IMPERIAL COLLEGE of ENGINEERING

(Affiliated by Rajshahi University Code: 385)

Department of CSE

Report no- 1

Course Title: Computer Peripherals and Interfacing lab

Course Code: CSE4142

Submitted by,

Name: Teresa Jency Bala
ID: 1938520113
Part: 4, Semester: Odd

Use 8086 Interfacing Trainer in Kit mode to make a traffic controlling system with three LEDs as the representing light of Red, Green, Yellow color light. Here the lights will be displayed in the following sequence and time period:

1. Red light will be on for 10 seconds
2. Yellow light will be on for 5 seconds
3. Green light will be on for 15 seconds
4. Red light will be on for 10 seconds
5. Repeat the sequence.

Date: 4-September-2023
Day: Monday

Submitted to,

Shovon Mandal
Adjunct Lecturer, CSE

Title:

Using emu8086 and Emulation Kit for 8086 Microprocessor to Display Traffic Light in Sequence

Objective:

The objective of this lab experiment is to create a traffic controlling system using the 8086 Interfacing Trainer in Kit mode, where three LEDs represent the Red, Green, and Yellow lights. The lights will be displayed in the following sequence from right to left and time periods:

- Red light will be on for 10 seconds.
- Yellow light will be on for 5 seconds.
- Green light will be on for 15 seconds.
- Red light will be on for 10 seconds.
- This sequence will repeat continuously.

Theory:

In this experiment, we utilize the 8086 Interfacing Trainer in Kit mode to simulate a traffic controlling system with three LEDs representing the Red, Yellow, and Green lights. The 8086 microprocessor serves as the control unit for managing the sequence and timing of these lights. The main objective is to create a traffic light cycle, starting with the Red light for 10 seconds, followed by the Yellow light for 5 seconds, the Green light for 15 seconds, and finally, the Red light for 10 seconds. This sequence repeats indefinitely, simulating a basic traffic signal.

Requirements:

- emu8086 emulator.
- 8086 microprocessor emulator.
- Provided 8086 assembly code.

Procedure:

1. **Initialization:** Set up the 8086 Interfacing Trainer in Kit mode and load the provided 8086 assembly code into its memory**.**
2. **Main Loop (MAINLOOP):**
    - Enter the main loop to control the traffic light sequence.
    - Initialize the AL register with '01h' (binary '00000001') to activate the Red LED.
    - Set CX to 4 to indicate the Red light phase**.**
3. **LED Display (LED):** Output the value in AL to the port address specified in DX to activate the Red LED. Rotate the value in AL to the left (ROL) to prepare for the next LED activation**.**

    Delay and Phase Control: Compare the value in CX to determine the current phase of the traffic light cycle:

    - If CX is 4, jump to D10 for a 10-second delay (Red light).
    - If CX is 3, jump to D5 for a 5-second delay (Yellow light).
    - If CX is 2, jump to D15 for a 15-second delay (Green light).
    - If CX is 1, jump to D10L for another 10-second delay (Red light).

4. **Delay Functions:**
    - Implement delay functions using the 8086's timing capabilities (INT 15h) to control the duration of each traffic light phase.
    - After the delay, reset CX to its initial value and update the port address for the LED display.

Looping and Continuation: Use the LOOP instruction to repeat the LED display and delay functions until CX reaches zero. After displaying the Red, Yellow, Green, and Red lights in sequence, jump back to the MAINLOOP to repeat the traffic light cycle indefinitely.

Completion and Return (RET): Upon manual reset or system reset, the program terminates.

Code:

```
DSEG    SEGMENT 'DATA'
DSEG    ENDS
SSEG    SEGMENT STACK   'STACK'
 DW     100h    DUP(?)
SSEG    ENDS
CSEG    SEGMENT 'CODE'
START   PROC    FAR
; Store return address to OS:
          PUSH    DS
          MOV     AX, 0
          PUSH    AX
; set segment registers:
          MOV     AX, DSEG
          MOV     DS, AX
          MOV     ES, AX
          MOV DX, 2070h
MAINLOOP:
         MOV AL, 01h
          MOV CX, 4
LED:
         OUT DX, AL
          ROL Al, 1

          cmp cx,4
          je D10

          cmp cx,3
          je D5

          cmp cx,2
          je D15

          cmp cx,1
          je D10L
```

```
D10: ;Red Light 10 sec Delay
   mov bx,cx
   mov CX,0098h
   mov DX,9680h
   mov ah,86h
   int 15h
   mov cx,bx
   MOV DX, 2070h
   LOOP LED

D15: ;Green Light 15 sec Delay
   mov bx,cx
   mov CX,004Ch
   mov DX,4B40h
   mov ah,86h
   int 15h
   mov cx,bx

   mov bx,cx
   mov CX,0098h
   mov DX,9680h
   mov ah,86h
   int 15h

   mov cx,bx
   MOV DX, 2070h
   LOOP LED

D10L: ;Red Light 10 sec Delay
   mov bx,cx
   mov CX,0098h
   mov DX,9680h
   mov ah,86h
```

```
D5: ;Yellow Light 5sec Delay
   mov bx,cx
   mov CX,004Ch
   mov DX,4B40h
   mov ah,86h
   int 15h
   mov cx,bx
   MOV DX, 2070h
   LOOP LED
```

```
   int 15h
   mov cx,bx
   MOV DX, 2070h

   JMP MAINLOOP
 ; return to operating system:
          RET
START   ENDP

CSEG    ENDS

      END    START
```

Result:

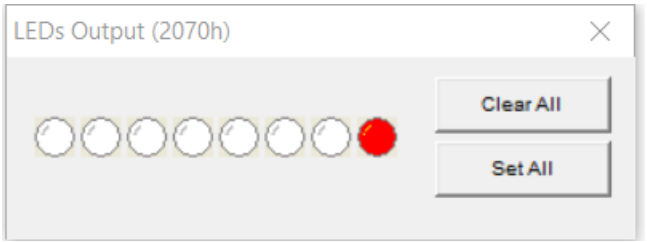

Figure1: Traffic Light Simulation - Red Phase: This lasts for 10 seconds, as per the specified timing

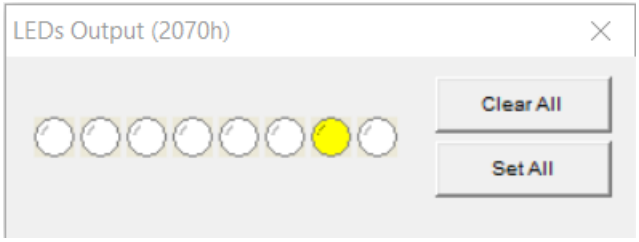

Figure2: Traffic Light Simulation – Yellow Phase: This lasts for 5 seconds, as per the specified timing

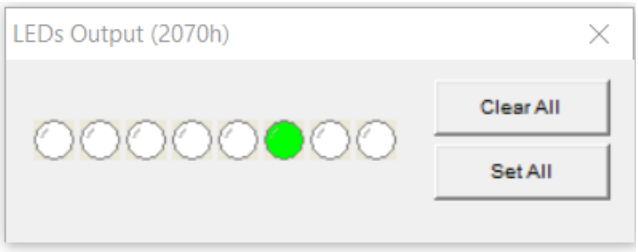

Figure3: Traffic Light Simulation - Green Phase: This lasts for 15 seconds, as per the specified timing

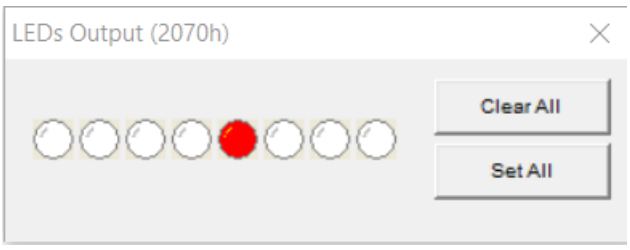

Figure4: Traffic Light Simulation - Red Phase: This lasts for 10 seconds, as per the specified timing

- Execute the provided 8086 assembly code on the 8086 Interfacing Trainer in Kit mode.
- Observe the three LEDs representing the Red, Yellow, and Green lights.
- Confirm that the traffic light sequence is as follows: Red (10 seconds) -> Yellow (5 seconds) -> Green (15 seconds) -> Red (10 seconds).
- Observe the sequence repeating indefinitely until manually reset or system reset.

Conclusion:

In this laboratory experiment, we successfully programmed the 8086 microprocessor using the emu8086 emulator and an Emulation Kit to simulate a realistic traffic light sequence. The program accurately emulated the specified traffic light timings and positions, following the sequence of Green, Yellow, Red, Green, Yellow, Red, and repeating for these 6 positions. This experiment demonstrated the principles of microprocessor programming, precise timing control, and the use of emulation environments to recreate real-world scenarios.