Tutorial A Wiring and Testing Circuits

Section I: Integrated Circuits

In order to perform computation using digital logic, we must assemble a series of *gates*, which represent some function combining binary inputs into an output. These gates can consist of functions such as AND, OR, NAND, NOR, and NOT. While these gates can be made out of individual transistors, we will be using pre-packaged gates, called *integrated circuits* or *ICs*.

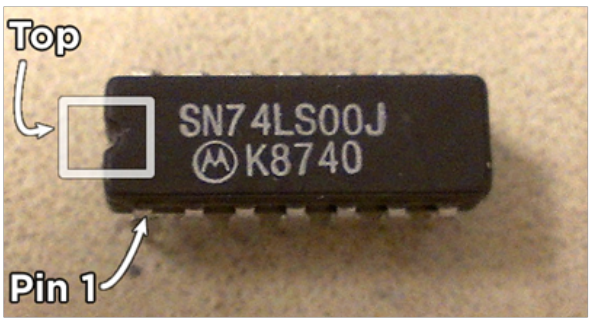

Figure 1. The 74LS00, an integrated NAND circuit.

ICs were invented in 1958 at Texas Instruments by Jack Kilby. Instead of manually assembling transistors together, we can draw (or *etch*) the parts of the transistor onto a *semiconductor* surface (often silicon). This allows us to make logic components that are much smaller, more power-efficient, and most importantly for us, pre-assembled. An example of an IC can be seen in Figure 1.

There are lots of ICs available on the market today. We will initially limit ourselves to the *7400 series* of ICs. The 7400 series provides chips that contain simple gates and the most basic of logic components, well-suited for our purposes. These chips are standardized throughout the industry, and their purpose can be figured out using their model number, printed on the body of each chip (see Figure 1). The chips we will use have a model number format of 74LSxx or 74LSxxx, where the x represents a digit. For example, the 74LS00, shown in Figure 1, contains quad 2-input NAND gates. (For the curious, LS means that these chips are low-power and fast-switching.) The 74LS32 contains quad 2-input OR gates.

Each of the pins on the edge represents a different input or output to the gates. In order to know how to wire these chips together, we use a *pin diagram*, which explains how the inside of the chip is designed. These are typically found in a *datasheet*, a document describing all of the technical details of the chip. Figure 2 shows the pin diagram for the 74LS00. In order to determine how pins are numbered, we look for a notch at the top of the chip (Figures 1 and 2). Using this orientation, we number the pins, starting with 1 at the top left and continuing in counter-clockwise order. As you can see in Figure 2, pins 1 and 2 are the two inputs into one of the NAND gates, and pin 3 will output the value of pin 1 NAND pin 2.

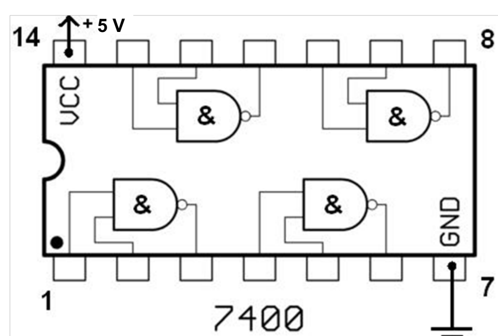

Figure 2. 74LS00 pin diagram. Wikipedia.

The chips used in lab belong to the *TTL* family of circuits (transistor-transistor logic). While these types of circuits are no longer used in commercial designs, they remain an excellent educational tool. TTL chips require two voltage values to be connected: **V_{cc}** (which is logical value 1) at +5V, and **GND** (logical value 0) at 0V. These voltages are used to power all of the transistors in the chip. The remaining pins are used to input and output logic.

NOTE 1 In TTL chips, **V_{cc}** is usually the top right pin, while **GND** is the bottom left. However, there are rare exceptions where this is not the case, so you should **always check the pin diagram**. Incorrectly hooking up these voltage pins will kill the chip.

Section II: Breadboard

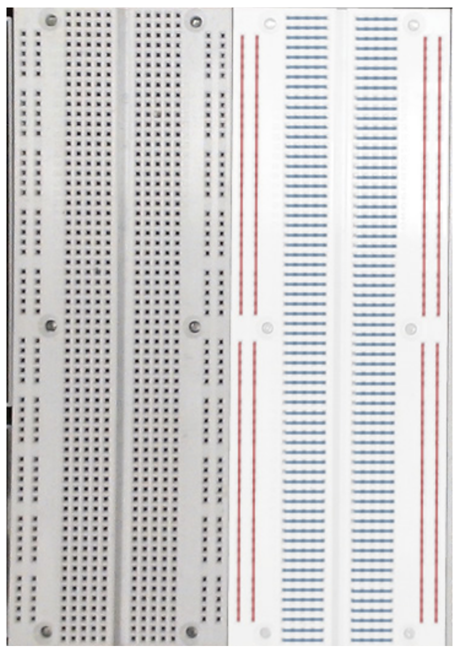

Figure 3. Breadboard used in lab. The internal wiring of the board holes has been overlaid on the right breadboard. The red lines show how the vertical rails are connected, while the blue lines show the horizontal rails.

We will use a breadboard to simplify designing our logic circuits. Figure 3 shows the breadboard that we will use. On the breadboard, we can plug in our ICs and use wires to connect them to inputs, outputs, and other chips. Each hole contains an electric contact that is wired inside the board. This wiring is shown in Figure 3, on the right segment of the breadboard. The lines show the wiring underneath. It is important to note that lines of the same color are **not** connected together. This means that on each side of the breadboard, each vertical column of holes has a discontinuity between the top and the bottom. Likewise, the horizontal rows are separated in the middle by a deep groove. There is no wire connecting the left side of a row to the right side. This allows us to straddle the IC over the groove without *shorting* different pins together. (A short is when an unintended connection incorrectly closes the circuit, which can often lead to overheating or sparking, both of which are dangerous.)

Section III: Assembling and Disassembling a Circuit

A. Placing an IC on the Breadboard

The chips in lab are contained inside *dual inline packages*, or DIP. This is a common type of chip packaging that allows us to easily fit the chip into a breadboard. You will notice that these pins will align nicely with the board, as seen in Figure 7.

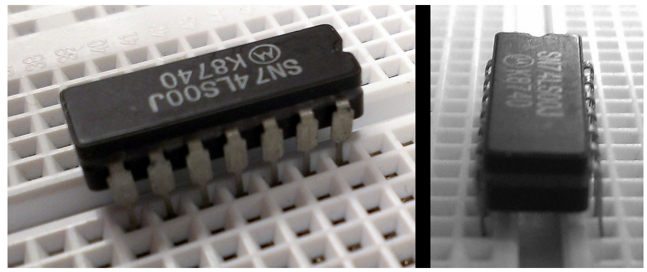

Figure 7. Placing a DIP chip onto a breadboard. Note in the right image how the chip sits over the U-channel.

Now, as shown in Figure 3, the horizontal rows are divided by the deep U-channel groove in the middle of the breadboard. We do not want two pins to come in contact with each other via the internal breadboard wiring. To get around this, we line up the chip such that it straddles this U-channel (see Figure 7, on the right).

When you insert chips, do not jam them in directly. First, align the pins to the openings without pressing too hard. Once the pins are properly aligned, use your thumb to press the IC in until it goes all the way in.

B. Connecting Everything Together

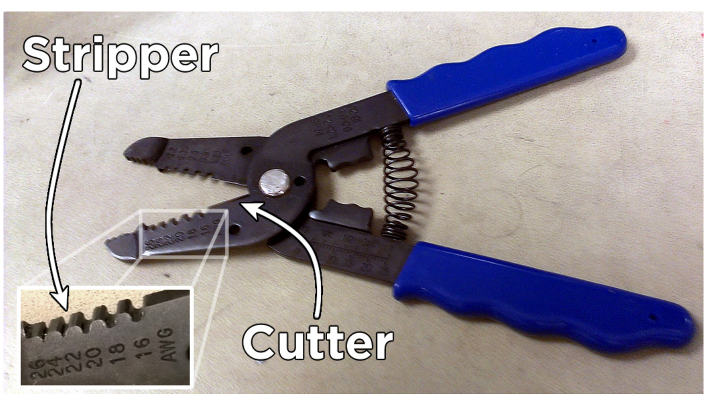

Figure 8. Wire strippers with cutter. Inset: stripper slots sized for specific wire gauges. 22 gauge wire is typically used.

Once the ICs are in place, we can use *wires* to connect them to each other (and to the switches and output LEDs). There is some prep work that needs to be done with wires. First off, there are two types of wires – *stranded* wires and *solid-core* wires. You should always use solid-core wires inside breadboards, as stranded wires cannot be easily inserted into the pin openings.

We will use a *wire stripper* (Figure 8) to cut the wire down to the right length. In its innermost part, the stripper contains a *cutter* that can be used to cut the wire down to a certain length. After that is finished, we must then strip the *insulation* (the plastic wrapping around the metal) down at the ends of the newly-cut segment, so we can have contact points. These contact points are known as *leads*.

Figure 9. Cut wire.

The stripper contains slots that allow you to strip insulation from wires of different thicknesses. The thickness of a wire is referred to as its *gauge*. The *American Wire Gauge* system, or AWG, is typically used. In AWG, as the number goes *up*, the thickness of the wire goes *down*. The wire available in the lab is predominantly 22 gauge, so we will use the slot labeled 22 on the stripper (see inset of Figure 8). Stick the part of the wire that you want to strip in, then clamp down, and pull the wire back out. Once you have completed this, you can choose to bend the leads to make your wiring layout a little less cluttered. The end result is shown in Figure 9 for a short wire.

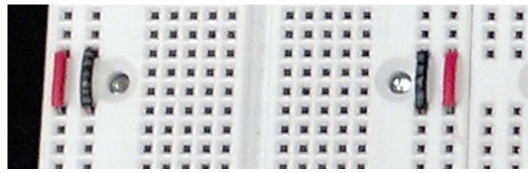

Figure 10. Bridges connecting the disjoint columns together in the middle of the board.

Short wires such as the one shown in Figure 9 are especially useful for the vertical rails in the breadboard. As you may recall from Section II-A, the vertical columns of the breadboard are split into two disconnected halves. Typically, we use these vertical columns to bring $\mathbf{V_{cc}}$ and **GND** case, it is convenient to connect the two halves of the vertical columns together, so they operate as a single voltage rail. These short wires can be used to *bridge* the columns, as shown in Figure 10.

When connecting a wire to the pin of an IC, remember from Figure 3 that in the breadboard, each horizontal row of 5 pin openings is connected together. The DIP design should only take up the pin opening closest to the U-channel, so you can use the remaining four openings to connect a wire to that pin.

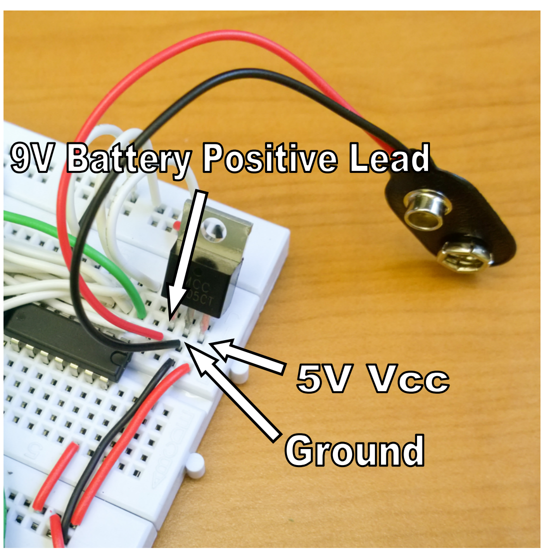

Figure 11. Wiring the power transistor

C. Power Sources

Once this is done, we can then hook up a power source to the rails. For our breadboard, we will use a 9V battery in combination with a power transistor (voltage regulator) to provide a +5V $\mathbf{V_{cc}}$. This setup can be seen in Figure 11. While not necessary, we often **use color coding on the supply voltage lines** to ease debugging. Typically, red is used for $\mathbf{V_{cc}}$, while black is used for **GND**. Inter-chip connections use the remaining colors. In our power setup, the power transistor's rightmost pin is used to distribute $\mathbf{V_{cc}}$. The middle pin is used for **GND**. The leftmost pin is connected to the positive terminal of the 9V battery lead (also red).

NOTE 3 It is extremely important to make sure that **only a single output is connected to a wire at any time**. If two or more outputs are connected to the same wire, a short will occur, which can damage both the breadboard and the ICs.

D. Sending Values to the Chips

In order to send binary values to the chips as our initial input, we must output a voltage of either $\mathbf{V_{cc}}$ or **GND** and wire it up to the input pins. For example, in order to input '1' to an input pin of a NAND gate, connect $\mathbf{V_{cc}}$ and the input pin using a wire. (Yes, input and output might get a little confusing here. Just remember – if we're setting the voltage, we are outputting it. This output will go into another input somewhere else, which will be connected using a wire.)

E. Reading Chip Output

Once you have wired your circuit together, you will want to see the output of the circuit in some form. We will insert some LEDs (*light-emitting diodes*) into the breadboard to show the value being output from the circuit. LEDs are polarized devices and only allow current to flow in one direction, from the anode to the cathode. Usually they are indicated by a long leg (anode, +) and short leg (cathode, -). Wire the cathode (-, short leg) of the LED to **GND** and the anode (+, long leg) to the signal you want to monitor. The LED will light up when that signal is $\mathbf{V_{cc.}}$ Please don't wire LEDs directly between $\mathbf{V_{cc}}$ and **GND**. This won't kill the LEDs immediately but it is not good for them.

F. Checking Your Circuit

After everything is wired, turn on the breadboard by connecting the power leads from the power transistor to the 9V battery. You can hold the leads on top lightly; just make sure they are touching. You should be able to see results being output on the LEDs. If your circuit is not operating as expected, first make sure that all of the wires are properly inserted. Most importantly, don't forget to wire $\mathbf{V_{cc}}$ and **GND** on **every chip**! Also make sure that your ICs are properly seated, and that the notch is facing the top. Another thing to check is that all of the configuration switches are set to either +5 or TTL.

NOTE 4 If you feel one of your chips getting very hot, **disconnect the 9V battery immediately!** This is a telltale sign of a short, which causes the IC to overheat. Make sure $\mathbf{V_{cc}}$ and **GND** are connected properly, and that no two outputs are connected. You may need to replace the chip.

If your circuit is producing an output, but that output does not match the expected truth table, you can always connect a wire from the middle of your circuit to the output lights to see if part of your circuit is working. Doing such piecemeal debugging is extremely helpful, and can often quickly isolate a problem spot. Another highly recommended approach is to test each part of your circuit as you assemble it onto the breadboard. If you wait till the end, the circuit may be too complex to easily hunt down the issue.

G. Circuit Disassembly

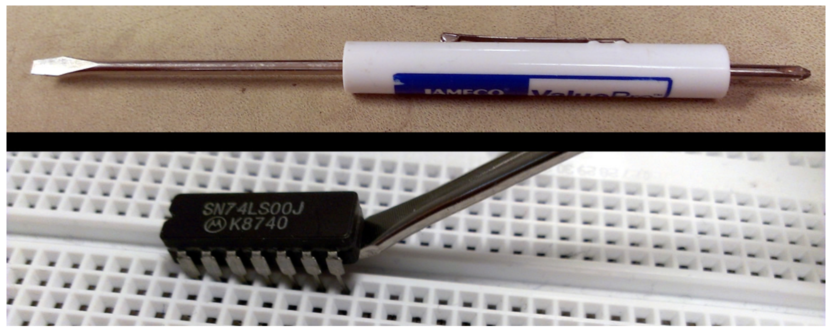

Figure 12. Prying the IC off using a screwdriver.

Once you are done, make sure to first disconnect the 9V battery. Afterwards, remove all of the wires. At this point, only the ICs should be left. **Do not use your hands to directly remove the ICs, as this will damage them.** We will use the flat edge of a screwdriver to remove these. Simply stick the flat edge under one end of the IC, parallel to the U-channel (Figure 12). Gently pry the IC up using the screwdriver. Once one side is out, you can stick the screwdriver under the IC body to pop it out of the breadboard completely. To avoid bending pins, you can use the screwdriver from both sides of a chip, pulling out each side gently. After this is done, you can lift it out the rest of the way by hand.

Section IV: Lab Protocol

Before leaving the lab you should always do the following:

- Turn off the breadboard by disconnecting the 9V battery.
- If you have used the computer, make sure to **save all files remotely** and then log off. Upon logging off, the computer automatically deletes any files that you created, so be careful!
- Clean your bench.
    - Place all wires in the plastic box at the front of class. Return any borrowed cutters and screwdrivers.
    - If you are not taking the chips home:
        - Use a small flat-head screwdriver to gently pop all chips out of the breadboard.
        - Return all ICs to their correct slot in the box, and return the box to a TA.
    - Throw out any stripped insulation or other garbage.