NAND/NOR Logic Gate Equivalent Training Tool Design Document

Team 34

TA: Xinrui Zhu

ECE 445 - Fall 2017

Jeremy Diamond and Matthew LaGreca

Table of Contents

1.0 INTRODUCTION

1.1 Objective

1.2 Background

1.3 High-level Requirements List

2.0 DESIGN

2.1 Block Diagram

2.2 Physical Design

2.3 Block Design

2.4 Tolerance Analysis

3.0 COST AND SCHEDULE

3.1 Cost Analysis

3.2 Schedule

4.0 ETHICS AND SAFETY

5.0 CITATIONS

1.0 Introduction

1.1 Objective: It can be difficult to learn circuit equivalents going from a design using ANDs, ORs, and NOT gates to a design using only NAND or only NOR gates. As a proposed solution to simplify the learning process of translating to NAND/NOR equivalent circuits, our goal is to design a learning board that will be split into two halves: one half will only allow the use of AND, OR, and NOT gates to build an original circuit and the other half will allow only the use of NAND or NOR to build the equivalent circuit. All gates will be represented by small pieces, instead of IC chips, to take out any necessary base knowledge of IC chips. Each halve will output to a truth table based on the circuit built. The two truth table outputs will be compared to each other to see if the two circuits built are logically equivalent; if the two circuits are equivalent a green LED will light up, and if not a red LED will light up. Also, based on the original circuit there will be a display showing the number of NAND/NOR gates for the most simplified version for the user to try and match on the NAND/NOR side of the board.

1.2 Background: In beginner Electrical and Computer Engineering courses, students are tasked with learning how to design basic circuits using TTL gates and then implementing those designs on breadboards. In the beginning these circuit designs are completed with AND, OR, and NOT gates. The use of many types of gates creates an inefficiency in design since an unnecessary amount of IC chips must be used. To reduce the number of IC chips, NAND or NOR gates are used to replace all of the AND, OR, and NOT gates. The circuits can now be built out of entirely NAND or NOR gates, thus reducing the total number of IC chips. A problem that many introductory students encounter with this conversion is the use of IC chips, whether it be reading their datasheets or making sure power is supplied. Our design uses simple pieces that represent the gates to avoid the need for the user to understand IC chips. Taking focus off of the IC chips allows for the user to focus on finding the NAND/NOR equivalent.

1.3 High-level Requirements List:

- Based on the position of the switch, the NAND/NOR side of the board will only accept either NAND or NOR gates and never both
- The training board must have comparator logic abstracted away to compare the two halves to make sure that the two circuits designed are logically equivalent.
- The correct number of rows are lit up in the truth table based on the position of the switch setting the circuit as 2-input or 3-input

2.0 Design

2.1 Block Diagram: Our design consists of three modules that work together for a complete training board. The power module takes in power from the wall and performs the necessary step downs to the correct voltages to power each of the components in the board. The circuit design module consists of two parts, one for the AND, OR, NOT original circuit and one for the NAND/NOR equivalent circuit. There is a switch here to tell the board whether or not it is looking for a 2- or 3-input circuit. A microcontroller here will also decide whether or not the right pieces are used on the NAND/NOR side of the board and output the correct logic to fill the truth tables. Finally, is our circuit comparison and truth table display module which has the flip-flops holding the correct values for the truth tables. Also, this module has the comparison circuit to tell if the two sides of the board are logically equivalent.

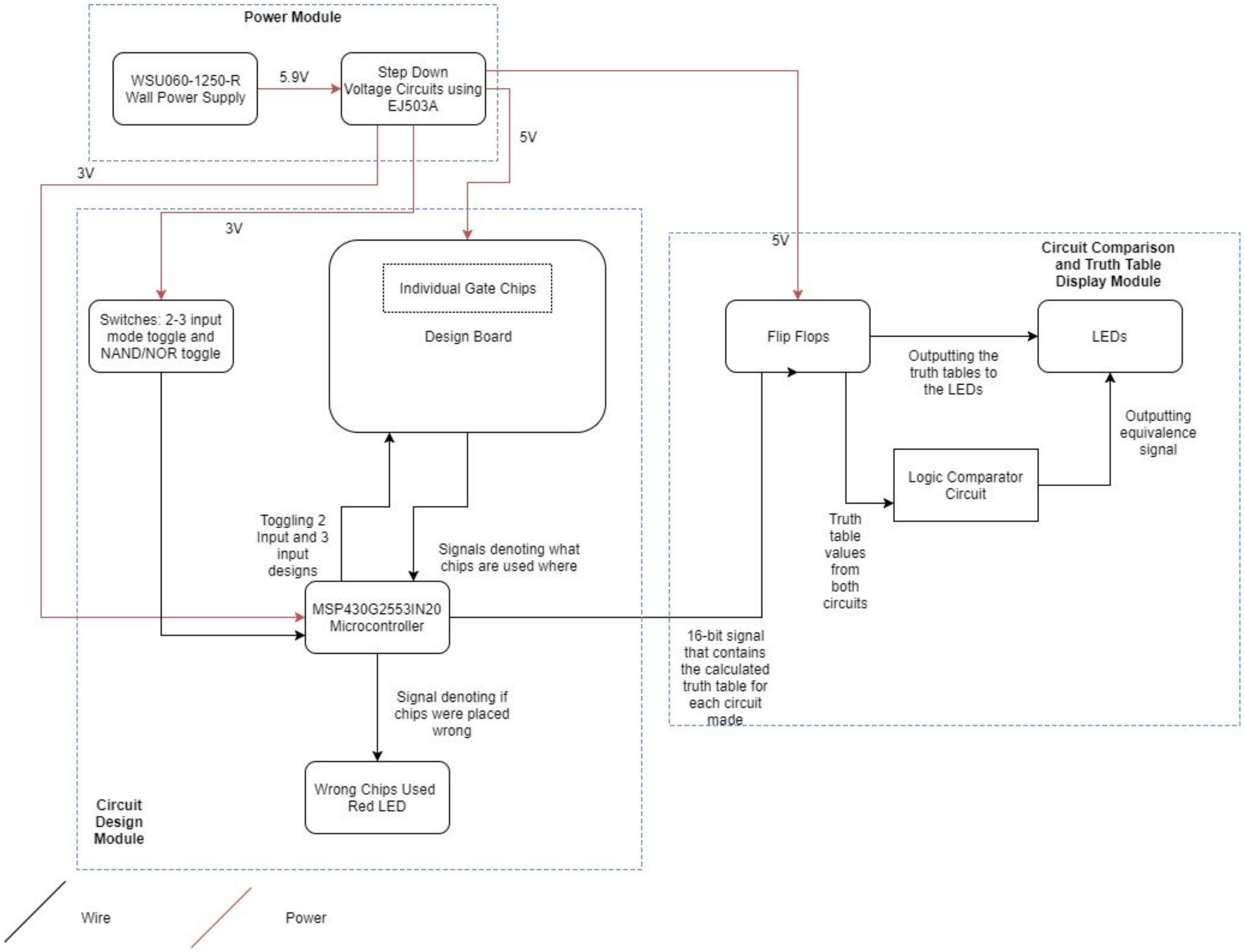
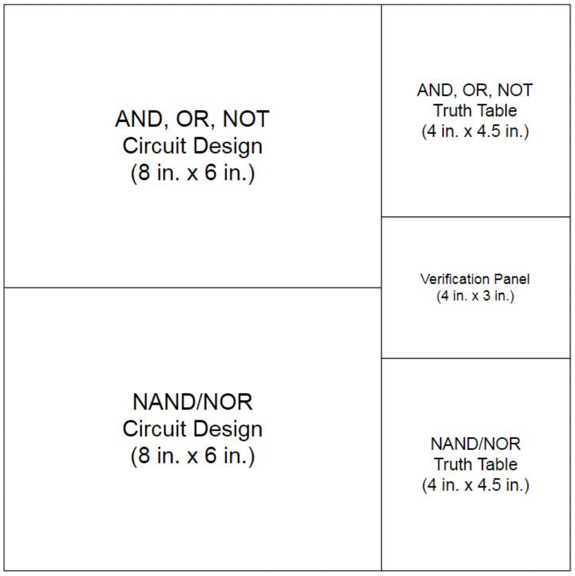

Power Module
WSU060-1250-R Wall Power Supply
5.9V
Step Down Voltage Circuits using EJ503A
5V
3V
3V
5V
Individual Gate Chips
Design Board
Switches: 2-3 input mode toggle and NAND/NOR toggle
Toggling 2 Input and 3 input designs
Signals denoting what chips are used where
MSP430G2553IN20 Microcontroller
Signal denoting if chips were placed wrong
Wrong Chips Used Red LED
Circuit Design Module
Circuit Comparison and Truth Table Display Module
Flip Flops
Outputting the truth tables to the LEDs
LEDs
Outputting equivalence signal
Logic Comparator Circuit
Truth table values from both circuits
16-bit signal that contains the calculated truth table for each circuit made
Wire
Power

Figure 1: Block Diagram

2.2 Physical Design:

Our training board is to be a total of 12 inches wide by 12 inches long with a thickness of no more than 1.5 inches. The board is composed of 5 main sections: two to build the original and equivalent circuits, two to display the truth tables for the circuits built, and one verification panel which will display if the two circuits are equivalent and if the NAND/NOR is in its most

Figure 2: Physical Design



simple form. The correctness of the equivalent circuit will be shown by a GREEN light, with an incorrect circuit displaying a RED light. The truth tables will be that for a three-input, one-output circuit and the first four rows will light up for a 2-input circuit and all eight rows will light up for a 3-input circuit.. Our verification panel will consist of a red LED, a green LED, a switch between 2-input and 3-input, and a hexadecimal display (for minimum number of NAND/NOR gates required).

2.2.1 AND, OR, NOT Circuit Design

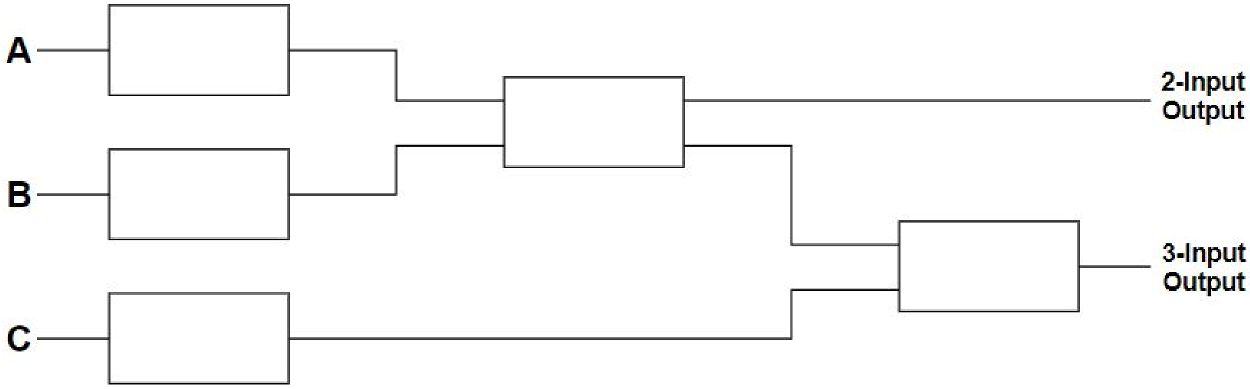


Figure 3: AND, OR, NOT Side Chip Placements

This half of the board will have five spaces that can hold pieces representing logic gates. The three spaces connected to each input can only hold NOT logic pieces, while the other two spaces can hold either AND or OR logic pieces. When the switch is put to 2-input the truth table will be built based on the ‘2-Input Output’ and when the switch is thrown the truth table will be built based on the ‘3-Input Output’.

2.2.2 NAND/NOR Circuit Design

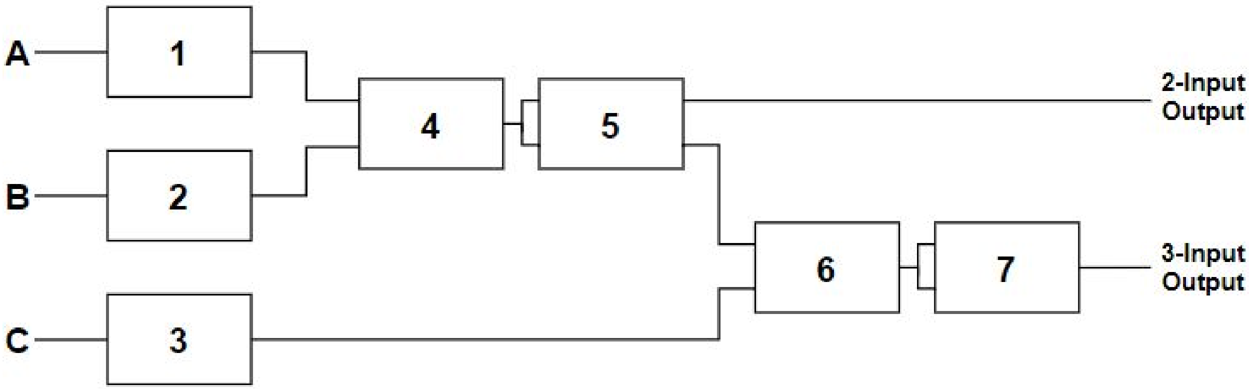
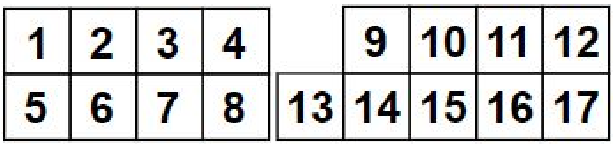


Figure 4: NAND/NOR Side Chip Placements

This half of the board will have seven spaces that can hold pieces representing logic gates. Spaces 1, 2, 3, 5, and 7 can only hold one input NAND or NOR (i.e. both inputs are the same) pieces. The other two spaces (4 and 6) can only hold the normal two-input NAND or NOR gates. When the switch is put to 2-input the truth table will be built based on the '2-Input Output' and when the switch is thrown the truth table will be built based on the '3-Input Output'. There will also be an additional switch and red LED on this side of the board that will say whether NAND or NOR pieces should be used. This side of the board will never accept both types of gates and will light up the red LED if the wrong pieces are used.

2.2.3 Logic Pieces and Board Placement

Each location on the board for the logic pieces will consist of an array of banana plug inserts. The plug placement on the individual pieces and the arrays have been designed to guarantee that only the correct types of pieces are put on each side of the board. These are the arrays:



1	2	3	4
5	6	7	8

	9	10	11	12
13	14	15	16	17

Figure 5: AND, OR, NOT (left) and NAND/NOR (right) Banana Plug Connection Points

There are nine different pieces that can be placed on the board and four types of connection points to ensure the correct operation of the board. There are two pieces that represent a wire instead of a gate, the top one for the AND, OR, and NOT side of the board and the bottom one for the NAND/NOR side of the board.

Equivalent Logic Gate	Piece Banana Plug Positions	Board Banana Plug Connection Array
AND	1 2 4 8	1 2 3 4 8
OR	1 3 4 8	1 2 3 4 8
NOT	1 5 6 8	1 5 6 7 8
Top - Wire	1 5 7 8	1 5 6 7 8
NAND (1-Input)	9 12 13 17	9 10 11 12 13 17
NOR (1-Input)	10 12 13 17	9 10 11 12 13 17
NAND (2-Input)	12 13 14 17	12 13 14 15 16 17
NOR (2-Input)	12 13 15 17	12 13 14 15 16 17
Bottom - Wire	11 12 13 17	9 10 11 12 13 17

Table 1: Exact Banana Plug Connections

2.3 Block Design:

2.3.1 Requirements and Verification

a) Power Module

The power module provides the rest of the circuit with a maximum of 5.9V at all times with the use of the WSU060-1250-R Wall Power Supply. The plug of the WSU will go into the EJ503A Board Power Connection, and from there the EJ503A will be connected to some voltage divider circuits all connected in parallel, each having a voltage drop of 5.9V. Through the use of these voltage dividers, the design board receives an input voltage of anywhere from 4.5V to 5.25 volts, as well as the flip flops, AND chips, and the 2-1 MUXes, the while the micro controller inputs receive a maximum of 3.6V and a minimum of 1.8V.

Requirement	Verification
Outputs a voltage of 5V for the design board, flip flops, AND chips, and 2-1 MUXes, and 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 B. Then using the same multimeter, measure the voltage across the 5V voltage supplier and make sure it it is less than 5.25V C. Finally, use the multimeter to measure the voltage coming from the 3V voltage supplier and make sure it is less than 3.6V

b) Circuit Comparison and Truth Table Display Module

When the user places chips down to be used in the AND/OR/NOT circuit, as well as when they place chips down for the NAND/NOR equivalent, the microcontroller will automatically calculate the truth tables for the circuits. Whenever the appropriate spots are filled for the 2-input/3-input circuits, an 8-bit/16-bit signal coming from the microcontroller will be sent as the input to a set of flip flops along with a clock signal to be used to show the truth tables of their designed circuits. The outputs of the flip flops will be connected to a set of LEDs as well as a logic comparator circuit. The purpose of the LEDs will be to show the individual truth table values for the circuits designed, while the logic comparator will very clearly tell the user whether or not the two circuits designed are logically equivalent.

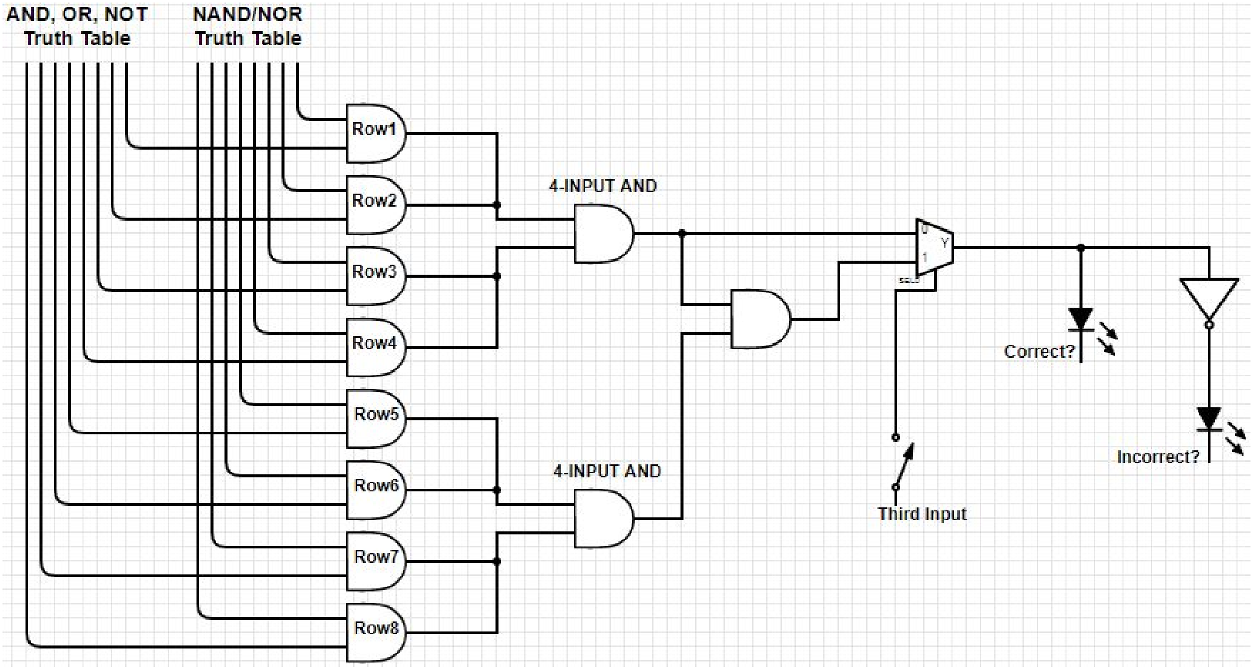


Figure 6: Logic Comparator Circuit

Requirement	Verification
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 clock pulse to all 16 flip-flops that hold the values for the two truth tables B. Apply a logic HIGH of 5V to all flip flops and verify that the green light goes on C. Apply a logic LOW of .5V to all flip flops and verify that the green light goes on D. Apply a logic HIGH of 5V only to the eight flip flops that represent the NAND/NOR side of the board and verify that the red light goes on

c) Circuit Design Module

i) Proper Custom Gate Chip Usage

The circuit design board is split into two halves; the left half is only supposed to use AND, OR, and NOT gate chips, while the right half is only supposed to use NAND and NOR gate chips. The NAND/NOR side is constrained even further with the switch that determines the NAND/NOR side is in NOR mode or NAND mode. If the user decides to put a NAND on the NAND/NOR side while it is in NOR mode, or if the user decides to put a NOR on the NAND/NOR side while it is in NAND mode, the microcontroller is programmed to tell the user that they used a NAND or a NOR when they should only be using NORs/NANDs to complete the circuit equivalent, respectively. The way that the microcontroller is able to tell when the chips are placed incorrectly, is that each gate that we are providing to the user has a specific banana plug configuration to put into the board, and a signal will be sent to the microcontroller based on which plugs are used that tells the microcontroller which gate is being used. The microcontroller then does some decoding of the chip and which mode it was placed on, and then an LED lights up if the chip was placed incorrectly.

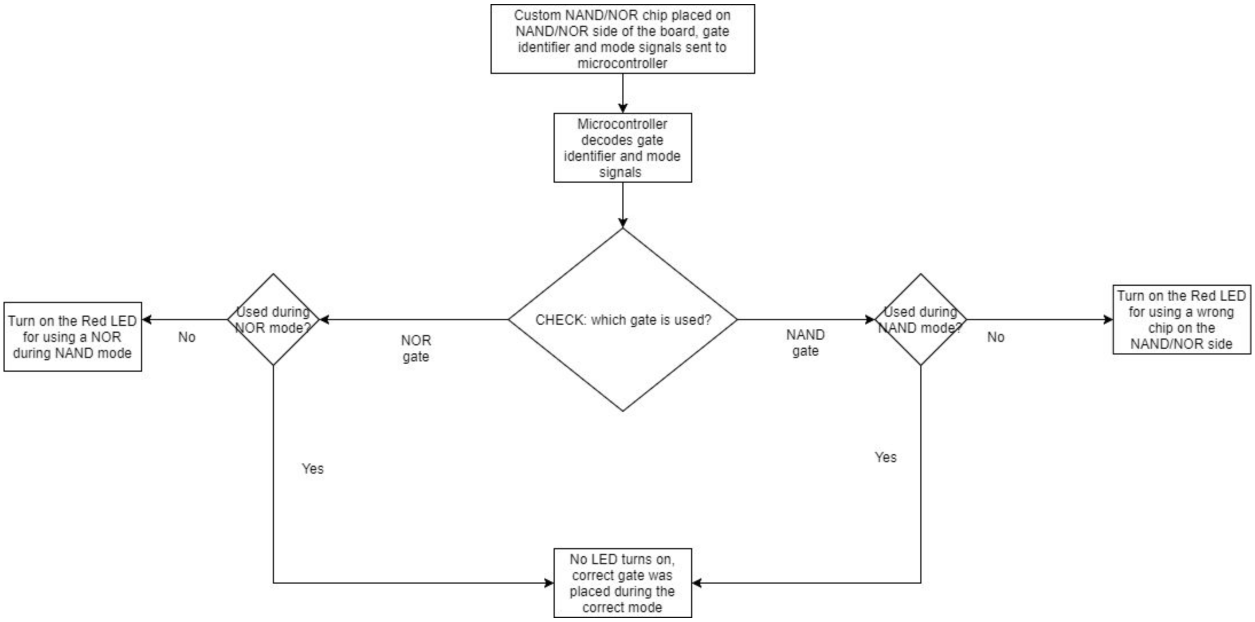

Custom NAND/NOR chip placed on NAND/NOR side of the board, gate identifier and mode signals sent to microcontroller
Microcontroller decodes gate identifier and mode signals
CHECK: which gate is used?
NOR gate
Used during NOR mode?
No
Turn on the Red LED for using a NOR during NAND mode
NAND gate
Used during NAND mode?
No
Turn on the Red LED for using a wrong chip on the NAND/NOR side
Yes
Yes
No LED turns on, correct gate was placed during the correct mode

Figure 7: Software Flow Chart: Chip Placed Incorrectly?

Requirement	**Verification**
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.

ii) Mode select and calculating truth tables

When the power supply is plugged in, depending on the position of the switches, the microcontroller will decide whether the mode for the design board is for a 2 input design or a 3 input design, as well as decide if the NAND/NOR side should be using NANDs only or NORs only. If the mode is set to a 2 input design, then the user will have to put gates into all 3 usable slots for the 2 input mode. Similarly, if the mode is set to a 3 input design, the user will have to put gates into all 6 usable slots for 3 input mode. Once all of the slots are filled for whatever mode the board is currently set to, the microcontroller will calculate the truth table of the design circuit, and output the calculated truth table to the flip-flops corresponding to the truth table storage of the designed circuit (AND/OR/NOT or NAND/NOR).

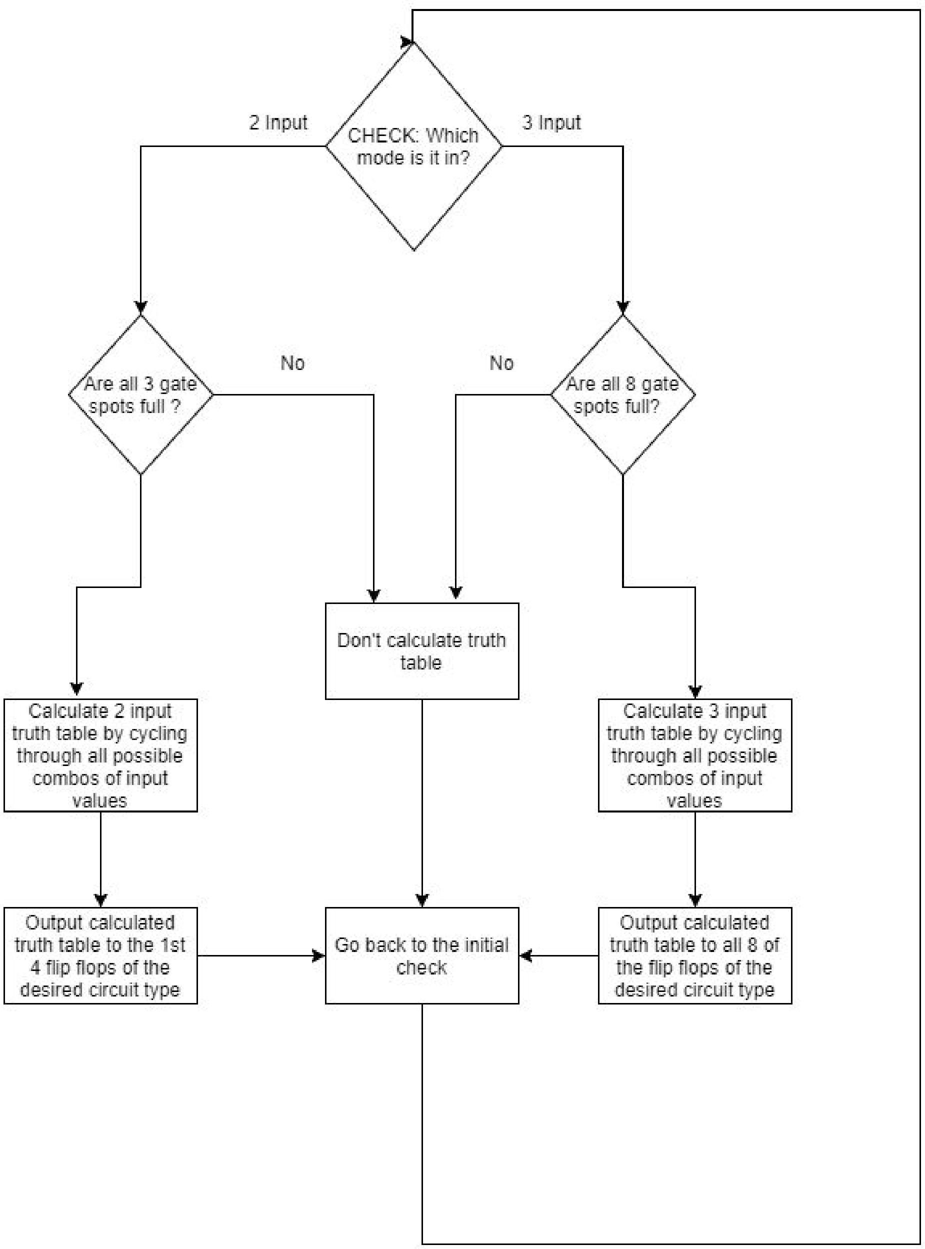

2 Input
CHECK: Which mode is it in?
3 Input
Are all 3 gate spots full ?
No
No
Are all 8 gate spots full?
Don't calculate truth table
Calculate 2 input truth table by cycling through all possible combos of input values
Calculate 3 input truth table by cycling through all possible combos of input values
Output calculated truth table to the 1st 4 flip flops of the desired circuit type
Go back to the initial check
Output calculated truth table to all 8 of the flip flops of the desired circuit type

Figure 8: Software Flow Chart: Calculating Truth Tables

Requirement	Verification
Once all the required spots are filled, the microcontroller outputs the correct truth table for the AND/OR/NOT circuit to the AND/OR/NOT flip flops, and similarly for the NAND/NOR circuit.	A. If not all spots are filled, (default) 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

NOTE: “Default” is behavior for when a truth table is not being calculated. The truth table displays all 0s, and a special LED lights up.

2.4 Tolerance Analysis:

In order for our project to run as smoothly as possible, the power supply must be able to provide a stable output voltage of 5.9V and a certain number of amps. This is because it has to keep powered the microcontroller, all of the gates, the flip flops, and be able to provide enough current so that 21 LEDs can be illuminated at all times. To check this, we need to plug all of these components into our circuit and then we measure the voltage drops across everything, and measure the current being supplied by the power supply. We should run this test over an extended period of time, probably around 5 hours, and see how stable the values are. If we see fluctuations in the current supplied or the voltage drops across each element, we will probably have to decide on another power supply. This project ideally has its consumer base originating from a teaching setting, so these devices probably need to last throughout a school day, multiple times a year.

3.0 Cost and Schedule

3.1 Cost Analysis:

Labor -

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

Table 2: Labor Costs

Parts -

Description	Manufacturer	Part #	Quantity	Cost
Microcontroller	Texas Instruments	MSP430G2553IN20	1	$10.00 (with development kit)
AC- DC Power Supply	Triad Magnetics	WSU060-1250	1	$6.85
Board Power Connection	Memory Protection Devices	EJ503A	1	$2.02
White LED	N/A	N/A	48	$12.00
Blue LED	N/A	N/A	16	$8.00
Orange LED	N/A	N/A	16	$4.00
Red LED	N/A	N/A	4	$1.40
Green LED	N/A	N/A	1	$0.35
AND IC Chips	Texas Instruments	74LS08	4	$2.80
2-to-1 MUX Chip	Texas Instruments	74LS157	1	$0.71
Resistors	N/A	N/A	10	$1.00
Dual J-K Flip Flops	Texas Instruments	74LS74	8	$2.40

SPST Switch	N/A	N/A	2	$3.90
Board Casing	N/A	N/A	1	$15.00
Board Pieces	N/A	N/A	30	$10.00

Table 3: Parts Costs

Total Cost:

Total Labor = $18000

Total Parts = $75.73

Grand Total = Total Labor + Total Parts = $18075.73

3.2 Schedule:

Week	Matt LaGreca	Jeremy Diamond
October 2nd	Prepare Design Document for review	Prepare Design Document for review
October 9th	Order parts and design logic pieces	Design board placement of logic pieces and overall board layout
October 16th	Finalize logic piece design and board connection	Design necessary circuit or PCB to supply wall power to components
October 23rd	Write programs for microcontroller	Assemble electrical layout for powering the logic design boards
October 30th	Assemble casing to hold all necessary components	Assemble electrical layout for powering the LEDs and comparison logic
November 6th	Test all microcontroller function	Assemble the LED array for both logic tables and verification panel
November 13th	Assemble all components together	Assemble all components together
November 20th	Prepare for mock demonstration	Test the power supplied at each component of the circuit
November 27th	Prepare first draft of final presentation and test verifications	Prepare first draft of final paper
December 4th	Check first draft of final paper and perform any necessary corrections	Check first draft of final presentation and perform any necessary corrections

December 11th	Finalize overall project for presentation	Complete final paper and prepare for lab checkout

Table 4: Schedule

4.0 Ethics and Safety

This project is designed to be used by introductory ECE students, whether that be at the university level or by PLTW students in high school. Our design must be one that is functional and safe to use for all students. The biggest concern is that all electrical components be contained within the board casing to prevent damage to any of these components. All logic pieces have small components that must be secured to the pieces in a manner that makes them difficult to break. While constructing this board the power supply will never be connected when any of the work is done and the power supply connection will be flush with the board and well secured. With all electrical elements it is critical that the circuit does not get wet and is used in dry conditions, any potential moisture will lead to damage of the circuit.

This project enables young engineers to easily expand their knowledge and understanding, hitting on IEEE Code of Ethics, #5: “to improve the understanding of technology” [1]. Our training tool only requires a basic understanding of logic gate equivalence, we hope that simplify the process will enhance the education process. Parts of this project also contain elements that neither group member have any experience with. To have a fully complete project we will have to consult with our TA, professors, and fellow classmates. We hope to give them credit where it is due and follow IEEE Code of Ethics, #7: “to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others” [1].

References

[1] IEEE.org, “IEEE Code of Ethics”, 2017. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 03- Oct- 2017].