Khoa Tran

07/03/2020

EE 271

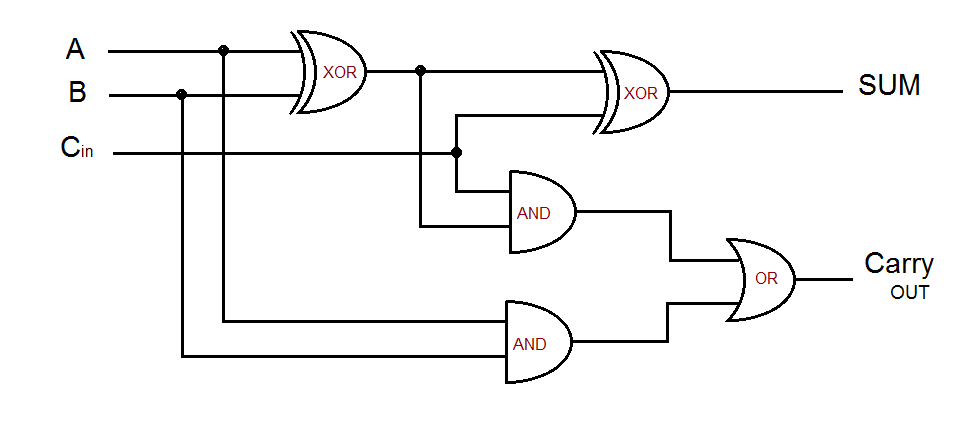
Lab 1 Report

**Procedure**

Task 1: Creating schematic diagram and simulating in Quartus

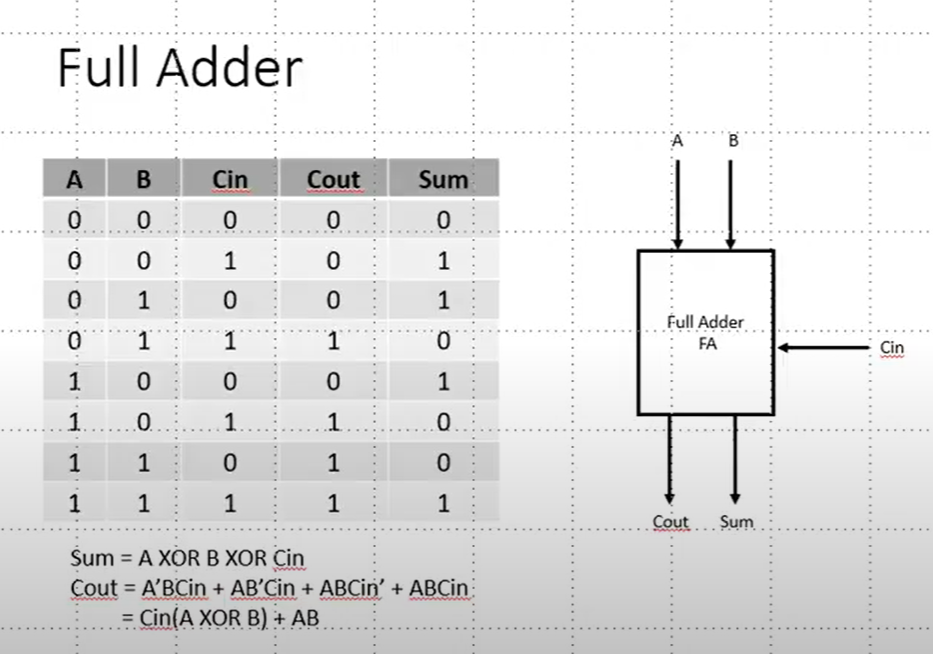
Approaching this problem, I first drew up the schematic diagram for a fullAdder and followed the tutorial for the setup and compilation of the diagram. Below is an image of the schematic diagram that I used to base the modeling.

In order to run the waveform simulation, I created a new simulation waveform editor and inserted the input and output. I also changed the interval timings to microseconds.



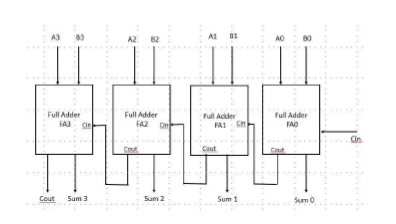
Task 2: Implementing and simulating the full adder with Verilog and ModelSim

I wrote the code based on the instructions with the inputs and outputs but the equations and truth table was derived from the image below. Afterwards, I simulated on ModelSim and tested on the FPGA board.

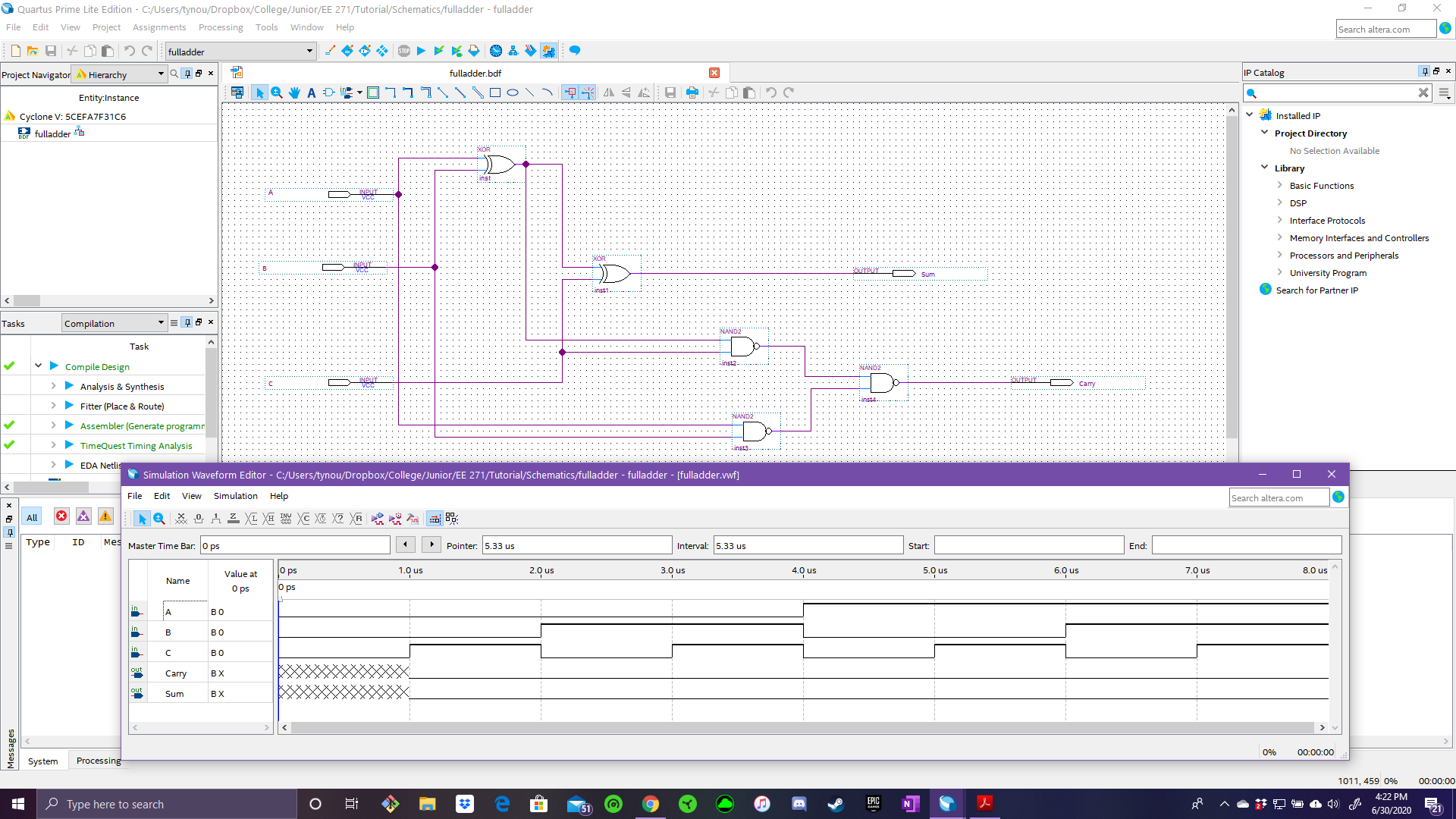
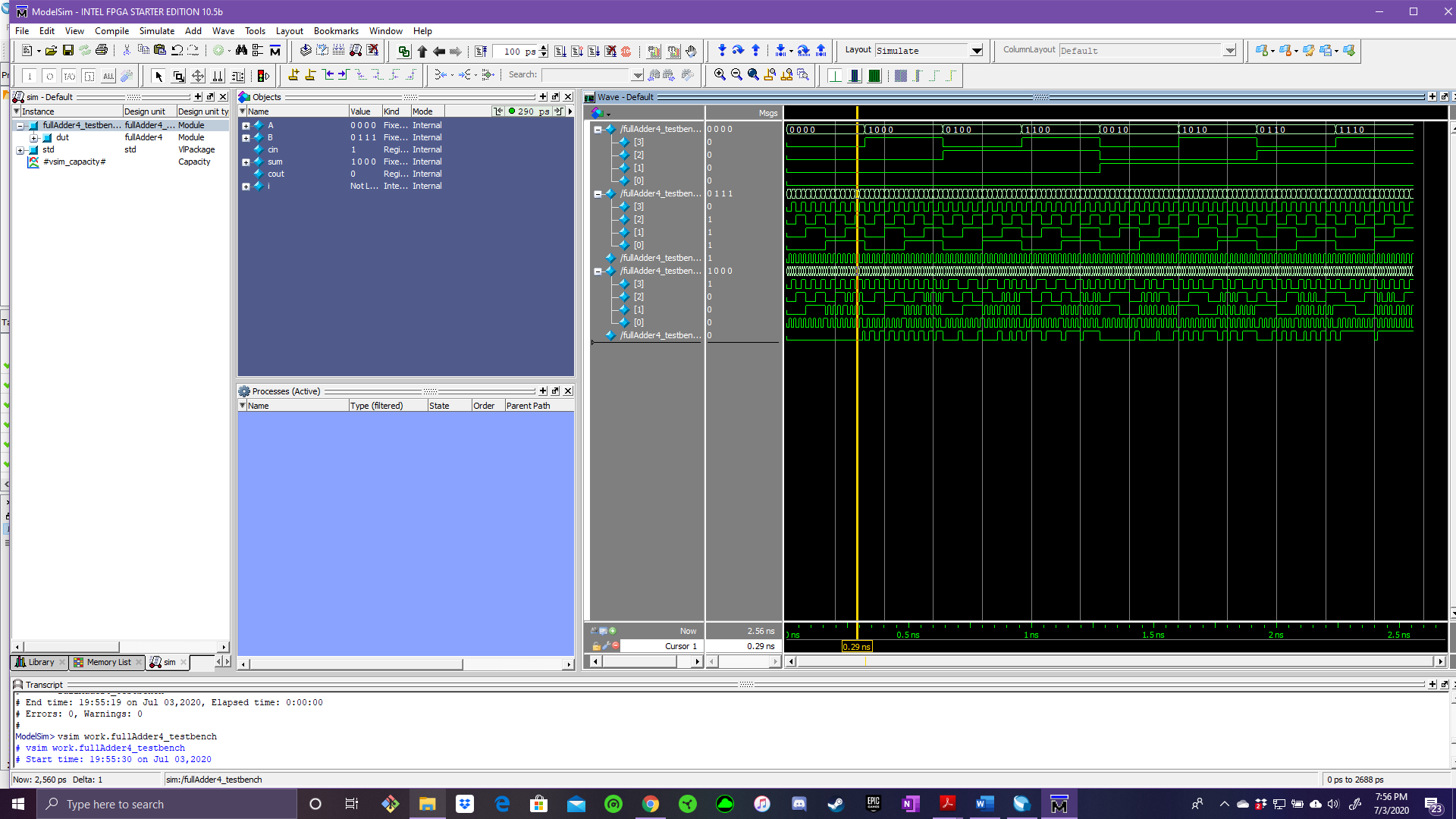


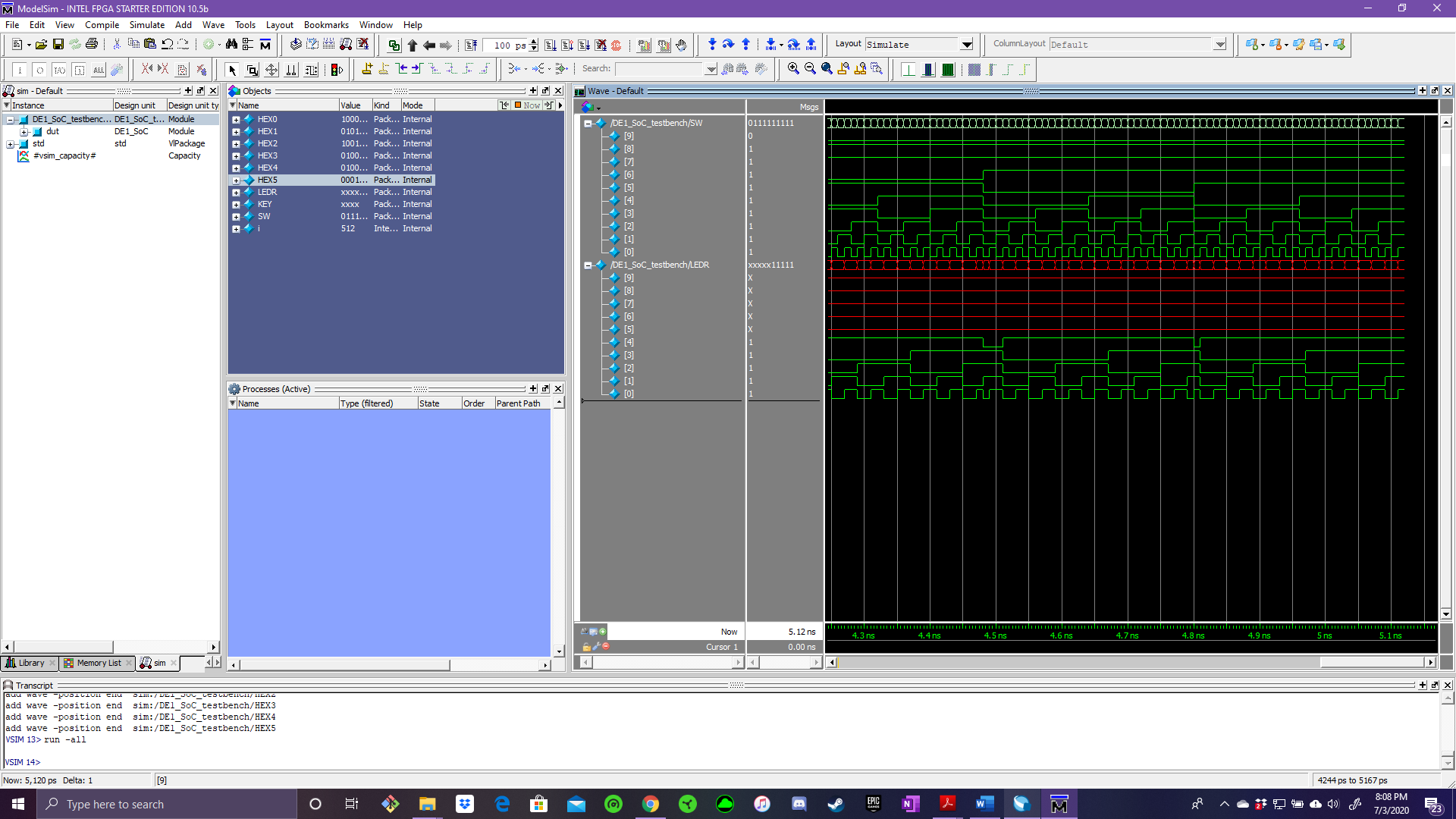
Task 3: Design 4-bit adder

To approach this problem, I first looked at the construction of the fullAdder with 4-bits. From this figure, I was able to determine that the fullAdder with 4-bits implements the module from the original fullAdder with the equations of sum and cout remaining the same. I also used the lecture build of fullAdder with 2-bits as a guild. For the DE1\_SoC board, I assigned the variables with the correct input or output with the switches, LEDs, and HEX board. I found out how to display different characters on the 7 segment HEX board. I also wrote the testbench by looking at the truth table and understanding it as a count binary count of decimal numbers from 0 to 255 for a total of 256 combinations. As a result, I used a for loop to make it easier and simulated on ModelSim.



**Results**

1. Waveform for Task 1
2. Task 3



Description:

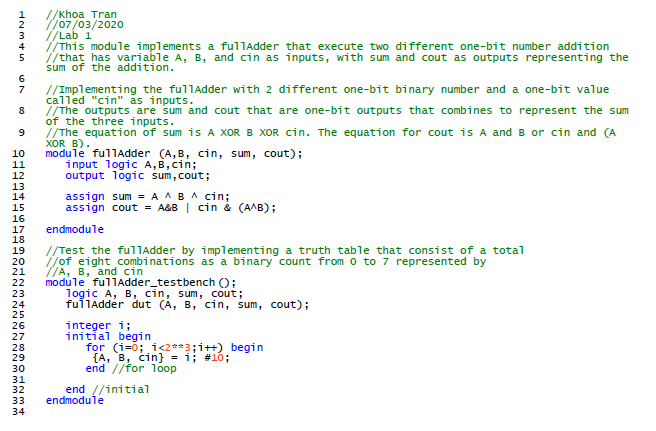
The simulation is a waveform that allows you to compare the different values of the inputs and the results of the outputs. The simulation allows for a variety of input numbers in order to see the outputs with the associated equation. In the second screenshot, there is a yellow line through the bars as it shows the values of the different inputs and outputs at a single point in time. This allows for testing of the truth table and seeing if the outputs are correct according to the row of inputs.

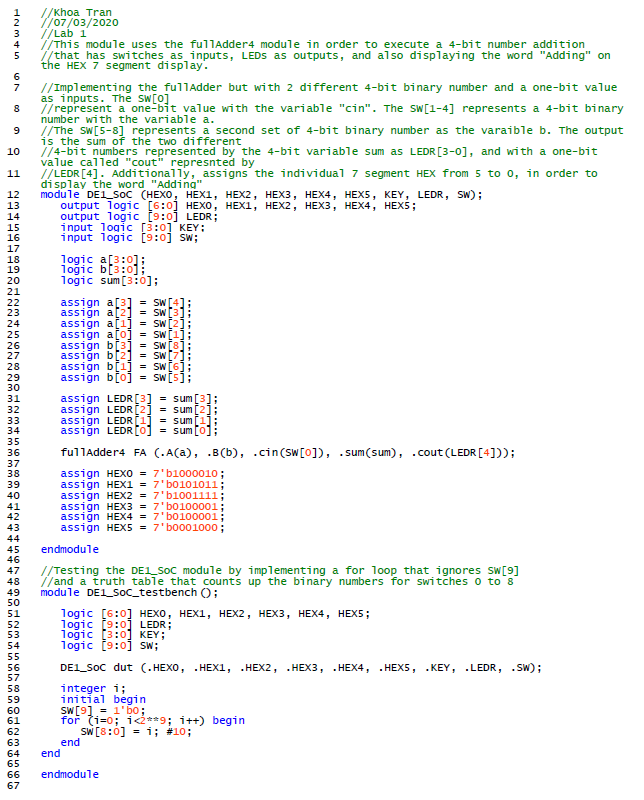
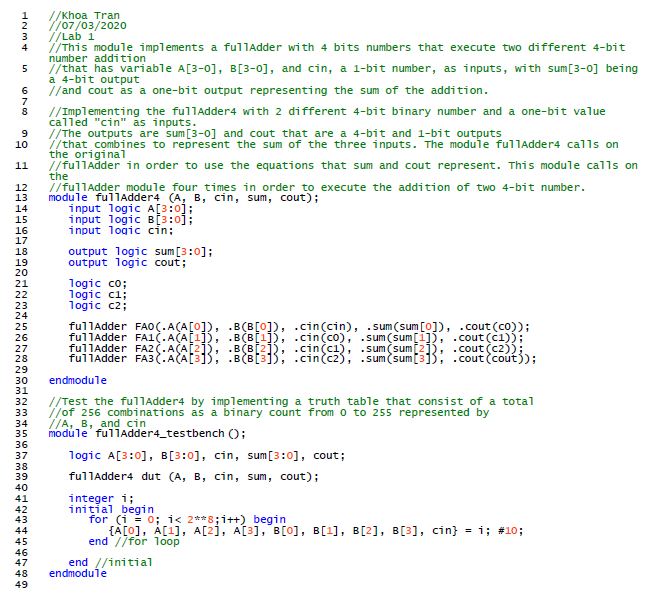
Overview:

Overall, this project was designed to learn how to develop a schematic diagram and writing the Verilog code that represents the fullAdder in order to connect with the FPGA board and use as two different 4-bit binary number addition. Additionally, the project also allows for simulation of the different fullAdder to be able to check if the outputs are correct according to the input before implementing on the FPGA board. In my finished project, I accurately and correctly implemented each step of this project by being able to successful test the fullAdder with the addition of two 4-bit number on the FPGA board. Additionally, I also implemented the 7 segment HEX board by displaying the word “Adding”. In conclusion, the finished product was everything the project asked for and was done with accuracy and precision.

**Appendix**

1. FullAdder module



1. DE1\_SoC module
2. FullAdder4 module