Owen Himsworth

ENGR3410

Brad Minch

3/3/2025

Miniproject 2: RGB LED Color Cycle on iceBlinkPico

1. Introduction

This project implements a digital circuit for the iceBlinkPico board (iCE40UP5K-SG48) that smoothly cycles an onboard RGB LED through the colors of the HSV (Hue, Saturation, Value) color wheel. The design leverages a 12 MHz clock to drive a counter that generates a hue angle ranging from 0 to 359 degrees over one second. The hue value is converted into red, green, and blue intensities via a simplified piecewise linear HSV-to-RGB conversion. An 8-bit PWM (Pulse-Width Modulation) generator then uses these intensity values to drive the LED outputs, which are active low.

2. Design Overview

2.1. Hue Angle Generator

Clock Division:
A counter (`angle_count`) counts clock cycles until it reaches a predefined maximum (approximately 33,333 cycles at 12 MHz). Once the maximum is reached, the hue angle (`hue_angle`) is incremented by one.

Hue Range:
The hue cycles from 0 to 359, generating 360 discrete steps, which result in a complete color cycle over one second.

2.2. HSV-to-RGB Conversion

Piecewise Linear Mapping:
The hue angle is divided into six segments (each 60° wide). For each segment, one color channel remains at its maximum value (255), one channel stays at minimum (0), and the third channel transitions linearly (increasing or decreasing).

Output:
The conversion outputs three 8-bit values (`pwm_red`, `pwm_green`, `pwm_blue`) corresponding to the intensity for each RGB channel.

2.3. PWM Generation

PWM Counter:
An 8-bit counter (`pwm_counter`) runs continuously at the 12 MHz clock rate.

Output Comparison:
Each RGB channel is driven by comparing `pwm_counter` with the corresponding color value. Because the LED outputs are active low, the signal is driven low (0) when `pwm_counter` is less than the computed intensity.

3. Operation

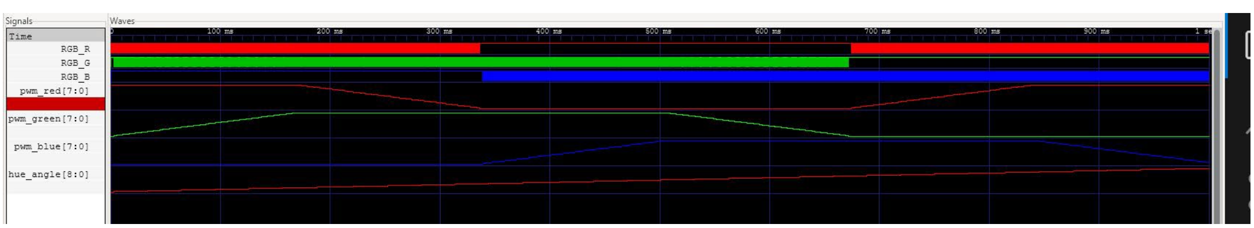
Color Cycling:
As the clock runs, the `angle_count` increments until it resets at ~33,333. Each reset increments the `hue_angle`. The hue value is then converted into red, green, and blue intensities by the HSV-to-RGB block.

PWM Signal Generation:
The PWM counter continuously runs, and the RGB outputs are generated by comparing the counter against the respective intensity values. The result is a PWM signal for each color that, when averaged over time, produces the desired brightness.

Active-Low Outputs:
The LED channels are active low. A low PWM output corresponds to an “on” state for that LED channel.

4. Simulation and Results

The design was simulated using Icarus Verilog, and the generated waveform was viewed in GTKWave. The following plot shows the RGB signals over a 1-second cycle, demonstrating the color cycle operation:



Note: The plot shows that the internal hue value increments approximately every 2.78 ms, and the PWM outputs reflect the calculated intensities. Due to the high frequency of the PWM counter, individual PWM pulses are not visible when zoomed out, but the overall duty cycle of each channel corresponds to the computed intensities.

5. Conclusion

This design successfully implements a smooth color cycling function for the onboard RGB LED of the iceBlinkPico board. By converting a slowly changing hue value into RGB intensities and using PWM to drive the LED channels, the circuit achieves the desired visual effect over a one-second cycle. The simulation results confirm that the design functions as intended.