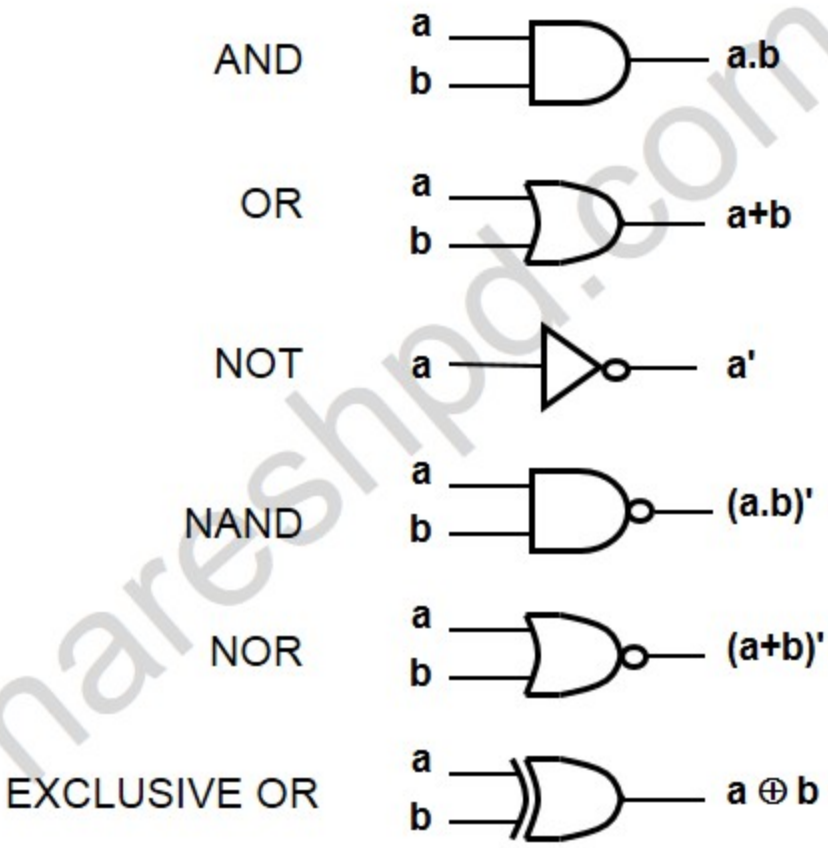
Digital (logic) Elements: Gates

- Digital devices or gates have one or more inputs and produce an output that is a function of the current input value(s).
- All inputs and outputs are binary and can only take the values 0 or 1
- A gate is called a combinational circuit because the output only depends on the current input combination.
- Digital circuits are created by using a number of connected gates such as the output of a gate is connected to the input of one or more gates in such a way to achieve specific outputs for input values.
- Digital or logic design is concerned with the design of such circuits.

nareshmd.com.np

Logic Gates

- **Gate Symbols**



nareshpd.com.np

Truth Tables

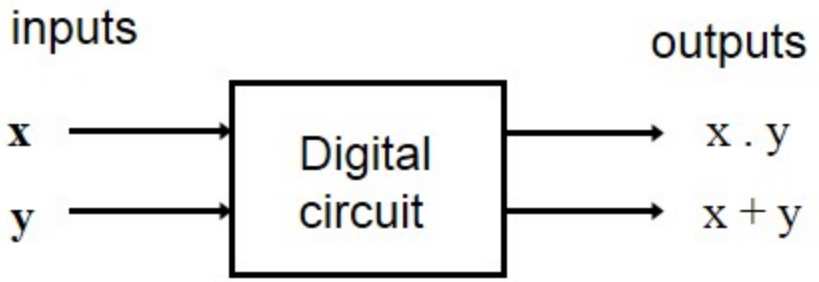
- Provides a **listing** of every possible combination of values of binary inputs to a digital circuit and the corresponding outputs.

INPUTS	OUTPUTS
…	…
…	…

- Example (2 inputs, 2 outputs):

inputs **Truth table** outputs

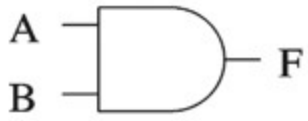
x	y	x . y	x + y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1



nareshpd.com.np

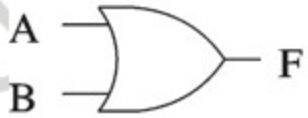
Basic Concepts

- Simple gates
  - AND
  - OR
  - NOT
- Functionality can be expressed by a truth table
  - A truth table lists output for each possible input combination
- Other methods
  - Logic expressions
  - Logic diagrams



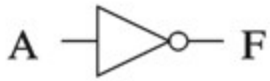
AND gate

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1



OR gate

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



NOT gate

A	F
0	1
1	0

Logic symbol

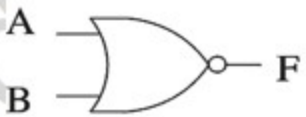
Truth table

Basic Concepts (cont'd)

- Additional useful gates
  - NAND
  - NOR
  - XOR
- NAND = AND + NOT
- NOR = OR + NOT
- XOR implements exclusive-OR function
- NAND and NOR gates require only 2 transistors
  - AND and OR need 3 transistors!

NAND gate

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0



NOR gate

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

XOR gate

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Logic symbol

Truth table

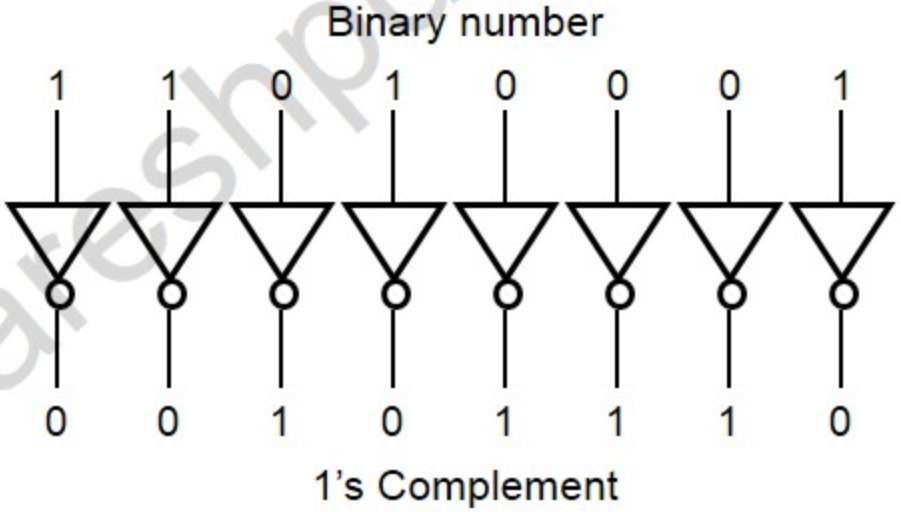
Ireshpc

Logic Gates: The Inverter

- The **Inverter**

A	A'
0	1
1	0

- Application of the inverter: complement.



nareshpa.com.np

Logic Gates: The AND Gate

- The **AND** Gate

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

nareshpd.com.np

Logic Gates: The OR Gate

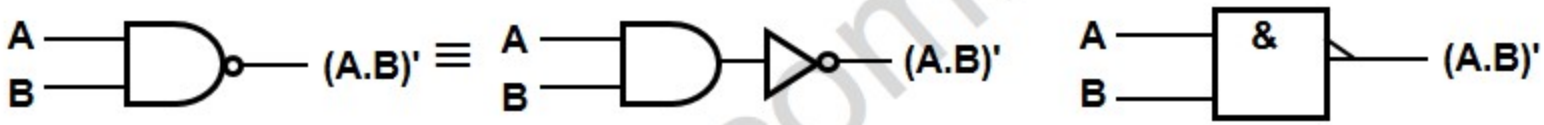
- The **OR** Gate

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

nareshpd.com.np

Logic Gates: The NAND Gate

- The **NAND** Gate



A	B	(A.B)'
0	0	1
0	1	1
1	0	1
1	1	0

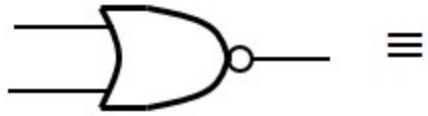
NAND

nareshpd.com.np

Logic Gates: The NOR Gate

- The **NOR** Gate

A	B	(A+B)'
0	0	1
0	1	0
1	0	0
1	1	0



NOR

nareshpd.com.np

Logic Gates: The XOR Gate

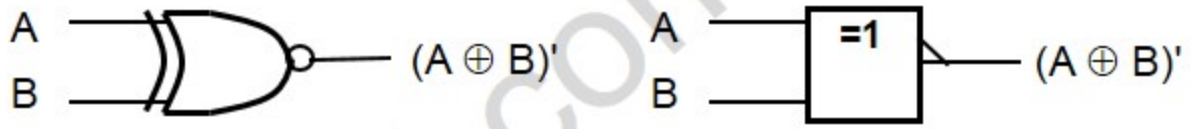
- The **XOR** Gate

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

nareshnd.com.np

Logic Gates: The XNOR Gate

- The **XNOR** Gate

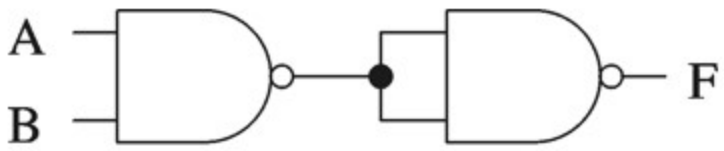


A	B	(A ⊕ B) '
0	0	1
0	1	0
1	0	0
1	1	1

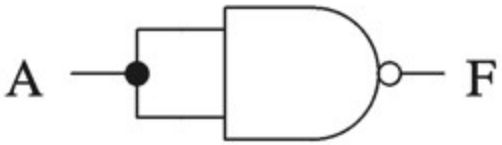
nareshnd.com.np

Basic Concepts (cont'd)

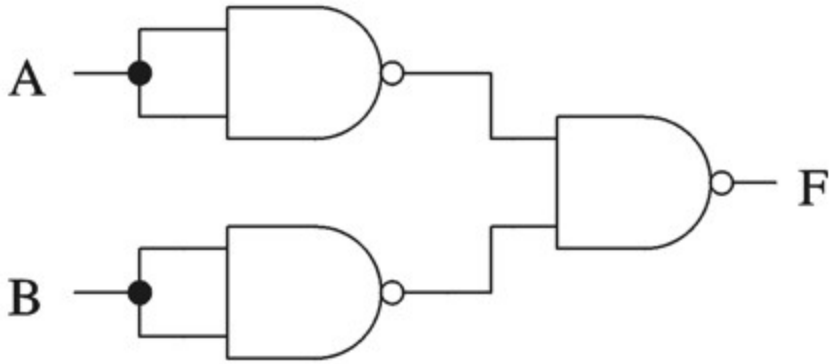
- **Proving NAND gate is universal**



AND gate



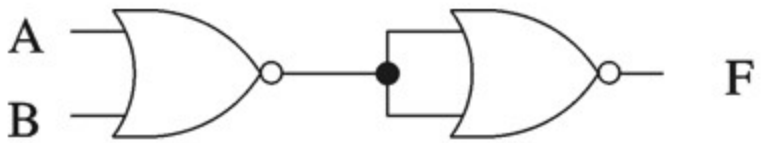
NOT gate



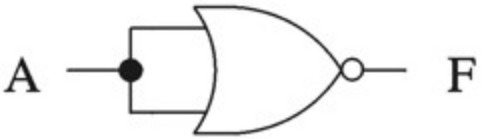
OR gate

Basic Concepts (cont'd)

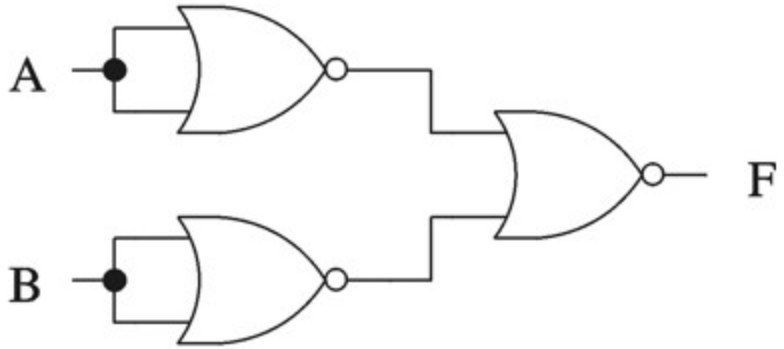
- Proving NOR gate is universal



OR gate



NOT gate

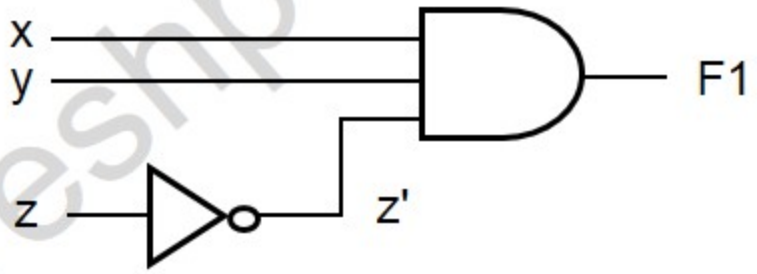


AND gate

Drawing Logic Circuit

- When a Boolean expression is provided, we can easily draw the logic circuit.
- Examples:

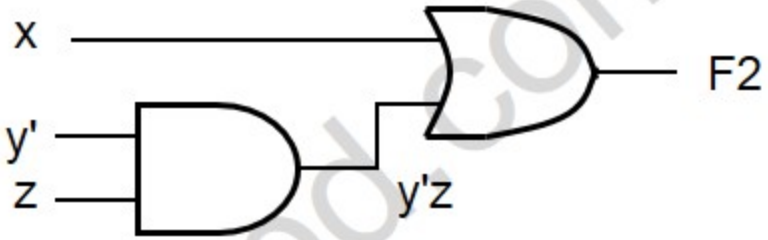
  (i) F1 = xyz' (note the use of a 3-input AND gate)



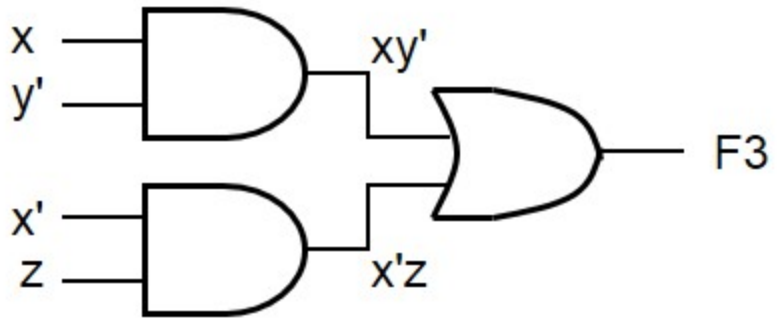
nareshpd.com.np

Drawing Logic Circuit

(ii) F2 = x + y'z (can assume that variables and their complements are available)



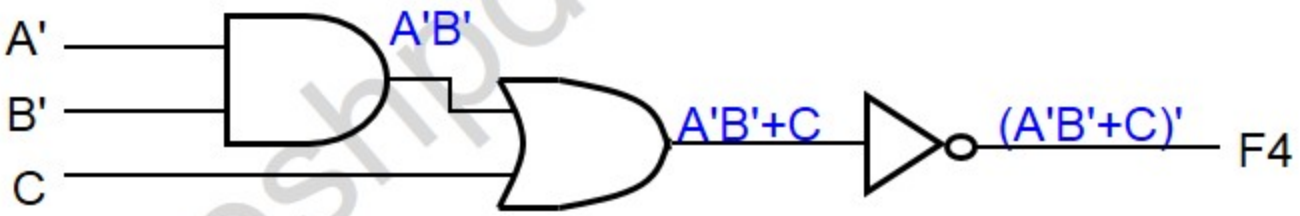
(iii) F3 = xy' + x'z



nareshpd.com

Analysing Logic Circuit

- When a logic circuit is provided, we can analyse the circuit to obtain the logic expression.
- Example: What is the Boolean expression of F4?



F4 = (A'B'+C)' = (A+B).C'

nareshpd.com.np

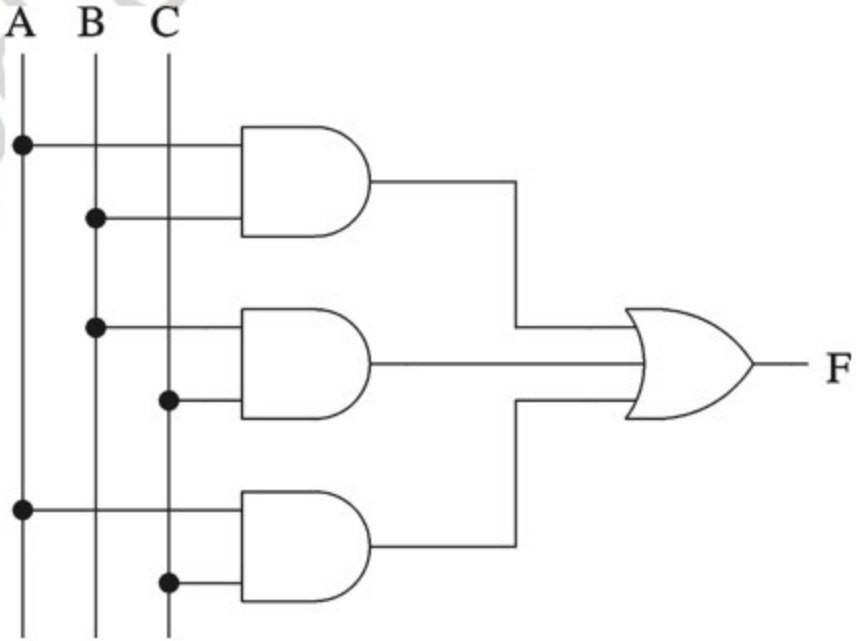
Logic Functions (cont'd)

3-input majority function

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- **Logical expression form**

$$F = A\,B + B\,C + A\,C$$



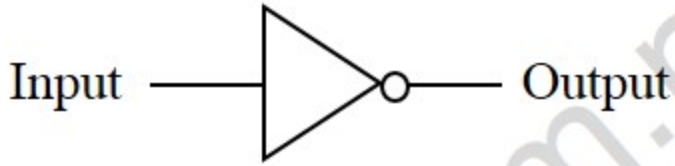
nareshp@com.np

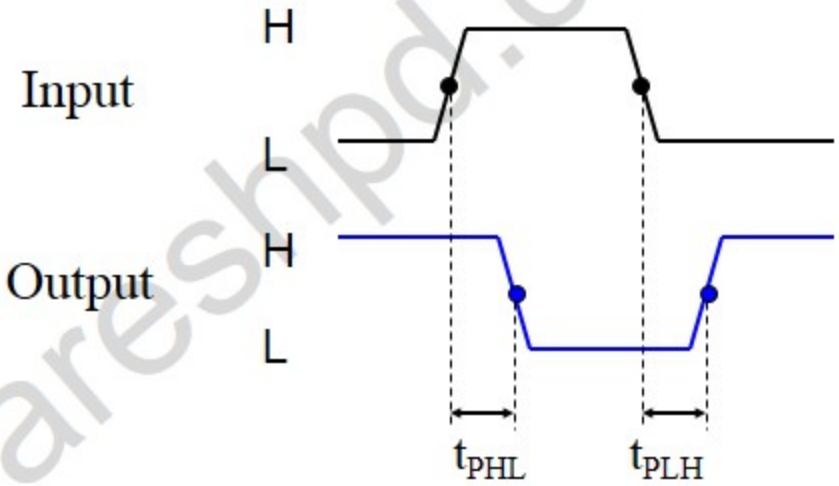
Propagation Delay

- Every logic gate experiences some delay (though very small) in propagating signals forward.
- This delay is called **Gate (Propagation) Delay**.
- Formally, it is the average transition time taken for the output signal of the gate to change in response to changes in the input signals.
- Three different propagation delay times associated with a logic gate:
  - t_{PHL}: output changing from the High level to Low level
  - t_{PLH}: output changing from the Low level to High level
  - $t_{PD}=(t_{PLH} + t_{PHL})/2$ (average propagation delay)

nareshpd.com.np

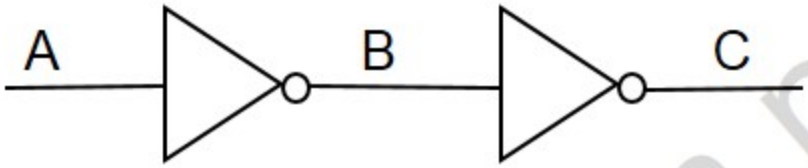
Propagation Delay



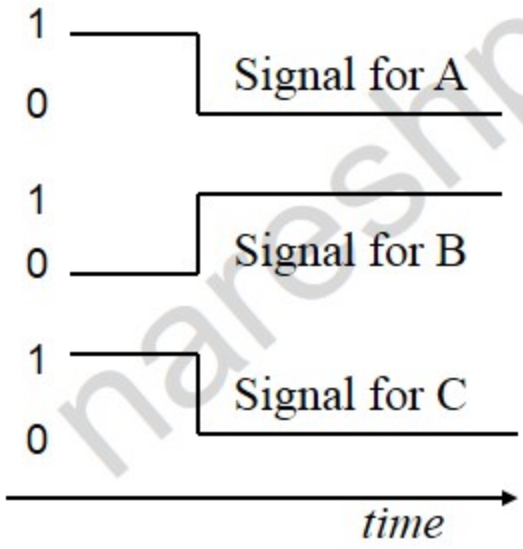


nareshpd.com.np

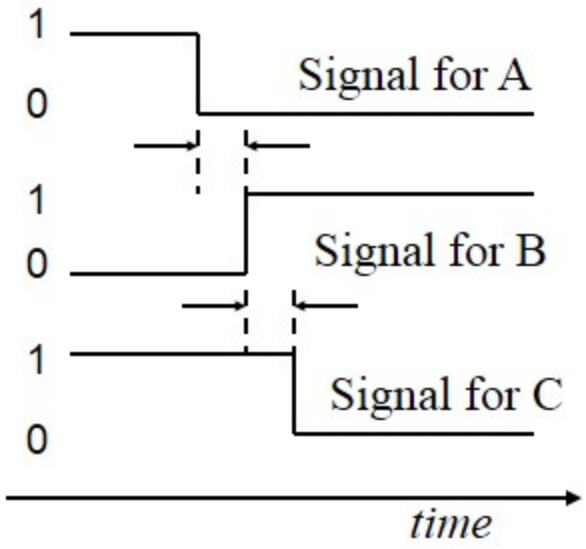
Propagation Delay



- Ideally, no delay:



- In reality, output signals normally lag behind input signals:



nareshpd.com.np

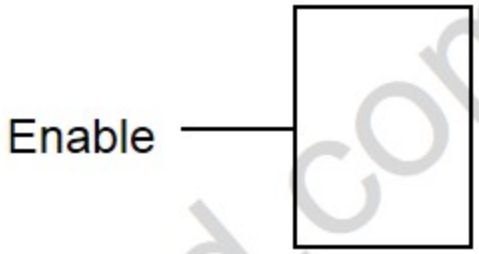
Positive & Negative Logic

- In logic gates, usually:
  - H (high voltage, 5V) = 1
  - L (low voltage, 0V) = 0
- This convention – **positive logic**.
- However, the reverse convention, **negative logic** possible:
  - H (high voltage) = 0
  - L (low voltage) = 1
- Depending on convention, same gate may denote different Boolean function.

rareshpd.com.np

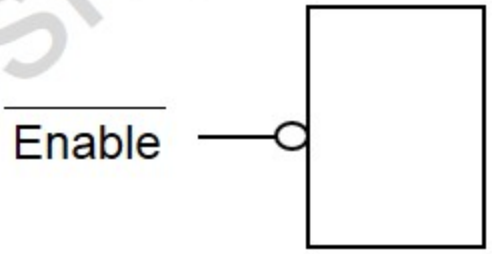
Positive & Negative Logic

Positive logic:



Active High:
0: Disabled
1: Enabled

Negative logic:

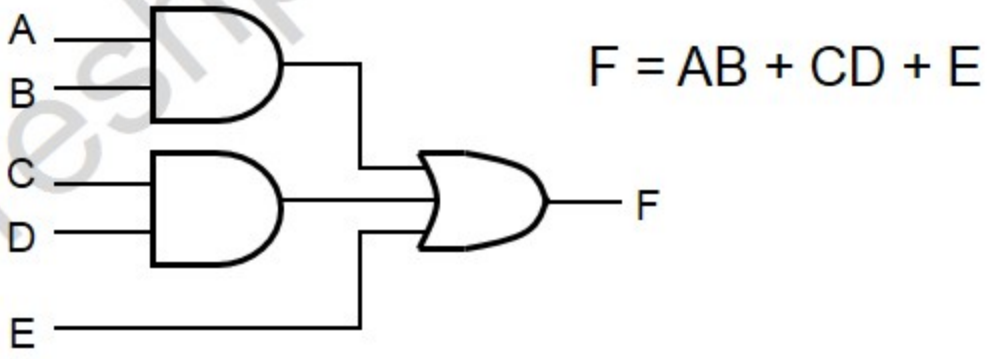


Active Low:
0: Enabled
1: Disabled

nareshpd.com.np

Implementation of SOP Expressions

- Sum-of-Products expressions can be implemented using:
  - 2-level AND-OR logic circuits
  - 2-level NAND logic circuits
- AND-OR logic circuit

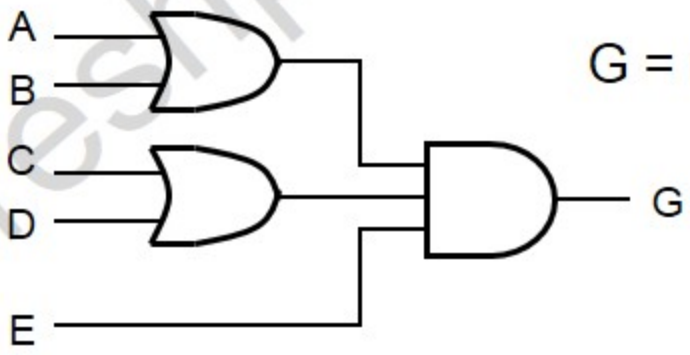


F = AB + CD + E

nareshpd.com.np

Implementation of POS Expressions

- Product-of-Sums expressions can be implemented using:
  - 2-level OR-AND logic circuits
  - 2-level NOR logic circuits
- OR-AND logic circuit



G = (A+B).(C+D).E

nareshpd.com.np

Minterm And Maxterm

- If a product term of SOP expressions contains every variable of that function either in true or complement form then it is defined as a **Minterm or standard product**.
- This **Minterm** will be true only for one combination of input values of the variable.
- Maxterm is a term of POS expression which contains all the variables of the function in true or complement form.
- The Maxterm has value 0 (false), for only one combination of input values.

nareshmd.com.np

Boolean Algebra

- Boolean algebra is an algebra for the manipulation of objects that can take on only two values, typically true and false
- It returns to its inventor, the famous mathematician and logician George Boole
- **Boolean variable**: it is a variable that can take only two values : 0 (false) or 1 (true)
  - Example: x, y, z, … Where, for instance, “x” could be 0 or 1
- **Boolean expressions**: Combination of Boolean variables and operators (AND, OR, NOT, …)
  - Example: x AND y, x OR y, …
- **Boolean function:** typically has one or more input values and yields a result, based on these input values, in the range {0,1}
  - Example: F(x, y, z) = (x AND y) OR z

hareshpd.com.np

Boolean Algebra...

- Frequently, a Boolean expression is not in its simplest form
- Recall from algebra the expression $2x + 6x$ can be simplified to $8x$
- Boolean expressions can also be simplified
- We need new identities, or laws, that apply to Boolean algebra instead of regular algebra
- These laws are grouped in the following table

nareshnd.com.np

Boolean Algebra...

Identity Name	AND Form	OR Form
Identity Law	$1x = x$	$0+x = x$
Null (or Dominance) Law	$0x = 0$	$1+x = 1$
Idempotent Law	$xx = x$	$x+x = x$
Inverse Law	$x\bar{x} = 0$	$x+\bar{x} = 1$
Commutative Law	$xy = yx$	$x+y = y+x$
Associative Law	$(xy)z = x(yz)$	$(x+y)+z = x+(y+z)$
Distributive Law	$x+yz = (x+y)(x+z)$	$x(y+z) = xy+xz$
Absorption Law	$x(x+y) = x$	$x+xy = x$
DeMorgan's Law	$(\overline{xy}) = \bar{x}+\bar{y}$	$(\overline{x+y}) = \bar{x}\bar{y}$
Double Complement Law	$\bar{\bar{x}} = x$	

Basic Identities of Boolean Algebra

Simplification of Boolean Expression

- Algebraic Simplification
- Karnaugh Maps
- Quine McCluskey Method

nareshpd.com.np

Simplification of Boolean Expression

- **Example 1:** Simplify the function F(*x*,*y*,*z*) = *xyz* + *xyz* + *xz*
  - F(x,y,z) = *xyz* + *xyz* + *xz*
    = *xyz* + *xz* (idempotent)
    = xz(y + 1) (Distributive)
    = xz(1) (Null)
    = xz (Identity)
  - The simplest form for F(x,y,z) is F(x,y,z) = xz

nareshpo.com.np

Simplification of Boolean Expression

- **Example 2:** Simplify the function $F(x,y,z) = \bar{x}yz + \bar{x}y\bar{z} + xz$
  - $F(x,y,z) = \bar{x}yz + \bar{x}y\bar{z} + xz$
    $= \bar{x}y(z + \bar{z}) + xz$ (Distributive)
    $= \bar{x}y\,(1) + xz$ (Inverse)
    $= \bar{x}y + xz$ (Identity)
  - The simplest form for F(x,y,z) is $F(x,y,z) = \bar{x}y + xz$

nareshod.com.np

Karnaugh Maps

-

nareshpd.com.np