

CmpE Internship Summer 2018

Design Doc Template

Rotary Encoder Controlled RGB LED

Authors & Editors:

Colin Schardt

Revision History:

|  |  |  |
| --- | --- | --- |
| Revision | Date | Description |
| B.1 | 6-25-18 | New Project Proposal |

Table of Contents

[Objective](#_f3ut0x3v9lxf) 2

[Background](#_52hjxc9p7ctt) 2

[Overview](#_yy2djj5i8omd) 3

[Detailed Design](#_avw48lx4cmyr) 4

[Caveats](#_op45shsa5n8y) 5

[Testing Plan](#_eejuf5av24ql) 5

[Unit testing scheme](#_h5m3fobrdlly) 5

[Integration Testing](#_fubwzje1thtf) 5

[Demonstration Project](#_6z1lojsl95dl) 5

# Objective

The objective of this project is to control the color of the onboard RGB LEDs of the Nexys 4 DDR FPGA board via a Rotary Encoder interfaced through the board’s GPIO pins using PMOD protocol.

# Background

My previously submitted project proposal included several flaws that would have made implementation impractical or impossible. Originally, I had planned on using a pushbutton PMOD module to change the colors of an OLED display. The framework for the OLED would have taken too much time to complete for this project, and the push button module is far to simplistic of an input device for purposes of this project. Also, my original designed focused heavily on controlling interrupts and their respective vector addresses to access interrupt service routines. While I would still be willing to include this in my new proposal, the framework within the MIPS processor module is still under development, and my design would be unable to function without it.

My new design will still include memory mapping and address decoding, which I believe to be of great importance to any potential late-term project for CMPE 127. It also now includes I/O devices for which the framework is already in place.

# Overview

This project takes input from a rotary encoder, a push button (inline with the encoder), and a binary switch, and provides output to the two onboard RGB LEDs. The rotary encoder will cycle through the entire color spectrum when turned in either direction, albeit in opposite order. Turning the encoder to the right will cycle Red to violet, while turning to the left will cycle violet to red. The inline button will save the current color to memory to be later accessed. The binary switch will determine of the color displayed on the LED is determined by the current position of the rotary encoder or by the color stored in memory. To successfully complete this project, the 127-toolkit provided by Khalil Estell will need to be utilized in addition to an additional logic for memory decoding that must be designed.

Implementation and final results of this design are not yet finalized, but two variations have been planned. One variation will toggle both of the LEDs with the switch while the other will only toggle one of the LEDs, allowing for the other LED to be used for continued adjustment of the displayed color.

This Project will be designed in Verilog using Vivado Design Suite on the Nexys 4 DDR FPGA board.

# 

# Detailed Design

This project will be written in Verilog for the Atrix-7 FPGA architecture on the Nexys 4 DDR board from Digilent. It will be written in Vivado Design Suite.

The 127-toolkit provided for this internship contains many useful modules ready for implementation. This includes a module for rotary encoders, which will be crucial to final functionality of this design. While not directly included in the provided framework, a module to handle input from the pushbutton as well as the binary switch will also need to be created. Lastly, a module for SPI interfacing may be necessary for future design changes, if any arise. This module is near completion by Nelson Wong.

With these modules complete, the main aspect of the project can be designed. A module or several modules will need to be created that will take inputs and send outputs to and from the Inputs, the LEDs, and the MIPS processor. The input from the encoder will increment or decrement a counter that will indicate the current position of the device. This number will be translated into a value corresponds to a color to be displayed on the LED. Input from the pushbutton will be interpreted as a pulse that when received will change the memory address for where the data from the encoder will be stored for a single clock cycle. The input from the binary switch will be used in address decoding to determine where in memory the output device should read from. The final functionality of this module or modules should require two read and write cycles to store data from the rotary encoder into memory and then to display that data on the LEDs in the form of various colors.

# 

# Caveats

* Most design decisions are very much in the preliminary stage
* Output device may be changed at team leader’s discretion
* This would likely cause drastic changes to overall functionality

# Testing Plan

### Unit testing scheme

I will write self-checking testbenches for every module I create.

### Integration Testing

In addition to testbenches, I will set several milestones of functionality that I plan on implementing on the Nexys 4 board to be certain that certain systems worm both independently and with other modules.

### Demonstration Project

The final project will be implemented on the Nexys 4 DDR board and will include all necessary physical components in order to test and demonstrate functionality. There are a low number of input devices, so it should also be possible to demonstrate every possible combination of inputs to prove the desired and expected results if need be.