- Boolean Functions (Switching Functions)

→ Boolean algebra is an algebra that deals with binary variables and logic operations.

→ A Boolean function described by an algebraic expression consists of binary variables, the constants 0 and 1, and the logic operation symbols.

→ For a given value of the binary variables, the function can be equal to either 1 or 0.

→ A Boolean function expresses the logical relationship between binary variables and is evaluated by determining the binary value of the expression for all possible values of the variables.

→ A Boolean function can be represented in a truth table.

→ The number of rows in the truth table is 2^n where n is the number of variables in the function

→ The binary combinations for the truth table are obtained from the binary numbers by counting from 0 through $2^n - 1$.

→ Consider a Boolean function $F_1 = x + y'z$

$F_2 = x'y'z + x'yz + xy'$

x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

Logic Gates

→ Logic gates are electronic circuits that operate on one or more input signals to produce an output signal.

→ Electrical signals such as voltages (or currents) are interpreted to be either of two recognizable values, 0 or 1.

→ Voltage-operated logic circuits respond to two seperate voltage levels that represent a binary variable equal to logic 1 or logic 0.

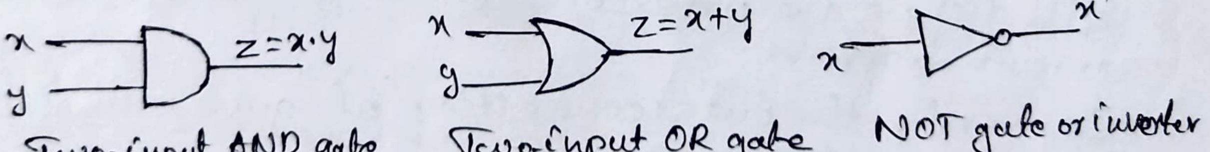
→ In practice, each voltage level has an acceptable range.

→ The input terminals of digital circuits accept binary signals within the allowable range and respond at the output terminals with binary signals that fall within the specified range.

→ The graphic symbols are used to designate gates

→ The gates are blocks of hardware that produce the equivalent of logic-1 or logic-0 output signals if input logic requirements are satisfied.

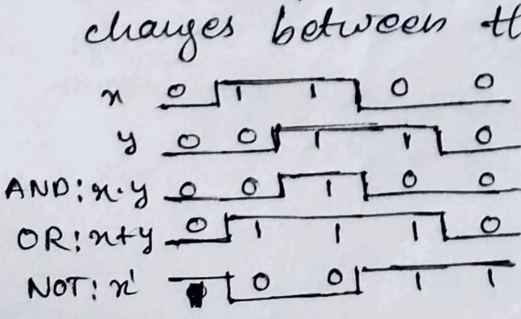
→ The input signals x and y in the AND and OR gates may exist in one of four possible states: 00, 01, 10 ~~or~~ and 11



Two-input AND gate   Two-input OR gate   NOT gate or inverter

→ The timing diagrams illustrate the idealized response of gate to the four input signal combinations.

→ The horizontal axis of the timing diagram represents the time, and the vertical axis shows the signal as it changes between the two possible voltage levels.



→ In reality, the transitions between logic values occur quickly, but not instantaneously.

→ The low level represents logic 0, the high level represents logic 1.

→ The AND gate responds with a logic 1 output signal when both input signals are logic 1.

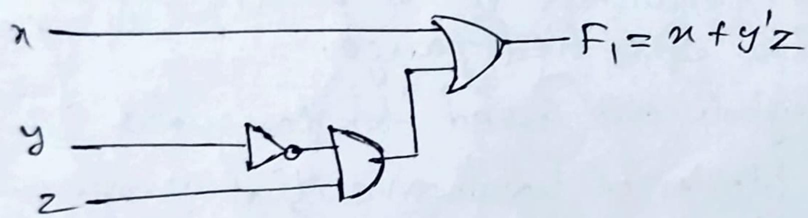
→ The OR gate responds with a logic 1 output signal if any input signal is logic 1.

→ The NOT gate is commonly referred to as an inverter, because the signal response in the timing diagram, shows that the output signal inverts the logic sense of the input signal.

→ AND and OR gates may have more than two inputs.

→ A Boolean function can be transformed from an algebraic expression into a circuit diagram composed of logic gates connected in a particular structure.

→ The logic-circuit diagram (also called a schematic) for $F_1 = x + y'z$ is shown below.



→ When function is in algebraic form, it can expressed in a variety of ways, all of which have equivalent logic.

→ The particular expression used to represent the function will dictate the interconnection of gates in the logic-circuit diagram.

→ Conversely, the interconnection of gates will dictate the logic expression.

→ By manipulating a Boolean expression according to the rules of Boolean algebra, it is sometimes possible to obtain a simpler expression for the same function and thus reduce the number of gates in the circuit and number of inputs to the gate.

→ Reducing the complexity and number of gates ~~because~~ can significantly reduce the cost of a circuit.

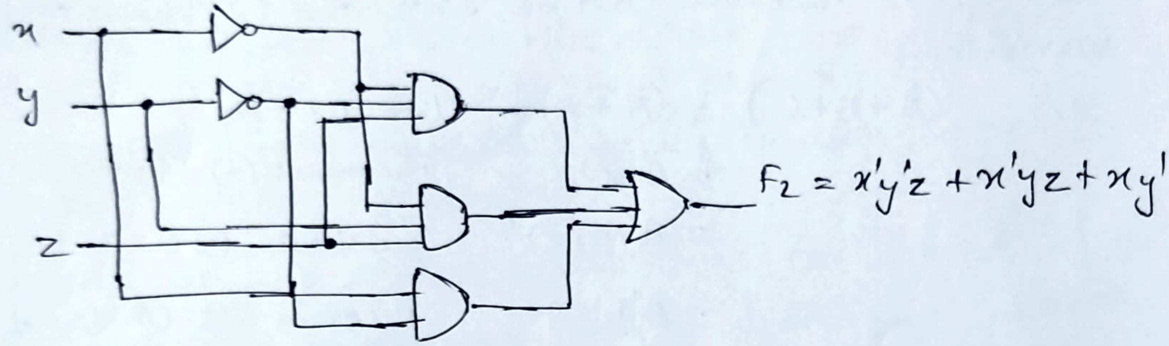
→ Consider the Boolean function:

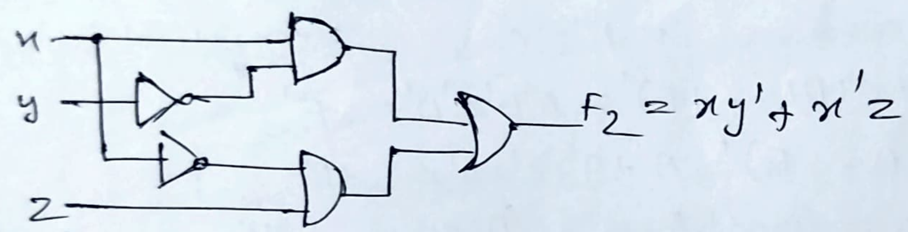
$$F_2 = x'y'z + x'yz + xy'$$

→ Consider the possible simplification of the function by applying some of the identities of Boolean algebra:

$$F_2 = x'y'z + x'yz + xy' = x'z(y'+y) + xy'$$
$$= x'z + xy'$$

→ Since both expressions produce the same truth table, they are equivalent.





→ Each circuit implements the same identical function, but the one with fewer gates and fewer inputs to gates is preferable because it requires fewer wires and components.

Q → Simplify the following Boolean functions to a minimum number of literals (A literal is a single variable within a term, in complemented or uncomplemented form.

1) $x(x'+y) = xx' + xy = 0 + xy = xy$

2) $x + x'y = (x+x')(x+y) = 1(x+y) = x+y$

3) $(x+y)(x+y') = x + xy' + xy + 0 = x(1+y+y') = x$

4) $xy + x'z + yz = xy + x'z + yz(x+x')$
$= xy + x'z + xyz + x'yz$
$= xy(1+z) + x'z(1+y)$
$= xy + x'z$

5) $(x+y)(x'+z)(y+z) = (x+y)(x'+z)(xx'+y+z)$
$= (x+y)(x'+z)(x+y+z)(x'+y+z)$
$= (x+y)(1+z)(x'+z)(1+y)$
$= (x+y)(x'+z)$

Complement of a Function

→ The complement of a function F is F' and is obtained from an interchange of 0's for 1's and 1's for 0's in the value of F.

→ The complement of a function may be derived algebraically through DeMorgan's theorems.

→ DeMorgan's theorems can be extended to three or more variables.

$$(A+B+C)' = (A+x)' \quad \text{let } B+C = x$$
$$= A'x' \quad \text{Theorem 5(a) (DeMorgan)}$$
$$= A'(B+C)' \quad \text{Substitute } B+C = x$$
$$= A'(B'C') \quad \text{Theorem 5(a) (DeMorgan)}$$
$$= A'B'C' \quad \text{Theorem 4(b) (associative)}$$

$$(A+B+C+D+\cdots+F)' = A'B'C'D'\ldots F'$$
$$(ABCD\ldots F)' = A'+B'+C'+D'+\cdots+F'$$

→ The generalized form of DeMorgan's theorems states that the complement of a function is obtained by interchanging AND and OR operators and complementing each literal.

Q. Find the complement of the function $F_1 = x'yz' + x'y'z$ and $F_2 = x(y'z' + yz)$.

$$F_1' = (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x+y'+z)(x+y+z')$$
$$F_2' = [x(y'z'+yz)]' = x' + (y'z'+yz)' = x' + (y'z')'(yz)'$$
$$= x' + (y+z)(y'+z')$$
$$= x' + yz' + y'z$$

Q. Find the complement of the functions F_1 and F_2 by taking their duals and complementing each literal

1) $F_1 = x'yz' + x'y'z$

Dual of F_1 is $(x'+y+z')(x'+y'+z)$

complement each literal: $(x+y'+z)(x+y+z') = F_1'$

2) $F_2 = x(y'z'+yz)$

Dual of F_2 is: $x + (y'+z')(y+z)$

Complement each literal: $x' + (y+z)(y'+z') = F_2'$