International Journal of Mathematics and Physical Sciences Research ISSN 2348-5736 (Online)
Vol. 6, Issue 1, pp: (72-75), Month: April - September 2018, Available at: www.researchpublish.com

Review on Boolean Algebra in Mathematics and Computing

[1]Ramanpreet Kaur, [2]Gagandeep

Punjabi University, Patiala
[1,2]Baba Farid Group of Institutions, Bathinda, India

Abstract: **Boolean algebra, symbolic system of mathematical logic that represents relationships between entities—either ideas or objects. The basic rules of this system were formulated in 1847 by George Boole of England and were subsequently refined by other mathematicians and applied to set theory. Today, Boolean algebra is of significance to the theory of probability, geometry of sets, and information theory. Furthermore, it constitutes the basis for the design of circuits used in electronic digital computers.**

Keywords: **EX-OR, EX-NOT, VLSI, CMOS, CPU.**

I. INTRODUCTION

The Greek philosopher Aristotle founded a system of logic based on only two types of propositions: true and false. The English mathematician George Boole (1815-1864) sought to give symbolic form to Aristotle's system of logic. Boole wrote a treatise on the subject in 1854, titled An Investigation of the Laws of Thought, on Which Are Founded the Mathematical Theories of Logic and Probabilities, which codified several rules of relationship between mathematical quantities limited to one of two possible values: true or false, 1 or 0. His mathematical system became known as Boolean algebra.

A set of rules formulated by the English mathematician George Boole describe certain propositions whose outcome would be either true or false. With regard to digital logic, these rules are used to describe circuits whose state can be either, 1 (true) or 0 (false). In order to fully understand this, the relation between the AND gate, OR gate and NOT gate operations should be appreciated**.**

Boolean Algebra is used to analyze and simplify the digital (logic) circuits. It uses only the binary numbers i.e. 0 and 1. It is also called as **Binary Algebra** or **logical Algebra**.

Example 1. The power set 2^X of X, consisting of all subsets of X. Here X may be any set: empty, finite, infinite, or even uncountable.

Example 2. The empty set and X. This two-element algebra shows that a concrete Boolean algebra can be finite even when it consists of subsets of an infinite set. It can be seen that every field of subsets of X must contain the empty set and X. Hence no smaller example is possible, other than the degenerate algebra obtained by taking X to be empty so as to make the empty set and X coincide

II. RULE IN BOOLEAN ALGEBRA

Following are the important rules used in Boolean algebra.

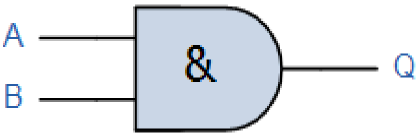
- Variable used can have only two values. Binary 1 for HIGH and Binary 0 for LOW.
- Complement of a variable is represented by an overbar (-). Thus, complement of variable B is represented as $\overline{B}$. Thus if B = 0 then $\overline{B}$ = 1 and B = 1 then $\overline{B}$ = 0.

- ORing of the variables is represented by a plus (+) sign between them. For example ORing of A, B, C is represented as A + B + C.
- Logical ANDing of the two or more variable is represented by writing a dot between them such as A.B.C. Sometime the dot may be omitted like ABC.
- **Boolean Operators** are used to connect and define the relationship between your search terms. When searching electronic databases, you can use Boolean operators to either narrow or broaden your record sets. The three Boolean operators are **AND**, **OR** and **NOT**.

2-input AND Gate:

For a 2-input AND gate, the output Q is true if BOTH input A “AND” input B are both true, giving the Boolean Expression of: (Q = A and B).

Symbol- .



2-input AND Gate

Truth Table

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

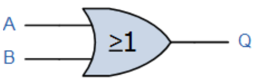
Boolean Expression Q = A.B
Note that the Boolean Expression for a two input AND gate can be written as: A.B or just simply AB without the decimal point.

Read as A AND B gives Q

2-input OR (Inclusive OR) GateE

For a 2-input OR gate, the output Q is true if EITHER input A “OR” input B is true, giving the Boolean Expression of: (Q = A or B).

Symbol - +



2-input OR Gate

Truth Table

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

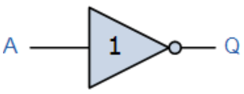
Boolean Expression Q = A+B

Read as A OR B gives Q

NOT Gate

For a single input NOT gate, the output Q is ONLY true when the input is “NOT” true, the output is the inverse or complement of the input giving the Boolean Expression of: (Q = NOT A).

Symbol-



Inverter or NOT Gate

Truth Table

A	Q
0	1
1	0

Boolean Expression Q = NOT A or A

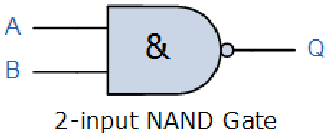
Read as inversion of A gives Q

The NAND and the NOR Gates are a combination of the AND and OR Gates with that of a NOT Gate or inverter.

2-input NAND (Not AND) Gate

For a 2-input NAND gate, the output Q is true if BOTH input A and input B are NOT true, giving the Boolean Expression of: (Q = not(A and B)).

Symbol- **(.)**'



Truth Table

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

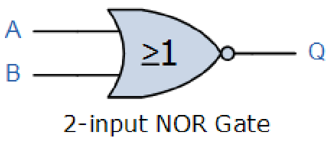
Boolean Expression $Q = (A.B)'$

Read as A AND B gives NOT-Q

2-input NOR (Not OR) Gate

For a 2-input NOR gate, the output Q is true if BOTH input A and input B are NOT true, giving the Boolean Expression of: (Q = not(A or B)).

Symbol – **(+)**'



Truth Table

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

Boolean Expression $Q = (A+B)'$

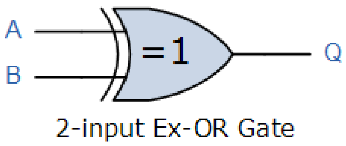
Read as A OR B gives NOT-Q

As well as the standard logic gates there are also two special types of logic gate function called an Exclusive-OR Gate and an Exclusive-NOR Gate. The actions of both of these types of gates can be made using the above standard gates however, as they are widely used functions, they are now available in standard IC form and have been included here as reference.

2-input EX-OR (Exclusive OR) Gate

For a 2-input Ex-OR gate, the output Q is true if EITHER input A or if input B is true, but NOT both giving the Boolean Expression of: (Q = (A and NOT B) or (NOT A and B)).

Symbol – ($\oplus$)



Truth Table

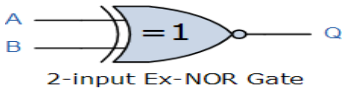
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Boolean Expression $Q = A'B + AB' = (A \oplus B)$

2-input EX-NOR (Exclusive NOR) Gate

For a 2-input Ex-NOR gate, the output Q is true if BOTH input A and input B are the same, either true or false, giving the Boolean Expression of: (Q = (A and B) or (NOT A and NOT B)).

Symbol - $\odot$



Truth Table

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

Boolean Expression $Q = AB + A'B' = (A \odot B)$

Boolean algebra uses for some definitions. If real life means in the form of practical objects, and used means doing functionally useful things - Boolean Algebra is the underlying mathematical structure of digital logic which is most often implemented via VLSI of CMOS circuits in the form of CPU, memory, and other digital technologies which are used to do things like run your laptop, or route packets over networks and so on.Perhaps you meant in the ordinary life of people thinking without the aid of computing devices - well, most 2-state logic problems can be attacked with boolean algebra. Say you knew if the ground is wet then either the sprinkler.

Over time, mathematicians have developed sophisticated mathematical techniques for analyzing very complex logical statements.

III. CONCLUSION

Binary mathematics is the number system most often used with computers. Computer systems consist of magnetic cores that can be switched on or switched off. The numbers 0 and 1 are used to represent the two possible states of a magnetic core. Boolean statements can be represented, then, by the numbers 0 and 1 and also by electrical systems that are either on or off. As a result, when engineers design circuitry for personal computers, pocket calculators, compact disc players, cellular telephones, and a host of other electronic products, they apply the principles of Boolean algebra.

REFERENCES

[1] Girard, Jean-Yves; Paul Taylor; (1990) [1989]. Proofs and Types. Cambridge Press (Cambridge Tracts in Theoretical Computer Science, 7). ISBN 0-521-37181-3..

[2] Steven R. Givant; Paul Richard Halmos (2009). Introduction to Boolean algebras. Springer. pp. 21–22. ISBN 978-0-387-40293-2

[3] Koppelberg, Sabine (1989). "General Theory of Boolean Algebras". Handbook of Boolean Algebras, Vol. 1 (ed. J. Donald Monk with Robert Bonnet). Amsterdam: North Holland. ISBN 978-0-444-70261-6.

[4] Rajaraman & Radhakrishnan. Introduction To Digital Computer Design. PHI Learning Pvt. Ltd. p. 65. ISBN 978-81-203-3409-0

[5] The name Boolean algebra (or Boolean 'algebras') for the calculus originated by Boole, extended by Schröder, and perfected by Whitehead seems to have been first suggested by Sheffer, in 1913." E. V. Huntington, "New sets of independent postulates for the algebra of logic, with special reference to Whitehead and Russell's Principia mathematica", in Trans. Amer. Math. Soc. **35** (1933), 274-304; footnote, page 278

[6] O'Regan, Gerard (2008). A brief history of computing. Springer. p. 33. ISBN 978-1-84800-083-4.

[7] Ben Goertzel (1994). Chaotic logic: language, thought, and reality from the perspective of complex systems science. Springer. p. 48. ISBN 978-0-306-44690-0.