

COMPUTER ARCHITECTURE

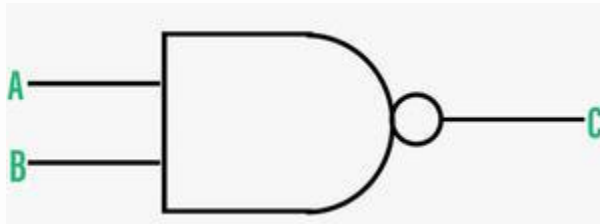
UNIT-I

NAND and NOR

We introduce two new gates, the NAND and NOR. These have advantages in size and speed because of the way transistor circuits implement gates.

NAND gate

A NAND gate (short for NOT-AND) is the same as AND followed by NOT. You can see how we've squished the symbols together,



The equation for a NAND gate is $C = A \cdot B$

The truth table for NAND is the opposite of AND. You list the same inputs but invert the output on every row,

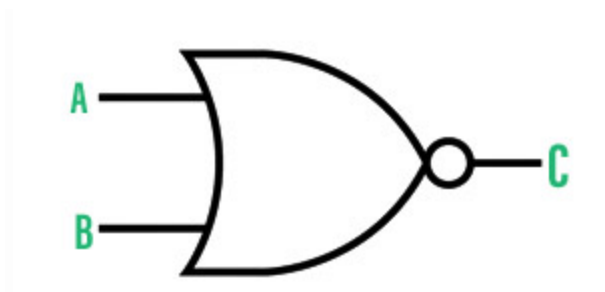
Input		AND	NAND
A	B	$A \cdot B$	$(A \cdot B)'$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

There are a few ways to think about the NAND truth table.

- Describe when the output is TRUE: The output is TRUE if *any* of the inputs are FALSE.
- Describe when the output is FALSE: The output is FALSE if *both* inputs are TRUE.

NOR gate

A NOR gate (NOT-OR) is the same as OR followed by NOT. We merge the OR and NOT symbols together to get NOR,



The equation for a NOR gate is $C = A+B$

The truth table for NOR is the opposite of OR. You list the same inputs but invert the output on every row,

Input		OR	NOR
A`	B	A+B	(A+B)'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

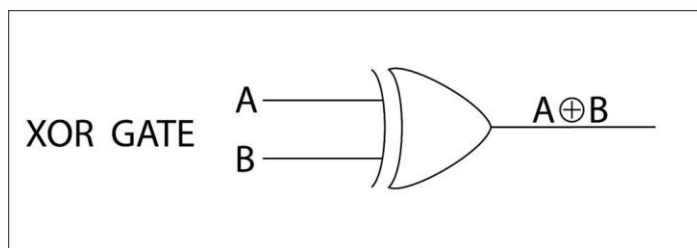
There are a few ways to think about the NOR truth table.

- Describe when the output is TRUE: The output is TRUE if *both* of the inputs are FALSE.
- Describe when the output is FALSE: The output is FALSE if *any* inputs are TRUE.

XOR Gate

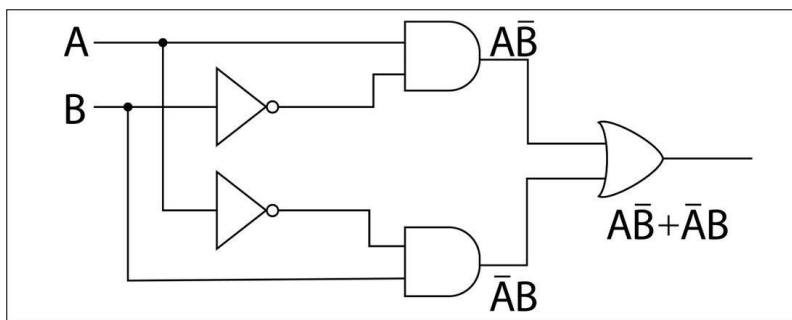
It is termed as “**Exclusive or Gate**”. If the input values are 1 and 0 the output will be 1 or true otherwise output will be 0 or false.

In XOR Gate , we take only two inputs.



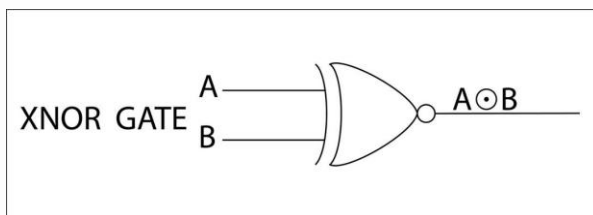
A	B	OUTPUT
0	0	0
0	1	1
1	0	1
1	1	0

The Boolean expression for XOR gate is: $AB' + A'B$ or $A \oplus B$



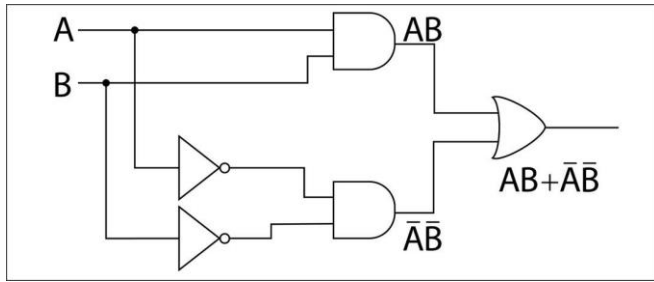
XNOR Gate

It is termed as “**Exclusively NOR gate**”. XNOR gate is basically a XOR gate added with a NOT gate. If both the input values are same (i.e. both 0 and both 1) then output will be true or 1, otherwise output will be 0 or false.



A	B	OUTPUT
0	0	1
0	1	0
1	0	0
1	1	1

The Boolean expression for XNOR gate is: $AB + A'B'$ or $A \odot B$



Laws of Boolean Algebra:-

AND Laws

In Boolean algebra, there are four AND laws given below –

- Law 1 – $A \cdot 0 = 0$ (This law is called null law).
- Law 2 – $A \cdot 1 = A$ (This law is called identity law).
- Law 3 – $A \cdot A = A$
- Law 4 – $A \cdot A' = 0$

OR Laws

There are four OR laws described below –

- Law 1 – $A + 0 = A$ (This law is called null law).
- Law 2 – $A + 1 = 1$ (This law is called identity law).
- Law 3 – $A + A = A$
- Law 4 – $A + A' = 1$

Complementation Laws

There are following five complementation laws in Boolean algebra –

- Law 1 – $0' = 1$
- Law 2 – $1' = 0$
- Law 3 – If $A = 0$, Then $A' = 1$
- Law 4 – If $A = 1$, Then $A' = 0$
- Law 5 – $(A')' = A$ (This is called double complementation law)

Commutative Laws

- $A + B = B + A$
- $A + B + C = C + B + A = B + C + A = C + A + B$
- $A \cdot B = B \cdot A$
- $A \cdot B \cdot C = A \cdot C \cdot B = C \cdot B \cdot A = C \cdot A \cdot B$

Associative Laws

- Law 1 –
 $(A + B) + C = A + (B + C)$
- Law 2 –
 $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

Distributive Laws

- Law 1 –
 $A \cdot (B + C) = AB + AC$
- Law 2 –
 $A + BC = (A + B)(A + C)$

Idempotence Laws

- Law 1 –
 $A \cdot A = A$
- Law 2 –
 $A + A = A$

DeMorgan's Theorem

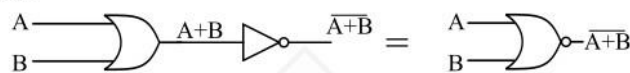
In Boolean algebra, DeMorgans theorem defines two laws which are explained below.

Law 1 – According to this law, the complement of a sum of variables is equivalent to the product of complement of each of the variables, i.e.,

$$\overline{A+B} = \overline{A} \cdot \overline{B}$$

This law can be extended to any number of variables.

LHS:



RHS:

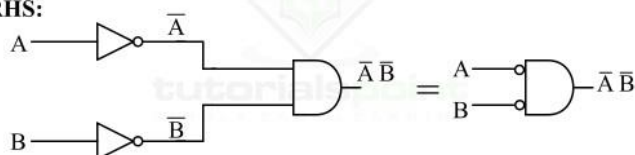


Figure 1 - De Morgan's First Law

Left Side			Right Side		
Input		Output	Input		Output
A	B	$(A + B)'$	A'	B'	$A' \cdot B'$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0

Law 2 – The second law of DeMorgans theorem states that the complement of a product of variables is equivalent to the sum of complement of each of the variables, i.e.,

$$\overline{AB} = \overline{A} + \overline{B}$$

This law can also be extended to any number of variables.

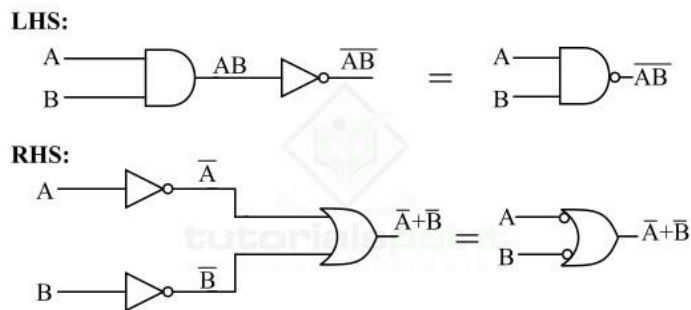


Figure 2 - De Morgan's Second Law

Left Side			Right Side		
Input		Output	Input		Output
A	B	AB	A'	B'	$A' + B'$
0	0	0	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	1	0	0	0