ECE385

Spring 2020

Experiment #1

Introductory TTL Experiment

Yizhen Lu

Section ABD

TA: Nicholas Cebry, Wenjie Pan

Introduction

This lab is primarily used to get students familiar with the available lab equipment and the basic frame structure of a lab experiment.
Equipment available and necessary in this lab are among I/O board switches, pulse/waveform generator, DC supply generator, oscilloscope and a lab kit with a protoboard, wires and various IC chips included.
In the process of building a TTL circuit, students should understand procedures of setting up waveform generator as signal input, standard circuit assembly manner on the breadboard, and debugging techniques involving either LEDs or oscilloscope.

Written Description of Circuit

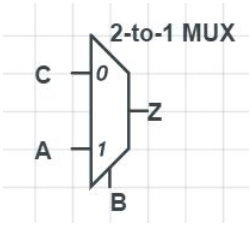


*Figure 1. Schematic diagram of 2-to-1 MUX with input **A** and **C**, select input **B** and output **Z**.*

The circuit is implementing a 2-to-1 multiplexer, hence requiring two(2) inputs, one(1) select signal and one(1) output. In the following schematics, we denote inputs as A, C; select signal as B; output as Z. Following this notation, the circuit we built is to achieve the data selector function the same as the diagram shown on the right side. When B is 0(Voltage Low), output Z follows C, whereas Z follows A when B is 1(Voltage High).
In part A of the pre-lab, we develop the minimal SOP form of the logic expression of the circuit based on the principle above:

$$Z = BA + B'C$$

This logic expression gives rise to the basic idea of the circuit we should have for part A.

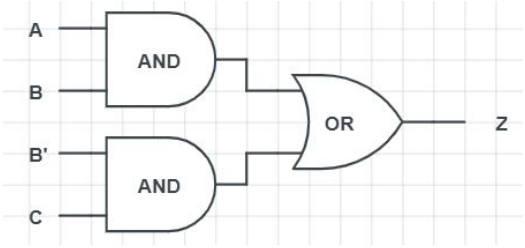


Figure 2. Direct logic diagram for part A of pre-lab from SOP form.

Then, by DeMorgan's Law, the circuit can also achieve the same function by simply replacing all AND and OR gates with NAND gates, as shown below.

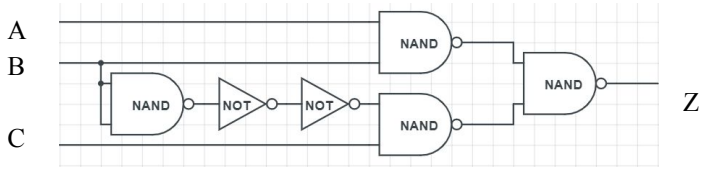


Figure 3. Modified logic diagram for part A of pre-lab from SOP form.

In the actual wiring of the circuit, we deliberately use more NAND/NOT gates to create B' signal. In this manner, the signal would be generated undergoing more time delay, which consequently makes the 'glitch' phenomenon more obvious to be observed.

In part B, we modify the circuit designed in part A by adding more components to overcome the static hazards/'glitch' phenomenon occurred during the falling-edge of signal B going from 1(Voltage High) to 0(Voltage Low), when both A and C are 1(Voltage High). The logic expression for the new circuit is as follows:

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

And the basic schematic for the circuit is shown below.

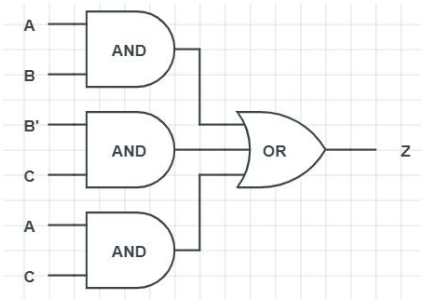


Figure 4. Direct logic diagram for part B of pre-lab from SOP form.

By adding the additional component of A AND C in the logic diagram, the 'glitch' effect can be nicely removed due to the continuing 1(Voltage High) signal provided by this extra term. Similar to part A, this diagram can also be updated to use NAND gates only at the place for all AND and OR gates, as shown below.

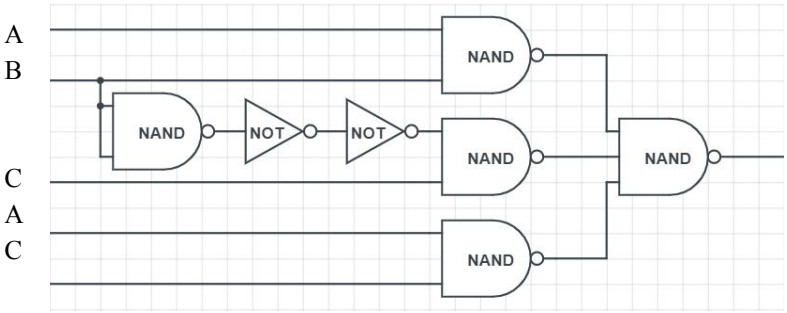


Figure 5. Modified logic diagram for part B of pre-lab from SOP form.

Component Layout

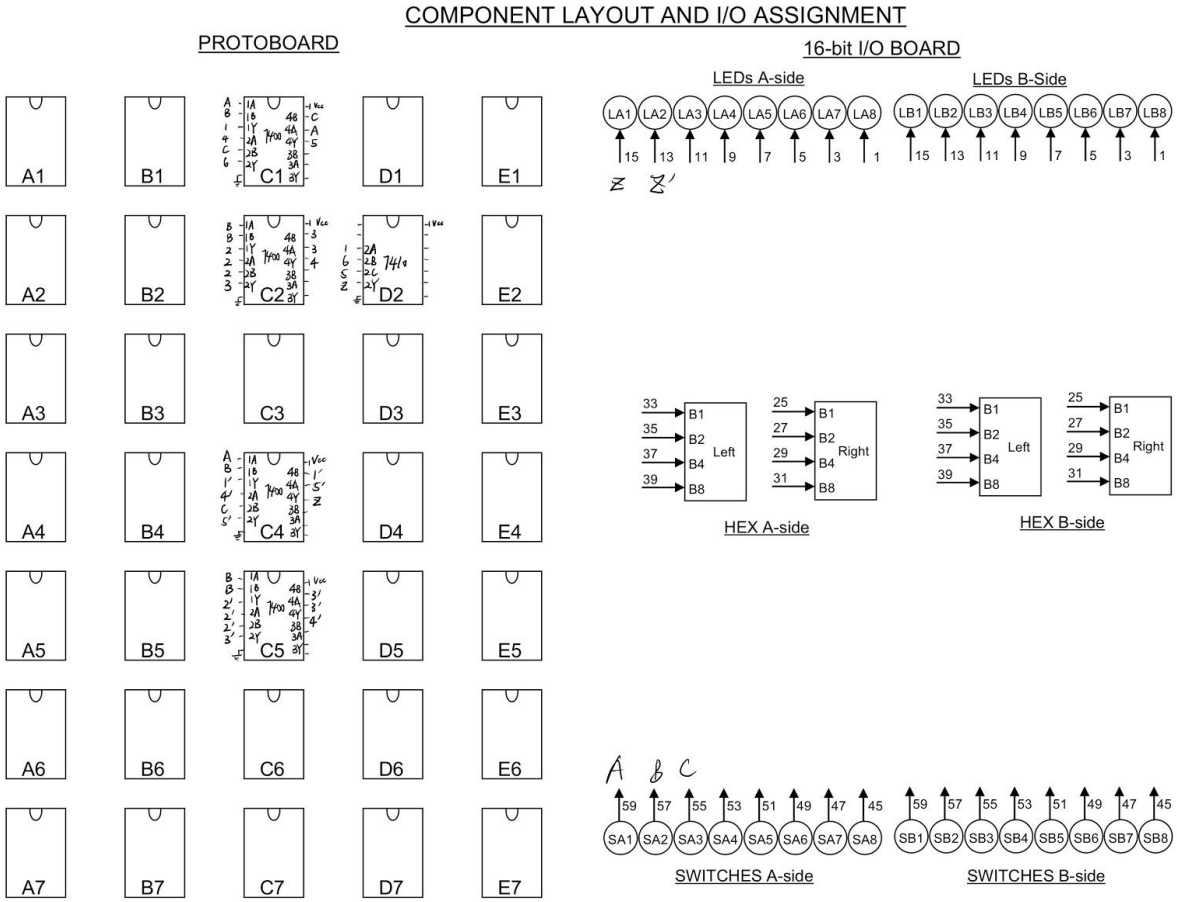


Figure 6. Component layout of lab 1 overall circuit.

Circuit Diagrams

Circuit diagram for part A of the pre-lab. As mentioned above, this design results in the presence of the ‘glitch’ effect.

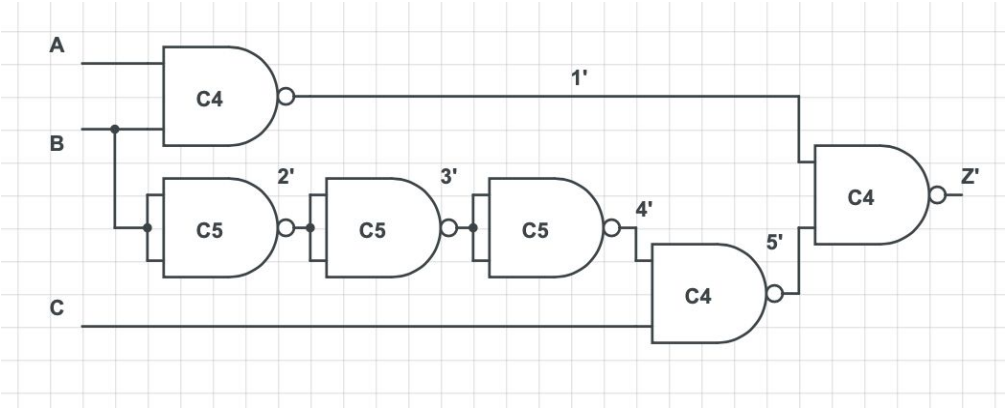


Figure 7. Circuit diagram for part A of pre-lab.

Circuit diagram for part B of the pre-lab. This modified version prevents the happening of obvious ‘glitches’.

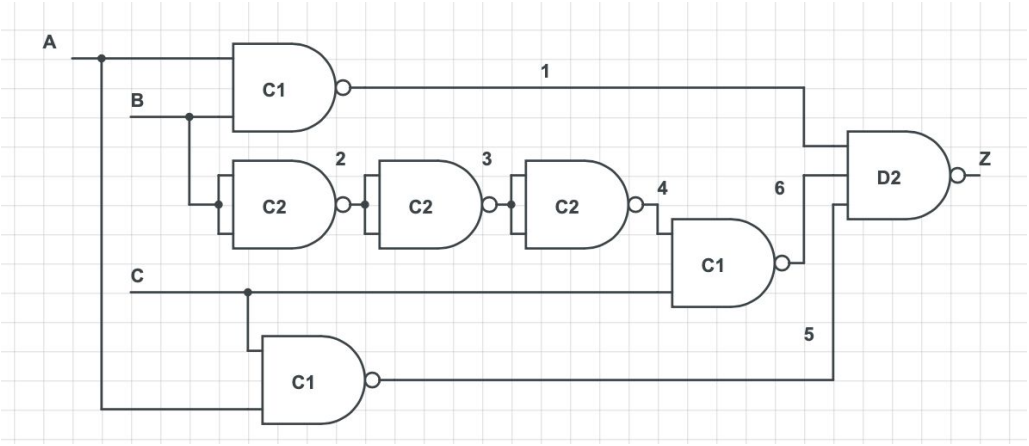


Figure 8. Circuit diagram for part B of pre-lab.

For both diagrams, the labels for each gate are placed in accordance with the location code of the chips used to implement the gates in the protoboard during the actual wiring of the circuit. And the numbers attached to each wire are also assigned matching the numbers noted on the component layout graph.

Documentation of Lab

Truth Table for Part A:

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

K-map for Part A (Z = BA+B'C):

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

From the circuit diagram and K-map for part A, we can analyze and predict the happening of static hazards.

Since the logic expression for the circuits is using the minimal SOP form, Z = BA + B'C, the transition for output Z between ABC = 001 and ABC = 101 is guaranteed to be smooth and consistently 1(Voltage High) because "B'C" term remains 1(Voltage High) throughout the process, confining the output to be in Voltage High region. The same mechanism applies to the transition between ABC = 110 and ABC = 111, thanks to the "BA" term. However, nothing is there to ensure the smooth transition between ABC = 101 and ABC = 111. And this is exactly when the static hazards, or a glitch shown on the oscilloscope, would occur.

When B switches from 1 to 0, "B NAND A" gives a 1, while since B has to first go through an inverter to generate a complement signal, while takes some extra time delay, the meantime "B' NAND C" remains 1 and no yet updated. So at this very instant, the output Z would have a 0 output until the slight time delay passes and pushes Z back to 1. This is how the 'glitch' forms. By understanding this mechanism, it is clear that how severe the 'glitch' is depends on how long the time delay for flipping signal B takes.

Thus, if a static hazard is not observed, it is because the time delay, which is taken to flip the B signal, is subtle enough for the time delay of the NAND gate. So, if an inverter is used, it usually doesn't present an obvious 'glitch' because time delay for an inverter is shorter than the time delay of a NAND gate. Oppositely, if you connect a chain of an odd number of inverters in series, or add a capacitor at the output end of the inverter, you deliberately increase the time delay, which could signify the effect of static hazards and hence making the 'glitch' easier to be observed. (In our case of implementing the circuit, we use a chain of three(3) NAND gates, each functioning as inverters to even magnify the time delay) Below is the screenshot from the oscilloscope for part A circuit when both A and C are 1(Voltage High), where the upper, yellow signal represents Z, and the lower green signal represents B, the select signal of the 2-to-1 MUX. The picture shows clear 'glitches' at every falling-edge of signal B.

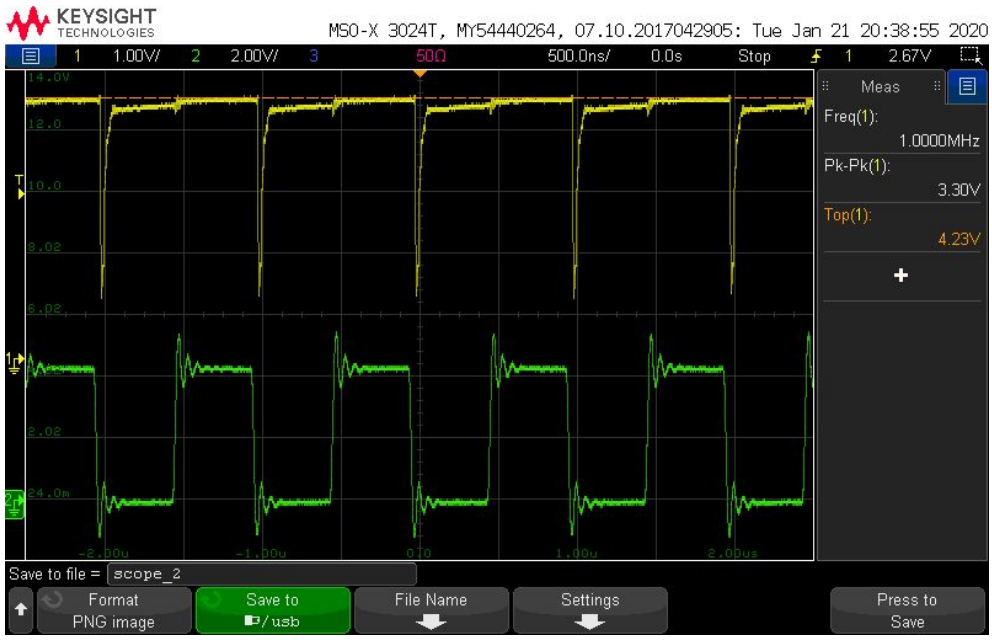


Figure 9. Oscilloscope printout for part A of pre-lab.

Thus, to enhance the performance of the circuit and to remove 'glitches'. As we pinpointed earlier, the problem occurs during the transition between ABC = 101 and ABC = 111. So the solution is simple to build some extra components to compensate and ensure the smooth transition. By adding "A NAND C", it consistently provides a 0(Voltage Low) signal throughout the transition. Therefore it restricts the output Z to be always 1(Voltage High).

The truth table and K-map for part B circuit are shown below, which are identical to its counterpart from part A.

Truth Table for Part B:

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

K-map for Part B (Z = BA+B'C+AC):

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

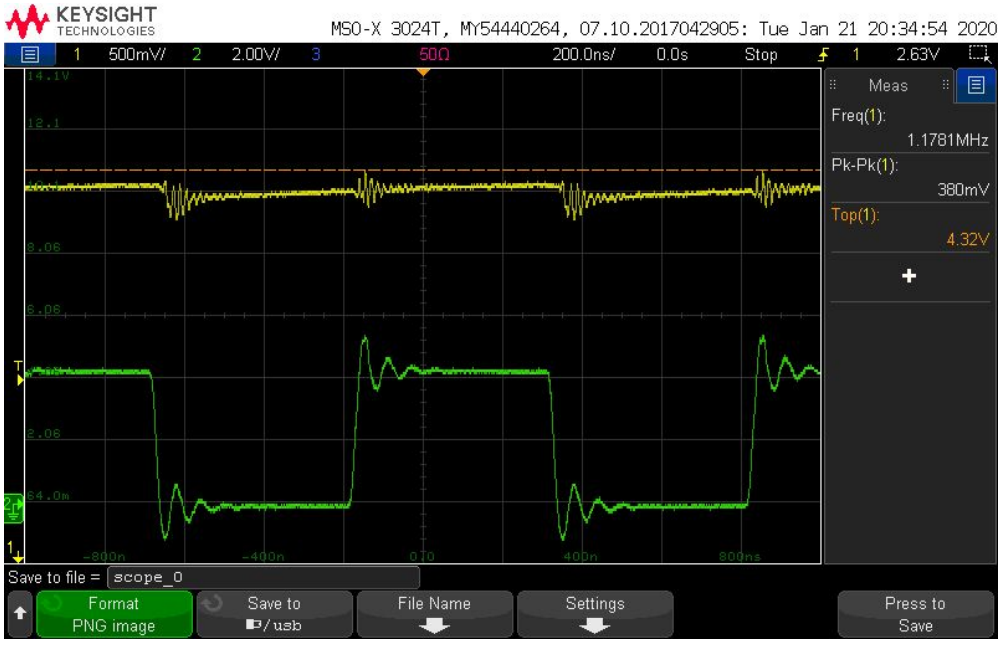


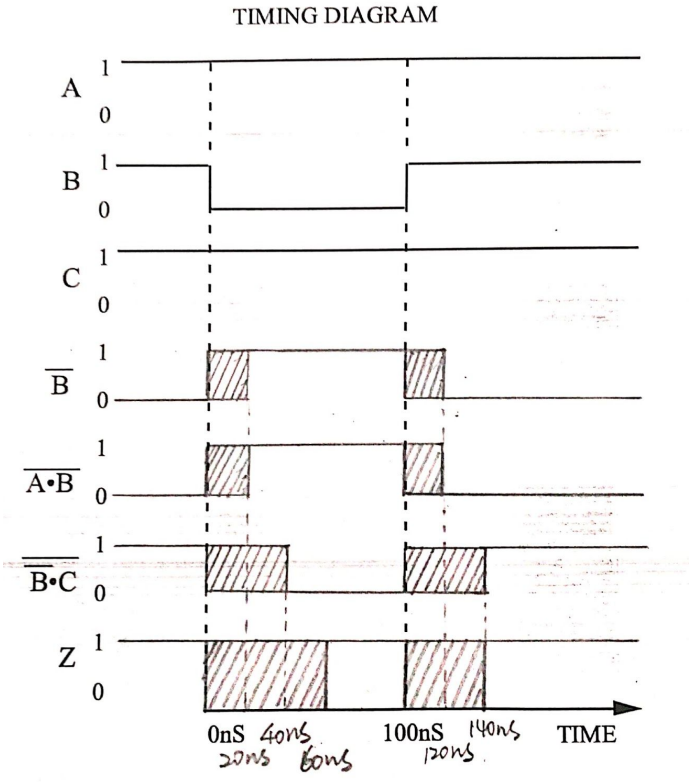
Figure 10. Oscilloscope printout for part B of pre-lab.

Picture above is the screenshot from the oscilloscope for part B circuit under the same conditions as for part A. Comparing two screenshots, it is quite clear that the huge ‘glitch’ is now reduced into small turbulences, meaning that the approach to solving the static hazards works well.

Post-Lab Questions

1.)

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. (See GG.23 if you are not sure how to proceed.)



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 Z to stabilize on the falling edge of B.

- It takes 40ns for Z to stabilize on the rising edge of B. It takes 20ns less than the falling edge case because as long as (AB)' gives 0(Voltage Low) output, the output of the NAND gate it follows has to be 1(Voltage High), and due to the internal structure of NAND gates, it already starts to process the output without requiring the output from (B'C)', this saves the entire process with 20ns and also explains why the 'glitch' on the rising edge is not as obvious as the falling one.
- There are potential glitches in the output Z. From the timing diagram above, it is clear that there are shaded regions for output Z after both the rising and falling edge of B. In the shaded area, the behavior of the circuit/ the output would be uncertain, and it is highly likely to have an instantaneous 0(Voltage Low) output, which would result in the glitch.

2.)

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?

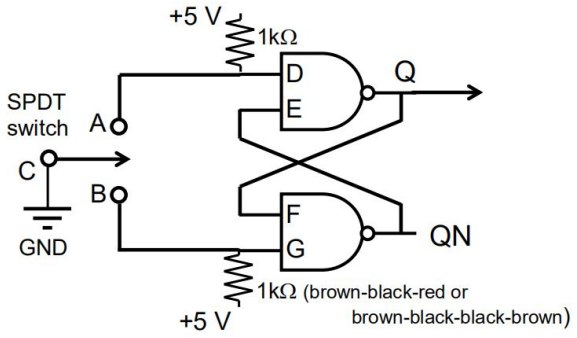


Figure 11. Schematic diagram for debouncer circuit.

The switch comprises of one(1) SPDT switch connected to one(1) SR latch using NAND gates. The idea is to use the flip-flop to eliminate the switch bounce. Considering the static behavior of the debouncer circuit, when switch node CA is connected, output Q is 1(Voltage High), whereas Q goes to 0(Voltage Low) when switch node CB is connected. So different choice from the SPDT switch changes

the output between 1 and 0. This enables the circuit to provide both 1 and 0 signals, satisfying the fundamental behavior as a switch.

Then we come to the dynamic behavior of the circuit to see why the bounces can be nicely eliminated. Assume CA is originally connected and reaches the steady state, under this assumption, we can induce signal value for all nodes:

D = 0; E = QN= 0; F = Q = 1; G = 1;

Then, as we disconnect CA nodes and before CB are connected, signal D is pulled up to 1(Voltage High) by the 5V supply gradually. However, since E = 0, the output of the upper NAND gate is confined to 1 all the time, making the change of signal D unable to affect the output. This eliminates the bouncing when disconnecting CA nodes.

Moving on, as CB nodes are connected, G is pulled down to 0(Voltage Low) with potential bounces. Even a short instant of G = 0 is loaded into the lower NAND gate, it outputs 1(Voltage High) and then the upper NAND gate would output Q = 0 due to both 1(Voltage High) inputs. As soon as Q = 0 is transmitted into F node, the 0(Voltage Low) signal would then ensure the output QN = 1 for the following output. In this manner, the bouncing during the reconnection period is greatly alleviated. Summing up, this circuit is capable of performing switch functions with all switch bounces highly eliminated.

General Guide Questions(GG.6, GG.29)

GG.6

What is the advantage of a larger noise immunity?

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?

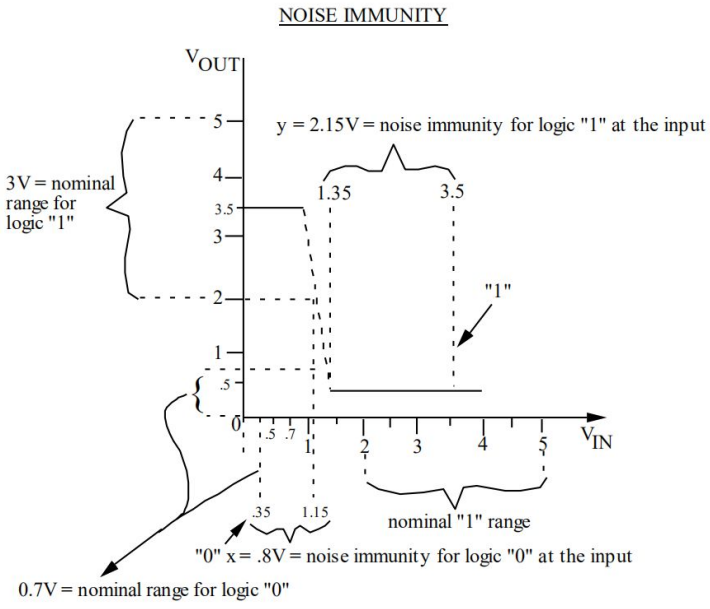


Figure 12. output voltage (VOUT) vs. input voltage (VIN) for an inverter

- Larger noise immunity is helpful when subject to input with noise turbulences. It helps eliminate the effect of noise by providing the correct level of output as if no noise is present at the input end.
- To find the nominal 1 and 0, we need to determine the level of HIGH and LOW voltage of the inverters. Due to the restorative property of inverters, as long as the input is within a reasonable range, when loaded into a chain of inverters, the voltage of the output would tend to converge to two(2) values(depending on the input and the number of inverters connected in the chain), which in this case represents the voltage corresponding to nominal logic 1 and 0. If directly measure the first inverter, it is unlikely to provide the correct result, the input may lie between the transition band of the inverter and thus provides an output that is between the voltage representing 1(Voltage High) and 0(Voltage Low).
- The noise immunity region is where is output is unaffected by changes in the input, on the graph, it is where the slope of the VOUT-VIN curve is zero, a flat horizontal line. Take the graph above as an example, we can tell that the curve has two parts of flat segments. The first one is from 0.35V-1.15V, while the second one is from 1.35V-3.5V. So the noise immunity at the 0(Voltage Low) level is 0.8V(1.15V-0.35V=0.8V). In the same manner, the noise immunity at the 1(Voltage High) level is 2.15V(3.5V-1.35V=2.15V). Taking the smaller number as the overall immunity gives the inverter an overall noise immunity of 0.8V.

GG.29

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

Connecting LEDs with resistors is to limit the current going through the LED to protect it from damage. However, if the resistor is shared among several LEDs in parallel, the current dissipated into each LED is greatly reduced. This would make LEDs show a dimmer light, harder to be observed and might cause ambiguity. Besides, different colors of LEDs possess various internal resistance, when

connected in parallel, currents may not be drawn into each LED evenly. This would damage some LEDs with low internal resistance due to the high current and making light emitted from LEDs with high internal resistance even dimmer. Also, it makes it hard to design the circuit because all LEDs are taking in different signals, they might misbehave and create great confusion if not properly designed. With all the reasons stated above, it would be a bad practice to share resistors.

Conclusions

In this lab, we make good use of IC chips like SN7400, SN7410 NAND gates and de-bounced switches to build a simple logic circuit, performing the functionality as a standard 2-to1 MUX. Also, during the wiring and testing period, we learned to use LEDs and oscilloscope to find and pinpoint where the circuit goes wrong and behaves not as desired. This greatly facilitates our debugging process. We figured out the underlying principle of the occurrence of static hazards, and took advantage of that to make the ‘glitches’ in our circuit more or less obvious by manipulating the overall time delay. The further modification eliminates the ‘glitch’ issue directly, ensuring a smooth transition where the ‘glitches’ would originally occur by adding one more term in the logic expression of the circuit. We also learned the cause of switch bounces and measures taken to reduce that effect. As for the noise present in the circuit, the noise immunity property of inverters would nicely solve the problem and leave no side-effect. So we can apply a chain of inverters to fix the imprecise signal.