VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Computer Science & Engineering

CSE2006

Microprocessor and Interfacing

LAB ASSIGNMENT 2

Submitted to **Prof. SANJAY R**

TOPIC: ASSEMBLY LANGUAGE PROGRAMMING

NAME: PUNIT MIDDHA

REG.NO: 19BCE2060

SLOT: L43+L44

DATE: 24/09/2021

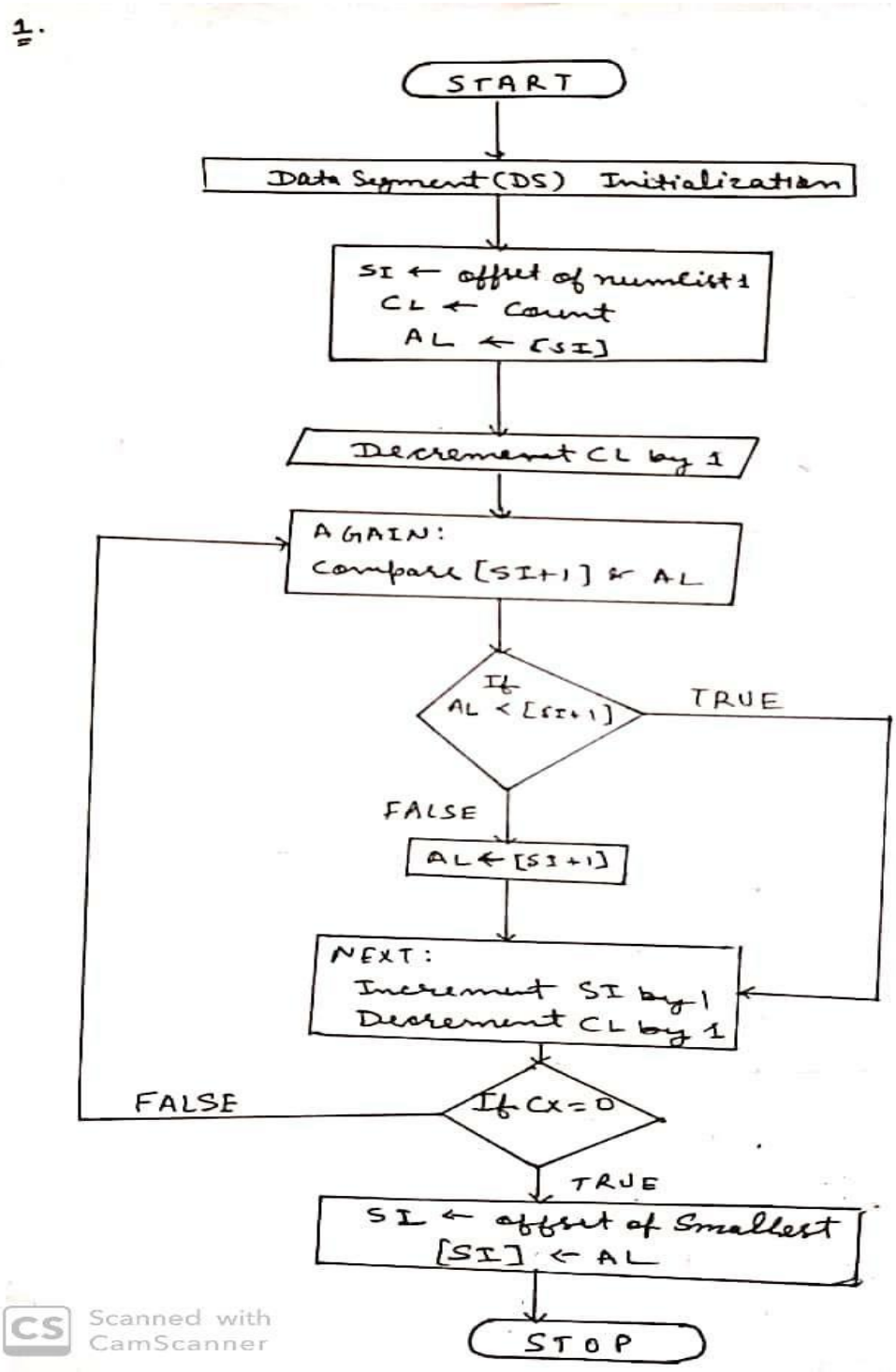
➢ *Task – 1*

1. **Write the ALP to find out the smallest number from a given unordered array of 8-bit numbers. Assume the array of numbers as AB, DF, FF,1E,5C, D1.**

Aim:

To find the smallest number from a given unordered array of 8-bit numbers i.e., AB, DF, FF, 1E, 5C, D1.

Handwritten Flow Chart:



Handwritten Program:

1.

```
ASSUME CS:CODE , DS:DATA
DATA SEGMENT
 NUMLIST1 DB  0ABH , 0DFH, 0FFH , 1EH, 5CH , 0D1H
 COUNT EQU 6D
 SMALLEST DW 01H DUP(?)
 DATA ENDS

 CODE SEGMENT
 START:    MOV  AX , DATA
           MOV  DS , AX
           MOV  SI , OFFSET NUMLIST1
           MOV  CL , COUNT
           MOV  AL , [SI]
           DEC  CL
 AGAIN:    CMP  AL , [SI+1]
           JC   NEXT
           MOV  AL , [SI+1]
 NEXT:     INC  SI
           DEC  CL
           JNZ  AGAIN
           MOV  SI , OFFSET SMALLEST
           MOV  [SI] , AL
           HLT
 CODE ENDS
 END START
```

ret

Snapshots of typed program and Output:

edit: C:\Users\Punit Middha\Desktop\Micro\assign2_a.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  NUMLIST1 DB 0ABH,0DFH,0FFH,1EH,5CH,0D1H
12  COUNT EQU 6D
13  SMALLEST DW 01H DUP(?)
14  DATA ENDS
15
16  CODE SEGMENT
17
18  START: MOV AX,DATA
19         MOV DS,AX
20         MOV SI,OFFSET NUMLIST1
21         MOV CL,COUNT
22         MOV AL,[SI]
23         DEC CL
24
25  AGAIN: CMP AL,[SI+1]
26         JC NEXT
27         MOV AL,[SI+1]
28
29  NEXT:  INC SI
30         DEC CL
31         JNZ AGAIN
32         MOV SI,OFFSET SMALLEST
33         MOV [SI],AL
34         HLT
35
36  CODE ENDS
37  END START
38
39  ret
```

emulator: assign2_a.exe_

file math debug view external virtual devices virtual drive help

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

registers

	H	L
AX	07	1E
BX	00	00
CX	00	00
DX	00	00

CS	0711
IP	0020
SS	0710
SP	0000
BP	0000
SI	0006
DI	0000
DS	0710
ES	0700

0710:0006

```
07106: 1E 030 ▲
07107: 00 000 NULL
07108: 00 000 NULL
07109: 00 000 NULL
0710A: 00 000 NULL
0710B: 00 000 NULL
0710C: 00 000 NULL
0710D: 00 000 NULL
0710E: 00 000 NULL
0710F: 00 000 NULL
07110: B8 184 ╕
07111: 10 016 ►
07112: 07 007 BEEP
07113: 8E 142 Ä
07114: D8 216 ╪
07115: BE 190 ╛
07116: 00 000 NULL
07117: 00 000 NULL
07118: B1 177 ▒
07119: 06 006 ♠
0711A: 8A 138 è
0711B: 04 004 ♦
```

0710:0006

```
PUSH DS
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI] + 00710h, BH
MOV DS, AX
MOV SI, 00000h
MOV CL, 06h
MOV AL, [SI]
DEC CL
CMP AL, [SI] + 01h
JB 026h
MOV AL, [SI] + 01h
INC SI
DEC CL
JNE 01Eh
MOV SI, 00006h
MOV [SI], AL
HLT
NOP
...
```

screen source reset aux vars debug stack flags

Inference:

The result obtained is 1E as seen at address 0710:0006. So, in the given array of numbers - AB, DF, FF,1E,5C, D1 – 1E is the smallest number.

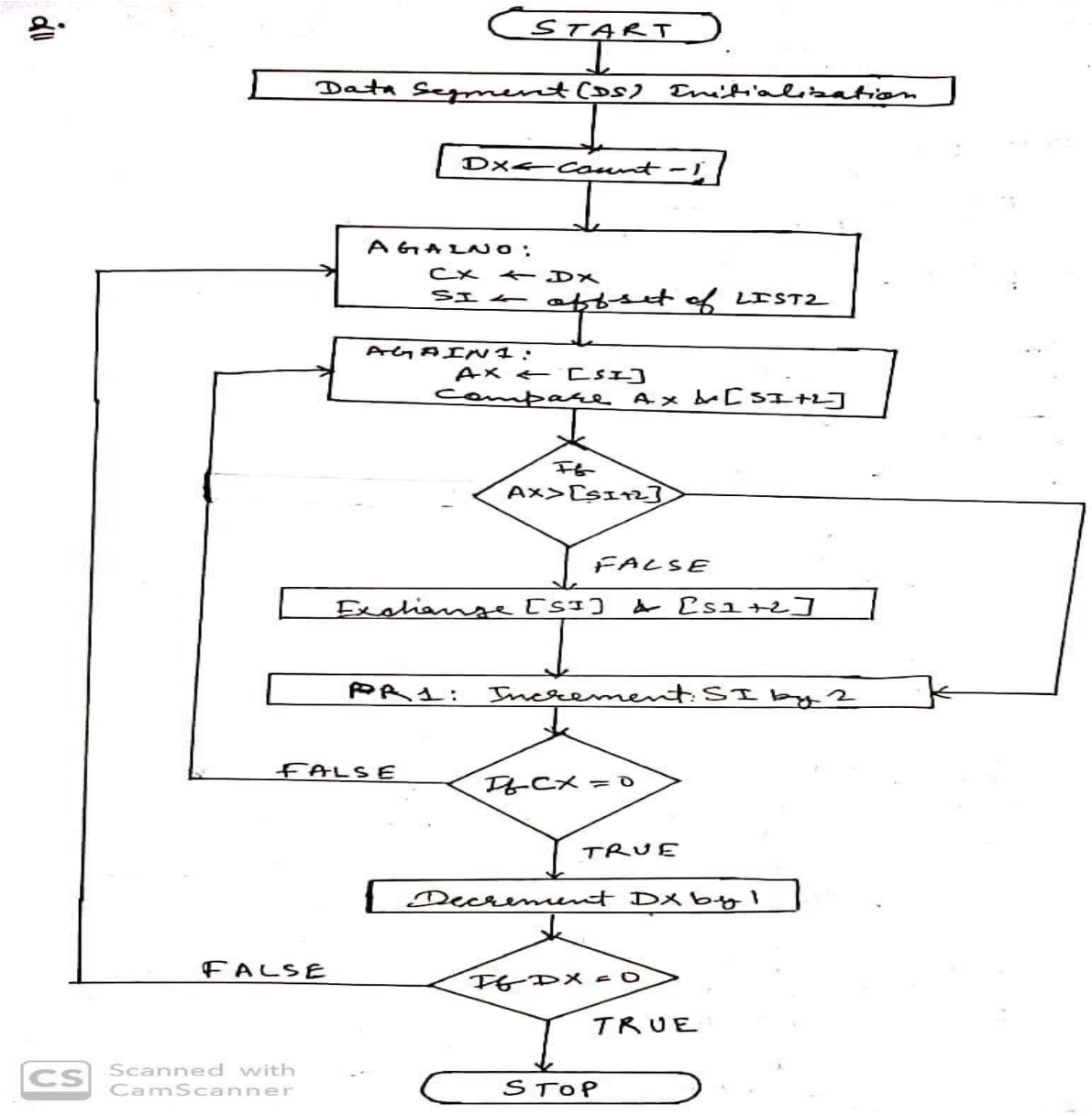
➢ *Task – 2*

2. Write the ALP to arrange a given series of hexadecimal bytes in descending order. Assume the array of numbers as: first 2 digits of your rollno, 3rd and 4th digits of your rollno, 6th and 7th digits of your rollno, last 2 digits of your rollno.

Aim:

To arrange a given series of hexadecimal numbers i.e., (19)H, (BC)H, (20)H, (60)H in descending order.

Handwritten Flow Chart:



Handwritten Program:

2.

```
ASSUME  CS:CODE , DS:DATA

DATA SEGMENT
LIST2 DW 19H, 0BCH, 20H, 60H
COUNT EQU 04
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
        MOV DS, AX
        MOV DX, COUNT-1
AGAINO: MOV CX, Dx
        MOV SI, OFFSET LIST2
AGAIN1: MOV AX, [SI]
        CMP AX, [SI+2]
        JNC PR1
        XCHG [SI+2], AX
        XCHG [SI], Ax
PR1:    ADD SI, 02
        LOOP AGAIN1
        DEC Dx
        JNC AGAINO
        MOV AH, 4CH
        INT 21H

CODE ENDS
END START
```

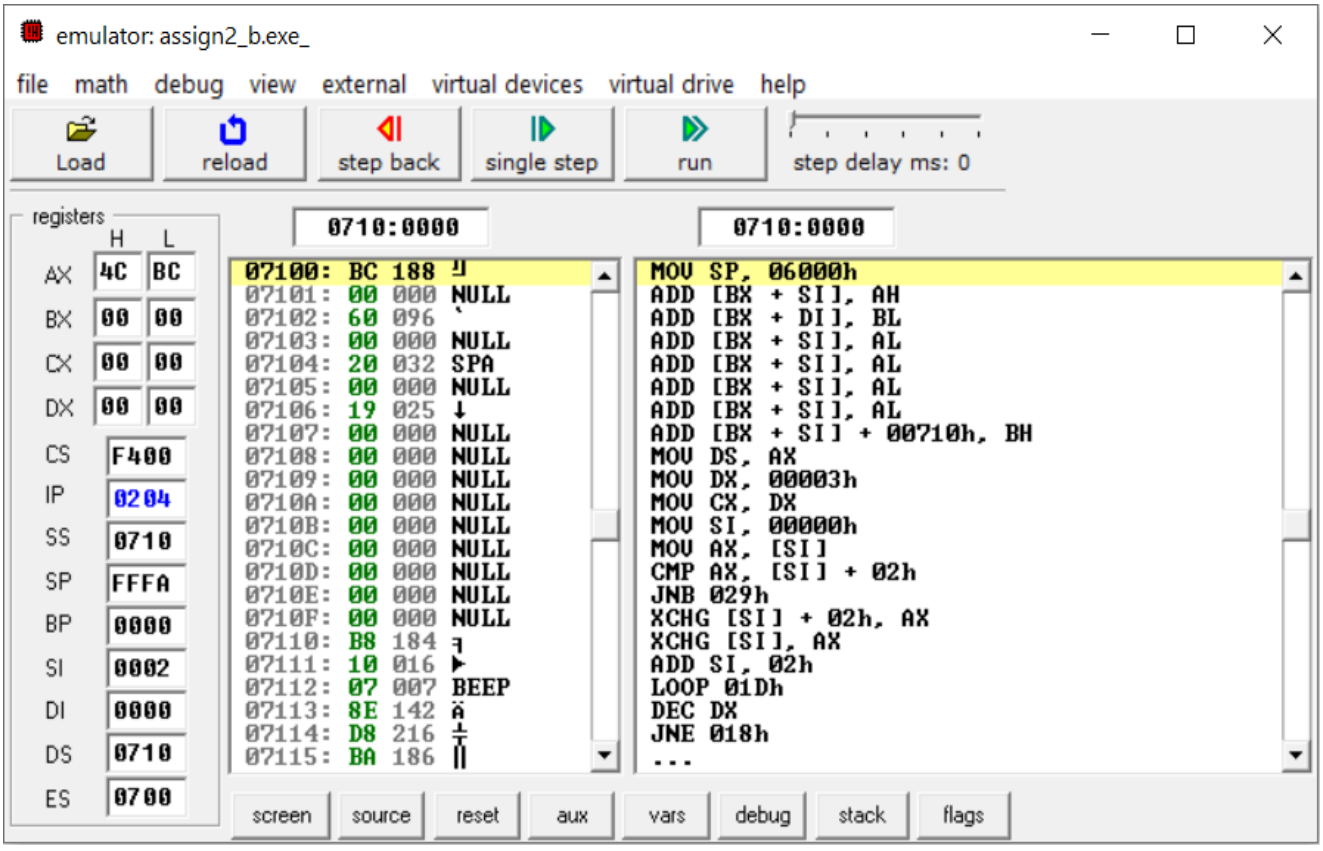
Snapshots of typed program and Output:

edit: C:\Users\Punit Middha\Desktop\Micro\assign2_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 LIST2 DW 19H,0BCH,20H,60H
12 COUNT EQU 04
13 DATA ENDS
14
15 CODE SEGMENT
16
17 START: MOV AX,DATA
18        MOV DS,AX
19        MOV DX,COUNT-1
20
21 AGAINO: MOV CX,DX
22         MOV SI,OFFSET LIST2
23
24 AGAIN1: MOV AX,[SI]
25         CMP AX,[SI+2]
26         JNC PR1
27         XCHG [SI+2],AX
28         XCHG [SI], AX
29
30 PR1: ADD SI,02
31      LOOP AGAIN1
32      DEC DX
33      JNZ AGAINO
34      MOV AH,4CH
35      INT 21H
36
37 CODE ENDS
38 END START
39
40 ret
```



Inference:

Array: 19H, BCH, 20H, 60H.
After executing order: BCH, 60H, 20H, 19H.

The resultant array after arranging it in descending order is stored at the following addresses:

07100 – BC, 07102 – 60, 07104 – 20, 07106 – 19.