**Daffodil International University**

**Department of Electrical and Electronic Engineering**

**EEE 324: Microprocessors and Interfacing Laboratory**

**Experiment No: 05**

**NAME OF THE EXPERIMENT: Write a program to turn ON/OFF the LEDs present in MDE-8086 kit**

## Objective:

### Turning ON/OFF LED using microprocessor kit by creating a list file writing a program in which the coding of this program ensures the switching on the LEDs.

## Theory:

### I/O address map

|  |  |  |
| --- | --- | --- |
| ADDRESS | MEMORY | DESCRIPTION |
| 00H ~ 07H | LCD & KEYBOARD | LCD Display00H : INSTRUCTION REGISTER02H : STATUS REGISTER04H : DATA REGISTERKEYBOARD01H : KEYBOARD REGISTER( Only read)01H : KEYBOARD REGISTER(Only write) |
| 08H ~ 0FH | 8251/8253 | 8251(data communication)08H : DATA REGISTER0AH : INSTRUCTION/STATUS REGISTER8253(TIMER / COUNTER)09H : TIMER 0 REGISTER0BH : TIMER 1 REGISTER0DH : TIMER 2 REGISTER0FH : CONTROL REGISTER |
|  10H ~ 17H | 8259/SPEAKER | 8259 (Interrupt controller)10H : COMMAND REGISTER12H : DATA REGISTERSPEAKER11H : SPEAKER |
| 18H ~ 1FH | 8255-CS-1 /8255-CS-2 | 8255-CS1 (DOT & ADC INTERFACE)18H : A PORT 1AH : B PORT 1CH : C PORT 1EH : CONTROL REGISTER8255-CS2 (LED & STEPPIN MOTOR)19H : A PORT 1BH : B PORT 1DH : C PORT 1FH : CONTROL REGISTER |
| 20H ~ 2FH | I/O EXTEND CONNECTOR |
| 30H ~ FFH | USER’S RANGE |

## 8255A (LED & 7-SEGMENT & STEPPING MOTOR INTERFACE)

8255-CS2 (LED & STEPPIN MOTOR)

19H : A PORT (connected to 7-segment LED)

1BH : B PORT 4 bits (LSB) are connected to 4 coloured LED

 B0 → LED 1 (Red)

B1 → LED 2 (Green)

B2 → LED 3 (Yellow)

B3 → LED 4 (Red)

 4 bits (MSB) are connected to Stepping Motor Driver

1DH : C PORT (connected to D/A converter)

1FH : CONTROL REGISTER

## 8255 control Word:



## Procedure:

### First we create an .ASM source file using the DOS editor/Notepad like follows.

### The editor is opened and we entry the following codes

Port B Address

B1=1 and B0 = 1 to turn ON LED 1 and LED 2

Control register address

Control word

B0 = 1 to turn ON LED 1

#  CODE SEGMENT

#  ASSUME CS: CODE

#  ORG 1000H

#  MOV AL, 10000000B

#  OUT 1FH, AL

#  MOV AL, 11110001B

#  OUT 1BH, AL

#  MOV AL, 11110011B

#  OUT 1BH, AL

# MOV AL, 11110111B

#  OUT 1BH, AL

#  MOV AL, 11111111B

#  OUT 1BH, AL

#  INT 3

# CODE ENDS

# END

#

#  CODE ENDS

#  END

1. After editing the above codes we save the file and thus the .ASM file is created.
2. Then we go back to the Command Prompt and run MASM.EXE to create the object file, list file.
3. In list file you will see machine code for each instruction you wrote.
4. Enter Machine code in MDE kit mode manually from location 0000:1000 as default CS=0000 and IP=1000 at start up.
5. Press STP button to execute each instruction.
6. Observe the result of execution of OUT instruction.

## Result:

### Pressing STP Key causes the CPU to execute one instruction associated with the memory address CS:IP. After execution of OUT 1BH,AL the LEDs are turned ON according to the lower 4 bits of data (stored in AL) sent to the port address 1BH.

# After the execution of OUT 1BH,11110001, the LED is shown as follows:



 LED

 As this way we can verify the changing lights of the LED according to the coding bits.

## 7-segment display:

7-segment display (figure: 1) is used to form the decimal characters 0 through 9. One common arrangement uses Light –emitting diodes (LEDS) for each segment. By controlling the current through each LED, some segments will be light and others will be dark so that the desired character pattern will be generated.

 

 Figure1: 7-Segment arrangement

LEDs are connected in common anode arrangement.

Binary “0”indicates LED is ON and binary “1”indicates LED is OFF.

To display ‘5’ we need to turn ON LED no: 1,6,7,3,4

So

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 =A7 | 7 =A6 | 6 =A5 | 5 =A4 | 4 =A3 | 3 =A2 | 2 =A1 | 1 =A0 |
|  | 0 | 0 |  | 0 | 0 |  | 0 |

Blanks will be filled with ‘1’.

So A = 10010010

The decimal characters 0 through 9 is displayed in the 7-Segment display by the corresponding required segment’s ON or OFF.

Write an assembly code to output your group number in 7 segment display.

1. Put value of A to AL so that your group number will be shown according to above table.
2. Use instruction OUT (port address), AL
3. Use I/O address of port A where 7 segment LED is connected
4. Use instruction INT 3 to terminate program.
5. Use MASM.EXE to create object and list file.
6. Enter Machine code in MDE kit mode manually from location 0000:1000 as default CS=0000 and IP=1000 at start up.
7. Press GO to execute your program.

**Report:**

Write code so that 7 segment display will display 1,2,3,4,5 each after few seconds.