EEC 181 – Lab Report #1

Chris Bird, Lillian Deas, Kaila Balancio

For lab 1, after getting through the trials and tribulations of getting our Qsys generated code to work, we had to perform a few tasks, get HELLO UUORLD to display and scroll on the board, get the switches to control specific segments, have functionality for scroll speeds, and a pause feature. Due to some mishaps with communication, we didn’t leave ourselves much time to complete the third part of the lab, and we didn’t have all three people to help us with it. The hardest part of the lab was figuring out how to work with decimal numbers and still having it work with the binary bits in the way we want it to. Meaning, when we want to load a specific set of 8 bits to the upper bits of a 32-bit integer, figuring out how to put that in without disturbing the rest ended up causing some issues.

After figuring this out, putting the other parts together wasn’t too hard. Delay speeds, calculated on paper and then hard coded into the code was the easiest as it was a simple variable that we add to or remove from based on speed up or slow down. This controls a ‘useless’ loop that will force delay on the next scroll cycle. This works with both message and random segment display scrolling speeds. To pause, we added in an infinite while loop that will continue to loop forever until the button is pressed again, so in the loop is only a check to see if the button has been pressed yet, if not, continue looping. The hardest part was displaying the message HELLO UUORLD and the specific segments based on button press. In the end, we decided to save time and hard code a ‘brute force’ approach to display HELLO UUORLD, this is a great place that we could improve our code efficiency.

In our specific segment display scroller, there was an issue with interfacing the switch value generated and the hex values as there was a dead bit on the board (where a decimal point would normally go) that we cannot use. Bits 9-0 of the switch value show the 10 switch values, but bits 6-0 show HEX0, and bits 14-8 control HEX1. Therefore, if we were to direct map it, bit 7 of the switch values would be wasted as it would map to an unused decimal place on the board. To fix this, we saved the bottom 7 bits to hex0 by shifting left 25 places and then back right 25 places to mask with 0s. As for hex1, we shifted right 7 to mask the lower bits, then back left 8 bits to put them in the right spot.

All in all, the best place for improving efficiency would be in the HELLO UUORLD scroller, and the hardest part to code wasn’t any specific part, but the inner workings of binary/decimal and how we can interact with the bits while only using decimal values.

Approximate hours spent on section 3 of lab 1: 19 hours