SCOMP DE10-Standard LED Controller Peripheral

Hamza Waraich

Submitted

April 22, 2025

# Introduction

Text text text.

# Device Functionality

Users interact with the LED Controller by writing to three input registers over the SCOMP bus:

**Table 1. SCOMP LED Controller Peripheral Input Registers**

|  |  |
| --- | --- |
| **Register** | **Function** |
| MODE\_DATA | Selects one of three control modes—pattern (00), toggle (01), or PWM (10). |
| PATTERN\_DATA | Provides 10-bit mask indicating which LEDS to update. |
| DUTYCYLE\_DATA | Specifies PWM duty cycle applied in PWM mode. |

On assertion of the appropriate chip-select signal (CS\_MODE, CS\_PATTERN, or CS\_DUTYCYCLE) with WRITE\_EN, input values are stored in internal registers. In pattern mode, selected LEDs switch fully on or off according to the mask. In toggle mode, each selected LED inverts its current state. In PWM mode, the peripheral drives all selected LEDs with a free-running PWM counter using the provided duty cycle. Hardware reset (RESETN) clears registers and outputs to a known state.

# Design Decisions and Implementation

The design was partitioned into three key units to maximize flexibility and robustness.

## Mode Selector Unit

Implements combinational logic that interprets mode and pattern inputs, producing per-LED duty values of toggle signals based on MODE\_DATA, REG\_DUTYCYCLE\_DATA, and PATTERN\_DATA.

## Latch Unit

Uses CS\_PATTERN signal to update the ARRAY\_LED\_DUTYCYCLES array, allowing the user to seamlessly update the pattern displayed on the LEDs.

## PWM Unit

The pulse width modulation engine uses a free-running binary counter driven by PWM\_CLK. Duty cycles are compared in parallel across all ARRAY\_LED\_DUTYCYCLES entries, generating individual on/off outputs.

## Key Implementation Choices

Generics such as ARRAY\_LED\_DUTYCYCLES, MODE\_WIDTH, PATTERN\_WIDTH, DUTY\_WIDTH, and LED\_COUNT were implemented to enable the designer to easily scale to different LED counts or duty-cycle resolutions. Additionally, a synchronous reset behavior across all registers was implemented to ensure invalid mode codes default to pattern mode, preventing undefined states.

FI
A screen shot of a computer

Figure 1: Example of user interfacing with the LED peripheral’s pattern mode through SCOMP

The SCOMP code above demonstrates how the user can seamlessly select a mode (pattern mode), assign a duty cycle, and set a LED pattern using the peripheral. A similar procedure applies when interfacing with the other peripheral modes.

# Conclusions