EECE 444 Microprocessor System Design

LAB 3

Introduction:

The objective of this lab is to teach you how to design finite state machines in Verilog. The state machine will control the taillights of a 1965 Ford Thunderbird[1]. There are three lights on each side that operate to indicate left turn, right turn, break, and hazard as shown in figure 1. The lights operate in sequence to indicate left turn or right turn. Figure 2 shows the flashing sequence for left and right turns.

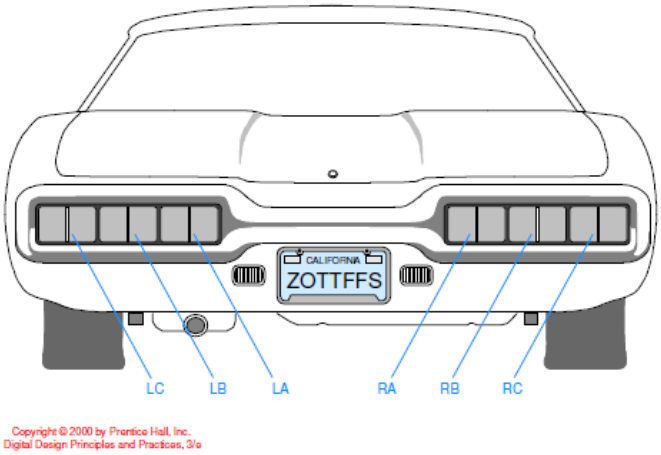

Figure 1

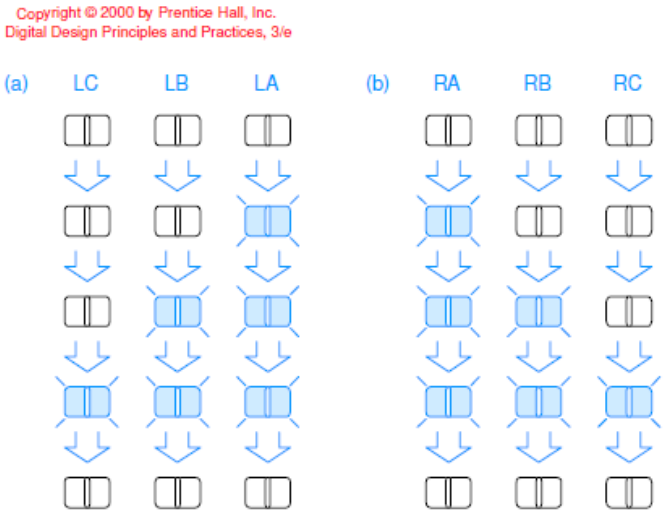

Figure 2

[1]This lab is derived from the 2nd edition of "Digital Design and Computer Architecture", Harris

This lab is divided into 4 parts: Design, Verilog coding, simulation and implementation.

1. **Design**

   Design a Finite State Machine to implement the taillight controller. The inputs to the FSM are: clk, reset, left, right, hazard, and break. The outputs will be the taillight display, LA, LB, LC, RA, RB, and RC. The state machine will act as follows:

   - On reset the FSM should enter a state where all lights are off.
   - When you press left, the LA light should light up then LB then LC, and finally all lights turn off again. The sequence should repeat as long as the left is pressed.
   - Pressing right should light up RA, RB, and RC following the same sequence and order as described before.
   - When break is pressed all lights should turn ON, and turn off when it is released.
   - When hazard is pressed, all lights should flash ON and OFF.
   - All lights should turn ON If break is pressed while left, right or hazard is ON at the same time. (Break has highest priority)
   - All lights should flash ON/OFF if hazard is pressed while left or right is ON.

   Start the design by constructing the STG diagram first. Indicate all possible states, inputs and outputs on the graph. It is up to you to choose binary or one-hot state encoding, justify your choice. Make sure to check for any identical states before you start building your design.

2. **Verilog code**

   Create a project to implement the taillight controller in Verilog. Define your inputs and outputs as described in the introduction.

3. **Simulations**

   Create a testbench file to test the functionality of your design. Simulate and take a snapshot of the timing diagram displaying all inputs and outputs. Test your design for all possible inputs.

4. **Implementation**

   Assign inputs and outputs to ports on your board, create the bit file and download your code to the FPGA. Demonstrate results to your TA.

   You will need to connect the clk input to the clock on the FPGA. The clock's frequency on the board is 50MHZ. Please create a new module, clkdiv, and add the following code to the module:

```
module clkdiv(
    input clk,
    input clr,
    output clk3
    );
reg [24:0]q;

always @ (posedge clk or posedge clr)
   begin
      if (clr==1)
         q<=0;
      else
         q<=q+1;
   end
assign clk3=q[24]; //1.5 Hz


endmodule
```

You will need to instantiate the clkdiv inside your program and use clk3 as your clock source. Connect clr to your reset signal.

What to Turn In?

1. Please indicate how many hours you spent working on this lab.
2. Your FSM design with all inputs and outputs labeled
3. Your States encoding
4. Verilog .v file
5. Testbench file
6. Include in your reports snapshot of your simulation
7. Describe how you tested the design on your FPGA, indicate any problems.