Mini Project 2 Final Report

Ivy Mahncke

System Architecture

The **src/** folder in this directory contains the following files:

- **top_tb.sv**
- **top.sv**
- **color_cycle.sv**
- **fade.sv**
- **pwm.sv**

Each file possesses its own module, with **top** at the highest level and **fade** and **pwm** at the lowest level. **top_tb** is used to record and view the RGB output of the system.

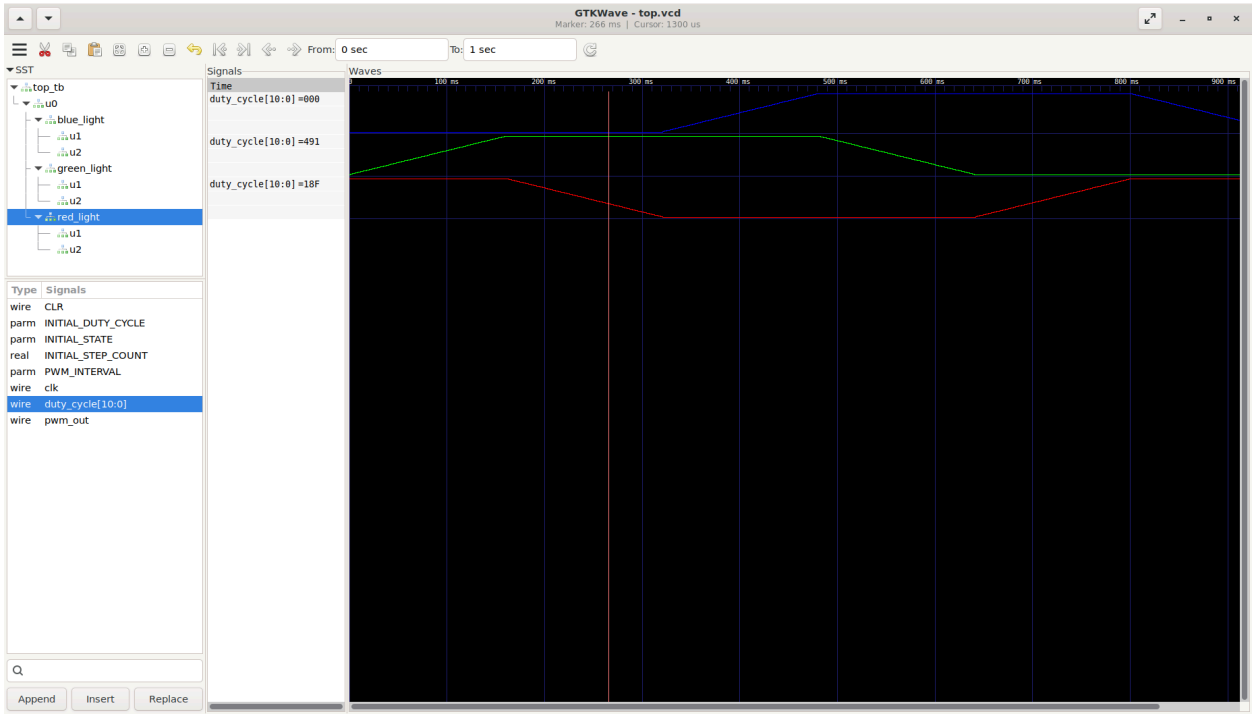
top contains three instances of the **color_cycle** module, representing red, green, and blue respectively. **top** passes each color a standard PWM interval and a clock; additionally, **top** sets the initial state and cycle position of each color. Therefore, **top** could be modified if one wanted the colors to rise and fall at different intervals, such as all together or in reverse. **top** also outputs RGB values from each color module.

color_cycle simply possesses an instance of the **fade** module and the **pwm** module. Meant to be used as a connective structure representing any of the three RGB values, **color_cycle** passes timing and state parameters to its **fade** module, and pipes the output value of fade to an input parameter for **pwm**. Finally, **color_cycle** returns the output signal from **pwm** as its own output.

fade is the most complex module in the system. **fade** maintains a finite state machine for a color signal to travel through, with the following states: increasing, holding high, decreasing, and holding low. **fade** calculates the proper durations for the step states (increasing, decreasing) and steady states (holding high and low), with the steady states twice as long as the steps. **fade** maintains a time counter to ensure that the state machine shifts when the proper duration is achieved. On each clock cycle, **fade** outputs a **cycle_value** variable that is ultimately translated into PWM. During steady states, this value is constant; during step states, the value is either increasing or decreasing at a constant rate.

pwm is the simplest module. Given a **cycle_value** parameter from an associated **fade** instance, **pwm** transforms this value into a PWM output signal to pipe to the actual board.

GTKWave Demonstration



This is a screenshot of the GTKWave-generated representation of the RGB signal produced by my system in a single second. Present on the graph is the **duty_cycle** variable from each **color_cycle** instance, which is actually the output of the **fade** module. Each of the three color signals rise, hold, and fall at equivalent intervals; however, they are offset equally from each other to output a continuous color wheel on the RGB light.

Demo Video

https://drive.google.com/file/d/1sRCiiqMccPAP1sCuxfxUzFoTtjXb1gbl/view?usp=sharing