Systems I
Lab 1
An introduction / Building an adder
Due Thursday, Sep-11, 11:59 pm

1 Half-adder: Introduction to building circuits

We begin with a problem we have already solved: Adding two 1-bit values to produce a 2-bit result—a circuit known as a *half-adder*. Specifically, if a and b are those 1-bit inputs, and y and z are the 1-bit outputs,[1] we seek to perform:

$$\begin{array}{r} a \\ \underline{+b} \\ zy \end{array}$$

We worked out, in class, that the truth table for this addition problem, where 0 is *false* and 1 is *true*, is:

a	b	z	y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$z = ab$$
$$y = a \oplus b$$

1.1 CircuitVerse

Our first tool of the semester is *CircuitVerse*, which is an online tool that you use in your browser at `circuitverse.org`. It is a *logic circuit simulator* that allows you not only to draw a circuit using gates, but also to test the circuit by simulating the values that would move through those gates. We will use it to here to implement our half-adder as follows:

1. Go to the web site. In the upper left, click the *Log In* button.

[1] Early letters in the alphabet are typically used to name inputs, while later letters are for outputs. Sticking to these types of naming conventions will keep things clearer for us all.

2. On the login page, you may need to verify that you are not a robot, and doing so may involve a captcha. Go ahead and do that. Then click the button to login with *Google*. Doing so will bring you to a page in which you can select your college Google account; select that, and confirm you want to login using that account.[2]

3. Once you are logged in, click on the *Launch Simulator* button.

4. Here is where the work actually happens. You will see, on the left, the ability to choose *Input* devices that provide 0 and 1 values to the circuit, *Gates* that perform the logic and can be connected to one another, and *Output* devices that display the values on selected connections in the circuit. **I will demonstrate** how to select a and b inputs, an AND gate, and an LED for the a output to complete the circuit to generate one of the two bits of our half-adder.

5. Repeat what was demonstrated, creating the z function yourself. Test that it works correctly for all possible combinations of a and b values.

6. In the upper-left of the simulator window, click on the *Project* drop-down menu. Select *Save Online*. Name this project `lab-1-half-adder` and save it.

7. **Your task:** Complete your half-adder by implementing the y function, **using only** AND, OR, and NOT gates to implement XOR logic function.[3] Test the complete result again with all combinations of a and b.

8. In the upper right of the window, click on your own name to get a drop-down menu. Select *Dashboard*.

9. Your dashboard should show your `lab-1-half-adder` circuit. At the bottom of the pane for that circuit, click the *More* button.

[2] CircuitVerse may require that you go back to the login page, enter your college email address in the email line, and then select *Google* again. It will send an email to verify your email address, which you should receive and follow that email's instructions so that you may finally login.

[3] Yes, there are XOR gates, but this is an exercise in getting familiar with Boolean logic and gates, so do it the hard way.

10. A new window will open for more options of what to do with that circuit. Near the lower right, click *Add a collaborator*. A window will open in which you can enter email addresses. In that window, enter:

    ```
    sfkaplan@amherst.edu
    eleroy27@amherst.edu
    twstephens26@amherst.edu
    ```

2 Full-adder

Now that you have constructed a half-adder, it is time to create a *full-adder*. That is, you need to devise an adder than can sum **three** 1-bit values. This device is useful because it can add not only the two bits of some value, but also a third bit that can be considered the *carry* value from a less significant 3-bit addition. That is, a full-adder is capable of adding the bits in one column of significance in a larger addition problem. Thus, this device will have a, b, and c inputs, where the third is the *carry-in* value. Likewise, you will find that there are again two output bits, z and y, where z is really the *carry-out* of this addition. To perform this task, you should:

1. Lay out a truth table that enumerates the possible input values for a, b, and c. Notice that with 3 inputs, there are $2^3 = 8$ combinations of input values. Therefore, your truth table should contain the 8 entries that enumerate all of the possible input values, and then list the correct output values.

2. Define the outputs (z and y) in terms of the inputs (a, b, and c). If you have never worked with logic functions before, you should **ask for help**—I don't expect you yet to define these functions with ease merely by looking at the truth table. We will learn more in the coming week about formulating logic functions from truth tables.

3. Draw circuits that implement z and y.

4. Build the circuits that you've drawn in *CircuitVerse*, naming this one `lab-1-full-adder`.[4] Evaluate your work by testing every possible combination of inputs, determining that the outputs generated by your circuit match those shown in your truth table.

5. When you are done, go to the *CircuitVerse* dashboard, and repeat the steps your followed with the half-adder to *Add a collaborator*, sharing this full-adder circuit with myself and the TAs.

[4] You may now use any of the logic gates, including XOR.