Building circuits to do math

Toki Nishikawa

September 20 2022
6-8pm Tuesday lab with Kim Nguyen/Nick Boudreau

1 Design

The objective of this lab is to construct a circuit that compares two 2-bit numbers. The first step is to create a logic equation which can be done using a truth table and its corresponding Karnaugh map as seen in **Figure 1**. M and N are each 2-bit numbers where M_1, M_0, N_1 and N_0 are the variables associated with the individual bits. M_1 and N_1 are the most significant bits. **Figure 1** shows that the logic equation corresponding to $M > N$ is

$$Y = M_1\bar{N}_1 + M_0\bar{N}_1\bar{N}_0 + M_1M_0\bar{N}_0 \tag{1}$$

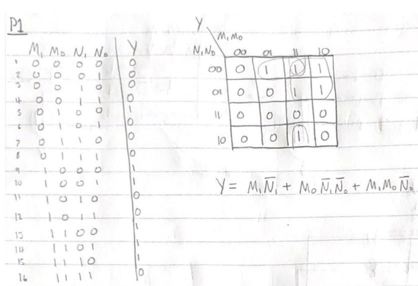

Figure 1: Truth table and Karnaugh map for $M > N$

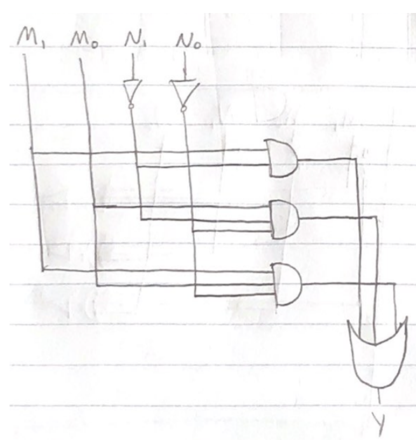

Figure 2: Circuit without constraints

A two-level logic implementation of **Equation 1** is shown in **Figure 2**. However, we will construct our circuit under the constraint that we only have access to the following chips:

- 74HC00 quad NAND
- 74HC04 hex inverter
- 74HC4075 3x3 OR
- 74HC266 quad XNOR

Thus, the circuit we will create is the one shown in **Figure 3**. This circuit will only use one 74HC00 quad NAND, one 74HC04 hex inverter, and one 74HC4075 3x3 OR. In order to control the voltages of our inputs (M_1, M_0, N_1 and N_0), we will use a DIP switch and pull-up resistors. An open switch will cause current to flow from V_{cc} to the desired input pin whereas a closed switch will cause current to flow straight to ground. The output of our circuit will be connected to an LED so that a lit LED indicates a true output. The schematic of the DIP switch and LED implementation are shown in **Figure 4**.

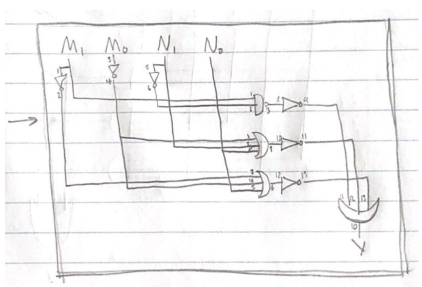

Figure 3: Circuit with constraints

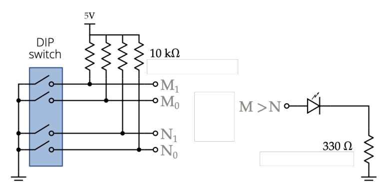

Figure 4: DIP switch and LED implementation

2 Testing

The final circuit is shown in **Figure 5**. It's pretty gross. In order to test its functionality, I tried all 16 combinations of the input variables. These combinations are shown in the truth table in **Figure 1**. The circuit obtained the correct outputs for all of the combinations, and so I'm confident that the implementation is correct.

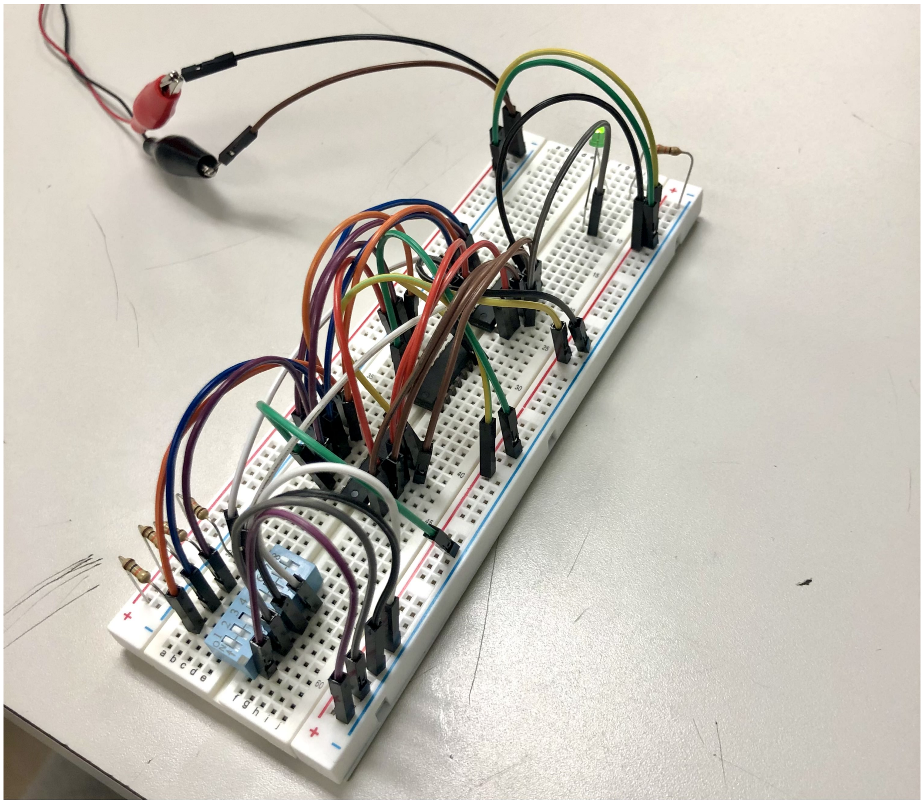

Figure 5: Final circuit on breadboard

3 Lab Journal

Lab 2 Notes

helpful notes

- because we're using pull-up resistors, the logic value will be 0 when the switch is ON.
- I color coded the wires so that each color corresponded to a different variable. this proved to be very helpful.
- the inverters are facing opposite directions on the left and right sides of the inverter chip.
- Sometimes the LED is hard to see, but it's actually working

problems

- I put my NAND chip in between my inverter and OR chips even though the majority of connections were between the inverter and OR chips.
  - ⟹ think about the layout of your circuit BEFOREHAND.
- I spaced out the circuit components quite a bit on the breadboard. it got really hard to connect components that were far away. this wouldn't have been that big of a problem if I had female-female wires, but I did not. I should've placed the components closer together from the start.
- It is very challenging to flip the switches on the DIP switch. I could just have fat thumbs but I theorize that it's a universal problem. the TA suggested I use a pencil. the pencil technique worked flawlessly.
- it is very inconvenient to make short connections with these wires. I want smaller ones.
- I failed to connect my IC's to power and ground despite the warning on the lab manual.
- I forgot to connect my breadboard grounds – I should always do this first.
- I didn't really think to connect the individual IC'S to power + ground. I forgot that they are their own little circuits.
- after I had finished everything, I tried one of the combinations and it failed. turns out I just entered the wrong combination and nothing was wrong with the circuit. I'm glad I didn't just start debugging.

Strategy

I had different checkpoints so that I never did too much work without testing it. this was very helpful since my circuit did not pass my first checkpoint test. this was much easier to debug as opposed to if I had finished the entire circuit. I debugged by checking the voltages at every step in the circuit with the multimeter. I'll start at the last functioning part of the circuit.

4 Reflection

This was my first experience with designing a circuit that implements a logic equation. It is important to understand the translation from logic to a hardware implementation. Sometimes it's difficult to imagine how hardware could represent even the simplest ideas.

I ran into various issues during this project. One of the most frustrating being the layout of the breadboard. When there are many different logic gate ICs, it becomes difficult to make all the necessary connections between circuit elements. It is especially difficult to make these connections when you don't plan the setup of your ICs beforehand. For example, the IC in the middle of the breadboard shown in **Figure 5** is the NAND gate. It may be difficult to see, but there are hardly any connections associated with this gate. The majority of the connections were between the inverter and the OR gate. It would have been a lot more convenient to have these two gates placed next to each other. In future projects, I plan to consider the connections I'll be making between ICs before I begin building the circuit.

I was able to complete the circuit board in around two hours. With that said, it can definitely be done quicker if you don't make as many mistakes as I did.