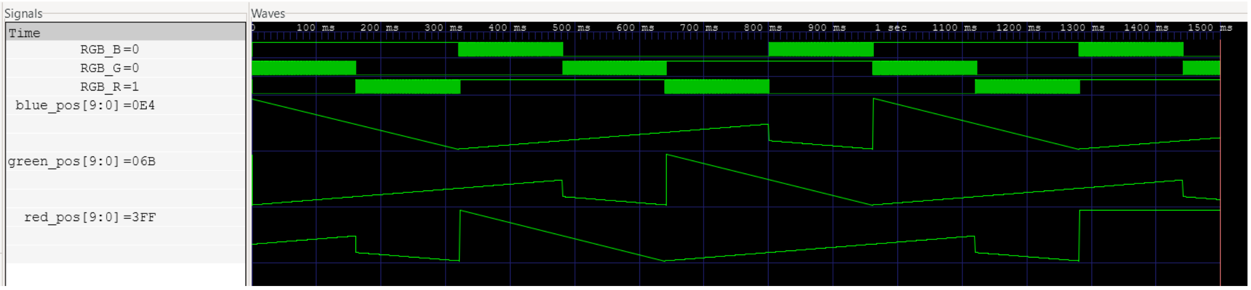
Computer Architecture Mini-Project II: RGB LED

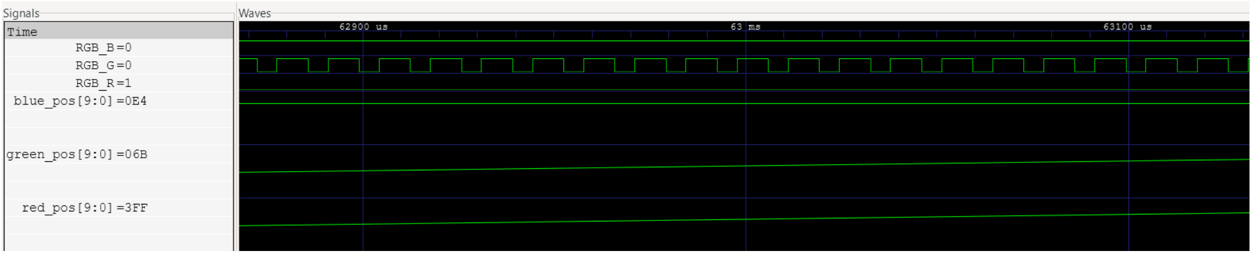
Charlie Sands

10/06/2025

I used a combination of seven finite state machines, four of which count time and three of which act as “controllers,” toggling on and off each LED. The first state machine counts “global” time and sets the monitors the start conditions which causes each color to begin fading. The next three state machines (one for each color) are activated by the global state machine and count the “local” time for each LED, setting flags for whether the LED should be increasing in brightness, decreasing in brightness, staying fully on or staying fully off. The final three state machines resolve the status of each LED by considering the current LED state (increasing brightness, decreasing brightness, on or off, as mentioned above) and the “local time” of the LED to determine how bright or dim it should be at a given time. The PWM signal is directly derived from the LED’s “local” time counter, kept track of by the second, third and fourth state machine.

Below is a set of screengrabs from GTKWave. The first shows the overall structure of the PWM pulses delivered to the LEDs, and the analog representation of the value the PWM pulse width is derived from. The second shows a zoomed in view of some of the PWM pulses delivered to the green LED.





GitHub Repository for code here.

Link to demonstration video here.