**Lab 9: Counters, Clock Dividers, and Debounce Circuits**

Faizan Bangash

TA : Saurav Kumar Sahu

ECEN 248 – 302

July 25, 2018.

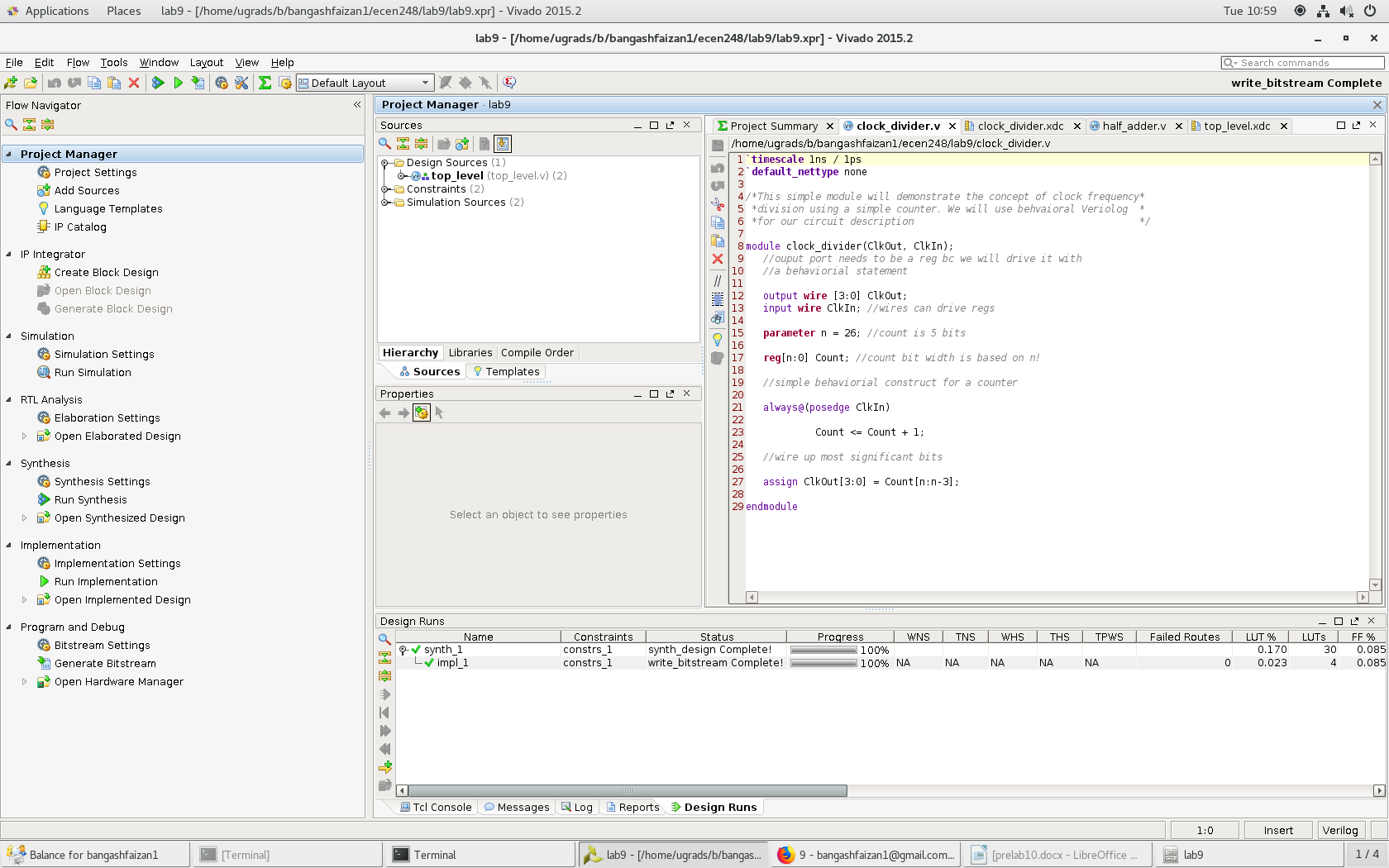
**Objectives** –

In this lab, we looked at more sequential circuits by implementing a binary counter made from half-adders and flip flops. Also, this lab looked at clock frequency and I/O debouncing. Additionally, we reinforced our knowledge of dataflow and structural circuits in verilog.

**Design** –

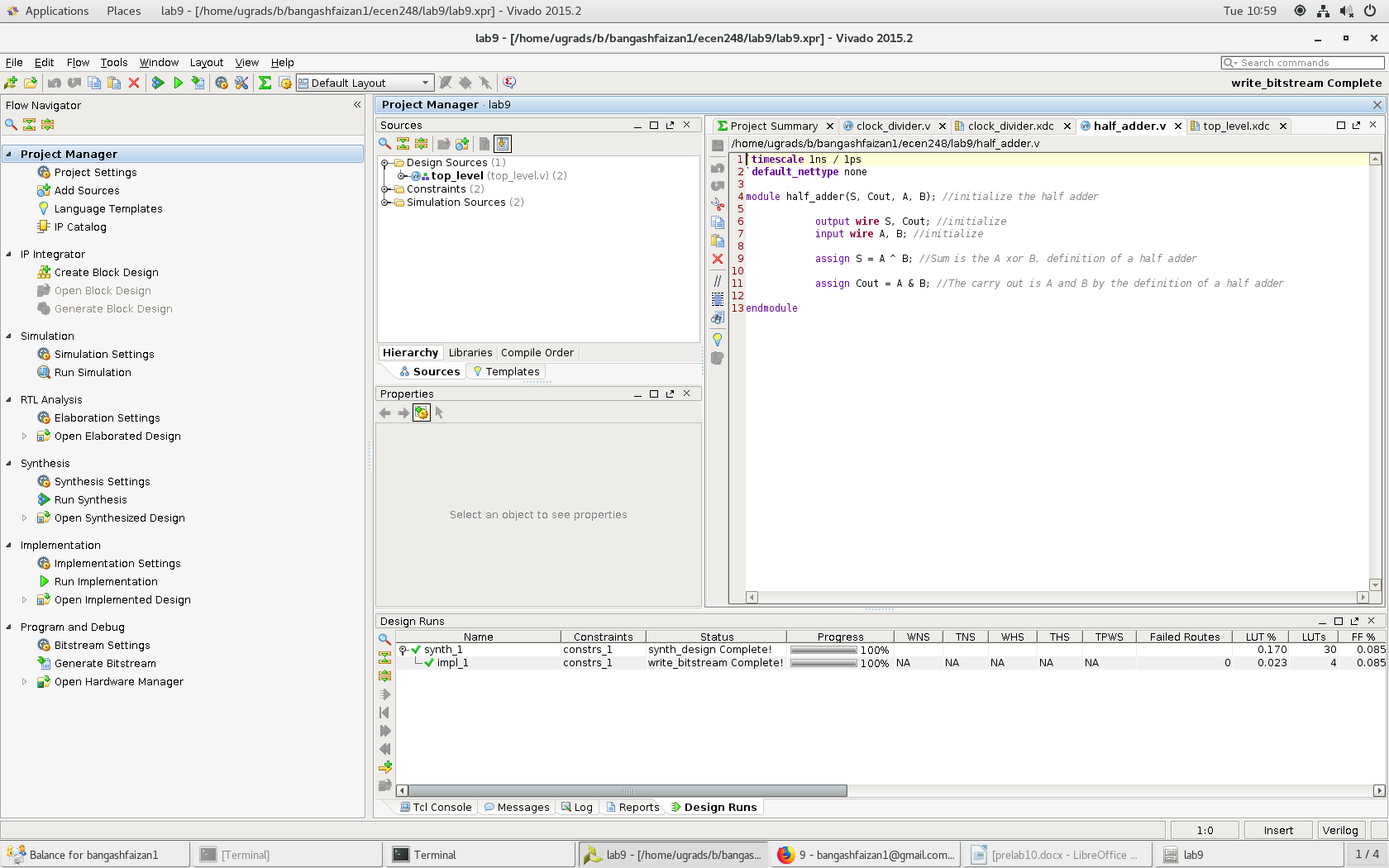
Part 1: For part 1 of the lab, we first made Clock\_divider.v which can be seen below. After this, we implemented it to the logic analyzer.

**Clock\_divider.v**

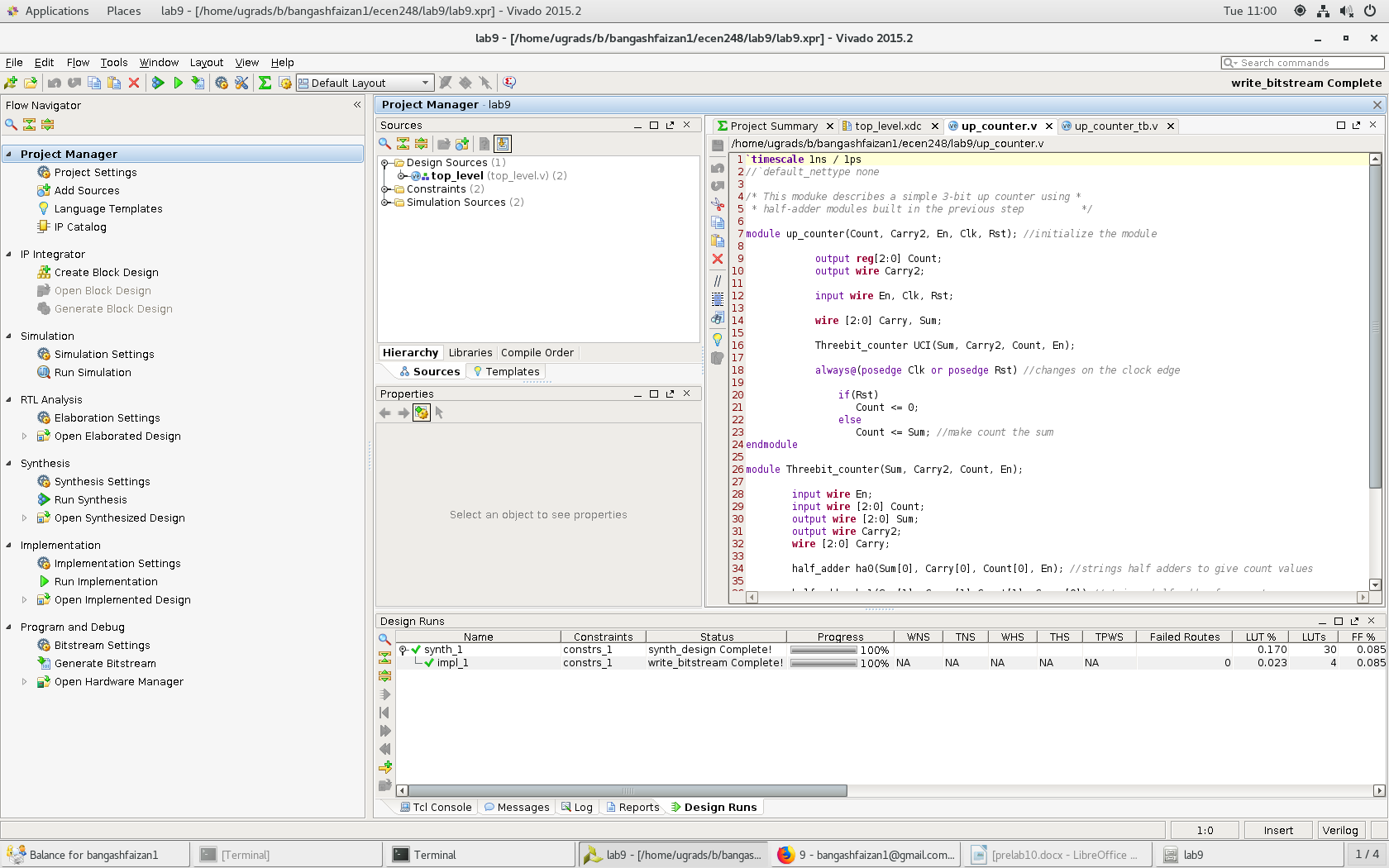


**clock\_divider.v**

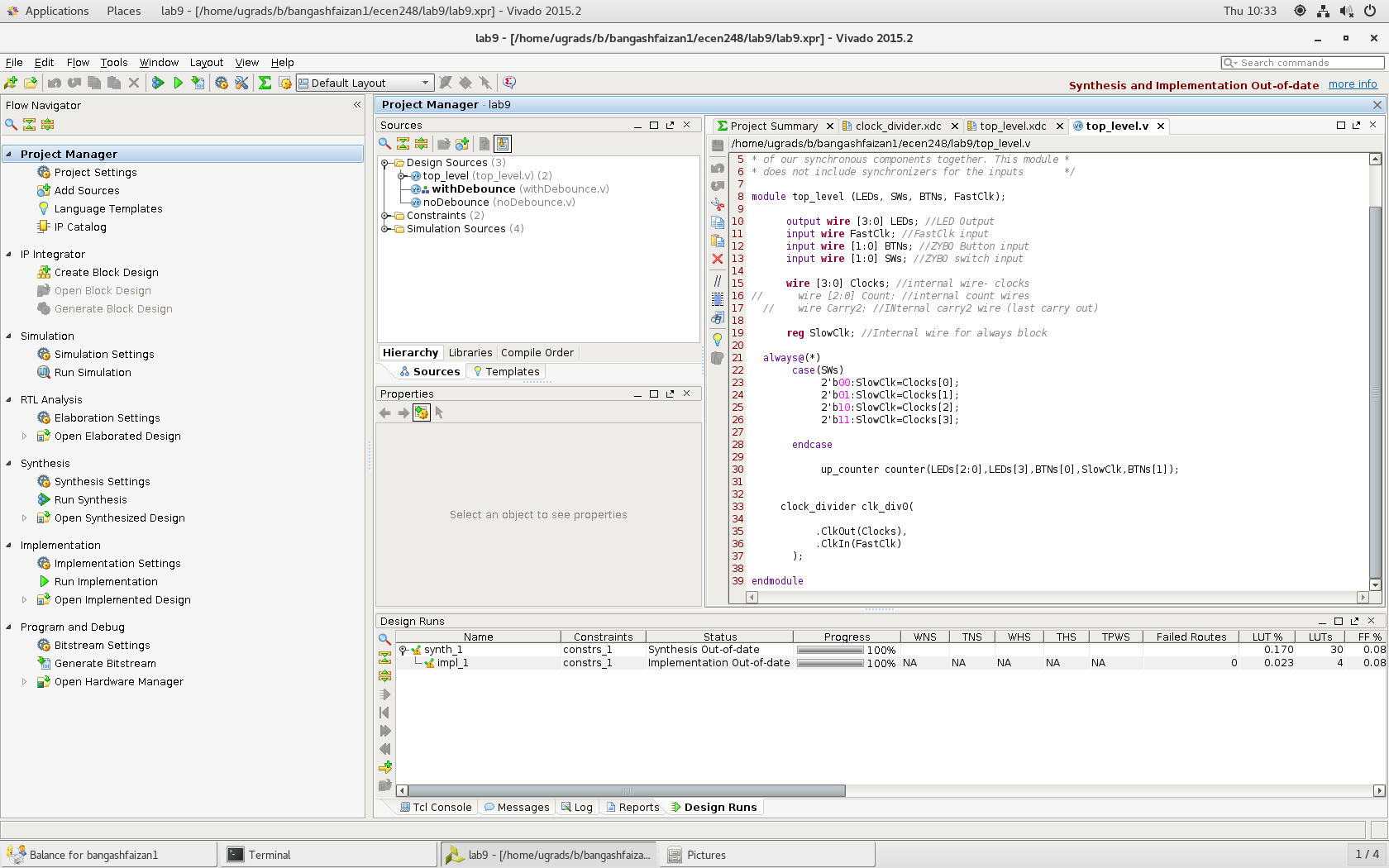
Part 2: For part 2, we initially created a source file half\_adder.v which can be seen .After this, we created up\_counter.v which can be seen. After this, we tested the up counter and it’s results can be seen below. After this, we then edited the clock divider program and created a top\_level.v program which can be seen below. After this, we made top\_level.xdc to implement the program on the Zybo board which can be seen below. From here we implemented the design onto the Zybo board which can be seen.



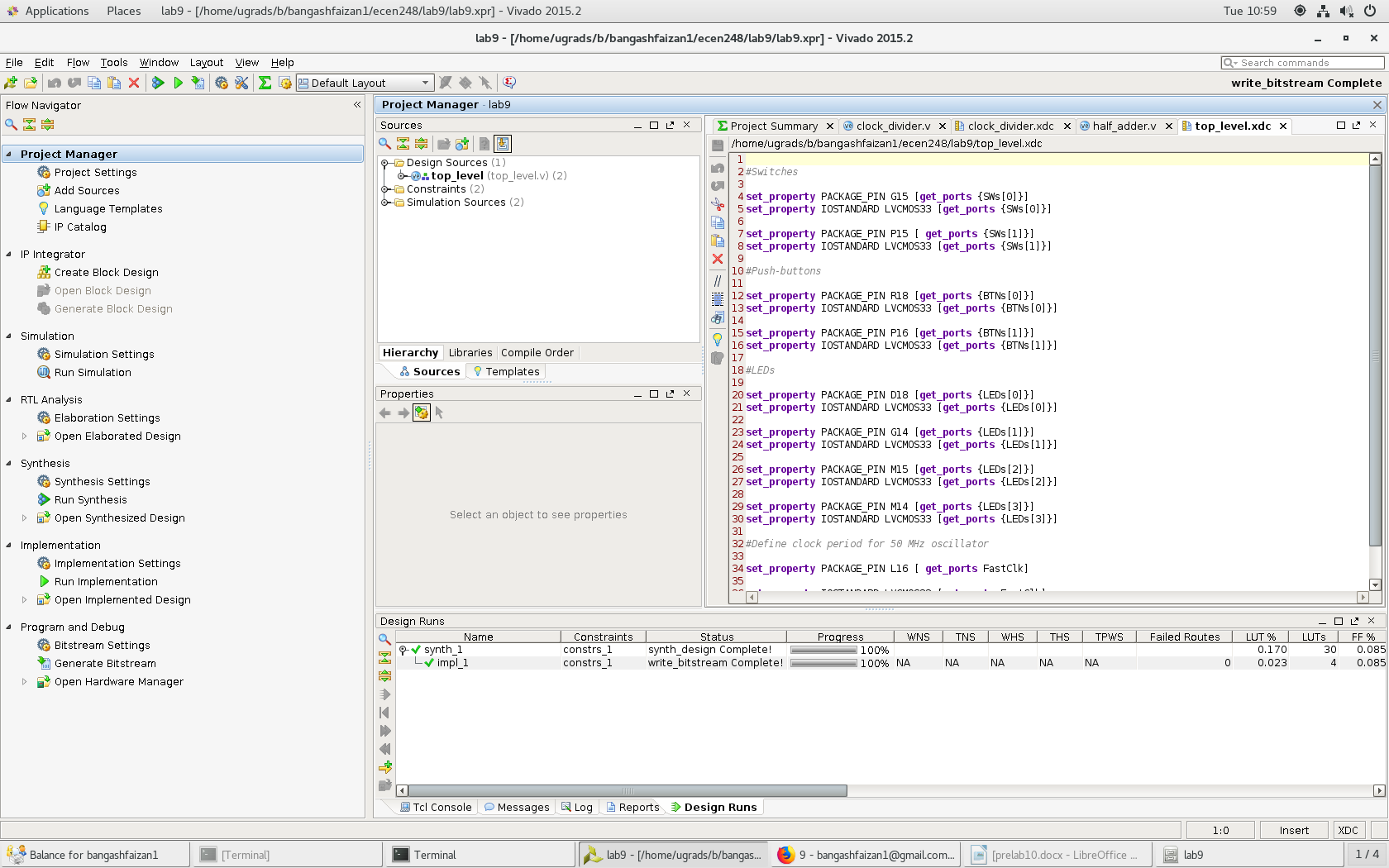
**The code for half\_adder.v**



**The code for up\_counter.v**

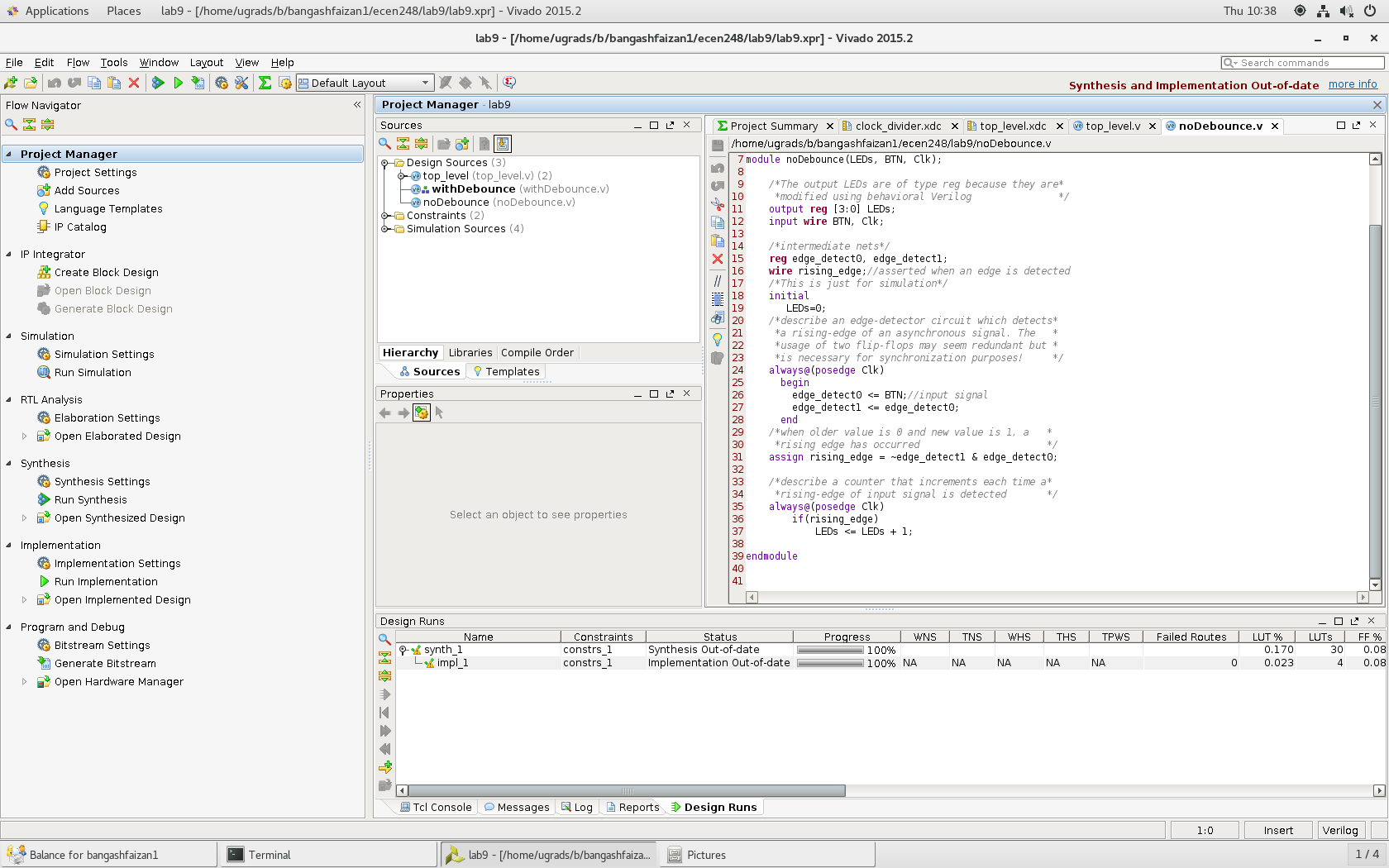
****

**code for top\_level.v**

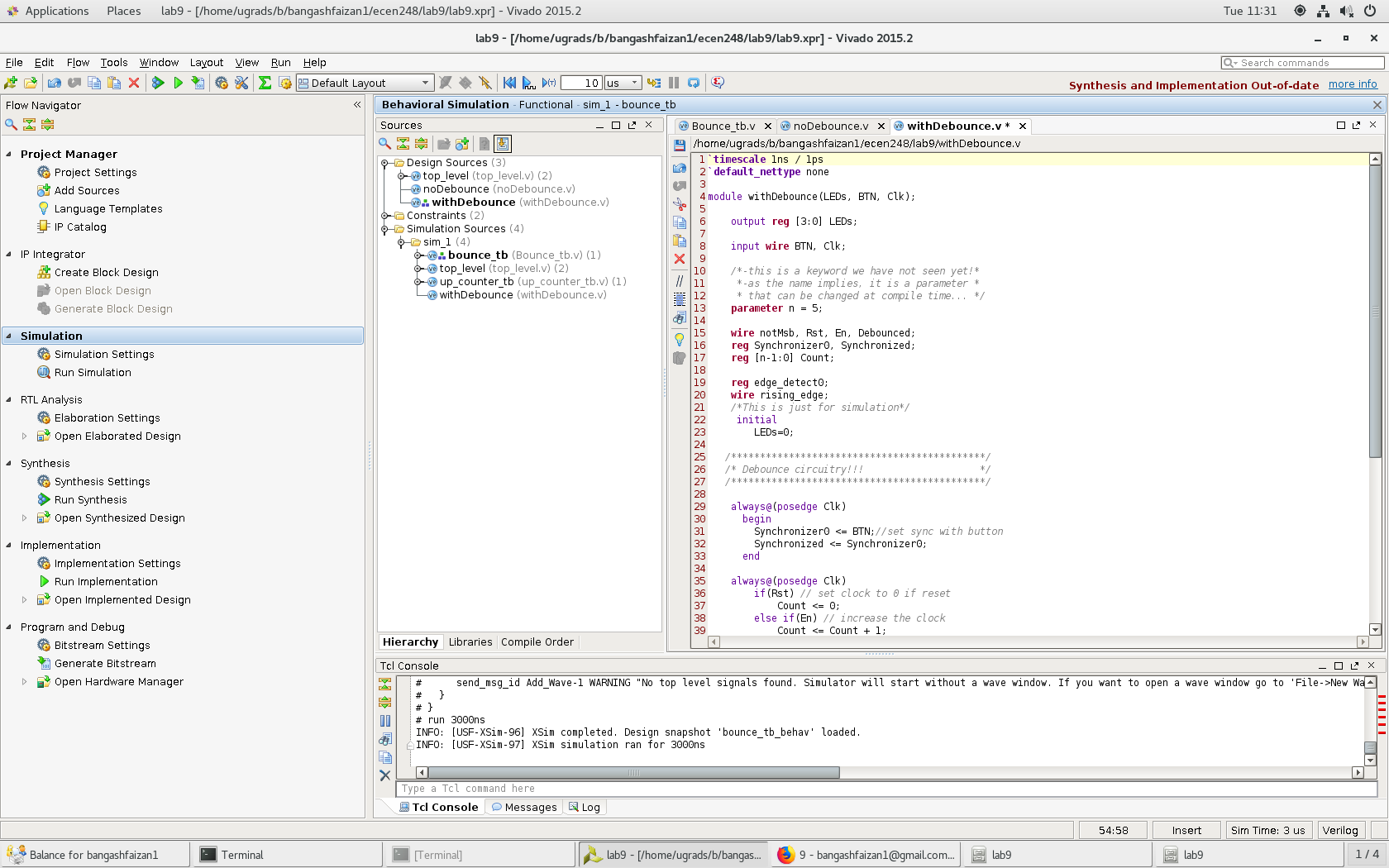
****

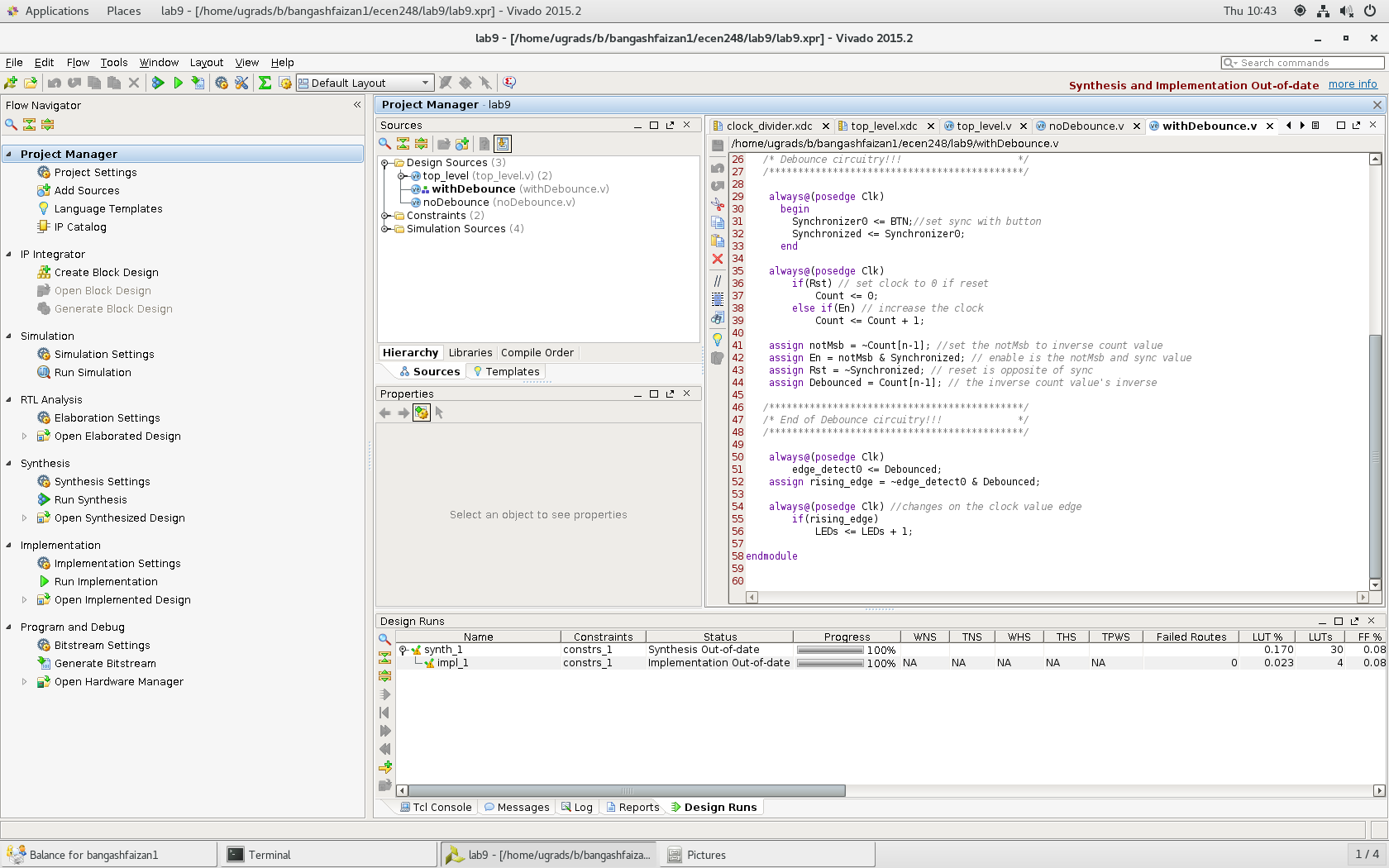
**code for top\_level.xdc**

Part 3: For part 3, we add noDebounce.v from the course directory to the project .We then test it with Bounce\_tb.v. After this, we add withDebounce.v which is seen test it again with Bounce\_tb.v.



**The code for noDebounce.v**

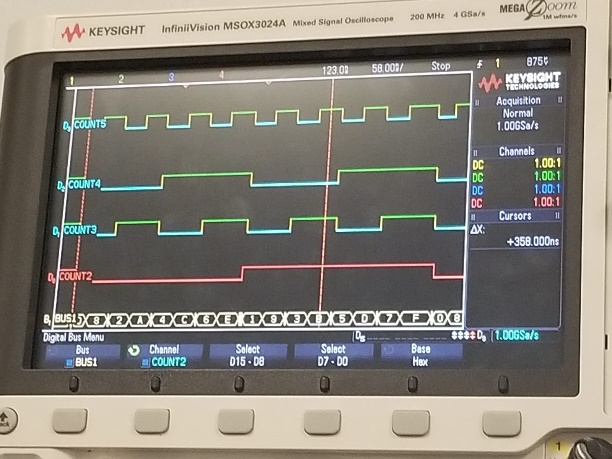
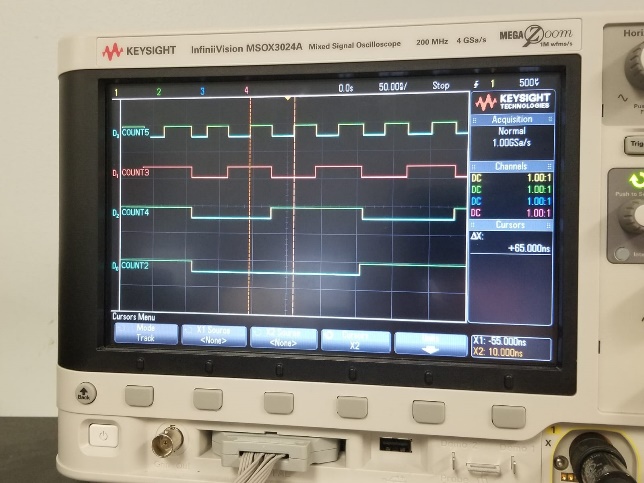
****

****

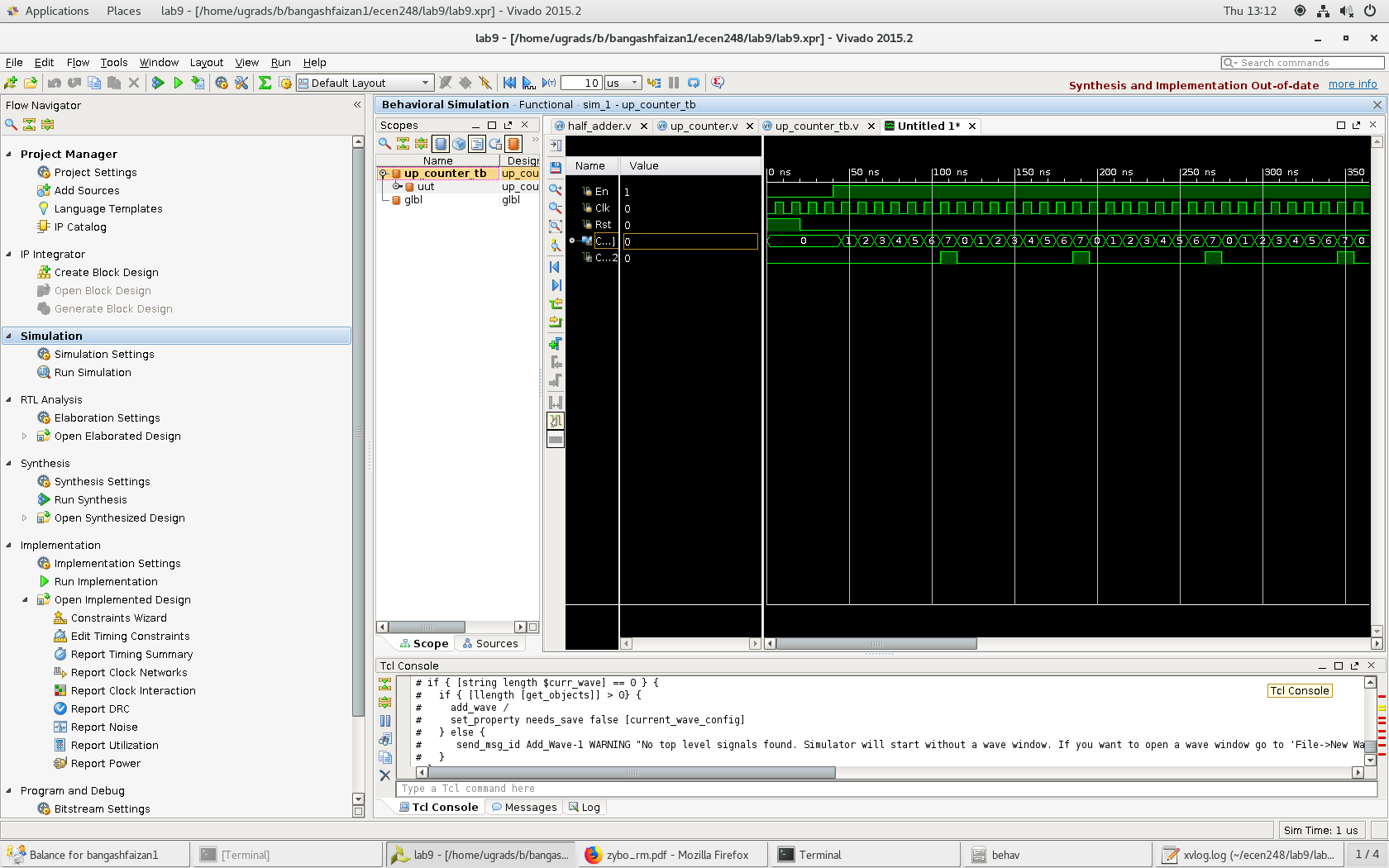
**The code for withDebounce**

**Results –**

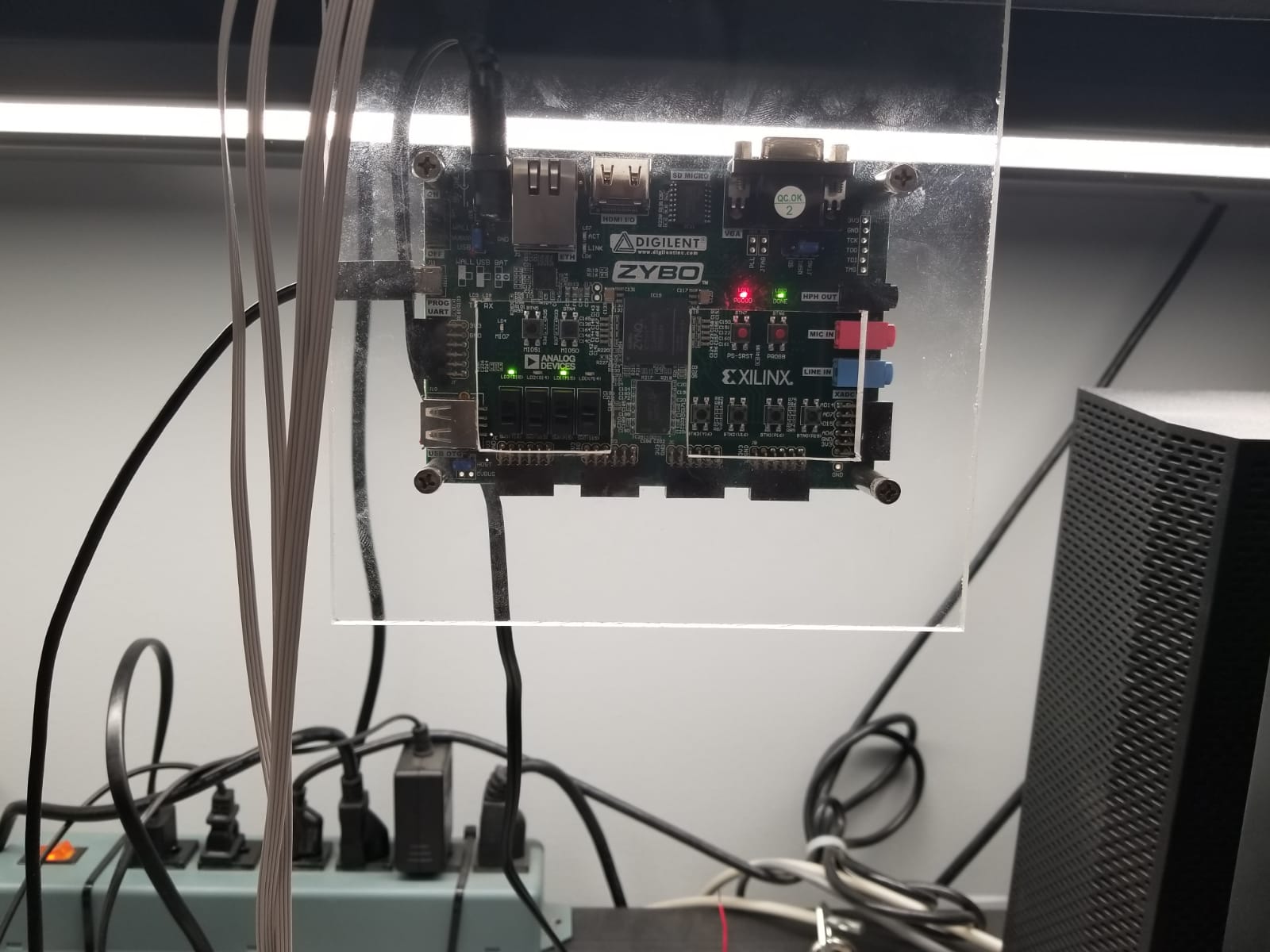
Overall, the lab was successful and went as expected. I did not run into any issues and got the results I was expecting. I was able to see how bouncing works and how to implement counters. Additionally, I was able to witness how clock dividers work which will be beneficial for future labs.



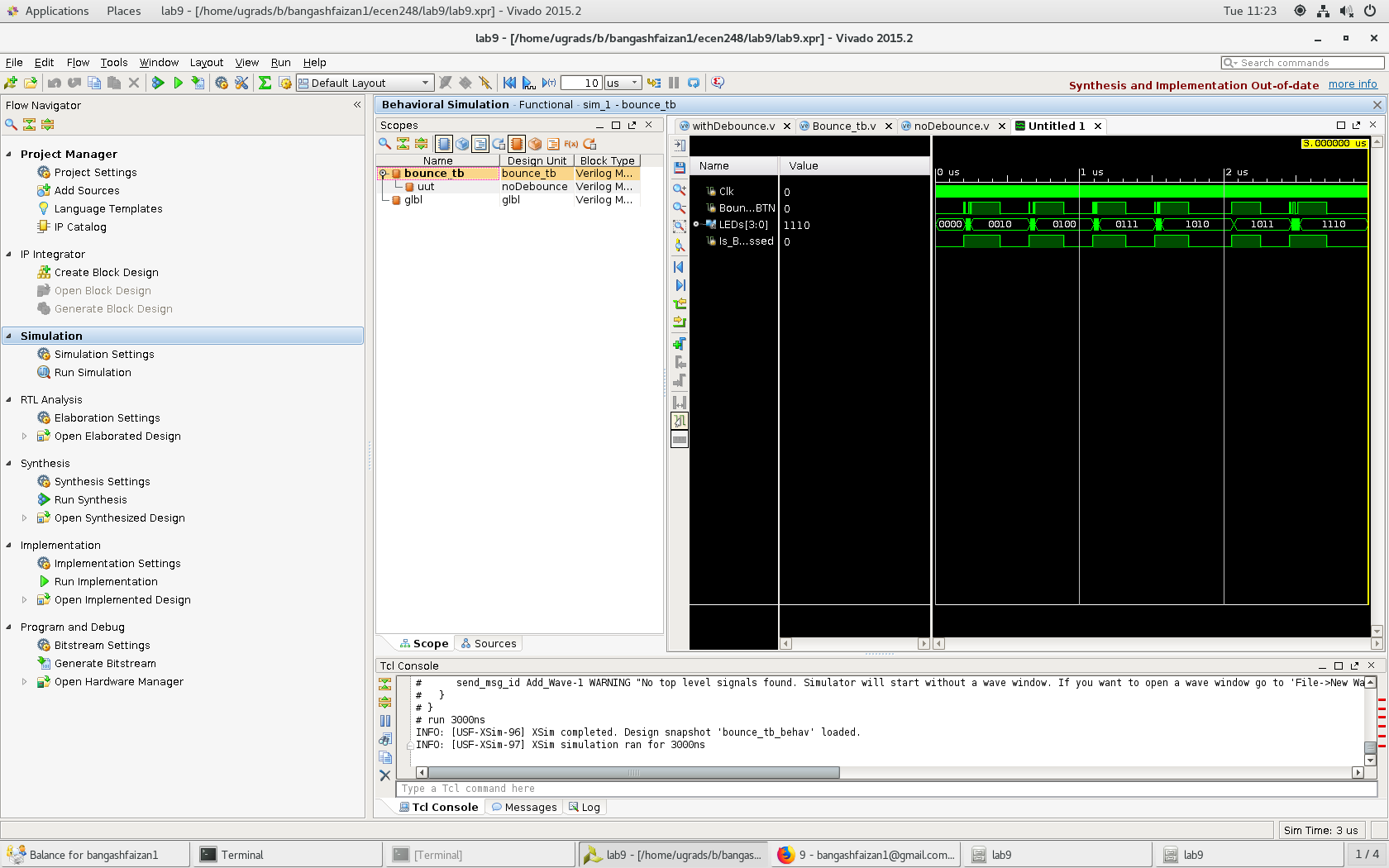
Waveform results from clock divider



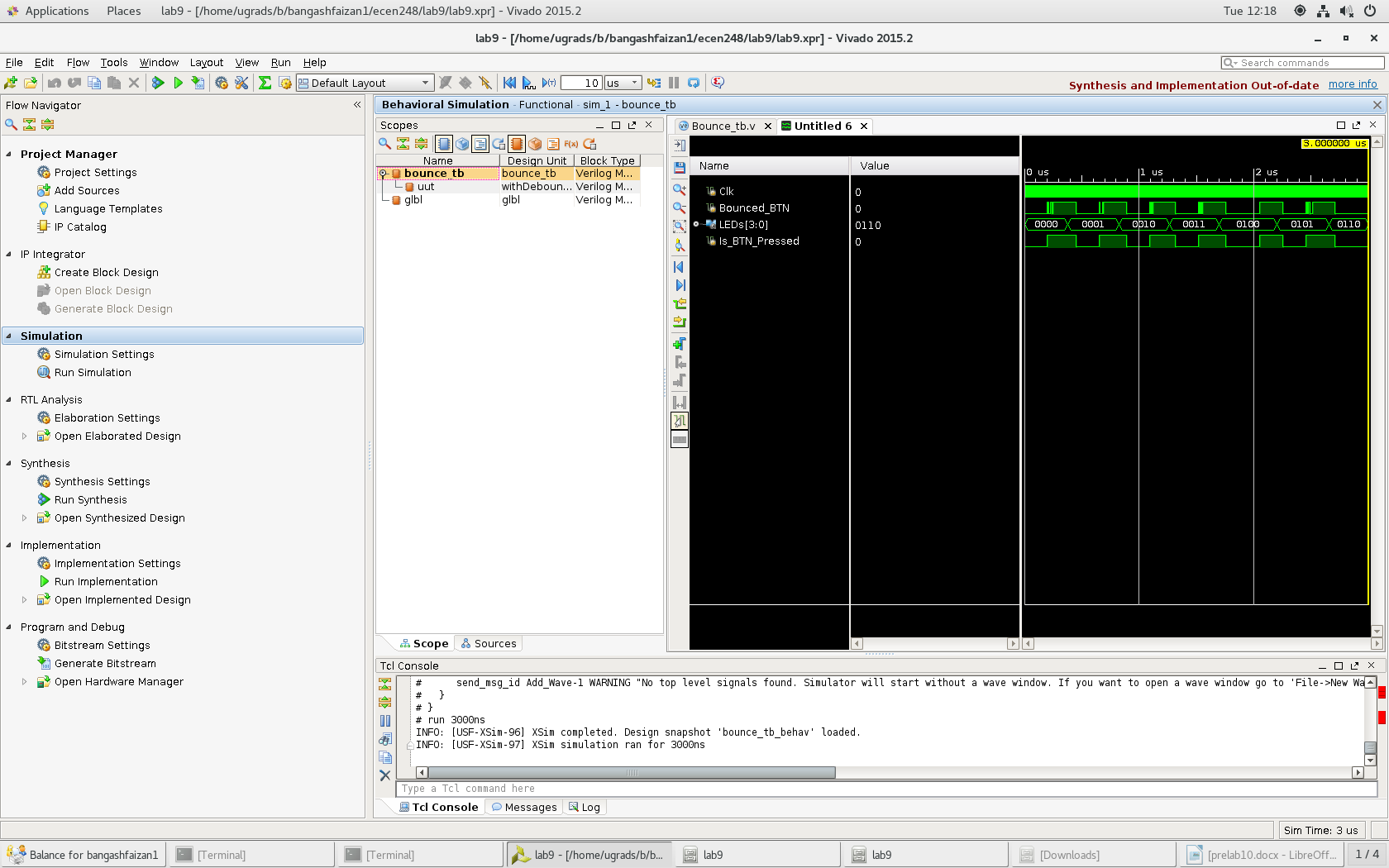
Results for upcounter



Implemented zybo for toplevel



Nodebounce result



With Debounce

**Conclusion –**

This lab was reinforced my knowledge of Verilog and introduced multiple new concepts. We have used sequential circuits a lot in the lab which demonstrates their importance. In this lab, we hit hard on circuit timing and how to manage this. We learned that circuits do not act ideally and we have to handle this.

**Questions:**

In the post lab deliverables

1. Source code is included in the design section
2. XDC are included in the design section
3. Waveforms and outputs are in the design section
4. Questions throughout the lab manual
   1. Experiment 1- No additional questions
   2. Experiment 2-
      1. Part 2b- What is the frequency of the signal?
         1. 1/10 ns= 100 MHz
      2. Part 2c- How long is the interval?
         1. 20 ns
      3. Part 2d- How long does it hold the enable LOW?
         1. 20 ns
      4. Part 2f- What is the maximum count value and what signal in the waveform could we use to know exactly when the counter is going to roll over?
         1. The max count value is 1111 and it resets when the bottom signal goes from 1 to 0
      5. Part 3a- What rate will the most significant bit of the divider oscillate at if we use a 125 MHz clock?
         1. 0.745 Hz
      6. Part 3f- Note what happens with various combinations of SW0 and SW1
         1. SW0 and SW1 set the frequency of the counter. 0 is the highest frequency and 3 is the lowest frequency.
      7. Part 3g- What does BTN1 do?
         1. BTN1 acts as a reset for the counter
   3. Experiment 3-
      1. Part 1b- Does the circuit in noDebounce.v work as expected? Why or why not?
         1. Yes without a de-bounce the circuit will oscillate rapidly once the button is pressed
      2. Part 2a- Does the circuit in withDebounce.v work as expected? Why or why not?
         1. It removes the random oscillation however it breaks the buttons so overall it does not work as expected.
      3. Part 2c- Explain withDebounce.v
         1. withDebounce has several catchers that prevent the button from oscillating rapidly. It works by adjusting for errors and outputting what results should be or what they should be close to.