Multiple-Input Gates

Many Boolean functions of three or more inputs exist. The most common are AND, OR, XOR, NAND, NOR, and XNOR. An *N*-input AND gate produces a TRUE output when all *N* inputs are TRUE. An *N*-input OR gate produces a TRUE output when at least one input is TRUE.

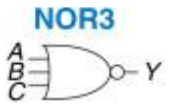
Example

Three-Input NOR Gate

The figure below shows the symbol and Boolean equation for a three-input NOR gate. Complete the truth table.

Solution

Figure bello shows the truth table. The output is TRUE only if none of the inputs are TRUE.



$Y = \overline{A + B + C}$

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

A	B	C	Y
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

Example

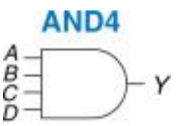
Four-Input And Gate

The figure below shows the symbol and Boolean equation for a four-input AND gate. Create a truth table.

Solution

Figure bellow shows the truth table. The output is TRUE only if all of the inputs are TRUE.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



$Y = ABCD$

Canonical and Standard Form

Canonical Form – In Boolean algebra, the Boolean function can be expressed as Canonical Disjunctive Normal Form known as minterm and some are expressed as Canonical Conjunctive Normal Form known as maxterm.

In Minterm, we look for the functions where the output results in "1" while
In Maxterm we look for functions where the output results in "0".
We perform the Sum of minterm also known as the Sum of products (SOP).
We perform Product of Maxterm also known as Product of sum (POS).
Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form.

A Boolean function can be expressed algebraically from a given truth table by forming a :

- minterm for each combination of the variables that produces a 1 in the function and then takes the OR of all those terms.
- maxterm for each combination of the variables that produces a 0 in the function and then takes the AND of all those terms.

SOP Truth Table

Consider a function X, whose truth table is as follows:

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

The function X can be written in POS form by multiplying all the max-terms when X is LOW(0).
While writing POS, the following convention is to be followed:

```
If variable A is Low(0) - A
            A is High(1) - A'

X (POS) = ΠM(0, 2, 4, 5, 7)

        = (A+B+C).(A+B'+C).(A'+B+C).(A'+B+C').(A'+B'+C')
```

Difference between SOP and POS in Digital Logic

S.No.	SOP	POS
1.	A way of representing Boolean expressions as sum of product terms.	A way of representing Boolean expressions as product of sum terms.
2.	SOP uses minterms. Minterm is product of Boolean variables either in normal form or complemented form.	POS uses maxterms. Maxterm is sum of Boolean variables either in normal form or complemented form.
3.	It is sum of minterms. Minterms are represented as 'm'	It is product of maxterms. Maxterms are represented as 'M'
4.	SOP is formed by considering all the minterms, whose output is HIGH(1)	POS is formed by considering all the maxterms, whose output is LOW(0)
5.	While writing minterms for SOP, input with value 1 is considered as the variable itself and input with value 0 is considered as complement of the input.	While writing maxterms for POS, input with value 1 is considered as the complement and input with value 0 is considered as the variable itself.

What is the symbol for SOP and POS?

SOP and POS are two forms of Boolean expression where SOP is denoted with the sign summation ∑ and POS is denoted by pi notation ***Π****.*

Example of SOP form = AB + BC + CA
Example of POS form = (A + B)(B + C)(C + A)

Converting an SOP Expression into a Truth Table

Consider the following *sum of product* expression:

$$Q = A.B.\overline{C} + A.\overline{B}.C + \overline{A}.B.C$$

Sum of Product Truth Table Form

Inputs			Output	Product
C	B	A	Q	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$A.B.\overline{C}$
1	0	0	0	
1	0	1	1	$A.\overline{B}.C$
1	1	0	1	$\overline{A}.B.C$
1	1	1	0	

Example :

The following Boolean Algebra expression is given as:

$$Q = A.B.C + \overline{A}.\overline{B}.C + \overline{A}.B.C + \overline{A}.B.\overline{C}$$

1. Use a truth table to show all the possible combinations of input conditions that will produces an output.

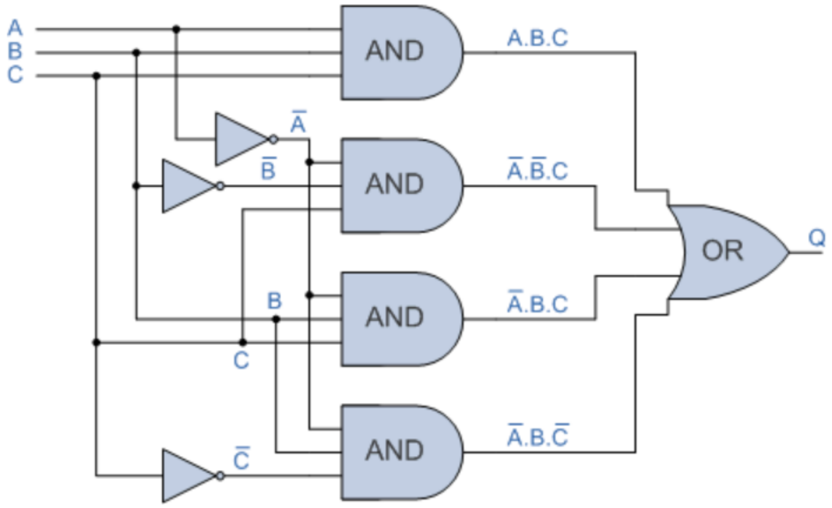
2. Draw a logic gate diagram for the expression.

Solution :

1.Sum of Product Truth Table Form

Inputs			Output	Product
C	B	A	Q	
0	0	0	0	
0	0	1	0	
0	1	0	**1**	$\overline{A}.B.\overline{C}$
0	1	1	0	
1	0	0	**1**	$\overline{A}.\overline{B}.C$
1	0	1	0	
1	1	0	**1**	$\overline{A}.B.C$
1	1	1	**1**	A.B.C

2. Logic Gate SOP Diagram



Converting an POS Expression into a Truth Table

Consider the following *product of sum* expression:

$$Q = (A + B + C)(A + \overline{B} + C)(A + \overline{B} + \overline{C})$$

We can now draw up the truth table for the above expression to show a list of all the possible input combinations for A, B and C which will result in an output "0".

Product of Sum Truth Table Form

Inputs			Output	Product
C	B	A	Q	
0	0	0	0	$A + B + C$
0	0	1	1	
0	1	0	0	$A + \overline{B} + C$
0	1	1	1	
1	0	0	1	
1	0	1	1	
1	1	0	0	$A + \overline{B} + \overline{C}$
1	1	1	1	

Example:
The following Boolean Algebra expression is given as:

$$Q = (A + B + C)(A + \overline{B} + C)(\overline{A} + B + \overline{C})(A + \overline{B} + \overline{C})$$

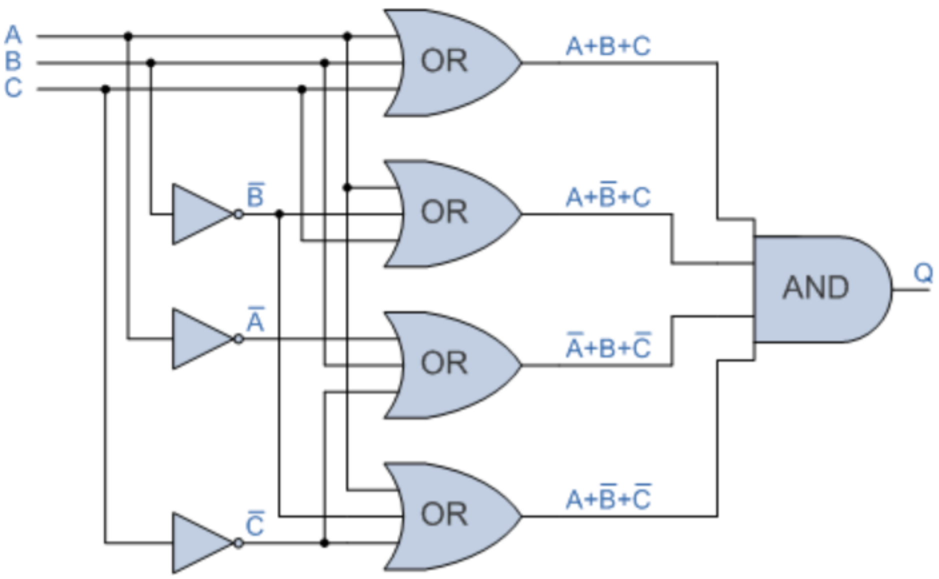
1. Use a truth table to show all the possible combinations of input conditions that will produces a "0" output.
2. Draw a logic gate diagram for the POS expression.

1. Truth Table

Product of Sum Truth Table Form

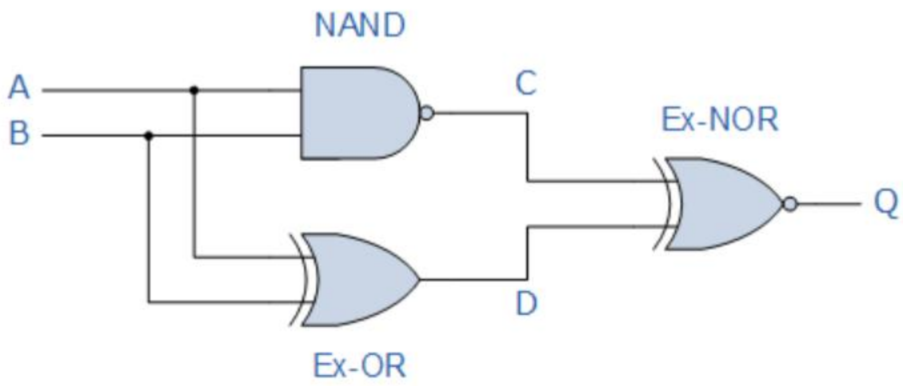
Inputs			Output	Product
C	B	A	Q	
0	0	0	**0**	$A+B+C$
0	0	1	1	
0	1	0	**0**	$A+\overline{B}+C$
0	1	1	1	
1	0	0	1	
1	0	1	**0**	$\overline{A}+B+\overline{C}$
1	1	0	**0**	$A+\overline{B}+\overline{C}$
1	1	1	1	

2. Logic Gate Diagram



Boolean Algebra Examples:

Construct a Truth Table for the logical functions at points C, D and Q in the following circuit and identify a single logic gate that can be used to replace the whole circuit.



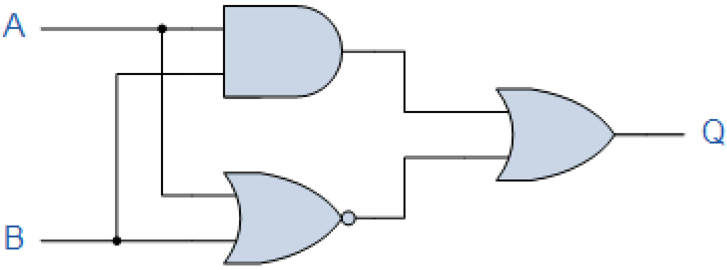
Inputs		Output at		
A	B	C	D	Q
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	0	1

From the truth table above, column C represents the output function generated by the NAND gate, while column D represents the output function from the Ex-OR gate. Both of these two output expressions then become the input condition for the Ex-NOR gate at the output.

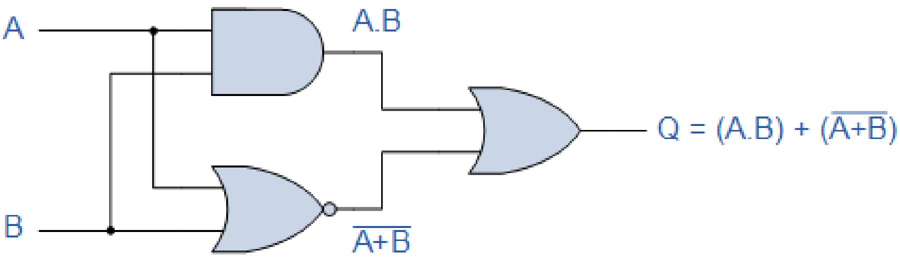
It can be seen from the truth table that an output at Q is present when any of the two inputs A or B are at logic 1. The only truth table that satisfies this condition is that of an OR Gate. Therefore, the whole of the above circuit can be replaced by just one single 2-input OR Gate.

Boolean Algebra Examples No2

Find the Boolean algebra expression for the following system.



The system consists of an AND Gate, a NOR Gate and finally an OR Gate. The expression for the AND gate is A.B, and the expression for the NOR gate is A+B. Both these expressions are also separate inputs to the OR gate which is defined as A+B. Thus the final output expression is given as:

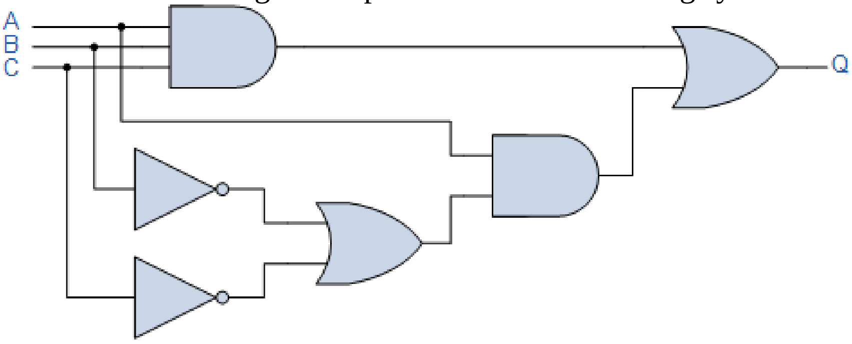


Inputs		Intermediates		Output
B	A	A.B	$\overline{A+B}$	Q
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

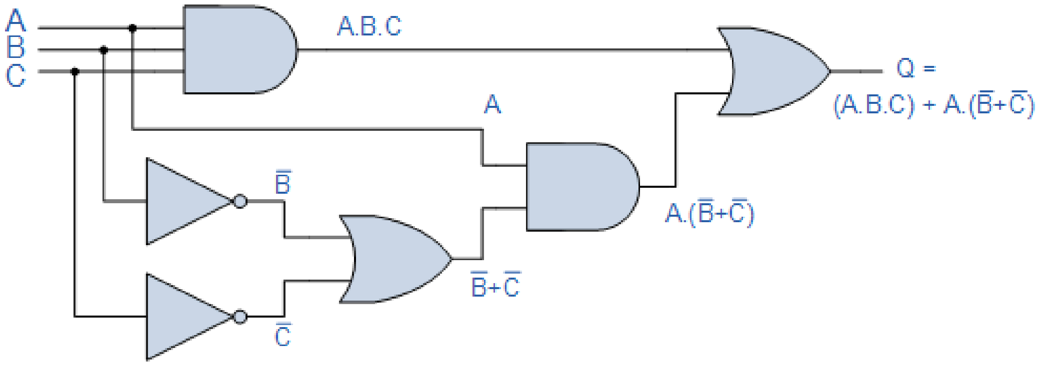
Then, the whole circuit above can be replaced by just one single Exclusive-NOR Gate and indeed an Exclusive-NOR Gate is made up of these individual gate functions.

Example No3

Find the Boolean algebra expression for the following system.



This system may look more complicated than the other two to analyse but again, the logic circuit just consists of simple AND, OR and NOT gates connected together.



Inputs			Intermediates					Output
C	B	A	A.B.C	$\overline{B}$	$\overline{C}$	$\overline{B}+\overline{C}$	$A.(\overline{B}+\overline{C})$	Q
0	0	0	0	1	1	1	0	0
0	0	1	0	1	1	1	1	1
0	1	0	0	0	1	1	0	0
0	1	1	0	0	1	1	1	1
1	0	0	0	1	0	1	0	0
1	0	1	0	1	0	1	1	1
1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1