Khoa Tran

EE 271

July 20, 2020

Lab 3 Report

**Procedure**

**Task #1**

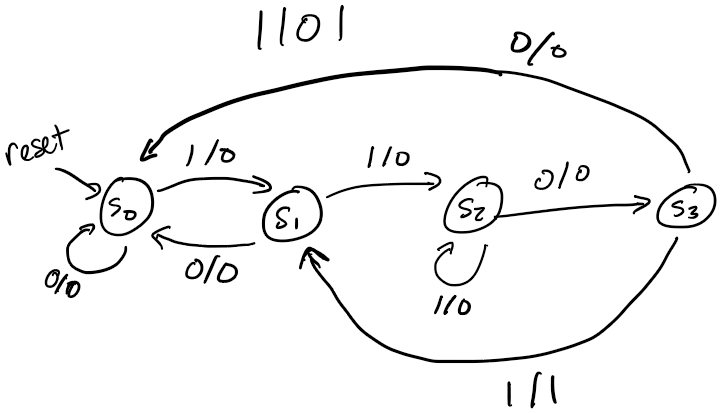
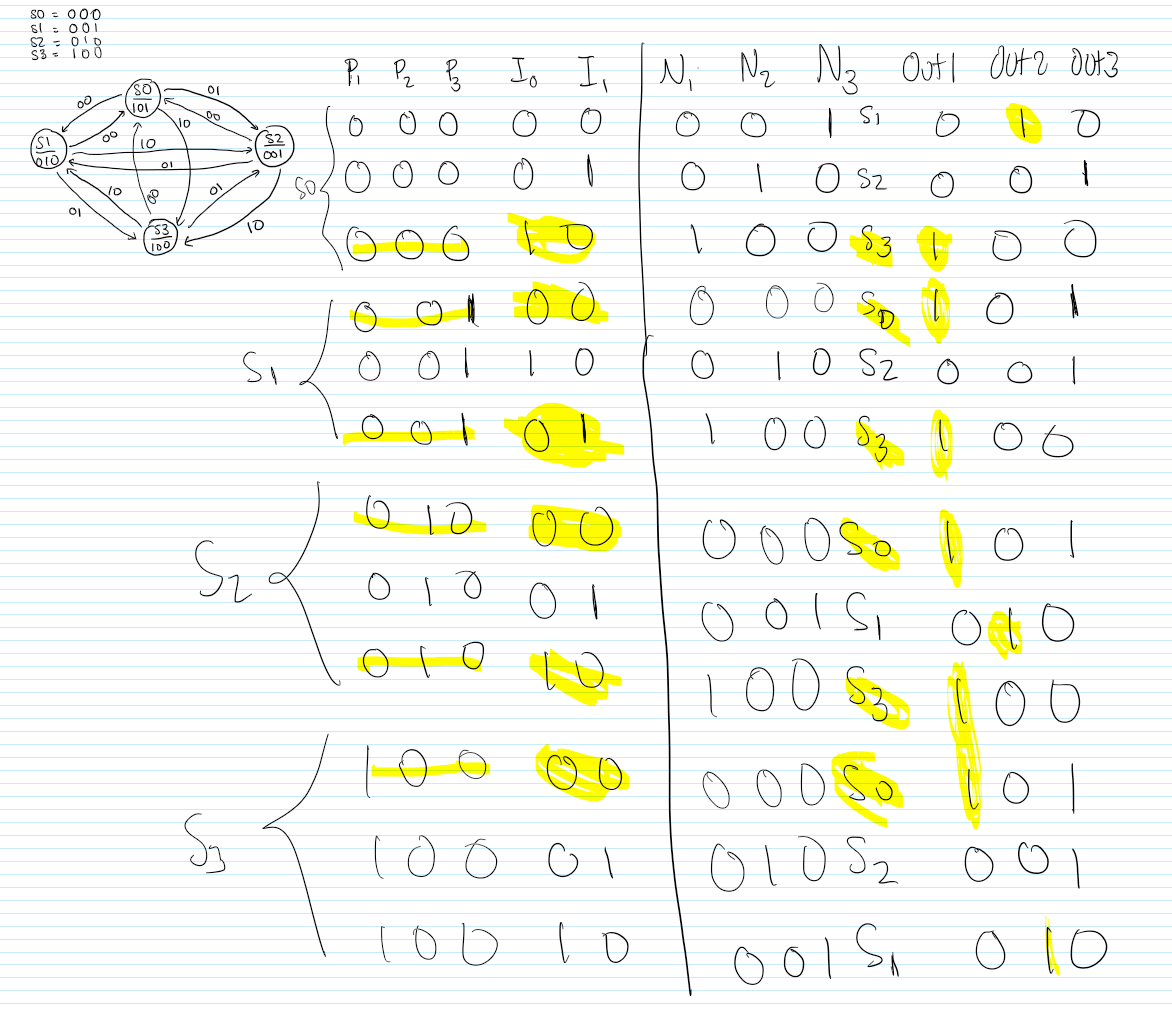
Approaching this problem, I first drew up the state diagram in order to figure out the number of states that I need and when to progress to each state depending on the input in order to reach 1101 and output 1. After this, I used the previous fsm code and modified it in order to have another state and change the output equations as well as the conditions for progressing to a different state.

Figure 1: State diagram for 1101 sequence detector

**Task #2**

Approaching this problem, I tried to understand the output sequence with the corresponding input sequence and noticed that there are a total of 4 states that the outputs need to be in order to cover all the possible patterns of each wind conditions. From recognizing this, I drew up the state diagram, which is shown below with the corresponding state table in order to understand what the equations for each output is and the conditions of progressing on to the next state. Afterwards, I implemented this design on Verilog with two inputs and three outputs that correspond to the individual switches and LEDRs.

Figure 2: State diagram and state table for wind indicators

**Results**

**Task 1:**

A screenshot of a computer

Description automatically generatedFor the first part, I tested the 101-sequence detector in order to check for the correct output whenever 101 is detected. For the second part, I tested the 1101 sequence detector to see if out is 1 when the sequence is detected. For both simulations, the input and output progresses if the clock is at the positive edge.

Figure 3: The waveform generated by the FSM detector of sequence 101

A screenshot of a computer

Description automatically generatedFigure 4: The waveform generated by the FSM detector of sequence 1101

**Task 2:**

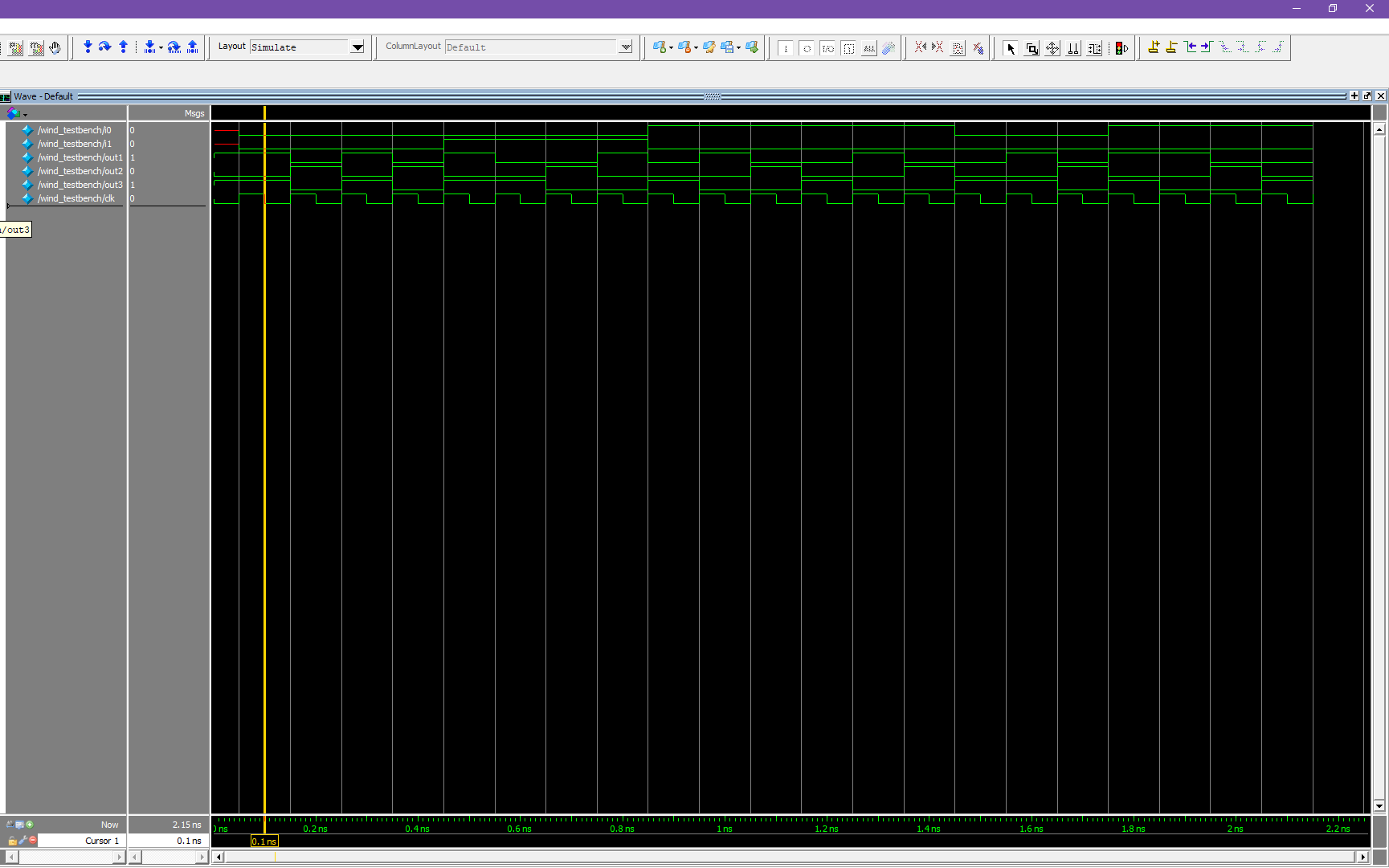
This simulation tested the inputs of i0 and i1 with the outputs of out1, out2, and out3 in order to check if the input wind conditions outputs the correct sequence of out1, out2, and out3. With the correct sequence, the LEDR can output the correct pattern with the specified wind conditions through inputs i0 and i1. The inputs only take effect when the clock hits the positive edge.

Figure 5: The waveform generated by the wind module

The size of FSM (wind module) is 30+28-26-26, which equals 6. This is the size of the design in terms of FPGA logic and DFF resources without the cost of the clock\_divider

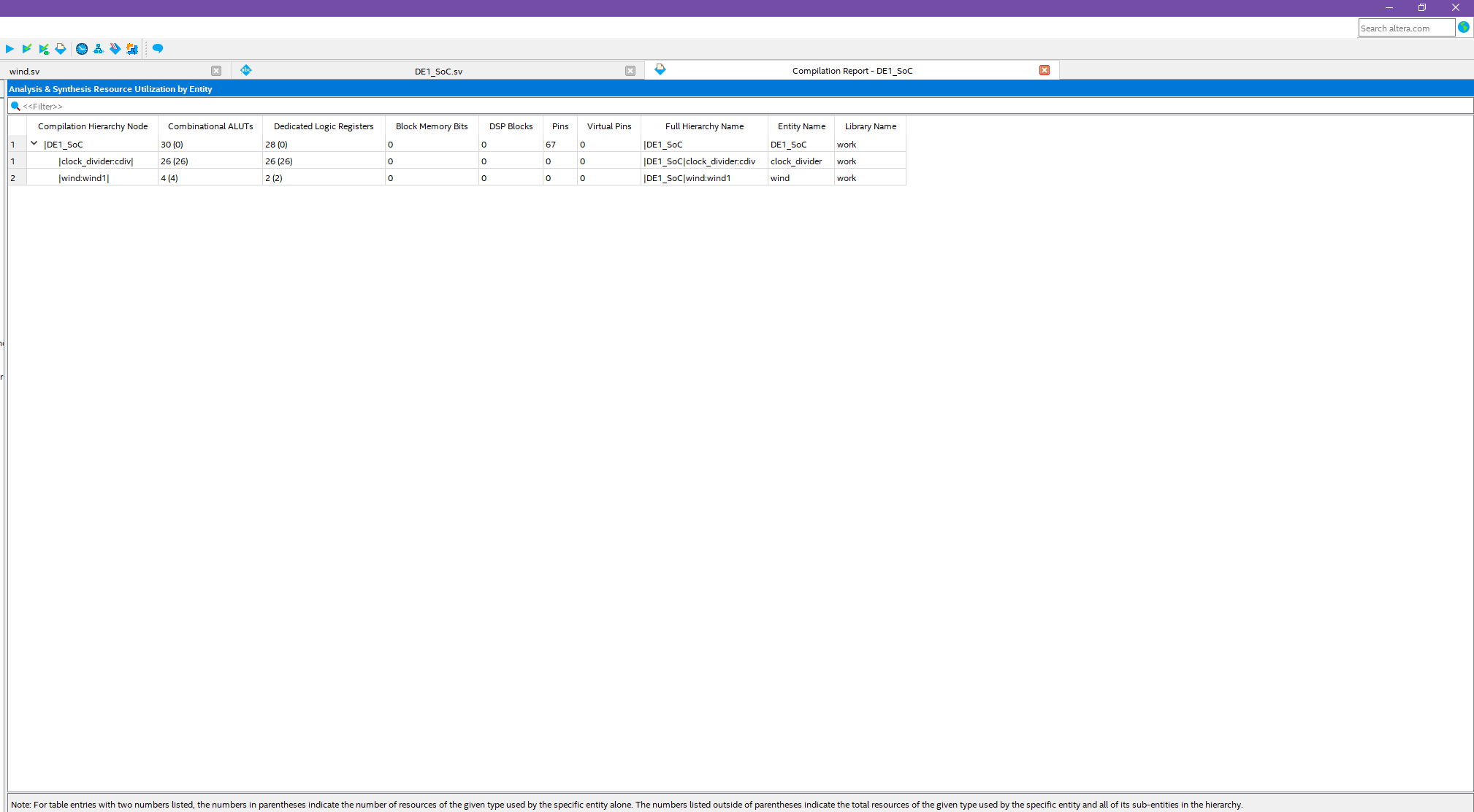
Figure

Figure 6: Analysis and Synthesis Resource Utilization of the wind module

**Final Product**

Overall, this project was designed to learn how to develop a FSM system on Verilog in order to learn how to convert a state diagram and a state table onto code with different states and understanding how and when to progress to another state. The project also is developed in order to show the output conditions as well as understand the progressing of the clock divider. For the first task, it is asked to develop the 1101 sequence detector and I designed so it would detect 1101 with the clock and output 1. For the second part, I was tasked to develop a detector of wind conditions and output a sequence of LEDR patterns for each specific conditions. I developed like asked as for the calm condition, it flickers between 101 and 010. For right to left and left to right, the LEDR pattern repeatedly cycle through in which the light moves from right to left or left to right in respect to the input condition. Overall, the finished project is similar to what was asked.

**Appendix**

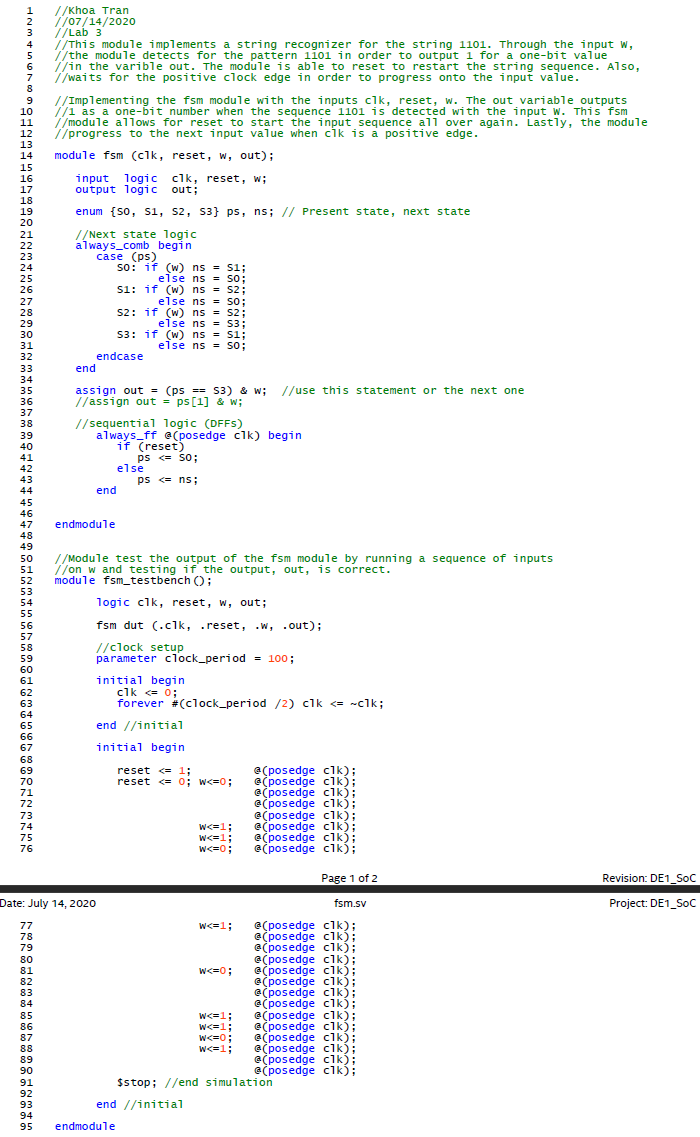


Figure 7: The FSM 1101 sequence detector file

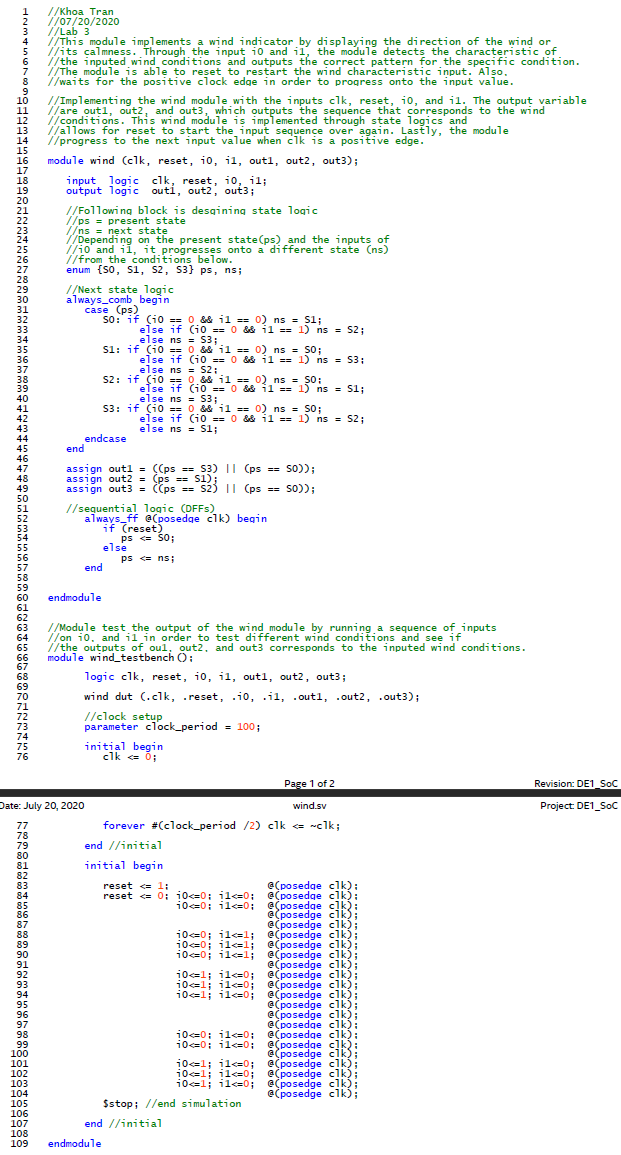


Figure 8: The wind file

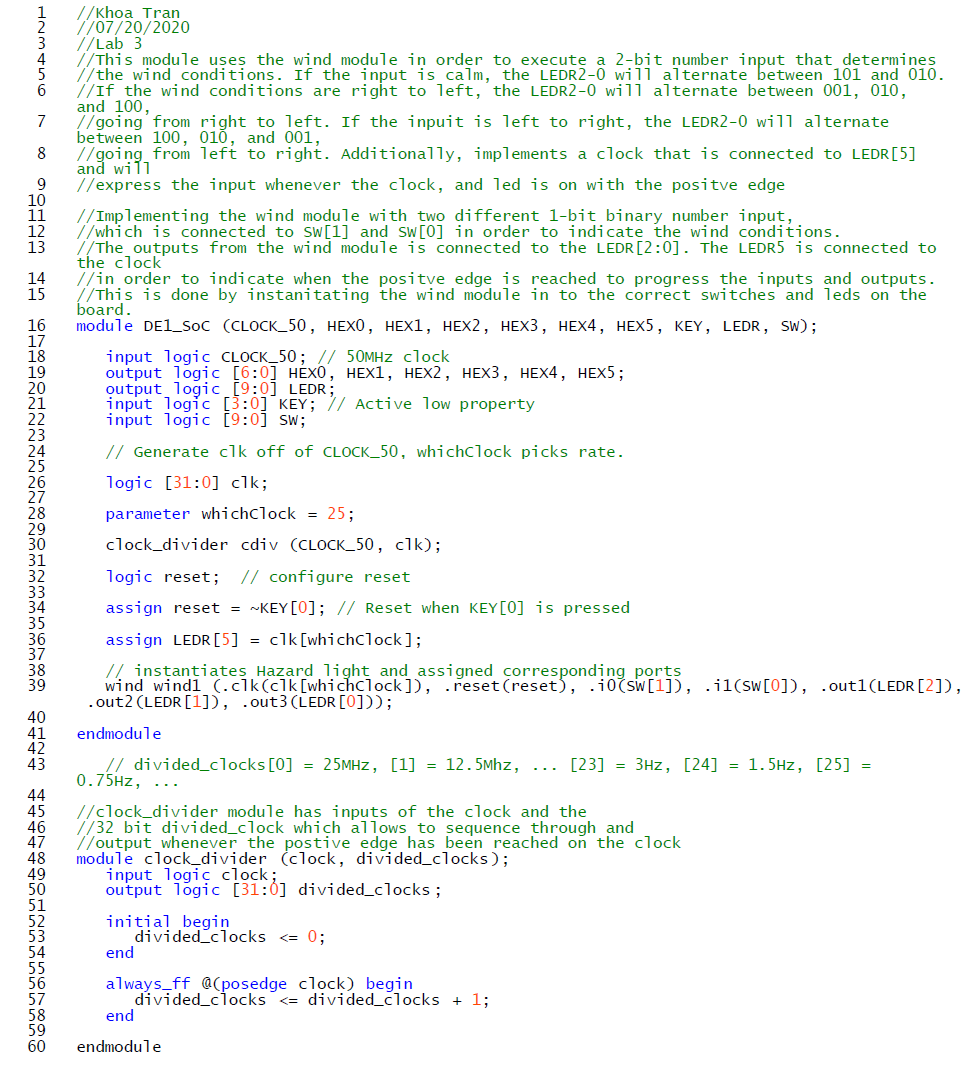


Figure 9: The DE1\_SoC file