FACULTY OF ENGINEERING & TECHNOLOGY

Dileep Kumar
Assistant Prof. EE Deptt

LOGIC GATE

Logic gates are used to carry out logical operations on single or multiple binary inputs & give one binary output. In simple terms, logic gates are the electronic circuits in a digital system.

Truth Table

A truth table shows how a logic circuit's output responds to various combinations of the inputs, using logic 1 for true and logic 0 for false.

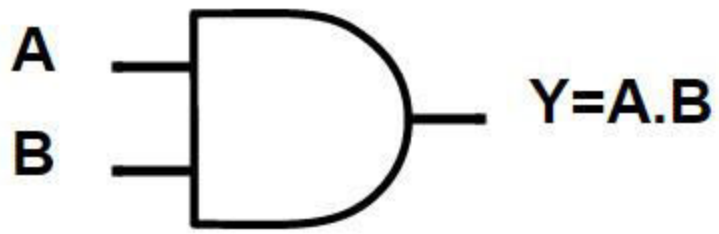
Type of Logic Gate

There are three types of logic gate

1. Basic Gate (AND,OR & NOT)
2. Universal Gate (NAND & NOR)
3. Arithmetic Gate (EXOR & EXNOR)

1. Basic Gate

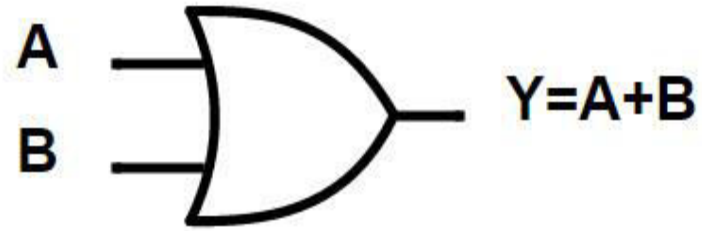
- **AND Gate:**
  In AND gate the output of an AND gate attains the state 1 if and only if all the inputs are in state 1.

Symbol	Truth Table		
2-input AND Gate 	A	B	Y
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolean Expression **Y = A.B**	Read as A **AND** B gives Y		

LOGIC GATE

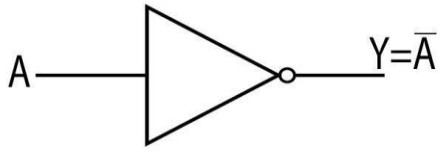
- **OR Gate:**

In OR gate the output of an OR gate attains the state 1 if one or more inputs attain the state 1.

Symbol	Truth Table		
2-input OR Gate	A	B	Y
	0	0	0
	0	1	1
	1	0	1
	1	1	1
Boolean Expression **Y = A + B**	Read as A **OR** B gives Y		

- **NOT Gate:**

In NOT gate the output of a NOT gate attains the state 1 if and only if the input does not attain the state 1.

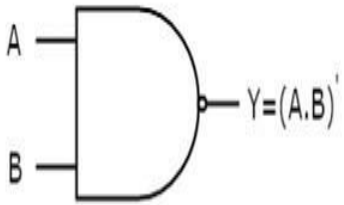
Symbol	Truth Table	
	A	Y
	0	1
	1	0
Boolean Expression Y = not A or Á	Read as inverse of **A** gives Y	

LOGIC GATE

2. Universal Gate:

• **NAND Gate**

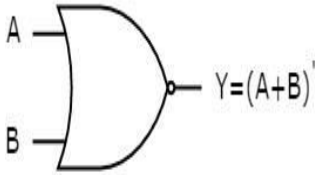
This gate is the combination of AND & NOT gate.

<table>
<tr><th>Symbol</th><th colspan="3">Truth Table</th></tr>
<tr><td rowspan="5">2-input NAND Gate
 
</td><td>A</td><td>B</td><td>Y</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>Boolean Expression Y = (A . B)’</td><td colspan="3">Read as A NAND B gives Y</td></tr>
</table>

LOGIC GATE

•**NOR Gate**

This gate is the combination of OR & NOT gate.

Symbol	Truth Table		
2-input NAND Gate 	A	B	Y
	0	0	1
	0	1	0
	1	0	0
	1	1	0
Boolean Expression Y = (A + B)'	Read as A N**OR** B gives Y		