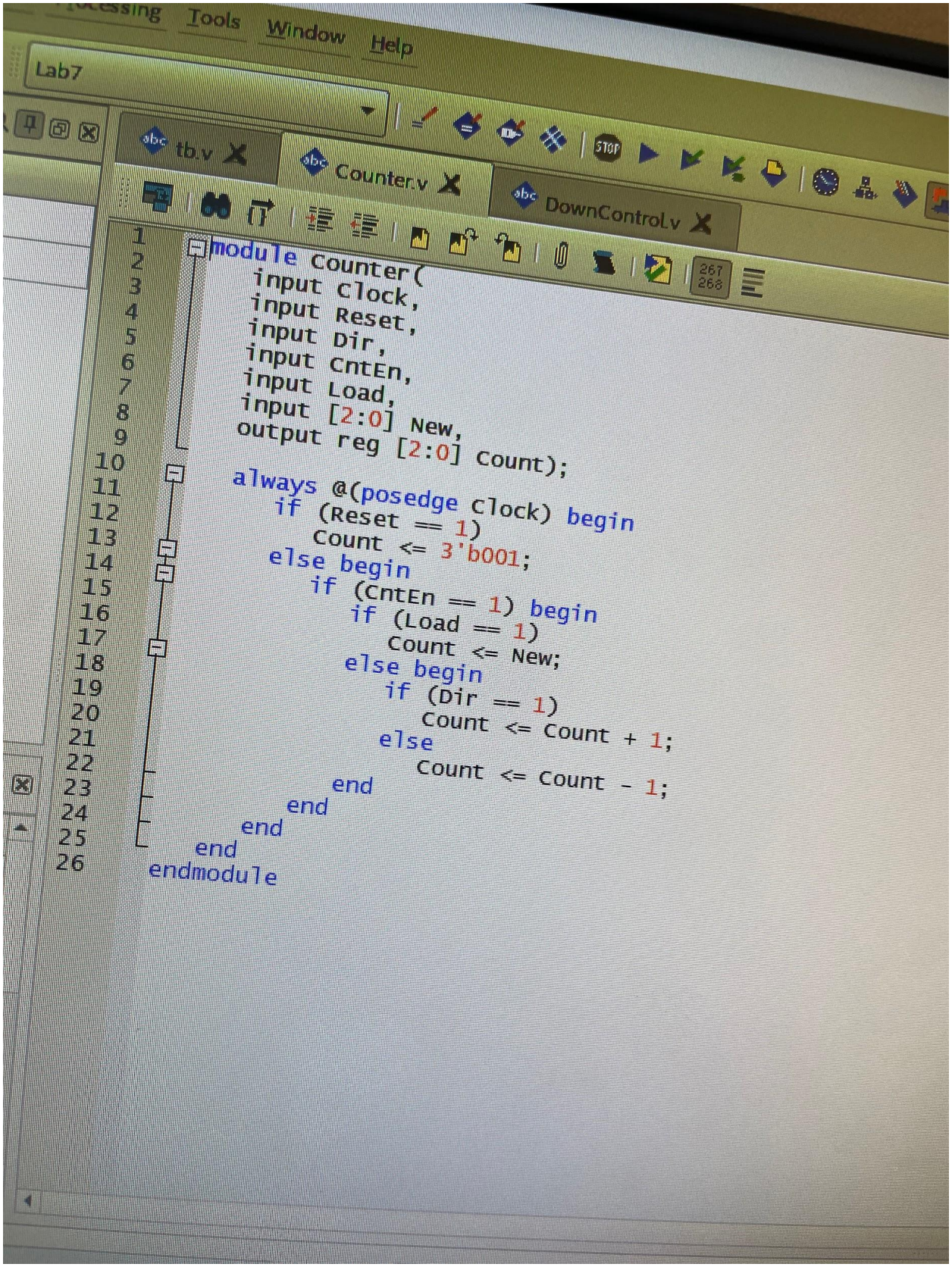
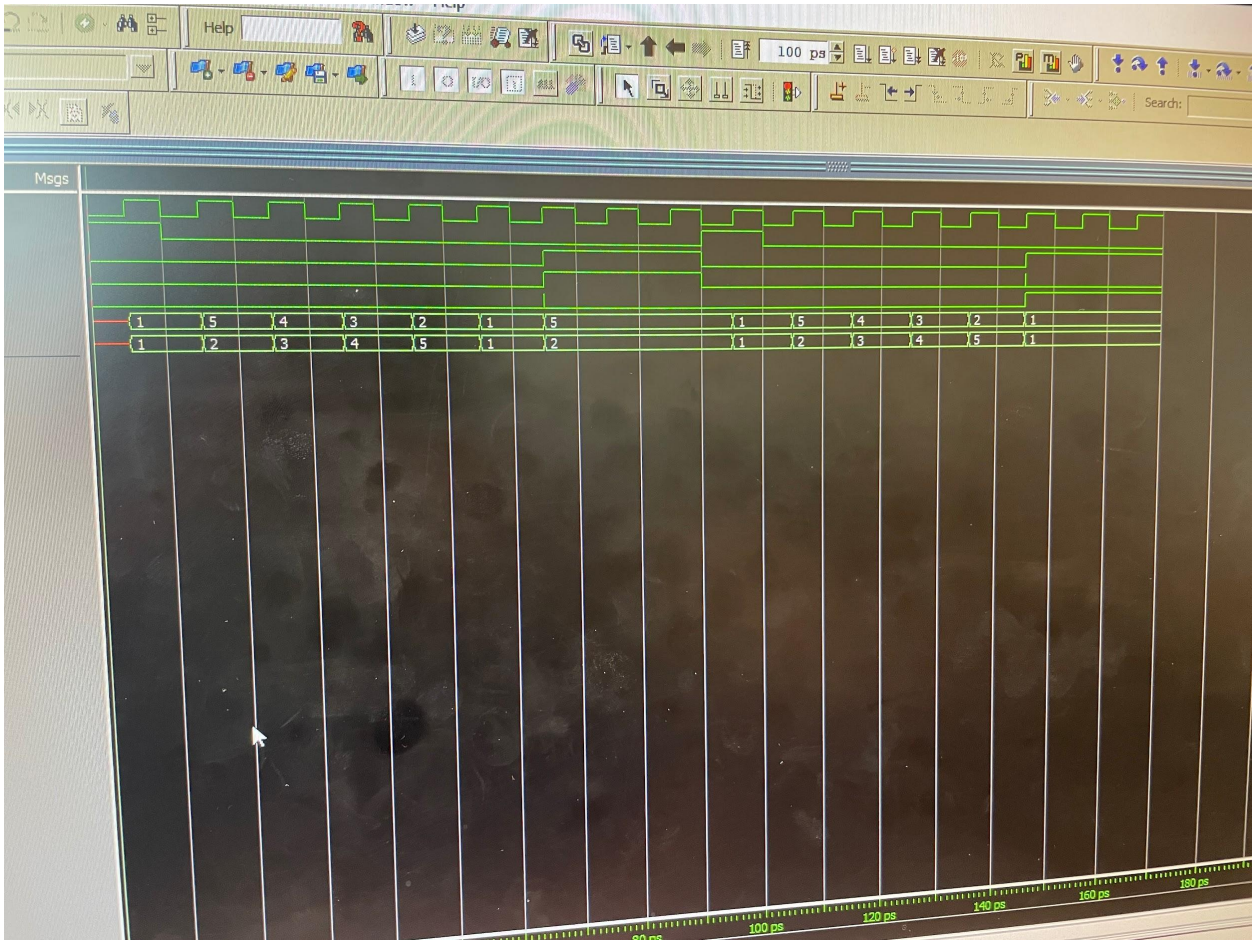
Ayden Dauenhauer
Seana Corners

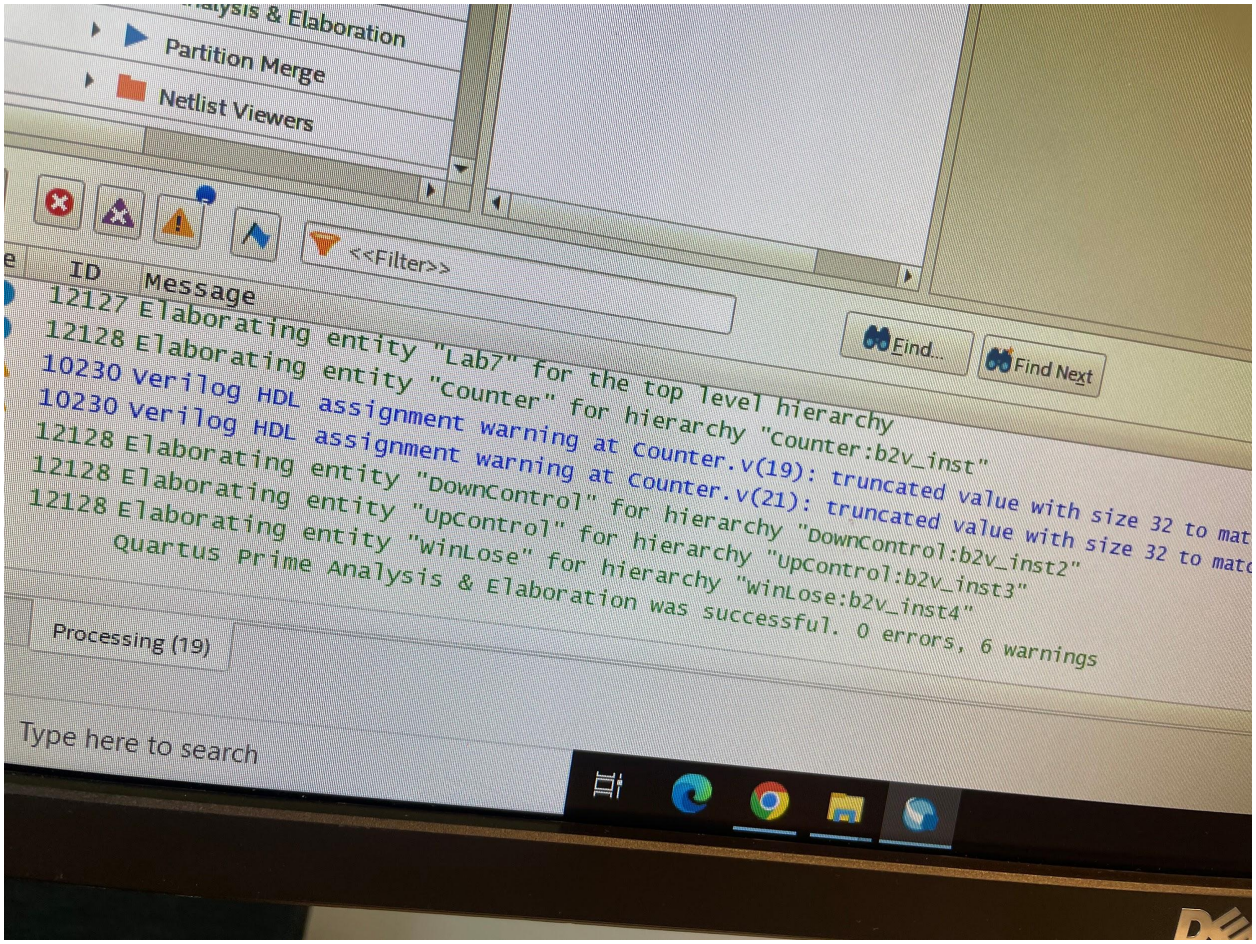
Lab 7 Post Lab

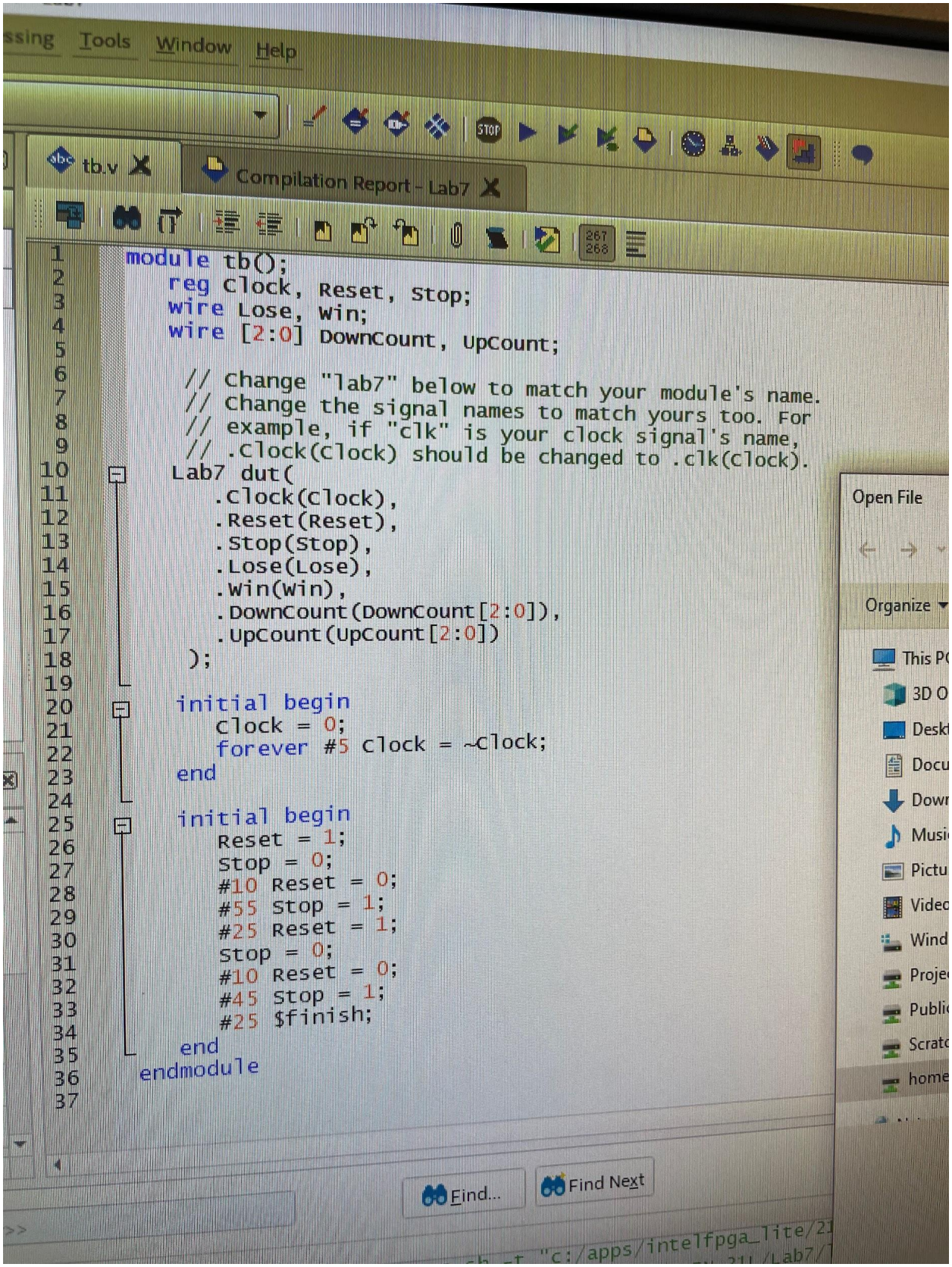
1. Introduction: In this lab we will be creating a game that will test a person's reaction time. One counter counts up and the other counter counts down. The objective of the game would be for the game to stop when the same value is shown. To execute this game we will use a 3-bit counter to express 8 possible values from 0 to 7 (an even number of values). They will always match at a value if they run through their sequence because they are running on the same clock. Moreover, we are going to change the way the counter behaves in order for the values to be between 1 and 5. We will also call "stop" which will be the inverse to a hypothetical button being pressed. The button being pressed serves as beginning the counting of the two counters. Finally, there is also an indication of the results of the game via signals. The first signal is a "win" signal which is when the two counter values match. The second signal is a "lose" signal which is when the two counters do not match. Stop is asserted for both signals. When "stop" is not asserted, the counter continues to count without the assertion of "win" or "lose".

2. If we were to change our design to have the counters count from 2 to 6 (or 6 to 2) instead of only 1 to 5, there are a couple things that we would need to change. The Verilog code needs to be changed, replacing the 3'b101 with 3'b110 and the 1'b1 to 2'b10. Next, the schematic should include a 2 bit input/output wherever the old 1'b1 and new 2'b10 is since an additional bit is needed.

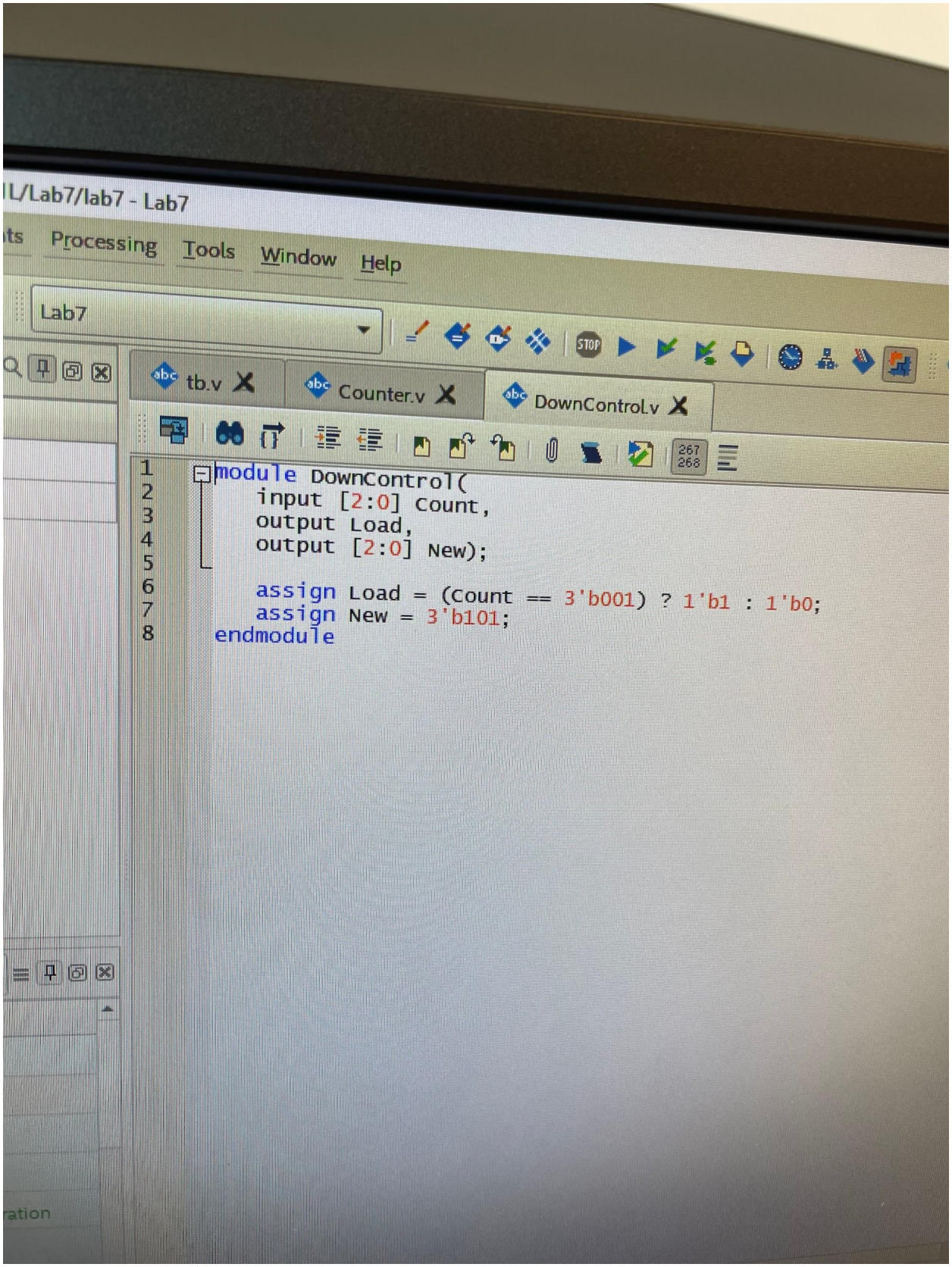

Tools
Window
Help
Lab7
tb.v
Counter.v
DownControl.v
module Counter(
input Clock,
input Reset,
input Dir,
input CntEn,
input Load,
input [2:0] New,
output reg [2:0] Count);
always @(posedge Clock) begin
if (Reset == 1)
Count <= 3'b001;
else begin
if (CntEn == 1) begin
if (Load == 1)
Count <= New;
else begin
if (Dir == 1)
Count <= Count + 1;
else
Count <= Count - 1;
end
end
end
end
endmodule

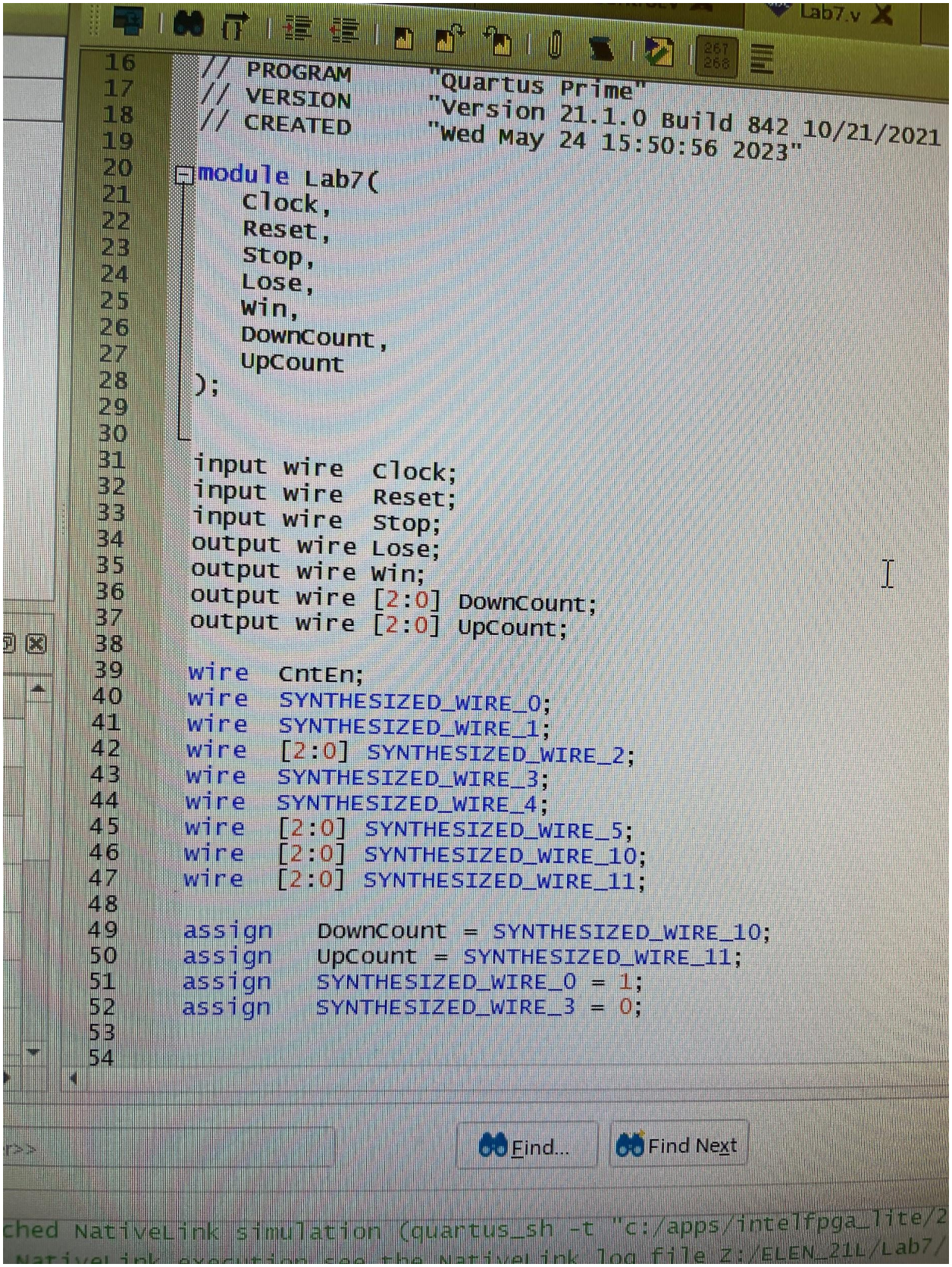

Help
100 ps
Search:
Msgs
1 5 4 3 2 1 5 1 5 4 3 2 1
1 2 3 4 5 1 2 1 2 3 4 5 1
100 ps
120 ps
140 ps
160 ps
180 ps

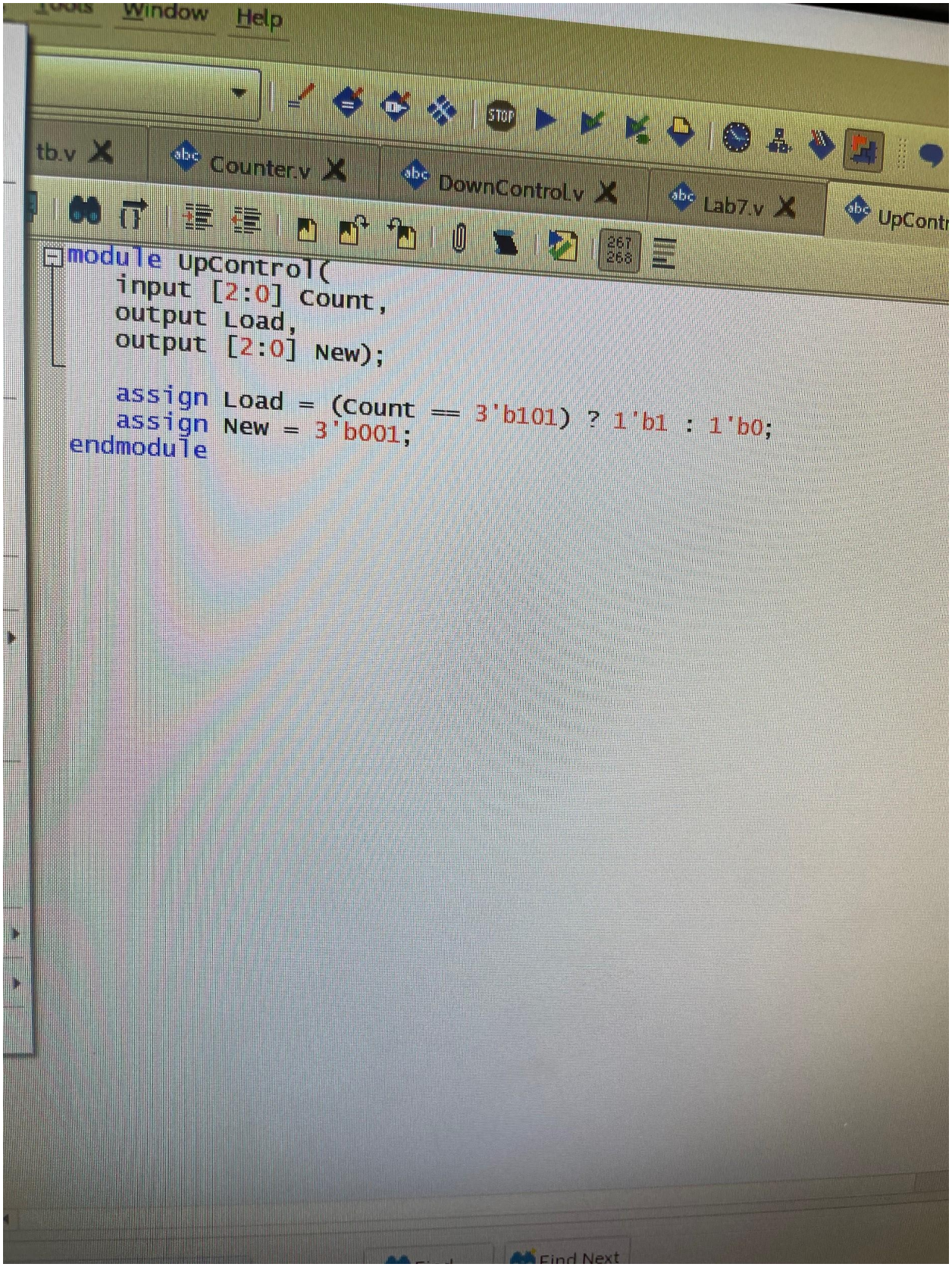

Partition Merge
Netlist Viewers
<<Filter>>
Find...
Find Next
ID Message
12127 Elaborating entity "Lab7" for the top level hierarchy
12128 Elaborating entity "counter" for hierarchy "counter:b2v_inst"
10230 Verilog HDL assignment warning at counter.v(19): truncated value with size 32 to mat
10230 Verilog HDL assignment warning at counter.v(21): truncated value with size 32 to mat
12128 Elaborating entity "DownControl" for hierarchy "DownControl:b2v_inst2"
12128 Elaborating entity "UpControl" for hierarchy "UpControl:b2v_inst3"
12128 Elaborating entity "WinLose" for hierarchy "WinLose:b2v_inst4"
Quartus Prime Analysis & Elaboration was successful. 0 errors, 6 warnings
Processing (19)
Type here to search

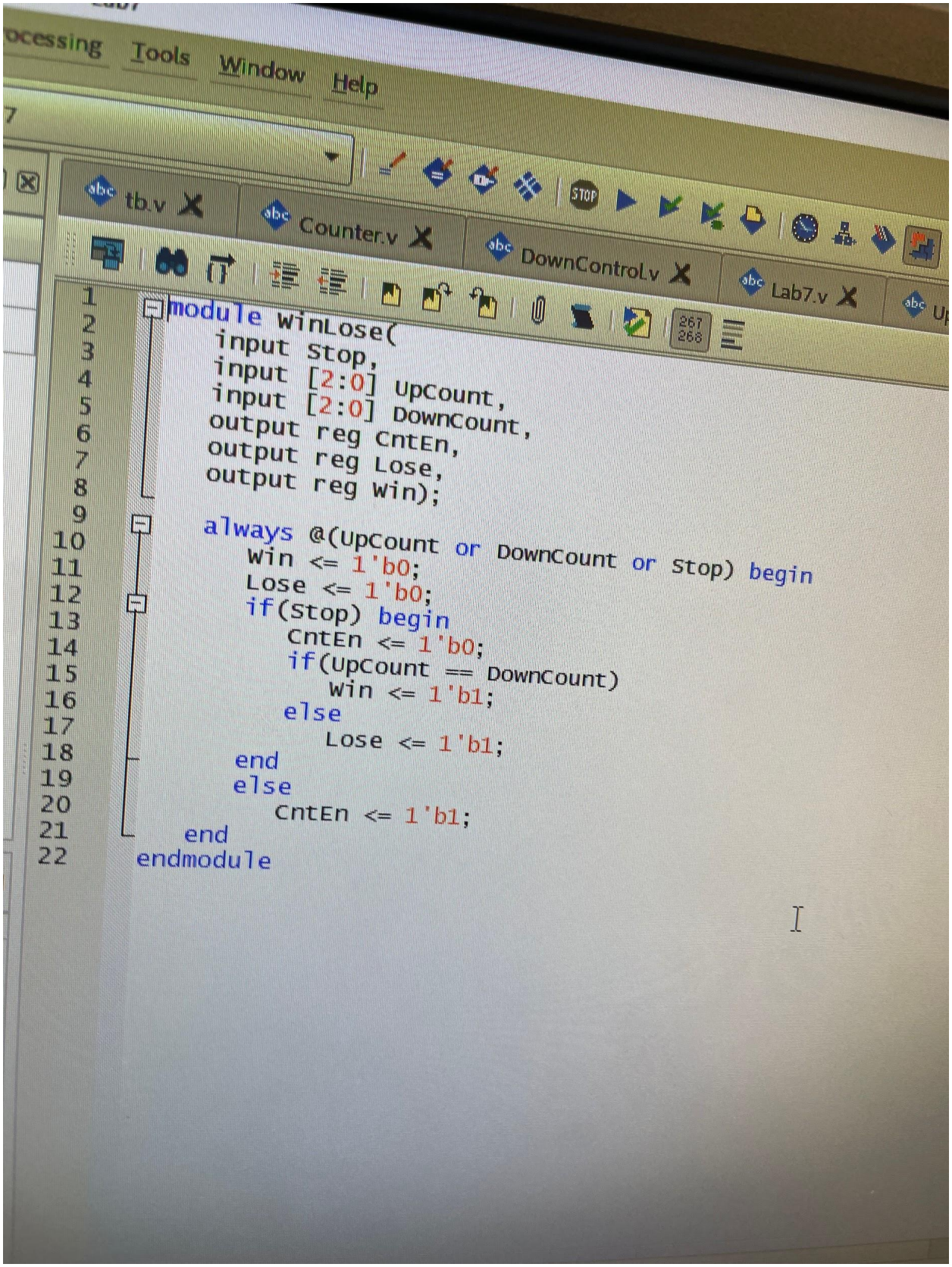


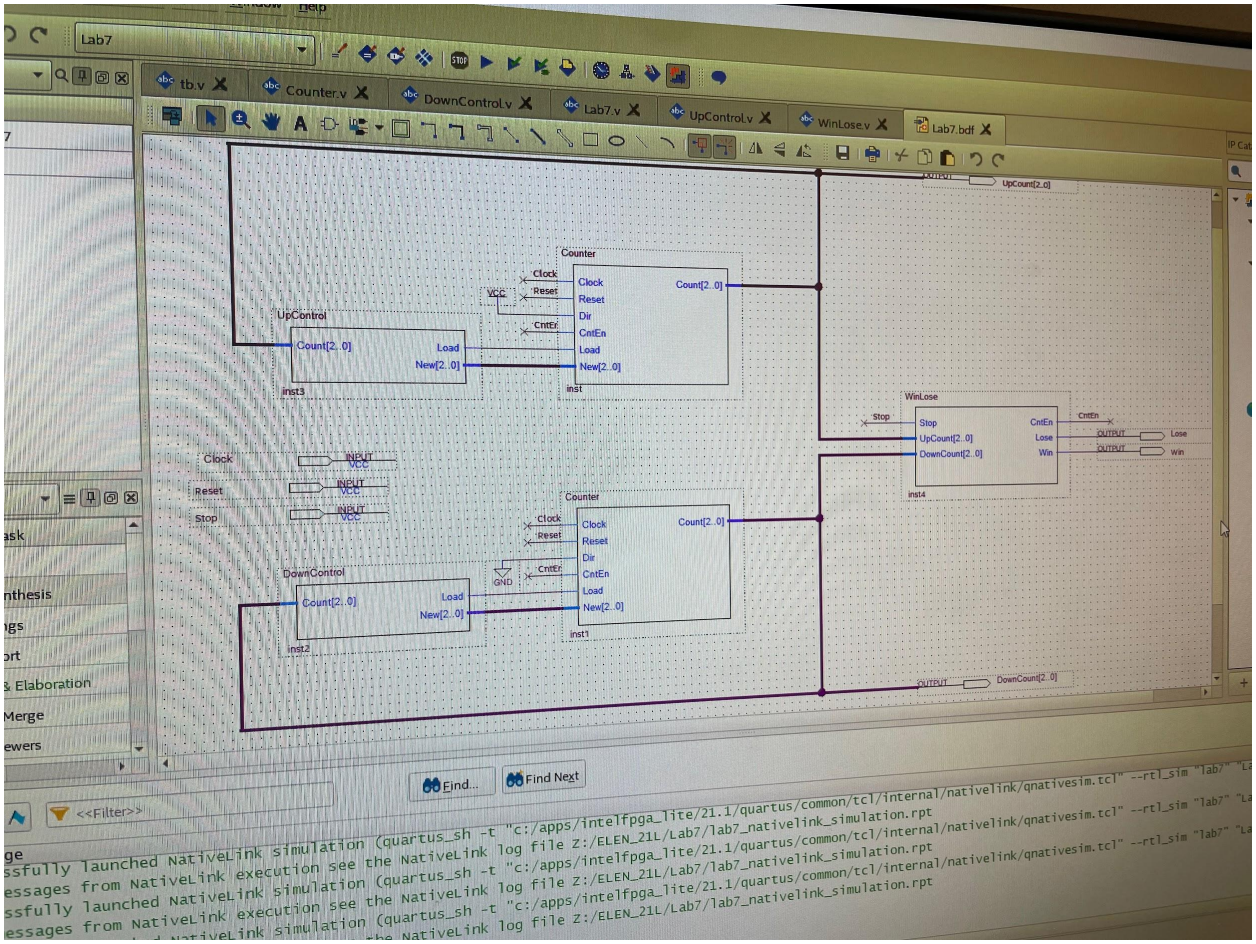
```
module tb();
    reg Clock, Reset, Stop;
    wire Lose, Win;
    wire [2:0] DownCount, UpCount;

     // Change "lab7" below to match your module's name.
     // Change the signal names to match yours too. For
     // example, if "clk" is your clock signal's name,
     // .Clock(Clock) should be changed to .clk(Clock).
    Lab7 dut(
        .Clock(Clock),
        .Reset(Reset),
        .Stop(Stop),
        .Lose(Lose),
        .Win(Win),
        .DownCount(DownCount[2:0]),
        .UpCount(UpCount[2:0])
     );

    initial begin
        Clock = 0;
        forever #5 Clock = ~Clock;
    end

    initial begin
        Reset = 1;
        Stop = 0;
        #10 Reset = 0;
        #55 Stop = 1;
        #25 Reset = 1;
        Stop = 0;
        #10 Reset = 0;
        #45 Stop = 1;
        #25 $finish;
    end
endmodule
```


L/Lab7/lab7 - Lab7
Processing
Tools
Window
Help
Lab7
tb.v
Counter.v
DownControl.v
1 module DownControl(
2 input [2:0] Count,
3 output Load,
4 output [2:0] New);
5
6 assign Load = (Count == 3'b001) ? 1'b1 : 1'b0;
7 assign New = 3'b101;
8 endmodule


Lab7.v
// PROGRAM "Quartus Prime"
// VERSION "Version 21.1.0 Build 842 10/21/2021
// CREATED "Wed May 24 15:50:56 2023"
module Lab7(
Clock,
Reset,
Stop,
Lose,
Win,
DownCount,
UpCount
);
input wire Clock;
input wire Reset;
input wire Stop;
output wire Lose;
output wire Win;
output wire [2:0] DownCount;
output wire [2:0] UpCount;
wire CntEn;
wire SYNTHESIZED_WIRE_0;
wire SYNTHESIZED_WIRE_1;
wire [2:0] SYNTHESIZED_WIRE_2;
wire SYNTHESIZED_WIRE_3;
wire SYNTHESIZED_WIRE_4;
wire [2:0] SYNTHESIZED_WIRE_5;
wire [2:0] SYNTHESIZED_WIRE_10;
wire [2:0] SYNTHESIZED_WIRE_11;
assign DownCount = SYNTHESIZED_WIRE_10;
assign UpCount = SYNTHESIZED_WIRE_11;
assign SYNTHESIZED_WIRE_0 = 1;
assign SYNTHESIZED_WIRE_3 = 0;
Find...
Find Next
ched NativeLink simulation (quartus_sh -t "c:/apps/intelfpga_lite/2
NativeLink execution see the NativeLink log file Z:/ELEN_21L/Lab7/


Window
Help
STOP
tb.v
Counter.v
DownControl.v
Lab7.v
UpContr
267
268
module UpControl(
input [2:0] Count,
output Load,
output [2:0] New);
assign Load = (Count == 3'b101) ? 1'b1 : 1'b0;
assign New = 3'b001;
endmodule
Find Next


ocessing Tools Window Help
tb.v
Counter.v
DownControl.v
Lab7.v
module winLose(
input Stop,
input [2:0] UpCount,
input [2:0] DownCount,
output reg CntEn,
output reg Lose,
output reg Win);
always @(UpCount or DownCount or Stop) begin
Win <= 1'b0;
Lose <= 1'b0;
if(Stop) begin
CntEn <= 1'b0;
if(UpCount == DownCount)
Win <= 1'b1;
else
Lose <= 1'b1;
end
else
CntEn <= 1'b1;
end
endmodule


Lab7
tb.v
Counter.v
DownControl.v
Lab7.v
UpControl.v
WinLose.v
Lab7.bdf
Counter
Clock
Reset
Dir
CntEn
Load
New[2..0]
Count[2..0]
UpControl
inst3
inst
WinLose
Stop
UpCount[2..0]
DownCount[2..0]
Lose
Win
inst4
INPUT
VCC
GND
DownControl
inst2
inst1
OUTPUT
Find...
Find Next
<<Filter>>
Elaboration
Merge
successfully launched NativeLink simulation (quartus_sh -t "c:/apps/intelfpga_lite/21.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "lab7" "lab7"
successfully launched NativeLink simulation (quartus_sh -t "c:/apps/intelfpga_lite/21.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "lab7" "lab7"