Foothill High School

Protect the Tank Report



Honors Digital Electronics
Period: 5
12/7/2021
Arunan Elamaran and Anthony Mitine

Table of Contents

Table of Contents **1**

Introduction & Problem Statement **2**

Truth Table & Expression **3**

Brainstorming **4**

Circuit Schematics **5**

Building the Circuit **7**

Confusion Strategies 7

Planning the Wiring Process 7

Wiring Phase 8

Final Circuit Design **9**

Results/Conclusion **11**

Cars We Broke Into **11**

Introduction & Problem Statement

The objective of the Car Alarm Project is to design and build a circuit that uses no more than 5 Integrated Circuit (IC) chips and has one combination with four switches. Everything else was allowed. The past unit that we have been working on has taught us expressions for 5 different logic gates (AND, OR, INV, XOR, NAND); how to build circuits on a breadboard using a power supply, jumper cables, switches, LEDs, resistors; and how to create circuit schematics based off of a truth table and/or expression. Also, we learned different theorems to simplify circuit schematic expressions. With this knowledge, we will be able to create and design a difficult car alarm circuit, as our classmates need to analyze the circuit and attempt to "break into" our vehicle with the proper combination.

We want to design a circuit that will have an LED light up only when the four switches are in the correct combination. We will start off by creating a truth table with only one output equaling 1. From there we will derive a simple expression and a simple circuit schematic. Once this has been completed, we will use different theorems to expand the expression and develop the complex circuit schematic, ensuring that the expression stays true. When the circuit is designed, we will re-derive the complex expression to ensure we did it correctly. After we have done all of these steps, we will start to build the physical circuit on the smart breadboard. If any modifications are needed, we will make them accordingly.

Truth Table & Expression

A	B	C	D		F
0	0	0	0		0
0	0	0	1		0
0	0	1	0		0
0	0	1	1		0
0	1	0	0		0
0	1	0	1		0
0	1	1	0		0
0	1	1	1		0
1	0	0	0		0
1	**0**	**0**	**1**		**1**
1	0	1	0		0
1	0	1	1		0
1	1	0	0		0
1	1	0	1		0
1	1	1	0		0
1	1	1	1		0

Key: A = S0, B = S1, C = S2, D = S3, F = Output

Simple Expression: $F = AB^cC^cD$
The circuit should only work when switch A and D are turned on and when switch B and C are turned off.

Complex Expression: $F = (B^cB^c((B^cB^c(A(A+(A+B)(A^c+B^c))))*(BB(A(A+(A+B)(A^c+B^c))))))*((B+(((A^cC+C)(A^cC+C))^c(D+((D+C^c)(D^c+C))^c)))*((B^c+(((A^cC+C)(A^cC+C))^c(D+((D+C^c)(D^c+C))^c)^c))$

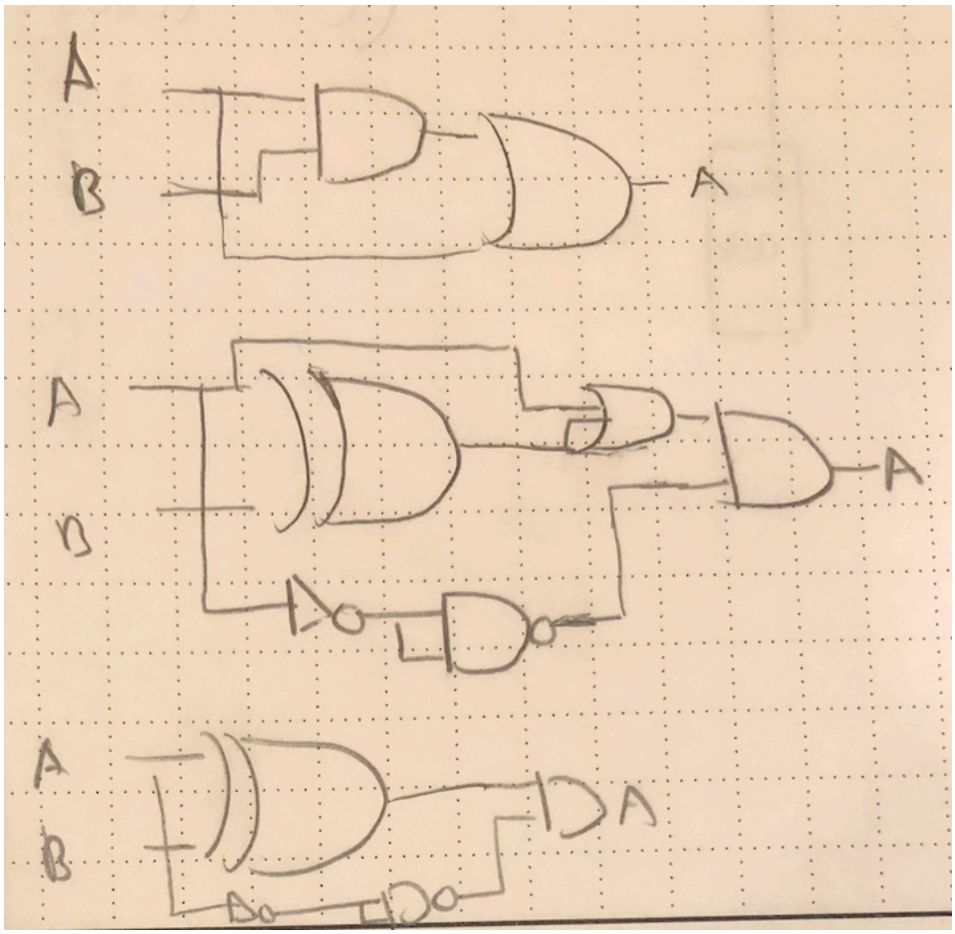
Derivation of Complex Expression discussed in next section (“Brainstorming”)

Brainstorming

After determining our code (1001), we started working on making our expression complex. In order to do this, we first listed the various properties and theories that we could use.

Binary Arithmetic (that differs from the perspective of base 10):

- $1 + 1 = 1$ $X(Y+Z) + XY^c = XY+XZ+XY^c = X(Y+Z+Y^c) = X(Z+1) = X * 1 = X$
- $A(A+B) = A$ $X(Y+Z) = XY + XZ$
- $(AA)^c = A^c$ $(A+B)^c = A^c + B^c$ $(A*B)^c = A^c * B^c$



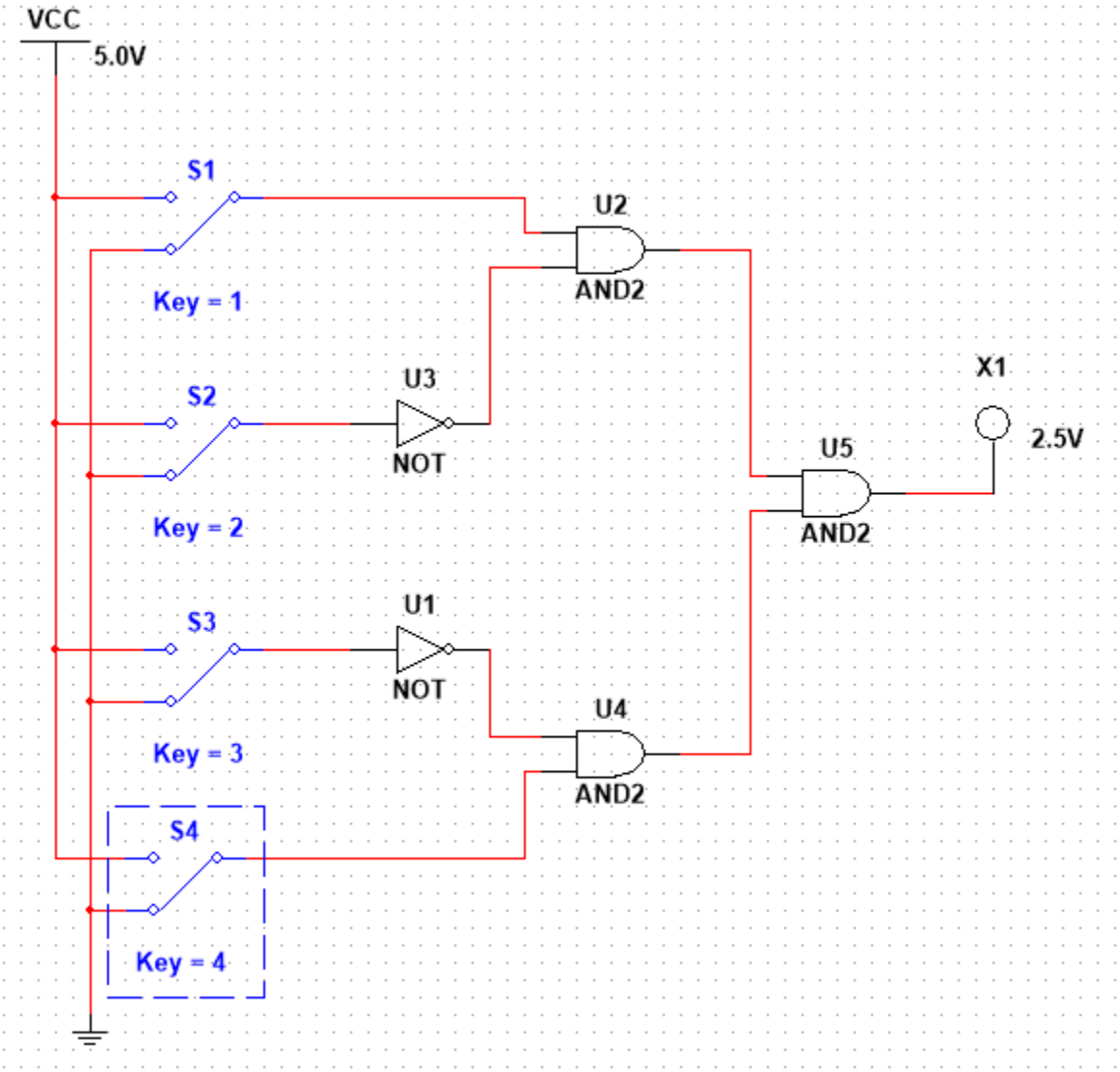
Through doing more experimentation with logic gates, we were able to derive the following logic gate combinations such that if we passed in the inputs A and B, the end output would always be A. Three such combinations are featured on the left.

By implementing the properties and the logic gate combinations featured above in the list and picture, we were able to create our complex circuit schematic shown in the “Circuit Schematics” section (the next section).

Circuit Schematics

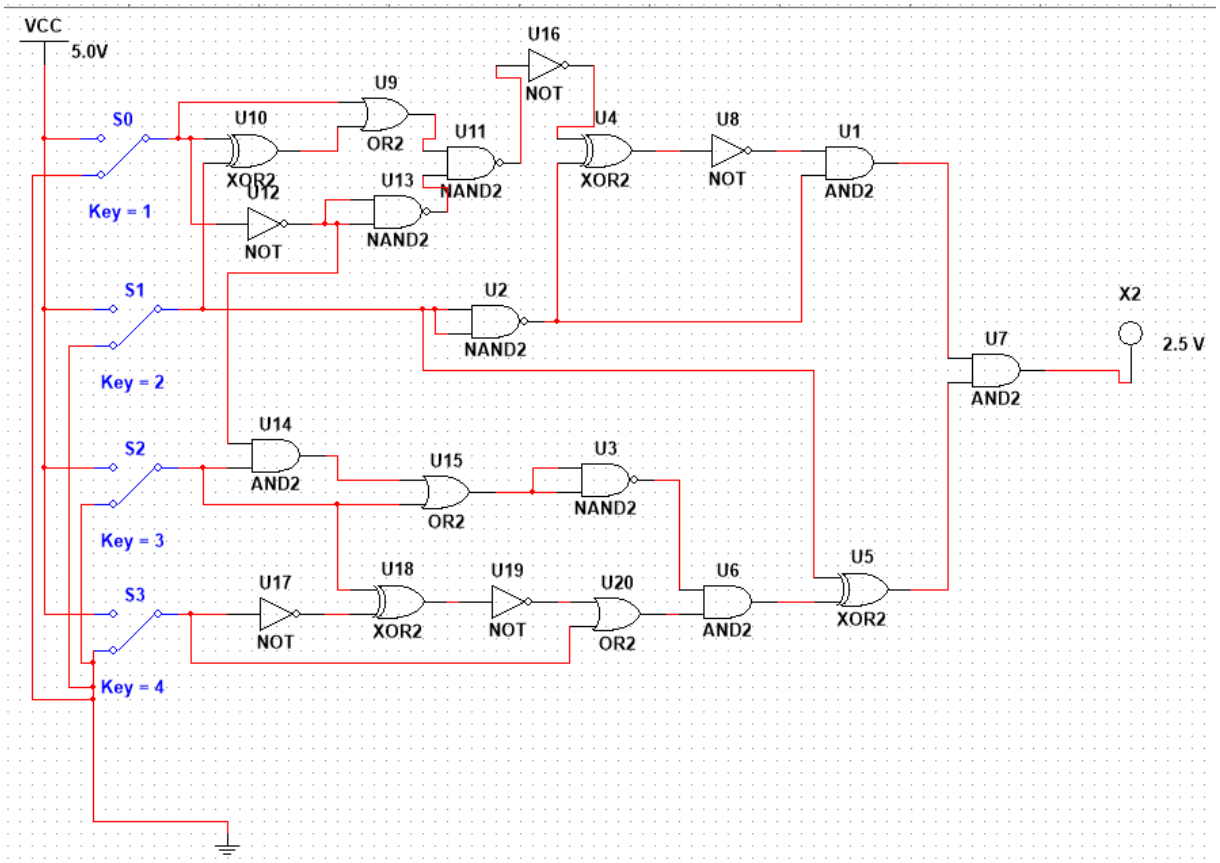
We created the circuit schematics on NI Multisim.

Simple Circuit Schematic:



This circuit is the simplest circuit possible that correctly corresponds to our truth table.

Complex Circuit Schematic:



This is the most complex we were able to make our circuit while still being with the constraints and making sure it correctly corresponded to our truth table.

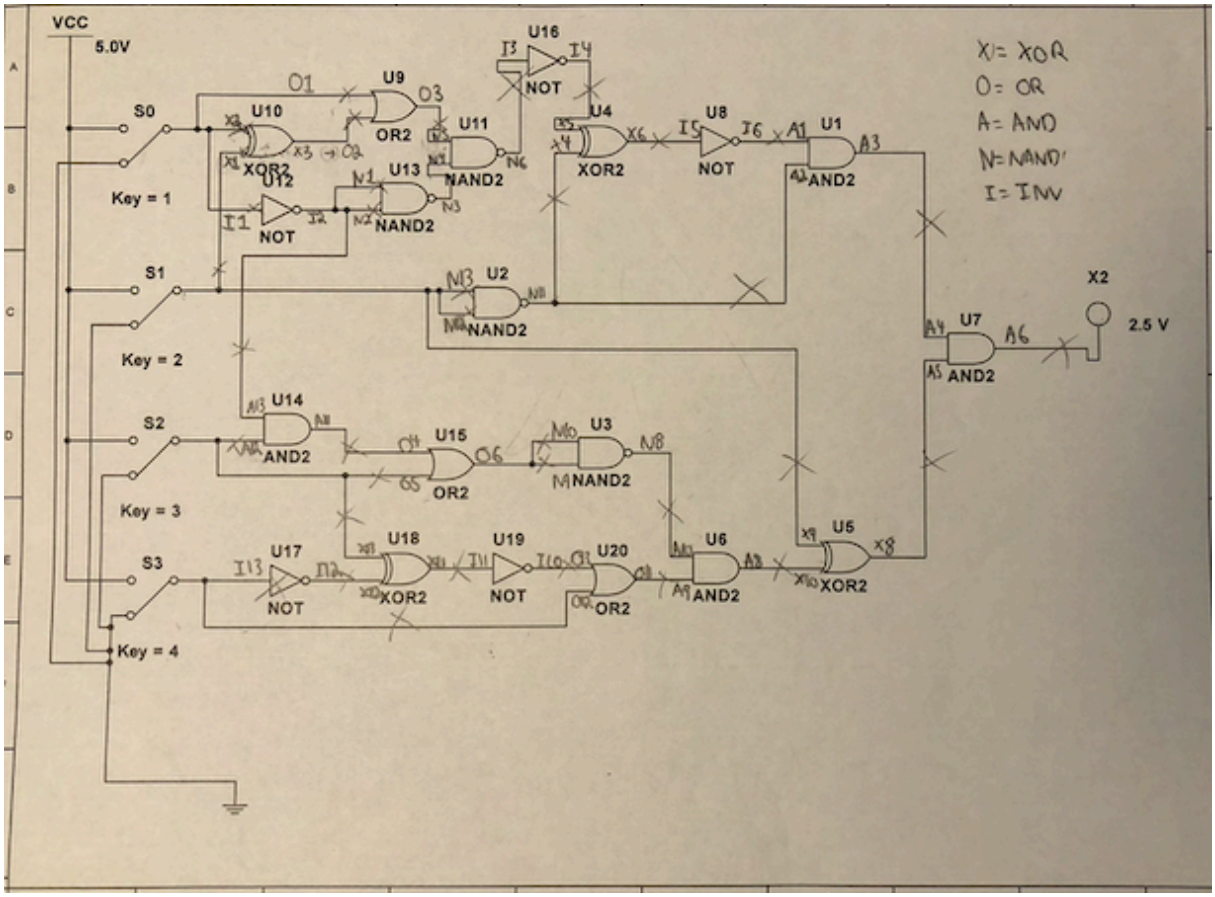
Building the Circuit

Confusion Strategies

The goal of the project being to create a non-hackable car circuit, creating a confusing circuit was essential. One such strategy was to use all the same color wires. This way, the hijacker would not be able to determine where each wire leads. Additionally, by tangling the wires, it would be nearly impossible to trace the wires. Finally, by braiding the initial input wires, the initial step of figuring out where the initial inputs (from the switches) would be nearly impossible.

Planning the Wiring Process

Since all the wires were the same color, it was essential to come up with a diagram to differentiate what wire connected to what terminal strip or IC gate. Additionally, if a wire were to come unplugged, we would not know where it was supposed to go. We were able to solve this issue by giving each wire connection a number of a specific IC gate on our circuit schematic.



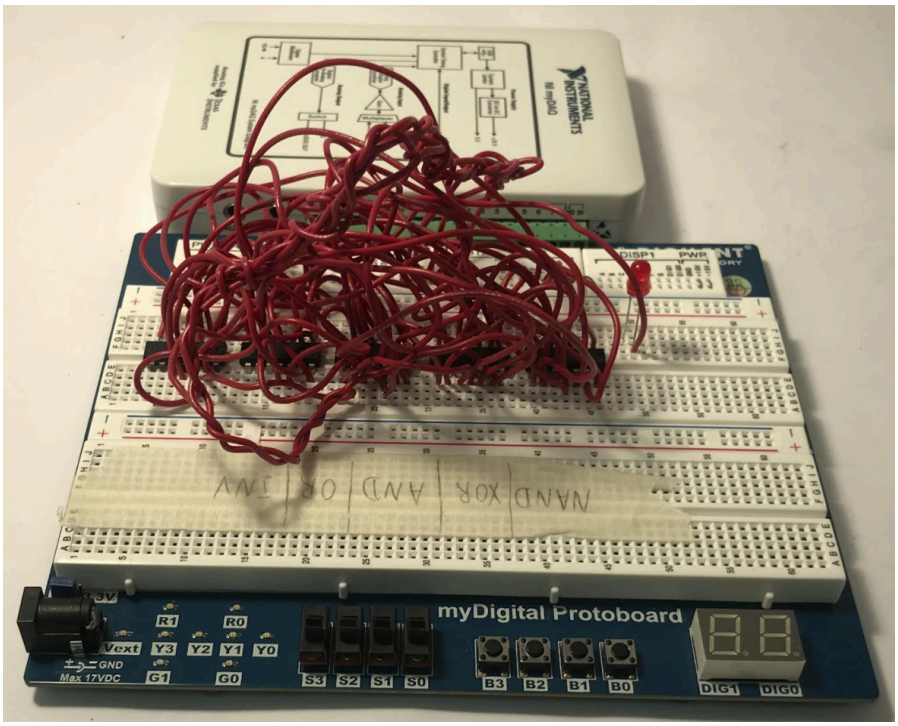
Pictured above is the wire-logic gate labeled diagram we created to help us wire the circuit.

Wiring Phase

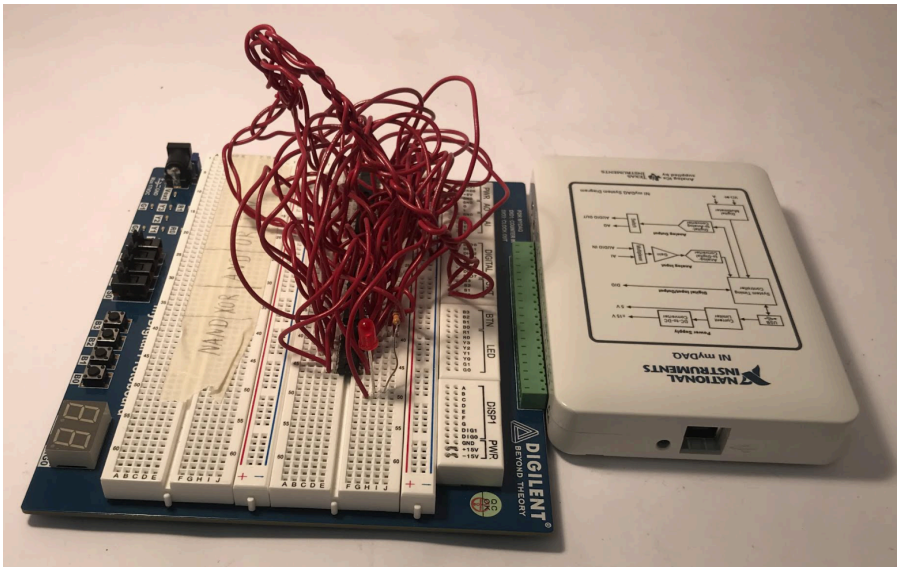
We first placed the Integrated Circuits (ICs) on the board. Using tape, we labeled the order of them on the side. The first wiring steps we took were to connect the power and ground wires for each of the ICs. We then moved on to connecting the logic gates.

We made sure to strictly follow the labels we wrote on the schematic print-out. This way, we were able to minimize the chances of misplacing any of the wires. Additionally, we were able to divide the circuit into two main sections: the top half and the bottom half. We completed most of the top half prior to starting the bottom half. This helped with organization and guided us on what to complete next. Finally, to keep track of which wires we had already placed.

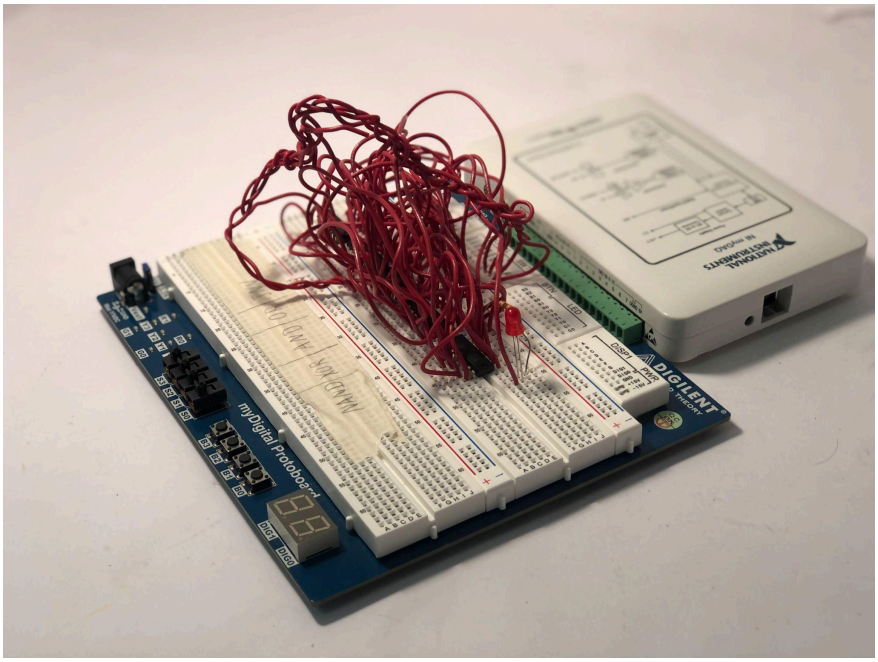
Final Circuit Design



The photo shows the front view of the circuit. All 5 of the IC chips are labeled with tape. The purpose of using only one color for the wiring was to make it very difficult for our classmates to try to figure out the schematic of our car alarm.



This image depicts the side view of our circuit. We also used a red LED to keep the “red theme” of our car alarm.



The image shows the isometric view of our car alarm circuit. To ensure that it would almost be impossible to break into our tank, the four wires that connected the outputs of the four switches to the different IC gates were braided together. All of the other wires were wrapped over and under each other.

Due to its complexity, the final design of the car alarm circuit took roughly 3 full class periods to build. The most challenging parts while building the circuit were simply putting the wires into the right terminal strip of the breadboard and ensuring that all of the wires were connected to the proper gates. Despite our organized planning, we ran into a few dilemmas while putting the wires together, as some would come loose from the terminal strips when connecting other wires. Luckily, we were able to make quick fixes to solve this problem. After the circuit was built, we tested all of the combinations, and the LED only lit up for the intended switch combination.

Results/Conclusion

Our circuit worked perfectly the first few times, and the LED turned on when the correct combination of switches were placed. However, we assume that a few wires came loose and were replaced in the wrong terminal strip, as our circuit now had two more combinations. We were able to find the problem and were able to fix the circuit. Though we faced many difficulties throughout the testing of the circuit, we are glad that it works.

At the end of our project, we learned how to analyze and build simple and complex circuits. This will be a valuable skill to have when we take engineering classes in the future. We both had a lot of fun building our car alarm circuit as well as testing and trying to “break into” other peoples’ cars. If we had more time on this project, we most likely would try to use more theorems to complexify the circuit even more.

Cars We Broke Into

We were able to successfully solve Chris and Arnav’s circuit (winning us a Cybertruck) and Charlie’s circuit (winning us Charlie’s whip). By solving this, we were able to practice our schematic drawing techniques. We learned how to properly draw and solve complex logic circuit problems. The most difficult part when solving each was not losing track of our spot in the circuit. The easiest way for that to happen was when we would trace back through the input nodes of an AND logic gate. However, by persisting, we were able to find patterns and develop techniques that allowed us to solve the circuits.