IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 7, July 2021

DOI: 10.17148/IARJSET.2021.8760

IMPLEMENTATION OF QUINE MC-CLUSKEY METHOD ON FPGA

Ayesha Rosheen,Bindu J,Chaithra VM,Sunil Kumar G R

K S Institute of Technology, Bangalore, India

Abstract: The Quine-McCluskey method is one of the popular method and is particularly used when both karnaugh method and variable entrant map (VEM) approach fails, it is useful in minimizing logic expressions for larger number of variables when compared with minimization by Karnaugh Map or Boolean algebra. It was developed in 1956 by Edward J.MC Cluskey. Karnaugh map could solve only for smaller variables whereas Quine Mc- Cluskey can solve for higher variables. Compared to other techniques, QM method is more executable and can handle more variables. The digital gates are basic electronic component of any digital circuit. Digital circuit should be simplified to reduce its cost by reducing number of digital gates required to implement it. To achieve this, we use Boolean expression that helps in obtaining minimum number of terms and does not contain any redundant pair. Karnaugh map (K-map) and Quine-McCluskey(QM) methods are well known methods to simplify Boolean expression. K-map method becomes complex beyond five variable Boolean expression. Minimization of Boolean function is able to optimize the algorithms and circuits. Optimization to Quine-McCluskey Method increases its performance by reducing number of comparisons between minterm list in determination of prime implicants. Modified Quine-McCluskey method (MQM) can be implemented to any number of variables. The concept that runs behind Quine McCluskey is the combining method approach. It has easy algorithm than Karnaugh, so it is efficient and can be easily implemented in computer algorithms.

Keywords: Quine mc-cluskeey algoritm, Xilinx ISE software, Filed programmable gate array FPGA, Karnaugh map, higher or larger variables.

A. INTRODUCTION

The Quine-Mc-cluskey algorithm is one such algorithm which is easy to implement on the hardware. This tabular method of prime implicants is used for minimizing the Boolean expression, Simplification of Boolean expression is a practical tool to optimize programing algorithms and circuits. Several techniques have been introduced to perform the minimization, including Boolean algebra (BA), Karnaugh Map (K-Map).

K-Map is a diagrammatic technique based on a special form of Venn diagram, It is easier to use than BA but usually it is used to handle Boolean expression with no more than six variables, When the number of variable in Karnaugh map exceeds six, the complexity of the map is exponentially enhanced, and it becomes more and more cumbersome, with increasing in the number of inputs, the pattern recognition in Karnaugh map can be difficult or sometimes even impossible.

The quine mc-cluskey method does not require pattern recognition. It includes steps like finding all prime implicants of the function and selecting a minimal set of prime implicants of the function. Functionally identical to K-Map, QM method is more executable when dealing with larger number of variables and is more suited for programming the computer and gives the simplified minimal form of Boolean functions that can be reached.

Algebraic method of minimization is slow and error prone. For these reasons some other procedures have been developed. Karnaugh map provides the ordinary method for simplifying switching functions, although it is limited to the problems with five or less input variables. When the number of input variables is greater than 5, the tabular method for simplifying switching functions developed by Quine and McCluskey is used. This technique is suitable also for problems with more than one output.

Besides, the Quine-McCluskey method is easier to be implemented as a computer program. Quine (1952) and McCluskey (1956) have suggested the above method of simplification which is considered the most useful tabular procedure and described in most books for logical minimization. In recent years some modified (simplified) algorithms with higher speed of execution have been observed too.

Using Boolean laws, it is possible to minimize digital logic circuits by reducing the original number of digital components (gates) required to implement digital circuits. This will reduce the chip size, the cost and increases the speed of circuit. Since minimization with the use of Boolean laws is neither systematic nor suitable for computer implementation, several algorithms were proposed in order to overcome the implementation issue. Karnaugh proposed a technique for simplifying

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 7, July 2021

DOI: 10.17148/IARJSET.2021.8760

Boolean expressions using an elegant visual technique, which is a modified truth table intended to allow minimal sum-of products (SOP) and product-of-sums (POS) expressions to be obtained (Karnaugh, 1953).

B. PROPOSED METHODOLOGY

This paper talks about the The simplification of large variables of an expression is our main concern over here, for which we are using QuineMcCluskey algorithm
or the method of prime implicants which is used for minimization of Boolean functions. The other algorithms like Karnaugh method uses maps, which becomes very difficult to design as the number of input variables increases and, also the Pattern recognition of adjacent cells becomes impossible, therefore we are using an alternative method, that is Quine-McCluskey method. So, this method firstly includes the writing of code in hardware description language (HDL), this code helps us to minimize the higher variables to simple form using Quine Mc-Cluskey algorithm. Then verify the HDL code by executing it, and then implement the verified code on Field- programmable gate array kit (Spartan-6). This algorithm steps includes listing all the minterms in binary form and arrange the minterms according to number of 1's. Compare each binary number with every term in the adjacent next higher category and if they differ only by one position, put a check mark and copy the term in the next column with '-' underscore symbol in the position that they differed. Then apply the same process for the result column and continue these cycles until a single pass through, cycle yields no further elimination of literals. Then list all the prime implicants. Ultimately select the minimum number of prime implicant's which must cover all the minterms. This HDL code which is written according to the step's involved in solving the Quine-McCluskey algorithm should be executed finally and implement it on FPGA kit.

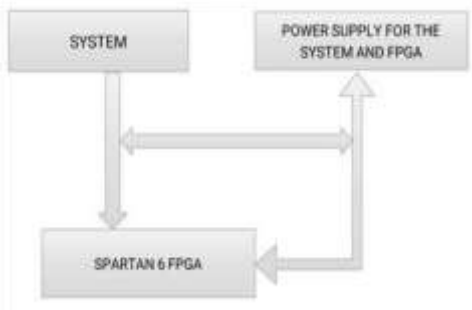

Fig no 1 Block diagram

As shown in the block diagram above we must first provide the power supply for both the system and FPGA kit. The HDL code which is written according to the step's involved in solving the Quine-Mc-cluskey algorithm should be executed later. After executing the HDL code successfully, it should be implemented on the Field Programmable Gate Array(FPGA)/Spartan 6 FPGA board.
Spartan-6 built on a proven 45nm technology; it offers the 1. Shipments to at least 2027 2.12-week lead times 3.ISE tools on windows 10. Spartan®-6 devices offer industryleading connectivity features such as high logic-to-pin ratios, small form-factor packaging, Micro Blaze, and a diverse number of supported I/O protocols. Ideally suited for a range of advanced bridging applications found in consumer, automotive infotainment, and industrial automation.

STEPS INVOLVED IN QUINE MC-CLUSKEY ALGORITHM

List all minterm's in the binary form.
Arrange the minterm's according to number of 1's.
Compare each binary number with every term in the adjacent next higher category and if they differ only by one position, put a check mark, and copy the term in the next column with '-' in the position that they differed.
Apply the same process described in step3 for the result column and continue these cycles until a single pass through, cycle yields no further elimination of literals List all prime implicant's.
Select the minimum number of prime implicant's which must cover all the minterm's

C. HARDWARE AND SOFTWARE DESCRIPTION

FPGA

FPGA stands for Field Programmable Gate Array, and it consists of array of logic gates which stores digital logic specified from Hardware Description Language (HDL). The Spartan FPGA Project kit provides a powerful, self-contained development platform for designs targeting the new Spartan FPGA from Xilinx.

IARJSET
ISSN (O) 2393-8021, ISSN (P) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 8, Issue 7, July 2021
DOI: 10.17148/IARJSET.2021.8760

Fig no 2 FPGA

XILINX ISE TOOL

Xilinx ISE (Integrated Synthesis Environment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize (compile) their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device.

Xilinx officially supports Microsoft Windows, Red Hat Enterprise 4, 5, & 6 Workstations (32 & 64 bits) and SUSE Linux Enterprise 11 (32 & 64 bits). Certain other GNU/Linux distributions can run Xilinx ISE Webpack with some modifications or configurations, including Gentoo Linux, Arch Linux, FreeBSD and Fedora.

Fig no 3 XILINX TOOL

The Design hierarchy consists of design files (modules), whose dependencies are interpreted by the ISE and displayed as a tree structure. For single-chip designs there may be one main module, with other modules included by the main module, similar to the main() subroutine in C++ programs. Design constraints are specified in modules, which include pin configuration and mapping. The Processes hierarchy describes the operations that the ISE will perform on the currently active module. The hierarchy includes compilation functions, their dependency functions, and other utilities. The window also denotes issues or errors that arise with each function.

VERILOG HDL

Fig no 4 Verilog symbol

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 7, July 2021

DOI: 10.17148/IARJSET.2021.8760

Verilog, standardized as IEEE 1364, is a hardware description language (HDL) used to model electronic systems. It is most commonly used in the design and verification of digital circuits at the register-transfer level of abstraction. It is also used in the verification of analog circuits analog circuits and mixed-signal circuits, as well as in the design of genetic circuits. In 2009, the Verilog standard (IEEE 1364-2005) was merged into the SystemVerilog standard, creating IEEE Standard 18002009. Since then, Verilog is officially part of the SystemVerilog language. The current version is IEEE standard 1800- 2017.

Verilog was one of the first popular hardware description languages to be invented. It was created by Prabhu Goel, Phil Moorby and Chi-Lai Huang and Douglas Warmke between late 1983 and early 1984. Chi-Lai Huang had earlier worked on a hardware description LALSD, a language developed by Professor S.Y.H. Su, for his PhD work.

D RESULTS

The proposed methodology provides us the simplified expression using the quine mc- cluskey algorithm, which has a lot of advantages as in the gates used will be very less and the space occupied will also be very less. The quine mc cluskey algorithm is a efficient algorithm when compared to Karnaugh map and MEV methods as in this algorithm we can use higher variable expression which is limited up to 4 variables in Karnaugh map aad in other algorithms and it does not require the pattern recognition.

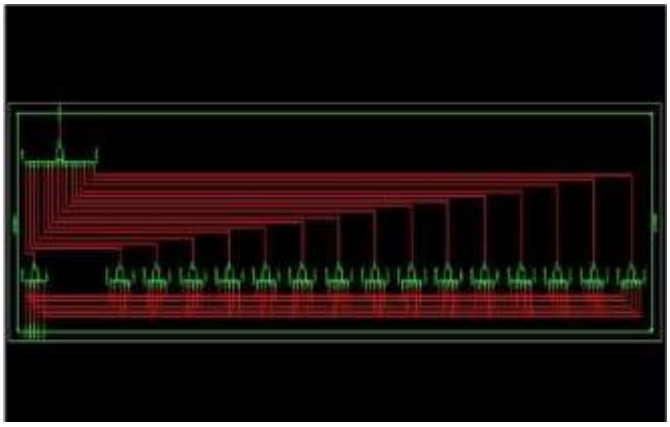

Fig no 4 RTL View for un-simplified expression

Register-transfer level (RTL) is a design abstraction which models a synchronous digital circuit in terms of the flow of digital signals (data) between hardware registers, and the logical operations performed on those signals. The above RTL view is for the un-simplified expression which contains 17 gates, and it was done before executing it in algorithm of quine mc-cluskey method. By which we get to know that the complexity of the un-simplified expression execution and it also involves many gates, and it is difficult to simplify in other methods except quine mc-cluskey method.

Fig no 5 Simulation result for un-simplified expression

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 7, July 2021

DOI: 10.17148/IARJSET.2021.8760

The output waveform of the un-simplified expression is as shown in fig 6.2. After executing the Verilog HDL code in Xilinx software for the un-simplified expression, simulation is done. Here we have consider the 5 bits 11011 as the input and got the output as low which is compared with manual calculation also and the same result is obtained. But the disadvantage over here is that the gates are more, when the gates are more then relatively the space will also be more and the execution time will also be more, which is the main concern that should be solved.

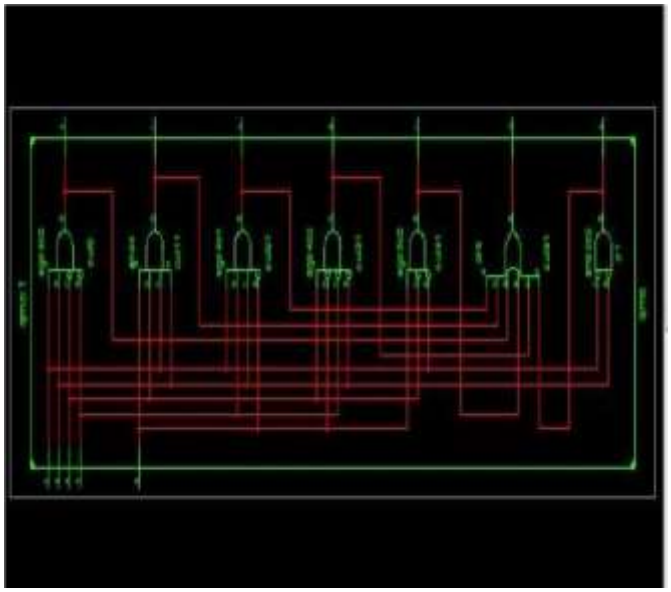

Fig no 6 RTL view for simplified expression

Here we can notice that the gates required after minimizing the expression is significantly reduced as compared to the before un-simplified expression and simultaneously with the gates, the space and the execution time is also been reduced. When we compare the RTL schematic of unsimplified expression in the figure and the RTL schematic of minimized expression in the figure The difference is clearly noticeable.

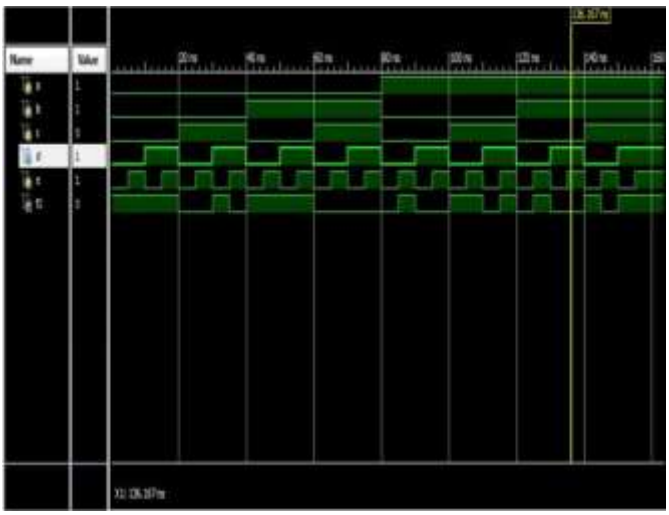

Fig no 7 Simulation result for simplified expression

The final output waveform is as shown in fig 6.5. The output of the Simplified expression. After executing the Verilog HDL code in Xilinx software for the simplified expression, simulation is done. Here again we have considered the same 5 bits 11011 as the input as we considered Previously for the un-simplified expression and got the output as low which means we are getting the proper result for the simplified or minimized expression too And is again compared with the manual calculation which is also the same as obtained.

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 7, July 2021

DOI: 10.17148/IARJSET.2021.8760

HARDWARE OUTPUT

Fig no 8 FPGA RESULT

The hardware output is as shown in fig 6.5. The output of the Simplified expression is implemented using the Spartan 6 FPGA kit. As we can clearly see in the above fig 6.5 that when we have given the input of 11011 5 bit, the output is turned out to be low as we needed, and likewise we have executed for various inputs and got the proper output, in which one result is as shown above.

E FURTHER WORK

As a future scope, not only for 5 input simple POS and SOP expression can be implemented but we can also implement for more higher expression which involves more than 5 variables. Further, we can also implement the algorithm to the circuits which includes 5 variable comparator, 5 variable subtractor, 5 variable Adder, 5 variable multiplexer etc.., Not only for 5 variables, we can even design the circuits which includes much higher variables as a input and also implementiton FPGA.

F CONCLUSION

As we know after been implementation, we have compared the un-simplified and simplified expression considering the input as 11011, by which we got to know that the number of gates required to execute the expression is more before simplifying it, as shown in RTL Schematic of un-simplified from the shown figures. So, we have used Verilog code to simplify the expression using quine mc-cluskey algorithm, and after executing the simplified expression the number of gates, space and percentage is decreased as shown in RTL Schematic of simplified expression in the figure mentioned. And this will not affect the simulation results as well, and it will remain the same as shown in simulation results of both un-simplified and minimized i.e, the simplified expression and successfully been implemented on FPGA. And checked the results for not only 11011 but for all the terms.

G REFERENCES

[1] J. Safaei and H. Begiy. Boolean function minimization: The information theoretic approach. In Proc. 15th IEEE/ACM workshop on Logic and Synthesis, pages 37– 41, 2006.

[2] Jain, Tarun Kumar, Dharmender Singh Kushwaha, and Arun Kumar Misra. "Optimization of the quine-mccluskey method for the minimization of the boolean expressions." Autonomic and Autonomous Systems, 2008. ICAS 2008. Fourth International Conference on. IEEE, 2008.

[3] BASÇÍFTÇÍ, Sirzat KAHRAMANLI Fatih. "Boolean functions simplification algorithm of O (N) complexity." Mathematical & Computational Applications 8.3 (2003):

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 7, July 2021

DOI: 10.17148/IARJSET.2021.8760

271-278.

[4] Mahmoud, H. A. H. "A new method for two-level logic minimization." Circuits and Systems, 2003 IEEE 46th Midwest Symposium on. Vol. 3. IEEE, 2003.

[5] Martins, Mayler GA, et al. "Efficient method to compute minimum decision chains of Boolean functions." Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI. ACM, 2011.

[6] Vladislav Manojlović, Kneza Miloša Kosovska
Mitrovica Faculty of Technical Sciences.
Minimization of Switching Functions using QuineMcCluskey Method Vladislav

[7] Milan Petr´ık, Center for Machine Perception, department of Cybernetics Faculty of Electrical Engineering, Czech Technical University Technick´a 2, 166 27 Prague 6, Czech Republic. Quine-McCluskey method
for many-valued logical functions

[8] P. W. Chandana Prasad, Azam Beg, Ashutosh Kumar Singh. Effect of Quine-McCluskey Simplification on Boolean Space Complexity..

[9] Jiangbo Huang department of Biological Sciences, Faculty of Science National University of Singapore, Singapore 117604 Programming implementation of the Quine-McCluskey method for minimization of Boolean expression.

[10] Jadhav Vitthal and Buchade Amar. Modified QuineMcCluskey Method. This paper proposes E-sum based optimization to Quine-McCluskey Method to increase its performance by reducing number of comparisons between minterm list in determination of prime implicants. Modified Quine-McCluskey method (MQM) can be implemented to any number of variables.

