ECE 385

Spring 2019

Lab 1

Introduction to TTL

Soham Karanjikar
Section ABJ-Friday 2PM
Xinbo Wu, David Zhang

Purpose of Circuit

This lab was intended to show us the basics of TTL logic and how ECE 385 labs generally work. The experiment was to create a 2:1 multiplexer using only NAND gates. We had to go from a general diagram of logic which used NOT, AND, and OR gates and convert it to a circuit which performed the same operation using only NAND gates. We also learned what a static-1 hazard is and how to prevent it by adding additional terms to our SOP. A MUX allows users to pick an output from many different inputs depending on what is necessary. This is very relevant to electrical design as MUX's are very widely used in many different technologies, it is one of the core components.

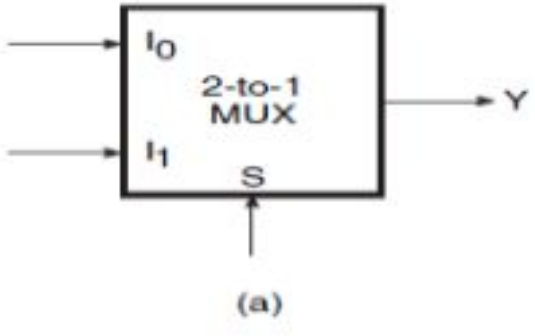


Figure 1.1: 2:1 MUX Block Diagram

Description of Circuit

Since this was a 2:1 MUX I knew that we only had to have 1 select bit as it allows to pick from 2 things. For example if we have 2 inputs A and B, our select S can be set to 1 to select input B
and set to 0 to select input A.

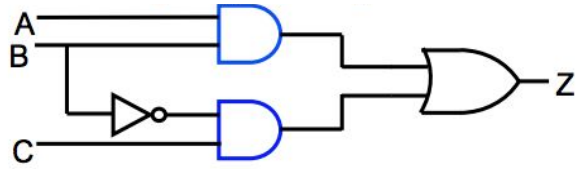
Karnaugh Map

		AB			
		00	01	11	10
S	0	0	0	1	1
	1	0	1	1	0

Truth Table

A	B	S	F(ABC)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1

Figure 2.1 : K-MAP and Truth Table



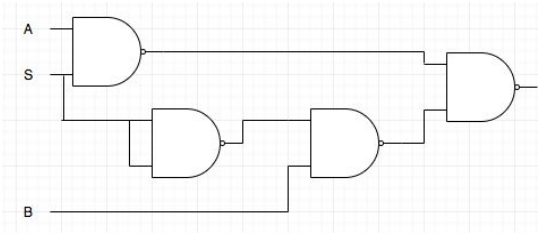


Figure 2.2: Logic Diagram (B is select in first)

We can see this again when we make the K-MAP and truth table for our MUX using the logic implemented. The logic diagram above has a glitch in it, specifically the static-1 hazard. This hazard means that there is a possibility of the circuit creating a 0 pulse output when according to steady state analysis of the logic should produce a 1. This pulse occurs when your inputs are 1 and the select bit changes from a 1 to a 0. As there is a delay from the NOT gate, the input to the OR gate is 0 for a small instance, and this is called a static-1 hazard.

The SOP which has the static-1 hazard is: **Z=S'B + SA.** This is derived from the minterms of the K-MAP. However, to fix this we need to add an additional term **AC** to prevent the glitch; so our final SOP is **Z= S'A + SB + BS.**

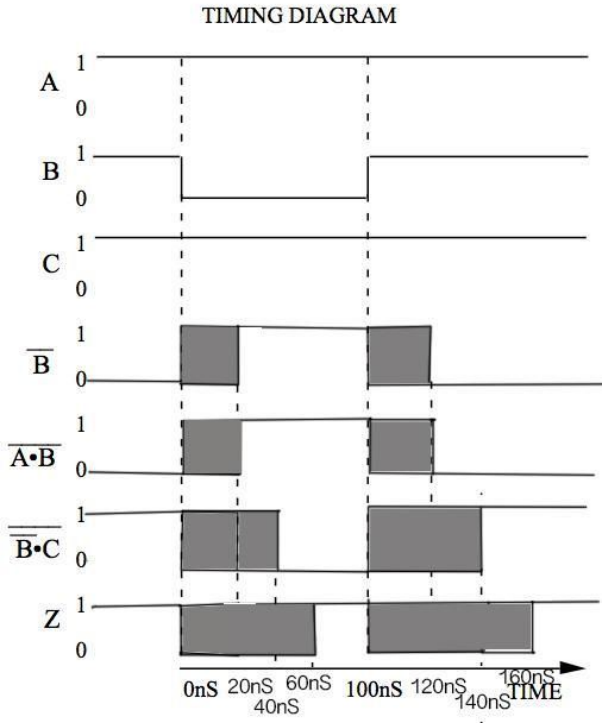


Figure 2.3: Timing Diagram

According to the timing diagram it is visible that it takes at most 60nS to stabilize Z on the falling edge of B and 60nS on the rising edge. The potential glitch occurs in a similar situation described above, when the input B changed from a 1 to a 0 a static-1 hazard is possible.

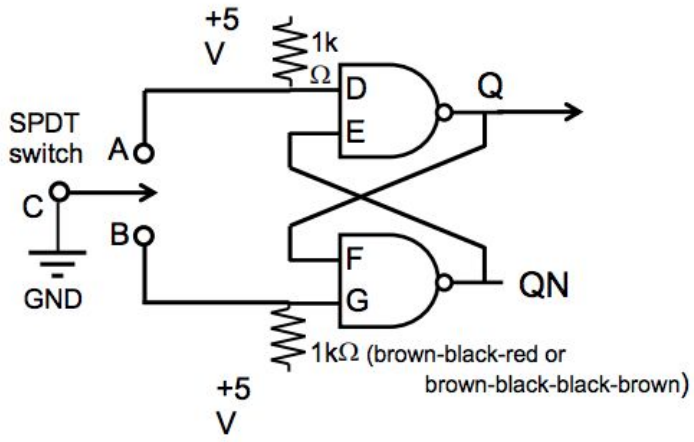


Figure 2.4: SPDT Debounce Switch

The reason we want to use debounce switches is because mechanical switches are usually spring loaded and will have some ill-effects when turned on or off. This occurs due to the terminals hitting each other with force and then bouncing back a little. This circuit prevents any ill-effects that occur when switching, this is because the switch acts like an SR latch. Once you send 0 input to A or B, that input is held constant even if there are bounce effects due to mechanical switching as the output of one NAND gate is connected to the input of the other.

Eliminating the Static-1 Hazard

According to what we learned in lecture, adding extra terms which cover all adjacent 1's in a K-MAP eliminates Static-1 Hazards. This means that we had to add another term **AC**.

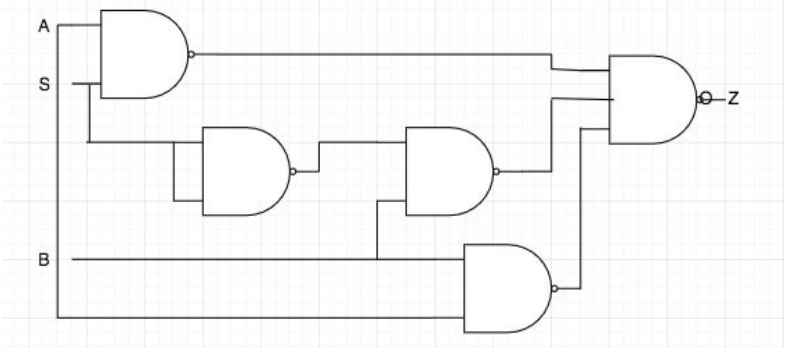


Figure 2.5: 2:1 MUX without Glitch

Lab Results and Oscilloscope Traces

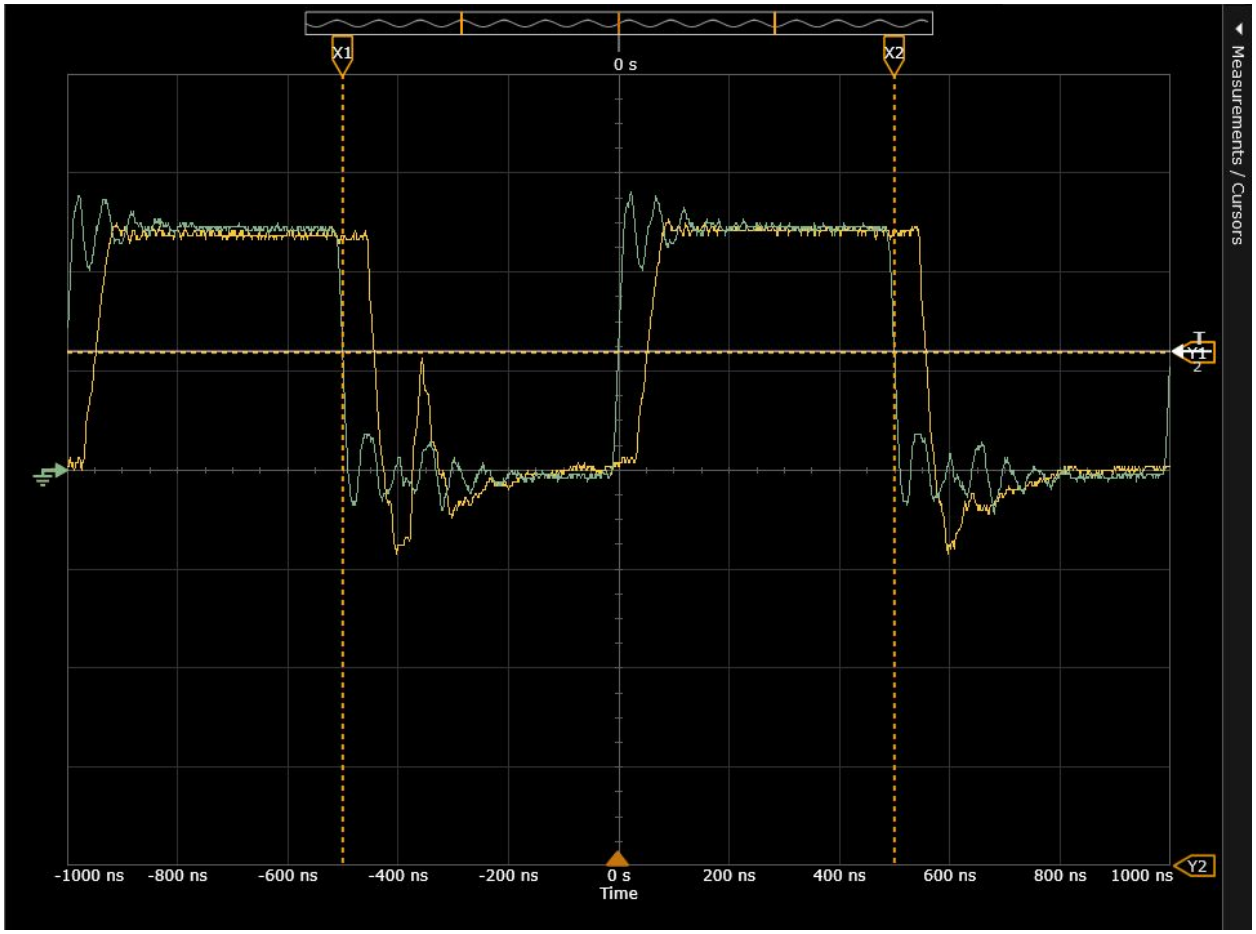


Figure 3.1: Trace with Delay

We did not get a glitch when we first tested our glitched circuit so the TA advised to add more inverters to increase delay, however, we were still unsuccessful in achieving a glitch. The TA checked over this plenty of times and finally told us to use this trace and specify that no glitch occured. TA approved.

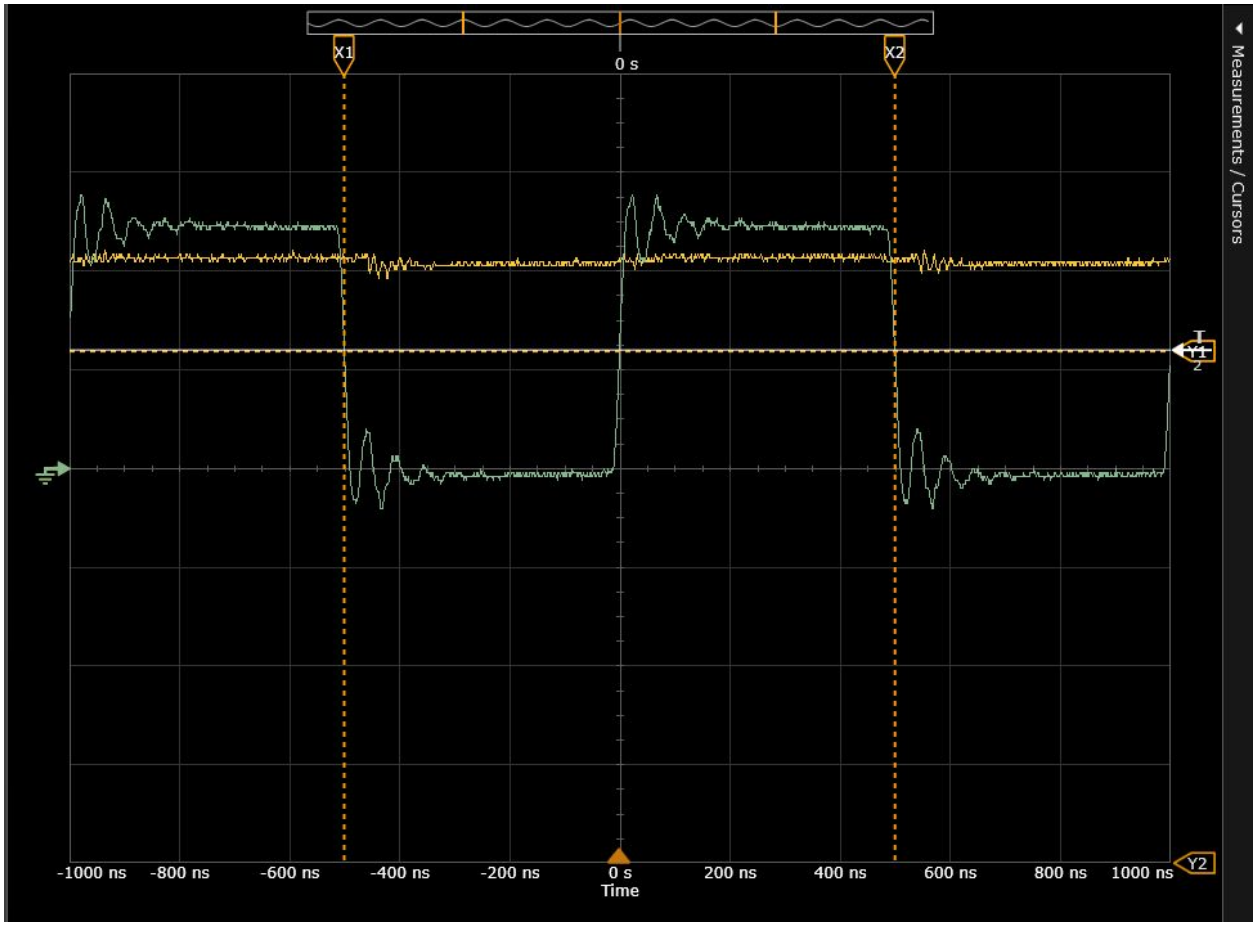


Figure 3.2: Glitchless Trace

After adding the **AC** term it is visible that there is no glitch. Even when going on a rising edge or falling edge of the select signal, the circuit behaves exactly how the steady state analysis showed.

The bouncing of the switches is shown from these traces when the select signal bounces a little after a rising/falling edge.

Circuit Layout

The two circuits described above were implemented as shown in the layout diagram. First only one 7400 chip was used for the circuit with a glitch since only 4 NAND gates were sufficient. To make the glitchless circuit, two 7400 chips and one 7410 chip was used as one more NAND gate and a 3 input NAND gate was necessary.

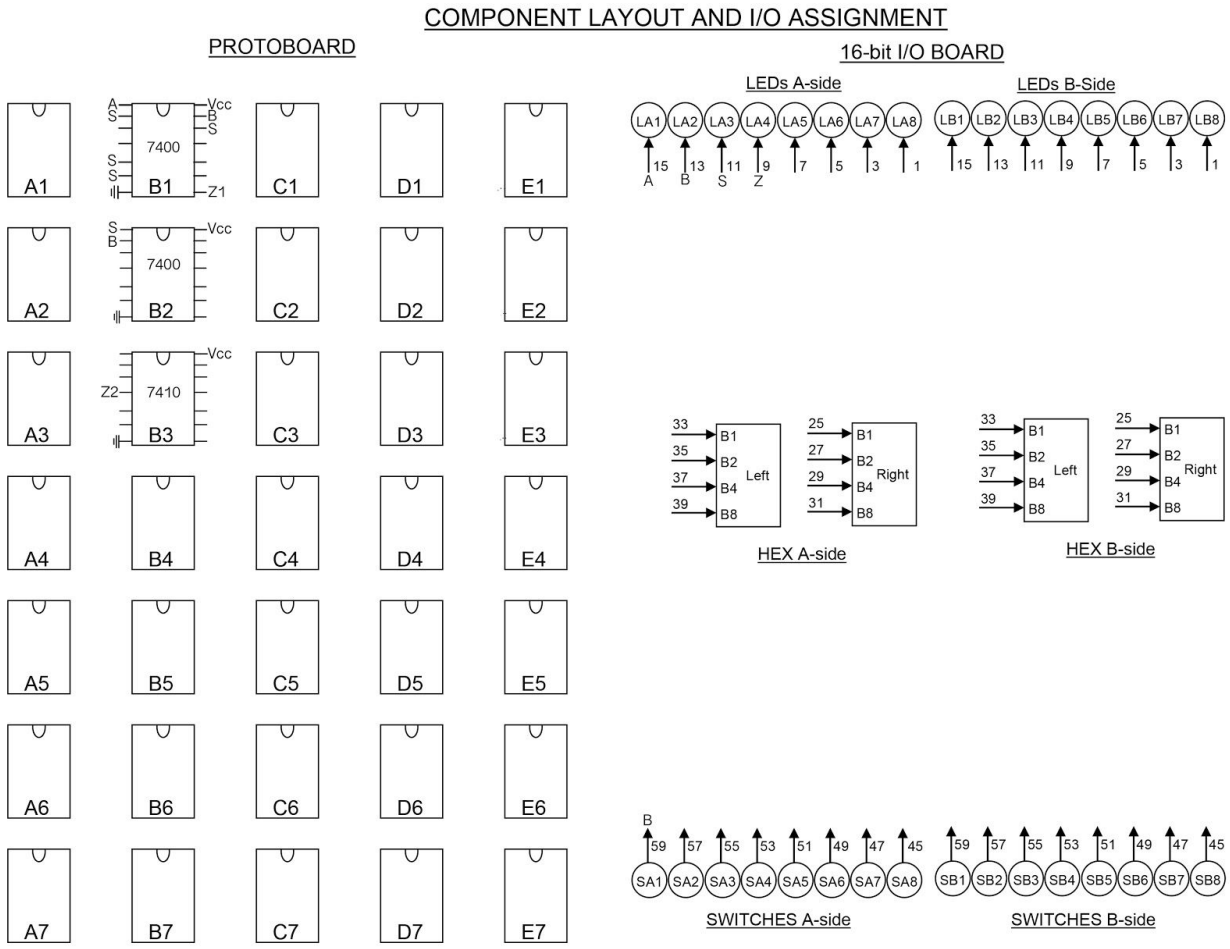


Figure 4.1: Layout Diagram

Questions

Why did some groups not observe the glitch?

Since the 7400 chips are not all identical, some of them are very good at their functionality and their propagation delay is very low so they do not experience any glitches.

GG.6:

Having as much noise immunity as possible in your design is good as any interference can create unwanted results. The last inverter is observed because the first inverter might give you an instant output but linking multiple one causes considerable delay. To calculate noise immunity you can just compare input to the output and see the range of values you get.

GG.29:

It is bad to share resistors when using more than 1 led is bad practice because driving a lot of current through a single LED can cause heat damage to resistor and also not turn on LED's. Also there might be considerable voltage sag, which affects the functionality.

Conclusion

As this was an introductory lab, it showed us the working of ECE 385 and how a lab report should be written and how a lab situation works. We also learned that testing along the way is good practice as having damaged chips is a very common issue. Static Hazards are something that definitely should be taken into consideration because if the circuit does not behave like you expect it to and causes damage to components, it can be very costly. Not only in the lab, but even in industry, taking such things into question when you are working with 3-6Ghz of clock speed, these errors can instantly kill a project. All in all, it was a very informative introductory lab.