EE 337: Introduction to PT-51 Board
Lab 3

13^{th} August, 2018

This set of experiments has the following objectives:

- Familiarization with pt-51 board.
- Familiarization of breaking down a problem into subproblems and using subroutines to solve a problem.

1 Homework

1. Install FLIP (for windows user) or DFU_Programmer for Linux users.This software is used to program the pt-51 board.
2. After installing FLIP go through the slides and video uploaded on moodle and test the pt-51 kit. You have to demonstrate the same to your TA in the lab.
3. Load the hex file for Nibble display (generated from Problem 2.1 of Lab 2) on the PT-51 kit and verify the working. (5 points)

2 Lab work

2.1 Problem: LED Display

Read the value of the switches to get the delay D and perform the following task.

- Toggle the first LED (p1.7) with a delay of D seconds.
- Toggle the second LED with a delay of $\frac{D}{2}$ seconds.
- Toggle the third LED with a delay of $\frac{D}{4}$ seconds.

The minimum value of D can be 1 second. As an example, if D is 4, then p1.7 (LED) should toggle with 4s delay, second LED should toggle with 2s delay and third LED should glow with 1s delay. Before loading the hex code to the kit, demonstrate the delay of each port pins in the logic analyser to your TAs. (5 points)

2.2 Problem: Gray Coding and Decoding

Write a program to convert binary code to gray code, and vice-versa. Display the same on LED using the following algorithm.

- Switch P1.3 will acts as a status switch.
  - If P1.3 is 0 then perform binary to gray conversion.
  - If P1.3 is 1 then perform gray to binary conversion.

- The remaining three switches will act as inputs.
  Suppose the switches are 1110, then you have to perform gray to binary of 110 (i.e., the input is a gray code). If the switches are 0110, then you have to perform binary to gray of 110 (i.e., the input is a binary code). (10 points)

2.3 Problem: Read nibble and pack nibbles

Accept the most significant nibble and least significant nibble of a byte from the user, using 4 switches (P1.3 P1.0) and store the value to the location 4Eh and 4Fh. Also combine the numbers at 4Eh and 4Fh into a 8 bit number and store the result at 50h.(10 points)
readNibble:

```
; First configure P1.3-P1.0 as input
; Set pins P1.7-P1.4(LEDs) (indication that routine is ready to accept input)
; wait for 5 sec during which user can give input
; Clear pins P1.7-P1.4
; wait for one sec
; read the input on P1.3-P1.0 (nibble)
; show the read value on pins P1.7-P1.4(LEDs)
;wait for 5 sec
```

The following code is to verify that the user input is right

```
; USER sets all switches if I/P is verified.  (0Fh)
; Follow the above-mentioned procedure to accept a nibble
; If the nibble reads 0Fh, USER input is verified.
; Else, User has to input the nibble again.
RET
```

packNibbles:

```
; Combine the two nibbles and store the byte to 50H.
RET
```

main:

```
call readNibble; Read the MSB from the user and store in into locations 4EH
call readNibble; Read the LSB from the user and store in into locations 4FH
call packNibbles;
end
```

2.4 Bonus Question: Compute Square-root

Write a program to find the squareroot of a number stored in 50H and display it on LEDs (P1.7 MSB). If the number is a perfect square display as such, otherwise display the most significant nibble of the number stored.
Note:

- You should not use look-up table to find the square-root.
- If you are not able to complete the bonus question in lab, you have to complete it before coming to next lab. (10 points)