Understanding Basic Logic Gates: AND, OR, XOR, NOT, NAND, NOR and XNOR

2023-10-26 Author: Quarktwin

Logic gates are the foundation of intricate operations in digital electronics. While theoretically, we can integrate an unlimited number of these gates into a device, there are practical physical constraints on how many can fit into a given space. Such assemblies of gates are commonly found in digital integrated circuits (ICs).

As advancements in IC technology continue, the physical die size of each gate shrinks, empowering digital devices to conduct increasingly complex tasks at unparalleled speeds, all while maintaining or even reducing their overall size.

Building on the foundation of these technological advancements, it's essential to understand the core elements that make these complexities possible. This article focuses specifically on seven fundamental logic gates. By understanding these basics, you'll gain a clearer insight into the marvels of modern digital devices and the intricate operations they perform.

What are Logic Gates?

Logic gates execute basic logical functions that are fundamental to digital circuits. These basic gates are the basis of more complex digital devices and systems that enable our computers, smartphones, and other digital devices to function.

What are Truth Tables?

The truth table displays the logical operations on input signals in a table format. Every Boolean expression can be viewed as a truth table. The truth table identifies all possible input combinations and the output for each. It is common to create the table so that the input combinations produce an unsigned binary up-count.

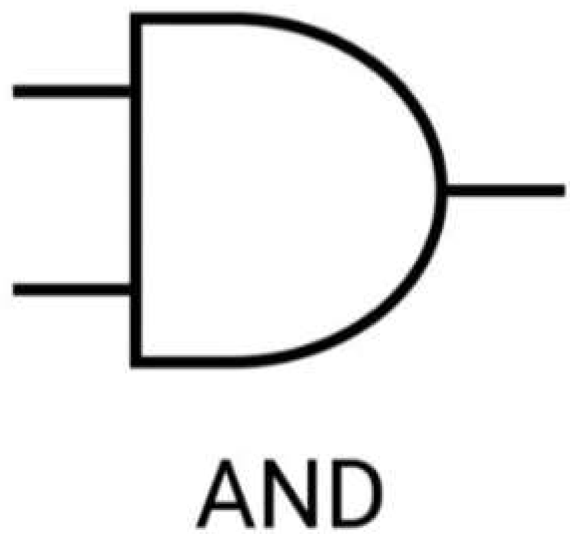
AND Gate

An AND gate produces a high output (1) only if all its inputs are high. If one or more of an AND gate's inputs are low (0), it produces a low output.

The name "AND gate" is derived from its function: if we label 0 as "false" and 1 as "true," the gate mimics the behavior of the logical "and" operation. The accompanying diagram and table display the symbol and logic outcomes for an AND gate.

In this symbol, inputs are on the left while the output is on the right. The output turns "true" only when both inputs are simultaneously "true." If not, the output remains "false." Simply put, the output registers as 1 only when both input one AND input two register as 1.

Symbol



Truth Table

RR Note: Truth Tables in this article should be placed in <table></table> html tags if possible

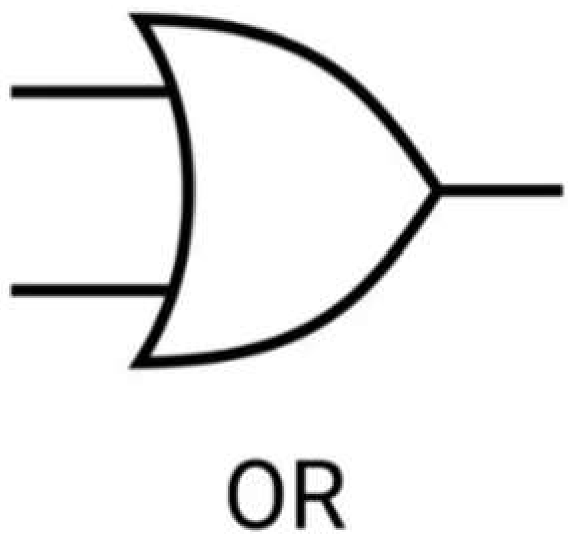
A	B	**Output**
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

An OR gate produces a high output (1) if one or more of its inputs are high. If all inputs are low, it produces a low output.

The "OR gate" is named for its resemblance to the logical inclusive "or" operation. The output is "true" when at least one of the inputs is "true." However, if both inputs are "false," the output remains "false." Simply put, the output will be 1 if either input one OR input two is 1.

Symbol



Truth Table

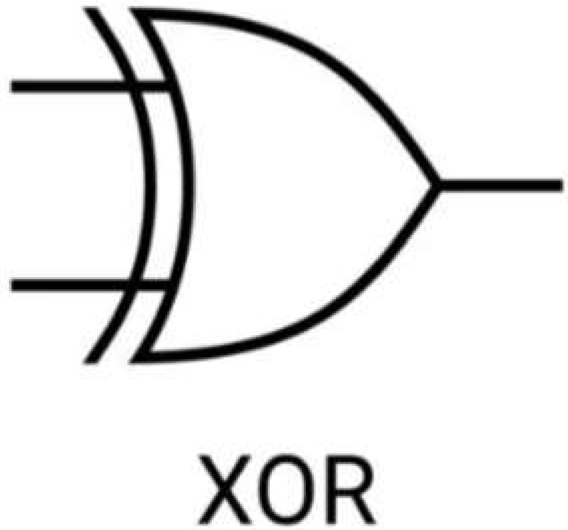
A	B	**Output**
0	0	0
0	1	1
0	0	1
1	1	1

XOR (Exclusive OR) Gate

An XOR gate produces a high output (1) only if an odd number of its inputs are high. For two-input XOR gates, this means the output is high only when one input is high and the other is low.

The XOR gate, standing for "exclusive-OR," operates similarly to the logical "either/or" concept. The output is "true" when one, and only one, of the inputs is "true." It becomes "false" if both inputs are the same, whether "true" or "false." Essentially, the output is 1 when the inputs differ and 0 when they match.

Symbol



Truth Table

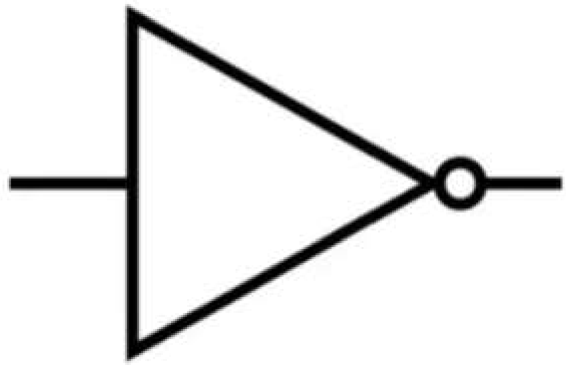
A	B	**Output**
0	0	0
0	1	1
1	0	1
1	1	0

NOT (Inverter) Gate

A NOT gate inverts its input. If the input is high (1), the output is low (0), and vice versa.

Often referred to as a “NOT” gate to distinguish it from other electronic inverter types, a logical inverter has a single input and flips the logic state. Specifically, an input of 1 yields an output of 0.

Symbol



Truth Table

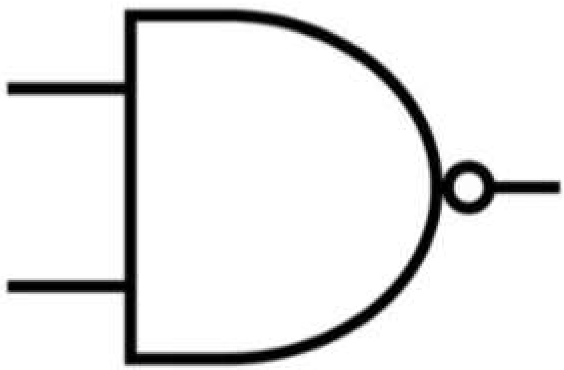
A	**Output**
0	1
1	0

NAND (NOT AND) Gate

A NAND gate is the inverse of the AND gate. It produces a low output (0) only if all its inputs are high.

The NAND gate functions like a combination of an AND gate succeeded by a NOT gate. It essentially performs the logical "and" operation and then negates the result. The output becomes "false" only when both inputs are "true"; for all other input combinations, the output is "true."

Symbol



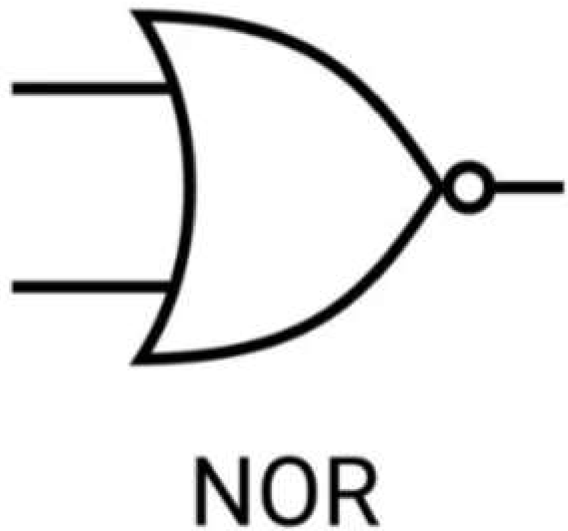
Truth Table

A	B	**Output**
0	0	1
0	1	1
1	0	1
1	1	0

NOR (NOT OR) Gate

A NOR gate is the inverse of the OR gate. It produces a high output (1) only if all its inputs are low. Its output is "true" if both inputs are "false." Otherwise, the output is "false."

Symbol



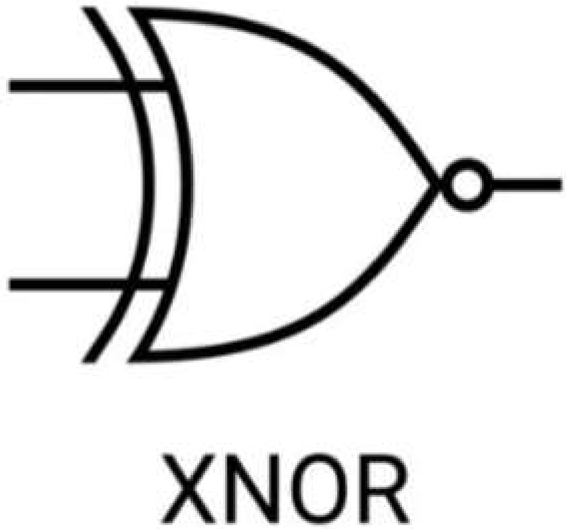
Truth Table

A	B	**Output**
0	0	1
0	1	0
1	0	0
1	1	0

XNOR (Exclusive NOR) Gate

An XNOR gate produces a high output (1) if an even number of its inputs are high. For two-input XNOR gates, this means the output is high when both inputs are identical, either both high or both low.

Symbol



Truth Table

A	B	**Output**
0	0	1
0	1	0
1	0	1
1	1	0

Logic Gates in Summary

These seven fundamental logic gates form the foundation of digital electronics. They can be combined in various ways to build more complex circuits, enabling the creation of memory units, processors, and the vast range of digital devices we use in everyday life.