ECE 385

SPRING 2019

EXPERIMENT #1

Introductory Experiment

Austin Lee

Tuesday 3pm (ABE)

Patrick Wang, Gene Shiue

Purpose

This laboratory experiment was designed to introduce the equipment which will be frequented throughout the course: the Agilent MSO6032A oscilloscope and the Hewlett Packard 33120A pulse generator. Using these two machines, a circuit was built to expose propagation delays in the logic gate chips. From there, we implemented changes in the circuit in order to prevent glitched output signals.

Written description

In order to obtain static-1 hazards, we built a circuit using logic gates in which different inputs had to run through different 'lengths' to reach the output: input B connected to both inputs of a NAND gate whose output then connected to another NAND gate whereas input C directly fed into the second NAND gate. A simple circuit following this prerequisite is the 2:1 Multiplexer, which can be constructed with four 2-input NAND gates using only one 7400 chip (see Logic Diagram 1).

A Karnaugh-map for a 2:1 MUX is as follows:

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

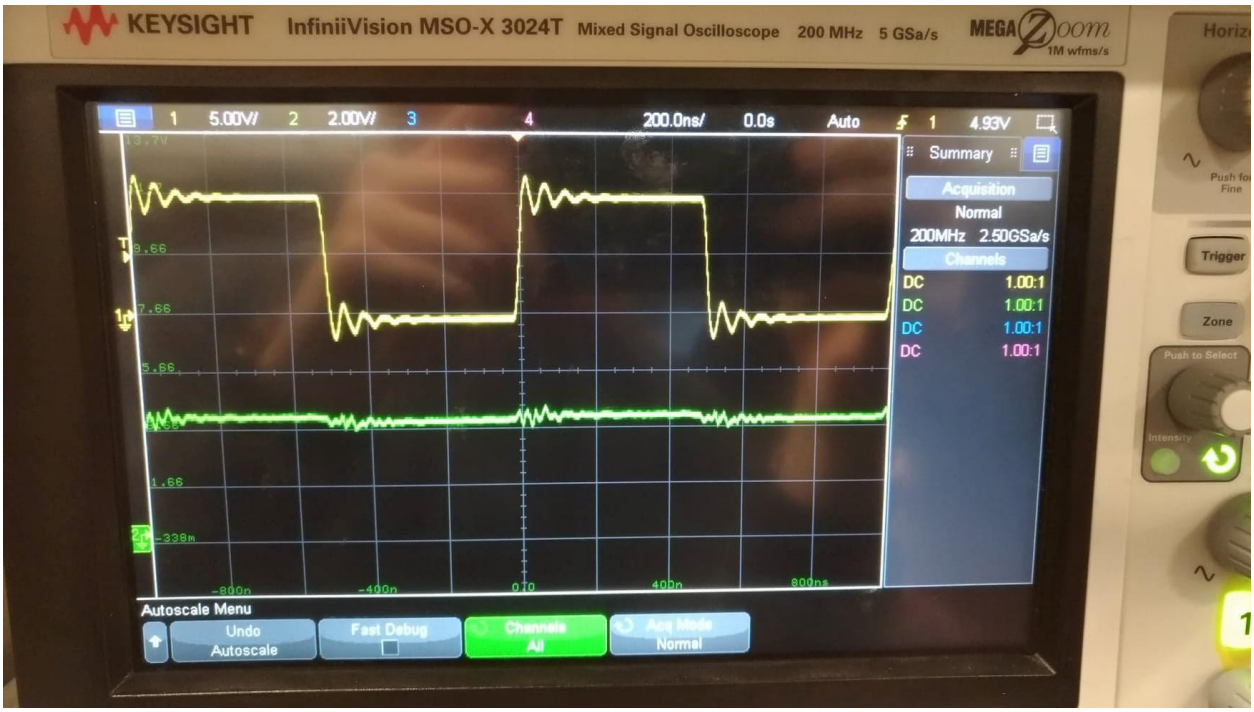
We controlled the three inputs A, B, and C using the I/O board switches. A complete truth table from all combinations of high/low settings of the inputs is as follows:

A	B	C	Z (output)
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

A static hazard may or may not be observed at this point. According to the General Guide, "all propagation gates have a nonzero propagation delay from input to output" (GG.22). This could result in a glitched output, where a 0 may temporarily occur where a 1 is meant to, or vice versa. This depends on the chip used, and it sometimes may not glitch. This is because although there is a guaranteed maximum propagation delay, there is no guaranteed minimum; it

is merely assumed to be 0 ns. Different chips also have a different range and typical delay. Therefore, while static hazards may occur, it is also possible that there the propagation delay is insignificant and does not affect the output. By inserting a chain of inverters or other logic gates before a signal's input is combined with another's, you can increase the potential propagation delay and therefore increase the chances of a glitch occurring. In our circuit, this can be accomplished by adding an odd number of NAND gates or inverters before the B signal is read by the NAND gate which takes the C signal as the other input.

Using the pulse generator, we drove input B with a 1MHz, 0 to 5-volt square wave instead of the I/O board switch. The output was displayed on the oscilloscope:



If a glitch was to occur, it would show dramatic divots in the green line—instances where the output was momentarily low instead of high. Our circuit, however, did not have these static-1 hazards. As mentioned above, the glitch effects could be dramatized by increasing the length of time before input B is read by the NAND gate with input C, but we chose to skip that step.

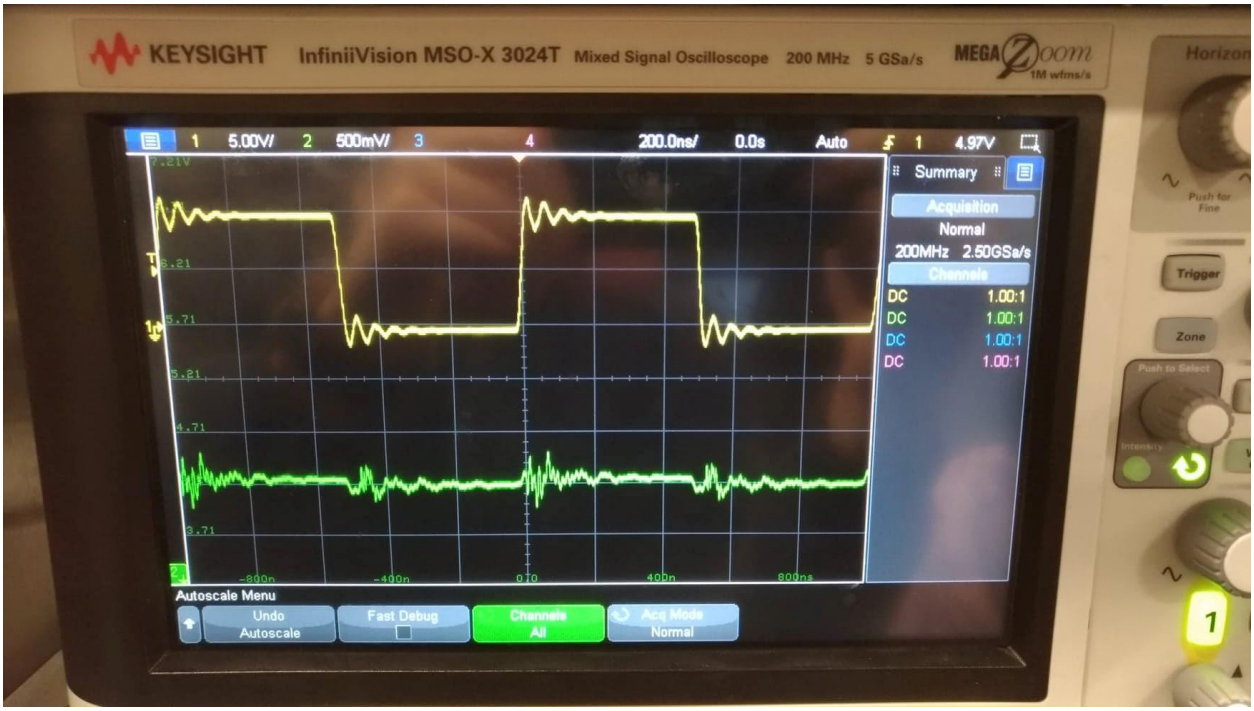
To solve the issue of static-1 hazards in this particular scenario, you can add a term to the Karnaugh-map:

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

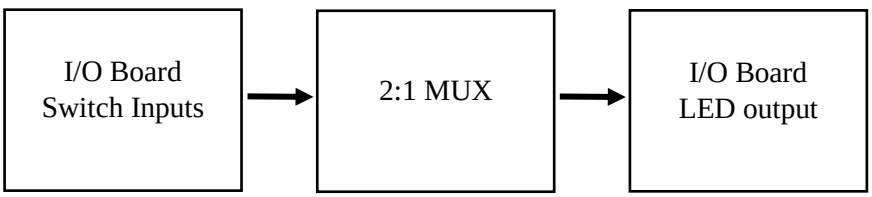
By grouping the two previously-separated combinations, we can introduce an additional NAND gate (see Logic Diagram 2). This would eliminate the dependency on the B input for combinations involving A and C driven to high. As you can see, the truth table created from this new circuit is identical to the previous:

A	B	C	Z (output)
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

So, without altering the output values, we have successfully eliminated any hazards and glitches that could have affected the circuit. This can be further derived from the output signals of the fixed circuit:

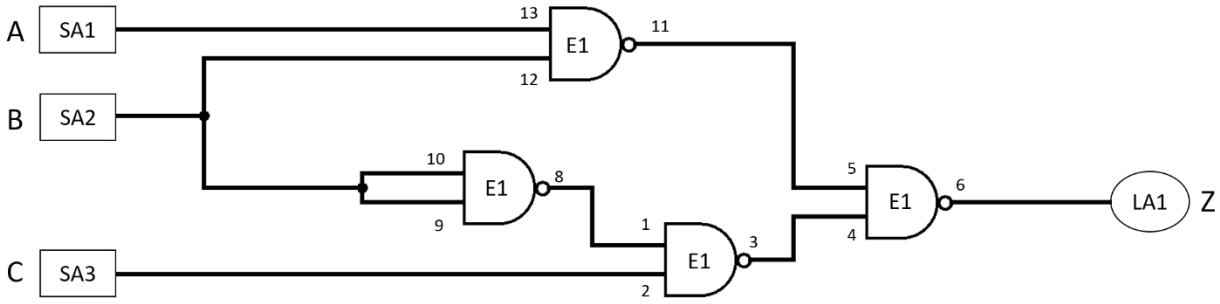


Block Diagram

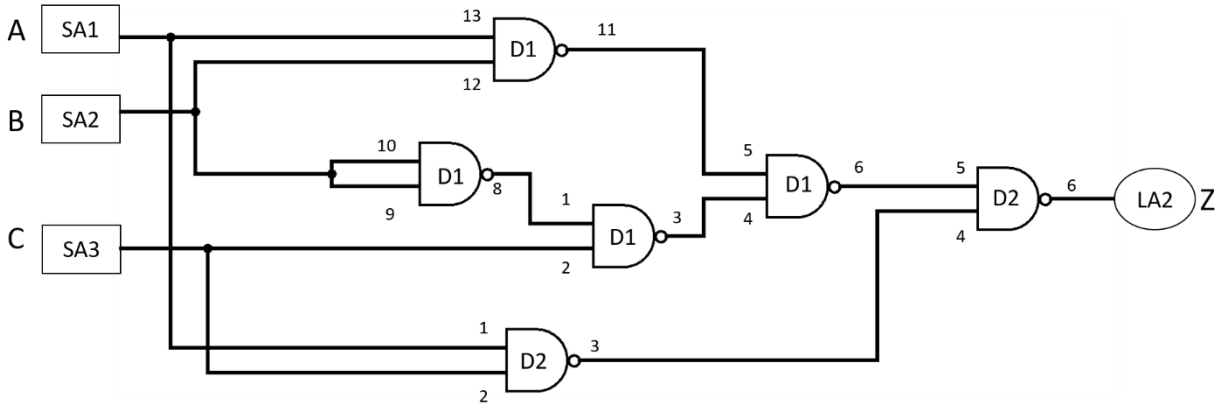


Logic Diagram(s)

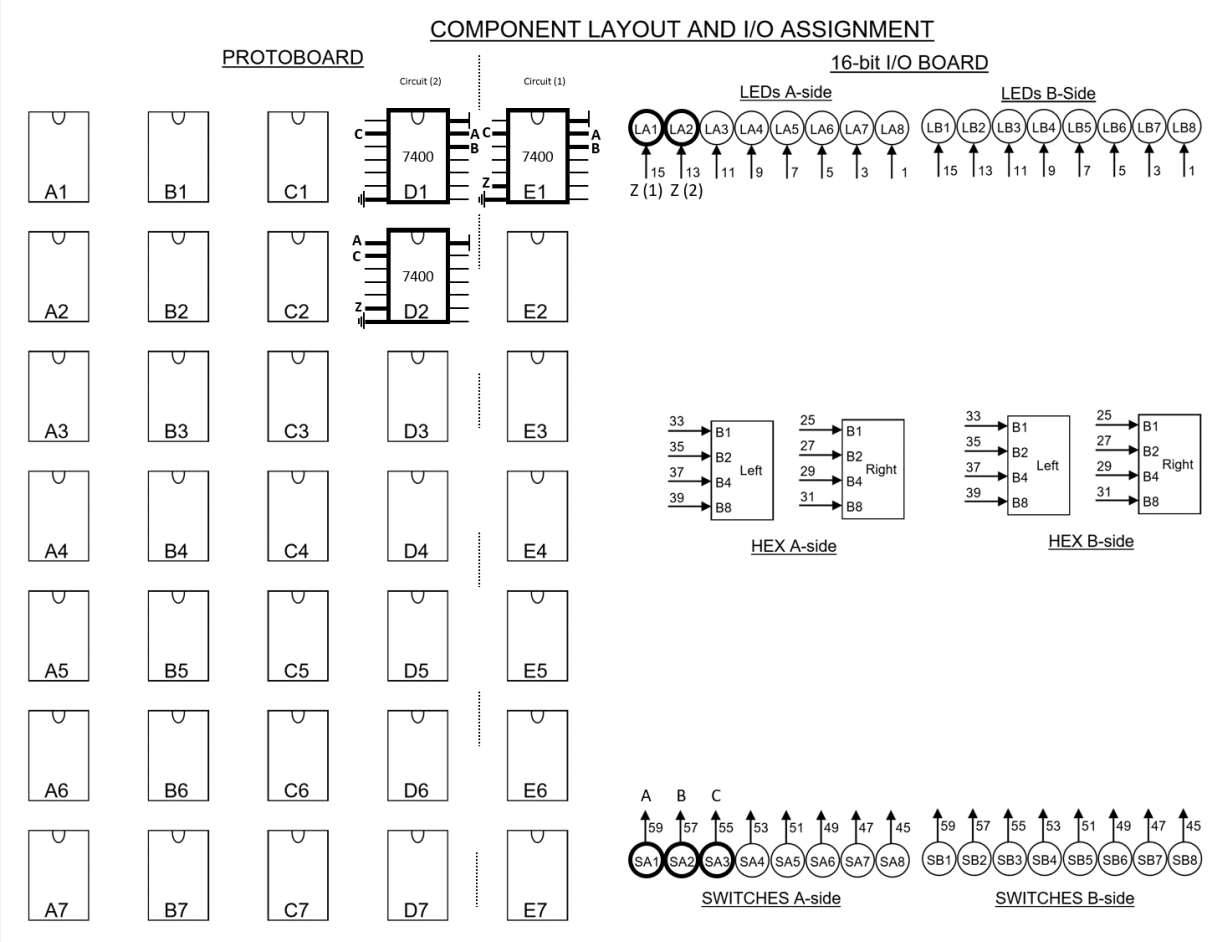
Logic Diagram 1



Logic Diagram 2



Component Layout



Pre-lab Questions

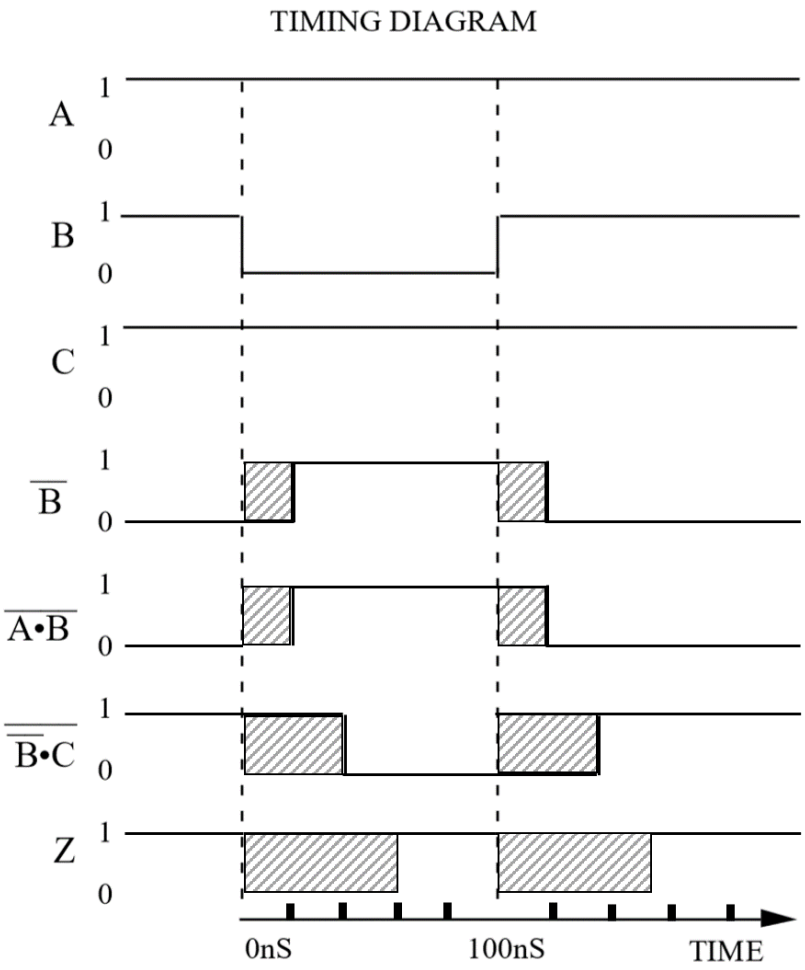
As mentioned, some groups may not observe static hazards because the propagation delay is a range with the minimum of no delay (0 ns). It is possible that the delay was insignificant enough to cause no effect at all on the output. By adding inverters/additional logic gates, we increase the potential maximum propagation time thus increasing the potential for a hazard to occur.

Lab Questions

We are more likely to observe glitches at the falling edges of input B. This is because the glitch happens when the output of the NAND of A and B is 1 but the output of the NAND of ~B and C is also 1. This would then cause the output Z of the circuit to be 0 instead of 1. This occurs during the falling edge, when the B input changes from 1 to 0. There is less possibility of a glitch at the rising edge because a propagation delay would not affect the output: Z would be 1 regardless.

Post-Lab Questions

1.



In theory, it takes 60 ns for Z to stabilize on both the rising and falling edge of B. However, the only place where glitches should occur is on the falling edge of B's signal. As mentioned, this is because the intermediate NAND gate values may have a propagation delay and therefore the output of Z would be incorrect, showing 0 instead of 1. On the other hand, there shouldn't be a glitch on the rising edge since a propagation delay would not affect the output since it would stay at 1 even if there is a delay error.

2. The debouncer circuit uses an S-R latch to prevent the output from bouncing when a switch is moved. A SPDT switch grounds one of the inputs while the resistors drive the other input to high. This means that one of the NAND gates' output will be affixed to 1, while the other NAND gate is affixed to 0. Even if the contact switch moves between the two terminals, it will not affect the S-R latch as the bottom NAND gate's output is connected back into the upper NAND gate's input. This arrangement ensures a stable output from both NAND gates. This circuit therefore makes sure that the output of the switch does not bounce as it stabilizes.

General Guide Questions

GG.6: A larger noise immunity means the device can function properly with higher amounts of noise. This would also decrease the possibility of glitches occurring; a gate with low noise immunity would be more susceptible to incorrect outputs from interpreting noise as input changes. You measure the last inverter because you want to discern the nominal voltage level of logic "0" and "1" after several logic combinations. It would be pointless to measure the first inverter as it may introduce very little noise and have little to no effect on the output voltage—an inaccurate conclusion of the noise immunity.

From a graph, you can calculate the noise immunity by finding the smaller of the difference of voltage ranges. In the figure on GG.7, the range for logic "1" is 1.35 to 3.5V input and 2 to 5V output. For logic "0", the range is .35 to 1.15V input and 0 to .7V output. The absolute difference in minimum output-input for logic "1" is 2 – 1.35 = 0.65V. The absolute difference for logic "0" is the maximum input-output: 1.15 – .7 = 0.45V. Since .45 is less than .65, the noise immunity of the gate is 0.45V.

GG.29: Sharing resistors could potentially damage the circuit components. For example, given a resistor is setup with the premise that it would be connected to multiple powered LEDs, if only a single LED is powered it might receive a greater current than it can safely handle and therefore burn out. On the other hand, a resistor setup in a circuit for only one LED might not provide enough current to multiple LEDs and therefore misreadings can occur if the LEDs do not illuminate.

Conclusions

In conclusion, I think this lab went very smoothly. There were very few issues if any as the documentation was rather simple and easy to follow. We did choose to omit verifying the glitch occurence using the oscilloscope and chaining inverters but only because we were instructed that we could alternatively explain it in this report. Otherwise, the rest of the experiment was completed successfully. We learned about propagation delays and glitches, how to display them, and ultimately how to effectively eliminate them through the study of debouncing switches.