ECE 385

Spring 2023

Experiment 1

Introductory Experiment

Matthew Wei
NL
Nick Lu

Purpose of Circuit

This lab demonstrated a glitch that can occur in digital circuits, specifically static hazards. Occurring as a result of gate delay, they can cause spontaneous errors in the circuit's output. This lab showed an application of when it could happen as well as specific methods to fix the error.

Written Description of Circuit

The circuit built is a basic 2-to-1 multiplexer using quad 2-input NAND gates (7400). Multiplexers are components that have multiple inputs routed into them with an option to select which input to go through. This 2-to-1 multiplexer will have two possible outputs given from its input. The two inputs are A and C with B being the select and Z as the output. Below is a diagram of the circuits pre-NAND layout, truth table, K-Map, and SOP Boolean function.

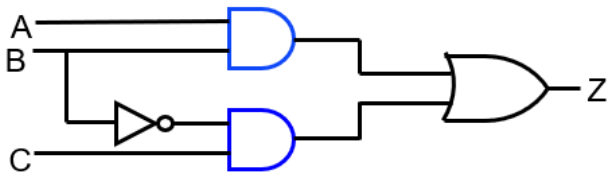


Figure 1: Logic diagram of 2-to-1 multiplexer

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

Figure 2: Truth table of 2-to-1 multiplexer

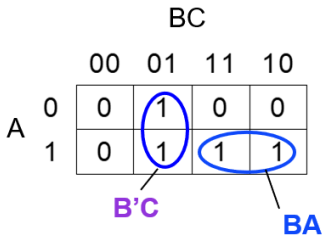


Figure 3: K-Map of 2-to-1 multiplexer

Figure 4: SOP of 2-to-1 multiplexer

$$Z = B'C + BA$$

This implementation is before the NAND gate conversion. Since this lab is implementing this design with only 2-input NAND gates, the circuit has to be converted over. To convert it to a NAND gate implementation, change each gate to its corresponding NAND gate representation by using DeMorgan's Law. Refer to figure 5 for a visual representation.

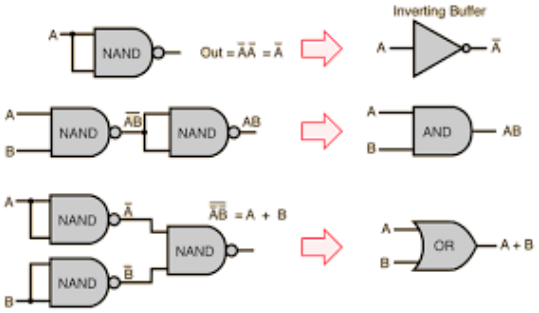


Figure 5: NAND conversion

Once converted, some simplifications can be further done to reduce the amount of gates required to implement this design. Notice the sections where there are two NAND gates back-to-back being fed with a single input. This is essentially inverting the input and then inverting it again, so setting it

back to its original value. There is no need for these gates and removing them will get rid of some unnecessary gates. This will also put the total gate needed at four, which perfectly fits into one chip

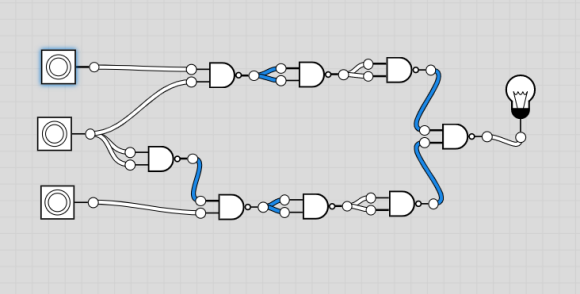


Figure 6: Pre-NAND simplification

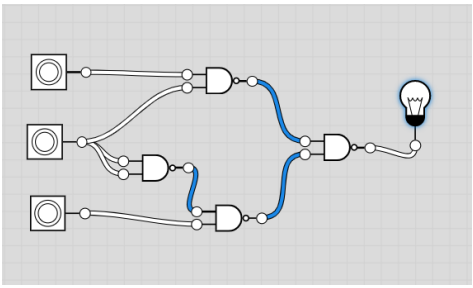


Figure 7: Post-NAND simplification

With the circuit shown above, there is a static hazard present caused by the gate delay of the inverter. This temporarily causes an incorrect value to be outputted. One way to fix the static hazard is to include its non-essential prime implicants. The current equation used for the circuit only includes essential prime implicants. Change this equation so that it includes non-essential prime implicants as well. One easy way to do this is to look at its K-Map. Non-essential prime implicants are the terms that are adjacent to the currently circled terms. In our case, this will be expression AC (refer to figure 9). Doing this will ensure that static hazards won't occur in our circuit.

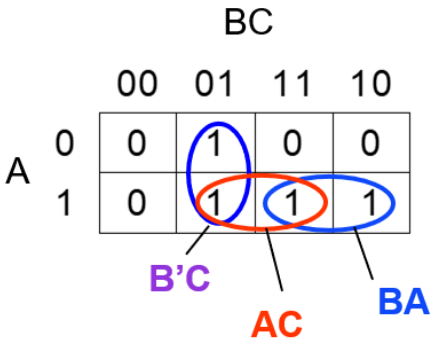


Figure 9: New K-Map

$$Z = B'C + BA + AC$$

Figure 10: New SOP

Using the conversion method to NAND gates as before, the new circuit can be built with 7 two-input NAND gates. For the design, refer to the next section.

By inputting signals and reading the output waveform, confirmation of the removal of the static hazard can be done. Below is data of the outputs before and after the addition of the non-essential prime implicant. The non-existence of the sudden dip in output is the evidence of the removal of the static hazard.

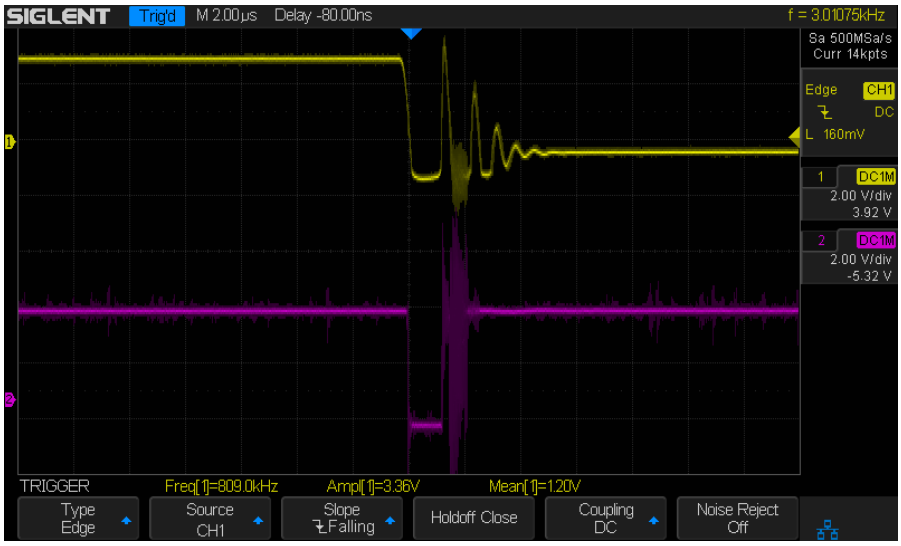


Figure 11: Before static hazard removal

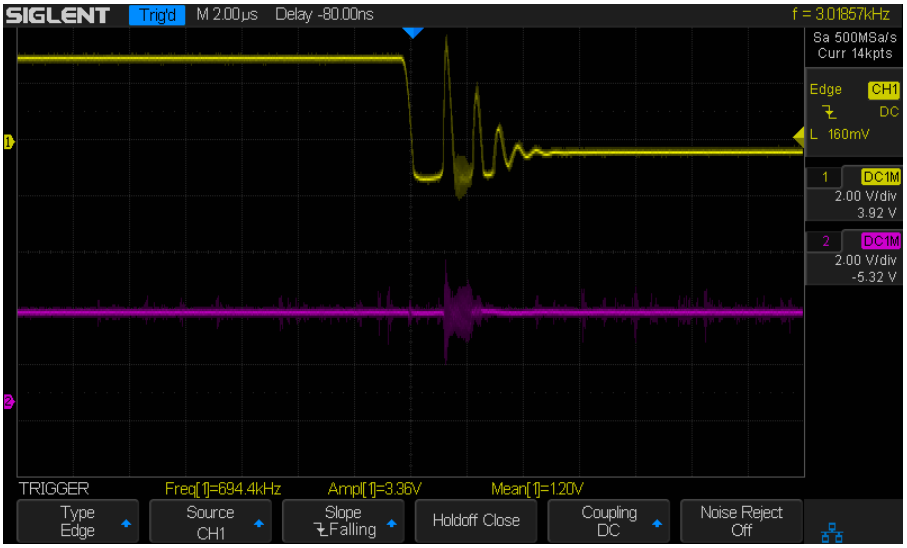


Figure 12: After static hazard removal

With this, the static hazard has been resolved. For the final design's digital logic diagram, refer to the next section.

Logic Diagrams

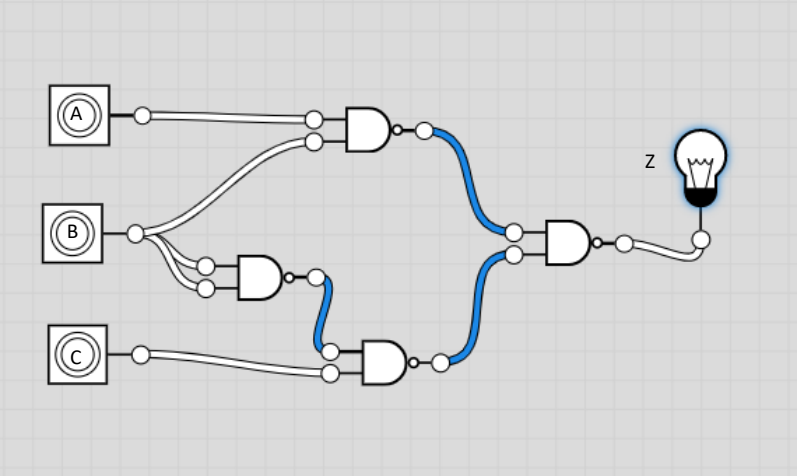


Figure 13: First digital circuit

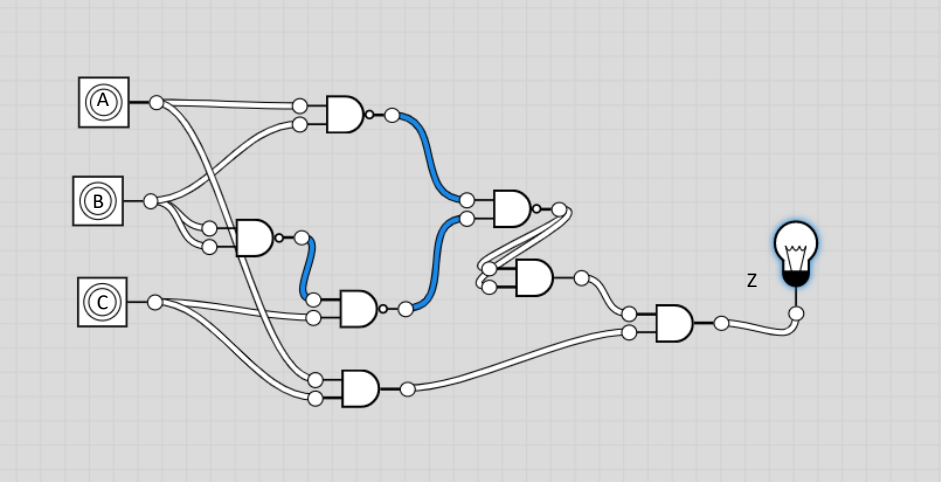


Figure 14: Final digital circuit

Component Layout

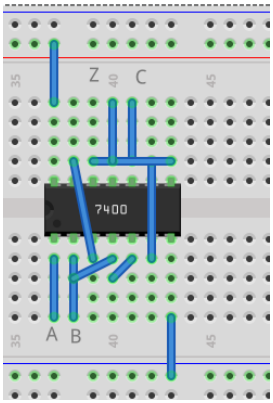


Figure 15: Before-removal breadboard layout

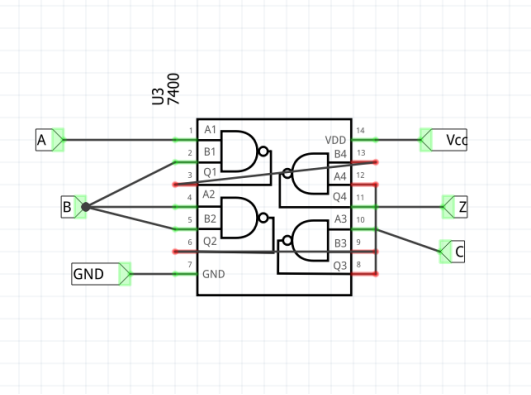


Figure 16: Before-removal component layout

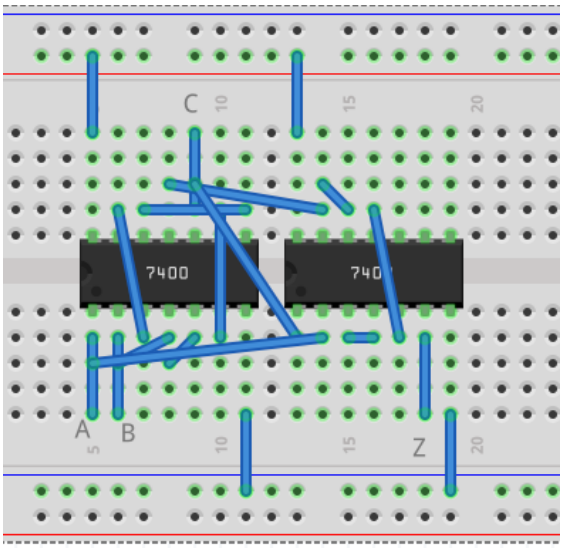


Figure 17: After-removal breadboard layout

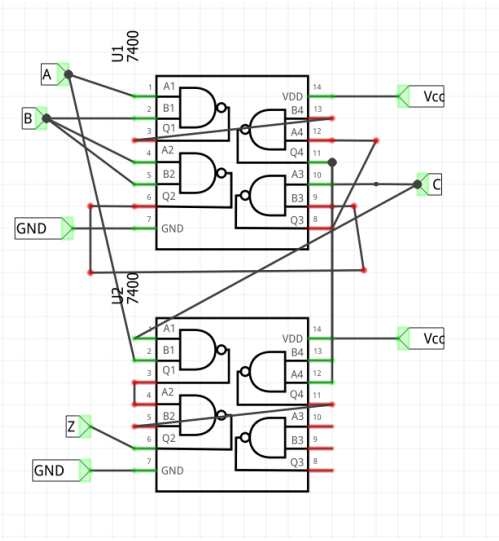


Figure 18: After-removal component layout

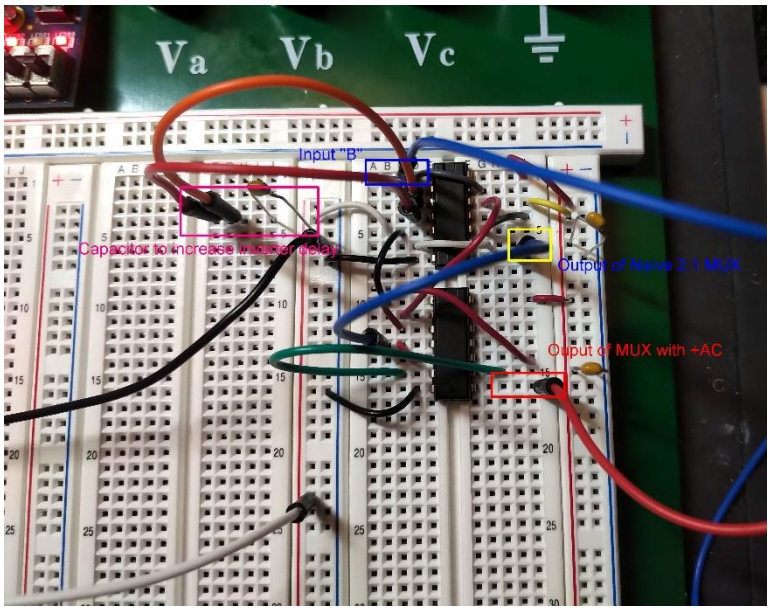


Figure 19: Breadboard layout

Pre-Lab

Some groups may not experience a visible gate delay as each chip is different. Gate delay is not a variable that designers try to set at a specific value, so it won't always be the same per chip, thus resulting in different results for some groups.

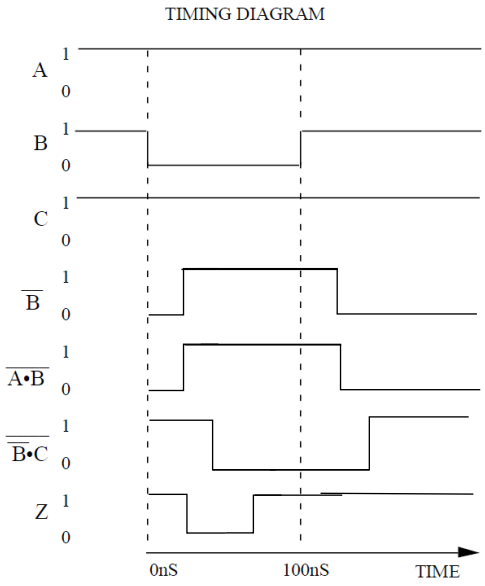
Increasing number of inverters before the logic goes into the gate will increase the number of gates it will have to go through. This will result in more gate delay as each one of the gates will have some delay that it will add on. Using an even number of inverters will simply result in its original value. An odd number will use of the circuit's correct logic.

Adding a capacitor also will increase the gate delay as capacitors require to be charged over time. This need to charge/discharge the capacitor will add sometime before the value can be read as high or low, resulting in a delay in time.

Lab Questions

The glitch would happen on both the rising edge and falling edge. This is because both would have to go through the inverter which is what causes the static hazard.

Post-Lab Questions



The falling edge stabilization time is 60 ns. The rising edge stabilization is 0 ns as A and C are kept constant. There are potential glitches in the output Z as shown in the diagram. The glitch causes Z to be the wrong value on certain inputs because of the gate delay in the IC chips.

Mechanical switches cause glitches while switching due to contact bounce. The debouncer circuit ensures a clean transition when going from high to low or vise versa by utilizing SR latches. It does this by retaining what the value was before the switch, so that the glitch doesn't occur.

Conclusions

Throughout the process of this lab, I have learned the presence of static hazards and glitches that may occur in digital logic circuits due to unideal components. Before while learning the content for digital logic, it was assumed no delay was present. However realistically, these delays will be present and can lead to bugs and glitches. Not only that, but I learned of methods to get rid of these glitches.

Because this lab wasn't done personally, most of it worked out and not much didn't work out. However, I will say creating the circuit schematics and component diagrams took a long time and they came out looking rough. I hope that I will get better at creating these diagrams and making them

look more professional. Overall, I thought this was a good lab with easy-enough content to introduce students to the whole lab process to writing the lab reports.