LAB 3: The T-Bird Taillight Control Using a CPLD

ELEC2607 - LOE, Mon 2:35pm – 5:25 pm

Saturday, March 9th, 2020

Youssef Ibrahim
Abdalrahman Shaheen

1.0 INTRODUCTION

The purpose of this lab is to create a circuit for the taillight of a T-Bird. The output of the circuit is controlled by the position of switches labeled L (Left), R (Right), B (Brake), and a clock. The clock will be further manipulated to output different patterns. The circuit is designed to simulate actions such as indicating a right or left turn, braking, turning while braking, and indicating an emergency. The circuit was built and simulated using a software called Xilinx ISE and ModelSim; These programs allowed us to download the circuit on a programable logic board (CPLD) and simulate it by lighting different bulbs. The CPLD contains three switches (L, B, R) which control the behavior of the output, and an output which is demonstrated by six bulbs. The following report shows the Specifications in part 2.0, Design in part 3.0, Implementation and Testing in part 4.0, and the Conclusion in part 5.0 of the telephone switch circuit.

2.0 SPECIFICATIONS

The T-bird tail-light circuit was designed using Xilinx software. ModelSim was used to generate waveforms which simulated the output of the gates which was then downloaded into a Programmable logic board (CPLD) to simulate the behavior of the circuit. The T-bird light is made of 2 sets of lights (right and left), each set contains 3 bulbs. The T-bird circuit consisted of three smaller circuits which is: A counter which emits four signals that turn on and off at certain times to control the bulbs, also a left and right control boxes that direct the counter signals to the correct bulbs. The circuit has a total of 8 different input combinations that will result in different output signals. The circuit has 3 inputs for the user to choose from which are the switches labeled as L, B, R. Pressing only on the switch L or R will indicate a left or right turn by lighting the 3 left or right bulbs in sequence starting from the first bulb and then repeat. Pressing only on switch B will indicate braking by lighting up all the six bulbs. Pressing on switches L and R

together will indicate emergency by flashing all bulbs with a half second delay between each flash. Pressing on switches L or R along with B will cause the lights at the left or right to flash as for a turn, while the lights on the opposite side is being steadily on to indicate braking. Pressing all switches L, R, and B together will indicate braking by making the braking light override the emergency flashers. There were not any limitations in designing this circuit.

3.0 DESIGN

The full design of the T-bird tail-light circuit is shown below in Figure 3.0.

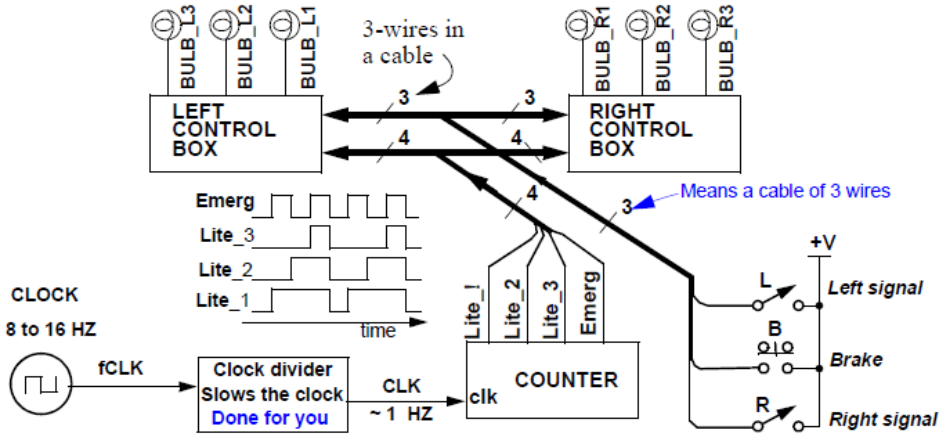


Figure 3.0: Block diagram of the T-Bird tail-lights [Lab 3 - p2, Modified]

3.1 Counter Circuit

The counter circuit consists of two sub-circuits which are: 1. A State-Generating circuit which generates 4 states, this State Generator contains a Divide-by-Two circuit. 2. An output circuit which generates Lite-1, Lite-2, Lite-3, and Emerg from the state and switch inputs.

3.1.1 State Generating Circuit

The State Generating Circuit generates a sequence of outputs that counts, the circuit uses a flip-flop to divide the input into two to provide a new signal which has half the clock frequency (Q_0), which allows inputs such as 00, 01, 10, and 11. The first bit of the sequence is called CLK, and

the second bit of the sequence is called Q_0. Figure 3.1 below shows waveforms of CLK and Q_0 for the sequence CLK,Q_0 = 00 -> 01 -> 10 -> 11 -> 00.

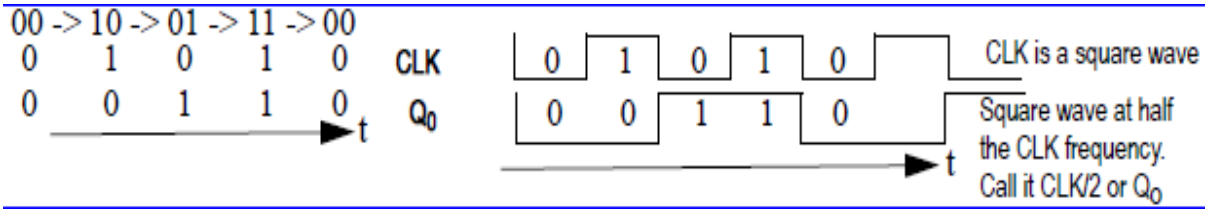


Figure 3.2: Waveforms obtained from the Divide-by-Two Circuit [Lab 3 – p3, Modified]

The states shown above will be used to turn on the tail-lights in sequence to give the pattern 000 -> 100 -> 110 -> 111 -> 000.

A D flip-flop is used as a memory element, which has inputs CLK (clock input) and D which remembers the input signal, and output Q. Figure 3.5 below shows how a D flip-flop works.

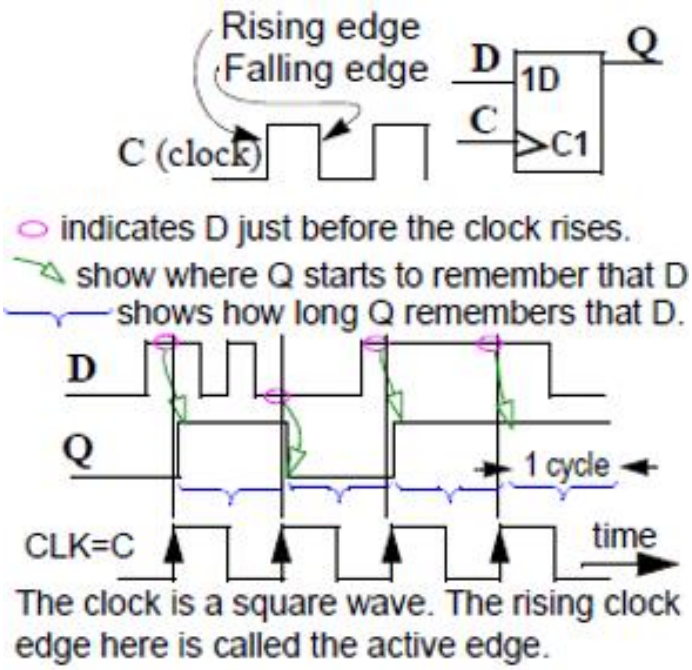


Figure 3.5: D flip-flop [Lab 3 – p3, Modified]

Figure 3.5 above shows that Q takes the value of D just before the rising clock edge, and Q will remain constant till the next rising edge.

Q_0 (CLK/2) changes only when D changes. To output Q_0, an inverter is added between input D and output as shown below in Figure 3.6.

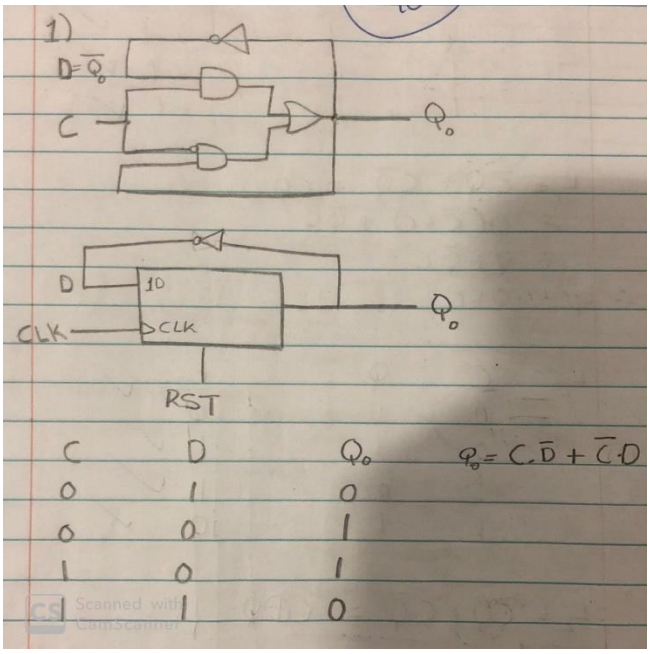


Figure 3.6: Divide-by-two Circuit [Prelab – Youssef Ibrahim]

Figure 3.6 above shows a Divide-by-two circuit. Practical flip-flops such as the one used here have an extra input called Reset (RST), when Reset = 1 this will cause the output Q to go to 0. The waveform for Q_0, and D relative to CLK is shown below in Figure 3.7.

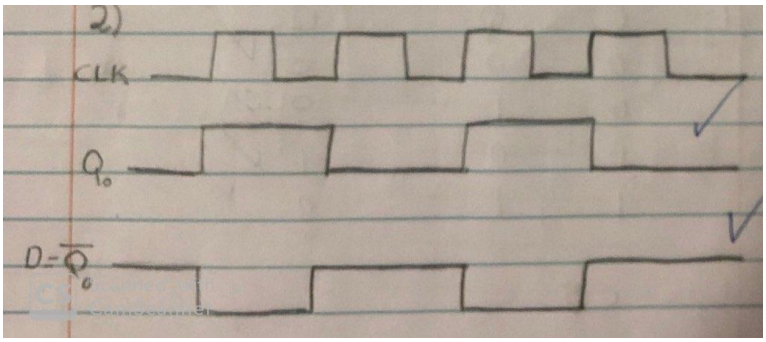


Figure 3.7: Q_0 and D waveforms relative to CLK [Prelab – Youssef Ibrahim]

3.1.2 Output Circuit

The output circuit has four outputs called Lite-1, Lite-2, Lite-3, and Emerg. Figure 3.8 below shows the waveforms of Emerg, Lite-3, Lite-1, Lite-2, Q_0, and D relative to CLK.

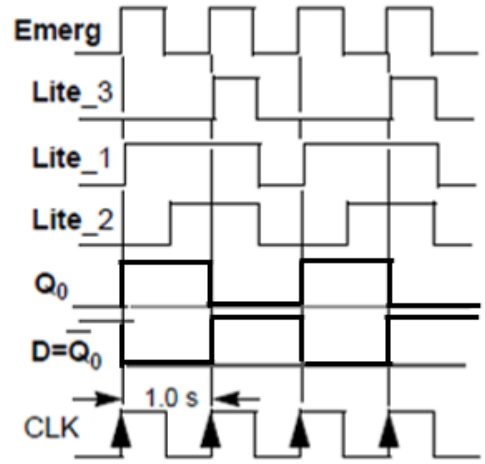
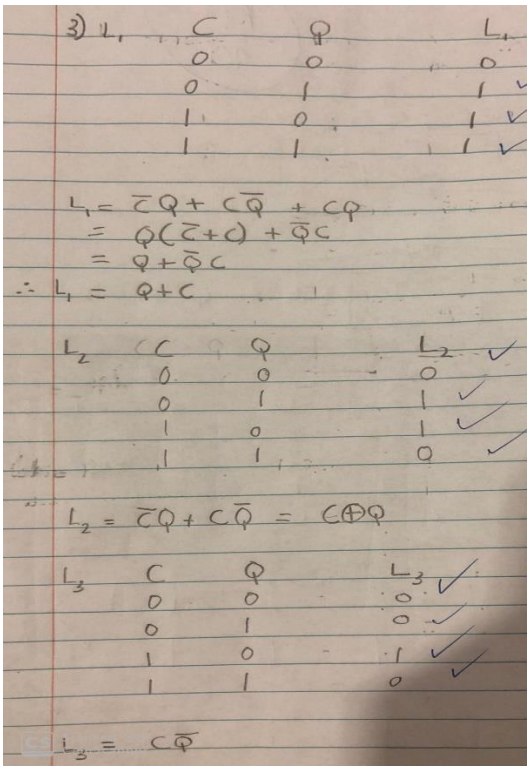


Figure 3.8: Waveforms of Emerg, Lite-3, Lite-1, Lite-2, Q_0, and D relative to CLK [Prelab – Youssef Ibrahim]

Using the waveforms shown in Figure 3.8 above, the logic of Lite and Emerge signals was obtained. The logic of Lite-1, Lite-2, Lite-3, and Emerg is shown below in Figure 3.9.



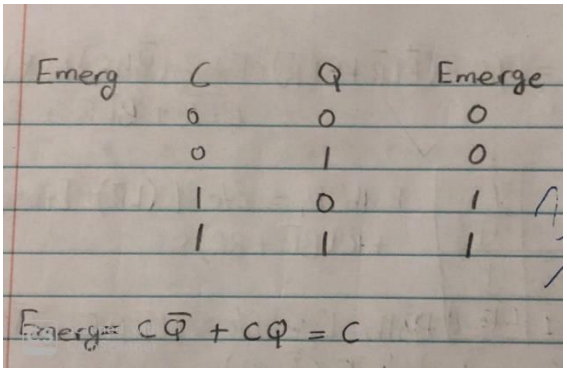


Figure 3.9: Logic of Lite-1, Lite-2, Lite-3, and Emerg [Prelab – Youssef Ibrahim]

Using the information shown above in Figure 3.9, the output circuit was designed as shown below in Figure 3.10.

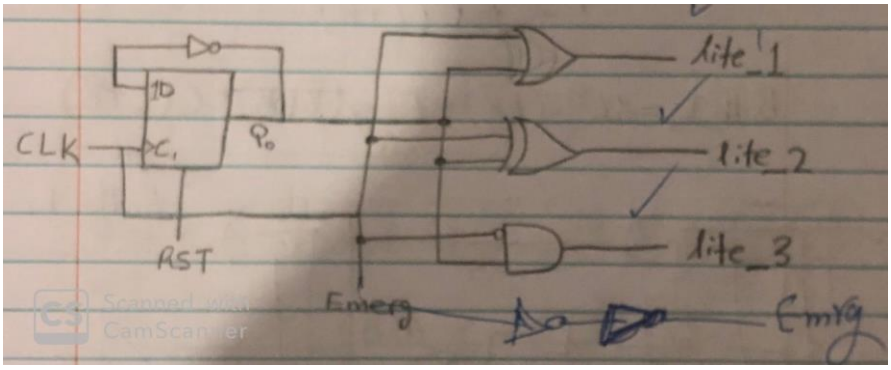


Figure 3.10: Output Circuit [Prelab – Youssef Ibrahim]

The truth table with values of Q_0CLK in time sequence is shown below in Figure 3.11.

State	Outputs			
Q_0CLK	Lite_1	Lite_2	Lite_3	Emerg
0 0	0	0	0	0
1 1	1	0	0	1
1 0	1	1	0	0
0 1	1	1	1	1

time ↓

Figure 3.11: Truth table with values of Q_0CLK shown in time sequence [Lab 3 – p5, Modified]

Using information from Figure 3.11 above, equations for the outputs is derived as shown below in Figure 3.12.

$$Emerg = CLK$$

$$Lite\_1 = Q_0 + CLK$$

$$Lite\_2 = Q_0 \oplus CLK$$

$$Lite\_3 = \overline{Q_0} \bullet CLK$$

Figure 3.12: Equations for Emerg, Lite_1, Lite_2, and Lite_3 [Lab 3 – p5, Modified]

Using all the previous information, the counter circuit was designed as shown below in Figure 3.13.

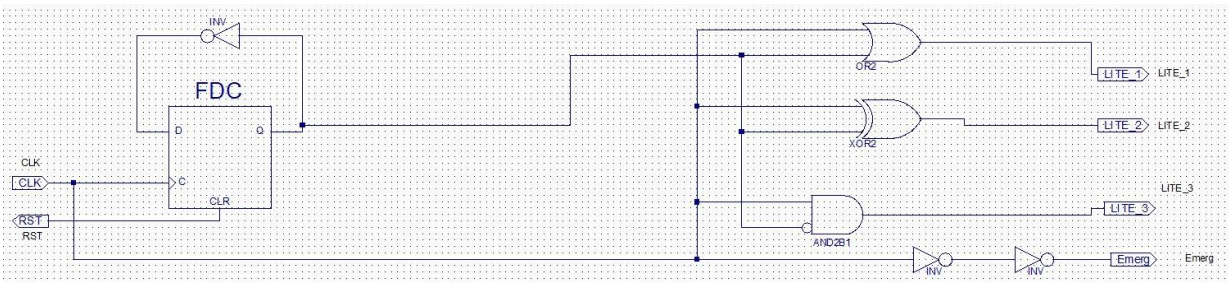


Figure 3.13: Counter circuit schematic

Since the software used doesn't allow direct connection between an input and output ports, two inverters are connected to the CLK input as shown above in Figure 3.13.

3.2 The Left Control Box

The Left Control Box inputs are the outputs of the counter circuit along with the switches L (left turn signal), R (right turn signal), and B (brake). Equations relating Bulb_L1, Bulb_L2, and Bulb_L3 to B, L, and R switches, Lite_1, Lite_2, Lite_3, and Emerg were derived from the table shown in Figure 3.14 below.

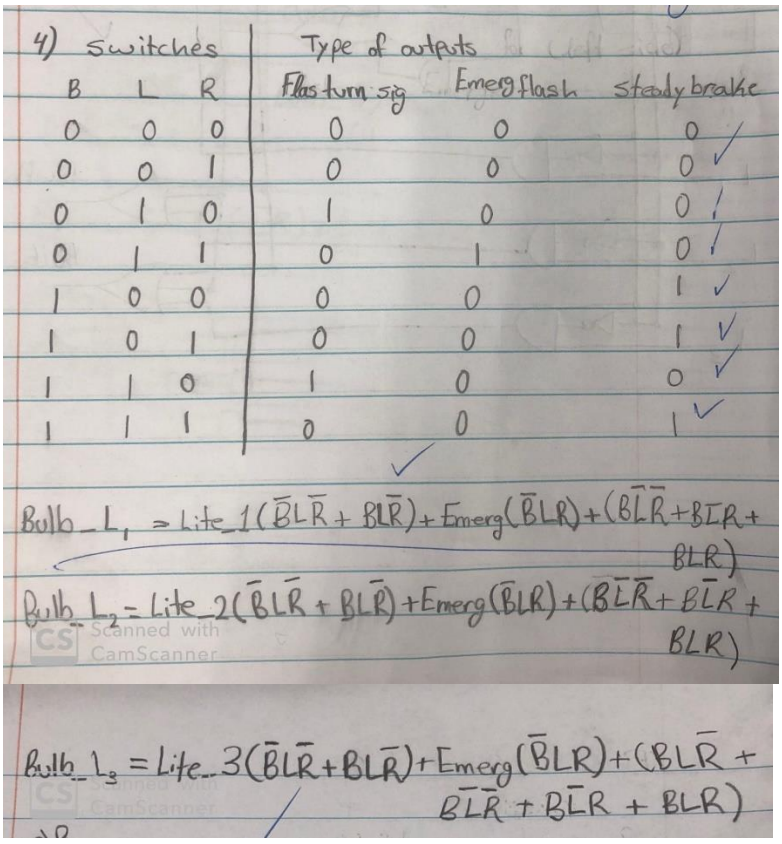


Figure 3.14: Table and Equations for the Left Control Box relating Bulb_L1, Bulb_L2, and Bulb_L3 to B, L, and R switches, Lite_1, Lite_2, Lite_3, and Emerg [Prelab – Youssef Ibrahim]

Equations shown above in Figure 3.14 were simplified using a K-map as shown below in Figure 3.15.

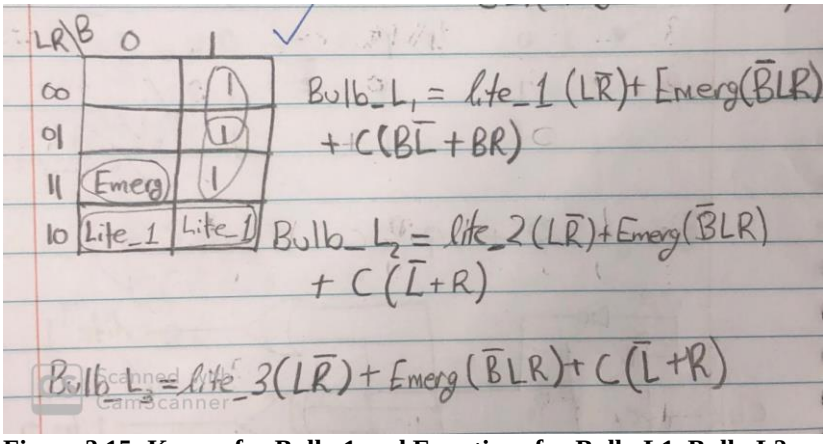


Figure 3.15: K-map for Bulb_1 and Equations for Bulb_L1, Bulb_L2, and Bulb_L3 [Prelab – Youssef Ibrahim]

Using the equations shown above in Figure 3.15, the Left Control Box was designed as shown below in Figure 3.16.

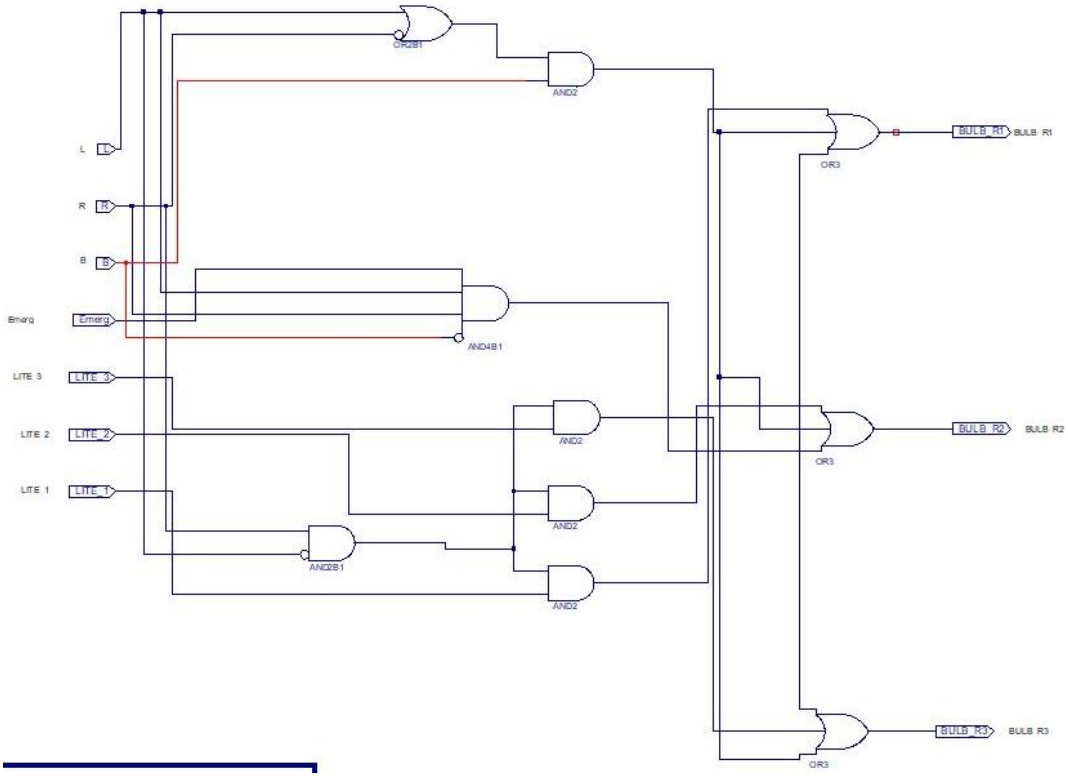


Figure 3.16: Left Control Box Schematic

3.3 The Right Control Box

The Right Control Box inputs are the outputs of the counter circuit along with the switches L (left turn signal), R (right turn signal), and B (brake). Figure 3.17 below shows a table containing the different types of outputs for The Right Control Box.

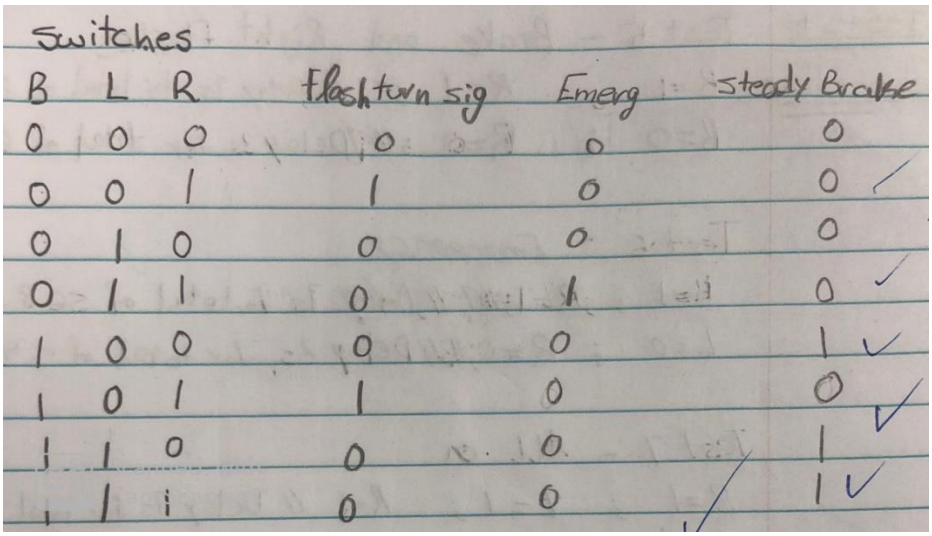


Figure 3.17: Outputs for Right Control Box for different switch inputs [Prelab – Youssef Ibrahim]

Using information from Figure 3.17 above, a K-map, and the equations for Bulb_R1, Bulb_R2, and Bulb_R3 were derived as shown below in Figure 3.18.

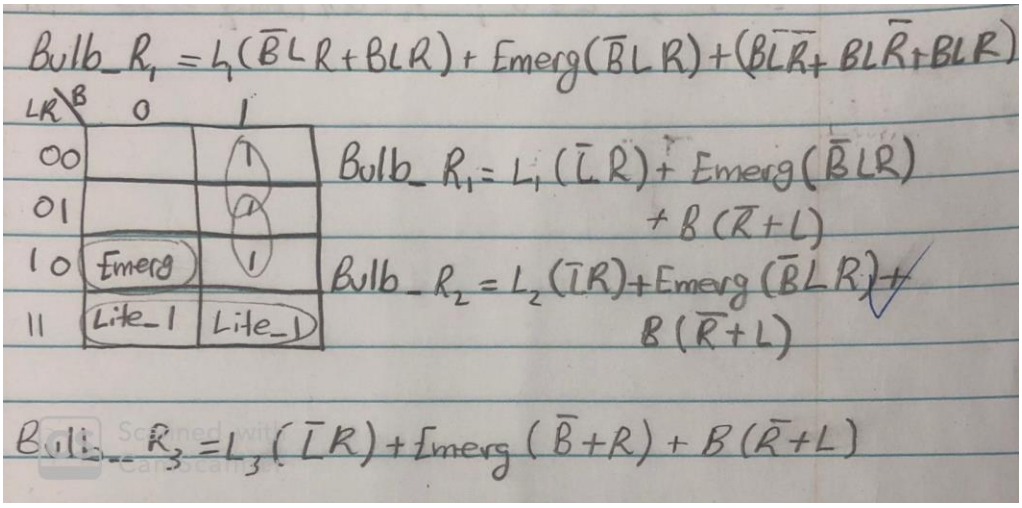


Figure 3.18: K-map and Equations for Bulb_R1, Bulb_R2, and Bulb_R3 [Prelab – Youssef Ibrahim]

Using the information above, The Right Control Box circuit was constructed as shown below in Figure 3.19.

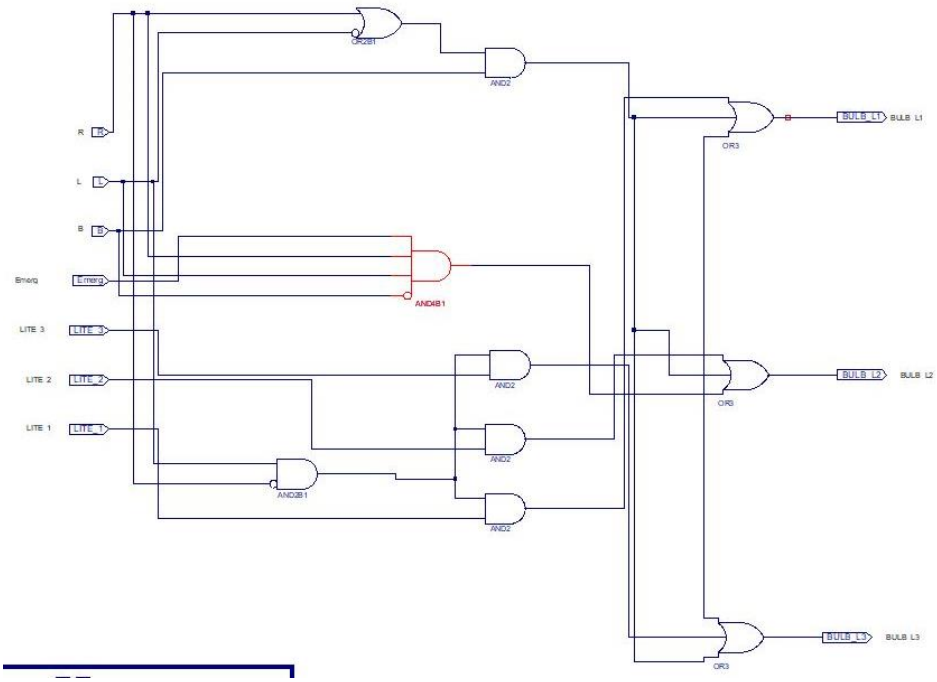


Figure 3.19: The Right Control Box

4.0 TESTING AND IMPLEMENTATION

The circuit was built using the designed circuits in the prelab and then implemented

In Xilinx according that. To run a simulation test on the circuit, we used ModelSim to provide the corresponding waveforms. Then, debugging was done to make sure there is no error messages. If the waveforms observed are matching the required test, then the lab was done successfully. CPLD was also used for further testing by recognizing the outputs of light bulb. Once more, if the light bulbs output was correct, then the lab is complete.

Working on this lab we faced many debugging issues while testing the circuit we constructed. Once we find which part of the circuit needs adjusting, we had to identify where exactly is the error and fix it. It can be an issue with the settings of the program in the beginning of the lab or after running the simulation. We return to the program and adjust the new update on the circuit. Mostly, wiring connections were not connected in the first part of the circuit the way they should. We had to do the design in prelab 3 times because we made the wires short and we

connect them to each other. Technically it works, but this way we made it harder to make sure the wires are touching ends. We worked on the circuit for the last time and we made the connections for the terminals to the gates immediately and it worked just fine. The main issue was how easy our prelab design compared to the Xilinx design we constructed. The final testing done on the circuit was the light bulbs. The output was random and incorrect at the first run. We noticed an inverter in the wrong place. Then, on the second attempt to make the light bulbs work, the lab period finished, and we couldn't make the required changes to our ModelSim and debug the issue.

We consider the lab to be complete if the circuit is working and all waveforms outputted matches the required forms with the TAs and the outputs of the bulbs in the CPLD operated in the correct right pattern.

The waveforms found are shown below:

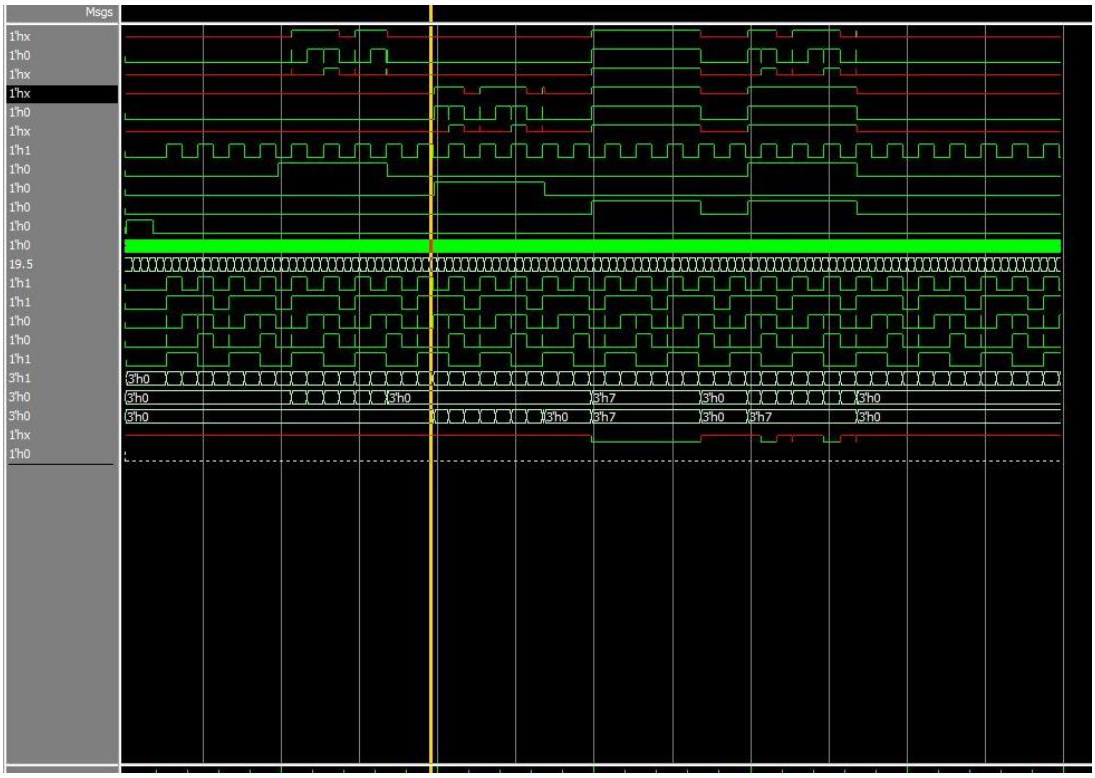


Figure 3.20: T-Bird Tail Light Xilinx waveforms

5.0 CONCLUSION

The circuit built in this lab did not actually meet the specifications required as the taillights of the T-bird were incorrectly simulated. Furthermore, the output waveforms were not all correct and thus the simulation of the taillights using the CPLD were incorrect as well. To do this lab the circuit was broken up into 3 sub circuits as seen in the design, otherwise this may have taken more time than only one circuit. The design of three circuit was not a big of a challenge however many issues were implemented in the program to be solved easily. These errors however were solved by debugging the circuits designed on Xilinx where the circuits are built in smaller sub-sections and to achieve the correct outputs for the circuit. A lot of new features and menu settings were learned in this lab and it is such a great practical application of Xilinx as well as CPLD was used. This lab also proved to be a good introduction to CPLD as this was not used prior to the lab and it also helped in further increasing knowledge on how to use the Xilinx program with more complex circuits we build in future.

References:

[1] Department of Electronics, " Lab #3: The T-Bird Tail-Light Control Using a CPLD ".Carleton
University, Ottawa Ontario, Feb 2011, Accessed: February 27, 2019