AMERICAN Journal of Engineering, Mechanics and Architecture
*Volume 4, Issue 03, 2026 ISSN (E): 2993-2637*

# Review on Consolidate of Combinational Logic Circuit

**Sumangla Pal**
*Assistant Professor, Sunrise Institute of Engineering Technology & Management*

**Abstract:** This paper's learning objectives are to illustrate the chronical of a logic switching circuit. The study focuses on the evolution and use of logic throughout history in the field of computer and electronic applications. This essay focuses on design considerations, a few interesting facts, and a brief analysis of the fundamental requirements of logic fusion. Additionally, it improves students' comprehension of various Boolean function minimization techniques.

**Keywords:** Combinational circuits, digital electronics, switching circuit simplification, multiplexers, decoders, maxterm, minimization, and numerous inputs.

## Introduction

Combinational circuits are made up of logic gates whose output is always dependent on the combination of logic levels at the input. Since a combinational circuit lacks memory, its output is solely dependent on the inputs' current values. Combinational circuits rely on Boolean expressions to implement their function, and the digital gate is its fundamental component. N inputs and M outputs make up the circuit. There are 2N potential combinations of binary input values for N inputs. There is only one output combination for any combination of inputs. The Boolean expression M can be used to describe this combinational circuit. One Boolean statement for each output can be used to describe this combinational circuit. N inputs are used to express every output.

On the contrary. Logic gates and memory components like flip-flops are components of sequential logic circuits.

A combinational circuit with n inputs and m outputs is depicted in Fig. 1 There are 2N potential permutations of bits at the input, as shown by the n number of inputs. Consequently, m Boolean statements are used to express the output.

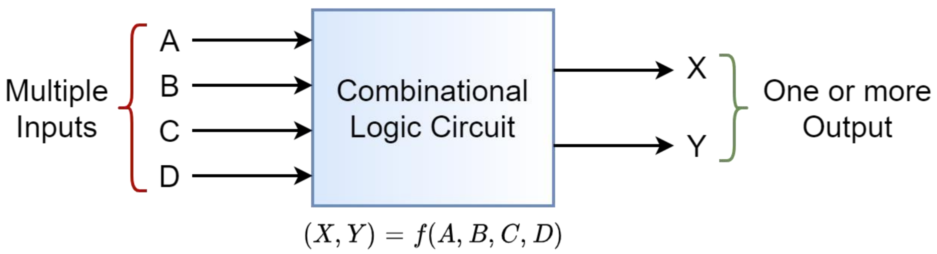


**Fig.1 Circuit Diagram of combinational circuit**

## Background

One well-known part of digital electronics is the combinational circuit. They are specifically made to work with several coupled logic gates.

**Classification of combinational circuit**

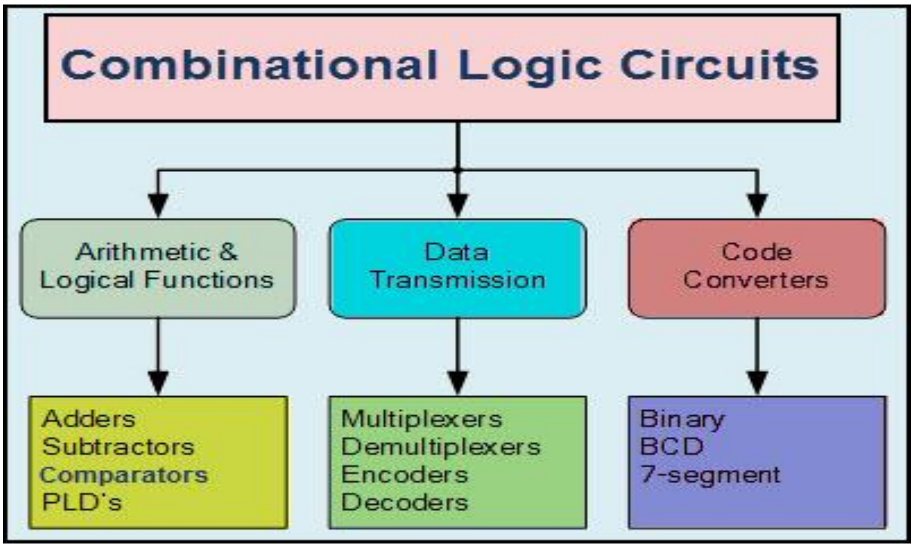


**Fig.2 Block diagram**

**Methodology**

Mathematical representation of the function, involves a systematic process, problem definition, identifying inputs/outputs, creating a truth table mapping all input combinations to desired outputs: deriving and simplifying Boolean expressions(using K-map or algebra) and finally implementing the logic diagram with AND,OR,NOT gates to build the circuit, logic implementation ,switching simplification. Problem specification, truth table construction, Boolean function derivation, circuit implementation, optimization technique.

Types of logic gate

1. Basic gate
2. Universal gate
3. Special gate

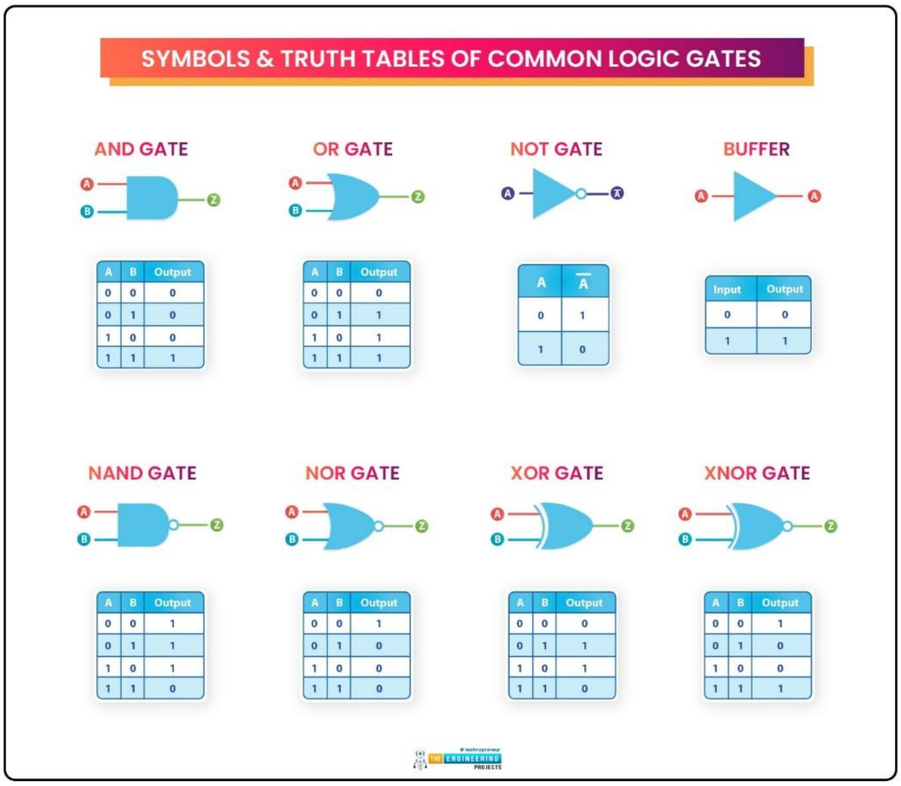


**Fig.3 Common Logic Gates: Symbols and Truth Table**

## Adder (Digital Adder)

An **adder** is a fundamental combinational logic circuit in digital electronics used to perform **binary addition**. Adders form the core of arithmetic operations in digital systems such as **computers, calculators, microprocessors, and ALUs**.

## Half Adder

Three single-bit binary inputs are added by a complete adder, a combinational logic circuit: two significant bits (A and B) and a carry-in (Cin) from the preceding stage. Sum (S) and Carry-out (Cout) are the two outputs it generates.

- **Sum (S)**
- **Carry-out (Cout)**

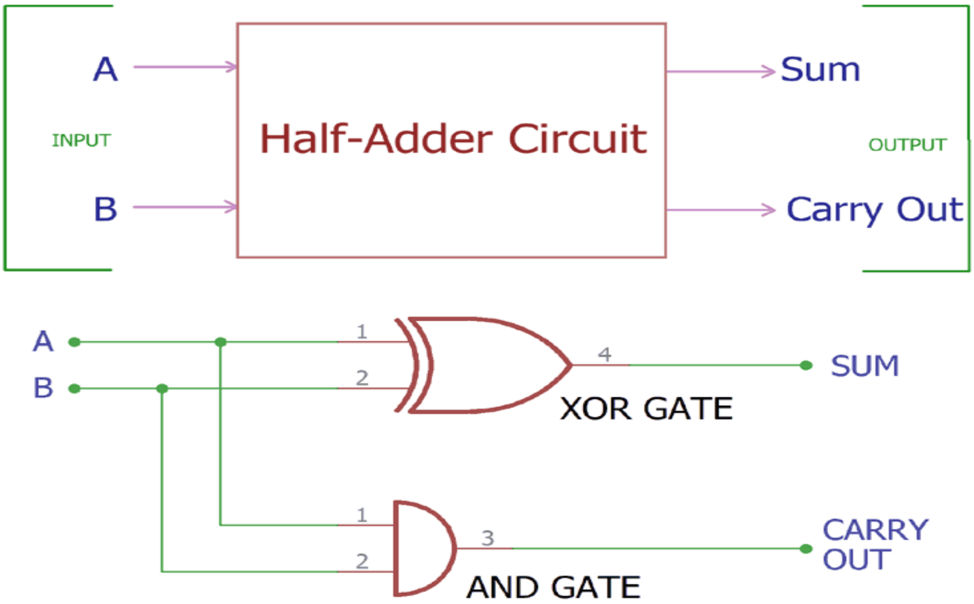


**Fig.4** Diagram of the Half Adder Circuit

Half Adder

| A | B | C Out | Sum |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

**Table.1 Truth Table of Half Adder**

## Full Adder

A simple combinational logic circuit for adding two single-bit binary integers is called a half adder. It has two outputs, Sum (S) and Carry (C), and two inputs, A and B.

- **Sum (S)**
- **Carry (C)**

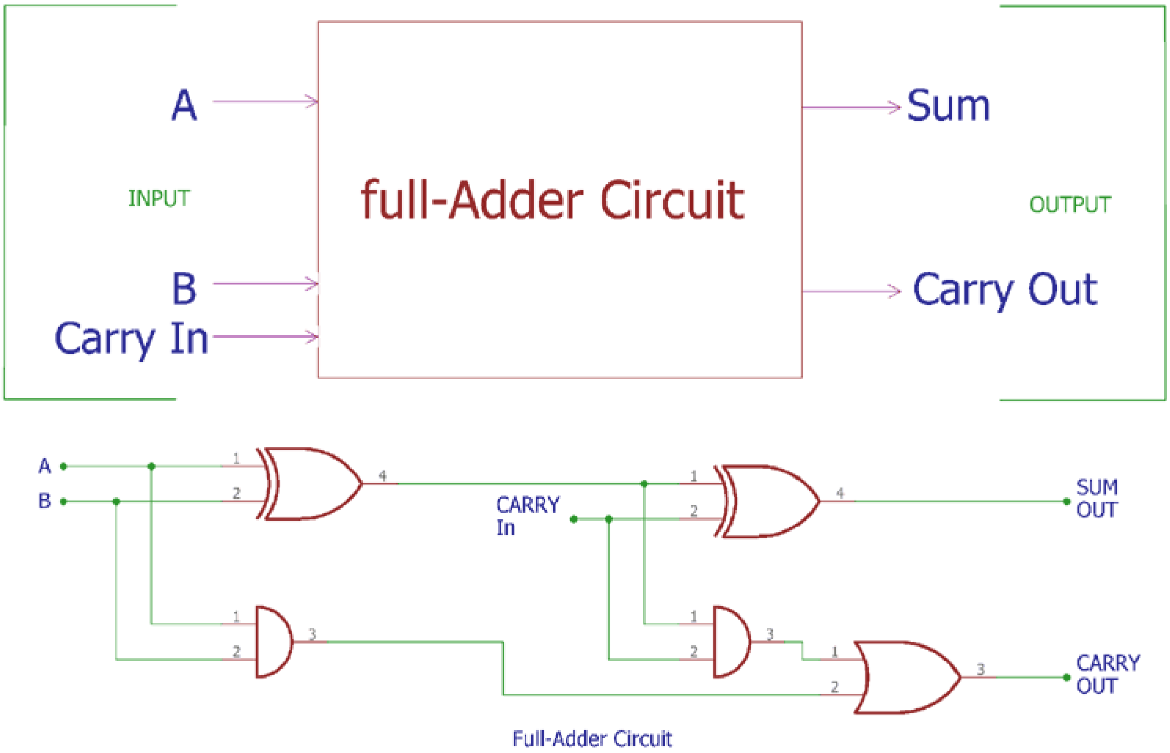


**Fig.4 Circuit Diagram of Full Adder Circuit**

| Input | | | Output | |
|---|---|---|---|---|
| A | B | Cin | Sum | Carry |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**Table2. Truth Table for Full Adder**

### Half Subtractor

One single-bit binary number can be subtracted from another using a simple combinational logic device called a half subtractor. Difference (D) and Borrow (Bo) are its two outputs, and Minuend (A) and Subtrahend (B) are its two inputs.

- **Difference (D)**
- **Borrow (Bo)**

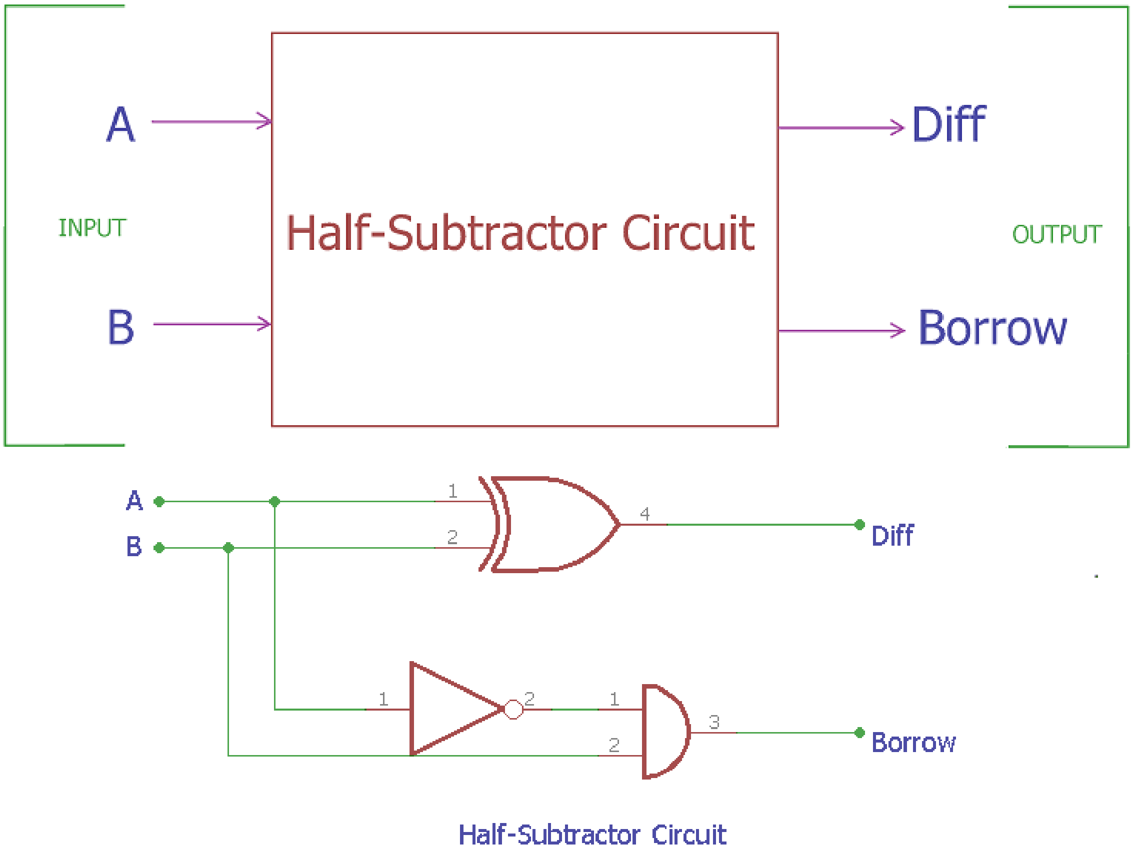


**Fig.4 Circuit Diagram of Half Subtractor Circuit**

| Inputs | | Outputs | |
|---|---|---|---|
| A | B | Difference | Borrow |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

**Table3. Truth Table for Half Subtractor**

**Full subtractor**

A combinational logic circuit called a full subtractor is used to subtract two single-bit binary integers while taking into account a borrow input from a lower-order stage. It has two outputs, Difference (D) and Borrow-out (Bout), and three inputs, Minuend (A), Subtrahend (B), and Borrow-in (Bin).

- **Difference (D)**
- **Borrow-out (Bout)**

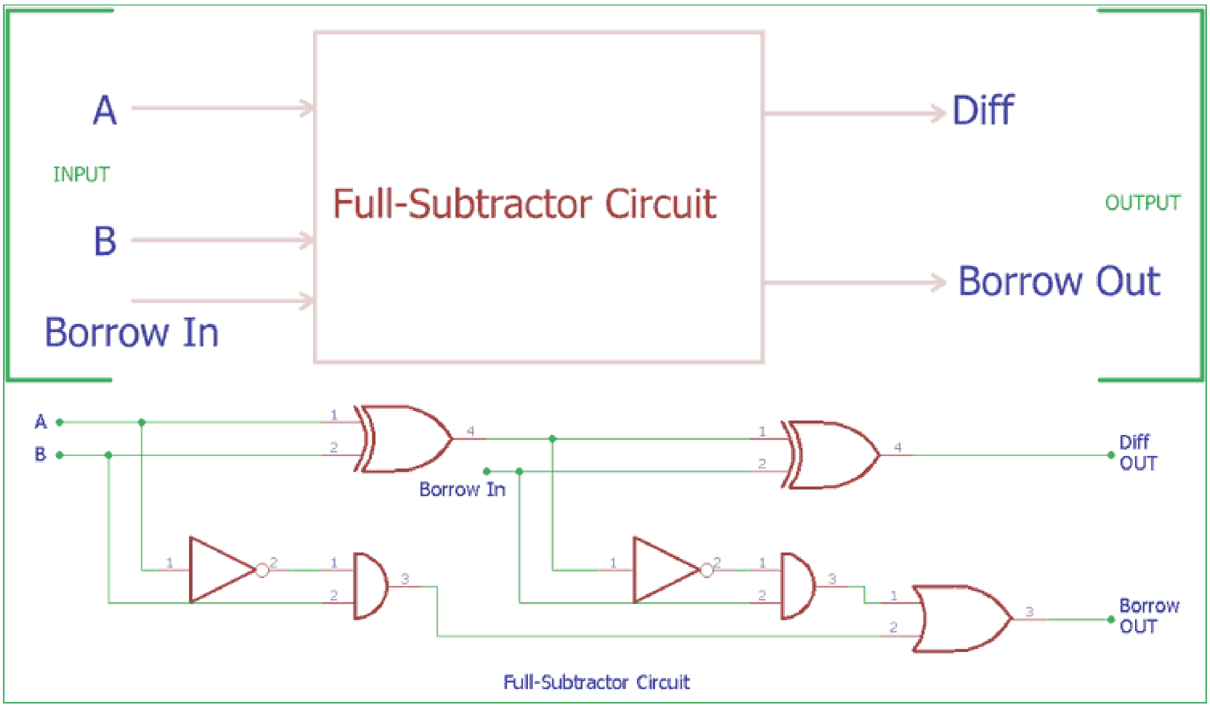


**Fig.6 Diagram of the Complete Subtractor Circuit**

| Inputs | | | Outputs | |
|---|---|---|---|---|
| A | B | bin | d (Difference) | b (Borrow) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**Table.4 Truth table for full subtractor**

**Encoder**

A combinational logic circuit known as an encoder transforms several input lines into fewer output lines, typically in binary form. It carries out a decoder's opposite operation. Typically, an encoder has n output lines and two input lines. The output is the binary code that corresponds to the active input, and it is assumed that only one input is active at a time.

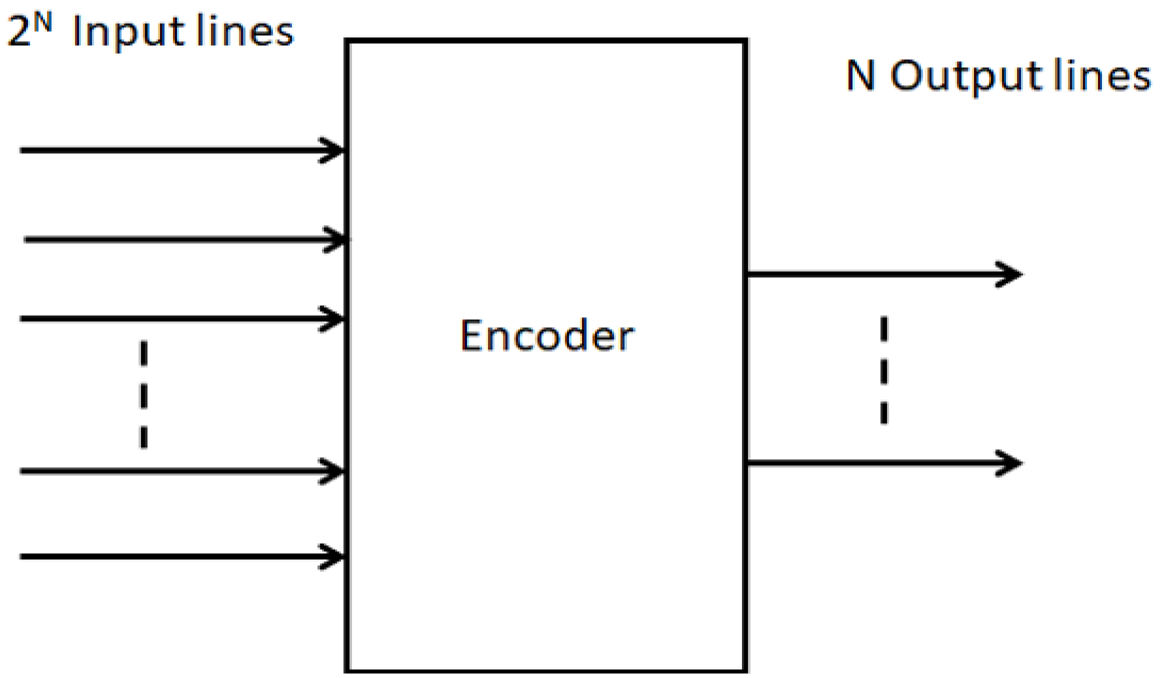


**Fig.7 Circuit Diagram for the Encoder**

**Decoder**

A combinational logic circuit that transforms binary input data into a one-of-many output format is called a decoder. It operates like an encoder in reverse.

Typically, a decoder has two output lines and n input lines. Only one output line is turned on at a time for a particular binary input combination.

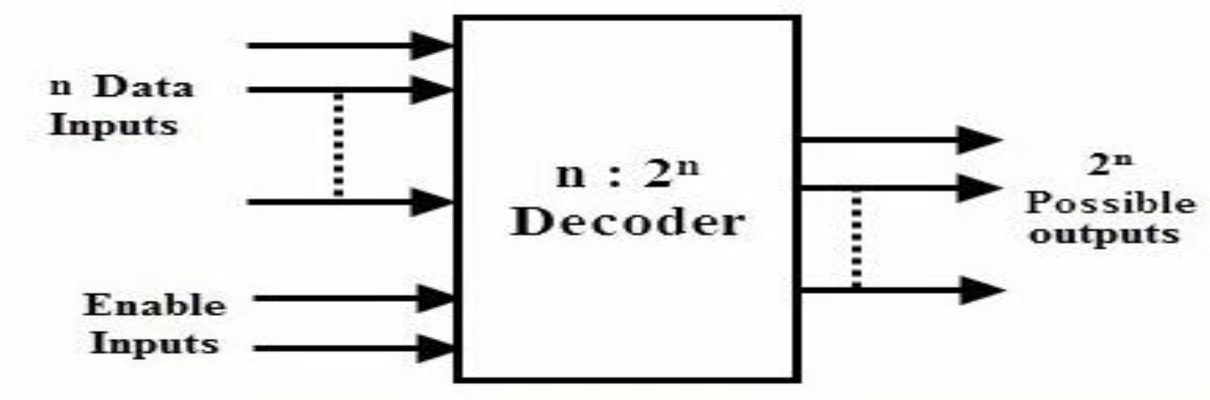


**Fig.8 The Decoder Circuit Circuit Diagram**

**Multiplexer (MUX)**

A multiplexer is a combinational logic circuit that, depending on the values of the selected inputs, chooses one input from several input lines and sends it to a single output line. It serves as a selector of data.

One output and n select lines are needed for a multiplexer with two input lines.

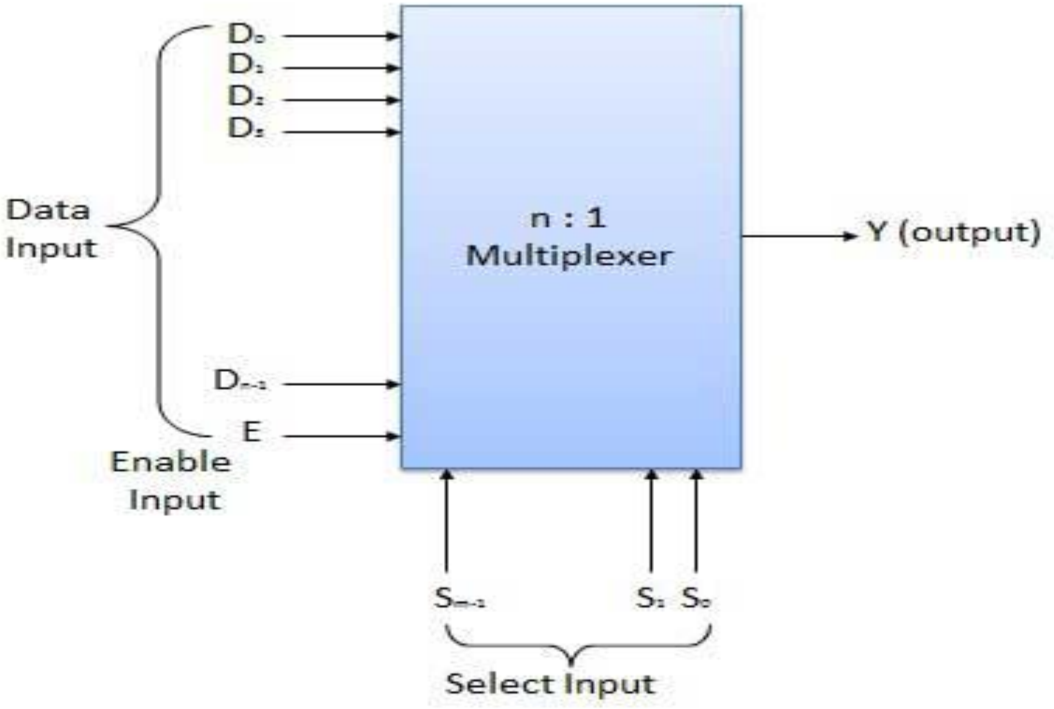


**Fig.9 Multiplexer Circuit Circuit Schematic**

**Demultiplexer (DEMUX)**

A demultiplexer is a combinational logic circuit that, depending on the values of specific inputs, divides a single input signal among several output lines. It does what a multiplexer does in reverse.

A demultiplexer with n select lines has one data input and two output lines.

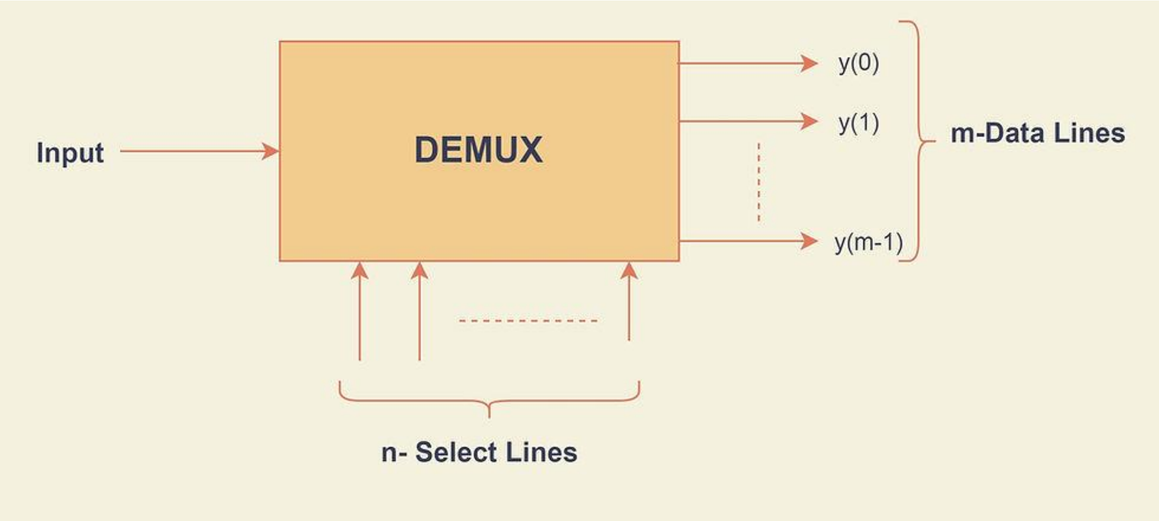


**Fig.10 Circuit Diagram of Demultiplexer Circuit**

**Recent advancements**

Reversible Logic, low-power combinational Approximate Computing, Nano technology logic circuits

**Research and findings**

This study is an overview of logic synthesis. Different representation techniques are needed for different minimization strategies and switching circuit implementation. Several minimization procedures and implementation strategies were discussed in this study. Logic minimization is an old topic, yet it is still relevant today. It is crucial for many aspects of VLSI synthesis, switching system design, built-in self-test, and many other applications.

**Future scope**

1. Higher label abstraction, automation, quantum computing, the Internet of Things, and integration with AI and machine learning


2. James A. Ryan. 1996. Leibniz' Binary System and Shao Yong's "Yijing". Philosophy East and West, Vol. 46, No. 1, 59-90.
3. Aschoff, V., "The early history of the binary code," in IEEE Communications Magazine, vol.21, no.1, (Jan. 1983), 4-10.
4. 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.
5. Boole, George, "The mathematical analysis of logic," in Philosophical Library, (1847). [
6. Marquand, Allan, "A new logical machine," in Proceedings of the American Academy of Arts and Sciences, John Wilson and Son (1885), 303-307.
7. 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.
8. Computer Museum. Computer museum of America. Available at http://history compuer.com/Modern Computer/thinkers/Stanhope.html last seen July (2015).
9. Peirce, Charles Sanders. "Writings of Charles S. Peirce: 1857-1866" (1982) vol. 1. Indiana University Press.
10. Computer Museum. Computer museum of America. http://history computer.com/ModernComputer/thinkers/Peirce.html last seen (July 2015).
11. 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.

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

18. 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.

19. 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.

20. T. Sasao, “Multi-Level Logic Synthesis, “Switching Theory for Logic Synthesis, Springer US, (1999), 229 261.

21. Jim Stiles, “Power Dissipation,” The Univ. of Kansashttp://www.ittc.ku.edu/~jstiles/312/handouts/Pow er%20Dissipation.pdf.2014.

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

23. 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 http://www.ddp116.org/products/pacs/logic_1955.pdf at

24. 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.

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

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

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

28. 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.