EEC 180 Lab 1

Bradley Manzo
916953788

Objective and Introduction:

The objective of the first part of this lab is to become familiar with the quartus and modelsim environments, while the objective of the second part is to review EEC 018 material and implement a simple verilog description. This is achieved through the use of karnaugh maps and using verilog assign statements. Finally the lab requires the construction of a test bench file in order to verify the behavior of the verilog description with the behavior of the DE10 FPGA board.

Experiment:

Part 1. Implementing Basic Combinational Logic Gates on the DE10-Lite Board

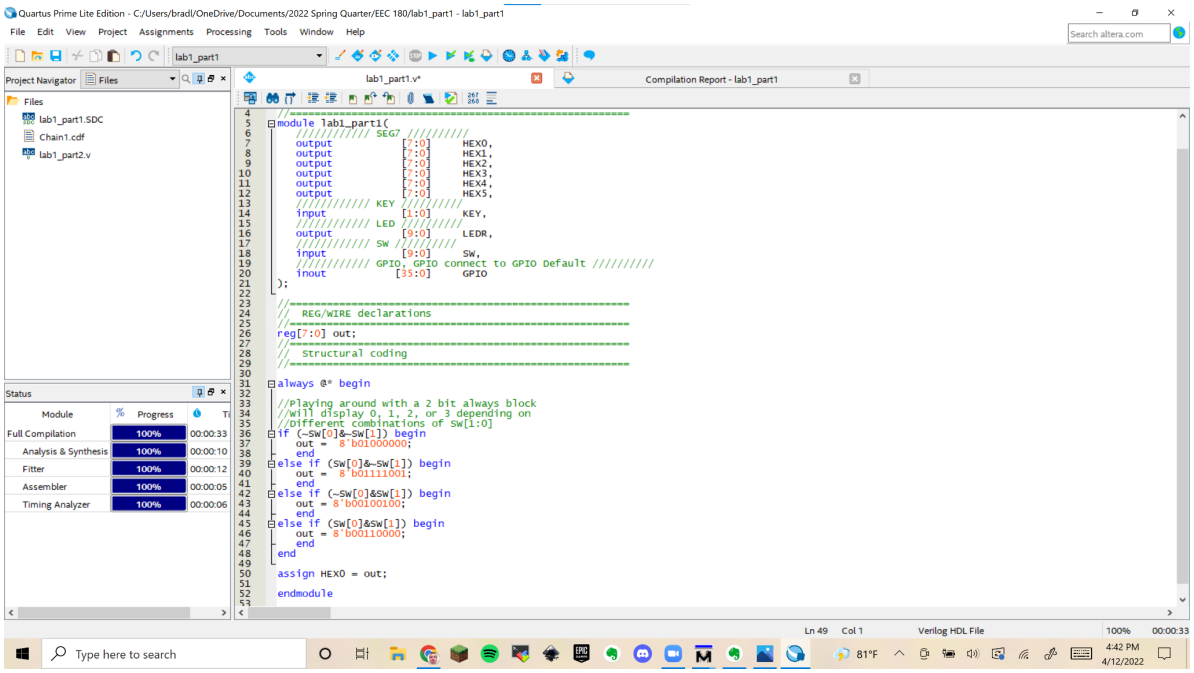


Figure 1: lab1_part1.v

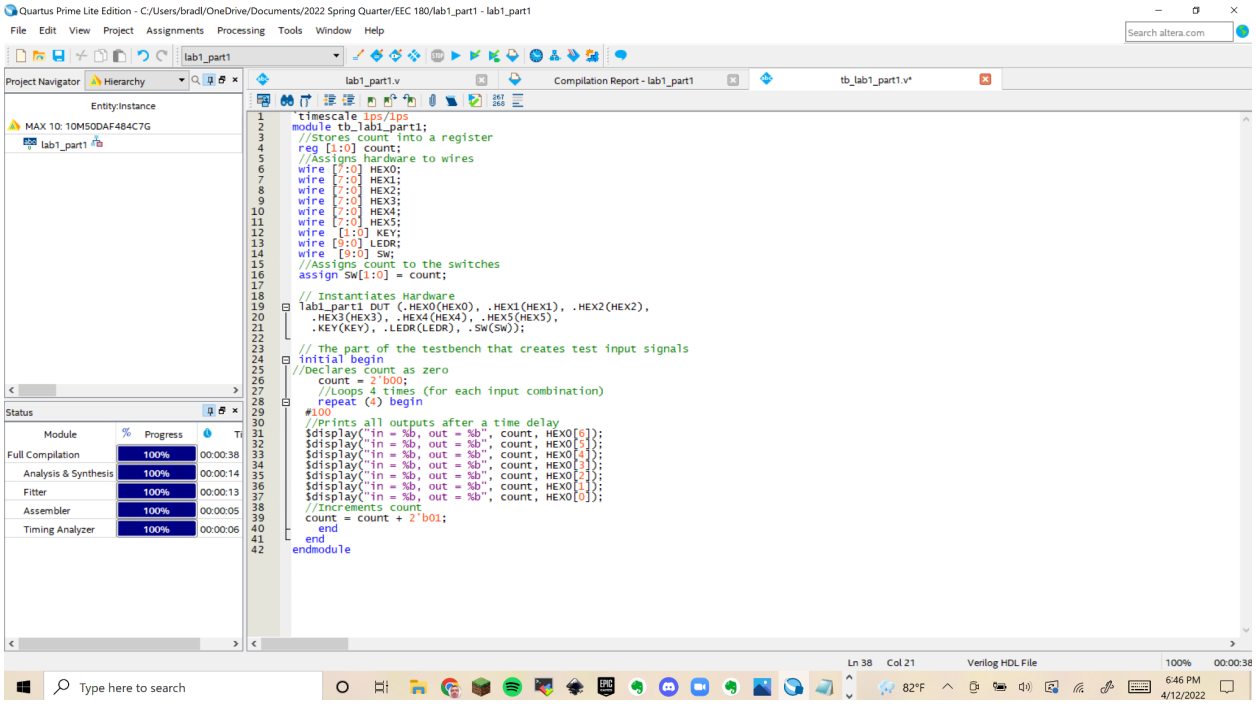


Figure 2: tb_lab1_part1.v

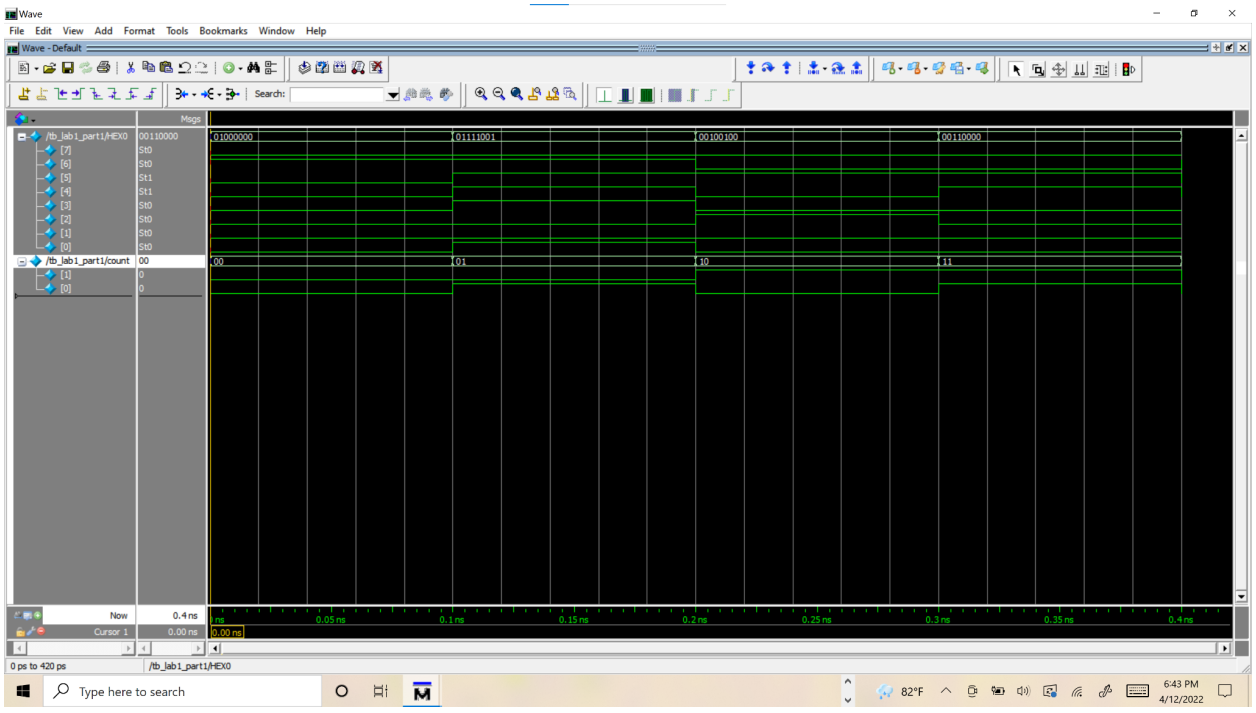
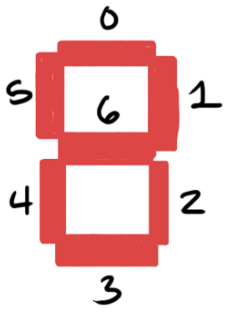


Figure 3: Testbench Simulation of tb_lab1_part1.v

Part 2. Implementing a Combinational Logic Decimal 7-segment Display for 4-bit Switch Inputs



	SW[3-0]				HEX1[6-0]							HEX0[6-0]						
	3	2	1	0	6	5	4	3	2	1	0	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0	0	0	1	1	1	1	0	0	1
2	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
3	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0
4	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1
5	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0
6	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
7	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
10	1	0	1	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0
11	1	0	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1
12	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	1	0	0
13	1	1	0	1	1	1	1	1	0	0	1	0	1	1	0	0	0	0
14	1	1	1	0	1	1	1	1	0	0	1	0	0	1	1	0	0	1
15	1	1	1	1	1	1	1	1	0	0	1	0	0	1	0	0	1	0
					1	–	–	–	0	0	–	–	–	–	–	–	–	–

Figure 4: Truth Table for Seven Segment Display

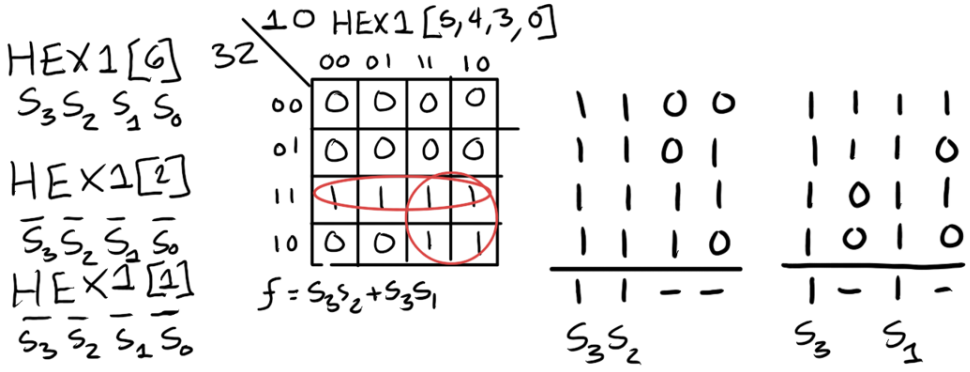


Figure 5: HEX1[7:0] K-Map and Equations

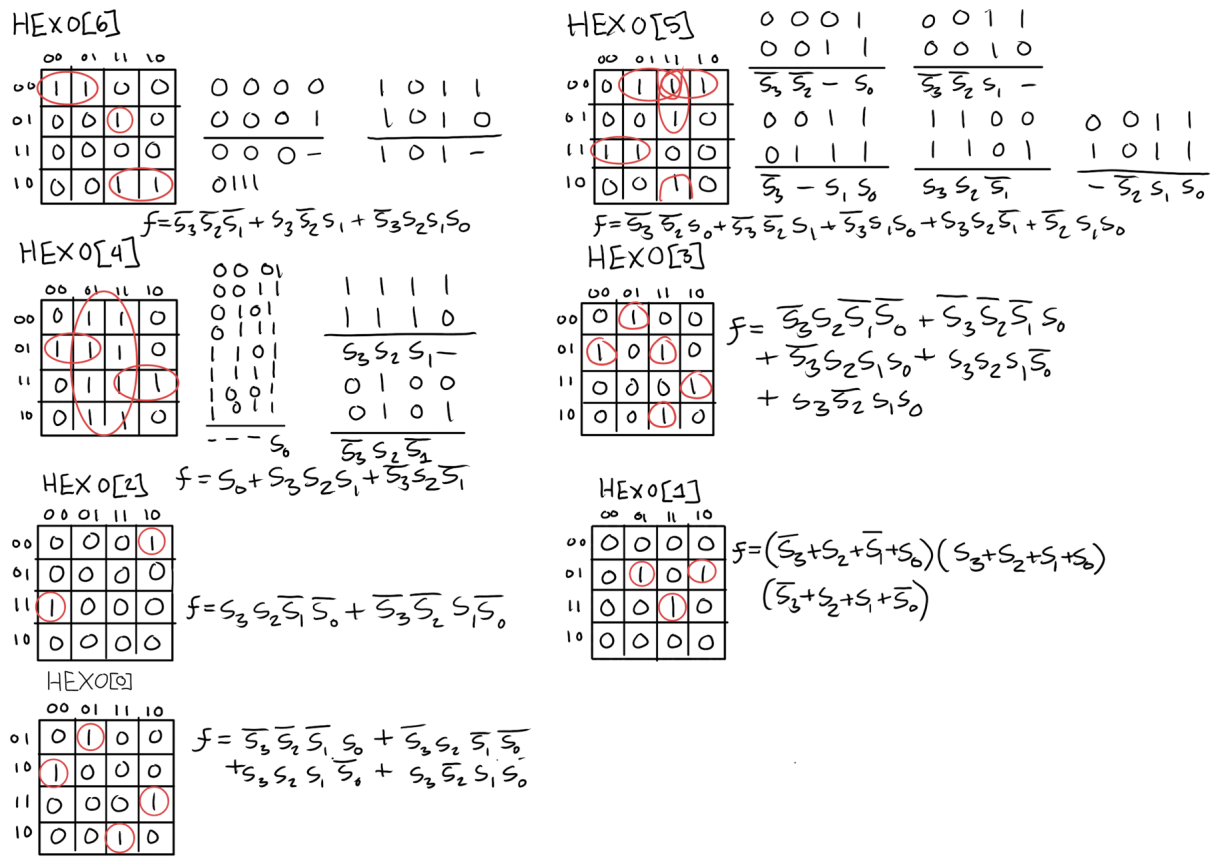


Figure 6: HEX0[7:0] K-Maps and Equations

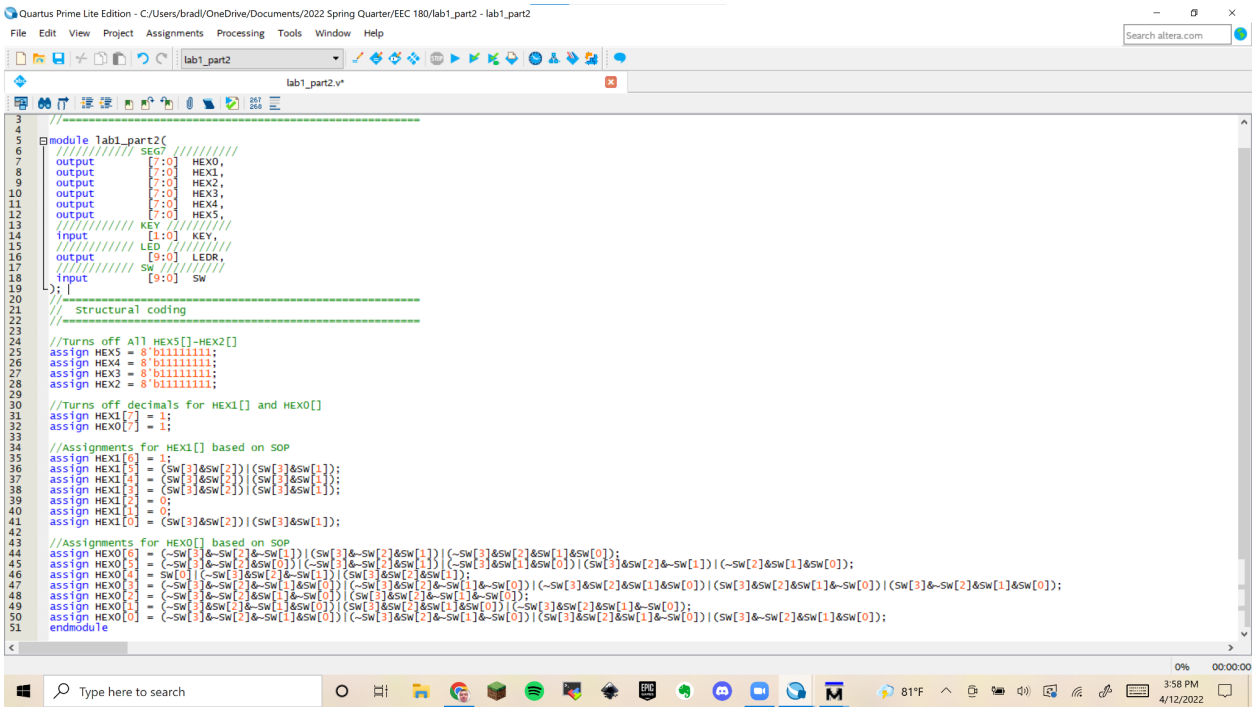
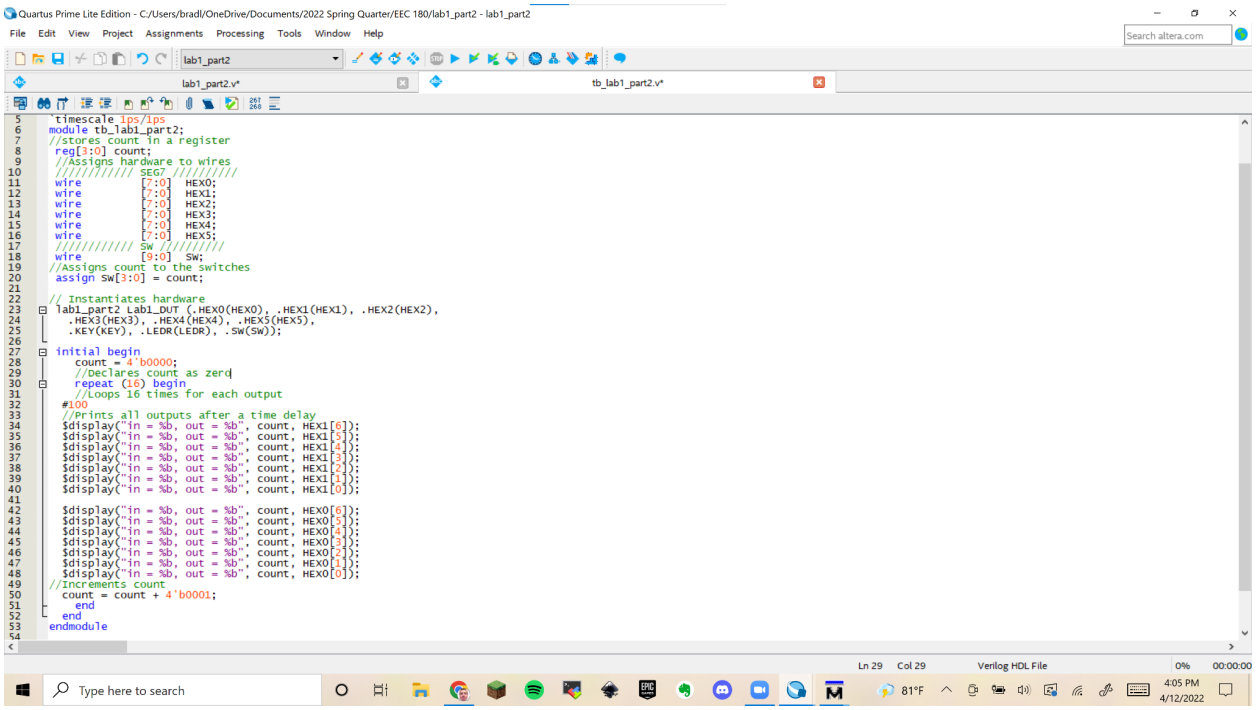


Figure 7: lab1_part2.v



```
`timescale 1ps/1ps
module tb_lab1_part2;
//stores count in a register
 reg[3:0] count;
 //Assigns hardware to wires
 /////////// SEG7 /////////
 wire        [7:0]  HEX0;
 wire        [7:0]  HEX1;
 wire        [7:0]  HEX2;
 wire        [7:0]  HEX3;
 wire        [7:0]  HEX4;
 wire        [7:0]  HEX5;
 /////////// SW //////////
 wire        [9:0]  SW;
//Assigns count to the switches
 assign SW[3:0] = count;

// Instantiates hardware
 lab1_part2 Lab1_DUT (.HEX0(HEX0), .HEX1(HEX1), .HEX2(HEX2),
    .HEX3(HEX3), .HEX4(HEX4), .HEX5(HEX5),
    .KEY(KEY), .LEDR(LEDR), .SW(SW));

 initial begin
     count = 4'b0000;
     //Declares count as zero
     repeat (16) begin
     //Loops 16 times for each output
   #100
   //Prints all outputs after a time delay
   $display("in = %b, out = %b", count, HEX1[6]);
   $display("in = %b, out = %b", count, HEX1[5]);
   $display("in = %b, out = %b", count, HEX1[4]);
   $display("in = %b, out = %b", count, HEX1[3]);
   $display("in = %b, out = %b", count, HEX1[2]);
   $display("in = %b, out = %b", count, HEX1[1]);
   $display("in = %b, out = %b", count, HEX1[0]);

   $display("in = %b, out = %b", count, HEX0[6]);
   $display("in = %b, out = %b", count, HEX0[5]);
   $display("in = %b, out = %b", count, HEX0[4]);
   $display("in = %b, out = %b", count, HEX0[3]);
   $display("in = %b, out = %b", count, HEX0[2]);
   $display("in = %b, out = %b", count, HEX0[1]);
   $display("in = %b, out = %b", count, HEX0[0]);
//Increments count
   count = count + 4'b0001;
    end
  end
endmodule
```

Figure 8: tb_lab1_part2.v

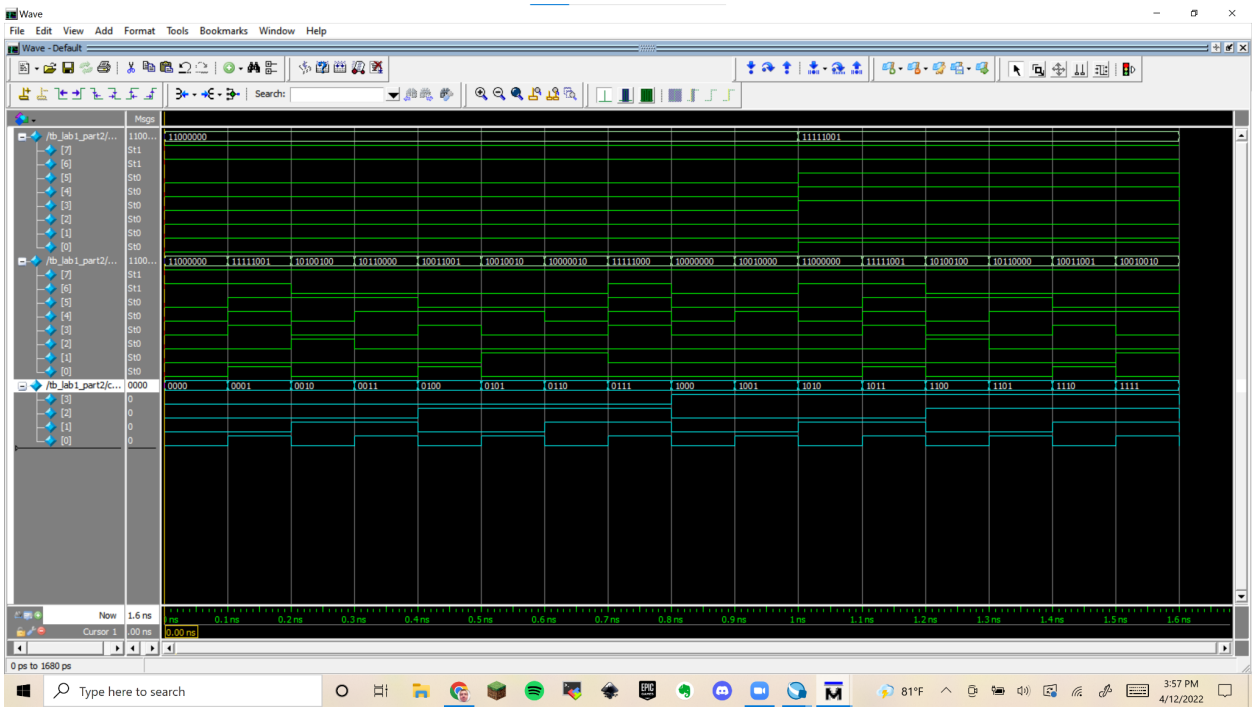


Figure 9: Testbench Simulation of tb_lab1_part2.v

Conclusion: After completing the lab, I have become more familiar with the Quartus and Modelsim environments as well as the verilog HDL, and I am eager to apply some concepts such as the technique I was playing around with in Part 1 to future labs in this class.