

Computer Assignment #3  
Due Su. 11/26

## 1965 Ford Thunderbird Tail Lights

Total Points: 55

### 1 Introduction

In this lab you will design and implement a finite state machine to control the tail lights of a 1965 Ford Thunderbird. There are three lights on each side that operate in sequence to indicate the direction of a turn. Figure 1 shows the taillights and Figure 2 shows the flashing sequence for (a) left turns and (b) right turns.

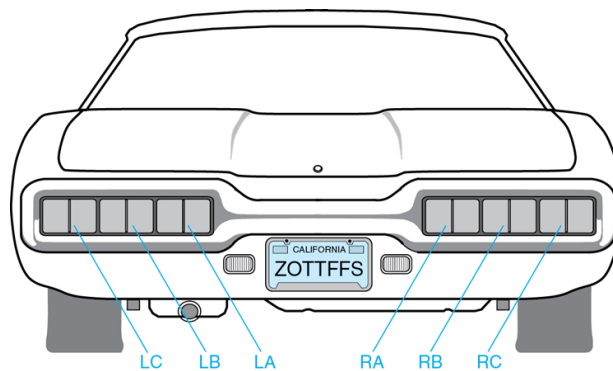


Figure 1: Thunderbird Tail Lights

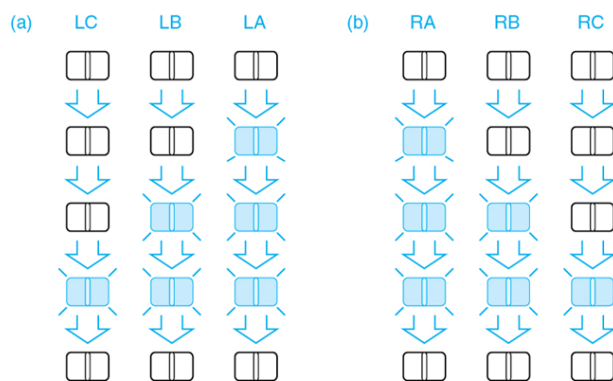


Figure 2: Flashing Sequence for T-Bird Tail Lights (a) Left Turn (b) Right Turn

The lab is divided into four parts: design, schematic entry, simulation, and implementation. Be sure to see the “What to Turn In” section at the end of the lab before you begin.

### 2 Design

Your FSM should have the following inputs and outputs:

- Inputs:  $clk$ ,  $reset$ ,  $left$ ,  $right$
- Outputs:  $LA$ ,  $LB$ ,  $LC$ ,  $RA$ ,  $RB$ ,  $RC$

where  $LA, LB, LC, RA, RB, RC$  are the lights for the left and right turn signals.

On reset, the FSM should enter a state with all lights off. When input *left* is pressed, the flashing sequence should be  $LA$ , then  $LA$  and  $LB$ , then  $LA, LB$ , and  $LC$ , then finally all lights off again. This pattern should occur even if *left* is released during the sequence and should continue if *left* is still down upon return to the lights off state (the pattern should repeat). The same operation works for *right*. You can assume that exclusively one of the inputs (*left* or *right*) can be asserted at once (i.e. only a left turn or right turn signal is on at a time). Also you should assume that the *clk* runs at a desired speed of 1 Hz.

Sketch the state transition diagram. define your state encodings. **Hint:** with careful choice of encoding, your output and next state logic can be quite simple. Give state transition and output tables. Give the next state and output Boolean equations. Sketch the schematic of the FSM. Note: this was all completed in HW #8.

### 3 Schematic Entry

Start Quartus and create a new project named “cpe100\_comp3\_xx” (where xx are your initials) and create your FSM schematic. Note that the flip-flop in the schematic editor is called “DFF”. It has asynchronous active-low set and reset inputs named “PRN” and “CLR”. You can connect “PRN” to VCC so that the element behaves as our ordinary flip-flop with reset.

### 4 Simulation

With your design completed, you are now able to use ModelSim to verify its operation. You will want to perform enough input variation to demonstrate that the FSM performs all functions correctly. Don't forget that your registers require a *clk* signal to advance state. When showing your simulation, your signals should be printed in the following order: *clk, reset, left, right, LC, LB, LA, RA, RB, RC*.

### 5 What to Turn In

1. (55 points) FSM: 1965 Ford Thunderbird Tail Lights
  - (a) (0 points) How many hours did you spend on the lab?
  - (b) (10 points) Full FSM design, including a completed state transition diagram for your FSM and schematic.
  - (c) (20 points) A printout of the Quartus II schematic.
  - (d) (20 points) A print out of the ModelSim simulation of the circuit. Make sure you test the circuit completely. Show the result for a reset, left signal, and right signal. Your signals should be printed in the following order: *clk, reset, left, right, LC, LB, LA, RA, RB, RC*.
  - (e) (5 points) A brief paragraph (2-4 sentences) highlighting what you learned on this lab assignment. List any difficulties you encountered.