ECE 385

Spring 2024

Experiment 1

Lab 1

Yanghonghui Chen / yc47

Section XC/ Friday 5:15-5:30 p.m.

TA: Shixin Chen

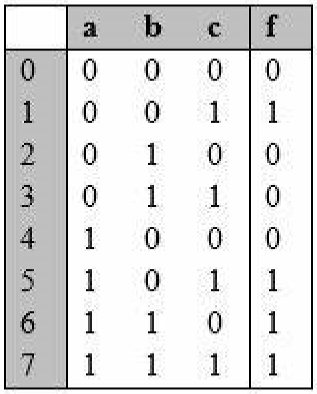
Introduction

This lab is intended to make us familiar with the equipment we will use throughout the class, including the lab kit, oscilloscope, and the function generator. And specifically, we will design a 2 to 1 multiplexor using only quad 2-input NAND gate chips. Additionally, we will explore the glitches in naïve designs, and further redesign circuit to remove possibility of static-1 hazard.

Written Description of Circuit and Lab

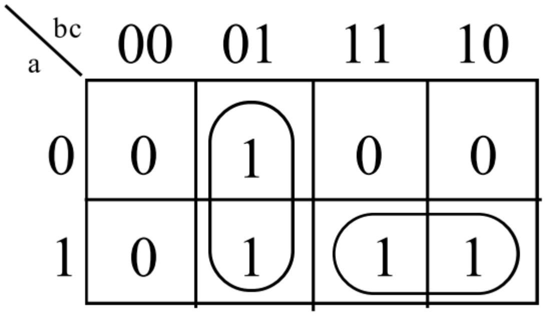
Since we are working on a 2 to 1 multiplexer, we should first derive the logic. MUX has two data inputs, A and C. The control input, B in this case, acts as the switch or selector. Depending on the state of B, either input A or input C is routed to the output Z. In detail, If B is high (1), the MUX selects input A, and the output Z reflects the value of A. On the other hand, if B is low (0), the MUX selects input C, and the output Z reflects the value of C.

1) For part A in the Prelab, here we first list out the truth table shown in Figure 1. Then we use the Karnaugh map shown in Figure 2 to derive the function used to implement the 2 to 1 multiplexer.



	a	b	c	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Figure 1- Truth Table for 2:1 MUX



a \ bc	00	01	11	10
0	0	1	0	0
1	0	1	1	1

Figure 2 - Karnaugh map of 2-to-1 Multiplexer Function

Based on the K-map, we can know that the Boolean function could be written in the SOP form as Z = BA + B'C for part A in the prelab. Then according to the function, a basic logic can be designed shown in Figure 3 with 1 inverter, 2 AND and one OR gates.

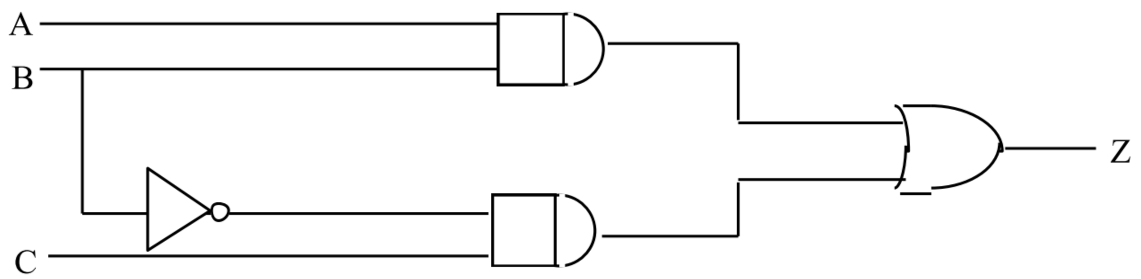


Figure 3 - Sum of Products implemented with 2-level AND-OR Logic

To further simplify the circuit on the breadboard, we can convert all the gates to NAND gates based on De Morgan's Law and practical implementation of chips. In this way, we can simply use a single NAND chip to implement the circuit shown in Figure 4.

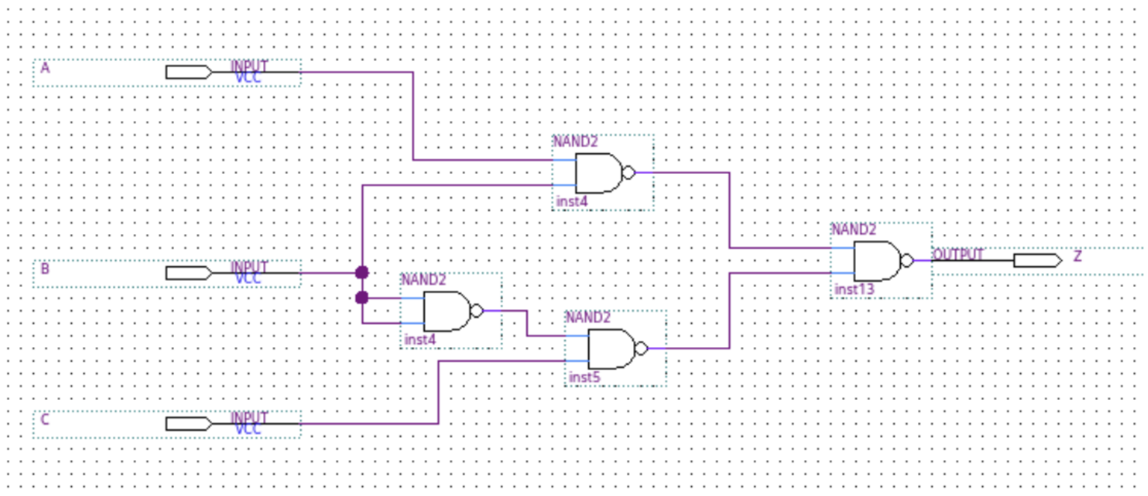


Figure 4 - Circuit Diagram in NAND Gates

2) For Part B in the Prelab, we discover that there is a problem with the circuit in Part A. Due to the additional NAND gate as an inverter in the bottom half circuit in Figure 4, there will be a propagation delay which may lead to the occurrence of glitches in a signal. A glitch is a momentary incorrect output value produced by a circuit during periods when the inputs are changing. For example, when both inputs A and C are held high (1), we change selector B from 1 to 0. For a period, the output will be at logic 0 since the delay of the inverter leads to the temporarily unchanged 0 as input to the OR gate shown in Figure 5.

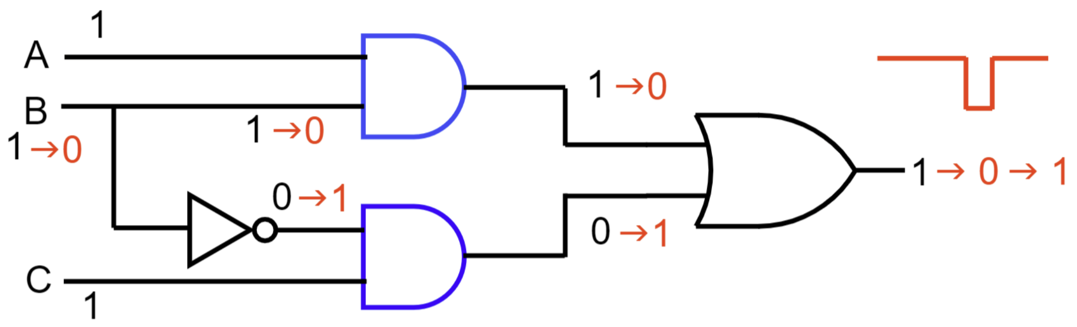


Figure 5 - Static Hazards – Example with 2 to 1 MUX

So here we redesign the circuit of part A to eliminate all static-1 hazards at the output based on the new K-map shown in Figure 6. We cover all adjacent min-terms in the K-map, which means that we just add one more term AC here. Therefore, the new SOP form of the logic expression is Z = BA + B'C + AC. And we redesign the circuit as shown in Figure 7, which uses two NAND chips to implement the logic.

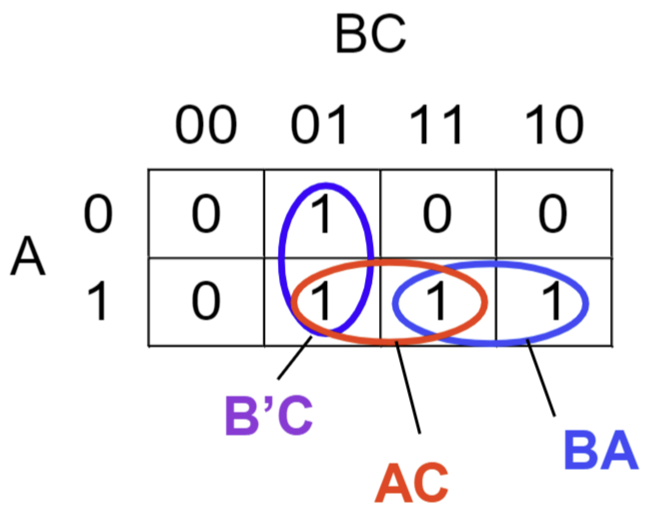


Figure 6 - New Truth Table for Part B

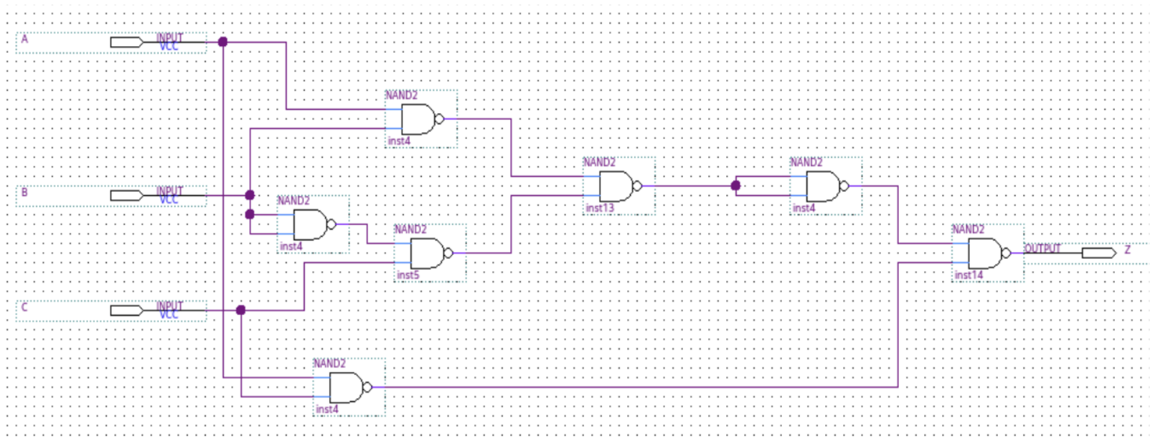


Figure 7- Redesigned Schematic Diagram for Part B

3) In fact, we can observe the difference between the two circuits using the oscilloscope. Here the yellow line in the graph denotes selector B and the purple line represents the output Z. As shown in Figure 8, the output is based on function Z = BA + B'C, which has static hazard. In addition, when B is switched from 1 to 0, the output wavers a couple times before stabilizing due to contact bounce. For our experiment, this cannot be fixed because this is an issue related to the switches used in our lab.

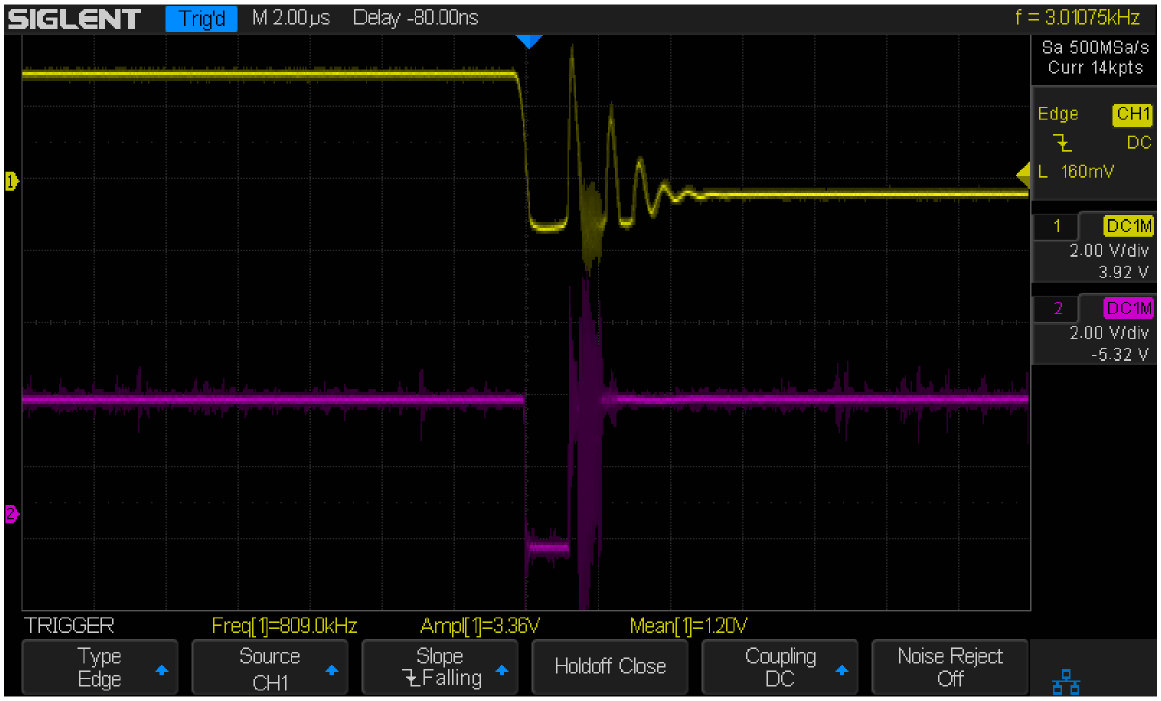
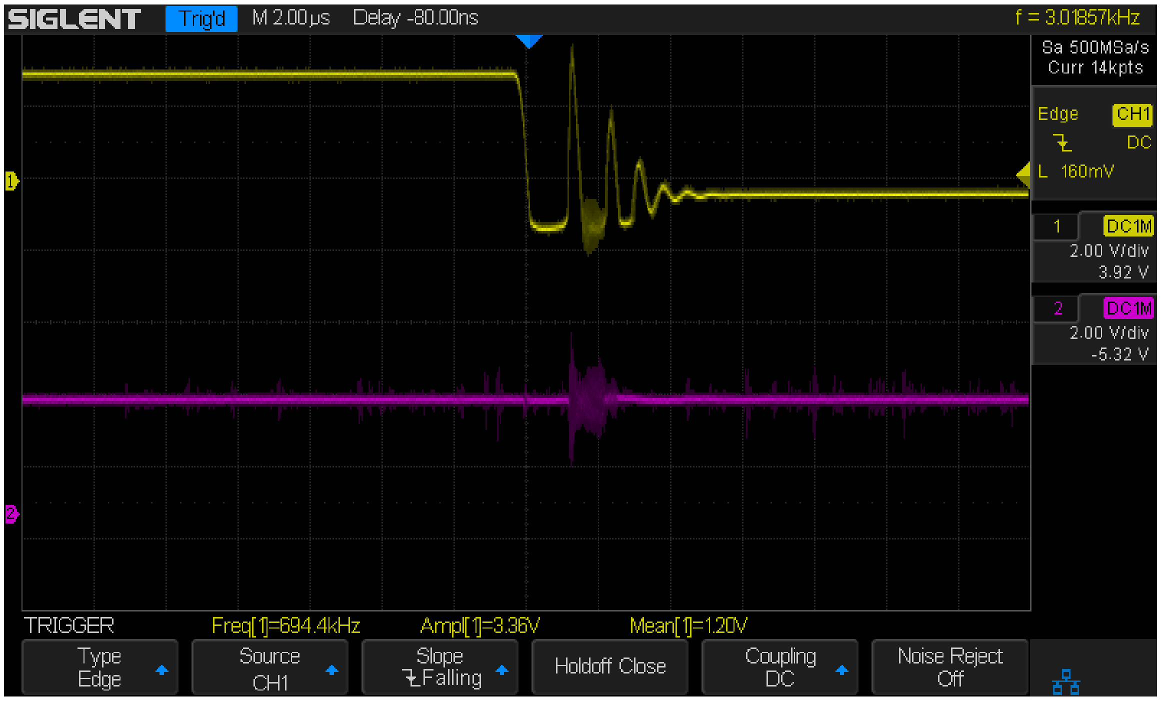


Figure 8 - Waveform With Static Hazard

But as shown in Figure 9, there is no static hazard since the function is redesigned as Z = BA + B'C + AC.



SIGLENT Trig'd M 2.00μs Delay -80.00ns f = 3.01857kHz

Sa 500MSa/s
Curr 14kpts

Edge CH1
DC
L 160mV

1 DC1M
2.00 V/div
3.92 V

2 DC1M
2.00 V/div
-5.32 V

TRIGGER Freq[1]=694.4kHz Ampl[1]=3.36V Mean[1]=1.20V

Type Edge | Source CH1 | Slope Falling | Holdoff Close | Coupling DC | Noise Reject Off

Figure 9 - Waveform without Static Hazard

Component Layout

1) Part A in pre-lab

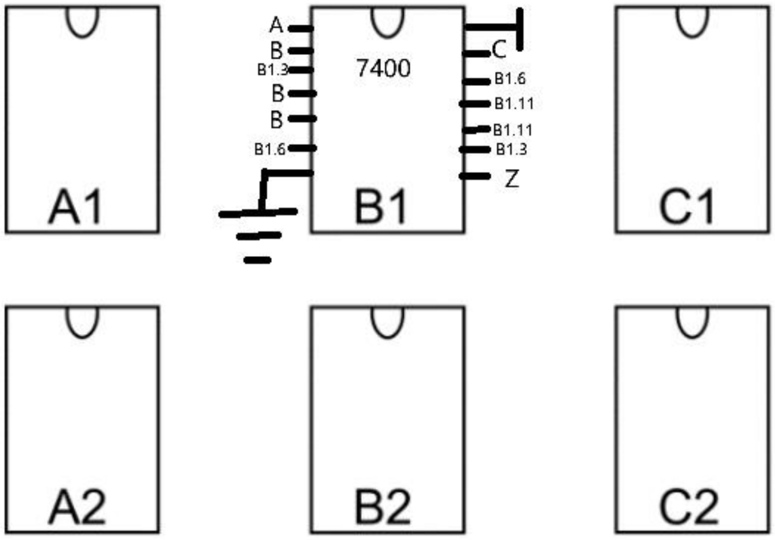


Figure 10 - Component Layout Sheet of Z= B'C+ BA

2) Part B in pre-lab

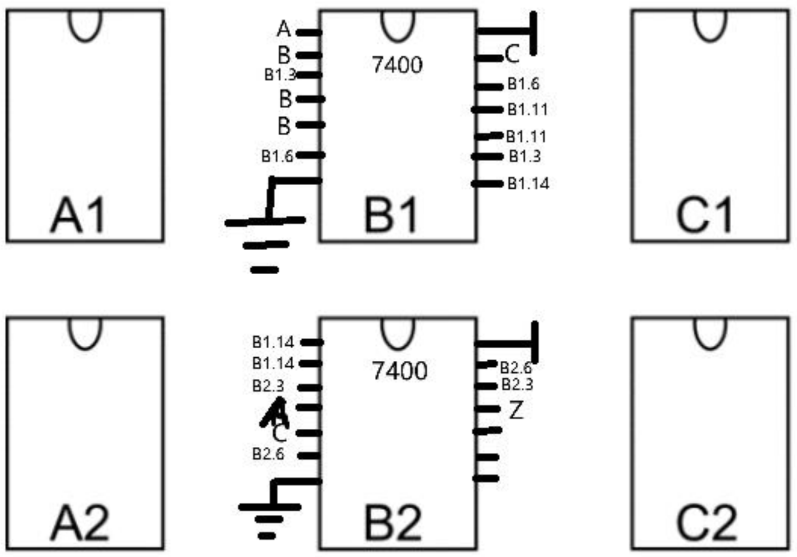


Figure 11- Component Layout sheet of Z=B'C+BA+AC

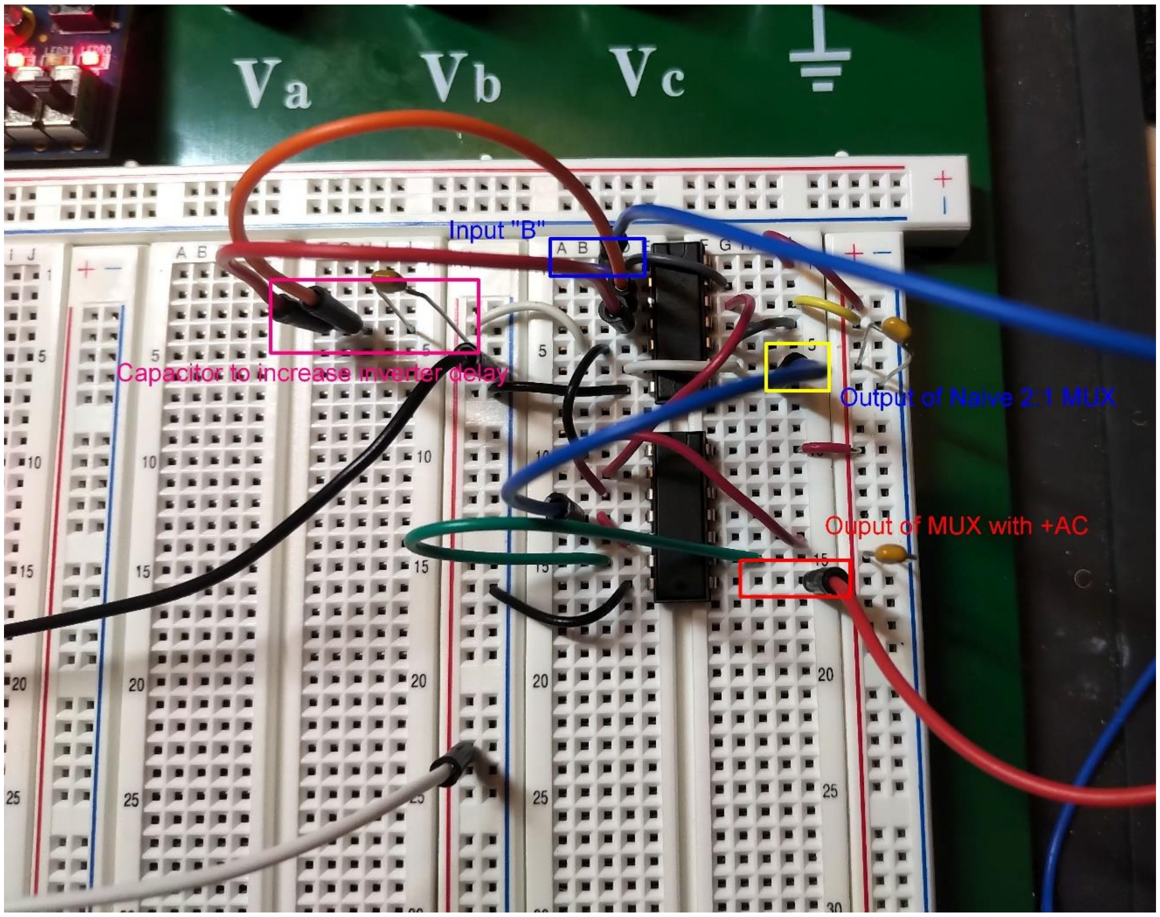


Figure 12 - Breadboard Layout

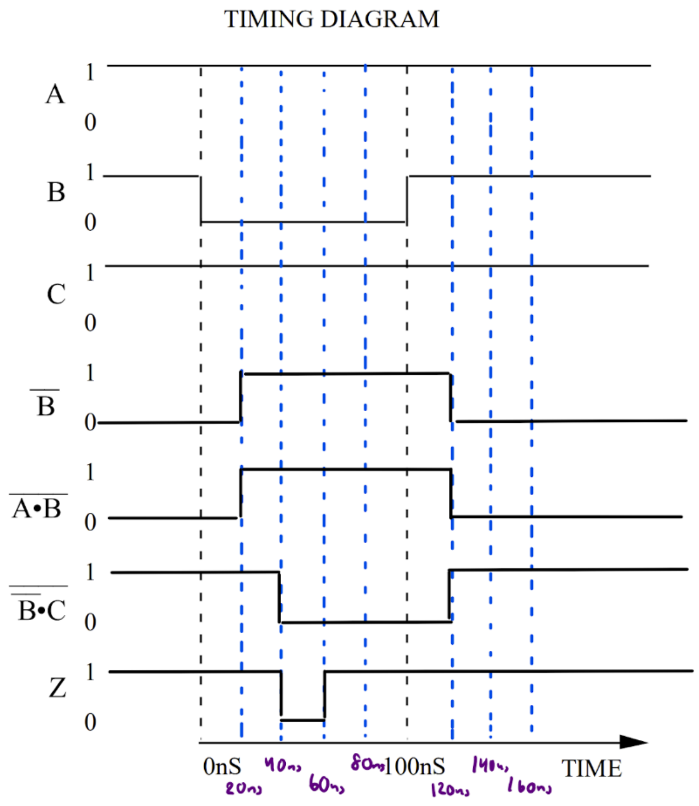
Answers to Pre-Lab Questions

1) *Q: Not all groups may observe static hazards (why?) If you do not observe a static hazard, chain an odd number of inverters together in place of the single inverter from Figure 17 or add a small capacitor to the output of the inverter until you observe a glitch. Why does the hazard appear when you do this?*

   Ans: Static hazards in our lab occur due to the delay in the IC chips / gates. Although the NAND gate created a delay for a couple of nanoseconds, some IC chips may have very little delay so that it is hard to observe all of them. Next, chaining inverters would cause the delay to build up. For example, if each inverter has a delay of 10 nanoseconds, then a chain of 5 inverters would be more than the delay of a single inverter. Adding a capacitor would also help us better observe the glitch because the capacitor will cause a short time of delay when selector bit B is switched from 1 to 0, making the delay period longer.

Answers to Post-Lab Questions

1) Q: *Given that the guaranteed minimum propagation delay of a 7400 is 0ns and that its guaranteed maximum delay time is 20ns, complete the timing diagram below for the circuit of part A. (See GG.17 if you are not sure how to proceed.)*



Ans:

2) Q: *How long does it take the output Z to stabilize on the falling edge of B (in ns)? How long does it take on the rising edge (in ns)? Are there any potential glitches in the output, Z? If so, explain what makes these glitches occur.*

Ans: It takes 60 ns for Z to stabilize on the falling edge of B, and it takes 0 ns for Z to stabilize on the rising edge. There are glitches in our system, as stated before, due to the delay in the gates. If there are multiple gates, the delay in each gate could build up, causing what is shown in the timing diagram above.

3) Q: *Explain how and why the debouncer circuit given in General Guide (Figure 22) works. Specifically, what makes it behave like a switch and how the ill effect of mechanical contact bounces is eliminated?*

Ans: First, the way a debouncer circuit works is that resistors from Vcc are connected to two latches. Then the latches put the voltage of the S-R latch inputs to 0. Now, once the switch is either set to A or B, it connects the node to ground and connects the unconnected latch to Vcc. This way, any disturbance can be eliminated.

Answers to Gen-Guide Questions

1） Q: *What is the advantage of a larger noise immunity?*

Ans: We have two advantages of larger noise immunity. First, a larger noise immunity allows the circuit to have a larger margin when deciding if output is a 1 or 0. Having a small noise immunity could result in more errors in deciding if an output is a 0 or 1. The second benefit is that the circuit will do a better job to ignore the noise and be affected by it.

2） Q: *Why is the last inverter observed rather than simply the first?*

Ans: The last inverter will have the most build-up of noise compared to the first inverter. Through this, we can determine how well the circuit is able to determine whether the final output is a 0 or 1. We can even find the noise immunity by observing the data from the last inverter by either inputting a 1 or 0 into the first inverter and seeing whether the output changes.

3） Q: *Given a graph of output voltage (VOUT) vs. input GG.7 voltage (VIN) for an inverter, how would you calculate the noise immunity for the inverter?*

Ans: We can find the difference between the nominal voltage and the inner voltage to calculate the noise immunity for both 1 and 0's noise immunity. First, for input logic level equal to 0, the average value of input is 0.35 V and for input logic level equal to 1, the average value of input is 3.5V. Second, we need to find the maximum value of input for the output not to change, from which we get for input equal to 0, the max value is 1.15 and for input equal to 1, the max value is 1.35. So we can derive that for noise immunity of 1, it should be 3.5 - 1.35 = 2.15 V. For the noise immunity of 0, it should be 1.15 - 0.35 = 0.8 V.

4） Q: *Why is a capacitor necessary close to each chip?*

Ans: Placing decoupling capacitors close to the chip ensures that there's a local source of charge readily available. When the circuit switches, these capacitors release stored charge to meet the sudden demand, preventing significant voltage droop.

5） Q: Irrespective of which polarity you choose for your LEDs, it is important that each LED has its own resistor. If we have two or more LEDs to monitor several signals, why is it bad practice to share resistors？

Ans: Using a single resistor for multiple LEDs may lead to unequal current distribution among them, causing some LEDs to be brighter than others. This lack of current matching can result in uneven illumination. And using individual resistors ensures current matching, consistent brightness, and reliability. It also provides isolation, preventing the failure of one LED from affecting others.

Conclusions

This lab helps me to review a lot of knowledge from previous classes such as ECE120. I learn to design the 2:1 MUX and it reminds me to consider the static hazard when it comes to the practice of the circuit. Additionally, I have a deeper understanding of the glitch due to the propagation delays of the gates. Besides, I get to know how to draw the component layout and the principle of debouncing circuits that are used to eliminate the noise due to bounce contact for traditional mechanical switches.