Application Brief

Controlling an RGB LED Using TPLD with a Single Input

Ian Graham

What is RGB?

When LEDs of different colors are close together, the light from the LEDs can blend together to seem to create a color array when viewed by a human. Designers using this property can create the rainbow of colors using three LEDs, each of a different color – red, green, and blue. One method to convert the three separate colors into one merged color is the use of the HSV (hue, saturation, and value) color model. This model describes the results of combinations of different relative power values of red, green, and blue light sources.

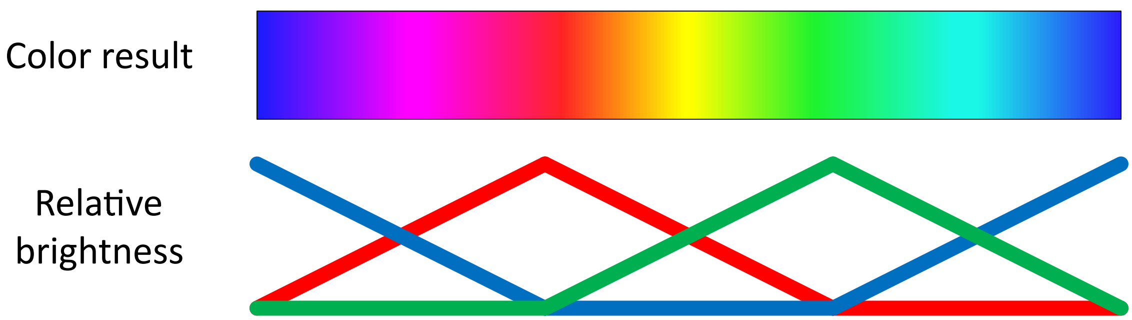

Figure 1. Power-Conscious HSV Model

Typically, to control and RGB light source, a designer can use three inputs to the LED: one to control each color. This can be an issue in systems with small MCUs or limited available control pins, or systems where the control signals need to be driven a long distance from the controller. It also complicates LEDs meant to be controlled directly by a user, as the user needs to effectively send three signals to control a single output.

Using TI's Programmable Logic Devices (TPLD), designers can convert a single control signal to an RGB value. By leveraging TPLDs' configurability, a user can implement an RGB control method that fits into their unique system. A few simple examples of possible control methods are shown below.

Continuous Sweep Control Method

This control method sweeps the RGB LED through a continuous power-conscious HSV pattern, as shown in Figure 2. Note that this control method follows the power-conscious HSV model shown in Figure 1.

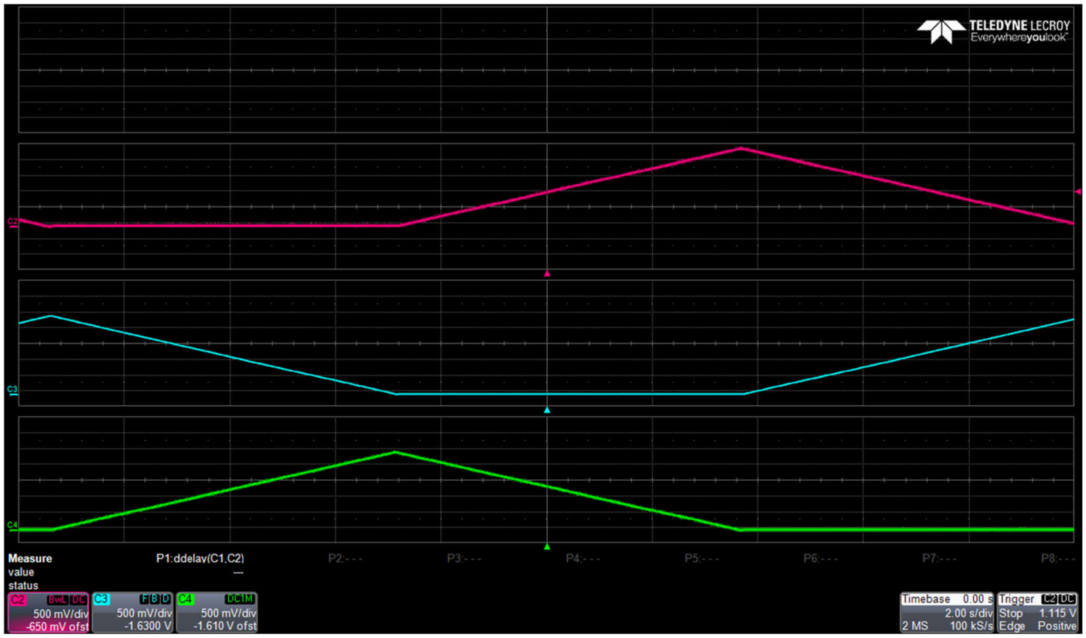

Figure 2. Power-Conscious HSV Model on TPLD

When the TPLD input receives a pulse, this input stops the sweep and the LED stays at whatever color it held when the input signal was detected. It holds this color until it receives a new pulse, as shown in Figure 3.

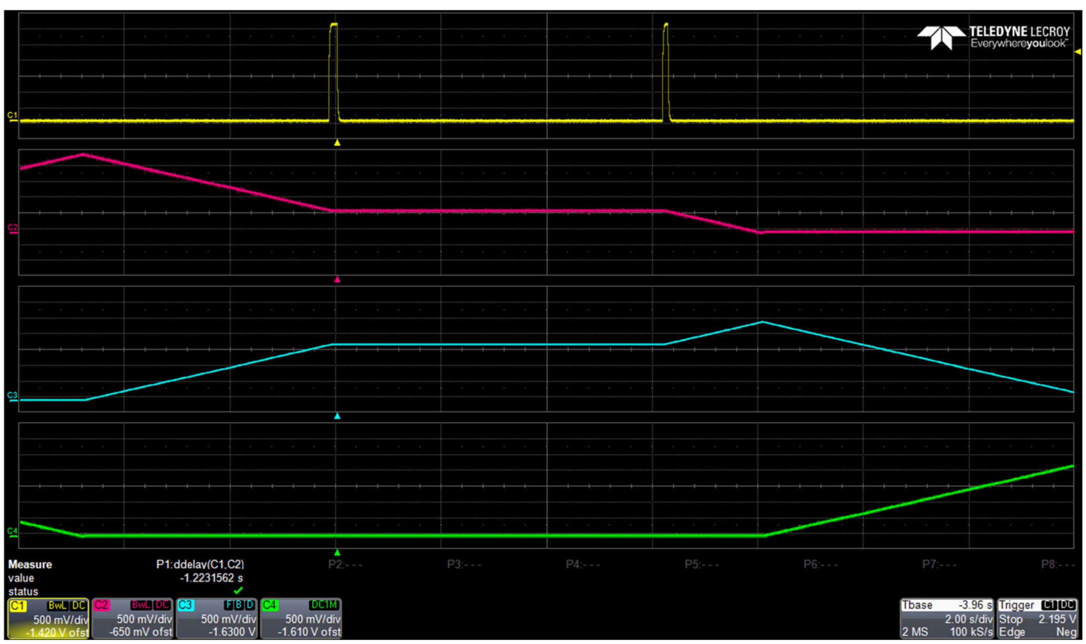

Figure 3. Continuous Sweep RBG Control Method Output

This method works well for a human controlled system, as it is simple for the user to select from a range of colors with no other user-interface. The InterConnect Studio (ICS) design of the continuous sweep control method is shown below. Groups 0 and 1 are equivalent to the circuit shown to create the blue output.

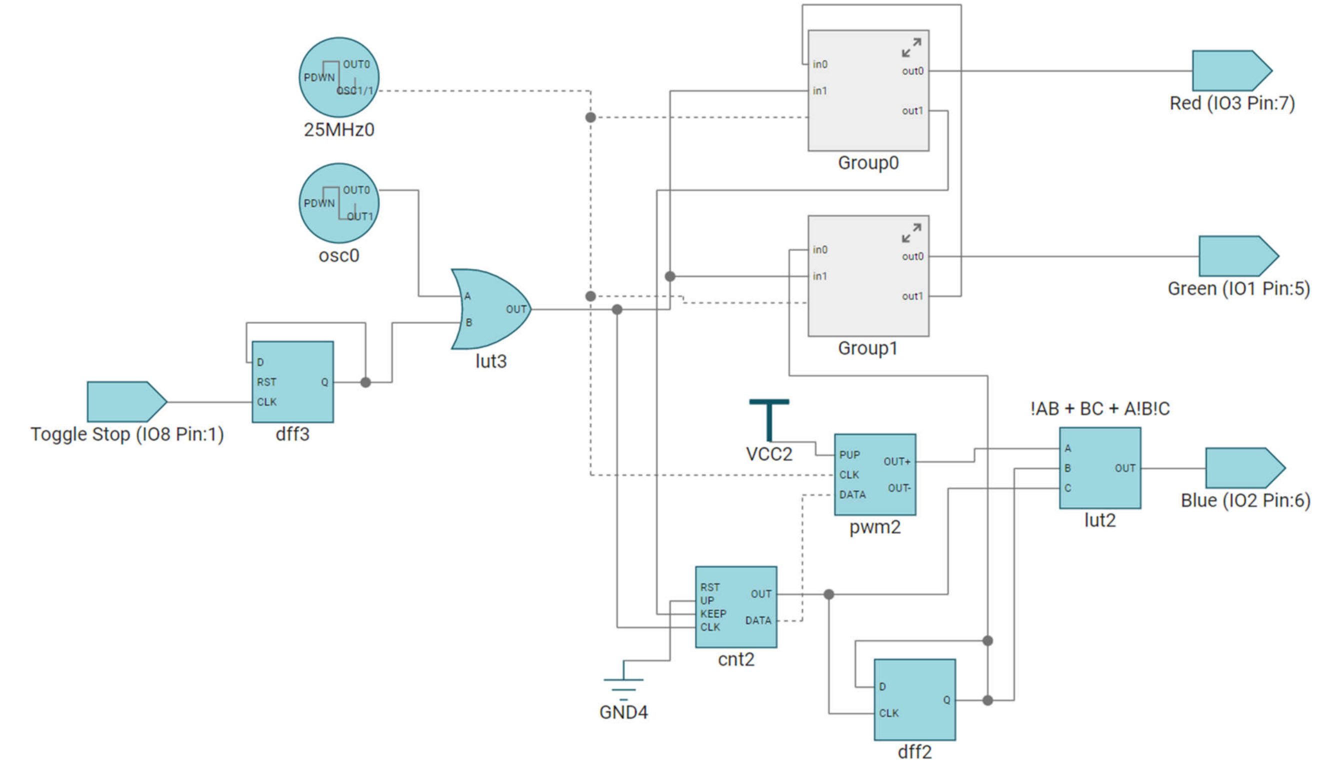

Figure 4. Continuous Sweep RBG Control Method in ICS

Rotating LED Control Method

This control method allows the designer to control each color one at a time. On a high input pulse, one of the red, green, or blue LEDs increases in relative power, looping back to 0 once the max potential power is reached. When the next input pulse is sent, this pulse controls the next color in the RGB sequence, looping back to red after blue. Figure 5 illustrates channel 1 showing the input, and the other channels showing the RGB outputs.

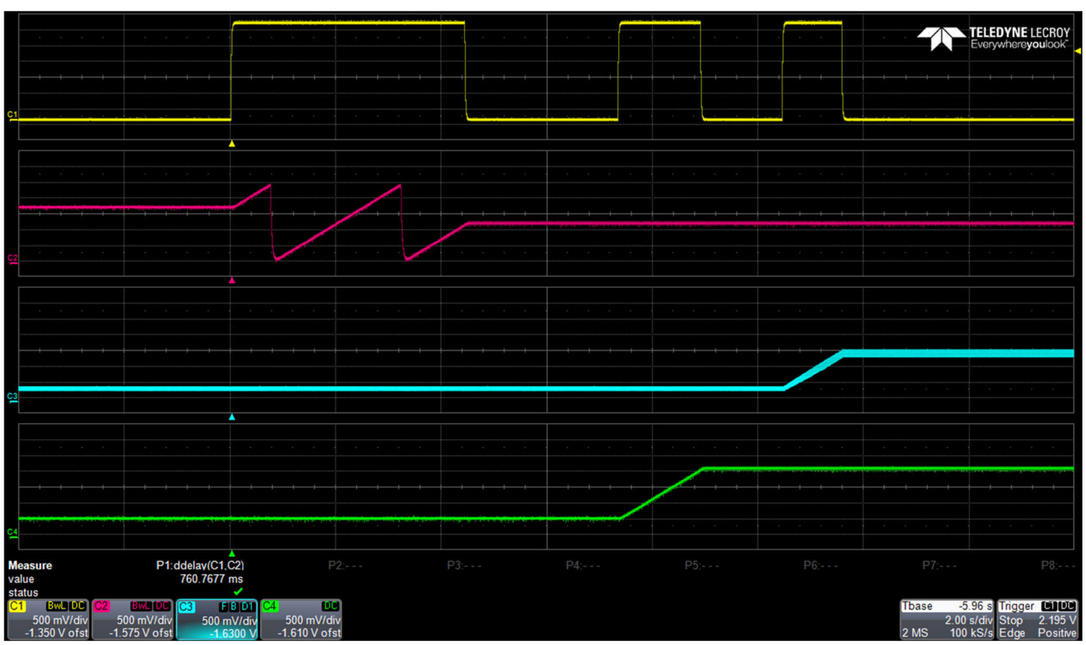

Figure 5. Rotating LED Control Output

This method works well for an MCU controlled system, which can easily keep track of the current LED color and can calculate the durations required of the pulses to change to the desired color. While this method can cycle through an array of colors to reach the goal color, with an MCU controlling the input pulse, this method can be easy to cycle through colors faster than a human can perceive to arrive at the desired color. Unlike the continuous sweep example, which outputs at a consistent brightness, this method also allows the designer to control the brightness of each LED independently. The InterConnect Studio (ICS) design of the rotating LED control method is shown in Figure 6.

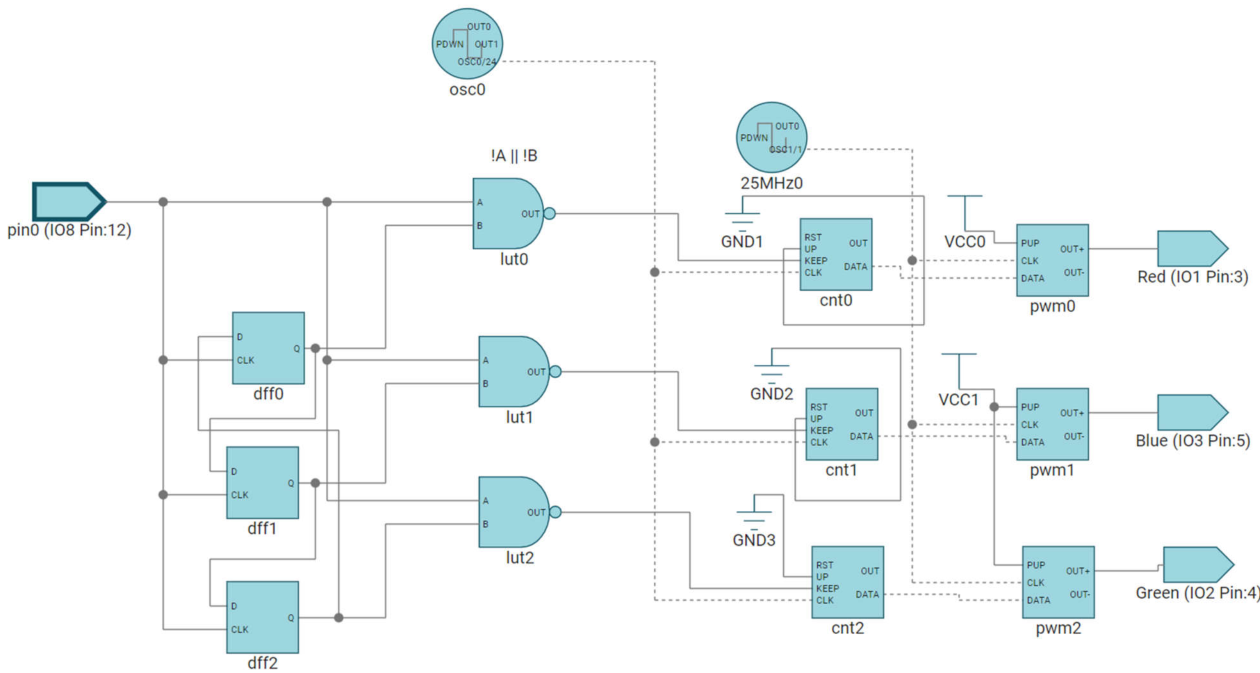

Figure 6. Rotating LED Control Method in ICS

Designing for a System

The two examples shown above are meant to be baselines to be built off of, if desired. There are many improvements and features that can be added to a controller to make this baseline more compatible with the designer's system. For example, a smart button can be added to the inputs of the system so that, if the input is held high for long enough, the system can reset into a known state.

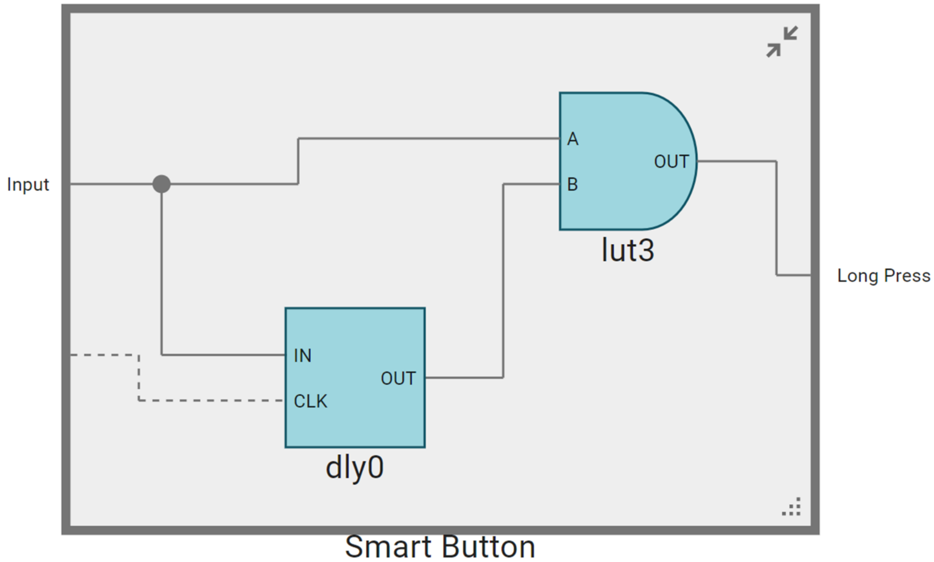

Figure 7. Smart Button in ICS

Implemented into the Rotating LED Control schematic, the smart button output can reset the flip-flops and PWM generators to the system's initial condition. The ICS design of the rotating LED control method with added reset functionality. In this design, the smart button outputs high once the input has been high for slightly longer than a complete LED cycle.

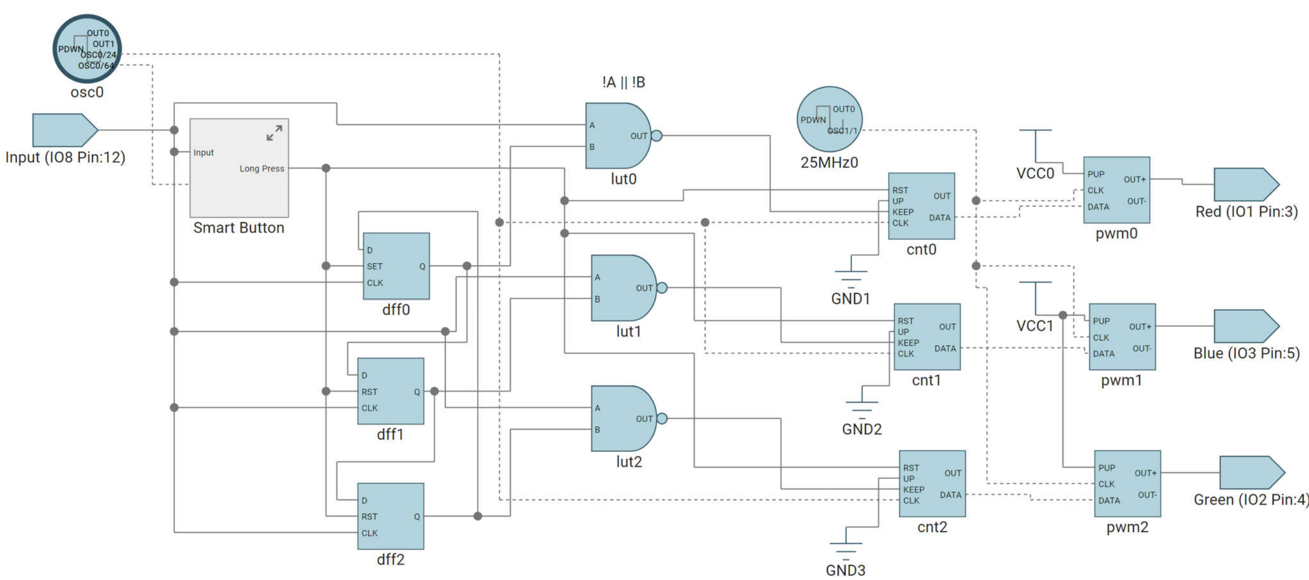

Figure 8. Rotating LED Control Method With Reset in ICS

Adding this to the control scheme allows an MCU to regain control over the LED if necessary, making the LED easier to control electronically. If the MCU needs to reset, the LEDs can be forced back to the start-up conditions. A scope shot showing power-averaged outputs of the rotating LED control design with added reset functionality is shown in Figure 9. Note that the RGB outputs reset to startup conditions after a long input pulse.

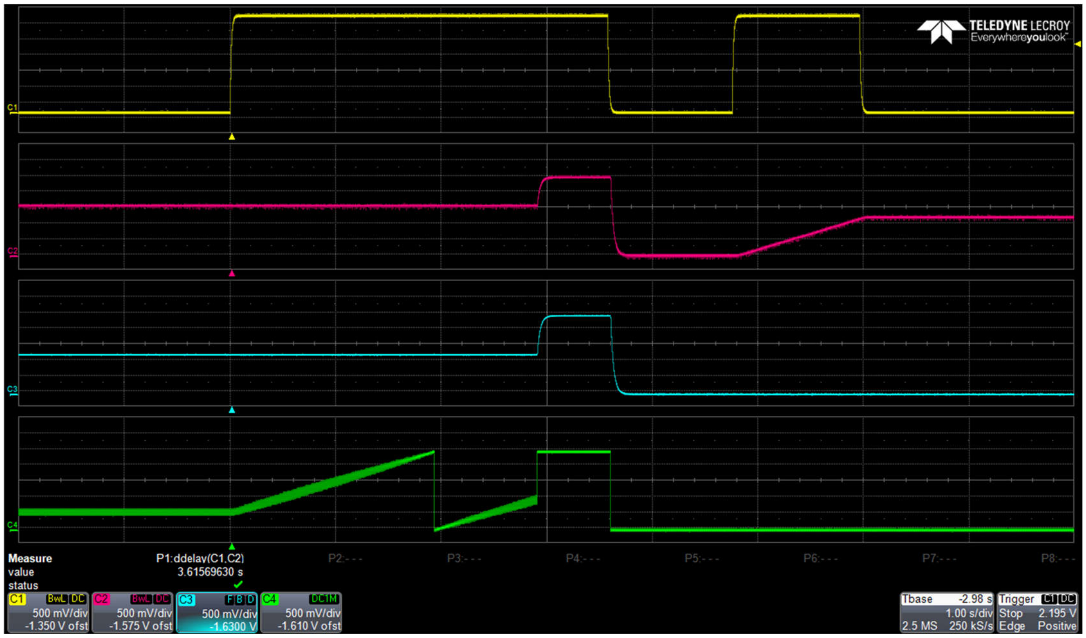

Figure 9. Rotating LED Control Method With Reset Output

Another use of a smart button on the input of the rotating LED control design is to allow for the change in which color is being controlled using a short pulse. The (ICS) design of the rotating LED control method with added channel select functionality is shown in Figure 10. In this design, the smart button outputs high once the input has been high for longer than about one fourth of a complete LED cycle.

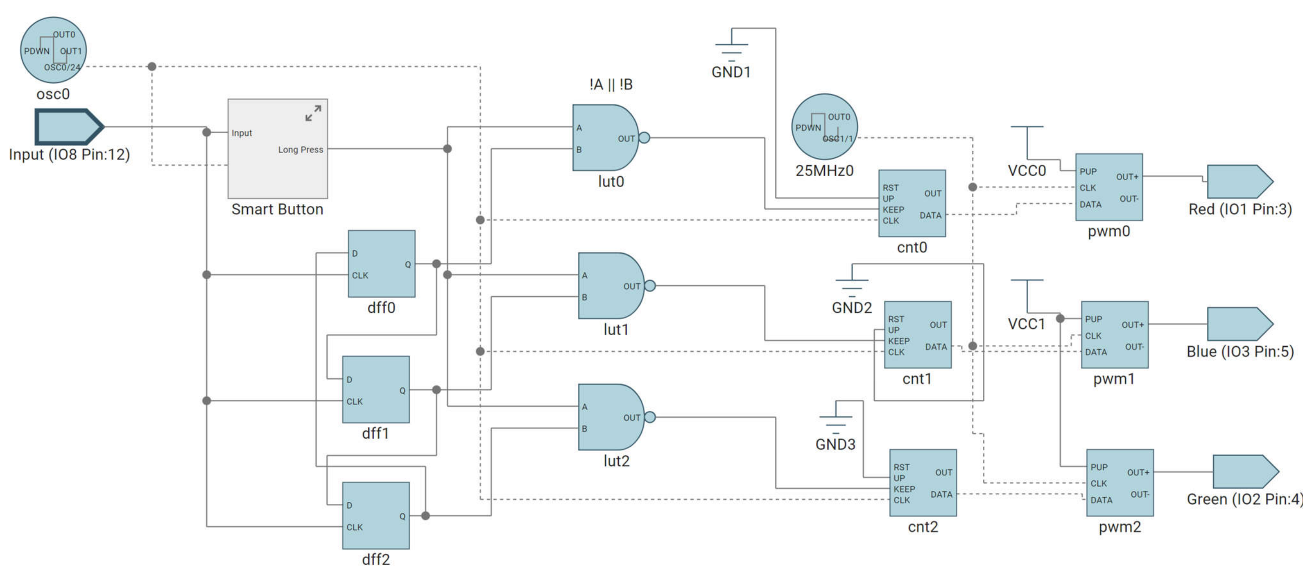

Figure 10. Rotating LED Control Method With Channel Select Feature in ICS

In this design, the LED brightness only changes after a short delay on a high input, so brief input pulses can rotate through the LEDs without affecting the output.

A scope shot showing the power-averaged outputs of the rotating LED control design with added channel select functionality is shown in Figure 11. Note that the blue output does not change due to a short pulse, allowing the controller to move from the green output to the red output.

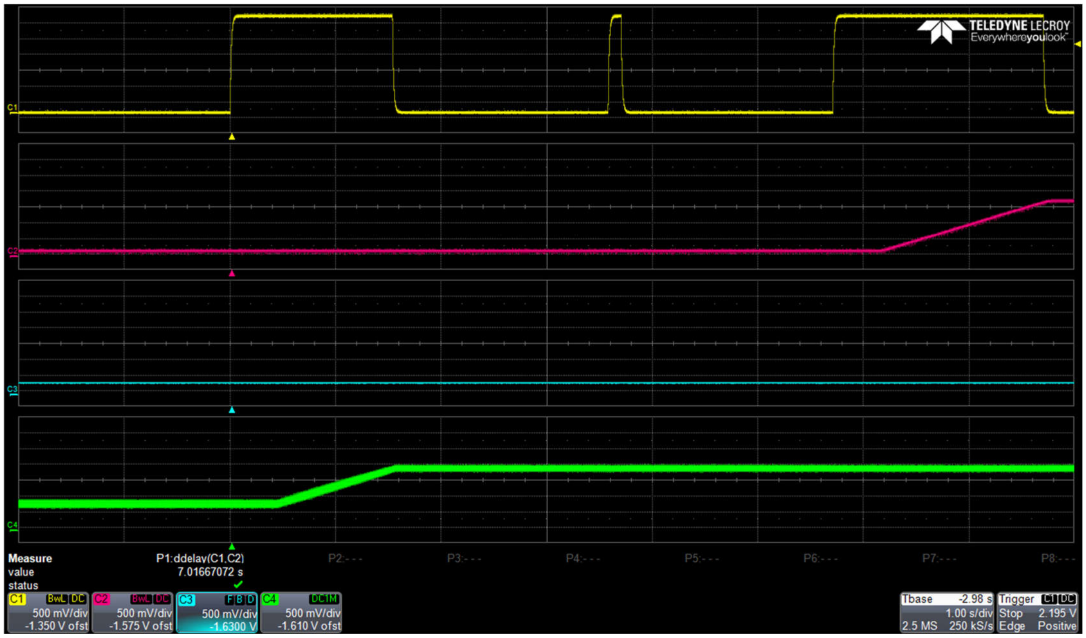

Figure 11. Rotating LED Control Method With Channel Select Feature Output

TPLD allow a designer to customize their control scheme to meet the unique constraints of their system using only a single device. This can reduce the total design size and offer unique benefits to the system, such as limiting the tax on the MCU to control the outputs or simplifying the user-facing interface. For more information on TPLD, visit the TPLD1202 product page or ask our engineers a question on the TI E2E™ Logic Support Forum.

Trademarks

All trademarks are the property of their respective owners.