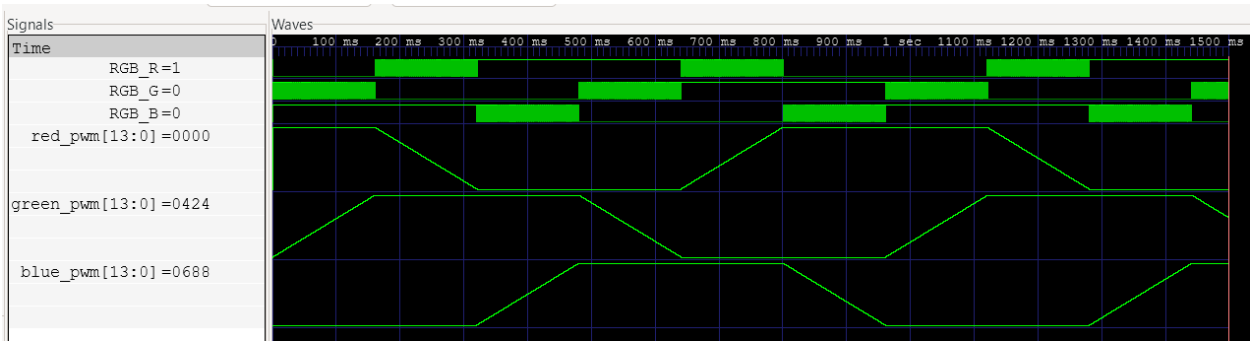
Computer Architecture Mini-Project II: RGB LED

Charlie Sands

10/06/2025

I used a combination of seven finite state machines to generate the flowing rainbow LED pattern. The first finite state machine counts global time (time with respect to power on) and sets a start status flag for each LED when each should begin to fade on. The next three finite state machines count local time for each of the LEDs, starting from when they begin to fade on, and use it to advance the state flag through each step of the face cycle (increasing, solid, decreasing and off). The final three finite state machines determine the duty cycle of each LEDs PWM signal by considering the state flag and local time of each LED. The LEDs are toggled on and off for a fraction of the PWM period of 1670 clock cycles, which is ~0.13ms long, to generate the fading pattern. This time period was chosen for mathematical convenience.

Below is a screengrab from GTKWave during a 1.5 second simulation of my code. The first shows the overall structure of the PWM pulses delivered to the LEDs (RGB_R, RGB_G, RGB_B), and the analog representation of the value the PWM pulse width is derived from (red_pwm, green_pwm and blue_pwm).



GitHub Repository for code here.

Link to demonstration video here.