

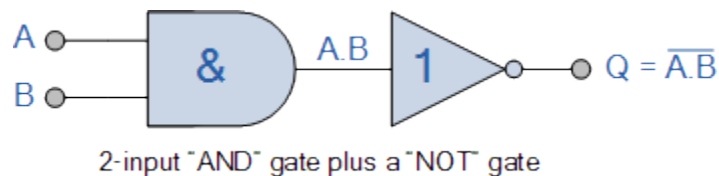


Logic NAND Gate Tutorial

The Logic NAND Gate is a combination of a digital logic AND gate and a NOT gate connected together in series

The NAND (Not – AND) gate has an output that is normally at logic level “1” and only goes “LOW” to logic level “0” when **ALL** of its inputs are at logic level “1”. The **Logic NAND Gate** is the reverse or “*Complementary*” form of the AND gate we have seen previously.

Logic NAND Gate Equivalence



The logic or Boolean expression given for a logic NAND gate is that for *Logical Addition*, which is the opposite to the AND gate, and which it performs on the *complements* of the inputs. The Boolean expression for a logic NAND gate is denoted by a single dot or full stop symbol, (.) with a line or *Overline*, ($\bar{}$) over the expression to signify the NOT or logical negation of the NAND gate giving us the Boolean expression of: $A.B = Q$.

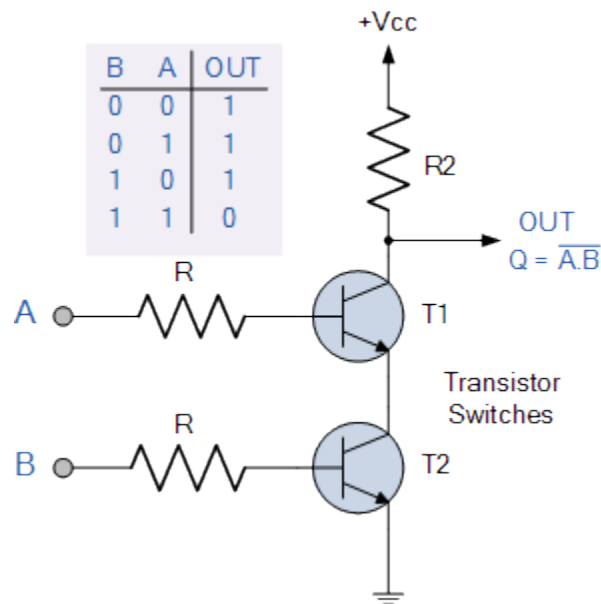
Then we can define the operation of a 2-input digital logic NAND gate as being:

“If both A and B are true, then Q is NOT true”

Transistor NAND Gate

A simple 2-input logic NAND gate can be constructed using RTL Resistor-transistor switches connected together as shown below with

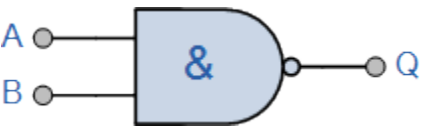
the inputs connected directly to the transistor bases. Either transistor must be cut-off “OFF” for an output at Q.



Logic NAND Gates are available using digital circuits to produce the desired logical function and is given a symbol whose shape is that of a standard AND gate with a circle, sometimes called an “inversion bubble” at its output to represent the NOT gate symbol with the logical operation of the NAND gate given as.


The Digital Logic “NAND” Gate

2-input Logic NAND Gate

Symbol	Truth Table		
 <p>2-input NAND Gate</p>	B	A	Q
	0	0	1
	0	1	1
	1	0	1

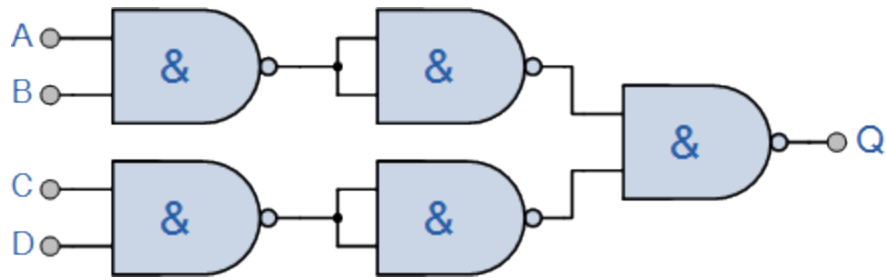
	1	1	0
Boolean Expression $Q = A.B$	Read as A AND B gives NOT Q		

3-input Logic NAND Gate

Symbol	Truth Table			
 <p>3-input NAND Gate</p>	C	B	A	Q
	0	0	0	1
	0	0	1	1
	0	1	0	1
	0	1	1	1
	1	0	0	1
	1	0	1	1
	1	1	0	1
	1	1	1	0
Boolean Expression $Q = A.B.C$	Read as A AND B AND C gives NOT Q			

As with the AND function seen previously, the NAND function can also have any number of individual inputs and commercial available NAND Gate IC's are available in standard 2, 3, or 4 input types. If additional inputs are required, then the standard NAND gates can be cascaded together to provide more inputs for example.

A 4-input NAND Function



The Boolean Expression for this 4-input logic NAND gate will therefore be: **$Q = A.B.C.D$**

If the number of inputs required is an odd number of inputs any “unused” inputs can be held HIGH by connecting them directly to the power supply using suitable “Pull-up” resistors.

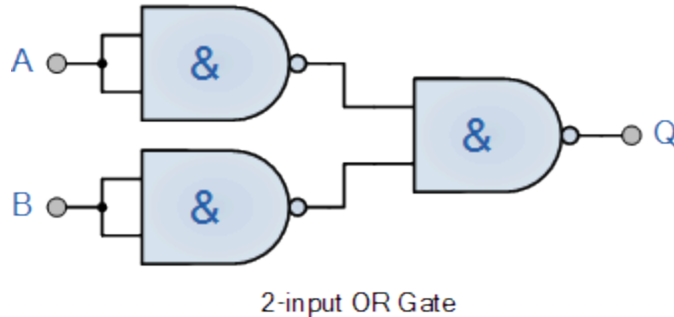
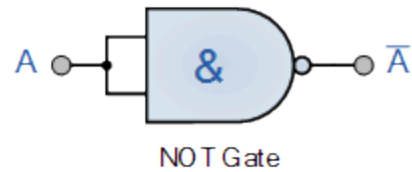
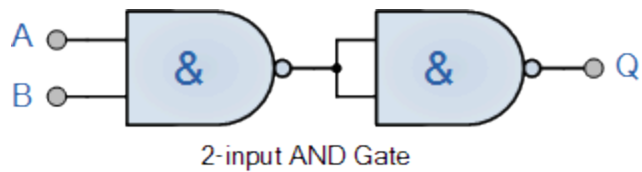
The **Logic NAND Gate** function is sometimes known as the **Sheffer Stroke Function** and is denoted by a vertical bar or upwards arrow operator, for example, A NAND B = $A|B$ or $A\uparrow B$.

The “Universal” NAND Gate

The **Logic NAND Gate** is generally classed as a “Universal” gate because it is one of the most commonly used logic gate types. NAND gates can also be used to produce any other type of logic gate function, and in practice the NAND gate forms the basis of most practical logic circuits.

By connecting them together in various combinations the three basic gate types of AND, OR and NOT function can be formed using only NAND gates, for example.

Various Logic Gates using only NAND Gates



As well as the three common types above, Exclusive-OR, Exclusive-NOR and standard NOR gates can be formed using just individual NAND gates.

Commonly available digital logic NAND gate IC's include:

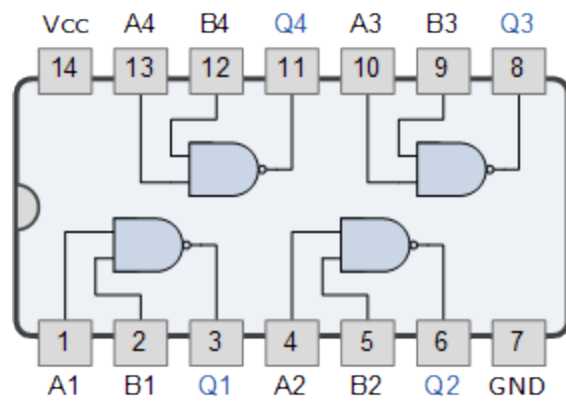
TTL Logic NAND Gates

- 74LS00 Quad 2-input
- 74LS10 Triple 3-input
- 74LS20 Dual 4-input
- 74LS30 Single 8-input

CMOS Logic NAND Gates

- CD4011 Quad 2-input
- CD4023 Triple 3-input
- CD4012 Dual 4-input

7400 Quad 2-input Logic NAND Gate



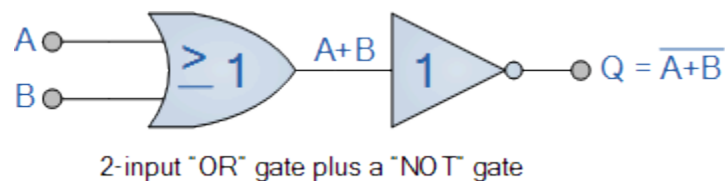


Logic NOR Gate Tutorial

The Logic NOR Gate is a combination of the digital logic OR gate and an inverter or NOT gate connected together in series

The inclusive NOR (Not-OR) gate has an output that is normally at logic level “1” and only goes “LOW” to logic level “0” when **ANY** of its inputs are at logic level “1”. The **Logic NOR Gate** is the reverse or “*Complementary*” form of the inclusive OR gate we have seen previously.

Logic NOR Gate Equivalent



The logic or Boolean expression given for a logic NOR gate is that for *Logical Multiplication* which it performs on the *complements* of the inputs. The Boolean expression for a logic NOR gate is denoted by a plus sign, (+) with a line or *Overline*, ($\bar{}$) over the expression to signify the NOT or logical negation of the NOR gate giving us the Boolean expression of: $A+B = Q$.

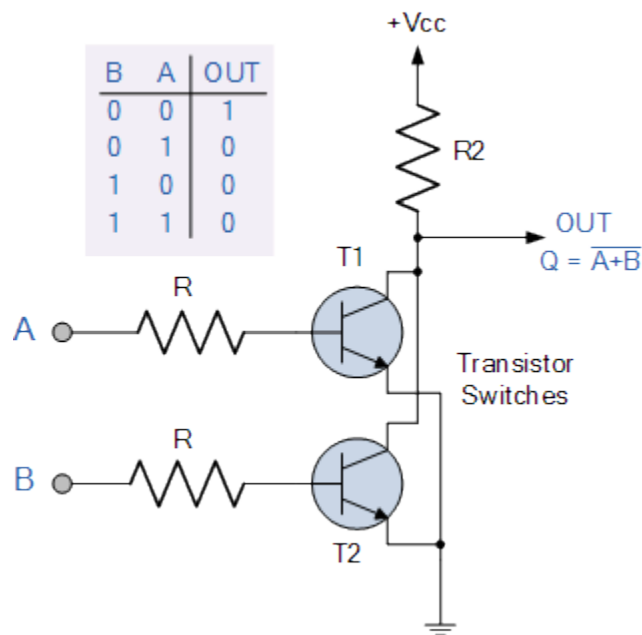
Then we can define the operation of a 2-input digital logic NOR gate as being:

“If both A and B are NOT true, then Q is true”

Transistor NOR Gate

A simple 2-input logic NOR gate can be constructed using RTL Resistor-transistor switches connected together as shown below with

the inputs connected directly to the transistor bases. Both transistors must be cut-off “OFF” for an output at Q.



Logic NOR Gates are available using digital circuits to produce the desired logical function and is given a symbol whose shape is that of a standard OR gate with a circle, sometimes called an “inversion bubble” at its output to represent the NOT gate symbol with the logical operation of the NOR gate given as.

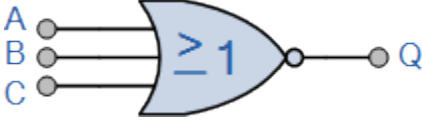
The Digital Logic “NOR” Gate

2-input NOR Gate

Symbol	Truth Table		
<div>2-input NOR Gate</div>	B	A	Q
	0	0	1
	0	1	0
	1	0	0

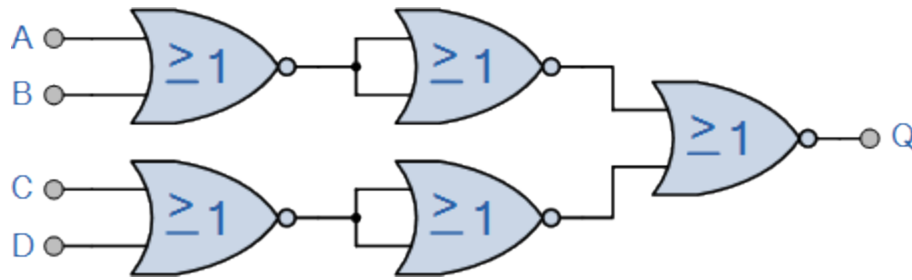
	1	1	0
Boolean Expression $Q = A+B$	Read as A OR B gives NOT Q		

3-input NOR Gate

Symbol	Truth Table			
 <p>3-input NOR Gate</p>	C	B	A	Q
	0	0	0	1
	0	0	1	0
	0	1	0	0
	0	1	1	0
	1	0	0	0
	1	0	1	0
	1	1	0	0
	1	1	1	0
Boolean Expression $Q = A+B+C$	Read as A OR B OR C gives NOT Q			

As with the OR function, the NOR function can also have any number of individual inputs and commercial available NOR Gate IC's are available in standard 2, 3, or 4 input types. If additional inputs are required, then the standard NOR gates can be cascaded together to provide more inputs for example.

A 4-input NOR Function



The Boolean Expression for this 4-input NOR gate will therefore be: **$Q = A+B+C+D$**

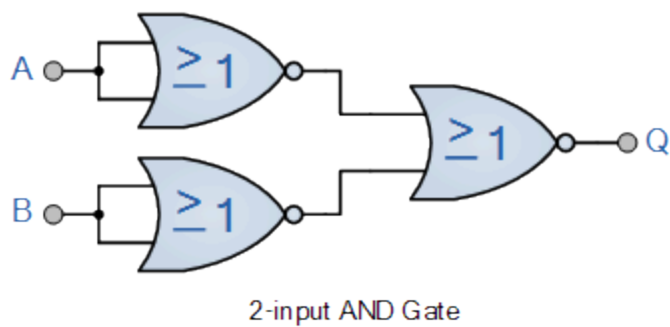
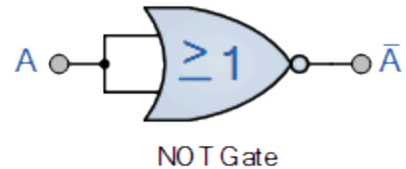
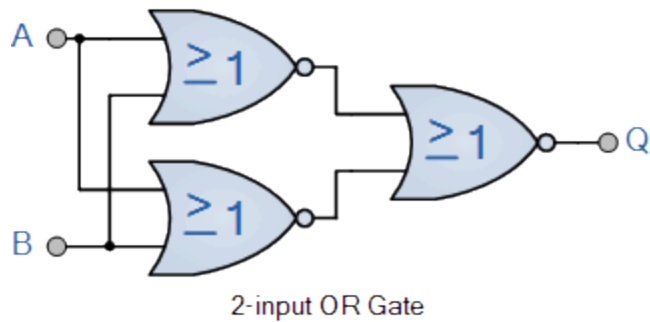
If the number of inputs required is an odd number of inputs any “unused” inputs can be held LOW by connecting them directly to ground using suitable “Pull-down” resistors.

The **Logic NOR Gate** function is sometimes known as the **Pierce Function** and is denoted by a downwards arrow operator as shown, $A \downarrow B$.

The “Universal” NOR Gate

Like the NAND gate seen in the last section, the NOR gate can also be classed as a “Universal” type gate. NOR gates can be used to produce any other type of logic gate function just like the NAND gate and by connecting them together in various combinations the three basic gate types of AND, OR and NOT function can be formed using only NOR gates, for example.

Various Logic Gates using only NOR Gates



As well as the three common types above, Exclusive-OR, Exclusive-NOR and standard NOR gates can also be formed using just individual NOR gates.

Commonly available digital logic NOR gate IC's include:

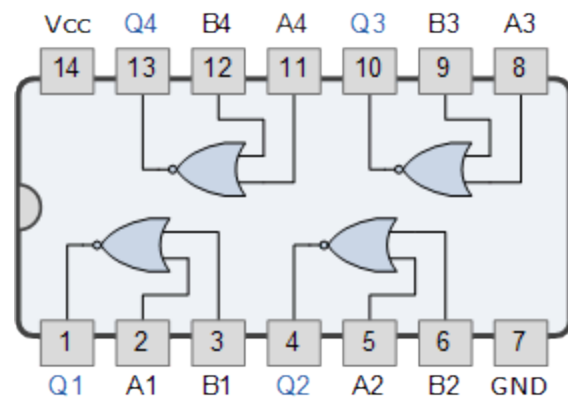
TTL Logic NOR Gates

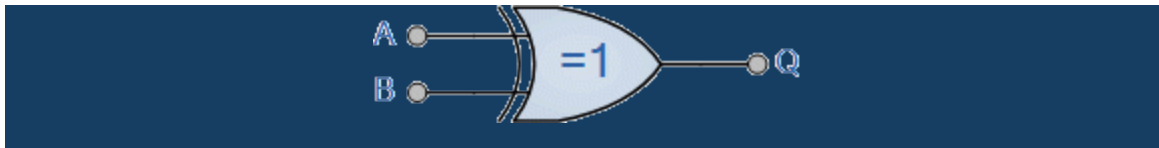
- 74LS02 Quad 2-input
- 74LS27 Triple 3-input
- 74LS260 Dual 4-input

CMOS Logic NOR Gates

- CD4001 Quad 2-input
- CD4025 Triple 3-input
- CD4002 Dual 4-input

7402 Quad 2-input NOR Gate





Exclusive-OR Gate Tutorial

The Exclusive-OR logic function is a very useful circuit that can be used in many different types of computational circuits

In the previous tutorials, we saw that by using the three principal gates, the AND Gate, the OR Gate and the NOT Gate, we can build many other types of logic gate functions, such as a NAND Gate and a NOR Gate or any other type of digital logic function we can imagine.

But there are two other types of digital logic gates which although they are not a basic gate in their own right as they are constructed by combining together other logic gates, their output Boolean function is important enough to be considered as complete logic gates. These two “hybrid” logic gates are called the **Exclusive-OR (Ex-OR) Gate** and its complement the **Exclusive-NOR (Ex-NOR) Gate**.

Previously, we saw that for a 2-input OR gate, if A = “1”, **OR** B = “1”, **OR BOTH** A + B = “1” then the output from the digital gate must also be at a logic level “1” and because of this, this type of logic gate is known as an Inclusive-OR function. The logic gate gets its name from the fact that it *includes* the case of Q = “1” when both A and B = “1”.

If however, an logic output “1” is obtained when **ONLY** A = “1” or when **ONLY** B = “1” but **NOT** both together at the same time, giving the binary inputs of “01” or “10”, then the output will be “1”. This type of gate is known as an Exclusive-OR function or more commonly an Ex-Or function for short. This is because its boolean expression *excludes* the “**OR BOTH**” case of Q = “1” when both A and B = “1”.

In other words the output of an Exclusive-OR gate **ONLY** goes “HIGH” when its two input terminals are at “**DIFFERENT**” logic levels with respect to each other.

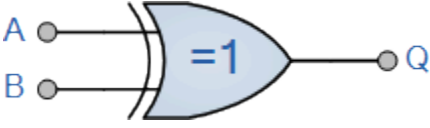
An odd number of logic “1’s” on its inputs gives a logic “1” at the output. These two inputs can be at logic level “1” or at logic level “0” giving us the Boolean expression of: $Q = (A \oplus B) = A.B + A.B$

The **Exclusive-OR Gate** function, or **Ex-OR** for short, is achieved by combining standard logic gates together to form more complex gate functions that are used extensively in building arithmetic logic circuits, computational logic comparators and error detection circuits.

The two-input “Exclusive-OR” gate is basically a modulo two adder, since it gives the sum of two binary numbers and as a result are more complex in design than other basic types of logic gate. The truth table, logic symbol and implementation of a 2-input Exclusive-OR gate is shown below.

The Digital Logic “Exclusive-OR” Gate

2-input Ex-OR Gate

Symbol	Truth Table		
 <p>2-input Ex-OR Gate</p>	B	A	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	0
Boolean Expression $Q = A \oplus B$	A OR B but NOT BOTH gives Q		

Giving the Boolean expression of: $Q = AB + \bar{A}B$

The truth table above shows that the output of an Exclusive-OR gate **ONLY** goes “HIGH” when both of its two input terminals are at “DIFFERENT” logic levels with respect to each other. If these two inputs, A and B are both at logic level “1” or both at logic level “0” the output is a “0” making the gate an “odd but not the even gate”. In other words, the output is “1” when there are an odd number of 1’s in the inputs.

This ability of the *Exclusive-OR gate* to compare two logic levels and produce an output value dependent upon the input condition is very

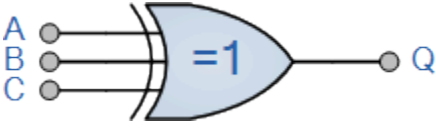
useful in computational logic circuits as it gives us the following Boolean expression of:

$$Q = (A \oplus B) = A.B + A.B$$

The logic function implemented by a 2-input Ex-OR is given as either: “A OR B but NOT both” will give an output at Q. In general, an Ex-OR gate will give an output value of logic “1” ONLY when there are an **ODD** number of 1’s on the inputs to the gate, if the two numbers are equal, the output is “0”.

Then an Ex-OR function with more than two inputs is called an “odd function” or modulo-2-sum (Mod-2-SUM), not an Ex-OR. This description can be expanded to apply to any number of individual inputs as shown below for a 3-input Ex-OR gate.

3-input Ex-OR Gate

Symbol	Truth Table			
 <p>3-input Ex-OR Gate</p>	C	B	A	Q
	0	0	0	0
	0	0	1	1
	0	1	0	1
	0	1	1	0
	1	0	0	1
	1	0	1	0
	1	1	0	0
	1	1	1	1
Boolean Expression $Q = A \oplus B \oplus C$	“Any ODD Number of Inputs” gives Q			

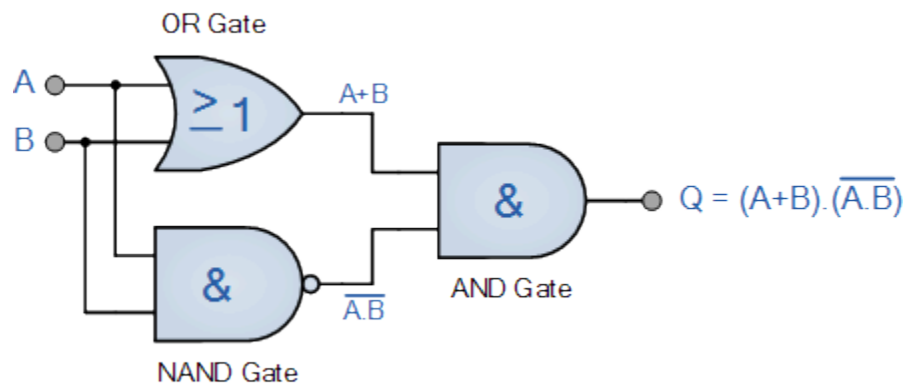
Giving the Boolean expression of: $Q = ABC + ABC + ABC + ABC$

The symbol used to denote an Exclusive-OR odd function is slightly different to that for the standard Inclusive-OR Gate. The logic or Boolean expression given for a logic OR gate is that of logical addition which is denoted by a standard plus sign.

The symbol used to describe the Boolean expression for an **Exclusive-OR** function is a plus sign, (+) within a circle (\oplus). This exclusive-OR symbol also represents the mathematical “direct sum of sub-objects” expression, with the resulting symbol for an *Exclusive-OR* function being given as: (\oplus).

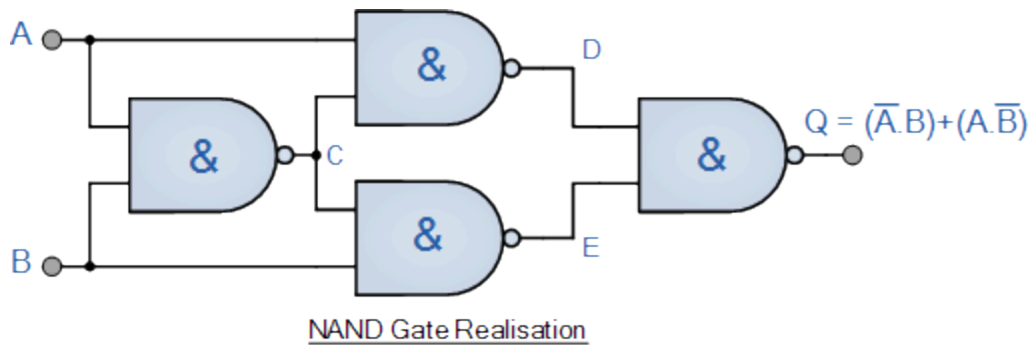
We said previously that the Ex-OR function is not a basic logic gate but a combination of different logic gates connected together. Using the 2-input truth table above, we can expand the Ex-OR function to: $(A+B).(A.B)$ which means that we can realise this new expression using the following individual gates.

Ex-OR Gate Equivalent Circuit



One of the main disadvantages of implementing the Ex-OR function above is that it contains three different types logic gates OR, NAND and finally AND within its design. One easier way of producing the Ex-OR function from a single gate is to use our old favourite the NAND gate as shown below.

Ex-OR Function Realisation using NAND gates



Exclusive-OR Gates are used mainly to build circuits that perform arithmetic operations and calculations especially **Adders** and **Half-Adders** as they can provide a “carry-bit” function or as a controlled inverter, where one input passes the binary data and the other input is supplied with a control signal.

Commonly available digital logic Exclusive-OR gate IC’s include:

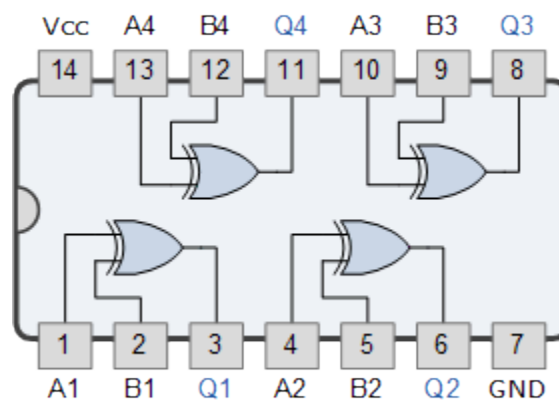
TTL Logic Ex-OR Gates

- 74LS86 Quad 2-input

CMOS Logic Ex-OR Gates

- CD4030 Quad 2-input

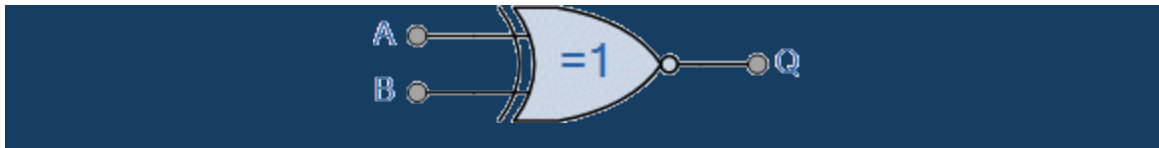
7486 Quad 2-input Exclusive-OR Gate



The **Exclusive-OR** logic function is a very useful circuit that can be used in many different types of computational circuits. Although not a basic logic gate in its own right, its usefulness and versatility has turned

it into a standard logical function complete with its own Boolean expression, operator and symbol. The *Exclusive-OR Gate* is widely available as a standard quad two-input 74LS86 TTL gate or the 4030B CMOS package.

One of its most commonly used applications is as a basic logic comparator which produces a logic “1” output when its two input bits are not equal. Because of this, the exclusive-OR gate has an inequality status being known as an odd function. In order to compare numbers that contain two or more bits, additional exclusive-OR gates are needed with the 74LS85 logic comparator being 4-bits wide.



Exclusive-NOR Gate Tutorial

The Exclusive-NOR Gate function is a digital logic gate that is the reverse or complementary form of the Exclusive-OR function

Basically the “Exclusive-NOR” gate is a combination of the Exclusive-OR gate and the NOT gate but has a truth table similar to the standard NOR gate in that it has an output that is normally at logic level “1” and goes “LOW” to logic level “0” when **ANY** of its inputs are at logic level “1”.

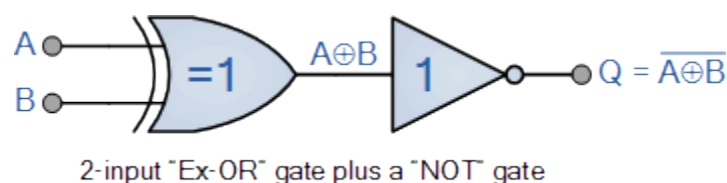
However, an output “1” is only obtained if **BOTH** of its inputs are at the same logic level, either binary “1” or “0”. For example, “00” or “11”. This input combination would then give us the Boolean expression
of: $Q = (A \oplus B) = A.B + A.B$

Then the output of a digital logic Exclusive-NOR gate **ONLY** goes “HIGH” when its two input terminals, A and B are at the “**SAME**” logic level which can be either at a logic level “1” or at a logic level “0”. In other words, an even number of logic “1’s” on its inputs gives a logic “1” at the output, otherwise is at logic level “0”.

Then this type of gate gives an output “1” when its inputs are “*logically equal*” or “*equivalent*” to each other, which is why an **Exclusive-NOR** gate is sometimes called an **Equivalence Gate**.

The logic symbol for an Exclusive-NOR gate is simply an Exclusive-OR gate with a circle or “inversion bubble”, (o) at its output to represent the NOT function. Then the **Logic Exclusive-NOR Gate** is the reverse or “*Complementary*” form of the Exclusive-OR gate, $(A \oplus B)$ we have seen previously.

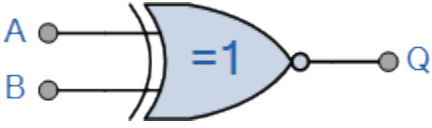
Ex-NOR Gate Equivalent



The **Exclusive-NOR Gate**, also written as: “Ex-NOR” or “XNOR”, function is achieved by combining standard gates together to form more complex gate functions and an example of a 2-input Exclusive-NOR gate is given below.

The Digital Logic “Ex-NOR” Gate

2-input Ex-NOR Gate

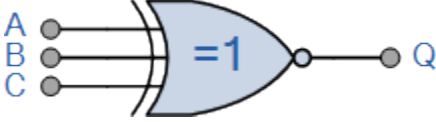
Symbol	Truth Table		
 <p>2-input Ex-NOR Gate</p>	B	A	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	1
Boolean Expression $Q = A \oplus B$	Read if A AND B the SAME gives Q		

Giving the Boolean expression of: $Q = AB + \overline{A}\overline{B}$

The logic function implemented by a 2-input Ex-NOR gate is given as “when both A AND B are the SAME” will give an output at Q. In general, an Exclusive-NOR gate will give an output value of logic “1” ONLY when there are an **EVEN** number of 1’s on the inputs to the gate (the inverse of the Ex-OR gate) except when all its inputs are “LOW”.

Then an Ex-NOR function with more than two inputs is called an “even function” or modulo-2-sum (Mod-2-SUM), not an Ex-NOR. This description can be expanded to apply to any number of individual inputs as shown below for a 3-input Exclusive-NOR gate.

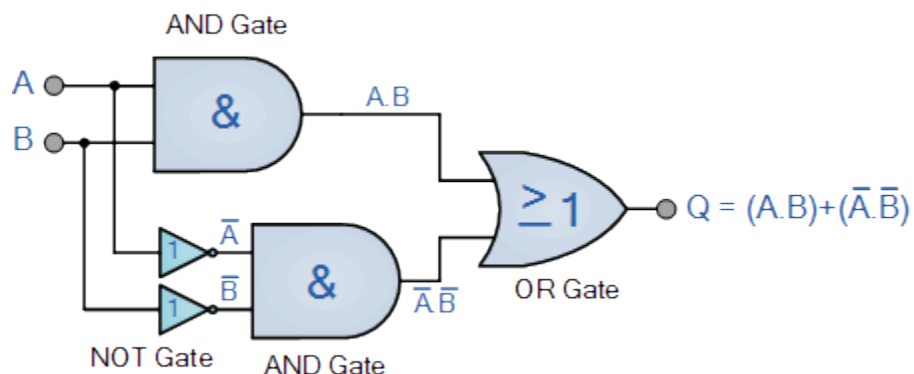
3-input Ex-NOR Gate

Symbol	Truth Table			
 <p>3-input Ex-NOR Gate</p>	C	B	A	Q
	0	0	0	1
	0	0	1	0
	0	1	0	0
	0	1	1	1
	1	0	0	0
	1	0	1	1
	1	1	0	1
	1	1	1	0
Boolean Expression $Q = A \oplus B \oplus C$	Read as “any EVEN number of Inputs” gives Q			

Giving the Boolean expression of: $Q = ABC + ABC + ABC + ABC$

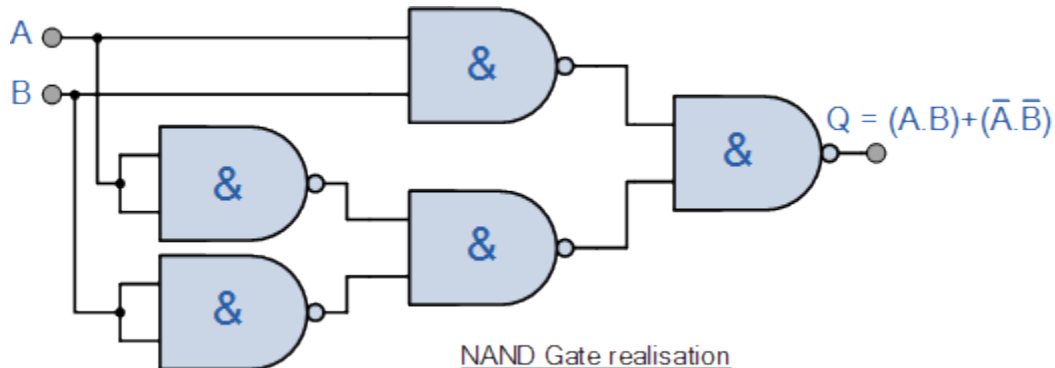
We said previously that the Ex-NOR function is a combination of different basic logic gates Ex-OR and a NOT gate, and by using the 2-input truth table above, we can expand the Ex-NOR function to: $Q = A \oplus B = (A.B) + (\bar{A}.\bar{B})$ which means we can realise this new expression using the following individual gates.

Ex-NOR Gate Equivalent Circuit



One of the main disadvantages of implementing the Ex-NOR function above is that it contains three different types logic gates the AND, NOT and finally an OR gate within its basic design. One easier way of producing the Ex-NOR function from a single gate type is to use NAND gates as shown below.

Ex-NOR Function Realisation using NAND gates



Ex-NOR gates are used mainly in electronic circuits that perform arithmetic operations and data checking such as *Adders*, *Subtractors* or *Parity Checkers*, etc. As the Ex-NOR gate gives an output of logic level “1” whenever its two inputs are equal it can be used to compare the magnitude of two binary digits or numbers and so Ex-NOR gates are used in Digital Comparator circuits.

Commonly available digital logic Exclusive-NOR gate IC's include:

TTL Logic Ex-NOR Gates

- 74LS266 Quad 2-input

CMOS Logic Ex-NOR Gates

- CD4077 Quad 2-input

74266 Quad 2-input Ex-NOR Gate

