Erich Viebrock

ECPE 174

**Post Lab #8**

Problem Summary:

An old car has tail lights controlled by 6 LEDs. In order to understand which LED is which, the names LA, LB, LC, RA, RB, and RC are used to represent each LED, from left to right. When the car is in IDLE (no switches used), all LEDs should be off. For the switch LEFT, the LEDs flash in a particular order: LB, LA and LC, all left LEDs on, and all off. For the switch RIGHT, the LEDs flash in a particular order: RB, RA and RC, all right LEDs on, and all off. In order to signify HAZARD lights are on, all LEDs flash on, and all LEDs flash off. When LEFT and RIGHT are both turned on, LEDs will go to IDLE. When more than one switch is used and HAZARD is involved, the sequence for HAZARD will begin. For all situations, LEDs will repeat their respected loop until the switch is changed. A clock divider is used in order to display LEDs at a rate conceivable to the human eye. The design is done in a Moore FSM.

Design Approach:

My design approach is the same as Lab #1, as the design is a Moore FSM. However, the design is written in Verilog. All of the case and special case statements are met, as depicted in the Vector Waveform.

Verification Procedure:

Initially, the design was written without a clock divider, in order to test the FSM with the Vector Waveform. After the design was correct and functioning in the desired manner, the next step was to create the clock divider. Interestingly, the clock divider for Verilog is very similar to VHDL. Once the code compiled for the clock divider, the final stage was to port the clock divider into the FSM. This was the simplest step, as it requires making the clock a wire, and Verilog’s version of PORT MAPing the clock into the FSM.

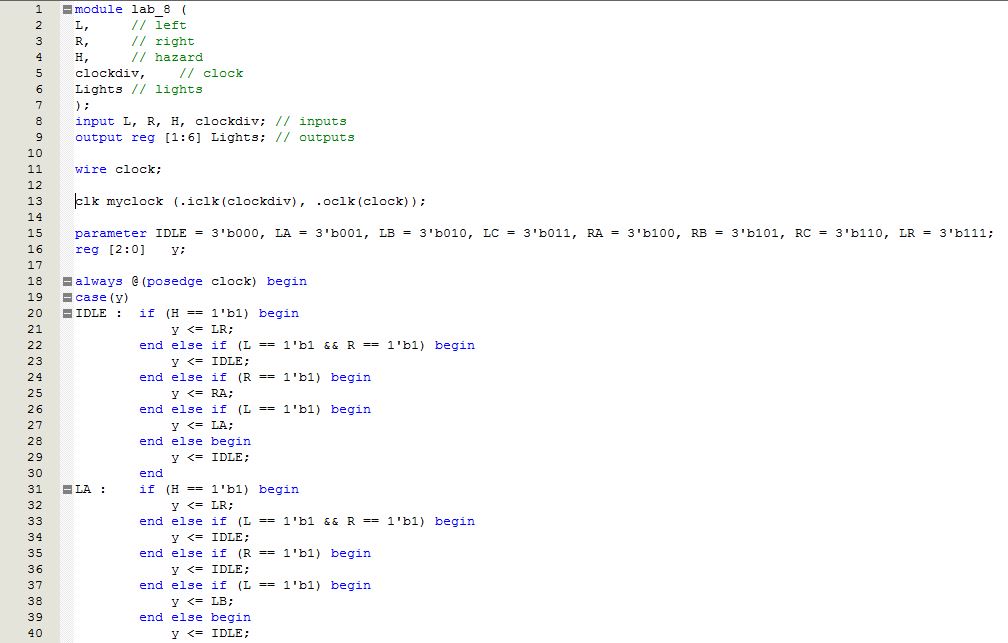
Post-Lab Questions:

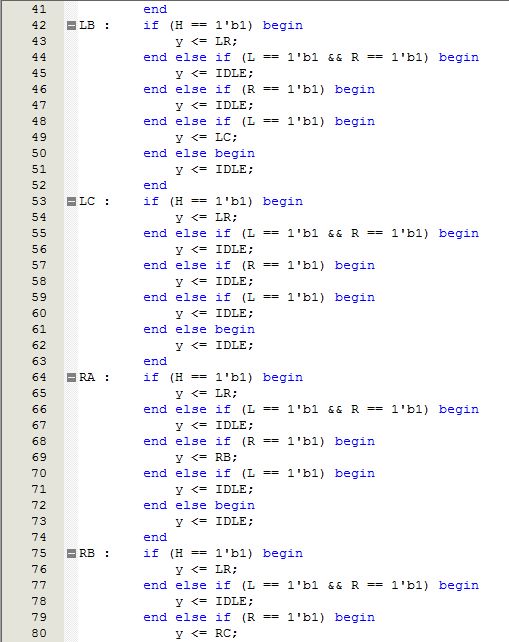
1. What, if any, differences do you see between the VHDL and Verilog designs based on the Quartus generated information? Explain these differences (or lack of them).
   1. In VHDL, there are 26 logic elements, whereas Verilog has 16 logic elements. Therefore, the delay in Verilog is shorter than VHDL. However, the delay in Verilog is only about 1 ns.
2. What do you see as the advantages and disadvantages between VHDL and Verilog? Which do you prefer? Why?

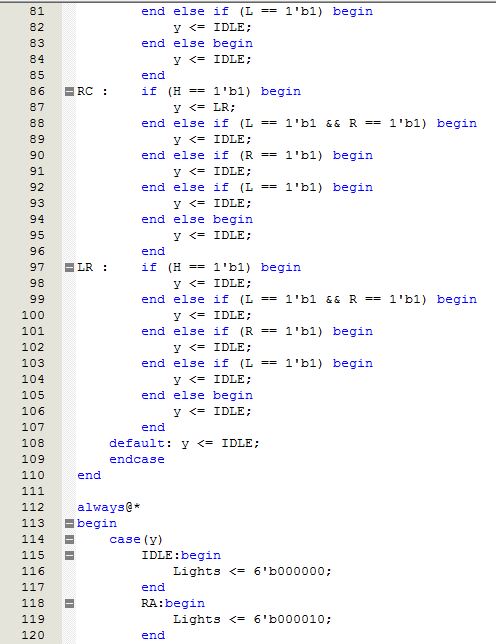
I see VHDL as a more user friendly language, as the syntax is very easy to learn. However, if you have had more of a C background in programming, Verilog would be a much more suitable language. I prefer VHDL, because I have more experience, and I think VHDL is easier to learn.

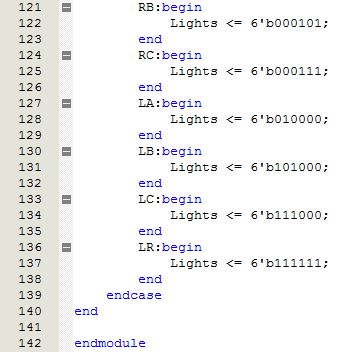
Appendix:

Verilog:

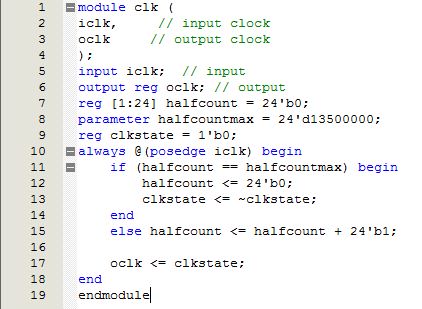




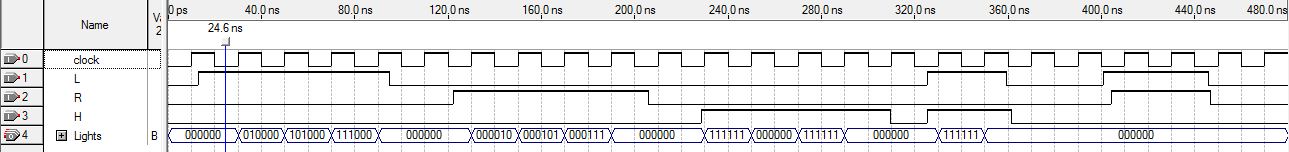




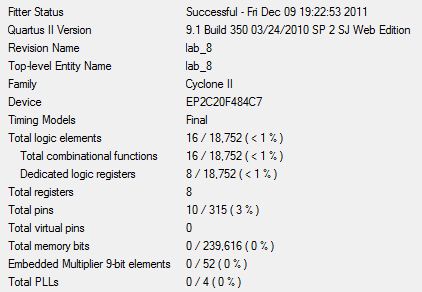
Verilog Clock Divider:



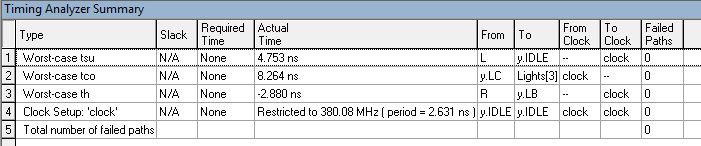
Verilog Simulation:



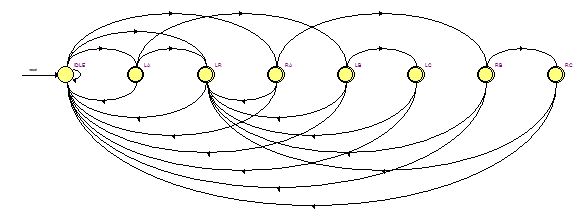
Verilog Fitter Summary:



Timing Summary:



Verilog State Diagram:



Verilog RTL:

