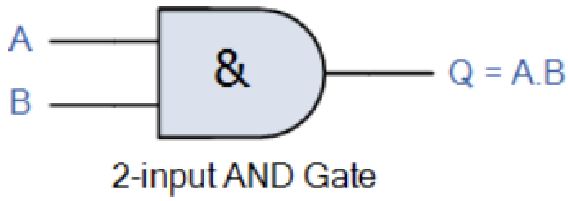


Boolean Algebra Truth Tables

Boolean Algebra Expressions can be used to construct digital logic truth tables for their respective functions

As well as a standard Boolean Expression, the input and output information of any **Logic Gate** or circuit can be plotted into standard Boolean Algebra truth tables to give a visual representation of the switching function of the system.

The table used to represent the boolean expression of a logic gate function is commonly called a **Truth Table**. A logic gate truth table shows each possible input combination to the gate or circuit with the resultant output depending upon the combination of these input(s).

For example, consider a single **2-input** logic circuit with input variables labelled as A and B. There are "four" possible input combinations or 2^2 of "OFF" and "ON" for the two inputs. However, when dealing with Boolean expressions and especially logic gate truth tables, we do not general use "ON" or "OFF" but instead give them bit values which represent a logic level "1" or a logic level "0" respectively.

Then the four possible combinations of A and B for a 2-input logic gate is given as:

Input Combination 1. – "OFF" – "OFF" or (0, 0)

Input Combination 2. – "OFF" – "ON" or (0, 1)

Input Combination 3. – "ON" – "OFF" or (1, 0)

Input Combination 4. – "ON" – "ON" or (1, 1)

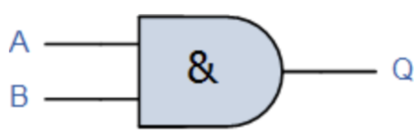
Therefore, a 3-input logic circuit would have 8 possible input combinations or 2^3 and a 4-input logic circuit would have 16 or 2^4, and so on as the number of inputs increases. Then a logic circuit with "n" number of inputs would have 2^n possible input combinations of both "OFF" and "ON".

So in order to keep things simple to understand, in this tutorial we will only deal with standard **2-input** type logic gates, but the principals are still the same for gates with more than two inputs.

Then the Truth tables for a 2-input AND Gate, a 2-input OR Gate and a single input NOT Gate are given as:

Boolean Algebra Truth Tables For A 2-input AND Gate

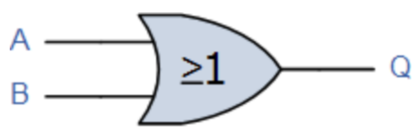
For a 2-input AND gate, the output Q is true if BOTH input A "AND" input B are both true, giving the Boolean Expression of: (Q = A and B).

<table>
<tr><th>Symbol</th><th colspan="3">Truth Table</th></tr>
<tr><td rowspan="5">2-input AND Gate</td><td>A</td><td>B</td><td>Q</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>Boolean Expression Q = A.B</td><td colspan="3">Read as A AND B gives Q</td></tr>
</table>

Note that the Boolean Expression for a two input AND gate can be written as: A.B or just simply AB without the decimal point.

Boolean Algebra Truth Tables For A 2-input OR Gate

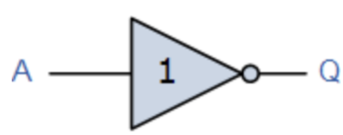
For a 2-input OR (Inclusive-OR) gate, the output Q is true if EITHER input A "OR" input B is true, giving the Boolean Expression of: (Q = A or B).

<table>
<tr><th>Symbol</th><th colspan="3">Truth Table</th></tr>
<tr><td rowspan="5">

2-input OR Gate</td><td>A</td><td>B</td><td>Q</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>Boolean Expression Q = A+B</td><td colspan="3">Read as A OR B gives Q</td></tr>
</table>

Boolean Algebra Truth Tables For The NOT Gate

For a single input NOT (Inverter) gate, the output Q is ONLY true when the input is "NOT" true, the output is the inverse or complement of the input giving the Boolean Expression of: (Q = NOT A).

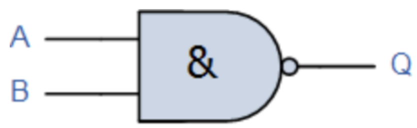
<table>
<tr><th>Symbol</th><th colspan="2">Truth Table</th></tr>
<tr><td rowspan="3">

Inverter or NOT Gate</td><td>A</td><td>Q</td></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>Boolean Expression Q = NOT A or $\overline{A}$</td><td colspan="2">Read as inversion of A gives Q</td></tr>
</table>

The NAND and the NOR Gates are a combination of the AND and OR Gates respectively with that of a NOT Gate (inverter).

2-input NAND (Not AND) Gate

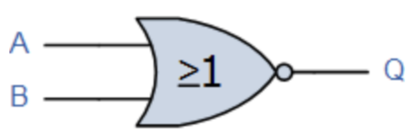
For a 2-input NAND gate, the output Q is NOT true if BOTH input A and input B are true, giving the Boolean Expression of: (Q = not(A AND B)).

<table>
<tr><th>Symbol</th><th colspan="3">Truth Table</th></tr>
<tr><td rowspan="5">

2-input NAND Gate</td><td>A</td><td>B</td><td>Q</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>Boolean Expression $Q = \overline{A.B}$</td><td colspan="3">Read as A AND B gives NOT-Q</td></tr>
</table>

2-input NOR (Not OR) Gate

For a 2-input NOR gate, the output Q is true if BOTH input A and input B are NOT true, giving the Boolean Expression of: (Q = not(A OR B)).

<table>
<tr><th>Symbol</th><th colspan="3">Truth Table</th></tr>
<tr><td rowspan="5">

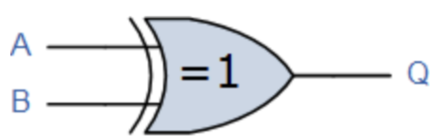
2-input NOR Gate</td><td>A</td><td>B</td><td>Q</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>Boolean Expression $Q = \overline{A+B}$</td><td colspan="3">Read as A OR B gives NOT-Q</td></tr>
</table>

As well as the standard logic gates there are also two special types of logic gate function called an Exclusive-OR Gate and an Exclusive-NOR Gate. The Boolean expression to indicate an Exclusive-OR or Exclusive-NOR function is to a symbol with a plus sign inside a circle, (⊕).

The switching actions of both of these types of gates can be created using the above standard logic gates. However, as they are widely used functions they are now available in standard IC form and have been included here as reference.

2-input EX-OR (Exclusive OR) Gate

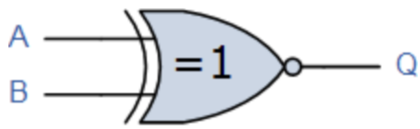
For a 2-input Ex-OR gate, the output Q is true if EITHER input A or if input B is true, but NOT both giving the Boolean Expression of: (Q = (A and NOT B) or (NOT A and B)).

<table>
<tr><th>Symbol</th><th colspan="3">Truth Table</th></tr>
<tr><td rowspan="5">

2-input Ex-OR Gate</td><td>A</td><td>B</td><td>Q</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>Boolean Expression Q = A ⊕ B</td><td colspan="3"></td></tr>
</table>

2-input EX-NOR (Exclusive NOR) Gate

For a 2-input Ex-NOR gate, the output Q is true if BOTH input A and input B are the same, either true or false, giving the Boolean Expression of: (Q = (A and B) or (NOT A and NOT B)).

Symbol	Truth Table		
	A	B	Q
	0	0	1
2-input Ex-NOR Gate	0	1	0
	1	0	0
	1	1	1
Boolean Expression $Q = \overline{A \oplus B}$			

Summary of 2-input Logic Gates

The following Boolean ALgebra Truth Tables compare the logical functions of the 2-input logic gates above.

Inputs		Truth Table Outputs For Each Gate					
A	B	AND	NAND	OR	NOR	EX-OR	EX-NOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

The following Boolean Algebra truth tables gives a list of the common logic functions and their equivalent Boolean notation.

Logic Function	Boolean Notation
AND	A.B
OR	A+B
NOT	$\overline{A}$
NAND	$\overline{A.B}$
NOR	$\overline{A+B}$
EX-OR	$(A.\overline{B}) + (\overline{A}.B)$ or $A \oplus B$
EX-NOR	$(A.B) + (\overline{A}.\overline{B})$ or $\overline{A \oplus B}$

2-input logic gate truth tables are given here as examples of the operation of each logic function, but there are many more logic gates with 3, 4 even 8 individual inputs. The multiple input gates are no different to the simple 2-input gates above, So a 4-input AND gate would still require ALL 4-inputs to be present to produce the required output at Q and its larger truth table would reflect that.