Digital Logic :

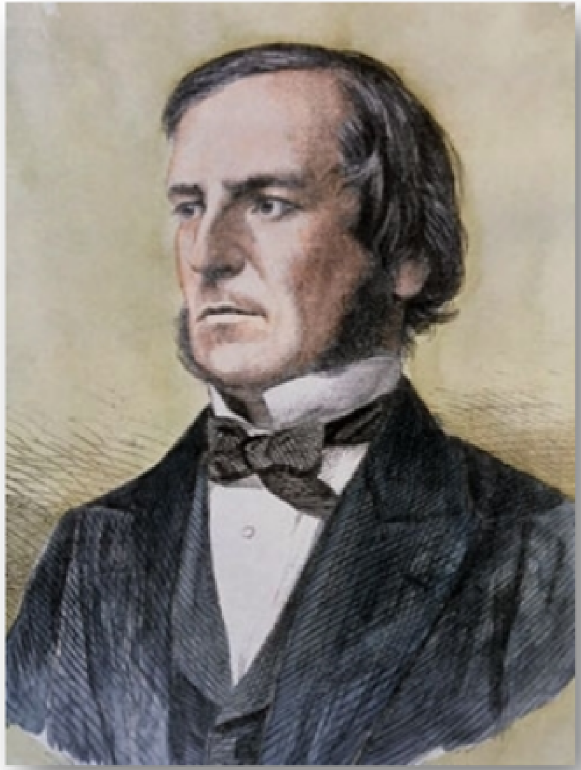
Coming Back to our main Topic :

Boolean Algebra

Website : https://www.goclasses.in/

Introduction

- Boolean Algebra is used to analyze and simplify the digital (logic) circuits.
- It uses only the binary numbers i.e. 0 and 1. It is also called as **Binary Algebra** or **logical Algebra**.
- It is a convenient way and systematic way of expressing and analyzing the operation of logic circuits
- Boolean algebra was invented by **George Boole** in 1854.



Introduction

- Variable used in Boolean algebra can have only two values. Binary 1 for HIGH and Binary 0 for LOW.
- Complement of a variable is represented by an overbar (-). Thus, complement of variable B is represented as B' . Thus if B = 0 then B'= 1 and if B = 1 then B'= 0.
- ORing of the variables is represented by a plus (+) sign between them. For example ORing of A, B, C is represented as A + B + C.
- Logical ANDing of the two or more variable is represented by writing a dot between them such as A.B.C. Sometime the dot may be omitted like ABC.

Boolean Operations

AND

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

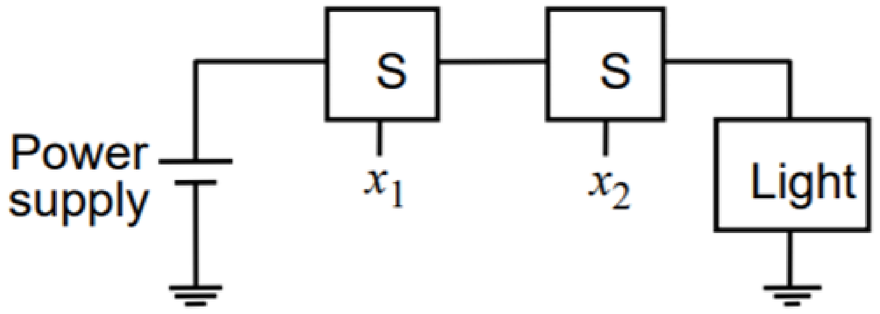
Complement

Not

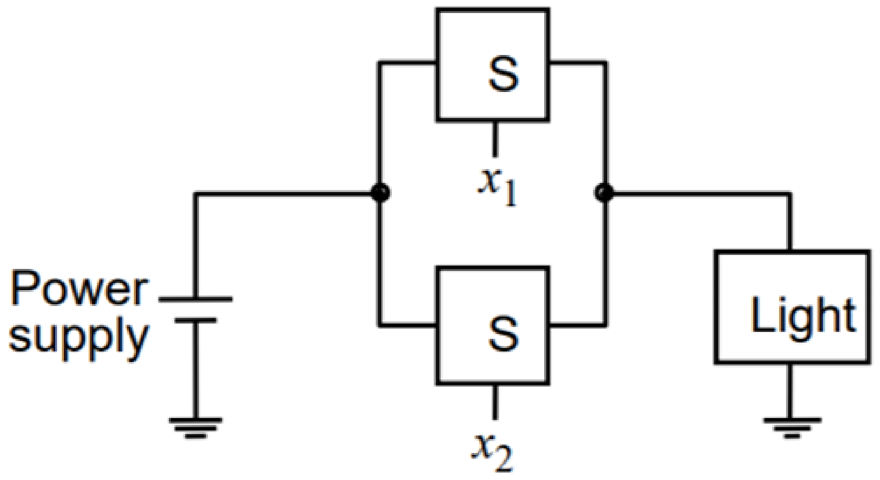
A	A'
0	1
1	0

Two Basic Functions

(a) The logical AND function (series connection)



(b) The logical OR function (parallel connection)



Boolean Logic

- Binary digits (or bits) have two values: {1,0}
- All logical functions can be implemented in terms of three logical operations:

NOT

x	$\overline{x}$
0	1
1	0

AND

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

The complement of 0 is 1, and the complement of 1 is 0. Symbolically, we write

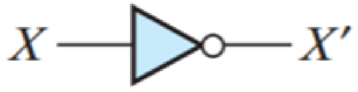
$$0' = 1 \quad \text{and} \quad 1' = 0$$

If X is a switching variable,

$$X' = 1 \text{ if } X = 0 \quad \text{and} \quad X' = 0 \text{ if } X = 1$$

An alternate name for complementation is inversion, and the electronic circuit which forms the inverse of X is referred to as an inverter. Symbolically, we represent an inverter by

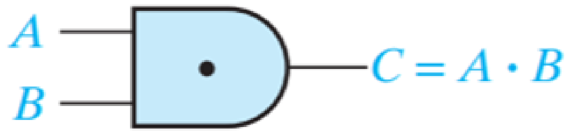
NOT GATE



Symbol

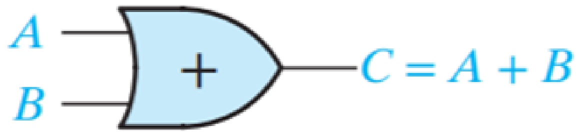
where the circle at the output indicates inversion. A low voltage at the inverter input produces a high voltage at the output and vice versa.

A logic gate which performs the AND operation is represented by



The gate output is $C = 1$ if and only if the gate inputs $A = 1$ and $B = 1$.

A logic gate which performs the OR operation is represented by



The gate output is $C = 1$ if and only if the gate inputs $A = 1$ or $B = 1$ (or both).

Digital Logic :

Next Topic :

More Standard Boolean Operations

NAND, NOR, ExOR, ExNOR

NAND Operation:

Symbol: $\uparrow$

NAND

$\underline{\text{N}}\ \underbrace{\text{AND}}$

NOT of AND

$\underbrace{a \uparrow b} = \overline{ab}$

NOR operation:

Symbol: $\downarrow$

N OR

NOT of OR

$$\underbrace{a \downarrow b} = \overline{a+b}$$

Truth Table:

a: 0 or 1

b: 0 or 1

Row	a	b	ab	$a \uparrow b = \overline{ab}$	$a+b$	$a \downarrow b = \overline{a+b}$
0	0	0	0	1	0	1
1	0	1	0	1	1	0
2	1	0	0	1	1	0
3	1	1	1	0	1	0

NAND

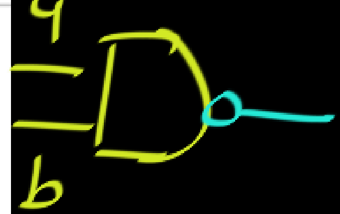
NOR

Universal gates: NAND, NOR

will see later

NAND = $\overline{\text{AND}}$

x	y	$z = \overline{xy}$
0	0	1
0	1	1
1	0	1
1	1	0



x	y	$z = \overline{x+y}$
0	0	1
0	1	0
1	0	0
1	1	0

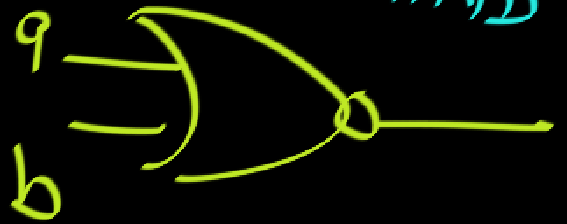
Operation

$$a \uparrow b = \overline{ab}$$

$$a \downarrow b = \overline{a+b}$$

Digital Ckt

implementing that operation



Exclusive OR (ExOR) Symbol: $\oplus$

→ Exclusively one of the variable must be 1.

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Analogy:

Exclusive news

$$\boxed{a \oplus b = 1 \quad \text{iff} \quad a \neq b}$$ **

$$\left.\begin{array}{l} 0 \oplus 1 = 1 \\ 1 \oplus 0 = 1 \end{array}\right\} 6$$

$$\left.\begin{array}{l} 1 \oplus 1 = 0 \\ 0 \oplus 0 = 0 \end{array}\right\} 6$$

Convention while writing Expression:

$a = 0 \equiv \overline{a}$

$a = 1 \equiv a$

$\overline{a} = 0 \equiv a$

Explain later

$\alpha = 0 \equiv \alpha'$

$\alpha = 1 \equiv \alpha$

Exclusive-OR and Equivalence Operations

The *exclusive-OR* operation ($\oplus$) is defined as follows:

$$0 \oplus 0 = 0 \qquad 0 \oplus 1 = 1$$
$$1 \oplus 0 = 1 \qquad 1 \oplus 1 = 0$$

The truth table for $X \oplus Y$ is

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

From this table, we can see that $X \oplus Y = 1$ iff $X = 1$ or $Y = 1$, but *not* both. The ordinary OR operation, which we have previously defined, is sometimes called inclusive OR because $X + Y = 1$ iff $X = 1$ or $Y = 1$, or both.

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

$a \oplus b = 1$ iff

$a=0, b=1$ OR $a=1, b=0$

$$a \oplus b = \bar{a}b + a\bar{b}$$

$\boxed{\text{ExNOR}}$ ≡ Complement of ExOR

Symbol: $\odot$, $\ominus$

$a \odot b \equiv a \ominus b$

a	b	$a \oplus b$	$a \odot b$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

$$\boxed{a \odot b = 1 \text{ iff } a = b}$$

$$\underbrace{a \Leftrightarrow b = 1}\ \text{iff}\ a = b$$

$a \oplus b = 1$ iff $\boxed{a \neq b}$

$a \odot b = 1$ iff $\boxed{a = b}$

Complement

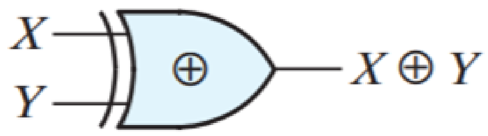
$$\boxed{\overline{a \oplus b} = a \odot b}$$

Exclusive OR can be expressed in terms of AND and OR. Because $X \oplus Y = 1$ iff X is 0 and Y is 1 or X is 1 and Y is 0, we can write

$$X \oplus Y = X'Y + XY' \tag{3-6}$$

The first term in (3-6) is 1 if $X = 0$ and $Y = 1$; the second term is 1 if $X = 1$ and $Y = 0$.

We will use the following symbol for an exclusive-OR gate:



Operations:

ExOR

$a \oplus b$

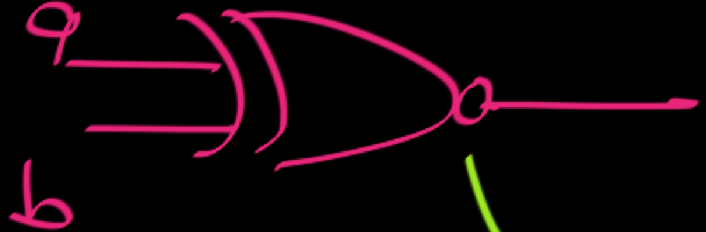
ExNOR

$a \odot b$

Digital Ckt

ExOR gate

symbol



ExNOR gate

NOT

The *equivalence* operation (≡) is defined by

$$(0 \equiv 0) = 1 \qquad (0 \equiv 1) = 0$$
$$(1 \equiv 0) = 0 \qquad (1 \equiv 1) = 1 \tag{3-16}$$

The truth table for $X \equiv Y$ is

X	Y	$X \equiv Y$
0	0	1
0	1	0
1	0	0
1	1	1

From the definition of equivalence, we see that $(X \equiv Y) = 1$ iff $X = Y$. Because $(X \equiv Y) = 1$ iff $X = Y = 1$ or $X = Y = 0$, we can write

$$(X \equiv Y) = XY + X'Y' \tag{3-17}$$

a	b	$a \odot b$
0	0	1
0	1	0
1	0	0
1	1	1

$a \odot b = 1$ iff

$a=0, b=0$ OR $a=1, b=1$

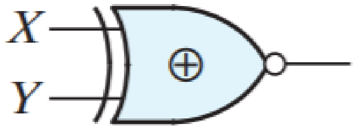
$$\underbrace{a \odot b = \bar{a}\bar{b} + ab}_{\text{Write equation}}$$

a	b	$f(a,b)$
0	0	1
0	1	0
1	0	1
1	1	1

Write equation for f:

$$f = \bar{a}\bar{b} + a\bar{b} + ab$$

Because equivalence is the complement of exclusive OR, an alternate symbol for the equivalence gate is an exclusive-OR gate with a complemented output:



$(X \oplus Y)' = (X \equiv Y)$

The equivalence gate is also called an exclusive-NOR gate.

Boolean Algebra :

Next Topic :

Boolean Expressions and Truth Table

(Standard form of Truth Table)

a	b	ab
1	0	0 ✓
1	1	1 ✓
0	1	0 ✓
0	0	0 ✓

Truth Table ✓

Standard form of Truth Table

Row Number	a	b	ab
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

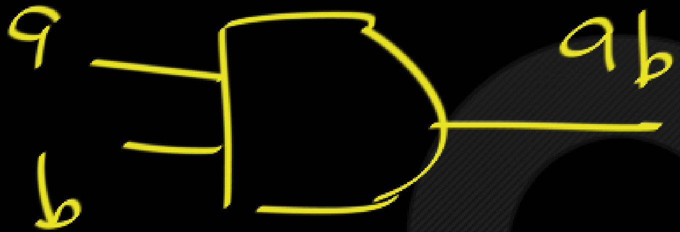
	a	b	c	$(a+b).c = E$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

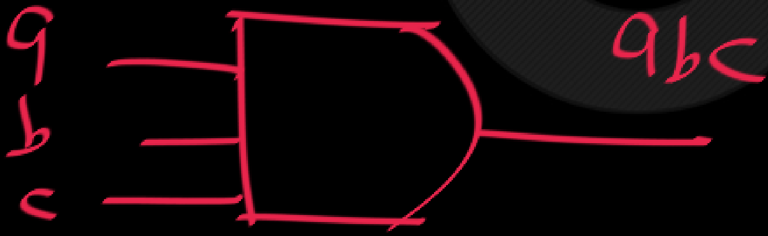
NOTE:

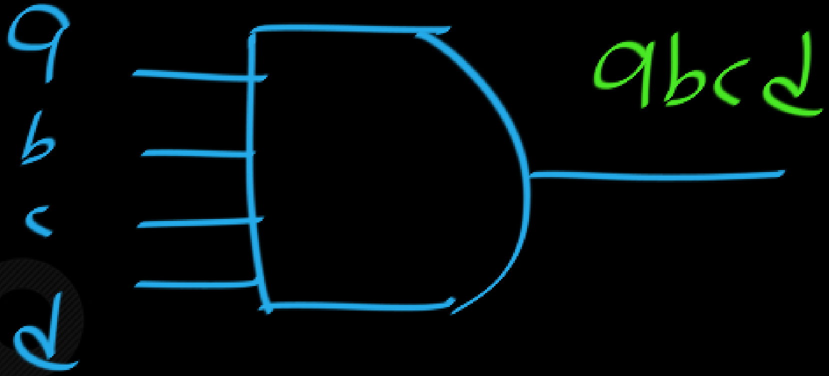
when $c=0$ then $E=0$

when $c=1$ $E=(a+b)$

AND GATE:







Truth Table of 3-Input AND and OR Operations

Row#	x_1	x_2	x_3	$x_1 \bullet x_2 \bullet x_3$	$x_1 + x_2 + x_3$
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	0	1
4	1	0	0	0	1
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

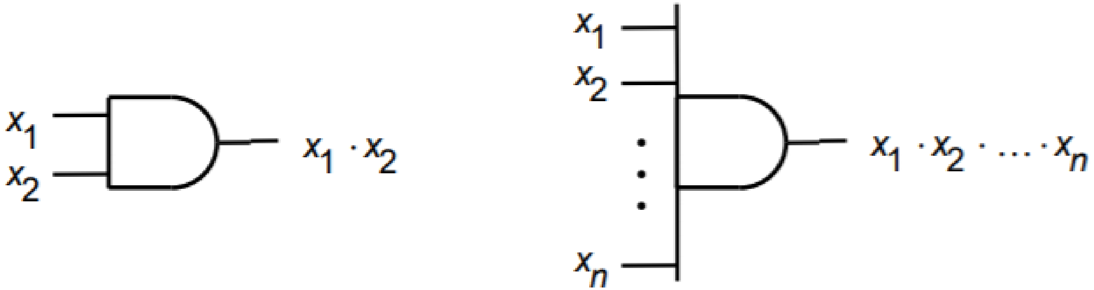
Symbols

- AND
  - Dot ($\bullet$)
  - Imagine it to be like multiplication
  - Example $x \bullet y$
  - Called “x and y”
- OR
  - Plus (+)
  - Imagine it to be like addition
  - Example $x + y$
  - Called “x or y”

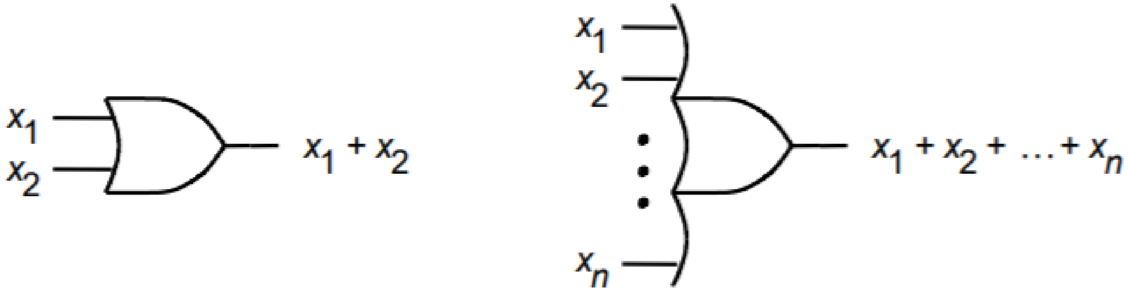
NOT Operation

- Symbol
  - Closing single quote ’
  - Also overline $\overline{}$ and ! symbol
  - Example: x', $\overline{x}$, $!x$
- Calling
  - x complement
  - “not of x”
  - Simpler: “x bar”

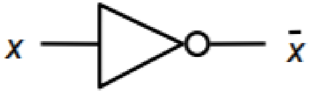
Symbols



(a) AND gates



(b) OR gates



(c) NOT gate

Literal:

"Each appearance" of a variable or its complement in an expression will be referred to as a *literal*. Thus, the following expression, which has three variables, has 10 literals:

$$\boxed{ab'c + a'b + a'bc' + b'c'}$$

variable = a

literal = a, $\overline{a}$

$E = \underbrace{a + a + a}$

#variables = 1

#literals = 3

$= a + \bar{a} + ab \Rightarrow$ #variables = 2

#literals = 4

$= \bar{a}\, a\, a\, \bar{a} \Rightarrow$ #literals = 4