NAND/NOR EQUIVALENCY TRAINING BOARD

By:

Jeremy Diamond

Matthew LaGreca

Final Report for ECE 445, Senior Design, Fall 2017

TA: Xinrui Zhu

13th of December 2017

Project No. 34

Abstract

The NAND/NOR Equivalent Logic Circuit Training Tool is a visual aid where students can make 2 or 3 input circuit using provided AND OR and NOT gates, and then make the logical equivalent circuit out of provided NAND or NOR gates. After both circuits have been built, LEDs will light up denoting to the user if the two circuits are logically equivalent or not. After going through the process of building this device, we concluded that our design needs refining in terms of the materials and space used so that device is more portable as safer to the user while also being less prone to breaking.

Contents

1. Introduction
Coming into the world of Electrical and Computer Engineering, one of the introductory topics encountered is creating logic circuits using logic gate IC chips. While it might make sense to those new to the concept to use AND, OR, and NOT gates for the circuit design, it is quickly taught that designing the same circuit using NANDs or NORs is critical since they are more power efficient by utilizing less transistors. However, it can be difficult to learn the equivalency between the two circuits, thus we saw a need to simplify the learning process of equivalent circuits. This is what the NAND/NOR Equivalent Logic Circuit Training Tool comes to accomplish. Without the need to read datasheets or power IC chips, a user can practice their NAND/NOR equivalency knowledge using the board.

1.1 Background
Growing up in the Northwest suburbs of Chicago, both of our team members were lucky enough to go to high schools that had Project Lead the Way programs (4 years of high school engineering classes). One of the years taught Electrical Engineering and just like most people when they first learn EE we were taught how to use logic gate ICs. However, we both struggled in our learning of NAND and NOR gate only equivalents for AND, OR, NOT circuits. Upon entering the program at the University of Illinois, this topic was taught to us again. Being that we experienced the concept of NAND/NOR equivalency before relearning this material made it easy for us to excel in the concept, but this did not hold true for our fellow students who were experiencing a totally new idea. Flash forward to Senior Design, we came up with this idea for the NAND/NOR Equivalent Circuit Training Tool because we felt that this process of learning how to use NAND/NOR gates could be alleviated some, and that the way it is currently taught seems more complicated than necessary.

1.2 Objective
The objective of this project is to create a tool for those new to logic gate circuits. We wanted to make it so that creating circuits would be simple, and would not require any knowledge of how to use an IC chip or read a datasheet. With this device, we wanted to make it so that it would be easy to build a circuit using AND, OR, and NOT gates and then be able to easily toy around with NAND and NOR equivalents. In this learning tool, we wanted to make sure that when both circuits are built, the user has instant feedback as to if the two circuits that they created are logically equivalent and thus making it easier to learn NAND/NOR equivalency in logic circuits.

2 Design

Our final product is composed of three major modules that interface with each other to create a user-friendly experience. One module provides the necessary power to all the components within the board, the next module is the actual part of the board the user interacts with, and the final module evaluates the work done by the user and outputs visual indicators.

2.1 Block Diagram

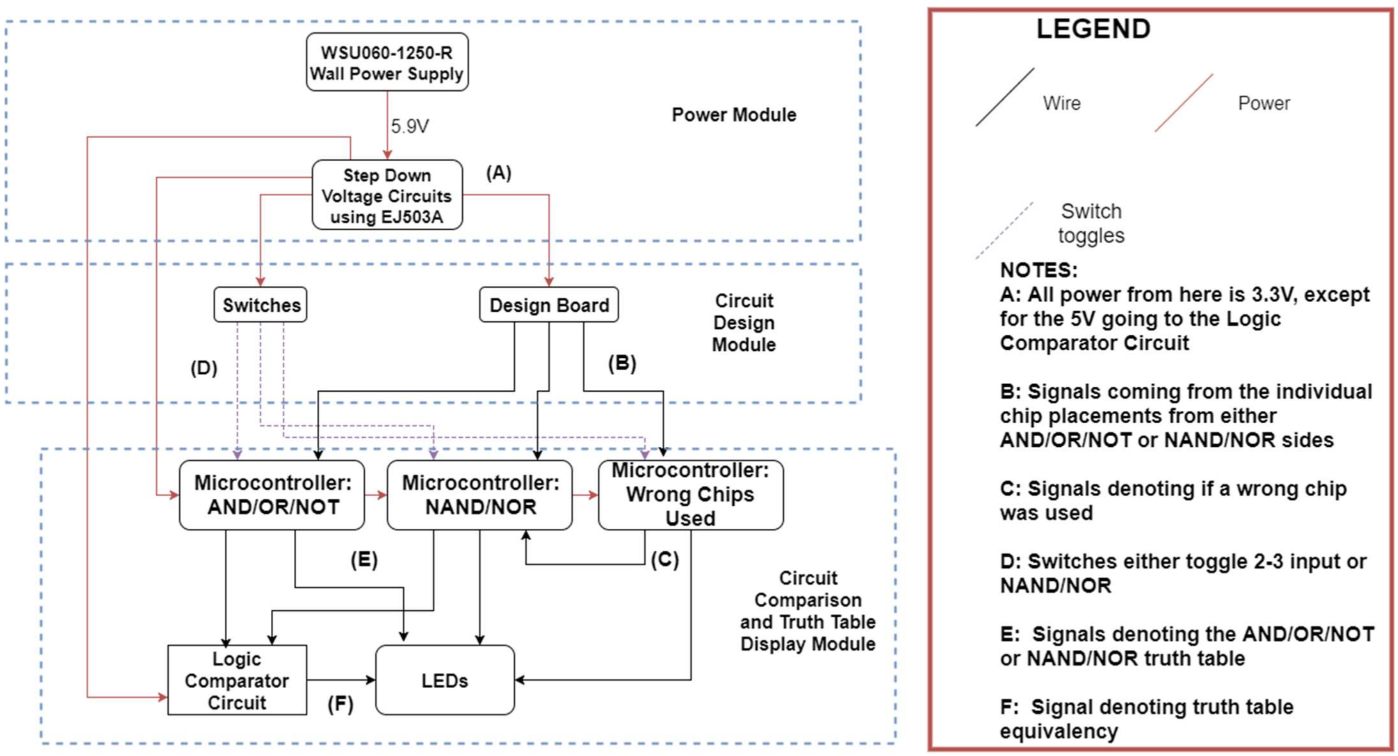

Figure 1: Overall Block Diagram for NAND/NOR Equivalency Training Board

As seen in Figure 1, our design is broken down into a Power Module, a Circuit Design Module, and a Circuit Comparison and Truth Table Display Module. The Power Module takes in power from a wall outlet and steps it down to two separate voltages, one voltage level for the circuit comparison components and another voltage provided to the microcontrollers and their inputs. The Circuit Design Module simply consists of the connections from the power board through either logic gate positions or the switches to the microcontroller inputs. The internal calculations and visual outputs are performed by the microcontrollers and logic comparator circuits of the Circuit Comparison and Truth Table Display Module.

2.1.1 Power Module

We had drawn part of our inspiration for this project from the work of a group in a previous semester. In their design they had used banana plugs to provide the power to their board which also requires a DC voltage supply, something not readily available to the average user. Taking this into consideration we wanted our board to be portable and easily powered anywhere a user could find an outlet. Our solution was to use a wall outlet with a barrel port connection as the power source of our board, a powering method almost everyone is familiar with. We selected an input value of 6V since it guaranteed we would have enough voltage at the input to power all the IC chips in our design.

Once we determined all the necessary components for our Circuit Comparison and Truth Table Display Module, we realized that we needed two voltage levels to power everything. The chips used for our circuit comparison circuit all required 5V to be powered. The other voltage level that had to be considered was for the microcontroller functionality. The voltage needed to power the microcontrollers could fall anywhere in the range of 1.8V to 3.6V, but the voltage that is chosen also determines the levels the microcontroller reads as a digital HIGH or LOW as well as the voltage of its outputs. Since our microcontroller outputs had to power on LEDs with turn-on voltages between 2V and 2.5V, we chose a value of 3.3V to power the microcontroller leaving enough extra voltage that we could dissipate through a current limiting resistor if necessary.

With the necessary voltages now known for the components of our board, we determined linear voltage regulators would take our input voltage and step it down to the correct levels.

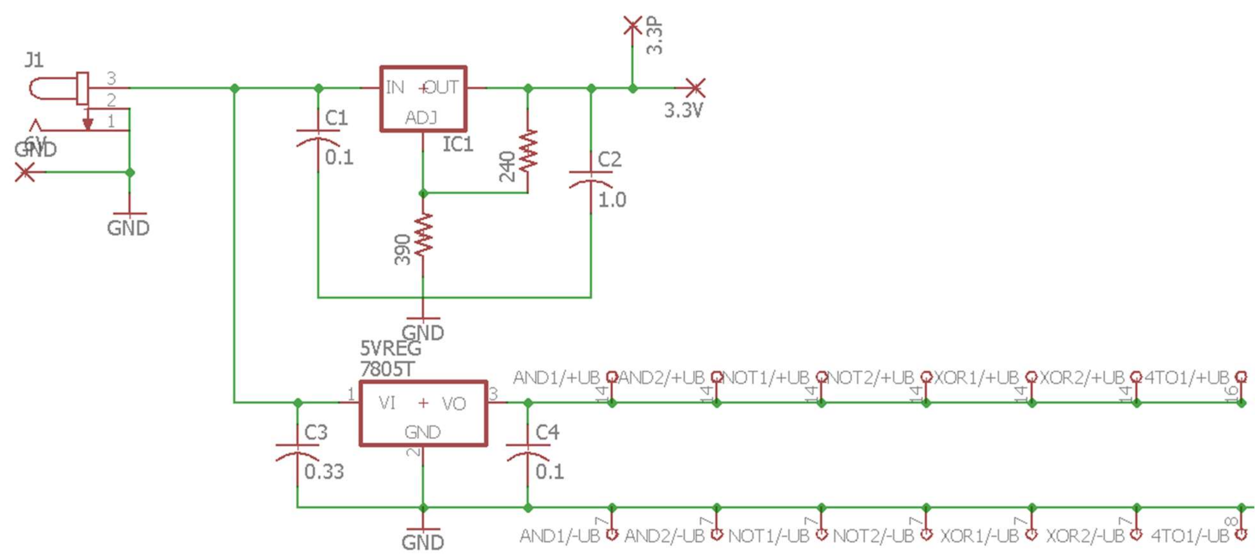

Figure 2: Power Module Schematic

Obtaining the 5V output simply required a LM7805 regulator as seen in the bottom circuit of Figure 2. This regulator will output a constant 5V, the exact level needed to power the IC chips of the logic comparator circuit. Achieving our 3.3V for the other components required a LM317 regulator, whose output voltage is adjustable. As seen in the top circuit of Figure 2, the resistor divider between the ADJ and OUT terminals of the regulator determine the output voltage. The exact equation is given by:

$$V_{out} = 1.25 \times (1 + \frac{R_2}{R_1})$$

For our design $R_2 = 390\Omega$ and $R_1 = 240\Omega$, giving us an output voltage of 3.28V.

2.1.2 Circuit Design Module

This module is the part of our board that lets the user interface with the underlying circuitry. The user places plastic pieces that represent logic gates into two separate circuit designs, one using AND, OR, and NOT gates and the other using either NAND or NOR gates. Also, this module contains a switch that alters the board functionality from 2-input circuits to 3-input circuits and a switch that allows the user to change between practicing NAND equivalent circuits or NOR equivalent circuits. It is the underlying circuitry that allows for the user's interaction with this module to produce the correct visual indicators.

For the microcontrollers of the Truth Table Display Module to output the correct table values, they must know which type of chip is placed in each position. Our solution for this concept was to pass either a digital HIGH or LOW to our microcontroller from each position, with each value representing a different type of chip. To pass the voltages from the Power Module to each of the microcontroller inputs was a straightforward process. Each gate position is composed of banana plug sockets, with one side of sockets taking voltages from the Power Module (3.3V and ~0V) and the other side outputting to the microcontroller inputs. Every piece then internally contained sets of banana plugs that were wired together, thus connecting either the 3.3V socket or the ~0V socket to the microcontroller input socket. This created a basic voltage loop and provided the correct voltage levels for each microcontroller to determine which gate piece was in each location.

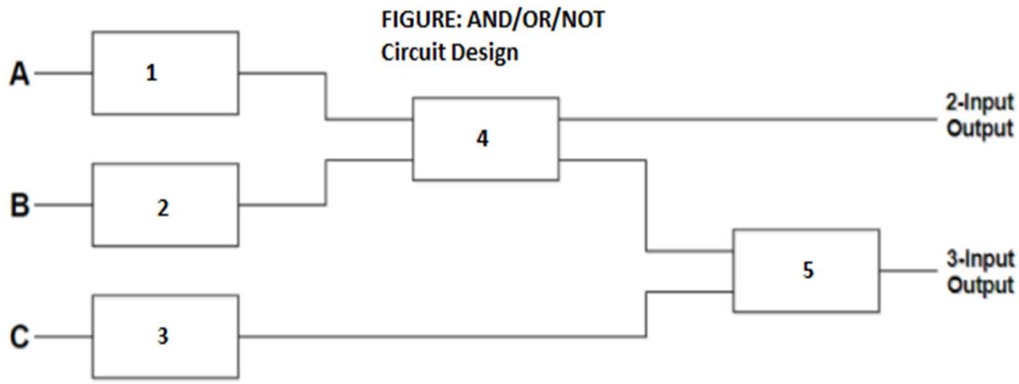

Figure 3: AND, OR, NOT Gate Placement Positions

Figures 3 and 4 show the layout of gate positions for the user to place pieces into. For the AND, OR, NOT circuit of Figure 3, positions 1 through 3 can hold either a through wire or a NOT gate and positions 4 and 5 can hold either AND or OR gates. For every possible combination of circuits to have an equivalent circuit, there are two additional spaces on the NAND/NOR circuit design. In Figure 4, positions 4 and 6 can hold either 2-input, 1-output NAND or NOR gates. All the remaining positions can be occupied by either a 1-input, 1-output (both inputs are the same) NAND or NOR gate or a through wire.

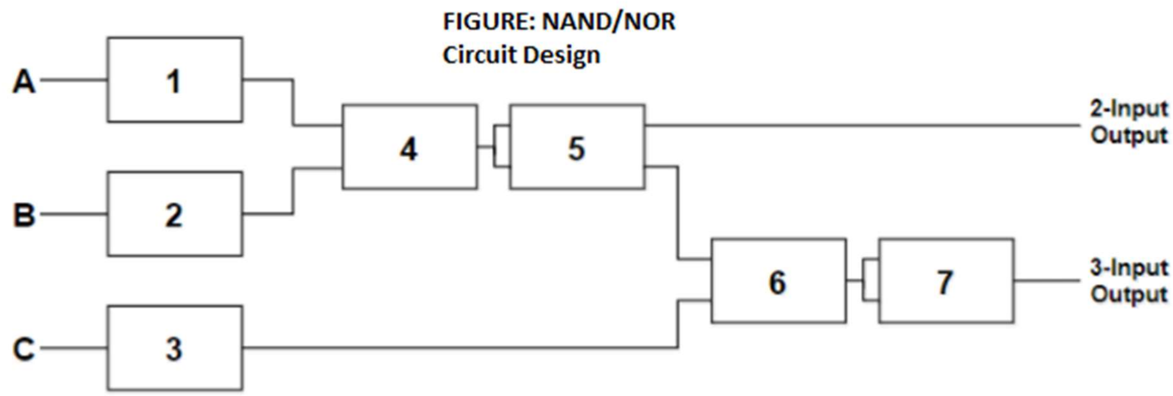

Figure 4: NAND/NOR Gate Placement Positions

A problem we encountered was that for certain positions on the NAND/NOR side of the board, there were three different logic gate possibilities: a NAND, a NOR, or a through-wire. Since the microcontroller can only determine whether it is HIGH or LOW, a third value would have been needed. We overcame this by using the NAND/NOR switch as an additional input to the NAND/NOR truth table microcontroller. By evaluating which mode it is in first, there is only two possibilities for the gate selection: a NAND/NOR gate or a through-wire.

Each of the gate positions for the AND, OR, NOT side of the board only provided one voltage level to the microcontroller that calculated its truth table. The gate positions on the NAND/NOR side of the board had to pass two voltages. One bit operated in the same function as the AND, OR, NOT side to produce the truth table, while both bits were passed to the wrong chip microcontroller to ensure that the correct type of gates were being used on the NAND/NOR side of the board. Our two switches also had to pass the correct voltage levels to the microcontroller. Using a SPDT switch, when put into the forward position the switches pass a HIGH voltage while in the down position the switches pass a LOW voltage.

2.1.3 Circuit Comparison and Truth Table Display Module

Our final module is the one that ties everything together and produces the visual indicators for the user to understand what they have built on the board. This module is broken down into three parts: truth table displays, NAND/NOR wrong chips display, and circuit equivalency display.

Truth Table Display:

Part of our design was to have instantaneous calculation of truth tables and equivalency display as the user adds pieces to the board. To properly evaluate the inputs from each gate position and instantly produce the correct truth table required the use of microcontrollers. Since our Power Module provided enough power to use three different microcontrollers, we implemented one each for the AND, OR, NOT truth table, NAND/NOR truth table, and the NAND/NOR wrong chip indicator.

As discussed in the Circuit Design Module, each of these microcontrollers takes inputs from the various gate positions on the board to determine the truth tables. In Figure 5 is our printed circuit design. On the left side of the board are all the inputs from each of the gate positions on the board. The

AND/OR/NOT microcontroller on top takes 6 inputs, one from each gate position and one from the input number switch. The NAND/NOR microcontroller at the bottom has 7 inputs: one from each position 1, 2, 3, 5, and 7 of Figure 5, one from the NAND/NOR switch, and one from the input number switch. Using these inputs, the NAND/NOR truth table will only output the correct number of rows based on the input number and the NAND or NOR truth table based on the position of the NAND/NOR switch.

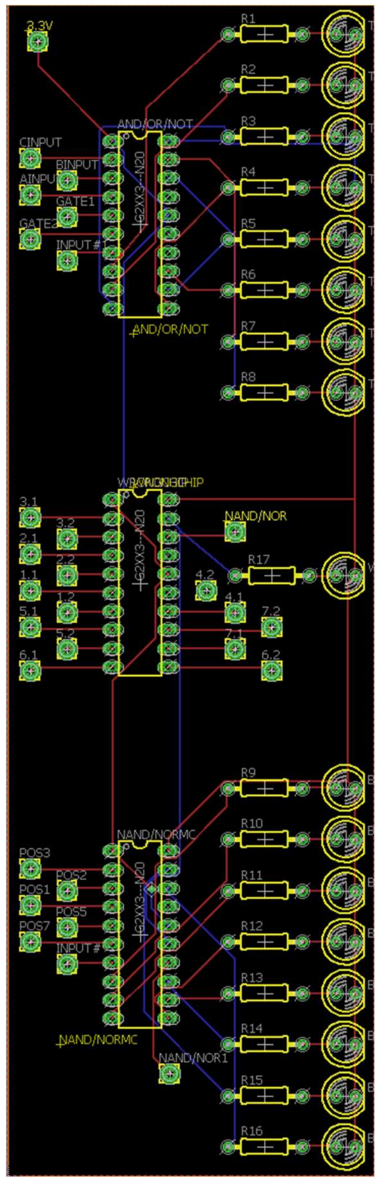

Figure 5: Truth Table and Wrong Chip PCB

The wrong chip microcontroller also checks on the NAND/NOR side of the board that the user is implementing the correct type of logic gates. As touched on in the background, the goal of NAND/NOR equivalents is to reduce the total number of IC chips for a circuit so using both NAND and NOR gates together eliminates the advantage. A red LED will light up when the wrong chip is used and all the NAND/NOR truth table will be blank, indicating to the user that they have made an error.

Circuit Equivalency Display:

The second part of this module is the circuit that compares whether the two circuits built are logically equivalent. This is accomplished using a basic logic circuit composed of NOT, XOR, and AND gates to determine if all active rows of the truth table are equivalent. In Figure 6 is the schematic that makes up the logic comparator circuit.

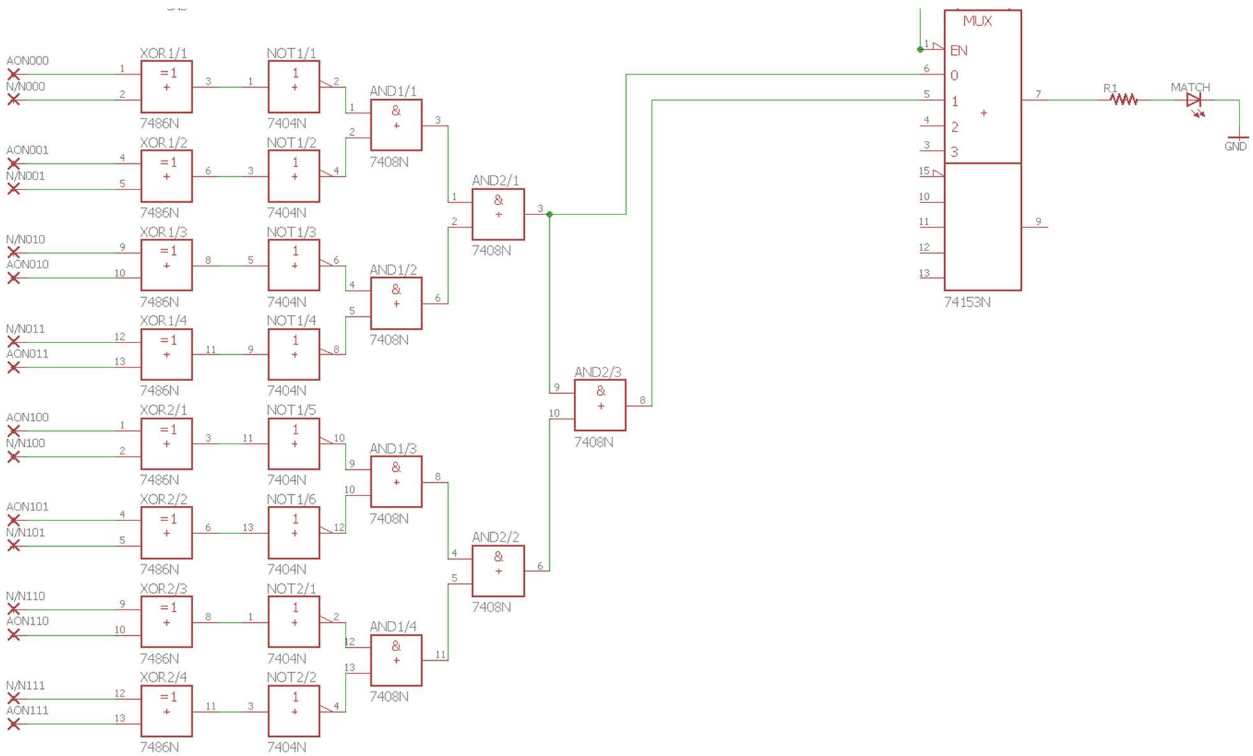

Figure 6: Logic Comparator Circuit Schematic

When the circuits are equivalent it will light up the LED seen on the right side of the schematic, thus letting the user know they successfully found an equivalent circuit.

2.2 Final Physical Design

2.2.1 Board Layout

In the design phase of our project, we had chosen a physical layout that we felt to be large enough to house all the components of our board. Figure 7 shows our original physical design, with overall dimensions of 12 inches long by 12 inches wide.

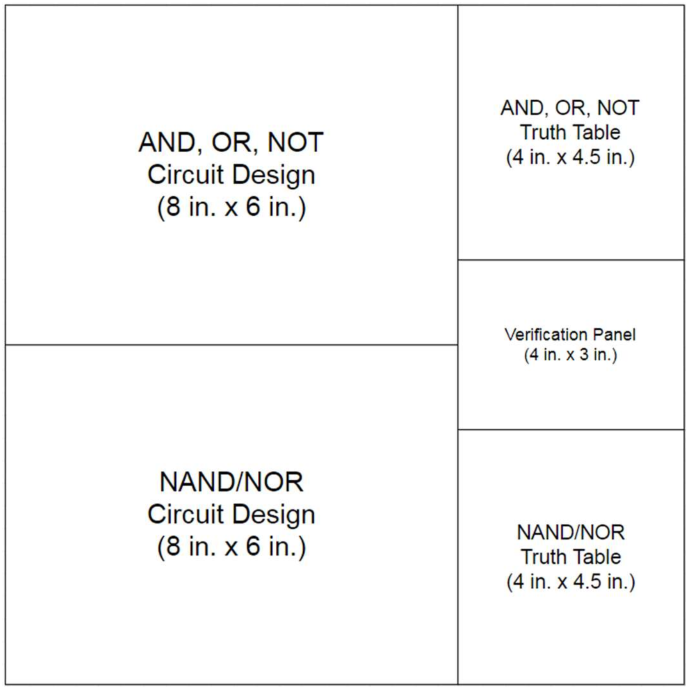

Figure 7: Original Physical Layout Design

By the time we got to the integration part of our project, we realized that our original design was not nearly large enough to house all the necessary components. The main driving factor of our physical layout was the size of the logic pieces for the NAND/NOR side of the board. Due to these pieces being large and there needing to be enough horizontal space to handle 5 pieces in a row, the final board layout was significantly larger than our original design. Our final design ended up being 13 inches wide by almost 24 inches long and can be seen in Figure 8.

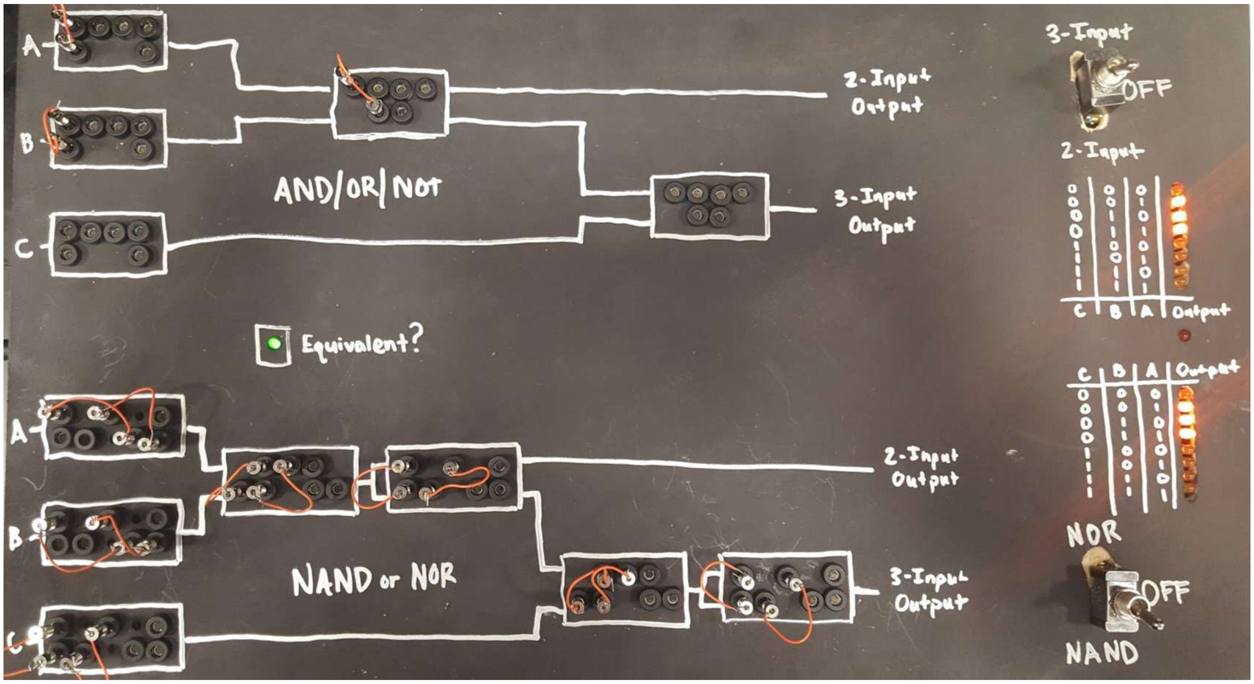

Figure 8: Final Board Physical Layout

2.2.2 Logic Pieces

For the user to implement the correct circuit without the use of IC chips requires pieces that represent their respective logic gates. For these pieces to work the banana plugs seen in Figure 8 are contained within the pieces, so they had to be tall enough to hide the wires and the plugs. The piece design is shown in Figure 9.

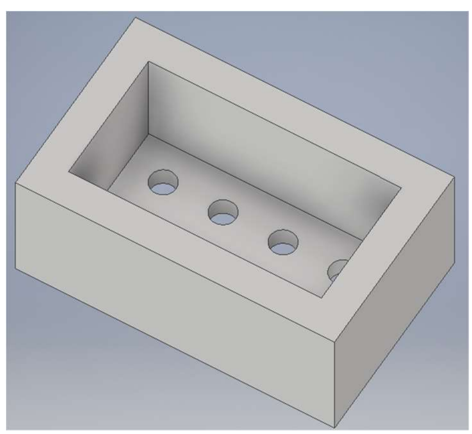

Figure 9: AND, OR, NOT Logic Piece 3D Design

Another function of our board to ensure that pieces from one side of the board can only be used on that side of the board. For this to happen the pieces for the NAND/NOR side of the are bigger than the pieces on the AND, OR, NOT side, with an additional two slots for banana plugs. Figures 10 and 11 show a top view of both pieces.

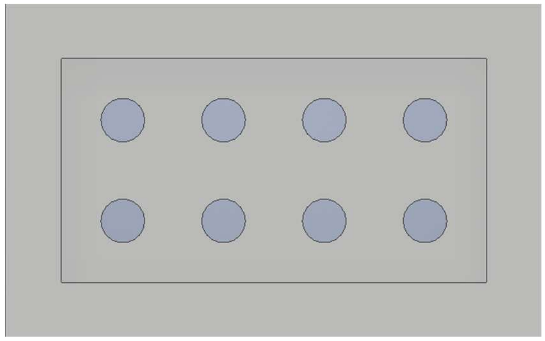
Figure 10: AND, OR, NOT Logic Piece Top View

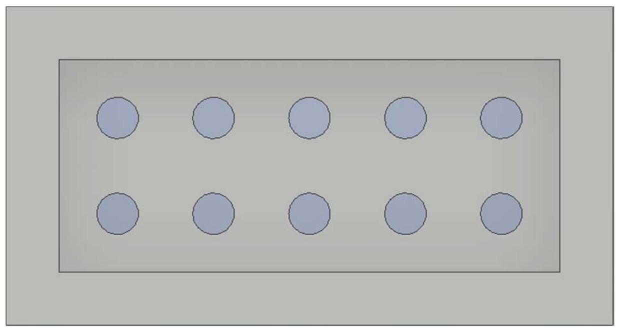

Figure 11: NAND/NOR Logic Piece Top View

3. Design Verification

3.1 Power Module

As discussed in our design section, the power module had two main requirements that needed to be met: a 3.3V output and a 5V output. We were able to verify these values based on the operation of the IC chips they were connected to. By measuring the input to the logic comparator circuit components and seeing that the circuit worked as designed, we were able to verify that the top portion of our power module in Figure 2 produced the correct 5V output. As for the 3.3V output we were able to verify the output by measuring the voltage at the outputs of our microcontroller. The microcontroller outputs at the same level it is powered at, which we were able to verify to 3.298V.

3.2 Circuit Design Module

The main portion of this module that needed to be verified was that when a NAND chip is placed when in NOR mode, the wrong chip LED should light up. The LED should light up if the opposite situation occurs.

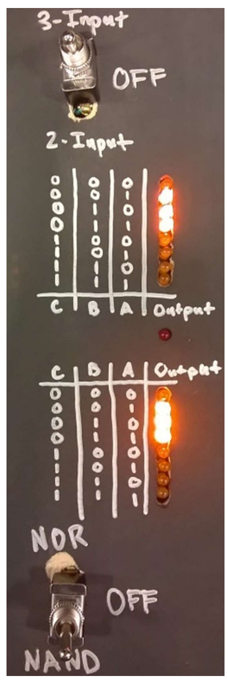

Figure 12: Truth Table and Wrong Chip Output in NAND Mode

Figure 12 shows the truth table LEDs while the switch is in NAND mode. As seen, the truth table for the NAND/NOR side of the board is lit up showing the correct truth table. Figure 13 shows that same truth table when the switch is placed in the NOR mode. The figure shows that the red Wrong Chip LED is on and all the LEDs for the NAND/NOR truth table are turned off.

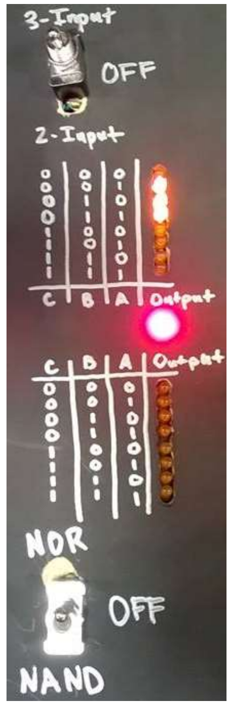

Figure 13: Truth Table and Wrong Chip Output in NOR Mode

3.3 Circuit Comparison and Truth Table Display Module

For our final module to be fully functional, there are two main requirements that need to be satisfied as seen in Tables 5 and 6. The verification of the truth table display LEDs can be seen in Figures 14 and 15.

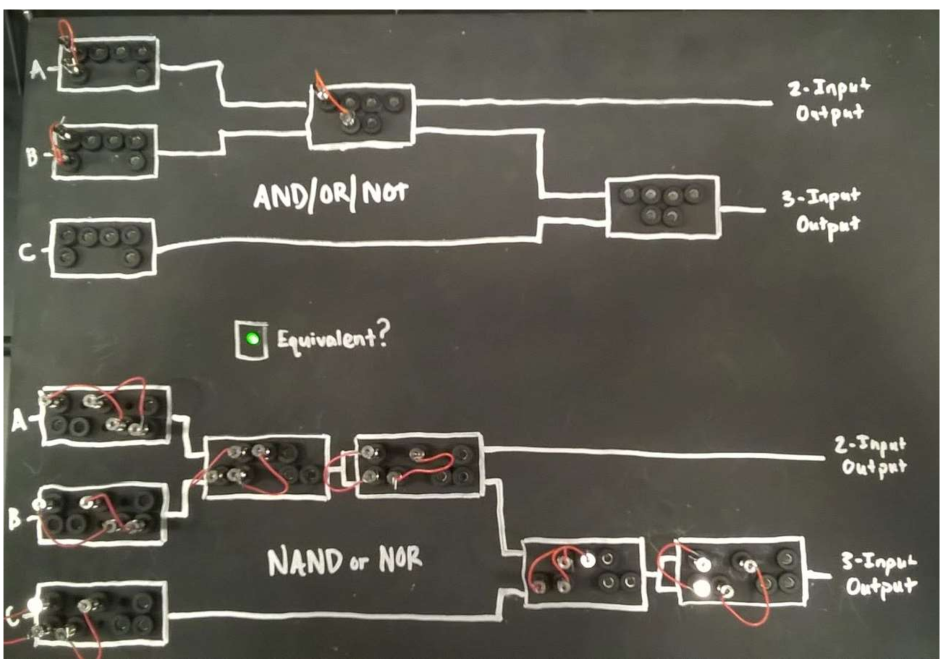

Figure 14: Physical Board with 2-Input OR Circuit and Equivalent NAND Circuit

Figure 14 shows the AND, OR, NOT circuit built with a through wire from inputs A and B to the logic gate and an OR gate in the placement. In the NAND/NOR portion of the circuit there is a 1-input, 1-output NAND gate at positions 1 and 2 of Figure 4, a through wire at position 5, and a 2-input, 1-output NAND gate at position 4. This produces the truth tables seen in Figure 15.

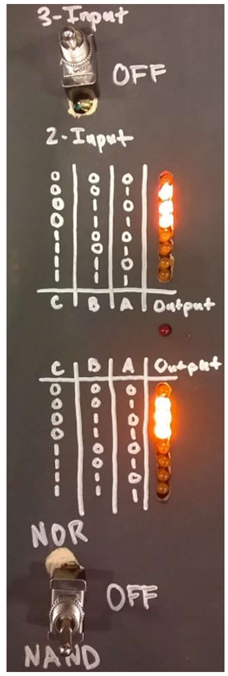

Figure 15: AND, OR, NOT and NAND/NOR Truth Tables

The other verification for this module was that the logic comparator circuit correctly lit the equivalency LED when the circuits had the same logical output. Based on the truth table shown in Figure 15, the equivalency LED is correctly lit as seen in Figure 14.

4. Costs

4.1 Parts

Table 1: Material and Parts Cost

Part	Manufacturer	Total Cost ($)
AND Gates	Texas Instruments	0.00
XOR Gates	Texas Instruments	0.00
NOT Gates	Texas Instruments	0.00
Dual 4-to-1 MUX	Texas Instruments	0.00
Microcontroller	Texas Instruments	20.00
Orange LEDs	N/A	3.20
Red LED	SparkFun	1.40
Green LED	SparkFun	0.85
SPDT Switch	GB	6.98
Banana Sockets	Cinch Connectivity Solutions	75.00
Banana Plugs	Cal Test Electronics	80.00
Power/Comparator Circuit Board	PCBway	49.00
Microcontroller/LED Circuit Board	PCBway	76.00
AC-DC Power Supply	Triad Magnetics	6.85
Board Power Connection	Memory Protection Devices	2.02
Wood Board		1.50
Total		**322.80**

4.2 Labor

Table 2: Labor Costs

Employee	Hourly Rate	Hours per Week	Number of Weeks	Total Cost
Matt LaGreca	$30.00	10	12	$3600 x 2.5 = $9000
Jeremy Diamond	$30.00	10	12	$3600 x 2.5 = $9000
Total:	$60.00	20	12	$18000.00

5. Conclusion

After the final integration of our project in Senior Design, there are things we learned that we would consider a success, and other things that we would work on should we continue with this project. We learned that due to our usage of the number the banana plugs for each gate chip, our prototype ended up being bigger than we would have liked. As such, we would reduce the redundancy of the number of plugs used in each gate, that way we can allow for less space to be used on future models, making the device much more portable for the user.

In addition to using less space, we'd like to use less wiring between our two current PCBs by combining both boards into one solid board. This would also allow for greater ease in the manufacturing of the device. Lastly, there was a lot of exposed wiring and such that the user could mess around with and potentially injure themselves or disconnect a critical wire. Safety is top priority for us, and we'd want to further encapsulate the device such that the user couldn't get into the wiring of the device too easily.

Coming from high schools that ran four-year engineering programs, we are very interested in bringing our product after implementing our design changes to show off to our past teachers. Hopefully with their feedback and additional feedback by letting their students try the device we will come out with a highly useful product that will enhance the learning process.

Appendix A Requirement and Verification Tables

Table 3: Power Module Requirement and Verifications

Requirement	Verification	Points
Outputs a voltage of 5V for the design board, AND chips, and 4-1 MUX, and 3.3V for the microcontroller, making sure that the microcontroller voltage never exceeds 3.6V and the design board never exceeds 5.25V	A. Power on the power module and use a multimeter to measure the open circuit voltage to see if it is 5.9V +/- 0.1 V B. Then using the same multimeter, measure the voltage across the 5V voltage supplier and make sure it is less than 5.25V C. Finally, use the multimeter to measure the voltage coming from the 3.3V voltage supplier and make sure it is less than 3.6V	25

Table 4: Wrong Chip Identifier Requirement and Verification

Requirement	Verification	Points
Microcontroller properly decodes gate identifier and mode signals, and lights up an LED if a gate is placed on the wrong side of the board.	A. Plug a NAND gate into the NAND/NOR side during NOR mode, see if the wrong chip LED goes off. B. Plug a NOR gate into the NAND/NOR side during NAND mode, see if the wrong chip LED goes off.	15

Table 5: Truth Table Display Requirement and Verification

Requirement	Verification	Points
Once all the required spots are filled, the microcontroller outputs the correct truth table for the AND/OR/NOT circuit, and similarly for the NAND/NOR circuit.	A. If all spots are not filled, a default output happens B. For a 2-input circuit, calculate an expected truth table beforehand using a K-Map/Boolean Algebra C. Build said circuit on Design Board using AND/OR/NOT gates D. Compare microcontroller truth table with expected E. Repeat B-D but with a NAND/NOR design F. Repeat B-E but using 3 input mode	35

Table 6: Logic Comparator Circuit Requirement and Verification

Requirement	Verification	Points
Show that when the truth tables are equivalent a green light goes on and when they do not match a red light goes on.	A. Apply a logic HIGH of 5V to all inputs of the logic comparator circuit and verify that the green light goes on B. Apply a logic LOW of .5V to all inputs of the logic comparator circuit and verify that the green light goes on C. Apply a logic HIGH of 5V only to the eight inputs that represent the NAND/NOR side of the board, leaving every other input as a LOW of .5V and verify that the red light goes on	25