Simplification and Simulation of Digital Circuits in Object-Oriented Programming

Shrinidhi Gindi\(^1\) Momin Uzair\(^1\) Chotta Mohommad\(^3\) Pankaj Bhayde\(^4\)

\(^1\)Assistant Professor, \(^2,3\)B.E Students
\(^4\)Department of Information Technology

M.H. Saboo Siddik College of Engineering, Mumbai University, Mumbai, India.

Abstract— the purpose of proposed system is to implement desktop software whose main aim is Simplification of Digital Circuit. The paper aims in automating all the steps that are involved in solving the digital circuit. This paper focuses to cover and integrate all features like Boolean equation simplification, simulation, circuit simplification etc. On the other hand, electronic industry requires designing of very complex circuits which consumes a lot of time. For various purposes different circuits has to be designed. The objective of this paper is to identify how Digital Circuit Simplifier will help in creating new and better circuits.

Key words: Digital Circuit Simplifier, Product of Sum, Sum of Product, Problem Solver, Boolean equation Simplification

I. INTRODUCTION

Digital Circuit Simplifier is software which helps the user to solve the problems related to digital circuit simplification. The user will first have to install the software, and after installation they will be allowed to use it. However the user must have some basic knowledge about the digital circuit and should be able to provide the correct input. If the input is in not proper format the software will not be able to process or analyze it.

The software allows the user to design a circuit with the help of drag and drop feature provided in our software. The software is able to provide the output of the designed circuit. This software also has the feature of simplifying a complex circuit. The Software can also develop a digital circuit based on the Boolean expression provide by the user as input. with the help of various features available in software the user will be able to solve his/her problem quickly and accurately. Digital Circuit Simplifier uses a completely new algorithm for solving complex digital circuit problems. The software can also provide an equivalent Boolean expression of a digital circuit.

Digital circuits are collections of devices that perform logical operations on two logical states, represented by voltage levels. Standard operations such as AND, OR, INVERT, EQUIVALENT, etc. are performed by devices known as gates. Groups of compatible gates can be combined to make yes/no decisions based on the states of the inputs. For example, a simple warning light circuit might check several switch settings and produce a single yes/no output. More complicated circuits can be used to manipulate information in the form of decimal digits, alphanumeric characters, or groups of yes/no inputs. \(^[1]\) These notes are intended to familiarize you with the elementary principles of this field. In order to solve or simplify Boolean expression or digital circuits various rules are used, some of the are listed below.

A. Commutative Law of Addition and Multiplier:
- \(A + B = B + A\)
- \(AB = BA\)

The commutative law of addition for two variables is written as:

![Image](https://via.placeholder.com/150)

Fig. 1: Commutative Law

B. Associative Law of Addition and Multiplier:
- \((A + (B + C)) = ((A + B) + C)\)
- \((A(BC)) = (AB)C\)

The associative law of addition is written as follows for three variables:

- \((A + (B + C)) = ((A + B) + C)\)

This law states that when ORing more than two variables, the result is the Same regardless of the grouping of the variables. Fig below illustrates this law as applied to 2-input OR gates.

![Image](https://via.placeholder.com/150)

Fig. 2: Associative Law

C. Distributive Law:
- \((A(B + C)) = AB + AC\)
- \((A + B)(C + D) = AC + AD + BC + BD\)
- OTHER RULES
  - \(A \cdot 0 = 0\)
  - \(A \cdot 1 = A\)
  - \(A + 0 = A\)
  - \(A + 1 = 1\)
  - \(A \cdot A = A\)
  - \(A \cdot \bar{A} = 0\)
D. Karnaugh Map:
The Karnaugh map, also known as the K-map, is a method to simplify boolean algebra expressions. Maurice Karnaugh introduced it in 1953 as a refinement of Edward Veitch’s 1952 Veitch diagram. The Karnaugh map reduces the need for extensive calculations by taking advantage of humans’ pattern-recognition capability. It also permits the rapid identification and elimination of potential race conditions.

The required boolean results are transferred from a truth table onto a two-dimensional grid where the cells are ordered in Gray code, and each cell position represents one combination of input conditions, while each cell value represents the corresponding output value. Optimal groups of 1s or 0s are identified, which represent the terms of a canonical form of the logic in the original truth table. These terms can be used to write a minimal boolean expression representing the required logic.

Karnaugh maps are used to simplify real-world logic requirements so that they can be implemented using a minimum number of physical logic gates. A sum-of-products expression can always be implemented using AND gates feeding into an OR gate, and a product-of-sums expression leads to OR gates feeding an AND gate. Karnaugh maps are used to simplify real-world logic requirements so that they can be implemented using a minimum number of physical logic gates. A sum-of-products expression can always be implemented using AND gates feeding into an OR gate, and a product-of-sums expression leads to OR gates feeding an AND gate.[3]

Karnaugh maps can also be used to simplify logic expressions in software design[4]. Boolean conditions, as used for example in conditional statements, can get very complicated, which makes the code difficult to read and to maintain. Once minimized, canonical sum-of-products and product-of-sums expressions can be implemented directly using AND and OR logic operators.[2]

Karnaugh maps can also be used to simplify logic expressions in software design. Boolean conditions, as used for example in conditional statements, can get very complicated, which makes the code difficult to read and to maintain. Once minimized, canonical sum-of-products and product-of-sums expressions can be implemented directly using AND and OR logic operators.

1) Resulting Product Terms:
If X is a variable that has value 0 in all of the squares in the grouping, then the literal X’ is in the product term.
If X is a variable that has value 1 in all of the squares in the grouping, then the literal X is in the product term.
If X is a variable that has value 0 for some squares in the grouping and value 1 for others, then neither X’ nor X are in the product term.

II. PROPOSED SYSTEM
Our software will follow these steps for solving digital equation
1) Step 1: It will first except the equation from user
2) Step 2: It will interpret the equation and then place it into the K-Map
3) Step 3: After correctly entering the values in K-Map it will start the formation of pairs.
4) Step 4: It will first search for the of eight
5) Step 5: If a pair of eight is found successfully, it will group them and will generate the output for the group
6) Step 6: If the pair of eight is not found it will look for the group of four, two.
7) Step 7: If the groups are found then it will generate the output for it else it will select the single element and produce the output
8) Step 8: The individual output of the groups will be merged together and will be stored in a node.
9) Step 9: The process will go on until all the possible results are obtained.
10) Step 10: Once all the possible results are obtained it will scan the tree for the best result.
11) Step 11: The best result i.e. the one with the least o of groups will be displayed as the final output In order to simulate the output of the Boolean Equation the following steps will be performed
1) Step 1: The software will accept the the equation as input from user
2) Step 2: It will accept the values i.e. true or false from of the variables from the user
3) Step 3: Once it has the values of the variables it will start to simulation process by using Artificial Intelligence.
4) Step 4: With the help of truth table it will develop the correct output.
5) Step 5: The output generated will be displayed to the user.

The paper focuses on these functionalities:
- Boolean equation simplification
- Digital equation simulation
- Karnaugh map simplification

A. Boolean Equation Simplification:
- This system provides functionality to solve the digital equation which is very DIFFICULT to solve manually.
- The system starts with taking equation as input from the user.
- Check for correctness of equation.
- If any error found in the equation it will not accept the input.
- If the input is in correct format it will process the input and provide simplified equation of it.

B. Digital Equation Simulation:
- In order to simulate the output of the digital equation the user will first have to enter the equation in the text field.
- After entering the equation the user will be asked to enter the values of the variables i.e. true or false.
- Once the value has been provided by the user he/she will click on the GetOutput button.
- The GetOutput button will display the result of the equation.

C. Karnaugh Map Simplification:
- In K-map simplification the user will first have to select the option from drop down menu (SOP or POS) and insert one’s or zero’s accordingly. Compute button will give simplified output.
- Reset button will set all the value to zero.

III. RESULT AND DISCUSSION

In equation simplification (refer Fig. 4), the software will first take equation as input in standard SOP or POS form. After accepting the equation it will validate it. If equation is incorrect then it will notify to the user. If equation is in standard SOP or POS form then software will simplify it and show output on screen. Clear button is used to clear input text field.

K map simplifier (refer Fig. 5) is used to simplify the equation. In this user will first select the option from drop down menu (SOP or POS) and insert one’s or zero’s accordingly. Compute button will give simplified output. Reset button will set all the value to zero.

IV. CONCLUSION

The paper focuses on developing the software which will help the users to successfully solve, simplify, simulate digital equation. With the help of this software users will be able to solve complex digital equations easily.

REFERENCES