# ECE 385

Spring 2024

# Lab 1

Ziheng Li (zihengl5)

Section AL1

## Introduction:

Lab 1 primarily aims to develop an understanding of glitches (also known as static-1 hazards) and the methods to eliminate them. Glitches arise due to the propagation delay in gates, as the signal transmission takes time, the output for our circuit could be temporarily incorrect. Since designers can easily overlook glitches, gaining some knowledge about them is crucial to prevent related problems.
In this lab, we will use a 2-1 Mux implemented by NAND gates as an example circuit for understanding glitches.

## Description:

Pre-lab part A is a circuit for 2 to 1 MUX. The basic Boolean function for 2-1 MUX is Z = B'C + BA, where A and C are two inputs and B is a selector. Implement it with NAND gate, we have Z = (C NAND (B NAND B)) NAND (A NAND B), as shown below. Note that this is the naïve layout that might cause glitches.

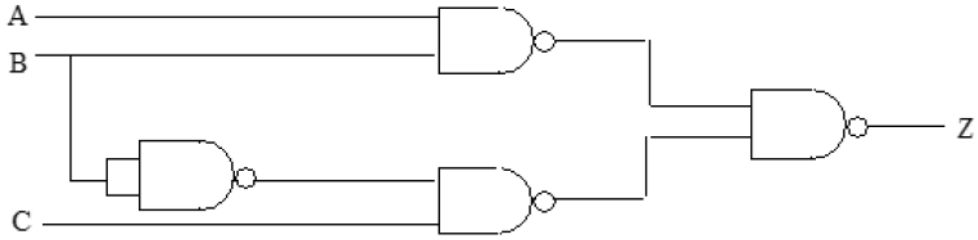
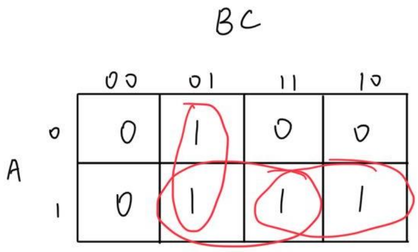


Fig1. Circuit diagram for prelab A

Based on the property of a 2-1 MUX, if both A and C are high, the output should always be high and not affected by the selector B. However, due to the presence of the extra NAND gate after B, the propagation delay will affect the output momentarily. Therefore, if we change B from high to low, we will observe the output Z fluctuated for a short period, and then change back to correct output.

Pre-lab part B ask us to modify the circuit to eliminate all glitches. In order to avoid Static-1 hazard, we have to cover all the adjacent min-terms in the K-map, as shown below.



| A \ BC | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |

Fig2. K-map for 2-1MUX with glitches-free implementation

Right now, the output is Z = B’C + BA + AC. NAND representation is
Z = (C NAND (B NAND B)) NAND (A NAND B) NAND (A NAND C).

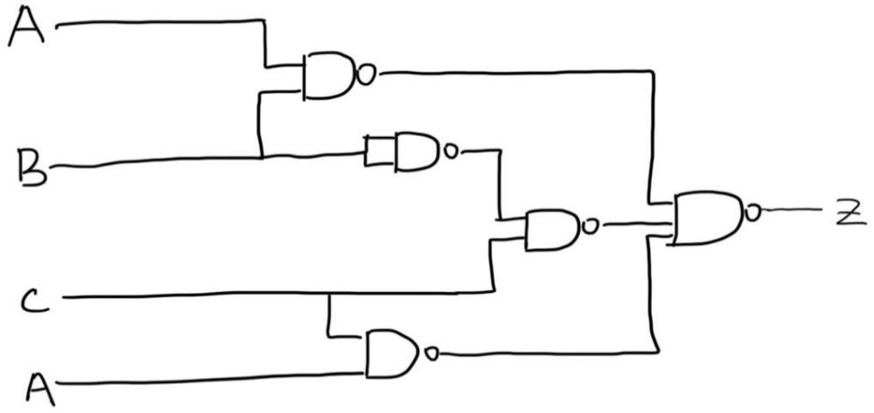


Fig3. Circuit diagram for prelab B

In this case, even if the is propagation delay caused by B, the extra AC term would be smart enough to maintain the output correctly. Hence, cover all adjacent min-terms in the K-map is a solution to glitches in AND-OR circuit.

## Circuit Diagram:

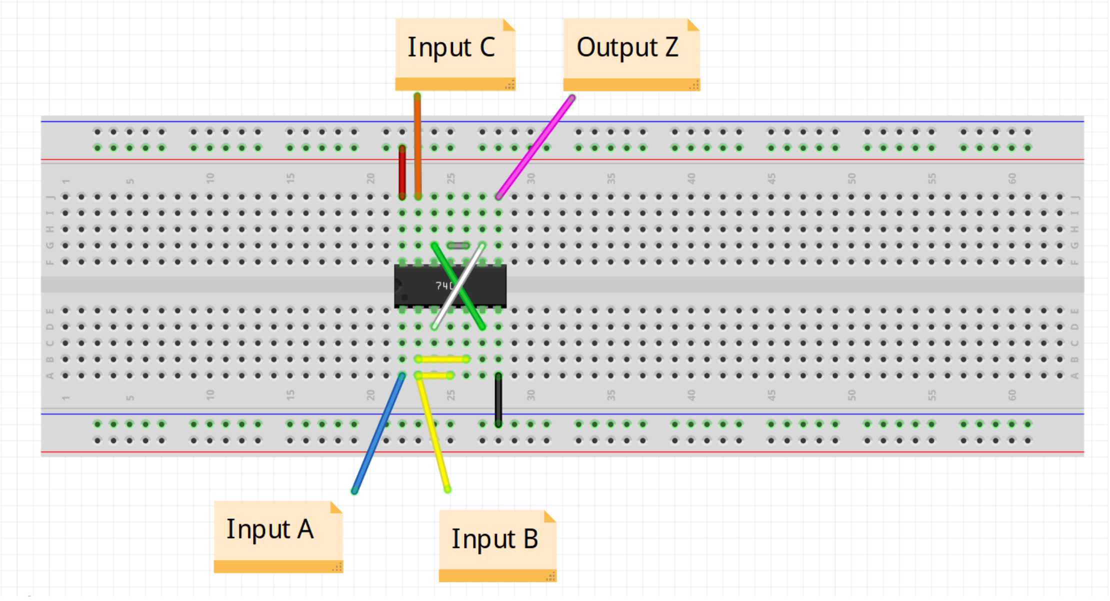


Fig4. Circuit diagram for prelab A using one 7400 chip

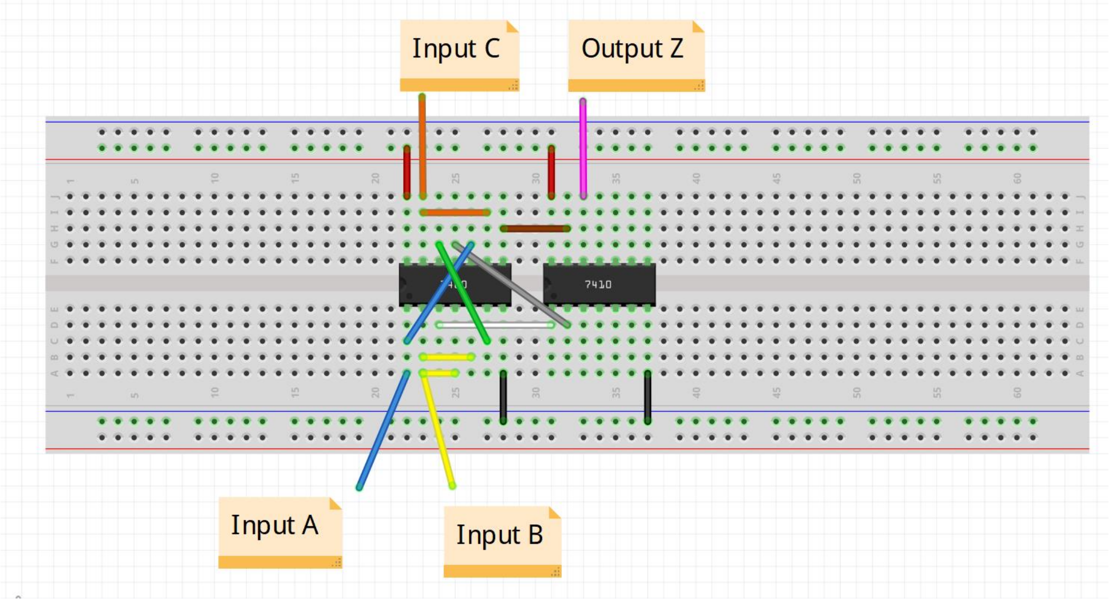


Fig5. Circuit diagram for prelab B using one 7400 chip and one 7410 chip

## Component Layout Sheet:

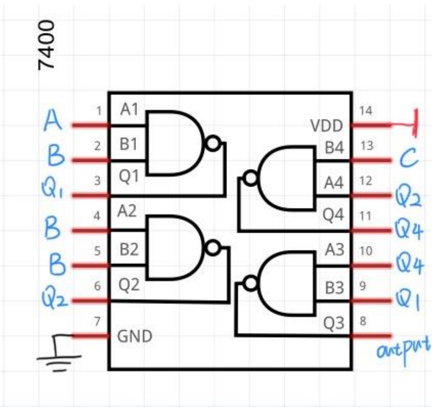


Fig6. Component layout sheet for part A

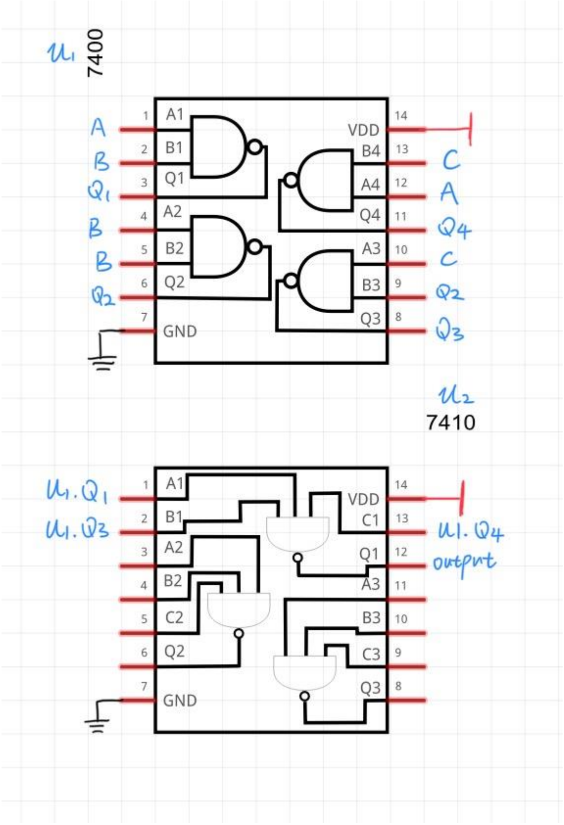
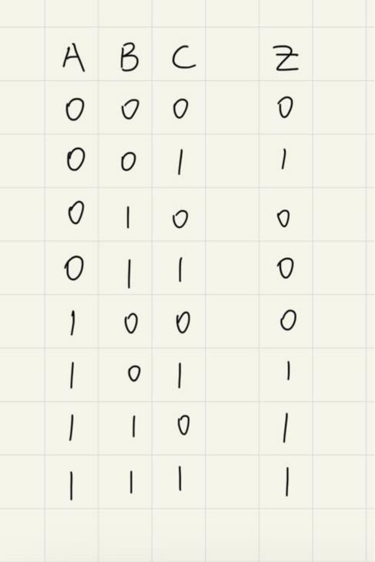


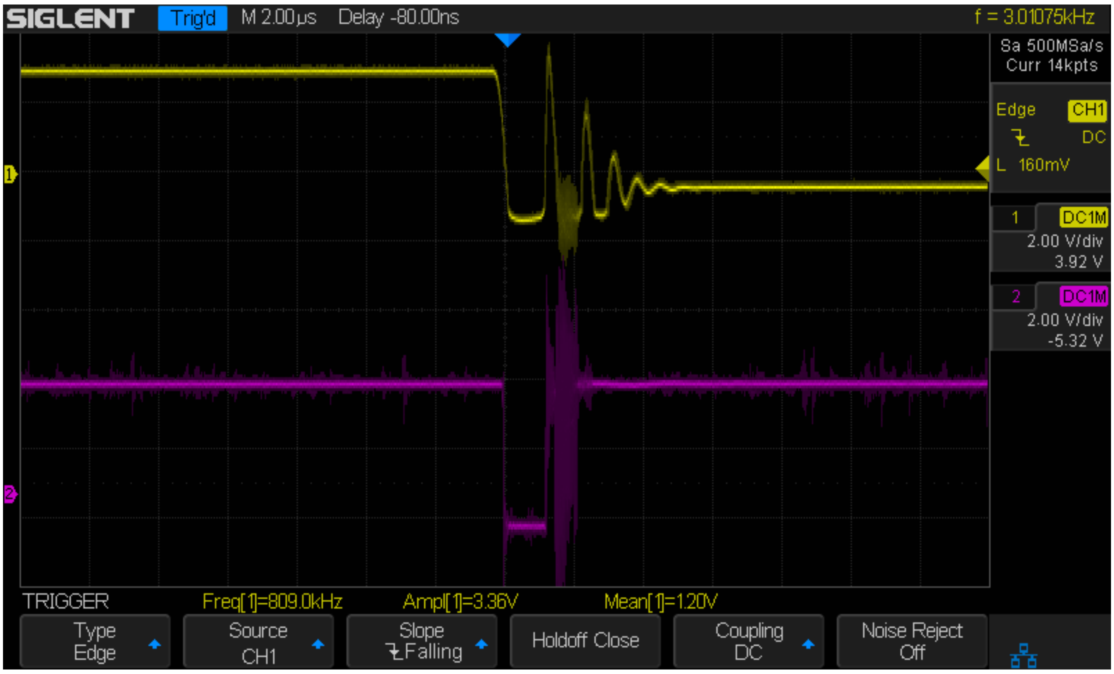
Fig7. Component layout sheet for part B

# Documentation:



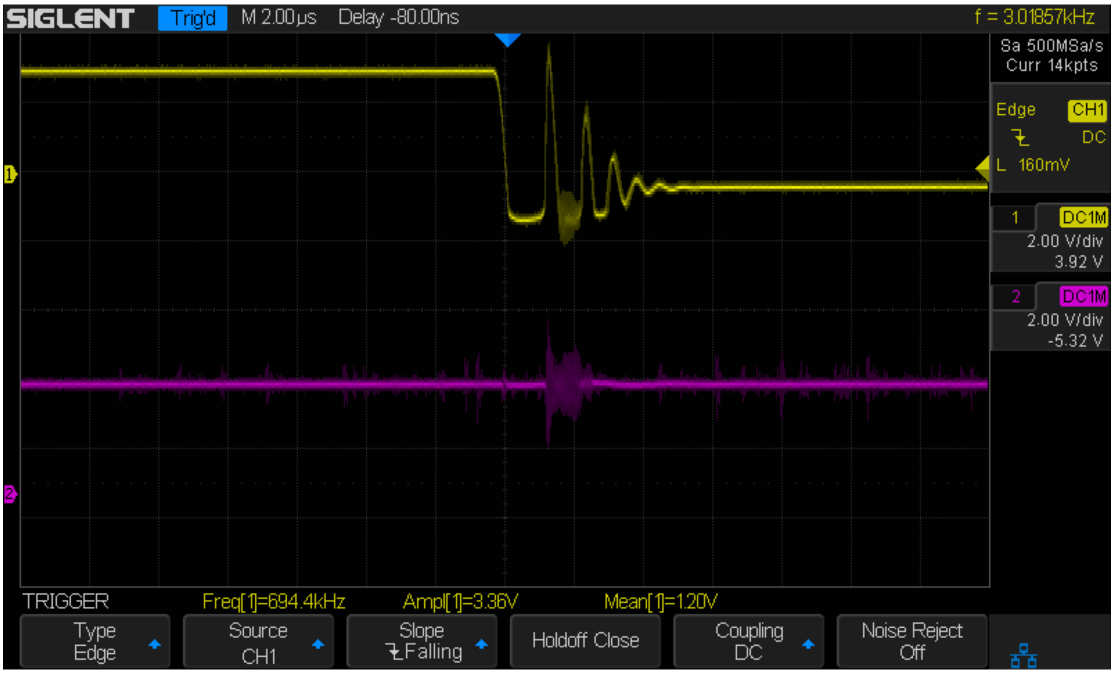
| A | B | C | Z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Fig8. Truth table for circuit A and B (same for both part)



```
SIGLENT  Trig'd  M 2.00μs  Delay -80.00ns                f = 3.01075kHz
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]=809.0kHz   Ampl[1]=3.36V   Mean[1]=1.20V
Type Edge | Source CH1 | Slope Falling | Holdoff Close | Coupling DC | Noise Reject Off
```

Fig9. Oscilloscope output for prelab A (with glitches)


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

Fig10. Oscilloscope output for prelab B (glitches free)

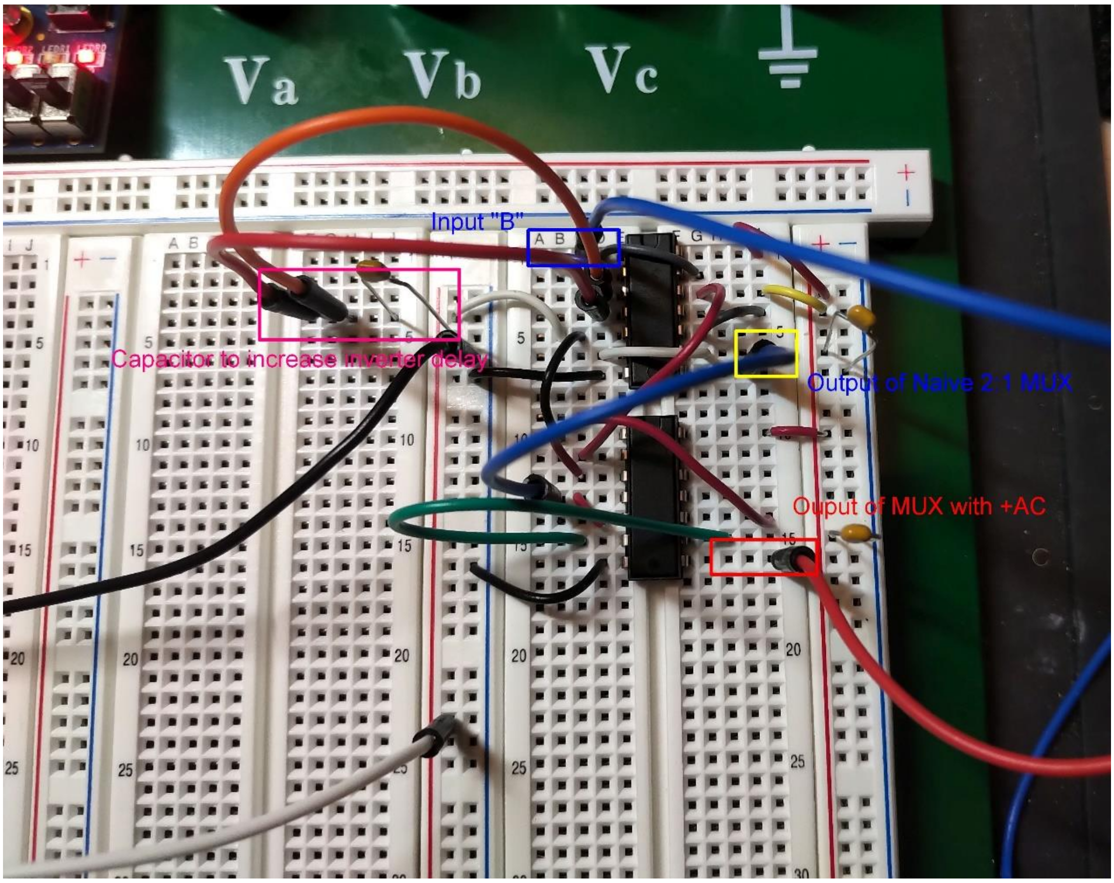


Fig11. Actual layout for prelab A and B

## Answer to pre-lab & lab questions:

**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?**

From the lab document, we know that the propagation delay of a 7400 chip varies from 0ns to 20ns. If the delay is too small, then the static hazard happens too short for us to observe.
However, when we chain up multiple inverters or add a small capacitor, we are increasing the propagation delay to a more observable time.

**Which edge (rising/falling) of the input B are we more likely to observe a glitch at the output?**

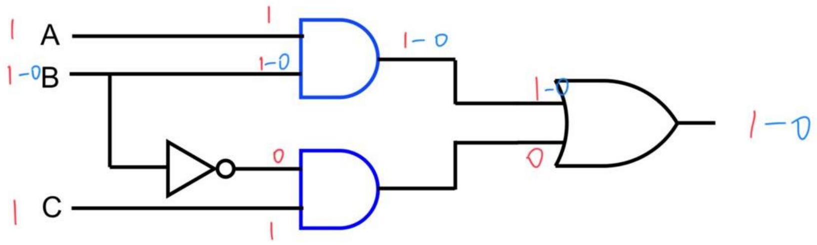


Fig12. Glitch happens at the falling edge

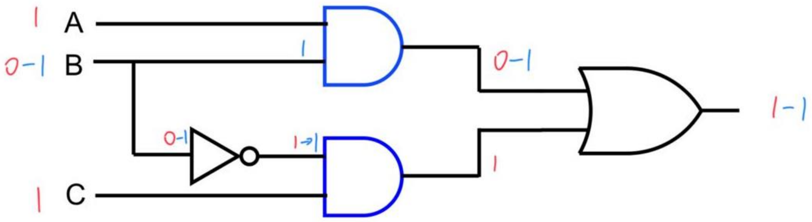


Fig13. Glitch does not happen at rising edge

If we set both inputs A and C 1, and then change B from 1 to 0, the output of the circuit will became 1-0(due to glitches)-1. However, if change B from 0 to 1, the output of the circuit will became 1-1(even with delay)-1. Moreover, from Post Lab Q1, we know that the unstable period is longer for falling edge than rising. Hence, I conclude that we are more likely to observe glitch at falling edge of input B.

## Answers to post-lab questions:

**1.)**

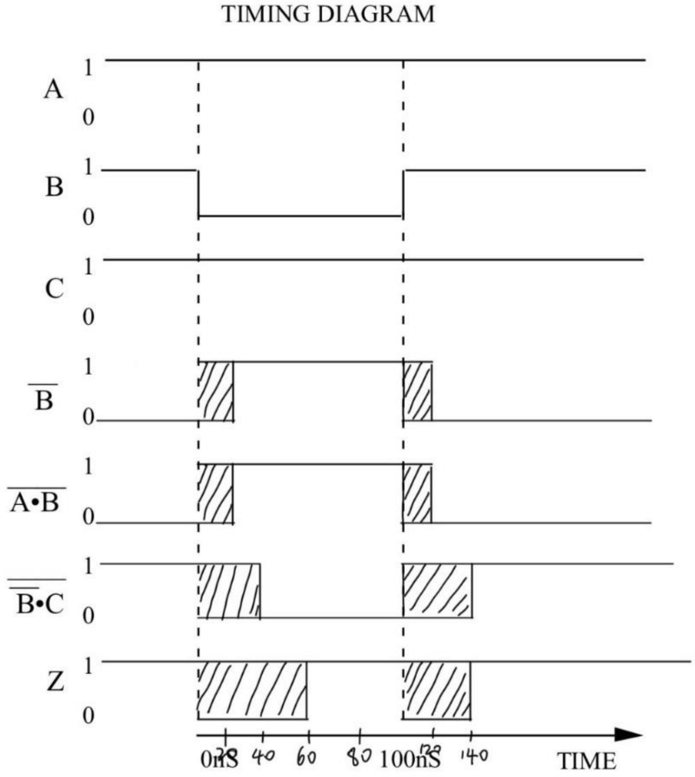


Fig14. Timing diagram for circuit in part A

**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.**

It takes 60ns for the output Z to stabilize on the falling edge of B. It only takes 40ns for the rising edge of B to stabilize. There are potential glitches in the output Z, since the output will be undetermined during 0-60ns (shaded region). The output of Z might be incorrect due to propagation delay of the extra NAND gate after B.

**2.) 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?**

As shown in the figure15, when the switch is at position A, the Q was set to 1 and QN was set to 0. When the switch is at position B, the Q was set to 0 and QN was set to 1. Moreover, the output will not change during switching due to the property of SR latch.

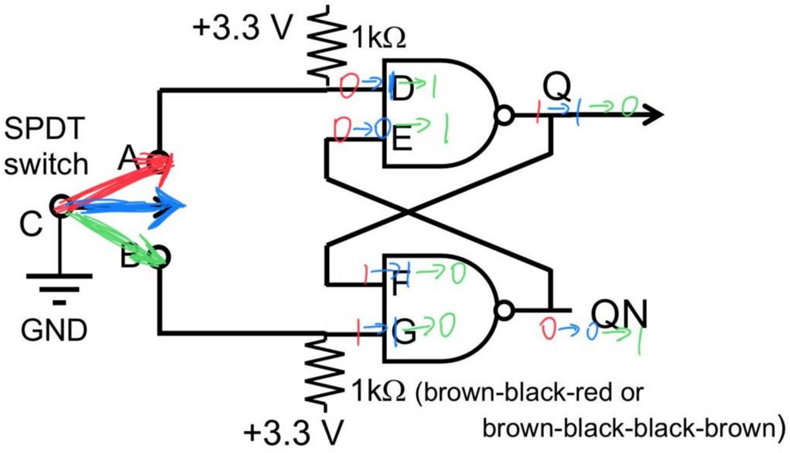


Fig15. Debouncer circuit

In the graph above, suppose the switch is at A position (in red), the output of Q will be 1 and QN will be 0. When the switch is separated (in blue), the output of Q and QN will not be changed since the debouncer circuit was able to hold the output value until next input. When the switch is at B position (in green), the output Q changes to 0 and QN changes to 1.

The debouncer circuit solve the contact bounce issue by holding the output value, and by connecting with A or B, the circuit can act like a switch.

## Answers to GG.7:

With a larger noise immunity, the chip could take a larger range of voltage and considering them as logic “1” or logic “0”.

If the input for the first inverter is higher/lower than the typical voltage for logic “0/1”, after passing through a series of inverter, the output voltage of each inverter will become more and more stable until it reach the nominal logic “0/1”. Therefore, the last inverter might have the most stable output voltage and very close to nominal logic “0/1”.

The noise immunity for logic “1” is 2.15V and the noise immunity for logic “0” is 0.8V. We are picking the lower value to be the overall noise immunity. Hence, the noise immunity for the inverter is 0.8V.

## Answers to GG.23:

If two or more LEDs are sharing the same resistor, we cannot guarantee each LED receives the appropriate current to operate. Each LED will have slightly varying resistance; hence, current distribution could be unequal.

## Conclusion:

In lab 1, we explored the concept of gate delays and how they might lead to glitches. The lab session also introduce methods for resolving these glitches, one of them is to add the adjacent min-term in the K-map. We also revisited foundational concepts like MUX and K-map. Additionally, the lab demonstrated how contact bounce was caused and how could it be addressed with a debouncer circuit. The experiment went on really smoothly, and it reinforce the importance for us to meticulously deal with any potential problems in circuit design.