



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

## **Computer Science & Engineering**

CSE2006

Microprocessor and Interfacing

### **LAB ASSIGNMENT 3**

Submitted to **Prof. SANJAY R**

**TOPIC: ASSEMBLY LANGUAGE PROGRAMMING**

NAME: PUNIT MIDDHA

REG.NO: 19BCE2060

SLOT: L43+L44

DATE: 21/10/2021

➤ Task - 1

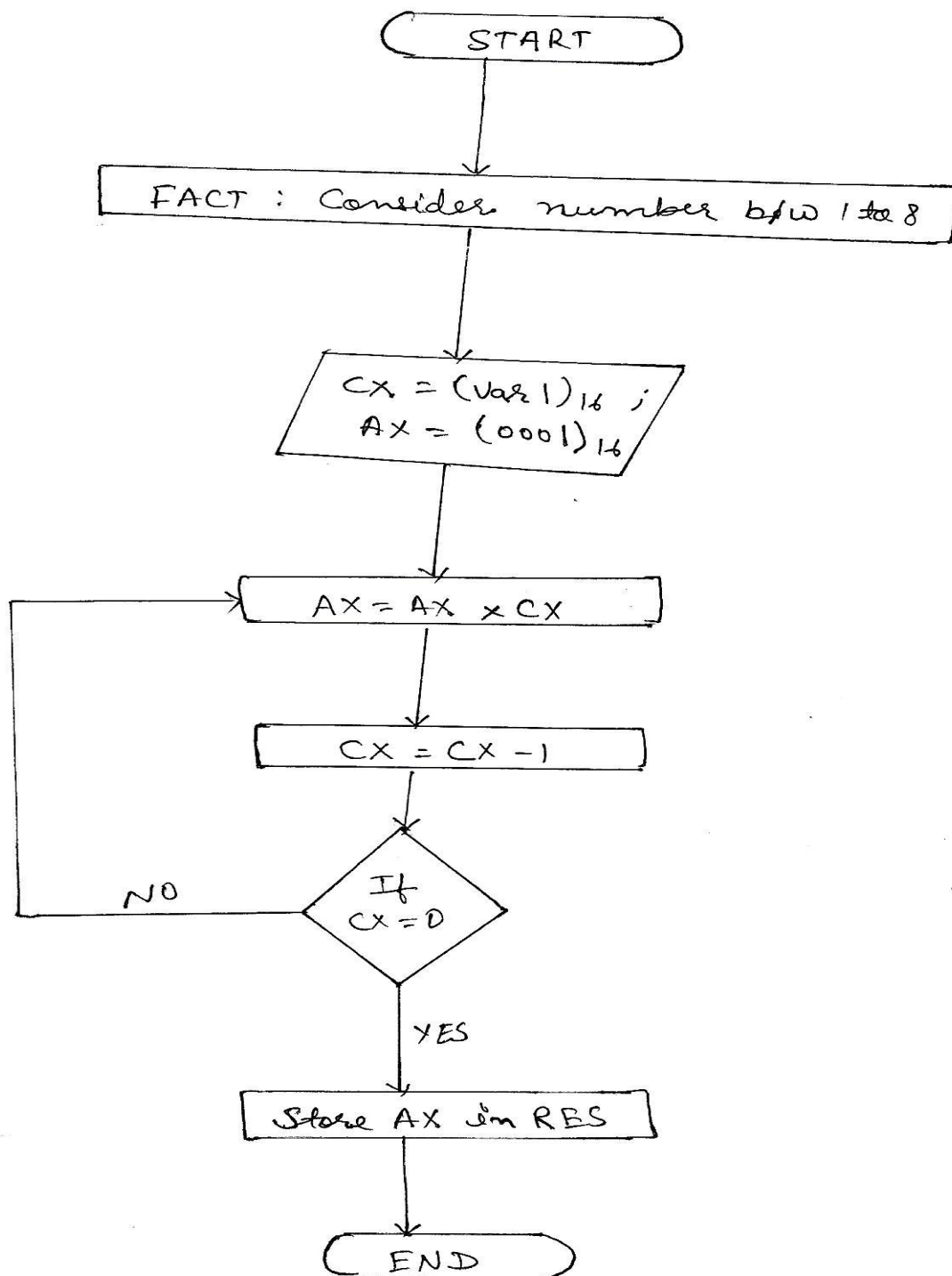
1. Write the ALP to find the factorial of the given number.

Aim:

To find the factorial of the given number.

Handwritten Flow Chart:

4.



### Handwritten Program:

1.

```
; Name: PUNIT MIDDHA
; Regno: 19BCE2060

ASSUME CS:CODE, DS:DATA

DATA SEGMENT
FACT DW 07H
RES DW ?
DATA ENDS

CODE SEGMENT
START: MOV AX, DATA
      MOV DS, AX
      MOV CX, FACT
      MOV AX, 01H

L01:   MUL CX
      LOOP L01

      MOV RES, AX

CODE ENDS
END START
```

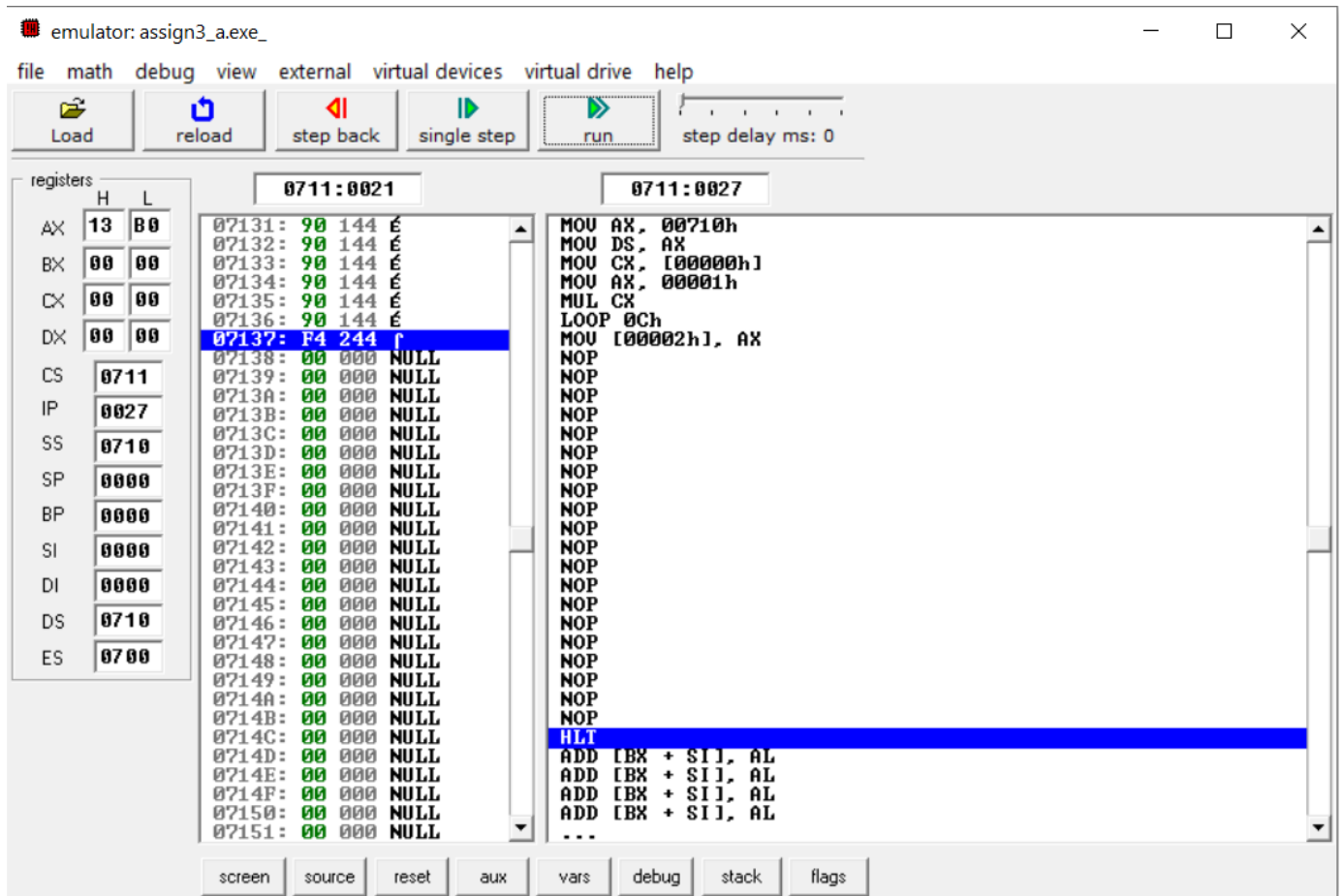
### Snapshots of typed program and Output:

edit C:\Users\Punit Middha\Desktop\Micro\assign3\_a.asm

file edit bookmarks assembler emulator math ascii codes help

new	open	examples	save	compile	emulate	calculator	converter	options	help	about
-----	------	----------	------	---------	---------	------------	-----------	---------	------	-------

```
01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt
03
04
05 ; Name: PUNIT MIDDHA
06 ; Regno: 19BCE2060
07
08 ASSUME CS:CODE, DS:DATA
09
10 DATA SEGMENT
11 FACT DW 07H
12 RES DW ?
13 DATA ENDS
14
15 CODE SEGMENT
16
17 START: MOV AX, DATA
18        MOV DS, AX
19        MOV CX, FACT
20        MOV AX, 01H
21 L01:   MUL CX
22        LOOP L01
23
24 MOV RES, AX
25
26 CODE ENDS
27 END START
```



### **Inference:**

Since, Factorial of any number is calculated as  $1*2*3*....*(n-1)*n$ . So, for the given number we have to run the loop from 1 to n and multiply in each iteration. We found the factorial of 07H, and the result, 13B0, which is stored in both AX and the RES variable.

## ➤ **Task – 2**

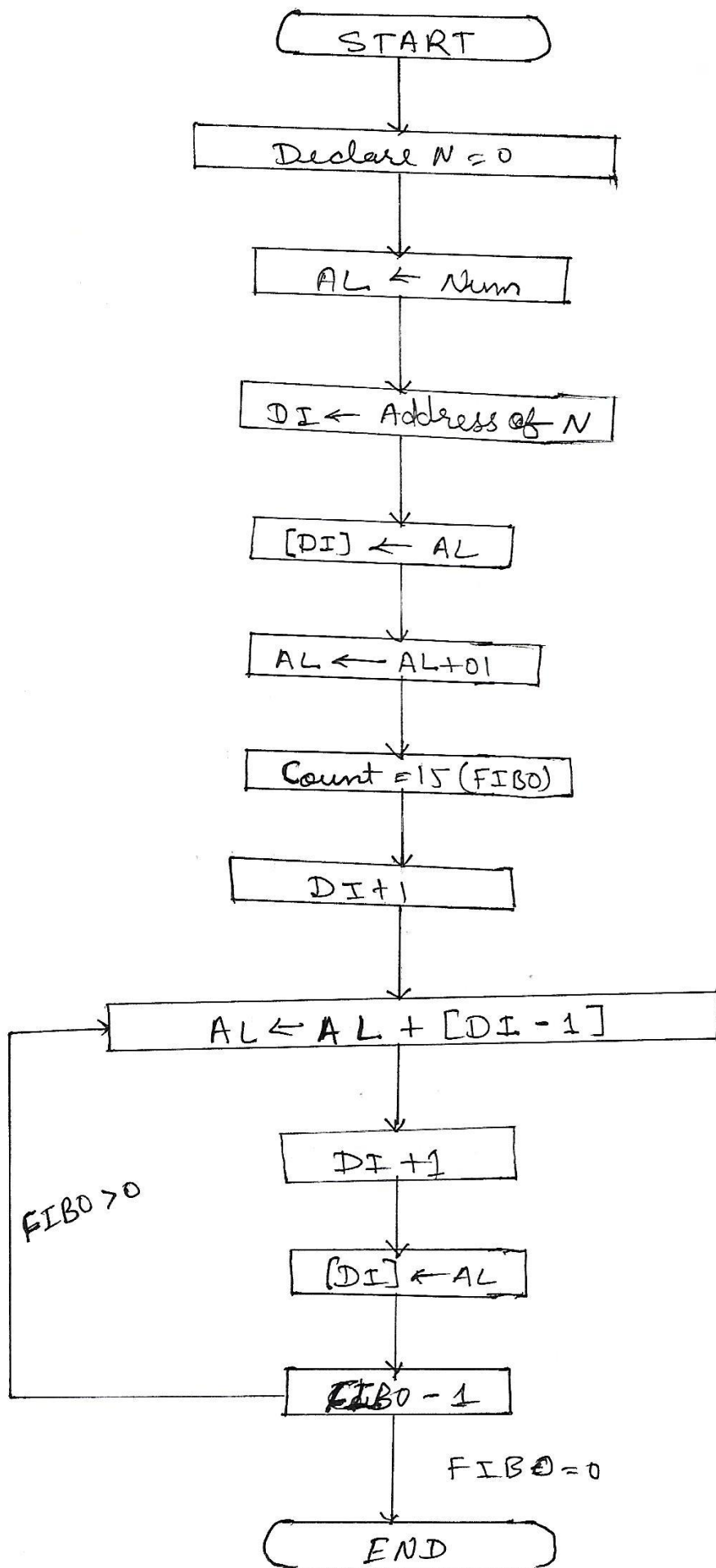
**2. Write the ALP for the Fibonacci series (up to 15 numbers)**

### **Aim:**

To find the Fibonacci series up to 15 numbers

### **Handwritten Flow Chart:**

2.



Handwritten Program:

Q.

; Name : PUNIT MIDDHA

; Regno : 19BCE2060

ASSUME CS:CODE, DS:DATA

DATA SEGMENT

FIBO DW 0EH

N EQU 00H

DATA ENDS

CODE SEGMENT

START: MOV AX, DATA

MOV DS, AX

MOV CX, FIBO

SUB CX, 0002H

MOV AL, N

MOV DI, OFFSET N

MOV [DI], AL

INC DI

ADD AL, 01H

MOV [DI], AL

LO4: MOV AL, [DI-1]

ADD AL, [DI]

INC DI

MOV [DI], AL

LOOP LO4

CODE ENDS

END START

## Snapshots of typed program and Output:

edit: C:\Users\Punit Middha\Desktop\Micro\assign3\_b.asm

file edit bookmarks assembler emulator math ascii codes help

```
new open examples save compile emulate calculator convertor options help about

01
02 ; You may customize this and other start-up templates;
03 ; The location of this template is c:\emu8086\inc\0_com_template.txt
04
05 ; Name: PUNIT MIDDHA
06 ; Regno: 19BCE2060
07
08 ASSUME CS:CODE,DS:DATA
09
10 DATA SEGMENT
11     FIBO DW 0EH
12     N EQU 00H
13     DATA ENDS
14
15 CODE SEGMENT
16     START:MOV AX,DATA
17            MOV DS,AX
18            MOV CX,FIBO
19            SUB CX,0002H
20            MOV AL,N
21            MOV DI,OFFSET N
22            MOV [DI],AL
23            INC DI
24            ADD AL,01H
25            MOV [DI],AL
26
27            L01:MOV AL,[DI-1]
28                ADD AL,[DI]
29                INC DI
30                MOV [DI],AL
31                LOOP L01
32            CODE ENDS
33 END START
```

emulator: assign3\_b.exe\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers		0710:0000		0711:0000	
H	L				
AX	07 E9	07100: 00 000 NULL		JMP 00010h	
BX	00 00	07101: 01 001 0		MOV AX, 00710h	
CX	00 00	07102: 01 001 0		MOV DS, AX	
DX	00 00	07103: 02 002 0		MOV CX, [00000h]	
CS	0711	07104: 03 003 0		SUB CX, 02h	
IP	0036	07105: 05 005 0		MOV AL, 00h	
SS	0710	07106: 08 008 BACK		MOV DI, 00000h	
SP	0000	07107: 0D 013 CRET		MOV [DI], AL	
BP	0000	07108: 15 021 5		INC DI	
SI	0000	07109: 22 034 "		ADD AL, 01h	
DI	0000	0710A: 37 055 7		MOV [DI], AL	
DS	0710	0710B: 59 089 Y		MOV AL, [DI] - 01h	
ES	0700	0710C: 90 144 E		ADD AL, [DI]	
		0710D: E9 233 0		INC DI	
		0710E: 00 000 NULL		MOV [DI], AL	
		0710F: 00 000 NULL		LOOP 028h	
		07110: B8 184 1		NOP	
		07111: 10 016 1		NOP	
		07112: 07 007 BEEP		NOP	
		07113: 8E 142 8		NOP	
		07114: D8 216 1		NOP	
		07115: 8B 139 i		NOP	
		07116: 0E 014 j		NOP	
		07117: 00 000 NULL		NOP	
		07118: 00 000 NULL		NOP	
		07119: 83 131 a		NOP	
		0711A: E9 233 0		NOP	
		0711B: 02 002 0		NOP	
		0711C: B0 176 8		NOP	
		0711D: 00 000 NULL		NOP	
		0711E: BF 191 1		NOP	
		0711F: 00 000 NULL		NOP	
		07120: 00 000 NULL		...	

screen source reset aux vars debug stack flags

**Inference:**

Each number in the Fibonacci sequence is the sum of the two preceding ones. The series' first two numbers are 0 and 1. Addresses ranging from 0710:0000 to 0710:000D contain the series of up to 14 numbers.