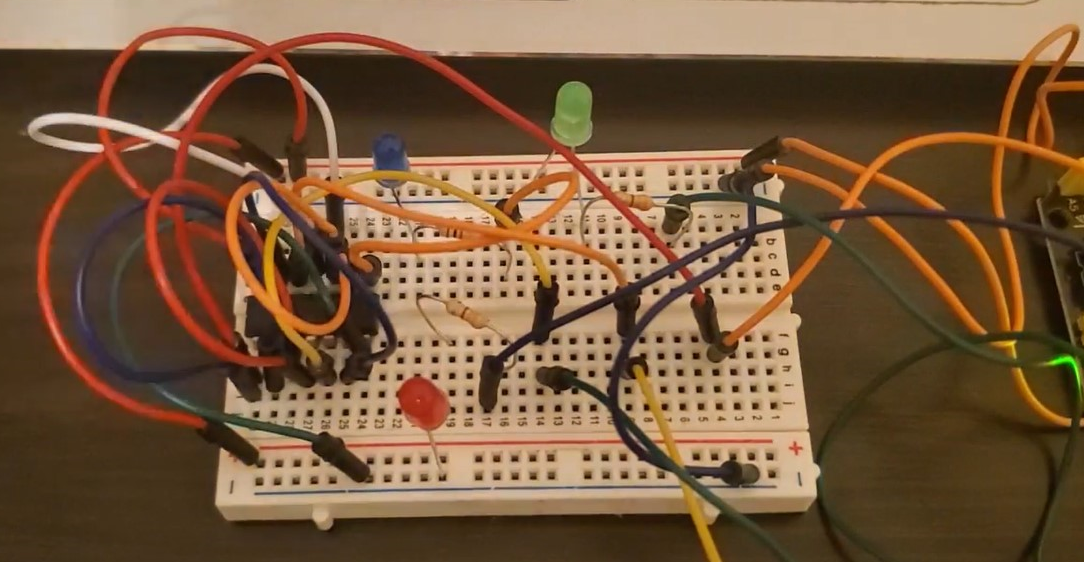
Digital Logic Basics

**ECE2300L Module 1 Report**



Kyler Martinez

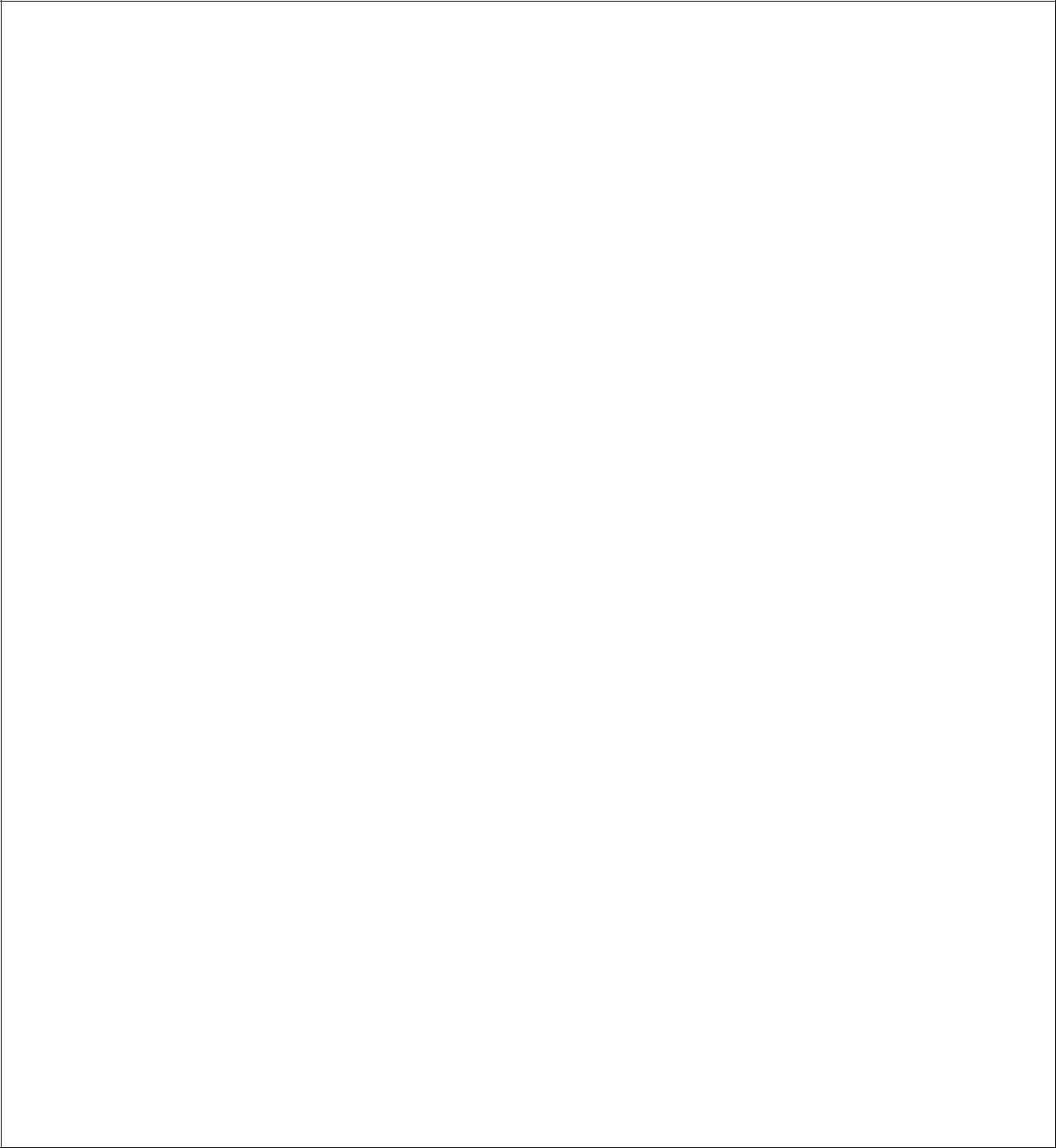
September 9, 2020

**Introduction**



The learning outcomes of the module is to be able to translate real life situations into truth tables that can then be transformed Boolean logic functions and simplified using Boolean algebra and its theorems and axioms. Another learning outcome is to be able to use logic gates to create combinational logic circuits and then be able to verify the design of the circuits against their truth tables to ensure proper designing. We also would learn why the NAND and NOR gates are considered universal gates and to then utilize these gates to create equivalent circuits. We also are to gain hands on experience for implementing logic gates using different logic chips and using the Arduino software. Finally, we then are to explore logic gate parameters and the effects of implementing logic gates improperly and to compare measured parameters to parameters from the data sheets for specific gates.

**Activity 1.1 — Boolean function, logic gates, truth table**



Problem Statement:

Since you are a successful college student, you undoubtedly developed good study habits when you were younger. Study habits at your home probably included something like the following:

* Bedtime is 10:00 p.m. on school nights.
* Homework must be completed before any playing, school night or not.

Based on these rules we can make the following statement:

*If it’s before 10:00 p.m. or it’s not a school night, and your homework is finished, then you can play video games.*

1. The Boolean Variables x, y, and z are defined as the following
   1. = It’s a school night
   2. = Homework is finished
   3. = It’s before 10:00 p.m.

Then the statement above can be written as a Boolean function V(x,y,z), which returns TRUE when you can play video game and FALSE when you cannot.

V(x,y,z) = ( x′ + z) y

1. Logic Diagram

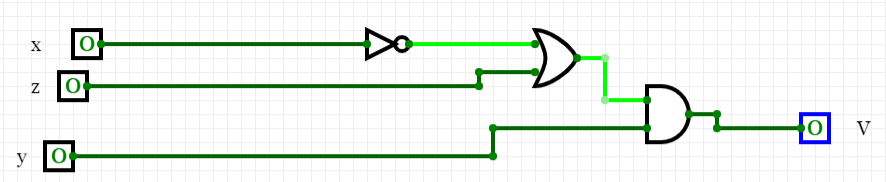


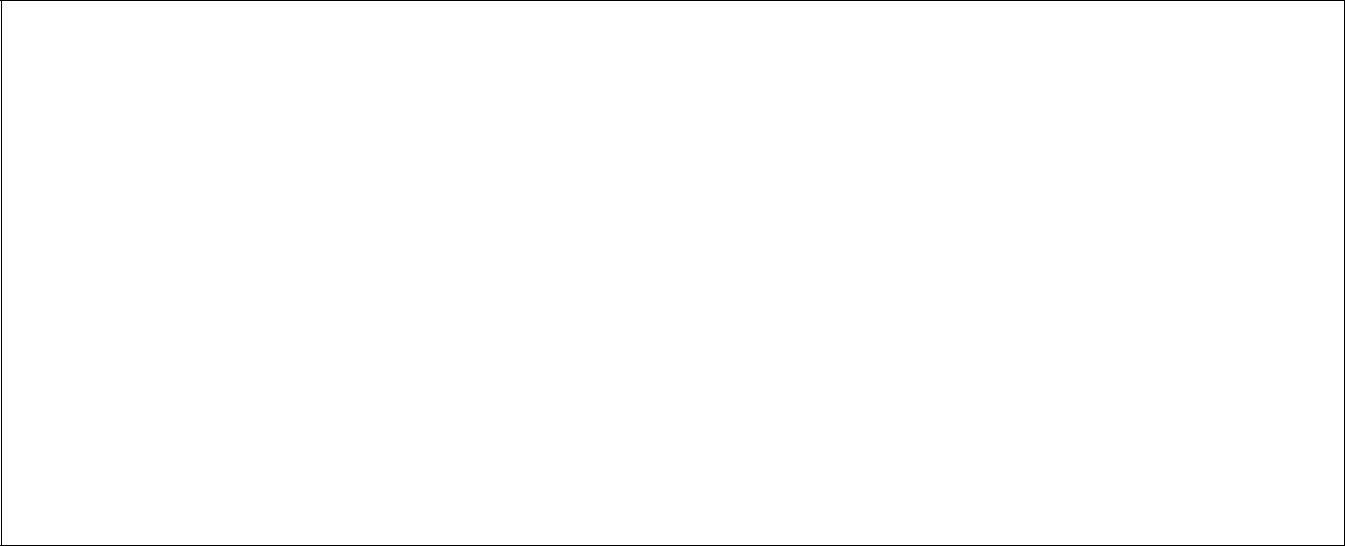
Figure : Logic Diagram of V Using Basic Logic Gates

1. Truth Table For V

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | y | z | x′ | x′ + z | V = (x′ + z) y |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |

Table : Truth Table For V

**Activity 1.2 — Implementations using NAND/NOR**



We can implement the same Boolean function V(x,y,z) using NAND gates only or NOR gates only.

1. Implementation of V using NAND gates only.

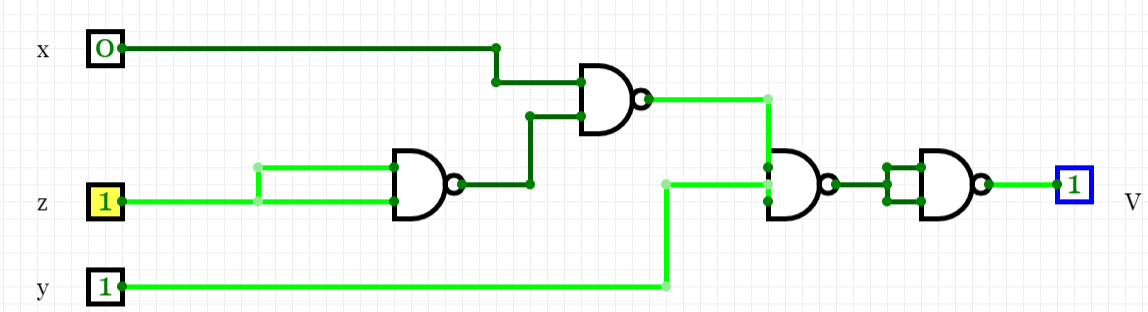
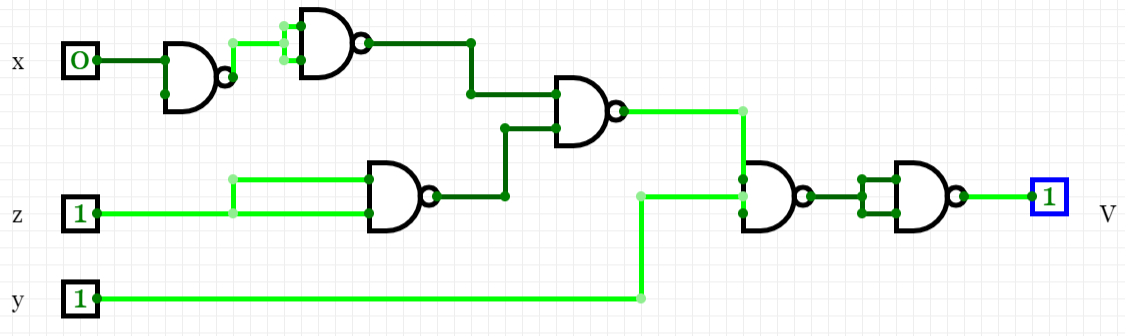


Figure : NAND Gate Implementation of V & Simplified Diagram

VNAND = (((x•z′)′ • y)′)′ = (x•z′)′ • y

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | y | z | z′ | x•z′ | (x•z′)′ | V = (x•z′)′ • y |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |

Table : Truth Table For NAND Implementation Of V

b. Implementation of V using NOR gates only.

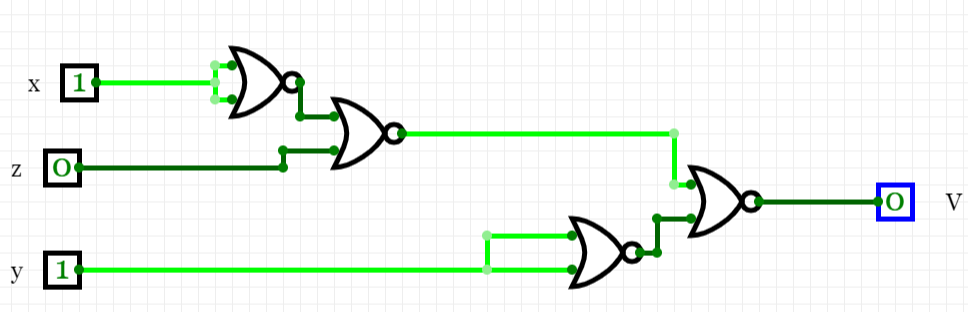
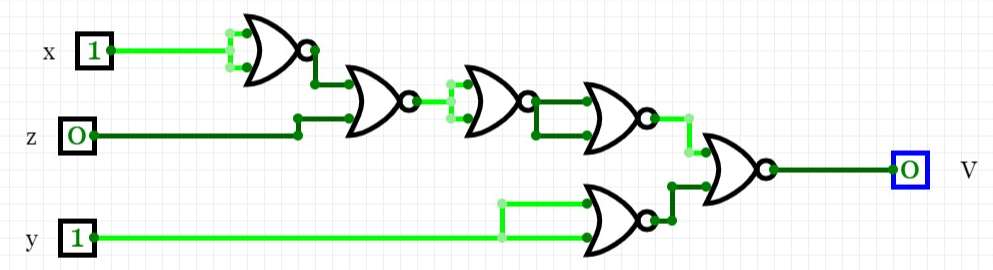
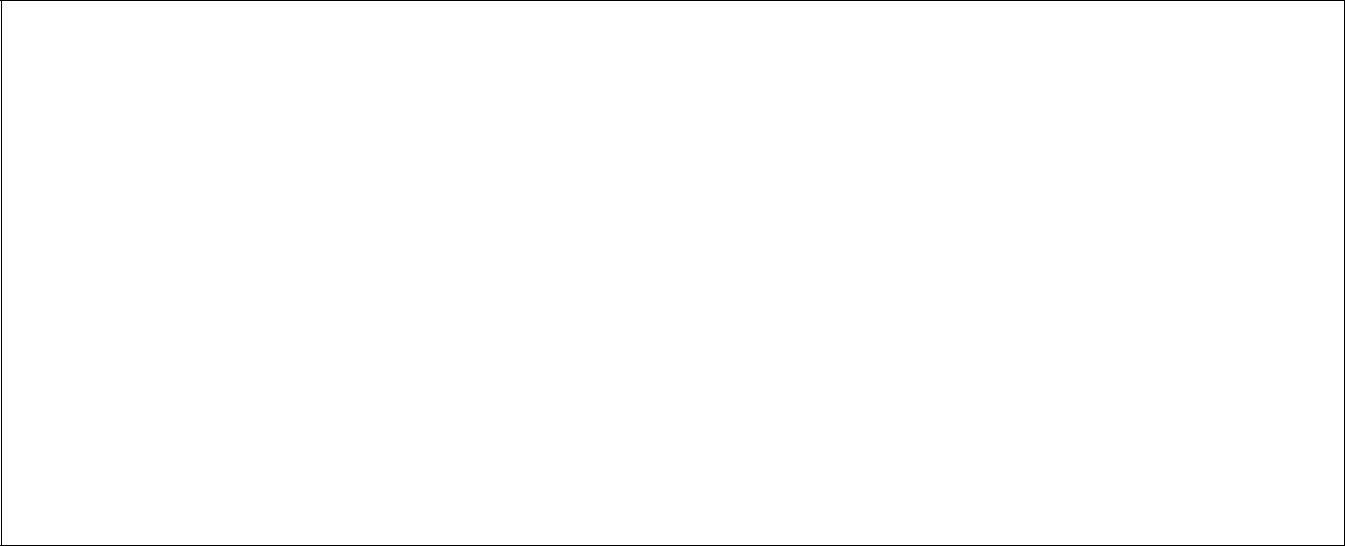


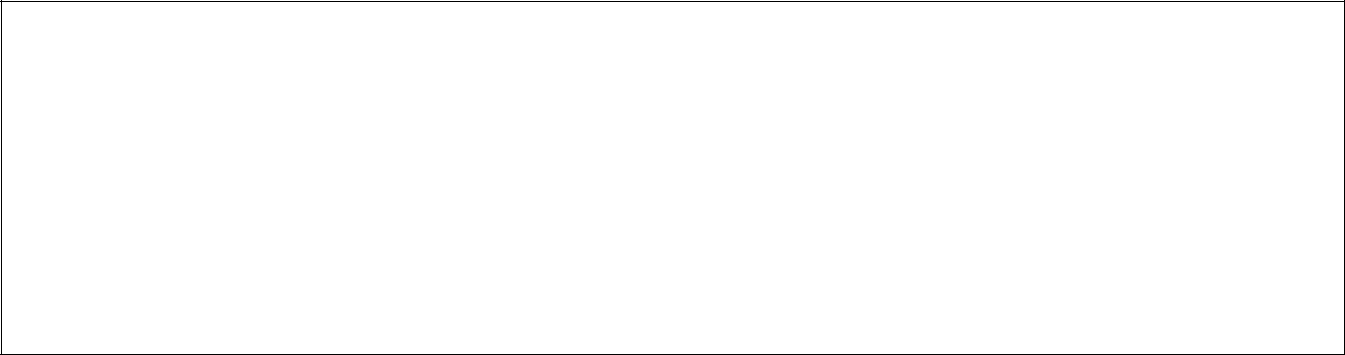
Figure : V Implementation Using NOR Gates & Simplification

VNOR = (((((x+x)′+z)′+ ((x+x)′+z)′)′) +(y+y)′)′ = ((x′ + z)′ + y′)′

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | y | z | x′ | y′ | (x′ + z) | (x′ + z)′ | ((x′ + z)′ + y′) | V = ((x′ + z)′ + y′)′ |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

Table : Truth Table For NOR Implementation Of V

**Activity 1.3 — Simulation with CircuitVerse**



Through simulation, we can verify that all three implementations of V(x,y,z) produce the same truth table.

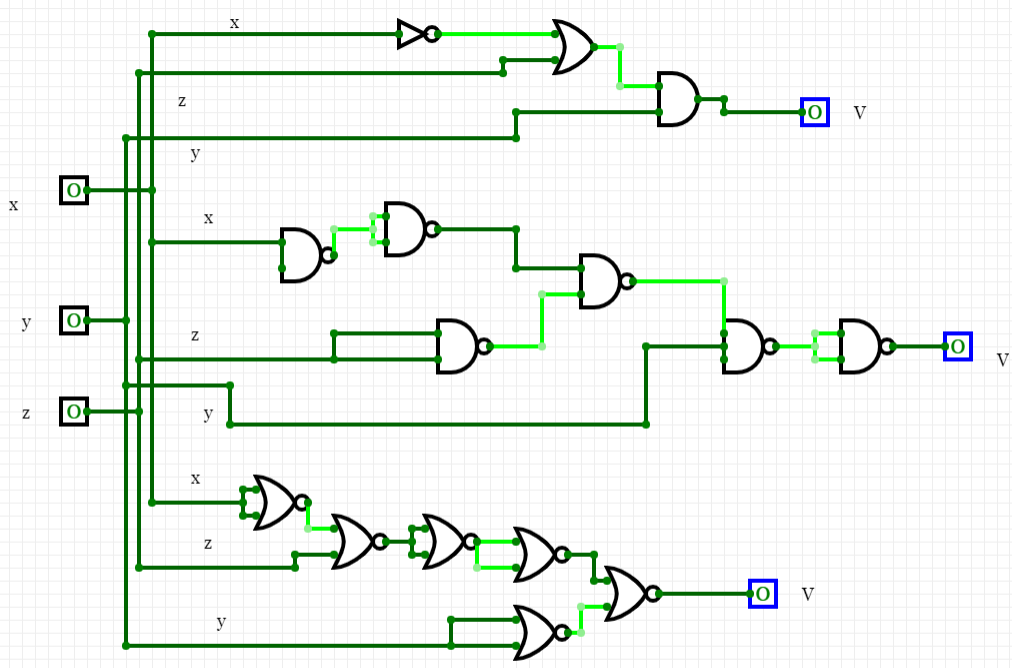


Figure : Inputs x = 0, y = 0, z = 0, and V=0

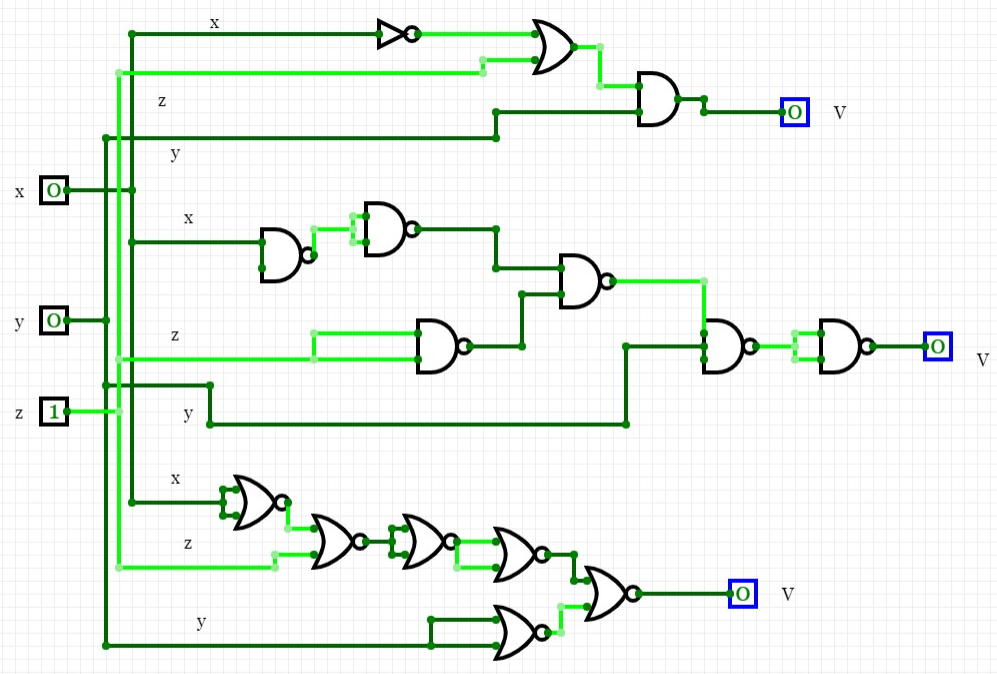


Figure : Inputs x = 0, y = 0, z = 1 and V=0

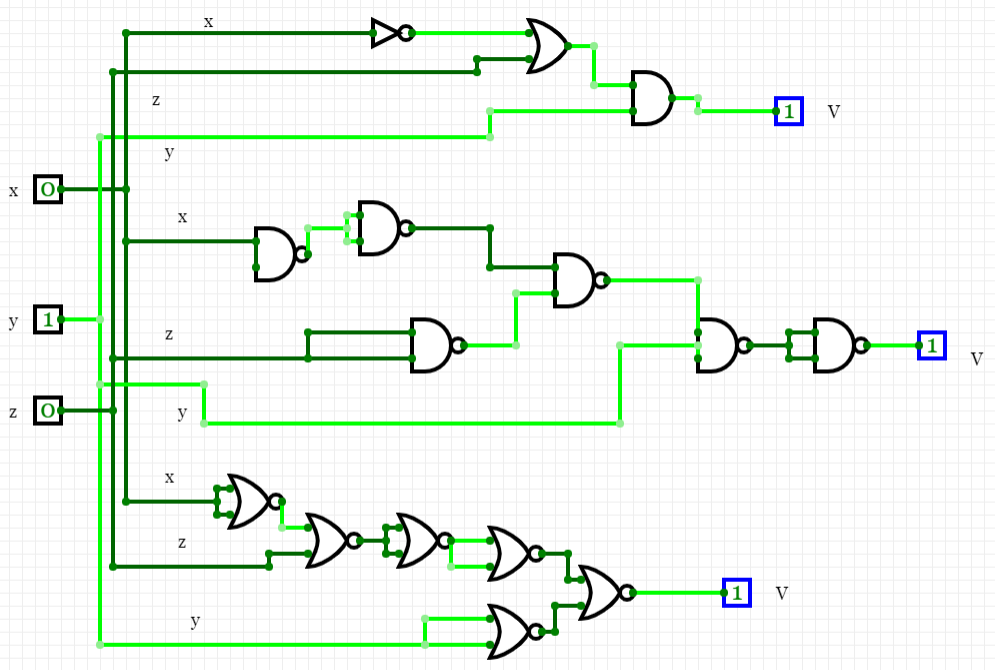
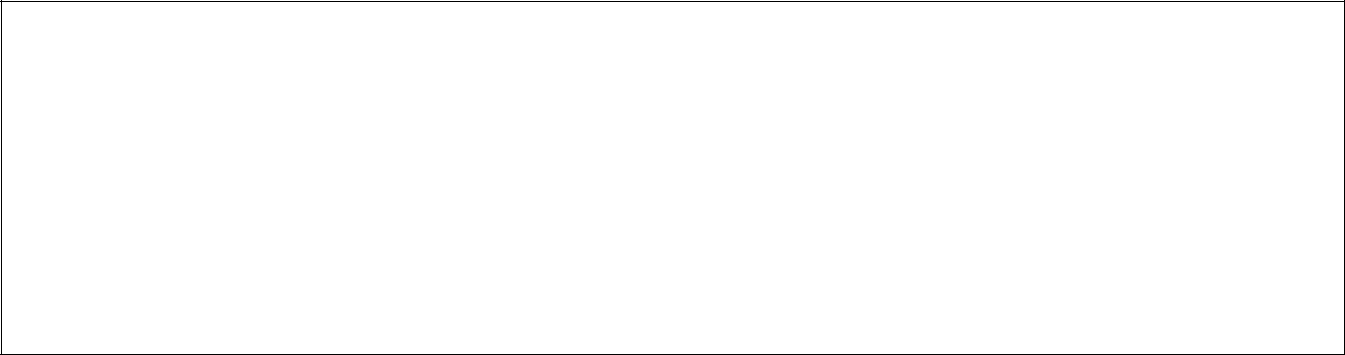


Figure : Inputs x = 0, y = 1, z = 0 and V=1

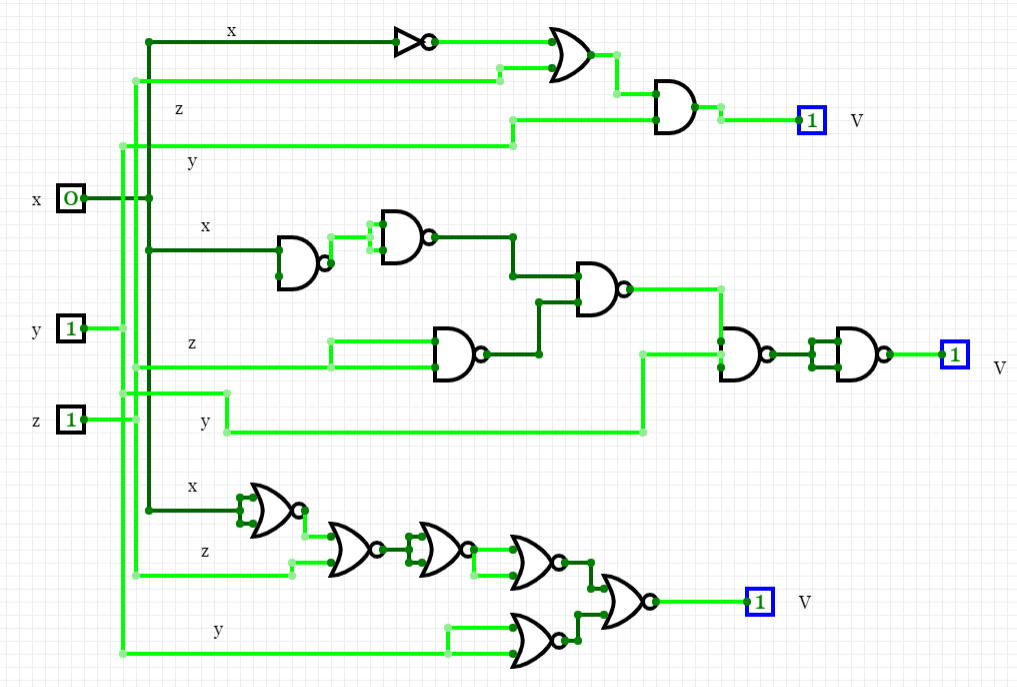


Figure : Inputs x = 0, y = 1, z = 1 and V = 1

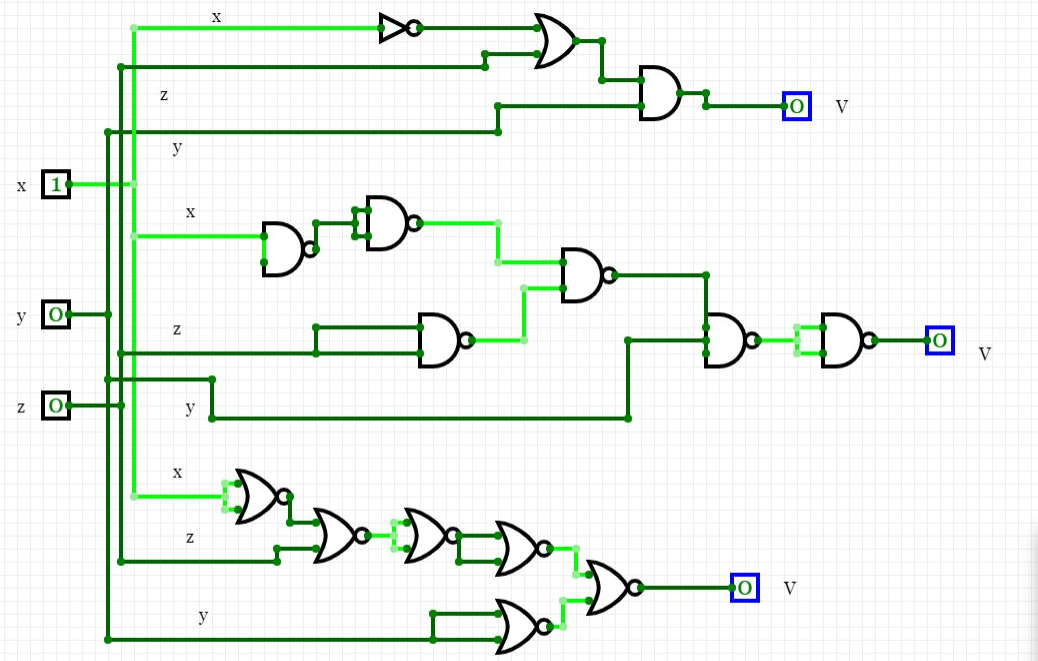


Figure : Inputs x = 1, y = 0, z = 0 and V=0

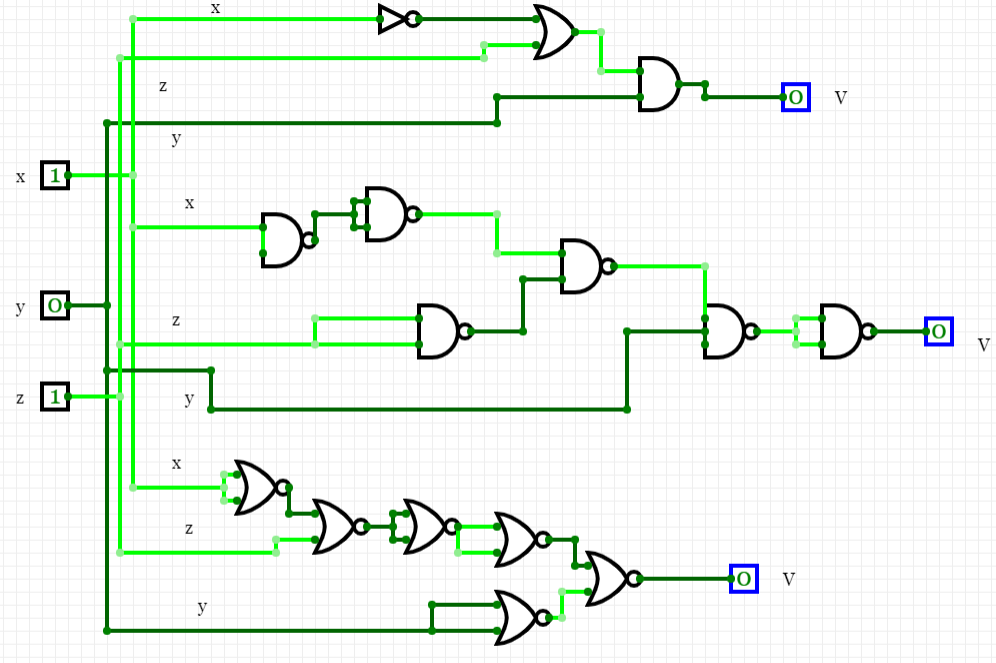
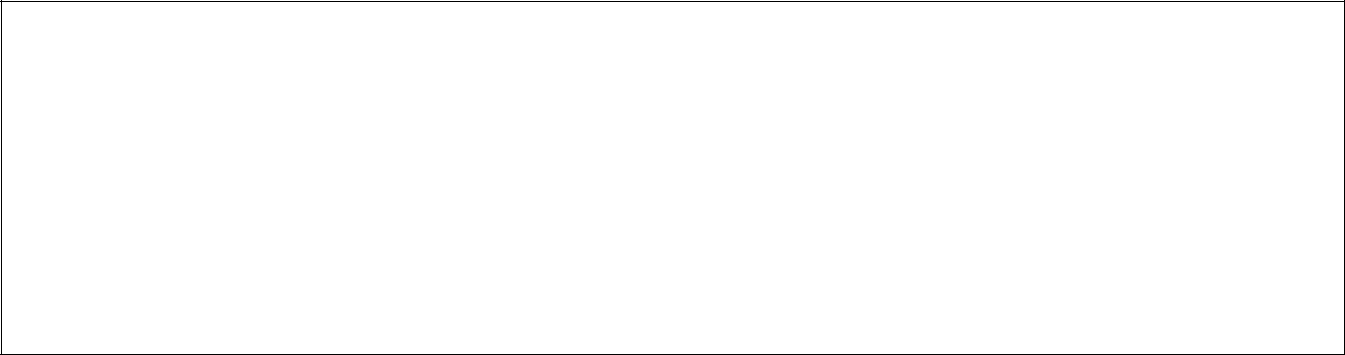


Figure : Inputs x = 1, y = 0, z = 1 and V=0

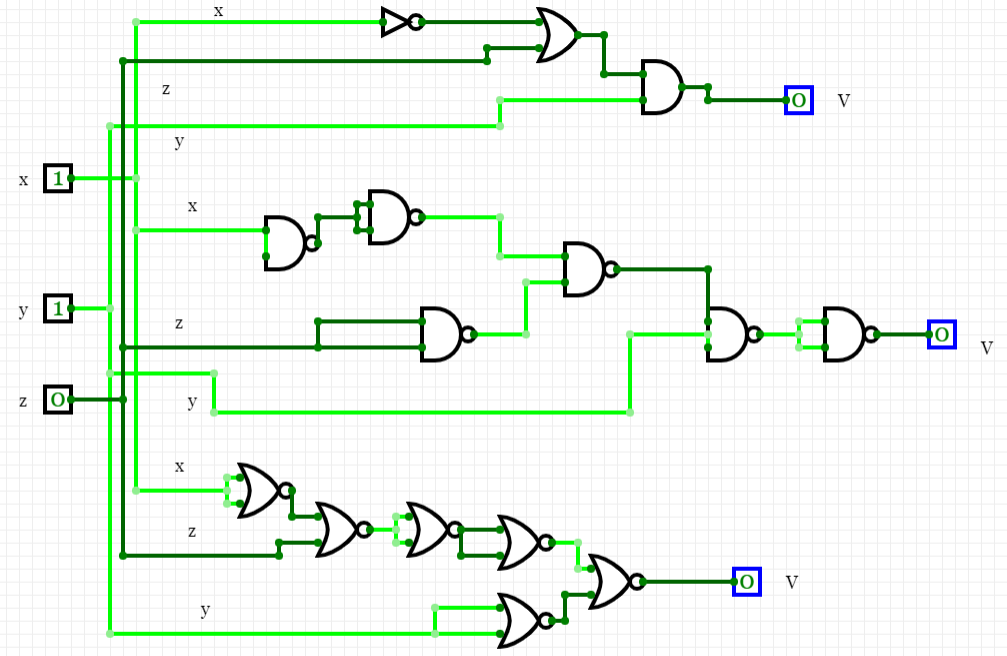


Figure : Inputs x = 1, y = 1, z = 0 and V=0

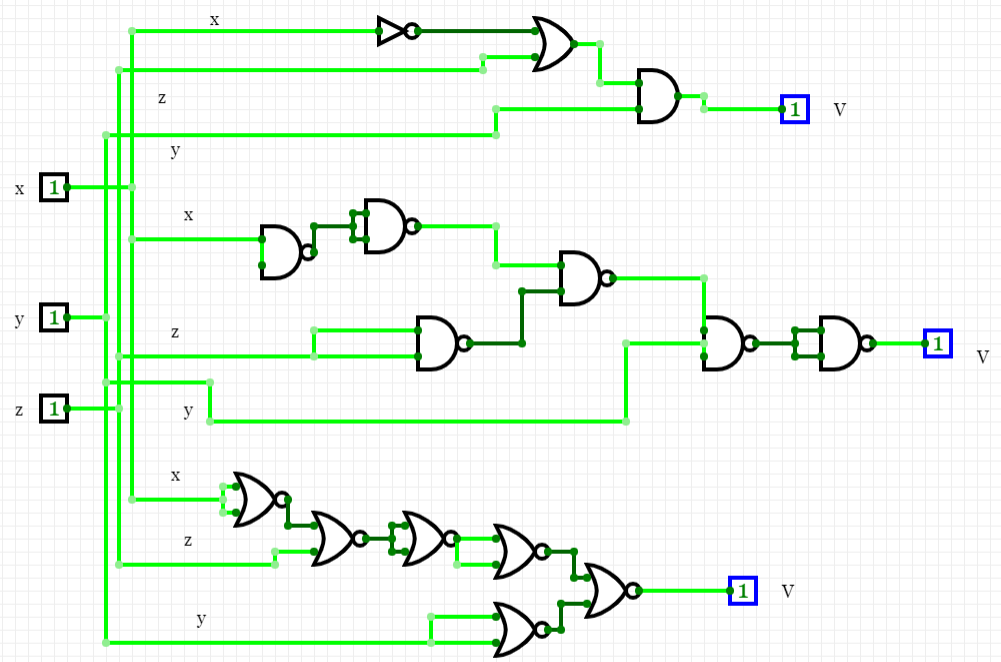
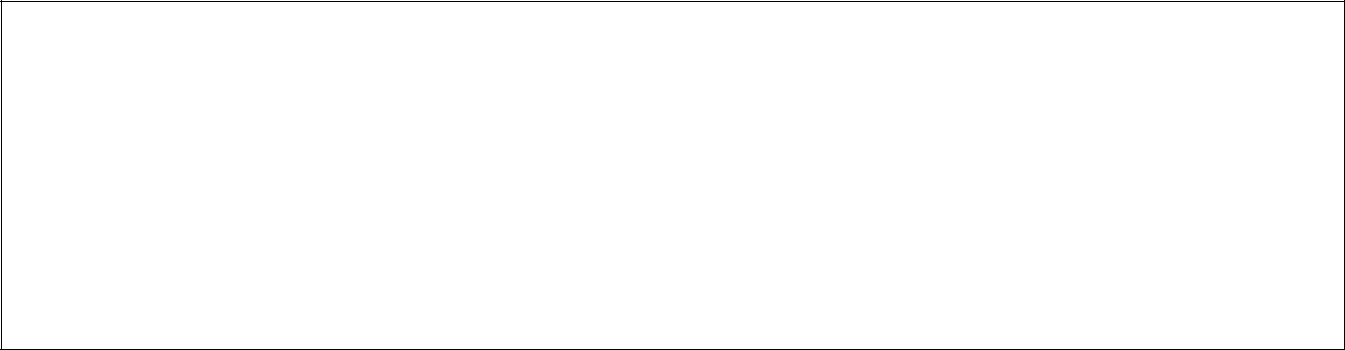


Figure : Inputs x = 1, y = 1, z = 1 and V=1

**Activity 1.4 — Logic gate parameters**



During activity 1.4 we explored some logic gate parameters by building the circuits in figures 12, 13, and 14 and recording voltages and current between logic gates and their outputs.

The breadboard was configured to reflect the circuit described in figure 12 to record voltage.

When pin one is at logic level low:

The voltage from pin 2 and 3 to ground was measured to be 4.99 V which is VOH and VIH.

The voltage from pin 4 to ground, VOL, was found to be .02 V.

When pin one is at logic level high:

The voltage from pin 2 and 3 to ground was measured to be .03 V which is VOL and VIL.

The voltage from pin 4 to ground, VOH, was found to be 4.55 V.

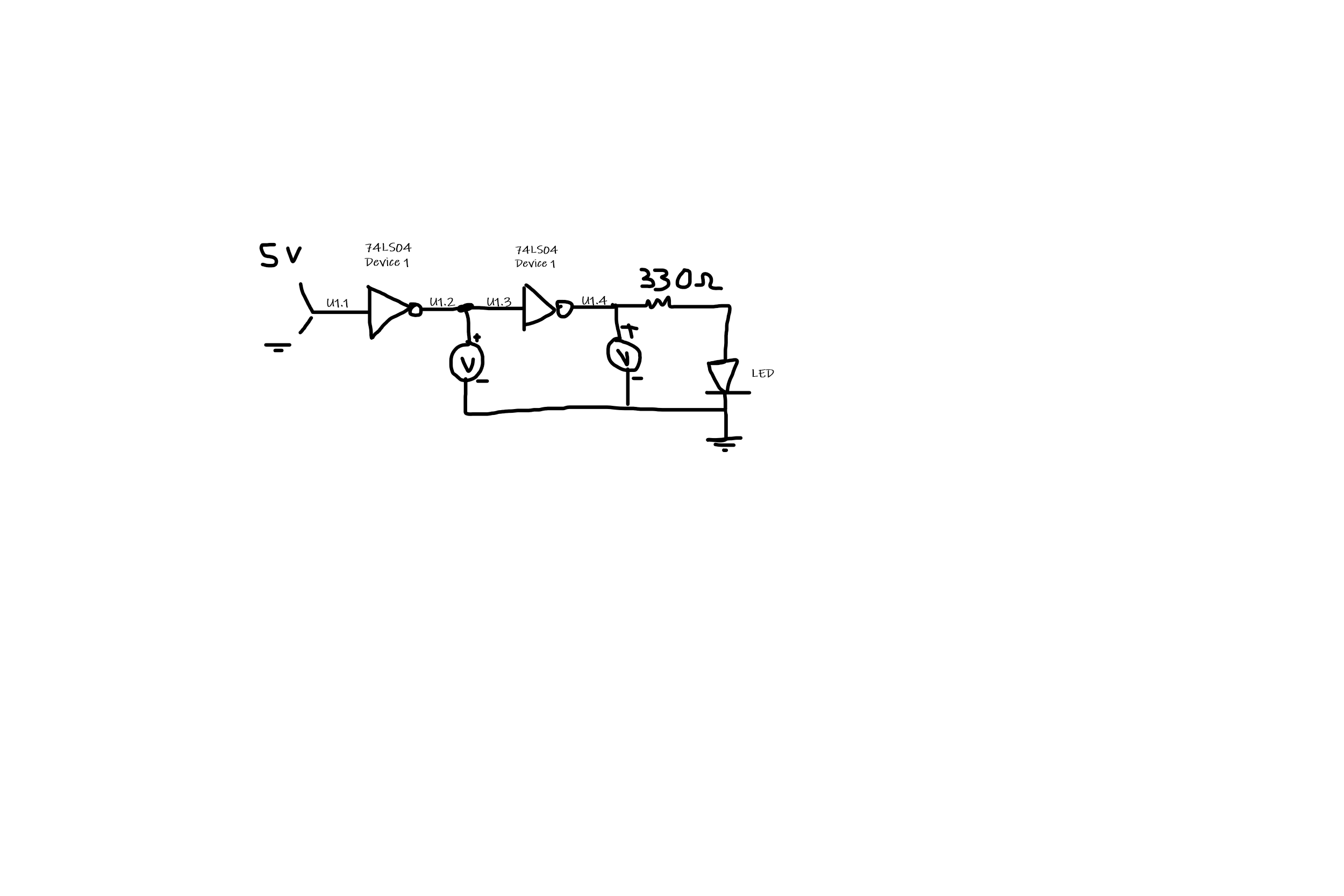


Figure : Voltage Measurements Diagram

The breadboard was configured to reflect the circuit described in figure 13 to record current when pin 1 is at logic level low.

The current from pin 3 to pin 2 was measured to be 1μA which is IOL.

The current from pin 4, IOH, was found to be 7.57mA.

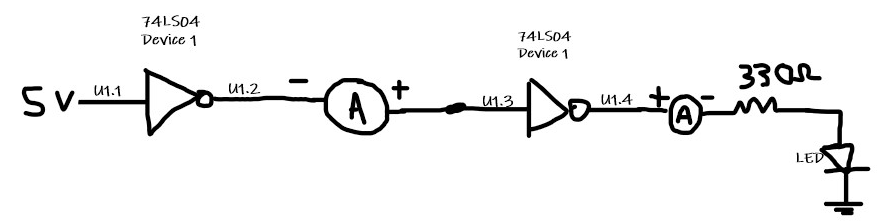


Figure 13: Current Measurements With Pin 1 At 5V Diagram

The breadboard was configured to reflect the circuit described in figure 13 to record current when pin 1 is at logic level low.

The current from pin 3 to pin 2 was measured to be 2μA which is IOH.

The current from pin 4, IOL, was found to be 7.44mA.

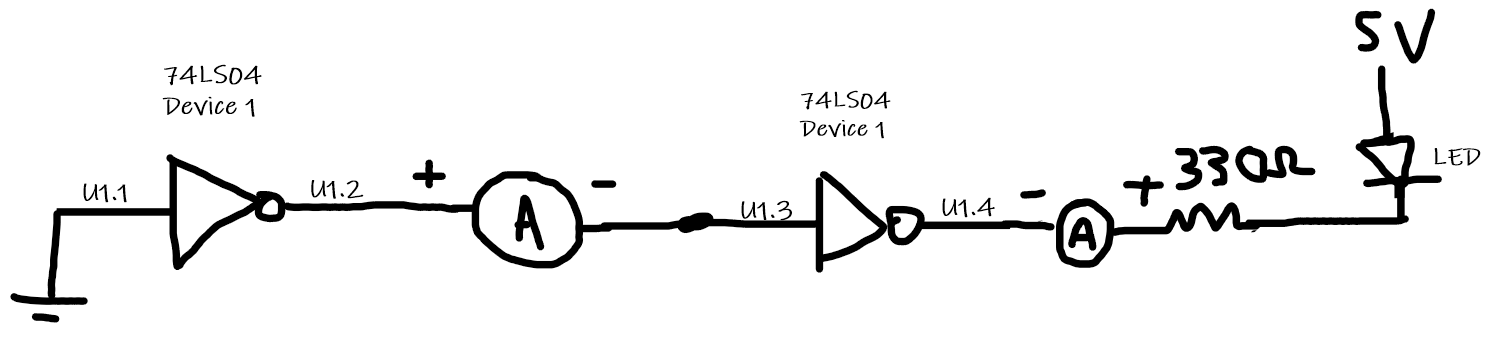
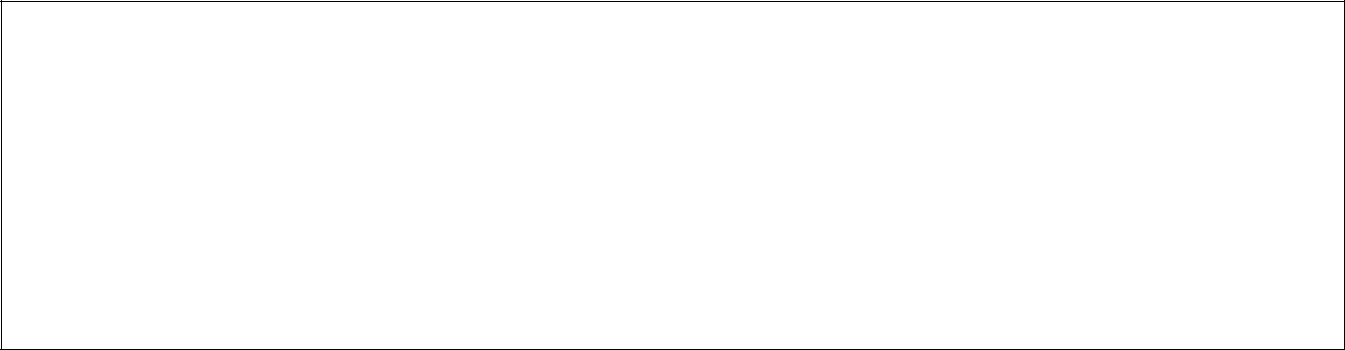


Figure 14: Current Measurements With Pin 1 At Ground Diagram

**Activity 1.5 — When logic gates are not properly connected**



We altered the circuit in figure 14 to represent the circuit in figure 15, this was done in order to determine the effects of leaving a logic gate floating, thus we choose to not connect the input of the first inverter directly to ground or 5V.

The voltage between pin 1 and ground was .06-.7V. The voltage between pins 2 and 3 and ground was 2.2 - 3.4V. The voltage between pin 4 and ground was .5-1.3V.

As shown in the data we see values, there is a large amount of variation which near the cut off voltages for determining if an input is high or low, which could result in voltage potentially moving out of range and the device no longer being able to function properly. Another point of interest is that the output of pin 2 has a minimum of 2.7 V when the input is considered low, which the voltage between pin 1 and ground was less than .8V, however the voltage found was in this range at times but out of range at times, and with pin 4 and ground the voltage was above the maximum output voltage for a low signal. It would not be wise to use a floating input due to the potential to damage the device by having the device output voltages it is not meant to and the variation could result in errors for the user.

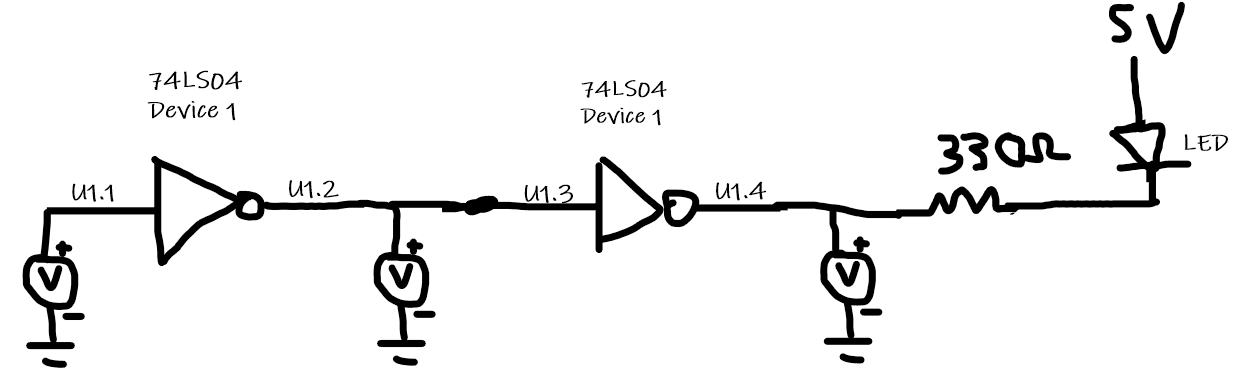


Figure 15: Voltage Measurements With Not Gate One Input Floating

We altered the circuit to represent the circuit in figure 15, we did this to determine the effects of shorting the outputs of two logic gates.

The voltage between pins 2/3 and ground was 2.3V and the current between pins 2 and 3 was 40mA.

As shown in the data, we can see that the voltage is close to the forbidden zone which runs the risk of registering as neither high nor low. The voltage that measured would not have normally been a normal output high voltage, since according to the data sheet of the 74LS04, the minimum high output voltage is 2.7V, thus showing that the chip is not working properly. In addition the current between pins 2 and 3 is 40mA and according to the data sheet, is five times the high output current that is recommended this would in turn damage the devices since the chips are working to output currents that are far greater than their recommended maxes, and producing irregular voltages.

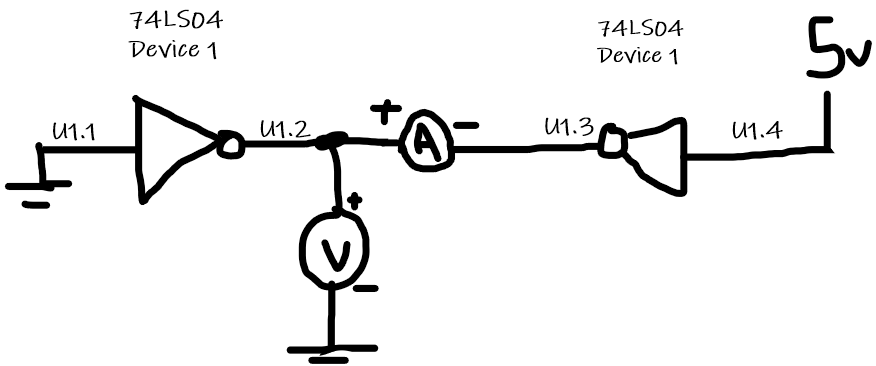
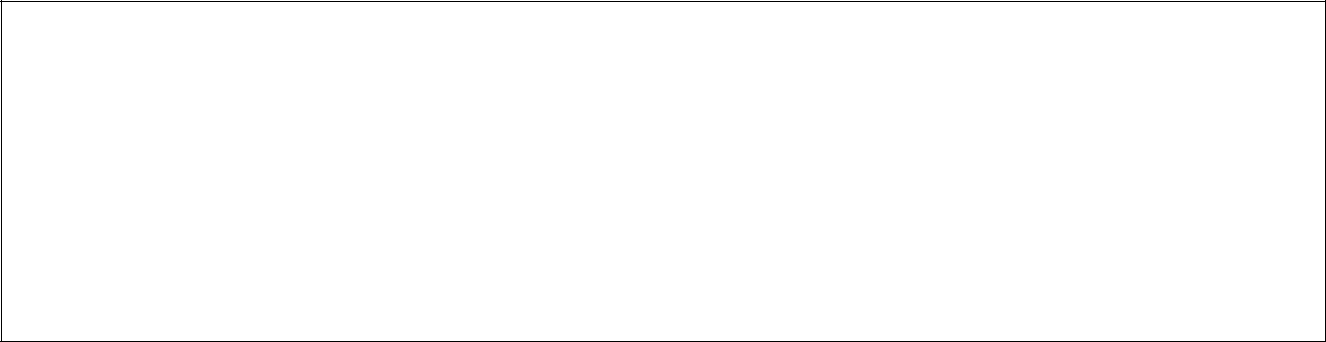


Figure 16: Voltage and Current Measurements With Outputs Connected

**Exercise 1.1 — Boolean algebra / Logic gates revisited**



For all gates discussed, the two inputs will be A and B, where A will be manipulated

1. For the AND gate:

If A is tied to logic zero then the logic function will become f(A,B) = 0 since the only way for a AND gate each input must be logic 1 for the output to be logic 1 and if any input is logic 0, then the output is logic 0.

If A is tied to logic one, then the logic function will become f(A,B) = B, since as mentioned before the AND gate will equal logic 1 if all inputs are logic level 1 and 0 if any input is logic level 0. However we can’t jump to that conclusion since B can be tied to logic 0 or logic 1 which will alter the output of the function, thus the function is solely dependent on B when A is tied to logic one.

1. For the OR gate:

If A is tied to logic zero then the logic function will become f(A,B) = B since for an OR gate only one input needs to be at logic one for the function to be at logic one, and due to the fact that A is at logic zero, then the function will follow the logic level of the B input.

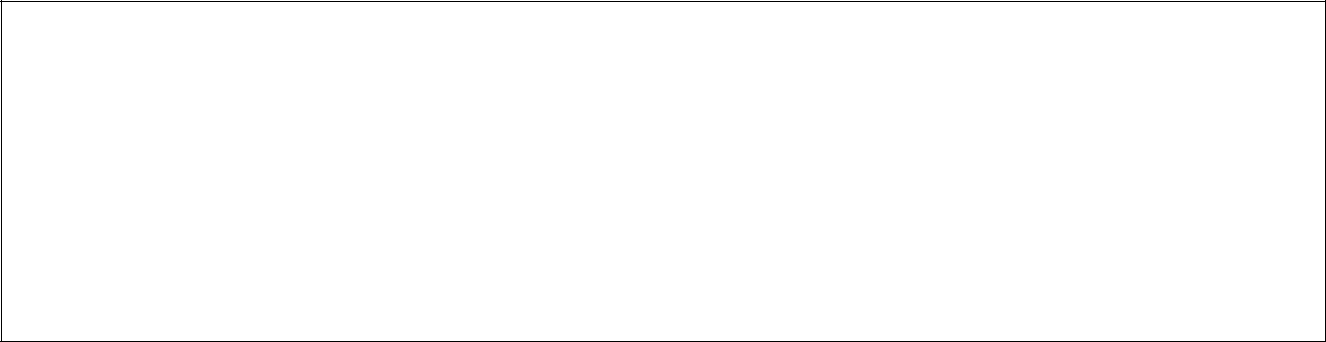
If A is tied to logic one, then the logic function will become f(A,B) = 1, since as mentioned before the OR gate will equal logic 1 if at least one input is at logic level one, thus B’s logic level would not affect the logic level of f

1. For the XOR gate:

If A is tied to logic zero then the logic function will become f(A,B) = B since the only way for an XOR gate is for one input to be logic 1 and logic 0, so if B is logic 1, then f will be at logic 1, if B is at logic 0, then f will be at logic 0.

If A is tied to logic one, then the logic function will become f(A,B) = B’, since as mentioned before the XOR gate will equal logic 1 the inputs are not the same, so if B is logic 1, then the output will be logic zero, and the output will be logic one if B is logic zero, so f will follow the compliment to the logic level of B.

**Exercise 1.2 — Data sheet of the 74LS04 device**



Using the data sheet of the 74LS04:

The minimum output voltage that is considered high (VOHmin) = 2.7V.

The maximum output voltage that is considered low (VOLmax) = .4V.

The minimum input voltage that is considered high (VIHmin) = 2V.

The maximum input voltage that is considered low (VILmax) = .8V.

The maximum output current that is considered high (IOHmax) = -.4mA.

The maximum output current that is considered low (IOLmax) = 8mA.

Some of the measurements that show out of range values in activity 1.4 are when pin 1 was at logic level low and IOL was found to be 1μA and IOH was found to be 7.57mA, the currents are out of range, I believe this is due to the LED resistor system causing a current to move out of the output. I am unsure of a fix; however, the voltages are in range and do not appear to be affected by the LED resistor load. Most of the measurements in activity 1.5 were out of range or close to being out of range, for the floating input a measurements the fix that would be employed is either connecting the input to ground or the input source to determine the exact state of the input of the first inverter. For the NOT gates that have their outputs connected the measurements the voltage from pins 2/3 was still in a position that is considered high, however it is close to the region where it may not be considered high or low, however the current was 40mA and as mentioned before it is a current much higher than what the NOT gate could output under normal circumstances, and the suggested fix is to simply not connect the gates in this way since it is contradictory to how logic gates work.