ES 4 Lab 1: Building circuits to do math

Prelab due 24 hours before your lab session, Feb 14-18
Lab report due at your lab time, Feb 22-26

1 Introduction

We've talked about how to represent numbers in binary, and how to manipulate bits with logic gates. In this lab, you'll combine that knowledge to build a circuit that multiplies two numbers.

After successfully completing this lab, you should be able to:

- Design a circuit from scratch to implement a collection of meaningful boolean equations
- Implement a circuit on your FPGA which has multiple inputs and multiple outputs
- Wire up DIP switches and LEDs to act as digital inputs and outputs

2 Prelab

We are going to build a circuit to multiply two 2-bit binary numbers, which we'll designate M and N. You can use M_1 and M_0 to represent the values of the individual bits, where M_1 is the most significant bit (i.e., $M = 2M_1 + M_0$). The result R will have four bits, which we'll designate R_3 through R_0.

> **P1**: Write a set of logic equations describing the output bits $R_3 \ldots R_0$ as a function of the input bits to implement $M \times N$. There are two ways to do this: either write out a big truth table and minimize the equations from there, or do the math symbolically to derive the equations directly.

P2: Once you implement your circuit on the FPGA, how will you prove that it always gives the correct result?

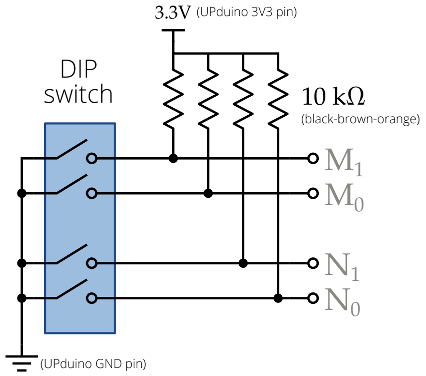
P3: What will you do when your circuit doesn't work? Give at least three things you could test to help figure out the problem. Remember that the problems could be outside or inside your FPGA.

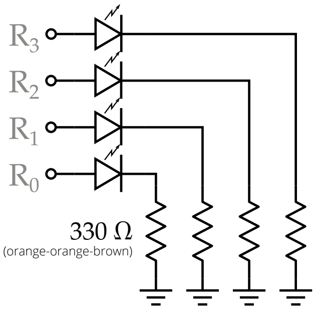
The following section is labeled "in lab", but your lab kit has everything you need to complete it. Feel free to get started right away!

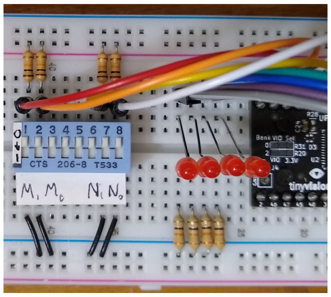
3 In lab

L1: Build the switch inputs and LED output according to the schematic below, and power it using 3.3 V and ground from the UPduino (the pins labeled `3V3` and `GND`). If you have access to a multimeter, use it to confirm that the switches are correctly producing high and low logic levels (0 and 3.3 V).

Note that because we're using pull-*up* resistors (instead of the pull-*down* configuration from the previous lab), the logic value will be 0 when the switch is "on" and 1 when it is "off", as shown in the example on the right (where $M = N = 00$). This is counter-intuitive, but get used to it now because it'll show up in several more labs.







L2: Build one part of your circuit (perhaps the R_0 output) in icestudio, and confirm that it works. You can use whatever FPGA pins you'd like (any of the numbered pins should work).

L3: Complete your circuit in icestudio, test it according to your plan, and demonstrate it to a TA.

4 What to turn in

Your lab report should contain:

- Standard "front matter" (see the lab reports handout).
- Your logic equation and logic circuit diagrams (copied from your prelab and corrected as necessary).
- A screenshot of your logic diagram in icestudio.
- A description of your plan for testing your circuit, and a table or diagram of your results.
- A description of problems you encountered, your debugging process for identifying them, and your solutions. This will be easy if you are taking notes as you work.
- Answers to the following questions:
  - What was the most valuable thing you learned, and why?
  - What skills or concepts are you still struggling with? What will you do to learn or practice these?
  - How long did it take you to complete the lab?