S.Khedika[1], R.Rathnathevi[2], S.Sagirtha[3],
T.Mathialakan[4] and S.Kanaganathan[1]

Learning tool for converting Boolean expression and Logic circuit diagram

(1), (2), (3),(4) Department of Physical Science, Faculty of Applied Science,
Vavuniya Campus of the University of Jaffna, Vauniya, Sri Lanka .
(e-mail: kthysubra@yahoo.com)

Abstract: An endeavor has made to develop an application software tool called "Logic Circuit Diagram Designer". Logic Circuit Diagram Designer is a learning tool for Logical Circuit Designing and simplifying Boolean expression. This paper explains algorithm and methodology for transforming Boolean Expression into Logic Circuit diagram and transforming Logic circuit diagram into Boolean Expression. The Karnaugh-map technique is used to simplify the Boolean Expression. The versatile software Logic Circuit Diagram Designer has developed using C# language in Microsoft Visual Studio.Net 2008.

Keywords: Logical circuit, Boolean expression, Karnaugh-map

Introduction

Electronic Digital Design is playing a major role in today's world in every routine. Digital design is the design of hardware for computers.

Problem Specification and Background

Transformation between Logic circuits and Boolean expression is an essential for the secondary educational students. Since the Logic circuit is a diagram it is needed to transform it into Boolean Expression, simplify it and transform it back to the Logic circuit diagram is a good practice for secondary educational students. This learning tool is developed to facilitate user friendly interface to draw Logic circuit diagram and to simplify the Boolean expression using the Karnaugh map technique.

Logic Circuits

An electronic circuit that is designed for two-state operation is called digital circuits. Logic circuits are part of the digital circuits which is constructed by logic gates. Logic gate is a combination of different electronic components, which takes one or more logic level inputs and produces a single logic level output. The actual logic gates circuits are made with diodes and Transistors. In logic gates, the output can have only one of the two possible states, i.e., either a high level 1 or low level 0.

Digital circuits are made up with three basic logic gates: NOT gate, AND gate and OR gate.

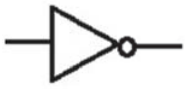


Fig. 1: Symbol of NOT gate

Fig. 2: Symbol of AND gate

Fig. 3: Symbol of OR gate

Boolean Expression

The function of the individual gates or combinations of gates may be expressed in a logical statement known as Boolean Expression. Boolean Expression is simplified by using many techniques some of them are: Boolean algebra, Logical Adjacency, Karnaugh map, Quine-McClusky algorithm.

There are two methods for obtaining the Boolean expression that perform the logic specific in the Truth table.

1. Sum of Product Method
   E.g.:X = (A.B.C)+(A.$\overline{B}$.C)+($\overline{A}$.B.$\overline{C}$).
2. Product of Sum Method
   E.g.:X = (A+B+C).(A+$\overline{B}$+C).($\overline{A}$+B+$\overline{C}$).

Karnaugh map

Karnaugh map provides a pictorial method of grouping together expressions with common factors and therefore eliminating unwanted variables. The Karnaugh map provides a simple and straight-forward method of minimizing Boolean expressions.

The Karnaugh map uses the following rules for the simplification of expressions by grouping together adjacent cells containing ones.

1. No zeros allowed.
2. No diagonals.
3. Only power of 2 numbers of cells in each group.
4. Groups should be as large as possible.
5. Everyone must be in at least one group.
6. Overlapping allowed.
7. Wrap around allowed.
8. Fewest numbers of groups are possible.

Methodology

Represent the Boolean expression in the Karnaugh-map

An algorithm was developed to represent the input Boolean expression in the Karnaugh-map. The flow of the algorithm is shown below. The input string is decomposed into sub strings by splitting using '+'. Then each sub string is recognized by each character to identify the respective location for representing in the Karnaugh-map.

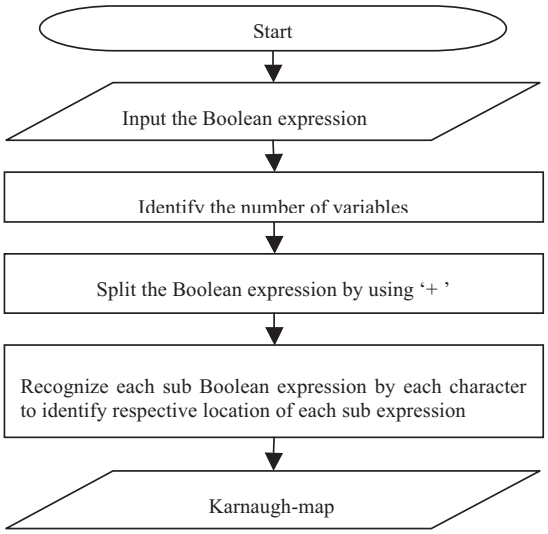


Fig. 4: Represent the Boolean expression in the Karnaugh-map

Simplify the Boolean expression

Karnaugh–map technique was used to simplify the Boolean expression. This technique is applied using a simplification algorithm proposed by Terry A. Scott on his paper. The following is the part of text of the paper submitted by Terry A. Scott.

"There are two possible approaches to solve a Boolean expression. The approach that is usually used by humans is to find the largest number of cells that can be joined and move down to smaller groupings. After some consideration it was concluded that for the computer algorithm, starting with the smallest groupings (pairs) and moving up toward larger groupings would be easier. Once an algorithm is established for pairing minterms then higher order terms can be obtained by joining terms of the previous level grouping. Here these are the basics steps for the algorithm.

One: Where possible, pair up all minterms or cells that contain ones. You must obtain all legal pairs even though this may lead to extra terms. All pairs are required because it is not possible to tell at this stage which pairs may join together to make groups of four.

Two: Remove those single cells that are incorporated into the pairings. Without this, some terms would unnecessarily be included as minterms and also in pair terms.

Three: Group four pairs to obtain groups of four. Again you must obtain all groups of four for the same reasons as given above.

Four: Once all groups of four have been obtained, drop out of the pair list those that have been included in the groups of four. No need to include something in the pairs list that has already been handled in the groups of four.

Five: Continue to try to group together previous groupings essentially repeating steps three and four above until you reach 2 raised to the power of the number of variables.

Six: Output results by displaying the terms for the minterms that weren't joined into higher order groupings, groupings of two cells, groupings of four cells, . . . until all the terms are displayed."

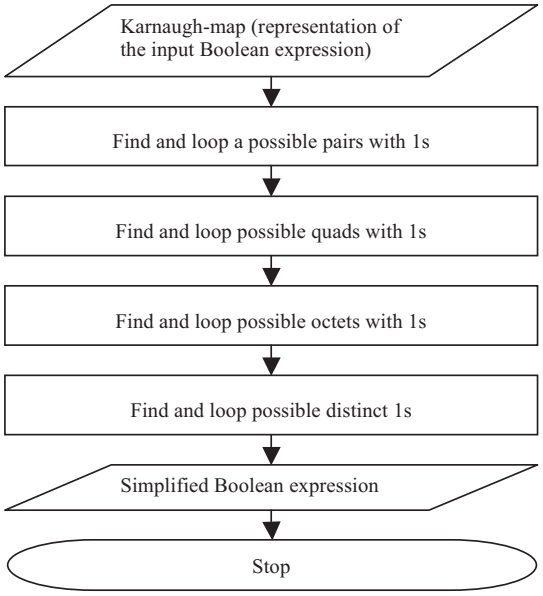


Fig. 5: Simplify the Boolean expression

The Karnaugh-map is represented in a 1- diamensional array by retrieving the locations of 0s and 1s. Then number of 1s in the 1-diamensional array is counted to loop distinct 1s.Then all possible pairs of adjacent 1s are found and stored in a 2-dimensional array. Then possible quads are found using the pairs. Then octets are found using the quads. Then pairs are extracted which have not been looped for quads or octets. Finally, the simplified Boolean expression is formed using the looped 1s.

Boolean expression into circuit

An algorithm was developed to transform the Boolean expression into Logic circuit. The flow of the algorithm is shown below.

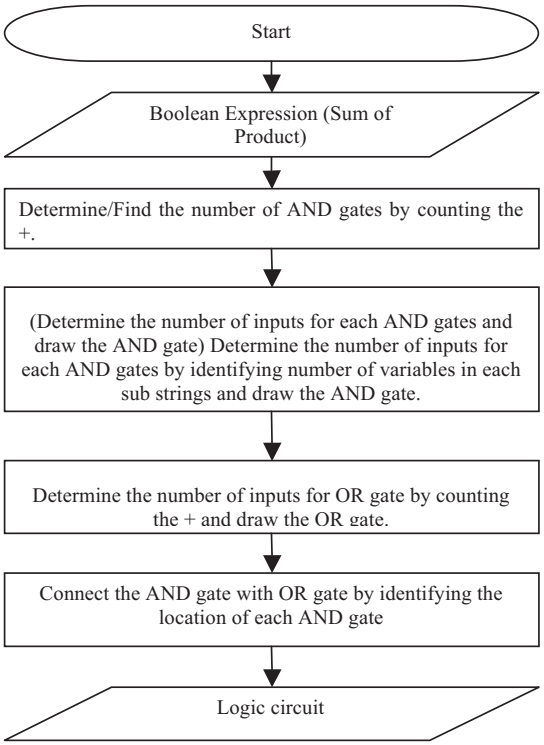


Fig. 6: Transform the Boolean expression into circuit

Logic circuit to Boolean expression

A user-friendly UI is designed with tools for drawing circuit diagram. The end-user could draw a logic circuit diagram by drag and drop the gate tools and connectors from the tool box. Each gate tool and connectors generate indexes when they drop into the drawing space.

A program was coded to generate index and gather the coordinates of those tools. This information is stored in a data structure called "HaspMap".

HaspMap is a data structure, where the data are stored as key-value pair. An algorithm was developed to generate Boolean expression using the data stored in the HashMap.

Results & Discussion

The Fig. 8 represents the user interface to input the Boolean expression for transforming to Logic circuit diagram. The Boolean expression should be a Sum of Product and should have maximum of four variables.

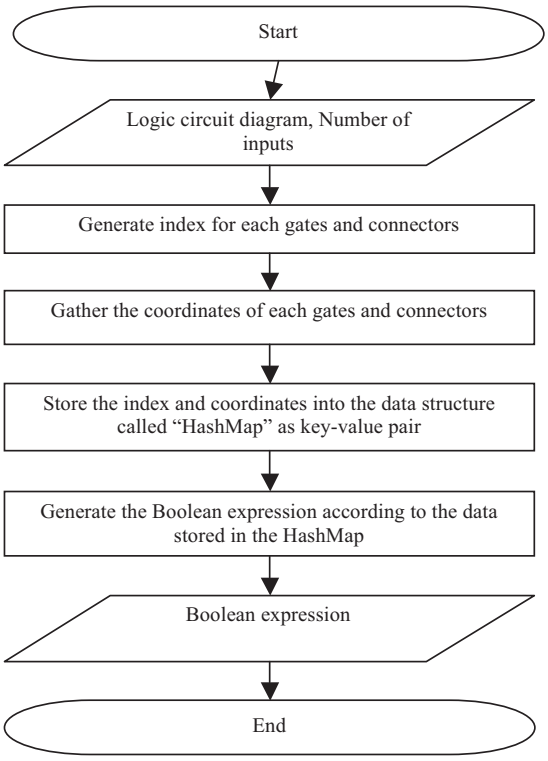
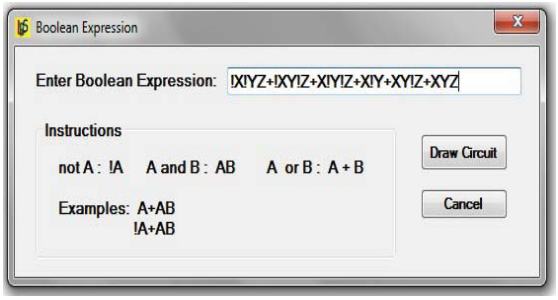


Fig. 7: Transform Logic circuit diagram into Boolean expression



The Fig. 9 represents the logic circuit diagram for the above Boolean expression when click the "Draw Circuit" button.

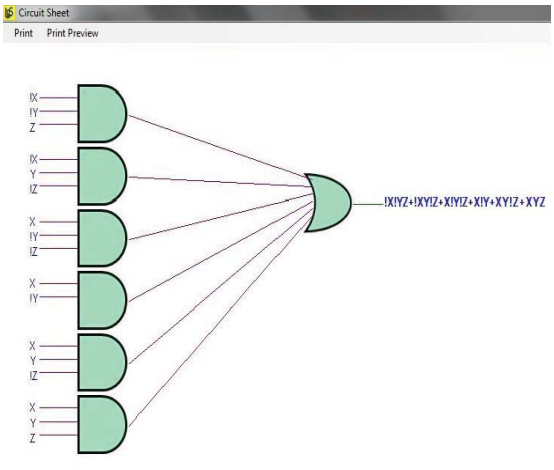


Fig. 9: Screenshot of output showing corresponding logic circuit diagram for the Boolean expression

The Fig. 10 represents the user interface to input the Boolean expression for the simplification. The Boolean expression should be a Sum of Product and should have maximum of four variables. The simplified Boolean expression is displayed when click the button "Simplify".

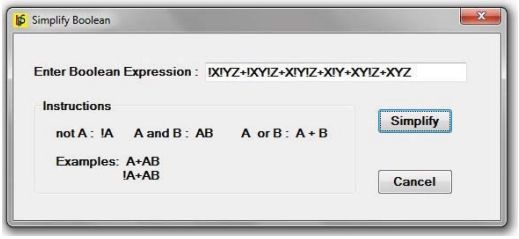


Fig. 10: Screenshot of the user interface for input Boolean expression for the simplification

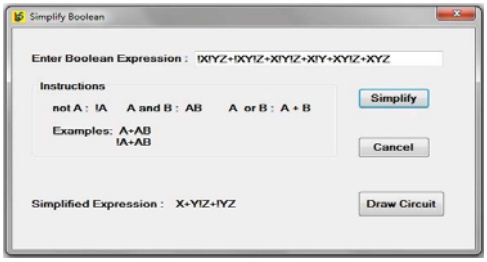


Fig. 11: Screenshot showing the simplified expression

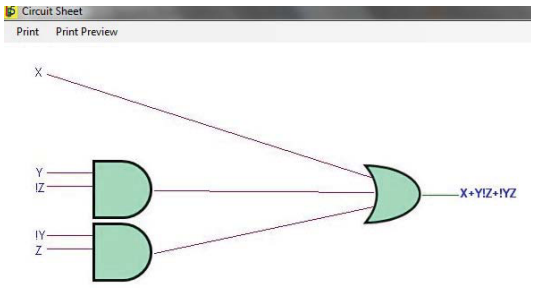


Fig. 12: Screenshot showing the circuit diagram for the simplified expression

The results show that the *Logic Circuit Diagram Designer* transformed the Boolean expression into Logic circuit diagram and simplified the complex Boolean expression (Sum of Product).

Conclusion

Logic Circuit Diagram Designer to transform the logic circuit diagram into Boolean expression and vice versa.

Logic Circuit Diagram Designer consists of 3 main parts. They are:

1. Transform the logic circuit diagram into Boolean expression.
2. Transform the Boolean expression into logic circuit diagram.
3. Simplify the Boolean expression.

The limits for ***Logic Circuit Diagram Designer:***

- Boolean expression must be Sum of Product
- The maximum number of variables is four.

Future Plans:

1. Display the output signal to an input (Boolean expression/Logic circuit diagram) through port.
2. Design an interface to input the Boolean expression through the K-map as well as through the Truth table rather than Boolean function.

References

Belton, D., 1998. Karnaugh Maps. [Online] Available at:http://www.ee.surrey.ac.uk/Projects/Labview/minimisation/karnaugh.html [Accessed 02 September 2011].

Chappell, D., n.d. In Understanding.NET-A tutorial & analysis.

J.Oberg-Delhi, R., 2002. In Introduction to C# using.NET. Pearson Education.

Rajendra, D.J.C.N., 2007/2008. In BASIC ELECTRONICS AND DIGITAL LOGIC DESIGN.