ECEN2350 Digital Logic – Lab 3

Fall, 2019

Due Wednesday, December 11th , 2019, at 4:00pm

10% of Course Grade (100 points)

Successful completion of this lab will require use of iVerilog, GTKWave, Quartus, and the DE10-Lite board. If you are having problems with any of these design tools, contact your instructor as soon as possible.

Everyone is responsible for turning in their own project. You may work with others to complete the lab.

This lab will require you to code and compile the design using iVerilog and Quartus, simulate the design using both GTKWave and/or textual output, and submit a working project and a lab report. The goal of Lab 3 is to extend your design knowledge to finite state machine (FSM) design techniques, and to create a testbench that simulates this FSM design.

Upon completion of this lab, you will be familiar with FSM design coding techniques, and instantiation and usage of memory blocks contained within the MAX10 FPGA.

Lab Description:

Lab3 consists of a design that emulates the behavior of taillights found on a 1965 Ford Thunderbird automobile.

https://www.youtube.com/watch?v=Qwzxn9ZPW-M shows how the taillights operate. This project will not emulate brake light behavior.

You should read the remainder of this document before beginning to design.

Grading

1. Successful completion and submission of base design 45%
2. Automation of design using MAX10 memory block 20%
3. Successful completion and submission of testbench 15%
4. Lab report. 20%

If you are unable to complete the base design, submit what you have for partial credit. Your lab report should fully explain the final state of your project.

Note: The lab report is mandatory. No Lab3 credit will be given if the lab report is not submitted. Document as much of the lab as you were able to complete. You may not submit a lab report that you did not write yourself.

Submission guidelines:

1. You must submit your project via Canvas.
2. Submit all files including the final report in a single .zip file. Make sure the folder names you use do not contain spaces or special characters. The name of the submitted zip file must contain your last name.
3. Verify that the .zip file you submit contains all files required to rebuild your project. This includes all verilog source and testbench files, and at a minimum the Quartus .qpf and .qsf files.
4. If your submitted project will not build successfully, you will lose credit. Test the contents in your .zip file before submission.
5. Late submission of the lab will result in a 20% reduction in your project grade. The deadline for late submission is Thursday, December 12th at 1159pm. No submissions will be accepted after this time.

Guidelines for lab report:

a) Your lab report should consist of the following sections:
    a. Problem statement
    b. Theory of operation
    c. State bubble diagram and state table
    d. Hierarchy of source files
    e. Description of testbench operation and output
    f. Summary of project success, what works and what doesn't
b) Keep the lab report concise.

DE10-Lite Definitions:

Per the DE10-Lite User Manual, the slide switches are defined as DOWN or Logic 0 when the switches are nearest the edge of the DE10-Lite board, and UP or Logic 1 when the switches are pushed toward the center of the DE10-Lite board.

With the DE10-Lite board positioned with the 7 segment displays and slide switches at the bottom of the board:

HEX5 is the leftmost 7 segment display, HEX0 is the rightmost 7 segment display.
SW[9] is the leftmost slide switch, SW[0] is the rightmost slide switch.
LEDR[9] is the leftmost LED, LEDR[0] is the rightmost LED.
KEY[0] is the pushbutton switch closest to the VGA video connector.

Detailed Requirements for Lab3 (Read carefully, get clarifications early)

1. You must code this lab as a FSM using separate current state, next state, and output logic blocks. These blocks can be separate Verilog modules, or simply separate groups of code. Use meaningful names to reference the states in your design.

2. Create a display update frequency that allows for easy viewing of the circuit operation.

3. Design the user interface as follows:
    a. Use KEY[0] as system reset. You may latch the pushbutton KEY[0] if you choose, however either latched or unlatched behavior is acceptable.
    b. SW[0] is the hazard light switch, 0 = off, 1 = on.
    c. SW[1] is the turn signal enable switch, 0 = off, 1 = on.
    d. KEY[1] will toggle between left turn signals (SW[1] = 0) and right turn signals (SW[1] = 1). Latching of KEY[1] is your choice.
    e. Use any hex display (or displays, if necessary) to display the numeric value of the current state register.
    f. The left turn signal will be displayed using LEDR[9:7], where LEDR[9] is the outside light. The right turn signal will be displayed using LEDR[2:0], where LEDR[0] is the outside light. All unused LEDRs should be turned completely off.
    g. The hazard indication will blink LEDR[9:7] and LEDR[2:0] simultaneously and at the same flash rate as the turn signal lights.
    h. The hazard condition has higher priority than turn signal lights.

4. Complete the design described above, where you manually manipulate the switches to operate the design.

5. Once step 4 is finished and functioning properly, add a MAX10 M9K memory block to your design. Based on the state of SW[9], the design will run normally using manual manipulation of the switches (SW[9] = 0), or will be automatically cycled through the states (SW[9] = 1) by using M9K outputs to emulate the switches.
    a. When SW[9] = 1, the display will spend approximately 5 seconds in the state where lights are off, approximately 5 seconds with the hazard lights flashing, approximately 5 seconds with the left turn signal lights flashing, approximately 5 seconds with the right turn signal lights flashing, then return to the lights off state. This loop will run continuously.

Testbench Requirements

1. **Simulate the design before adding the additional logic that automates the design using the M9K memory block. There is no expectation that the automated design is simulated.**

2. You must create a testbench that demonstrates aspects of the design. At a minimum:
    a. Simulate the behavior of the state machine by changing inputs.
    b. Simulate the behavior of the turn signal lights and hazard lights.

3. Your simulation output can be waveform captures, text file(s), or a combination of both.