



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Microprocessor and Interfacing
CSE2006

Lab Assignment 5

Slot: L3+L4

Name: Kulvir Singh

Register Number: 19BCE2074

1)7 Segment Display using 8086

Aim :

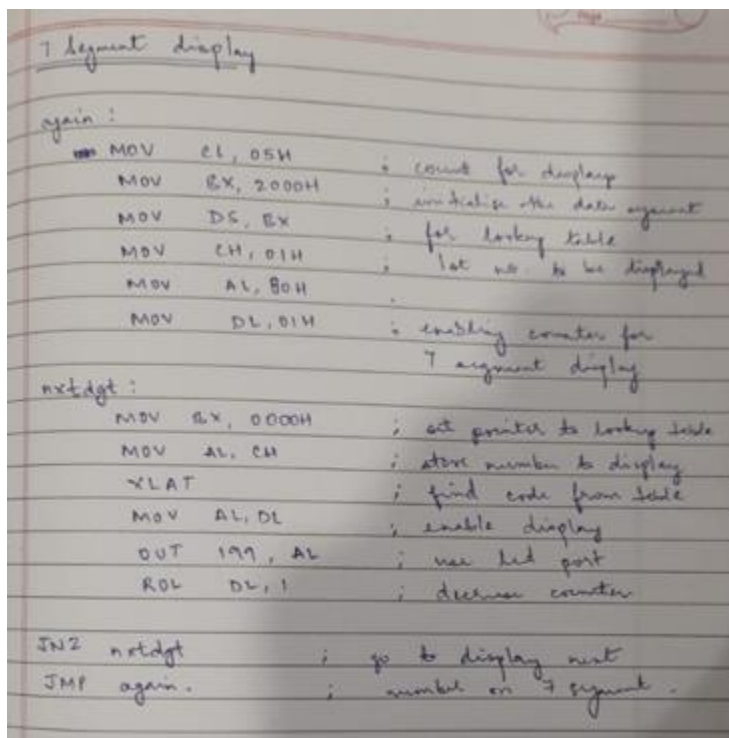
Write a program in 8086 Assembly Language to display numbers in 7 segment format in an LED display.

Requirements :

8086 EMU - An emulator to run the 8086 Assembly Language Code

Operating System - Any valid operating system that can execute the emulator

Handwritten Program :



The image shows a handwritten assembly program on lined paper. The title '7 segment display' is underlined. The program is divided into two main sections: 'again:' and 'nextdgt:'. The 'again:' section initializes registers: CL to 05H (commented as 'count for display'), EX to 2000H (commented as 'initialise the data segment'), DS to EX (commented as 'for looking table'), CH to 01H (commented as '1st no. to be displayed'), AL to 80H, and DL to 01H (commented as 'enabling counter for 7 segment display'). The 'nextdgt:' section moves the pointer to the look-up table (MOV BX, 0000H), moves the number to display (MOV AL, CH), performs a table lookup (XLAT), enables the display (MOV AL, DL), outputs to port 199 (OUT 199, AL), and decrements the counter (ROL DL, 1). A loop is formed by the JNZ nextdgt instruction, which jumps back to the 'nextdgt:' label if the zero flag is not set. The program ends with a JMP again instruction, which jumps back to the 'again:' label to display the next number.

```
7 segment display

again:
MOV CL, 05H      ; count for display
MOV EX, 2000H    ; initialise the data segment
MOV DS, EX       ; for looking table
MOV CH, 01H      ; 1st no. to be displayed
MOV AL, 80H
MOV DL, 01H      ; enabling counter for 7 segment display

nextdgt:
MOV BX, 0000H    ; set pointer to looking table
MOV AL, CH       ; store number to display
XLAT             ; find code from table
MOV AL, DL       ; enable display
OUT 199, AL      ; use led port
ROL DL, 1        ; decrease counter

JNZ nextdgt      ; go to display next
JMP again        ; number on 7 segment.
```

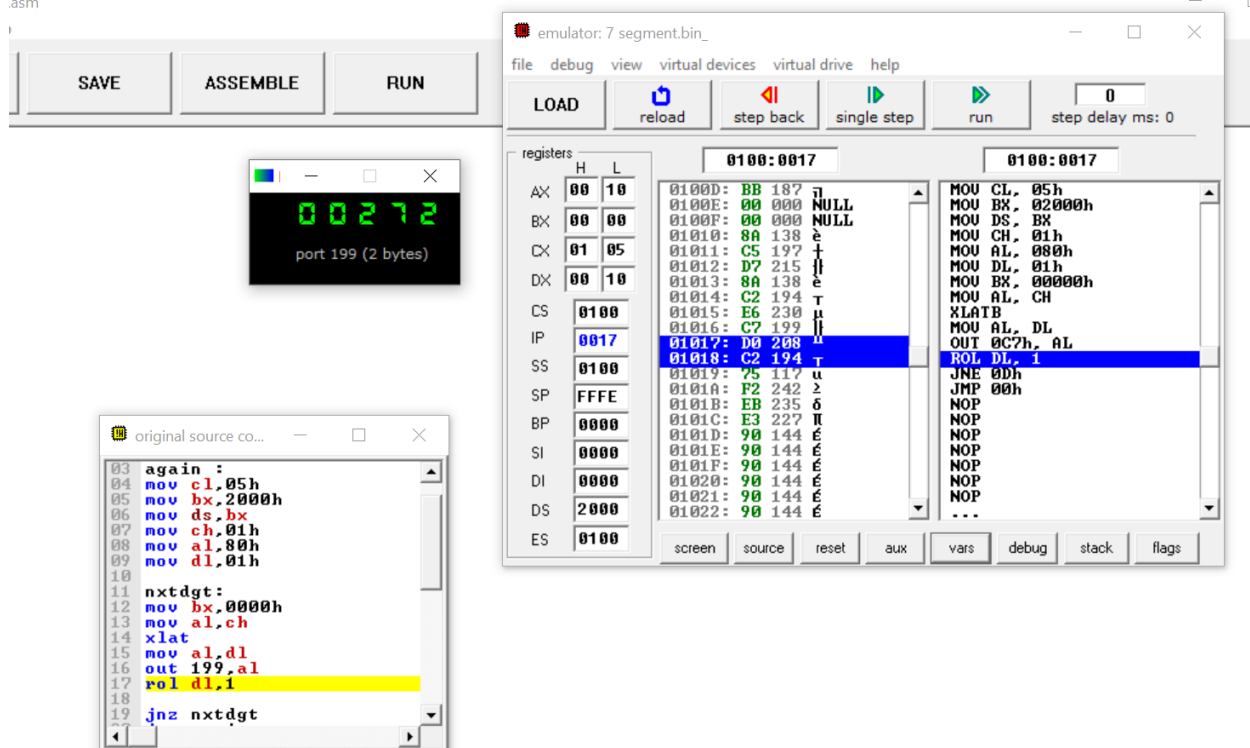
Screenshots :

```

01 ;19BCE2074
02 ;KULVIR SINGH
03 again :
04 mov cl,05h
05 mov bx,2000h
06 mov ds,bx
07 mov ch,01h
08 mov al,80h
09 mov dl,01h
10
11 nextdgt:
12 mov bx,0000h
13 mov al,ch
14 xlat
15 mov al,dl
16 out 199,al
17 rol dl,1
18
19 jnz nextdgt
20 jmp again

```

asm



Inference :

The program can successfully display the numbers in a seven segment format on the led display simulator.

2) Stepper Motor using 8086

Aim :

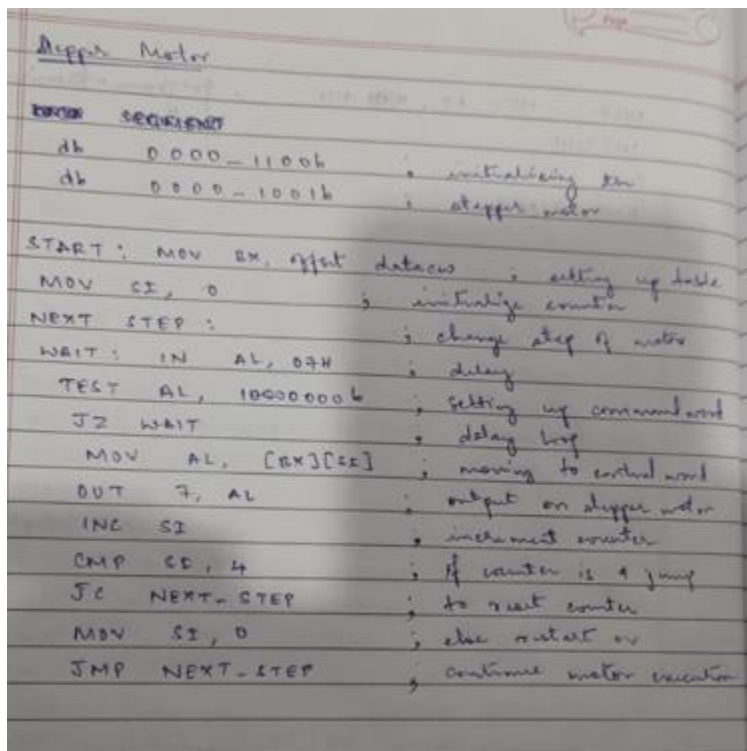
Write a program in 8086 Assembly Language to simulate a stepper motor.

Requirements :

8086 EMU - An emulator to run the 8086 Assembly Language Code

Operating System - Any valid operating system that can execute the emulator

Handwritten Program :



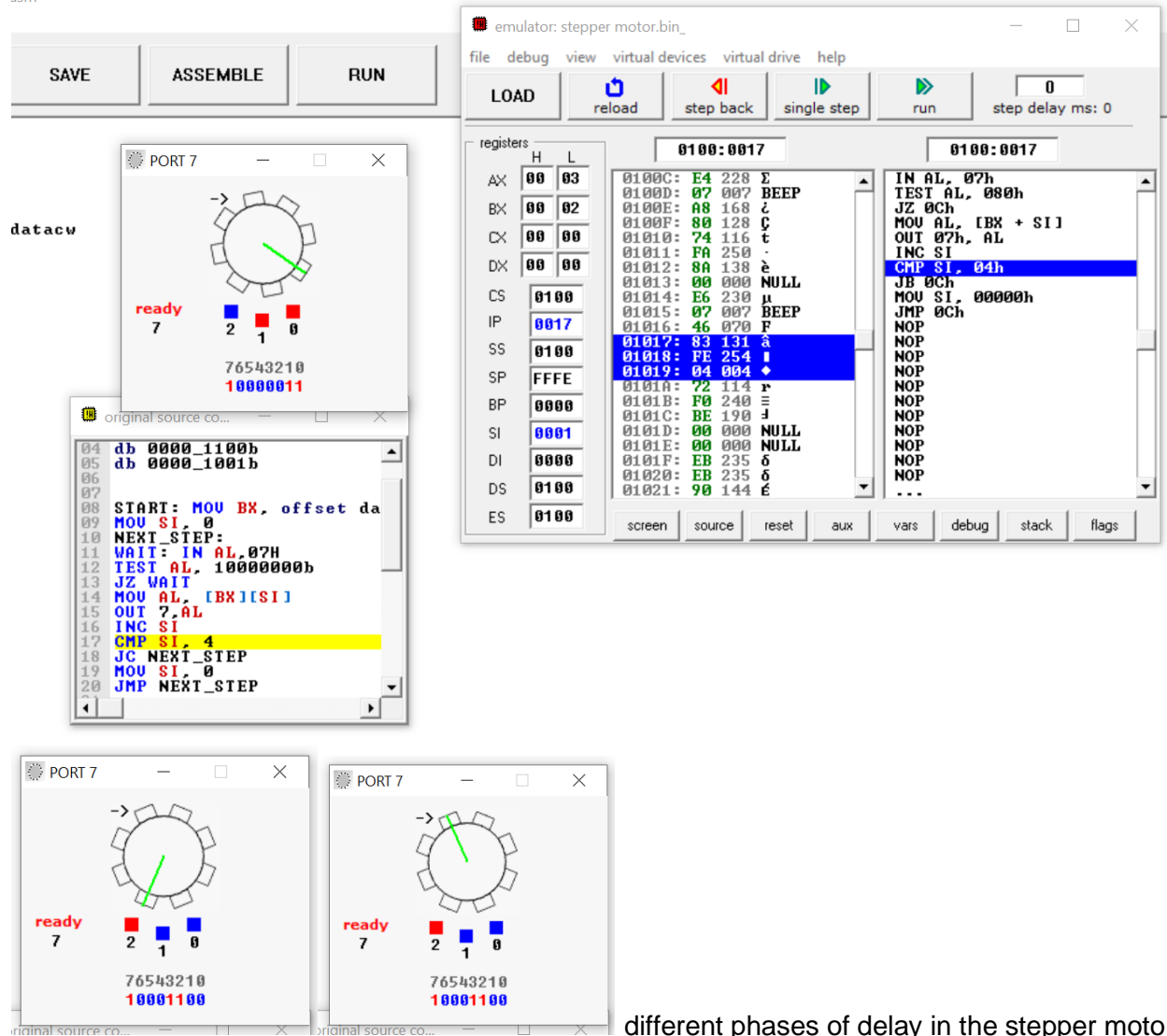
Screenshots :

```

01 ;19BCE2074
02 ;KULVIR SINGH|
03 jmp start
04 datacw db 0000_0011b
05         db 0000_0110b
06         db 0000_1100b
07         db 0000_1001b
08
09
10 START:  MOV BX, offset datacw
11         MOV SI, 0
12 NEXT_STEP:
13 WAIT:   IN AL, 07H
14         TEST AL, 10000000b
15         JZ WAIT
16         MOV AL, [BX][SI]
17         OUT 7, AL
18         INC SI
19         CMP SI, 4
20         JC NEXT_STEP
21         MOV SI, 0
22         JMP NEXT_STEP

```

asm



different phases of delay in the stepper motor

Inference :

The program can successfully simulate a stepper motor