2025 ASEE Northeast Section Conference, March 22, 2025, University of Bridgeport, Bridgpeort, CT, USA.

Learning Through Logic: An Educational Digital Guessing Game with LED Feedback

Steven Bercik[1], Mehmet Furkan Baylan[1], Ansa Brew-Smith[1], Don Heiman[1], Bala Maheswaran[2], Haridas Kumarakuru[1]
[1]Department of Physics, [2]Department of Electrical and Computer Engineering
Northeastern University, Boston, MA 02115 USA

***Abstract*—This project introduces a digital guessing game, where player-1, the guesser, attempts to deduce a correct combination of 4 sequential bits set by player-2. After player-1 makes their guess of the 4-bits, they receive instantaneous feedback from a row of LEDs telling them how many bits were chosen incorrectly, but not which ones were incorrect. Using this information, player-1 makes a second guess for the 4 bits and receives second feedback from the LEDs. Player-1 gets a total of four trial guesses to find player-2's original 4-bit code. The system contains two user consoles—switchboards for player interaction—and a central computing unit consisting of four interconnected digital circuits that drive the LED outputs. The theoretical foundation is based on Boolean algebra, with four sum-of-products expressions generated and simplified using Karnaugh Maps. These expressions were electronically implemented using logic gates, specifically AND, OR, and XOR. Troubleshooting the initial design of the circuits involved various laboratory instruments, including digital multimeters and a function generator. The final design ensures enjoyable gameplay while providing mental stimulation, allowing both players to practice logical problem solving as they refine their strategies.**

Keywords—digital electronics, logic gates, LED, Boolean algebra

I. Introduction

The purpose of this academic project was to fully realize a functional prototype of a digital gaming device meant for home entertainment. At its simplest, digital electronics deals with signals representing a sequence of discrete values, as opposed to the continuous signals of analog electronics. Digital circuits are constructed with logic gates, physical electronic components that represent Boolean operations. With this, truth tables are a useful tool to visualize a digital gate or circuit's output as a function of its digital inputs [1].

Digital games like this have valuable applications in education. Electronic models serve as important educational tools, allowing students to bridge the gap between their theoretical knowledge and real-world application. Students can gain hands-on experience with digital electronics—understanding the fundamental principles of logic gates, circuit design, Boolean algebra, and how these ideas are applied in real-world devices [2]. Additionally, a project like this can spark interest in STEM fields, encouraging students to explore careers in technology and engineering. This hands-on approach to learning can make complex concepts more accessible, engaging, and fun, fostering an overall deeper understanding and appreciation of electronics.

Students who investigate this game may also experiment with different configurations of logic gates and observe how these changes affect the device's behavior. This process would help reinforce theoretical knowledge through practical application. Due to the strategic nature of the game, students can also develop problem-solving skills as they search for the best algorithm to deduce their opponent's 4-bit code. Through troubleshooting and refining their guesses, they are able to analyze the logical structure of the circuit. Understanding digital electronics is critical for understanding many advanced technologies. Educational tools like this project act as stepping-stones to more complex topics such as microprocessor design or embedded systems.

An important application of this digital design is its scalability and adaptability. Students can use the gate-level diagram as a template for more complex digital systems, adding new functionality or improving performance in a way that they seem fit. In this way, our game also acts as a powerful educational tool.

II. Methodology

A. Rules of Gameplay

Among the first factors that were considered, a concrete set of rules governing the mechanics of the game was necessary to formulate before moving onward. The final rules are as follows:

- One player takes the role of the setter (player-1), the other the guesser (player-2).
- The setter sets a passcode by holding down a combination of buttons on their console.
- The guesser guesses the passcode by holding down a combination of buttons on their console.
- The number of LEDs illuminated displays the number of mismatching (incorrect) buttons chosen by the guesser.
- The guesser has four more guesses to turn off all the LEDs (representing zero mismatching buttons) for a correct passcode.

B. Boolean Analysis and Optimization

With the simple gameplay procedure defined, the theoretical framework behind the circuitry can start to be developed. Since both the input (buttons on user console) and output (number of LEDs activated) of the device could be represented by bits, digital signals were chosen over their analog counterparts. For the scope of this project, the passcode was selected to be composed of four bits, meaning that the output would likewise be composed of four LEDs.

While the total number of LEDs illuminated was to represent the total number of incorrect bits guessed, each *individual* LED was to represent something distinct. For example, the second LED from the left lights up when there are *at least* two incorrect bits, meaning when either the third or fourth LED lights up—signifying at least three or four incorrect bits—so must it too. Four discrete circuits can be designed, one for each LED, to facilitate these conditions.

Truth tables for each output LED were generated, as well as their equivalent canonical sum-of-products expressions (see Appendix). The four input bits were labeled A through D, with A being the most significant bit. Within these truth tables, an input 0 denotes a mismatch between the setter and the guesser, while an input 1 signifies a match. Regarding the output, a 0 represents an inactivated LED, and a 1 indicates one that is activated. Since the canonical sum-of-products expressions were rather long and ill-adapted for immediate conversion into a sequence of logic gates, further simplification was vital.

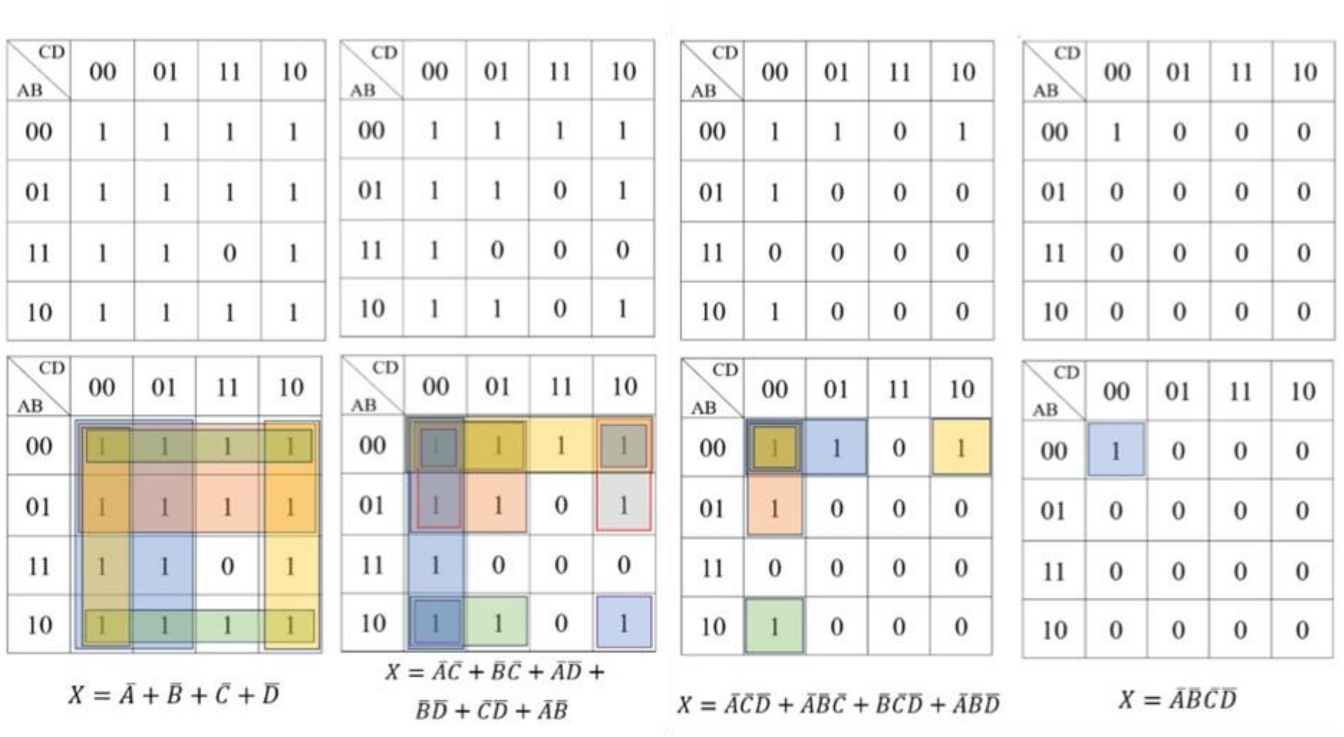
The Karnaugh map is a standard and efficient way of obtaining minimal Boolean expressions from canonical forms. Information stored in truth tables can be faithfully characterized by Karnaugh maps, wherein optimal groupings with size 2^n of similar bits are subsequently identified and translated into reduced Boolean expressions [3]. Fig. 1 shows the representative Karnaugh maps for each of the four LED truth tables, as well as the appropriate optimal groupings. Note that these maps have toroidal geometry—the left and right edges, the top and bottom edges, and crucially the corners, are respectively adjacent. To generate terms for each one of the groupings, the bits held constant are identified. For example, the simplified term for the large, eight-cell peach-colored grouping for the first LED is $\overline{A}$, as it is held constant through the entirety of the group.

Although now more manageable, two sum-of-product expressions still remained unoptimized when implemented with logic gates. Since logic gates function as Boolean operators, the expressions for the second and third LEDs could be simplified by reducing the number of operations. The expressions for the second and third LEDs are given by

$$\overline{A}\overline{C} + \overline{B}\overline{C} + \overline{A}\overline{D} + \overline{B}\overline{D} + \overline{C}\overline{D} + \overline{A}\overline{B} \tag{1}$$

and

$$\overline{A}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C} + \overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{D}. \tag{2}$$



AB \ CD	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	0	1
10	1	1	1	1

AB \ CD	00	01	11	10
00	1	1	1	1
01	1	1	0	1
11	1	0	0	0
10	1	1	0	1

AB \ CD	00	01	11	10
00	1	1	0	1
01	1	0	0	0
11	0	0	0	0
10	1	0	0	0

AB \ CD	00	01	11	10
00	1	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$X = \overline{A} + \overline{B} + \overline{C} + \overline{D}$

$X = \overline{A}\overline{C} + \overline{B}\overline{C} + \overline{A}\overline{D} + \overline{B}\overline{D} + \overline{C}\overline{D} + \overline{A}\overline{B}$

$X = \overline{A}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C} + \overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{D}$

$X = \overline{A}\overline{B}\overline{C}\overline{D}$

Fig. 1. Karnaugh maps and groupings for LEDs 1-4, shown in columns from left-to-right. The respective optimized sum-of-product expressions are shown below the groupings.

These expressions were simplified to

$$\bar{A}(\bar{B} + \bar{C} + \bar{D}) + \bar{B}(\bar{C} + \bar{D}) + \bar{C}\bar{D} \tag{3}$$

and

$$\bar{A}\bar{B}(\bar{C} + \bar{D}) + \bar{C}\bar{D}(\bar{A} + \bar{B}), \tag{4}$$

decreasing the number of logic gates from eleven each to eight and seven, respectively.

Each of the four LED circuits was to be almost entirely realized with AND and OR gates—represented in Boolean algebra as the multiplicative and additive operators respectively [4]—except for the connection with the user consoles. To attach the switchboards to the circuit body, XOR gates were chosen, both since this gate outputs identical signals for two matching inputs (0,0 or 1,1) and because the simplified Boolean expressions were fully composed in terms of $\bar{A}$, $\bar{B}$, $\bar{C}$, and $\bar{D}$ (all LOW signals). Table 1 displays the truth table for XOR gates, where A and B are inputs, and X is the output.

C. Circuit Implementation

With the theoretical foundation, coupled with access to a university electronics laboratory, everything was in place to begin the construction of the prototype.

Digital logic integrated circuits (ICs), or chips containing logic gates, were used for the primary computing body. Five such chips were used, two SN74S08N-AND chips, two SN74S32N-OR chips, and one SN74S86N-XOR chip. A breadboard served as the circuit base, allowing easy assembly, layout adjustments, troubleshooting, and providing five volt and ground channels for power.

As seen in Figs. 2 and 4, the full schematic and the initial prototype, each of the eight inputs (four from the setter, four from the guesser) are routed into XOR gates before entering a complex series of copper wires, gradually building up the simplified Boolean expressions derived above. Finally, they are linked to LEDs, each with its own current-limiting resistor of 470 Ω, and grounded.

Another place in which resistors were employed were the user consoles, or switchboards, as displayed in Figs. 3 and 4. Current-limiting resistors of 470 Ω were placed before purple LEDs, which would shine when their corresponding pushbutton was pressed, acting as visual conformation of the players' passcode/guess. A 1500 Ω resistor was also wired to each pushbutton, before being grounded, to eliminate any residual floating voltage once the button was lifted.

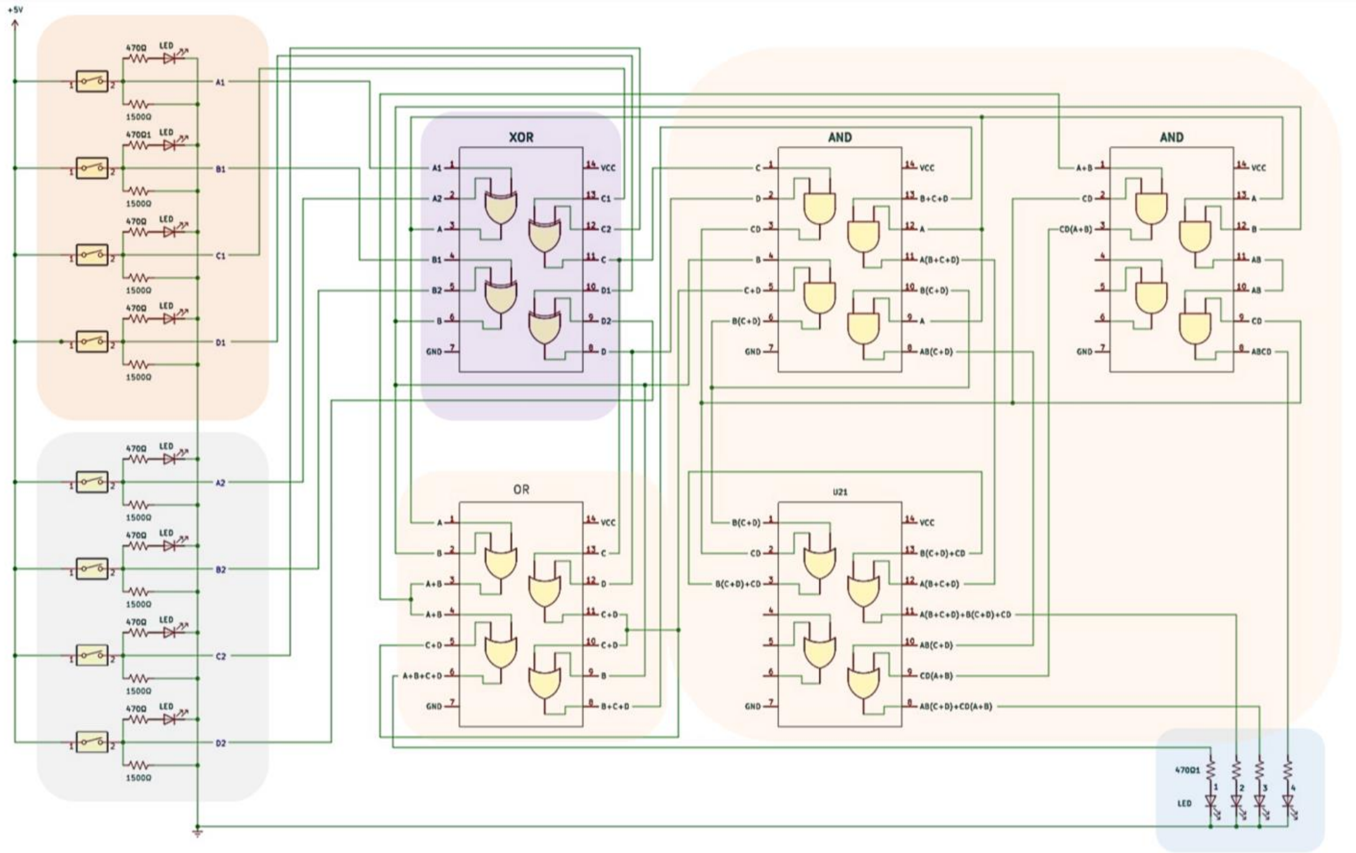


Fig. 2. Circuit schematic of the digital guessing game. The setter's and guesser's consoles (orange and gray) relay signals to the XOR chip (purple) and the main computing unit (peach). The LED output appears at the bottom right (blue).

TABLE I. TRUTH TABLE FOR XOR GATE

A	***B***	***X*** (output)
0	0	0
0	1	1
1	0	1
1	1	0

III. ANALYSIS

Although the preliminary design faced a handful of problems, careful circuit analysis and manipulation proved successful in creating a working game. Amid the challenges faced, the most prevalent was improving the functionality of the switchboard. Initial attempts were plagued with faulty LEDs, floating voltages, and a general lack of understanding surrounding the pushbuttons. Eventually, voltage fluctuations were successfully mitigated by additional ground connections, and the pushbuttons were understood to have three states, not two as originally thought. Specific values of resistance were precisely selected in accordance with maximum current tolerances of circuit components, namely the LEDs and logic gate chips.

Defective components also convoluted the assembly of the device. Tedious measurements were thus conducted using a digital multimeter and function generator, probing the inputs and outputs of each logic gate, as well as connections to ground and the switchboard; parts including damaged LEDs and integrated circuits were removed, effectively restoring functionality [5].

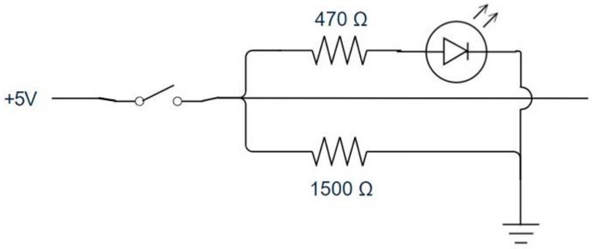


Fig. 3. Circuit diagram of an individual switch.

Early on, gameplay was feasible but cumbersome—the DC power supply needed to be toggled on and off for each new guess posited by the guesser. Nonetheless, in due time this obstacle was overcome by the implementation of a dependable switchboard. Providing five attempts for the guesser resulted in a correct guess approximately 50% of the trials conducted. The game offered an excellent way to test mental dexterity and deduction, particularly due to how it displayed wrong guesses—showing the total number of incorrect bits, instead of specifying which ones are wrong.

In the end, all initial goals for the project were met—the procedural steps outlining gameplay could be faithfully followed in both an enjoyable and mentally stimulating way.

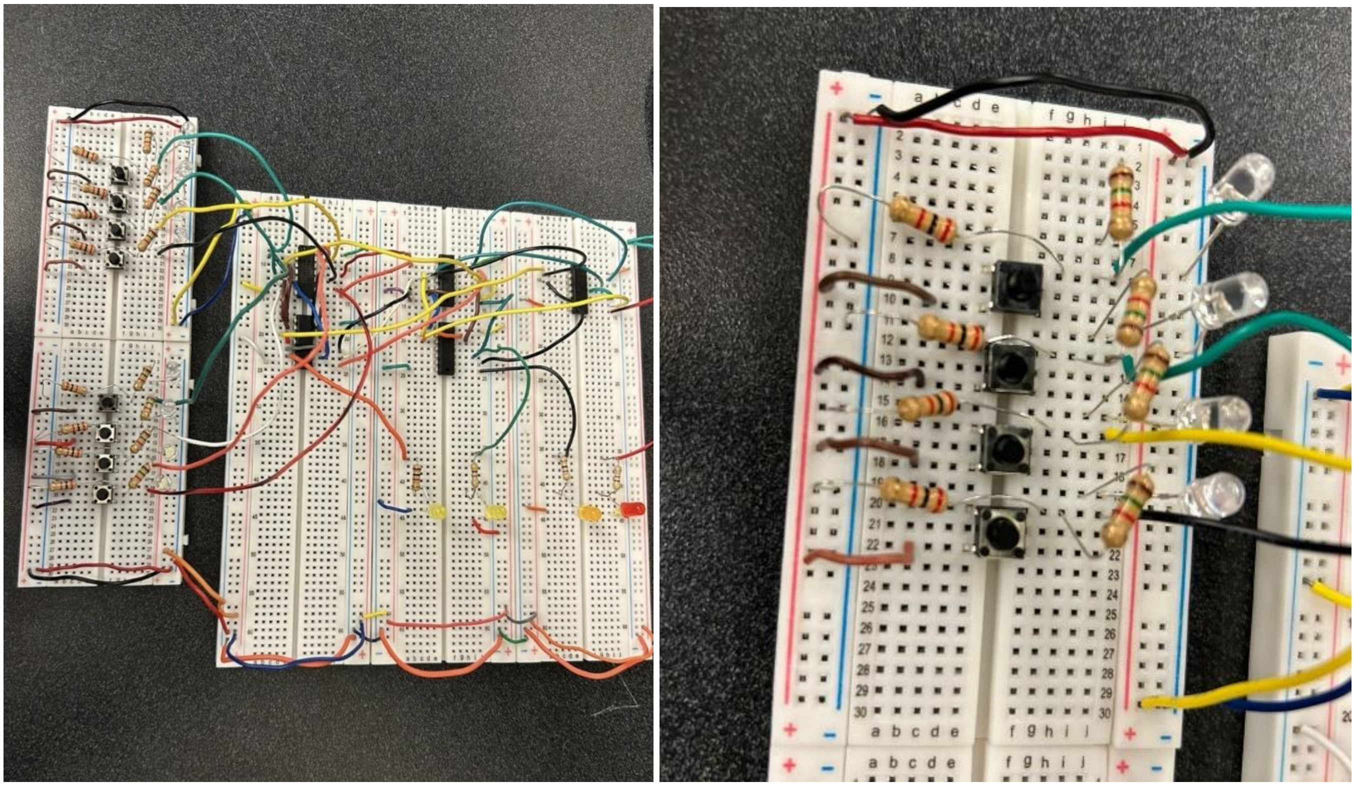


Fig. 4. (Left) Initial test prototype featuring user consoles on separate breadboard, constructed in laboratory. (Right) Close-up of a single-user console showing pushbuttons, pre-LED current-limiting resistors, grounding resistors, and purple LEDs.

IV. Conclusions

The goal of this project was to fully realize an original design for an at-home digital guessing game for two players—the setter, who sets a unique 4-bit passcode using a switchboard, and the guesser, who attempts to deduce the correct combination of bits. Instantaneous feedback regarding the amount of incorrectly guessed bits is provided by a row of LEDs. A strong theoretical framework was supported by Boolean algebra. First, four truth tables encoding all the cases in which each LED would light up were generated, and the appropriate sum-of-products expressions were written. By using the method of Karnaugh maps, these expressions were simplified from their canonical forms. To translate the mathematics into a digital circuit, further manipulation of two expressions was necessary to minimize the amount of logic gates. The four interconnected circuits were built with AND and OR gates, while XOR gates were used to attach the user consoles to the main body. Current-limiting resistors of 470 Ω and 1500 Ω were employed before each LED, as well as to facilitate an essential link to ground after each pushbutton to avoid floating voltages.

Of the challenges faced, ensuring the proper functionality of the switchboards was the greatest. With various laboratory equipment including a digital multimeter and a function generator, as well as meticulous bookkeeping, a careful analysis of each component was conducted. Faulty elements and incorrect connections were located and fixed. The final prototype fulfilled all preliminary goals—players were able to play with ease, and the game proved neither to be too difficult nor too easy. Given five attempts, the guesser was able to correctly determine the passcode nearly half of the time.

V. Future Work

In terms of future improvements, the project's design can easily be augmented or modified to increase its functionality. Besides aesthetic considerations of the prototype, improving the feedback mechanism would allow for a more engaging game. For instance, the addition of a green LED activated when the guesser wins, accompanied by a buzzer or chime, would provide both clear visual and auditory reinforcement of success. Additional modes of gameplay would also enhance the strategic value of the design. Examples include multiple guessers competing to guess the 4-bit code the fastest, or a setup where the setter and guesser work together—alternating turns and trying to converge on a common code.

These modifications would benefit from a more "modular" overall design of the circuitry. Printed circuit boards (PCBs) are much more reliable and durable than their breadboard counterparts. PCB modules could be designed for "plug and play" design—each mode of gameplay would have a corresponding PCB to which the user consoles are connected. In this way, the prototype could be transformed into a more professional-looking, user-friendly product.

Transitioning to a modular design would also facilitate the game's educational value. On the breadboard, the function of individual components is easily obfuscated, as the board's primary use is for prototyping. Employing sockets, between which logic gate ICs can be exchanged, provides a method for students to interact with logical design.

Acknowledgment

The authors extend their sincere gratitude to the electronic laboratory teaching assistants, Harrison Adler, Nathaniel Beaver, and Nicholas Otero, for their invaluable support in the teaching lab. Additionally, the authors wish to acknowledge the staff at the University of Northeastern's Makerspace at EXP building for their essential guidance in the development of a physical device. Finally, the authors express their deep appreciation to the Department of Physics at the University of Northeastern for facilitating their participation in the ASEE-NE 2025 conference.

References

[1] H. B. Enderton, A Mathematical Introduction to Logic, Oxford: Academic, 2013.

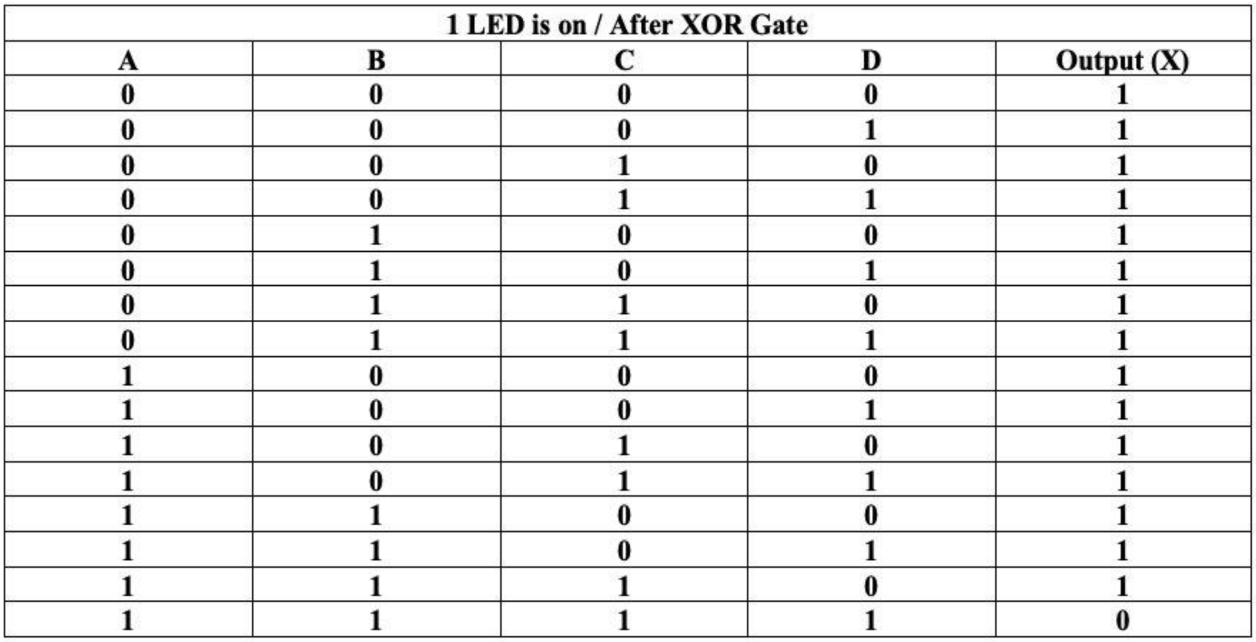
[2] J. Pang, "Active learning in the introduction to digital logic design laboratory course," ASEE Zone III Conference, 2015.

[3] M. Karnaugh, "The map method for synthesis of combinational logic circuits," Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics, vol. 72, no. 5, pp. 593–599, 1953.

[4] L. Null and J. Lobur, The Essentials of Computer Organization and Architecture, Sudbury, MA: Jones and Bartlett Publishers, p. 95, 2012.

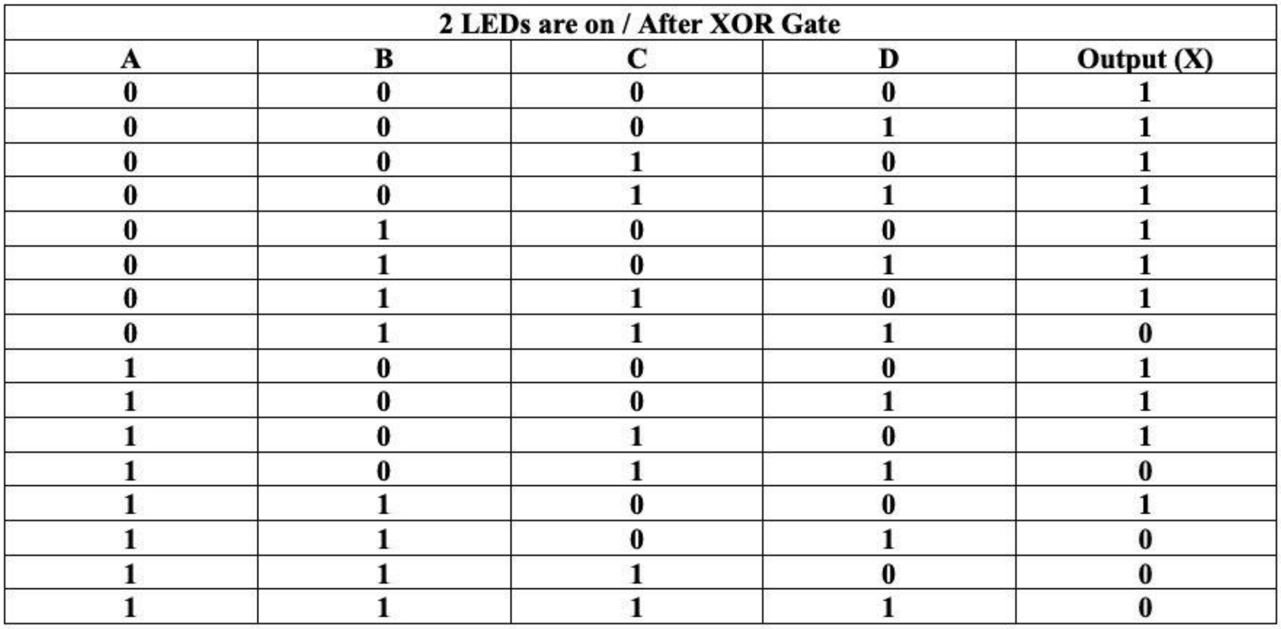
[5] T. L. Floyd, Digital Fundamentals, 11th ed., Harlow: Pearson Education, pp. 170-175, 2015.

Appendix



1 LED is on / After XOR Gate				
A	**B**	**C**	**D**	**Output (X)**
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

$$\mathbf{X} = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD + AB\overline{C}\overline{D} + AB\overline{C}D + ABC\overline{D}$$



2 LEDs are on / After XOR Gate				
A	**B**	**C**	**D**	**Output (X)**
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

$$\mathbf{X} = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + AB\overline{C}\overline{D}$$

3 LEDs are on / After XOR Gate				
A	**B**	**C**	**D**	**Output (X)**
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
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

$$\mathbf{X} = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}B\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D}$$

4 LEDs are on / After XOR Gate				
A	**B**	**C**	**D**	**Output (X)**
0	0	0	0	1
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	0
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

$$\mathbf{X} = \overline{A}\overline{B}\overline{C}\overline{D}$$