Lab 4

Ex.No.4. Program to demonstrate string operation.

4.1 Introduction:

The purpose of this experiment is to find the number of character in a string

4.2 Hardware Requirement:

The 8086 Microprocessor kit, Power Supply.

4.3 Program Logic:

Addressing the string is done using the SI register, and the DX register is used to store the number of characters. End of the string is detected using FF. Hence each character is fetched from memory and is compared with FF. If the zero flag is set, then it denotes end of string, the count has been stored in DX, by incrementing it after each comparison.

4.4 Program:

Introduction of general purpose registers, indirect addressing, and loop instructions, compare instruction, increment instruction:
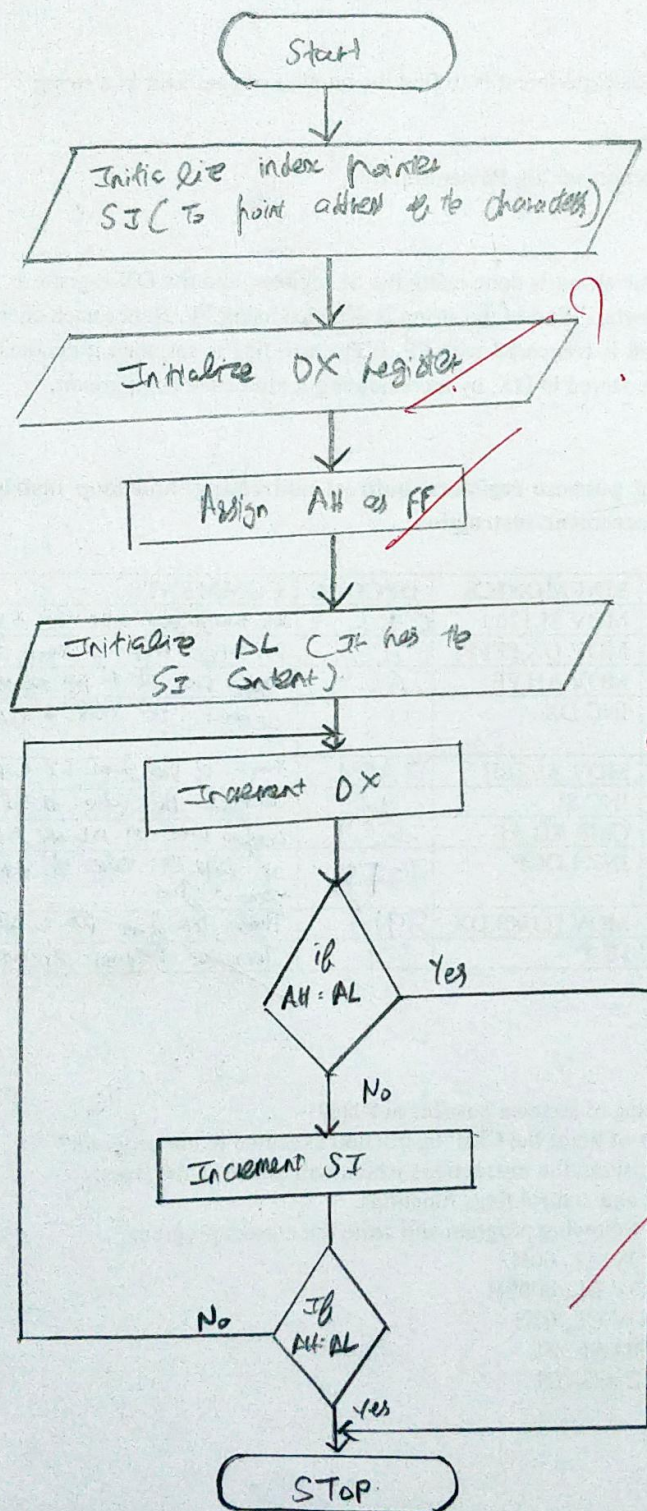
ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1200		MOV SI,1200	C7C6	Data transfer from address 1200 to register SI.
1004		MOV DX,FFFF	C7C2	Data transfer FFFF to register DX
1008		MOV AH,FF	C6C4	Transfer Data FF to AH register
100B	LOOP	INC DX	42	Increment the value of DX.
100C		MOV AL,[SI]	8A04	Transfer of Data from SI to AL.
100E		INC SI	46	Increment the value of SI
100F		CMP AH,AL	38C4	Compare Data in AL and AL.
1011		JNZ LOOP	75F8	It looks to value if non-zero is found.
1013		MOV [1100],DX	8916	Transfer Data from DX to address 1100
1017		HLT	F4	Terminate Program execution.

4.5 Pre-Lab Questions:

1. Does the overlapping of segment possible in 8086?
2. Count the number of times the CMP instruction executed in our program?
3. In our program, List out the instructions which will affect by the flags.
4. Write conditional and control flags functions.
5. Find errors in the following program and write the correct program?

```
         MOV AL, 00H
         MOV BL, 0005H
         MOV CL, 02H
AGAIN:   ADD AL, BL
         JNZ AGAIN
```

```
                    ┌─────────────┐
                    │    Start    │
                    └──────┬──────┘
                           │
                           ▼
         ╱────────────────────────────────────────╲
        ╱  Initialize index pointer                 ╲
       ╱   SI ( To point address of the characters)  ╲
      ╱──────────────────────────────────────────────╲
                           │
                           ▼
         ╱────────────────────────────────╲
        ╱   Initialize DX register          ╲
       ╱────────────────────────────────────╲
                           │
                           ▼
              ┌──────────────────────┐
              │   Assign AH as FF     │
              └──────────────────────┘
                           │
                           ▼
         ╱────────────────────────────╲
        ╱   Initialize DL (It has the   ╲
       ╱    SI Content)                  ╲
      ╱──────────────────────────────────╲
                           │
                           ▼
              ┌──────────────────────┐
              │    Increment DX       │
              └──────────┬───────────┘
                         │
                         ▼
                       ╱───╲
                      ╱ if  ╲        Yes
                     ╱ AH=AL ╲───────────────┐
                      ╲      ╱                │
                       ╲    ╱                 │
                        ╲──╱                  │
                         │ No                 │
                         ▼                    │
              ┌──────────────────────┐        │
              │    Increment SI       │        │
              └──────────┬───────────┘        │
                         │                    │
          No           ╱───╲                  │
     ┌───────────────╱ If   ╲                 │
     │               ╲ AH=AL ╱                │
     │                ╲     ╱                 │
     │                 ╲───╱                  │
     │                   │ Yes                │
     │                   ▼                    │
     │            ┌─────────────┐             │
     └──────────▶ │    STOP      │ ◀──────────┘
                  └─────────────┘
```

Redo

```
DEC CL
MOV DI, 1300H
MUV [DI], AL
HLT
```

4.6 Post-Lab Questions:

1. Write the program for Fibonacci series and execute in emulator software.
2. List the instructions that can be used to clear the accumulator or any registers?
3. Write the use of pointers and index registers?

INPUT ADDRESS	DATA
1200	12
1201	34
1202	56
1203	78
1204	AB
1205	FF

OUTPUT ADDRESS	DATA
1100	05

Experiment 4 - Program to Demonstrate String Operation

I   Pre-lab Questions:

1   Does the overlapping of segment possible in 8086?

Soln.   There is no restriction on the locations of these segments in the memory. These segments can be separate from each other or they can overlap. In the users program, there can be many segments but 8086 can deal with only four of them at any given time because it has only four segment registers.

2   Count the number of times CMP instruction executed in our program?

Soln.   CMP is executed 5 times in our program as it is looping till the end of the string.

3   In our program, list out the instructions which will effect by the flags.

Soln.   CMP, AH, AL, JNZ these are the instructions which will effect by the flags.

4   Write Conditional and Control flags functions.

Soln.   Control flags enable or disable certain operations of the Microprocessor. There are 3 Control flags in 8086 microprocessor.
  i) Directional flag - specifically used in string instructions.
  ii) Interrupt flag - for interrupts signal.
 iii) Trap flag - used for on-chip debugging.
  In 8086 there are 6 different flags which are set or reset after 8-bit or 16-bit operation.

5. Find errors in the following code and write the correct program,

Sol. In the given code, JNZ AGAIN should come after DEC CL otherwise it will become infinite loop until the value of AL becomes 00.

II Post-Lab Questions

1. Write the program for Fibonacci Series and execute in emulator Software.

Sol.
```
MOV AL, 00H
MOV SI, 50H
MOV [SI], AL
INC SI
ADD AL,01H
MOV [SI], AL
MOV CX, [60H]
SUB CX, 1H
Loop1:
MOV AL, [SI-1]
ADD AL, [SI]
INC SI
MOV [SI], AL
LOOP Loop1
HLT
```
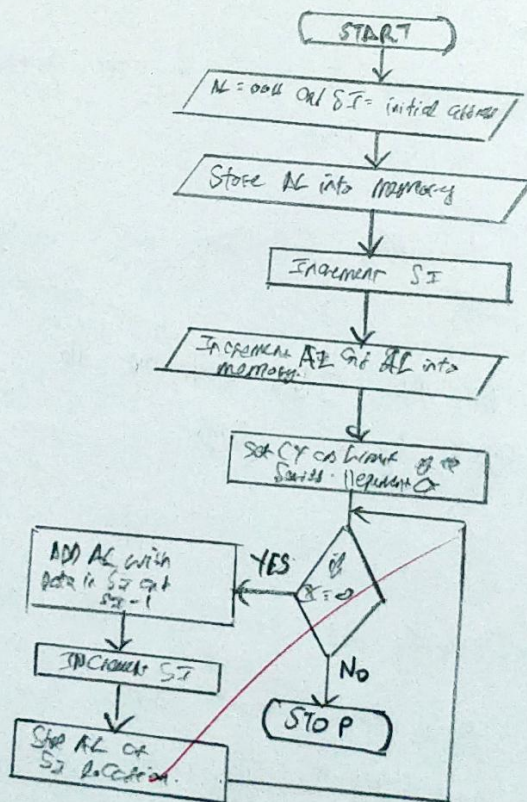

2. List the instructions that can be used to clear the accumulator (or) any register.

Sol. · SUB is the instruction that can be used to clear the accumulator (or) any register and it is a one-byte instruction. Sub is used for

arithmetic operations whereas another instruction XOR is logical and performs much faster. So, XOR is better for clearing and resetting accumulator cons any register

3 Write the use of pointers and index registers.

Ans Stack and base pointers are 2 pointer registers whereas source and destination index are index group of registers.