Unit 3
Boolean Algebra

Lecture 1

Basic Gates

Binary logic

- Binary logic consists of binary variables and logical operations.
- The variables are designated by letters of the alphabet such as *A, B,* C, *x, y,* Z, etc., with each variable having two and only two distinct possible values: 1 and 0. There are three basic logical operations: AND, OR, and NOT.

a. AND: This operation is represented by a dot or by the absence of an operator. For example, *x.y* = z or *xy* = z is read "*x* AND *y* is equal to *z.*" The logical operation AND is interpreted to mean that z= 1 if and only if *x* = 1 *and y* = 1; otherwise z = 0. (Remember that *x, y,* and z are binary variables and can be equal either to 1 or 0, and nothing else.)

b OR: This operation is represented by a plus sign. For example, *x* + *y* = z is read "*x* OR *y* is equal to *z,*" meaning that z = 0 if *x* = 0 *or* if *y* = 0 otherwise z = 1.

c. NOT: This operation is represented by a prime (sometimes by a bar). For example, *x'* = z is read "not *x* is equal to *z,*" meaning that z is what *x* is not. In other words, if *x* = 1, then z = 0; but if *x* = 0, then z = 1.

- These definitions may be listed in a compact form using truth tables.

Truth Table:

A **truth table** is a table of all possible combinations of the variables showing the relation between the values that the variables may take and the result of the operation.

Truth Tables of Logical Operations

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

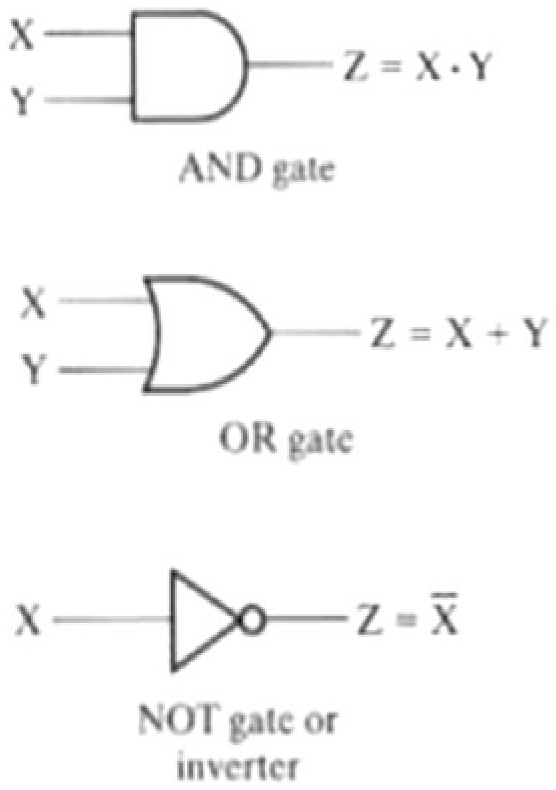
NOT

x	x'
0	1
1	0

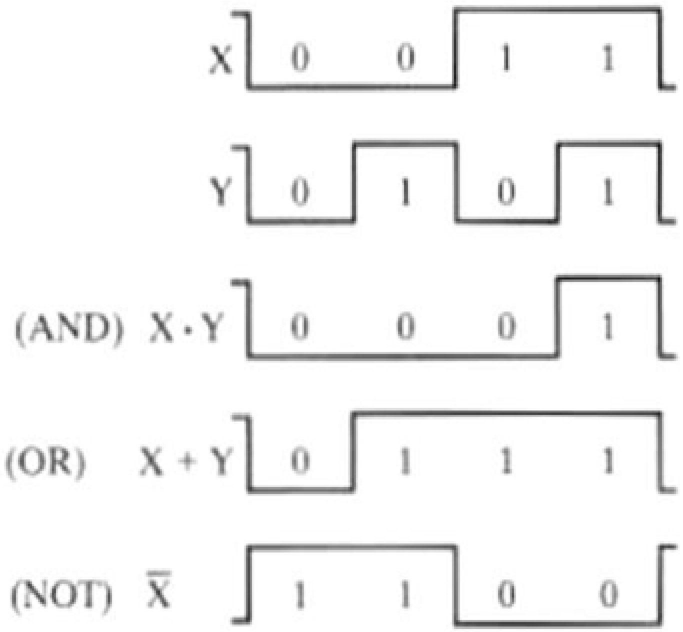
- Binary logic should not be confused with binary arithmetic (However we use same symbols here). You should realize that an arithmetic variable designates a number that may consist of many digits.
- A logic variable is always either 1 or 0. For example, in binary arithmetic, 1 + 1 = 10 (read: "one plus one is equal to 2"), whereas in binary logic, we have 1 + 1 = 1 (read: "one OR one is equal to one").

Basic 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 exist throughout a digital system in either one of two recognizable values (bi-state 0 or 1).
- Voltage-operated circuits respond to two separate voltage ranges (Example of voltage ranges is discussed in unit 1) that represent a binary variable equal to logic 1 or logic



- These circuits, called *gates,* are blocks of hardware that produce a logic-1 or logic-0 output signal if input logic requirements are satisfied.
- Note that four different names have been used for the same type of circuits: digital circuits, switching circuits, logic circuits, and gates.
- AND and OR gates may have more than two inputs.
- NOT gate is single input circuit, it simply inverts the input.



(b) **Timing diagram**

The two input signals X and Y to the AND and OR gates take on one of four possible combinations: 00, 01, 10, or 11. These input signals are shown as timing diagrams, together with the timing diagrams for the corresponding output signal for each type of gate. The horizontal axis of a timing diagram represents time, and the vertical axis shows a signal as it changes between the two possible voltage levels. The low level represents logic 0 and the high level represents logic I. 1l1e 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 either input signal is logic-1.

Boolean algebra

- In 1854 George Boole introduced a systematic treatment of logic and developed for this purpose an algebraic system now called *Boolean algebra.*
- In 1938 C. E. Shannon introduced a two-valued Boolean algebra called *switching algebra,* in which he demonstrated that the properties of bistable electrical switching circuits can be represented by this algebra.
- Thus, the mathematical system of binary logic is known as **Boolean or switching algebra**.
- This algebra is conveniently used to describe the operation of complex networks of digital circuits. Designers of digital systems use Boolean algebra to transform circuit diagrams to algebraic expressions and vice versa.
- For any given algebra system, there are some initial assumptions, or postulates, that the system follows.

Basic theorems and Properties of Boolean algebra

Duality

- Postulates of Boolean algebra are found in pairs; one part may be obtained from the other if the binary operators and the identity elements are interchanged. This important property of Boolean algebra is called the **duality principle**.
- It states that "***Every algebraic expression deducible from the postulates of Boolean algebra remains valid if the operators and identity elements are interchanged***".
- In a two- valued Boolean algebra, the identity elements and the elements of the set *B* are the same: 1 and 0. If the ***dual*** of an algebraic expression is desired, we simply interchange OR and AND operators and replace 1's by 0's and 0's by 1's.

Basic Theorems

- The theorems, like the postulates, are listed in pairs; each relation is the dual of the one paired with it. The postulates are basic axioms of the algebraic structure and need no proof.
- The theorems must be proven from the postulates. six theorems of Boolean algebra are given below:

Theorem1:	Idempotence	(a) $x + x = x$	(b) $x.x = x$	One variable theorems
Theorem2:	Existence: 0&1	(a) $x + 1 = 1$	(b) $x.0 = 0$	
Theorem3:	Involution	$(x')' = x$		
Theorem4:	Associative	(a) $x + (y + z) = (x + y) + z$	(b) $x(yz) = (xy)z$	2 or 3 variable theorems
Theorem5:	Demorgan	(a) $(x + y)' = x'y'$	(b) $(xy)' = x' + y'$	
Theorem6:	Absorption	(a) $x + xy = x$	(b) $x(x + y) = x$	