- A logic gate is a device that acts as a building block for digital circuits.
- They perform basic logical functions that are fundamental to digital circuits.
- Most electronic devices we use today will have some form of logic gates in them. For example, logic gates can be used in technologies such as smartphones, tablets or within memory devices.
-
- In a circuit, logic gates will make decisions based on a combination of digital signals coming from its inputs. Most logic gates have two inputs and one output.
- Logic gates are based on Boolean algebra.
- At any given moment, every terminal is in one of the two binary conditions, false or true. 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).

Basic logic gates

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

Basic Gates

AND Gate

OR Gate

NOT Gate

Universal Gates

NAND Gate

NOR Gate

Special Purpose Gates

XOR Gate

XNOR Gate

AND Gate

An AND gate is a logic circuit that consists of two or more inputs and only one output.

In an AND gate, the output is high if and only if all the inputs to the gate are high. Otherwise, the output will be low.

Here, high indicates 1 and low indicates 0. In a digital circuit, we only talk about 0 and 1.

Symbol of the AND Gate

Here is the symbol for the two-input AND gate:

Here, A and B are the two inputs, and Y is the output.

AND Gate Boolean Equation

Here is the boolean equation of a two-input AND gate:

$$Y = A.B$$

AND Gate Truth Table

The table given below represents or shows the truth table of a two-input AND gate:

Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

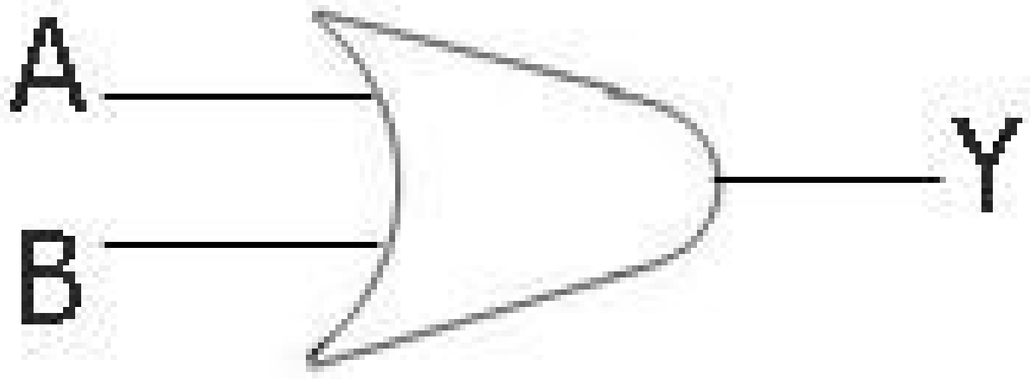
OR Gate

An OR gate is a logic circuit with two or more inputs and only one output.

The function of an OR gate can be expressed as follows: if either of the two inputs is true, then the output will be true; otherwise, the output will be false. Here, true means 1 and false means 0.

Symbol of the OR Gate

Below is the symbol for the two-input OR gate:



Boolean Expression for OR Gate

The boolean expression of the two-input OR gate will be:

$$Y = A + B$$

OR Gate Truth Table

The table given below shows the truth table of an OR gate:

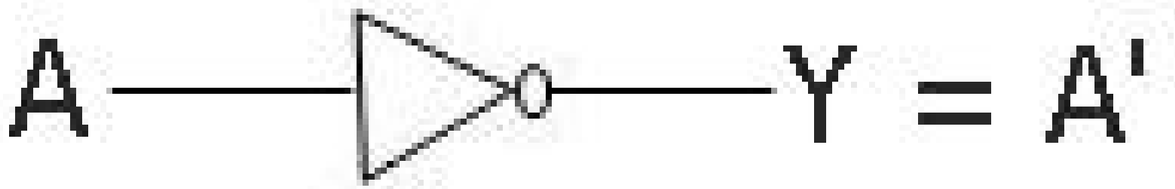
Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate

A NOT gate is used to reverse the input, which means that when the input is true, the output will not be true, implying that the output is false.

Symbol of the NOT Gate

Here is the symbol used for the NOT gate:



NOT Gate Boolean Expression

The boolean expression of the NOT gate is:

$$Y = A'$$

The Truth Table of the NOT Gate

The NOT gate's truth table is as follows:

Input	Output
A	Y
0	1
1	0

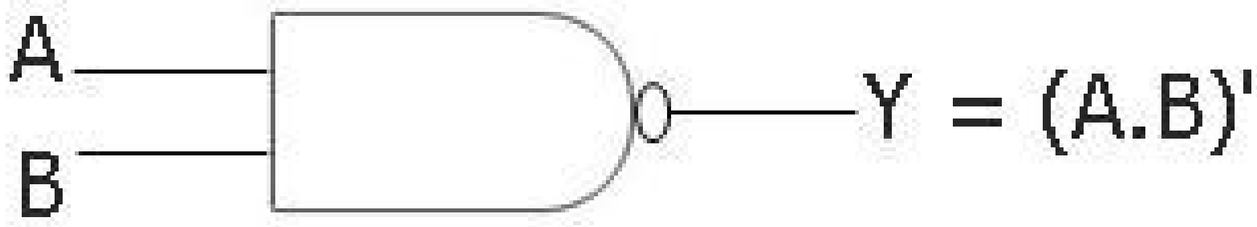
NAND Gate

The NAND gate, or NOT-AND gate, is made up of two gates. The first one is an AND gate followed by a NOT gate.

We can use the NAND gate whenever we want to reverse the value of the AND gate's output.

Symbol of the NAND Gate

The symbol of a two-input NAND gate is given below:



Boolean Expression of the NAND Gate

The boolean expression of a two-input NAND gate will be:

Y = (A.B)'

As we can see from the above expression, it reverses the value of the AND gate output's value.

Input		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

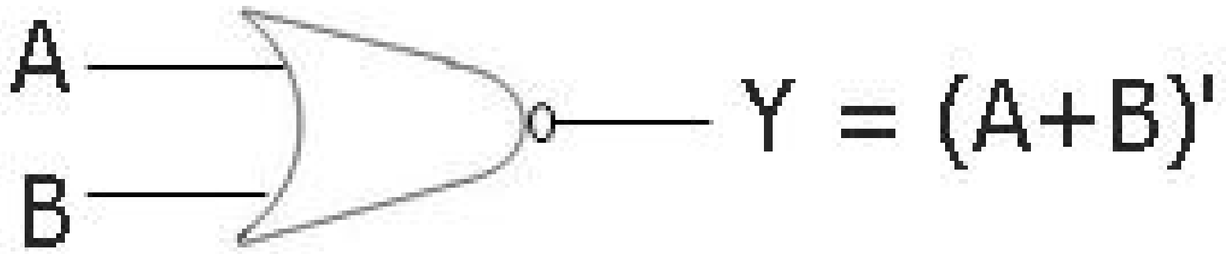
NOR Gate

The NOR gate is a little bit similar to the NAND gate. OR gate is used instead of AND gate in this case.Therefore, we can say that a NOR gate is a combination of an OR gate followed by a NOT gate.

This gate is used to reverse the value of the OR gate's output. It is abbreviated as the NOT-OR gate.

Symbol of the NOR Gate

The symbol used for a two-input NOR gate is:



Boolean Expression for NOR Gate

The boolean expression of a two-input NOR gate will be:

$$Y = (A+B)'$$

NOR Gate's Truth Table

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

XOR Gate

The XOR gate can also be called the EX-OR gate or EOR gate and is pronounced as the Exclusive OR gate.

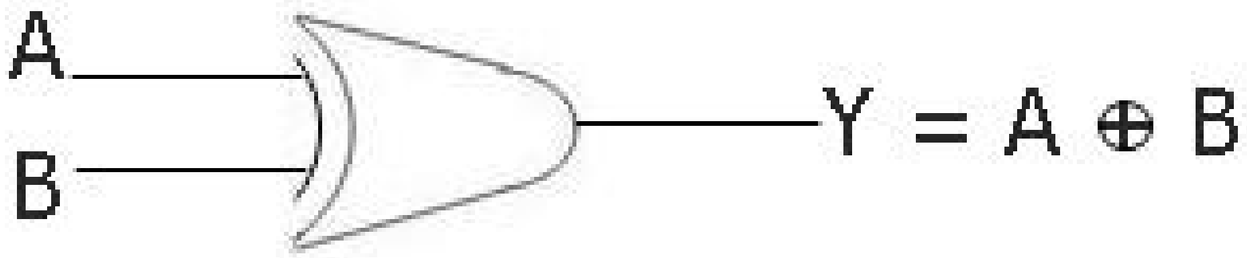
An exclusive OR gate can have two or more input terminals and only one output terminal.

The XOR Gate's symbol

Here is the symbol given for the two-input XOR gate:

$$x \oplus 1 = x'$$

$$x \oplus 0 = x$$



Boolean Expression of the XOR Gate

The boolean expression of a two-input XOR gate is:

$$Y = A \oplus B$$

The XOR Gate Truth Table

Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

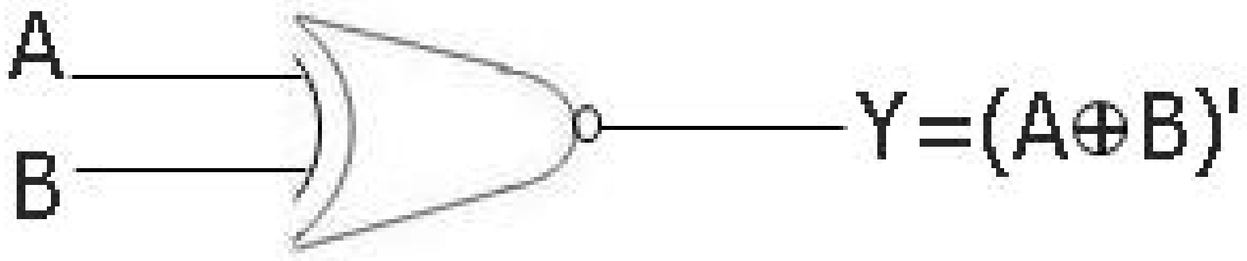
XNOR Gate

XNOR, EX-NOR, or ENOR gates are pronounced as Exclusive NOR or Exclusive NOT-OR gates. Or, to put it another way, exclusive NOR means NOT Exclusive OR. It means it is the reverse of an XOR gate.

This gate is used to reverse the value of the XOR gate.

XNOR Gate Symbol

Below is the symbol of a two-input XNOR gate:



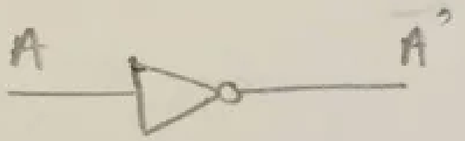
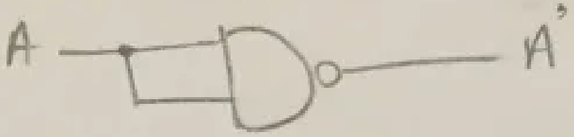
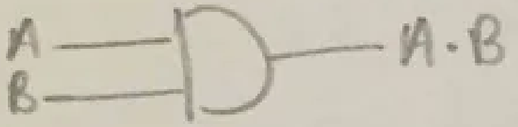
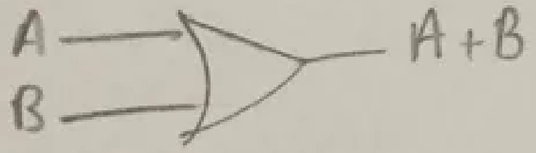
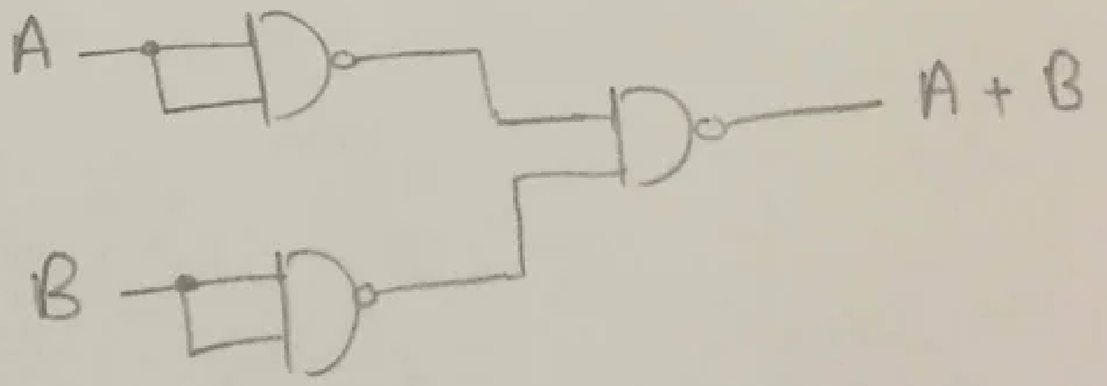
XNOR Gate Boolean Expression

Here is the boolean equation of a two-input XNOR gate:

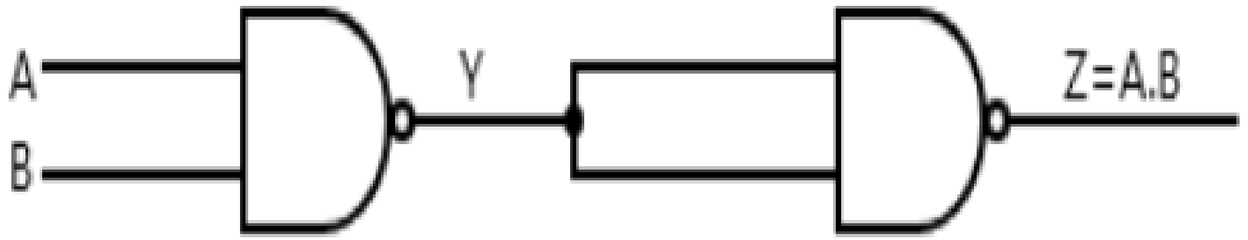
$$Y = (A \oplus B)'$$

The Truth Table of XNOR Gate

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Logic Gates	Symbol	Equivalence using NAND gates
NOT		
AND		
OR		

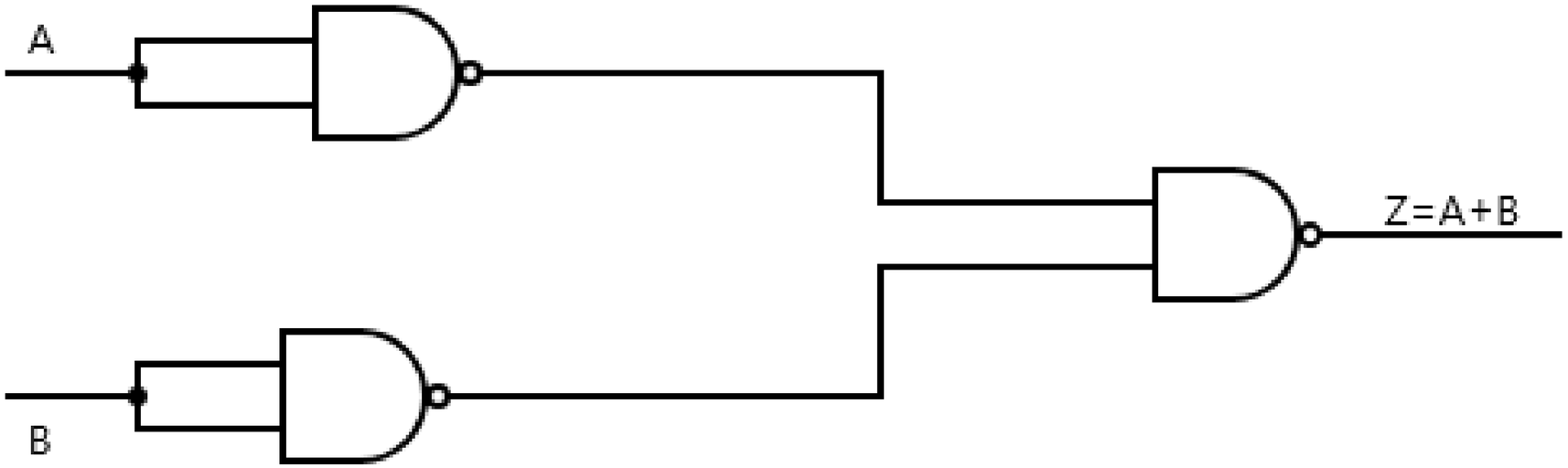
AND GATE USING NAND GATE



NAND = (AB)’

AND GATE = AB = ((AB)’)’

OR Gate using NAND GATE



NAND GATE = (AB)’

OR GATE = (A+B)

= (A’B’)’ = A+B

Implementation of XOR Gate from NAND Gate

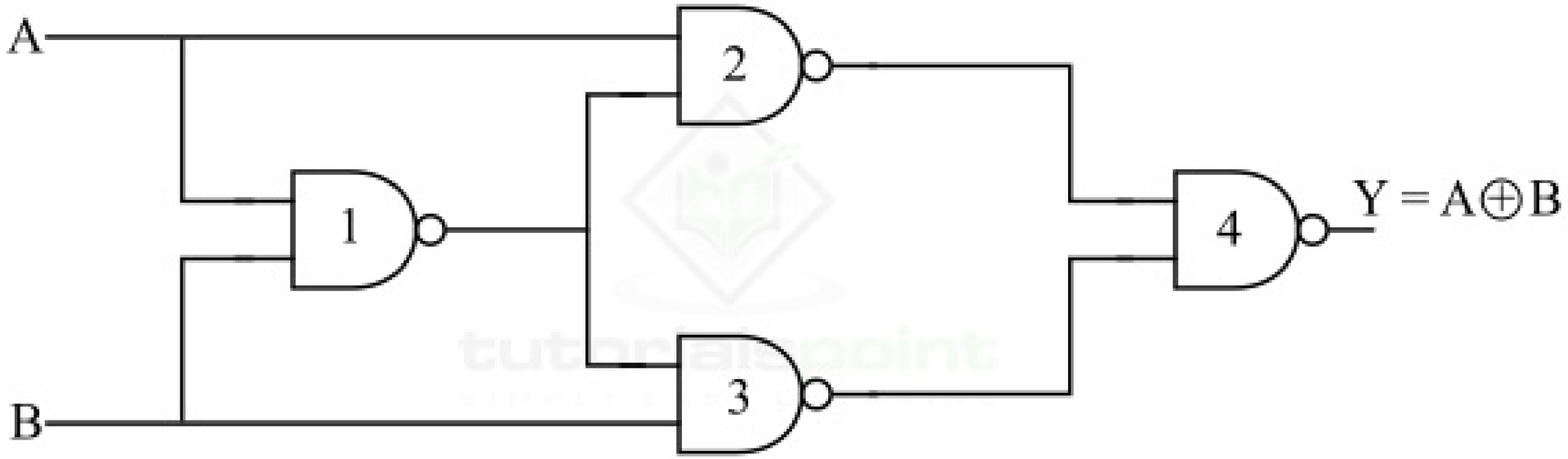


Figure 3 - XOR Gate using NAND Gates

The output of the first NAND gate is,

$$Y1 = (AB)'$$

The outputs of the secondary and third NAND gates are,

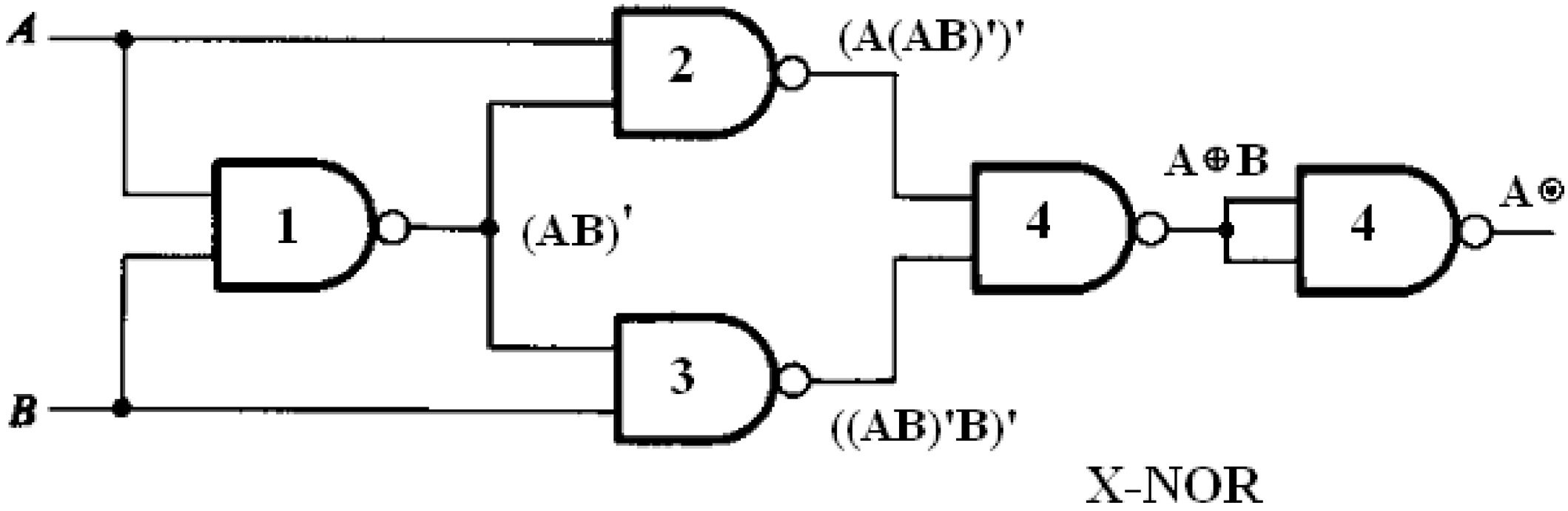
$$Y2 = (A.(AB)')'$$

$$Y3 = (B.(AB)')'$$

Finally, these two outputs (Y2 and Y3) are connected to the fourth NAND gate. This NAND gate will produce an output which is,Finally, these two outputs (Y2 and Y3) are connected to the fourth NAND gate. This NAND gate will produce an output which is,

$$Y = ((A.(AB)')'.(B.(AB)')')' = A.(AB)'+B.(AB)' = A.(A'+B')+B.(A'+B') = AB'+A'B = A \oplus B$$

NAND gates as Ex-NOR gate



Realization of logic function using NAND gates

Any logic function can be implemented using NAND gates.

To achieve this, first the logic function has to be written in Sum of Product (SOP) form.

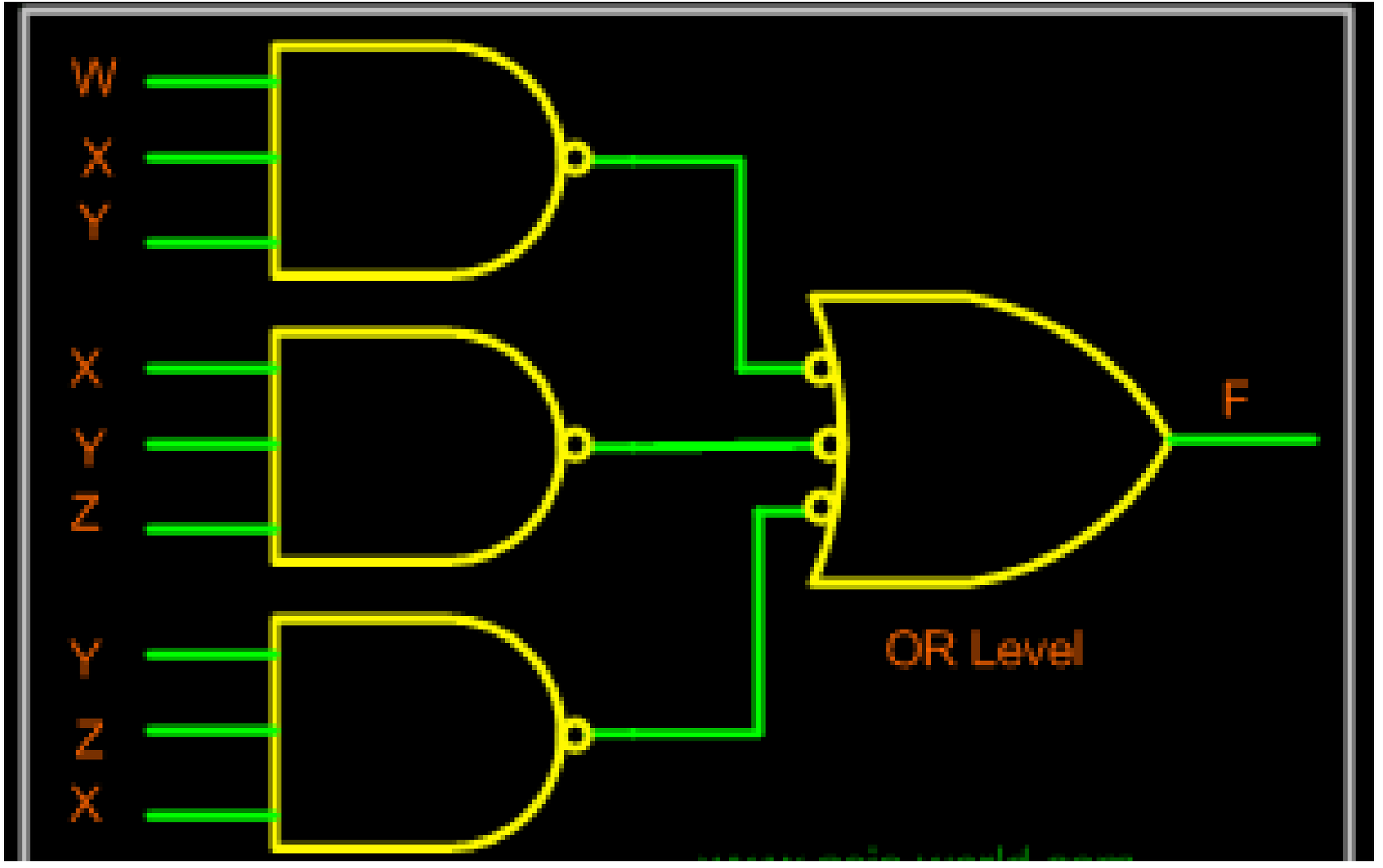
Once logic function is converted to SOP, then is very easy to implement using NAND gate.

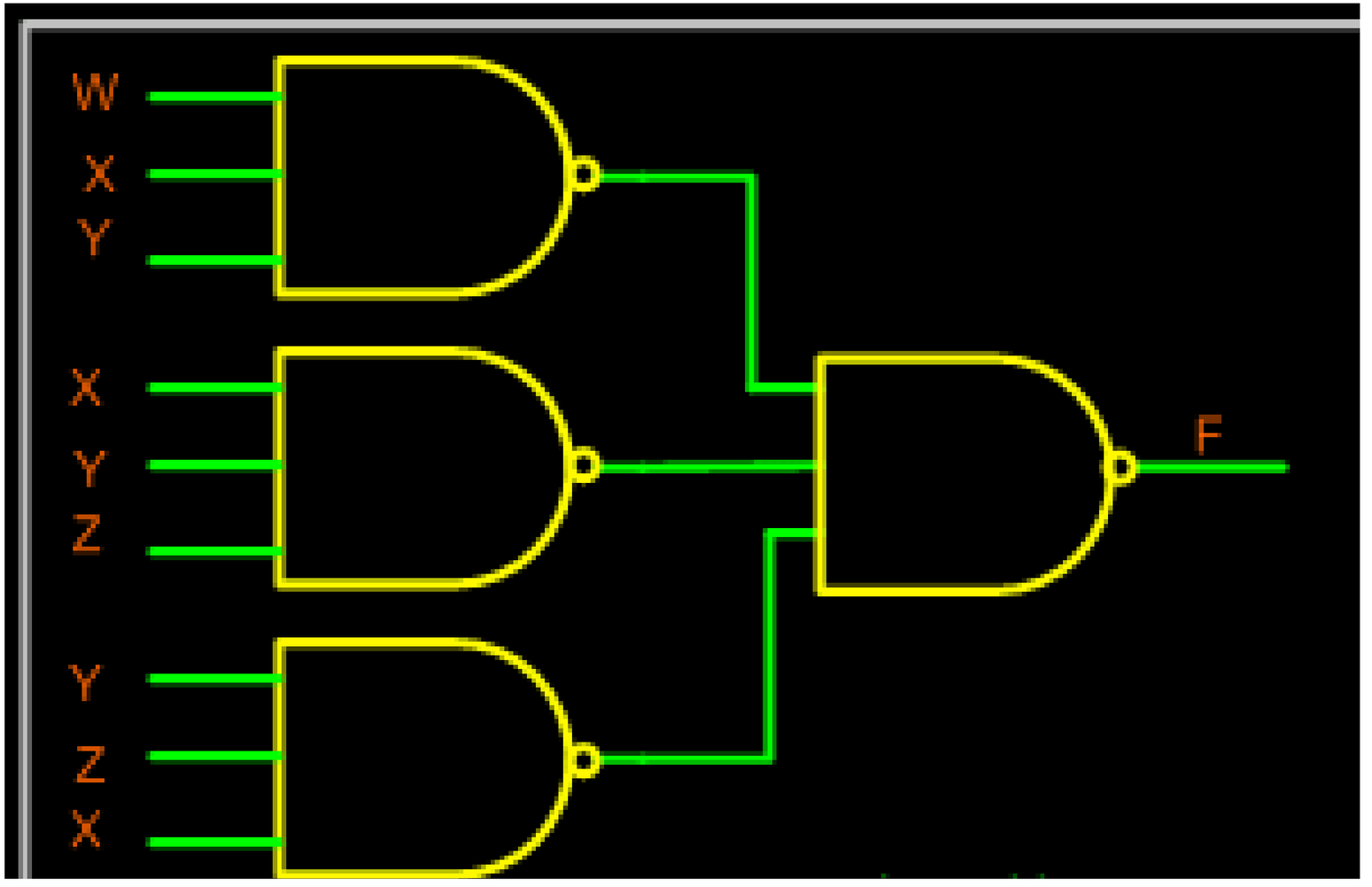
logic circuit with AND gates in first level and OR gates in second level can be converted into a NAND-NAND gate circuit.

NAND GATE IS EQUIVALENT TO BUBBLED OR GATE

Consider the following SOP expression

$$F = W.X.Y + X.Y.Z + Y.Z.W$$

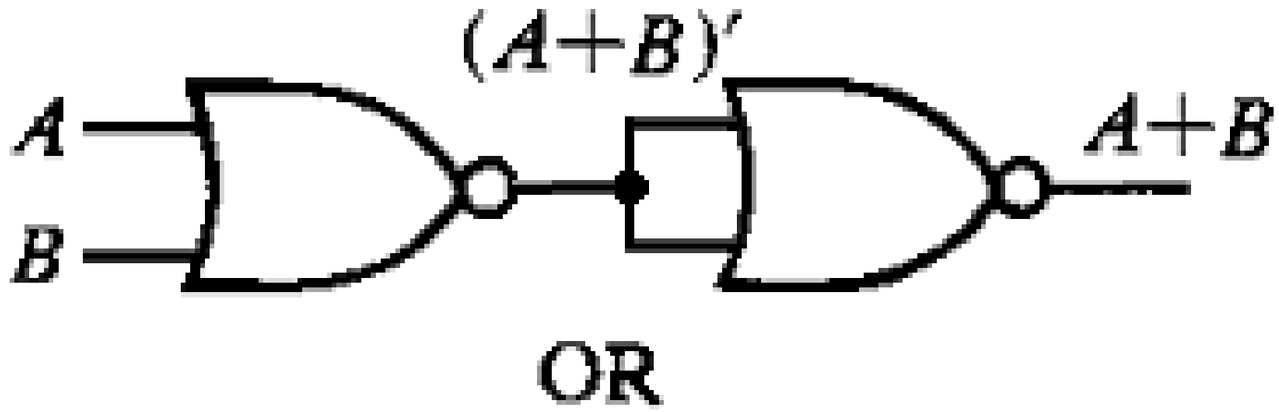

W
X
Y
X
Y
Z
Y
Z
X
F
OR Level


W
X
Y
X
Y
Z
Y
Z
X
F

Nor gate as Universal Gate

NOR gates as OR gate

- A NOR produces complement of OR gate. So, if the output of a NOR gate is inverted, overall output will be that of an OR gate.
- Y = ((A+B)')'
- Y = (A+B)



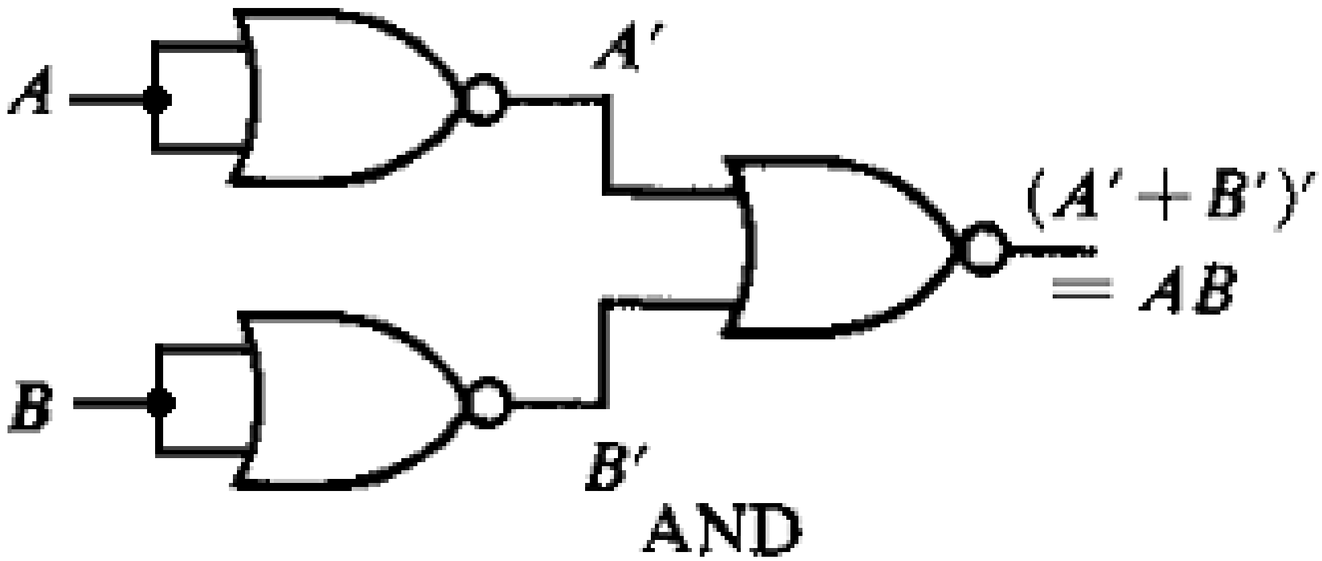
NOR gates as AND gate

From DeMorgan's theorems:

(A+B)' = A'B'

(A'+B')' = A''B'' = AB

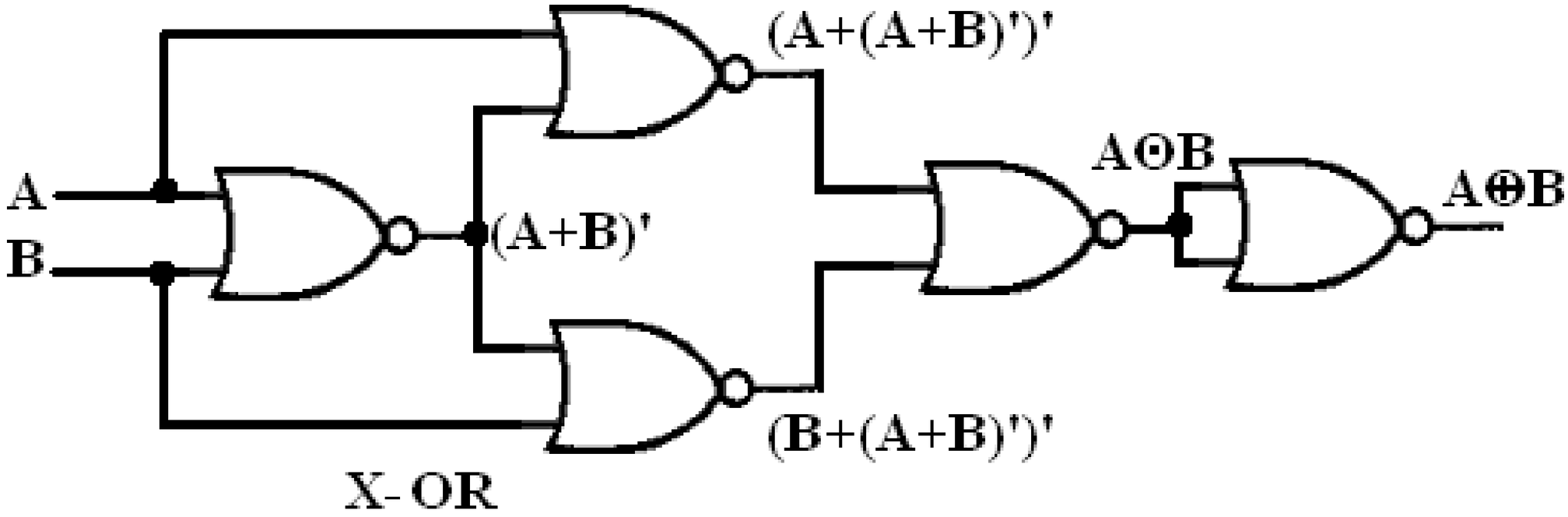
So, give the inverted inputs to a NOR gate, obtain AND operation at output.



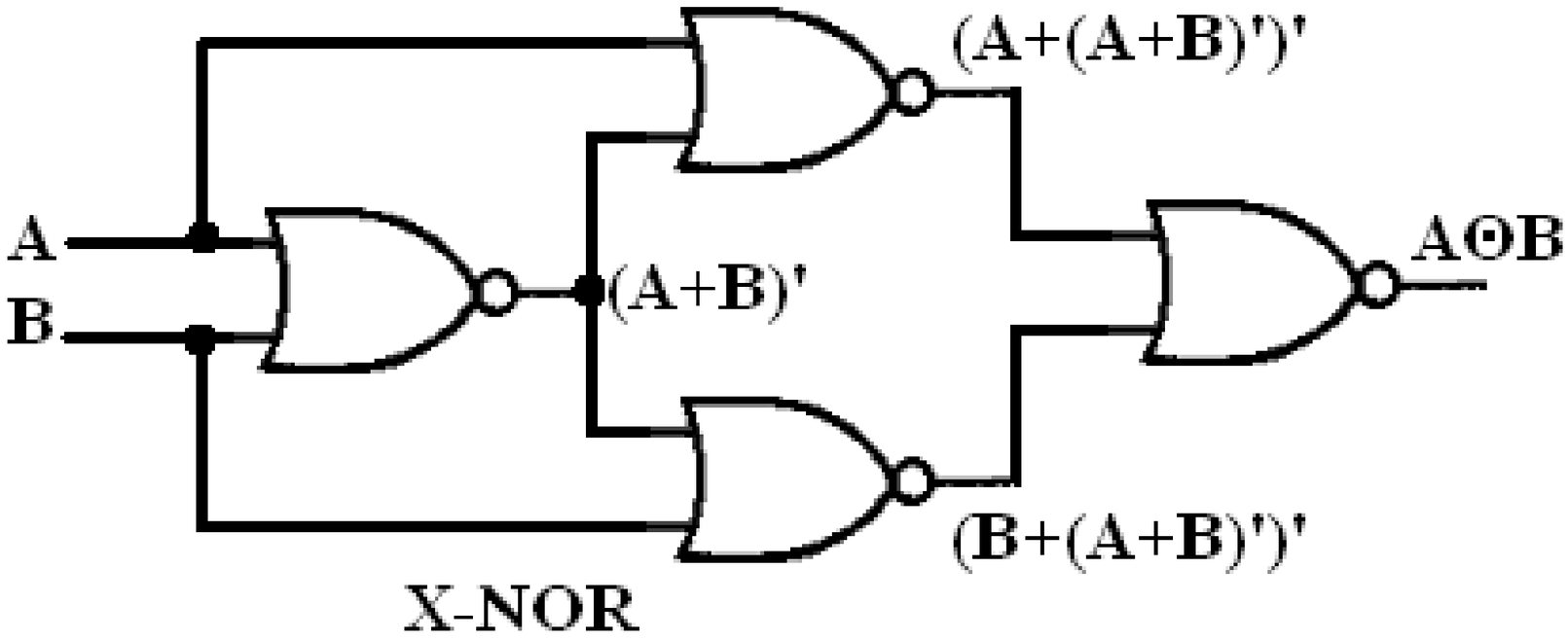
NOR gates as Ex-OR gate

Ex-OR gate is actually Ex-NOR gate followed by NOT gate. So give the output of Ex-NOR gate to a NOT gate, overall output is that of an Ex-OR gate.

Y = A’B+ AB’



NOR gates as Ex-NOR gate



Gate No.	Inputs	Output
1	A, B	$(A + B)'$
2	A, $(A + B)'$	$(A + (A+B)')'$
3	$(A + B)'$, B	$(B + (A+B)')'$
4	$(A + (A + B)')'$, $(B + (A+B)')'$	$AB + A'B'$

Now the ouput from gate no. 4is the overall output of the configuration.

$$
\begin{aligned}
Y &= ((A + (A+B)')' (B +(A+B)')')' \\
&= (A+(A+B)')''.(B+(A+B)')'' \\
&= (A+(A+B)').(B+(A+B)') \\
&= (A+A'B').(B+A'B') \\
&= (A + A').(A + B').(B+A')(B+B') \\
&= 1.(A+B').(B+A').1 \\
&= (A+B').(B+A') \\
&= A.(B + A') +B'.(B+A') \\
&= AB + AA' +B'B+B'A' \\
&= AB + 0 + 0 + B'A' \\
&= AB + B'A'
\end{aligned}
$$

$$
\Rightarrow \quad Y = AB + A'B'
$$

Realization of logic function using NOR gates

Any logic function can be implemented using NOR gates.

To achieve this, first the logic function has to be written in Product of Sum (POS) form.

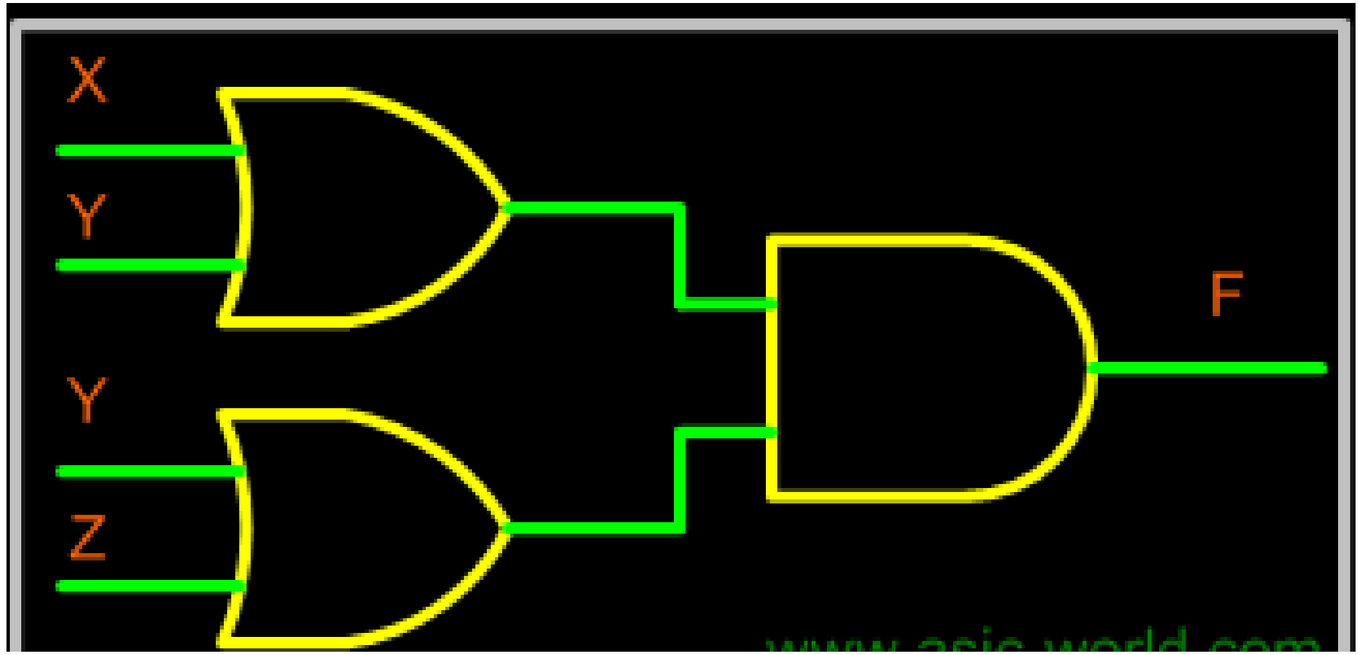
Once it is converted to POS, then it's very easy to implement using NOR gate.

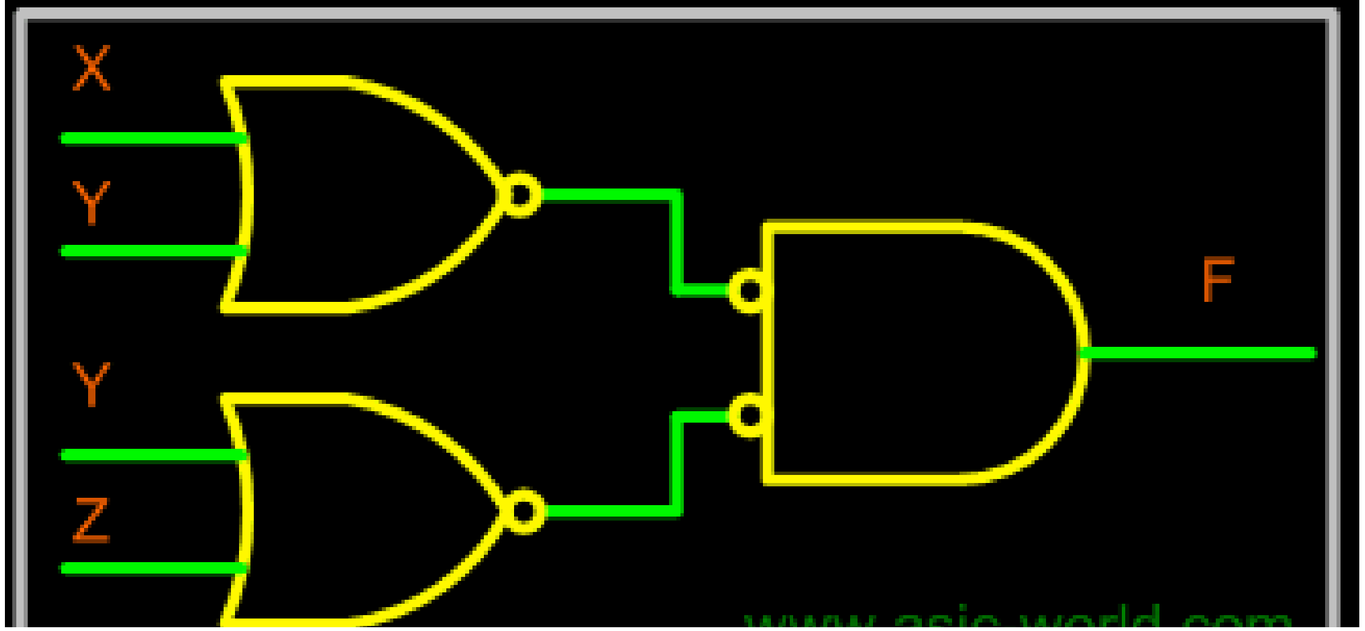
In other words any logic circuit with OR gates in first level and AND gates in second level can be converted into a NOR-NOR gate circuit.

NOR = BUBBLED AND GATE

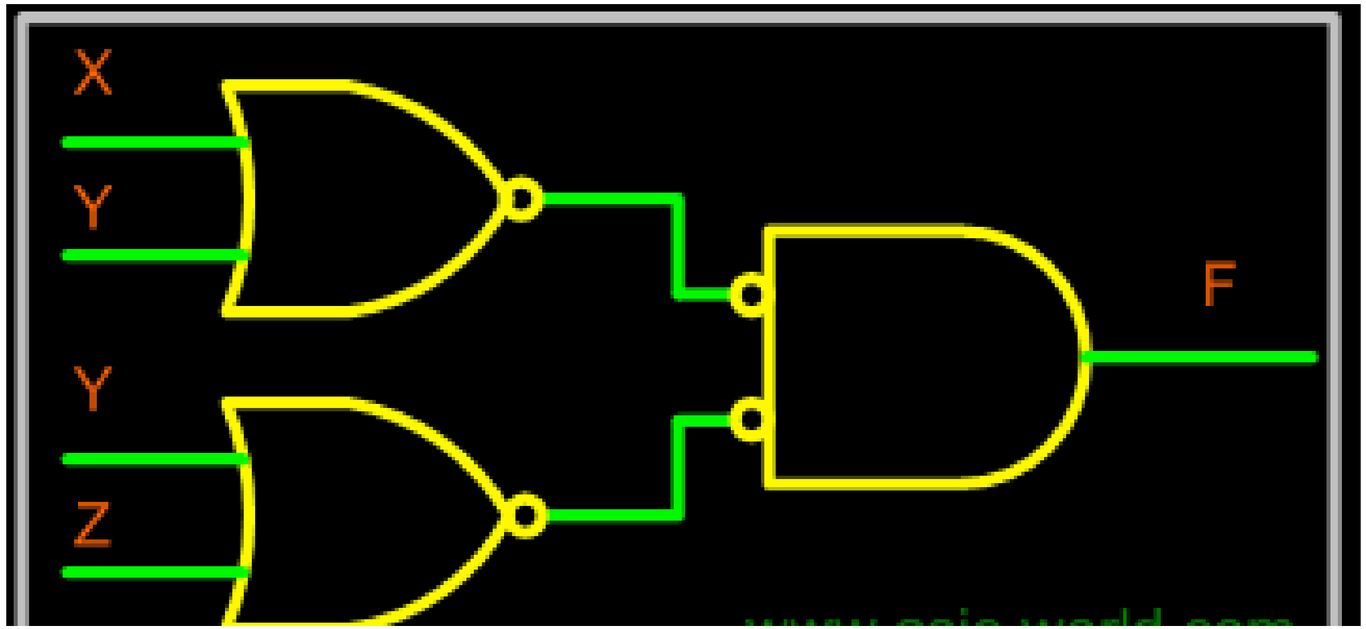
Consider the following POS expression

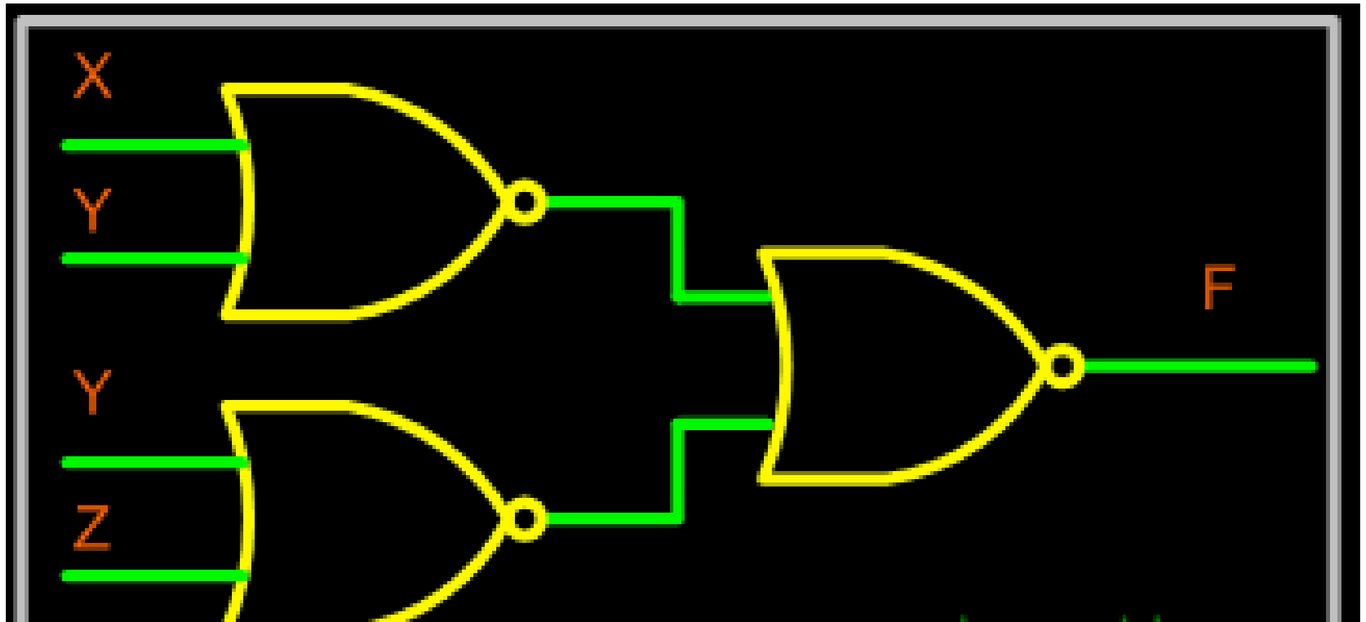
F = (X+Y) . (Y+Z)


X
Y
Y
Z
F
www.asic-world.com



If bubble are introduced at the output of the OR gates and the inputs of AND gate, the above circuit becomes as shown in figure.





Now replace AND gate with input bubble with the NOR gate. Now we have circuit which is fully implemented with just NOR gates.