Cynthia Li

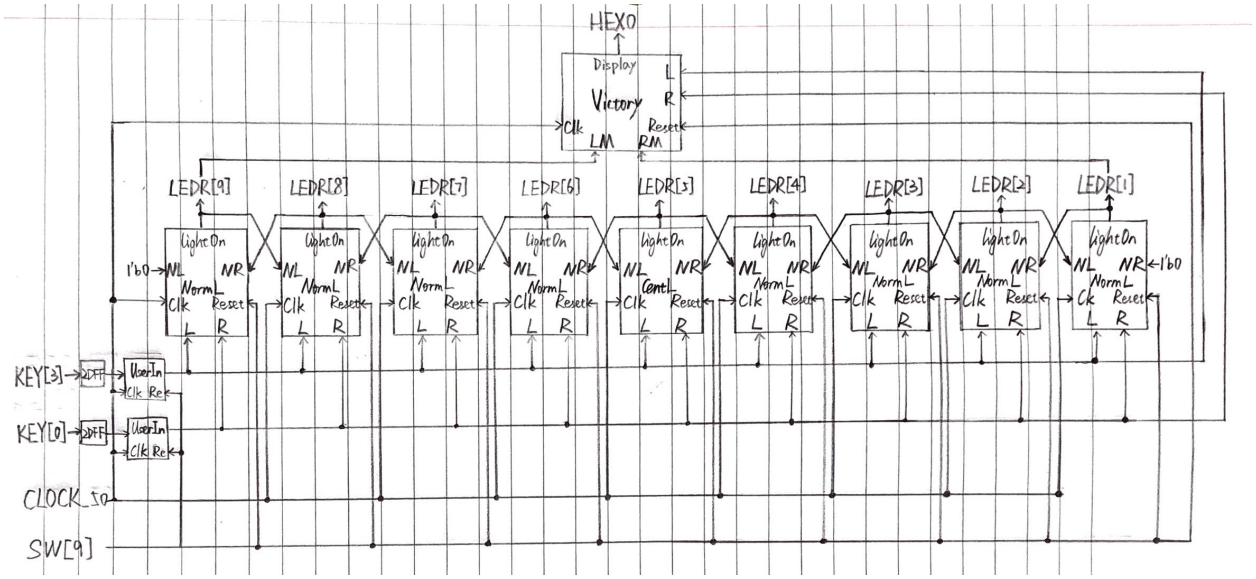
EE 271

LAB 6 REPORT

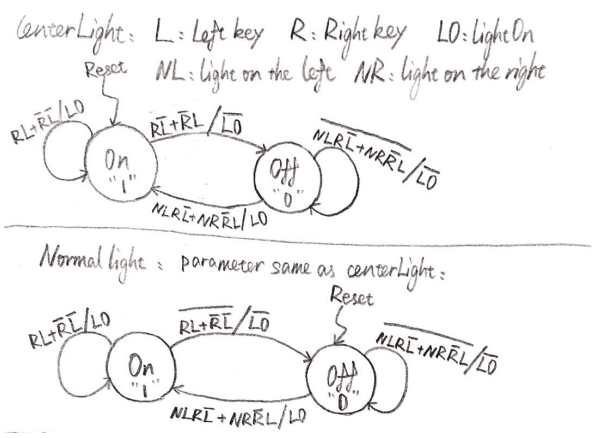
1. Video demo:

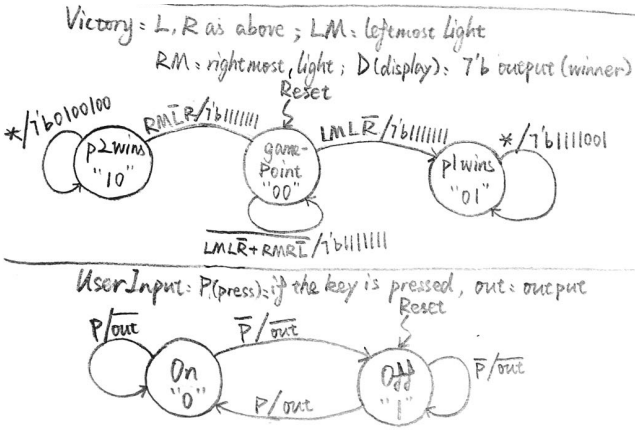
https://drive.google.com/file/d/1ae4maf1We6nTEY3HO3CRNcrv_f0AgeQd/view?usp=sharing

2. A drawing of your the top-level block diagram for your entire design, showing the major modules and how they are interconnected. It needs to show how the player input (KEYs) reach the lightfield output (LEDRs).

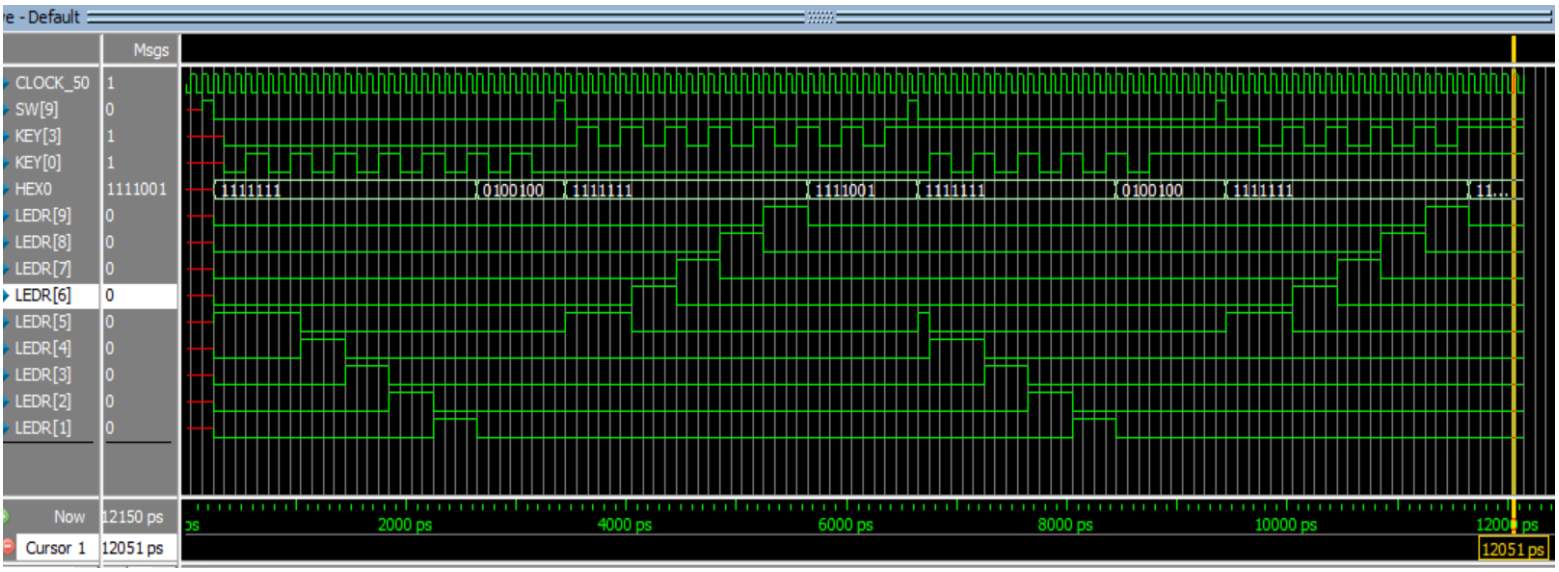


3. Drawings of your state diagrams for all FSM that you built.





4. A screenshot of ModelSim simulation of your top-level entity (DE1_SoC)



5. Screenshots (3+) of ModelSim simulation of each important element you built for this lab.

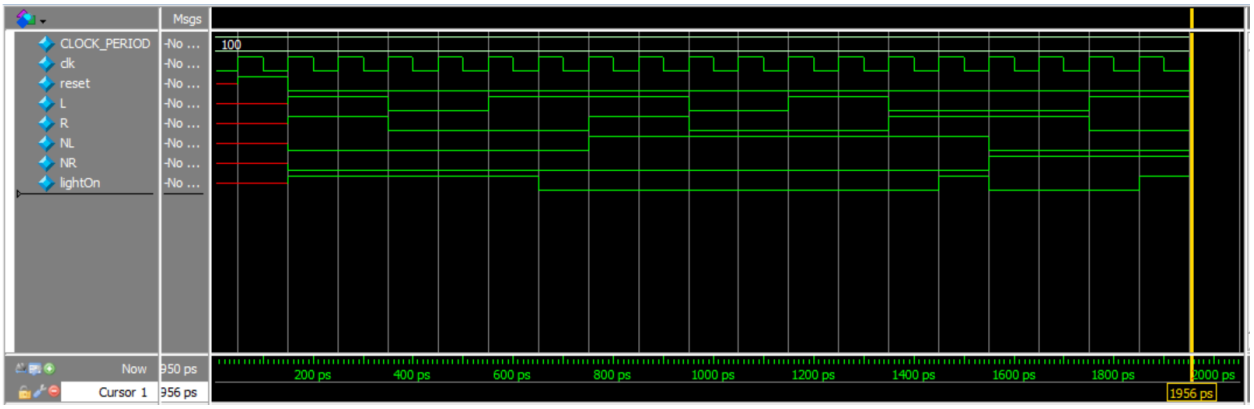


Fig.1 centerLight Simulation

centerLight_testbench: for each state (on, off), test p1 and p2 both pulling, neither pulling and one of them pulling and test NL, NR to both 0 or one of them getting 1 to move transition between states. (specific cases please reference code comment)

It should work as it tests all possible player inputs and regulated NL and NR, since in real game, NL and NR will not both be true, we don't really care about that case, so current cases are sufficient to show when the center light should be on.

We should expect to see lightOn is 1 once the module is reset, and will stay true when L and R are same value, lightOn will also be 1 when it's previously 0 and NL&R&~L occurs or NR&L&~R occurs.

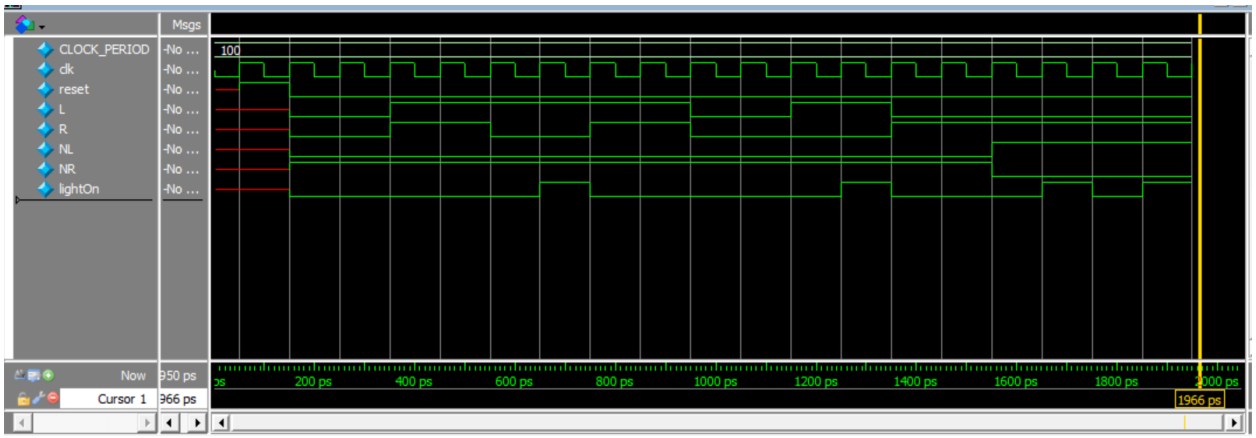


Fig 2. NormalLight Simulation

normalLight_testbench: for each state (on, off), test p1 and p2 both pulling, neither pulling and one of them pulling and test NL, NR to both 0 or one of them getting 1 to move transition between states. (specific cases please reference code comment)

It should work as it tests all possible player inputs and regulated NL and NR, since in real game, NL and NR will not both be true, we don't really care about that case, so current cases are sufficient to show when the center light should be on. We should expect to see lightOn is 1 when NL&R&~L and NR&L&~R, and sometimes when R and L have the same value (during on state, not that obvious from graph). In fact, since centerLight and normalLight uses the same code except reset to a different state, when centerLight works, normalLight also works.

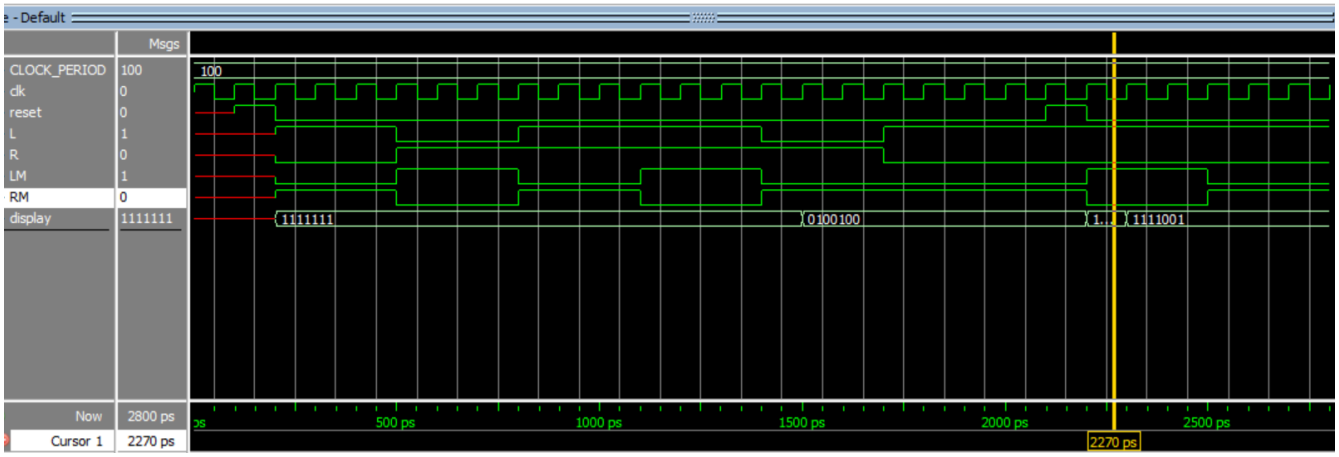


Fig 3. *Victory* simulation

Victory_testbench tests cases:

1) at game point (leftmost light is on or the right most light is on), the next pull is dragging it to the opposite (disadvantaged) side;

2) at the game point, the next pull is both player pulling at the same time;

3) at the game point, the next pull makes player2 wins;

4) after player2 wins, the displayed name doesn't change until reset;

5) at game point, the next pull makes player1 wins;

6) after player1 wins, the displayed name doesn't change until reset.

It basically tested all the possible cases that the players might be pulling in a real game, matching with the described functionality (continues to display the name, determines which player wins, and determines if the game has a winner yet), so it should ensure the module working correctly. Based on the tested cases, we should see that "*display"* will stay as 7'b1111111 (as turned off display) until RM=R=1, and L=0 occurs (player 2 wins), which will then displaying 7'b0100100 ("2" for player2). Then we see reset happens, and LM=L=1, R=0 (player1 wins), and "*display"* is now 7'b1111001 ("1" for player1).

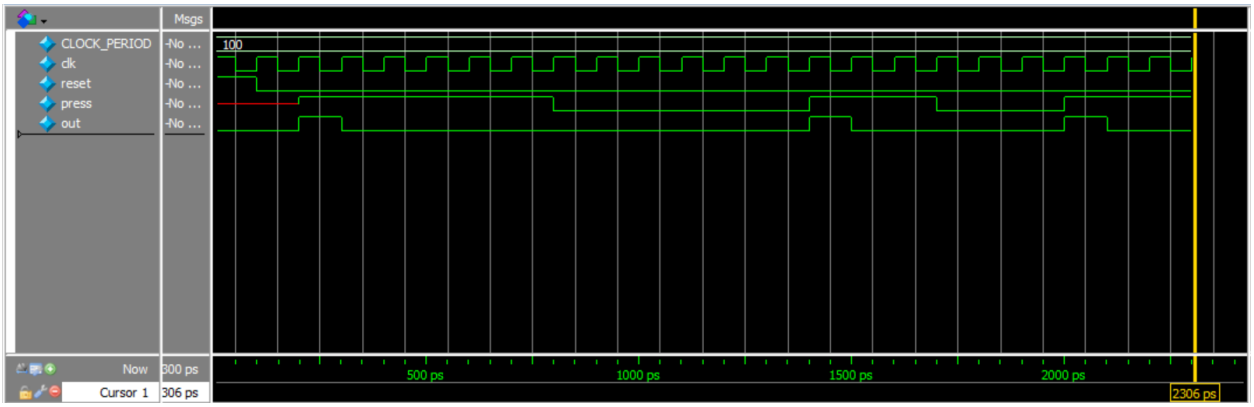


Fig 4. userInput Simulation

userInput_testbench cases:

1) Currently at off state: key is pressed
2) Currently at on state: key is pressed
3) Currently at on state: key is released
4) Currently at off state: key is unpressed

This testbench should be able to check whether the module works correctly as it goes through every edge in the state diagram to check if the output is correct.

We expect to see that each time *"press"* changes to "1," "*out"* will immediately show "1" but only for one cycle.

6. Screenshot of "Resource Utilization by Entity," along with the computed size of your design.

	Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers
1	|tugOfWar	18 (0)	17 (0)
1	|centerLight:l5|	1 (1)	1 (1)
2	|metastability:m1|	1 (1)	2 (2)
3	|metastability:m2|	1 (1)	2 (2)
4	|normalLight:l1|	1 (1)	1 (1)
5	|normalLight:l2|	1 (1)	1 (1)
6	|normalLight:l3|	1 (1)	1 (1)
7	|normalLight:l4|	1 (1)	1 (1)
8	|normalLight:l6|	1 (1)	1 (1)
9	|normalLight:l7|	1 (1)	1 (1)
10	|normalLight:l8|	1 (1)	1 (1)
11	|normalLight:l9|	1 (1)	1 (1)
12	|userInput:leftkey|	2 (2)	1 (1)
13	|userInput:rightkey|	2 (2)	1 (1)
14	|victory:vic|	3 (3)	2 (2)

Size = 18 + 17 = 35

7. Approximately how much time did you spend on this lab (including reading, planning, design, coding, debugging etc.)?

30 hours.