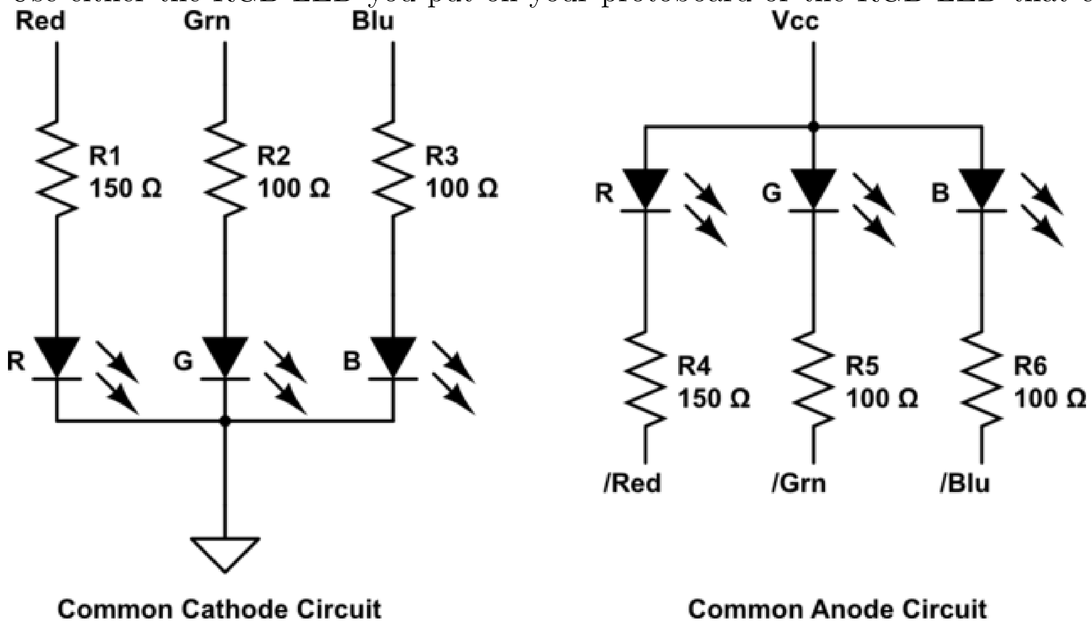
Digital System Design I CST 231
Winter 202X
Lab Assignment 9: Softly blinking LED
Check website for due date

This lab assignment is to be completed individually. You are allowed to use course materials, the internet, and ask the instructor or teaching assistant for limited assistance. You may discuss your lab with others in the class but your final implementation must be your own.

You have one week to complete this lab. It is your responsibility to manage your time in such a way that your lab is complete before the due date without requiring last minute instructor assistance or checkoff. While instructor will attempt to assist and check off the lab at the last minute, there is no guarantee.

1 Introduction

Use either the RGB LED you put on your protoboard or the RGB LED that came on your board.



`http://www.element14.com/community/community/raspberry-pi/blog/2013/08/29/embedded-pi--part-2`

- Your LED is either common cathode or common anode.

  Common anode LEDs have the higher voltage end tied together and you can turn each LED on by supplying a logic '0' to each LED's cathode.

  Common cathode LEDs have the lower voltage end tied together and you can turn each LED on by supplying a logic '1' to each LED's anode.

- You will need to calcuate clock division for this lab. Divide the input clock by the desired output clock to determine the count value. In order to determine the count value's register size, use the equation: ceiling(log(n)/log(2)) where n represents the maximum count value. Copy over an existing module and modify it.

- To determine how to implement a M second delay, you will need to take the reciprocal of the clock frequency. Once you have the period, divide M seconds by the period to determine number of times you will need to count up for an M second delay.

- Pulse width modulation means that the clock duty cycle is being varied. Clocks are generally varied in their duty cycle. By varying the duty cycle, we can increase or decrease the power supplied to a device. PWM is used in the control of various devices such as brushless DC motors, LEDs, servos, etc.

- In order to implement PWM, start with a basic clock divider. For a 50 percent duty cycle clock and a count up value of 1000, you would count up to 500 and switch the output each time. Eg. At zero set clockOutput = 0. At c=500, set the clockOutput = 1. Count up to c=1000 and reset to c = 0. Recall that your input clock speed would be reduced by 1000.

- To change this to a 25 percent duty cycle with the same count up value, you would count up to 750 and switch the output each time. Eg. At zero set the clockOutput = 0. At c=750, set the clockOutput = 1. Count up to c=1000 and reset to c = 0.

- Example code can be found at this website, although it probably isn't exactly how you should write yours:

  `http://www.fpga4fun.com/PWM_DAC_1.html`

2 Instructions

2.1 Wiring up the LED: Calculations

You already did this.

2.2 Testing and coding

1. Turn all three LEDs on to verify that they work. A simple assign statement setting each LED should suffice.

2. Blink the LED at a rate of 1 second (On for 1 second, off for one second.)

   Show calculations and explain how you arrived at the register size and counter values. Your clock on your FPGA board will vary. Verify your oscillator if needed.

3. Use pulse width modulation (PWM) to softly increase and decrease the brightness of all three LEDs at the same time.

4. Using pulse width modulation (PWM), implement the following in order. No glitching may occur.

   Slowly increase brightness on red LED (0 to 100 percent).

   Slowly decrease brightness on red LED.(100 to 0 percent).

   Slowly increase brightness on blue LED (0 to 100 percent).

   Slowly decrease brightness on blue LED.(100 to 0 percent).

   Slowly increase brightness on green LED. (0 to 100 percent).

   Slowly decrease brightness on green LED.(100 to 0 percent).

   Repeat.

3 Submission

Electronically submit the lab. No lab report is required. You will be graded based on the following criteria.

1. Demonstration will be due by (check website) 9 PM PST.
2. Submit the RTL and state machine diagram (if used).
3. Submit calculations for clock dividers and switching frequencies used.
4. Submit a one paragraph summary of how your lab works.
5. Submit your zipped project files. Remember to comment.