

ECE 385

Fall 2019

Experiment #1

Introductory TTL Experiment

Xinglong Sun
ABD
Mihir Iyer

Purpose of Circuit:

This lab serves as an introductory experiment to later labs. The equipment get introduced and we're going to get more familiar with the format and procedure of ECE385 lab. The lab is about building a 2-to-1 Mux with only NAND gates, observing the scope traces, and solving the issues of glitches.

Written Description of Circuit:

- Part A:

The requirement of this part is to build a 2-to-1 Mux with NAND Gates as shown in the General Guide Figure 16. In order to come up with this design, I start from the Boolean Algebra equation. With Z representing the output, A and C representing the inputs, and B representing the select signal, the equation is $Z = AB + CB'$. Using DeMorgan's Law, the equation can easily be transformed to $Z = ((AB)' \cdot (B'C)')'$. Then, with $B' = (BB)'$, we can achieve the Boolean Equation with 4 NAND Gates. We choose to do this on one 7400 quad chip.

- Part B:

The requirement of this lab is to design a glitch-free 2-to-1 Mux with NAND Gates. In order to avoid Static – 1 Hazard in SOP circuit, we just have to cover all adjacent min terms in the K-map. The K-map is as follows:

		AB			
		00	01	11	10
C	0	0	0	1	0
	1	1	0	1	1

In this case, we add the term AC, and the expression becomes

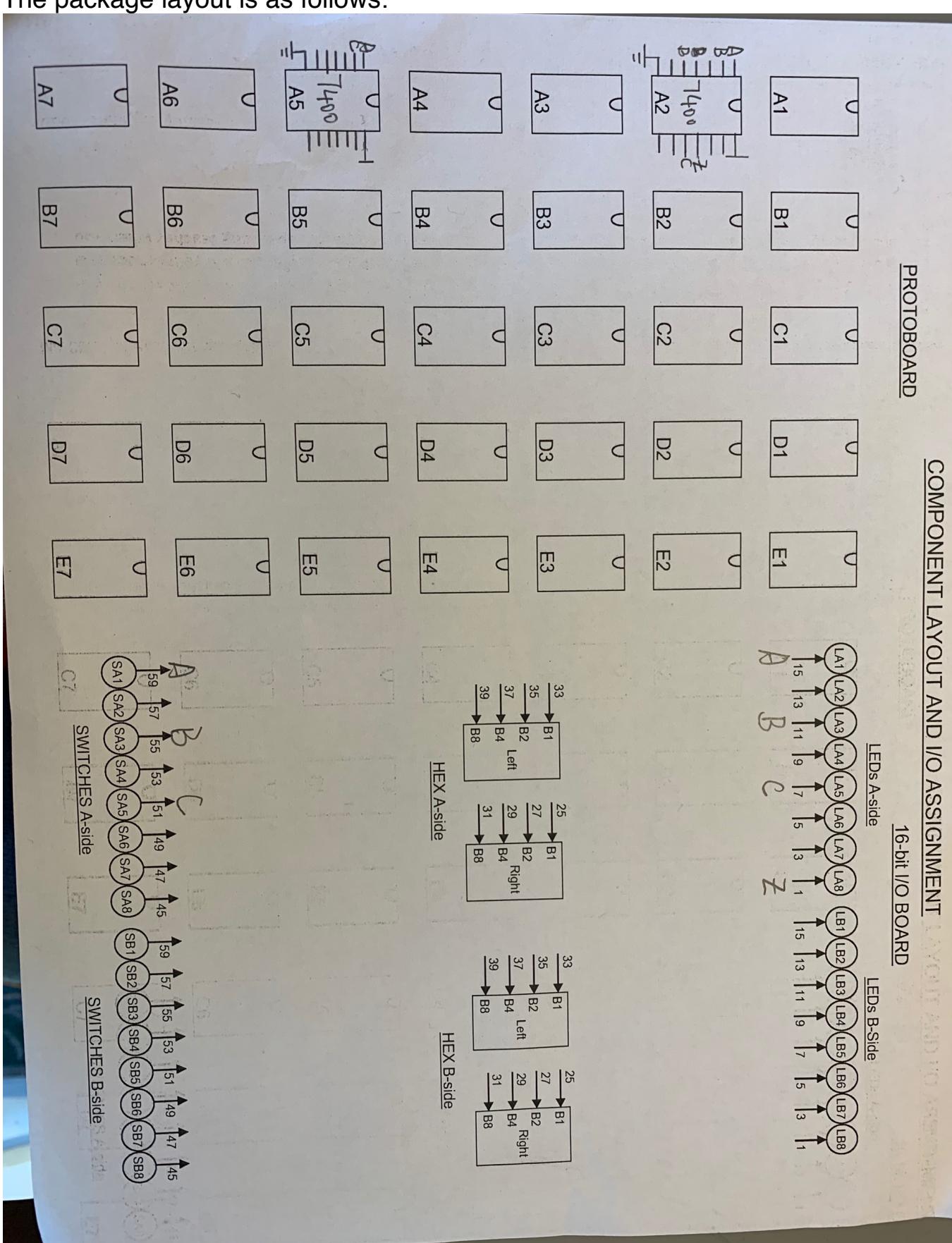
$$Z = AB + CB' + AC = ((AB)' \cdot (CB')' \cdot (AC)')'$$

Again, with $B' = (BB)'$, we can achieve the Boolean Equation with six NAND Gates. We choose to do this on two 7400 quad chips.

PROTOBOARD

COMPONENT LAYOUT AND I/O ASSIGNMENT

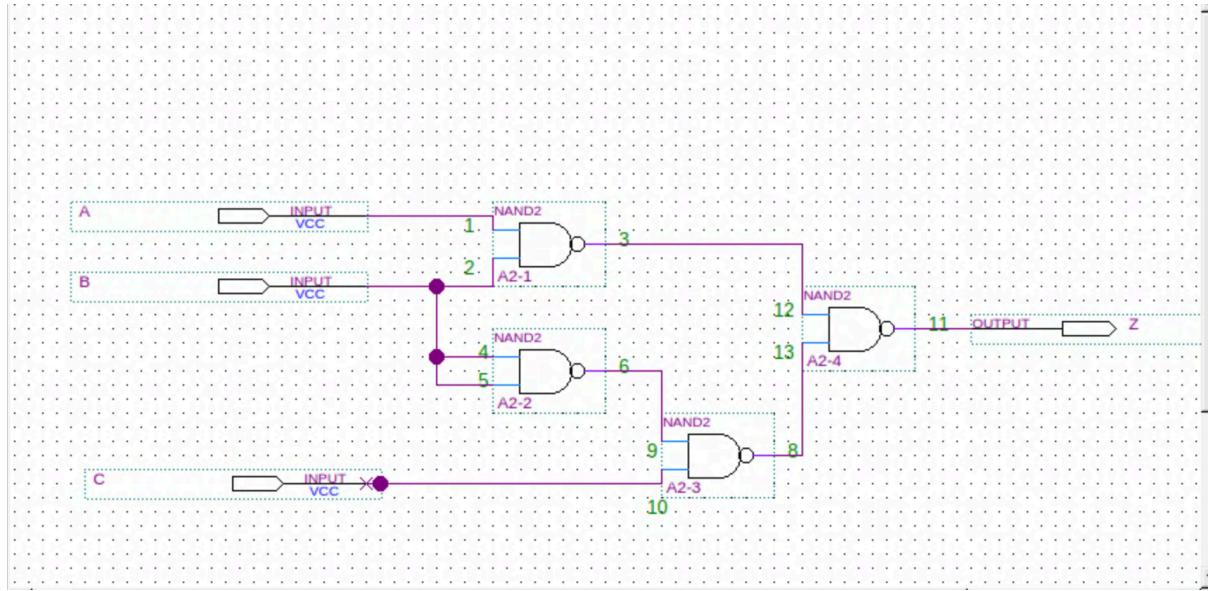
16-bit I/O BOARD



Logic Diagram and Circuit:

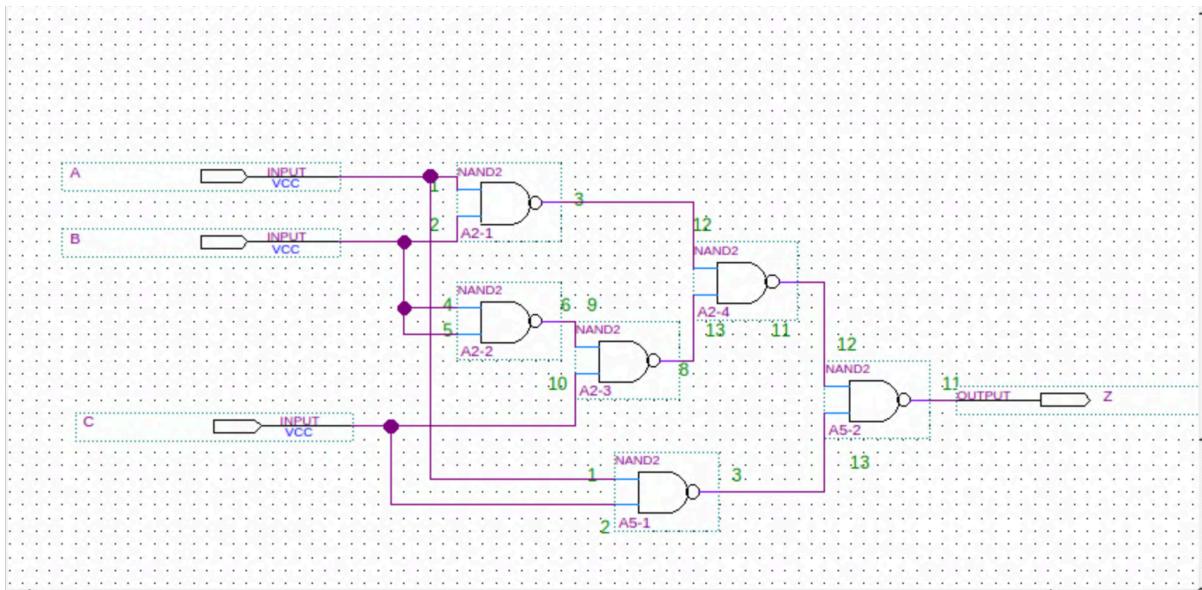
- Part A

The gate-level layout of the circuit is as follows:



- Part B

The gate-level layout of the circuit is as follows:



Documentations:

- Prelab Truth Tables

As the purpose of MUX is to select between two signals, we verified that the truth table for the circuit of pre-lab part A is as follows:

A	C	B	Z
0	0	0	0
0	0	1	0
0	1	1	0
0	1	0	1
1	1	0	1
1	1	1	1
1	0	1	1
1	0	0	0

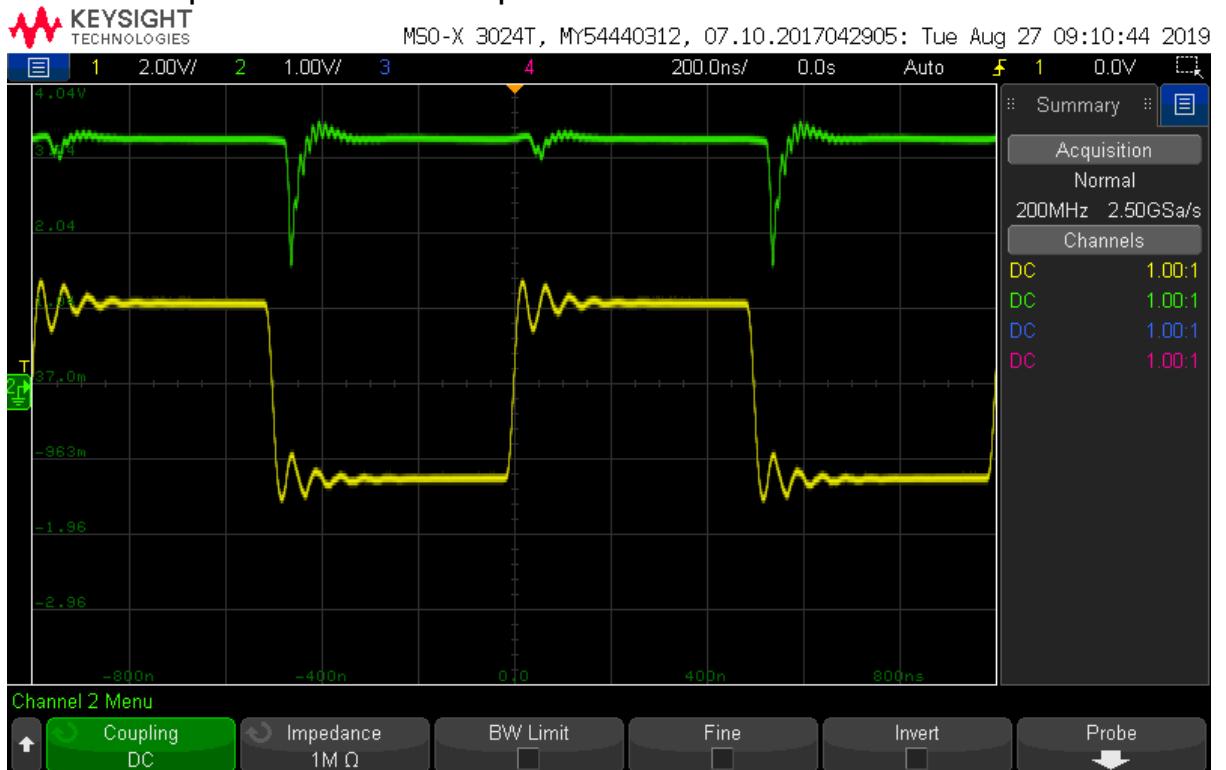
As the input and output and the function of the circuit in part B is unchanged, the truth table remains the same as that of the part A. We also verify this while doing the lab.

- Prelab Answers:

1. **Static Hazards:** We do not observe static hazards in our circuit. The reason that not all groups can observe static hazards is that there is no specific amount of propagation delay time of 7400, but exists a minimum propagation delay of 0ns. The chip of each group is different.
2. **Why does the hazard appear when you do this? :** We artificially increase the propagation delay time by either increasing the number of inverters used or by adding a capacitor at the output. Therefore, the period where the output produces an incorrect value is increased, and human eyes can physically capture a glitch.

- Scope Printout and Analysis:

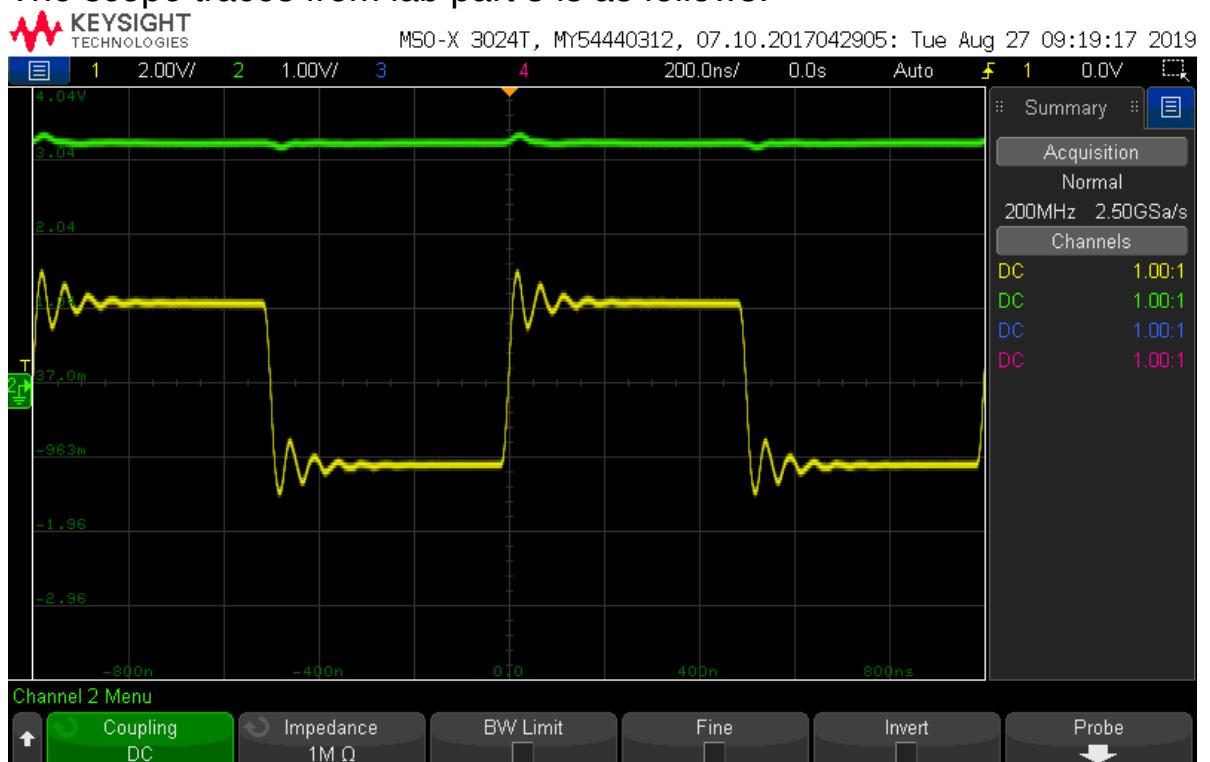
1. The scope traces from lab part 2 is as follows:



Yellow Curve: Signal B

Green Curve: Output

2. The scope traces from lab part 3 is as follows:



Yellow Curve: Signal B

Green Curve: Output

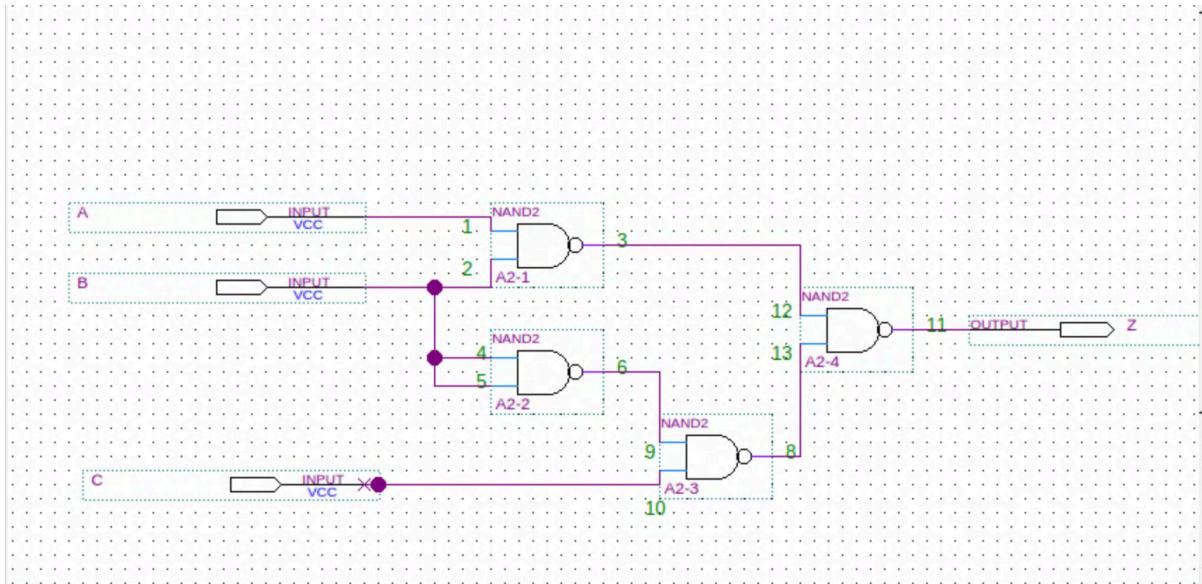
3. An analysis: As we can see from the scope traces, the glitch is effectively suppressed by using our redesigned circuit in prelab part B.

- Lab Answers:

1. **Part B circuit behaves the same as part A?** :The circuit of part B responds the same as that of part A.

2. **At which edge of the input B are we more likely to observe a glitch:** By observing the scope traces and theoretical analysis, we learn that at the **falling edge** ($1 \rightarrow 0$) of select signal, the glitch is more likely to happen. A concrete analysis is as follows, given the following logic diagram and pin labels, with input 1 and 10 held high, at the falling edge ($1 \rightarrow 0$), the output 8 will momentarily be held at the previous value (1) due to one more gate delay when the output 3 has already changed (0 \rightarrow 1). This will momentarily cause the output 11 be 0, which produces a static-0 hazard. However, at the rising edge ($0 \rightarrow 1$), though signal B is still one more gate delay to input 13 than to input 12, the output 12 will change from 1 to 0. For a NAND Gate, when one input is held at 0, the output will always be 1 regardless of the other input.

This can also be shown in the timing diagram in the following page. At the rising edge of select signal, the output took 20ns shorter to be stabilized than at the falling edge.



Answers to Post-Lab Questions:

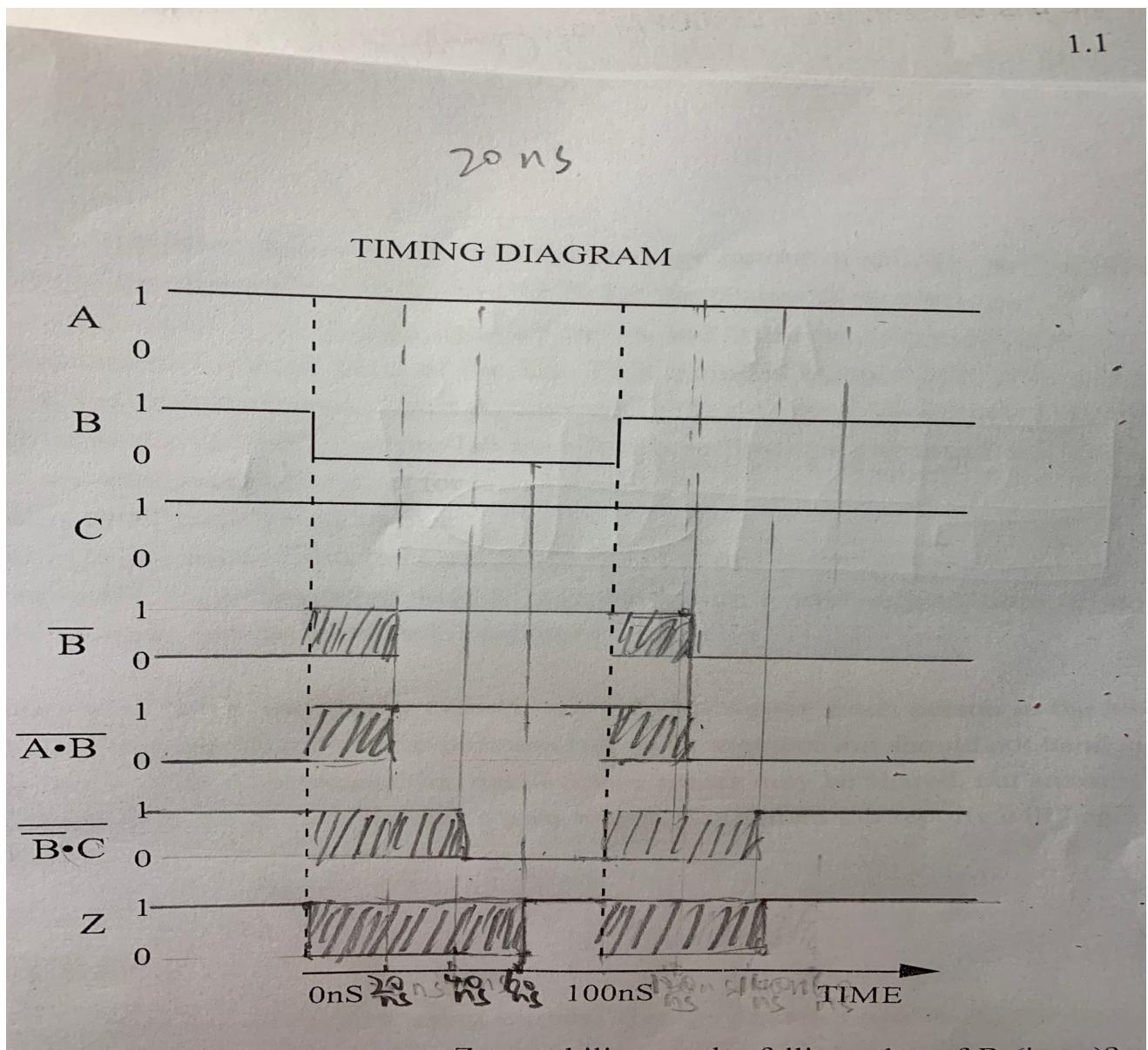
1. The timing diagram is as follows.

It takes the output Z **60 ns** to stabilize on the falling edge of B.

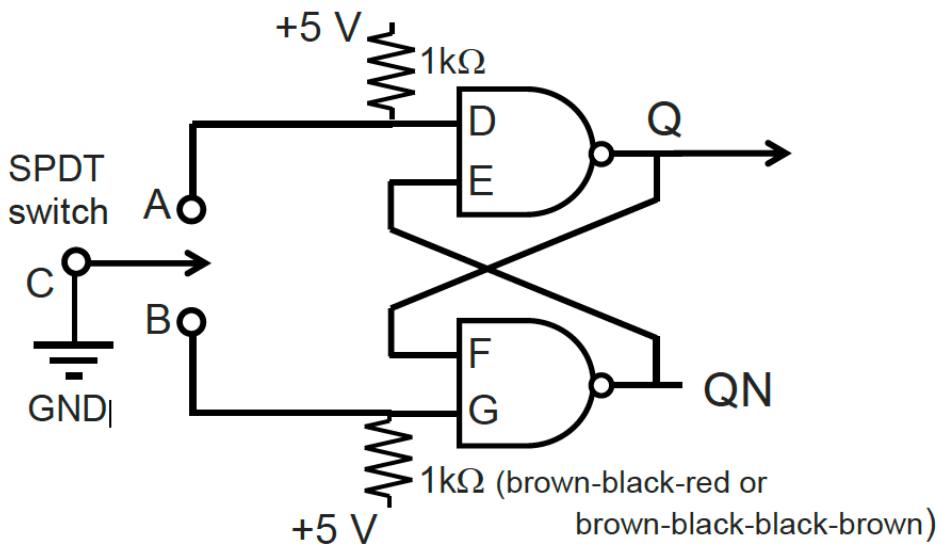
It takes the output Z **40 ns** to stabilize on the rising edge of B.

The **shaded areas** in the timing diagram are the areas where the output is unknown and the potential glitches might occur.

The propagation delay time of the 7400 is the root cause of the glitches. As a designer, we have to consider the worst – case scenario, when the chips take the guaranteed maximum delay time.



2.



The circuit diagram above can act like a debounced switch without ill effect due to the following reason:

Initially, the inputs D and G are pulled-up to 5V via 1kohm resistors, and E, F, Q, and QN are floating. For example, when we toggle the switch to A, input D will be connected to the GND. **For a NAND Gate, when either of the two inputs become LOW, the output signal will become HIGH.** Q and F are turned to HIGH in this case. Since F and G are both HIGH, QN and E will be LOW. At this point, even if the switch bounces and the input D is again turned to HIGH, E is still held at LOW and will ensure the output Q held at HIGH. When the switch is toggled to B, for the similar reason, QN is going to be HIGH, and Q will be LOW. The diagram can therefore act like a switch producing stable output signals.

Answers to GG Questions:

1. GG.6

A larger noise immunity is beneficial because the gate will be less sensitive to weaker signals, and thus will not change its output value due to some tiny fluctuations of the input signal.

We observe the output of the last inverter rather than the first one. We do this because we can artificially increase the noise by using multiple gates. This is more helpful for the observation and our measurement of the noise immunity.

We first determine x, the noise immunity for logic “0” at the input. In order to measure it, we increase slowly the input voltage from “0” to a certain threshold voltage that the output turns from “1” to “0”. We then determine y, the noise immunity for logic “1” at the input. In order to measure it, we decrease slowly the input voltage from “1” to a certain threshold voltage that the output turns from “0” to “1”. The overall noise immunity of the gate is just the smallest of the x and y.

2. GG.29

The resistor acts like a current limiter in the circuit. Therefore, if you have multiple LEDs sharing the same resistor and several LEDs are turned on at the same time, they will divide the total current flowing through. This will cause each LED and IC being fed in a lower current than usual.

Conclusions:

In this lab, I got more familiar with the tools used in ECE385 and some basic digital design rules and techniques. I've also got a chance to review the knowledge of ECE120 which will be fundamental for the later labs. More importantly, I learnt more concretely what static hazards are and how we can deal with them in the future. The first lab moved very smoothly for my partner and me. We collaborated perfectly and we finished the lab within 20 minutes.