Name: Rami Omer
Student ID: 215855620

Lab 3: T-Bird Tail Light Design

In this document, I will explain the FSM design that I have constructed to achieve the objectives and deliverables described in EECS3216_Lab3 document. I will also briefly and concisely explain the structure and content of my Verilog code that implements the T-Bird Tail light patterns.

Please use this link to find the demonstration video for this lab: https://www.youtube.com/watch?v=9oOhSFElNvc

The FSM Design:

In this design, I defined 9 states. Each state responds to a 6-bit input command. A sketch of the design is shown in **figure 1**.

The 9 states shown in the graph are defined as follows:

- S0 or the state best known as the "All LEDs are OFF" state.
- **Six** states defined for all 6 LEDs with the output of each state being the LED status "ON".
- A state was also assigned to the case where all LEDs are ON in the pattern.
- Finally, a state was assigned to the last case of the design in which all LEDs are off and an error message "E" is displayed on the screen.

The 6-bit input command that allows the circuit to transition between any two states in the design is compromised of two components. As shown in **Table 1**, the first component (the two MSBs) is allocated to switch 1 and switch 0. The remaining 4 bits houses the values stored in a 4-bit register that was used to store the values of a counter in the clock divider circuit.

Bit#	5	4	3	2	1	0
Input	Switch 1 (SW1)	Switch 0 (SW0)	Counter[3]	Counter[2]	Counter[1]	Counter[1]

Table 1. shows the content of the 6-bit input commands that is used to transition between states in the FSM design of the T-Bird Tail Light patterns.

Name: Rami Omer
Student ID: 215855620

As shown in **figure 1**, when SW1 and SW0 are each in the "0" position, the circuit is in the idle state- the state in which all LEDs are off. It is also worth noting that no matter in which state the circuit is currently residing, the moment SW0 and SW1 are turned to the "0" position, the design immediately transitions to the "All LEDs are OFF" state. When SW1 is in the "0" position and SW0 is in the "1" position, the circuit moves in the following direction:

Counter = 0 → RA is ON
Counter = 1 → RA is ON and RB is ON
Counter = 2 → RA, RB and RC are all ON
Counter = 3 → All three LEDs are OFF

This patter repeats itself 3 times until the register counter store the value 12 in it. When the value stored in counter is 12, the design transitions to the "All LEDs are ON" state and stays in that state as long as the SW1 and SW0 remain in the "0" and "1" positions respectively.

The remaining 3 LED states (LA, LB and LC) follow the same logic discussed above except that SW1 needs to be in the "1" position and SW0 needs to be in the 0 position. It is also important to note that the instant SW1 moves to "0" and SW0 moves to "1", the design transitions to the "RA ON" state and starts the right LEDs sequence from start to the finish-counter = 0 to counter = 12. Same design applies to the left LEDs

Finally, when the two switches move the "1" position, the circuit transitions to the "All LEDs are OFF and Display "E" on the screen state". In this state, all 6 LEDs are off and an error message "E" is displayed on one of the board's screens.

Name: Rami Omer
Student ID: 215855620

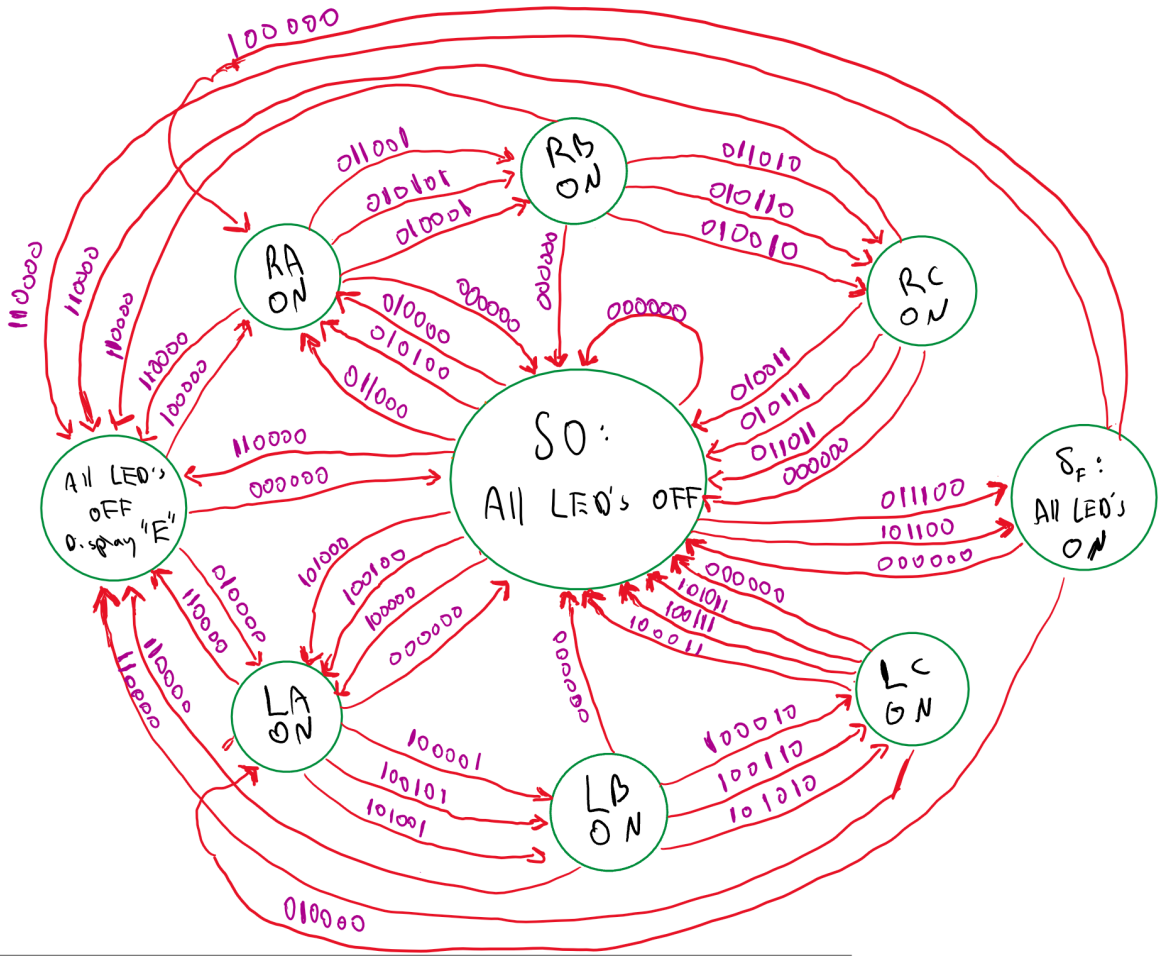

Figure 1. shows the FSM design for the T-bird Tail light pattern.

Verilog Code:

My Verilog code consists of 3 modules. The first module is a clock divider and a counter. The second module is a design of a combinatorial circuit that ties the LEDs and the screen to the output counter of the first module. The third module is the top-level module in which I connect the wires between the first two modules.

Name: Rami Omer
Student ID: 215855620

The first module:

In the counter module, I used the following inputs and outputs:

- Inputs:
  DE-10 Clock, Switch 0 and Switch 1.
- Output:
  A 4-bit register that stores the value of the counter.

I used an always block sensitive to the positive edge of the clock cycle to do the following:

- When switch1 and switch0 are both in the “0” position, sore at all time the value 0 in the output register.
- When either one of the switches, but not both, is in the “1” position, increment the value stored in the counter register by 1 every 25E6 cycles i.e., every 0.5 seconds. The incrementing starts from 0 all the way to 12. When the value in the counter register is 12, the incrementing process is stopped and the value stored in the register stays at 12.
- Finally, when both switches are in the “1” position, the counter stores the value 0 in it at all times.

The second module:

In the second module I used the combinatorial logic to tie the results stored in the counter of the first module to the output LEDs and the screen that will be used to display the error message.

In this module, I used the following inputs and output:

- Inputs:
  Switch 1 and Switch 0 and a 4-bit wire that will later be connected to the output register of the first module in the top-level entity.

- Outputs:
  A 6-bit wire for the 6 LEDs and a 7-bit wire for the seven segments screen.

I used truth tables and k-maps to construct combinatorial circuits comprised of AND and OR gates to connect the counter to the outputs of thee second module.

Name: Rami Omer
Student ID: 215855620

The third module:

In this module, I used the following inputs and output:

- Inputs:
  DE10 Clock, Switch 0 and Switch 1.

- Outputs:
  The seven-segment screen and the 6 LEDs.

I used a 4-bit wire to connect the counter register, which is an output of the first module to the second module, in which the register is now and input of that circuit.