ECE 385
SPRING 2024

EXPERIMENT 1
Introductory Experiment

Name: Hanjun Luo
INTL ID: hanjun.21@intl.zju.edu.cn
Section: LA3
TA: Yinan Yao

Purpose of Lab

This experiment is an introduction to the basic operations of ECE 385.
In the design of TTL circuit, due to the delay of gates, we need to deal with the occurrence of glitches in signals. In this lab I learned a method to handle the glitches. Also, as the initialization of ECE385, I learned how to use the VHDL again and how to do simulations without kits. It helps me with reviewing the concept including delays, K-maps, and logic diagrams, too.

Purpose of Circuit

For the part. A, I draw a diagram which fits the requirements. However, due to the delay of the gates, it may cause glitches.
The circuit of part. B is designed for diminishing the glitches. We know it's impossible to eliminate the delay, but by adding additional gates, we can make the output steady.

Written Description of Circuit

The part. A is a 2-TO-1 MULTIPLEXER, which is consisted of all NAND gates.
If B=0, we output the value of C, otherwise we output the value of A.
For the part. A
The K-MAP table is below:

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

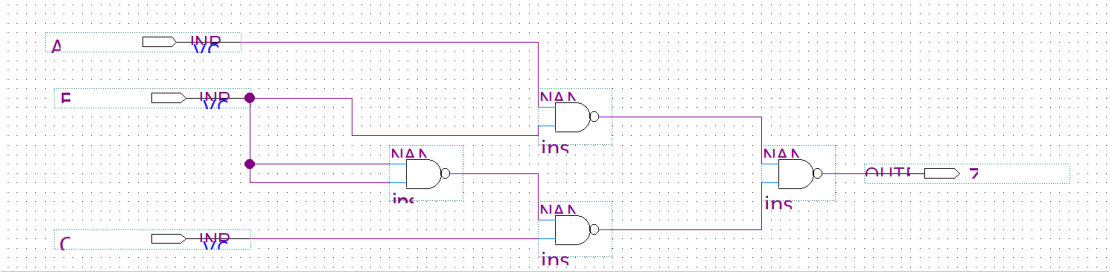
Z = AB+B'C
The K-MAP table of part. B is the same as above:

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

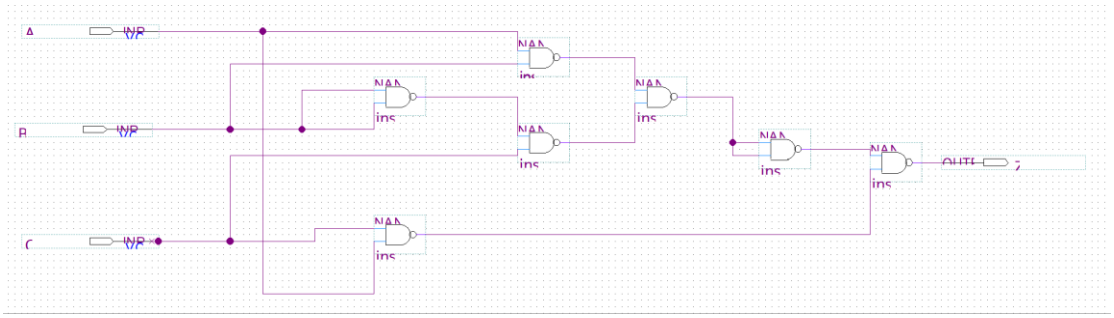
However, there is a crucial change.
Firstly, we make an analysis. If the Z is originally 1 before the value of B changed, then there is at least one 1 in A and C. Only the delay will cause glitches when both A and C are 1 and B changes its value. The value of B and the output of the NAND which is used as NOT may be equal for a short time. Then the Z is 0, which is an error.
To solve this problem, we add one more AC. This change will make sure that when both A and C are 1, the output won't be influenced by the value of B, so the delay won't cause a glitch now.
The equation is shown below:
Z=AB+AC+B'C
The last thing we need to do is transforming the logic expression to a circuit consisting of NAND gates.

Logic Diagrams

Part. A



Part. B



State Diagrams and Tables

Truth table of part. 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

Truth table of part. B:

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

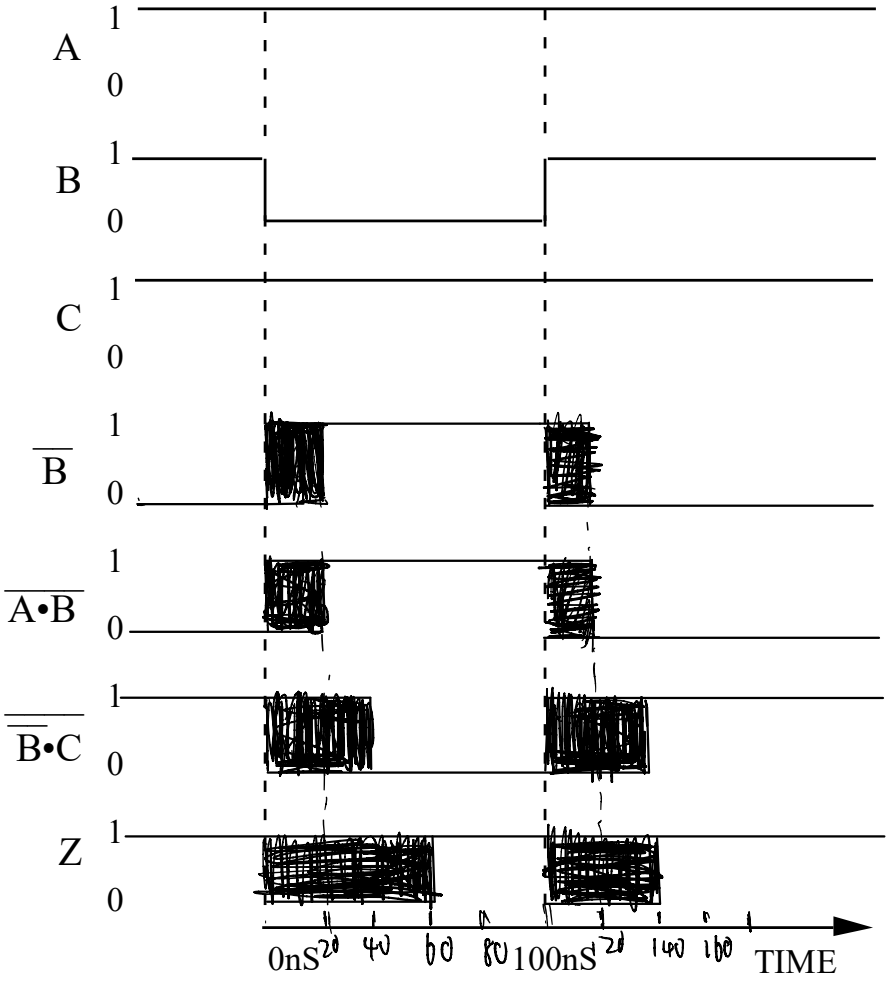
Answers to Lab Questions

1. Does it respond like the circuit of part. A? Describe the output and explain any differences between it and the results obtained in part 2.
   No. There is a glitch in A but not in B. The term AC prevent the happening of glitch and stabilize the output Z.

2. 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?
   Falling. The delays happen on the NAND functioning as a NOT. Assume both A and C is 1. When falling happens, because of the delay the input signal of the B' and C will get result 1. The output is 0. Glitch happens. By the following diagram and calculation, the stabilizing time of falling is 20ns longer than the rising due to the gate which causes the glitch is used by the NAND of B and C.

Answers to Post-Lab Questions

1. Timing diagram below for the circuit of part A. when the guaranteed minimum propagation delay of a 7400 is 0ns and the guaranteed maximum delay time is 20ns.



2. How long does it take the output Z to stabilize on the falling edge of B (in ns)?
   60ns.

3. How long does it take on the rising edge (in ns)?
   40ns.

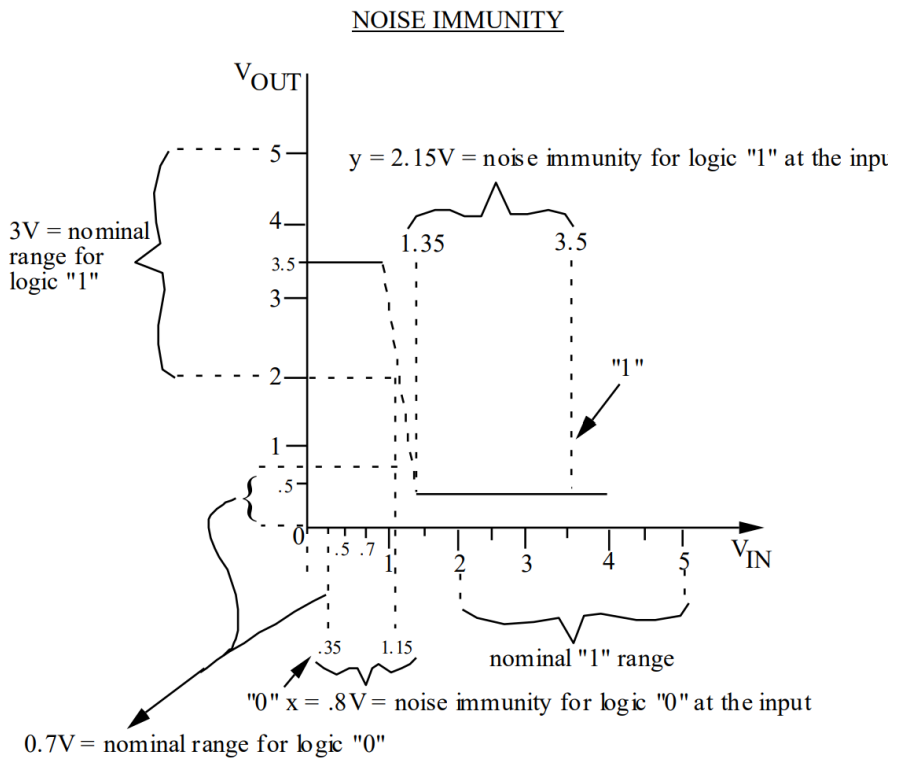
4. Are there any potential glitches in the output, Z? If so, explain what makes these glitches occur.
   Yes. Because the delay of the NANDs especially the inverter of B, the both input signal of the last NAND will be 1 in a very short time. The output is 0.

Answer to GG Questions

1. What is the advantage of a larger noise immunity?
   The large immunity can increase the stability of the circuit as the output won't be changed if the change of the input is quite small.

2. Why is the last inverter observed rather than simply the first?
   Because input may contain noise, the first inverter may fail to invert the input signal. We observe the last inverter so we can make sure that the observed signal is inverted successfully.

3. Given a graph of output voltage (VOUT) vs. input voltage (VIN) for an inverter, how would you calculate the noise immunity for the inverter?



THE OVERALL NOISE IMMUNITY OF THE GATE IS THE SMALLEST OF THE RANGES X AND Y.

The noise immunity for logic 0: 1.15 - 0.35 = 0.8 V
The noise immunity for logic ”1”: 3.5 - 1.35 = 2.15 V.
0.8<2.15, so the overall noise immunity of inverter is the smaller one, which is 0.8.

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

Because it may decrease the currency. Parallel LEDs share not only resistors but also currency. If the currency is too small, the LED may fail to light on. Then it can’t monitor anything.

Conclusions

Now the first lab is ended. For this lab, I reviewed so much knowledge which I used to learn in ECE120, especially K-map, the basic usage of Quartus, and logic expressions. Also, I learned new things like glitch, noise immunity and static hazards. Realizing how to deal with the glitch took a tremendous amount of time, but it’s incredibly interesting to me. It is a great step from theory to practice. I think it’s a good start of this course.