



# Working Of Logic Gates In Digital Computers

<sup>1</sup>Mr.Durvesh Kumar, <sup>2</sup>Mr. Himanshu Sen, <sup>3</sup>Mr. Praveen Dahaliya

<sup>1</sup>Assistant Professor, <sup>2</sup>Assistant Professor, <sup>3</sup>Assistant Professor

<sup>1</sup>Department of Computer Applications,

<sup>1</sup>Accurate Institute Of Management and Technology, Greater Noida, U.P., India

**Abstract:** Logic gates are the basic building blocks of digital computers. They perform simple logical operations using binary values, 0 and 1. This research paper explains the working of common logic gates such as AND, OR, NOT, NAND, NOR, XOR, and XNOR, and their role in digital computer systems. The paper discusses how logic gates use Boolean logic to process input signals and produce output results. It also describes how combinations of logic gates are used to perform important functions inside a computer, including arithmetic operations, data comparison, and decision-making. Through truth tables and simple circuit diagrams, this study helps in understanding how logic gates form the foundation of digital electronics and computer hardware. The objective of this paper is to provide a basic and easy explanation of logic gates and their importance in digital computers.

The paper also describes how logic gates are used in real-life applications like burglar alarms and security systems. By studying different physical systems such as mechanical, electrical, optical, thermal, and natural systems, it is possible to represent them using Boolean expressions. This makes it easier to study and analyze these systems using mathematical methods.

**Index Terms** - AND, OR, NOT, NAND, NOR, XOR, XNOR Gates.

## I. Introduction

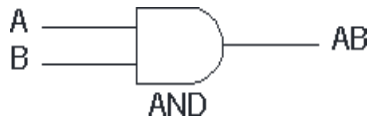
Logic gates are the basic units used to perform logical operations in digital systems. They work by following simple logical rules and produce outputs in binary form, that is 0 or 1, based on the given inputs. Logic gates play an important role in digital electronics and computer systems. These gates can be implemented using electronic, mechanical, optical, or biological methods. Most logic gates have one output, while the number of inputs may differ. The NOT gate is special because it has only one input and one output. Logic gates are the building blocks of digital circuits and help in designing complex digital systems. They receive one or more input signals and produce an output signal that is either 0 or 1, representing “off” or “on.” Different types of logic gates perform different basic logical functions. For example, a NOT gate has only one input and one output; it simply reverses the input so that an “on” input becomes “off,” and vice versa.

Logic gates are usually made from electronic components such as diodes and transistors, which act as tiny switches. When many logic gates are combined, they can perform very complex operations. A truth table is commonly used to describe the behavior of a logic gate by showing the output for every possible combination of inputs.

**II. Types of logic gates:**

**A. AND GATE:** An AND gate is a basic digital device that only gives an output of 1 (ON) when all of its inputs are 1. If even one input is 0 (OFF), the output will be 0. It is often used in digital circuits to show a logical “and” condition.

Where  $A \wedge B = \{ 1 \text{ if } A=1 \text{ and } B =1 \text{ (otherwise } 0)$



TRUTH

2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

TABLE

**B. OR GATE:** An OR gate is a digital logic gate that has two inputs, usually labeled A and B. The output of the gate is written as  $A + B$ . The OR gate produces an output of 1 if either A or B is 1, or if both are 1. The output is 0 only when both A and B are 0.

In symbolic form:  $A \vee B = \{ 1 \text{ if } A=1 \text{ or } B=1 \text{ (otherwise } 0)$

TRUTH TABLE

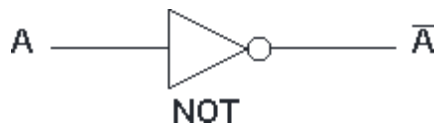


2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

**C. NOT GATE:** A NOT gate, also called an inverter, is a basic digital device that has only one input, called A. This input can be either 0 or 1. The output is written as  $\bar{A}$ .

If  $A = 1$ , then output is 0  
 If  $A = 0$ , then output is 1

TRUTH TABLE



NOT gate	
A	$\bar{A}$
0	1
1	0

**D. NAND Gate:** A NAND gate works like an AND gate followed by a NOT gate. This means the output will be 0 only when all the inputs are 1. If any input is 0, the output will be 1. The symbol looks like an AND gate with a small circle at the output. The small circle shows inversion (NOT action).

TRUTH TABLE



2 Input NAND gate		
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

**E. NOR Gate:** A NOR gate works like an OR gate followed by a NOT gate. This means the output will be 1 only when all the inputs are 0. If any input is 1, the output will be 0. The symbol looks like an OR gate with a small circle at the output. The small circle shows inversion.

TRUTH TABLE



2 Input NOR gate		
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

**F. Exclusive-OR (XOR) Gate:** An XOR gate gives an output of 1 only when the two inputs are different. If both inputs are the same (both 0 or both 1), the output will be 0. It is usually shown with an OR gate symbol and an extra curved line, and sometimes a  $\oplus$  sign is used.

TRUTH TABLE



2 Input EXOR gate		
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

**G. Exclusive-NOR (XNOR) Gate:** An XNOR gate does the opposite of the XOR gate. It gives an output of 1 when both inputs are the same (both 0 or both 1). If the inputs are different, the output is 0. The symbol looks like an XOR gate with a small circle at the output.

TRUTH TABLE



2 Input EXNOR gate		
A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

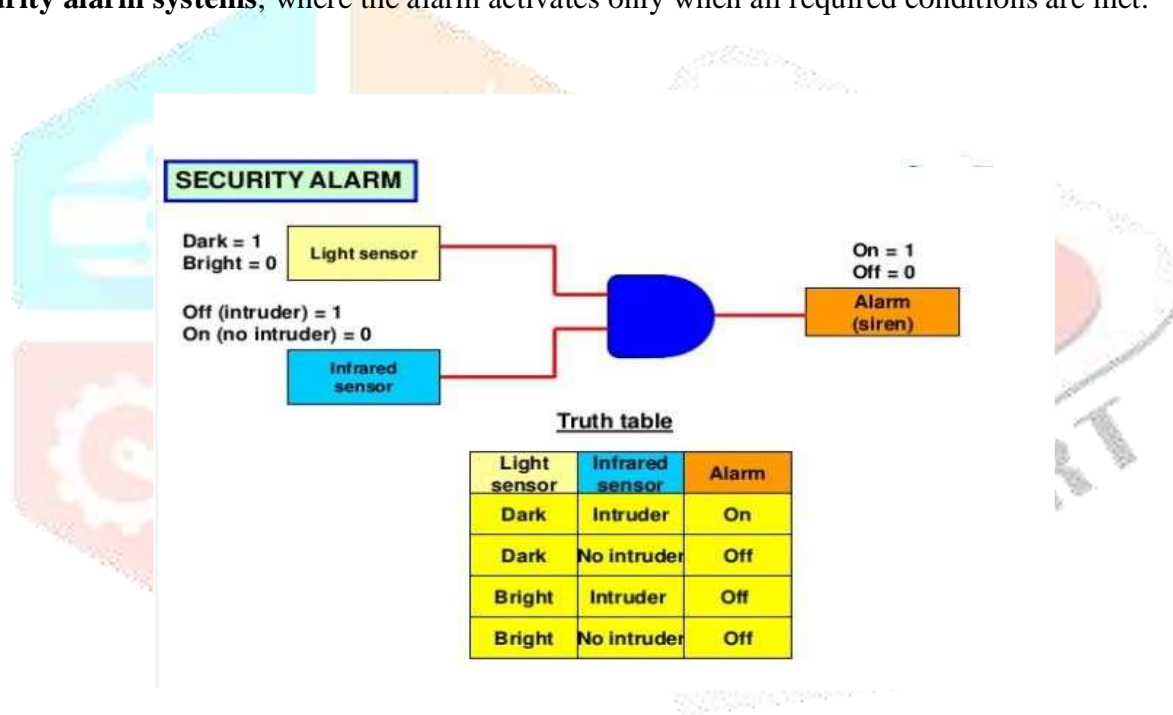
**III. APPLICATIONS OF LOGIC GATES:** Logic gates are mainly used according to their truth tables, which define how they operate. Basic logic gates are found in many everyday electronic circuits such as push-button locks, light-activated burglar alarms, safety thermostats, and automatic watering systems. Digital

communication would not be possible without logic operations. Practical examples of XOR (exclusive OR) and XNOR (exclusive NOR) gates can be seen in ladder-type light switches used in homes and in water-level detection circuits.

Logic gates have a wide range of applications in both everyday devices and complex digital systems. They form the basic building blocks of digital electronics, and their operation is determined by the truth table of each gate. Logic gates are used in security systems such as push-button locks and light-activated burglar alarms, as well as in control devices like safety thermostats and automatic watering systems. They are essential for digital communication, data processing, and decision-making circuits in computers and embedded systems. Overall, logic gates enable systems to perform logical decisions automatically, making modern electronic technology possible.

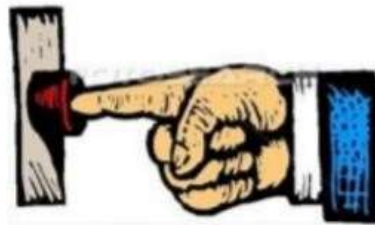
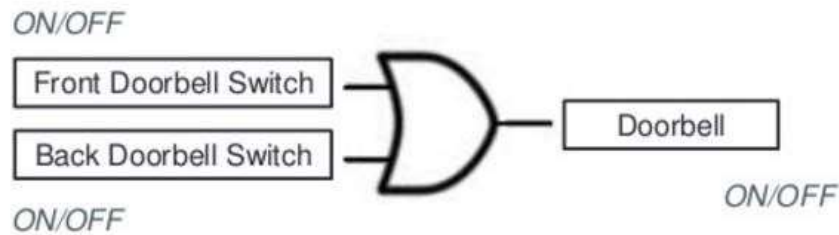
**(i). AND GATE: - AND gates are used to combine two or more input signals. The output will be TRUE (1) only when every input is TRUE. If even one input is FALSE (0), then the output becomes FALSE.**

In practice, AND gates are not used as often as NAND gates, because **NAND gates require fewer components and can also act as NOT gates when needed.** Still, AND gates are useful in many circuits, such as **security alarm systems**, where the alarm activates only when all required conditions are met.



**(ii) OR GATE:-**

**OR gates are used in situations where one or more events need to be detected.** If any of the inputs becomes active, the OR gate produces an output, which can then be used to trigger an action. So, whenever a circuit needs to respond when **any one of several conditions occurs**, an OR gate is the right choice.

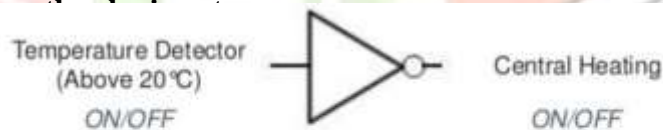


If either the Front Doorbell Switch OR the Back Doorbell Switch is pressed then the Doorbell rings.

**(iii) NOT GATE:-**

**NOT gates are called inverters because they always produce the opposite of the input signal.** If the input is 1, the output becomes 0, and if the input is 0, the output becomes 1.

Today, CMOS inverters are often used to create **square-wave oscillators**, which generate clock signals in digital circuits. They are popular because **they use very little power and are easier to interface with other circuits compared to mar**

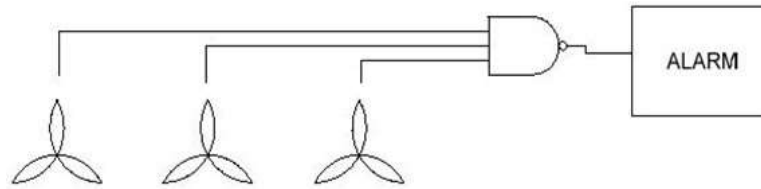


If the temperature is above 20°C then the Central Heating is switched off.

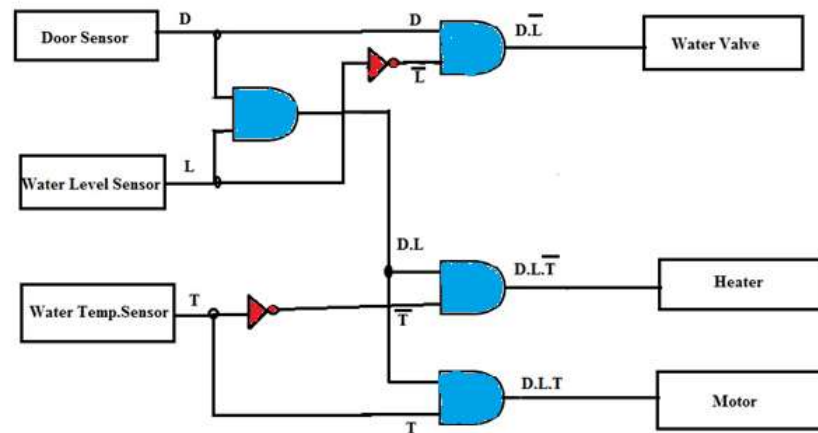
If the temperature is below 20°C then the Central Heating is switched on.

**(iv) NAND GATE:-**

**NAND gates are used in many real-world electronic circuits.** They are commonly found in devices such as alarm systems, push-button locking systems, and automatic street-light controllers. These are just a few examples — NAND gates are extremely important in digital electronics because they are flexible, efficient, and can be used to create many different types of circuits in modern technology.

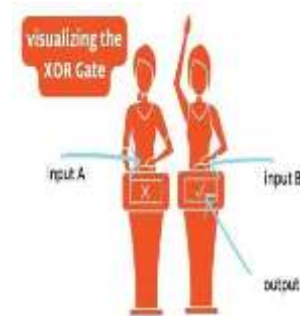
**(v) NOR GATE****Washing Machine Controller**

A washing machine is equipped with three sensors to monitor its operation: one detects if the lid is open, another checks whether the tub has reached the minimum water level, and the third measures the combined weight of the clothes and water in the tub. If any of these conditions are unsafe—lid open, water below minimum, or overloaded—the corresponding sensor outputs a 1. These signals are fed into a NOR gate, which then produces an output of 0. This 0 signal automatically turns the washing machine off, ensuring safety and preventing damage.



**(vi) EX- ORGATE**

An everyday example of an XOR gate can be seen in a game show buzzer system. When two contestants have buzzers, the system is designed so that only one person can buzz in successfully. If exactly one contestant presses their buzzer, the system accepts that response and locks out the other player. But if both press at the same time, neither is accepted. This works like an XOR gate, which gives an output of ‘1’ only when one —



and only one — of the inputs is ‘1’.

An XOR gate is a digital logic circuit that has two inputs and one output. It produces a HIGH (1) output only when exactly one of the inputs is HIGH. If both inputs are LOW, or both are HIGH at the same time, the output will be LOW (0).

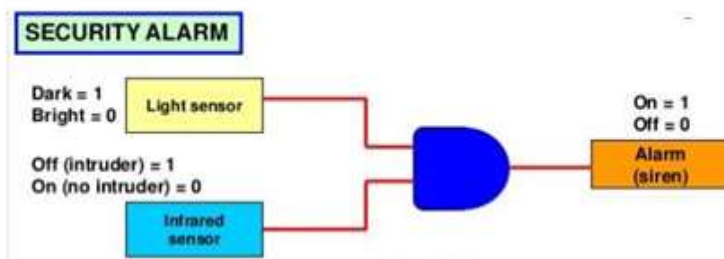
**(vii) EX-NORGATE**

The Exclusive NOR (EX-NOR or XNOR) gate is a digital logic gate whose output becomes HIGH (1) when there is an even number of HIGH inputs. In other words, the output is TRUE when both inputs are the same — either both 0 or both 1. A practical example of an XNOR gate can be found in some home security systems, where the alarm is triggered if the expected and actual signals don't match.

**XNOR GATE USING DIODES**

Inputs		Outputs
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

Light Sensor	Infrared Sensor	Alarm
Intruder	Dark	On
No Intruder	Dark	Off
Bright	Intruder	Off
Bright	No Intruder	On



## REFERENCES:-

- [1]. M. Morris Mano & Michael D. Ciletti, Digital Design, 5th Edition, Pearson Education, 2013.
- [2]. R.P. Jain, Modern Digital Electronics, 4th Edition, McGraw-Hill, 2010.
- [3]. Thomas L. Floyd, Digital Fundamentals, 11th Edition, Pearson, 2015.
- [4]. Charles H. Roth & Larry L. Kinney, Fundamentals of Logic Design, Cengage Learning, 2010.
- [5]. Neil H. E. Weste & David Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4th Edition, Addison-Wesley, 2010.
- [6]. S. Brown & Z. Vranesic, Fundamentals of Digital Logic with Verilog Design, McGraw-Hill, 2009.
- [7]. A. S. Sedra & K. C. Smith, Microelectronic Circuits, 6th Edition, Oxford University Press, 2010.
- [8]. Tocci, Widmer & Moss, Digital Systems: Principles and Applications, 10th Edition, Pearson, 2007.
- [9]. <https://brilliant.org/wiki/logic-gates/>
- [10]. <http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc>
- [11]. <https://www.electrical4u.com/some-common-applications-of-logic-gates/>
- [12]. <https://electronicsclub.info/gates.html>
- [13]. D.P.Vasudevan, P.K.Lala, J.Di and J.P.Parkerson, — Reversiblelogic design with online testability, IEEE Trans. on Instrumentation and Measurement, vol.55., no.2, pp.406- 414, April 2006
- [14]. Peres, —Reversible Logic and Quantum Computers, Physical review A, 32:3266- 3276, 1985
- [15]. Fundamentals of Digital Logic with Verilog Design — \*Stephen Brown & Zvonko Vranesic
- [16]. Logic and Computer Design Fundamentals — M. Morris Mano & Charles R. Kime
- [17]. Digital Logic and Computer Design — R. P. Jain
- [18]. Contemporary Logic Design— Randal E. Bryant & David R. O'Hallaron
- [19]. CMOS VLSI Design: A Circuits and Systems Perspective — Neil H. E. Weste & David Harris
- [20]. Computer Organization & Architecture — William Stallings
- [21]. <https://ieeexplore.ieee.org/>
- [22]. <https://www.sciencedirect.com/>
- [23]. <https://scholar.google.com/>
- [24]. <https://www.geeksforgeeks.org/digital-electronics/>
- [25]. [https://www.tutorialspoint.com/digital\\_electronics/index.html](https://www.tutorialspoint.com/digital_electronics/index.html)