**ECE 4250/ 7250: VHDL and Programmable Logic Devices**

**Laboratory**

**Lab #: 02**

**Lab Title: Sequential Process Design: BCD Counter**

**Group #: 5**

**Names: John Kelly | Zach Rump**

**Teaching Assistant Use Only:**

**Points Earned Reasons for Deduction**

**Pre-lab:**

**Post Lab report**:

**Demonstration:**

**Final Lab Grade:**

**Comments to students:**

Objective

The objective of Lab 02, Sequential Process Design: BCD Counter is to make students familiar with the sequential process design by modeling a counter VHDL process and verify the functionality of the model by using ModelSim.

Questions

1. In part a, step 2, what is the value of Co? Justify your answer?

The carry out (Co) is set to 1 when the COUNT transitions from 9->0 and from 0->9.

In figure 1 below, the Co is set for two clock cycles because the count is incremented from 9->0 and then immediately decremented from 0->9 on the next cycle.

1. If the counter current state is 0, and you have UP=0 (decrement)? How do you handle this case, explain?

If the counter is decremented while the current counter state is 0 then the counter will be assigned the value of 9 and the carry out will be set. In the VHDL code this is implemented as an extra if statement inside the decrement logic. (An extra 2-1 MUX)

3. In part b, when the counter current state is 00, and you have UP=0(decrement)? What is the expected output?

In this case the expected output will be “99” and Co will be set. This can be seen in figure 2 below, although in our implementation this doesn’t work correctly and the counter goes from “00” to “09” and then “98.”

Outputs and Results

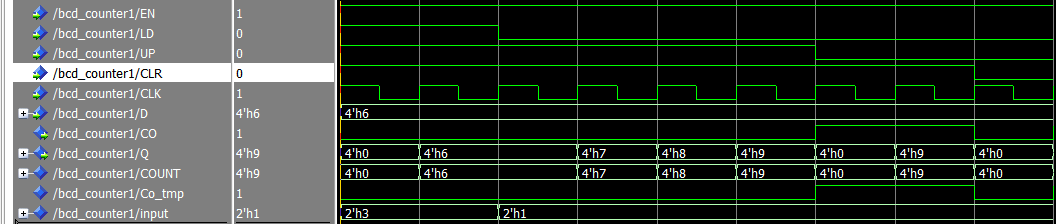


Figure 1: Shows the test waveforms associated with the single digit BCD counter in part one.

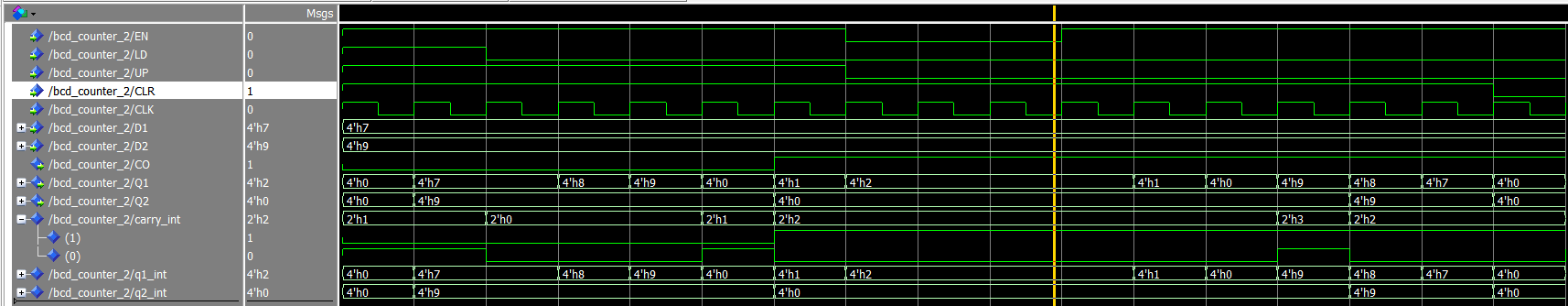


Figure 2: Testing waveforms for part two of the lab. Note that when decrementing from “00” the sets the carry out correctly but skips from “00” to “09” to “98.”

Higher resolution screenshots and code at: <https://github.com/nihilanth41/ece4250_lab2>

Conclusion

Part A was rather straightforward. The directions were clear and easy to follow. By modeling the instructions as VHDL statements, the BCD counter operated as expected. While the functionality was proper, the program was reevaluated and made more efficient by initially setting the Co to ‘0’ and only setting it when needed ( ‘1’ ), as opposed to specifying the result in each case. Error was encountered while implementing part B. Initially, the carry out was not properly enabling the second BCD and therefore only incrementing 00-09 rather than 00-99. First, it was observed that the port mapping arguments were incorrect. After making these changes and further evaluating the progress the carry out was still being set inside the process. This was moved so that the carry out would show up and actually enable the second set of bits in the BCD.

Another issue that we encountered had to do with misunderstanding how to actually use the simulator to test the circuit. We initially tried to force the internal count variable when loading the number into the counter, as opposed to driving the external signals to load the count. This misunderstanding propagated into the second part of the lab, as we didn’t realize initially that the EN input for the second BCD digit needed to be forced HIGH when doing the initial load.

Ultimately the lab proved helpful for becoming more familiar with the ModelSim design and simulation process and was likely necessary to refine our skills.