Name: Ningyuan Zhang

Course: Cpr E 381

Section: K

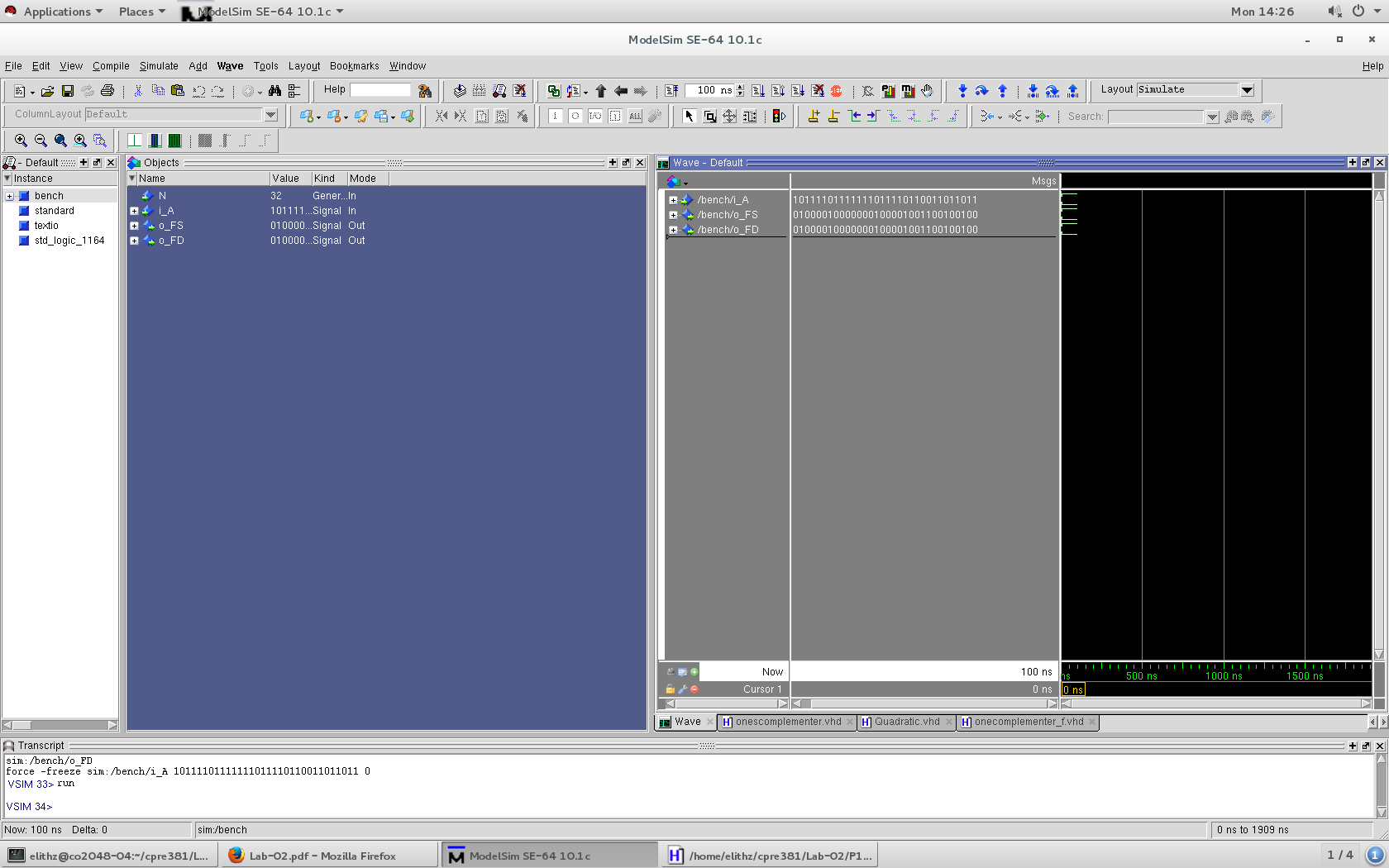
Date: 9/8/2016

TA: Taewoon Kim

Lab 2 Report

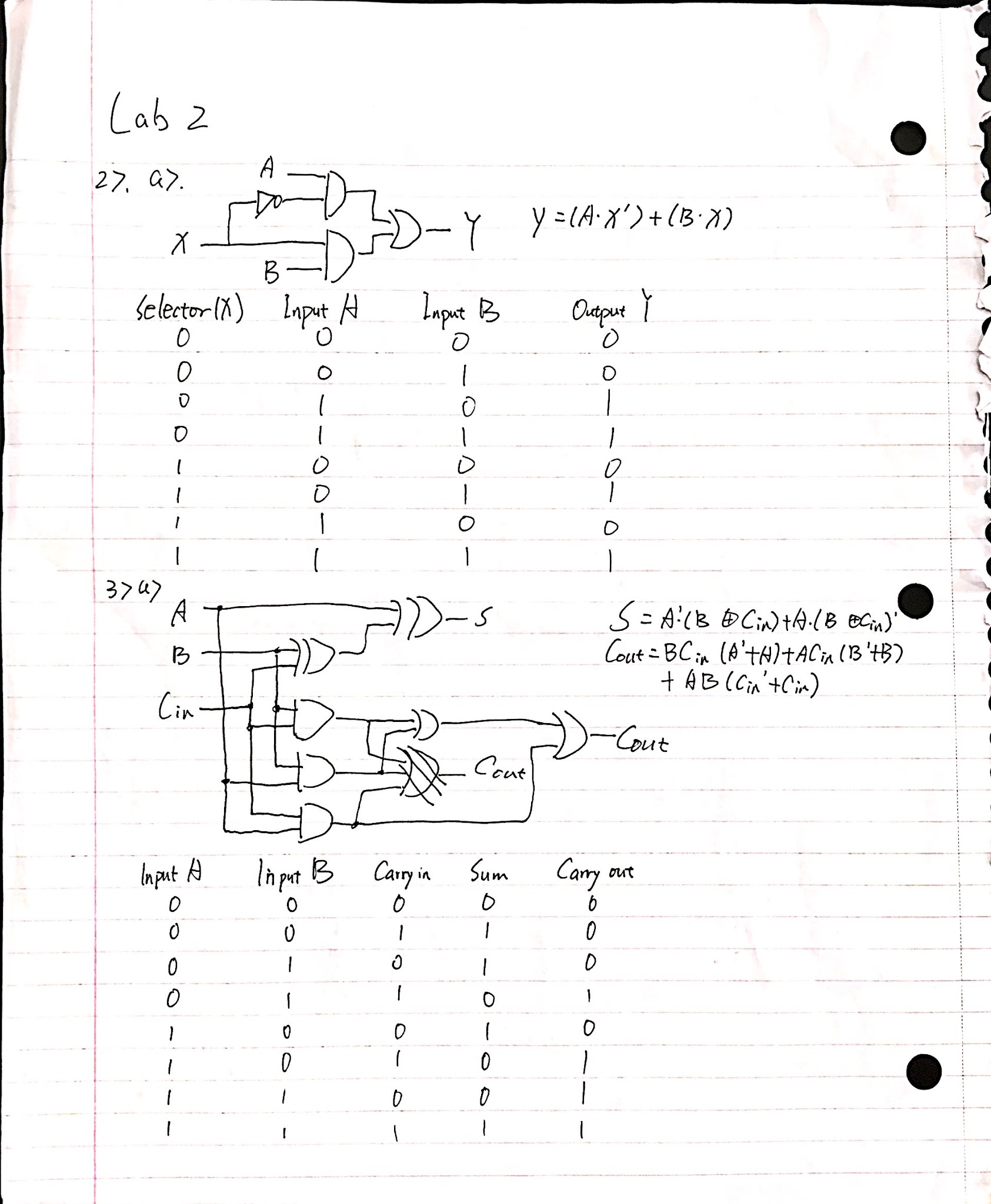
Part 1.

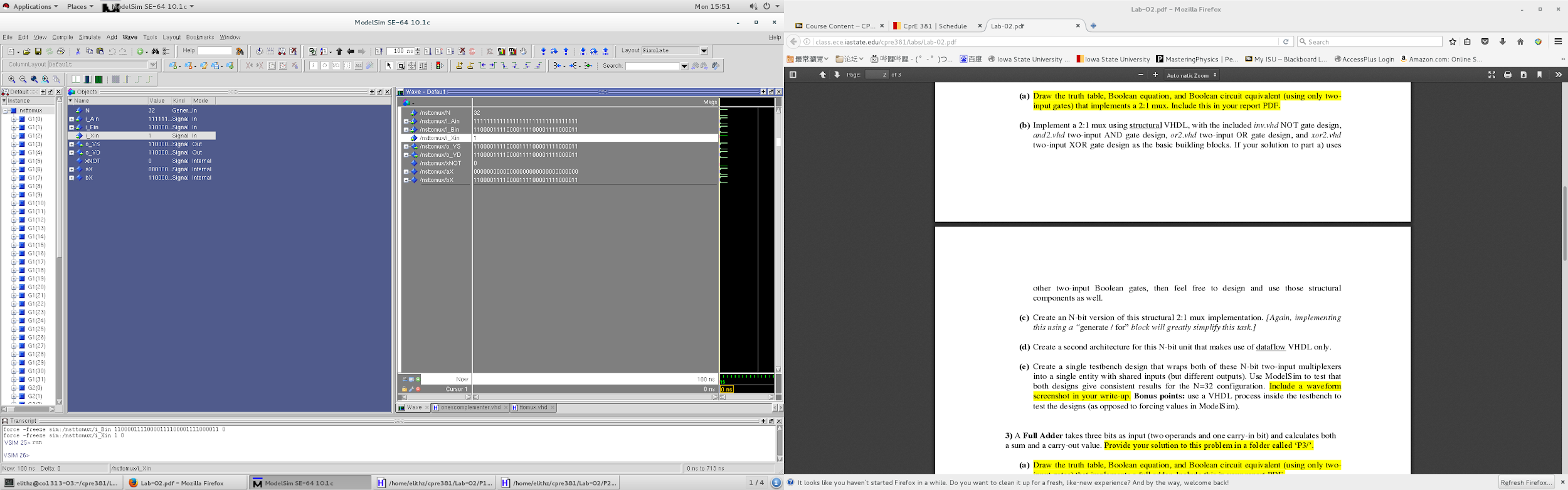
This part required us to implement an one’s complementer using VHDL. The code block that we need to use is a not gate which was already provided for us. This part also needed us to write the code in both structural and dataflow style. After finished the single bit one’s complementer, we also need to convert it to a n-bit one’s comlementer by using generic nember and test all two style code in testbench.



Part 2.

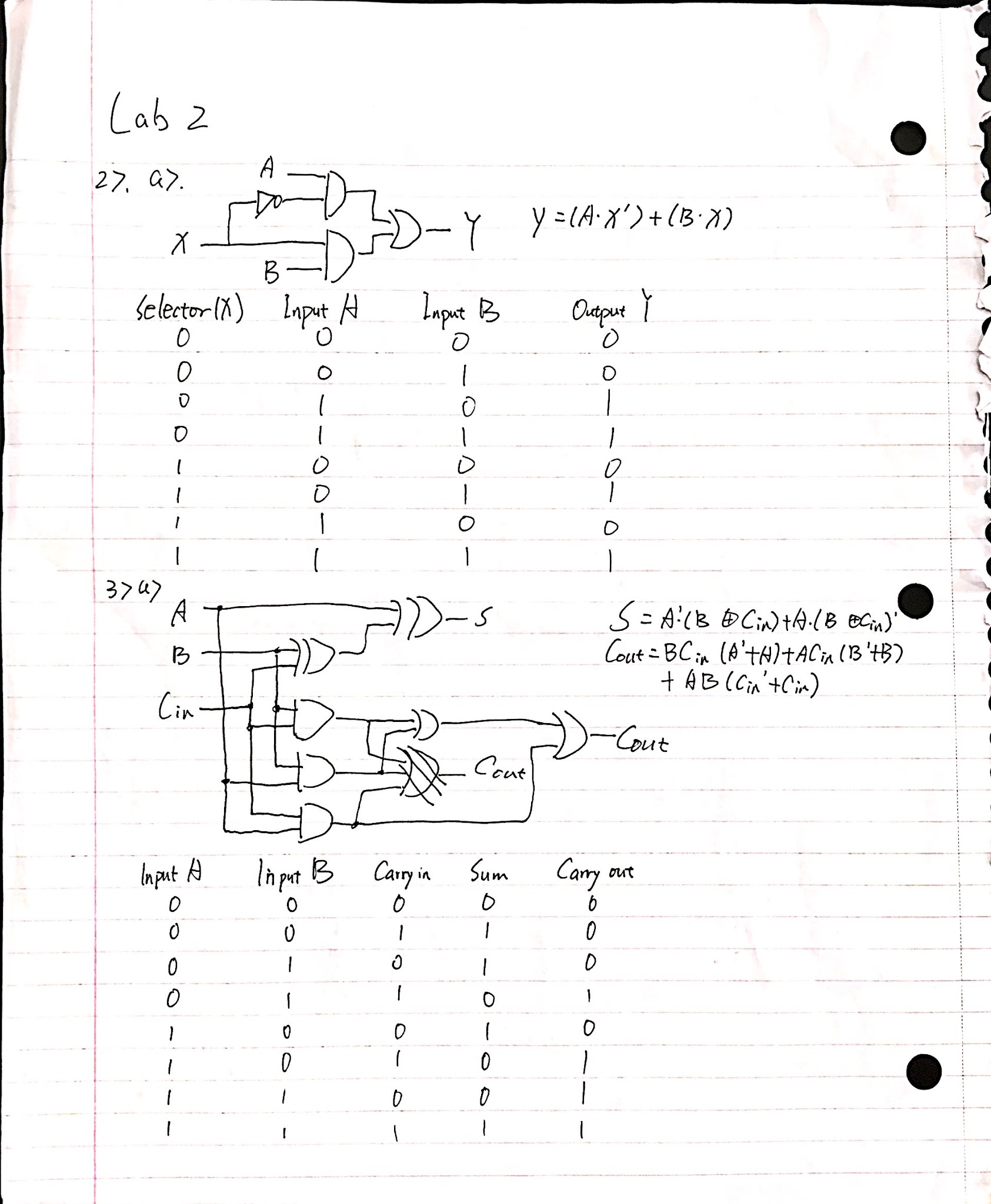
This part was requiring us to finish a n-bit 2 to 1 muxer in both structrural and dataflow style. I started from doing 1-bit version structural VHDL 2:1 mux. I drew the circuit and truth table, and also wrote the expression. Then I used the same way to inplement the n-bit 2:1 mux by using generic nember and for loop and also implemented the dataflow version of the code and let it run on testbench.

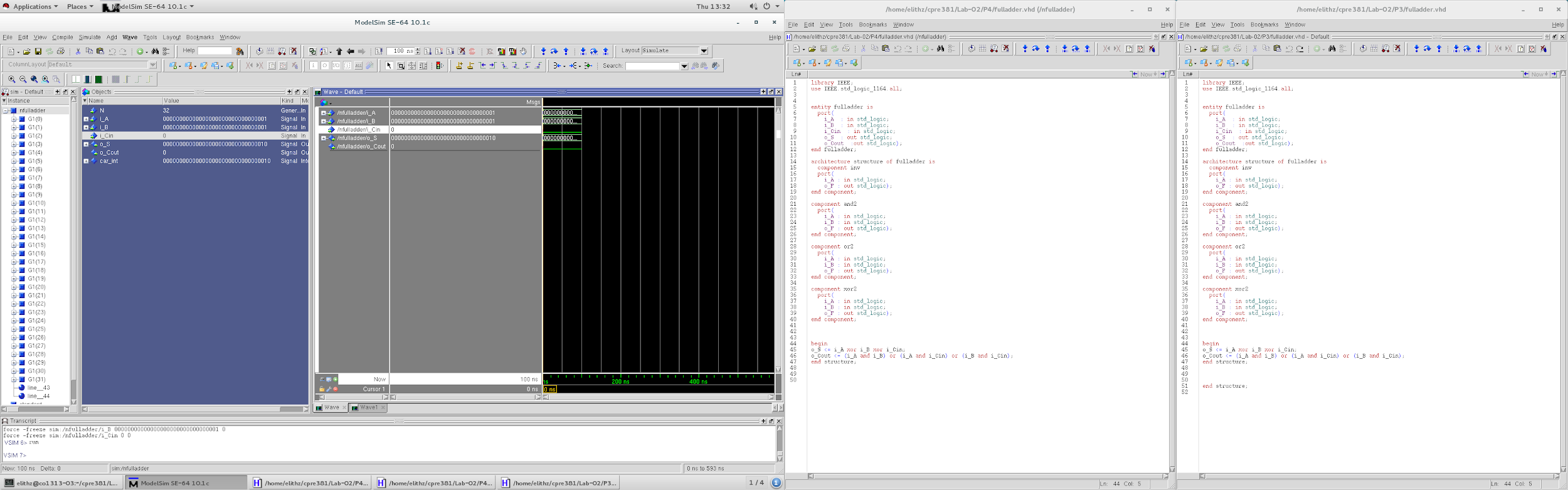


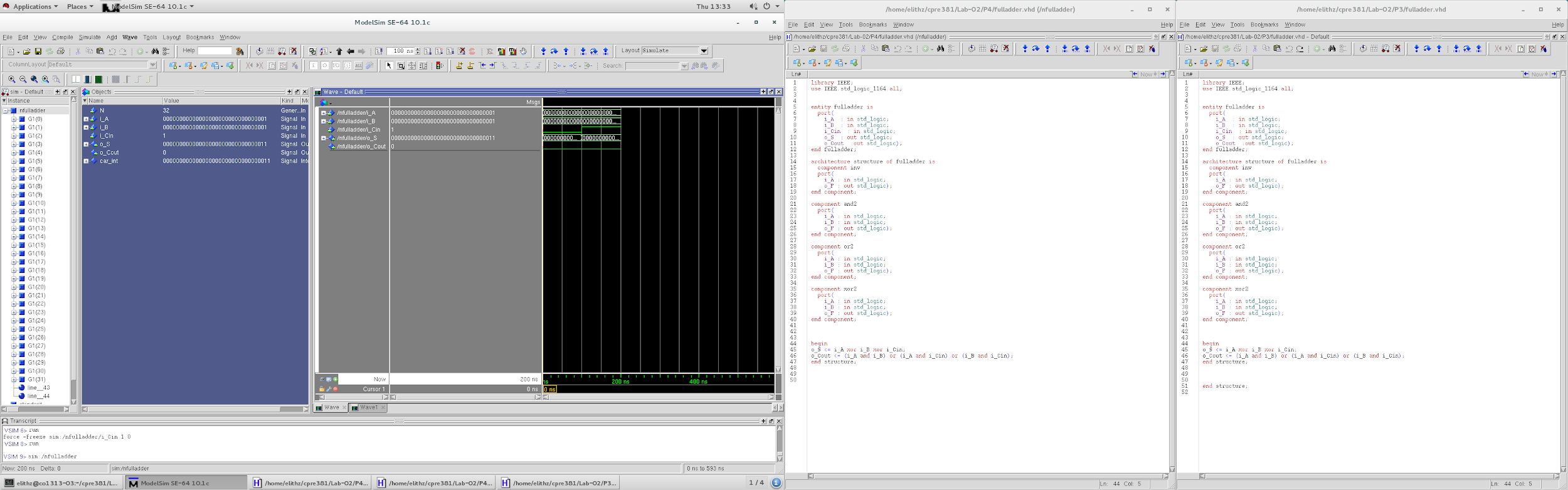


Part 3.

This part is the hardest part of this lab. Because of the n-bit full adder’s complicated structure, it is kind of hard to convert it into VHDL. As usual, I first finished the circuit, truth table and expression, and then I implemented the 1-bit full adder regarding to the expression. Then I found that I could create a new file to use the 1-bit version as a component so I could solve the problem of carry-in and carry-out in n-bit version. But because of the specific feature of the n-bit version, I did not provide the dataflow version for n-bit one.







Part 4.

In this part, I needed to implement a n-bit adder-subtractor using the VHDL entities that I finished in previous 3 parts as components. I drew the circuit of it and then implemented it in structural style (it did not require me to implement the dataflow version). This adder-substractor is controled by an switch which is also the carry-in for the full adder and the selector of the 2:1 mux. When switch is 1, it will perform a substract behavior and the result will be A - B, and if it is 0 it would simply add those two numbers. B will pass through a one’s complementer to enter the mux with its self and will be selected by the switch, the result of the mux will go into the full adder with A. the result of the full adder is the sum, and the carry-out of the full adder is the carry-out of the whole adder-substractor.

