Erich Viebrock

ECPE 174

**Lab Report #1**

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.

Design Approach:

Our design is simple; by VHDL we were able to create a Moore FSM to replicate the behavior of an old car’s tail lights. Loops allowed us to create the desired output for all different scenarios.

Verification Procedure:

Our design did not run into error, but confusion arose after seeing the Moore, One-Hot, and Mealy diagrams of our code. Surprisingly, each diagram appeared the same. To our knowledge, this is almost never the case. Because of this, we believed there was error in our VHDL. However, this was not the problem, which ensued to further confusion and frustration. Then, the professor approved the similar diagrams, as they were desired output.

Post-Lab Questions:

* What, if any, difference is there between the two State Machine Viewer outputs? How do they compare to the one you designed in pre-lab step 1?
  + We found no difference in the State Machine Viewer outputs. This is because both methods will produce the simplest design possible.
  + In terms of the difference between Quartus’s State Machine Viewer and my personal state machine, there was slight difference. I was unaware an IDLE state was needed in the middle of the transition from LEFT to RIGHT, or RIGHT to LEFT. Therefore, this was the only difference.
* What equations did the synthesis tool use to implement the design? How does this compare with your results and ideas outlined in pre-lab step 3?
* Outline the results of the fitter resource usage summary: how much space on the FPGA does the design require? What types of elements were used?
* How fast could you clock the design according to Quartus? What happens if you attempt to clock it at that speed? Faster than that speed?
  + According to Quartus, the design can be clocked at 27 MHz.
  + If you attempt to clock it at that speed, the system will be able to produce output, but will be too fast for the human eye to recognize.
  + If you attempt to clock faster than that speed, the system will not be able to keep up and skip clock cycles.

Appendix: