INTRODUCTION TO COMPUTER ARCHITECTURE
LAB 1
An introduction / Building an adder

1 Knowing what the toys are

Don't do anything yet! We're going to spend a few minutes going over the various chips, boards, power supplies, and other devices that you'll need to use for this and other labs. Here's a list of the devices you'll be using, and some key information about them:

- **Breadboard and the ETS-7000:** These big units have a number of parts, some of which we'll never use. Their key feature is the *breadboard*—white boards with lots of little holes in them. These will allow us to structure our circuits, as we'll have to plug all of the chips and wires into them to make them work. Each breadboard has **power and ground channels** that provide the digital zeros (0 *volts*, *ground*) and ones ($+5$ *volts*, *power*) that we will use as inputs into gates. More on these below. The breadboard also has **wells** that chips straddle, with independent rows of connected holes that allow you to connect a wire to a chip pin by putting that wire in the same row.

  Note that the ETS-7000 also contain a number of devices around the periphery for input and output. For now, we are interested in the eight **switches** at the bottom that allow us to easily select power and ground inputs; we are also going to use the eight **LEDs** in the upper-right corner, allowing us to see the value flowing out of our circuits.

- **Power channels:** There are several power channels on each board, marked by red or blue/black lines. The colors respectively mean **power** and **ground**. There should be wires from the posts to two of the power channels. Each power channel is electrically connected inside the board, meaning that once a channel is connected to a source (in the upper left corner of the ETS-7000), power is available at each post in the channel. There should be wires connecting all of the red channels together and all of the blue/black channels together. If these wires are not already on your board, you can add them as you go.

- **Chips:** We'll use three kinds of chips today: 7404 (NOT gates), 7408 (AND gates), and 7432 (OR gates). *Data sheets* describing the operation of these chips are attached. As the semester goes on, you'll be using a number of other chips. Data sheets for these will be posted on the class web pages, under the Documents section. Some important notes on the chips and their use:

  - **Orientation:** Each chip has a mark, either a small hole or a small cutout, to indicate the top end. The top pin on the left is pin 1. The pin numbers go down the left side and up the right side. Most of our chips have 14 pins, but some will have more.

  - **Power and ground pins:** Two pins on each chip must be connected to power channels. The pin marked V_{cc} on the data sheet should be connected to the red channel. The pin marked GND should be connected to the black/blue channel. *Don't get them backwards, and don't install the chip upside down.*

  - **Installing chips:** Chips must be laid on the board with the right end up and with the pins straddling the groove. Each row of five pins is electrically connected within the board. By placing the chip so it straddles the groove, you are ensuring that there are separate sets of holes for each pin of the chip. Be sure to press each chip in gently and evenly, without letting one end or the other twist up or down. This will help you avoid breaking pins.

  - **Removing chips**: Remove chips carefully and evenly, by **using a chip extractor** and pulling straight up. Do not try to remove the chips with your fingers, as you'll probably bend or break the pins in the process. Return each chip to the place you got it.

- **Wires:** There are bins of wires available for your use, as well as kits that contain pre-cut, pre-bent wires that are color-coded by length. When you make a connection on the board, try to do it neatly and try to avoid using wires that are too long for the purpose. Your work will be far easier to test and correct if the wires are short, well organized, and labeled. When you need a wire of a certain length, you can either look for one or you can make one from a longer wire, using wire strippers to cut and remove insulation. You may even want to devise for yourself a wire-color scheme, such that each color implies something about the purpose to which the wire is being put.

- **Logic probes**: Logic probes are great debugging tools. If you connect them to power you can insert the probe tip into holes to determine if there is a positive voltage (a red light and a high-pitched sound), a ground connection (a green light and a low-pitched sound), or an open connection (no light or sound). *Probes don't always work in locations that are connected to LEDs.* As an alternative, you can always use a wire connected to one of the ETS-7000's LEDs as an ad-hoc logic probe.

We may not use all of these parts in the first lab, but you will need a number of them, and eventually you will use all of them and more.

2 What to build

It is unwise to construct a complete solution without testing its parts along the way. With a large circuit, you want first to construct the smaller circuits that will later compose the large one. With each smaller circuit, you should debug it thoroughly before connecting it to other smaller circuits. That way, you can have some confidence that certain portions of your larger circuit work correctly, reducing the number of possible portions of your circuit that may be designed or constructed in error.

2.1 Half-adder

We begin with by constructing a half-adder. Since this circuit is our first, you can follow these instructions to construct it and become familiar with the components involved:

1. Set up a single AND gate by using a 7408 chip. Set up an LED so that you can see the output of this gate. Try all four possible inputs to be sure that it's operating correctly. You've just constructed the circuit the generate the **carry** bit for your half-adder, where one input is your a value, and the other your b value.

2. Begin constructing an XOR circuit by using the NOT gates in the 7404 chip. You will want use two gates, one to produce $\bar{a}$, and the other to produce $\bar{b}$.

3. Use two more of the AND gates from the same 7408 chip as you used above. To one gate, connect a and $\bar{b}$, and to the other $\bar{a}$ and b.

4. Finally, bring the outputs of the two AND gates used in the previous step, and connect them as inputs to an OR gate on your 7432 chip. The output of this particular gate is $a \oplus b$, also known as the **result** bit.

5. Wire the outputs of these two circuits so that they appear adjacent to other another on an LED. Once you've done that, you have a half-adder!

2.2 Full-adder

Now that you have constructed a half-adder, use another part of your board to construct 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_i inputs, where the third is the *carry-in* value. Likewise, you will find that there are two output bits, r (for *result*) and c_o (for *carry-out*). To perform this task, you should:

1. Lay out a truth table that enumerates the possible input values for a, b, and c_i. 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 (r and c_o) in terms of the inputs (a, b, and c_i). 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.

3. Draw circuits that implement r and c_o.

4. Build the circuits that you've drawn. 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.

3 Finishing up and submitting your work

Clean-up and storage: At the end of each visit to the lab, you must clear off your workspace and store your work. Specifically:

1. **Label your ETS-7000:** Grab some masking tape, put a piece at the top of your device, and put your name on it.

2. **Put away your ETS-7000:** Stack your device on the shelves designated for your lab section.

3. **Put away tools and materials:** Put probes, chip pullers, and any other tools back where you found them. Put wire segments into their bins, and wire spools back to their place on the shelves. Any unused chips should be returned to their tubes.

4. **Clean up your space:** Creating these circuits leaves behind a detritus of wire segments and wire insulation. Sweep any such bits off of your workstation and into a garbage bin.

Submitting your completed work: Since the finished product of these assignments is a circuit, you will need to demonstrate its operation. For this lab, you need only demonstrate your *full adder*.

1. **Take a video:** Use your phone[1] to caputure a video of your full-adder operating correctly on *all eight possible input patterns*. Move systematically through the inputs, clearly showing the setting of the input switches and the outputs shown on the LED's for each possible input combination.

2. **Submit the video:** Go to the CS submission site at:
   `<https://www.cs.amherst.edu/submit>`
   Login and navigate your way to COSC-163, Lab-1, and submit your video file.

This assignment is due at on Sunday, Sep-15, 11:59 pm.

[1] If you don't have a smartphone or other device that can capture video, then let me or the TA's know, and we'll find one for you to use.