



<p>Department of Electrical and Electronic Engineering EEE 302 MICROPROCESSORS & INTERFACING</p>

EXPERIMENT NO: 02

Part-1 Experimental Study of Input/ Output functions in Assembly Language

1.1 OBJECTIVE

In this experiment, the assembly language of Intel 8086 microprocessor will be observed where students will learn to achieve input & output in emu 8086 and get the outputs.

1.2 Pre- lab Preparation

- Read the experiment thoroughly and make a real effort to answer the questions of pre-lab.
- Review 8086 System commands from Microprocessor internal lab before you coming to the lab.

1.3 Equipment

- Personal Computer
- Emu 8086 Software

1.4 Theoretical background:

System Commands

Table 1 lists the most of the system commands used in the MTS-8088. The 8086 System has two command groups. These are the system commands and the I/O Driver commands. The system commands provide ways to utilize the system's resources, and the I/O Driver Commands are used to control I/O devices.

Command

Memory Management Commands

D	Display the contents of Memory
C	Compare the contents of Memory
E	Edit/Modify the memory contents
F	Fill memory
M	Move the contents of memory

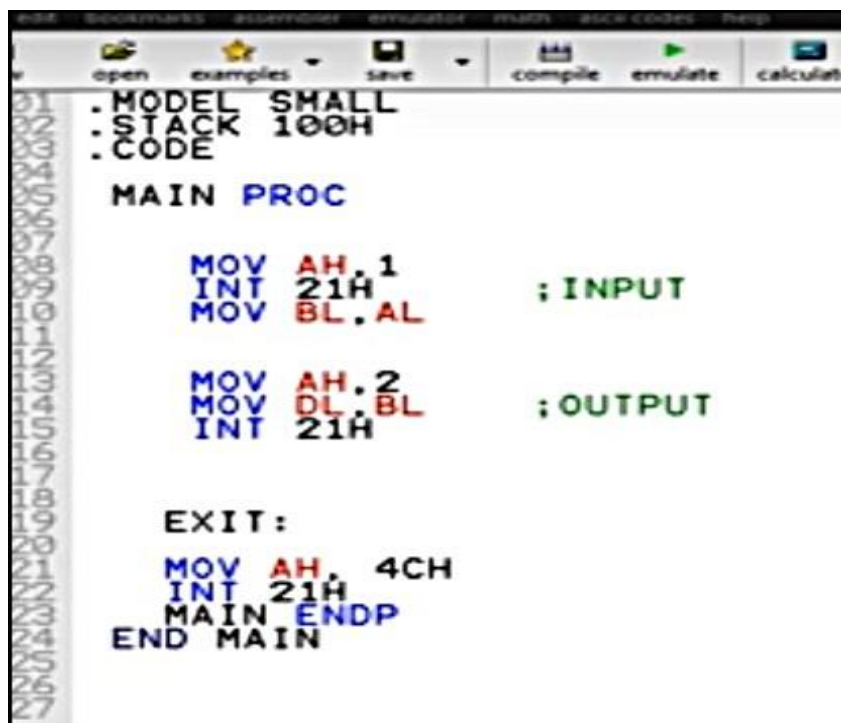
Assembler Commands

A	Command A is used to write an assembly language program.
I	Insert instruction in the program
U	Disassemble the assembly language instructions into machine code
Program Control Commands	
G	Executing Programs
R	Display / modify the contents of registers.
T	Trace the program execution
Numerical Operation/Conversion	
B	Convert a decimal number into binary
H	Sum or difference of two hexadecimal numbers
J	Convert a decimal number in Hexadecimal
S	Convert a hexadecimal number into decimal
V	Convert a binary number into decimal

1.5 Procedure

❖ Part-1 : Input from keyboard and output on screen

- Go to Start menu and run emu 8086 software.
- At first, you need to declare the library function.
- Then, you need to declare the main function like the following figure 1. Then, write down the code.



```

01  .MODEL SMALL
02  .STACK 100H
03  .CODE
04
05  MAIN PROC
06
07
08      MOV AH, 1
09      INT 21H          ; INPUT
10      MOV BL, AL
11
12
13      MOV AH, 2
14      MOV DL, BL      ; OUTPUT
15      INT 21H
16
17
18
19  EXIT:
20
21      MOV AH, 4CH
22      INT 21H
23      MAIN ENDP
24  END MAIN
25
26
27

```

Figure: 01

- As you can see, the 1st part of the program the code consist of input properties and 2nd part consist of output properties.

- After you emulate the code, you need to press the RUN button, where you will see an input and a copy of it as an output. But consecutively. To omit this, you need to add the following properties of NEW_LINE, as shown in the figure 02.

```

01 .MODEL SMALL
02 .STACK 100H
03 .CODE
04
05 MAIN PROC
06
07
08     MOV AH,1
09     INT 21H           ; INPUT
10     MOV BL,AL
11
12
13
14     MOV AH,2
15     MOV DL,0AH
16     INT 21H           ; NEWLINE
17     MOV DL,0DH
18     INT 21H
19
20
21     MOV AH,2
22     MOV DL,BL
23     INT 21H           ; OUTPUT
24
25
26
27     MOV AH,4CH
28     INT 21H
29     MAIN ENDP
30 END MAIN
  
```

Figure: 02

- After you emulate the code, you need to press the RUN button, where you will see an input and a copy of it as an output but in the next line like figure 03.

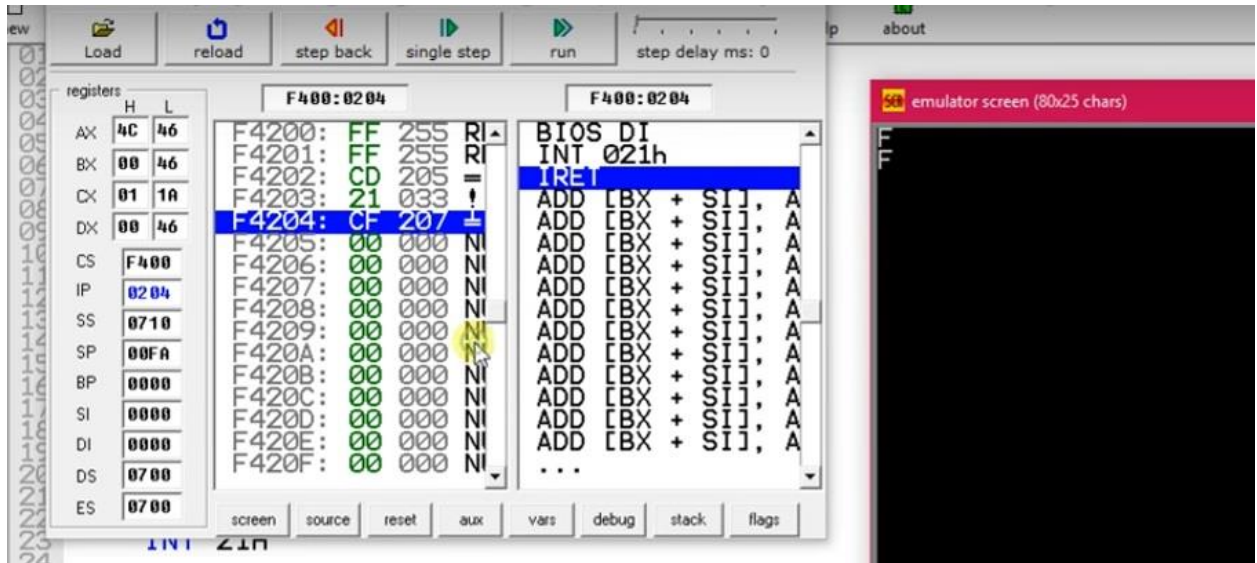
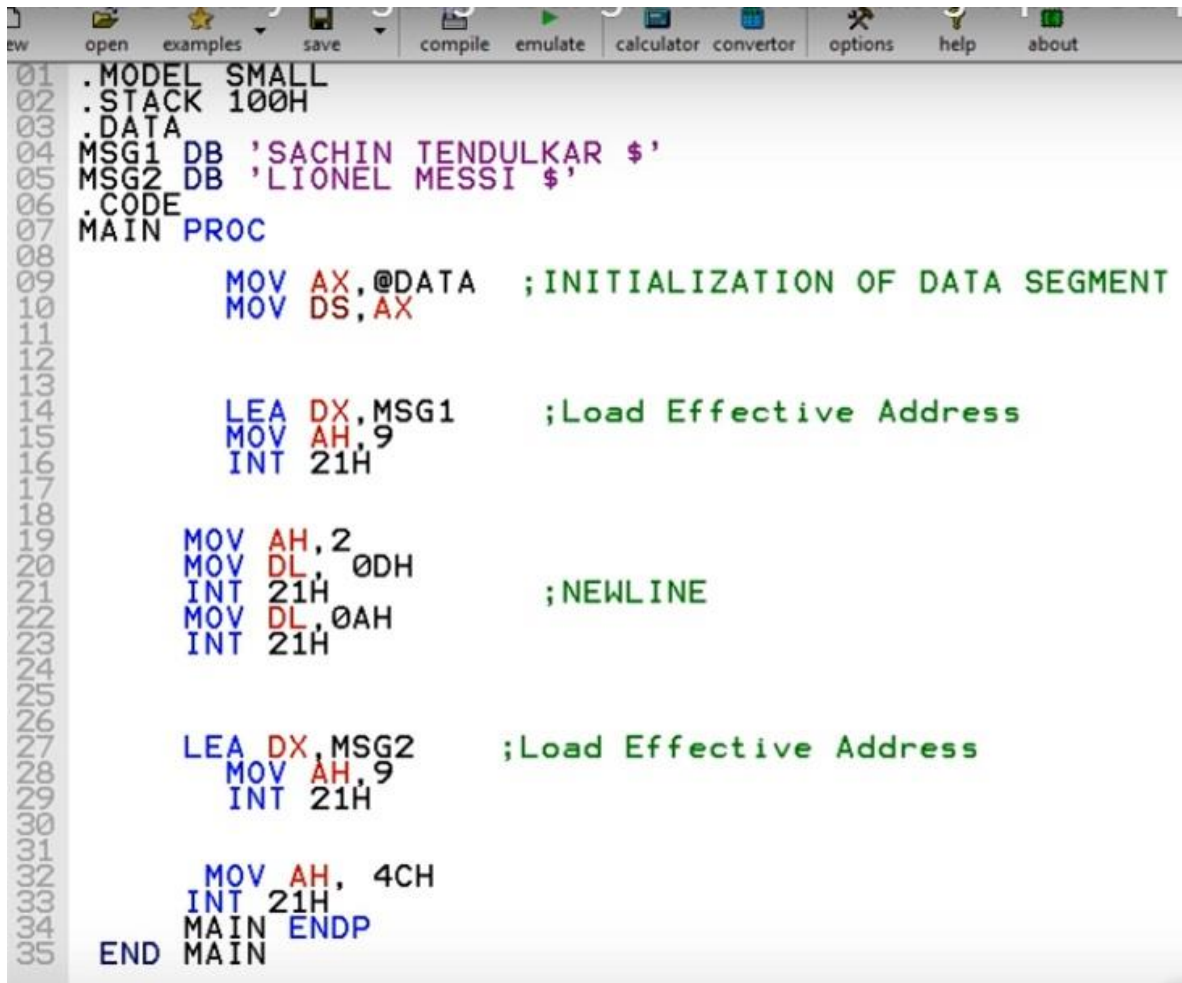


Figure: 03

❖ PART -2: String Input output

- Please follow what we have done in part 1. Here, you need to get a string output. As you can see in figure 04, we have used DX instead of DL, since we are using a string or DB i.e. double byte. The 2nd paragraph consists of print function which starts from load effective address command. Also, in part -1 we have used AH, 2 since it was only a integer value. But in this case, we have used MOV AH, 9; since it is a string.
- \$ = END OF STRING



```
01 .MODEL SMALL
02 .STACK 100H
03 .DATA
04 MSG1 DB 'SACHIN TENDULKAR $'
05 MSG2 DB 'LIONEL MESSI $'
06 .CODE
07 MAIN PROC
08
09     MOV AX,@DATA ;INITIALIZATION OF DATA SEGMENT
10     MOV DS,AX
11
12
13
14     LEA DX,MSG1 ;Load Effective Address
15     MOV AH,9
16     INT 21H
17
18
19     MOV AH,2
20     MOV DL,0DH
21     INT 21H ;NEWLINE
22     MOV DL,0AH
23     INT 21H
24
25
26
27     LEA DX,MSG2 ;Load Effective Address
28     MOV AH,9
29     INT 21H
30
31
32     MOV AH,4CH
33     INT 21H
34 MAIN ENDP
35 END MAIN
```

Figure: 04

- After you emulate the code, you need to press the RUN button. You will see an output like figure 05.

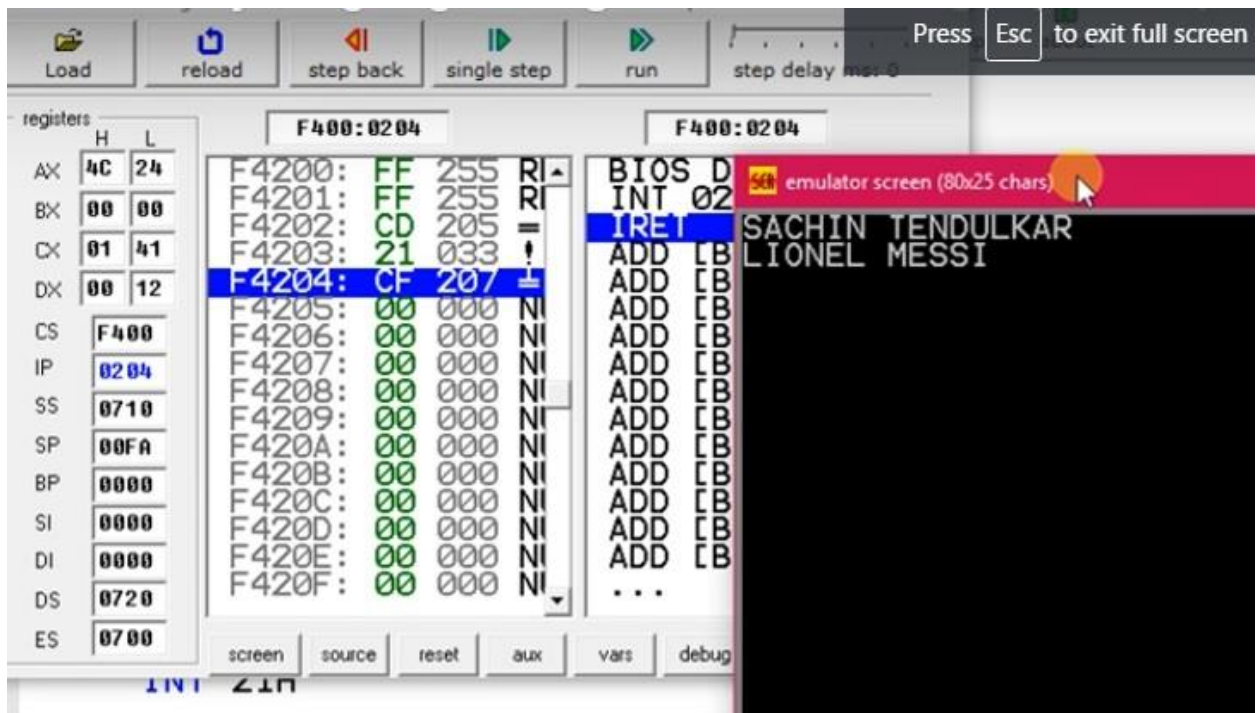


Figure: 05

- You need to remember that every time we are using the following commands

```

30
31
32      MOV AH, 4CH
33      INT 21H
34      MAIN ENDP
35  END MAIN

```

as an end function. You can get output, without writing these lines. But it is a good practice, so that is why we have added in 2nd part as well.

1.6 Post Lab Work:

- Take the screenshots, after executing each command.
- Randomly chose any two strings for getting a different set of outputs for part-2.
- Discuss all the steps in your own words, preferably with screenshots of each step.

Part-2 Experimental study of assembly language for finding out the greatest number among three decimal integer numbers.

2.1 OBJECTIVE

In this part of experiment, the loop instructions of assembly language of Intel 8086 microprocessor will be observed where students will learn to achieve input & output in emu 8086 using loop instructions and will find out the greatest number among three decimal numbers.

2.2 Procedure

❖ Part-1 : Input from keyboard and output on screen

- Go to Start menu and run emu 8086 software.
- At first, you need to declare the library function.
- Then, you need to declare the main function like the following.

.MODEL SMALL

.STACK 100H

.CODE

MAIN PROC

MOV AH, 1

INT 21H

MOV BL, AL

INT 21H

MOV BH, AL

INT 21H

MOV CL, AL

CMP BL, BH

JGE L2 ; // jump if greater than or equal to

L1:

CMP BH, CL

JGE LL1,

MOV AH, 2

MOV DL, CL

INT 21H

JMP EXIT

```

LL1:
MOV AH, 2
MOV DL, CL
INT 21H

JMP EXIT

L2:
CMP BL, CL
JGE LL2
MOV AH, 2
MOV DL, CL
INT 21H
JMP EXIT

LL2:
MOV AH, 2
MOV DL, BL
INT 21H

EXIT:
MOV AH, 4CH
INT 21H

MAIN ENDP
END MAIN

```

- After you emulate the code, you need to press the RUN button, where you can input any three decimal integer numbers. At the end, you will see the greatest number is printed after computation. Save the screenshot in your computer.

6
7
8
8

2.3 Post Lab Work:

- Take the screenshots, after executing each command.
- Randomly chose any three numbers for getting a different set of outputs.
- Discuss all the steps in your own words, preferably with screenshots of each step.