GUBJSE: ISSN: 2409-0476

Simplifying the Boolean Equation Based on Simulation System using Karnaugh Mapping Tool in Digital Circuit Design

Md. Jahidul Islam, Md. Gulzar Hussain, Babe Sultana, Mahmuda Rahman, Md. Saidur Rahman and Muhammad Aminur Rahaman

Abstract—In computerized integrated circuits, the fundamental principle intends to avoid the multifaceted nature of the circuitry by making it as brief as attainable and minimize the expenditure. Techniques like Quine-McCluskey (QM) and Karnaugh Map (K-Map) are often used approaches of simplifying Boolean functions. This study presents a recreation framework of simplification of the Boolean capacities by the utilize of the K-Map definition for beginner-level learners. It uses the algebraic expression of the Boolean function to decrease the number of terms, generates a circuit, and does not use any redundant sets. In this way, it gets to be competent to deal with lots of parameters and minimize the computational cost. The result of the assessment is performed in this paper by contrasting it with the C-Minimizer algorithm. In computation time terms, the result appears that our comprehensive K mapping tool outflanks in current procedures, and the relative error accomplishes a lower rate of percentage (2%), which fulfills the satisfactory level.

Index Terms—K-map, Digital Logic Design, Quine McCluskey(QM) technique, C minimizer .

I. Introduction

LOGIC-level simulation is one of the most commonly performed electrical circuit practices, during testing times and design. Amount of logic will be integrated in a simple circuits grows fast, there is a need for greater attempts to automate the designing scheme. One of the key objectives is to develop integrated logic circuit with the Programmable Array Logic (PAL) and Programmable Logic Array (PLA) [1], and here the amount of logic circuits are to be maintained in a minimal form, the production costs of such devices will be decreased. The K map is the graphical procedure to optimize Boolean function throughout the format of a minimal SOP form. Logic Gate-wise, these observations resulted such a two-tier minimal system. It's used mostly for many small design problems. It is obvious that several larger models are carrying out using various computer method implementations, and so we can obtain a pretty good insight in to digital logic gate circuit after learning K map. The process of logic to circuit design in digital logic circuit design, AND, OR, and NOT are treated as the basic boolean operations. These processes can be combined to construct complex expressions that can be directly converted into a hardware implemen- tation, too. The construction cost of a logic gate circuit is inferred by its Boolean Algebraic expression. Small configuration circuit's costs will be smaller than biggest one. For a wide range of logic gate configuration is being performed with a tiny one and the resulting functionality is the same as well, in that case, we will definitely take the small one because of the low cost. It can be difficult to use Boolean algebra to simplify Boolean expressions and can lead to solutions that are not, although they seem minimal. Numerous studies have been conducted to simplify the Boolean expression. First of all the Boolean function is simplified by postulates and theorems in which there are no special regulations [2]. The very first approach to minimize it seems to apply theorems of postulates and many other specific manipulation techniques. Best of our knowledge, literally, for a limited count of variable in the primary learning stage, the K map is very smooth and better, but it is difficult to do it side by side when comes to more than 5 variables. The goal of this research study is to work with the primary level digital logic design learners who are want to work with Boolean Expression and circuit containing a large number of variables while they want to use the mapping approach for minimizing Boolean function. The rest of our research paper is ornamented as follows. The background history of Boolean Logic Design and its simplification

DOI: https://doi.org/10.3329/gubjse.v7i0.54025

This paper was received on 9 January 2021, revised on 15 March 2021 and accepted on 19 April 2021.

Md. Jahidul Islam is with the Computer Science & Engineering, Green University of Bangladesh, Dhaka, Bangladesh. E-mail: jahid@cse.green.edu.bd

Md. Gulzar Hussain is with the Computer Science & Engineering, Green University of Bangladesh, Dhaka, Bangladesh. E-mail: gulzar@cse.green.edu.bd

Babe Sultana is with the Computer Science & Engineering, Green University of Bangladesh, Dhaka, Bangladesh. E-mail: babe@cse.green.edu.bd

Mahmuda Rahman is with the Computer Science & Engineering, Green University of Bangladesh, Dhaka, Bangladesh. E-mail: mahmuda@cse.green.edu.bd

Md. Saidur Rahman is with the Computer Science & Engineering, Green University of Bangladesh, Dhaka, Bangladesh. E-mail: saidur.it@green.edu.bd

Muhammad Aminur Rahaman is with the Computer Science & Engineering, Green University of Bangladesh, Dhaka, Bangladesh. E-mail: aminur@cse.green.edu.bd

techniques with the decade invention of this area are discussed in II. The demonstration and the process of the Kaurnaugh mapping are exhibited in section III. Overall demonstration of whole working procedures are exhibited in section IV. The impact of our research works with performance evaluation are described in section V. And at the end, we have concluded and addressed the future direction in section VI.

II. Related Works

In this section, significant research has been discussed below:

For big amount of variables, the common tabular method called the Quine McCluskey (QM) technique is sufficient. This tabular technique is a step-by- step procedure which is a system-wise instruction introduced by Quine [3] first and McCluskey [4] demonstrated it in later. It operates for huge numbers of variables, as far as we know. So it is perfect for controlling machines. Two main activities work for QM method is as follows: a) Prime Implicates Determination b) Collection of Critical Prime Implicates [5]. Main sickening aspect of QM seems: QM is a NP Complete problem and computation times of the algorithm enhance exponentially by the increasing length of input [6]. Therefore, it is quite difficult for primitive level DLD students to manage the tabular method of QM. The common approach of generalization, called the mapping method. Veitch [7] suggested it first and then Karnaugh modified it. [8], provides a smooth, transparent procedure for reducing Boolean expressions.

The K map is truly an important instrument with few parameters in the processing of algebra logic. In the engineering field, this system is often used for various purposes. Author prasad at el. [14] premises a K-map like online cable virtualization engaging algorithm that consists of two things, first one is scheduling technique which is based on online and second one is K-map-like embedding algorithm. Au- thor jinrong at el. [18] offers an alg at el.orithm which is one of the simplified way to use the idea of a relational database by using the Karnaugh map approach to manage all candidate keys with usable dependency sets. After all and above discussion, it can be said that, working with such a wide range of variables for the Karnaugh map will be a significant achievement throughout this sector. The researcher [19] discusses a prominent primary design of elec- tronics circuit issue that is compounded by the Type 2 problem. This resource gives an overview of their ongoing attempt to reform techniques to deal with basic digital circuit design problems through 'big' Boolean algebraic methods. In this research [20] has an integrated method since it teaches students how to easily and logically simplify Boolean equations step by step, including a description of the phases and instances to explain that. The author [21] suggests a modified method of the Karnaugh map that gives students a opportunity to analyze exactly how a K- map was generated and helps all to know some queries farther explicitly during the education in DLD. But for dealing with a large number of variables, there is no definition. To minimize different circuit outputs, the author [22] suggested an Expanded K-MAP that uses a single Karnaugh map. And essentially, in their algorithm, the multiple Karnaugh-maps accumulated in a single Karnaugh-map. The experimental outcome of their contribution and interpretation gives us an idea that they function only for no more than 5 vari- ables. Authors of [10] proposed a modified algorithm of K-Map algorithm and implemented a K-map solver which works up to 6 variables. It also performs better than C-Minimizer algorithm [7].

III. Overview of Kaurnaugh Map Tool

Kaurnaugh map optimizes the boolean function using the human's ability to identify patterns. This re- search paper presents a structure to implement the K- map for optimizing the boolean function. A boolean function is fed to the system as an input. Then the sys- tem generates 2^n positions in the K-map based on the input variables (n). In K-map each position contains a digit either zero or one to represent the output of the function for the corresponding combination of the input.The system then generates minterm groups with a region of 2^n (where, n = 0,1,2,...) with neighboring 1s. And then by evaluating which variables remain the same inside each box, algebraic minterms are derived.

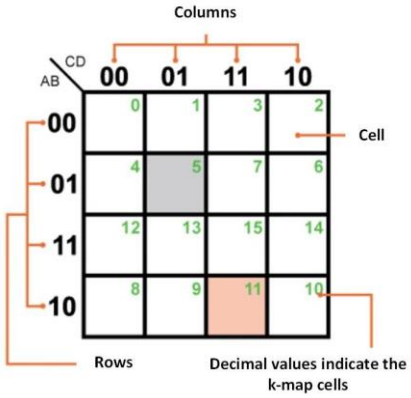

Fig. 1: A Karnaugh Map with 4 × 4 Cells

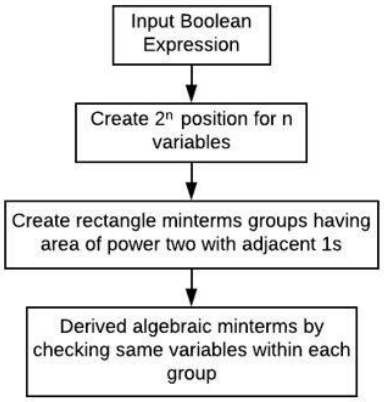

Fig. 2: Block Schematic of the Suggested Implementation Framework for the Karnaugh Map

TABLE I: EXISTING WORKS ANALYSIS

Authors	Contributions
Elham et al. [9]	Design a simulation system that finds optimal expression and reduces the cost of the circuit.
Rahman et al. [10]	Develop a Karnaugh mapping that can deal with a big number of variables with minimum cost and also a comparison with C-Minimizer algorithm.
Rushdi et al. [11]	Present a prominent basic digital circuit design issue using Variable-Entered Karnaugh Maps (VEKMs) with don't care notation.
Vyas et al. [2]	Propose a procedure widely linked to asymmetric base logic function systems using Karnaugh maps and also the technique thus makes it possible to minimize spintronic and memristive logic circuits.
Wang et al. [12]	Design a system that enhances the concept of Karnaugh maps also demonstrate the results and applications.
Abdalla et al. [13]	Develop a method that improves the concept of Karnaugh maps and better than the Quine-McCluskey method for low and large variables.
Prasad et al. [14]	Propose a generalized Karnaugh Map Method that uses "K–don't care" and it can easily solve seven variables k-map for all prime implicants.
Kim et al. [15]	Design a karnaugh map using web-based java applets based on Quine-McCluskey minimization technique which aim is to maximize the learning efficiency.
Murugavelneural et al. [16]	Design a Boolean circuit and also proposed an algorithm with many variables using modified karnaugh map
Nabulsi et al. [17]	Present a comprehensive procedure that simplified the logical function using a truth table Based on the Minterms combination.
Proposed	Design & development a simulation system that finds optimal expression, simplified the logical function using a truth table, reduces the cost of the circuits and also demonstrate the results.

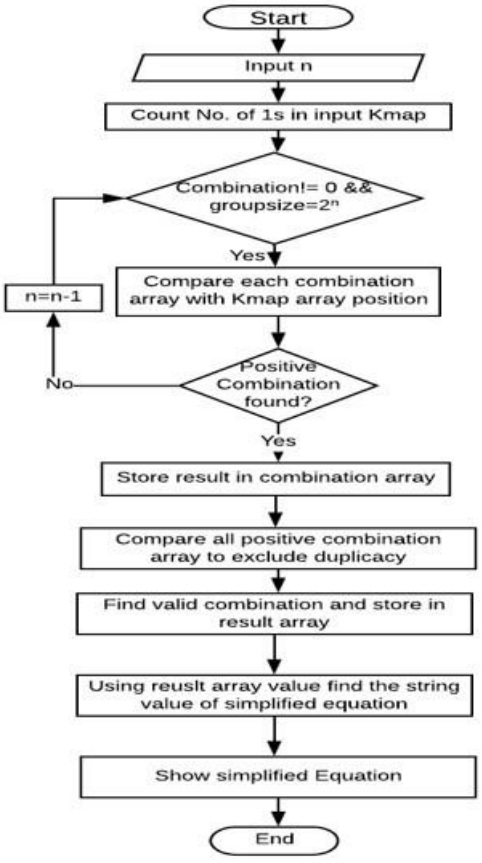

Fig. 3: System Flow Diagram for the K-Map

IV. METHODOLOGY

A. Creation of 2^n Positions and Min-term Groups

The framework creates 2^n placements for n parameters to build the K-map where gray code is used to order the locations as if the change occurs only in a single variable in each pair of adjoining cells. Then the output of the function for the corresponding input combinations is preserved using zero and one digit in every location of the Karnaugh-map. They suggested approach uses two arrays of one dimension where this one is used for holding the location of the K map & the other one is used for storing the result of the function for that location in the same array index. For instance, Fig. 4 displays the iterator repre- sentation of a 4-variable Boolean expression between both the location set as well as the method output list. The framework produces all the potential classes of minterms for both the periodic system dependent on the number of parameters and the variations are described by an Array List of two dimension. The List is then forwarded to the GroupChecker method to review the unique Boolean expression for a major combination of minterms containing adjacent 1s.

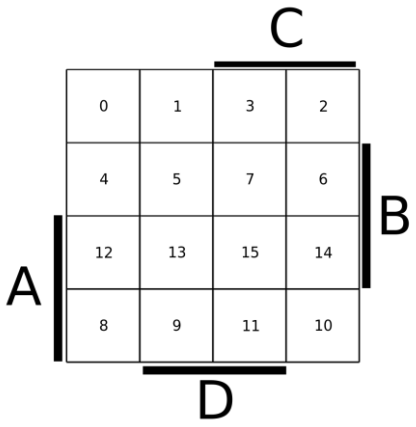

Fig. 4: Positional Values of K-Map (4 Variables)

B. Checking of Groups

Minterms are grouped possessing a region of 2^n (where, n = 0,1,2,...) in K-map. The method **Group Checker** has tested all possible combinations which can be generated by the neighboring 1's. The conflicting groups were often thought to establish bigger

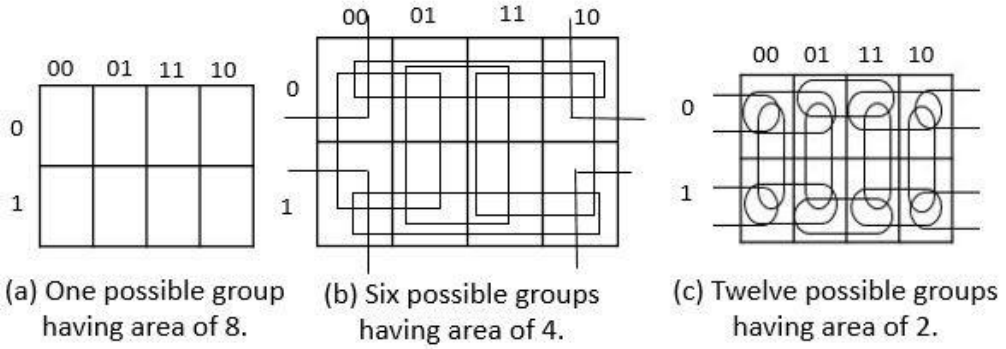

Fig. 5: An Overview of all Possible Permutations of the Boolean Method Min-term Consisted of Three Parameters.

Algorithm 1: K_Map

Input: k-Map_position_Array[], function_output[], variable_Size
Output: vCombinationArray[],rGroup[]

```
Initial variable_Size ← values from the users;;
Initial k-Map_position_Array ← values from the users;
Initial function_output ← Method input values as stated by the location;
Initial comb[][] ← Potential amalgamation values of the array;
Initial m ← 0;
Initial s ← 0;
Initial end ← combination[][].len;
for k= 0 to end do
    init vCombination ← call_group-Checker(k-Map[], comb[k], k);
    if vCombination = zero then
        increment m by one;
    end
end
Init rGroup[]← call pCombination[group-Size];
```

numbers of groups without any zeroes. Fig. 5 demonstrates that, there exist two potential minterms classes of getting area four for the scenario given in Fig. 4. Therefore the method **GroupChecker** returns amount of teams for that key method.

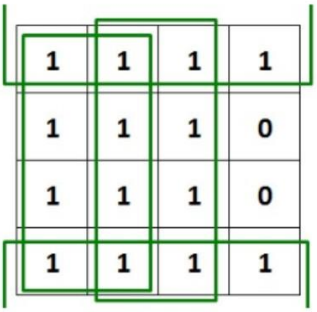

Fig. 6: Min-terms Groups for Boolean Methods

Algorithm 2: Algorithm of Group Checker

Input: k-Map[], combination-Array[], combination-Number, mCombinationCount, grpMax1
Output: validGrpCombinations

```
Initial pCombination[] ← mCombinationCount;
Initial s ← 0;
Initial end ← combination-Array[].length;
Initial f ← 0;
for i= 0 to end do
    if k-Map[combination-Array[i]] = 1 then
        increment f by 1;
end
if f = grpMax1 then
    pCombination[combination-Number]← 1;
    increment s by 1;
    return one;
else
    return zero;
end
```

C. Algebraic Min-terms Computation

The given algorithm 3, measures the numbers of ones in each grouping while groups are generated. It specified how many parameters are needed to describe this min-term as according to the number of ones. Whenever the number of parameters is specified, the frequent bit among all the places was regarded as well as the parameter for that location was obtained.

Algorithm 3: Counter of Numbers of one's

Input: k-Map[]
Output: The Numbers of 1's

```
Init count ← 0;
Init len ← k-Map[].len;
for k=0 to len do
    if k-Map[combination-Array[k]] is one then
        increment count by one;
    end
end
return count;
```

D. Circuit Drawing from Expression

The given algorithm 4, generates the required circuit form the simplified equation. In this algorithm operators and operands are considered to generate correspondingly circuits.

Algorithm 4: Algorithm for Circuit Genera- tion

```
Input: Expression[]
Output: Circuit
Init len ← Expression[].len;
for k=0 to len do
    if Expression[k] is '+' then
        Generate a OR circuit;
        Assign Expression[k-1] as left operand of OR;
        Assign Expression[k+1] as right operand of OR;
    end
    if Expression[k] is '.' or ' ' then
        Generate a AND circuit;
        Assign Expression[k-1] as left operand of AND;
        Assign Expression [k+1] as right operand of AND;
    end
    if expression[k] is '!' then
        Generate a NOT circuit;
        Assign Expression [k-1] as operand of NOT;
    end
end
return Circuit;
```

V. Performance Evaluation

A. Setup for Evaluating the System

With the following environment as shown in Table II, we have tested the established Karnaugh mapping visualization tool:

TABLE II: ENVIRONMENT SETUP

Operating System	Window's 10 64-bit
Processors	Intel's Core i5-3.5 GHz CPU
Memory	16 GB
Solid-State Drive	512 MB
IDE	MS Visual's Studio Express 2010
Language	C#

B. Design of the Simulation system

For testing this application the cell numbers (as an input) have been entered. Then, it have found the following Sum of Product (SOP) expression 1 as the input of four variables and also have found optimal expression 2 as output which is shown in Fig. 7.

$$X = \overline{A}\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + AB\overline{C}\overline{D} + AB\overline{C}D + ABC\overline{D} + ABCD \quad (1)$$

$$Y = \overline{A}\overline{C}D + AB\overline{C} + A\overline{C}D + ABD + \overline{A}CD \quad (2)$$

This system uses the input expression 1 and generates truth table and simplified equation shown in Fig. 7. The best part is our it also generates the circuit (with minimum cost) diagram as Fig. 8. Moreover, 5 variables K-Map truth table, optimal expression and circuit diagram also shown in Fig. 9 and 10 respectively.

C. Performance Discussion

1) Influences of increase in the number of Parameters with relative error: A team of 100 Digital Logic Design (DLD) students and undertaking a Bachelor's Degree in Computer Science and Engineering (CSE) inspected it after the implementation of the system. The students are given three(3) groups of Boolean algebra expressions with parameter number three(3), four(4), five(5), six(6), and seven(7) to evaluate soft- ware where various numbers of equations are included in each set. This is achieved because it is more difficult to optimize expression with a greater set of variables than just a lesser set of variables. For Boolean algebra equations with parameter numbers three (3), four(4), five(5), six(6), and seven(7), the Table III displays relative and absolute errors.

TABLE III: RELATIVE ERROR FOR EXPRESSIONS

No. of Variables	No. of expressions	Accurately identified Expressions	Perfect Error	Relative Error
4	200	196	4	0.022
	400	390	10	0.026
	600	584	16	0.027
5	200	192	8	0.042
	400	380	20	0.054
	600	564	36	0.065
6	100	93	7	0.075
	200	187	13	0.070
	300	279	21	0.075
7	50	44	6	0.136
	100	83	17	0.205
	150	126	24	0.190

TABLE IV: RELATIVE ERROR FOR CIRCUIT

No. of Variables	No. of Circuits	Correctly Identified	Perfect Error	Relative Error
4	50	49	1	0.02
	80	77	3	0.038
	100	95	5	0.05
5	50	47	3	0.06
	70	64	6	0.086
	90	80	10	0.11
6	10	9	1	0.1
	20	17	3	0.15
	30	25	5	0.17

From Table III, we do see that relative error increases like those of the numbers of expression as well as the numbers of variables increases. We identified a minimal relative error of 0.02 with 4 variables as well as a maximum of 7 expressions of the variables. Fig. 12 displays the relationship among the numbers of parameters, the numbers of expression, and also the respective error found in Table III. Moreover, Table IV shows the relative error for circuit generation with respect to the no. of variables. We have identified a

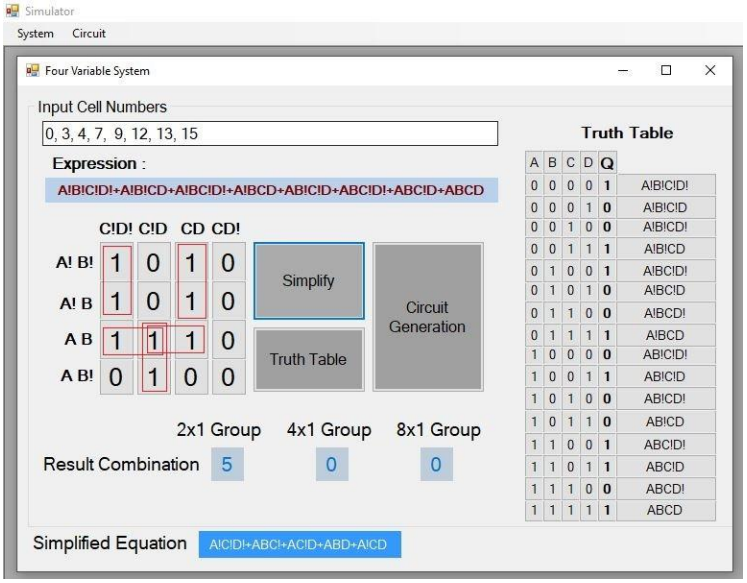

Fig. 7: Simplified Optimal Expression with Truth Table Created by the Expression System 1

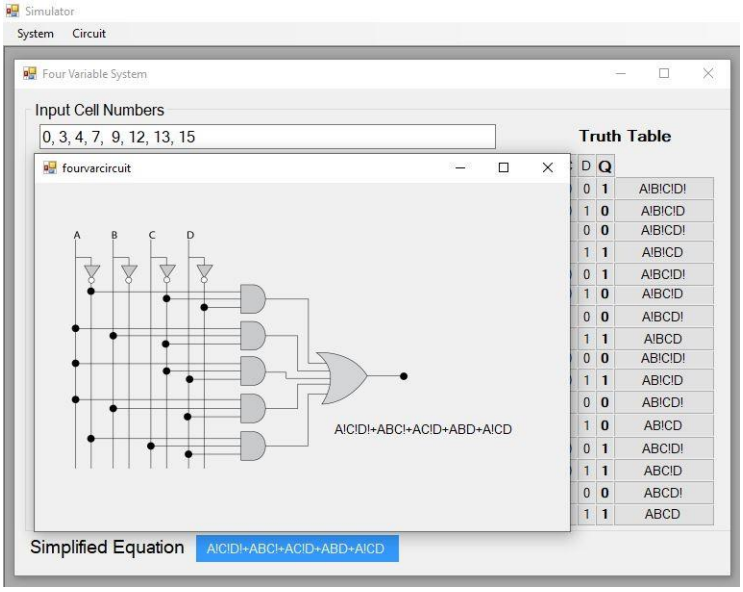

Fig. 8: The Circuit diagram produced by the expression (4 variables)

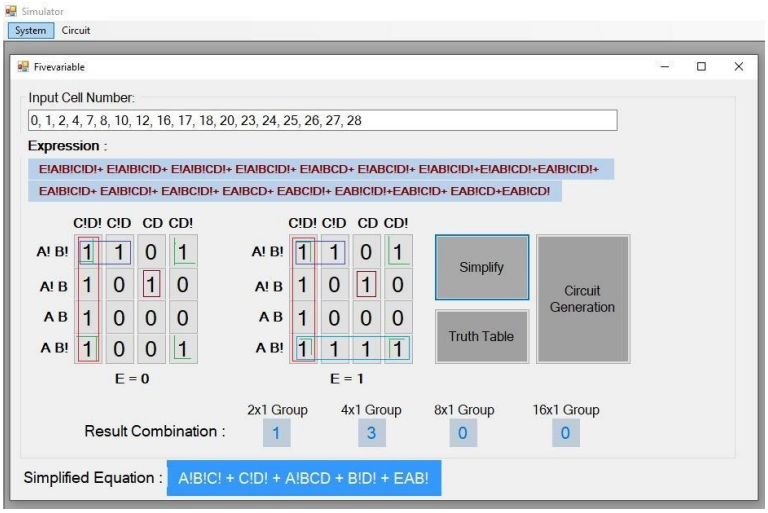

Fig. 9: Simplified Equation for 5 Variables K-Map

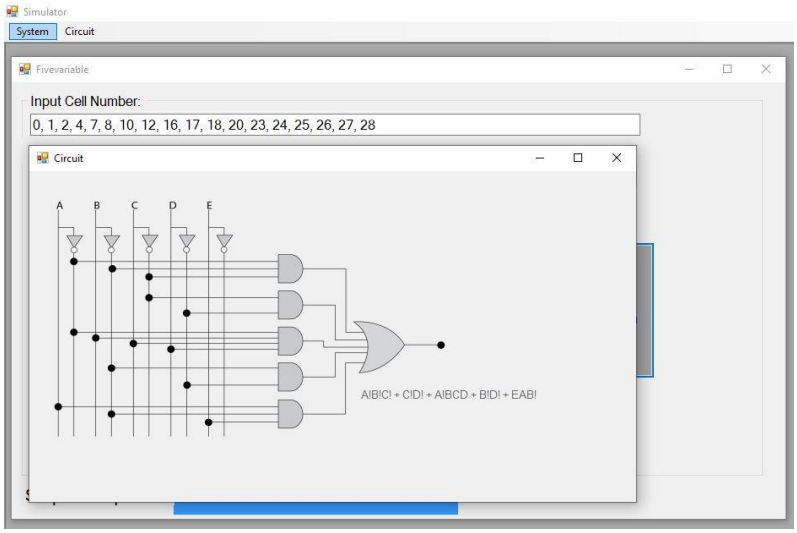

Fig. 10: The Circuit Diagram for the 5 Variables

TABLE V: COMPARATIVE ANALYSIS

References	Optimal Expression	Circuit Generation	No. of Variables used	Truth Table Generation
Elham et al. [9]	YES	NO	4 Variables	NO
Rahman et al. [10]	YES	NO	7 Variables	NO
Wang et al. [12]	YES	YES	4 Variables	NO
Abdalla et al. [13]	YES	NO	6 Variables	NO
Prasad et al. [14]	YES	NO	7 Variables	NO
Kim et al. [15]	YES	YES	3 Variables	YES
Nabul et al. [17]	YES	NO	4 Variables	YES
Our System	YES	YES	7 Variables	YES

minimal relative error of 0.011 with 4 variables as well as a maximum of 6 variables.

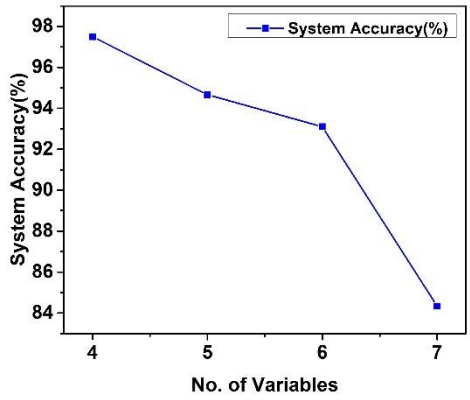

Fig. 11: System Accuracy (%) for Expressions of the System with Respect to the No. of Variables

2) System Accuracy: Fig. 11 shows the system accuracy for various expressions with respect to the no. of variables. With a small number of variable less than 4, the system accuracy almost 100%. Then, increasing the no variables, the accuracy is decreasing with a linear trend. Fig. 12 shows that when number of variables is increasing and number of expressions is decreasing then relative error is high and it is low when the number of variables is low but number of expression is high.

3) Computation Time Comparisons: We have evaluated the computational time analysis to evaluate the

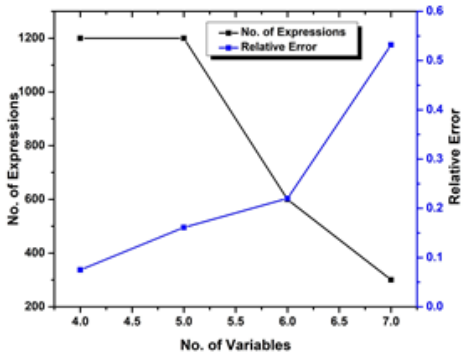

Fig. 12: Relative Error with Respect to the Numbers of Expressions Vs Variables

performance discovered from the application system. Fig. 13 shows the computational time increases with the number of parameters. Notably, with a small number of variables (i.e. $\approx$ 4), the computational time of our system and C-Minimizer assumes similar values. Then, increasing the no. of variables, the computational time also increasing We can assume from the graph that we need just 312.43 milliseconds for 7 variables, which is very efficient. This result proves the practicality that can effectively reduce the computational time with respect to the existing baseline.

To measure the effectiveness of this method as there

are no latest projects on system creation to reduce the Boolean expressions, a comparative analysis is made with the result found on paper [7]. The relation is seen in the tables III and IV and we're seeing our Karnaugh mapping system requires less effort for 3 to 7 parameters than the C-Minimizer Method. The explanation for this reduced time is, through the use of gray code series, our K-mapping system eliminates repeating indices for min-terms. Finally, Table V shows the comparative analysis of our system with various approaches.

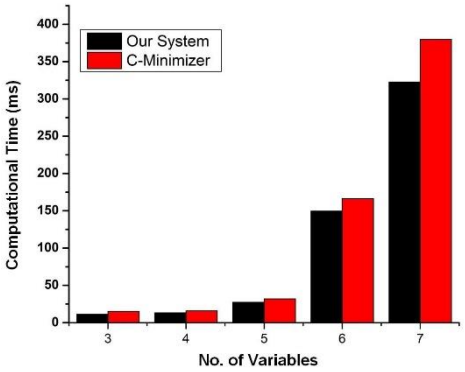

Fig. 13: Computational Time(ms) with Respect to the Numbers of Variables of Our System with C-Minimizer Algorithm [7].

VI. Conclusion & Future Work

It is provided an endeavor to demonstrate a comprehensive Karnaugh mapping method in this re- search paper, which provides an easy way to simplify Boolean Expressions and circuit where there is a large number of variables. In this paper, it is shown that the results expose a strong correlation between the conceptual solution and the mathematical model's predicted results. As it was said, the goal was to work for students of digital logic design at the primary level, so any relative situational error may cause them anxiety. Still, this level of anxiety is much lower.

Acknowledgement

This work was supported in part by the Center for Research, Innovation, and Transformation (CRIT) of Green University of Bangladesh (GUB) under grant No. 7/2019 (CRIT, GUB).

References

[1] A. M. Rushdi, "Improved variable-entered karnaugh map procedures," Computers & electrical engineering, vol. 13, no. 1, pp. 41–52, 1987.

[2] V. Vyas, L. Jiang-Wei, P. Zhou, X. Hu, and J. S. Friedman, "Karnaugh map method for memristive and spintronic asymmetric basis logic functions," IEEE Transactions on Computers, 2020.

[3] W. V. Quine, "The problem of simplifying truth functions," The American Mathematical Monthly, vol. 59, no. 8, pp. 521–531, 1952. [Online]. Available: http://www.jstor.org/stable/2308219

[4] E. J. McCluskey Jr, "Minimization of boolean functions," Bell system technical Journal, vol. 35, no. 6, pp. 1417–1444, 1956.

[5] T. Rathore, "A note on the size of a karnaugh map," IETE Journal of Education, vol. 58, no. 1, pp. 3–6, 2017.

[6] A. B. Marcovitz and C. M. Shub, "An improved algorithm for the simplification of switching functions using unique identifiers on a karnaugh map," IEEE Transactions on Computers, vol. 100, no. 4, pp. 376–378, 1969.

[7] D. Nasa and U. Ghose, "C–minimizer algorithm: A new technique for data minimization," International Journal of Computer Applications, vol. 44, no. 17, pp. 15–19, 2012.

[8] V. Saxena, M. Srivastava, and D. Arora, "Performance estimation of karnaugh map through uml," International Journal of Computer Science and Network Security, vol. 9, pp. 220–225, 2009.

[9] E. H. Aziz, "Design simulation system to simplifying boolean equation by using karnaugh map," AL-Rafidain Journal of Computer Sciences and Mathematics, vol. 14, no. 1, pp. 97–115, 2020.

[10] M. S. Rahman, R. Hasib, B. Sultana, M. G. Hussain, M. Rahman, and M. A. Rahaman, "An extensive karnaugh mapping tool for boolean expression simplification," in 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI). IEEE, 2019, pp. 1–5.

[11] A. M. A. Rushdi, "Handling generalized type-2 problems of digital circuit design via the variableentered karnaugh map," International Journal of Mathematical, Engineering and Management Sciences (IJMEMS), vol. 3, no. 4, pp. 392–403, 2018.

[12] C. Wang and Y. Tao, "Karnaugh maps of logical systems and applications in digital circuit design," Circuits, Systems, and Signal Processing, vol. 39, no. 5, pp. 2245–2271, 2020.

[13] Y. S. Abdalla, "Introducing the yasser-map as an improvement of the karnaugh-map for solving logical problems," in 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE, 2015, pp. 1–5.

[14] V. Prasad, "Generalized karnaugh map method for boolean functions of many variables," IETE Journal of Education, vol. 58, no. 1, pp. 11–19, 2017.

[15] D.-S. Kim and H.-K. Jeong, "Design and analysis of educational java applets for learning simplification procedure using karnaugh map," Journal of Internet Computing and Services, vol. 16, no. 3, pp. 33–41, 2015.

[16] P. Murugavel, "Neural modified karnaugh map for boolean circuit," International Journal of Trend in Research and Development (IJTRD), 2015.

[17] M. A. Nabulsi, A. A. Alkatib, and F. M. Quiam, "A new method for boolean function simplification," International Journal of Control and Automation, vol. 10, no. 12, pp. 139–146, 2017.

[18] T. C. D. Jinrong, "A methord of using the next state karnaugh map to analyse the sequential logic circuit [j]," JOURNAL OF SICHUAN TEACHERS COLLEGE (NATURAL SCIENCE), vol. 3, 1999.

[19] W.-d. BAO and G.-y. JI, "The calculator assistance the karnaugh map diagram turns over the logic function [j]," Computer Knowledge and Technology (Academic Exchange), vol. 5, 2007.

[20] L. Rong-de, "Applications of combined karnaugh map in analysis of logic circuits [j]," Journal of Yangtze University (Natural Science Edition) Sci & Eng V, vol. 4, 2009.

[21] P. Cobham, "An extension of karnaugh map technique to interface design," Radio and Electronic Engineer, vol. 38, no. 3, pp. 131–136, 1969.

[22] C. Yang and H. Jiao, "Low power karnaugh map approximate adder for error compensation in loop accumulations," in 2019 International Conference on IC Design and Technology (ICICDT). IEEE, 2019, pp. 1–4.

Md. Jahidul Islam received the B.Sc. and M.Sc. degrees in Computer Science and Engineering from Jagannath University (Jnu), Dhaka, in 2015 and 2017 respectively. Currently, he is working as a Lecturer and Program Coordinator (Day) at Computer Science and Engineering (CSE), Green University of Bangladesh (GUB), Dhaka, Bangladesh since May 2017 to present. He is a member of Computing and Communication and Human-Computer Interaction (HCI) research groups, CSE, GUB. His research interests include Internet of Things (IoT), Blockchain, Network Function Virtualization (NFV), Software Defined Networking (SDN), Digital Forensic Investigation (DFI), HCI, and Wireless Mesh Networking (WMN).

Md. Gulzar Hussain was born in Nimnagar, Dinajpur City, Bangladesh. He received the B.Sc. degree in computer science and engineering from the Green University of Bangladesh, Dhaka, in 2018. Since 2019, he has been a Lecturer with the Computer Science and Engineering Department, Green University of Bangladesh, Dhaka, Bangladesh. He is the author of one conference article. His research interests include machine learning, natural language processing, text mining, topic modeling etc. Mr. Hussain was a recipient of seven times Vice-Chancellor awards and once Dean award for excellent performance in trimester result. He is a Graduate Student Member of IEEE since 2019.

Babe Sultana was born in, Cox's Bazar, Bangladesh, in 1994. She received her B.Sc. Degree in Computer Science and Engineering from Green University of Bangladesh (GUB) in 2018. At present she is working as a Lecturer, Dept. of CSE, Green University of Bangladesh. Also, her publication was about "Multimode Project Scheduling with Limited Resource and Budget Constraints" published in International Conference on Innovation in Engineering and Technology (ICIET) 27-28 December, 2018 and she got Best Paper Award and IEEE Best Paper Award on this conference. Her research interests include Theory of Optimization, Natural language Processing, Renewable and Sustainable Energy.

Mahmuda Rahman received her B. Sc. Engineering degree in Computer Science and Engineering (CSE) from Green University of Bangladesh, in 2018. She is Currently working as a Lecturer of CSE Department in Green University of Bangladesh. Her research interests include Digital Image processing, Computer Vision and Natural Language Processing.

Saidur Rahman received his B.Sc. Engineering degree in Computer Science and Engineering (CSE) from Green University of Bangladesh. He is Currently working as an assistant programmer of IT department in Green University of Bangladesh. His research interests include Software Engineering and Robotics.

Muhammad Aminur Rahaman was born in Tangail, Bangladesh, in 1981. He received his Ph.D. from the Department of Computer Science & Engineering, University of Dhaka, Bangladesh in 2018. Rahaman has completed his B.Sc. and M.Sc. degree from the Department of Computer Science & Engineering, Islamic University, Kushtia, Bangladesh in 2003 and 2004, respectively. He is a founder Director of Worldgaon (Pvt.) Limited. He is working as an Assistant Professor and Program Coordinator at the Department of Computer Science and Engineering, Green University of Bangladesh, Dhaka, Bangladesh since September 2018 to present. His current research interests include Computer Vision, Image Processing, Human-Computer Interaction System, Robotics and Software Engineering. He is a senior member of IEEE.