EE 337: Microprocessors Laboratory (Spring 2023)
Indian Institute of Technology Bombay

Lab 3: 20 points — Date: January 19, 2023

1. [8 points] In order to convert an analog signal to a digital signal, the 3 steps to be followed are sampling, quantization and encoding. Assume that a sampled signal (consisting of 8 samples in the range 0 to 40) is present. Write an assembly code to quantize the signal into 4 discrete levels. Follow the below scheme.

```
if (sample >= 0 and sample < 10):
    output 5
else if (sample >= 10 and sample < 20):
    output 15
else if (sample >= 20 and sample < 30):
    output 25
else:
    output 35
```

(Note that these values are given in decimal. You need to use equivalent hexadecimal values in your code)

```
 // -- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
CALL QUANT
HERE: SJMP HERE
ORG 130H
// *****************

QUANT:
// ADD YOUR CODE HERE
RET
END
```

- The inputs samples should be present in locations `60H` to `67H`.
- The quantized output samples should be present in memory locations `70H` to `77H`.
- To reduce the effort involved in adding multiple items in memory locations, you can use the command window in Keil. Refer to Lab-2 lab-sheet.

TA Checkpoint 1

Check the following cases:

- Input samples : `11H, 2AH, 01H, 3FH, 18H, 1CH, 0EH, 06H`.
  Output samples : `0FH, 23H, 05H, 23H, 19H, 19H, 0FH, 05H`.
- Input samples : `00H, 09H, 0AH, 13H, 14H, 1DH, 1EH, FFH`.
  Output samples : `05H, 05H, 0FH, 0FH, 19H, 19H, 23H, 23H`.

2. [8 points] Moving Average Filter is used in Signal Processing and Data Analysis to smooth out high frequency noise and short-term fluctuations. Write an assembly language program to implement a 4-point Moving Average Filter on an input signal consisting of 8 samples.

   The formula for output signal is provided-

$$y[n] = \frac{x[n] + x[n-1] + x[n-2] + x[n-3]}{4}$$

   Assume that $x[-1]$, $x[-2]$ and $x[-3]$ are equal to 0.

```
// -- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
CALL FILT
HERE: SJMP HERE
ORG 130H
// ****************

FILT:
// ADD YOUR CODE HERE
RET
END
```

   - The input signal $x[n]$ samples must be stored in memory locations `60H` to `67H`.
   - The output signal $y[n]$ samples must be stored in memory locations `70H` to `77H`.
   - To reduce the effort involved in adding multiple items in memory locations, you can use the command window in Keil. Refer to Lab-2 lab-sheet.
   - You can assume that the addition result never crosses 8 bits (i.e there is no overflow and each operation can be done in a single instruction).

TA Checkpoint 2

Check the following cases:

- Input samples : `04H, 24H, 38H, 11H, 3AH, 2BH, 0EH, 69H`.
  Output samples : `01H, 0AH, 18H, 1CH, 29H, 2BH, 21H, 37H`.
- Input samples : `56H, 1AH, 33H, 4DH, 46H, 02H, 4EH, 56H`.
  Output samples : `15H, 1CH, 28H, 3CH, 38H, 32H, 38H, 3BH`.

3. [4 points] This will be an in-lab experiment, you will be demonstrating that your PT-51 kit works by running the test code for the kit. The test code also checks the peripherals on the kit. Using this, you can also verify that all required software is correctly installed. Before coming to the lab please install the softwares mentioned below and go through the slide decks to get familiarize with the board.

   - Get familiar with PT-51 kit by going through the slides here: **User Manual**
   - Install FLIP (on Windows) or DFU Programmer (on Linux/Mac). This software is used to load programs into the 8051 microcontroller on the PT-51 kit. The steps are here: **FLIP/DFU Installations**

   In the in-lab session, verify the following in front of your TAs.

   - The procedure to load a hex file onto the PT-51 kit using Flip is shown in the video given here: **Running a Hex file on Pt-51**

     Follow the procedure and load the `led.hex` file to see that the LEDs on your kit toggle.
   - Download the file `pt51_test.hex` and load it on your PT-51 kit, as described here: **Testing the peripherals of Pt-51**
   - Follow the steps in the slide deck titled "PT-51 Test Program" to do the self-test of the kit. Check if all tests run successfully. If there are failed tests, inform your respective TA.
   - Please use the installation given here **JRE_Installer** if you are not able to find JRE or the installation with JRE fails.
   - Please put your board in boot mode before installing drivers and flashing code. Detach → Press boot → Press Reset → Release Reset → Release Boot → Attach