جامعة المستقبل
AL MUSTAQBAL UNIVERSITY
كلية الهندسة والتقنيات الهندسية

Communication Technical Engineering Department

1st Stage

Digital Logic- UOMU028021

Lecture 5 – Boolean algebra and DeMorgan's theorem

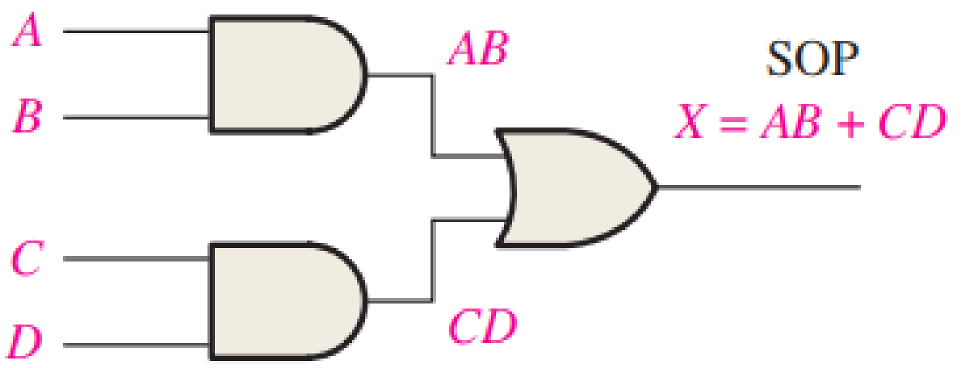
Dr. Mohammed Fadhil
PhD in Computer Networks
Email: mohammed.fadhil1@uomus.edu.iq

Basic Combinational Logic Circuits

- In this lecture, you learned that SOP (Sum of Products) expressions use AND gates for product terms and an OR gate to combine them, forming AND-OR logic.
- After completing this section, you should be able to
  - Analyze and apply AND-OR circuits
  - Analyze and apply AND-OR-Invert circuits
  - Understand Laws and Rules of Boolean Algebra
  - Understand DeMorgan's Theorems

AND-OR Logic

- The below figure depicts a 2-input AND-OR circuit
- The output *X* follows an SOP expression. AND-OR circuits can have multiple AND gates with varying inputs.



(a) Logic diagram (ANSI standard distinctive shape symbols)

AND-OR Logic

- The Table shows a 4-input truth table, including intermediate AND outputs (AB, CD).
- See the previous figure as this table for its logic.
- For a 4-input AND-OR logic circuit, the output X is HIGH (1) if both input A and input B are HIGH (1) or both input C and input D are HIGH (1).

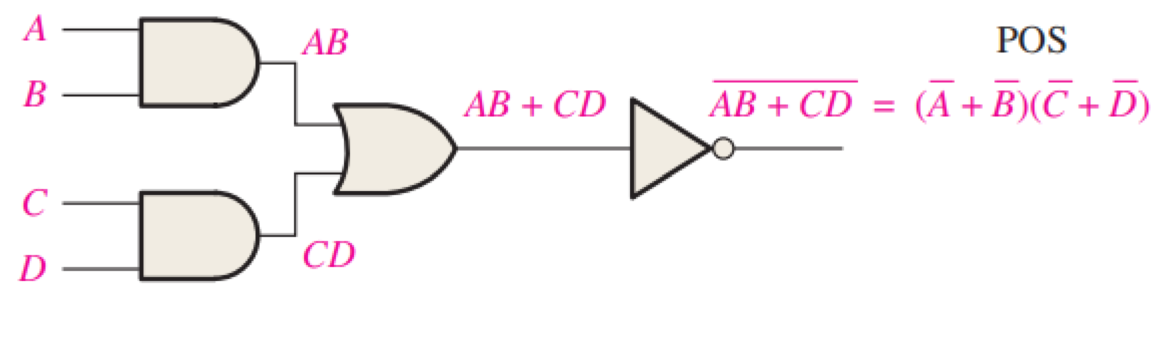
TABLE 5–1

Truth table for the AND-OR logic in Figure 5–1.

Inputs						Output
A	*B*	*C*	*D*	*AB*	*CD*	*X*
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

AND-OR-Invert Logic

- An AND-OR-Invert circuit inverts the output of an AND-OR configuration, enabling POS implementation.
- For example, $X = (\overline{A} + \overline{B})(\overline{C} + \overline{D}) = (\overline{AB})(\overline{CD}) = \overline{\overline{(\overline{AB})(\overline{CD})}} = \overline{\overline{\overline{AB}} + \overline{\overline{CD}}} = \overline{AB + CD}$
- Figure shows a 4-input AOI circuit.
- AND-OR-Invert circuits can have multiple AND gates with varying inputs.
- For a 4-input AND-OR-Invert logic circuit, the output X is LOW (0) if both input A and input B are HIGH (1) or both input C and input D are HIGH (1). A truth table can be developed from the AND-OR truth table in Table by simply changing all 1s to 0s and all 0s to 1s in the output column.



(a)

Laws and Rules of Boolean Algebra

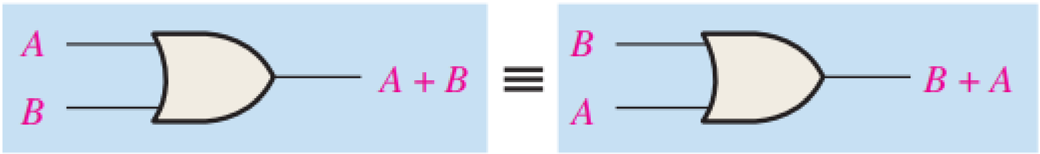
- The basic laws of Boolean algebra—**the commutative laws** for addition and multiplication.
- **The associative laws** for addition and multiplication, and
- **The distributive law**—are the same as in ordinary algebra.
- Each of the laws is illustrated with two or three variables, but the number of variables is not limited to this.

Commutative Laws

- The commutative law of addition for two variables is written as

$$A + B = B + A$$

This law states that the order in which the variables are ORed makes no difference. Remember, in Boolean algebra as applied to logic circuits, addition and the OR operation are the same.



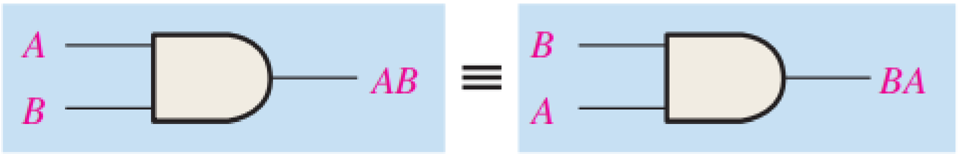
Application of commutative law of addition.

Commutative Laws

- **The commutative law of multiplication** for two variables is

$$AB = BA$$

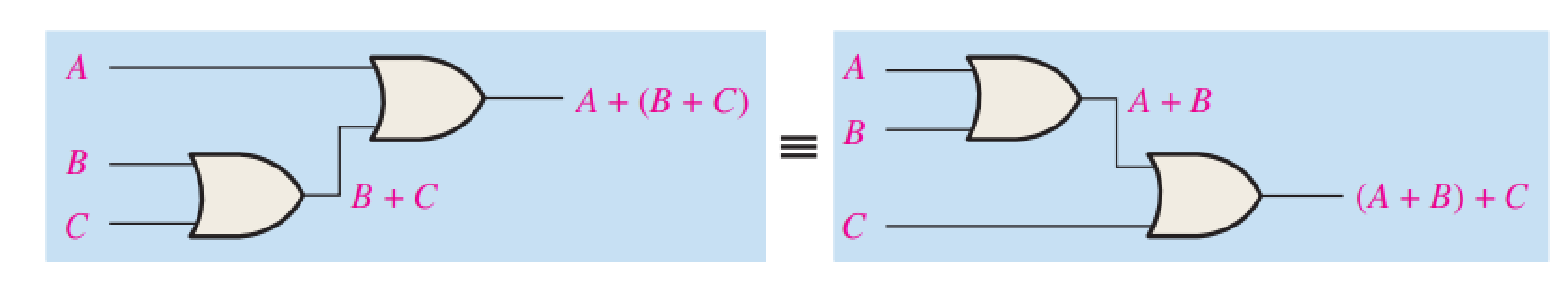
This law states that the order in which the variables are ANDed makes no difference.



Application of commutative law of multiplication.

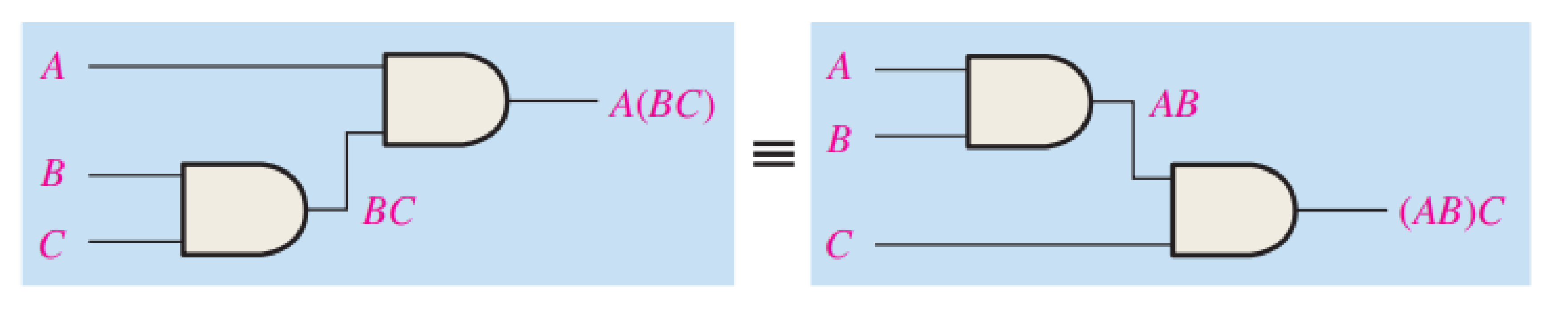
Associative Laws

- The associative law of addition is written as follows for three variables: $A + (B + C) = (A + B) + C$
- This law states that when ORing more than two variables, the result is the same regardless of the grouping of the variables.



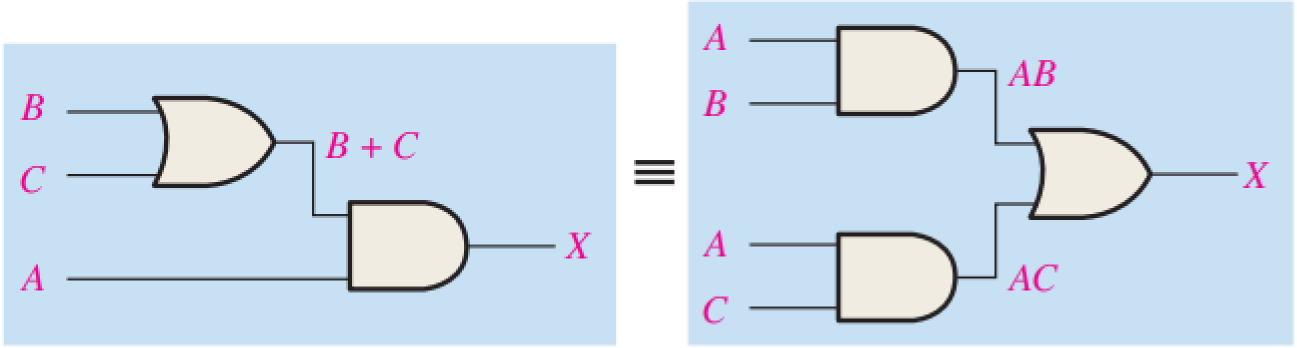
Associative Laws

- The associative law of multiplication is written as follows for three variables: $A(BC) = (AB)C$
- This law states that it makes no difference in what order the variables are grouped when ANDing more than two variables.



Distributive Law

- The distributive law is written for three variables as follows: $A(B + C) = AB + AC$
- This law states that ORing two or more variables and then ANDing the result with a single variable is equivalent to ANDing the single variable with each of the two or more variables and then ORing the products.
- The distributive law also expresses the process of factoring in which the common variable A is factored out of the product terms, for example, AB + AC = A(B + C).



$X = A(B + C)$ $\qquad$ $X = AB + AC$

Rules of Boolean Algebra

- The Table lists 12 basic rules that are useful in manipulating and simplifying Boolean expressions.
- Rules 1 through 9 will be viewed in terms of their application to logic gates. Rules 10 through 12 will be derived in terms of the simpler rules and the laws previously discussed.

Basic rules of Boolean algebra.

1. $A + 0 = A$	**7.** $A \cdot A = A$
2. $A + 1 = 1$	**8.** $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	**9.** $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	**10.** $A + AB = A$
5. $A + A = A$	**11.** $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	**12.** $(A + B)(A + C) = A + BC$

A, *B*, or *C* can represent a single variable or a combination of variables.

DeMorgan's Theorems

- DeMorgan's first theorem is stated as follows:
  - **The complement of a product of variables is equal to the sum of the complements of the variables.**
- Stated another way,
  - **The complement of two or more ANDed variables is equivalent to the OR of the complements of the individual variables.**
- The formula for expressing this theorem for two variables is

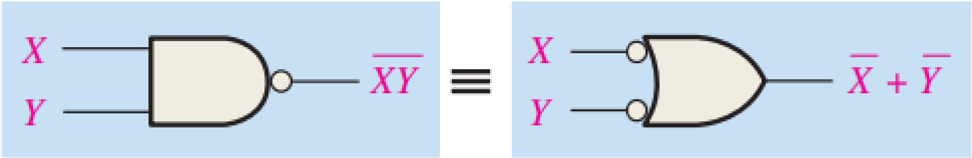
$$\overline{XY} = \overline{X} + \overline{Y}$$

DeMorgan's Theorems

- DeMorgan's second theorem is stated as follows:
  - **The complement of a sum of variables is equal to the product of the complements of the variables.**
- Stated another way,
  - **The complement of two or more ORed variables is equivalent to the AND of the complements of the individual variables.**
- The formula for expressing this theorem for two variables is

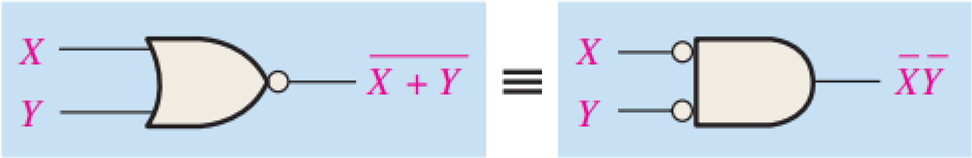
$$\overline{X + Y} = \overline{X}\,\overline{Y}$$

Gate equivalencies and the corresponding truth tables that illustrate DeMorgan’s theorems. Notice the equality of the two output columns in each table. This shows that the equivalent gates perform the same logic function.



NAND Negative-OR

Inputs		Output	
X	Y	$\overline{XY}$	$\overline{X} + \overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0



NOR Negative-AND

Inputs		Output	
X	Y	$\overline{X + Y}$	$\overline{X}\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

EXAMPLES

EXAMPLES 1

Apply DeMorgan's theorems to the expressions $\overline{XYZ}$ and $\overline{X + Y + Z}$.

Solution

$$\overline{XYZ} = \overline{X} + \overline{Y} + \overline{Z}$$
$$\overline{X + Y + Z} = \overline{X}\,\overline{Y}\,\overline{Z}$$

EXAMPLES 2

Apply DeMorgan's theorems to the expressions $\overline{WXYZ}$ and $\overline{W + X + Y + Z}$.

Solution

$$\overline{WXYZ} = \overline{W} + \overline{X} + \overline{Y} + \overline{Z}$$
$$\overline{W + X + Y + Z} = \overline{W}\,\overline{X}\,\overline{Y}\,\overline{Z}$$

Examples and Solve (C)!

Apply DeMorgan's theorems to each of the following expressions:

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

(b) $\overline{ABC + DEF}$

(c) $\overline{A\overline{B} + \overline{C}D + EF}$

Solution

(a) Let $A + B + C = X$ and $D = Y$. The expression $\overline{(A + B + C)D}$ is of the form $\overline{XY} = \overline{X} + \overline{Y}$ and can be rewritten as

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

Next, apply DeMorgan's theorem to the term $\overline{A + B + C}$.

$$\overline{A + B + C} + \overline{D} = \overline{A}\,\overline{B}\,\overline{C} + \overline{D}$$

(b) Let $ABC = X$ and $DEF = Y$. The expression $\overline{ABC + DEF}$ is of the form $\overline{X + Y} = \overline{X}\,\overline{Y}$ and can be rewritten as

$$\overline{ABC + DEF} = (\overline{ABC})(\overline{DEF})$$

Next, apply DeMorgan's theorem to each of the terms $\overline{ABC}$ and $\overline{DEF}$.

$$(\overline{ABC})(\overline{DEF}) = (\overline{A} + \overline{B} + \overline{C})(\overline{D} + \overline{E} + \overline{F})$$

Logic Simplification Using Boolean Algebra

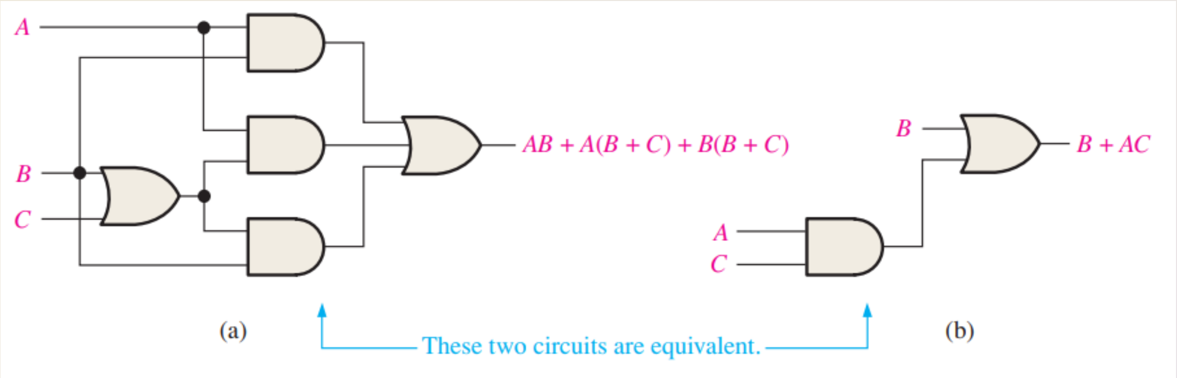
- A logic expression can be reduced to its simplest form or changed to a more convenient form to implement the expression most efficiently using Boolean algebra.
- The approach uses the basic laws, rules, and theorems of Boolean algebra to manipulate and simplify an expression.
- A simplified Boolean expression uses the fewest gates possible to implement a given expression

EXAMPLE

- **Question:** Using Boolean algebra techniques, simplify this expression:
  AB + A(B + C) + B(B + C)
- **Solution:**
- The following is not necessarily the only approach.
  - Step 1: Apply the distributive law to the second and third terms in the expression, as follows:
    - **AB + AB + AC + BB + BC**
  - Step 2: Apply rule 7 (BB = B) to the fourth term.
    - **AB + AB + AC + B + BC**
  - Step 3: Apply rule 5 (AB + AB = AB) to the first two terms.
    - **AB + AC + B + BC**
  - Step 4: Apply rule 10 (B + BC = B) to the last two terms.
    - **AB + AC + B**
  - Step 5: Apply rule 10 (AB + B = B) to the first and third terms.
    - **B + AC**
- At this point the expression is simplified as much as possible.

Cont. - Example Solution!

- The Figure shows that the simplification process in the previous Example has significantly reduced the number of logic gates required to implement the expression.
  - Part (a) shows that five gates are required to implement the expression in its original form; however, only two gates are needed for the simplified expression, shown in part (b).
  - It is important to realize that these two gate circuits are equivalent. That is, for any combination of levels on the A, B, and C inputs, you get the same output from either circuit.



Example!

Simplify the following Boolean expression:

$$\overline{\overline{AB} + AC} + \overline{A}\,\overline{B}C$$

Solution

Step 1: Apply DeMorgan's theorem to the first term.

$$(\overline{\overline{AB}})(\overline{AC}) + \overline{A}\,\overline{B}C$$

Step 2: Apply DeMorgan's theorem to each term in parentheses.

$$(\overline{A} + \overline{B})(\overline{A} + \overline{C}) + \overline{A}\,\overline{B}C$$

Step 3: Apply the distributive law to the two terms in parentheses.

$$\overline{A}\,\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C} + \overline{A}\,\overline{B}C$$

Step 4: Apply rule 7 ($\overline{A}\,\overline{A} = \overline{A}$) to the first term, and apply rule 10 [$\overline{A}\,\overline{B} + \overline{A}\,\overline{B}C = \overline{A}\,\overline{B}(1 + C) = \overline{A}\,\overline{B}$] to the third and last terms.

$$\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$$

Step 5: Apply rule 10 [$\overline{A} + \overline{A}\,\overline{C} = \overline{A}(1 + \overline{C}) = \overline{A}$] to the first and second terms.

$$\overline{A} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$$

Step 6: Apply rule 10 [$\overline{A} + \overline{A}\,\overline{B} = \overline{A}(1 + \overline{B}) = \overline{A}$] to the first and second terms.

$$\overline{A} + \overline{B}\,\overline{C}$$

Example!

Reduce the combinational logic circuit in Figure 5–14 to a minimum form.

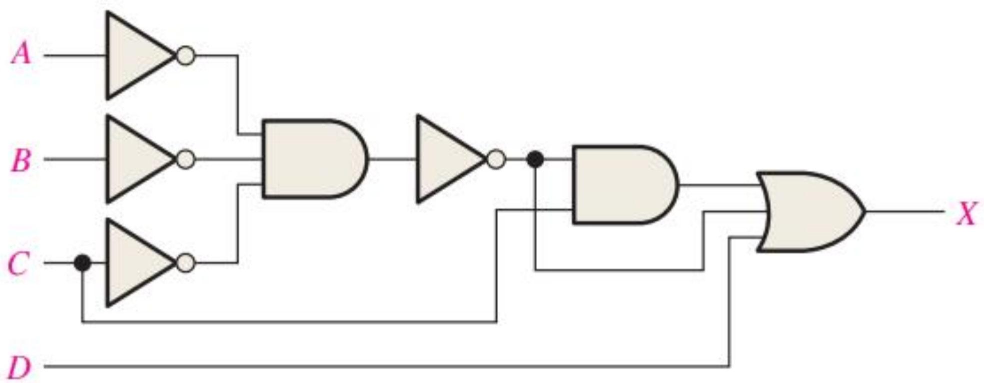


FIGURE 5–14
Open file F05-14 to verify that this circuit is equivalent to the gate in Figure 5–15.

MultiSim

Solution

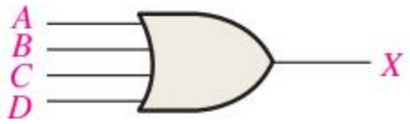
The expression for the output of the circuit is

$$X = (\overline{\overline{A}\,\overline{B}\,\overline{C}})C + \overline{\overline{A}\,\overline{B}\,\overline{C}} + D$$

Applying DeMorgan's theorem and Boolean algebra,

$$\begin{aligned} X &= (\overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}})C + \overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}} + D \\ &= AC + BC + CC + A + B + C + D \\ &= AC + BC + C + A + B + \cancel{C} + D \\ &= C(A + B + 1) + A + B + D \\ X &= A + B + C + D \end{aligned}$$

The simplified circuit is a 4-input OR gate as shown in Figure 5–15.



THANK YOU ☺