EXPERIMENT NO : ~~08~~ 07

DATE: 6/5/22

ROLL NO : 5139

AIM: To perform string instruction of 8086

THEORY:

STRING INSTRUCTIONS:

A string instruction is in 8086 is a series of same type of data items in sequential memory locations. Following are the string instructions of 8086:

1. MOVS
2. LODS
3. STOS
4. CMPS
5. SCAS

KEY POINTS FOR INSTRUCTIONS:

1. SI - Index offset for source string
2. DI - Index offset for destination string
3. CX - Default counter register
4. AL/AX - Register which required to store data for string instructions
5. CLD - Clear direction flag (DF=0)
6. STD - Set direction flag (DF=1)

   DF=0 : Auto increment SI, DI pointers

   DF=1 : Auto decrement SI, DI pointers.

1. MOVS B/W/D:

This instruction is used to transfer content from source to destination.

Operation: ES: [DI] ← DS: [SI]

2. LODS B/W/D:

This instruction is used to load string byte into AL and string word into AX register. This instructions copies a byte or word from a string location pointed by SI into AL/AX register

Operation: AL ← DS: [SI]

3. STOS B/W/D:

This instruction is used to load string byte from AL and string ward from AX register. This instruction copies a byte or ward from a AL/AX register into string location pointed by DI in extra segment.

Operation: AL → ES: [DI]

4. SCAS B/W/D:

This instruction is used to compare a byte/word into AL/AX with a byte pointed by DI in ES

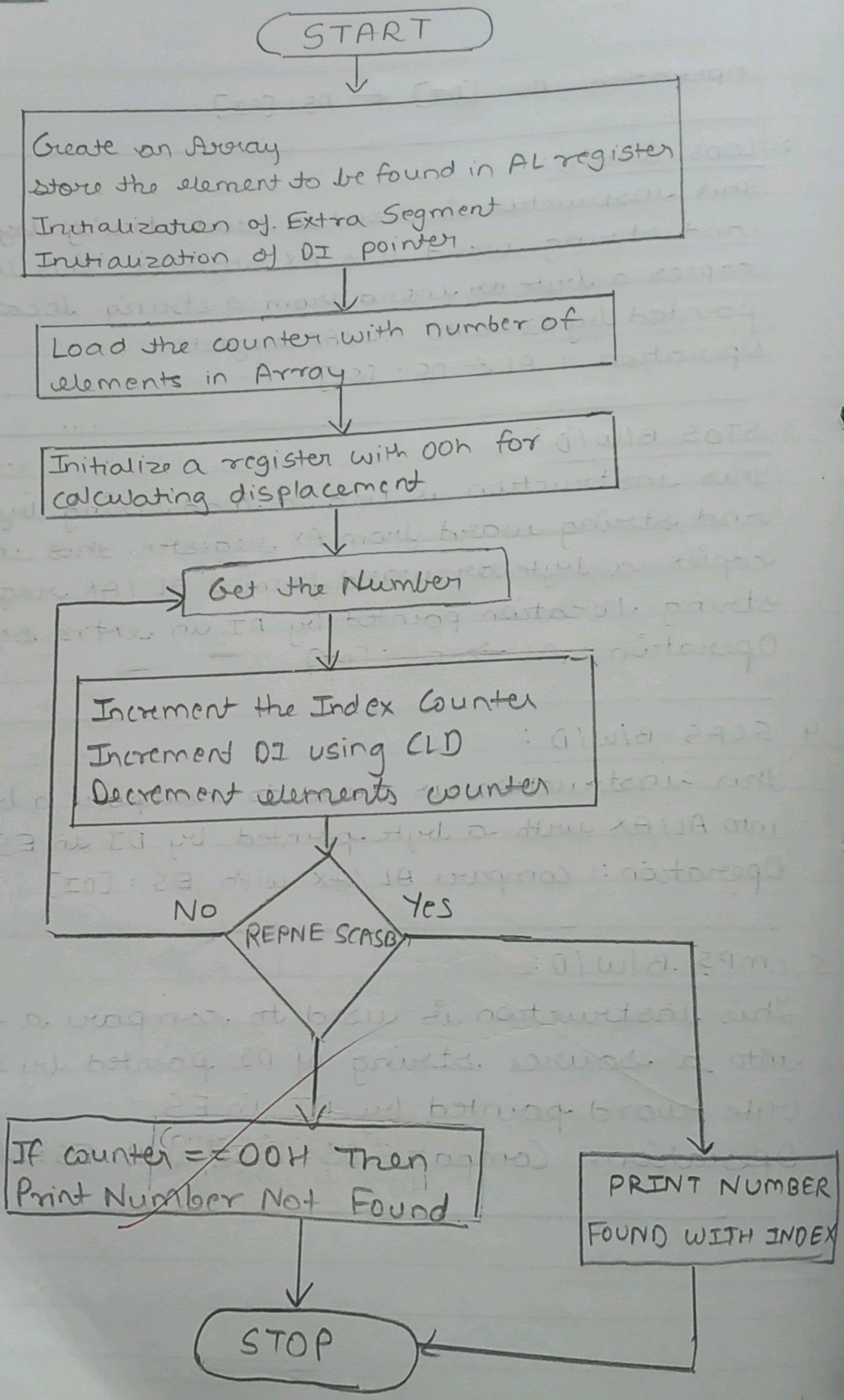
Operation: Compare AL/AX with ES: [DI]

5 CMPS B/W/D:

This instruction is used to compare a byte/word into a source string of DS pointed by SI with a byte/word pointed by DI in ES

Operation: Compare DS: [SI] with ES: [DI].

FLOWCHART


START
Create an Array
Store the element to be found in AL register
Initialization of Extra Segment
Initialization of DI pointer
Load the counter with number of elements in Array
Initialize a register with 00h for calculating displacement
Get the Number
Increment the Index Counter
Increment DI using CLD
Decrement elements counter
REPNE SCASB
No
Yes
If counter = 00H Then
Print Number Not Found
PRINT NUMBER
FOUND WITH INDEX
STOP

PROGRAM 1 :

- WAP to create an array of and find the displacement at which given data is present

Name of an Array : StrArr

Size of an Array : 07

Elements of an Array : 04, 03, 09, 05, 02, 01, 15

No and displacement to be search : 02H

ALGORITHM

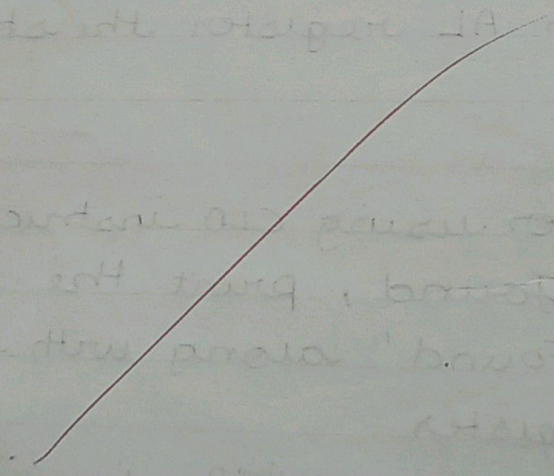
1) Create an array of 7 elements
2) Store the element to be found from the Array in AL register.
3) Initialization of extra segment
4) Initialization of DI pointer.
5) Load the counter with the number of elements in Array.
6) Initialize a register with 00h for calculating index of number present at the Array
7) Compare [di] with AL register the string Instruction REPNE SCASB
8) Decrement counter
9) Increment DI pointer using CLD instruction
10) If the element is found, print the Message "Number Found" along with its index
11) Increment index register.
12) If the counter register == 00H. Display "Number Not Found" Message
13) Stop the Program.

STEPS TO DISPLAY OUTPUT

```
C:\> tasm filename.asm
C:\> tlink filename.obj
c:\> filename
```

OUTPUT

```
NUMBER FOUND AT : 04
```



PROGRAM:

```
.model small
.data
strArr db 04,03,09,05,02,01,15
a db 02H
string db 'NUMBER FOUND AT : ',`$'
string1 db 'NUMBER NOT FOUND',`$'
.code
mov ax,@data
Mous ds,ax
mov es,ax
lea di,arr
mov al,a
mov cx,0007
mov bx,cx
cld
repne scasb
cmp cx,0000H
jz down3
sub bx,cx
dec bx
mov ax,bx
lea dx,string
mov ah,09h
int 21H
mov ch,02h
mov cl,04h
mov bh,al
i2: rol bh,cl
    mov dl,bh
    and dl,0fh
```

```
    cmp dl, 09h
    jbe i4
    add dl, 07h
i4: add dl, 30h
    mov ah, 02h
    int 21h
    dec ch
    jnz i2
    jmp f3
    down3:
    lea dx, string1
    mov ah, 09h
    int 21h
f3: mov ah, 4ch
    int 21h
    end
```

CONCLUSION

Hence, we implemented string instructions of 8086.

Roll no: 1315