Khoa Tran

EE 271

July 28, 2020

Lab 4 Report

**Procedure**

**Tug of War**

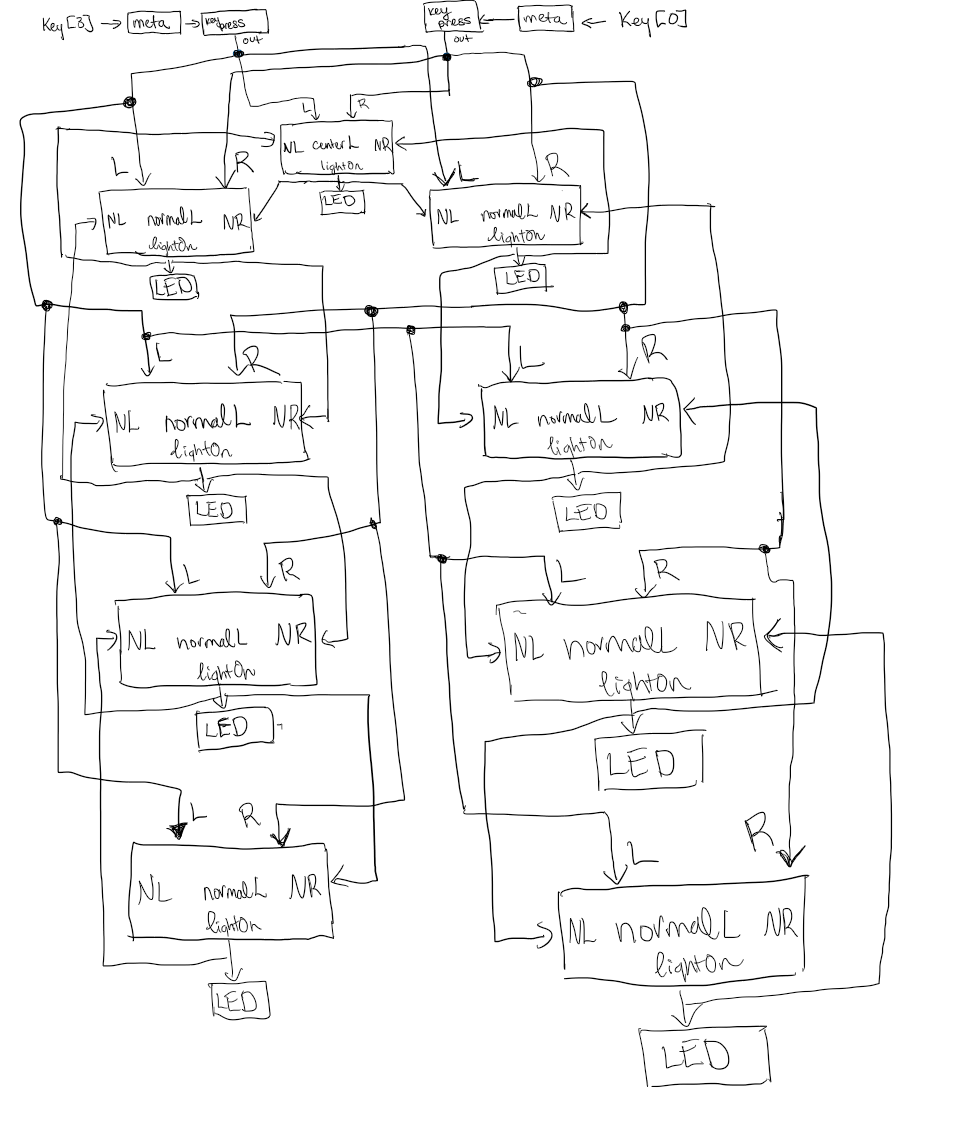
Approaching this problem, I first drew up the block diagram to figure out how to connect each FSM in order to produce the correct output for each user input that is possible. I drew the connection for the meta and keypress module to control the metastability of the user input and allows for any holding of the button to only count as a press. Next, it is connected to the center light and the light being on, sends out to the adjacent normal lights on the left and right side. This concept is applied to the other 8 normal lights surrounding the center one. Afterwards, I drew up the state diagram for each FSM in order to capture the state conditions that each FSM needs. The state diagram then allows me to transfer the FSM on picture to code. Simulating each module and connecting towards the FPGA board.

Figure 1: Block diagram for Tug of War System

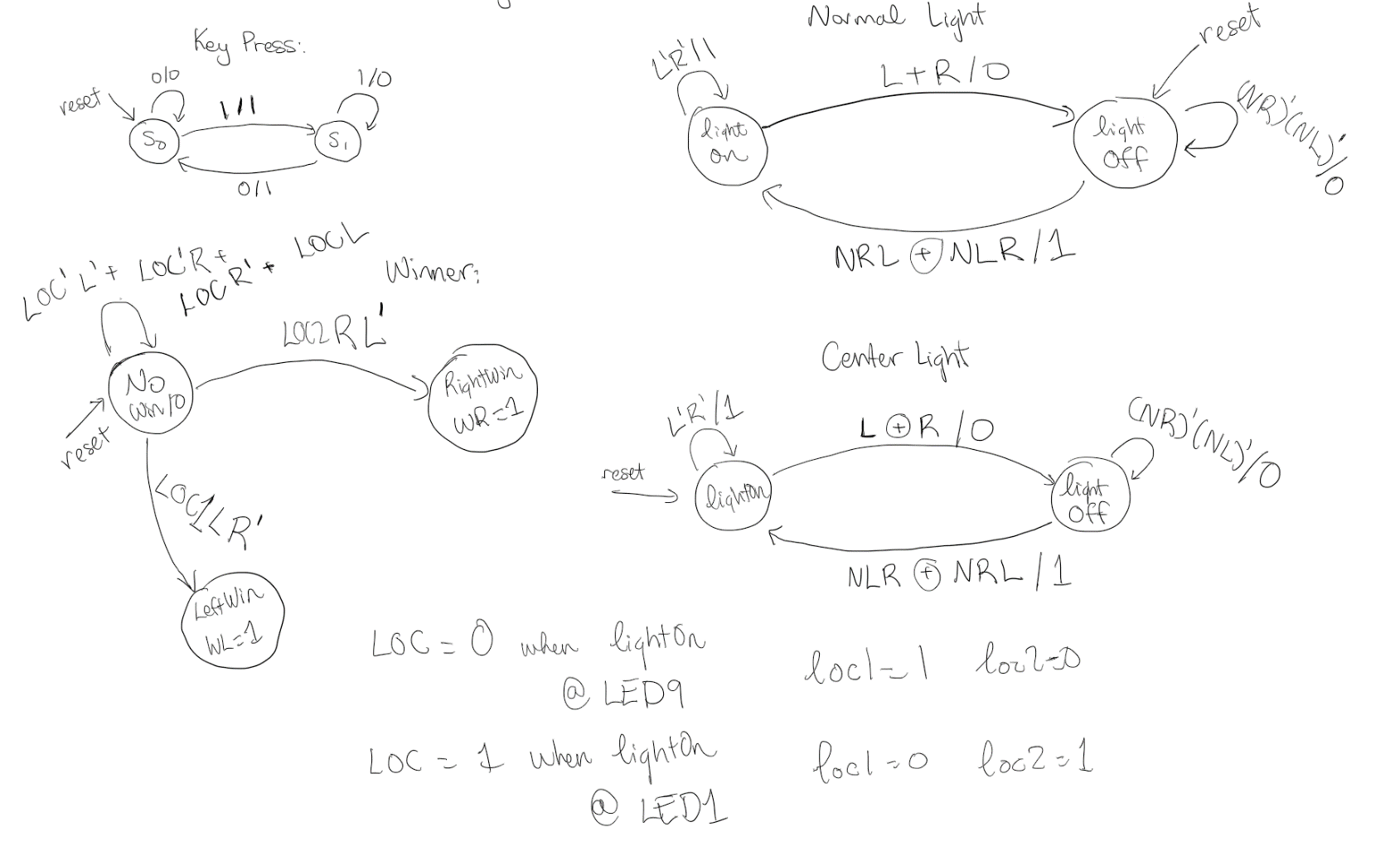


Figure 2: State diagram for each FSM in Tug of War system

**Results**

**Meta:**

For the first part, I tested the meta module to check if the metastability works with the input from the user as it allows for a two-clock cycle delay of the inputs. Also, the input and output progresses if the clock is at the positive edge.

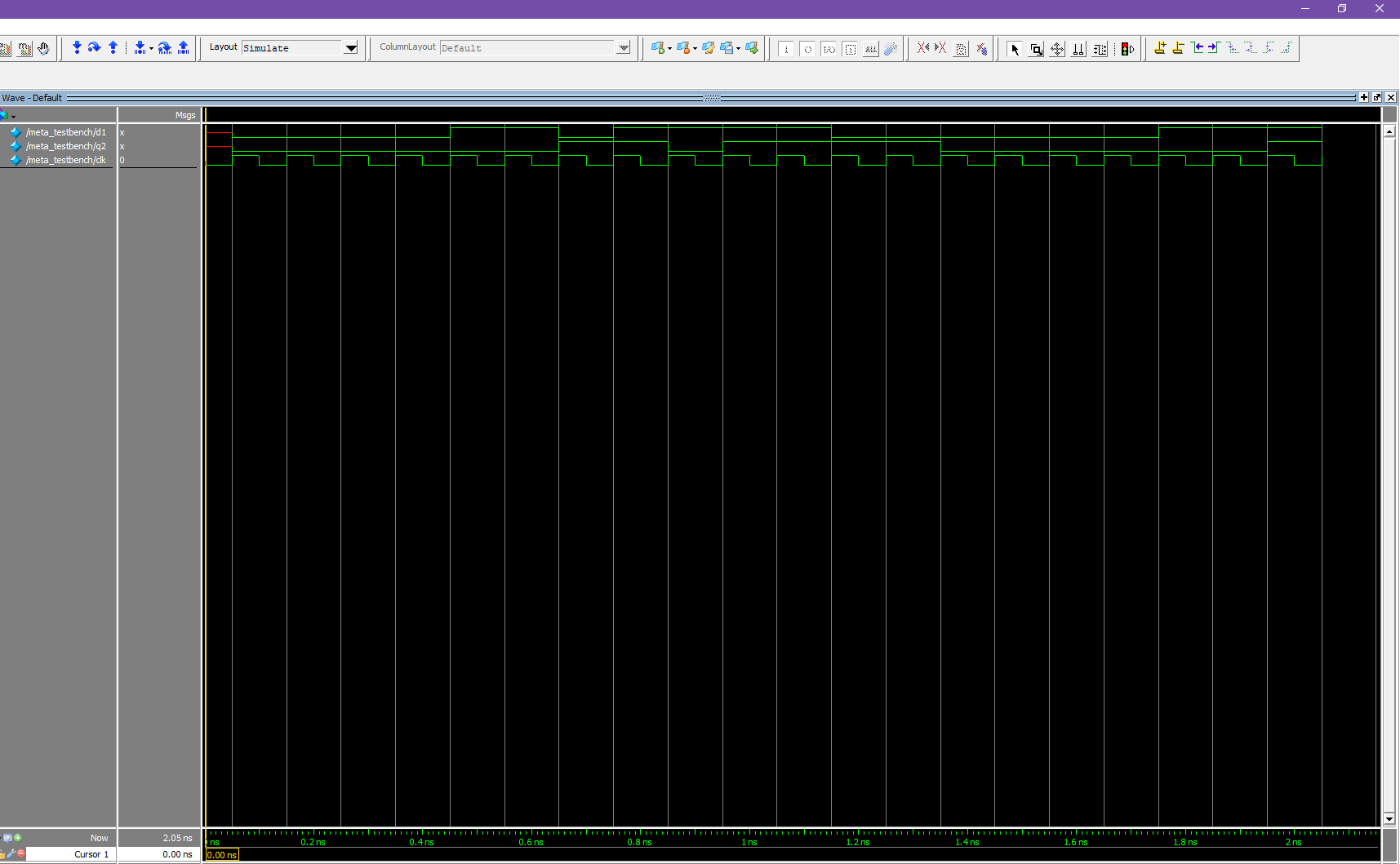
****

Figure 3: The waveform generated by the meta module for DFF

**Keypress:**

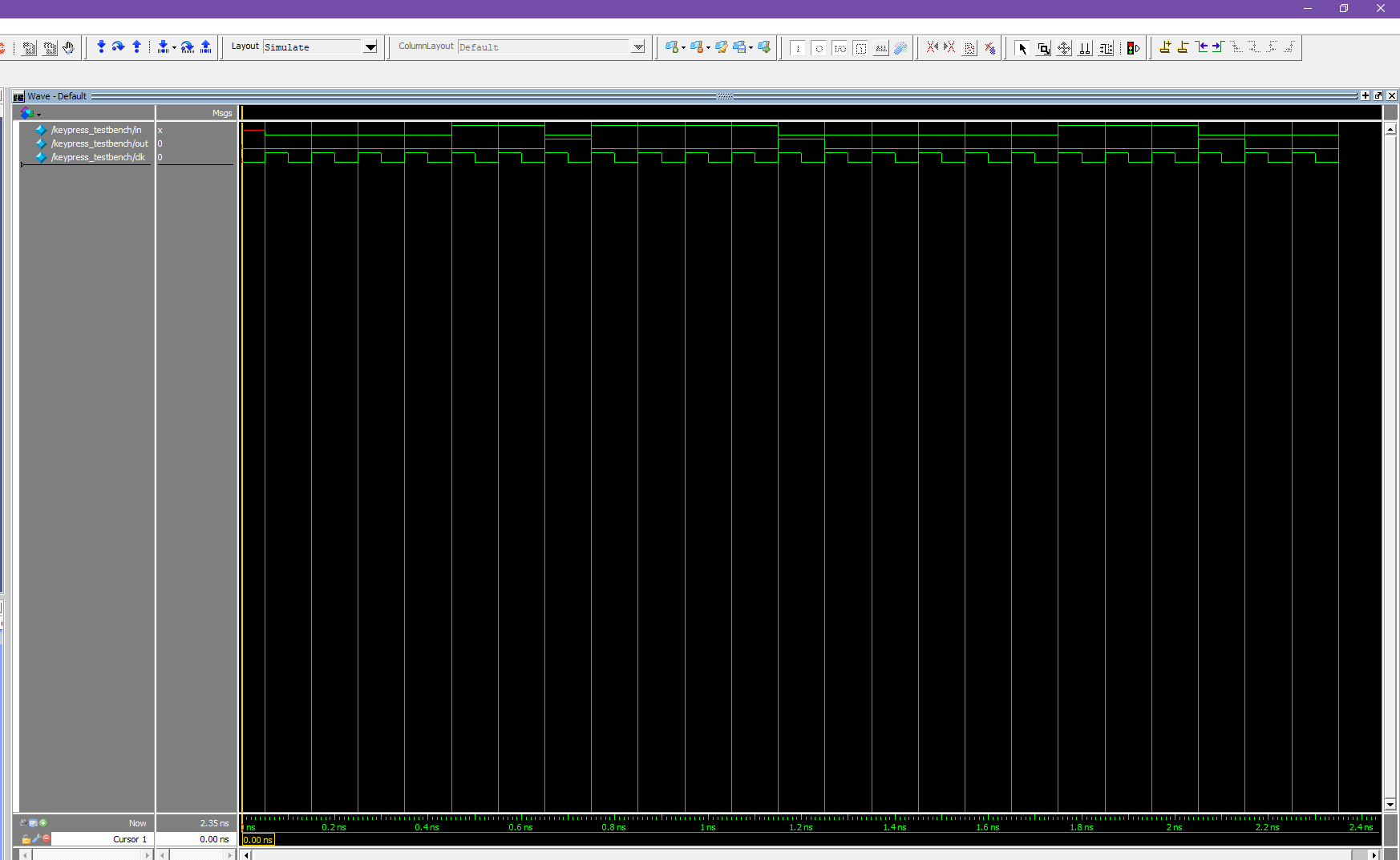
This simulation tested the inputs of in with the output of out. This FSM controls the input to allow a hold of the button as a press and forcing the user, not to hold the input. The inputs and outputs only take effect when the clock hits the positive edge.

Figure 4: The waveform generated by FSM of keypress

**normalLight:**

This simulation tested the output of the lightOn with the inputs of L, R, NL, and NR. This FSM allows for the light to be off when reset is called but turns on when either L and NR is true or when R and NL is true. When the light is on, either call to L or R would turn the light off. The inputs and outputs only take effect when the clock hits the positive edge.

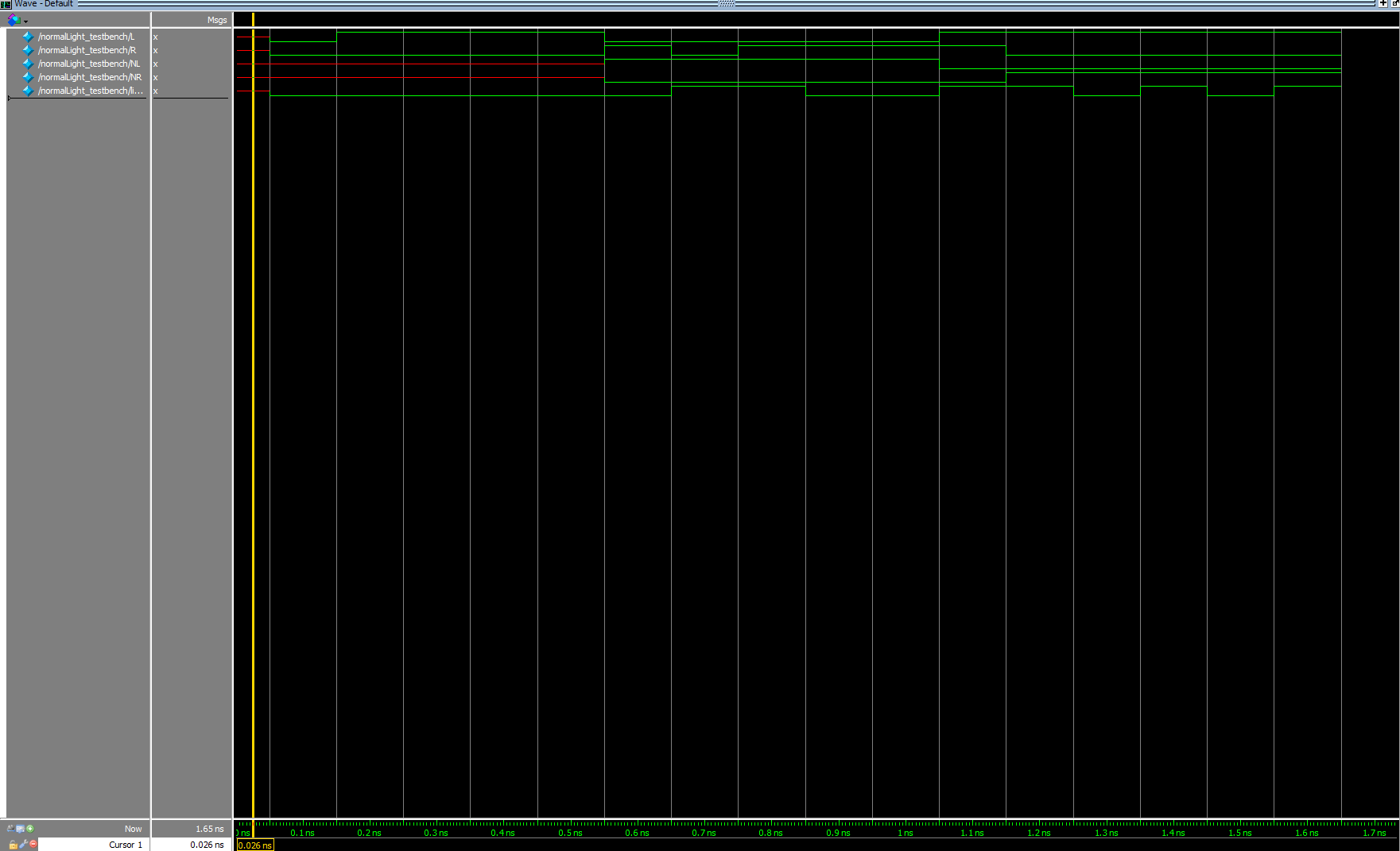
****

Figure 5: The waveform generated by FSM of normalLight

**centerLight:**

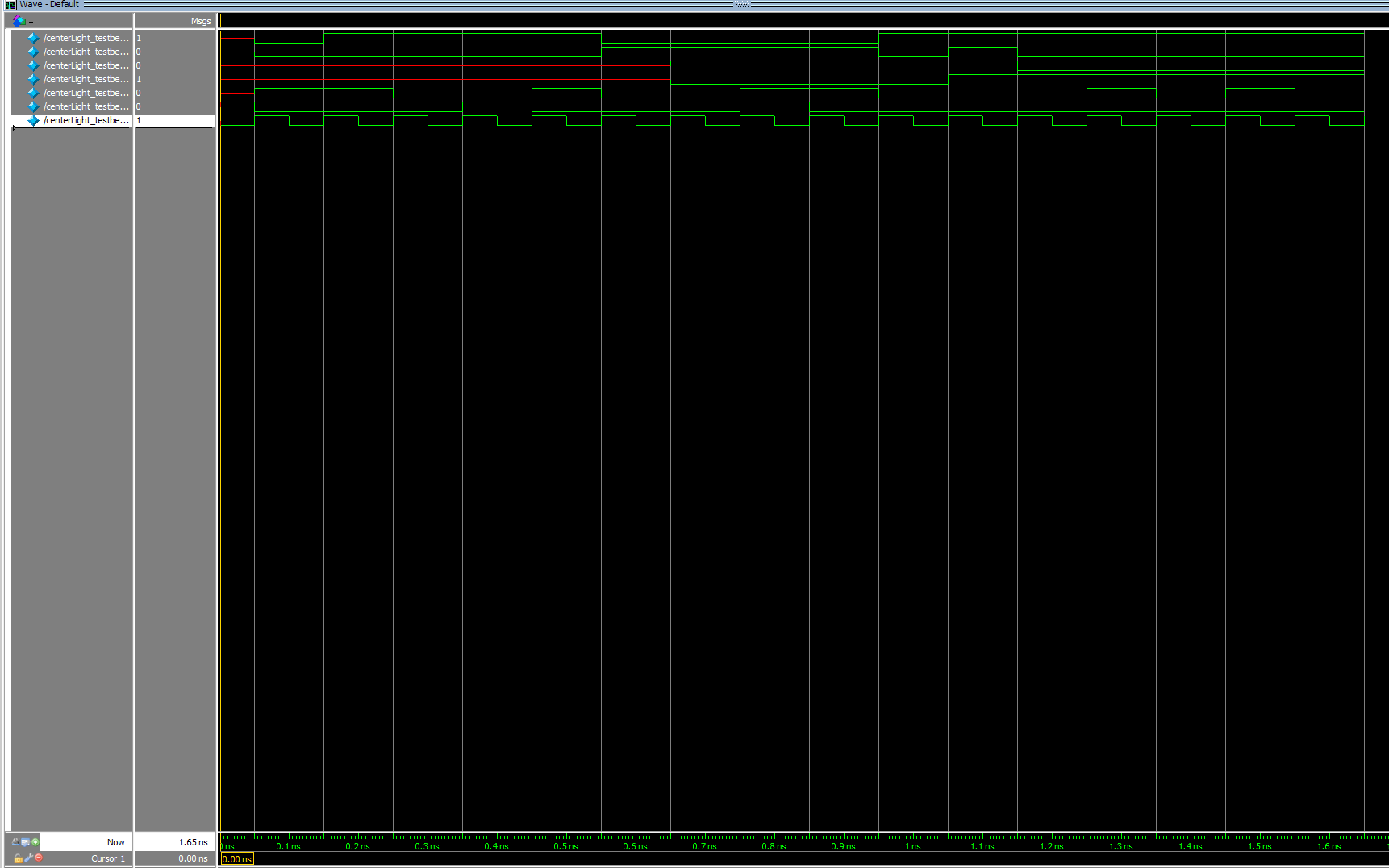
This simulation tested the output of the lightOn with the inputs of L, R, NL, and NR. This FSM allows for the light to be on when reset is called but turns on when either L and R is called. When the light is off, a call to L and NR is true or R and NL is true, allows for the light to turn on. The inputs and outputs only take effect when the clock hits the positive edge.

Figure 6: The waveform generated by FSM of centerLight

**Winner:**

This simulation tested the output of out as the 7-bit binary output of the HEX board display. The inputs are L, R, loc1, and loc2. The loc1 and loc2 are variables to indicate if either LEDR[9] or LEDR[1] is on to test if when the light are in those positions, how the calls on L and R have an effect on the out. The inputs and outputs only take effect when the clock hits the positive edge.

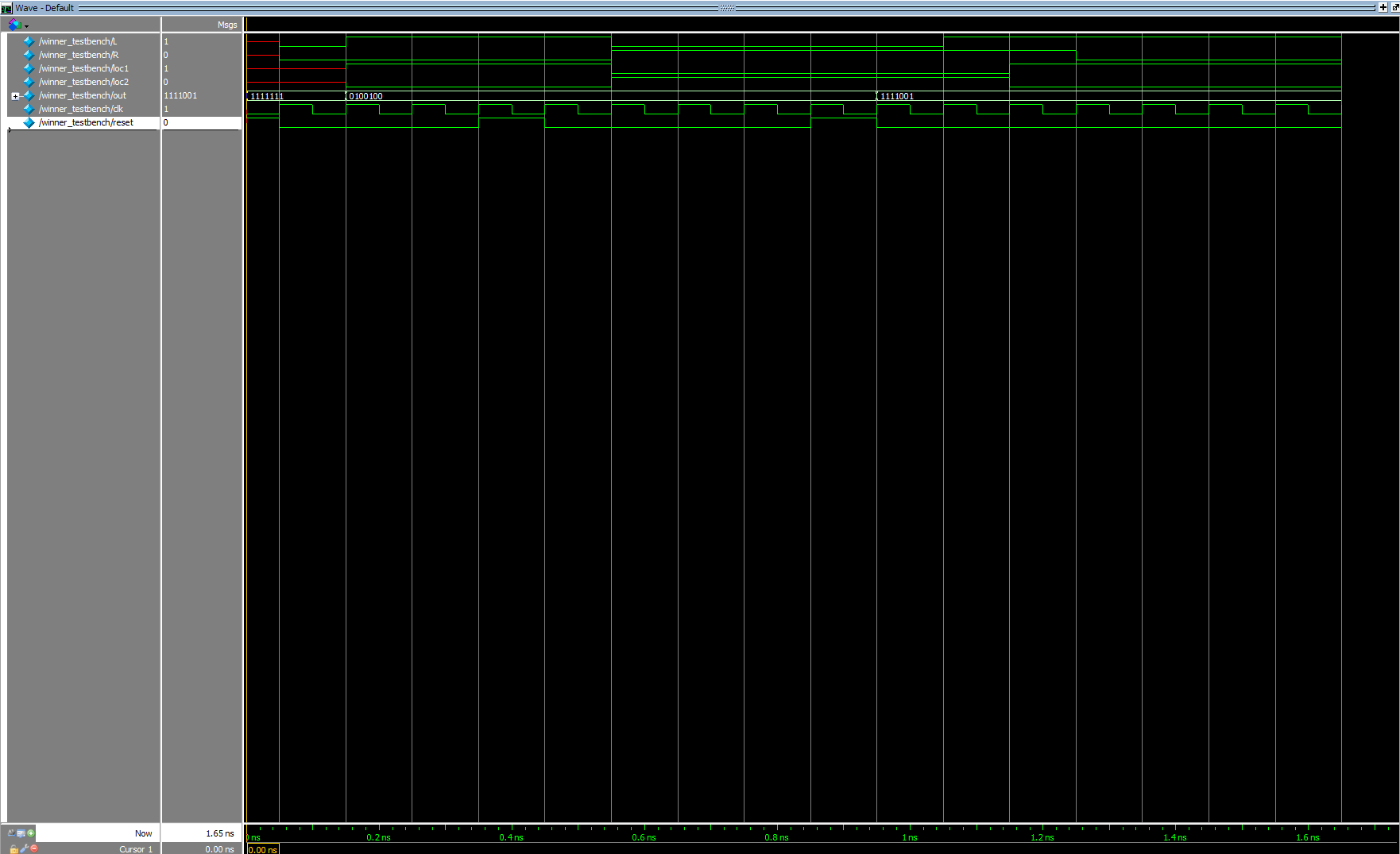
****

Figure 7: The waveform generated by FSM of winner

**DE1\_SoC:**

This simulation tested the inputs of the KEY[3] and KEY[0] on the outputs of the different LEDR and HEX0 display. When the inputs of KEY[3] and KEY[0] reaches either LEDR[9] or LEDR[1] and the right input is entered, the HEX0 displays either 1 or 2 for the winner of the first or second player. The inputs and outputs only take effect when the clock hits the positive edge.

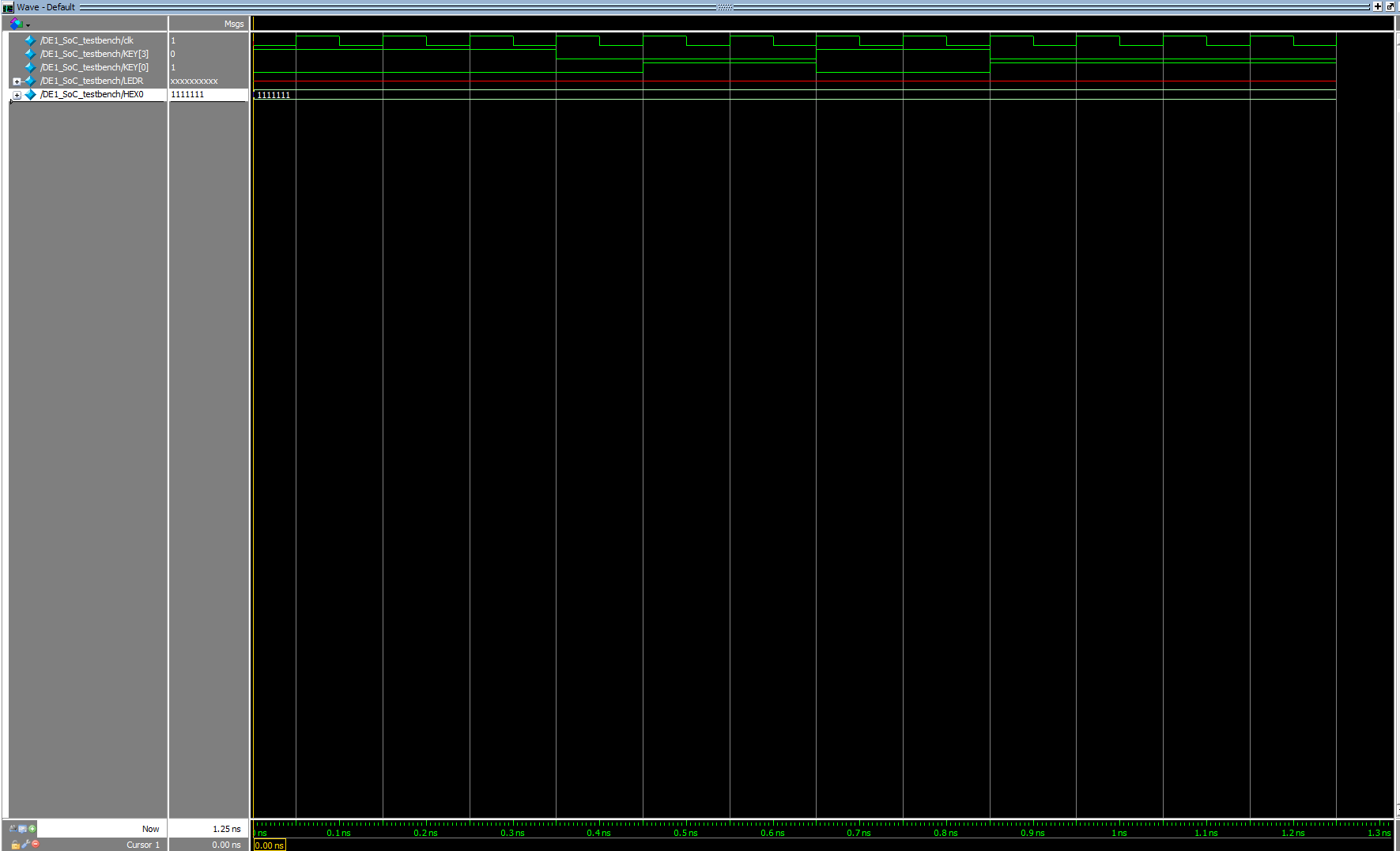


Figure 8: The waveform generated by DE1\_SoC

**Size:**

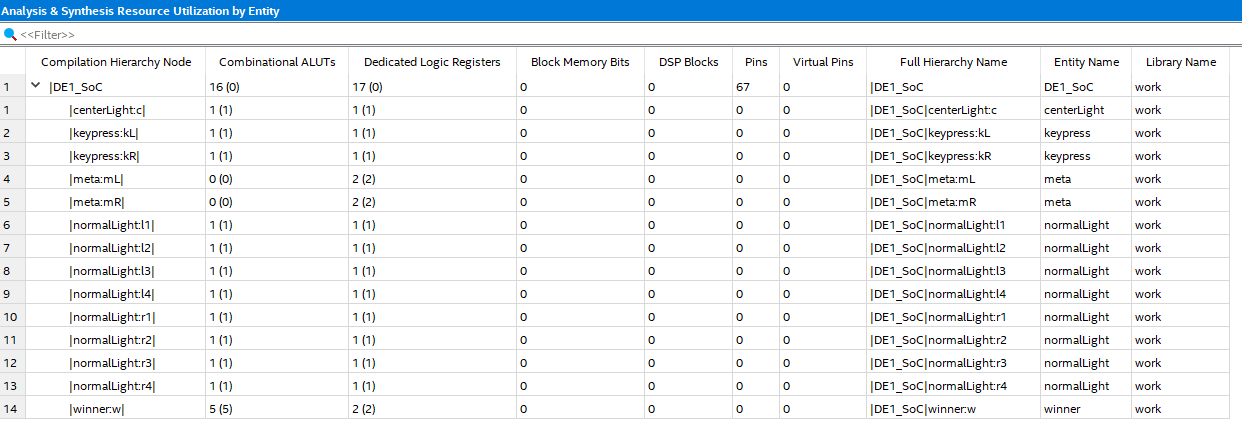
The size of system is 16+17, which equals 33. This is the size of the design in terms of FPGA logic and DFF resources.

Figure 9: Analysis and Synthesis Resource Utilization of the DE1\_SoC module

**Final Product**

Overall, this project was designed to learn how to develop a system that implements a design of Tug of War using a series of FSM along with D-Flip-Flops for metastability with the constant user input. The project asked to develop a system to control the metastability of the input as well as how an input was perceived. Afterwards, the focus is on developing the transition of the LEDR with the user input. I implemented this part with controlling the center light and each of the other normal lights separately and allowing for the input of each light to be from the output of the lights adjacent to it. This allows to track the current state that the light is on and allows for transition with the user input. From what is asked, this project implements all facets of the Tug of War game, allowing the user to reset to the original state as well as controlling metastability of the input with the 50MHz clock. Overall, there are no complications with this project as it allows for seamless play of Tug of War with the inputs from the FPGA board.

**Appendix**

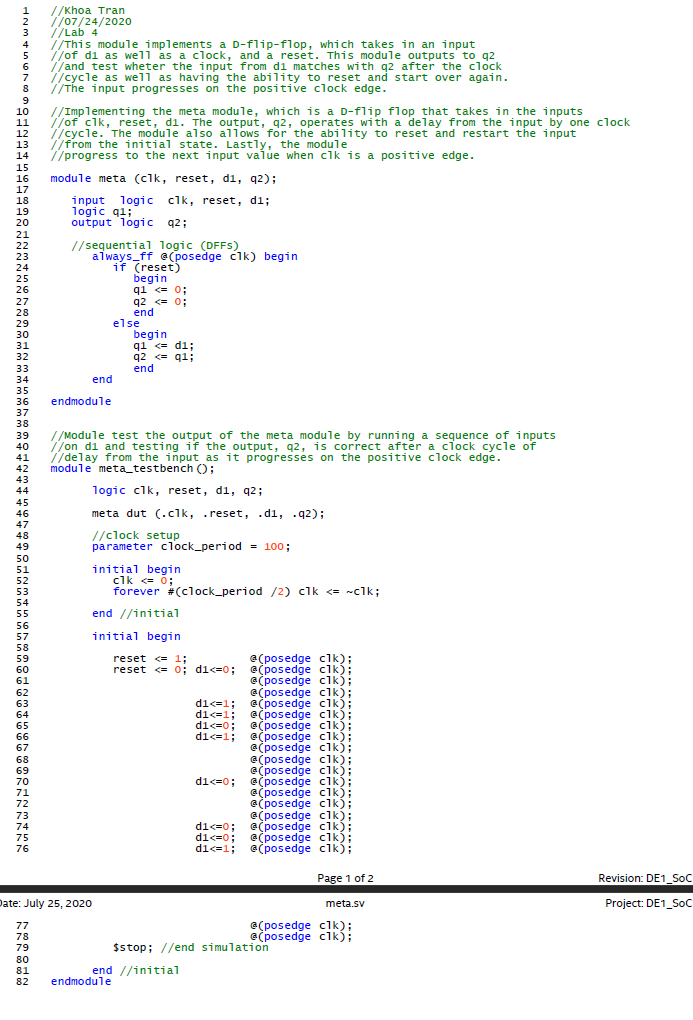


Figure 10: The Meta module

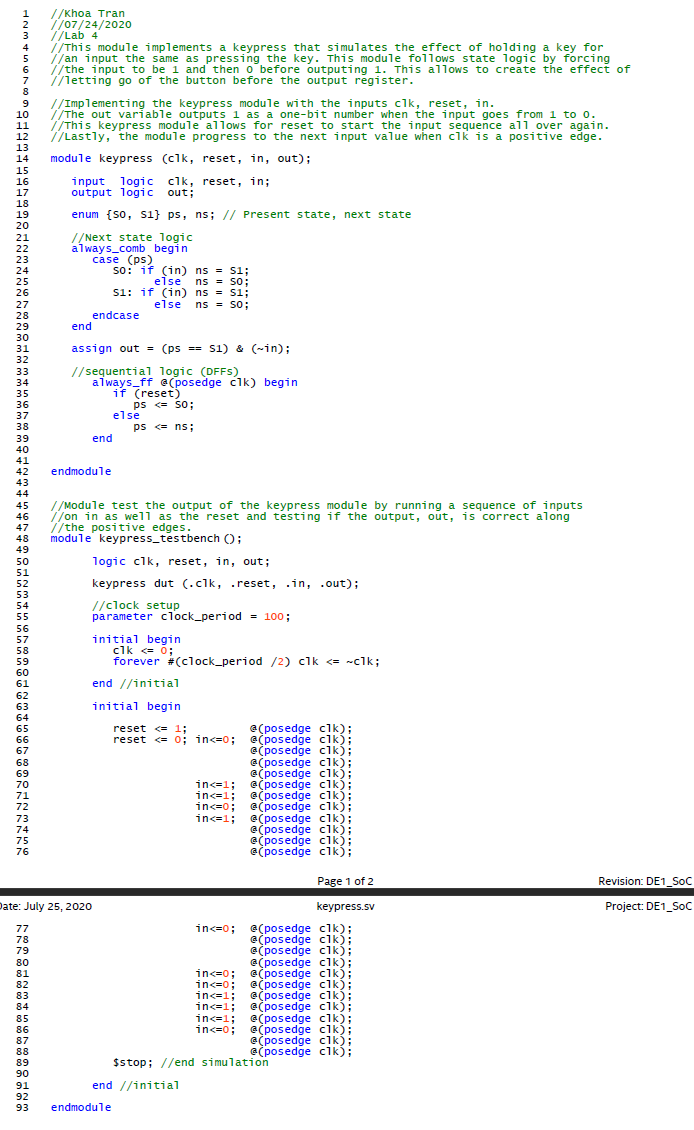


Figure 11: The keypress module

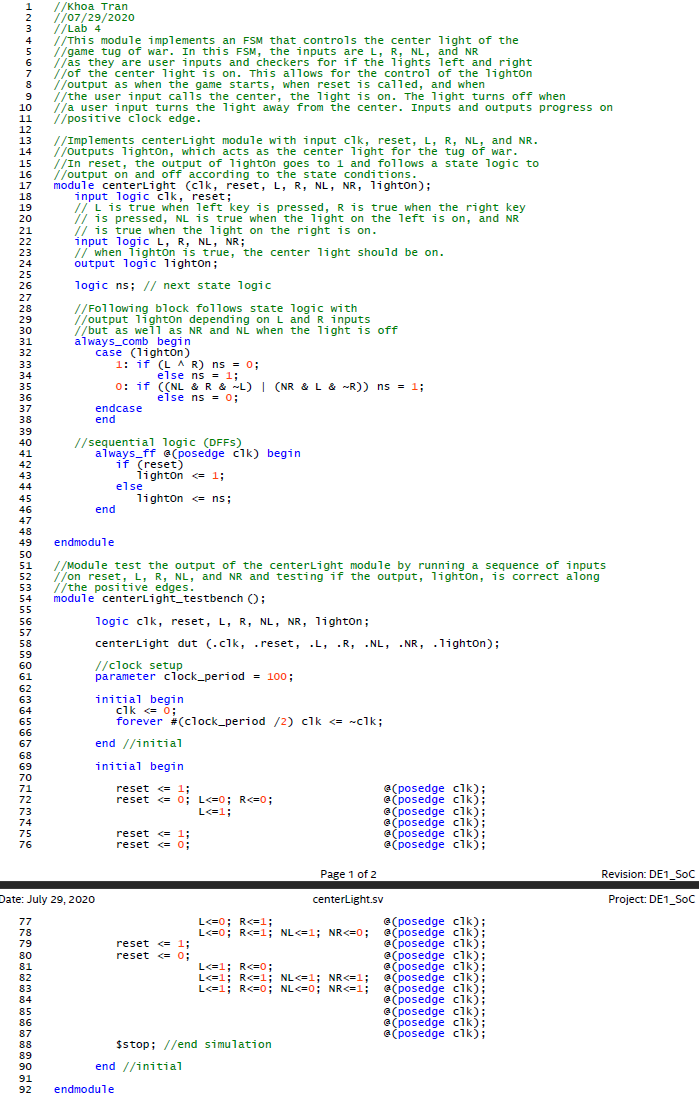


Figure 12: The centerLight module

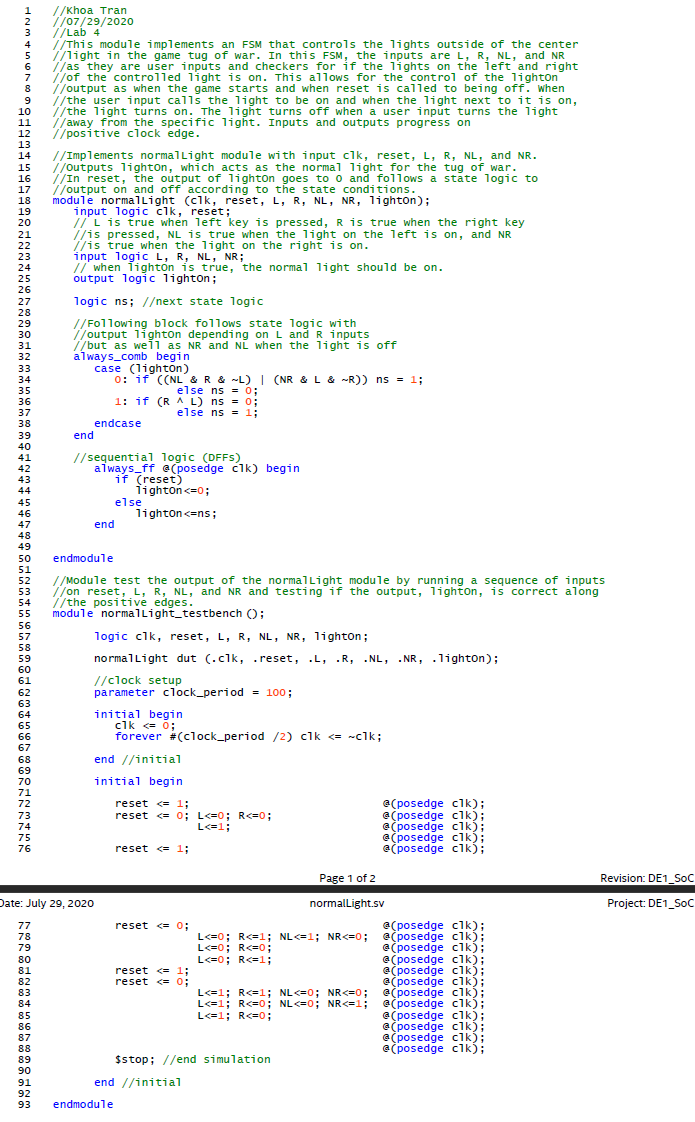


Figure 13: The normalLight module

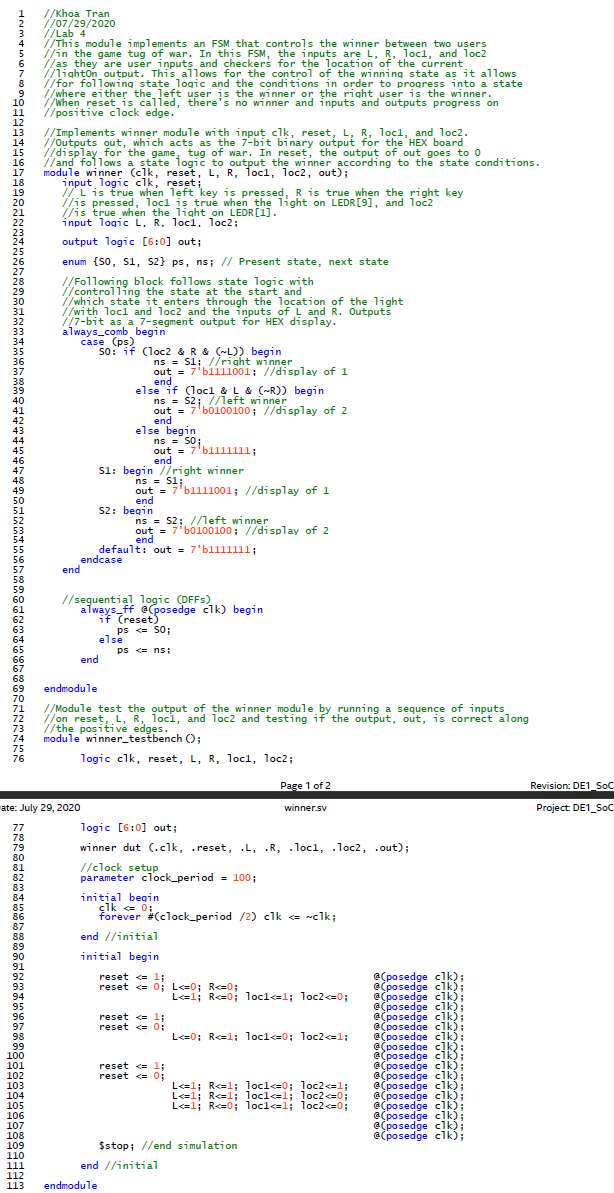


Figure 14: The winner module

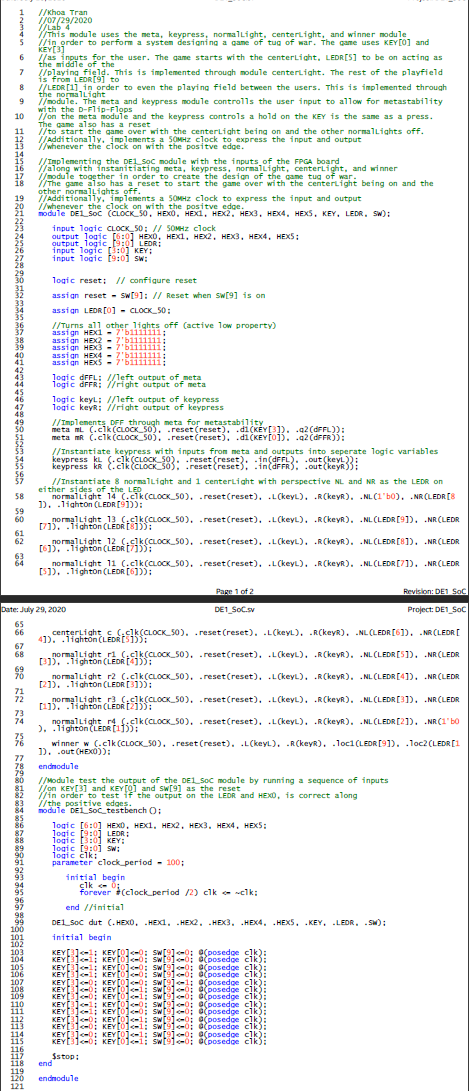


Figure 15: The DE1\_SoC module