Charles Tung Fang
EE 271
Professor Rania Hussein

Lab 4 Report

Procedure

In order to understand the whole project and know where to start, I began with drawing the top-level block diagram for my entire design (Figure 1). The whole diagram is my DE1_SoC module and in that I called userInput and playField.
First, I dealt with the metastability in the metastability function. With the output from metastability, I then used it to count the key press. The key press is counted as one when it is pressed then released (Figure 2).
Second, I passed in the L and R output from the previous action into the playField module. In the playField function I called eight normalLight modules representing LEDR1~LEDR4 and LEDR6~LEDR9. I also called centerLight module representing LEDR5. After calling the lights, I have the victory module to determine the winner and show the winner on the HEX0.

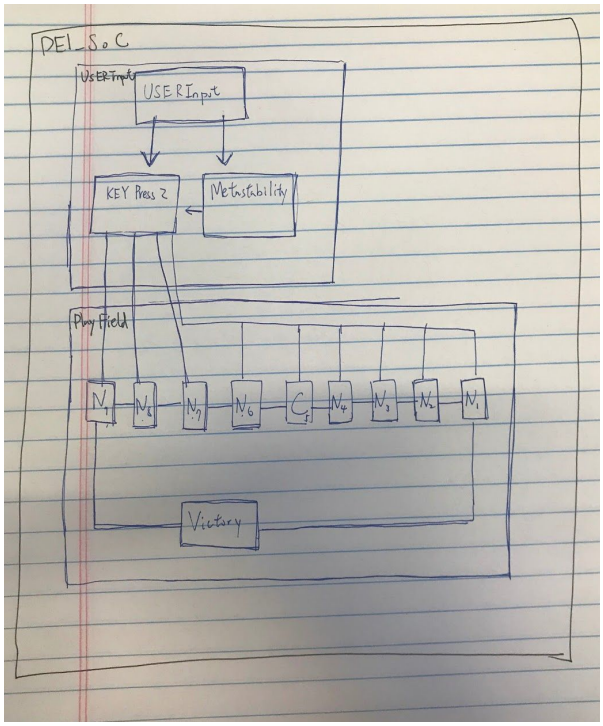


Figure 1. Block Diagram

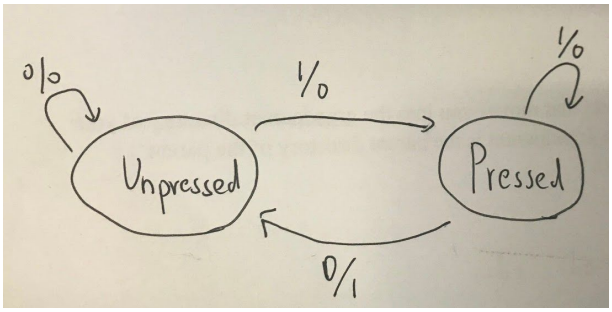


Figure 2. Key Pressed FSM

Results

After implementing each module, I tested every single one of them on the modelSim before I combined them all together. I made sure every module behaved the way they're expected through the modelSim. I tested larger modules, userInput and playField (Figure 3, Figure 4), to make sure inputs got modified and pass on. In the end, I tested the DE1_SoC (Figure 5) and wrote a testbench for player 1 & 2 to win each time in order to confirm if different LEDR values changed, and if ultimately the HEX0 showed the winner.

Figure 3. userInput ModelSim to test metastability & keypress2 modified values

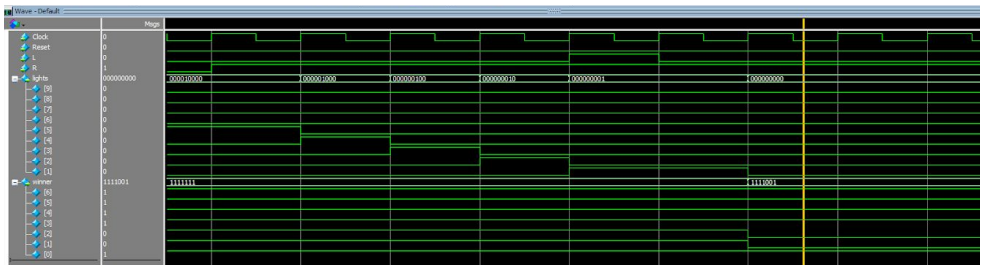


Figure 4. playField ModelSim to test each LEDR and HEX0's behavior

Figure 5. DE1_SoC ModelSim testing everything together

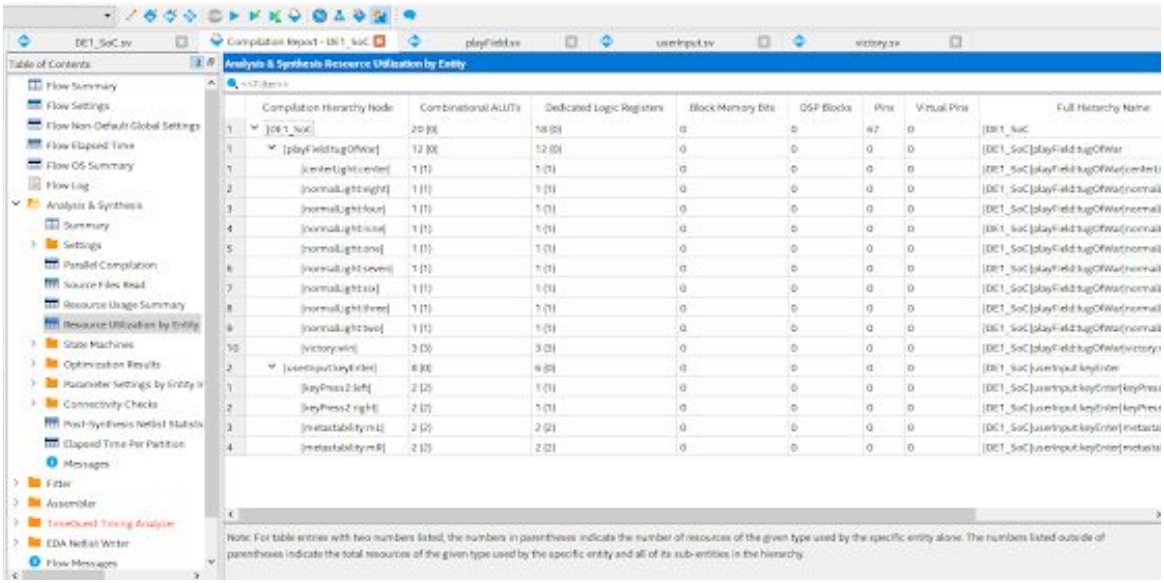


Figure 6. Resource Utilization by Entity

The size (Figure 6) of my design is 20 (Combinational ALUTs) + 18 (Dedicated Logic Registers) = 38. The final result performed successfully on the board. Player two press key 3 to make the light goes left and player one press key 0 to make the light goes right. HEX0 shows 2 if it reach to the beyond the leftmost light, and it shows 1 if it reach beyond the rightmost light.

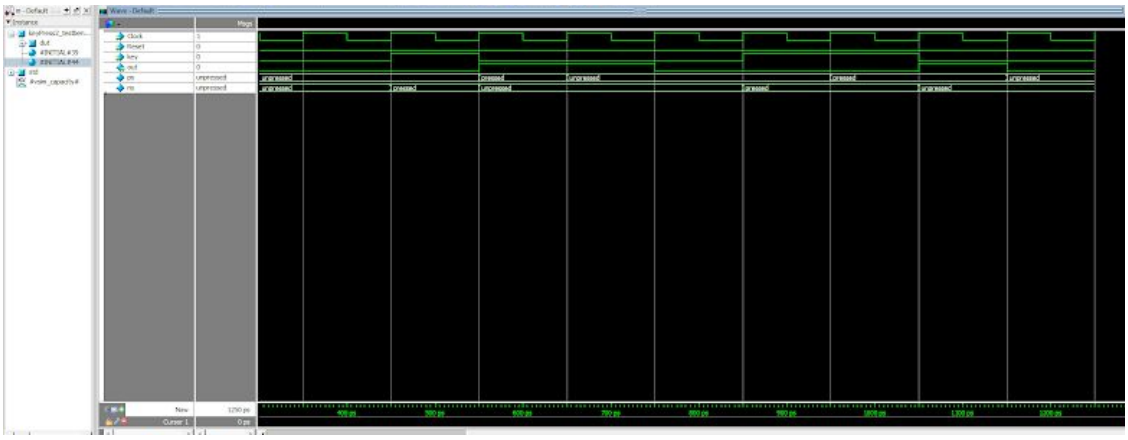
Problems Faced and Feedback

The biggest issue I had while completing the lab was to understand how to connect different logics from one module to another. I was confused at first because I was thinking everything in a software perspective. I was assuming the logic works as a variable so I assigned multiple values to an output at the same time, which gave me errors while compiling. However, everything became clear and straightforward after getting clarification from the TA.

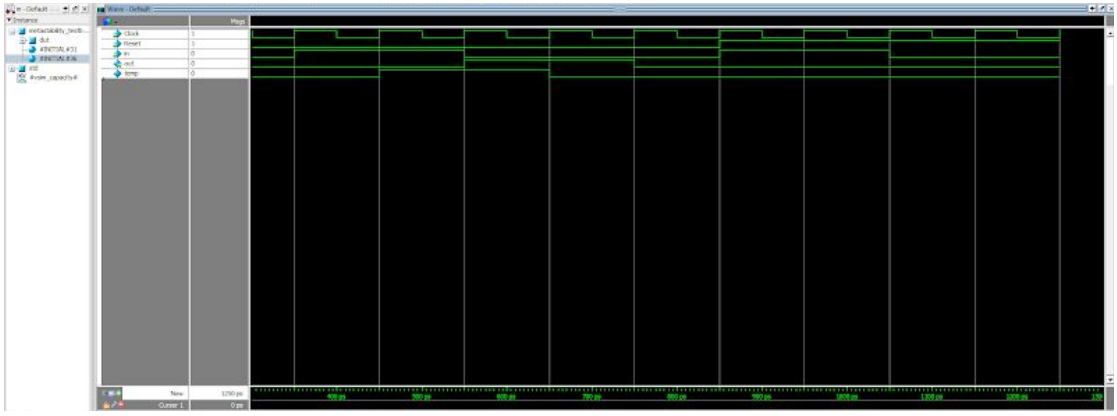
The tip and tricks I got away from this lab is to understand how to design and draw block diagram before I start the work. I realized that having a good block diagram really helps me to develop the design as I go along not only because I can always go back to examine my logic but also can check where I am at when I am lost.

Appendix

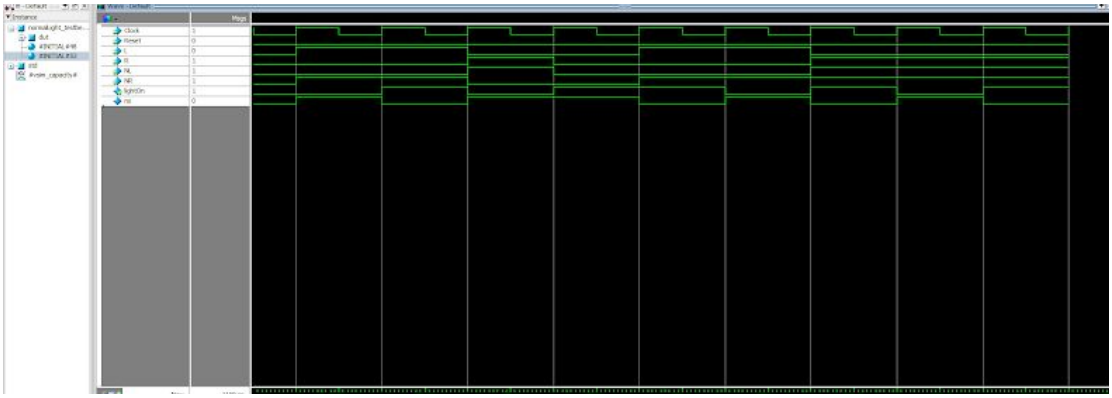
ModelSim for CenterLight



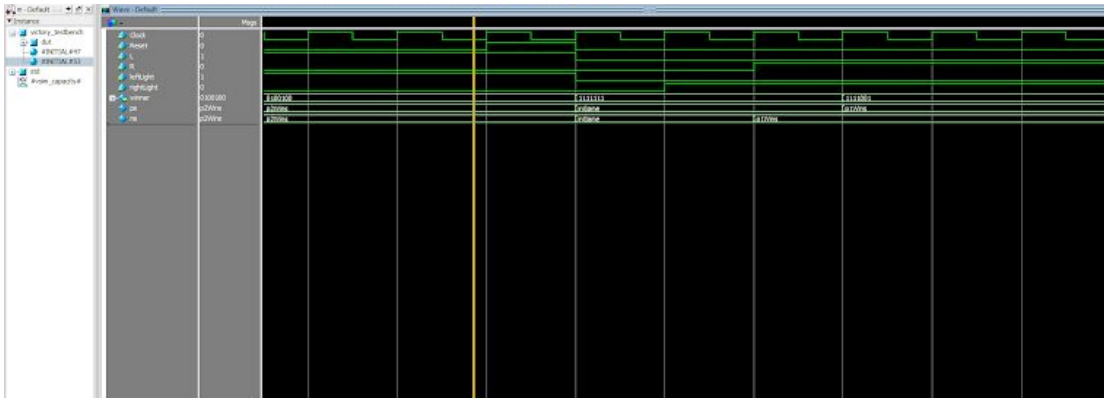
ModelSim for KeyPress2



ModelSim for Metastability



ModelSim for normalLight



ModelSim for Victory