

## Theory of Logic Gate

PushendraSoyal, BharatKumar, RajdeepMeena, NarsiYadav, BhuvneshYadav

Department of Physics, Students, B.Sc. Final year, Parishkar College of Global Excellence, Jaipur, India

### Abstract

*Digital Electronics is the back bone of current world due to integrated circuit fabrications process, electronic circuits includes very complex embedded digital block as serial receivers. Here, in this paper we will be discussing about various gates and their applications with which they made our life easier*

### Introduction

#### Logic gates

Digital systems are said to be constructed by using logic gates. These gates are the AND, OR, NOT, NAND, NOR, EXOR and EXNOR gates. The basic operations are described below with the aid of [truth tables](#).

#### AND gate



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

The AND gate is an electronic circuit that gives a **high** output (1) only if **all** its inputs are high. A dot (.) is used to show the AND operation i.e. A.B. Bear in mind that this dot is sometimes omitted i.e. AB

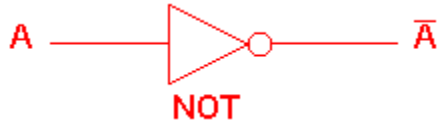
#### OR gate



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

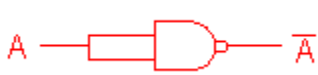
The OR gate is an electronic circuit that gives a high output (1) if **one or more** of its inputs are high. A plus (+) is used to show the OR operation.

## NOT gate



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

The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an *inverter*. If the input variable is A, the inverted output is known as NOT A. This is also shown as A', or A with a bar over the top, as shown at the outputs. The diagrams below show two ways that the NAND logic gate can be configured to produce a NOT gate. It can also be done using NOR logic gates in the same way.



## NAND gate



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

This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if **any** of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

## NOR gate



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

This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if **any** of the inputs are high. The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

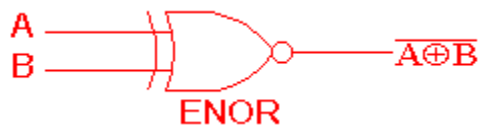
### EXOR gate



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

The 'Exclusive-OR' gate is a circuit which will give a high output if **either, but not both**, of its two inputs are high. An encircled plus sign ( $\oplus$ ) is used to show the EOR operation.

### EXNOR gate



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

The 'Exclusive-NOR' gate circuit does the opposite to the EOR gate. It will give a low output if **either, but not both**, of its two inputs are high. The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion.

The NAND and NOR gates are called *universal functions* since with either one the AND and OR functions and NOT can be generated.

Note:

A function in *sum of products* form can be implemented using NAND gates by replacing all AND and OR gates by NAND gates.

A function in *product of sums* form can be implemented using NOR gates by replacing all AND and OR gates by NOR gates.

**Table 1: Logic gate symbols**

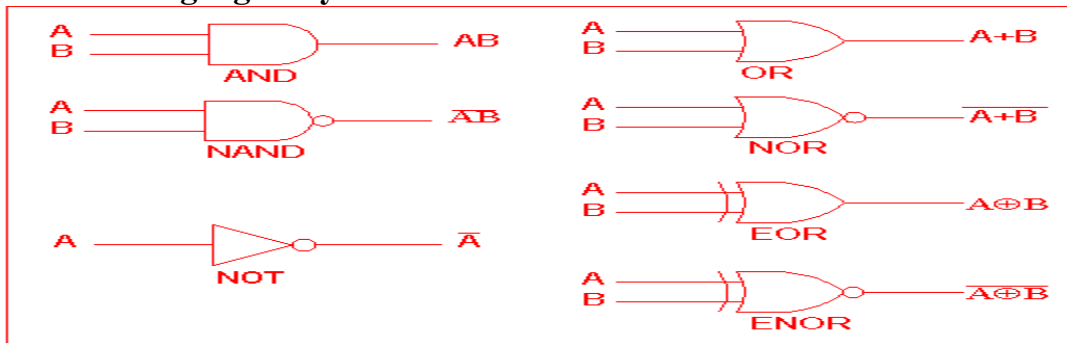


Table 2 is a summary truth table of the input/output combinations for the NOT gate together with all possible input/output combinations for the other gate functions. Also note that a [truth table](#) with 'n' inputs has  $2^n$  rows. You can compare the outputs of different gates.

**Table 2: Logic gates representation using the Truth table**

		INPUTS		OUTPUTS					
		A	B	AND	NAND	OR	NOR	EXOR	EXNOR
<b>NOT gate</b>		0	0	0	1	0	1	0	1
A	$\overline{A}$	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	0	1	0
1	0	1	1	1	0	1	0	0	1

## References

- [1] [http://minecraft.gamepedia.com/Tutorials/Basic\\_Logic\\_Gates](http://minecraft.gamepedia.com/Tutorials/Basic_Logic_Gates)
- [2] <http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Comb/gates.html>
- [3] <http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/index.html>
- [4] [http://en.wikipedia.org/wiki/Logic\\_gate](http://en.wikipedia.org/wiki/Logic_gate)
- [5] [http://www.electronics-tutorials.ws/logic/logic\\_1.html](http://www.electronics-tutorials.ws/logic/logic_1.html)
- [6] [http://www.electronicstutorials.ws/logic/logic\\_10.html](http://www.electronicstutorials.ws/logic/logic_10.html)
- [7] <http://www.circuitstoday.com/half-adder>