ECE 385

Fall 2019

Experiment #1

Introductory Experiment

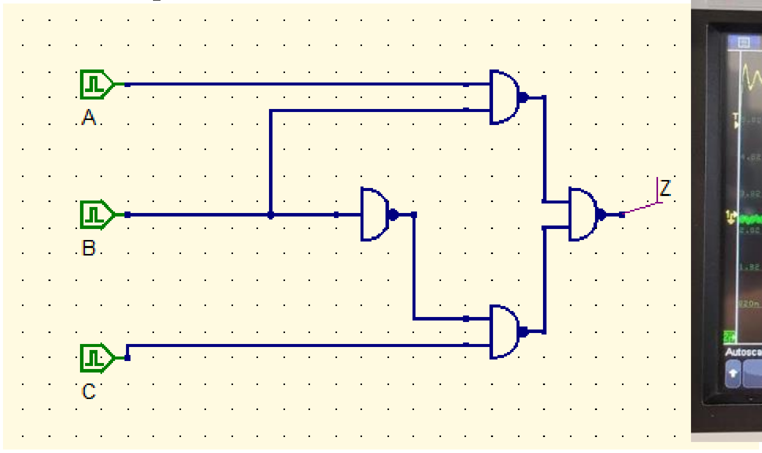
Pouya Akbarzadeh

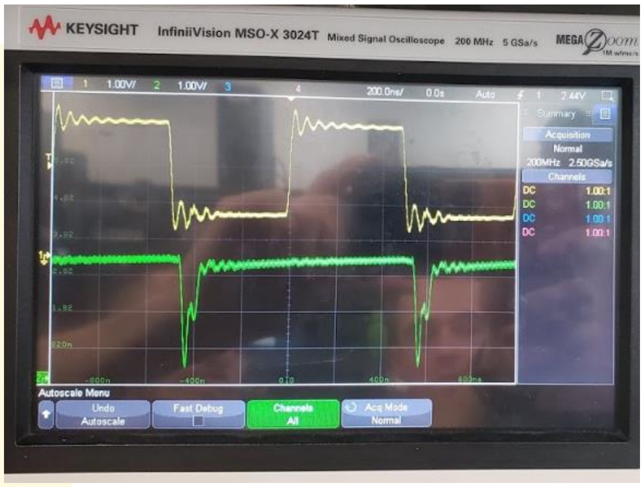
ABC, Tuesday, 8/27/2019, 2-4:50pm

Intro
In this lab, we study, observe, and solve the propagation delay of TTL chip and how it can cause static-1 hazard. Static-1 is best define as a glitch where we get the correct steady state result however, at one point we get the wrong output.

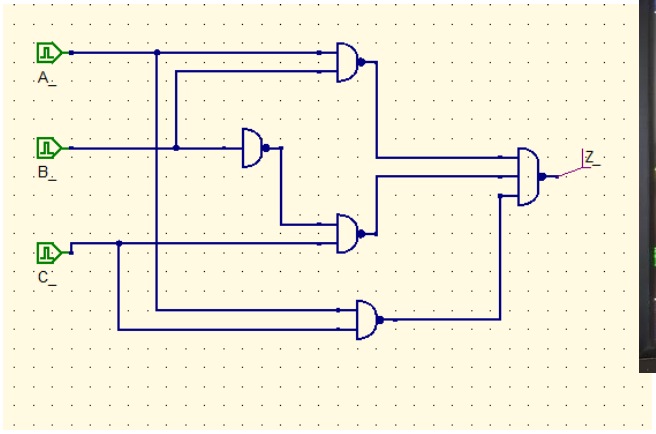
Purpose of Circuit
The purpose of this lab was to understand and review some basic transistor-transistor logic. This review included the ability of translating a logic circuit with various gates to only have NAND gates (Conversion showed in picture below). The circuitry used in this lab brought attention to the case of which the small-time difference would have caused the incorrect output while stepping through the circuit. The lab allowed us to think and try to problem solve around this issue.

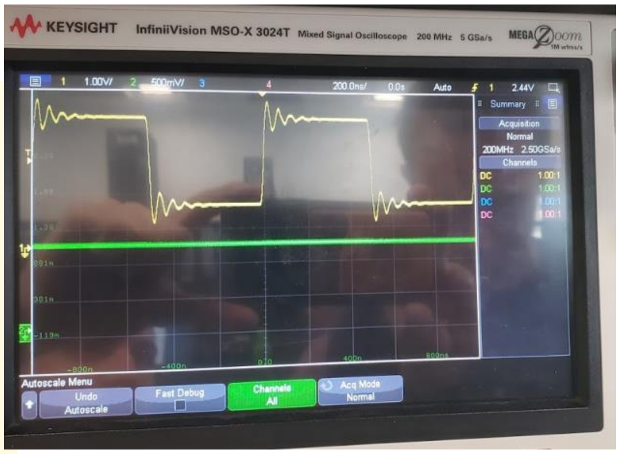
Written Description of Circuit
Old Circuit (With Static Hazard):
Boolean Expression: AB+CB'=Z

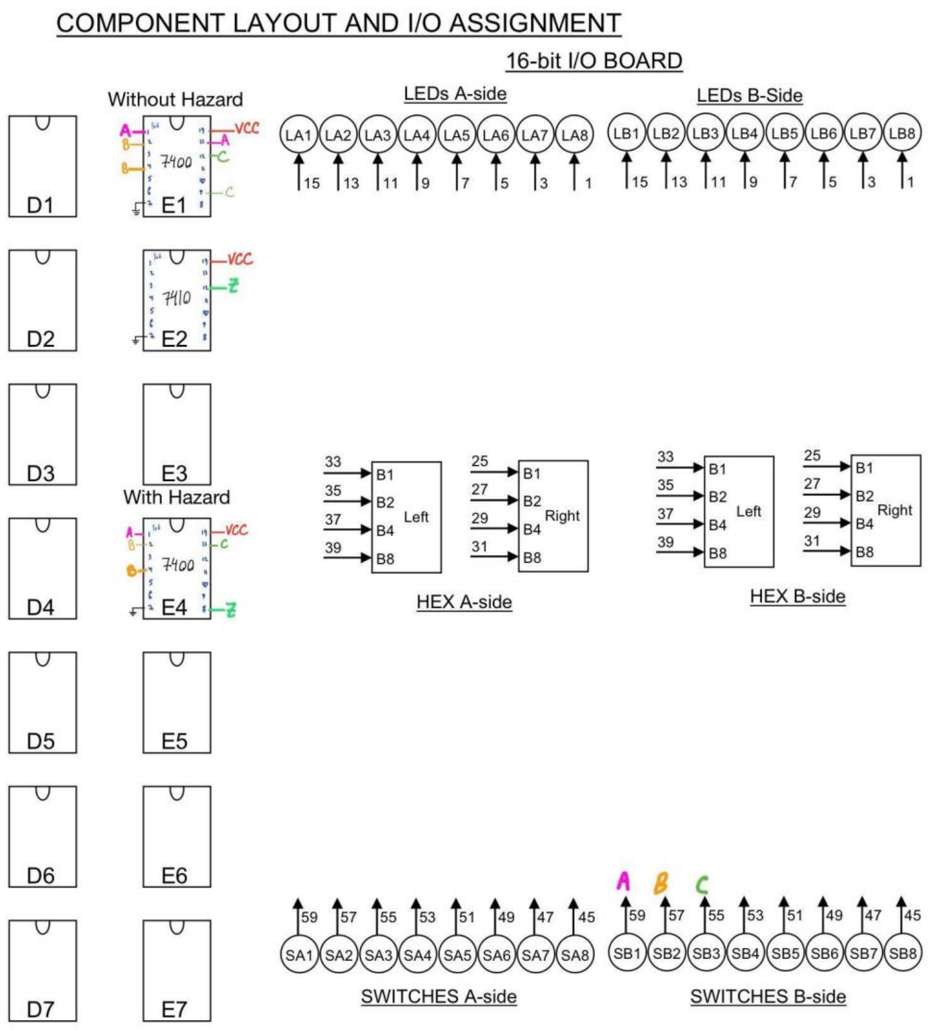


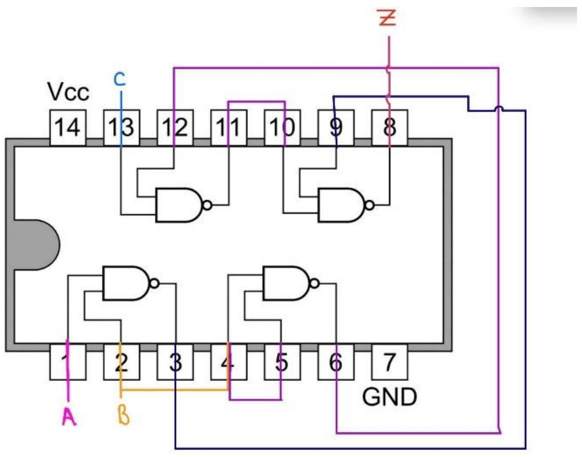


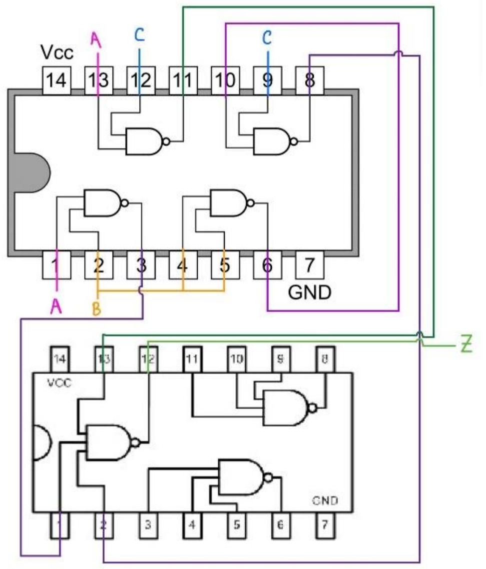
New Circuit (Without Static Hazard):
Boolean Expression: AB+CB'+AC=Z









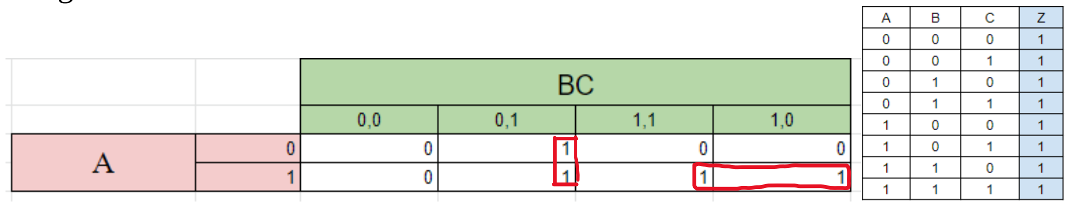


7400 Chip

7400 Chip and 7410 Chip

K-Map and Truth Table:

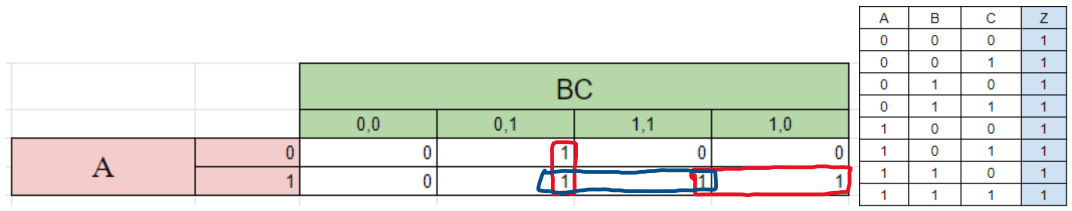
Design 1:



		BC			
		0,0	0,1	1,1	1,0
A	0	0	1	0	0
	1	0	1	1	1

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

Design 2:



		BC			
		0,0	0,1	1,1	1,0
A	0	0	1	0	0
	1	0	1	1	1

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

Clearly, we can see that the truth tables for both circuits are the same. This is because the truth table only shows the steady state. This goes to show how we have gotten away with static hazards so far in earlier courses.

The main difference between the new circuit and old circuit is the NAND gate that connects A and C and instead of a 2-input NAND gate we have a 3-input one. This change allows us to not have a static glitch. The circuit we created uses the 2 IC chips, (7400 and 7410) alongside the 16 bit I/O board to create outputs to our oscilloscope where we fist examined the glitch. Both of our circuits used a 2:1 MUX which controlled our two inputs A and C. We used B as our select bit.

Based on our knowledge in ECE120 both circuits should have the same result for Z. However, there was slight propagation delay in the gates in the first circuit. The brief lag created by the NOT gate caused the static hazard seen in the first screen capture of the oscilloscope. The green line should be straight however, it jumps.

Pre-Lab Questions

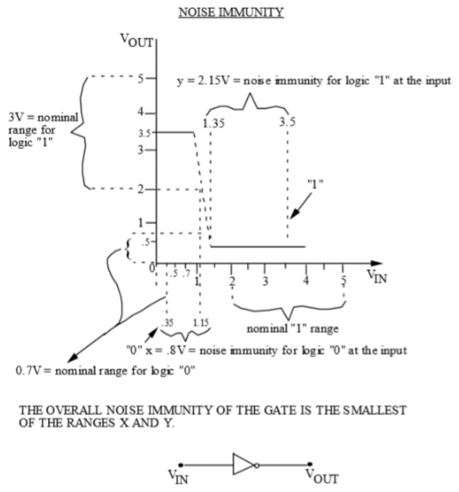
1. For new design look under “Written Description of Circuit” section. The reason all groups might not see the static hazard is due to the fact that no chip is truly identical. This the already small delay might be even smaller or so small that it’s not noticeable with the accuracy of our measuring device. The more you add to the circuit the more the delay increases. More uneven components = more delay. The reason the capacitor helps us see it is the quick charge and discharge from the bounce that we might see.
2. Look under “Written Description of Circuit” section

Lab Questions

1. The Noise Immunity of a gate is defined as the maximum amplitude of a positive going (noise) pulse added to the nominal logic "0" voltage level or a negative-going (noise) pulse added to the logic "1" voltage level at the input of a gate which does not cause the output of that gate to change its logic value. The nominal logic "0" and "1" voltage levels can be determined by connecting several inverter gates (e.g., 7404) in series and observe the voltage levels at the output of the last inverter when the input to the first inverter is at GND and at +5v. 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? See the following figure.

Noise immunity is a measure of the ability of a digital circuit to avert logic level changes on signal lines when noise causes voltage level changes. Thus, we have to observe it “backwards” and do the last inverter first. Also the last inverter is the one thing outputting to the terminals. The noise immunity can be calculated by finding the differences of the ranges given. The ranges shown in graph are the 1.35 is the lower and 3.5 is the upper range. Thus 3.5-1.35 gives us the 2.15V

Vin is logic low and Vout is logic high.

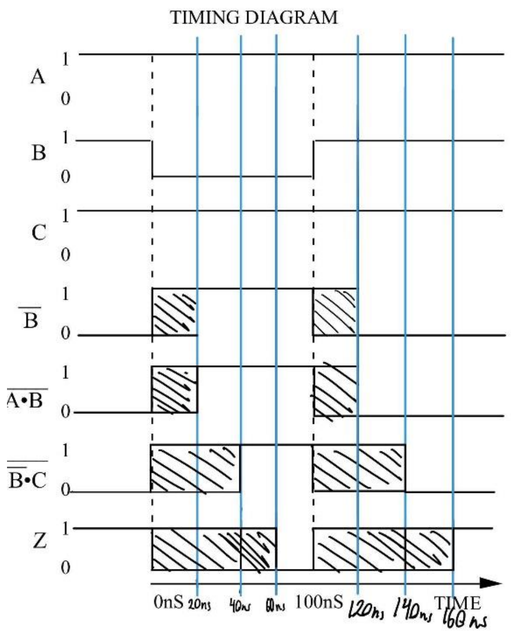


2. If we have two or more LEDs to monitor several signals, can we just use one 330Ω resistor for all LEDs?x

Since the circuit is configured with a parallel design. You would the same amount of current across the 2 LEDs which would not be enough to drive the LEDs.

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. 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.



The parts that are shades show where there could be delay or the delay be propagated. We can see how the small 20ns can cause much more as the signal passes through more gates. This uncertainty can be a glitch.

2. Explain how and why the debounce circuit given in General Guide Figure 17 works. Specifically, what makes it behave like a switch and how the ill effect of mechanical contact bounces is eliminated?

There are multiple ways to implement debouncing. Software and hardware are both options available, however, we took the approach and used the hardware option. To explain how the circuit works we first must understand what bouncing is. Bouncing is the result of two metal contacts creating multiple signals as they close and open. The debouncing circuit shown in Figure 17 takes care of that and allows the circuit to act like there was only one closing or opening. The flip-flop circuit uses a set-reset to get rid of the bouncing. The process takes place when the switch is flipped, the very first contact will cause the latch to change state, but any additional mechanical switch bounces will also have no effect. The flip-flop circuit can reset automatically after a short period of time, to register intentional repeated inputs from the same switch contacts on the rising edge of the clock.

Conclusion

In conclusion we see the importance of debouncing switches and how they can allow for less mishaps to occur and how they can allow easier debugging of the circuit later on. We also can see the impact that can occur by adding one component. This shows the level of attention required to ensure a smart and robust design. Propagation delays can occur easily occur with any design, even in a simple circuit that was used in this lab. Another important lesson learned is the documentation of circuits built and chips used. This allowed for faster building and debugging of circuits. I believe the circuit build is already simplified and doesn't need to be changed.

This lab was pretty simple, and we did not face that many issues. One issue that we did face was trying to understand and learn how to use the I/O board. We were not powering the I/O board and we were only powering the circuit.