

ECSE 324  
Laboratory No. 3 Report

Ismail Faruk 260663521

Tanbin Chowdhury 260578441

## Introduction

This is the report for Lab 3.

### Part 1: Slider switches and LEDs program

#### **Description**

The task was to turn on the LEDs using the corresponding Slide Switches on the board

#### **Approach Taken**

We used assembly files to read data from the Slide Switches data register. Then we wrote that data into the LED data register. With that, this part was solved. The assembly subroutines were used to access the data registers and reading and writing, accordingly.

#### **Challenge Faced**

None.

### Part 2: Basic I/O

#### **Description**

The task was to create an application that would use all of the drivers created. As before, the state of the slider switches would be mapped directly to the LEDs. Additionally, the state of the last four slider switches SW3-SW0 will be used to set the value of a number from 0-15. This number would be displayed on a HEX display when the corresponding pushbutton is pressed. Then to switch on all the segments of the displays HEX4 and HEX5. Lastly, using slider switch SW9 to clear all the HEX displays.

#### **Approach Taken**

The HEX display was programmed to with three subroutines. One of the subroutines would turn on all the segments of the corresponding HEX display, another subroutine would turn off all the segments of the corresponding HEX display and lastly to turn on particular segments based to show numbers between 0 and F. The push buttons were programmed similarly as the slider switch, to read data from the buttons. A C program was created to put together and use the subroutines accordingly.

#### **Challenge Faced**

Understanding enumeration and one-hot-encoding.

## Part 3: Polling based stopwatch

### **Description**

The task was to create a simple stopwatch using the HPS timers, pushbuttons, and HEX displays. The stopwatch would be able to count in increments of 10 milliseconds using a single HPS timer to count time. The board was to display milliseconds on HEX1-0, seconds on HEX3-2, and minutes on HEX5-4. PB0, PB1, and PB2 would be used to start, stop and reset the stopwatch respectively. We were to use another HPS timer set at a faster timeout value (5 milliseconds or less) to poll the pushbutton edgecapture register.

### **Approach Taken**

The HPS timer played a key role. Every time the timer would finish its count, it would increment a certain count variable in the C program created. The timeout was set as 10000 microseconds. Thus, the count would increment every 10 milliseconds. The rest of the program included the modulus and division function to extract data and plot onto the display accordingly. The button edgecapture was set up with another additional timer. The program was configured to use the buttons as start, stop and reset.

### **Challenge Faced**

Understanding the inner workings of the HPS timer.

## Part 4: Interrupt based stopwatch

### **Description**

The task was to modify the polling stopwatch application to use interrupts of the HPS timer and interrupts for the pushbuttons.

### **Approach Taken**

The program was modified to setup an interrupt subroutine for both the HPS timer and the pushbuttons. The interrupt subroutine would turn on a flag every time its count finished. The push button subroutine would trigger a flag every time a button was pressed. The C program was modified to run based on these interrupt flags.

### **Challenge Faced**

Implementing the subroutine for the pushbutton interrupt.

## Conclusion

Got to learn a lot about how interrupt and polling works.