

Basic Electrical and Digital Laboratory Concepts

Computer Science 240

Laboratory 5

This laboratory uses a combination of physical experiments, computer simulations, and paper and pencil exercises to help you investigate the electronic circuits level of computer organization.

The lab begins by investigating the operation of the PB-503 Proto-Board, the circuit design tool you will be using to perform physical circuit experiments during the next few labs. By making some simple connections and measurements, you will be introduced to the use of the proto-board for constructing and testing circuits, and also to basic electrical theory and practice.

You will then see how you can build simple logical gates from actual transistors, as you learned about in lecture, and verify their correct operation. You will do some simple designs using basic digital logic, and work through some problems using universal gates and proving circuit equivalency. You will also simulate a circuit using LogicWorks®, a digital design application that will be used for the rest of the semester.

Voltage, Current, and Resistance

Exercise 1. Connect the circuit in Figure 1 on your breadboard. Use the +5V voltage supply on your breadboard, a resistor, and some wires.



Figure 1. A simple circuit

1. IF we were to remove the resistor from the circuit and replace it with a piece of wire, what would happen (don't actually do it!)?

Short circuit (infinite current, possible destruction of the circuit).

- o What if we removed it and did not replace it with any device?

Open circuit, no current would flow.

- o Calculate the current in the circuit using Ohm's Law, with the indicated voltage and resistance values (label your answer with the proper unit for current):

$$I = V/R, 5V/22000\Omega = .23 \text{ mA}$$

- o If the resistor had a larger value (voltage source the same), would there be more or less current in the circuit?

Less current (inversely proportional to the resistance).

- o If the voltage source had a larger value (resistor the same), would there be more or less current in the circuit?

More current (directly proportional to the voltage).

- o Assume a resistor has the following bands of color.: (green, orange, yellow, gold). What is the value of this resistor?

530 KΩ

Transistors

Exercise 2: Basically, a transistor is a switch that can be turned on or off. Transistors are used to implement circuits that produce basic logic functions.

Get 3 2N3904 transistors from the drawer. Each transistor has 3 leads, marked E B C. Connect one of the transistors and a 1.8K resistor as shown below (use PB1 on the Protoboard for the A input):

1. Calculate the current in the circuit using Ohm's Law, with the measured voltage value, and the known resistance value (use the current units to label the current):

$$I = V/R, 5V/1800\Omega = 2.77 \text{ mA}$$

2. Record the value of Q at the output for the possible inputs for A:

A	Q
0	1
1	0

3. How would you describe what the circuit does with the input A to produce Q?

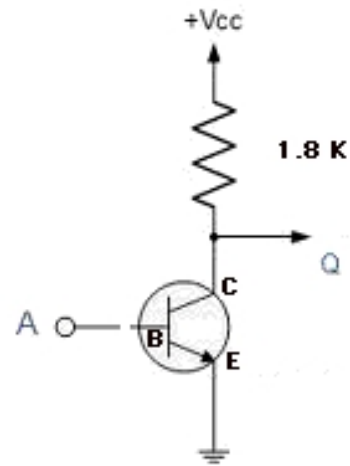


Figure 2. Transistor logic gate, version 1

Add another transistor to your current circuit, as shown in Figure 3. Use PB2 as the input for B.

4. Given the following inputs, record the value of Q at the output:

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

5. Describe the relationship between the two inputs and the output Q: $Q = A \text{ NOR } B$

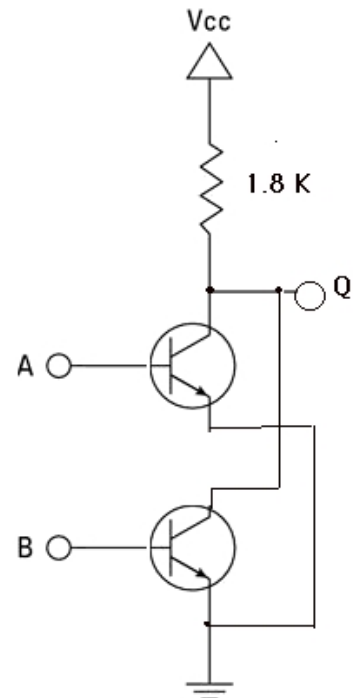


Figure 3. Transistor logic gate, version 2

DO NOT BUILD THE NEXT CIRCUIT

BUT, if you did add one more transistor and another resistor to your original circuit, as shown in Figure 4:

6. Using the circuit in Figure 4 with the following inputs, record the value of Q at the output:

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

7. Explain why adding the third transistor causes a different logic function to be produced:

The third transistor acts as a NOT gate, so produces the complement of the output from the NOR gate (which is an OR gate, as shown by the truth table).

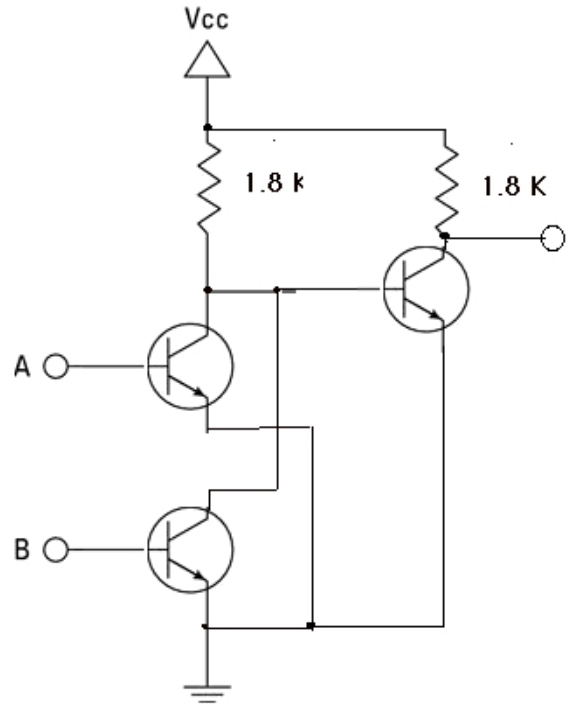


Figure 4. Transistor logic gate, version 3

Basic Gates and Boolean Logic

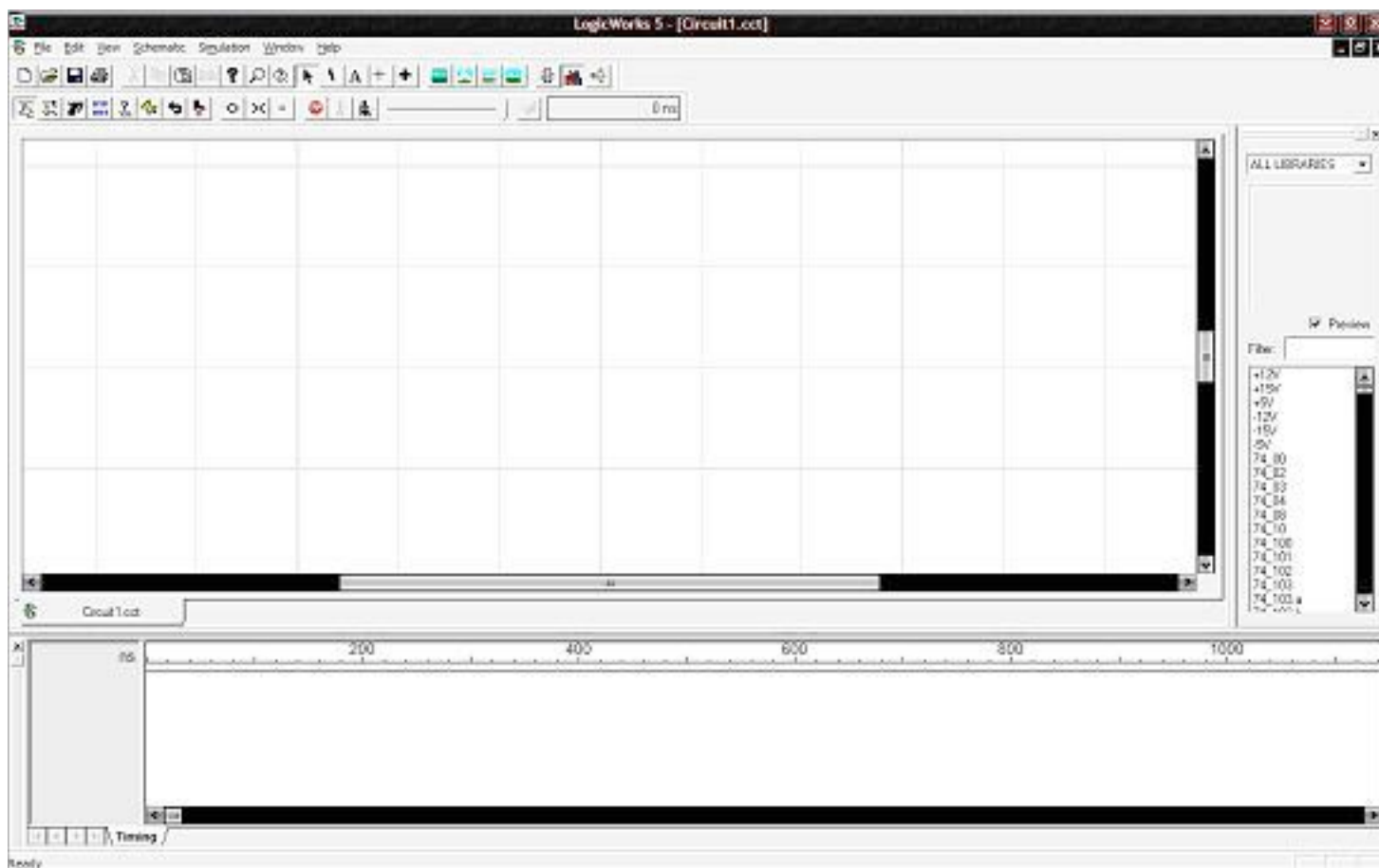
LogicWorks

Exercise 3: From lecture, you have learned that any Boolean function can be represented using three operations: NOT, AND, and OR. A NOT gate can be implemented with 1 transistor, AND/OR with 3 transistors, and NAND/NOR with 2 transistors. NAND and NOR are **universal gates**, and can implement any of the primitive operations NOT, AND, and OR.

1. Use LogicWorks to draw a circuit that implements the function $F=AB+BC$.

1. Go to the PC and launch LogicWorks (from the Start menu, you can find it in the All Programs pop-up menu, or as an icon on the Desktop).

When the application opens you will see **Drawing** and **Simulation Toolbars** at the top (as labeled below), a large **Design Window** below the toolbars, a **Parts Palette** on the right, and a **Timing Window** on the bottom:



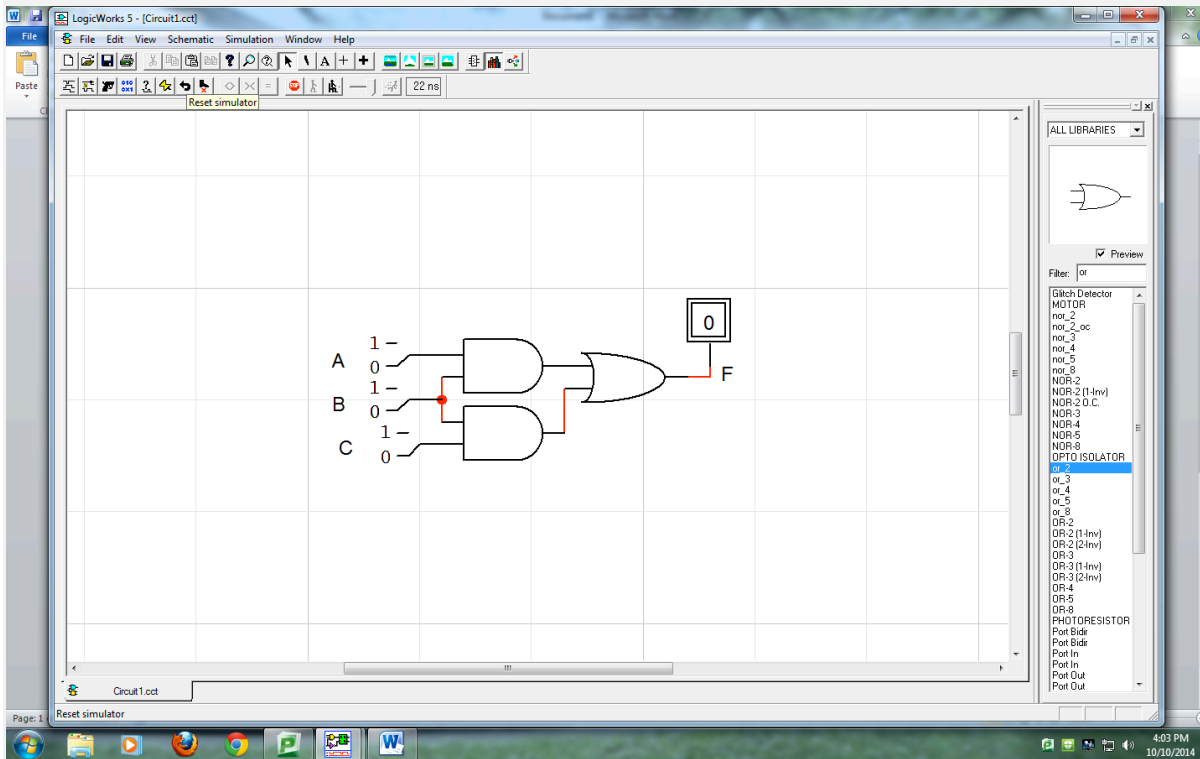
There will not yet be a circuit in the **Design Window** – you will see a pop-up window asking you whether you want to create a new empty circuit. Begin by clicking on the **Create** button.

Follow the steps below to simulate the circuit $F = AB + BC$:

- Go to the **Parts Palette**, and select the **Simulation Gates.clf** library.
- Find the **AND-2** gate, and double click on it.
- Move to the **Design Window** and place the gate somewhere in the window by clicking. Place a second gate by clicking again. Press the space bar to deselect.
- Repeat with an **OR-2** gate (double-click, click to place, and space bar to deselect).
- Connect the devices together by clicking and dragging the mouse between the two points on the design that you wish to connect (this may take a bit of practice – ask for help, if needed).
- Label the inputs and the outputs (A, B, C, and F) with the Text tool.

After creating the circuit in LogicWorks, do the following to copy and paste your circuit into this document:

- Close the Parts Palette and Timing Window.
- Zoom in to increase the magnification of the circuit displayed (use the Magnify tool for this purpose).
- Minimize the window size so it is just large enough to hold your circuit
- ALT-print screen on your keyboard to put a screenshot of the window into the copy buffer
- Paste into this document:



- Re-open the Parts Palette, and select the **Simulation IO.clf** library.
- Find **Binary Switch** in the list, and double click it (a binary switch will allow you to input a pulse to your circuit, similar to the pushbutton switch used in the earlier experiment).
- Place three binary switches in the Design Window, and hit the space bar when done, to deselect the switch.
- Connect the switches to the inputs of the AND gates to simulate A, B, and C. Find the **Binary Probe** in the Simulation IO library, and double click it (a binary probe is similar to an LED - it indicates whether a high or low value is being presented at an output).
- Place one binary probe in the Design Window.
- Connect the binary probe to the output of the OR gate.

At this point, the circuit should be operational (LogicWords assumes that +5V and ground are already connected). To actually test its operation, you must go to the Simulation Toolbar, and take the following steps to start the simulation:

- Click on the **Clear Unknowns** button from the Simulation Toolbar:



(you should see the binary probes go to '0')

- Click the binary switches to the different combinations of A, B, and C, and check that the output of the OR gates matches what you expect for $F = AB + BC$ (use your earlier truth table to verify your results).

Build the circuit on the protoboard

Gates come on a chip, or integrated circuit (IC) in standard packages, so keep the following in mind when wiring a circuit:

- Only **one** type of gate is available on a chip (for example, four 2-input AND gates are available on a single, 14-pin chip, IC 7408).
- Each chip has +5V and ground inputs, which must be connected to power the chip.
- To understand how to connect the gates, consult the TTL Data Book or an online source, such as:

<http://www.kpsec.freeuk.com/components/74series.htm>

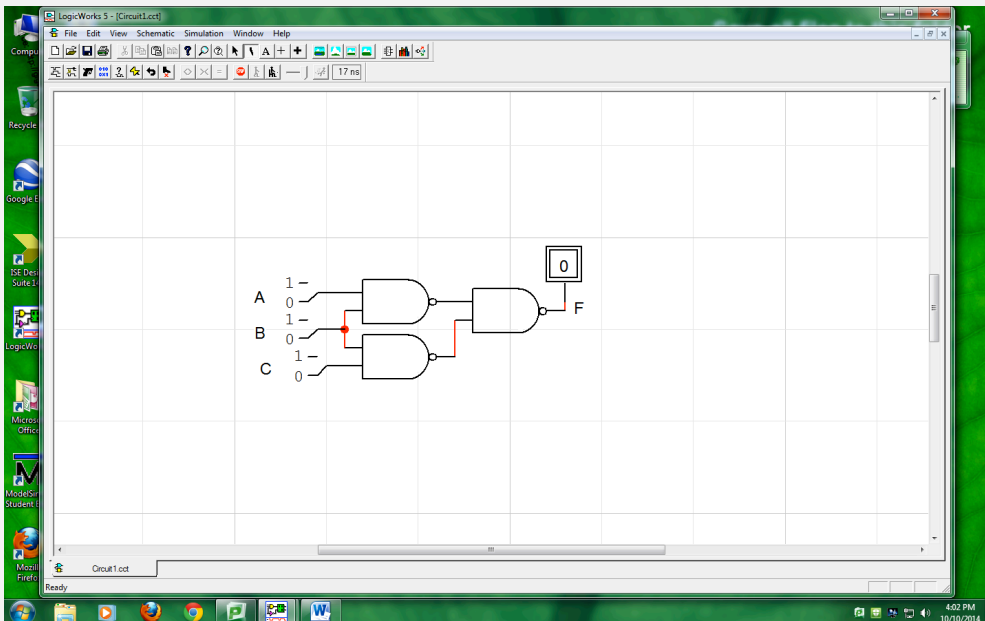
Build the circuit on your protoboard, using 3 Logic Switches for inputs A,B, and C, and an LED display for output F. You should only need two chips. DON'T FORGET to connect +5V and ground for each chip.

2. Record the results for all the possible input combinations, and verify that your results are correct:

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

3. Now, design a circuit which implements F using NAND gates only in LogicWorks (you should be able to edit your circuit from the previous problem).

Copy and paste the circuit using NAND gates from LogicWorks to this document:



To build this circuit, how many gates did you use?

3 NAND gates

How many chips are needed to supply this number of gates?

Only 1 NAND chip (which has 4 gates available)

How many total transistors (in the underlying electronics for the gates) are required for this design?

Each NAND gate uses 2 transistors, so $3 \times 2 = 6$ transistors

Compare your answer to what would have been required for the original circuit, using AND and OR gates, and describe the advantages and disadvantages of using a universal gate in a circuit design:

The original circuit used two AND gates and 1 OR gate, which required two chips (one AND chip and one OR chip). Each gate uses 3 transistors, so the total is $3 \times 3 = 9$ transistors. So, for this particular circuit, using only NAND gates is an advantage in terms of requiring fewer chips and transistors. However, for some functions, using universal gates only may result in a larger number of gates overall, so that the benefit is not achieved.