ECE 385

Experiment 1

Lab Report

Devul Nahar

danahar2

1/21/2022

Purpose

The purpose of this lab is to design a 2 – 1 MUX using only NAND gates. In the real-world NAND gates require far fewer transistors as compared to other gates. Therefore, NAND gates remain a default choice when designing circuits of all kinds. The first part of this experiment is going to be to use DeMorgan's laws and basic chip design to construct 2 – 1 MUX using NAND gates. In doing so we will realize a very common glitch in circuit design known as the static-1 hazard. This hazard occurs due to small-time delays as signals pass through gates. We will then modify our design by adding additional logic to create a glitch-free design. The lab is useful in the context of processor architecture as it teaches us what real-world problems arise when it comes to circuit designing and how we can fix them.

Description, High Level & Logic Diagrams

A 2 – 1 multiplexer uses a control signal C to select between two arbitrary signals A (when B = 1) and C (When B = 0). The K-map for a mux is:

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

Figure 1 – Kmap for a 2 – 1 MUX (GG.26)

From this K-map we get the minimal SOP expression:

$$Z = B'C + AB$$

The circuit equivalent of this expression is:

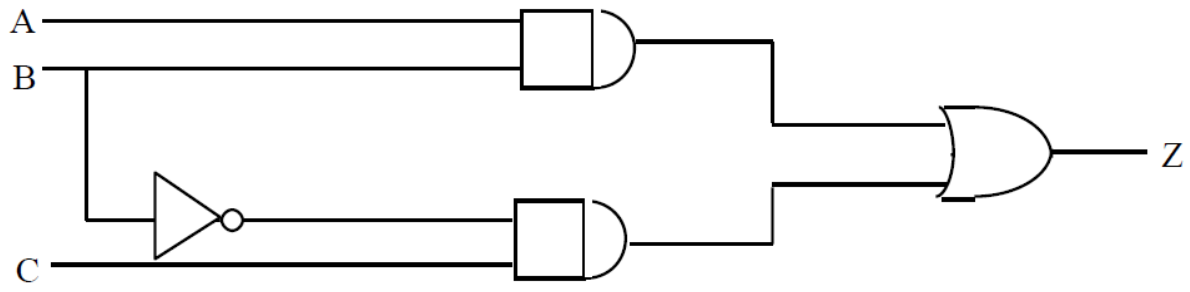


Figure 2 – SOP implemented in a circuit (GG.26)

As stated in the purpose statement, it is often best to convert all of our circuit designs into NAND form for space efficiency reasons. DeMorgans laws are used as a tool to convert the or gate into a NAND gate. Our final NAND-NAND circuit looks like this:

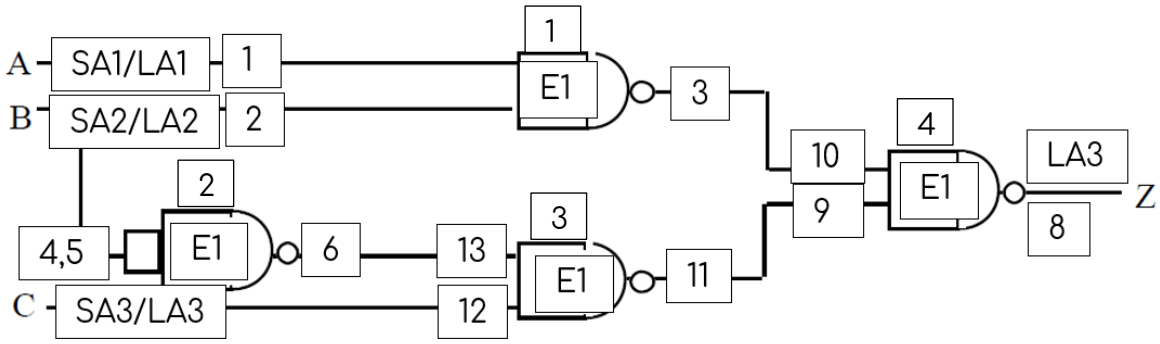


Figure 3 – Final implementation using NAND-NAND (GG.27) (part A)

This entire implementation can now be completed using only one 7400 Quad 2-input NANDs, thus making it a far superior implementation. The signals A, B, and C are connected to the following pins in the chip:

Logic pins	Logic description
A	Input 1
B	Select Signal
C	Input 2
Z	Output

Note that if the output of this circuit is connected to an oscilloscope, then you observe the static-1 hazard. For instance, if A and C are 1, and B is changed from 0 to 1, then because of the differences in the time delay caused by gates, for a brief period the output would drop to 0 even though it should always be 1. Sometimes this hazard may not be as apparent, so to make the static-1 hazard more obvious we can increase the time delay by adding a capacitor or inverters in series.

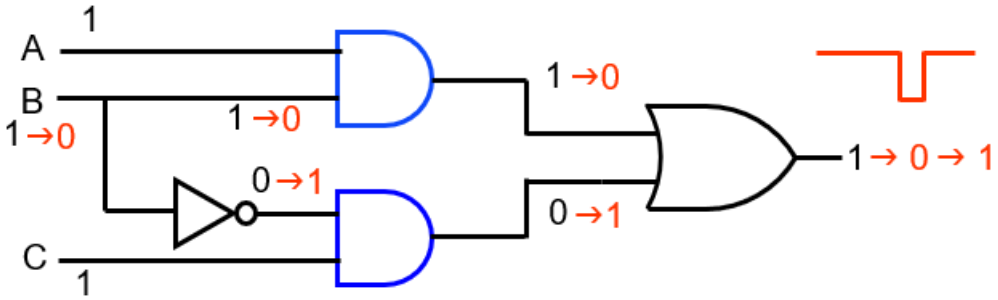


Figure 4 – Static-1 hazard (Powerpoint)

*Note: This would be similar for the NAND – NAND circuit we described earlier.

To eliminate the possibility of static-1 hazard or glitches, we add another term adjacent to both the minterms represented by the previous minimal boolean expression. Though this adds redundancy, this design prevents glitches from happening. Our final Boolean equation and circuit that prevents static-1 glitches is:

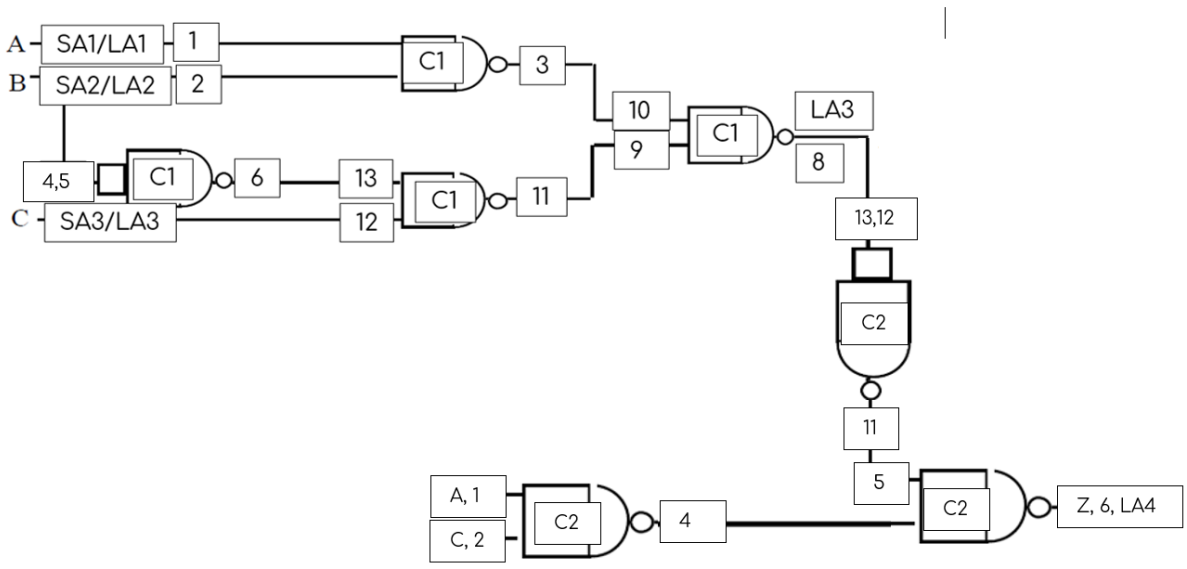


Figure 5 – Static-1 hazard free circuit design (Powerpoint) (part B)

The circuit design and modular breakdown of this lab are already described and presented above. This section shows the Fritzing schematic diagrams for more clarity when prototyping on the breadboard.

Fritzing Part A:

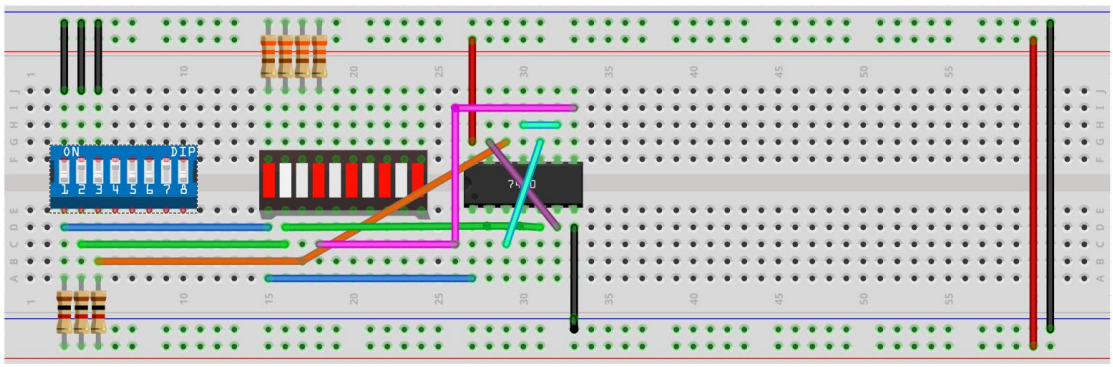


Figure 6 – 2-1 MUX implementation

Fritzing part B:

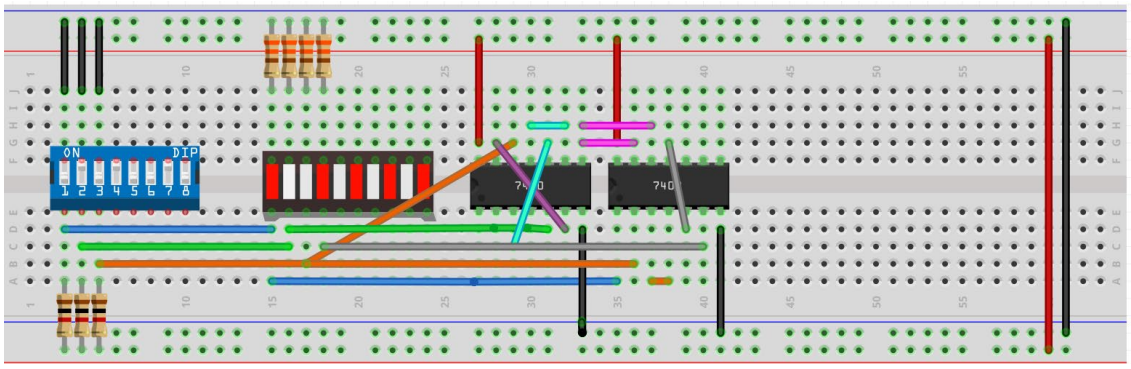


Figure 7 – 2-1 MUX implementation without static-1 hazard.

Component Layout

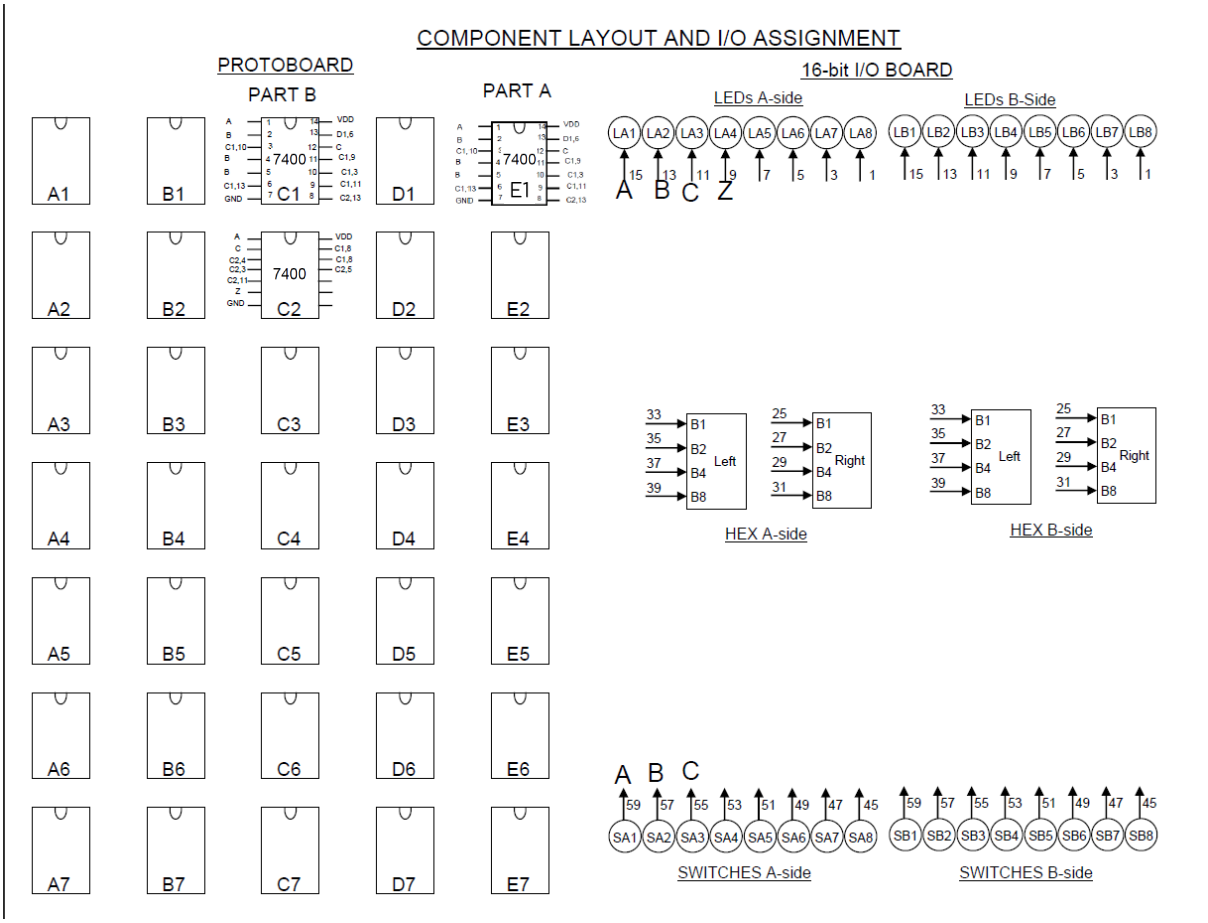


Figure 8 – component layout

Prelab, Lab, Post lab, General Guide Questions

Prelab Questions:

1) Not all groups may observe static hazards (why?)

A) Not everyone will observe the static hazard since the gate delay is quite small (a few nanoseconds). In addition, there is no guaranteed exact time delay for any gate, therefore some people's circuits may have much lower time delay as compared to other groups.

2) Why does the hazard appear when we use a capacitor?

A) By adding a 1uf capacity we increase the gate delay on the inverter and thus observe a glitch on the scope.

Lab Questions:

1) Does the part B circuit respond like the part A circuit?

A) Yes, the circuit in part B responds in the same way that the circuit in part A performed, with the key difference being that there are no longer any glitches in the outcome. But there is still some noise detected in the part B circuit, which was also present in part A.

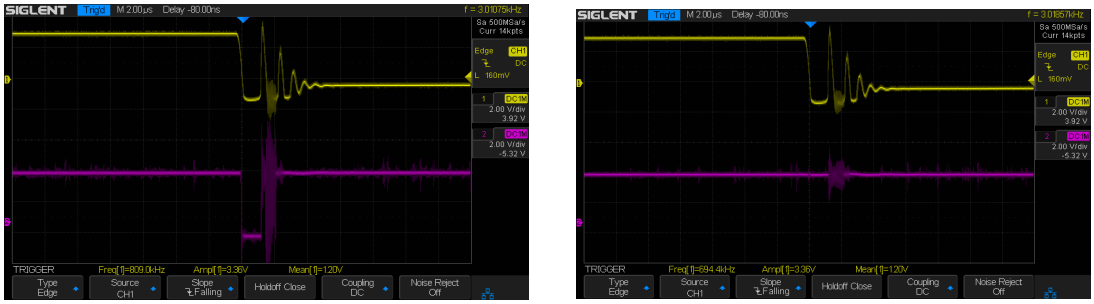


Figure 9 – the oscilloscope showing the glitch in part A circuit, and no glitch in part B circuit

2) Complete the truth table of the output

A)

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

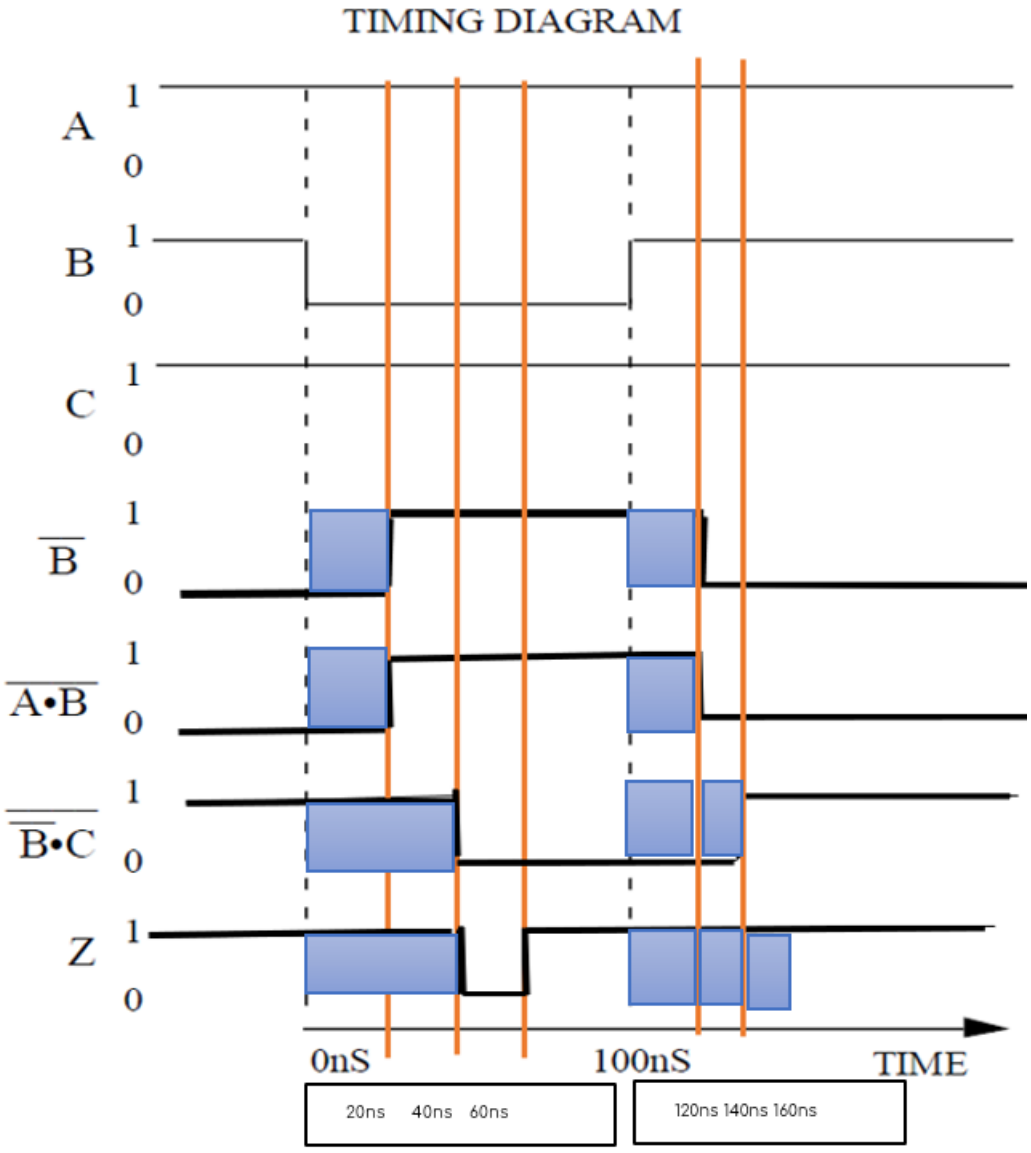
3) For the circuit of part A of the pre-lab, at which edge (rising/falling) of the input B are we more likely to observe a glitch at the output?

A) From my observation, the static-1 hazard happens at the falling edge of input B. This is because when B goes from 1 to 0, B and B' will both be 0 for a short amount of time (due to the short time delay caused by the inverter) leading to the next set of NAND gates to briefly output a 1 at the same time causing a glitch in the final output. On the other hand, when B goes from 0 to 1, the output of the NAND gates will therefore always contain at least one zero and there no glitch will occur in the final output.

Post Lab Questions:

1) Complete the timing diagram:

A)



2) 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?

A) Note that the output of Nand gates 1 and 2 (see Figure 3) take about 1 gate delay (max 20ns); the output of Nand gate 3 takes about 2 gate delays; and finally, the output of Nand gate 4 takes 3 gate delays because of all the circuitry that came before it. This means that whenever there is a change in A, B, or C there will always be uncertainty in what Z is for 60 ns. In other words, the output Z takes both 60ns to stabilize at both the falling and the rising edge of B, where the potential glitch could occur between 40-60ns (as the glitch occurs because of delay due to the immediate gate before it).

3) Explain how and why the debouncer circuit is 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?

A) One of the problems of using a mechanical design for a button in a digital circuit is that when the two plates that make up the button slam together it causes them to momentarily separate multiple times before coming to a rest (called contact bounce). Due to the sensitivity of digital logic, those momentary separations are recorded and thus affect the functioning of the circuit as planned. One way to eliminate this is by using a debouncer circuit. Figure 17 of G.G is one example of such a circuit. This circuit ensures a clean transition between 0 and 1. One of the ways it does this is by connecting an SR latch with two pull-up resistors connected to 3.3V. By doing this we ensure logical 1s and 0s as inputs to the SR latch. More specifically, when the switch is in position B, D will be 1, G will be 0, and thus QN will be 1 irrespective of what Q is. Now, since QN is 1 and A is 1, Q will be 0. Now, if the switch is moved to position A and in doing so is suspended in the middle for a while, both D and G will be 1 (as they are pulled by the resistor), but Q will not change as QN is still 1. Finally, when the switch touches position A, D will be 0 and G will be 1. This means that Q will be 1 irrespective of what QN is. Any bouncing that occurs at position A will not affect the output Q, and so things will remain constant until the next switch operation is performed. This is how the contact bounce problem is eliminated by using an SR latch.

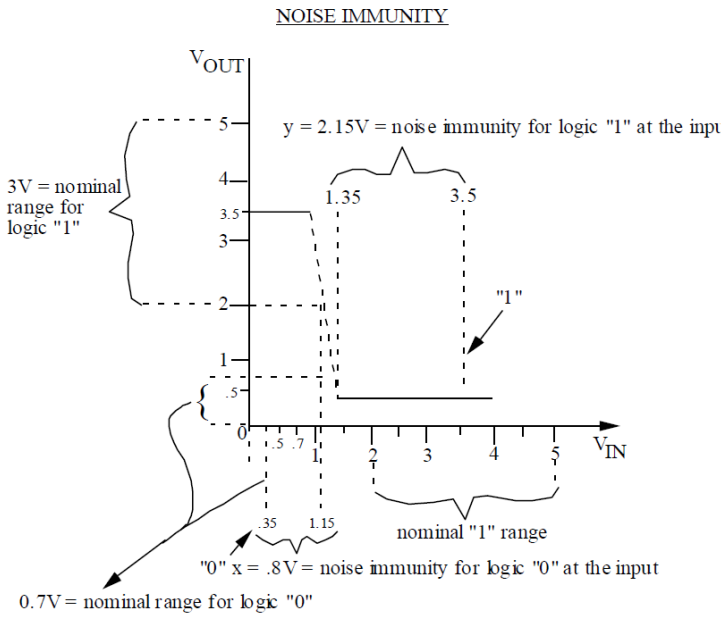
General Guide Questions:

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

A) Having a large noise immunity is highly beneficial as the larger the noise immunity the more immune the logic in a digital circuit is to the effects of noise.

2) Why is the last inverter observed rather than simply the first? Given a graph of output voltage (VOUT) vs. input voltage (VIN) for an inverter, how would you calculate the noise immunity for the inverter?

A) The last inverter is observed rather than the first one because we want to calculate the noise immunity. By connecting several inverters' gates, we can compare the voltage of the final inverter and the first inverter to see just how much noise immunity the inverter has. The noise immunity is given by the range for which the Vout is stable. By looking at the diagram it is clear that Vout is stable for 1 between 0.35 V – 1.15 V; this gives us a noise immunity of 0.8 V. Similarly, Vout is stable for 0 between 1.35 V to 3.5 V; this gives us a noise immunity of 2.15 V. In general, noise immunity is considered to be the smaller of the two ranges, and thus we can say that the noise immunity for the inverter is 0.8 V.



3) Why is a capacitor necessary close to each chip?

A) In CMOS chips, when a value is switched from high to low or vice versa, there is temporary short circuit between Vdd and ground, this shows up in the circuit as noise that propagates between the logic chips. The 1uF capacitor is necessary to be around the CMOS chips as it acts like a decoupling capacitor which provides just enough power for the integrated circuit to keep a stable voltage and thus reduce noise.

4) If we have two or more LEDs to monitor several signals, why is it bad practice to share resistors?

A) We should avoid multiple LEDs sharing resistors mainly because you cannot safely connect diodes in parallel. This is because, in the real world, not all diodes have the same characteristics, and therefore by placing just 1 resistor we allow diodes to control how much current goes through them. This means we run the risk of some diodes conducting much more current than other diodes and thus burning out as a result.

Conclusion

Experiment 1 gave us an introduction to glitches and hazards that may occur in circuit design. It also showed how we can sometimes improve this design by adding redundancy and eliminating the glitch in the process. This lab was generally trivial as it was solely designed to give us an introduction to how to write Lab reports and for us to revise some

concepts explained in ECE 120. However, despite this, the lab report took quite a long time to complete. In the future, I expect to complete labs quicker as I have a better understanding of the software as well as the lab requirements. In addition, I expect to get better at circuit design as I recall and learn more about the fascinating world of circuit design.

Upon further reading about avoiding hazards, it seems that we can easily avoid this issue by making the circuit synchronous where we make each clock cycle as long as the max delay of any gate. This would allow sufficient time for the static hazards to automatically resolve. This seems like a far better solution as we don't have to add additional redundant gates to avoid the hazard.