Lab 4: Transistors and Digital Circuits

I. Before you come to lab...

A. Please review the lecture from Thursday, March 6, on Binary Numbers and Manipulations. There were two different handouts containing the lecture notes; if you picked up the slides at the start of lecture, check online to make sure you have the version that is posted on the website.

B. Please read this handout.

C. No pre-lab questions this week! Just reading. But please, please do the reading, or the lab will be no fun at all. In particular, make sure you understand how a transistor acts like a switch.

II. Background

A. Binary arithmetic: see Prof. Whitesides's lecture notes. If you don't understand subtraction, don't worry. The lab only really deals with addition.

B. Boolean algebra

1. This is a branch of mathematics which deals with true/false statements. A boolean variable can be either 0 ("false") or 1 ("true").

2. There are three elementary boolean operators:

a. NOT

This is an operator which takes only a single argument. If X is defined to be NOT A, then X is false when A is true, and true when A is false. The NOT operation can be summarized by the following "truth table", in which inputs appear to the left of the vertical line and outputs appear to the right:

A	X
0	1
1	0

b. AND

This is an operator which takes two arguments. If X is defined to be A AND B, then X is true when both A and B are true; otherwise, X is false:

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

c. OR

OR also takes two arguments. If X is defined to be A OR B, then X is true if A is true, if B is true, or if both A and B are true. In other words, X is only false when both A and B are false:

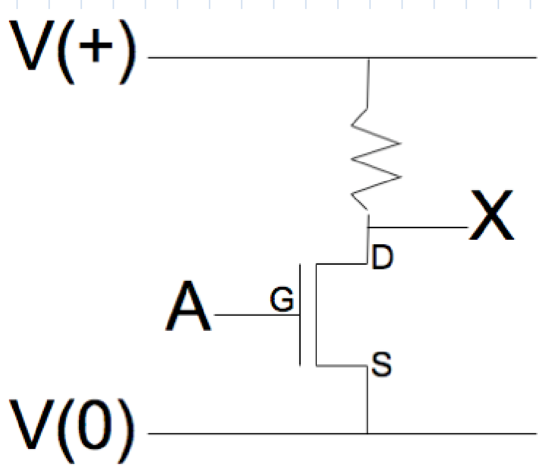
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

3. There are also other operators which can be expressed as combinations of NOT, AND, and OR. For example, NAND, which is short for "NOT AND." A NAND B is equivalent to NOT (A AND B). NOR is defined similarly.
4. The other important operation is XOR, which is "exclusive OR." XOR has the following truth table:

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

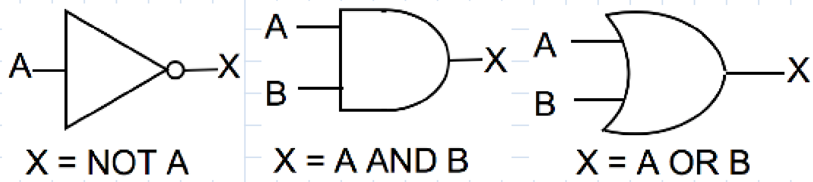
C. Digital circuits

1. Digital circuits are ones in which we are not interested in the exact value of a voltage, only whether it is "high" or "low." Most digital circuits are powered from +5 V ("high") and 0 V, or ground ("low").
2. Because any point in a digital circuit has a voltage which is either high or low, digital circuits naturally correspond to binary thinking: the binary digits 0 and 1 correspond to low and high, respectively. They're also great for doing Boolean logic: low corresponds to false and high corresponds to true.
3. The basic building block of digital circuits is the **transistor**, an electrical component which is essentially a voltage-controlled switch. One terminal of the transistor, the gate, determines whether current is allowed to flow between the other two terminals.
   - **a.** If the gate voltage is high, then the transistor is "on" and current will flow from drain to source, with essentially no resistance (and no voltage drop). It is as if you connected a wire between drain and source.
   - **b.** If the gate voltage is low, the transistor is "off" and current cannot flow from drain to source. It is as if there were no electrical connection at all between the drain and the source.
   - **c.** **In either case**, there is no electrical connection between the gate and either the drain or the source. The gate itself merely acts to allow or block the current from drain to source.
4. One key to understanding the operation of transistors in digital circuits is that you can generally assume that **outputs draw no current**. Consider the following arrangement (taken from page 11 of the 3/6/08 lecture, with a heading of INVERTER = "NOT"), which is extremely common:



Here, X is the output and is tied to the transistor drain; this arrangement is an implementation of the boolean statement X = NOT A. If you apply the rule that X, the output, draws no current, then it follows that the resistor current is equal to the drain-source current of the transistor. This makes it much clearer why V(X) = 0 when the transistor is on (so that there is a voltage across the resistor), but V(X) = V(+) when the transistor is off (there is no voltage across the resistor, because there is no current).
5. Boolean operators implemented in a circuit are called *logic gates*. (This use of the word "gate" has no relationship

to the gate of a transistor.) Some gates, like the NOT gate picture above, can be constructed out of a single transistor; others require complicated combinations. The symbols of the various gates are shown below:

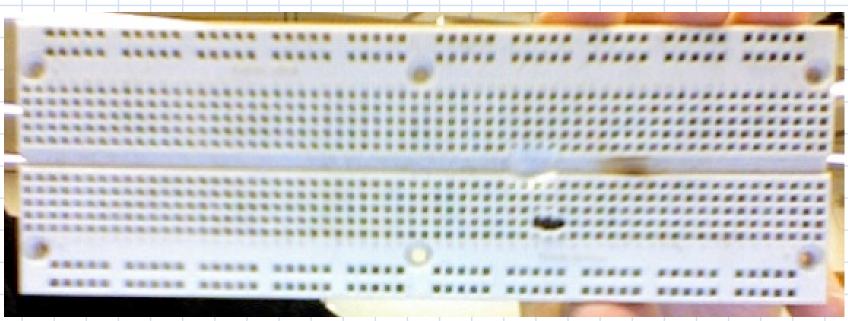


III. Introduction

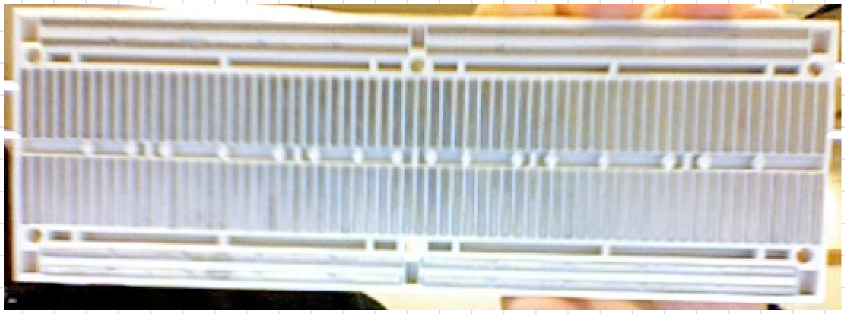
- **A.** In this lab, you'll receive a gentle introduction to digital circuits. Using transistors and resistors, you'll build some simple logic circuits, and then test your work. Finally, you'll demonstrate the ability to perform binary addition using the circuits you've constructed.
- **B.** Objectives for this lab:
  - **1.** Learn and understand how a breadboard works
  - **2.** Learn to use transistors to implement simple logic gates
  - **3.** Test the truth tables of logic gates using a voltage probe
  - **4.** Show how to combine several logic circuits to make a binary adder
  - **5.** Implement a new truth table using the components you've learned to build

IV. Materials

- **A.** Power supply
  - **1.** This DC power supply maintains a 5-volt difference across its leads. The black terminal is ground, and the red terminal is +5.0 V.
  - **2.** The power supply should connect to one of your breadboards; the top rail usually is at +5 V and the bottom rail is at ground.
- **B.** Breadboard
  - **1.** Instead of using lots of alligator clips, you'll be building your circuits on a breadboard. A breadboard is a piece of plastic with lots of little holes. If you stick a wire into one of the holes, it "grabs" the wire and makes an electrical contact with it.



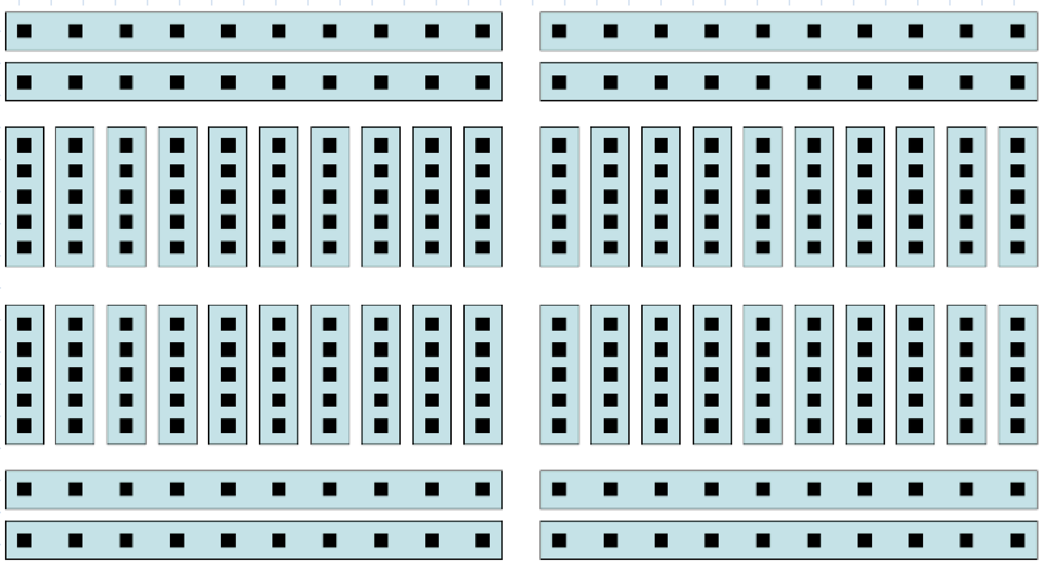
  - **2.** Beneath the surface of the breadboard, there are lots of electrical connections between the holes, as you can see from the following photo of the back of a breadboard whose casing has been removed:



3. The summary of the electrical connections is as follows:

- **a.** The top two rows and bottom two rows are called the "rails." Within a row, all the holes are connected to each other, except that there is a gap across the very middle of the board. Usually a small jumper wire connects the rails across the gap, making each row one long rail. The rails are used for the positive voltage supply (top rail) and ground (bottom rail). Of course, there are two top rails and two bottom rails, but you'll only need to use one of each.
- **b.** Other than the rails, the board has many columns of 5 holes which have a common (vertical) electrical connection. The columns are not connected across the gap which runs across the center of the board, which makes the gap a useful place to put, say, a transistor: you can put the drain above the gap and the source below the gap in the same column, and they won't be shorted to each other. (The gate would have to go in a different column. By convention, you'd use a column to the left, so as to make the actual placement of the transistor resemble the way a transistor is represented in a circuit diagram.)

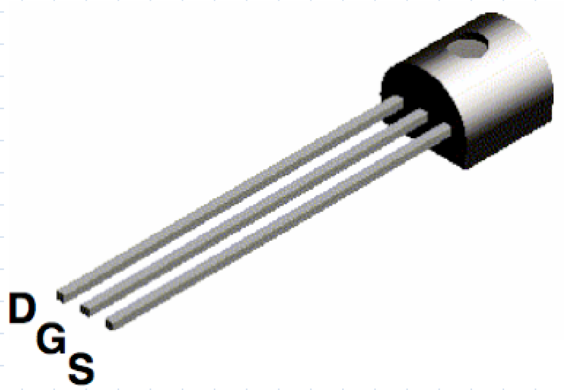
4. Here is a schematic diagram indicating the connections on a breadboard:



Each black square represents a hole where you can place a wire; each rectangle represents all the holes which are connected inside the board.

C. Transistors

1. The type of transistor we will be using in this lab is the field-effect transistor, or FET. It is an electronic component with three terminals: the gate (G), source (S), and drain (D).



The transistor casing is flat on one side and rounded on the other, so that the three terminals can be distinguished. The gate is always the middle pin, but you'll need to know how the drain and source are oriented in order to put the transistor into your circuit correctly.

- **2.** The transistors we'll be using in the lab are n-channel FETs. The "n" means the current is carried by negatively-charged electrons (just like in normal wires), which flow from the source to the drain. (In p-channel transistors, the current carriers are actually positively charged, through a minor miracle of semiconductor physics.) So the direction of the **current** is actually from drain to source. This means that the transistor operates correctly when the drain is at a higher voltage than the source. So a "right-side up" transistor (assuming that the +5V is supplied from the top rail and ground is at the bottom) has the flat side facing to the right and the rounded side facing to the left.

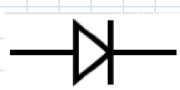
D. Resistors

E. Jumper wires

- **1.** You'll have a selection of short wires which are stripped at both ends, which you can use to make extra connections on your breadboard.

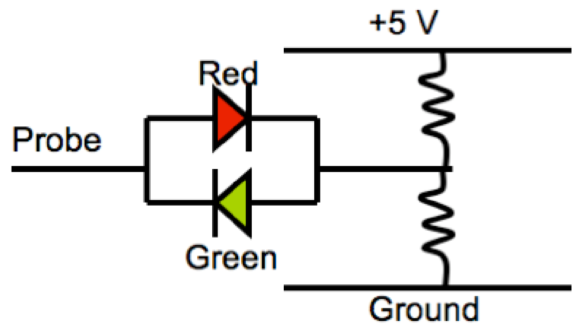
F. Digital logic probe

- **1.** This is a package consisting of two LEDs (light-emitting diodes) on a long wire. It is used to probe the digital voltage in a circuit. One end of the wire should already be plugged into the far right of your breadboard, between two resistors; it should stay there. The other end (the leg of the LED itself) is the terminal you'll use to probe your circuit.
- **2.** If you plug the LED end into a place on your breadboard which is at high voltage (+5 V), the LED lights up red. If you plug it into a place which is at ground, the LED lights up green. Mnemonic: red is generally used to refer to the positive voltage in a circuit.
- **3.** If you plug the diode in somewhere and neither LED lights up, it means that it's not connected to either a high voltage or ground. Perhaps there is a break in your circuit somewhere?
- **4.** How it works
  - **a.** A **diode** is a circuit element that only wants to pass current in one direction. The light-emitting diodes that are in the plastic casing (there are actually two of them in there) light if the voltage drop in the direction the diode is facing exceeds about 0.7 volts. There is no light if the voltage is less than that, or if the voltage drop is the opposite way from the diode orientation.
  - **b.** The circuit symbol for a diode is:



The arrow points in the direction that the diode will allow current.

  - **c.** The plastic casing contains a red LED and a green LED which are wired in *anti-parallel*; that is, in parallel, except the diodes are pointing in opposite directions. The resistors on the right-hand side of the breadboard form a voltage divider (as you have seen on HW 3), so that the middle point between is at +2.5 volts:



- **d.** If the probe end is connected to a place which is at +5 V, then current will flow through the red diode and it will light up, because its voltage drop of 2.5 V exceeds the 0.7 V minimum needed to light the diode. However, the green diode will not light because it is **reverse-biased**; the voltage drop is in the wrong direction. (In the diagram, the current wants to flow from right to left through the green LED, but the potential is higher on the left than the right, so the diode will not allow current to flow.)
- **e.** If the probe end is connected to ground, the opposite occurs: the green LED lights and the red does not, because it is reverse-biased.
- **f.** The brightness of the light depends on how much current flows through the diode, which in turn depends on the values of the resistors in the voltage divider. The smaller the resistances are, the more current flows, and the brighter the light. However, drawing a lot of current to light the diodes affects the circuit you are probing!

G. Vernier voltage probe

- **1.** This is just a pair of clip leads that can be used to measure the voltage in the circuit. The black lead must always remain clipped to ground. The voltage at the red lead is then monitored by Logger Pro.
- **2.** You can use the voltage probe instead of the digital logic probe to test your circuits, or you can use both in conjunction.

H. XOR gate

- **1.** On a separate breadboard, you'll be given an XOR gate which is actually made out of an integrated circuit, or IC (the black chip in the middle of the breadboard). It would require quite a few transistors to make an XOR yourself, but the chip actually contains four separate implementations of XOR inside it. You'll only need to use one. The green and yellow leads connected to the two leftmost pins on the bottom row are the inputs; the next pin over (3rd one on the bottom) is the output.
- **2.** There are two other pins which are already connected; they are the power supply and ground for the chip. Don't alter these connections.

V. Procedure

A. Tell us who you are!

- **1.** Take a picture of yourselves using Photo Booth and paste it here:
- **2.** Tell us your names:
  - **a.**
  - **b.**
  - **c.**

B. Verify the truth table of an XOR gate

- **1.** Start by taking the breadboard with the already-assembled XOR gate. Connect the top rail to +5 V and the bottom rail to ground of the power supply.
- **2.** Start by connecting both inputs of the XOR (the yellow and green wires) to ground.
- **3.** Open the file Lab4.cmbl in Logger Pro. Connect the black probe of the voltmeter to ground using a short jumper

wire. Connect the red probe to ground also and zero the voltmeter by clicking on the Zero button.

4. Now connect the red probe to the output of the XOR (the white wire on the 3rd pin). You can either collect data continuously for 60 seconds, or just look at the meter display in the lower left of the Logger Pro window.

5. What voltage do you read on the XOR output? Is this "high" or "low"?

6. Fill in the top row of the XOR truth table in the Logger Pro file with your result.

7. Locate your LED logic probe and test it by plugging the LED end directly into the +5 V supply and then directly into ground. Does it light up with the appropriate colors?

8. Now take the LED probe and connect it to the output also. Which color lights up? Does this agree with your voltage measurement (high or low)?

9.

Click the Collect button and watch what happens to the output voltage when you change the inputs from 0 to +5 V, one at a time. Fill in the rest of the truth table in the Logger Pro file by changing the inputs A and B (the green and yellow wires). When you are done, paste a copy of the table below:

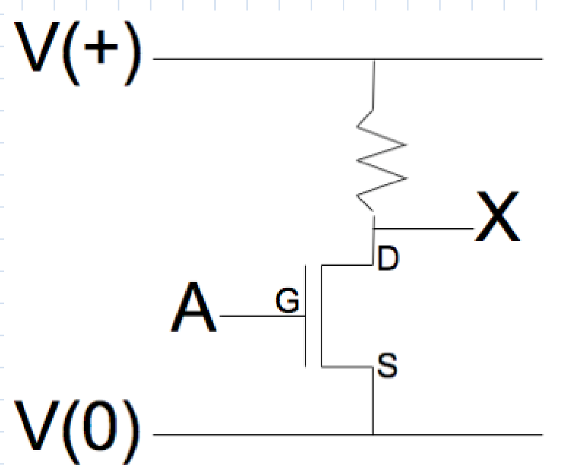
10. Choose a set of inputs that makes the output high (+5 V). Connect the voltmeter but not the LED to the output

and click Collect. Now notice what happens to the voltage reading when you plug in the LED probe at the output. How much does it change? Does this affect whether it is considered high or low?

11. Why do you think the voltage changes when you use the LED probe? (Hint: a "good" logic probe would draw almost zero current in making a measurement.)

C. Build a NOT gate out of a transistor

1. Now take the other breadboard (the empty one) and connect it to the power supply. Connect a transistor with a pull-up resistor to make a NOT gate. As a reminder, here is the circuit diagram:



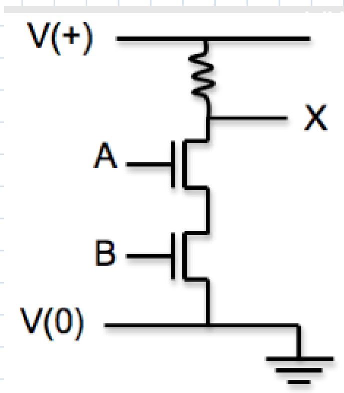
2. You should use a 100 Ω resistor as the "pull-up" resistor between the transistor drain and +5 V. The reason for the pull-up resistor is that a transistor can not handle more than about 100-200 mA of current from drain to source. The transistor itself provides a drain-source resistance of only a few ohms when it is on. So if there is no resistor in series with the drain-to-source path, the current will be too high and will destroy the transistor. A fried transistor is not the end of the world—they're very cheap—but you might not notice that it is broken and be unable to figure out why your circuit isn't working.

- **3.** When you think you have wired up the circuit correctly, try testing the output (X) for the two different values of the input (A). You can use either the voltmeter or the LED probe for this testing.
- **4.** Fill in the two rows of the NOT truth table in the Logger Pro file and paste it below:

- **5.** When everything is working, talk to your TF and show him or her your circuit and how it responds to the two different inputs. Then move on to the next part. By the way, you shouldn't take out the NOT circuit you've built; you'll use it again later in the lab.

D. Build a NAND gate out of several transistors

- **1.** Now, on another section of that breadboard, combine transistors (and pull-up resistors) to implement a NAND gate. NAND is a two-input operator whose output is 0 when both inputs are 1. Otherwise the output is 1. Prof. Whitesides gave this circuit in the lecture for X = A NAND B, or equivalently, X = NOT (A AND B):



- **2.** Explain in words why X is 0 when A and B are both 1, but 1 otherwise:

- **3.** Build a NAND on your breadboard and test it for all 4 possible input combinations, using either the logic probe or the voltmeter. Fill in the NAND truth table in the Logger Pro file and paste it below:

- **4.** Does this truth table match the definition of NAND?

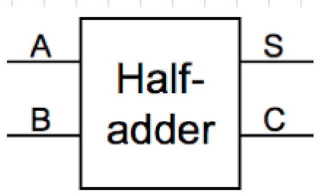
E. Combine NAND with NOT to form AND

- **1.** Now that you have both NOT and NAND, combine them to make AND, and test it in the same way.
- **2.** Fill in the truth table for AND in the Logger Pro file and paste it below:

- **3.** Talk to your TF and show him or her your AND gate. Clearly indicate where the inputs are and where the output is measured. Then you can move on to the last part.

F. Show that your circuits can be used to do binary addition

- **1.** In his lecture, Prof. Whitesides demonstrated how to perform addition on two binary numbers A and B, giving a Sum bit and a Carry bit. A gate which implements binary addition for single-digit binary inputs is called a **half-adder***:



*: "Why is this called a half-adder, and what's a full adder?" A half-adder is fine for adding two single-digit binary numbers. But if you

want to add two n-digit binary numbers, you need more than n of these half-adders, because there is also potentially a carry-in from the previous digit. So a full adder is a circuit with 3 inputs (A, B, and C_{in}) and 2 outputs (S and C_{out}), which adds all three inputs to produce a sum and a carry out. To add two n-digit binary numbers, then, you need n full adders... or actually, n-1 full adders and a half adder, to be minimalist about it. Each C_{out} then becomes the C_{in} of the next digit.

2. A half-adder has two inputs (A and B), and two outputs (S and C, for sum and carry). Given the circuits you have in front of you, explain how you would implement a half-adder.

3. Essentially all operations performed in a computer boil down to simple logic gates and, in particular, addition. So if all you did was make millions of copies of the circuits you've already made, you could, in principle, make a computer. (If the idea of putting together a computer from scratch appeals to you, be sure to check out Physics 123, Laboratory Electronics, next year.)

G. Design your own circuit

1. Now that you have built NOT, NAND, AND, and XOR, see if you can combine two or more of these gates to implement **either** of the following new truth tables:

	=		
	A	B	Output
1	0	0	1
2	0	1	0
3	1	0	0
4	1	1	1
5			

	>=		
	A	B	Output
1	0	0	1
2	0	1	0
3	1	0	1
4	1	1	1
5			

Both of these new operators take two inputs and produce one output, and in both cases, they compare the two outputs. The one on the left, **=** (or EQUALS), is true (1) when A and B are equal to each other, and false (0) when they are different. The one on the right, **>=**, is true when A is greater than or equal to B.

2. Choose which of the two you will build, and then try to actually build it. (Hint: you can get a lot of mileage out of your existing circuits, so don't tear them all out before you start!) If you are having trouble getting started, see if you can first write "A = B" or "A >= B" in terms of NOTs, NANDs, ANDs, and XORs of A and B.

a. Which table?

b. Describe your implementation of this logical operator (what circuits you used, how you connected inputs and outputs, etc.). If you would like to, take a photo of your breadboard(s), or of a circuit diagram describing your breadboard, and paste it below.

c. Test your circuit for all four input values (each row of the truth table). Does it match the truth table above?

VI. Conclusion

A. Clean up

1. Before leaving, please disconnect everything on your original (empty) breadboard **except**:

a. the short orange and gray jumper wires that connect the rails together across the gap in the middle of the board

b. the resistors forming a voltage divider at the far right of the board

c. the end of LED probe which is plugged into that voltage divider (you can leave the probe end dangling)

2. Return the XOR board to its original state (with the IC chip and all connections to it still there, and the LED probe circuit on the right side of the board). If you can't remember what it was like at the start of the lab period, ask your TF.

B. Be sure to export your lab report to the server in the usual way.

C. That's all for this week! Good luck on the midterm and we'll see you in the next lab after spring break.