ASSIGNMENT 2: Introduction to Electricity and Digital Logic
out: Mon Sep 8

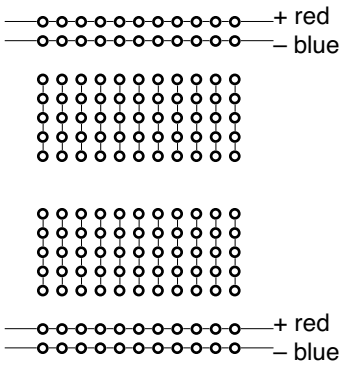
due at the start of class, Mon Sept 15.

For each section of the assignment, the work that you are supposed to turn in is *indicated in itali* the end of each problem or sub-problem. This result may be a drawing/schematic, a written answer, or an equation, or a combination of all three. I prefer that schematics are drawn neatly b hand (this is foryour benefit—it's quicker than using a draw program on a computer). Please turn hard copy.

2-1: Basic Prototyping and the 74HC00 Quad NAND Gate.

In this exercise, you will learn basic prototyping skills using your UML305DEV board and the solderless breadboard.

The figure below illustrates a section of the solderless breadboard:

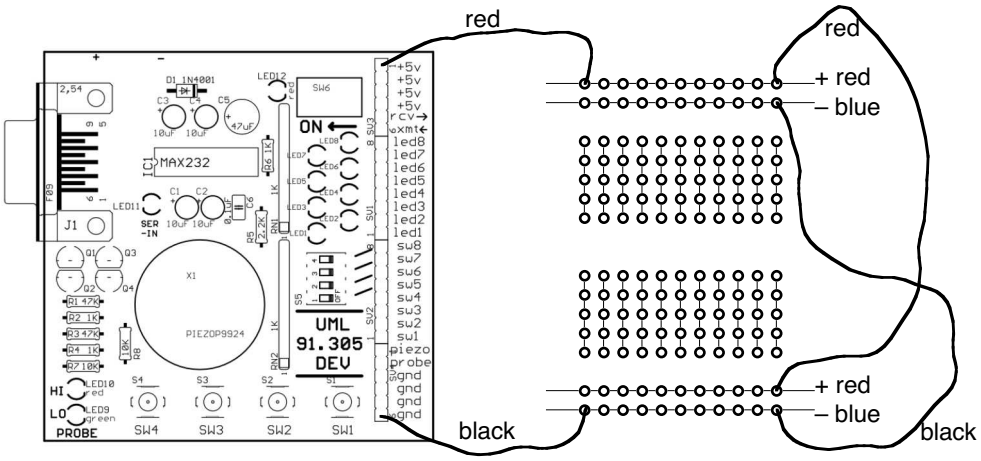


Each group of holes that is connected (in the drawing) with a thin line is connected electrically or board. Thus, there are two sets of horizontal "busses," along the top and bottom, and vertical groups of five holes. The horizontal busses are used for power and ground distribution, and the vertical holes are used to install and connect parts.

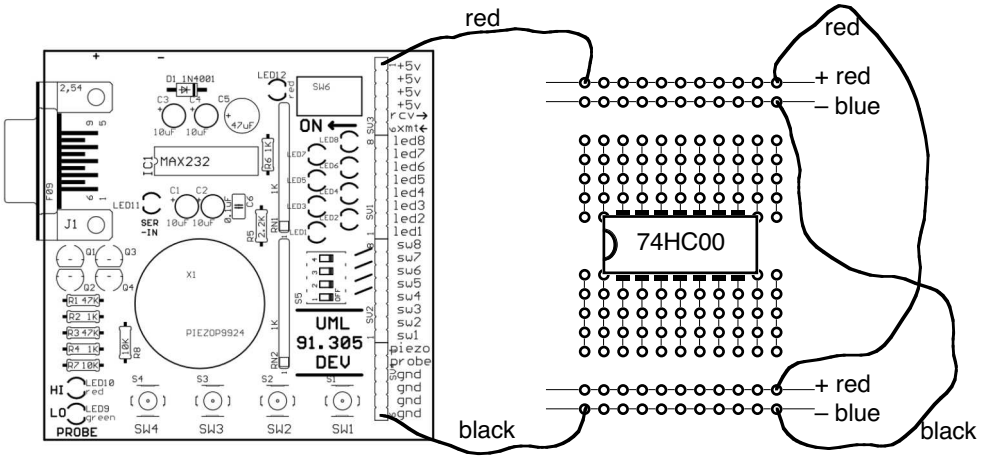
The busses are printed in red and blue, and labeled + and –. Start off by connecting power and ground from the UML305DEV board to one of your proto boards. Get a length of red wire, strip about 1/3 of an inch of insulation from both ends. Then run one end into one of the four +5v sockets on the UML305DEV board. Connect the other end to one of the red bus strips. Get a second red wire, and connect from one red bus to the other red bus.

Using a black wire, also connect ground from the UML305DEV board to one of the blue bus strips. Get a second black wire, and connect from the blue bus to the other blue bus.

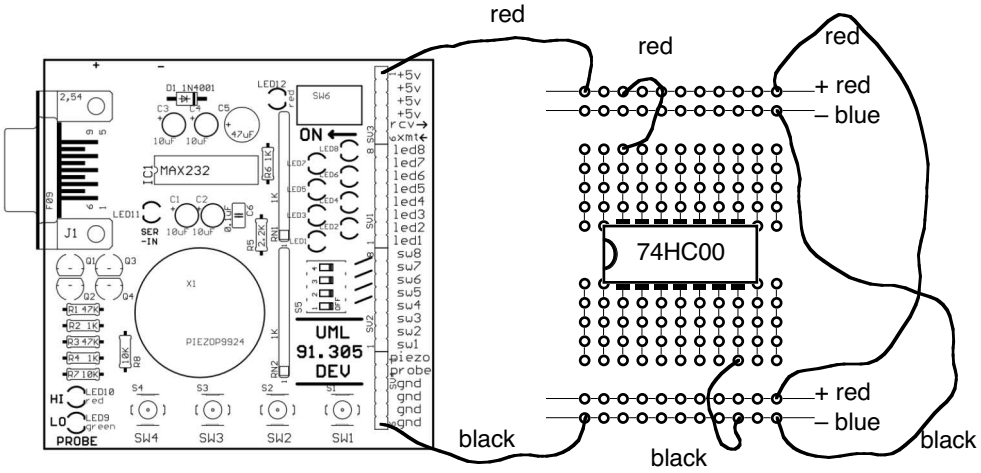
Your set up should now look like:



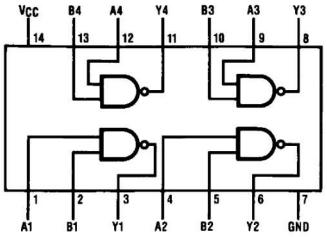
Now you're ready to install a chip and wire it up. Get the 74HC00 quad NAND gate chip, and plug it into the breadboard, straddling the vertical banks of pins. Arrange the chip so that the not is to the left, putting pin 1 in the lower left corner:



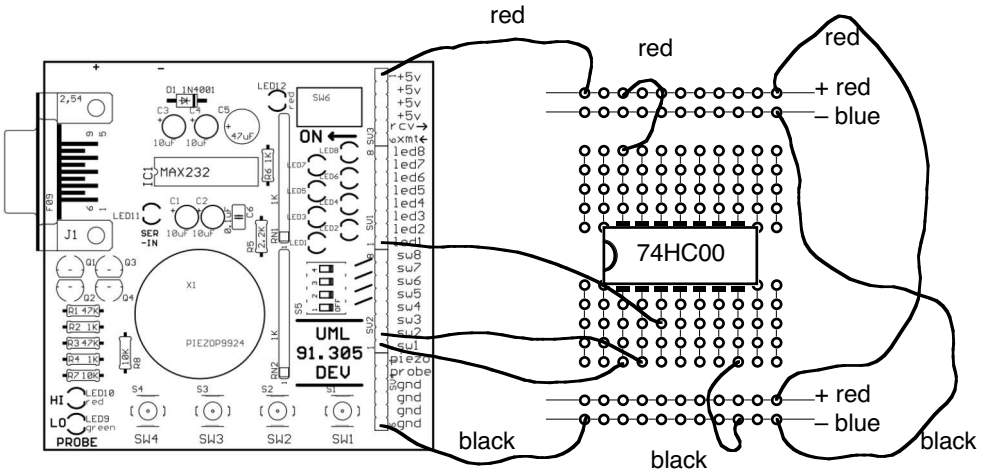
Now, connect power and ground to the chip by running wires to from the chip to the power busses. Pin 14 is the power pin; use a red wire to connect it to the red power bus. Pin 7 is the ground pin; use a black wire to connect it to the blue ground bus:



Now, let's connect to an actual gate from the chip. Looking at the data sheet, one can see that t gates are connected as illustrated:



We'll connect the #1 gate. The inputs A1 and B1 are at pins 1 and 2, and the output Y1 is at pin 3. Using white wires, connect the inputs to pushbuttons SW1 and SW2, and the output to LED1. The circuit should now look like:



At this point, turn on the UML305DEV board. Make sure the red power LED next to the power switch is on.

The pushbuttons generate logic low (0v) signals when they are not pressed. So, the input to the NAND gate is 0 0, and its output should be logic high (5v). Thus, LED1 should now be lit.

Test the NAND function: when both inputs are true (buttons pressed), the output goes low (LED off).

To turn in: nothing to turn in.

2-2: NAND as Inverter.

Design a circuit that allows the NAND gate to be used as an inverter. The circuit should have on input and one output. Use SW4 as the input and LED2 as the output. Use a different gate on th NAND package than the one you wired in Exercise 1. Build and test the design.

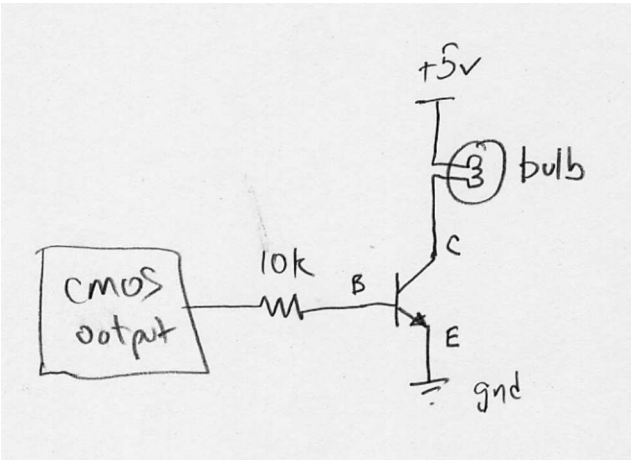
To turn in: Draw a schematic of your resulting circuit, including the pushbutton switch, output LE and power/ground connections.

2-3: Transistor/Lamp Circuit.

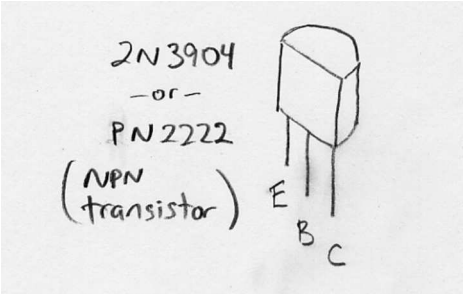
CMOS outputs can source between 5 and 20 mA of current. The small lamp in your kit requires about 50 mA of current to light. Therefore, CMOS outputs can not properly drive the lamp.

The NPN transistor in your kit is an ideal device to use to provide drive current for the lamp.

The schematic below shows how this works. The CMOS output is connected to the base of the transistor (B) through a 10k resistor. When the CMOS output is high, a small current flows between the base and the emitter (E). This small current causes a large current to flow through collector (C) - emitter (E) junction, turning on the lamp.



The diagram below shows how the three transistor signals are attached to the physical device (known as the TO-92 package).



Using this information, wire your NAND gate to control the lamp.

Before building,try wiring the lamp directly to the NAND output (from the output to ground). Does it light up at all when the output is high?

Draw a circuit diagram of the final configuration, including power and ground connections. When have the circuit work, answer: how much brighter does the lamp seem when running through th transistor?

2-4: Counters.

With the assistance of the data sheet, hook up the 74HC393 dual 4-bit binary counter. Use a pushbutton to generate the clock input; when it's working, you should see it increment one count per button press.

Draw a circuit diagram of the chip and how you have attached it so that it will count.
Does it count on the rising edge or falling edge of the clock signal?

Using the chips in your kit, design a circuit that will generate a carry from the first counter stage t as the clock input of the second, creating a full 8-bit counter. Do the counters increment on the r edge or the falling edge of the clock signal? Make sure to think this through, so that the second counter increments on the same edge as the first.

Draw a circuit diagram of your 8-bit counter.

Typically, counters are built with chains of flip-flops. Using your 74HC73 dual JK flip-flop chip, build a two-bit counter.

Draw a circuit diagram of your 2-bit counter.

2-5: Mystery Chips.

Synopsis. You are given two chips from the 74HCxxx series. Without destroying them in the process, identifity them.

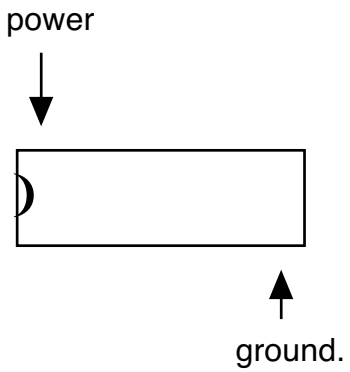
Process. The method is more important than the result. In other words, you must document your thought process and experimental method—the way that you go about finding the solutions. You must turn in a detailed description of the steps you take with each chip that led you to your conclusion.

About one full page of single-spaced, printed output is expected (per chip).

Just turning in the answer, e.g., "The 14-pin chip is the 74HC00" is not acceptable. (Note: this isn't the correct answer, so you've just had one possibility eliminated.)

Method. It is important to avoid burning out the chip during testing. This means you must not wire chipoutputs to power or ground. So you must first determine which pins are input and which are output.

First of all, make sure you wire the chip to power and ground properly. Holding the chip like this:



The +5v power pin is on the upper left, and the ground pin is on the lower right.

To determine if a given pin is an input pin, wire the pin to +5v **using a 1k resistor**. Then, measure the signal at the pin using your logic probe. If it's an input, it should be high. Now, connect the pin to ground with the resistor. Measure it again. If it's now low, then it is an input. **If the pin ever disagrees with the value you're asserting with the resistor, then it is an output.**

Go around all of the pins and determine which are input and which are output. Once you know a pin is an input, you may wire it high or low with a wire (the resistor isn't needed). **But be sure to not wire outputs to +5 or ground with a wire!** If the output is trying to generate a signal other the one you've wired, the chip will get hot and may burn up.

Connect the outputs to your LED indicators. Now vary the inputs, and figure out what the chip is doing. You may want so systematically record all possible inputs and what outputs are generated, or explore and try to figure it out differently.

Hints. The two chips (one 14-pin, one 16-pin) are both members of the 74HCxx or 74HCxxx series. Both are pure combinational logic—outputs are directly a function of inputs, with no internal state (no flip-flops or latches).