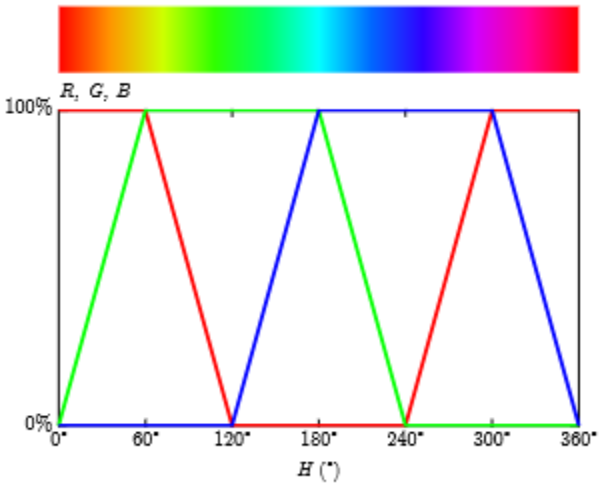
Computer Architecture Miniproject 2



RGB components versus hue angle, *H*.

Figure 1. The basic design of the digital circuit, and how it should function.

Understanding how to implement the digital circuit requires fundamental understanding of how this design works. We know that this should happen over 1 second, and each segment happens in a ⅙th of a second. Assuming this, that means that flat HIGHs and flat LOWs should happen in ⅓rd of a second and increasing and decreasing state should happen in that ⅙th of a second time. We also understand that each color has its own initial state, meaning that you have to precode them to initialize at being that state.

Starting with the initial state, I made a ternary operator that takes a color parameter that you pass into the module, and changes the initial LED state based on what color the function is taking in. From there, you can easily determine the second state, by reading the graph. In all cases, a flat line happens in 2 cycles, so flat lines can be split into 2 segments, and increasing and decreasing always lead to LOW or HIGH, respectively. Knowing these, we can determine initial states and next states, and on flat states, keep the PWM value at a full HIGH or full LOW value and on increase or decrease states, increment or decrement the PWM value up or down to HIGH or LOW. For transitioning, based on your given time slot, you can determine from parameters how many steps you need to take and how many change steps need to happen before you switch states, which leads to a very clean change, as seen in the figure below.

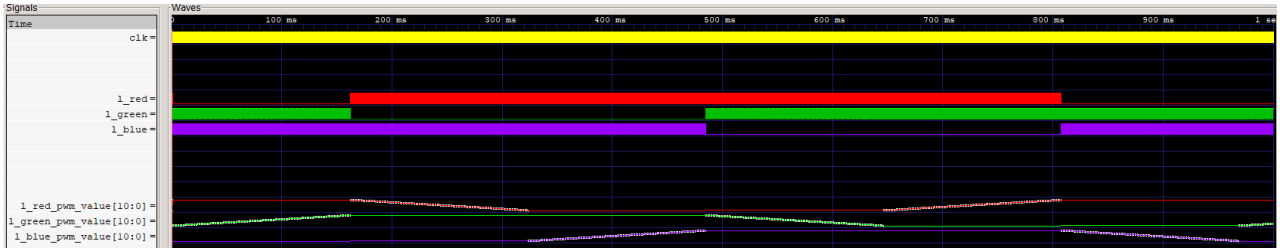


Figure 2. My implementation. The LEDs are one when it is either LOW or in an increasing or decreasing state, and off when it is at a HIGH state. Disclaimer that this testbench technically runs at 12.5MHz to save on space.

Github Link: https://github.com/foxwithahdie/miniproject_2