EE 337: Introduction to PT-51 Board
Lab 3

26^{th} August, 2019

This set of experiments has the following objectives:

- Familiarization with pt-51 board.
- Familiarize with few more instructions.

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. 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. Demonstrate the delay of each port pins in the logic analyser to your TAs. (5 points)

2 Lab work

2.1 Testing of Board

- 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.
- Load the hex file for LED Display (Homework 2) on the PT-51 kit and verify the working. (5 points)

2.2 Problem: Convolution

Write a subroutine `convolution` that can read two input signals of length-3, whose starting location is specified by the pointers saved in $50H$ and $51H$. The result of convolution is to be saved, starting from the location specified by pointer saved in $52H$. (10 points)

Example:

If the pointers saved in $50H$ and $51H$ are $40H$ and $45H$ and the memory contents of 40H, $41H$, $42H$, $45H$, $46H$, $47H$ are $x1, x2, x3, y1, y2, y3$ respectively. Then the result of convolution of $[x1, x2, x3] * [y1, y2, y3] = [x1 \times y1 + x2 \times 0 + x3 \times 0, x1 \times y2 + x2 \times y1 + x3 \times 0, x1 \times y3 + x2 \times y2 + x3 \times y1, x1 \times 0 + x2 \times y3 + x3 \times y2, x1 \times 0 + x2 \times 0 + x3 \times y3]$ is in the addresses pointed by $52H$,i.e., if we have $60H$ as pointer in $52H$, then the results are saved from $60H$ to $69H$ (2 memory locations for each convolution output of 16 bits). As the result would be 16 bits, 2 memory locations would be used to save the same where the higher byte goes in the first memory location and lower byte goes to the next.

One approach to implement the same is to zero-pad the content of second signal on both ends by 2 pair of zeroes , i.e new data would be [0, 0, y1, y2, y3, 0, 0]. Then in iteration do the following:

- multiply the value of first signal in swapped order i.e., [x3, x2, x1] with 3 elements in order of the zero padded second signal and add them
- save the result (16 bit) in the destination pointer
- Increment the start pointer used for reading the 3 elements from zero padded second signal
- Increment the resultant pointer

Note that it is not necessary to swap the first signal in physical memory. You can simply point to the data to be multiplied first to x3, then decrement the pointer to point at address of x2 and so on.

2.3 Debugger Related Question

After completing the convolution problem, answer the following question to your TA.(4 points)

- What is the data size, code size, external data size in bytes?
- Show the first output of convolution result on PORT P1 using logic analyzer and peripherals option available in debugger
- Check whether **overflows** happened either during ‘mul’ or ‘add’ operation
- How many machine cycle the complete code takes? Verify your hand calculation result with the debugger (use state available in register window)