

# Universal Logic Gates

Universal Logic gates can be used to produce any other logic or Boolean function with the NAND and NOR gates being minimal

Individual logic gates can be connected together to form a variety of different switching functions and combinational logic circuits. As we have seen through this *Digital Logic* tutorial section, the three most basic logic gates are the: AND, OR and NOT gates, and given this set of logic gates it is possible to implement all of the possible Boolean switching functions, thus making them a “full set” of **Universal Logic Gates**.

By using logical sets in this way, the various laws and theorems of Boolean Algebra can be implemented with a complete set of logic gates. In fact, it is possible to produce every other Boolean function using just the set of AND and NOT gates since the OR function can be created using just these two gates. Likewise, the set of OR and NOT can be used to create the AND function.

Any logic gate which can be combined into a set to realise all other logical functions is said to be a universal gate with a complete logic set being a group of gates that can be used to form any other logic function.

For example, AND and NOT constitute a complete set of logic, as does OR and NOT as cascading together an AND with a NOT gate would give us a NAND gate. Similarly cascading an OR and NOT gate together will produce a NOR gate, and so on. However, the two functions of AND and OR on their own do not form a complete logic set.

So by using these three *Universal Logic Gates* we can create a range of other Boolean functions and gates. However, the NAND and NOR gates are classed as minimal sets because they have the property of being a complete set in themselves since they can be used individually or together to construct many other logic circuits. Therefore we can define the complete sets of operations of the main logic gates as follows:

- AND, OR and NOT (a Full Set)
- AND and NOT (a Complete Set)
- OR and NOT (a Complete Set)
- NAND (a Minimal Set)
- NOR (a Minimal Set)

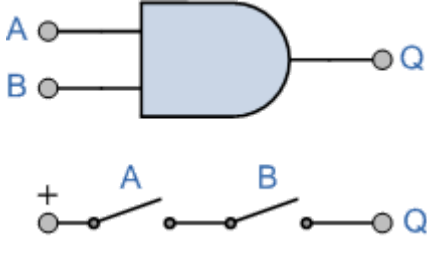
Thus we can use these five sets of gates, together or individually as the building blocks to produce more complex logic circuits called *combinational logic circuits*. But first let us remind ourselves of the switching characteristics of the three basic logic gates, AND, OR and NOT.

## The AND Function

In mathematics, the number or quantity obtained by multiplying two (or more) numbers together is called the *product*. In Boolean Algebra the AND function is the equivalent of multiplication and so its output state represents the product of its inputs. The AND function is represented in Boolean Algebra by a single “dot” (.) so for a two input AND gate the Boolean equation is given as:  $Q = A.B$ , that is Q equals both A AND B.

## The 2-input Logic AND Gate

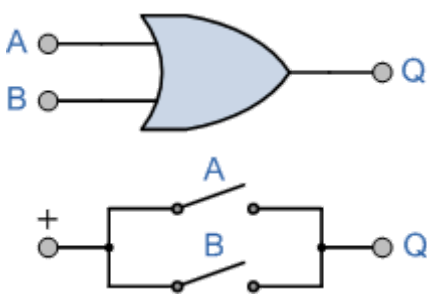
Symbol	Truth Table		
	B	A	Q
	0	0	0

	0	1	0
	1	0	0
	1	1	1

The OR Function

In mathematics, the number or quantity obtained by adding two (or more) numbers together is called the *sum*. In Boolean Algebra the OR function is the equivalent of addition so its output state represents the addition of its inputs. In Boolean Algebra the OR function is represented by a “plus” sign (+) so for a two input OR gate the Boolean equation is given as:  $Q = A+B$ , that is Q equals either A OR B.

The 2-input Logic OR Gate

Symbol	Truth Table		
	B	A	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1

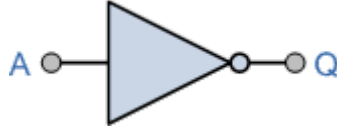
The NOT Function

The NOT gate, which is also known as an “inverter” is given a symbol whose shape is that of a triangle pointing to the right with a circle at its end. This circle is known as an “inversion bubble”.

The NOT function is not a decision making logic gate like the AND, or OR gates, but instead is used to invert or complement a digital signal. In other words, its output state will always be the opposite of its input state.

The NOT gate symbol has a single input and a single output as shown.

The Logic NOT Gate

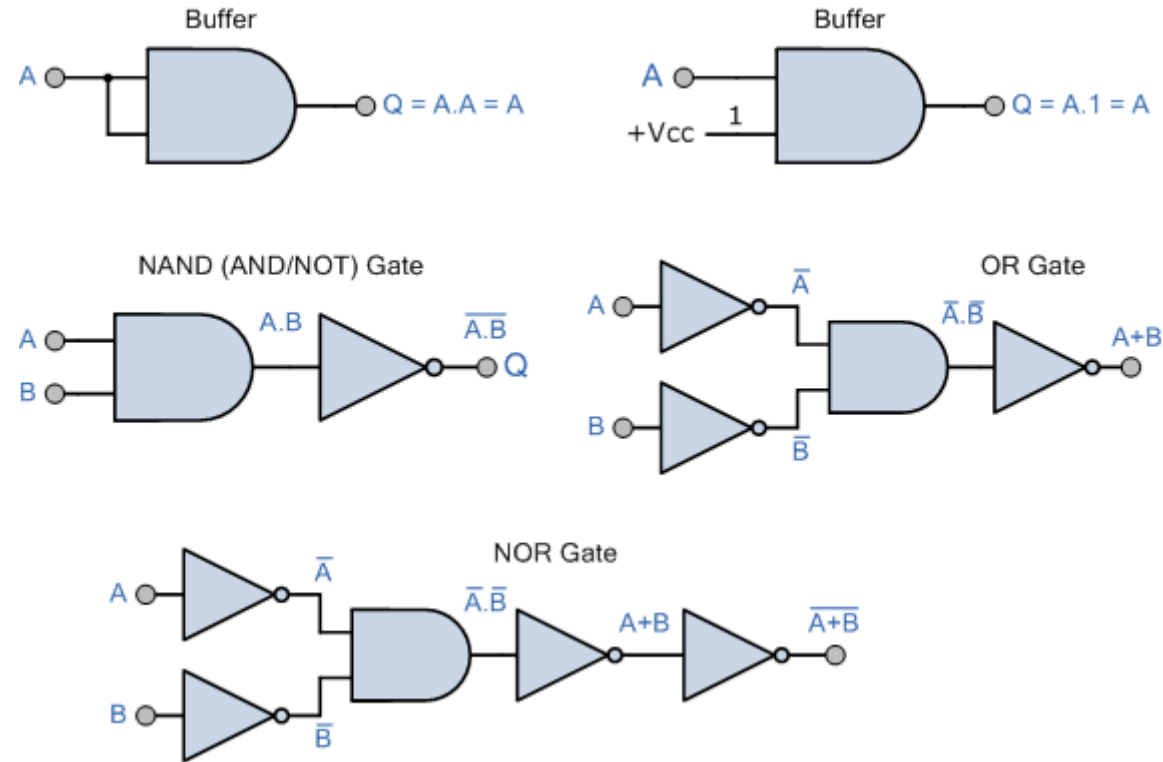
Symbol	Truth Table	
	A	Q
	0	1
	1	0

The single input NOT gate or invert function can be cascaded with itself to produce what is called a digital buffer. The first NOT gate will invert the input and the second will re-invert it back to its original level performing a double inversion of the single input. Non-inverting Digital Buffers have many uses in digital electronics as this double inversion of the input can be used to provide digital amplification and circuit isolation.

Using the AND and NOT Set

Using just the AND and NOT set of logic gates we can create the following Boolean functions and equivalent gates.

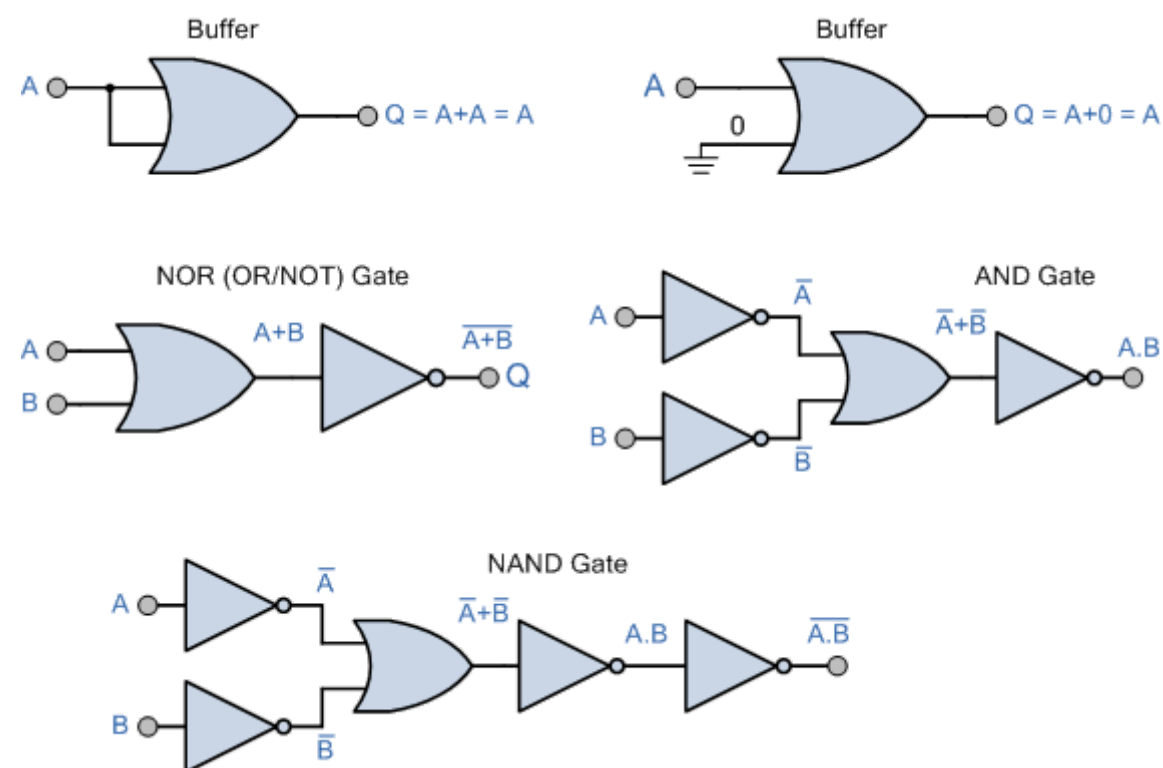
AND/NOT Set Equivalents



## Using the OR and NOT Set

Using the OR and NOT set of logic gates we can create the following Boolean functions and equivalent gates.

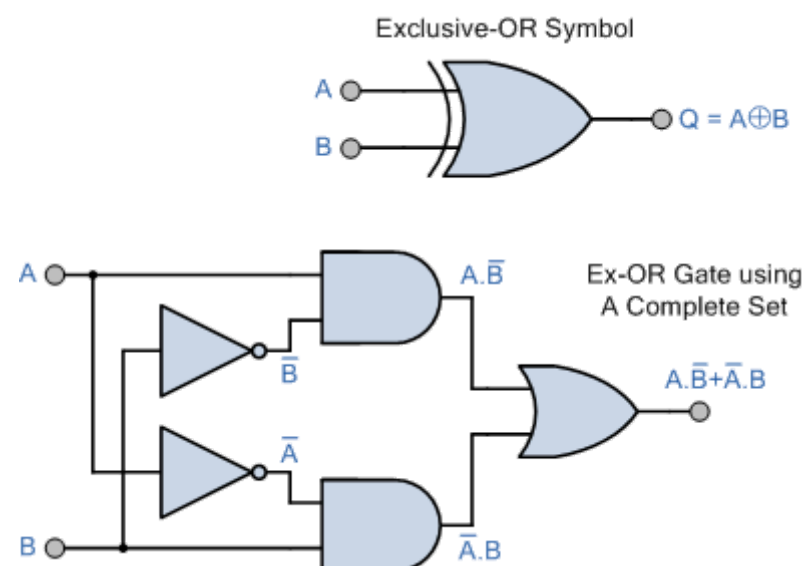
## OR/NOT Set Equivalents



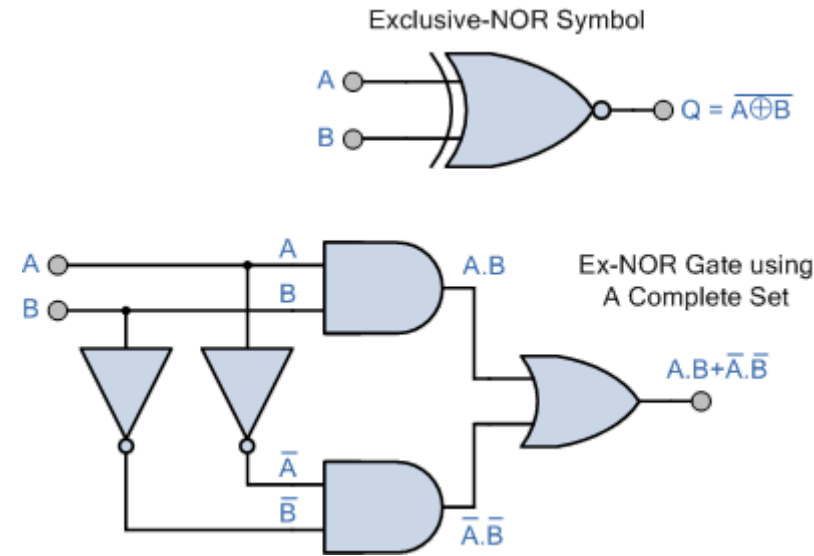
## Using the Full AND, OR and NOT Set

Using the full AND, OR and NOT set of logic gates we can create the Boolean expressions for the Exclusive-OR (Ex-OR) and the NOT Exclusive-OR (Ex-NOR) gates as shown.

## Full AND/OR/NOT Set to Implement Ex-OR



## Full AND/OR/NOT Set to Implement Ex-NOR



Note that neither the Exclusive-OR gate or the Exclusive-NOR gate can be classed as a universal logic gate as they can not be used on their own or together to produce any other Boolean function.

## Universal Logic Gates

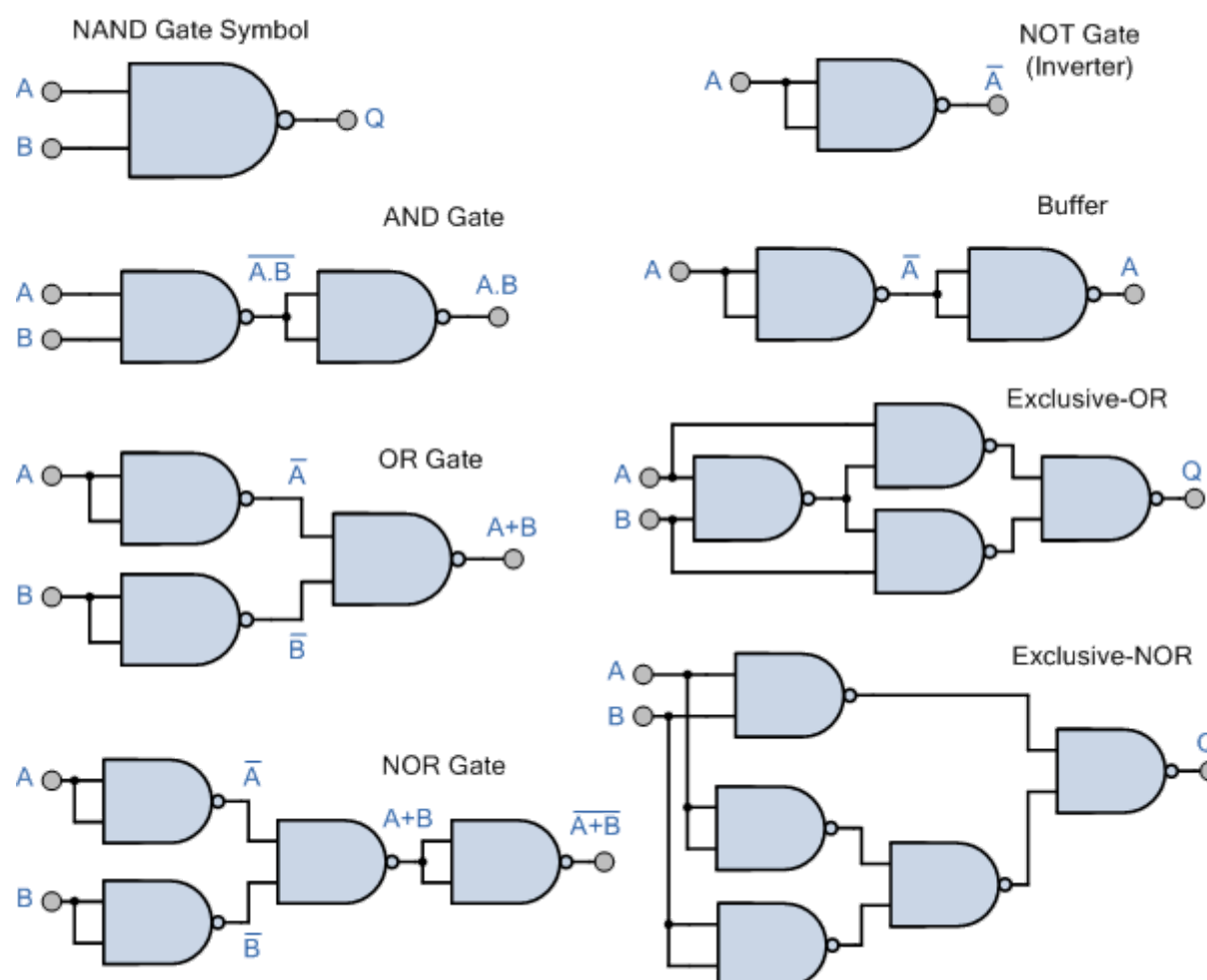
One of the main disadvantages of using the complete sets of AND, OR and NOT gates is that to produce any equivalent logic gate or function we require two (or more) different types of logic gate, AND and NOT, or OR and NOT, or all three as shown above. However, we can realise all of the other Boolean functions and gates by using just one single type of universal logic gate, the NAND (NOT AND) or the NOR (NOT OR) gate, thereby reducing the number of different types of logic gates required, and also the cost.

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**.

## 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

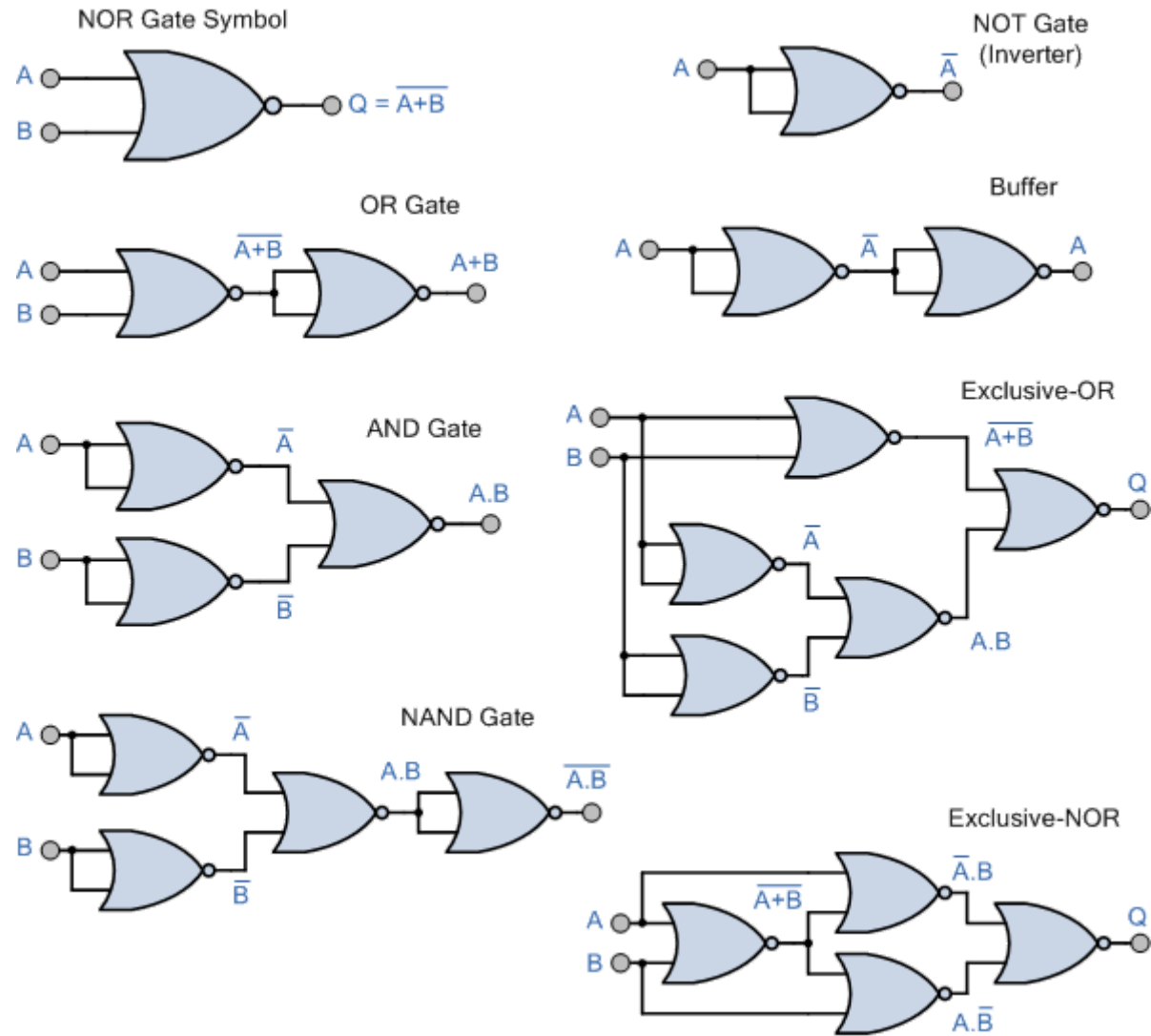


Thus ALL other logic gate functions can be created using only NAND gates making it a universal logic gate.

## 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.

## 21 Comments

Join the conversation

Your Name	Email Address
Write your comment here	

☐ Notify me of follow-up comments by email.

SUBMIT