# Lab 03 – The Decoder

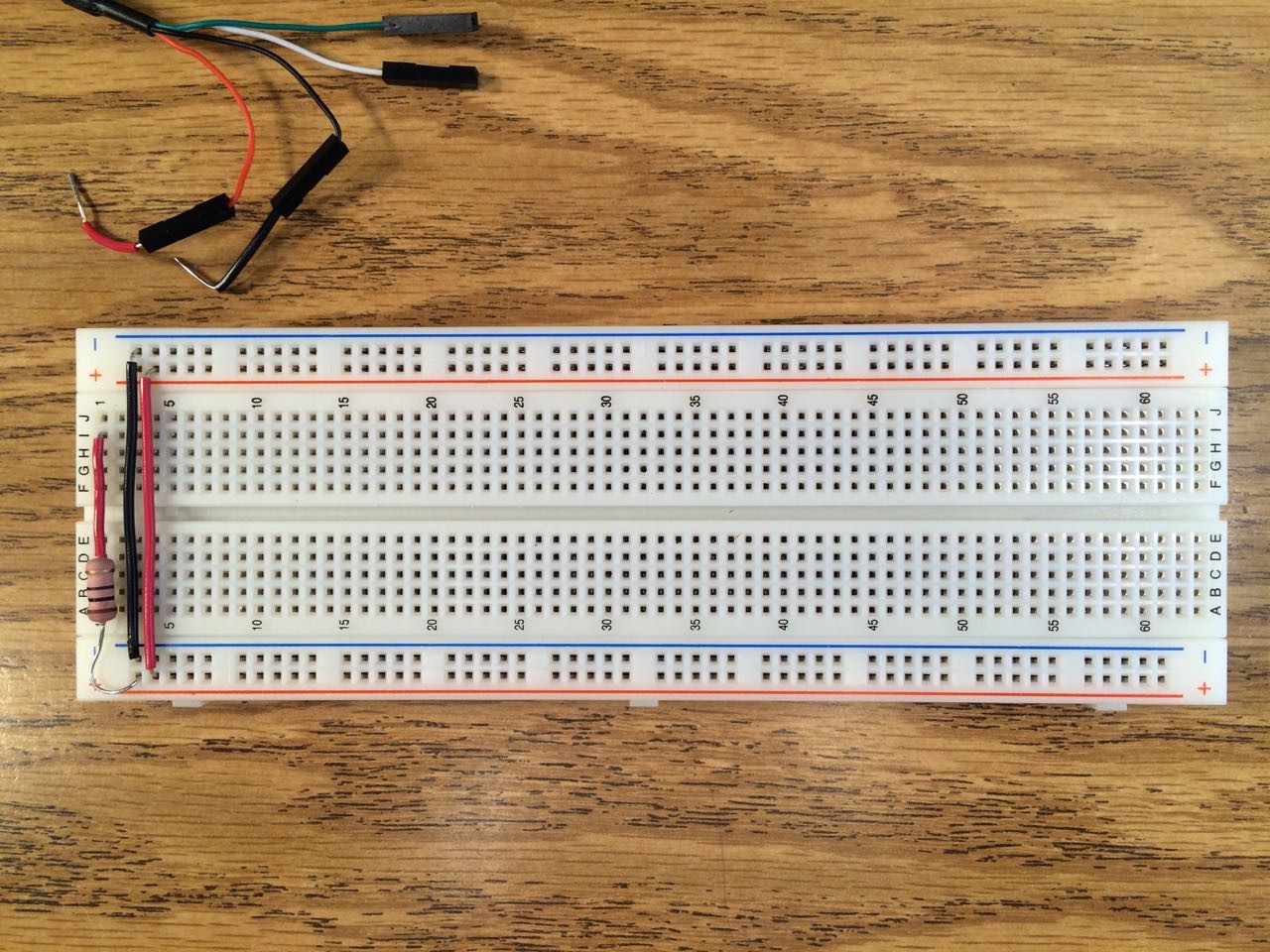
## Start construction for this assignment **during** Week 03 (Jan. 23-26) after your Lab 02 take-home breadboard circuit has been scored.

## Your completed decoder circuit and question responses are due at the start of your lab session in Week 04 (Jan. 30 – Feb. 02).

In this lab, you will build a circuit to implement a 2-to-4 Line Decoder. In general, a decoder is any circuit that translates a code into a set of (decoded) signals. A line decoder, often just called a decoder, takes an n-digit unsigned binary number, the “code” and often thought of as an address, and decodes it by asserting one of 2n signal lines (wires). To “assert” a wire is to make its voltage equal to the value desired for the condition when the signal is considered to be active. Thus, assertion is a logical concept and can be mapped to correspond to the physical condition of either a high voltage or a low voltage on the wire.

## Step 1

Ensure that your breadboard is wired with the 10 Ohm limiting resistor and wires for enabling the red and blue buses along the board edges.



## Step 2 ­– Design the decoder

The following truth table defines the decoder to build. The inputs A1 and A0 can be thought of as a 2-bit address, or unsigned base 2 integer, when read in left to right order, A1 A0. This decoder asserts one of the four outputs, D0, D1, D2, OR D3 for each possible input condition.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input | | Output | | | |
| A1 | A0 | D3 | D2 | D1 | D0 |
| L | L | H | H | H | L |
| L | H | H | H | L | H |
| H | L | H | L | H | H |
| H | H | L | H | H | H |

This truth table is at a lower level of abstraction than a table using logical 0 and 1 values. Instead, the voltage level to appear at various nodes in the circuit is specified as either “L” for low or ground or 0 volts, and as “H” for high or +5 volts.

Design a circuit using only components from your lab kit that will implement a 2-to-4 line decoder.

## Step 3 – Build the decoder and answer questions

Build your circuit design from Step 2 on your breadboard. Make use as needed of the Appendix: Helpful Reference Information section at the end of this lab document. Use either active-high or active-low inputs, but use the same type for both inputs A0 and A1. Show the result of each decoding computation by illuminating one of four green light-emitting diodes (LEDs). Place these diodes on your breadboard in a straight line in the sequence D3, D2, D1, and D0.

On a sheet of notebook paper, write labels that will clearly indicate which pushbutton controls the value of the signal A1, which pushbutton controls A0, what type the two pushbutton inputs are, and which LED indicates which one of the outputs D3, D2, D1, D0. Position to labels on this sheet such that when you set your breadboard on the sheet it will be clear to everyone, by looking down on the breadboard and the sheet, which breadboard component is meant by each label.

## Step 4 – Demonstrate your decoder in lab next week

## Grading:

Your lab grade will be based on your circuit demonstration and written answers to the take-home questions shown to your TA in lab during Week 04.

Your circuit demonstration can earn up to 60 points, comprised as follows:

[5 points] Breadboard circuit includes 10 Ohm resistor correctly protecting against short-circuit faults.

[5 points] Inputs are of matched type (high-active or low-active).

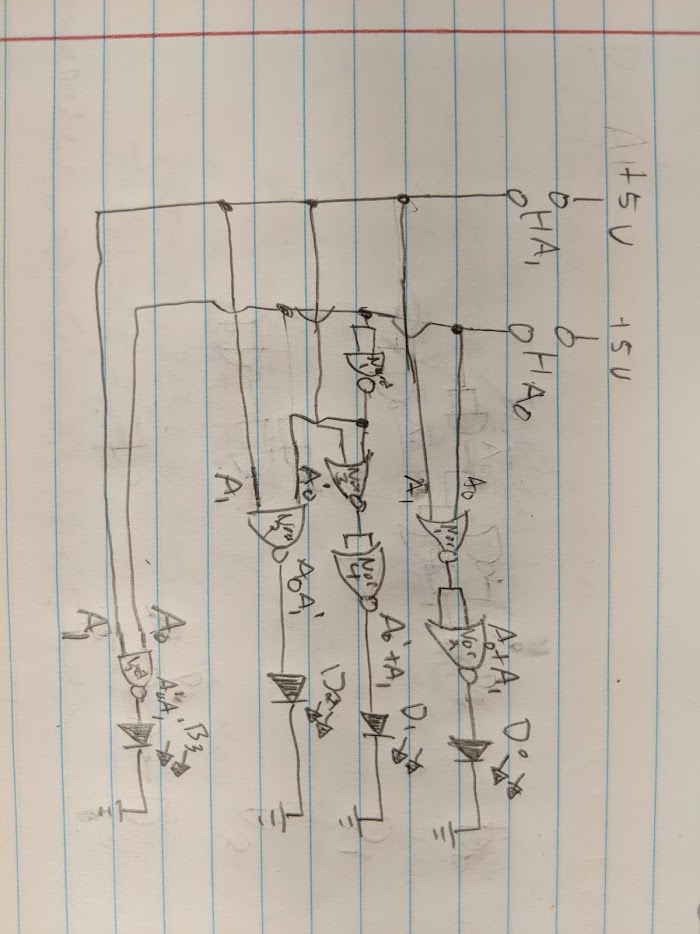
[15 points] The circuit asserts just one of D0, D1, D2, and D3 for any input.

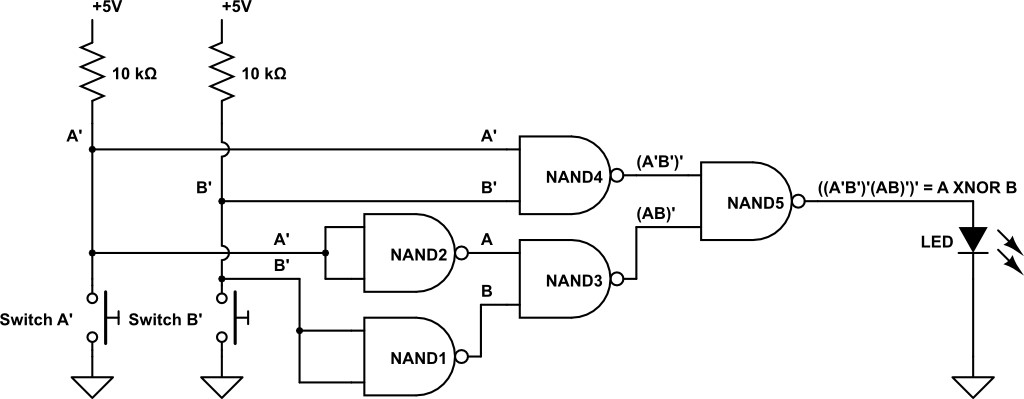
[15 points] Input decoding is consistent with label sheet, that is, each A1 A0 input combination lights the LED labeled as the output to be asserted according to the truth table.

[20 points; 5 points each] Four live-answer questions about your circuit and about decoding will be asked of you by your TA at the time that you demonstrate your circuit.

**Disassemble your decoder circuit in the presence of your TA. Do not remove the 10 Ohm resistor and power and ground bus jumper wires.**

Your written answers to the following questions can earn up to 40 points.

Question 1. [10 points]. Draw the circuit schematic for your design at the level of abstraction shown in the following schematic.



Question 2. [5 points] What is different about what circuit nodes the LEDs for your decoder outputs are connected between compared to what the LED in the above circuit is connected between? **The LEDs in my decoder are each connected to different outputs whereas the LED in the above circuit is to display all outputs.**

Question 3. [5 points] How many minterms are implemented in hardware for the 2-to-4 line decoder circuit? **4**

Question 4. [5 points] How many variables comprise each minterm of the 2-to-4 line decoder? **2**

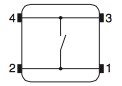
Question 5. [5 points] How many inputs are there to each gate connected to an output signal wire for an n-to-2n line decoder? **2**

Question 6. [10 points]. Give a formula for the number of CMOS transistors required to build, in a manner analogous to the 2-input NAND gate of Figure 2.5 in the textbook, an n-to-2n line decoder.

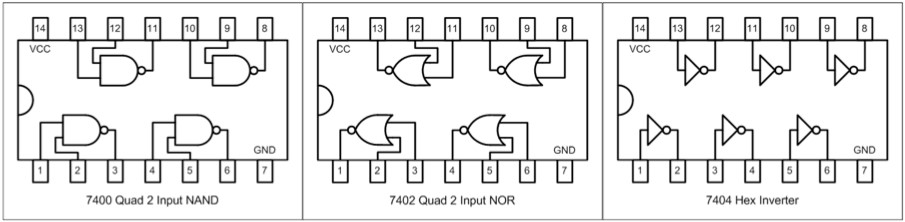
## Appendix: Helpful Reference Information



A pushbutton switch schematic that looks a bit like the actual switch is this. Rotate the schematic 30 degrees counterclockwise if you would like to align it with the image above.



The DIP packages containing NAND gates (also known as (a.k.a.) chips) are the ones labeled SN74HCT00N, which we can call 7400 for simplicity. Here are pin diagrams for three chips in the 7400 series included in your lab kit, the 7400 Quad 2-input NAND, the 7402 Quad 2-input NOR, and the 7404 Quad Inverter. These diagrams show you which pins on the chips correspond to which portion of a particular gate (each chip has multiple gates on it). The “notch” on one end of the package identifies the number 1 pin.



Make sure you can read the tiny label print on the top of the chip when referencing the diagram: if the part number on the chip is upside down, then you are holding the chip upside down.

At any point if you want to test your construction, you can use an LED as a visual probe. Think of it as a hardware version of a PRINT statement in a programming language. Just connect the LED anode (longer lead) to the circuit node you wish to interrogate (variable to print out) and connect the other LED lead to breadboard ground. The LED will light when the circuit node is at logic 1 (high voltage) and not light if the circuit node is at low voltage or is not connected to anything (called an open circuit).

**Parts Layout on the Breadboard.** A good starting strategy is to place the main components of your circuit on the breadboard first, particularly switches and DIP devices. Place switches where you can easily reach them. If there will be several, space the switches to give room for your fingers to reach each switch simultaneously. This will allow you easily press the switches in any combination that may be needed. DIP packages are large, which is a good reason to place them on the breadboard next.