

Introduction to Boolean Algebra and Logic Gates

LOGIC GATES

- Logic gates are the fundamental building blocks of digital systems. It consists of several inputs and only one output.
- There are just three basic types of gates- AND, OR and NOT. The interconnection of gates to perform a variety of logic operations is called logic design.
- The inputs and outputs of logic gates can occur only in two levels. These two levels are termed HIGH and LOW or TRUE and FALSE or ON and OFF or simply 1 and 0.

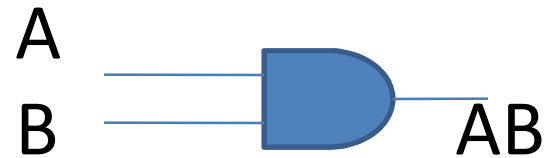
THE AND GATE

An AND gate has two or more inputs but only one output. The output is logic HIGH only when all the inputs are at logic high. The output is logic low, if any one of the inputs is logic LOW.

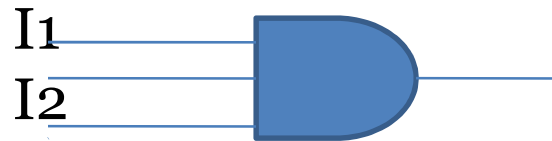
- The truth table of two input AND gate, and symbol of two input AND gate is shown below.

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

Symbol of 2 input AND gate:



Symbol of n input AND gate:



THE OR GATE

- The OR gate has two or more inputs but only one output. The output is logic HIGH if any one of the input is logic HIGH. The output is logic LOW only when all the inputs are at logic LOW. Logic symbol of two input OR gate, 'n' input OR gate and truth table of two input OR gate is shown below. The symbol for the OR operation is '+'.
The symbol for the OR operation is '+'.

TRUTH TABLE:

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

Symbol of 2 input OR gate:



Symbol of n input AND gate:



THE NOT GATE

- A NOT gate, also known as inverter, has only one input and only one output. Its output is always the complement of its input. That is, the output of a NOT gate is logic HIGH when its input is logic LOW and the output is logic LOW when its input is logic HIGH. The logic symbol of NOT gate and its truth table is shown below.

Symbol:



Truth Table:

INPUT	OUTPUT
A	Y
0	1
1	0

THE EXCLUSIVE - NOR (XNOR) GATE

An X-OR gate has two input but only one output. The output is logic High when the inputs are not equal. The output is logic Low when the inputs are equal to either logic HIGH or logic LOW. Logic symbol of two input X-

(le is

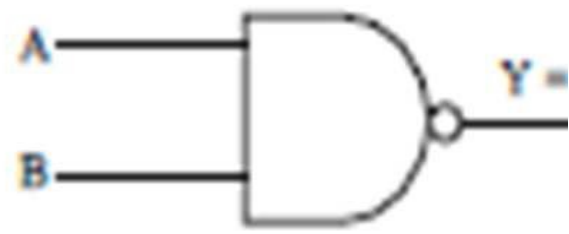


Logic symbol of 2-input X-OR gate

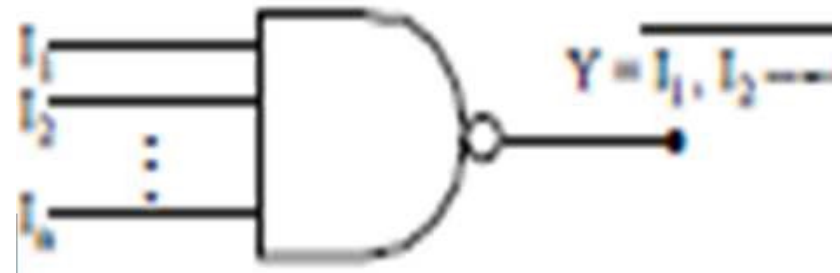
Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

THE NAND GATE

NAND means NOT AND ie the AND gate output is NOTed. So, a NAND gate is a combination of an AND gate and a NOT gate. It has two or more inputs but only one output. The output is High if any one of the input is logic LOW. The output is LOW only when all the inputs are HIGH. The logic symbol for two input NAND gate, logic symbol for “n” input NAND gate and truth table of two input NAND gate are shown below.



Logic symbol of 2-input NAND

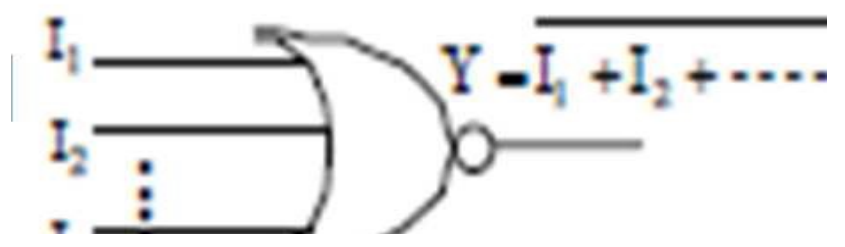


Logic symbol of n-input NAND gate

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

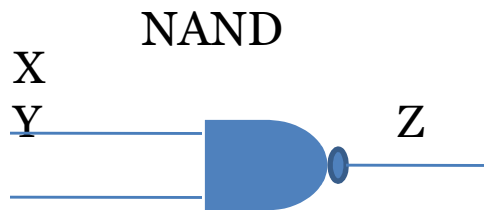
THE NOR GATE

- NOR means NOT OR ie OR gate output is NOT ed. So, a NAND gate is a combination of an OR gate and a NOT gate. It has two or more inputs but only one output. The output is LOW if any one of the input is HIGH. The output is HIGH only when all the inputs are LOW.
- The logic symbol for two input NOR gate, logic symbol for 'n' input NOR gate and truth table of two input NOR gate are shown below.



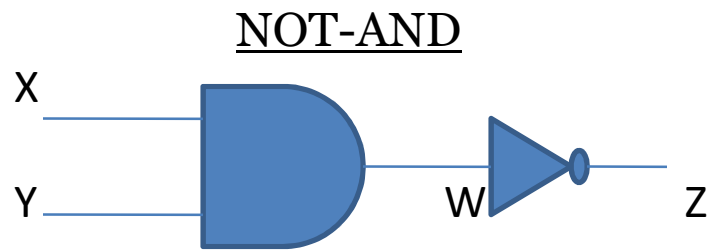
Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0

NAND Gate



$Z = \sim(X \& Y)$
 $\text{nand}(Z,X,Y)$

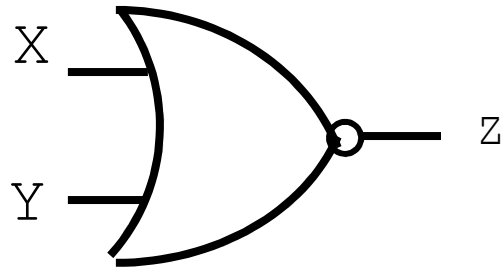
X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0



$W = X \& Y$
 $Z = \sim W = \sim(X \& Y)$

X	Y	W	Z
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

NOR Gate

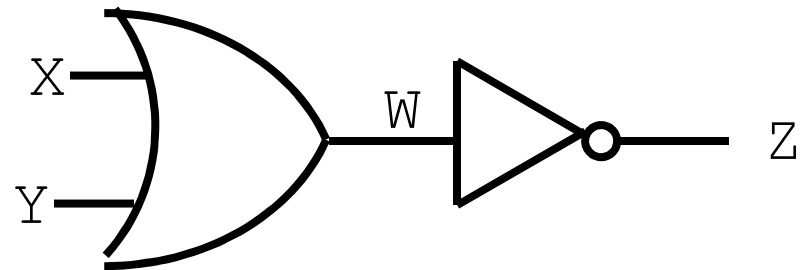


$$Z = \sim (X \mid Y)$$

nor (Z, X, Y)

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

NOT-OR Gate

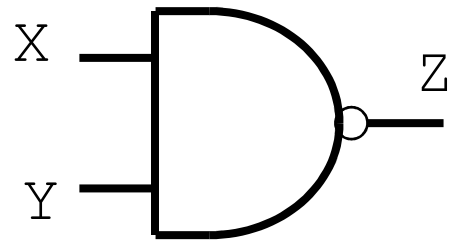


$$W = X \mid Y$$

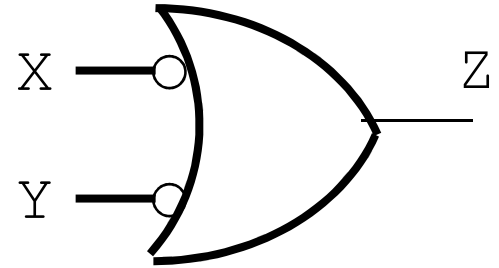
$$Z = \sim W = \sim (X \mid Y)$$

X	Y	W	Z
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

NAND Gate



=



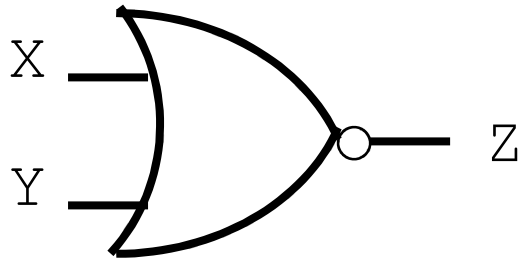
$$Z = \sim (X \ \& \ Y)$$

$$Z = \sim X \ | \ \sim Y$$

X	Y	W	Z
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

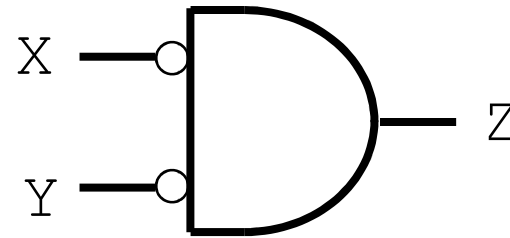
X	Y	$\sim X$	$\sim Y$	Z
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

NOR Gate



$$Z = \sim (X \mid Y)$$

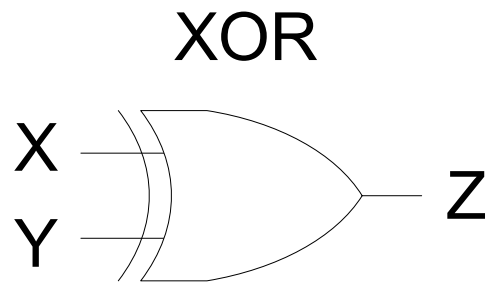
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0



$$Z = \sim X \ \& \ \sim Y$$

X	Y	$\sim X$	$\sim Y$	Z
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Exclusive-OR Gate



$$Z = X \wedge Y$$

xor (Z, X, Y)

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

XOR

• $X \wedge Y$ (Verilog)

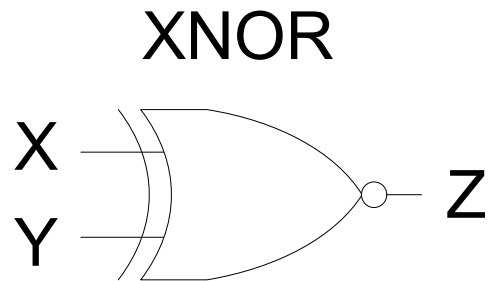
• $X \$ Y$ (ABEL)

• $X @ Y$

□ $X \oplus Y$ (textbook)

• **xor**(Z,X,Y) (Verilog)

Exclusive-NOR Gate



$$Z = \sim (X \wedge Y)$$

$$Z = X \sim \wedge Y$$

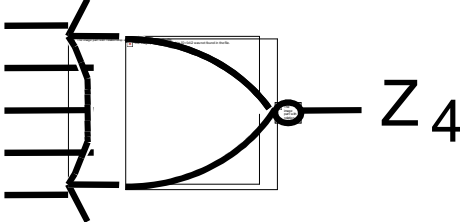
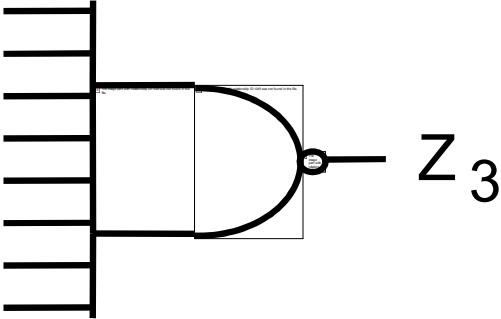
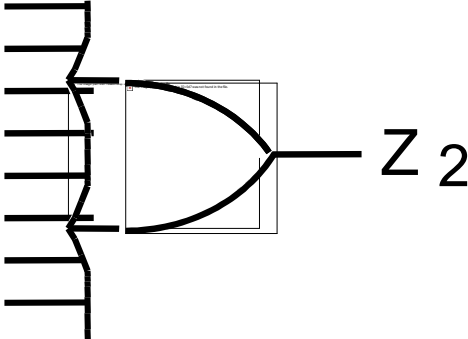
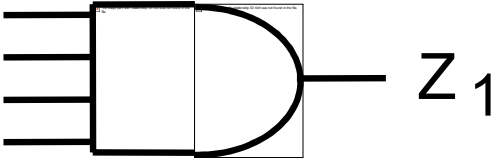
xnor (Z, X, Y)

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

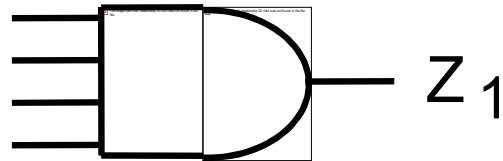
XNOR

- $X \sim^{\wedge} Y$ (Verilog)
- $!(X \$ Y)$ (ABEL)
- $X @ Y$
- $\square X \square Y$
- **xnor**(Z,X,Y) (Verilog)

Multiple-input Gates



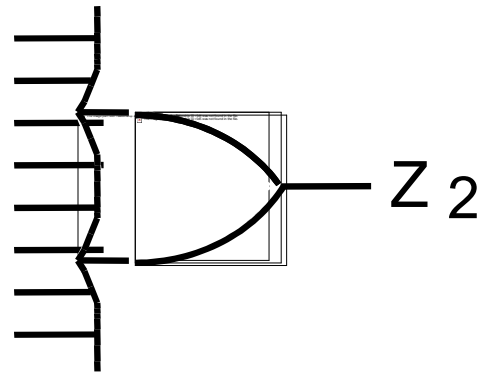
Multiple-input AND Gate



Output Z_1 is HIGH only if all inputs are HIGH

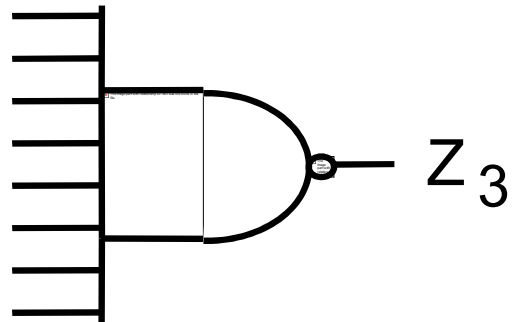
An open input will float HIGH

Multiple-input OR Gate



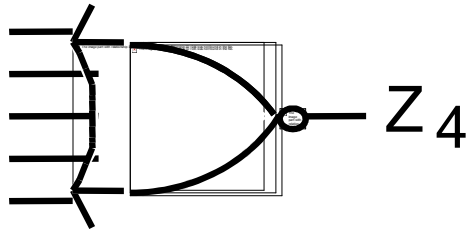
Output Z_2 is LOW only if all inputs are LOW

Multiple-input NAND Gate



Output Z_3 is LOW only if all inputs are HIGH

Multiple-input NOR Gate



Output Z_4 is HIGH only if all inputs are LOW

THANK YOU

Introduction:

- Boolean algebra is a system of mathematical logic.
- Any complex logic statement can be expressed by a Boolean function.
- The Boolean algebra is governed by certain rules and laws.
- The binary digits 0 and 1 are used to represent the two voltage levels that occur with in the digital circuit.
- If we follow positive logic, logic 1 represents (+5v, High, ON, True). Logic 0 represents (0v, Low, OFF, False).

POSTULATES OF BOOLEAN ALGEBRA

Postulate: Anything which is not proved but assumed to be true is known as postulate.

` If M is a set, x, y are two objects, $x, y \in M$

implies that X and Y all members of the set M .

1. (a) An operation '+' is defined such that if $Z = X + Y$ then $Z \in M$ for every pair of elements $x, y \in M$.
(b) An operation is defined such that if $W = x ; y$ then $W \in M$ for every pair of elements $x, y \in M$.
2. There exists an element '0' in M such that
 - (a) $x + 0 = x$ for every element $x \in M$
 - (b) There exists an element '1' in M such that $x + 1 = 1$ for every element. $1 \in M$.

3. For $x, y \in M$ the following commutative laws hold as follows.

(a) $x + y = y + x$

(b) $x \cdot y = y \cdot x$.

4. For $x, y, z \in M$ the following distributive laws hold as follows.

(a) $x \cdot (y + z) = x \cdot y + x \cdot z$

(b) $x + (y \cdot z) = (x + y) \cdot (x + z)$

5. For every element $x \in M$ there exists an element such that

(a) $x + x = 1$ (b) $x \cdot x = 0$ (*unary operation*)

6. There are at least two elements $x, y \in M$ such that $x \in y$

LAWS OF BOOLEAN ALZEBRA

Complementation laws or Negation laws:

(complement means change 0 to 1 or 1 to 0)

Law 1: $0 = 1$; Law 2: $1 = 0$.

Law 3: If $A = 0$; Then $\bar{A} = 1$

Law 4: If $A = 1$; Then $\bar{A} = 0$

Law 5: $\bar{\bar{A}} = A$; (*Double Negation law*)

Law 6: $A \cdot \bar{A} = 0$; Law 7: $A + \bar{A} = 1$

AND Laws:

Law 1: $A \cdot 0 = 0$; Law 2: $A \cdot 1 = A$; Law 3: $A \cdot A = A$; Law 4: $A \cdot \bar{A} = 0$

OR Laws:

Law 1: $A + 0 = A$; Law 2 : $A + 1 = 1$; Law 3 : $A + A = A$
; Law 4 : $A += 1$

Commutative Laws:

Law 1: $A + B = B + A$
2: $A . B = B . A$

Associative Laws:

Law 1: $(A + B) + C = A + (B + C)$
Law 2 : $(A . B) C = A (B . C)$

Distributive Laws:

Law 1 : $A (B + C) = AB + AC$
Law 2 : $A + BC = (A + B) (A + C)$
Law 3 : $A + B = A + B$

Idempotence Laws:

Law 1 : $A \cdot A = A$

Law 2 : $A + A = A$

Identity Laws:

Law 1: $A \cdot 1 = A$; Law 2 : $A + 0 = A$

Null Laws:

Law 1 : $A \cdot 0 = 0$; Law 2 : $A + 1 = 1$

Absorption Laws:

Law 1 : $A + A \cdot B = A$

Demorgan's Laws:

Law 1: = The complement of a sum of variables is equal to the product of their individual complements.

Law 2: = The complement of a product of variables is equal to the sum of their individual complements.

Theorem : $AB + AC = (A + C)(A + B)$

Proff : $= A \cdot A + A \cdot B + C \cdot A + C \cdot B$
 $= 0 + AB + CA + CB (A + A)$
 $= AB + CA + CBA + CBA$
 $= AB (1 + C) + A C (1 + B)$
 $= AB + C$
 $= \text{LHS.}$

TRANSPOSITION THEOREM

Theorem : $AB + AC = (A + C)(A + B)$

Proff : $= A \cdot A + A \cdot B + C \cdot A + C \cdot B$

$$= 0 + AB + CA + CB (A + A)$$

$$= AB + CA + CBA + CBA$$

$$= AB (1 + C) + A C (1 + B)$$

$$= AB + C$$

$$= \text{LHS.}$$

CONSENSUS THEOREM

First Theorem: $AB + AC + BC = AB + AC$

$$AB + AC + BC$$

$$AB + AC + BC (A + A) (QA + A = 1)$$

$$AB + CA + BCA + BCA$$

$$AB (1 + C) + CA (1 + B)$$

$$AB + C$$

$$= \text{RHS.}$$

Second Theorem: $(A + B) (A + C) (B + C) = (A + B) (A + C)$

Proof: LHS = $(A + B) (A + C) (B + C)$

$$= [AA + AC + BA + BC] (B + C)$$

$$= AAB + ABC + BBA + AAC + ACC + BAC + BCC$$

$$= 0 + ABC + AB + BC + 0 + AC + BAC + BC$$

$$= ABC + AB + BC + AC + BC \text{ (QBC + BC = BC)}$$

$$= BC (A + A) + BC + AB + AC$$

$$= BC + AB + AC.$$

$$\text{RHS} = (A + B) (A + C)$$

$$= A \cdot A + AC + AB + BC.$$

$$= 0 + AC + AB + BC$$

$$= BC + AB + AC.$$

$$\text{LHS} = \text{RHS}$$

DUALITY:

In Boolean Algebra we can produce dual by changing all “+” signs to “.” signs, all “.” signs to “+” signs, and complementing all 0s and 1s. The variables are not complemented in this procedure. Below table form gives the dual of some expressions.

S.no Expression Dual

(1)

$$0 = 1 \quad 1 = 0$$

S.No	Expression	Dual
(1)	$0 = 1$	$1 = 0$
(2)	$0 \cdot 1 = 0$	$1 + 0 = 1$
(3)	$0 \cdot 0 = 0$	$1 + 1 = 1$
(4)	$1 \cdot 1 = 1$	$0 + 0 = 0$
(5)	$A \cdot 0 = 0$	$A + 1 = 1$
(6)	$A \cdot 1 = A$	$A + 0 = A$
(7)	$A \cdot A = A$	$A + A = 1$
(8)	$A \cdot B = B \cdot A$	$A + B = B + A$
(9)	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$
(10)	$A \cdot (A \cdot B) = B \cdot A$	$A + (A + B) = A + B$
(11)	$(A + C) (A + B) (B + C) = (A + C) AB + AC + BC = A \cdot B + (A \cdot C)$	

THANK YOU