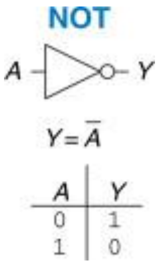
Logic gates

A 'Logic Gate' is a type of simple digital circuit that takes binary inputs and produces a binary output. It is used in digital systems to perform operations on binary variables.

Logic gates are simple digital circuits that take one or more binary inputs and produce a binary output. Logic gates are drawn with a symbol showing the input (or inputs) and the output. Inputs are usually drawn on the left (or top) and outputs on the right (or bottom). Digital designers typically use letters near the beginning of the alphabet for gate inputs and the letter *Y* for the gate output. The relationship between the inputs and the output can be described with a truth table or a Boolean equation. A *truth table* lists inputs on the left and the corresponding output on the right. It has one row for each possible combination of inputs. A *Boolean equation* is a mathematical expression using binary variables.

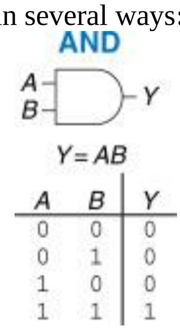
1. NOT Gate

A *NOT gate* has one input, *A*, and one output, *Y*, as shown in Figure bellow. The NOT gate's output is the inverse of its input. If *A* is FALSE, then *Y* is TRUE. If *A* is TRUE, then *Y* is FALSE. This relationship is summarized by the truth table and Boolean equation in the figure. The line over *A* in the Boolean equation is pronounced *NOT*, so $\mathbf{Y=\bar{A}}$ is read "*Y* equals NOT A." The NOT gate is also called an *inverter*.



2. AND Gate

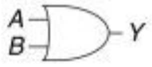
Two-input logic gates are more interesting. The *AND gate* shown in Figure bellow produces a TRUE output, *Y*, if and only if both *A* and *B* are TRUE. Otherwise, the output is FALSE.
The Boolean equation for an AND gate can be written in several ways**:** $\boldsymbol{Y = A \bullet B}$**,** $\boldsymbol{Y = AB}$**,** read "*Y* equals *A* and *B*



3. **OR Gate**

The *OR gate* shown in Figure 1.15 produces a TRUE output, *Y*, if either *A* or *B* (or both) are TRUE. The Boolean equation for an OR gate is written as $\boldsymbol{Y = A + B}$. Digital designers normally use the + notation, $\boldsymbol{Y = A + B}$ is pronounced "*Y* equals *A* or *B*."

OR



$Y = A + B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

4. NAND Gate

Any gate can be followed by a bubble to invert its operation. The *NAND gate* performs NOT AND. Its output is TRUE unless both inputs are TRUE as shown in figure bellow .

NAND

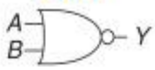
$Y = \overline{AB}$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

5.NOR Gate

The *NOR gate* performs NOT OR. Its output is TRUE if neither *A* nor *B* is TRUE as shown in figure bellow:

NOR



$Y = \overline{A + B}$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

6.EX-OR Gate

XOR (exclusive OR, pronounced “ex-OR”) is TRUE if *A* or *B,* but not both, are TRUE. The XOR operation is indicated by ⊕, a plus sign with a circle around it.

XOR

$Y = A \oplus B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

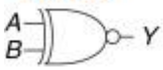
An *N*-input XOR gate is sometimes called a *parity* gate and produces a TRUE output if an odd number of inputs are TRUE. As with two-input gates, the input combinations in the truth table are listed in counting order.

7.XNOR Gate

Figure bellow shows the symbol and Boolean equation for a two-input *XNOR (pronounced ex-NOR) gate* that performs the inverse of an XOR.

The XNOR output is TRUE if both inputs are FALSE or both inputs are TRUE. The two-input XNOR gate is sometimes called an *equality* gate because its output is TRUE when the inputs are equal.

XNOR



$$Y = \overline{A \oplus B}$$

A	B	Y	A	B	Y
0	0		0	0	1
0	1		0	1	0
1	0		1	0	0
1	1		1	1	1

Summery :

- AND gate: the output is 1 if all inputs are 1; otherwise, the output is 0.
- OR gate: the output is 1 if at least one input is 1; otherwise, the output is 0.
- XOR gate: the output is 0 if both inputs are same; otherwise, the output is 1.
- NAND gate: the output is 1 if at lease one input is 0; otherwise, the output is 0.
- NOR gate: the output is 1 if both inputs are 0; otherwise, the output is 0.
- NOT gate or inverter: the output is 1 if the input is 0 and the output is 0 if the input is 1.