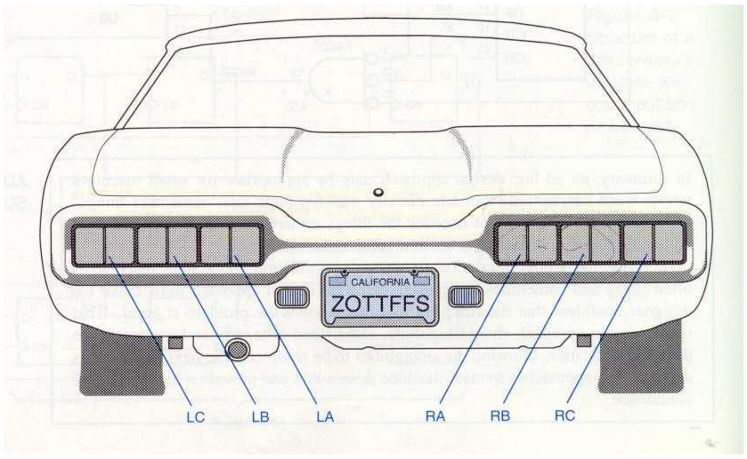
Tail Light Controller

Objective

The objective of this project is to implement a sequential taillight controller analogous to what can be found on 1966 Ford Thunderbirds or some Ford Mustangs.

Specifications

Consider the taillights of a 1966 Ford Thunderbird, shown below. There are three lights on each side, and for turns and hazard mode, they operate in the sequences shown. The goal of this project is to build a state machine that controls the operation of these taillights.



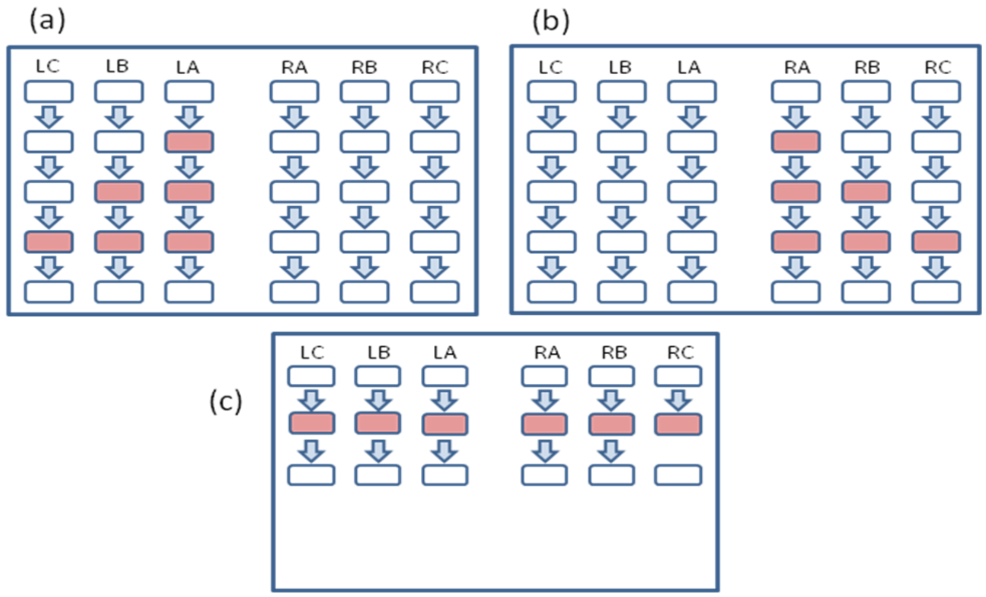


Figure 1: (a) Taillight sequence for left turn indication. (b) Taillight sequence for right turn indication (c) Taillight sequence for hazard.

The state machine has two *synchronous* input signals (LEFT and RIGHT) and six *synchronous* output signals (LC, LB, LA, RA, RB, RC) that control the taillights. The state machine to be designed must realize the following operational characteristics:

- Starting from the IDLE state (initial state), if none of the inputs are asserted, then all taillights should be OFF (outputs are not asserted).
- The assertion of the LEFT input signal alone indicates the driver's request for a left turn. If the other input is not asserted, the output signals should produce the appropriate sequence indicating a left turn. For left turns, the machine should cycle through four states, in which the right-hand lights are OFF and 0, 1, 2 or 3 of the left-hand lights are ON.
- Likewise, the assertion of the RIGHT input signal alone indicates the driver's request for a right turn. If the other input is not asserted, the output signals should produce the appropriate sequence indicating a right turn. For right turns, the machine should cycle through four states, in which the left-hand lights are OFF and 0, 1, 2 or 3 of the right-hand lights are ON.
- The assertion of both the LEFT and RIGHT input signals simultaneously, indicates the driver's request for hazard mode. In hazard mode, the machine should cycle through two states, in which all six lights are controlled as in Fig. 1(c) (All lights OFF then all lights ON then back to all lights OFF).
- We assume the existence of a free-running clock signal whose frequency equals the desired flashing rate for the lights. (You will use your board’s clock divided down with MUX for this, as usual.)

Simplifying Assumptions: Once a turn sequence begins (e.g. left turn, right turn, hazard) it completes without interruption.

NOTE: The design is to be implemented as a **synchronous state machine**.

DEMO:
Done in simulation with waveforms and on the FPGA board.