4-Way Traffic Light Controller

List of Authors (Matthew Monroe, Aaditya Dhiman, Matthew Fortman, Aadeesh Dhiman)
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: matthewmonroe@oakland.edu, aadityadhiman@oakland.edu, mfortman@oakland.edu, adhiman@oakland.edu

***Abstract*—We developed a 4-way traffic light as our final project for this course. The project followed and implemented a similar way as a commonly used traffic light works in the real world without lights for turns.**

I. Introduction

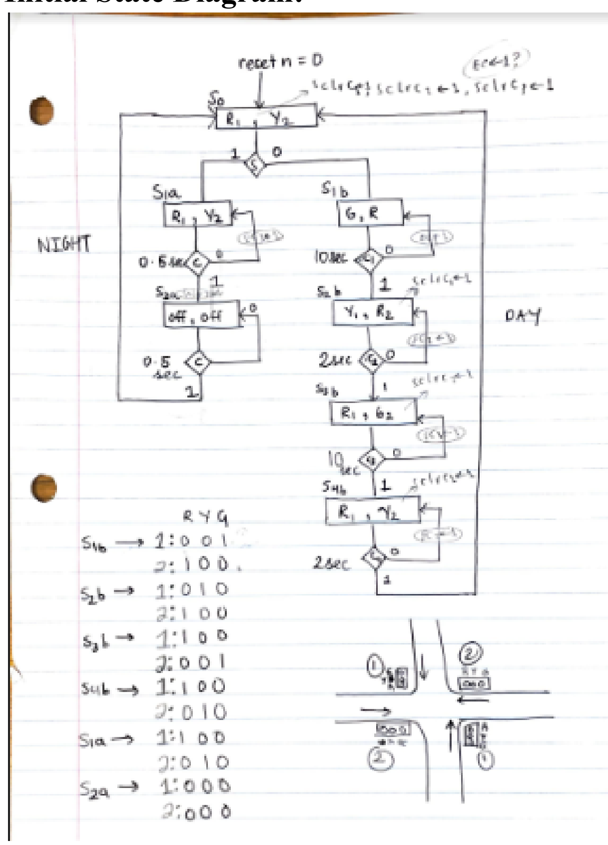
The 4-way traffic light controller is a device that will allow us to control the traffic. Since there are a lot of vehicles on the streets every day and with this number increasing day by day, a more managerial infrastructure is needed to ensure safety and smooth flow of the traffic. This device would address this issue and help by controlling traffic in 4 directions at a major intersection. The traffic light that we designed took into consideration between day and night where in the night it operated differently as it did in daytime.

II. Methodology

A. First Section

To begin this project, we first considered what was required. We needed to create a traffic light to control the flow of traffic in four different directions at an intersection, meaning we needed four sets of lights. In addition to this, we needed to consider operation for both daytime and nighttime, operating in two different modes. Finally, we also needed to consider the timing of each light. After these parameters were determined, we began by creating drafts of our finite state machine, which would be the main operational component of our circuit. When designing, we used our parameters, along with critical thinking in order to implement an efficient yet effective design. This included the use of counters and LED states, which were critical in determining which state would be displayed once the design was operating on a breadboard. Below shows an image of our initial finite state machine state diagram:

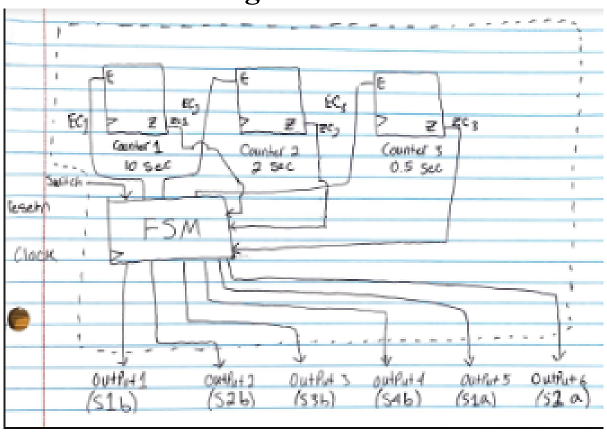
Initial State Diagram:



B. Second Section

After creating a rough draft of how our finite state machine worked, we next developed our full circuit diagram, which linked the counters with the finite state machine, along with creating a visual representation of our input and output values. The rough draft of this circuit diagram can be seen below:

Initial Circuit Diagram:



3. Third Section

After creating our initial finite state machine state diagram and circuit diagram, it was critical to our success that we met with Professor Llamocca in order to fix anything that did not seem correct. After meeting with Llamocca, we felt more confident with our design and were able to move forward with testing our design through the behavioral simulation. After confirming our values were correct, we implemented our design onto the Nexys A7 - 100T FPGA board, along with wiring our circuit onto a breadboard. These action steps taken throughout the development process allowed us to successfully complete this project.

III. Experimental Setup

IV. *First Section*

We made a circuit diagram which used different digital logic components including three counters and a finite state machine. A switch was also used to switch between day and night since traffic lights work differently. The daytime lights would flash just as a regular traffic light. It would show the Red to indicate a stop, yellow to prepare for a change in light, and green to indicate a go signal. Whereas in the nighttime, we had yellow lights flashing in two directions and red in the other directions. This was done to let the driver know to proceed safely in the nighttime.

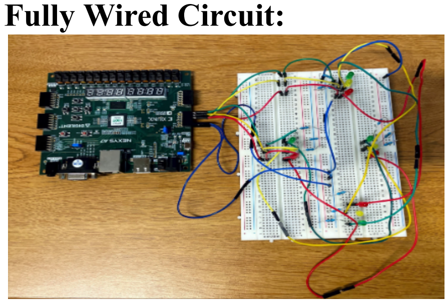
B. Second Section

With the finite state machine, we were able to design a process for our traffic lights to follow. When we had one set of lights turned green, the other way turned red. This was done for 10 seconds after which the red light stayed red, and the set of green lights changed to yellow. This process was completed within 2 seconds. After this, the other sets of lights turned green and the other turned red. This during the night time changed to one set of lights flashing red while the other set flashed yellow in a synchronized way. The time interval between each of these flashing was set to 0.5 seconds.

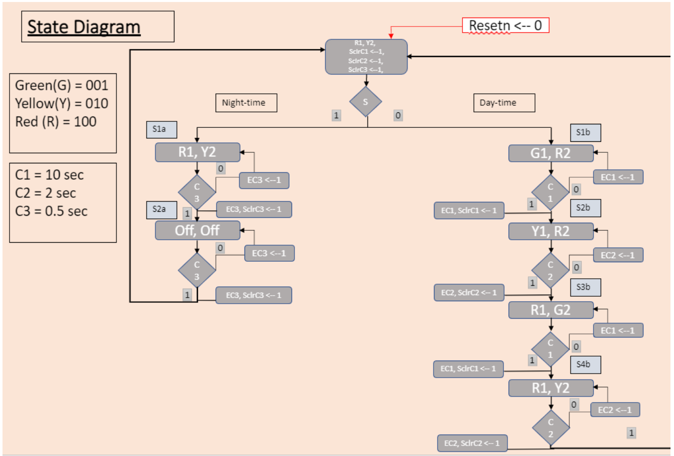
3. Third Section

This flashing was made possible using three counters which were set to different times. One of the counters was set to 10 seconds while the other one we had was for two seconds. These two counters were used during daytime. The switching between day and night was done using a switch on the Nexys A7 - 100T board. During nighttime, we only needed one set of lights to flash red while the other set to flash yellow. For this, we made use of a counter with a time of 0.5 seconds. All of these counters were connected to our finite state machine which was connected to the six-pin output on our board using wires. These wires were connected to the LED ports on our breadboard. Along with the LEDs, which were red, yellow and green, we also used 330 ohms resistors with each of these. Refer to the images below to see our FSM diagram, our circuit diagram and fully wired circuit which will put all these explanations in place.

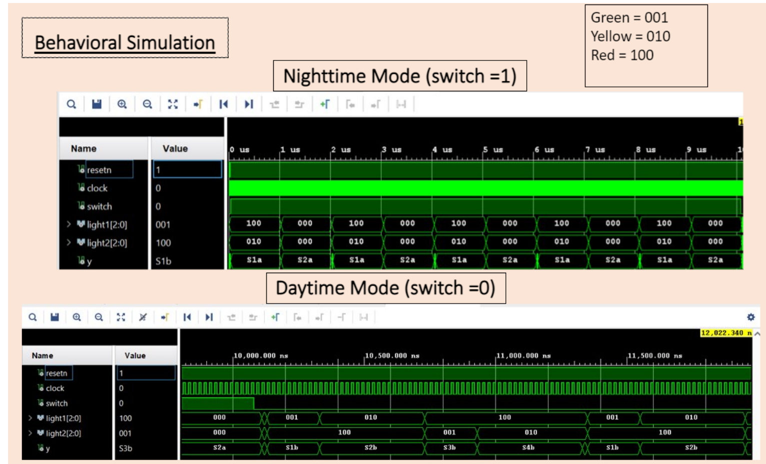
Fully Wired Circuit:



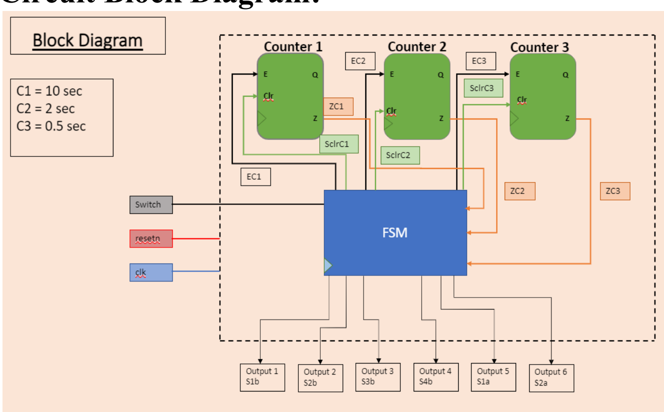
Finite State Machine:



Behavioral Simulation Results:

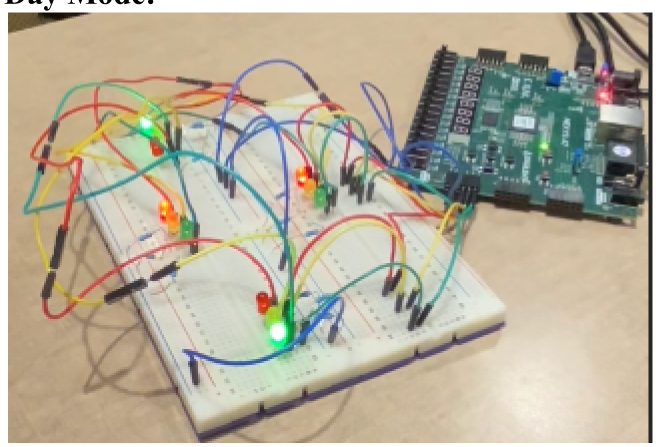


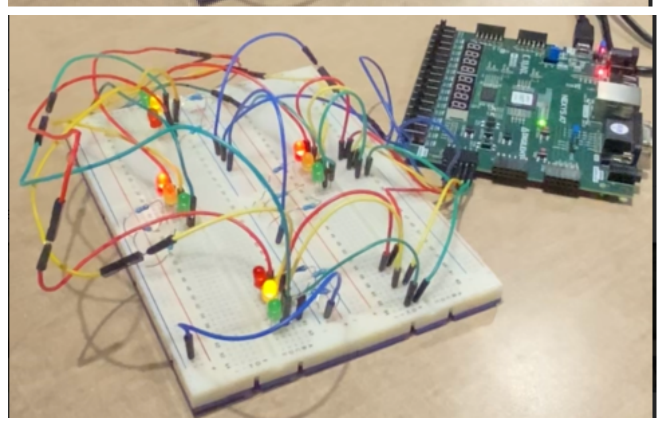
Circuit Block Diagram:



Full Implementation on FPGA Board Results:

Day Mode:





V. Results

The results of this experiment helped us get an understanding of how we can make use of an FPGA board. Using simulation, we were able to confirm that the values we predicted were right since our expected outputs were correctly shown at each stage in our behavioral simulation. Please find our results below. In addition, once the design was fully implemented onto the FPGA board and wired to the breadboard, we were able to conclude that our project was complete.

Night Mode: