Santa Clara University



Lab #7: Slot Machine Inspired Game

ELEN 21L 51306 Tuesday 2:15-5pm

May 23, 2017

Group 2

Angela Hwang

Yutong Li

Lab #7: Slot Machine Inspired Game

1. **Objectives:**

* Learn to use the design wizard to create logic components
* Learn to use a comparator and a counter in a circuit
* Use a multiplexor to switch multi-bit input

1. **Introduction:**

In this lab, we will build a slot machine, in which the player can press the push button to stop the “random” number displayed, and if the two separate displays match, the player wins, else, the player loses. For the display numbers, we just make them 0 or 1 or 2 or 3 instead of special symbols. For the result states, we just let d indicates win, E indicates lose, and f indicates in progress. Also, the game has four speeds: slow, medium, fast and test. For test, you can think this is “really slow” because we use this speed to test the circuit. In order to tell the player the result, there is a display showing whether the player wins, loses or unknown when the push button hasn’t been pressed.

1. **Procedure:**

**Part 1 - Create new components**

1. Create a 4-bit comparator that will create 3 outputs: equal, greater than and less than. We are going to use the comparator to compare to two 2-bit numbers that we generate from the 4to1 muxes. Because we are comparing only 2 bits, so we change the input from the default to [1..0], and the property of the inputs are the outputs from the 4to1 muxes.

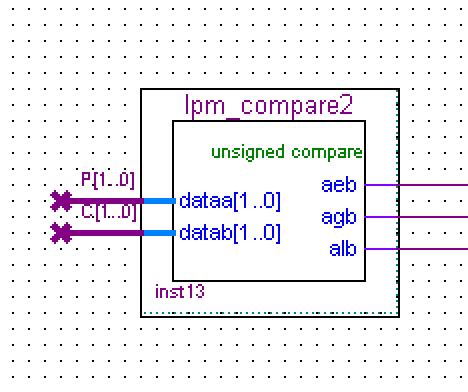


Figure 1. 4-bit comparator

1. Create an 8-bit binary up counter and an 8-bit binary down counter with a clock enable. The clock for both counters are the component “clock\_counter”, whose input should be the system clock50.

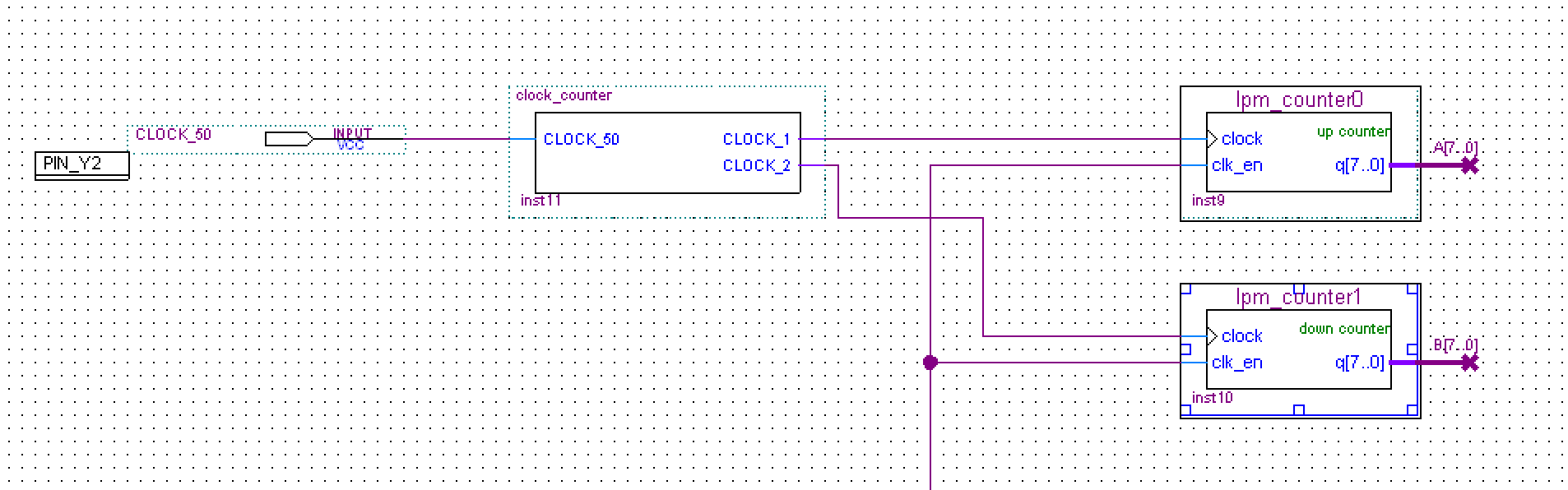


Figure 2. Clock counter, up counter, down counter, and their connection

**Part 2 - Connect your circuit**

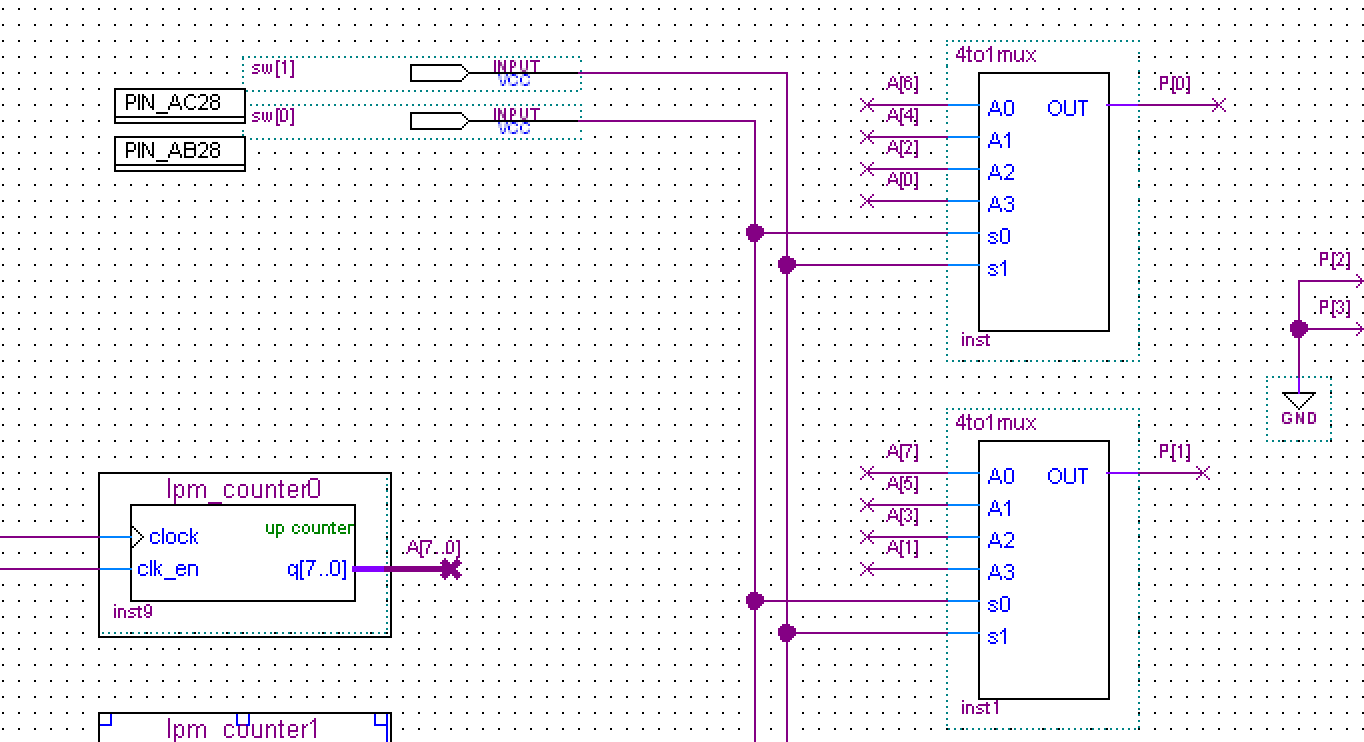
1. The selected bits from the 8-bit counters will provide distinct groups of 2 bits of the 4-bit input to the game display symbol generators. We set the switches for the 4to1mux such that 11 is fast, 10 is medium, 01 is slow and 00 is test. In fast mode, the outputs will be q0 and q1. For medium mode, the outputs will be q2 and q3. For slow mode, the outputs will be q4 and q5. For the test mode, the outputs will be q6 and q7.

Figure 3. The connection that we have for the up counter

1. Next we connect the first 2 4to1 mux to a spec\_7seg to show the random generated number. The spec\_7seg has 0s as the first 2 bits because we only have two bits from the outputs of the muxes, and the sepc\_7seg expects 4-bit input. We connect the remaining 2 4to1 mux to another spec\_7seg with the same technique.
2. Then we connect to comparator to the circuit to compare the two 2-bit inputs of the two game display symbol generator.
3. Next we need to come out with the logic expression for the inputs for the display as a function of pushbutton and the comparator equal output to show the “win(d, which is 13 in decimal, and 1101 in binary)”, “lose(E, which is 14 in decimal, and 1110 in binary)”, or “in progress(f, which is 15 in decimal, and 1111 in binary)”. We can see that all of these states has 11 in the first two bit, so the spec\_7seg has 1s as the first 2 bits. For the remaining two bits, we come out with a truth table shown in Figure 4, where P is push button, E is equal, N is not equal, which is the negate of equal, R1 is the second least significant bit, R0 is the least significant bit, and X is don’t care.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P | E | N | R1 | R0 |
| 0 | 0 | 0 | X | X |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | X | X |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Figure 4. Truth table of the display of the win or lose or in progress state

From the K-map that we create from the truth table in Figure 4, we have

R1 = P + E’

R0 = P + E

So the connection between the comparator and the spec\_7seg will look like what is shown in Figure 5.

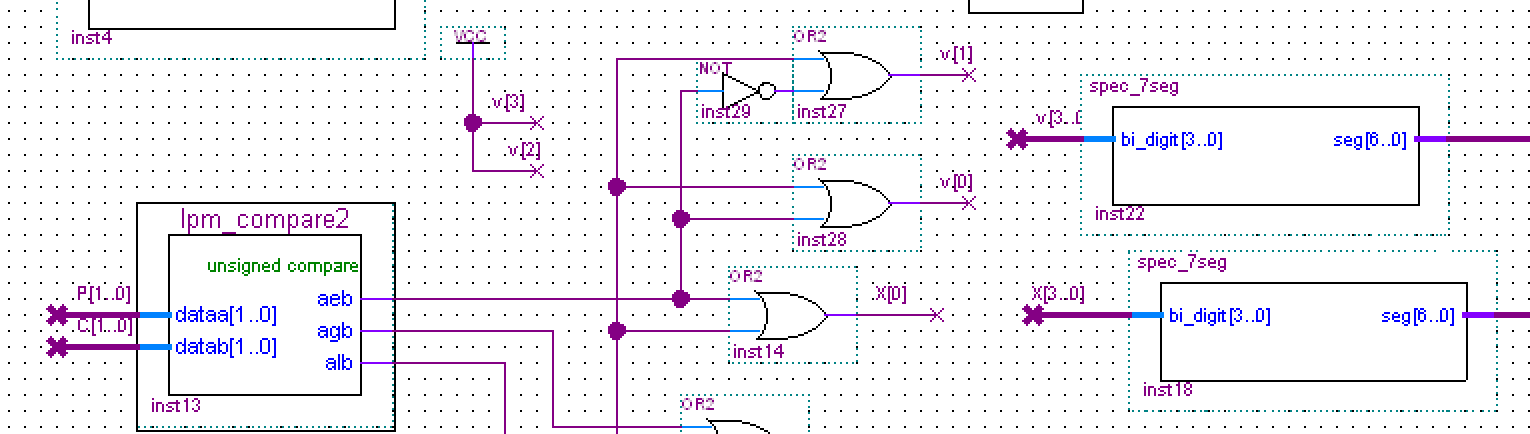


Figure 5. Connection between the comparator and the spec\_7seg

**Part 3 - Test your circuit**

The board works like this:

1. The counters are not counting when the push button is pressed.
2. Both counters are counting when the push button is pressed.
3. The up counter shows 0, 1, 2, 3, 0, 1, 2, … and the down counter shows 0, 3, 2, 1, 0, 3, 2, …
4. When the push button is pressed and the two displays match, the result state is d which is win.
5. When the push button is pressed and the two displays don’t match, the result state is E which is lose.
6. When the push button is not pressed, the counters will keep counting and the result state is f which is in progress.
7. When switches are 00, which is test, the speed is really slow. When switches are 01, which is low, the speed is a little faster. When switches are 10, which is medium, the speed is even faster. When switches are 11, which is fast, the speed is crazily fast.

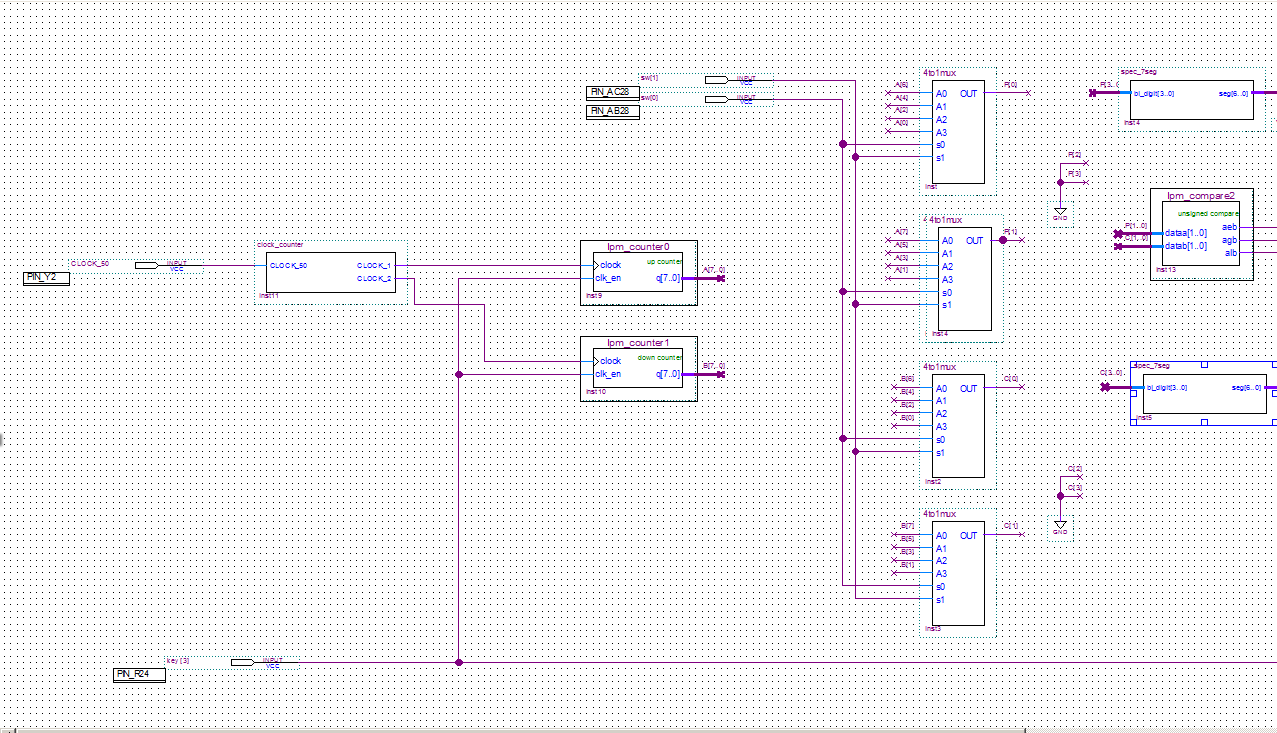


Figure 6. The zoomed left part of the circuit

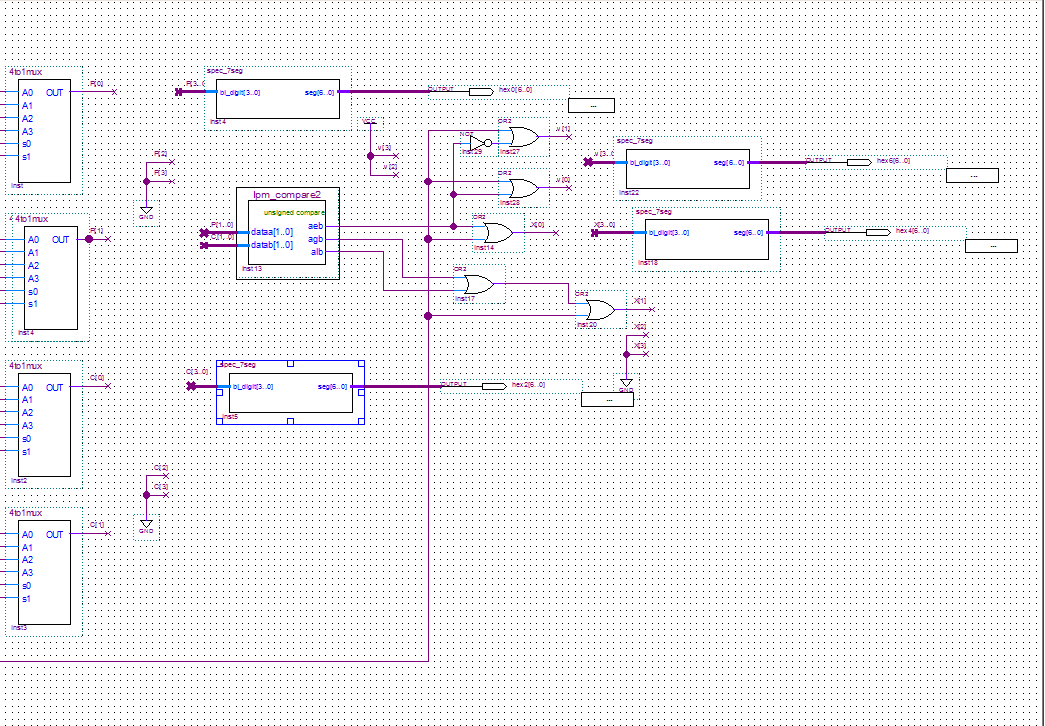


Figure 7. The zoomed right part of the circuit

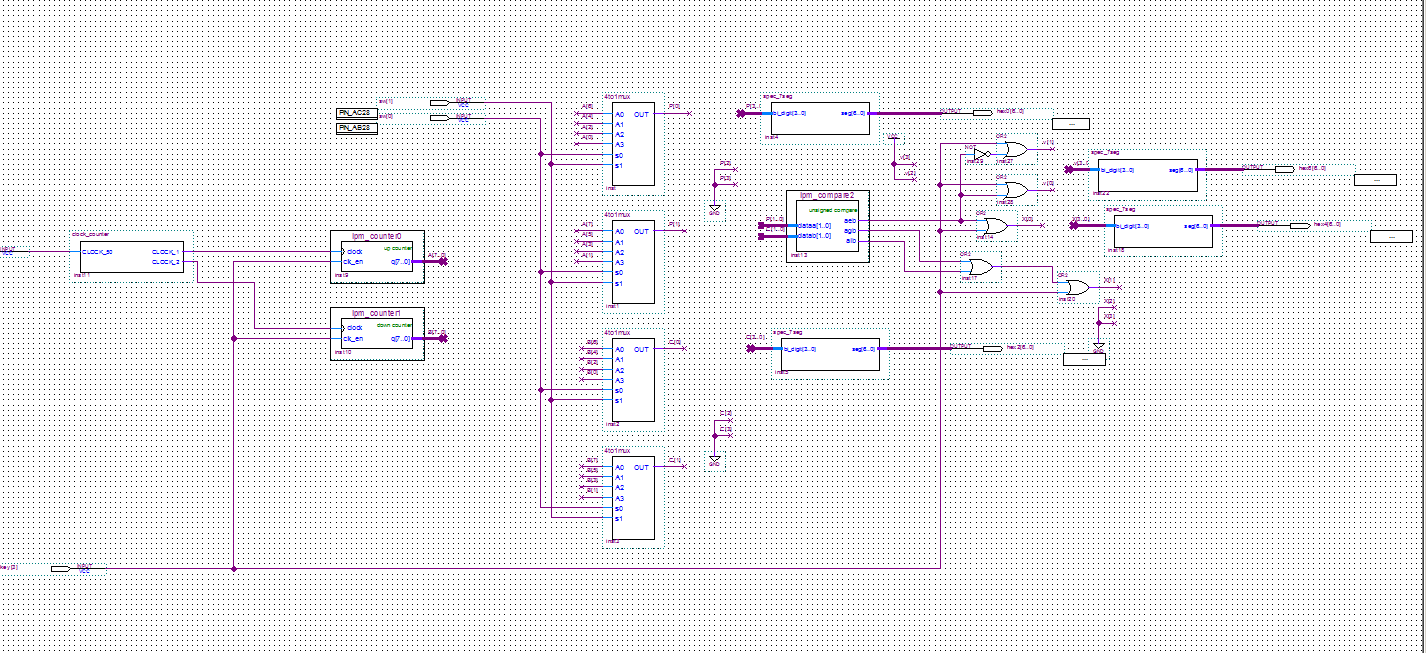


Figure 8. The whole circuit

1. **Final question:**
2. The two most significant bits should always be grounded, so they can always be 0, so the input to the special symbol generator is always 0, 1, 2, or 3.
3. First, If we use 8 symbols instead of only 4, we will have 3 valid bit, which mean only the first bit

or the most significant bit will always be grounded.

Second, we will need three 8to1mux, so we will need 3 switches.

Third, we will need more up counters and down counters in order to make the inputs for the muxes more random.

Fourth, for the comparator, the inputs will be 3 bits, which is [2..0]. What’s more, if it was me, I will just have two outputs, one is equal, and the other is not equal, or even just one output, because we can negate the output anyway.

1. **Conclusion:**

In this lab, we build a gambling machine. The purpose of this lab was to design and test a gambling machine. The counters will provide the inputs for the 4to1 muxes, the 4to1 muxes will provide the inputs for the comparator, the comparator will provide the inputs for the spec\_7seg through a designed logic circuit, and the spec\_7seg will tell the result of the gambling game. Overall, we accomplished all of the objects illustrated in the beginning of the lab instruction.

1. **Reference lists:**

* Dr. Sally Wood, Dr. Samiha Mourad, Dr. Radhika Grover. *Laboratory #7: Slot Machine Inspired Game.* Spring 2017. Print
* Bin\_7seg\_temp.txt. Spring 2017. Print
* Dr. Sally Wood, Dr. Shoba Krishnan. *7-Segment Display Tutorial.* Spring 2017. Print
* Altera Corporation - University Program. *Quartus II Introduction Using Schematic Designs.* Spring 2017. Print
* Altera Corporation - University Program. *Quartus II Introduction to Simulation of Verilog Designs.* Spring 2017. Print
* Terasic Technologies Inc. *DE2-115 User Manual.* Spring 2017. Print.