**ECEN 248 - Lab Report**

**Lab Number: 8**

**Lab Title: Counters, Clock, and Debounce Circuits**

**Section Number: 519**

**Student’s Name:** [Alex Allahar](mailto:alex.allahar@tamu.edu)

**Student’s UIN: 928009686**

**Date: 11/08/23**

**TA: Yi Deng**

**Objectives:**

The purpose of the lab is to build our knowledge of sequential circuits and introduce synchronous sequential circuits, and the binary counter. Also, the lab will demonstrate two important use cases for the binary counter, the clock frequency division and I/O debouncing.

**Design:**

The lab begins by creating the project lab8. The first design source that needs to be made is the clock\_divider. Using the code provided in the lab manual, design the clock\_divider. Add the constraint clock\_divider file in the course directory into the project. Synthesize and Implement the design.

In part 2 of the lab, start by building the up-counter. Using the starter code provided for up\_counter.v in the lab manual, create the design source for the up\_counter. Once designed add the up\_counter test bench file from the course directory into the project as a simulation source. Simulate the up\_counter using the test bench file. The next module to design is the top-level source which can be used to drive the counter on the FPGA and load onto the ZYBO Z7-10 board. First, change the clock\_divider module Count signal bit size from 4 to 26 bits wide. Then use the starter code in the lab manual under top\_level.v, design the top-level module. Next, create the top\_level constraint file using the code provided in the lab manual as a starting point. Once the top-level constraint is completed synthesis, and implement the top\_level. If no errors arise generate bitstream and load the design onto the FPGA board.

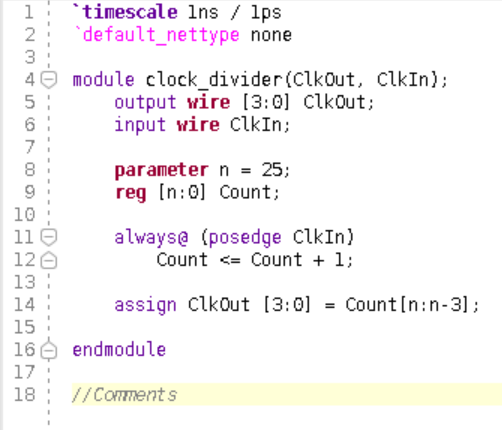
In part 3 of the lab, the effects of the button bounce will be demonstrated. Add the Bounce test bench file into the simulation sources from the course directory. Add the noDebounce design file to the project from the course directory. Modify the name of uut in the test bench file to test noBounce. Simulate with a runtime of 3000ns. Add the withDebounce design source into the project from the course directory. Change the uut name back to withDebounce and resimulate, with withDebounce as the top.

**Conclusion:**

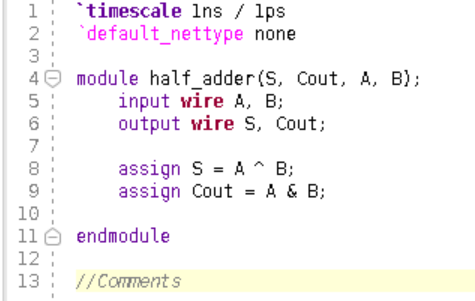
In this lab, I designed a clock divider, and an up-counter, and was shown the effects button bounce. Furthermore, I was able to use the FPGA to view the output of the counter.

**Post-lab Deliverables:**

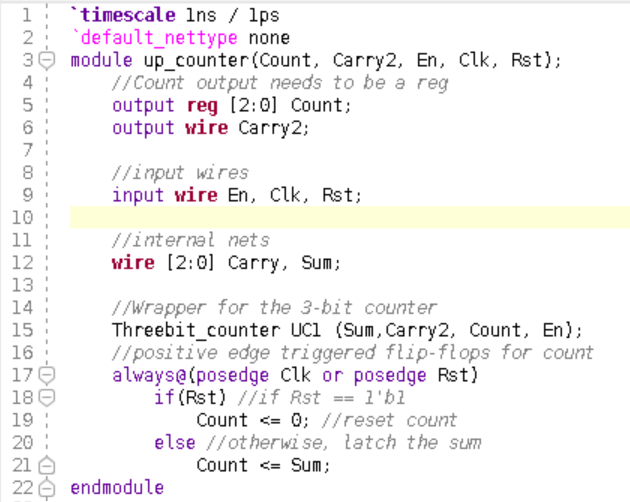
1. **Source Code:**

****

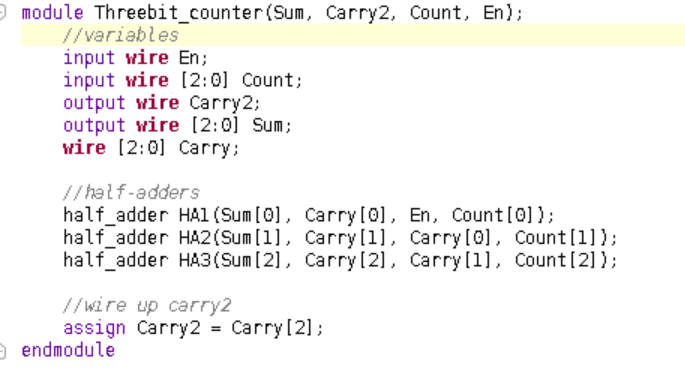
**Figure 1: Clock Divider Code**

****

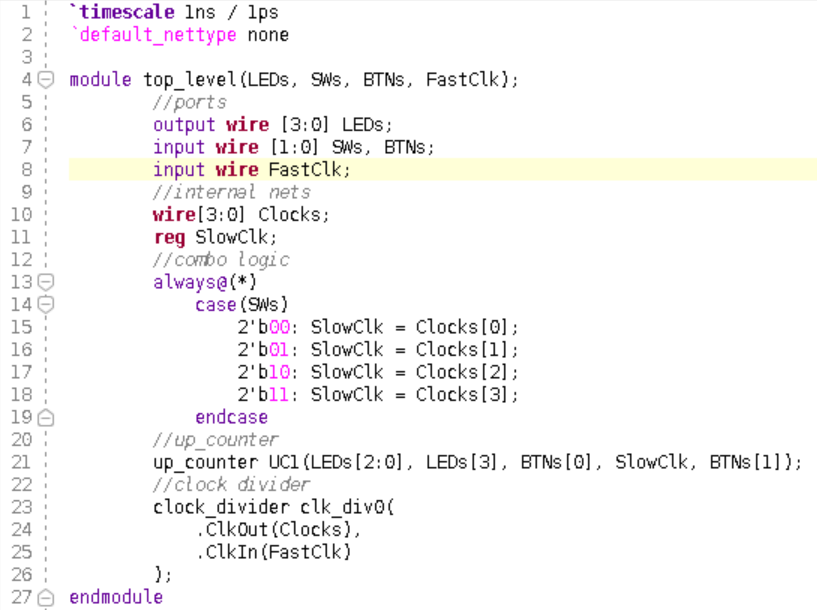
**Figure 2: Half-Adder Code**

****

**Figure 3: Up Counter Code**

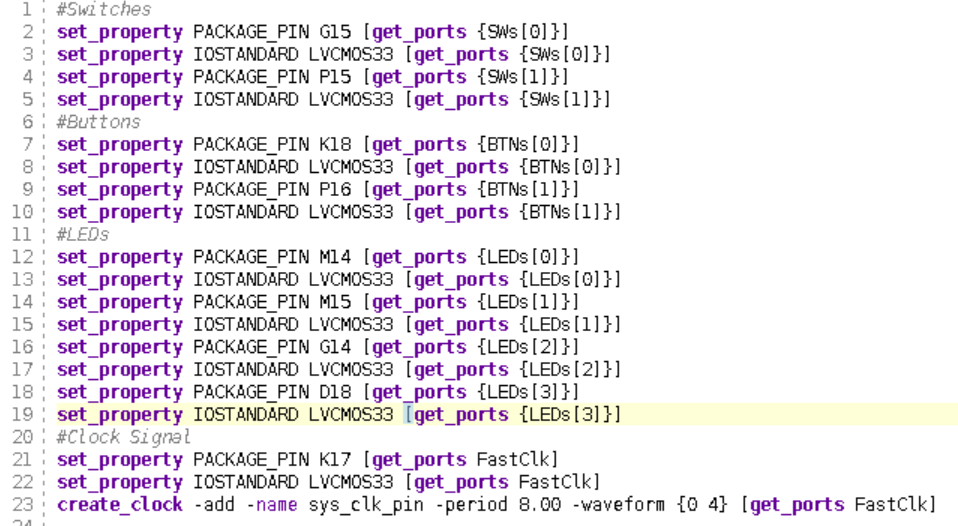
****

**Figure 4: Three bit counter Code**

****

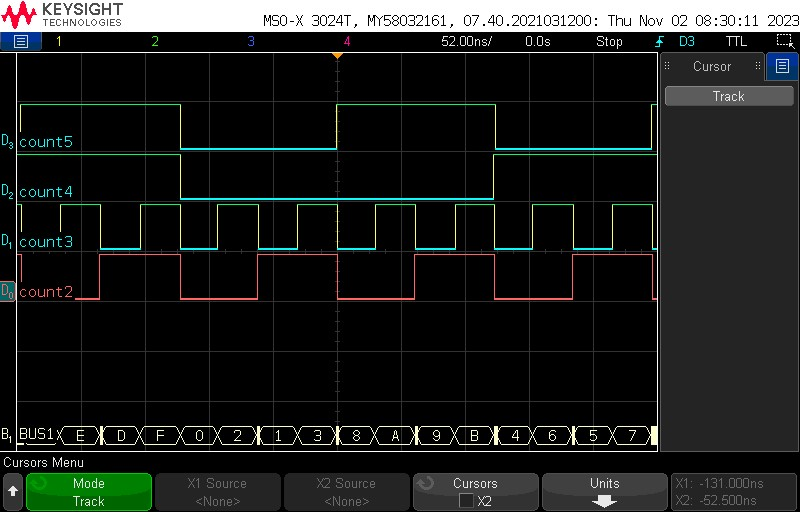
**Figure 5: Top Level Code**

1. **XDC Files**

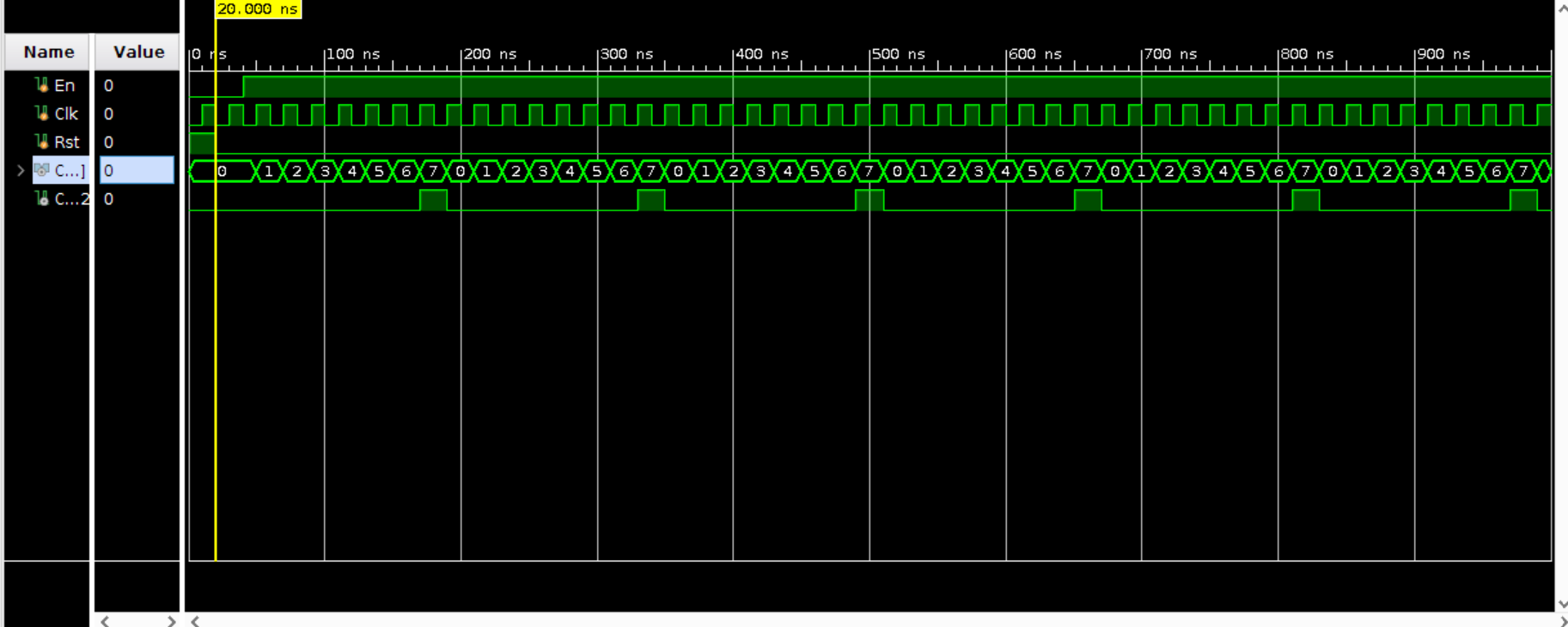
****

**Figure 6: Top Level Code (.XDC)**

1. **Waveforms**

****

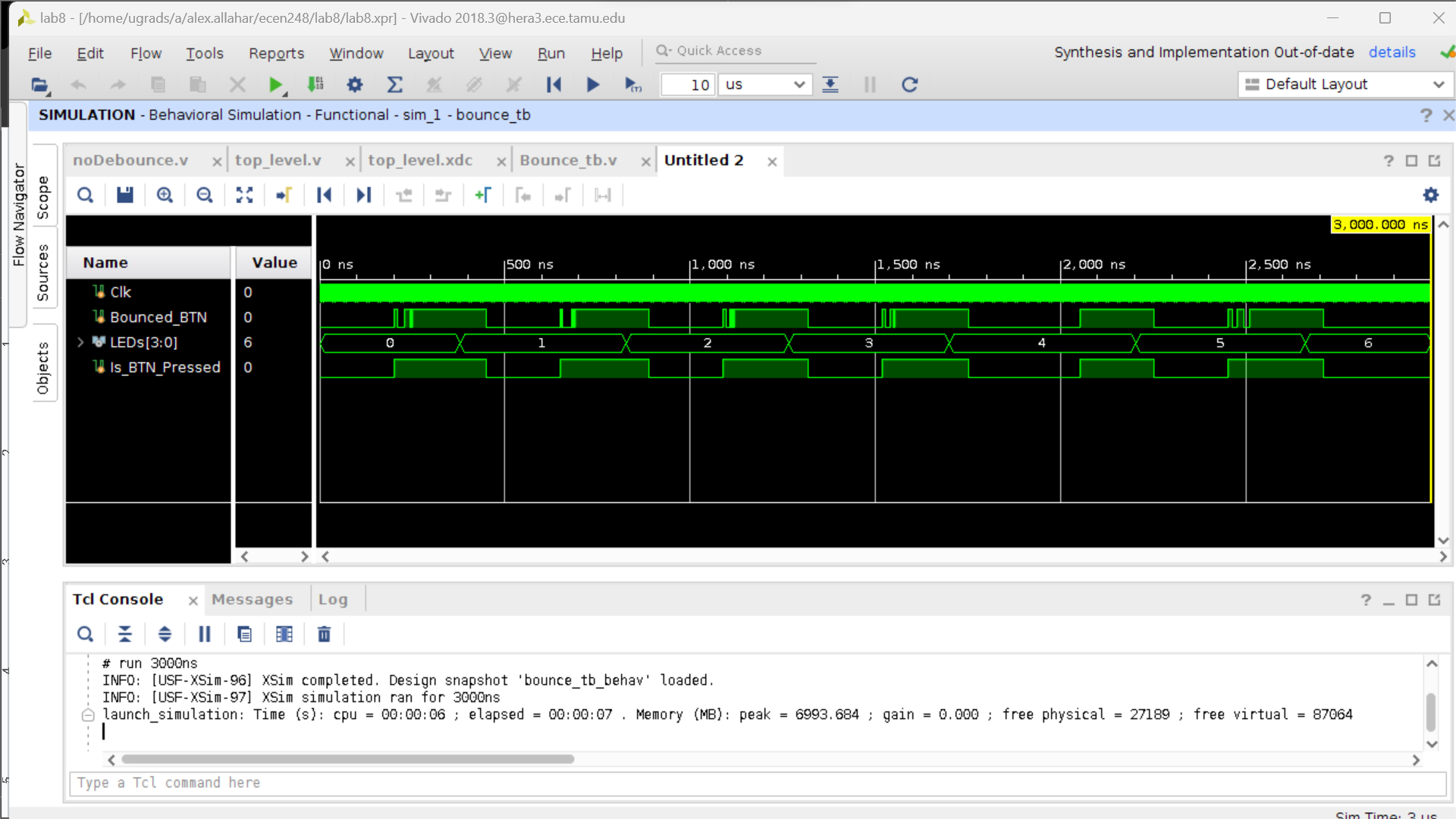
**Figure 7: Waveform of Clock Divider with Count in Hexadecimal**

****

**Figure 8: Up Counter Waveform**

****

**Figure 9: Waveform with noDebounce**

****

**Figure 10: Waveform with Debounce**

1. **Questions**
   1. **Part 1, 4. a**

The count interval were different from the clock due to the clock division. For count 5 = 128 ns, count 4 = 255 ns, count 3 = 31 ns, and count 2 = 64 ns.

* 1. **Part 2, 2. b**

The Clk is assigned with 10 ns; thus the frequency is the 1/10ns = 100MHz

* 1. **Part 2, 2. c**

The interval of the reset for the counter is 20 ns.

* 1. **Part 2, 2.d**

The interval of the testbench holding LOW for enable is also 20ns.

* 1. **Part 2, 2. f**

One the waveform the count contains 3 bits, thus has a maximum of 111 before overflowing and returning to 000.

* 1. **Part 2, 3. a**

If the frequency is 125 MHz and by increasing to 26 bits for the Count signal, we divide the frequency by 2^26 which results in a frequency of 1862.7 Hz for the MSB.

* 1. **Part 3, 1. b**

The noDebounce waveform shows a higher count which is not in line with the amount of times the button is pressed. This does not work.

* 1. **Part 3, 2. a**

The withDebounce works as the count goes up by one when the button is pressed.

* 1. **Part 3, 2. c**

The withDebounce circuit synchronizes the button input with the clock and holds it for a complete cycle. Using the positive rising edge to change the count, ignores the button bounce.