International Journal of Computer Applications (0975 – 8887)
Volume 127 – No.1, October 2015

On Synthesis of Combinational Logic Circuits

Sahadev Roy
Dept. of ECE
NIT Arunachal Pradesh
Yupia, India

Chandan Tilak Bhunia
Dept. of CSE
NIT Arunachal Pradesh
Yupia, India

ABSTRACT

Objective of this paper is to present historiography of logic switching circuits. The research mainly focuses on chronological development and application of logic in the field of electronic and computer applications. This paper briefly discussed on the basic needs of logic synthesis and also discuss few interesting facts and design consideration regarding logic synthesis. It also enhances student’s deep understanding of different logic function minimization technique during a lecture and practical implementation.

General Terms

Combinational circuits minimization and synthesis techniques.

Keywords

Boolean function, Combinational logic, Digital electronics, Maxterm, Minimization, Minterms, Multiple inputs, Switching circuit simplification.

1. INTRODUCTION

Combinational circuit output at any time depends on the present inputs applied on the circuit. For each possible combination of input variables there is only one output. Combinational circuits not contain any memory element and not have any closed feedback loops. Digital switching circuits mainly concern with input variables, output variables and logic gates. Both input and output information are represented by binary states, logic 1 and logic 0. Combinational logic function of n input variables comes from an external sources and m set of output go to an external destination. Theoretically m and n be any positive integer as shown in Fig. 1.

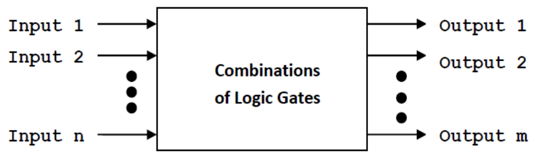

Fig 1: Combinational logic circuit

For n number of input variables, there are 2^n possible combinations of binary input values. This circuit can be described by m Boolean functions, one for each output. This is the basic building blocks of digital electronics. Based upon this simple concept, today world is converted into digital world. In this review paper mainly focused on the chronological development history of logic, representation of logic, different implementation technique.

2. LOGIC TO DIGITAL LOGIC

Digital logic circuits or digital systems originated from philosophy. Modern digital system is nothing but implementation of logic in electronically. The heart of digital system is binary number system, which was refined by Leibniz in early of 1700 [1]. Logic proposition and operations started to be represented by symbols although it had already been expressed by natural language in times of Aristotle, the fourth century B.C. Francis Bacon (1561- 1626), invented a "Bilateral alphabet code" a binary system that used the symbols A and B to develop ciphers and code [2]. The introduction of indices by Mitchell into the logic algebra is the greatest technique use till now to represent logic proposition by symbol means of letter [3]. This concept very useful in today digital design propositions means inputs to a device which are denoted as A, B etc or different combination using subscript.

George Boole in 1854 combine arithmetic and logic which has two truth values either true or false, which has known as Boolean algebra. The objective of these works was to model reasoning and that could be expressed using symbols and a mechanical method of deduction based on rules through the solution of algebraic equations. The main operations of Boolean algebra are the conjunction (AND), the disjunction (OR) and the negation (NOT) are his inventions. A “NOR logic” entry already exists in his famous book the mathematical analysis of logic at page number 21 in 1847 [4]. These relations was introduced for describing logical relations in the same way that addition, multiplication etc. used in general algebra to describes numeric relations. Boolean algebra has been play fundamental roles in the development of digital electronics.

Charles Sanders Peirce simplifies Boolean algebra in more simpler and attractive way. A basic difference between ordinary algebra and algebra of logic is clearly mentioned by Price. Price explain symbol which are used in ordinary algebra they denote things without describing them i.e. in ordinary algebra indices are blank and may be assigned any numerical value to get some result but for today logic design assignment of x as input 1, y as input 2 and z as input 3 means use different token for different input as a result designer don’t know what happen with this operation without translating output into a sensible image means again use of another token for output. He compares logical indices with indices of token and it is blank. He used two values ‘v’ and ‘f’ representing a proposition for true Value and false proposition respectively. Charles also state that “According to ordinary logic, a proposition is either true or false, and no further distinction is recognized. This is the descriptive conception, as the geometers say; the metric conception would be that every proposition is more or less false and that the question is one of amount”. He adopts former view i.e. two vale of logic either true or false Boole chose 1 for v and 0 for f respectively. Logic high and logic low in digital electronics is used for the same purpose to indicate logic high or low means high reference voltage or near to high reference voltage and low reference voltage or very near to low reference voltage value with same notation. Peirce described a notation practice to present any proposion which vale either true or false. Before Peirce any proposition x if true it was written as (v—x) or x otherwise for false case it was written as(x—f) or (1-x). He adopts a new notation style only x means true and x̄ for false value of that proposition which till now used. In his famous paper “On the Algebra of Logic: A Contribution to the Philosophy of Notation”, Peirce redefine propositional logic and presented as in form of icons. In the first icon of algebra is contained in the formula of identity (x −< x). The symbol ‘−<’ he uses to represent similar to implies (→). The icon of second kind i.e. ‘modus ponens’ of hypothetical inference is use to represent “if x, then y (y −< x)”. The third icon is used for the principle of the transitiveness of the copula i.e. the principle of the syllogism in Barbar or chain of consequence. According to this, if in any case y follows from x and z from y, then z follows from x, symbolically, (x −< y) −< (y −< z) then x −< z. The icon of forth kind is modus tollens used for negation. Modus tollens proposition is negation proposition of modus ponens. Using this icon he proof from 2nd icon for negation of y proposition x also false. A fifth icon used for the principle of excluded middle and other pro-positions connected with it. He represent OR logic using '+' sign and AND logic using simply multiplication [3]. Peirce also extended Mitchell’s notation for some (∑) and all (∏). He suggests ∑ for a sum means that function is true if some one of the individuals are true and ∏ for a product means that function is true if entire individuals are true. Till today these two symbols to represent SOP and POS with same meaning.

Allan Marquand student of Peirce in his historical paper "A new logical machine" at page number 20 in the foot note clearly mention A + B means either A OR B or both, which was used as non-exclusive sense and AB means individuals which belong to both class A and B [5]. Peirce expresses deMorgans low symbolically as $\overline{x+y} = \bar{x}\,\bar{y}$ and $\overline{xy} = \bar{x} + \bar{y}$ which was reported in 1885 [4]. This representation are very useful to represent any logic by a single operation, letter describe NOR and NAND. The first axiom system based on NAND was given by Henry Sheffer in 1913 [6]. Whitehead and Russell in 1927 promoted NAND as the appropriate foundation for axiomatic logic.

3. LOGIC IMPLIMENTATION

Charles Stanhope designed first logic demonstration machine which is famous as ‘Demonstrator’ was able to solve both traditional and numerical syllogism mechanically [7]. In 1881-82 Allan Marquand constructed ‘Logical Machine’ with Prof. Rockwood. The machine similar to Prof. Jevons and Prof. Venn but more advanced. The Logic Machine was able to handle four inputs condition and capable to expand desire number of inputs. The Logic Machine is the 1st full functional mechanical device design for a general purpose truth functional logic processor in which all the valid implications of a logical proposition displayed in a systematic way [5]. In 1886, Charles Sanders Peirce in a letter to Marquand described how logical operations could be performed by simply electrical switching circuits. Charles draw two simple electrical circuit where all switch are parallel which and series which able to perform logical multiplication i.e. ANDing and logical addition i.e. ORing respectively [8].

Around 1890, Price and Marquand improved the Logic Machine using electromagnetic switching devices [9]. In 1891 Strowger switching circuit based on electromagnetic relay was based on AND operation [10]. Strowger relay switch operate in step by step in a particular sequence which may be consider as first electromagnetic sequential circuit.

In 1907 Lee De Forest's proposed that modified Fleming valve can be used for implementation of AND logic. Walther Bothe, inventor of the coincidence circuit, the first modern electronic AND gate in 1924, got part of the 1954 Nobel Prize in physics. Konrad Zuse built electromechanical logic gates for his computer Z1 (from 1935–38) [11].

4. SWITCHING CIRCUIT SYNTHESIS

Claude E. Shannon introduced the use of Boolean algebra in the analysis and design of switching circuits in 1936. Shannon represents a switching function with X and its terminal by a, b. He used symbol Xab to represent circuit, the symbol 0 (zero) used to represent the hinderance of a closed circuit, and the symbol 1 (unity) to represent ·the hinderance of an open circuit which are exactly analogous to the Calculus of Propositions used in the symbolic study of logic. He use ‘+’ for each series connection i.e. ANDing operation and ‘•’ for each parallel connection of switch switching elements i.e. ORing operation. He established analogues relation between Boolean algebra and switching system. He successfully correlates De Morgan's low and Huntington’s postulates for symbolic logic with switching function. He also efficiently simplify switching circuit using algebra of logic and identify set of logic formula which are till now valuable for complex logical circuit minimization [12]. Though today use of Price representation i.e. ‘•’ for each series connection and ‘+’ for each parallel connection of switch switching elements. The symbol 0 (zero) used to represent the open circuit, and the symbol 1 (unity) to represent ·the close circuit which are exactly opposite of Shannon’s representation. Due to duality principal of logic all symbolic analysis are still valid. Symbolic analysis of switching circuit plays an important role in logic circuit synthesis. His research changed man’s understanding of digital systems. In November 1937, George Stibitz, at Bell Labs, designed a relay-based computer Model-K, which calculated by using binary addition [13].

5. REPRESENTATION TECHNIQUES

Any logical expression can be executed by using logic gates. Input variables generally indicate by English letters. Value of any inputs variables either without negation (1) or with negation (0) but not both and symbolic representation of that variable are called literal. A combinational circuit can be represented in five different forms or descriptions.

5.1 Algebraic forms

Algebraic forms such as canonical descriptions of logic statements are used most frequently. A sum of product (SOP) expression is a very efficient representation technique of any switching function, because there might be very few terms at which the function is contend and alternatively, complimented of that function expressed as product of sums (POS) expression might be very efficient because there are only a few terms at which the function is mask. As example,

$$Y = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC. \qquad (1)$$

$$Y = \bar{B}\bar{C} + AB. \qquad (2)$$

$$\bar{Y} = (B + C)(\bar{A} + C)(\bar{A} + \bar{B}). \qquad (3)$$

Equation (1) is canonical SOP form equation (2) is minimized SOP form and the same expression presented in POS form in (3), here A, B and C three Boolean input conditions. Bars indicate complement only.

5.2 Tabulated forms

The Inputs combinations are listed for which either outputs becomes logic high or logic low or in don’t care state. K-map procedure is not only tabulated input but also helpful for logic minimization [14]. Fig. 2(a) is Truth table and Fig. 2(b) K-Map representation of equation (1).

Truth Table

Input			Out Put
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

(a)

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1			
A	1		1	1

(b)

Fig 2:Tabular representation (a) Truth table (b) K-Map

5.3 Graphical forms

Graphical forms such as logic networks are other representation techniques. Using simple logic gate logic networks may be represented. Equation (2) may be expressed as fig.3. The circuit obtain from equation (1) after minimization using two inverter, two AND gate and one OR gate.

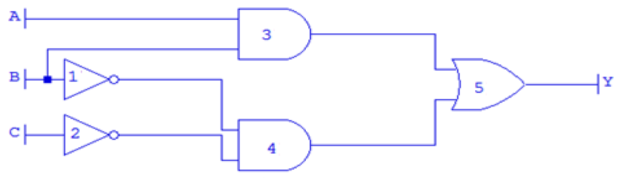

Fig 3: Graphical representation method.

5.4 Statements forms

Using set of statements, like as hardware description language is another representation method. Hardware descriptive language used to describe switching circuit, gate level Verilog code of equation (2) given below,

//Gate level verilog coding of Y = $\overline{\mathrm{B}}\overline{\mathrm{C}}$ + AB

module exp_y(Y, A, B, C) // Define variable for input and output.

input A, B, C; // Input variable

output Y; // output variable

wire b,c,a0,a1; // b, c → Component 1 to 4 and 2 to 4; a0,a1 → Component 3 to 5 and 4 to 5 (Fig.3)

not (b,B); // wire b logic value complement of B

not (c,C); // wire c logic value complement of C

and (a0,b,c); // output a0 is b AND c

and (a1,A,B); // output a1 is A AND B

or (Y,a0,a1); // Y is a0 OR a1

endmodule.

5.5 Coded from

Coded from like, binary coded, gray coded, decimal coded.

Different types of code used to express input output relation, generally decimal coded. Using decimal coded equation (2) and (3) as equation (4) and (5),

$$Y(A,B,C) = \sum m(0,4,6,7). \qquad (4)$$

$$Y(A,B,C) = \prod M(1,2,3,5). \qquad (5)$$

5.6 Drawbacks

All of the representation of logic is not symbolically complete. Switching circuit posses symbolic-value-dependent relationships depend on the interconnection of the logic gates related their truth tables. Another is time dependent relationships like propagation delay times of the component elements to express the validity of variables values and the invalidity of data values. Combinational logic analysis basically performs only functional analysis and treated that there is no propagation delay. Timing analysis of the switching circuit is essential and must be taken care otherwise circuit may not work according to the desired circuit which is obtained from functional analysis. Timing hazards are major problems arising as a result of propagation delays and to overcome this issue timing analysis must be perform with functional analysis. Digital signal have two value 0 or 1. When signal expected to be in a stable state from functional analysis, but it changes vale for a moment due to path difference of multiple source which from glitch [15].

The set of p input variables may assume 2^P states in certain AND or OR conditions called minterm or maxterm, respectively. For each possible combination of input variables output of the combinational circuit either 0 or 1.

There are some important methods to design combinational circuits one of them is traditional algebraic method. In this method Boolean expressions or truth tables can be simplified by using standard methods and simplified expression can be realized by using gates. A truth table concise lists all the possible combination of input variables and corresponding outputs.

Any combination of literals produce output either 0 or 1 for any combinational circuit which presented in each rows of the truth table. The easiest way to represent logic functions using minterms. A minterm is product of literals, in which each input literals appears exactly once for which output of the logic circuit high. In general practice right most variable in the truth table assigned to a weight of 1 and others variables weight increased by two times with respect to the nearest right variable. Using binary to decimal number conversation technique minterms are converted into decimal number. Decimal code of minterm depends on the variable assigned under which positional weight and choice of weight for particular variables varies from designer to designer. For particular minterm equivalent decimal code are not fixed until and unless exact sequence of variables are maintain. For standard Sum of Products (SOP) form or Disjunctive Canonical form (DCF) is logical ORing of the all minterms and represent by listing the decimal code of the minterm for which output of the function become high i.e. exact implementations of the target truth table as example in (6) and (7).

$$f(R,G,B,W) = \sum m(1,5,10,11). \qquad (6)$$

$$f(B,G,W,R) = \sum m(2,6,9,11). \qquad (7)$$

It is designer choice for which variable want to indicate by which letter. In equation (6) and (7), R, G, B and W are four input variable and have different decimal values into DCF. But minimal SOP expression is same $\overline{\mathrm{B}}\overline{\mathrm{R}}\mathrm{W} + \mathrm{B}\overline{\mathrm{G}}\mathrm{R}$ because in the equation (6) and (7) minterms are same as table 1. Cause of different minterm exactly to say Decimal Coded Minterm (DCM) sequence of variables are just change i.e. weight assigned to each variable position are different.

Table 2: Truth Table

W	B	G	R	Output
1	0	0	0	1
1	0	1	0	1
0	1	0	1	1
1	1	0	1	1

For different set of DCM but if minterms are same then minimal SOP expression must be same.

6. MINIMIZATION TECHNIQUES

In design it is required to realize logic expression converting in electronic switching circuits. There are two useful techniques for reducing combinational logic expression and logic diagram to the fewest possible elements, these are mapping and minimization respectively.

Different combinational circuits can implement for same logic function which are producing the correct output response for all input values called functionally equivalent circuit [16]. Best implementation depends on design goals and criteria of the design mainly area analysis, number of gates, number of gate inputs and outputs etc. Implementations after simplification has the following advantages less cost, reduce number of logic gates, reduce complexity, simple circuit and reduce delay [17]. Fan-out factor is difference between gate output and input load factors to which that gate output is connected. Many inputs gates required more area and design whit many gates interconnection between the gates increased and it more complex. The area or size determine by total of both gate area and the area of the interconnection. Minimization of area also play major role in design.

Number of gates in the longest path from input to output i.e. number of level increase intrinsic as well as extrinsic delay as a whole propagation delay increased. Combinational circuits having arbitrary finite gate and wire delays which also taken care in design process. Designing with considering arbitrary gate and wire delay have no grunted implementation due to output does not change monotonically during input transition which causes glitches [18]. The maximum delay obtained for all possible path of data flow and glitches should be minimized.

Combinational logic circuit always depends on present combination of inputs and maps to a single output for each combination of inputs not on the previous inputs [19].

For multilevel combinational logic circuit the time-dependent relationships must be carefully and explicitly coordinated to provide a single output as desired.

Multilevel implementation of logic networks often requires fewer gates and fewer connections than two-level logic networks but designing procedure more complex than those of two-level logic networks [20][21].

Numbers of gate count also increase both static and dynamic power dissipation. Static power dissipated when gates are remaining in the same state. Dynamic power dissipation is typically specified as a function of frequency is dependent on the speed of the switching inputs are changed its value, the faster the logic input changed lead the higher the dynamic power dissipation [22].

The cost of hardware realizations and implementations a logic expression are related to the number of terms present in the minimized expression and to the number of literals present in a term.

Switching circuit simplification become essential due to rapid development in application relay and digital design. Before Veitch, simplification mainly has done using algebraic approach. To avoid lengthily algebraic approach first proposed a chart method [23]. Brooks Vice-President of Computer Control Company extended Veitch Chart to make useful in computer design [24]. Technical note of that company reported by Brooks shows a systematic representation of switching circuit. Symbol of AND gate which was reported there is quite similar with present style. Complete output of NOT, AND, OR, NOT-AND (NAND), NOT-OR (NOR) represented using Truth-Table. Using 3C-PAC which mainly combine gate, buffer (OR), inverter and delay element details design procedure of binary addition, subtraction, multiplication, division, function generation and shift register also reported there [25]. In 1953, Veitch Chart modified by Karnaugh, called Mapping is a famous method for manual simplifying switching functions involving six to seven variables [14]. Application of Consensus concept in logic minimization by Quine [26] and generalized by Tison [27]. Tabular technique or Quine-McCluskey (QM) method of minimization is a special case of application iterative consensus very useful for computer-aided minimization [28].

Another concept of switching function minimization is by using identification of symmetricity, which is more economical than series parallel design. Shannon defines symmetric function as any interchanging between variables leaves the function identically the same. Caldwell classifies symmetry functions into two category where all variables are either all unprimed or all primed and the mixed variable class were some but not all of the variables of symmetry are primed. He further proposed extended K-Map for determination of symmetric function which helpful for synthesis of switching function [29], McCluskey [30], Franz [31], Michalski [32], S. R. Das [33] and many others improved the method of simplification by reducing complexity. Conjunctive and Disjunctive manipulation technique method originated by Nelson [34] which does not required prior expansion of a formula to develop normal form. This method extended by Slagle [35] S. R. Das [36], and many others.

Combinational function possesses topological characteristics using this property cubing algorithm proposed by Roth [37]. Another algorithmic iterative procedure for generating the prime implicants of switching functions by utilizing a new tabular proposed by Das and Khabra [38] using clause-column table. Every column of the clause-column table contained a clause determining the product or sum term. This algorithm was subsequently extended to find minimal realization of multiple output Boolean functions [39]. Bryant proposed the use of Reduced Ordered Binary Decision Diagrams (BDD) as a canonical representation and graph-based algorithms for Boolean function manipulation in 1986 [40]. Almost all of the methods treat the minimization process in to two separate parts, determination of all Prime Implicants and selection of essential Prime Implicants.

7. CONCLUSION

This article presents a state of the art review of logic synthesis. For different minimization techniques and implementation of switching circuit required different representation technique. This study covered different minimization process and implementation technique. The problem of logic minimization is quite old but not dead. It play important role in many area of VLSI synthesis, design of switching system, built in self-test and many other application.

8. REFERENCES

[1] James A. Ryan. 1996. Leibniz' Binary System and Shao Yong's "Yijing". Philosophy East and West, Vol. 46, No. 1, 59-90.

[2] Aschoff, V., "The early history of the binary code," in IEEE Communications Magazine, vol.21, no.1, (Jan. 1983), 4-10.

[3] Peirce, Charles Sanders. "On the algebra of logic: A contribution to the philosophy of notation." in American journal of mathematics, (1885), Vol. 7 No. 2, 180-196.

[4] Boole, George, "The mathematical analysis of logic," in Philosophical Library, (1847).

[5] Marquand, Allan, "A new logical machine," in Proceedings of the American Academy of Arts and Sciences, John Wilson and Son (1885), 303-307.

[6] Sheffer, Henry Maurice, "A set of five independent postulates for Boolean algebras, with application to logical constants." in Transactions of the American Mathematical Society (1913), Vol. 14, No. 4, 481-488.

[7] Computer Museum. Computer museum of America. Available at http://history compuer.com/Modern Computer/thinkers/Stanhope.html last seen July (2015).

[8] Peirce, Charles Sanders. "Writings of Charles S. Peirce: 1857-1866" (1982) vol. 1. Indiana University Press.

[9] Computer Museum. Computer museum of America. http://history-computer.com/ModernComputer/thinkers/Peirce.html last seen (July 2015).

[10] Almond B. Stronger, Automatic telephone-exchange. U.S. Patent No. 447,918. 10 Mar. (1891).

[11] Jess, Jochen AG. "Designing electronic engines with electronic engines: 40 years of bootstrapping of a technology upon itself." in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (2000) Vol.19 No.12 , 1404-1427.

[12] C.E. Shannon, "A symbolic analysis of relay and switching circuits," in Trans. of the American Institute of Electrical Engineers, (1938) Vol. 57, No. 12, 713-723.

[13] M. M. Irvine, "Early Digital Computers at Bell Telephone Laboratories." in IEEE Annals of the History of Computing, (2001), Vol.23 No. 3, 22-42.

[14] Karnaugh, M., "The map method for synthesis of combinational logic circuits," Transactions of the Communication and Electronics, American Institute of Electrical Engineers, Part I: Nov. (1953), Vol.72, No.5, 593-599.

[15] F. Dartu, N. Menezes, J. Qian, L.T. Pillage, "A gate-delay model for high-speed CMOS circuits," Proc. ACM. 31st annual Design Automation Conference (DAC '94), June (1994) 576-580. doi:10.1145/196244.196562.

[16] Y. Matsunaga, "An efficient equivalence checker for combinational circuits," Proceedings of the 33rd annual Design Automation Conference, DAC '96 (1996),629-634.doi: 10.1109/DAC.1996.545651.

[17] S. Roy and C.T. Bhunia, "Constraints Analysis for Minimization of Multiple Inputs Logic Programming," in Elsevier Proc. of International Conference on Signal and Speech Processing (ICSSP-14), Aug. (2014), 61-64, Kollam, India.

[18] K. P. Parker and E. J. McCluskey, "Probabilistic treatment of general combinational networks," IEEE Trans. Computers, June (1975), vol. C-24, 668-670.

[19] K. Fant and S.A. Brandt, "Null Conventional Logic: A Complete and Consistent Logic for Asynchronous Digital Circuit Synthesis," Proc. Int'l Conf. Application-Specific Systems, Architectures and Processors (ASAP 96), IEEE CS Press, (1996) Los Alamitos, Calif. 261-273.

[20] S.M. Nowick and D.L. Dill. "Exact two-level minimization of hazard-free logic with multiple-input changes," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,(1995) ,vol. 14, no. 8, 986-997.

[21] T. Sasao, "Multi-Level Logic Synthesis, "Switching Theory for Logic Synthesis, Springer US, (1999), 229-261.

[22] Jim Stiles, "Power Dissipation," The Univ. of Kansashttp://www.ittc.ku.edu/~jstiles/312/handouts/Power%20Dissipation.pdf.2014.

[23] Veitch, Edward W. "A chart method for simplifying truth functions." In Proceedings of the 1952 ACM national meeting (Pittsburgh), (1952), 127-133.

[24] R. W. Brooks, "Extension of the Veitch Chart Method in Computer Design," meeting of Association for Computing Machinery, Cambridge, Massachusetts, September, (1953), 1-15. Available at http://www.ddp116.org/products/pacs/logic_1955.pdf

[25] R. W. Brooks: Symbolic logic, binary calculation, and 3C- PACS, published by Computer Control Co. Inc., 93 Broad St., Wellesley 57, Mass. (1955). Available at: http://www.ddp116.org/products/pacs/logic_1955.pdf.

[26] Willard V. Quine. "A way to simplify truth functions." American mathematical monthly, (1955), 627-631.

[27] P.Tison, "Generalization of Consensus Theory and Application to the Minimization of Boolean Functions," , IEEE Transactions on Electronic Computers, Aug. (1967) vol. EC-16, no.4, 446-456, doi: 10.1109/PGEC.1967.264648.

[28] McCluskey, Edward J. "Minimization of Boolean Functions." Bell system technical Journal, (1956), Vol. 35, No.6, 1417-1444.

[29] Caldwell, S. H. "The recognition and identification of symmetric switching functions." American Institute of Electrical Engineers, Part I: Transactions of the Communication and Electronics, (1954), Vol.73 No.2, 142-147.

[30] McCluskey, E. J. "Detection of Group Invariance or Total Symmetry of a Boolean Function." Bell System Technical Journal, (1956), Vol.35. No.6, 1445-1453.

[31] Franz H,. "Some mathematical aspects of switching." American Mathematical Monthly, (1955), 75-90.

[32] Ryszard S Michalski. "Recognition of total or partial symmetry in a completely or incompletely specified switching function," Proceedings of the IV Congress of

the International Federation on Automatic Control (IFAC), June 16-21, (1969),Vol. 27, 109-129.

[33] S.R. Das, "On detecting total and partial symmetry of switching functions," Proceedings of the IEEE, May (1970),Vol.58, No.5, 840-841. doi: 10.1109/PROC.1970.7777.

[34] Raymond J. Nelson, "Simplest normal truth functions," The Journal of Symbolic Logic, (1955),Vol. 20, 105-108. doi:10.2307/2266893.

[35] J.R. Slagle, Chin-Liang Chang; R.C.T Lee, "A New Algorithm for Generating Prime Implicants," IEEE Transactions on Computers, April (1970), Vol.C-19, No.4, 304-310. doi: 10.1109/T-C.1970.222917.

[36] S.R., Das, "Comments on " A New Algorithm for Generating Prime Implicants",", IEEE Transactions on Computers, Dec. (1971), vol.C-20, no.12, 1614-1615,. doi: 10.1109/T-C.1971.223185.

[37] J. P. Roth, "Algebraic topological methods for the synthesis of switching systems. I." Transactions of the American Mathematical Society, (1958), 301-326.

[38] S.R. Das, N.S. Khabra, "Clause-Column Table Approach for Generating All the Prime Implicants of Switching Functions," IEEE Transactions on Computers, Nov. (1972), Vol.C-21, No.11, 1239-1246. doi: 10.1109/T-C.1972.223484.

[39] D. C Schmidt, L. E. Druffel, " An extension of the clause table approach to multi-output combinational switching networks" IEEE Transactions on Computers, (1974), Vol. 23, No. 4, 338-346.

[40] Randal E Bryant, "Graph-based algorithms for Boolean function manipulation.", IEEE Transactions on Computers, (1986), Vol.100, no. 8, 677-691.