DIGITAL ELECTRONICS CIRCUIT(BCA 103)

DEPARTMENT OF COMPUTER SCIENCE
PROGRAMME: BCA

CENTRAL UNIVERSITY OF ODISHA
KORAPUT

BINARY LOGIC

- Binary logic deals with variables that take on two discrete values and with operations that assume logical meaning.
- The two values the variables assume may be *true* and *false*, *yes* and *no* or 1 and 0.
- Binary logic consists of binary variables and a set of 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. Each operation produces a binary result, denoted by *z*.

- Binary logic resembles binary arithmetic, and the operations AND and OR have similarities to multiplication and addition, respectively.
- The symbols used for AND and OR are the same as those used for multiplication and addition.
- However, **binary logic should not be confused with binary arithmetic.**
- An arithmetic variable designates a number that may consist of many digits. But a logic variable is always either 1 or 0.
- Ex: **Binary Arithmetic :** 1 + 1 = 10 **Binary Logic :** 1 + 1 = 1

- 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				OR				NOT	
x	y	$x \cdot y$		x	y	$x + y$		x	x'
0	0	0		0	0	0		0	1
0	1	0		0	1	1		1	0
1	0	0		1	0	1			
1	1	1		1	1	1			

Boolean Algebra

- Boolean algebra may be defined with a set of elements, a set of operators, and a number of unproved axioms or postulates.
- A *set* of elements is any collection of objects, usually having a common property.
- If S is a set, and x and y are certain objects, then the notation $x \varepsilon S$ means that x is a member of the set S and y x S means that y is not an element of S.
- The postulates of a mathematical system form the basic assumptions from which it is possible to deduce the rules, theorems, and properties of the system.

Common Postulates

- ***Closure:*** A set S is closed with respect to a binary operator if, for every pair of elements of S, the binary operator specifies a rule for obtaining a unique element of S.

Ex: the set of natural numbers $N = \{1, 2, 3, 4, c\}$ is closed with respect to the binary operator + by the rules of arithmetic addition, since, for any $a, b \varepsilon N$, there is a unique $c \varepsilon N$ such that $a + b = c$.

- ***Associative law:*** A binary operator * on a set S is said to be associative whenever $(x * y) * z = x * (y * z)$ for all $x, y, z, \varepsilon S$.
- ***Commutative law:*** A binary operator * on a set S is said to be commutative whenever $x * y = y * x$ for all $x, y \varepsilon S$.
- ***Identity element:*** A set S is said to have an identity element with respect to a binary operation * on S if there exists an element $e \varepsilon S$ with the property that $e * x = x * e = x$ for every $x \varepsilon S$.

- ***Inverse:*** A set S having the identity element e with respect to a binary operator * is said to have an inverse whenever, for every $x \varepsilon S$, there exists an element $y \varepsilon S$
such that $x * y = e$.
- ***Distributive law:*** If * and . are two binary operators on a set S, * is said to be distributive over . whenever $x * (y . z) = (x * y) . (x * z)$

Two-Valued Boolean Algebra

- A two-valued Boolean algebra is defined on a set of two elements, $B = \{0, 1\}$, with rules for the two binary operators + and .

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

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

x	x'
0	1
1	0

- Distributive law: $x \,.\, (y + z) = (x \,.\, y) + (x \,.\, z)$

x	y	z
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

$y + z$	$x \cdot (y + z)$
0	0
1	0
1	0
1	0
0	0
1	1
1	1
1	1

$x \cdot y$	$x \cdot z$	$(x \cdot y) + (x \cdot z)$
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	1	1
1	0	1
1	1	1

Theorems of Boolean Algebra

- *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.
- 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.

Postulates and Theorems of Boolean Algebra

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

- The postulates are basic axioms of the algebraic structure and need no proof.
- The theorems must be proven from the postulates.
- **Theorem: x + x = x.**

Statement	Justification
L.H.S: $x + x = (x + x) .1$	$(x . 1 = 1)$
$= (x + x)(x + x')$	$(x + x' = 1)$
$= x + xx'$	$(x + yz = (x + y) . (x + z))$
$= x + 0$	$(x . x' = 0)$
$= x$	$(x + 0 = x)$

- **Theorem: $x . x = x$.**

Statement	Justification
L.H.S: $x . x = (x . x) + 0$	$(x + 0 = x)$
$= (x . x) + (x . x')$	$(x . x' = 0)$
$= x . (x + x')$	$(x . (y + z)) = (x . y) + (x . z))$
$= x . 1$	$(x + x' = 1)$
$= x$	$(x . 1 = x)$

- **Theorem: x + 1 = 1.**

Statement	**Justification**
L.H.S: $x + 1 = (x + 1) . 1$	$(x . 1 = x)$
$= (x + 1) . (x + x')$	$(x + x' = 1)$
$= x + (1 . x')$	$(x + yz = (x + y) . (x + z))$
$= x + x'$	$(1 . x = x)$
$= 1$	$(x + x' = 1)$

- **Theorem: x + xy = x.**

Statement	**Justification**
L.H.S: $x + xy = (x . 1) + (x . y)$	$(x . 1 = x)$
$= x . (1 + y)$	$((x . y) + (x . z) = x . (y + z))$
$= x . (y + 1)$	$(x + y = y + x)$
$= x . 1$	$(x + 1 = 1)$
$= x$	$(x . 1 = x)$

-

- The theorems of Boolean algebra can be proven by means of truth tables.
- Theorem: $x + xy = x$ DeMorgan's Theorem: $(x+y)' = x'y'$

x	y	xy	x+ xy
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

x	y	x + y	(x+ y)'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

x'	y'	x'y'
1	1	1
1	0	0
0	1	0
0	0	0

- The operator precedence for evaluating Boolean expressions is :

(1) parentheses, (2) NOT, (3) AND, and (4) OR.

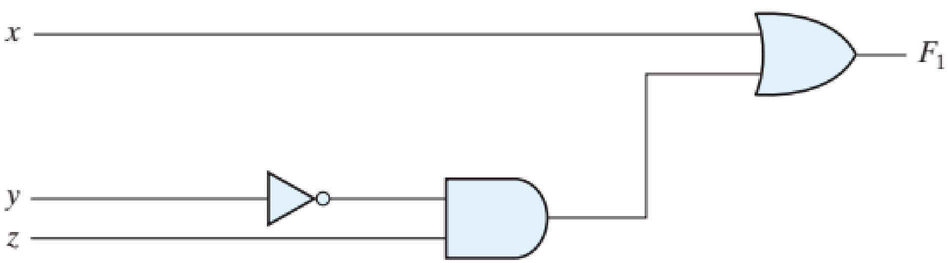
- In other words, expressions inside parentheses must be evaluated before all other operations.

Boolean Function

- 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.
- *Ex:* $F_1 = x + y'z$
- The function F_1 is equal to 1 if x is equal to 1 or if both y and z are equal to 1.
- F_1 is equal to 0 otherwise. The complement operation dictates that when $y' = 1$, $y = 0$.
- Therefore, $F_1 = 1$ if $x = 1$ or if $y = 0$ and $z = 1$.
- 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 $2^n - 1$.
-

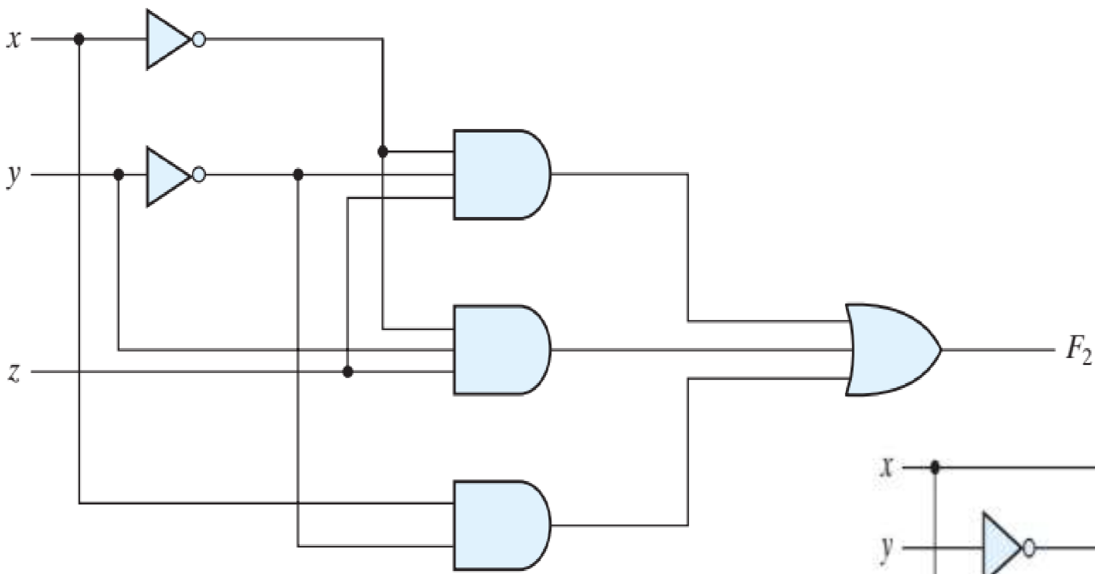
X	Y	Z	F_1	F_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0



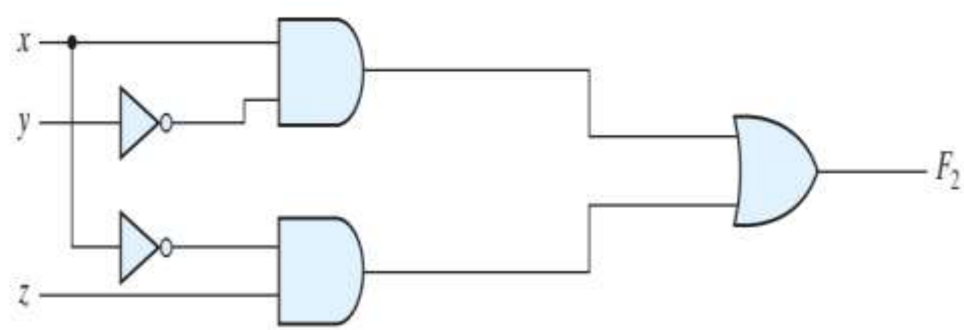
$F_1 = X + Y'Z$

- 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 is also called a schematic.
- In logic-circuit diagrams, the variables of the function are taken as the inputs of the circuit and the binary variable F_1 is taken as the output of the circuit.
- The schematic expresses the relationship between the output of the circuit and its inputs.
- There is only one way that a Boolean function can be represented in a truth table.
- However, when the function is in algebraic form, it can be expressed in a variety of ways, all of which have equivalent logic.

- **Ex:** $\boldsymbol{F_2 = x'y'z + x'yz + xy'}$
- The possible simplification of the function by applying some of the identities of Boolean algebra:
  $\boldsymbol{F_2 = x'y'z + x'yz + xy' = x'z(y' + y) + xy' = {\color{red}x'z + xy'}}$
- By means of a truth table, it is possible to verify that the two expressions are equivalent.



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



(b) $F_2 = xy' + x'z$

Algebraic Manipulation

- When a Boolean expression is implemented with logic gates, each term requires a gate and each variable within the term designates an input to the gate.
- We define a *literal* to be a single variable within a term, in complemented or un-complemented form.

- $F_2 = x'y'z + x'yz + xy'$ (Function has 3 terms and 8 literals)
- $F_2 = x'z + xy'$ (Function has 2 terms and 4 literals)

- By reducing the number of terms, the number of literals, or both in a Boolean expression, it is often possible to obtain a simpler circuit.
- The manipulation of Boolean algebra consists mostly of reducing an expression for the purpose of obtaining a simpler circuit.

- Simplify the following Boolean functions to a minimum number of literals.

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

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

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

4. $$\begin{aligned} xy + x'z + yz &= xy + x'z + yz(x + x') \\ &= xy + x'z + xyz + x'yz \\ &= xy(1 + z) + x'z(1 + y) \\ &= xy + x'z. \end{aligned}$$

5. $(x + y)(x' + z)(y + z) = (x + y)(x' + z)$

Functions 4 and 5 are together known as the *consensus theorem*.

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.
- $(A + B + C)' = (A + x)'$ let $B + C = x$

  $= A'x'$ by DeMorgan theorem (A+B)'=A'B'

  $= A'(B + C)'$ substitute $B + C = x$

  $= A'(B'C')$ by DeMorgan theorem (A+B)'=A'B'

  $= A'B'C'$ by associative

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.

$$\boldsymbol{(A + B + C + D + \ldots\ldots. + F)' = A'B'C'D'\ldots\ldots.\ F'}$$

$$\boldsymbol{(ABCD\ \ldots\ldots\ F)' = A' + B' + C' + D' + \ldots\ldots + F'}$$

- Find the complement of the functions:

  $F_1 = x'yz' + x'y'z$ **and** $F_2 = x(y'z' + yz)$
- **Ans:** $F'_1 = (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x + y' + z)(x + y + z')$

- $F'_2 = [x(y'z' + yz)]' = x' + (y'z' + yz)' = x' + (y'z')'(yz)'$

  $= x' + (y + z)(y' + z')$

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

- A simpler procedure for deriving the complement of a function is to take the dual of the function and complement each literal.
- The dual of a function is obtained from the interchange of AND and OR operators and 1's and 0's.

- A simpler procedure for deriving the complement of a function is to take the dual of the function and complement each literal.
- The dual of a function is obtained from the interchange of AND and OR operators and 1's and 0's.
- Find the complement of the functions:

  $\boldsymbol{F_1 = x'yz' + x'y'z}$ **and** $\boldsymbol{F_2 = x(y'z' + yz)}$
- **Ans:**
- The dual of F_1 is $\boldsymbol{(x' + y + z')(x' + y' + z)}$.
  Complement each literal: $\boldsymbol{(x + y' + z)(x + y + z') = F'_1}$

- $F_2 = x(y'z' + yz)$
  The dual of F_2 is $\boldsymbol{x + (y' + z')(y + z)}$
  Complement each literal: $\boldsymbol{x' + (y + z)(y' + z') = F'_2}$

-

BINARY LOGIC

- Binary logic deals with variables that take on two discrete values and with operations that assume logical meaning.
- The two values the variables assume may be *true* and *false*, *yes* and *no* or 1 and 0.
- Binary logic consists of binary variables and a set of 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. Each operation produ

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