ECE 385

Spring 2021

Experiment 1

Introductory Experiment

Adam Naboulsi
Lab Section: ABP

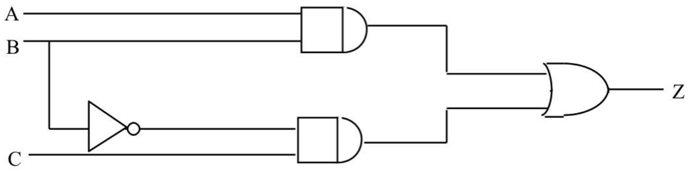
Purpose of Circuit

This lab served as an introduction to ECE 385 and lab equipment that will be used throughout the course including the student lab kit, I/O boards, an oscilloscope, and a pulse generator. It also served as a guide for future lab assignments particularly because of the information provided in the General Guide and the fundamental concepts I was introduced/reintroduced to and will most definitely revisit in the future including static hazards, debouncing, delays, glitches, K-maps, and logic diagrams.

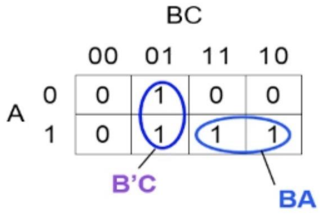
Written Description of Circuit

The circuit built in this lab was a basic 2-input MUX with two minterms and an extra/redundant term (added later in the lab) using NAND gates. The function of a 2-input MUX is to take two digital input signals (and a select input) and produce a single output. The purpose of the extra term added onto the base MUX was to prevent Static-1 hazard (through covering the minterms) which results from propagation delay in the logic (and may momentarily produce an unwanted output when an input is changed.

I was first given a design of the 2-input MUX that used 2 AND gates and an OR gate as shown below:

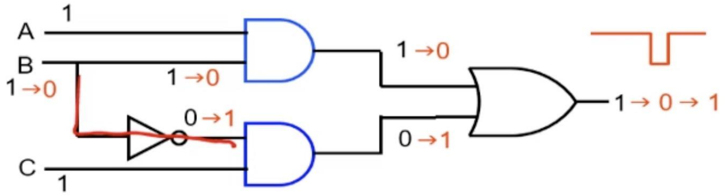


This design was simply an implementation of the following SOP K-map:



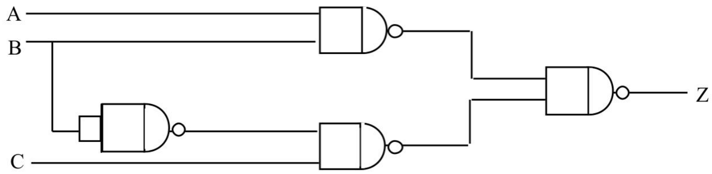
		BC			
		00	01	11	10
A	0	0	1	0	0
	1	0	1	1	1

The main issue with this circuit was the presence of the inverter in one of the circuit paths which caused a delay. When input B switches from a 1 to a 0, this delay resulted in a momentary drop to 0 in the output (does not affect the steady state, however it is a static hazard) as shown in the progression of logic values below:



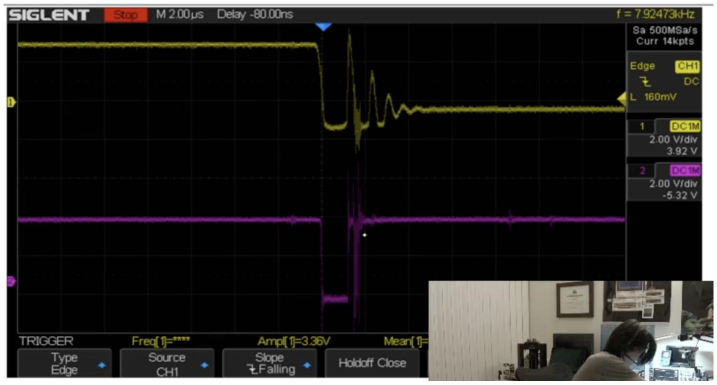
This static hazard (glitch) was addressed later in the lab.

To make this circuit, I had to modify it so that only NAND gates were used since the lab kit only contains 7408s and 7432s which only use NANDs, NORs, and inverters. Through inserting inverters and using DeMorgan's law, the following circuit was designed which was able to fit into a single 7400 chip:



In building this circuit, I powered it with the 3.3V source on the FPGA board. I attached a pulse generator (generating a 5V square wave) to input B, causing it to trigger. Additionaly, I placed a samll capacitor to the output of the inverter (the double input B NAND gate) to artificially increase its delay thus emphasizing the static-1 hazard (glitch) produced in the output when input B is triggered. I also attached two decoupling capacitors attached to the ground and power terminals near the 7400 power lines. These were attached to the circuit in order to minimize static caused by momentary short circuits between VDD and GND in the CMOS logic when the logic gate is switching. The effect of adding these capacitors is displayed below along with the glitch when input B changes from HIGH to LOW (yellow - input, purple - output):

Without Decoupling Capacitors



SIGLENT Stop M 2.00μs Delay -80.00ns f = 7.92473kHz

Sa 500MSa/s
Curr 14kpts

Edge CH1
DC
L 160mV

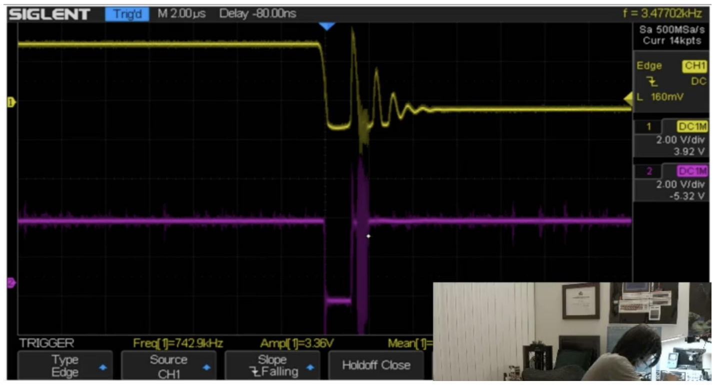
1 DC1M
2.00 V/div
3.92 V

2 DC1M
2.00 V/div
-5.32 V

TRIGGER Freq[1]=**** Ampl[1]=3.36V Mean[1]=

Type Edge
Source CH1
Slope Falling
Holdoff Close

With Decoupling Capacitors



SIGLENT Trig'd M 2.00μs Delay -80.00ns f = 3.47702kHz

Sa 500MSa/s
Curr 14kpts

Edge CH1
DC
L 160mV

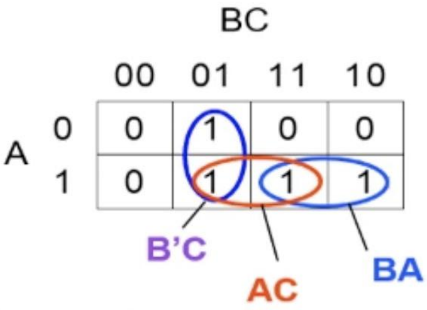
1 DC1M
2.00 V/div
3.92 V

2 DC1M
2.00 V/div
-5.32 V

TRIGGER Freq[1]=742.9kHz Ampl[1]=3.36V Mean[1]=

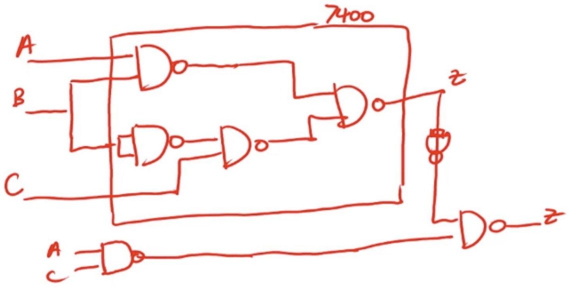
Type Edge
Source CH1
Slope Falling
Holdoff Close

To prevent this glitch caused by the presence of the inverter and its delay in one of the circuits paths, I modified the circuit by adding a redundant term (AC) that covered the adjacent minterms (BA and B'C) in the original SOP K-map:

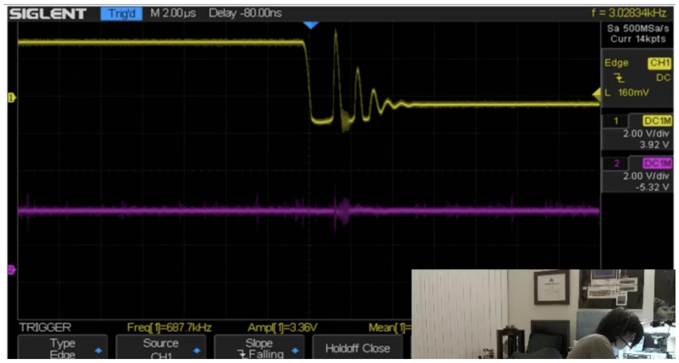


A \ BC	00	01	11	10
0	0	1	0	0
1	0	1	1	1

This extra term prevents/minimizes the glitch by making the output not solely dependent on input B in the case that it switches from HIGH to LOW. To implement this redundant term, I again used inverters and De Morgan's law such that only NAND gates were used:



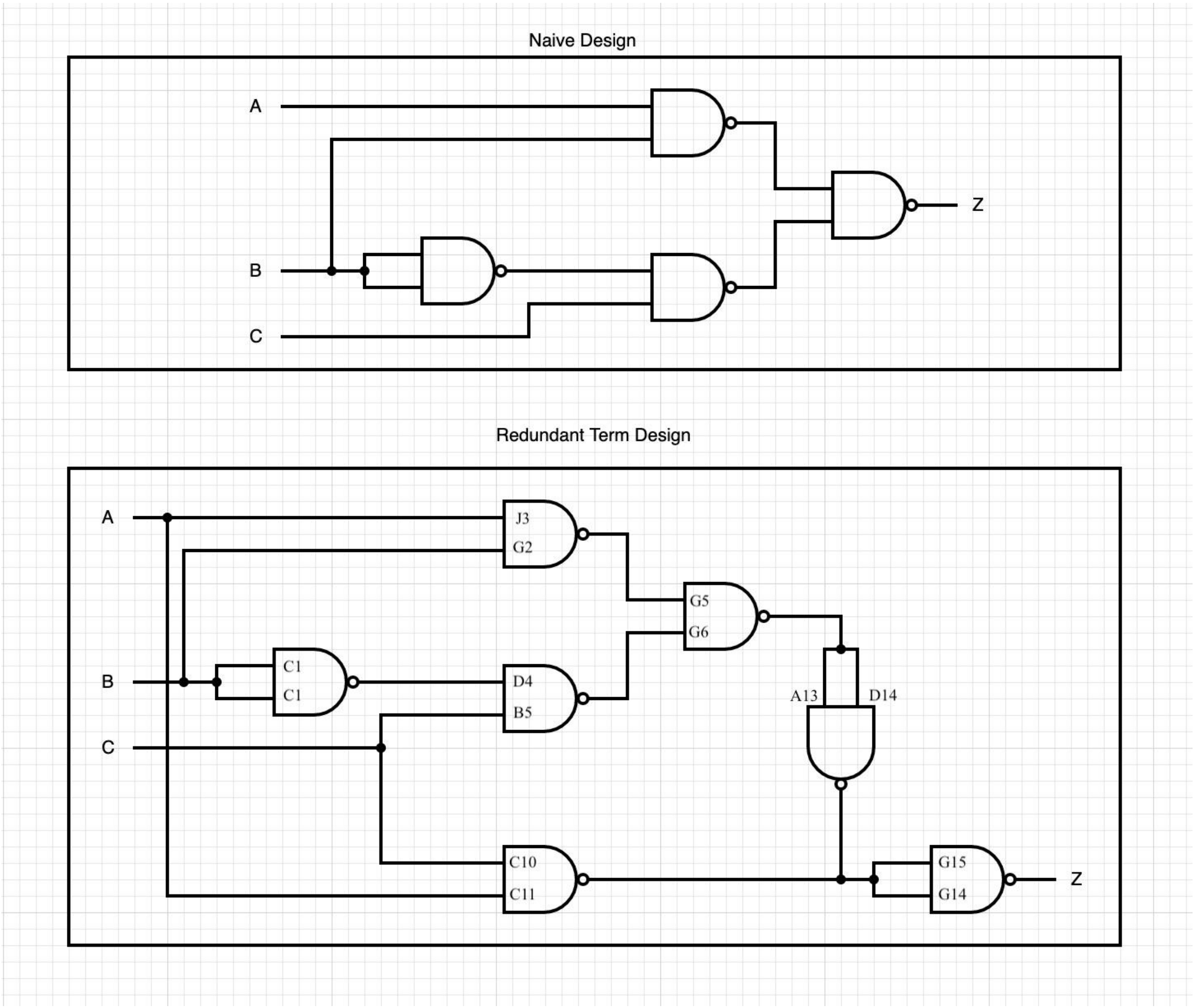
In order to make this circuit, I simply used an extra 7400 chip for the 3 new NAND gates needed in addition to the original 2 input MUX circuit. The effect of this additional AC term is shown below in the oscilloscope display:



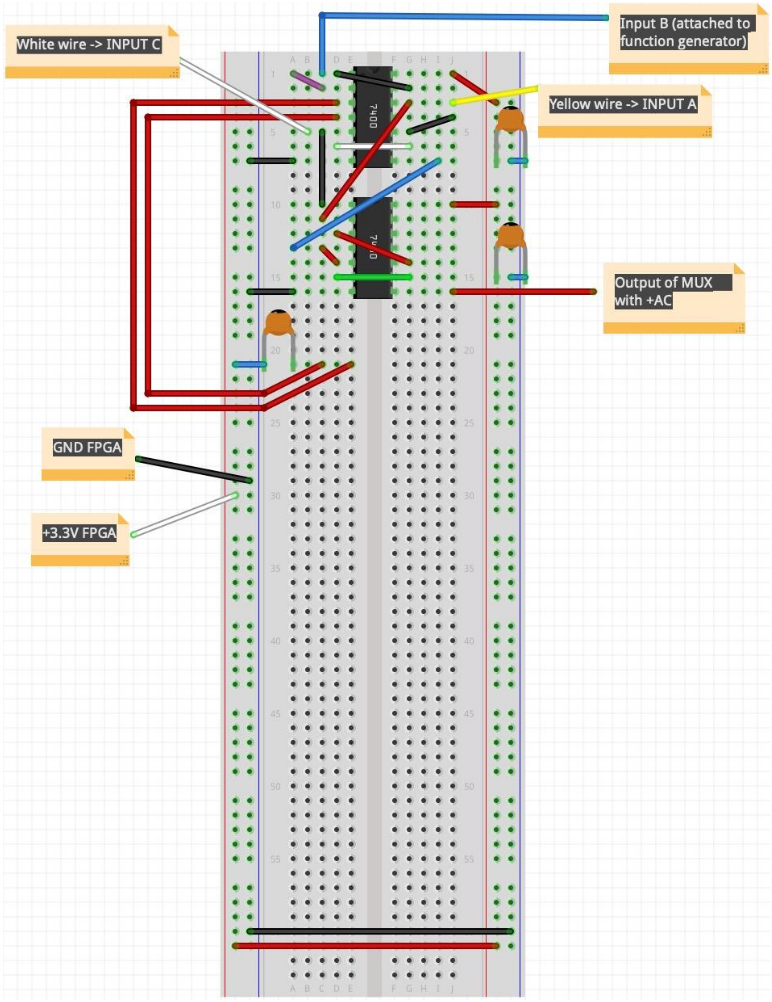
M 2.00µs Delay -80.00ns
f = 3.02834kHz
Sa 500MSa/s
Curr 14kpts
Edge CH1 DC
L 160mV
1 DC1M 2.00 V/div 3.92 V
2 DC1M 2.00 V/div -5.32 V
TRIGGER
Freq[1]=687.7kHz
Ampl[1]=3.36V
Mean[1]=
Type Edge
Source CH1
Slope Falling
Holdoff Close

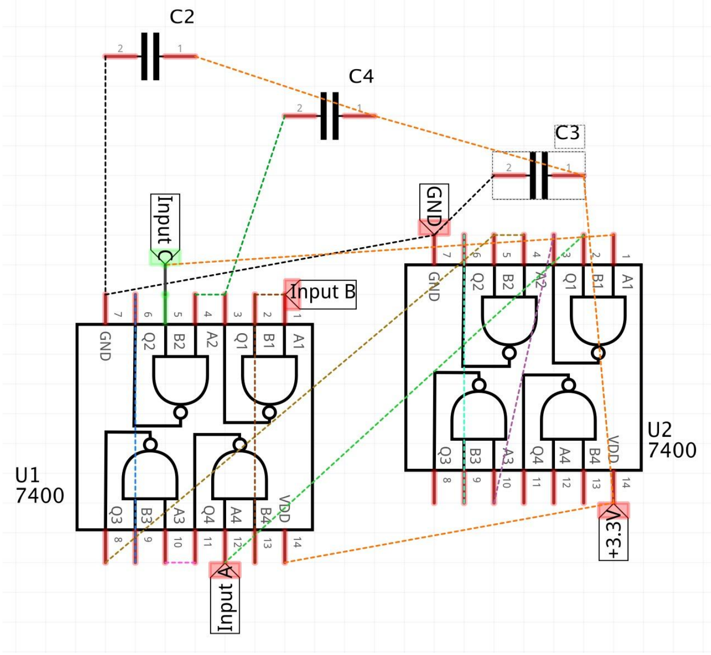
As seen, there is no more glitch (despite the presence of the capacitor at the output of the inverter)! There is of course still switching noise as the decoupling capacitors are not great, but again there is no apparent static hazard.

Logic Diagrams



Component Layout





Answers to Pre-Lab Questions

Q1: Display the three inputs and the output. Not all groups may observe static hazards, why? Not all groups (input combinations) will observe static hazards because certain changes in the inputs produce the same steady-state and delay-affected outputs, thus no glitch would be observed.

Q2: If you do not observe a static hazard, chain an odd number of inverters together in place of the single inverter from Figure 16 or add a small 1.2 capacitor to the output of the inverter until you observe a glitch. Why does the hazard appear when you do this? When a capacitor is attached to the output of a logic gate, the capacitor must charge before and discharge before reaching the next logic gate in the circuit thus resulting in a larger delay. This increase in delay emphasizes the glitch as the static hazard is drawn out for a longer period of time.

Answers to Lab Questions

Q1: Using the three input switches, test the circuit in part B of the pre-lab. Complete a truth table of the output. Does it respond like the circuit of part A?

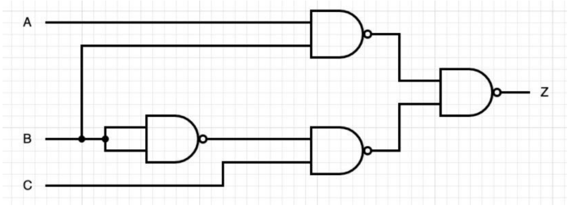
PART B: TRUTH TABLE

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

The truth table only shows the steady-state output of the circuit which are the same for part A and part B.

Q2: Next, disconnect the switch from input B and hook the pulse generator to input B. With inputs A and C high, observe input B and the circuit output on the oscilloscope. Describe and save the output and explain any differences between it and the results obtained in part 2. The oscilloscope in part A displayed a glitch in the output when input B changed from HIGH to LOW corresponding to the square wave generated by the function generator. Now however, after adding the redundant AC term to the circuit, the oscilloscope displayed no glitch in the output. Both parts, however, displayed the same amount of static noise because the changes made from part 2 (the original circuit) had no effect on the decoupling capacitors which are responsible for reducing static noise in the output.

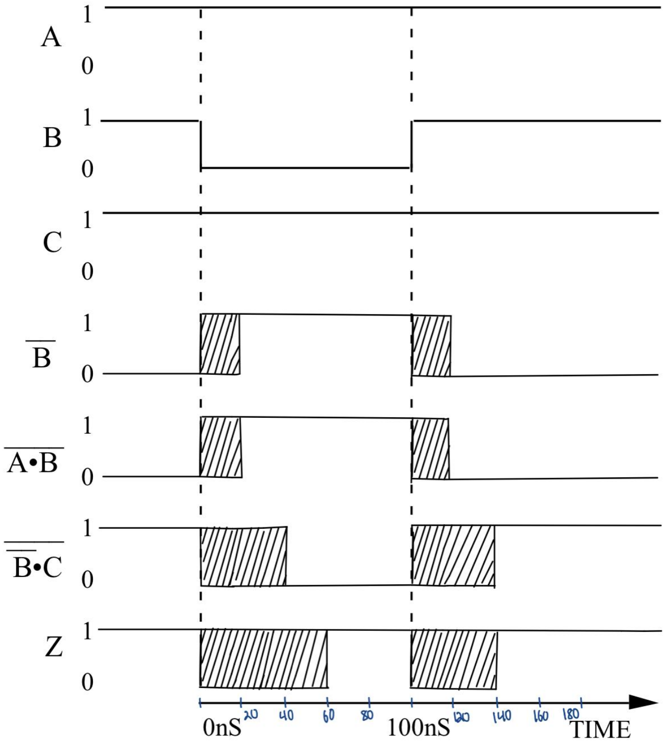
Q3: Consider the following question and explain: for the circuit of part A of the pre-lab, at which edge (rising/falling) of the input B are we more likely to observe a glitch at the output? For the circuit of part A of the pre-lab we are more likely to observe a glitch at the output on the falling edge. This is because on the rising edge, the output of the uppermost NAND gate is a 0, thus the delay caused by the inverter produces no effect on the output since Z is a guaranteed 1. On the falling edge, the uppermost NAND gate outputs a 1, thus the output (Z) is dependent on the output of the lower NAND gate preceding the last one. In this case, the delay makes it so that the lower NAND gate takes a 1 from C and (due to the delay from the inverter) a 0 from the inverter, outputting a 1 and momentarily causing Z to drop to 0. After the delay, the lower NAND gate takes in a 1 from C and a 1 from the inverter thus causing Z to rise back to 1.



Answers to Post-Lab Questions

Q1: 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.

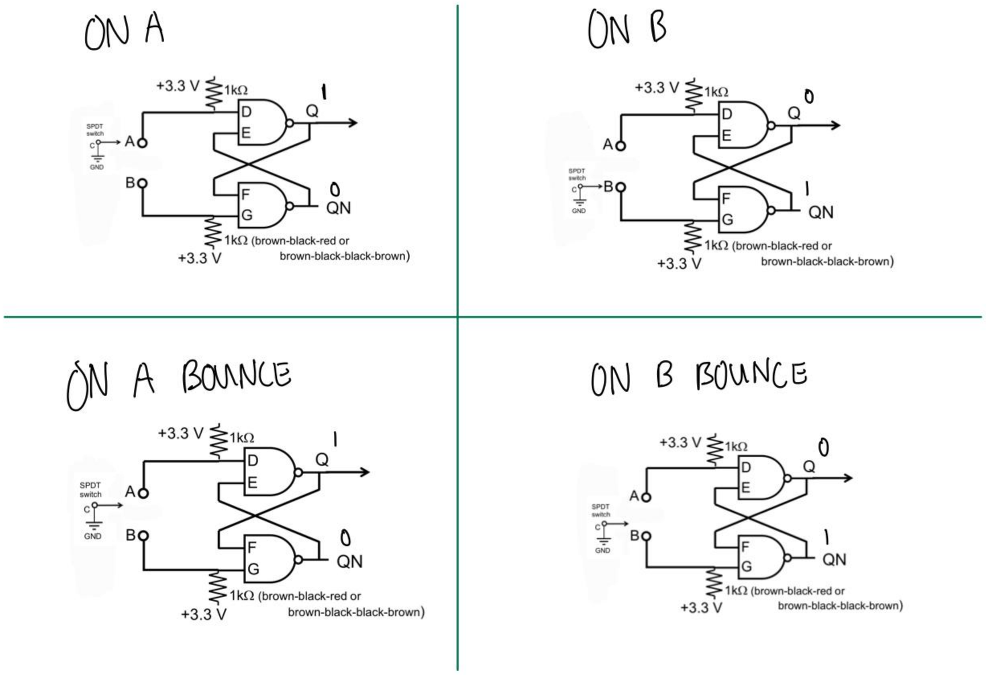
TIMING DIAGRAM



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 output Z to stabilize on the falling edge of B; It takes 40ns for output Z to stabilize on the rising edge; Yes there are potential glitches in the output Z particularly after the falling edge of input B. This is due to the delay of the inverter subsequent to input B which causes the output of subsequent logic gates dependent on it to momentarily output a different value than intended thus affecting the overall output of the circuit (Z) momentarily until the enough time passes such that the logic gate outputs have been corrected.

Q2: Explain how and why the debouncer circuit given in General Guide Figure 17 (GG.32) works. Specifically, what makes it behave like a switch and how the ill effect of mechanical contact bounces is eliminated?

As shown in the image below, when the switch bounces on A (assuming the switch began in the middle thus dra, the output changes to 1 and stays at one due to the introduction of the upper NAND gate and the feedback loop which causes the output of the switch to be dependent on the previous output. This feedback loop which resembles an SR latch essentially causes the output to be stored as a set when the terminal corresponding to it is grounded thus keeping the output stable when a mechanical contact bounce (which is not enough to move the switch all the way to the other terminal) occurs. For example, when the switch first lands on A and D is drawn to ground (0), the output Q is 1; when the switch bounces off of A, D is drawn to 1, however the previous output of Q (1) causes the lower NAND gate to output a 0 which is fed back into the upper NAND gate, thus keeping the output of Q at 1. The same situation occurs when the switch bounces on B. This shows that the ill effect of the mechanical contanct bounces is eliminated by the debouncer circuit.



Conclusions

In this lab, I learned about static hazards, mechanical bouncing, delays, glitches, the ill effect they may have on logic circuits, and how to prevent these effects. To prevent glitches, I learned that adding terms (and corresponding logic) that cover all adjacent minterms in a given circuit strongly prevents static hazards and the possibilities of glitches in the output of logic circuits. I originally thought that you could simply add various logic gates such that the delays all balance out, however, I later discovered that this method would require more materials and would be hard to coordinate since different gates have different delays. I also learned that switches bounce mechanically when switched on or off which may cause momentary effects on the output of a given circuit/component. This is especially prevalent when it comes to clock signals since the mechanical bouncing can cause excessive rising/falling edges and may significantly affect the output of a circuit that is dependent on the clock signal. To prevent this problem, I learned about debouncer circuits and how they behave like a switch without the negative effect of mechanical bouncing. Lastly I learned about CMOS logic and the importance of decoupling capacitors when it comes to reducing static noise in the output of CMOS circuits (momentary short circuits between VDD and GND in the CMOS logic when the logic gate is switching).