# Boolean Algebra

# 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 called by different names
  - 1 and 0
  - *true and false*
  - *yes and no*
  - *…*

# BINARY LOGIC

- 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,
  - NOT.
- Each operation produces a binary result( say *z)*

# AND

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

# OR

- This operation is represented by a plus sign.
- For example,
  - $x + y = z$
  - *is read "x OR y is equal to z,"*
- If the result of the operation *x+y is z, the logical operation* OR is interpreted to mean that
  - $z = 1$ *if* $x = 1$ *or if* $y = 1$ *or if both* $x = 1$ and $y = 1$.
  - $z = 0$ *if both* $x = 0$ *and* $y = 0$

# NOT

- This operation is represented by a prime or sometimes by an overbar.
- For example,
  - $x' = z$
  - $\overline{x} = z$
  - *is read "not x is equal to z,"*
- If the result of the operation *x' is z, the logical operation* NOT is interpreted to mean that
  - if *x = 1, then z = 0,*
  - *if x = 0, then z = 1.*
- *The NOT* operation is also referred to as the complement operation, since it changes a 1 to 0 and a 0 to 1

# Binary logic Vs Binary arithmetic

- Binary logic resembles binary arithmetic,
  - AND operation is similar to multiplication
  - OR operation is similar to addition
- Symbols used for
  - AND is same as multiplication
  - OR is same as addition
- Binary arithmetic variable designates a number that may consist of many digits.
  - in binary arithmetic, we have 1 + 1 = 10 (read “one plus one is equal to 2”),
- A logic variable is always either 1 or 0.
  - in binary logic, we have 1 + 1 = 1 (read “one OR one is equal to one”).

# Truth Table

- Definitions of logical operations may be listed in a compact form called *truth tables.*
- *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.

**AND**

| $x$ | $y$ | $x \cdot y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**OR**

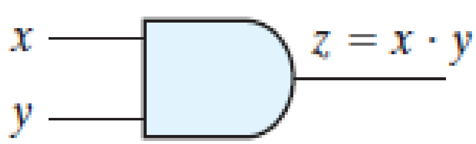
| $x$ | $y$ | $x + y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**NOT**

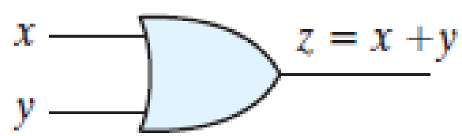
| $x$ | $x'$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

# Logic Gates & Graphic symbols

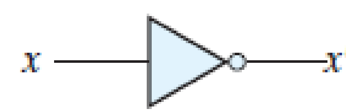
- Logic gates are electronic circuits that operate on one or more input signals to produce an output signal.
- The gates are blocks of hardware that produce the equivalent of logic-1 or logic-0 output signals if input logic requirements are satisfied.
- The graphic symbols used to designate the three types of gates (AND, OR, NOT).



Two-input AND gate



Two-input OR gate



NOT gate or inverter

# Boolean algebra

- Defined with a set of elements, a set of operators, and a number of unproved axioms or postulates.
- In 1854, George Boole developed an algebraic system now called *Boolean algebra*
- *In* 1938, Claude E. Shannon introduced a two-valued Boolean algebra called *switching algebra that represented the properties of bistable electrical switching circuits*

# Boolean Algebra(Formal Definition)

- Boolean algebra is an algebraic structure defined by a set of elements, *B, together* with two binary operators, + and ● , provided that the following (Huntington) postulates are satisfied:
  1. (a) The structure is closed with respect to the operator +.
     (b) The structure is closed with respect to the operator ● .
  2. (a) The element 0 is an identity element with respect to +; that is, $x + 0 = 0 + x = x$.
     (b) The element 1 is an identity element with respect to ● ; that is, $x \bullet 1 = 1 \bullet x = x$.
  3. (a) The structure is commutative with respect to +; that is, $x + y = y + x$.
     (b) The structure is commutative with respect to ● ; that is, $x \bullet y = y \bullet x$.
  4. (a) The operator ● is distributive over +; that is, $x \bullet (y + z) = (x \bullet y) + (x \bullet z)$.
     (b) The operator + is distributive over ● ; that is, $x + (y \bullet z) = (x + y) \bullet (x + z)$.
  5. For every element $x \in B$, *there exists an element* $x' \in B$ *(called the complement of x)* such that (a) $x + x' = 1$ *and* (b) $x \bullet x' = 0$.
  6. There exist at least two elements $x, y \in B$ *such that* $x \neq y$.

# Two-valued Boolean Algebra

- Two-valued Boolean algebra has applications in set theory (the algebra of classes) and in propositional logic.
- Boolean algebra can be applied to gate-type circuits commonly used in digital devices and computers.
- A two-valued Boolean algebra is defined on a set of two elements, *B = {0, 1}, with rules* for the two binary operators + and . as shown in the following operator tables (the rule for the complement operator is for verification of postulate 5):

**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 |

# BASIC THEOREMS AND PROPERTIES OF BOOLEAN ALGEBRA

- **Duality**
  - This important property of Boolean algebra is called the *duality principle*
    - *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.*
  - *The duality principle has many applications.*
  - *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) | $x(x + y) = x$ |

# Operator Precedence

- The operator precedence for evaluating Boolean expressions is
  - (1) parentheses,
  - (2) NOT
  - (3) AND
  - (4) OR.

# Proving Theorems

$x + x = x.$

| Statement | Justification |
|---|---|
| $x + x = (x + x) \cdot 1$ | postulate 2(b) |
| $= (x + x)(x + x')$ | 5(a) |
| $= x + xx'$ | 4(b) |
| $= x + 0$ | 5(b) |
| $= x$ | 2(a) |

$x \cdot x = x.$

| Statement | Justification |
|---|---|
| $x \cdot x = xx + 0$ | postulate 2(a) |
| $= xx + xx'$ | 5(b) |
| $= x(x + x')$ | 4(a) |
| $= x \cdot 1$ | 5(a) |
| $= x$ | 2(b) |

# Proving Theorems

$x + 1 = 1.$

| Statement | Justification |
|---|---|
| $x + 1 = 1 \cdot (x + 1)$ | postulate 2(b) |
| $= (x + x')(x + 1)$ | 5(a) |
| $= x + x' \cdot 1$ | 4(b) |
| $= x + x'$ | 2(b) |

**$(A + B)(A + C) = A + BC$**

| | |
|---|---|
| $(A + B)(A + C) = AA + AC + AB + BC$ | Distributive law |
| $= A + AC + AB + BC$ | $AA = A$ |
| $= A(1 + C) + AB + BC$ | Factoring (distributive law) |
| $= A \cdot 1 + AB + BC$ | $1 + C = 1$ |
| $= A(1 + B) + BC$ | Factoring (distributive law) |
| $= A \cdot 1 + BC$ | $1 + B = 1$ |
| $= A + BC$ | $A \cdot 1 = A$ |

# Proving theorems

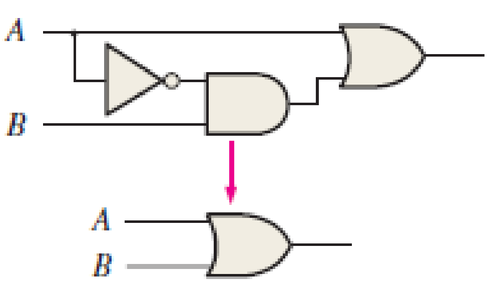
$A + \overline{A}B = A + B$

$$
\begin{aligned}
A + \overline{A}B &= (A + AB) + \overline{A}B && A = A + AB \\
&= (AA + AB) + \overline{A}B && A = AA \\
&= AA + AB + A\overline{A} + \overline{A}B && \text{adding } A\overline{A} = 0 \\
&= (A + \overline{A})(A + B) && \text{Factoring} \\
&= 1 \cdot (A + B) && A + \overline{A} = 1 \\
&= A + B && \text{drop the 1}
\end{aligned}
$$

$A + \overline{A}B = A + B.$

| $A$ | $B$ | $\overline{A}B$ | $A + \overline{A}B$ | $A + B$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |

equal



# Proving theorems

Apply DeMorgan's theorems to each expression:

**(a)** $\overline{(\overline{A + B}) + \overline{C}}$

**(b)** $\overline{(\overline{A} + B) + CD}$

**(c)** $\overline{(A + B)\overline{C}\,\overline{D} + E + \overline{F}}$

**Solution**

**(a)** $\overline{(\overline{A + B}) + \overline{C}} = (\overline{\overline{A + B}})\overline{\overline{C}} = (A + B)C$

**(b)** $\overline{(\overline{A} + B) + CD} = (\overline{\overline{A} + B})\overline{CD} = (\overline{\overline{A}}\,\overline{B})(\overline{C} + \overline{D}) = A\overline{B}(\overline{C} + \overline{D})$

**(c)** $\overline{(A + B)\overline{C}\,\overline{D} + E + \overline{F}} = (\overline{(A + B)\overline{C}\,\overline{D}})(\overline{E + \overline{F}}) = (\overline{A}\,\overline{B} + C + D)\overline{E}F$

# BOOLEAN 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.
- Example: Consider the Boolean function

$$F1 = x + y'z$$

  - *F1 is equal to 1 if x is equal to 1 or if both y' and z are equal to 1.*
  - *F1 is equal* to 0 otherwise.
  - The complement operation dictates that when $y = 1$, $y = 0$.
    - *Therefore,* $F1 = 1$ *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.

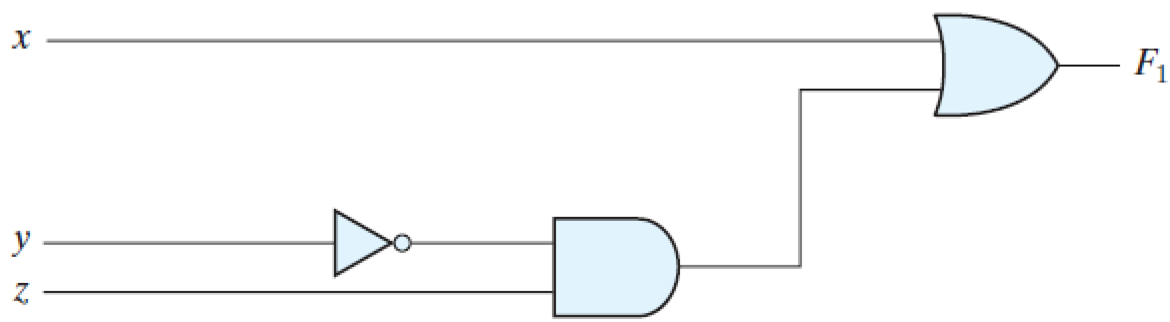
# BOOLEAN FUNCTIONS

- 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.*
  - *Example:* Truth table for the function *F1(F1 = x + y'z)*

| $x$ | $y$ | $z$ | $F_1$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

# Logic circuit diagram or Schematic

- 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 *F1*(***F1 = x + y'z)*** *is shown below:*

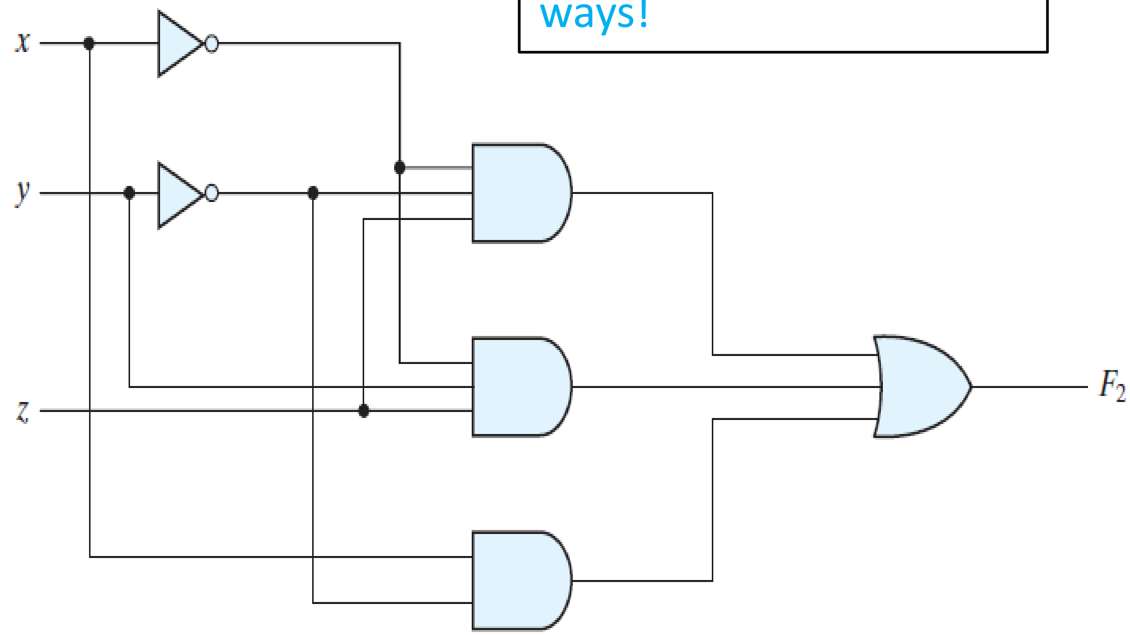


# Example

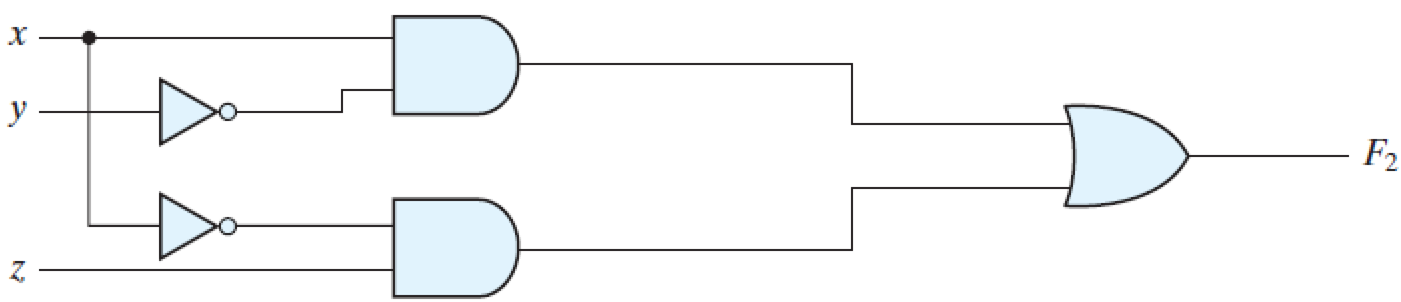
- *Function: F2 = x'y'z + x'yz + xy'*
  - *Simplification:*

$$F2 = x'y'z + x'yz + xy'$$
$$= x'z(y' + y) + xy'$$
$$= x'z + xy'$$

Note the number of gates required for implementing the same function in two different ways!




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




$F_2 = xy' + x'z$