Overall design flow:[1]

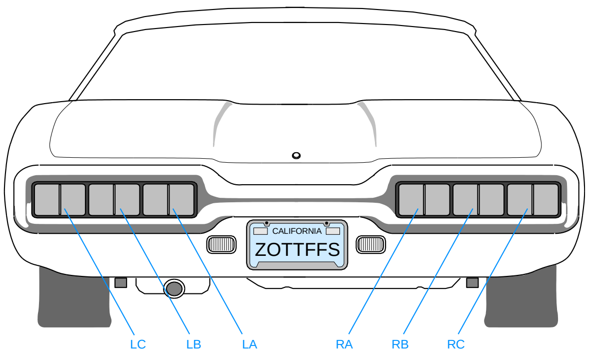
Big Picture: In this lab, you will design, write, test, and implement in hardware a finite state machine to simulate the taillights of a 1965 Ford Thunderbird.

Authorized Resources: For this lab, you may work in teams of two (including the prelab). You may seek help from any cadet or instructor, not limited to this course, and reference any publication in its completion. Online resources are acceptable. However, no resources containing solutions to the course homework, labs, exams, quizzes, or in class/out of class exercises are allowed. Your work (and your code) must **always** be your own. Normal documentation of all resources utilized is required.

Due Dates: Prelab – 1700 T23 via Gradescope
Demo – 1700 T25 in person or via Teams
Report – 1700 T26 via Gradescope

Background

The 1965 Ford Thunderbird has three lights on each side that operate in sequence to indicate the direction of a turn. Figure 1 shows the tail lights and Figure 2 shows the flashing sequence for (a) left turns and (b) right turns.



Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

Figure 1 – Thunderbird taillights

Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

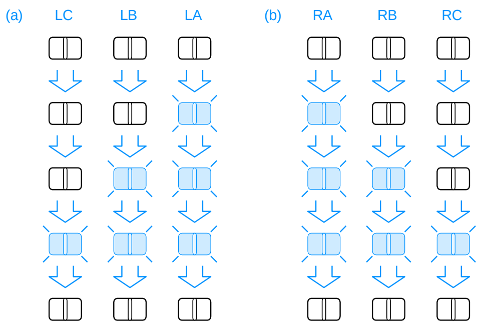


Figure 2 – Flashing sequence for taillights. The lights shaded light blue are illuminated.

[1] Modeled after a lab provided with instructor notes for Digital Design and Computer Architecture, David Money Harris & Sarah L. Harris, 2nd Edition

The objective of this lab is to create a turn signal that will engage the left and right light sets to mimic the Ford Thunderbird. This lab is divided into four parts: design (Prelab), VHDL creation, simulation, and implementation. If you follow the steps of FSM design carefully and ask questions at the beginning if a part is confusing, you will save yourself a great deal of time.

1. Prelab - FSM Design

Project functionality description

On reset, the FSM should immediately enter a state with all lights off. When you press left, you should see LA, then LA and LB, then LA, LB, and LC, then finally all lights off again. This pattern should occur even if you release left during the sequence. If left is still down when you return to the lights off state, the pattern should repeat. The logic for the right lights is similar. Finally, when both left and right switches are on your state machine should blink all lights on and off (implementing hazard lights).

Prelab Tasks (20 pts) Due Lesson 22 via Gradescope

You will use a MOORE FSM to implement the above functionality. Complete the following tasks to prepare yourself for the lab.

0. Create a new Vivado Project called **Lab3**. This will create your Lab3 folder where you can store your files.
1. Create a state transition diagram using the Lab3_StateTransitionDiagram_Template.pptx found in the Team under Lab 3. Take a screen shot and include it in the Lab3_Prelab_Template.docx found in the Team.
2. **Binary Encoding (TYPE YOUR ANSWERS INTO THE TEMPLATE)**
    a. Complete the encoding table that maps state names to encodings
    b. Complete the state transition table
    c. Complete the output table
3. **One-Hot Encoding (TYPE YOUR ANSWERS INTO THE TEMPLATE)**
    a. Complete the encoding table that maps state names to encodings
    b. Complete the state transition table
    c. Complete the output table
4. Write your next state and output equations using **One-Hot Encoding (TYPE YOUR ANSWERS INTO THE TEMPLATE)**

In the remainder of the lab, you will be creating your FSM in VHDL, testing it using a testbench, and then implementing your design on the FPGA. Since you are designing a synchronous sequential circuit, **we will be making use of a clock signal** to drive the logic. Finally, while not required for the Prelab, you should already be thinking about what **test cases** you will use to verify that your design is correct.