ECSE 324 – Lab Report #3 – Group 34

Jia Wei Sun, 260866206
Karl Michel Koerich, 260870321
15 November 2019

Overview

For this lab, we worked with the DE1-SoC Computer System and learned how to use the slider switches, push-buttons, LEDs and 7-Segment displays. After performing the required steps from Lab 3, we were able to use timers and interrupts to create a stopwatch while demonstrating polling and interrupts written in C language.

Firstly, we followed the steps from section #2: Basic I/O. By the end, we had a simple code that allows us to display hexadecimal characters on the LED displays in the board. For the next step, we used the provided code, combined to sample snippets of code in the lab manual, to implement a stopwatch. The last task was to improve the stopwatch implementation by making it interrupt based.

Although this lab was long and challenging to finish, we managed to perform all the proposed tasks and we were satisfied with the final results.

Approaches

To solve the proposed challenge in the first section, we were required to use the 7-segment display on the DE1-SoC board to display hexadecimal numbers. Also, we used the slider switches to define the hexadecimal value, and buttons to turn on the LEDs desired. The left-far switch was used to flood the LEDs, while the first 4 switches were used to compute the hexadecimal value we want to display. We use the buttons to display the hexadecimal in the LEDs according to the following pattern: Far-right button turns on the far-right LED. This pattern follows from right to left for 4 buttons and LEDS.

To solve the second task of Lab 3, we were challenged with implementing a stopwatch running increments of 10 milliseconds. We displayed milliseconds on HEX0 and HEX1; seconds on HEX2 and HEX3; and minutes on HEX4 and HEX5. We incrementing the HEX0 with the time clock from 0 to 9, reset it to 0 and incremented HEX1 by 1. When both HEX0 and HEX1 arrived to 9 at the same time, both were erased and HEX2 was incremented by 1. HEX2 also goes from 0 to 9 since it holds the most significant digit of seconds. When HEX2 reaches 9, it went to zero and HEX3 was increment. HEX3 went up to 6, to close 60 seconds. The pushbuttons were used to control the behaviour of the stopwatch. PB0 was used for the start button. PB1 was used for the pause button. Finally, PB2 was used for restarting the stopwatch. To perform the pause, we used another clock to pull the Edgecap register. When a button was pressed, the Edgecap register should have a value not equal to 0. The program started with the enable equals 0. The start button set enable to 1 and stop set it to 0. Meanwhile, the reset button restarted the counters to zero, cleared the LED displays, and set enable to 0.

To solve the third and last task of Lab 3, we were required to improve the stopwatch code from the previous section by implementing interrupters. The counters on the hexadecimal displays worked exactly the same, with the same encoding. Buttons still performed the same actions as before. However, the main difference was that, instead of using two timers, only one timer was used for the stopwatch. This means that, while the first stopwatch implementation relied on the clock tick of the timer to actually perform actions to the stopwatch (pause/start/restart), the improved version could perform these actions asynchronously. To make this improvement possible, we used the help of provided snippets of code and guidance from the lab manual. Instead of polling the Edgecap register, we were waiting for keys flag. Furthermore, when a button was pressed, the program would receive this flag signal and read the pb keys routine asynchronously, instead of waiting for the next iteration to perform an action. The Edgecap register would then we read and cleared.

Challenges

The first section of the lab was not very challenging. A big portion of the code was had already been provided. However, the specifications in the lab document were not very precise and we needed help from the TA to precisely understand what our goal was. Overall, there were not many challenges, except the fact we required extensive testing to make sure the right LED sections were lighting up.

The second task was challenging in the sense that there was a lot of actions being performed. Buttons and displays, timers and encoding – All of these were challenging to figure out and correctly implement. The TAs were very helpful and the lab manual provided us with valuable examples. The main challenge was understanding the Edgecaps, as well as reconfiguring the timer according to each action button pressed.

Similarly, the last task was challenging in the sense that there was a lot going on, but this time most of the code could be reused from the second task. Understanding how the flags work was the main challenge, but we were able to find helpful resources in the lab manual.

Conclusion

As we advance through the course, we are more and more mixing concepts and the Labs are getting more challenging, was requires us to invest more time on them. The main purpose of this lab was to not only introduce us to the basics of input output application, but also to make us more familiar with hexadecimal display, encoding, timers and interrupts. Even though it took us more time than expected to solve all the problems and demo, we were satisfied with the final results.