**LOGIC GATES AND TRUTH TABLE**

A logic gate is a building block of a [digital](https://whatis.techtarget.com/definition/digital) [circuit](https://whatis.techtarget.com/definition/circuit). Most logic gates have two inputs and one output and are based on [Boolean](https://whatis.techtarget.com/definition/Boolean) algebra. At any given moment, every terminal is in one of the two [binary](https://whatis.techtarget.com/definition/binary) conditions true (high) or false (low). False represents 0, and true represents 1. Depending on the type of logic gate being used and the combination of inputs, the binary output will differ.  A logic gate can be thought of like a light switch, wherein one position the output is off—0, and in another, it is on—1.  Logic gates are commonly used in integrated circuits ([IC](https://whatis.techtarget.com/definition/integrated-circuit-IC)).

**Basic logic gates**

There are seven basic logic gates: AND, OR, XOR, NOT, NAND, NOR, and XNOR.

[AND](https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR?vgnextfmt=print#and) | [OR](https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR?vgnextfmt=print#or) | [XOR](https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR?vgnextfmt=print#xor) | [NOT](https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR?vgnextfmt=print#not) | [NAND](https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR?vgnextfmt=print#nand) | [NOR](https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR?vgnextfmt=print#nor) | [XNOR](https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR?vgnextfmt=print#xnor)

The *AND gate* is so named because, if 0 is called "false" and 1 is called "true," the gate acts in the same way as the logical "and" operator. The following illustration and table show the circuit symbol and logic combinations for an AND gate. (In the symbol, the input terminals are at left and the output terminal is at right.) The output is "true" when both inputs are "true." Otherwise, the output is "false." In other words, the output is 1 only when both inputs one AND two are 1.



**Truth Table of AND gate**

|  |  |  |
| --- | --- | --- |
| **Input 1** | **Input 2** | **Output** |
|  0 |  0 |  0 |
|  0 | 1 |  0 |
| 1 |  0 | 0  |
| 1 | 1 | 1 |

Discrete AND gates may be realized by using diodes or transistors. The inputs represented as X and Y may be either 0V or +5V correspondingly. The output is represented by Z . In the diode of AND gate, when both the inputs are of same value, X=+5V and Y= +5V, then the diodes are in OFF condition. As a result, no current flows through the resistor and there will not be any voltage drop across the resistor. Here the output will be Z=+5V. Similarly, when both the inputs such as X and Y are equal to 0V, then the corresponding diodes such as either D1 or D2 or both the diodes are at ON state and act as short circuits. Here the output will be Z corresponds to 0V. In practical cases the output z corresponds to 0.6V or 0.7V, which is treated as logic 0 state.



2 Input Diode AND Gate

So the final output voltage corresponds to +5V. The truth table for this gate circuit is shown below:



Discrete AND Gate Truth Table

**OR GATE**

Similar to AND gate, an OR gate may also have two or more inputs but produce only one output. The OR gate produces an output of logic 1 state even if any of its inputs is in logic 1 state and also produces an output of logic 0 state if any of its inputs is in logic 0 state. The symbol for OR operation is ‘+’.  If the inputs are of X and Y, then the output can be represented as Z=X+Y. An OR gate may also be defined as a device whose output is 1, even if one of its input is 1. OR gate is also called as any or all gate. It is also called as an inclusive OR gate because it consists of the condition of ‘both the inputs can be present’.  The logic symbols and truth table for two-input and three-input OR gates are given below.

.2 Input OR Gate – Truth Table

3 Input OR Gate – Truth Table

Discrete OR gates may be realized by using diodes or transistors. The inputs represented as X and Y may be either  0V or +5V correspondingly. The output is represented by Z . In the diode of OR gate, when both the inputs are of same value, X=0V and Y= 0V, then both the diodes are in OFF condition. As a result, no current flows through the resistor and there will not be any voltage drop across the resistor. Here the output will be Z=0V. Similarly, when both the inputs or either the inputs such as X and Y are equal to +5V, then the corresponding diodes either D1 or D2 or both the diodes are at ON state and act as short circuits. Here the output will be Z corresponds to +5V. In practical cases the output Z corresponds to +5V-diode drop =  +5V – 0.7V = +4.3V, which is regarded as Logic 1 state.



2 Input Diode OR Gate

The truth table for this gate circuit is shown below:

Discrete OR Gate Truth Table

**NOT GATE**

The NOT gate is also called as an inverter, simply because it changes the input to its opposite. The NOT gate is having only one input and one corresponding output. It is a device whose output is always the compliment of the given input. That means, the NOT gate produces an output of logic 1 state when the input is of logic 0 state and also produce the output of logic 0 state when the input is of logic 1 state. The NOT operation is denoted by ’-‘(bar). When the input variable to the NOT gate is represented by ‘X’ and the output is represented by ‘Z’. In the NOT operation it can be read as ‘Z is equal to X bar’. The logic symbol and truth table are given below:

NOT Gate – Truth Table

Discrete NOT gate may be realized by using transistors. The inputs represented as X may be either 0V or +5V correspondingly. The output is represented by Z. When the input X = 0V, then the transistor Q1 will be reverse biased and therefore it remains OFF. As a result no current flows through the resistor and thereby there will not be any voltage drop across the resistor. As a result, the output voltage Z corresponds to +5V. When the input X= +5V, transistor Q1 is ON and the output voltage Z=Vce(sat) corresponds to 0V. The truth table for the NOT gate is given below:

Transistor Inverter NOT GateTransistor Inverter NOT Gate – Truth Table

**NAND GATE**

The NAND and NOR gates are the universal gates. Each of this gates can realize the logic circuits single handedly. The NAND and NOR are also called as universal building blocks. Both NAND and NOR has the ability to perform three basic logic functions such as AND,OR and NOT. NAND gate is a combination of an AND gate and a NOT gate. The expression for the NAND gate is ‘—‘whole bar. The output of the NAND gate is at logic 0 level only when each of the inputs assumes a logic 1 level. The truth table of two-input NAND gate is given below:

2 Input NAND Gate -Truth Table

A Discrete two-input NAND  gate is as shown in the figure. The two inputs are represented by X and Y. The output is represented by Z .  When the input X and Y= +5V, then both the diodes D1 and D2 are OFF. The transistor Q1 gets enough base drive from the supply through resistor and therefore transistor Q1 is ON and the output Z=Vce(sat) corresponds to 0V. Similarly when inputs either x=0V or Y=0V or when both inputs are equal to 0V, at that time the transistor Q1 is OFF and therefore, output voltage Z= +5V. The truth table is given below:

Discrete NAND GateDiscrete NAND Gate -Truth Table

**NOR GATE**

NOR means NOT OR. That means, NOR gate is a combination of an OR gate and a NOT gate. The output is logic 1 level, only when each of its inputs assumes a logic 0 level. For any other combination of inputs, the output is a logic 0 level. The truth table of two-input NOR gate is given below:

2 Input NOR Gate – Truth Table3 Input NOR Gate -Truth Table

Discrete two-input NOR gate is as shown in the figure. The inputs represented as X and Y may be 0V correspondingly. As a result the transistors Q1 and Q2 are OFF, as a result no current flows through the resistor and thereby there will not be any voltage drop across the resistor. Here, the output voltage Z corresponds to +5V. When either of the input X= +5V or Y=+5V or both the inputs corresponds to +5V, the corresponding transistor Q1or Q2 or both Q1 and Q2 are ON . Therefore the output voltage Z=Vce(sat) corresponds to ground and equal to 0V. The truth table for the NOR gate is given below:

Discrete NOR GateDiscrete NOR Gate – Truth Table

**EXCLUSIVE-OR GATE (X-OR) GATE**

An X-OR gate is a two input, one output logic circuit. X-OR gate assumes logic 1 state when any of its two inputs assumes a logic 1 state. When both the inputs assume the logic 0 state or when both the inputs assume the logic 1 state, the output assumes a logic 0 state. The output of the X-OR gate will be the sum of the modulo sum of its inputs. X-OR gate is also termed as anti-coincidence gate or inequality detector. An X-OR gate can also be used as inverter by connecting one of the two input terminals to logic1 and also by inputting the sequence to be inverted to the other terminal.

X-OR Gate – Truth Table

**EXCLUSIVE-NOR (X-NOR) GATE**

An X-NOR gate is a combination of an X-OR gate and a NOT gate. The X-NOR gate is also a two input, one output concept. The output of the X-NOR gate will be logic 1 state when both the inputs assume a 0 state or when both the inputs assume a 1 state. The output of the X-NOR gate will be logic 0 state when one of the inputs assume a 0 state and the other a 1 state. It is also named as coincidence gate, because its output will be 1 only when the inputs coincide. X-NOR gate can also be used as inverter by connecting one of the two input terminals to logic 0 and also by inputting the sequence to be inverted to the other terminal.



X-NOR Gate – Truth-Table

## Implementation of Logic Functions Using Only NAND Gates

The 7400 (or the 74LS00 or 74HC00) quad 2-input NAND TTL chip has four individual NAND gates within a single IC package. Thus we can use a single 7400 TTL chip to produce all the Boolean functions from a NOT gate to a NOR gate as shown.

### Logic Gates using only NAND Gates



Thus ALL other logic gate functions can be created using only NAND gates making it a universal logic gate.

## Implementation of Logic Functions Using Only NOR Gates

The 7402 (or the 74LS02 or 74HC02) quad 2-input NOR TTL chip has four individual NOR gates within a single IC package. Thus like the previous 7400 NAND IC we can use a single 7402 TTL chip to produce all the Boolean functions from a single NOT gate to a NAND gate as shown.

### Logic Gates using only NOR Gates



Thus ALL other logic gate functions can be created using only NOR gates making it also a universal logic gate.

Note also that the implementation of the Exclusive-OR gate is more efficient using NAND gates compared to using NOR gates, while the implementation of the Exclusive-NOR gate is more efficient with NOR gates compared to using NAND gates as in each case only four individual logic gates are required. In other words we can create all the Boolean functions using just one 7400 NAND or one 7402 NOR chip including its various sub-families.

**Universal Gates**

The NAND and NOR gates are the complements of the previous AND and OR functions respectively and are individually a complete set of logic as they can be used to implement any other Boolean function or gate. But as we can construct other logic switching functions using just these gates on their own, they are both called a minimal set of gates. Thus the NAND and the NOR gates are commonly referred to as **Universal Logic Gates**.

The *OR gate* gets its name from the fact that it behaves after the fashion of the logical inclusive "or." The output is "true" if either or both of the inputs are "true." If both inputs are "false," then the output is "false." In other words, for the output to be 1, at least input one OR two must be 1.



**Truth Table of OR gate**

|  |  |  |
| --- | --- | --- |
| **Input 1** | **Input 2** | **Output** |
|  0 | 0  | 0  |
|  0 | 1 | 1 |
| 1 | 0  | 1 |
| 1 | 1 | 1 |

The *XOR* ( *exclusive-OR* ) *gate* acts in the same way as the logical "either/or." The output is "true" if either, but not both, of the inputs are "true." The output is "false" if both inputs are "false" or if both inputs are "true." Another way of looking at this circuit is to observe that the output is 1 if the inputs are different, but 0 if the inputs are the same.

  

**Truth Table of XOR gate**

|  |  |  |
| --- | --- | --- |
| **Input 1** | **Input 2** | **Output** |
|  0 | 0  | 0  |
|  0 | 1 | 1 |
| 1 |  0 | 1 |
| 1 | 1 |  0 |

A logical *inverter*, sometimes called a *NOT gate* to differentiate it from other types of electronic inverter devices, has only one input. It reverses the logic state. If the input is 1, then the output is 0. If the input is 0, then the out is 1.

A logical *inverter*, sometimes called a *NOT gate* to differentiate it from other types of electronic inverter devices, has only one input. It reverses the logic state. If the input is 1, then the output is 0. If the input is 0, then the output is 1.

**Truth Table of Inverter or NOT gate**  

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 |  0 |
|  0 | 1 |

 The *NAND gate* operates as an AND gate followed by a NOT gate. It acts in the manner of the logical operation "and" followed by negation. The output is "false" if both inputs are "true." Otherwise, the output is "true."



**Truth Table of NAND gate**

|  |  |  |
| --- | --- | --- |
| **Input 1** | **Input 2** | **Output** |
|  0 | 0  | 1 |
|  0 | 1 | 1 |
| 1 | 0  | 1 |
| 1 | 1 |  0 |

The *NOR gate* is a combination OR gate followed by an inverter. Its output is "true" if both inputs are "false." Otherwise, the output is "false."



**Truth Table of NOR gate**

|  |  |  |
| --- | --- | --- |
| **Input 1** | **Input 2** | **Output** |
|   |   | 1 |
|   | 1 |   |
| 1 |   |   |
| 1 | 1 |   |

The *XNOR (exclusive-NOR) gate* is a combination XOR gate followed by an inverter. Its output is "true" if the inputs are the same, and "false" if the inputs are different.



**Truth Table of XNOR gate**

|  |  |  |
| --- | --- | --- |
| **Input 1** | **Input 2** | **Output** |
|  0 |  0 | 1 |
|  0 | 1 |  0 |
| 1 |  0 |  0 |
| 1 | 1 | 1 |

Using combinations of logic gates, complex operations can be performed. In theory, there is no limit to the number of gates that can be arrayed together in a single device. But in practice, there is a limit to the number of gates that can be packed into a given physical space. Arrays of logic gates are found in digital ICs. As IC technology advances, the required physical volume for each individual logic gate decreases and digital devices of the same or smaller size become capable of performing ever-more-complicated operations at ever-increasing speeds.

**Composition of logic gates**

High or low binary conditions are represented by different [voltage](https://whatis.techtarget.com/definition/voltage) levels. The logic state of a terminal can, and generally does, change often as the circuit processes data. In most logic gates, the low state is approximately zero [volts](https://whatis.techtarget.com/definition/volt) (0 V), while the high state is approximately five volts positive (+5 V).

Logic gates can be made of [resistors](https://whatis.techtarget.com/definition/resistor) and transistors, or diodes. A resistor can commonly be used as a pull-up or pull-down resistor. Pull-up or pull-down resistors are used when there are any unused logic gate inputs to connect to either a logic level 1 or 0 respectively. This prevents any false switching of the gate. Pull-up resistors are connected to Vcc (+5V), and pull-down resistors are connected to ground (0 V).

Commonly used logic gates are [TTL](https://whatis.techtarget.com/definition/transistor-to-transistor-logic-TTL) and CMOS. TTL, or Transistor-Transistor Logic, ICs will use NPN and PNP type Bipolar Junction [Transistors](https://whatis.techtarget.com/definition/transistor-to-transistor-logic-TTL). CMOS, or Complementary Metal-Oxide-Silicon, ICs are constructed from MOSFET or JFET type [Field Effect Transistors](https://whatis.techtarget.com/definition/field-effect-transistor-FET). TTL IC’s may commonly be labeled as the 7400 series of chips, while CMOS ICs may often be marked as a 4000 series of chips.

Boolean functions may be practically implemented by using electronic gates. The following points are important to understand.

* Electronic gates require a power supply.
* Gate **INPUTS** are driven by voltages having two nominal values, e.g. 0V and 5V representing logic 0 and logic 1 respectively.
* The **OUTPUT** of a gate provides two nominal values of voltage only, e.g. 0V and 5V representing logic 0 and logic 1 respectively. In general, there is only one output to a logic gate except in some special cases.
* There is always a time delay between an input being applied and the output responding.

  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](http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/index.html#truth).

**AND gate**



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**

 

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**

 

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**

 

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**

 

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**

 

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 () is used to show the EOR operation.

**EXNOR gate**

** **

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](http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/index.html#truth) with 'n' inputs has 2n rows. You can compare the outputs of different gates.

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

 

  Example

A [NAND gate](http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/index.html#nandgate) can be used as a [NOT gate](http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/index.html#notgate) using either of the following wiring configurations.

                                        (You can check this out using a truth table.)

###