Objectives

After this lab experiment you should be able to:

1. Identify the truth tables for some fundamental logic gates
2. Build a multi-level circuit consisting of multiple logic gates
3. Use the Cadet II to wire a simple logic circuit
4. Debug common wiring problems

ICs needed (quantity): 7400(1), 7402(1), 7404(1), 7408(1), 7432(1)

Notes

• Prior to starting this lab, read Appendix A, Introduction to the Cadet II. The Cadet II will be used in almost all labs. You will use many of its capabilities in the course of the semester.

• Prior to most labs, you are required to develop a pre-lab. In Appendix B you will find part of an example pre-lab for this lab unit. You should examine this example pre-lab as you read through the description of this lab, and then use it as the starting point for your own pre-lab. You will bring a copy of the pre-lab with you to lab, which you will turn in prior to the start of lab.

A pre-lab may have errors in it; the pre-lab in Appendix B does. That's OK. If you discover an error in your pre-lab during the lab, be sure your lab observations note the error and the correction. Don't change your pre-lab; make the changes in your observations.

• A little background about circuits, and a common wiring problem

Most of the ICs you will use in the lab have three types of pins: (1) power, (2) inputs, and (3) outputs. All the circuits you build can be broken down into connections from one output or power source to one or more inputs. You will never connect one output to another output or one output to a power source. Many problems encountered in circuit design can be traced to a violation of this rule.

The IC's power pins are necessary for the IC to function; if they are not connected, the IC will not function correctly. You will find each IC has two power pins; one labeled Vcc and one labeled GND. You can think of these two pins as the power connections for the positive and negative poles of a battery. Voltage is always measured relative to the negative pole of the battery (the ground). Because of this, the GND voltage is also called 0VDC (DC stands for direct current). The positive pole is the largest positive voltage difference from GND. However, different ICs require different Vcc voltages to operate. This is similar to how a battery-operated CD player may need different types of batteries than a cellular phone. The ICs we will use in lab use two voltage levels; +5VDC and +3.3VDC. The red connector of the Cadet II (see the description of #18 in Appendix A) is the +5VDC supply and the black connector is the GND supply.

Normally you (as a logic circuit designer) don't worry too much about these details; your main concern is logic 1 and logic 0. All the inputs and outputs in your digital circuit are in terms of
one of these two values. In class, we rarely worry about any other values. However, in the lab you need to keep them in the back of your mind. Almost always a logic 0 is represented by a GND voltage (or something very close to it) and a logic 1 by a Vcc voltage (or something close to it). So in reality when you're talking about inputting a logic 1 to a gate, you're in fact applying a voltage of Vcc to the gates input pin. Similarly when a gate outputs a logic 1 it is really outputting a voltage of Vcc. Sometimes you may even connect a wire from an gate's input directly to the Vcc power source if that input will always be a logic 1.

What's so important about these voltages? Well, remember that the purpose of any circuit is to route electrons from one place to another, hopefully through some useful pattern. The number of electrons that move through some point in a circuit during a given time is the current. These electrons, like everyone else, would prefer to move through the easiest path; the path of least resistance. The lower the resistance through a path, the higher the current. If you connect Vcc directly to GND you create a very low-resistance path for current to travel -- this is a short circuit, which is bad. One of two things can happen: (1) the wire heats up as lots of current moves through it or (2) the wire heats up to the point where it breaks. At either of these points fire is also possible so this clearly sounds like a bad idea.

Connecting two outputs together can, at certain times, cause a short circuit to occur. When both outputs are the same, the voltage difference between the two is small and only a negligible amount of current flows from one to another. If the outputs are different (one is logic 1 and the other a logic 0), you in fact have connected Vcc to GND and a short circuit occurs. Lots of current can flow which, as we said, is bad. Another thing which happens is at least one of the outputs (and probably both) will appear to have the wrong logic value.

One way to detect a short circuit on the Cadet II is to observe the 7 segment LED display when the power switch is turned on. Normally the display will light up to show "00" when your circuit is not using it. If there is a short somewhere on your board, the display will not light at all. You should turn off power immediately and trace your circuit for the problem.

• Other common problems

Most problems encountered in the lab can be traced to bad wiring. This is not to say incorrect wiring; you can construct a circuit with all the proper connections that will not work because of a loose connection. When this is the case -- you've double-checked and triple-checked the wiring and it is correct -- what should you look for?

1. Check for power and ground along the power buses
2. Check for power and ground to each chip
3. Check for bad chips (including bent or missing pins)
4. Check for any unconnected inputs (no wires at all)
5. Check for "unusual" voltages at each input and output
6. Check that each gate and device works properly
7. Rebuild the circuit from scratch as a last resort

Experiment descriptions

Before building or changing any circuit, be sure the power switch of the Cadet II is turned off. Never add or remove chips from the breadboard area while the power is on. This is not so much for your safety as for the safety of the chips!
Experiment 1: Typical Logic Gates

Boolean algebra is based upon three operators: AND, OR, and NOT. These symbols are also commonly used in schematic diagrams for circuits. It should come as no surprise then that there are IC chips for each of these types of operators. Additionally there are chips for NAND and NOR gates, and even chips with more than two inputs per gate (such as the 3-input NAND gates and 4-input NAND gates).

The logic gates above are among the most frequently-used ICs in lab and industry. The purpose of the first experiment is to become familiar with the logic gates as well as their implementation in an IC. The ICs we have available in the lab are almost all in the 7400 series. You should verify the truth table for the NOT, and the 2-input AND, OR, NAND and NOR ICs. Choose the proper IC by using your data sheets.

Show the designs to test the first gate of each of these ICs. Be sure to include chip number, GND and Vcc pin numbers, and input and output pin numbers for the gate you use. Note in your notes how you will confirm your design and what you expect to observe.

1. Connect the Vcc and GND connectors (#18) to the buses on the breadboard using copper wire.
2. Install the 7400 IC on the breadboard.
3. Using your pre-lab design as a guide, connect its power pins to the Vcc and GND buses on the breadboard.
4. Connect the inputs of the first gate to logic switches (#8) and connect the output of the gate to an LED. (#17)
5. Turn on the Cadet II and record your observations on the worksheet. Turn off the Cadet II before removing the circuit. Repeat this for each IC.

Experiment 2: A Multi-level Circuit

Almost every digital circuit is not composed of a single logic gate, but of many combinations of gates. Any complex boolean equation can be broken down into the three basic AND/OR/NOT operations. With this, it's possible to implement each equation using the basic gates you've seen so far. The proper combination of gates, wired one after the other, give the final output.

For example, consider the boolean equation

\[ X(A, B, C) = A' \cdot B' \cdot C + A \cdot C \]

By simple examination, we can see that this requires two inverters, one OR gate, and two (or three) AND gates. [Since AND is a two-input operator, \( A' \cdot B' \cdot C \) can be rewritten as \( (A' \cdot B') \cdot C \)]. So for the circuit implementation of this equation we could

(a) invert A
(b) invert B
(c) AND (a) with (b)
(d) AND (c) with C
(e) AND A with C
(f) OR (d) with (e)
Seems simple, right? It is. (Notice, by the way, that even though we need six logic gates, we only use three ICs. Each 7408 IC contains four 2-input AND gates and each 7404 IC contains six inverters). The only thing which can be tricky is making sure all the wires go to the right place. Therefore, one important point to emphasize is that it’s a good idea to test each part of the circuit as you build it. When you start to build a complex circuit, start from the bottom (with the inputs) and work you way toward the top (the final output). Test each gate's output as its inputs are wired. If you find that a gate gives the wrong output, you only need to debug the problem with that gate because you know all the gates prior to it work properly.

One common misconception is that each variable in the boolean equation comes from a separate switch, even if that variable appears multiple times. For example, some people (not you, of course), would connect one logic switch to AND gate (d) for input C and a second logic switch to AND gate (e) also for input C. Others will use one switch to generate input A and a second switch to generate A’. Only use one switch for each input variable.

1. Draw in your pre-lab the circuit schematic for the circuit implementing the boolean function above. Remember to label each input and output pin, as well as the Vcc and GND pins and the chip numbers.
2. Determine the truth table for each of the gates (a) through (f). Record them in your pre-lab as well. This truth table should include all eight input combinations (in other words, draw the truth table for the complete function, then show what each gate does).
3. Connect the logic switch (#17) for variable A to the inverter as shown in your pre-lab schematic. Temporarily connect the output to an LED (#18). Verify that the inverter’s output matches the truth table in your pre-lab. Disconnect the output from the LED when finished.
4. Repeat the wiring for gates (b) through (e), each time checking the output using an LED.
5. Connect the final gate (f) and verify the completed circuit against its truth table.