# Digital Fundamentals

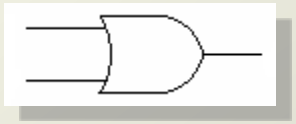
## CHAPTER 4

## Boolean Algebra and Logic Simplification

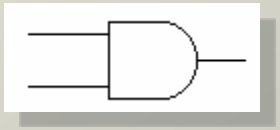
# Boolean Operations and Expressions

# Boolean Operations and Expressions

- Addition
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 0 = 1
  - 1 + 1 = 1



- Multiplication
  - 0 * 0 = 0
  - 0 * 1 = 0
  - 1 * 0 = 0
  - 1 * 1 = 1



# Laws and Rules of Boolean Algebra

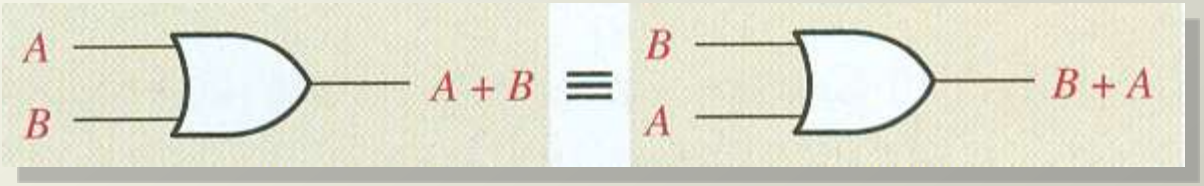
# Laws Boolean Algebra

- Commutative Laws
- Associative Laws
- Distributive Law

# Laws of Boolean Algebra

- Commutative Law of Addition:

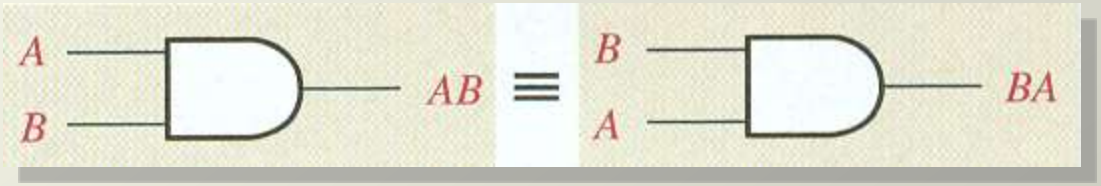
**A + B = B + A**



## Laws of Boolean Algebra

- Commutative Law of Multiplication:

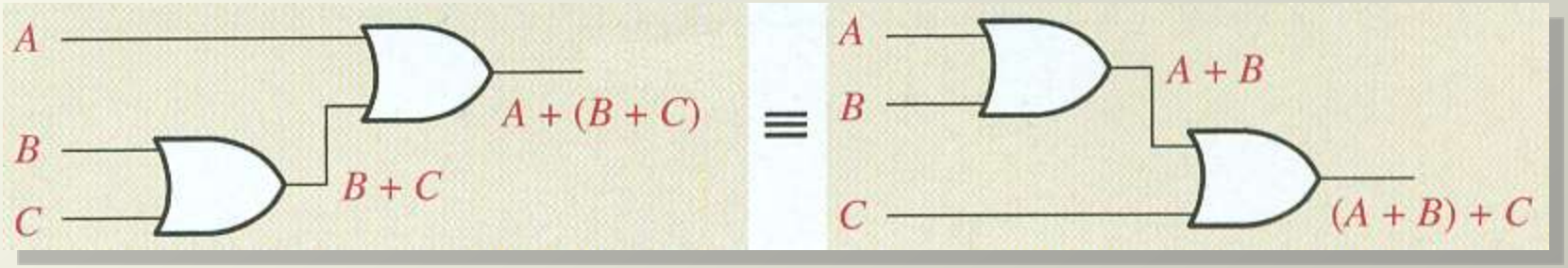
**A * B = B * A**



## Laws of Boolean Algebra

- Associative Law of Addition:

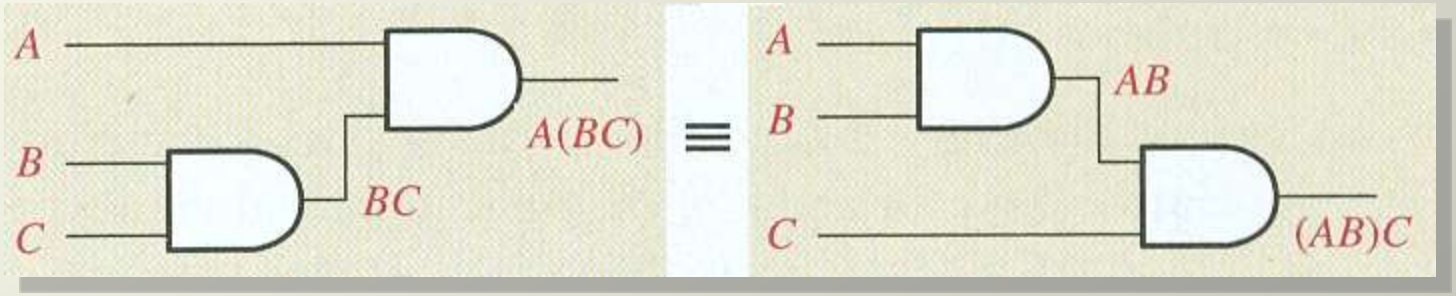
$$A + (B + C) = (A + B) + C$$



## Laws of Boolean Algebra

- Associative Law of Multiplication:

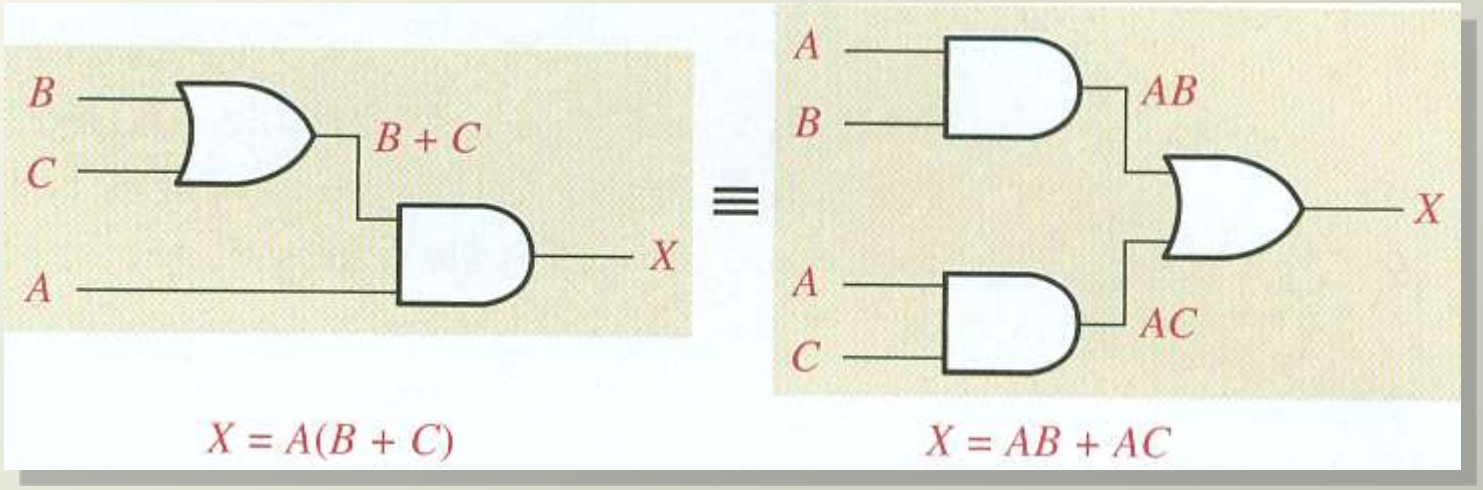
**A * (B * C) = (A * B) * C**



## Laws of Boolean Algebra

- Distributive Law:

$$A(B + C) = AB + AC$$

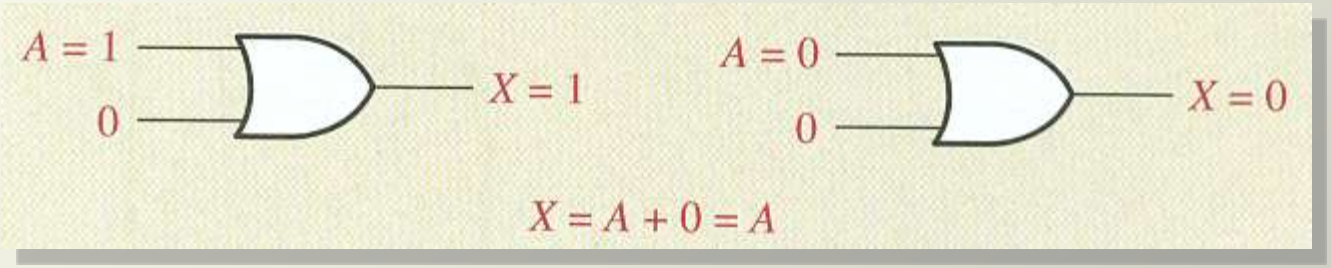


# Rules of Boolean Algebra

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

# Rules of Boolean Algebra

- Rule 1

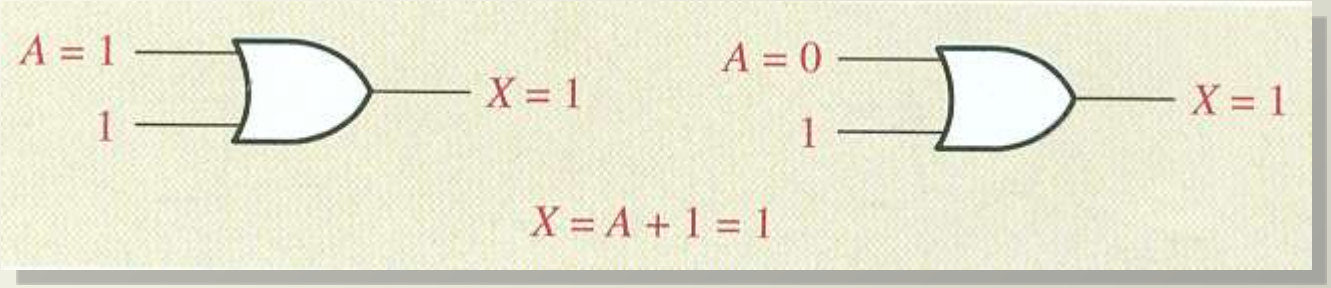



| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**OR Truth Table**

# Rules of Boolean Algebra

- Rule 2

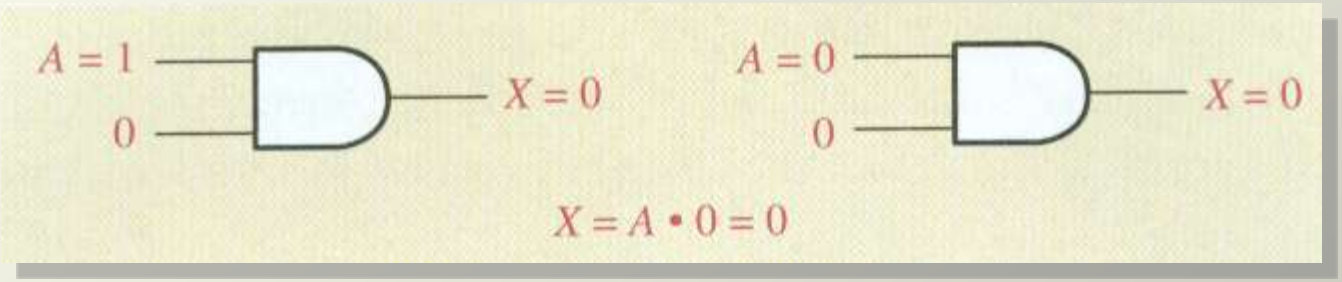



| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**OR Truth Table**

# Rules of Boolean Algebra

- Rule 3

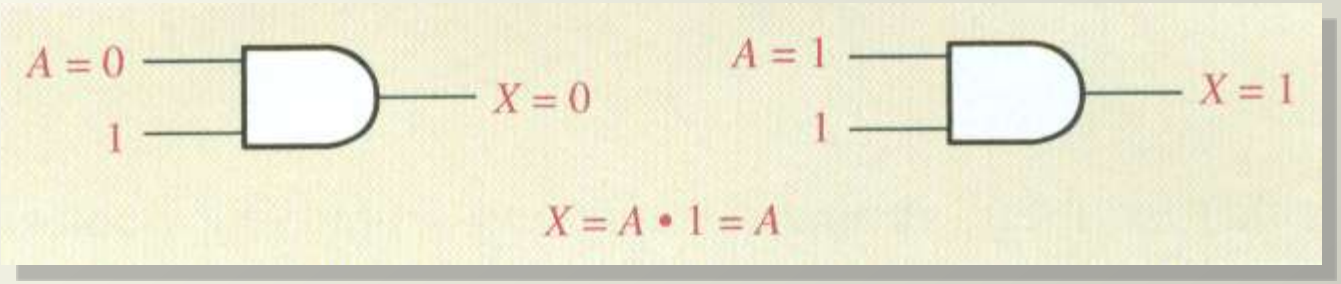



| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**AND Truth Table**

# Rules of Boolean Algebra

- Rule 4

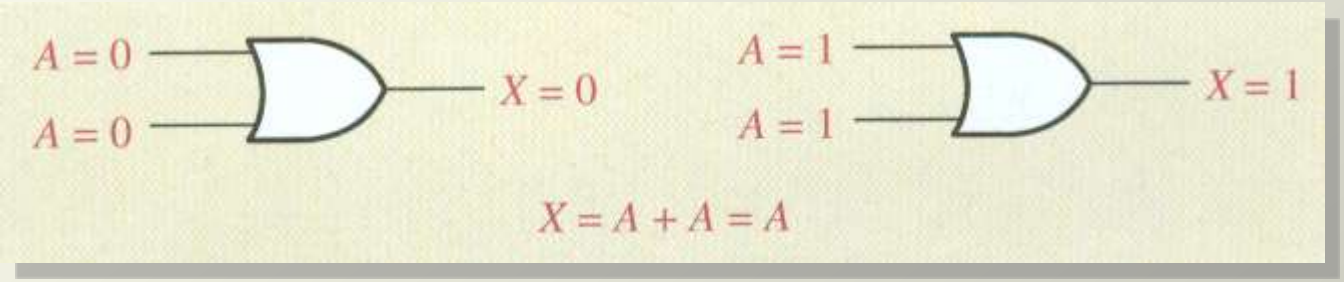



| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**AND Truth Table**

# Rules of Boolean Algebra

- Rule 5

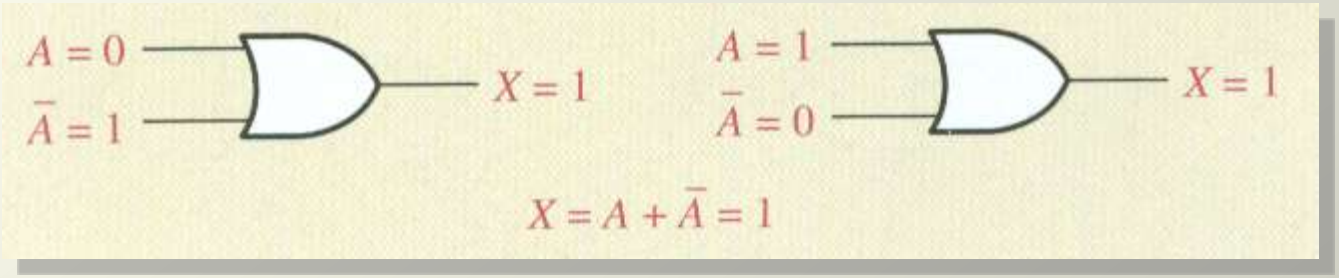



| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**OR Truth Table**

# Rules of Boolean Algebra

- Rule 6

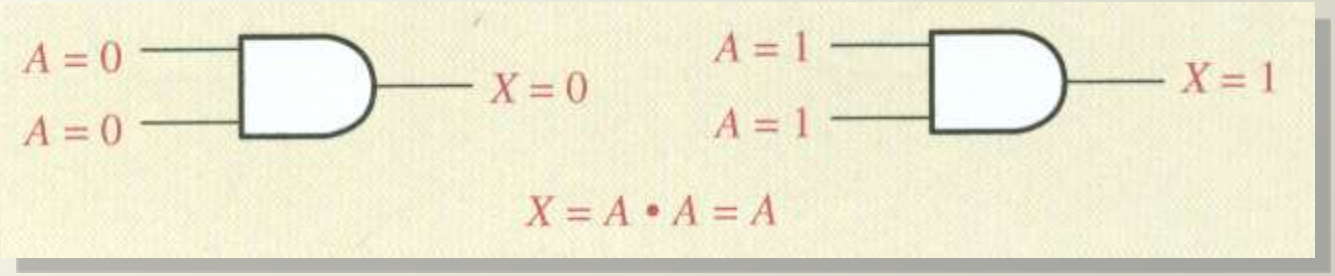



| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**OR Truth Table**

# Rules of Boolean Algebra

- Rule 7

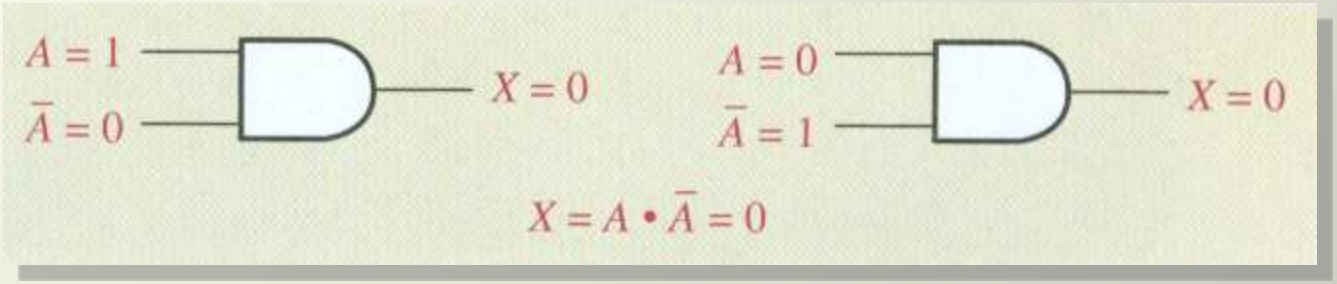



| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**AND Truth Table**

# Rules of Boolean Algebra

- Rule 8

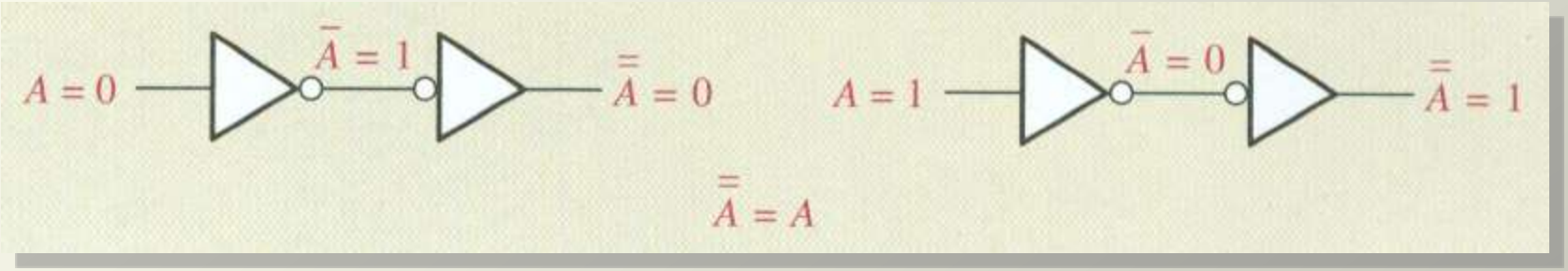



| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**AND Truth Table**

## Rules of Boolean Algebra

- Rule 9

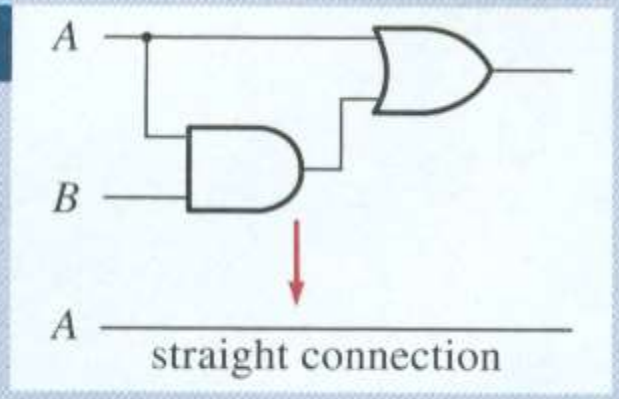


# Rules of Boolean Algebra

- Rule 10: A + AB = A

| A | B | AB | A + AB |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 |

equal




| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**AND Truth Table**

| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

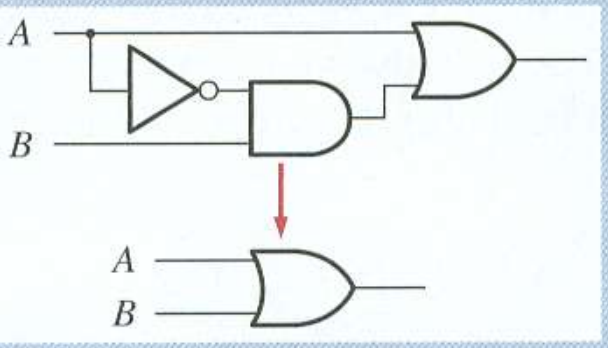
**OR Truth Table**

# Rules of Boolean Algebra

- Rule 11: $A + \overline{A}B = A + B$

| 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




| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**AND Truth Table**

| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

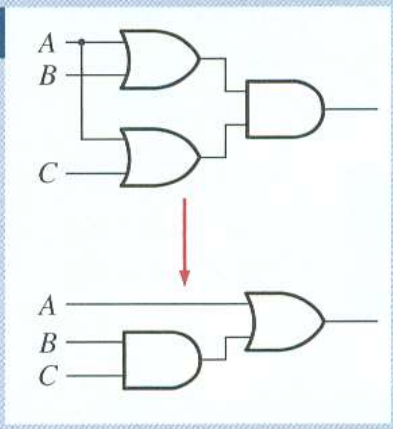
**OR Truth Table**

# Rules of Boolean Algebra

- Rule 12: (A + B)(A + C) = A + BC

| A | B | C | A + B | A + C | (A + B)(A + C) | BC | A + BC |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

equal




| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**AND Truth Table**

| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**OR Truth Table**

# DeMorgan’s Theorem

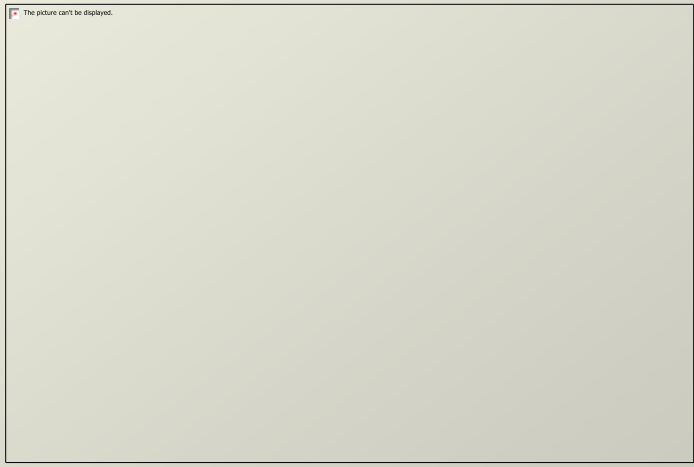
# DeMorgan's Theorems

- Theorem 1

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

- Theorem 2

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




**Remember:**

**"Break the bar, change the sign"**

Figure 4–16

**A logic circuit showing the development of the Boolean expression for the output.**

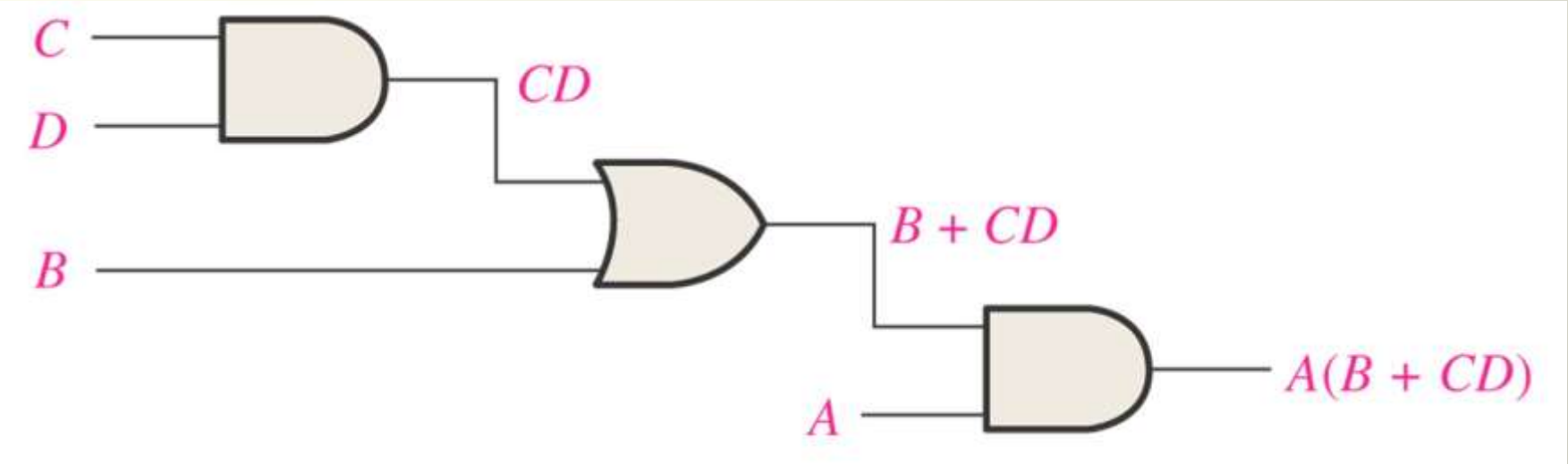
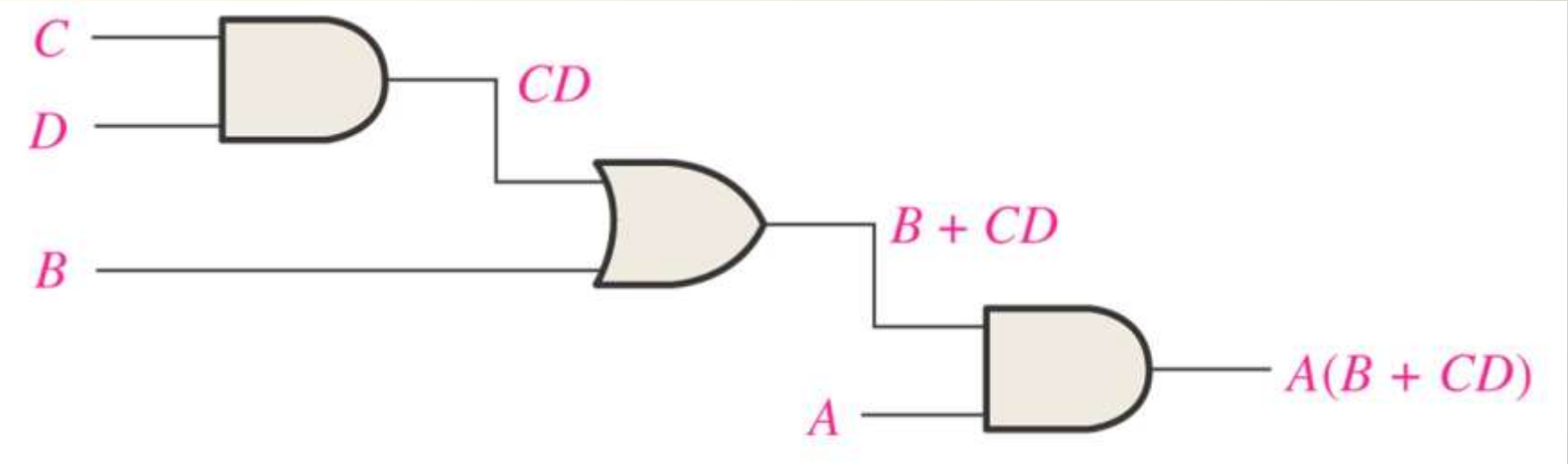


Figure 4–16

**A logic circuit showing the development of the Boolean expression for the output.**



# Standard Forms of Boolean Expressions

## Standard Forms of Boolean Expressions

- The sum-of-product (SOP) form

  Example: X = AB + CD + EF

- The product of sum (POS) form

  Example: X = (A + B)(C + D)(E + F)

Figure 4–18 **Implementation of the SOP expression *AB* + *BCD* + *AC*.**

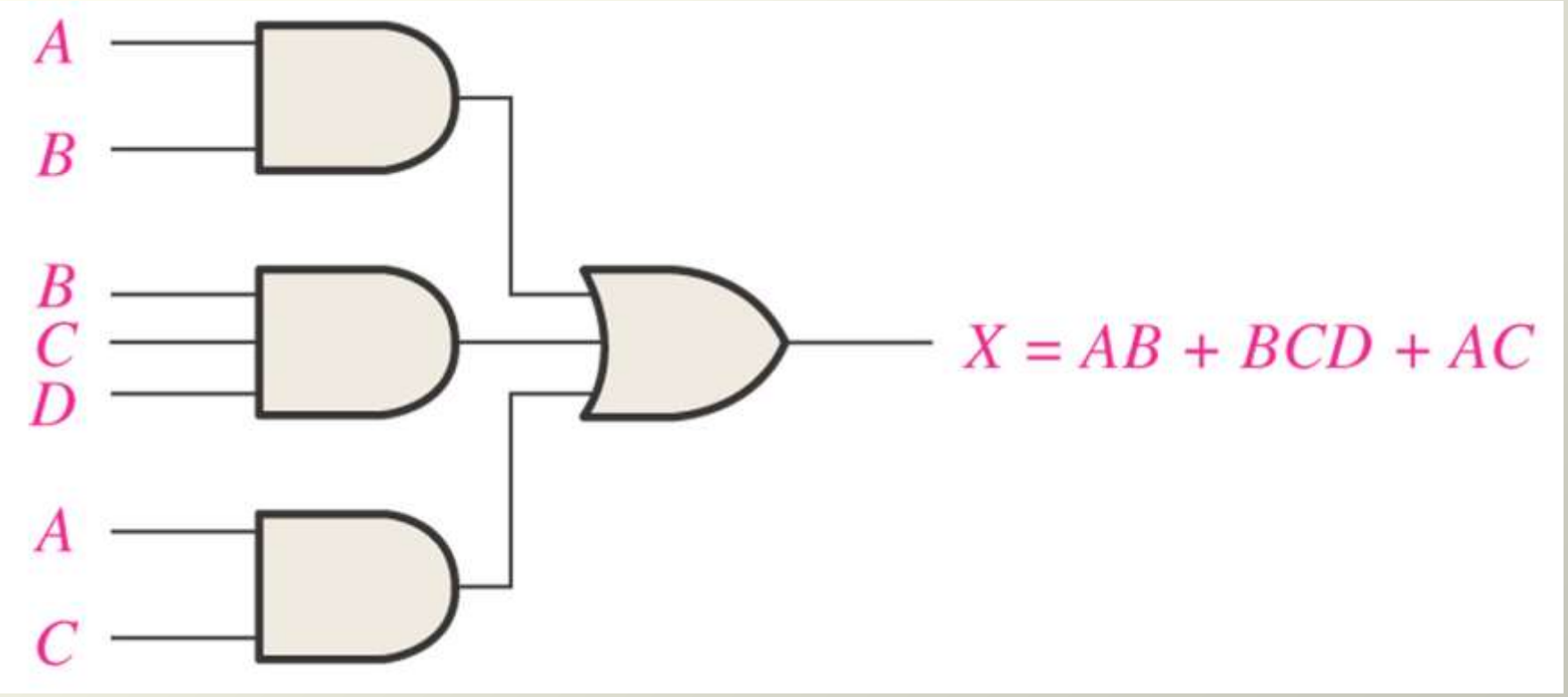


Figure 4–19

**This NAND/NAND implementation is equivalent to the AND/OR in Figure 4–18.**

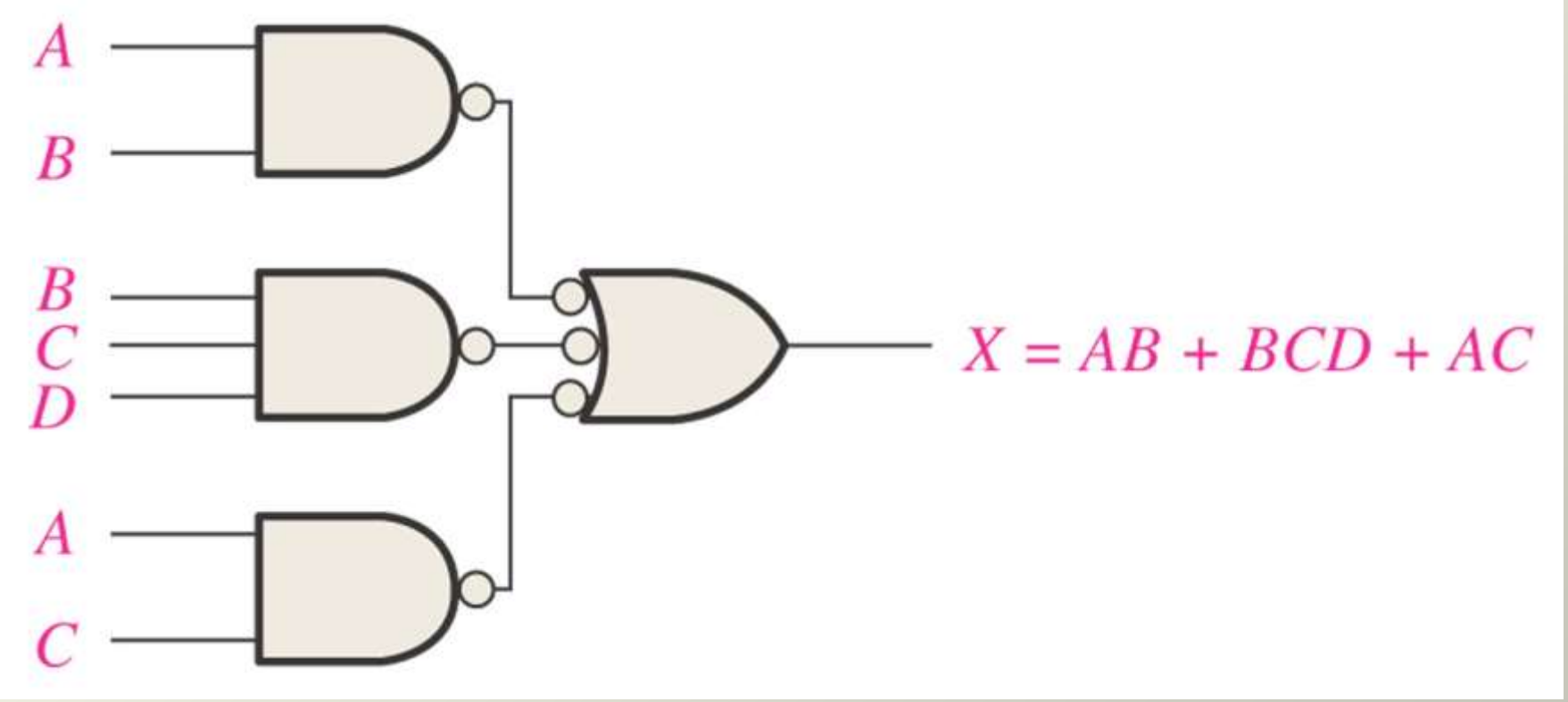
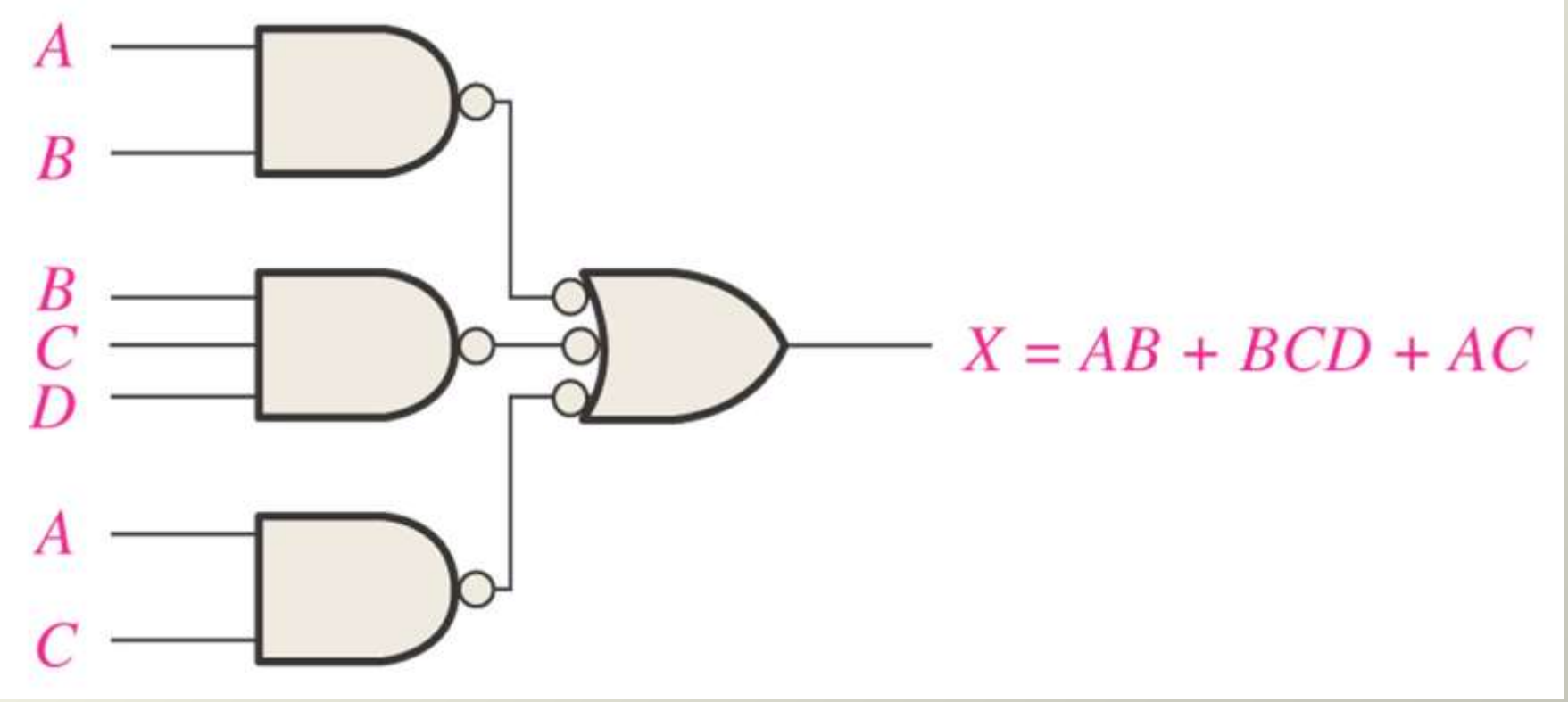


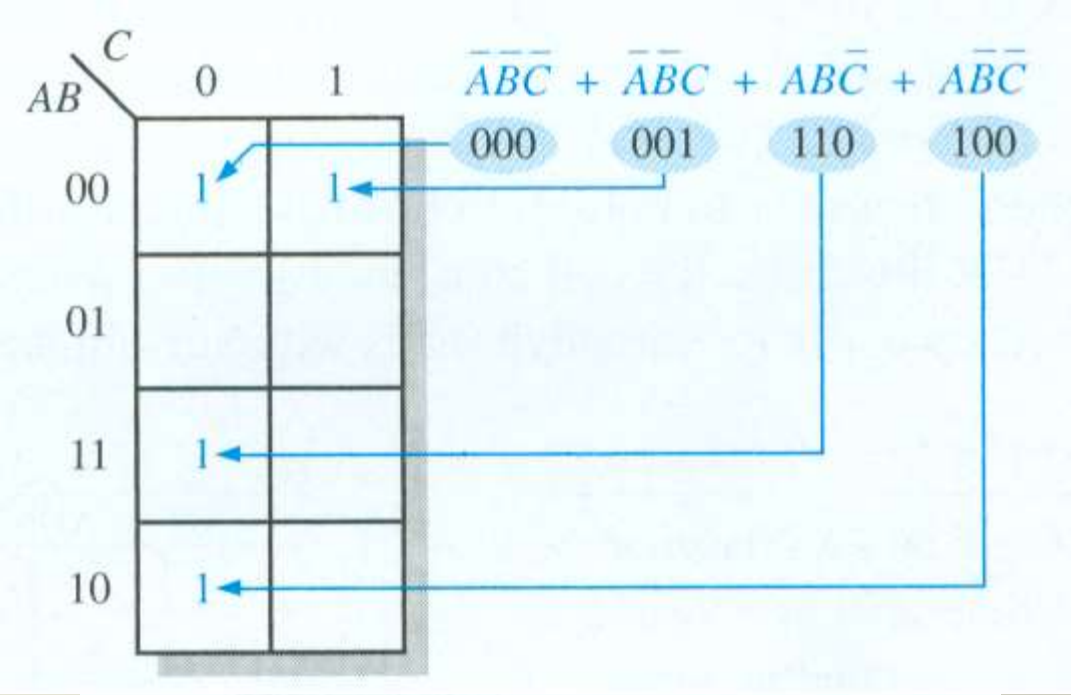
Figure 4–19

**This NAND/NAND implementation is equivalent to the AND/OR in Figure 4–18.**



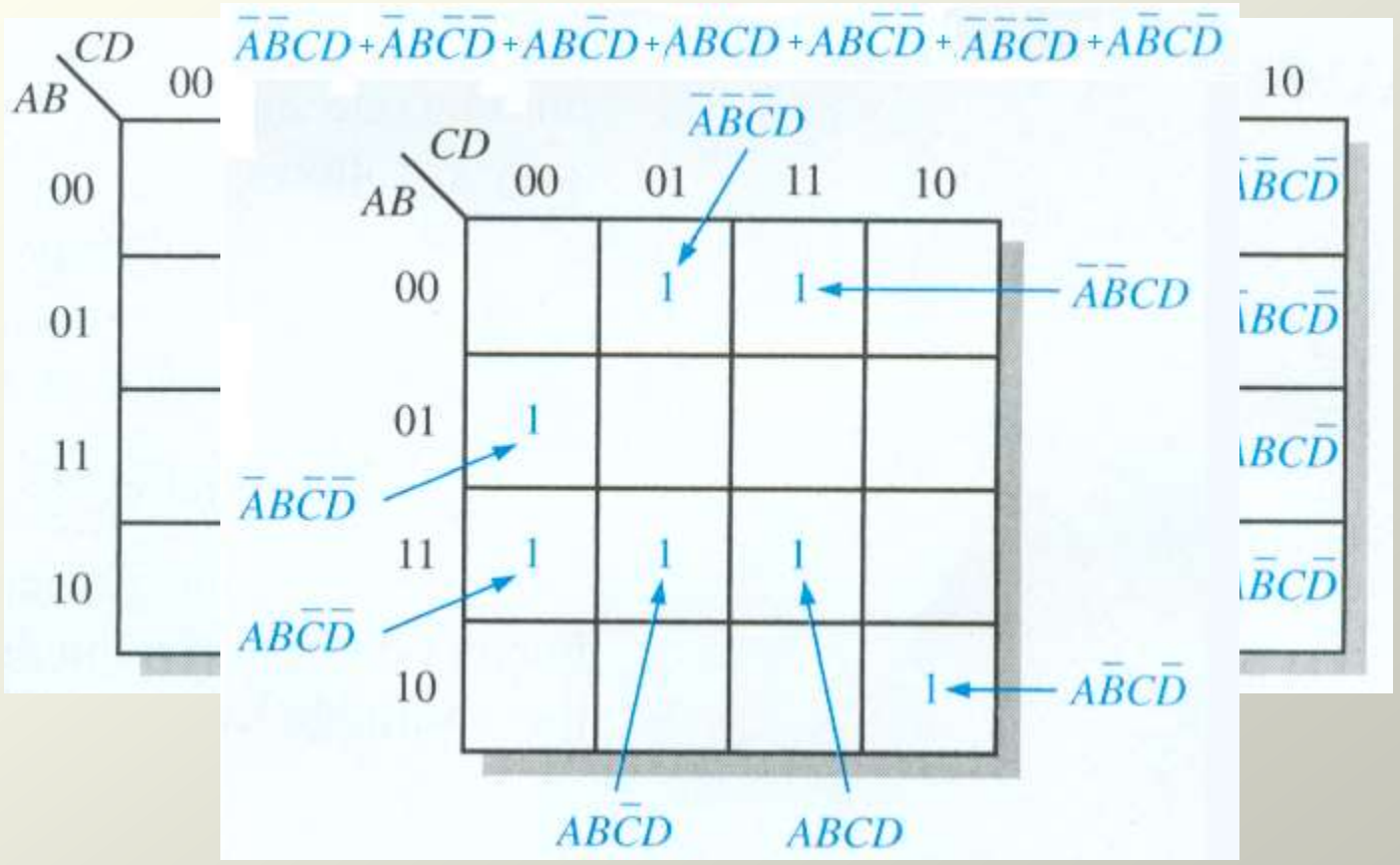
# The Karnaugh Map

# The Karnaugh Map




**3-Variable Example**
**3-Variable Karnaugh Map**

# The Karnaugh Map




**4-Variable Example**

Figure 4–23 **Adjacent cells on a Karnaugh map are those that differ by only one variable. Arrows point between adjacent cells.**

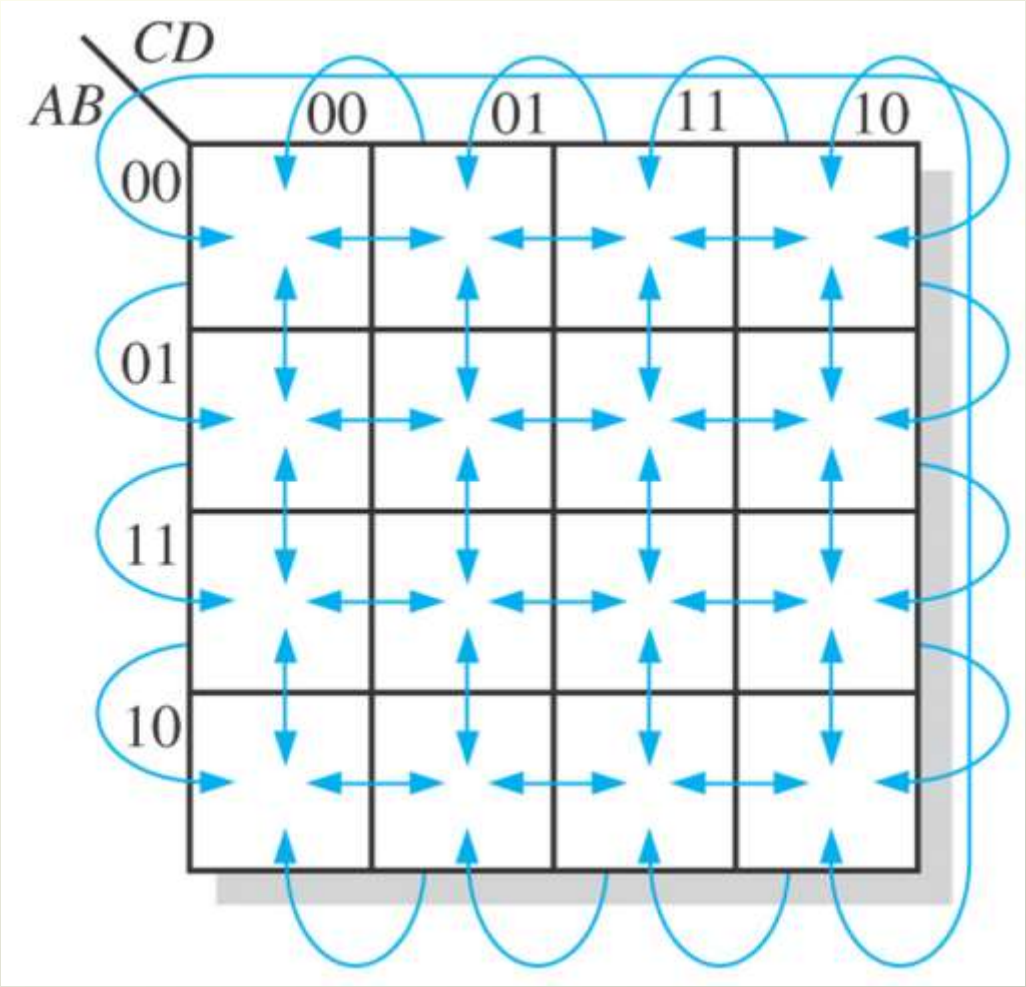


Figure 4–24 **Example of mapping a standard SOP expression.**

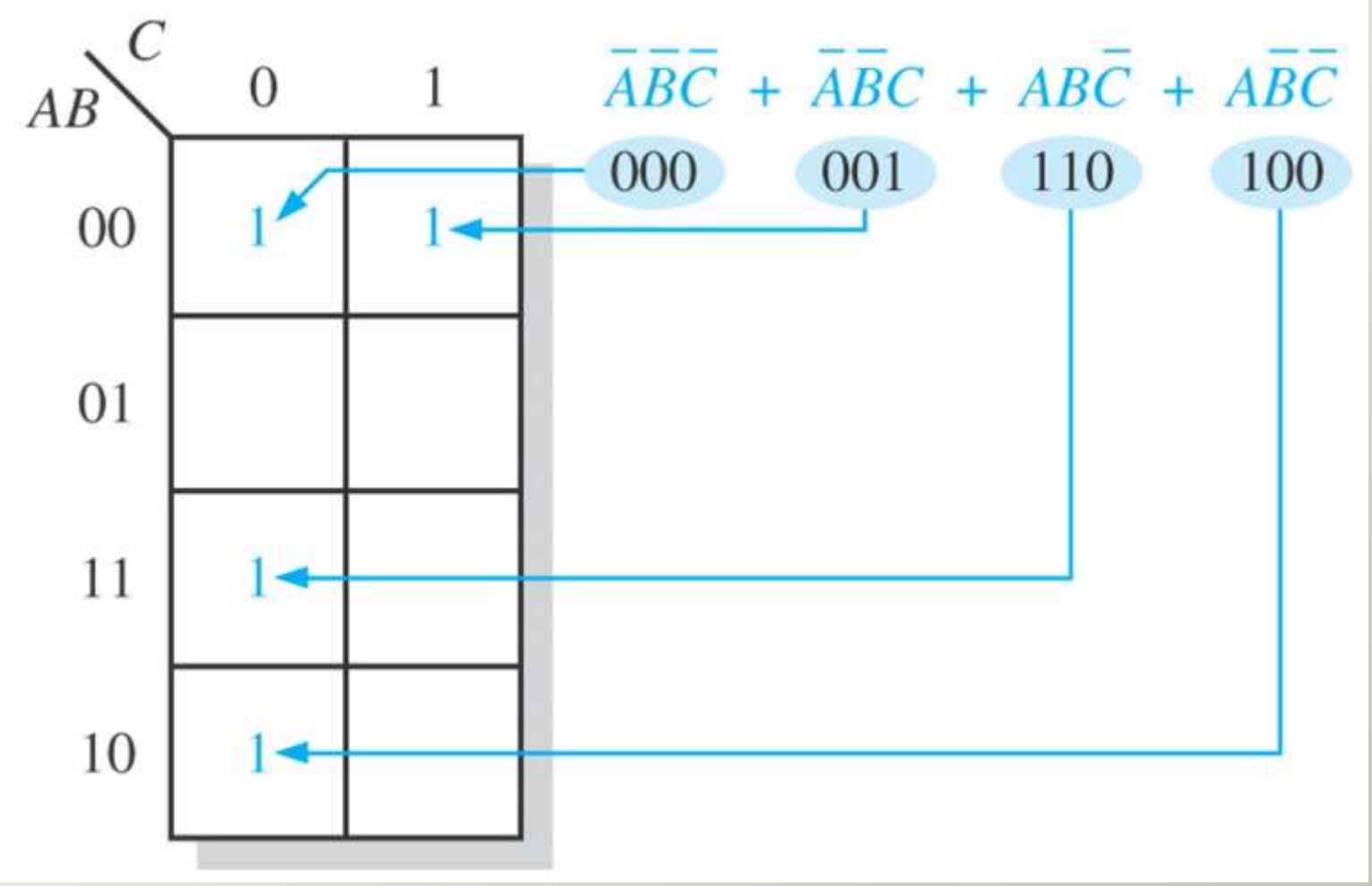


Figure 4–25

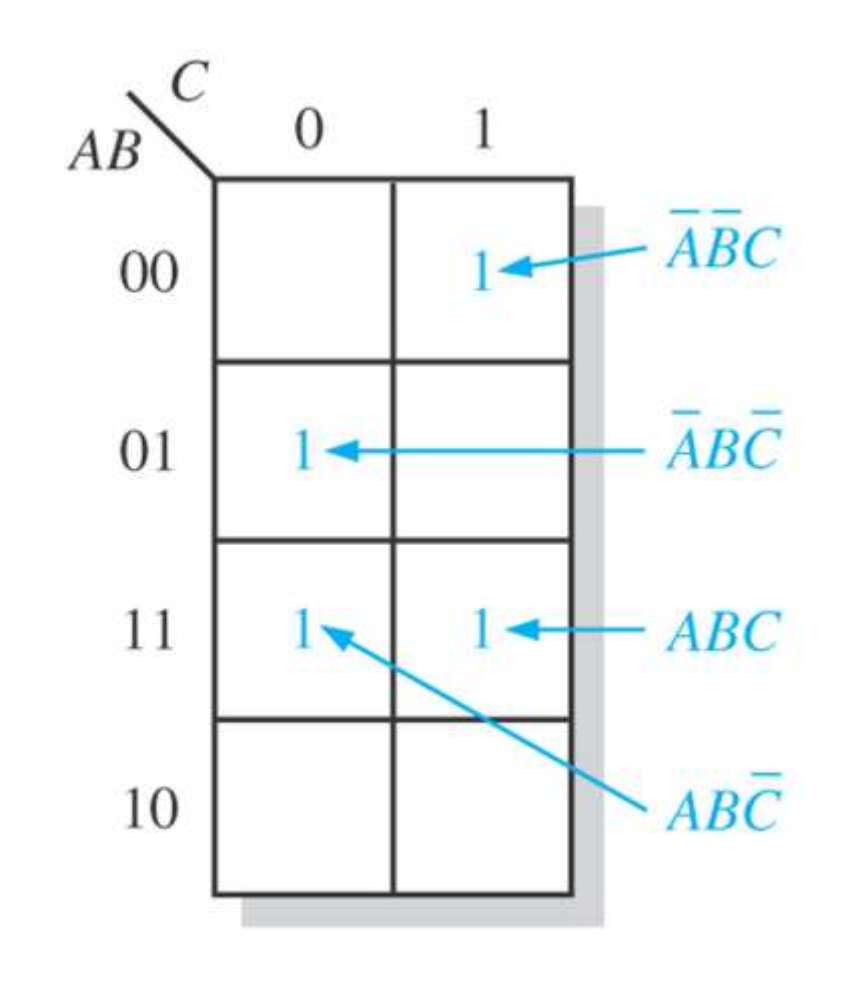


Figure 4–28

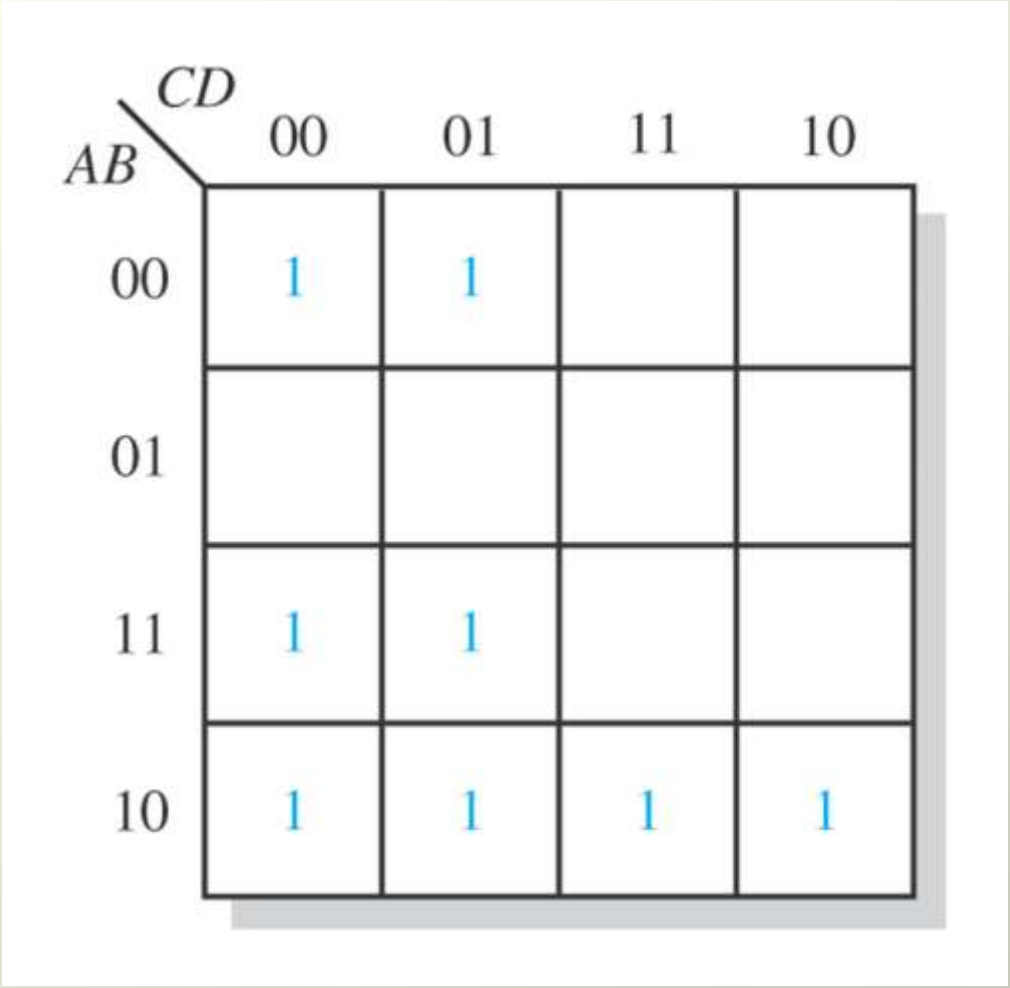


Figure 4–29

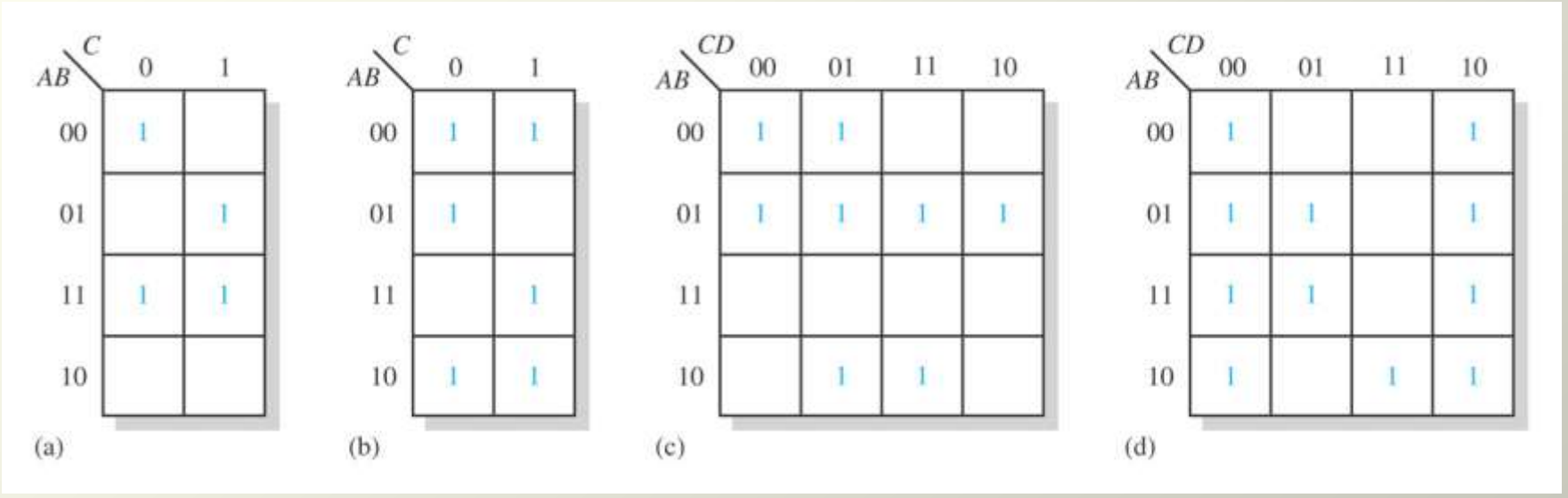

C
AB
0
1
00
01
11
10
1
(a)
C
AB
0
1
00
01
11
10
1
(b)
CD
AB
00
01
11
10
00
01
11
10
1
(c)
CD
AB
00
01
11
10
00
01
11
10
1
(d)

Figure 4–30

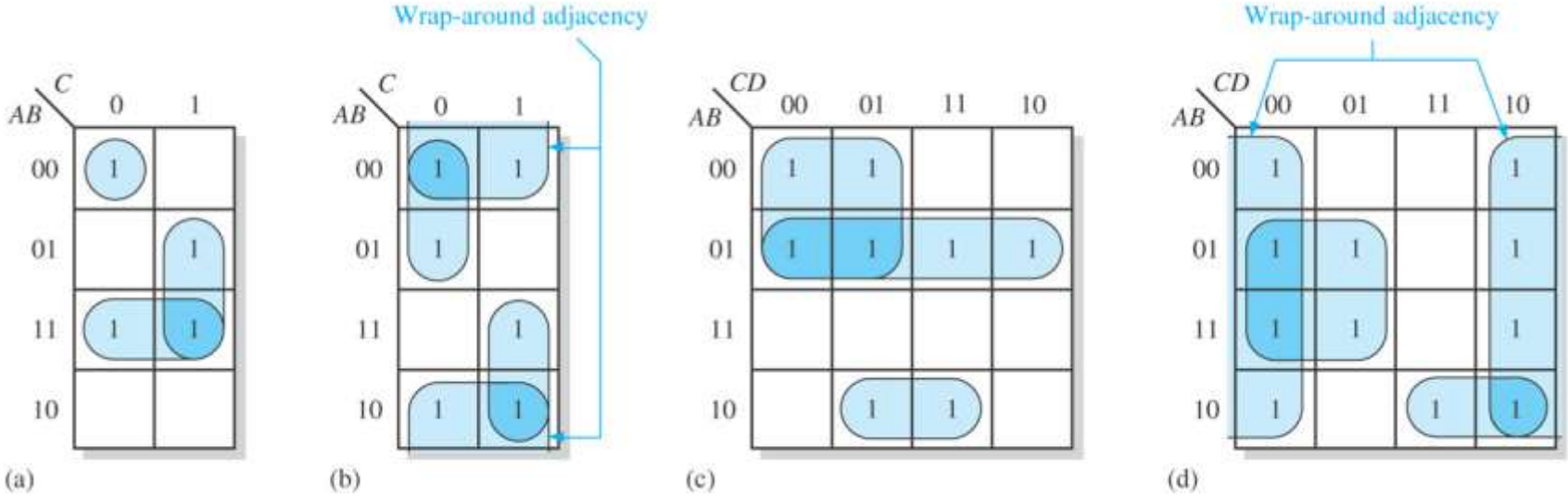

Wrap-around adjacency
Wrap-around adjacency
C
AB
0
1
00
01
11
10
CD
(a)
(b)
(c)
(d)

Figure 4–31

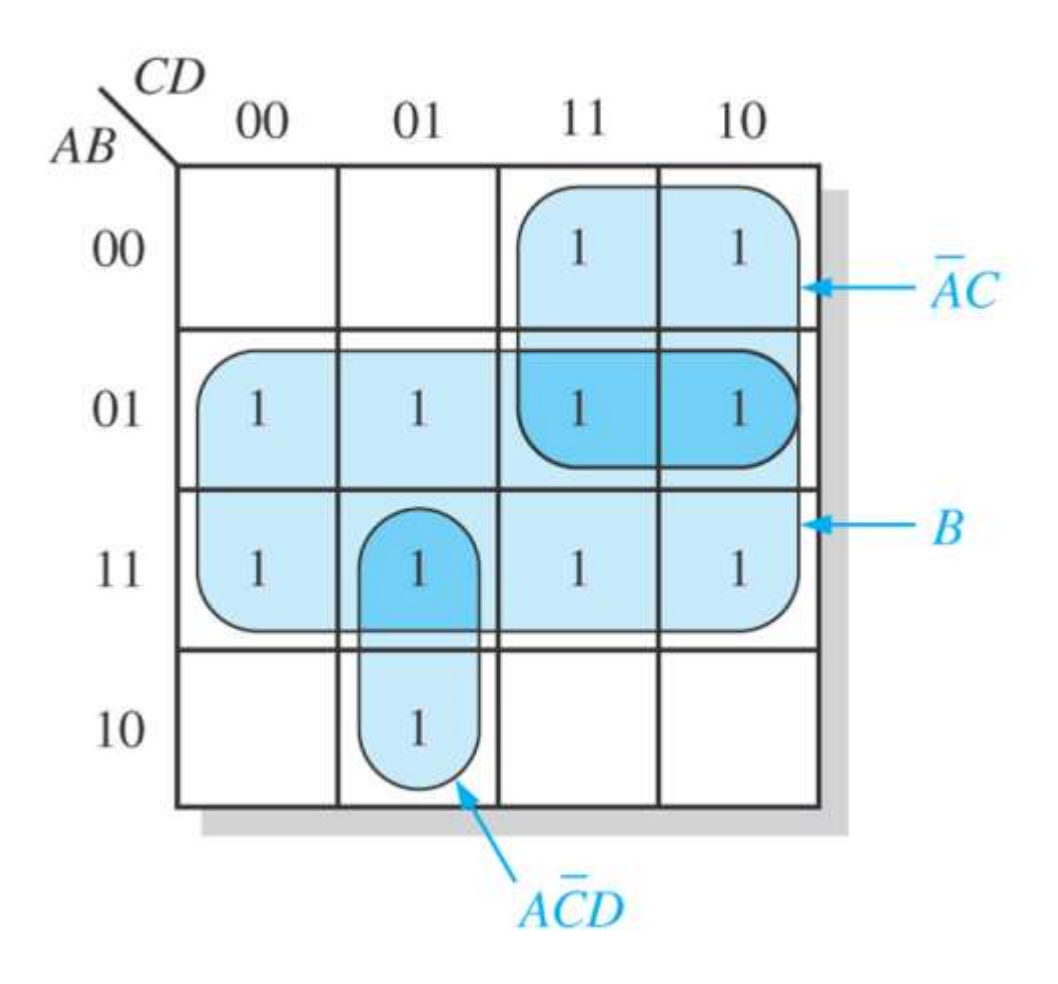


Figure 4–32

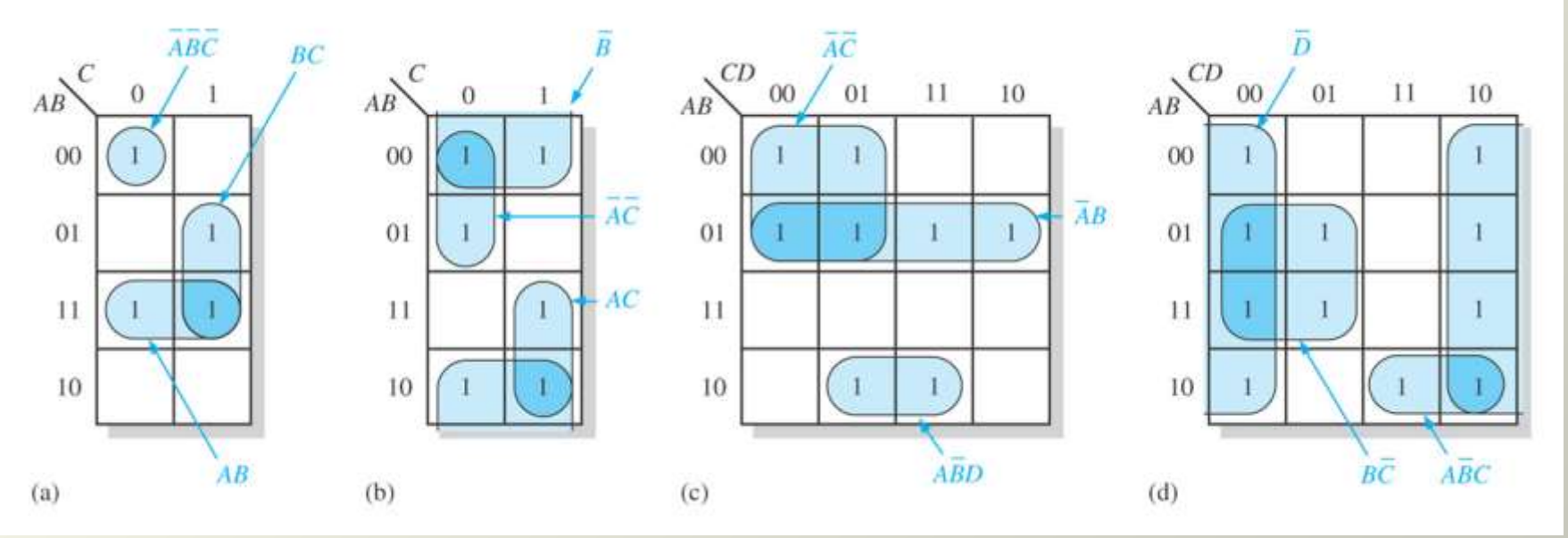

$\overline{A}\overline{B}\overline{C}$
$BC$
C
AB
0
1
00
01
11
10
1
$AB$
(a)
$\overline{B}$
$\overline{A}\overline{C}$
$AC$
(b)
$\overline{A}\overline{C}$
CD
00
01
11
10
$\overline{A}B$
$A\overline{B}D$
(c)
$\overline{D}$
$B\overline{C}$
$A\overline{B}C$
(d)

Figure 4–33

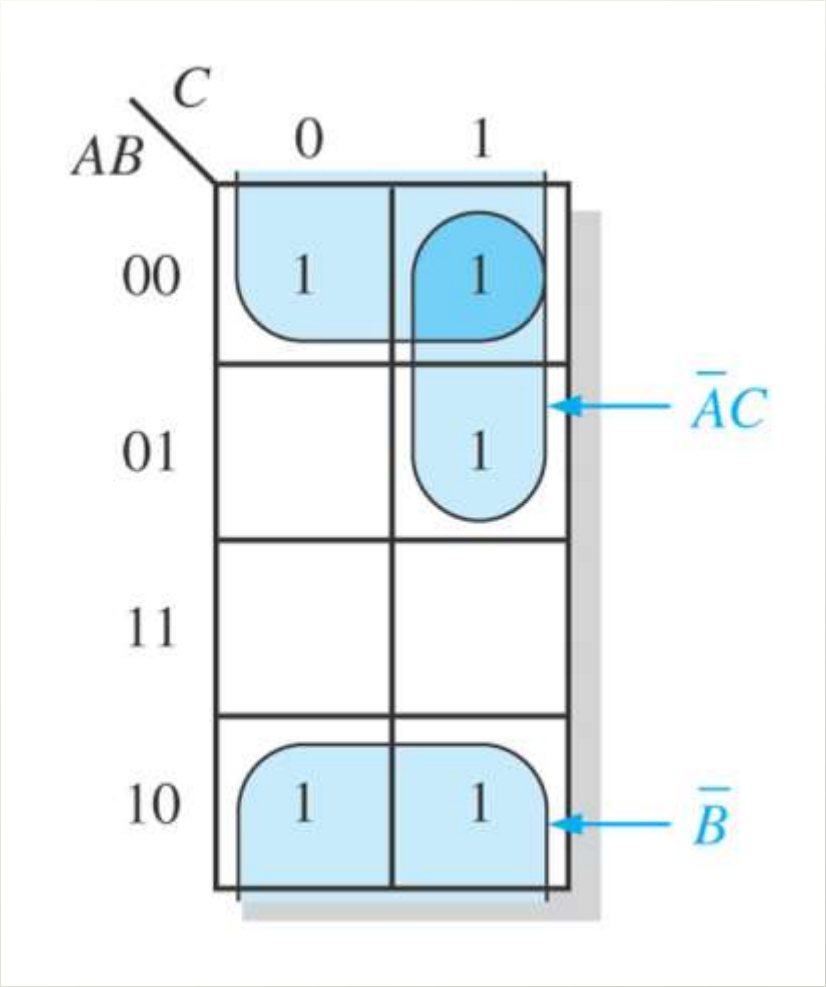


Figure 4–34

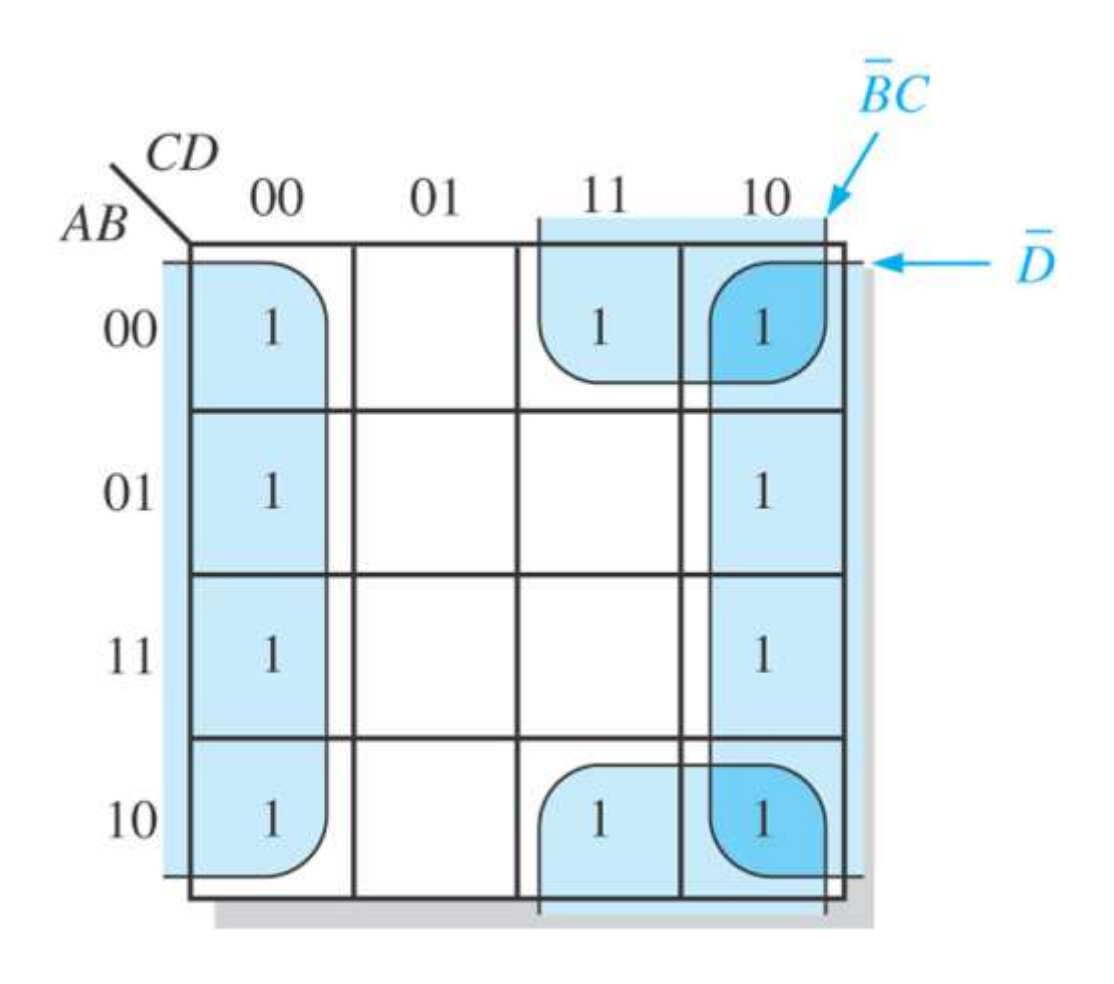


Figure 4–35 **Example of mapping directly from a truth table to a Karnaugh map.**

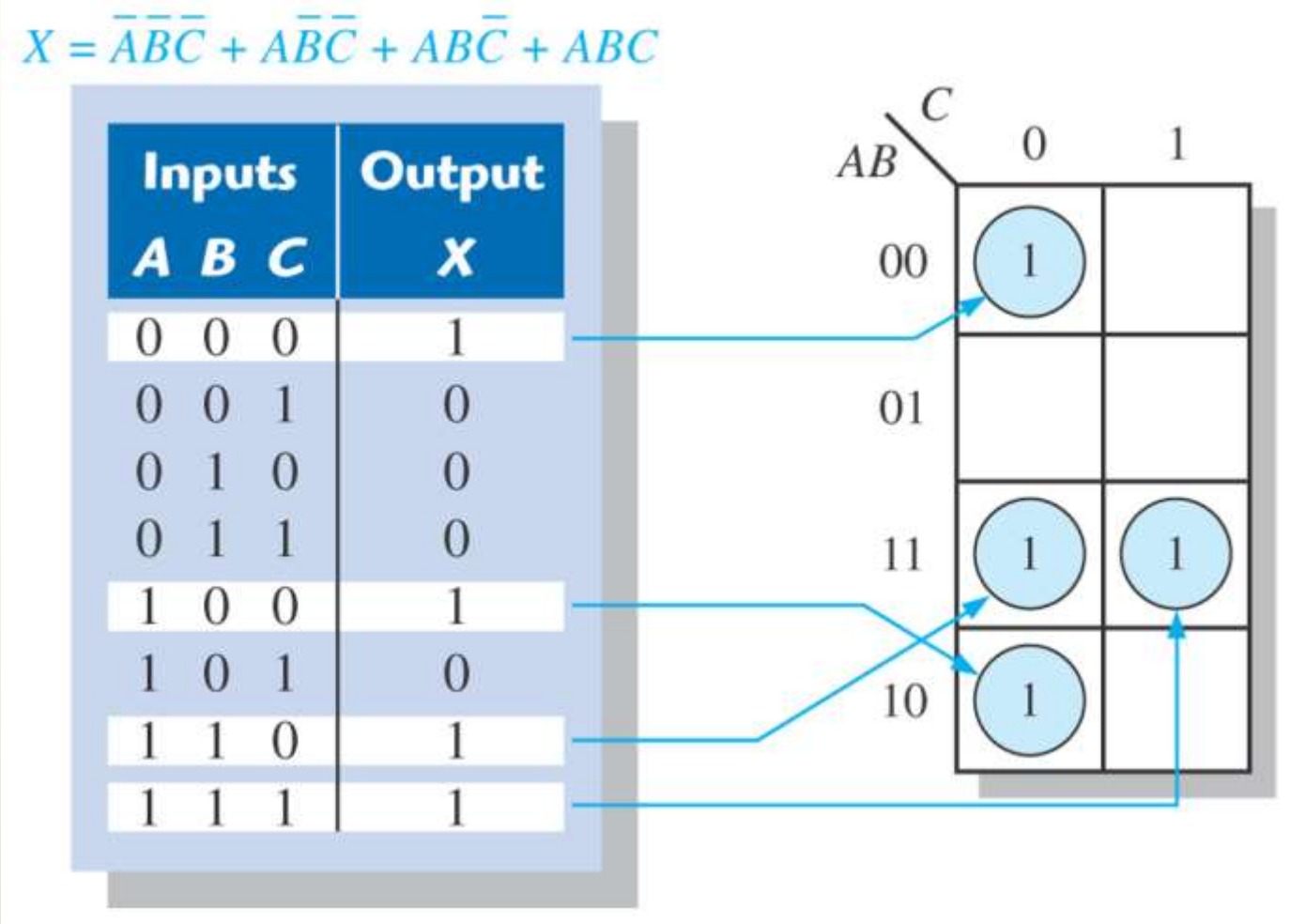
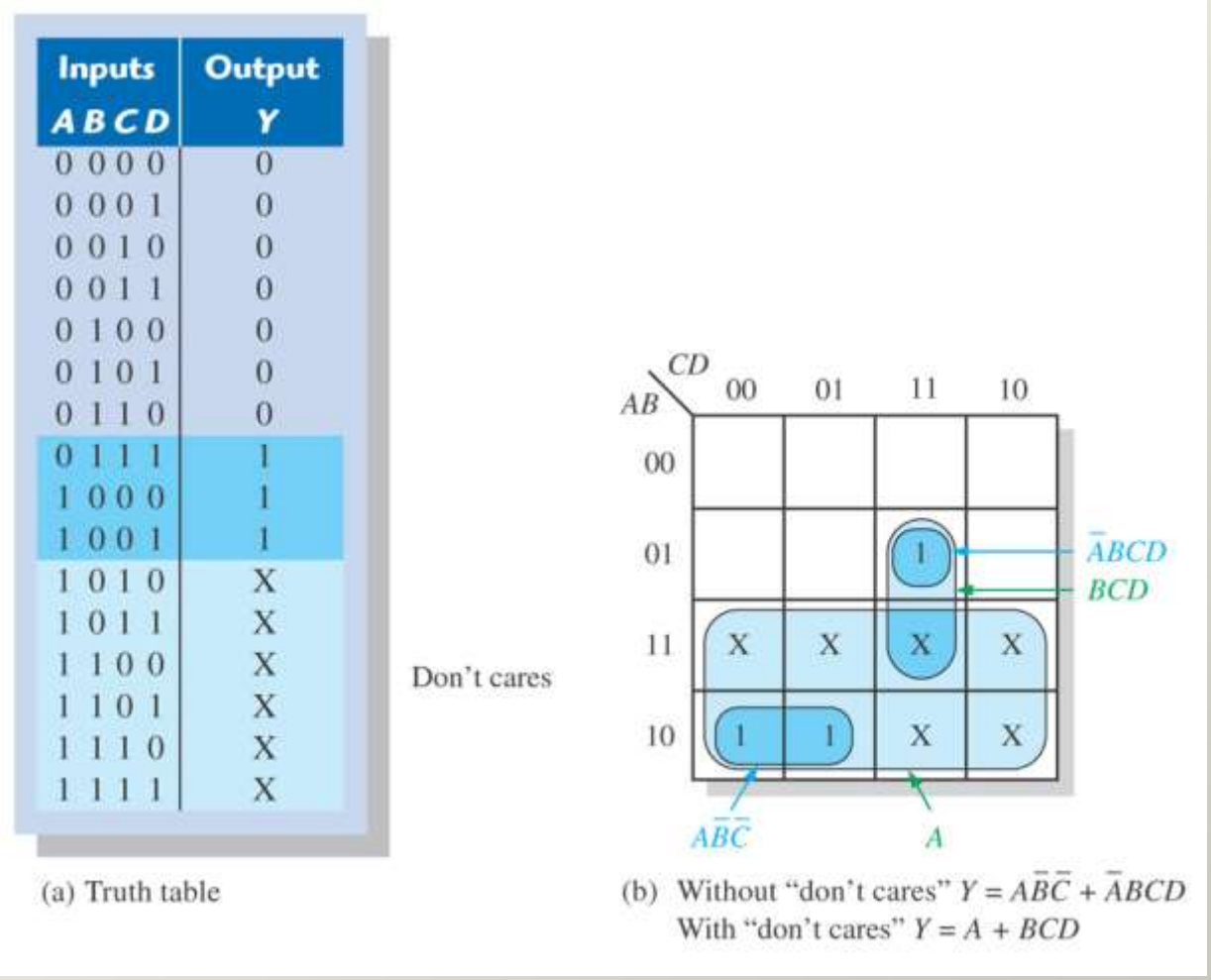


Figure 4–36 **Example of the use of "don't care" conditions to simplify an expression.**



| Inputs A B C D | Output Y |
|---|---|
| 0 0 0 0 | 0 |
| 0 0 0 1 | 0 |
| 0 0 1 0 | 0 |
| 0 0 1 1 | 0 |
| 0 1 0 0 | 0 |
| 0 1 0 1 | 0 |
| 0 1 1 0 | 0 |
| 0 1 1 1 | 1 |
| 1 0 0 0 | 1 |
| 1 0 0 1 | 1 |
| 1 0 1 0 | X |
| 1 0 1 1 | X |
| 1 1 0 0 | X |
| 1 1 0 1 | X |
| 1 1 1 0 | X |
| 1 1 1 1 | X |



(a) Truth table

(b) Without "don't cares" $Y = A\bar{B}\bar{C} + \bar{A}BCD$
With "don't cares" $Y = A + BCD$

Figure 4–37 **Example of mapping a standard POS expression.**

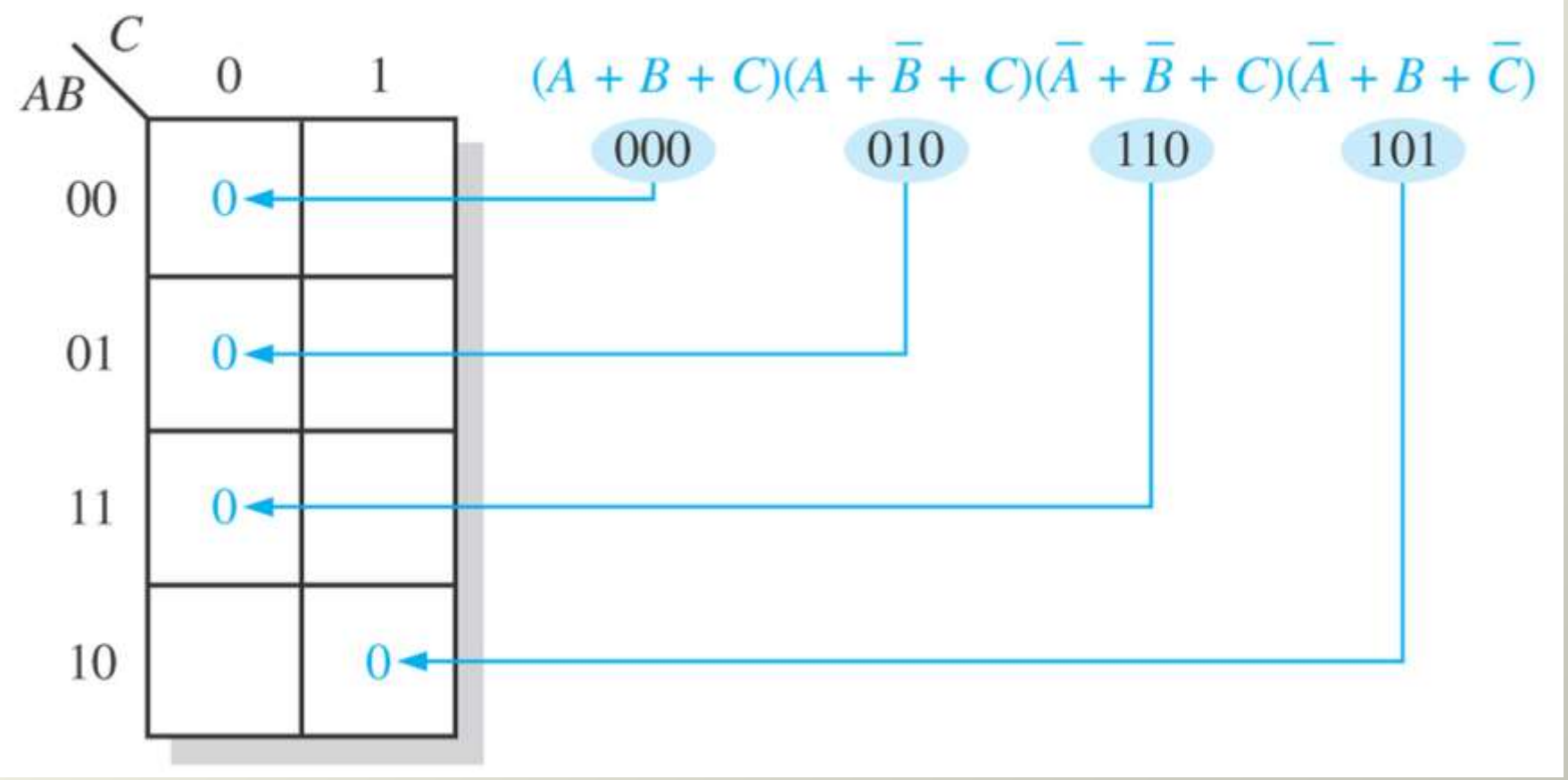


Figure 4–38

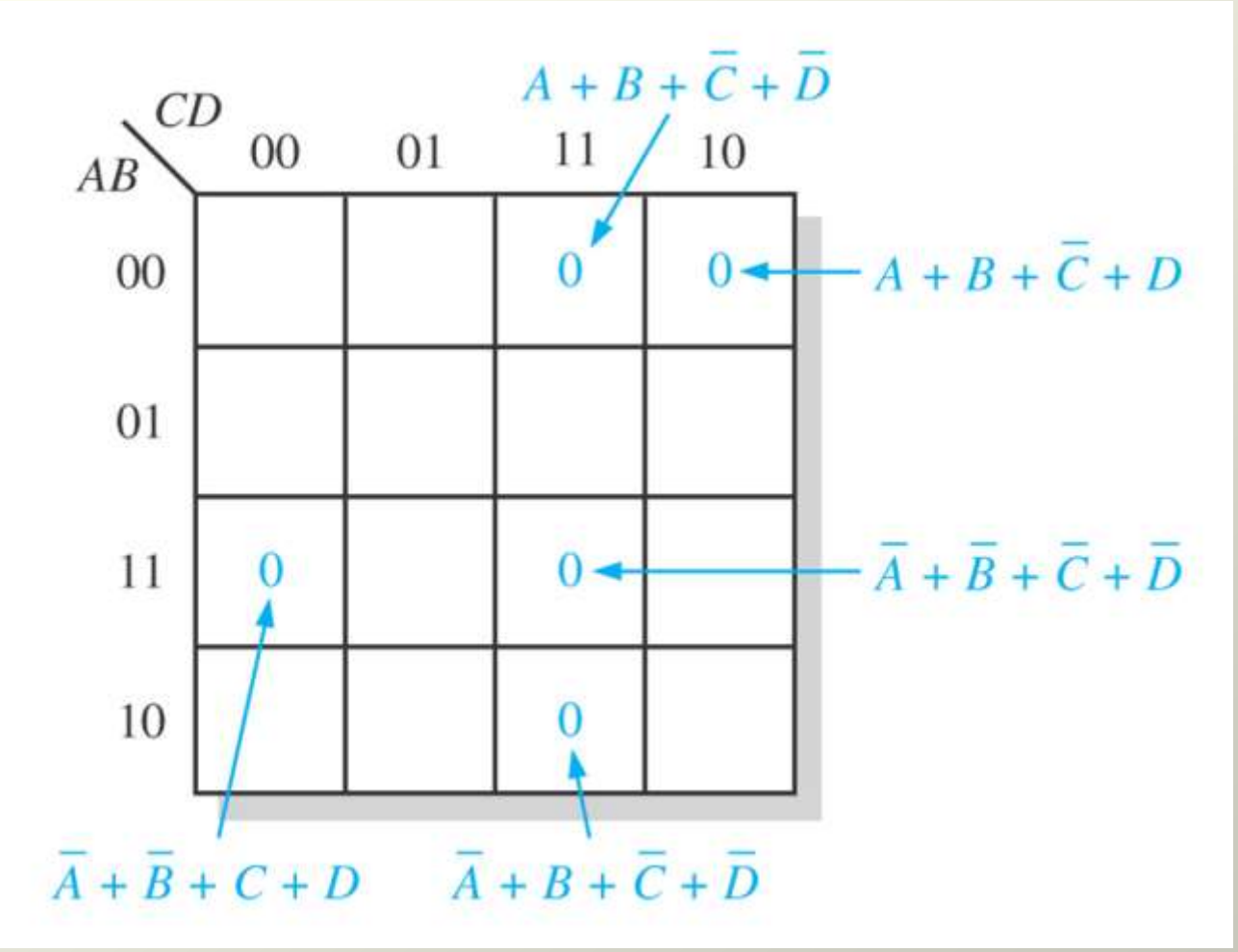

CD
AB
00
01
11
10
00
01
11
10
$A + B + \overline{C} + \overline{D}$
$A + B + \overline{C} + D$
$\overline{A} + \overline{B} + \overline{C} + \overline{D}$
$\overline{A} + \overline{B} + C + D$
$\overline{A} + B + \overline{C} + \overline{D}$
0

Figure 4–39

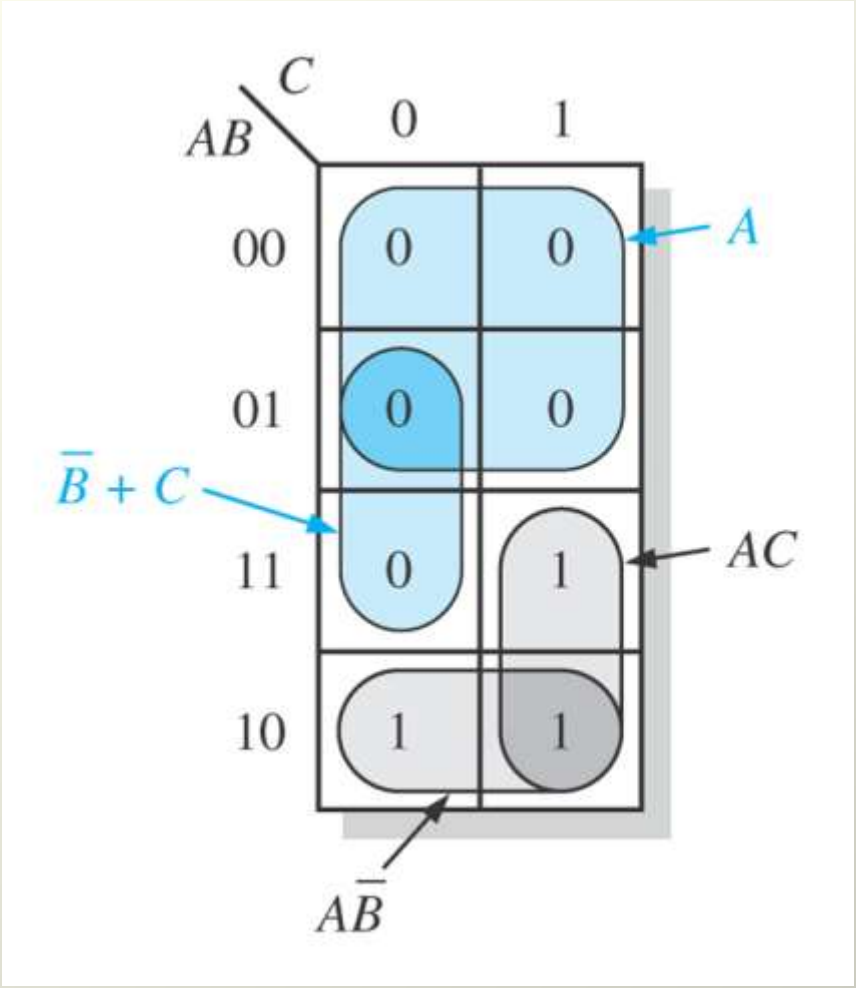


Figure 4–40

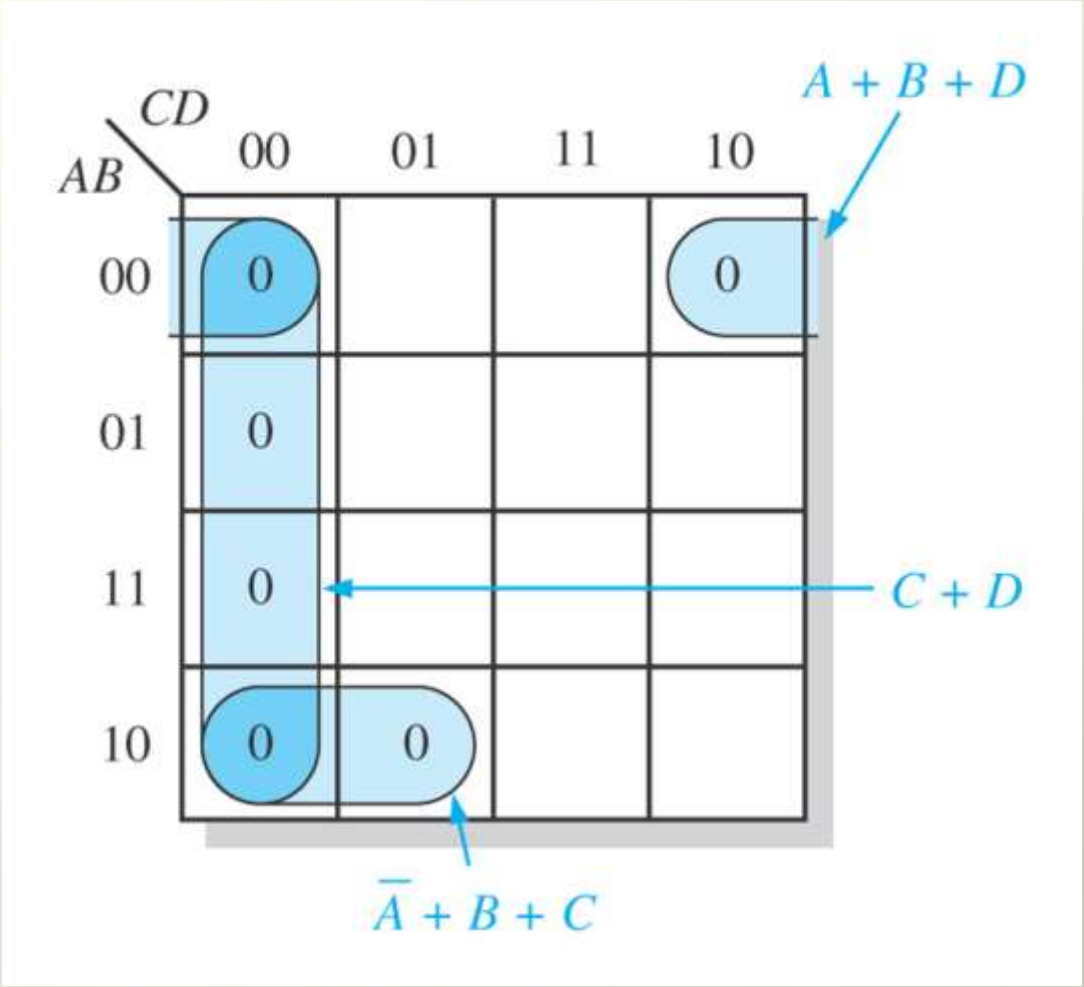
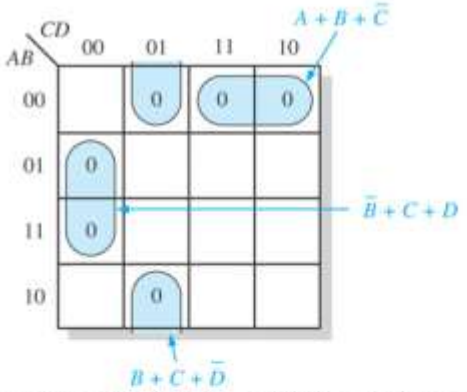
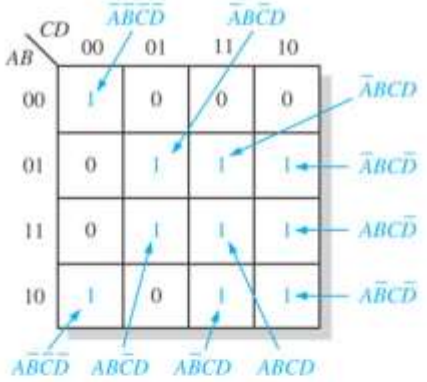


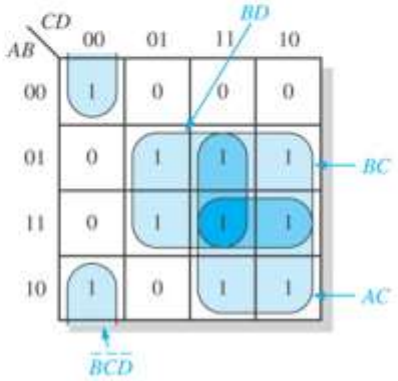
Figure 4–41




(a) Minimum POS: $(A + B + C)(\bar{B} + \bar{C} + D)(B + C + \bar{D})$

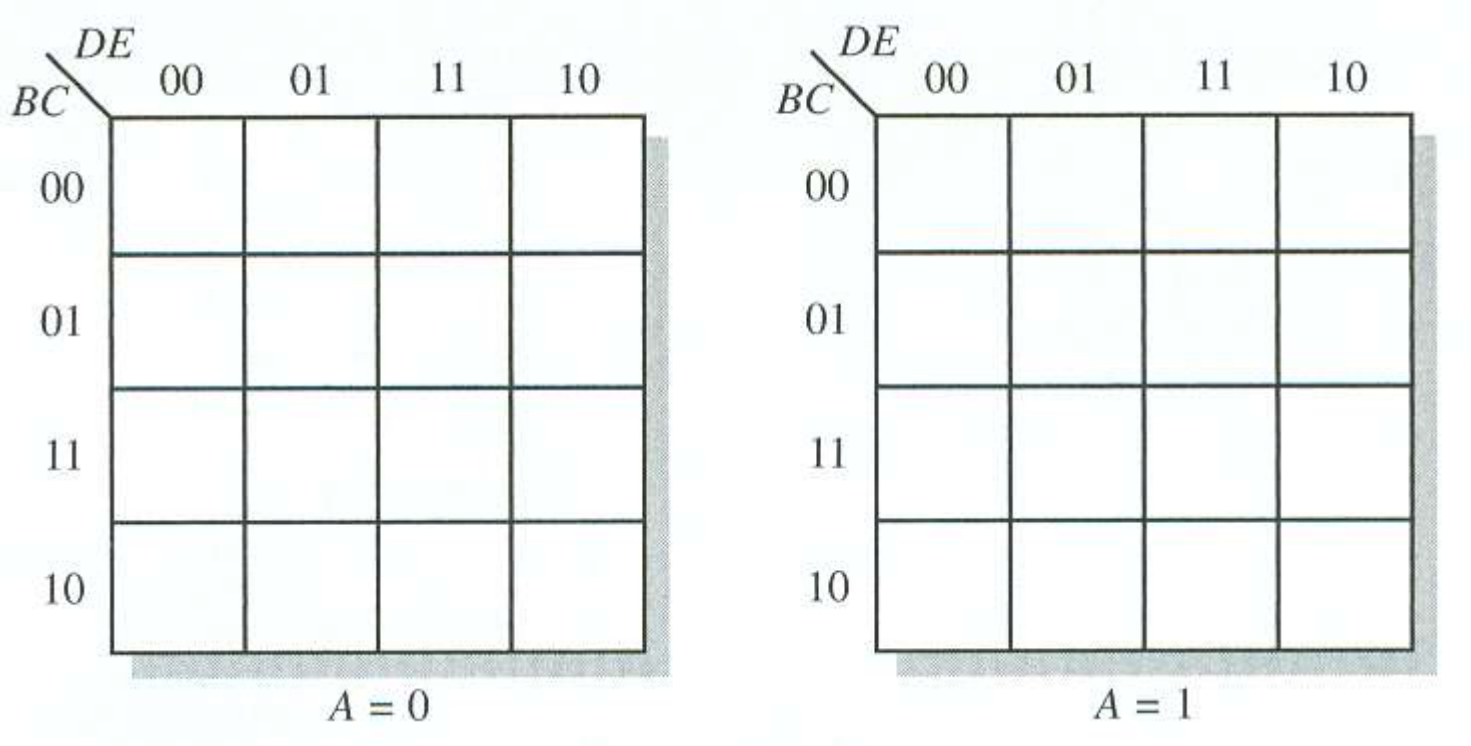



(b) Standard SOP:
$\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}BCD + \bar{A}BC\bar{D} + ABC\bar{D} + A\bar{B}C\bar{D} +$
$A\bar{B}\bar{C}\bar{D} + AB\bar{C}D + A\bar{B}CD + ABCD$




(c) Minimum SOP: $AC + BC + BD + \bar{B}\bar{C}\bar{D}$

# The Karnaugh Map




**5-Variable Karnaugh Mapping**

Figure 4–43 **Illustration of groupings of 1s in adjacent cells of a 5-variable map.**

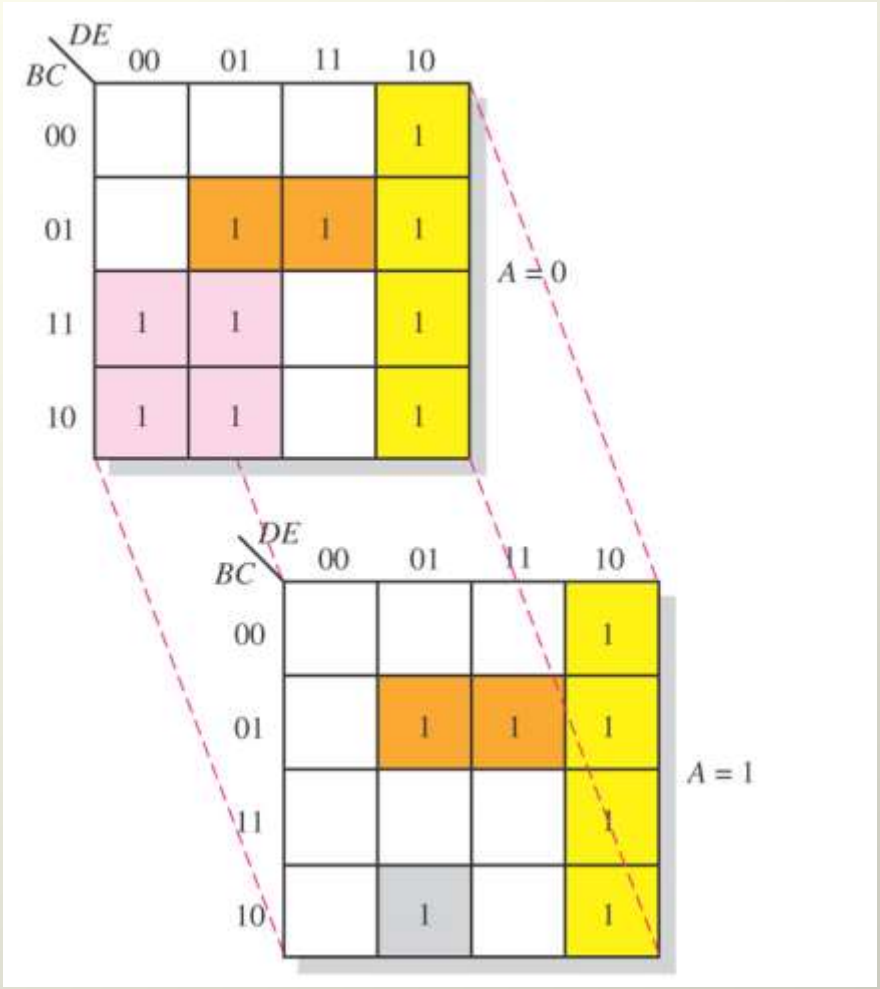
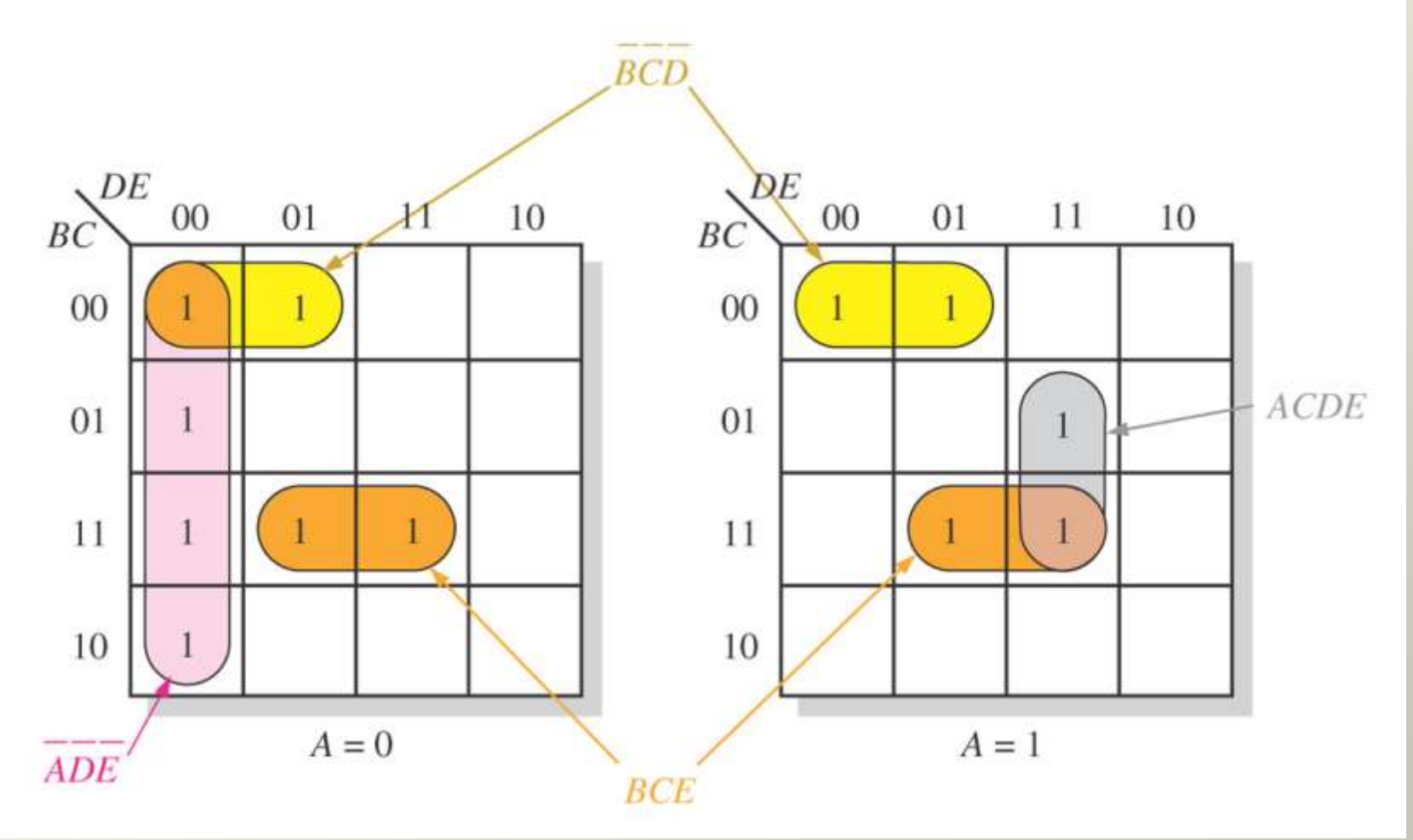


Figure 4–44

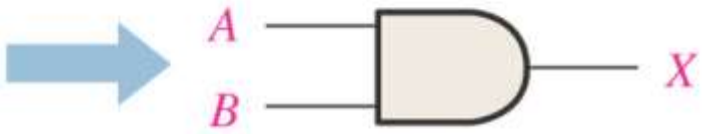


# VHDL

Figure 4–45 **A VHDL program for a 2-input AND gate.**

```
entity AND_Gate2 is
  port (A, B: in bit; X: out bit);
end entity AND_Gate2;


architecture LogicFunction of AND_Gate2 is
begin
  X <= A and B;
end architecture LogicFunction;
```



# VHDL

- VHDL Operators

  **and**
  **or**
  **not**
  **nand**
  **nor**
  **xor**
  **xnor**

- VHDL Elements

  entity
  architecture

# VHDL

- Entity Structure

Example:

```
entity AND_Gate1 is
    port(A,B:in bit:X:out bit);
end entity AND_Gate1
```

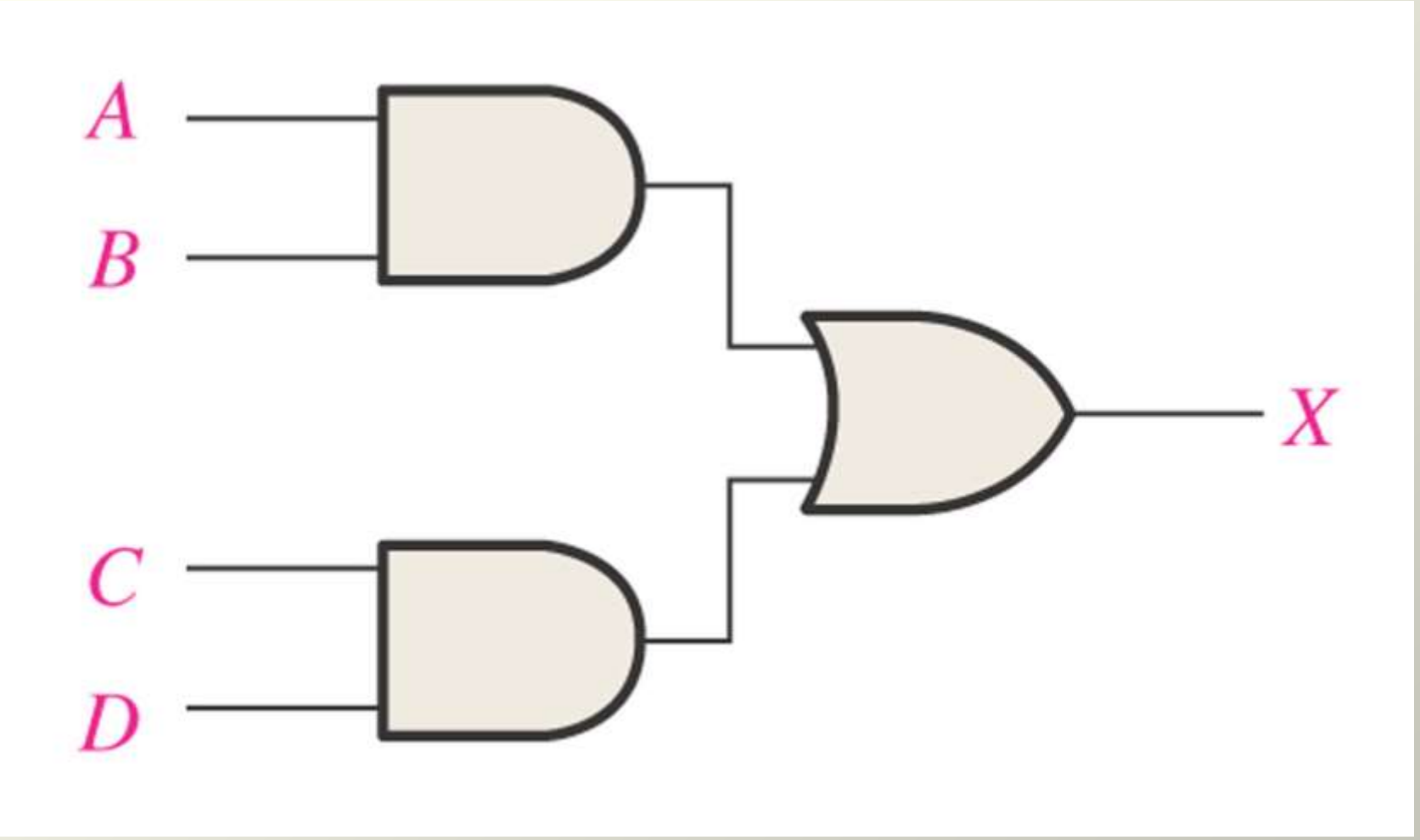
# VHDL

- Architecture

Example:

```
architecture LogicFunction of AND_Gate1 is
begin
    X<=A and B;
end architecture LogicFunction
```

Figure 4–46



# Hardware Description Languages (HDL)

- Boolean Expressions in VHDL

| | |
|---|---|
| AND | X **<=** A **and** B**;** |
| OR | X **<=** A **or** B**;** |
| NOT | X **<=** A **not** B**;** |
| NAND | X **<=** A **nand** B**;** |
| NOR | X **<=** A **nor** B**;** |
| XOR | X **<=** A **xor** B**;** |
| XNOR | X **<=** A **xnor** B**;** |

Figure 4–47 **Seven-segment display format showing arrangement of segments.**

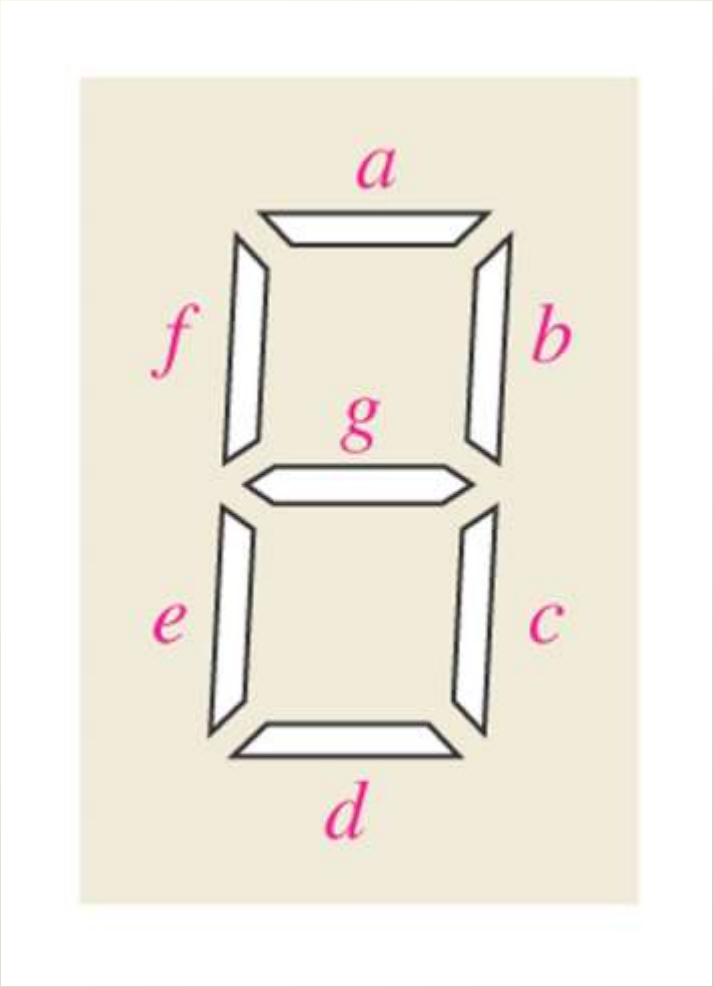


Figure 4–48 **Display of decimal digits with a 7-segment device.**

Figure 4–49 **Arrangements of 7-segment LED displays.**

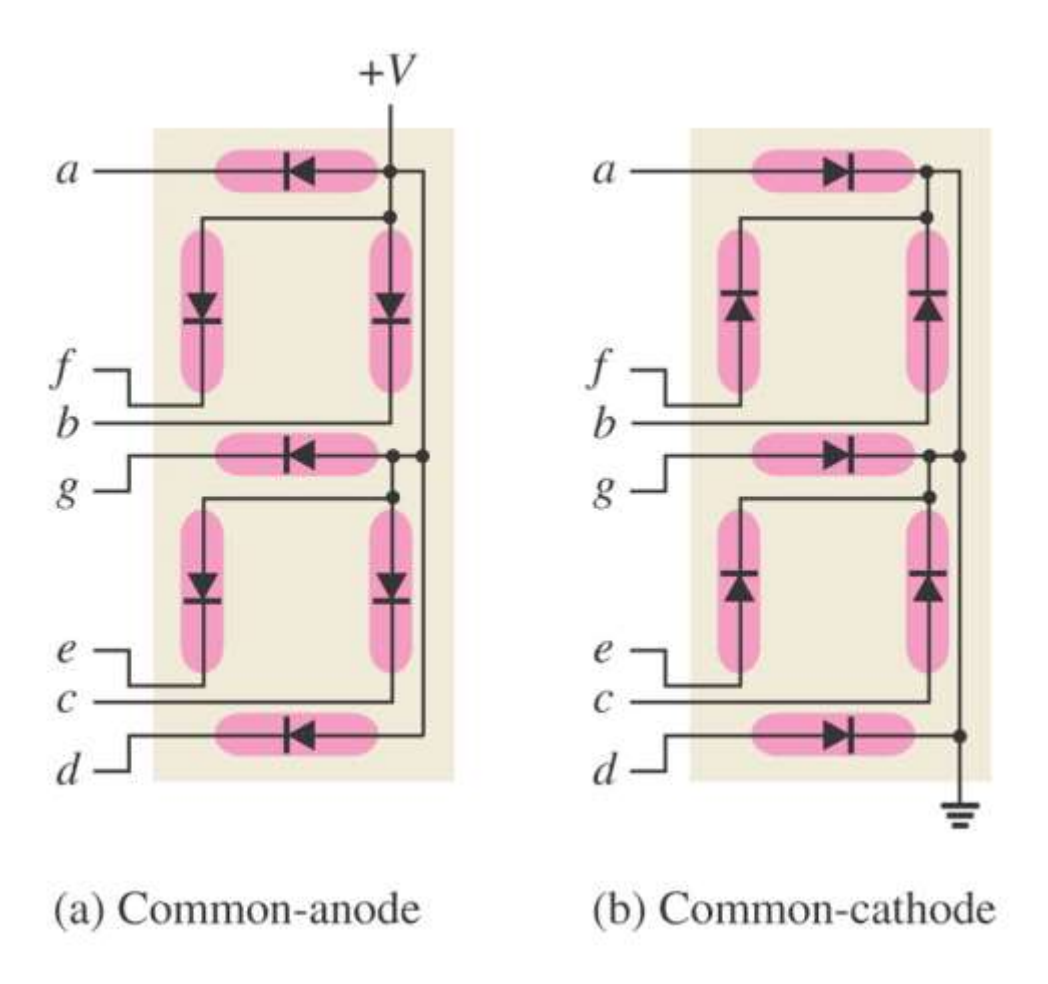


Figure 4–50 **Block diagram of 7-segment logic and display.**

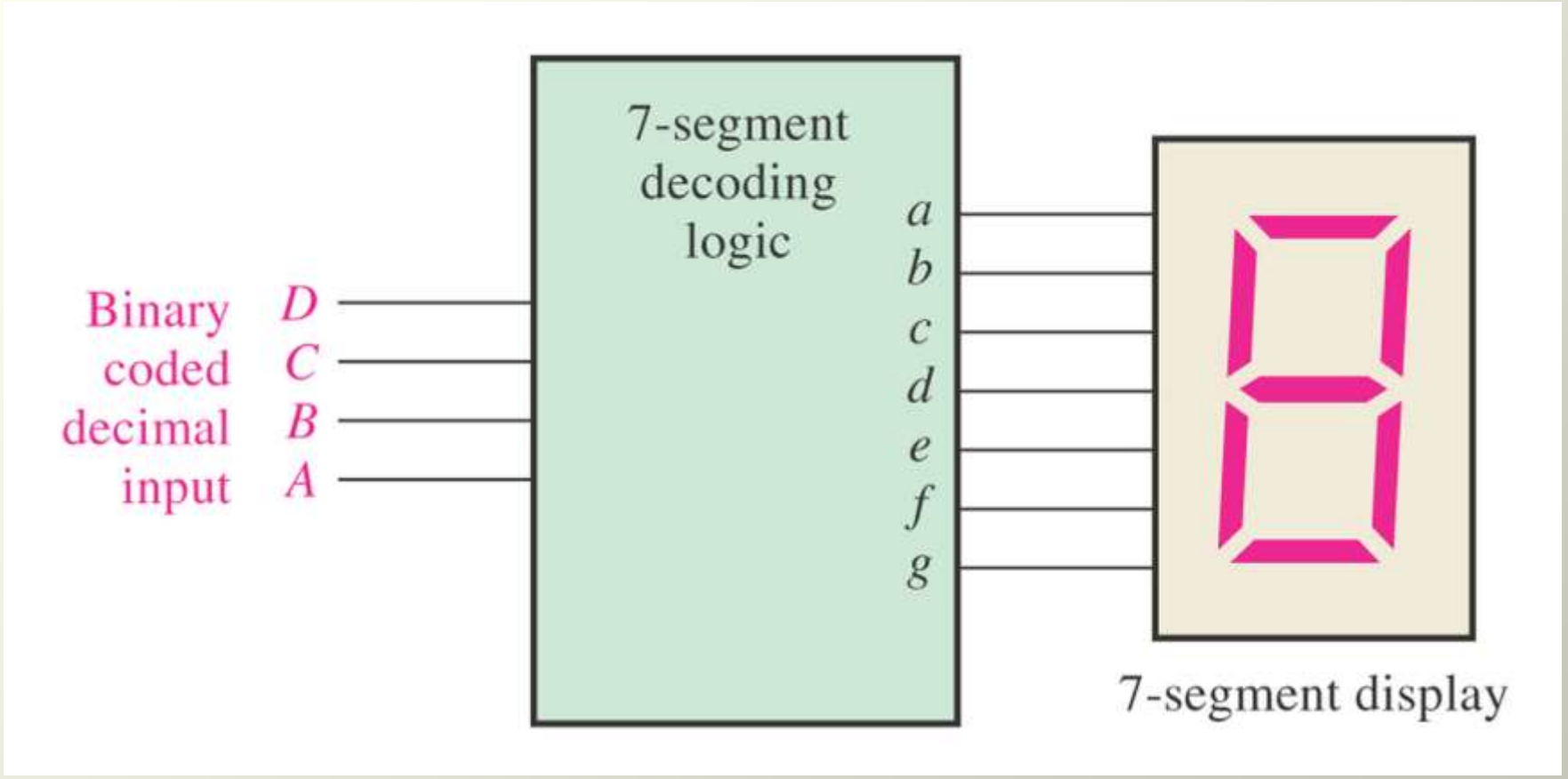
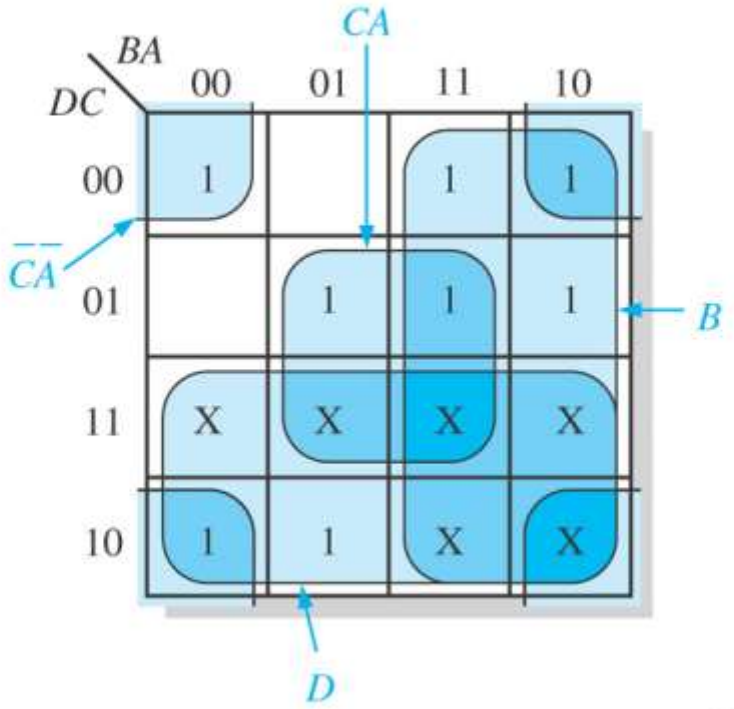


Figure 4–51 **Karnaugh map minimization of the segment-a logic expression.**

Standard SOP expression:

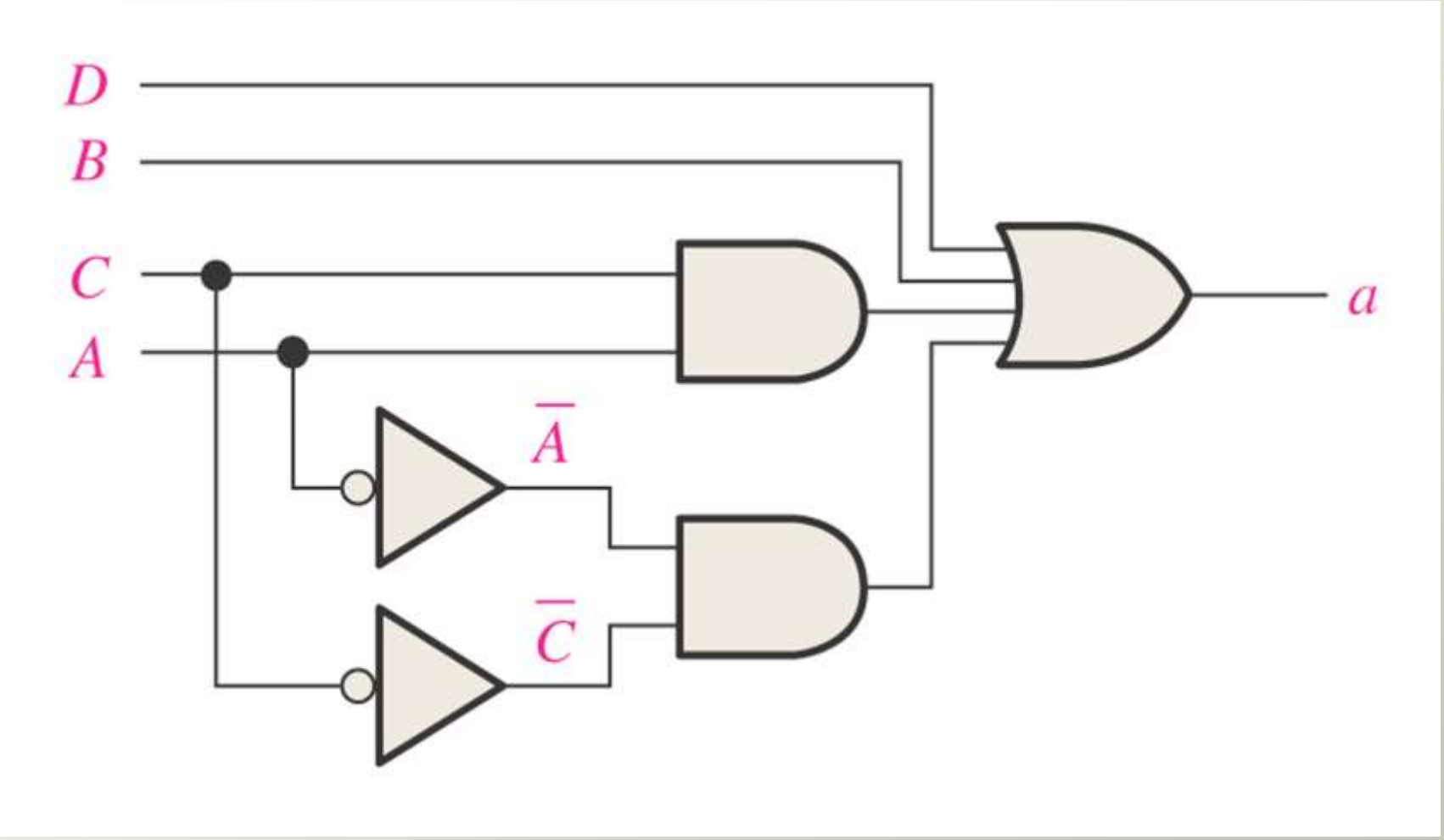
$\overline{D}\overline{C}\overline{B}\overline{A} + \overline{D}\overline{C}\overline{B}A + \overline{D}\overline{C}B\overline{A} + \overline{D}\overline{C}BA + \overline{D}C\overline{B}\overline{A} + \overline{D}C\overline{B}A + \overline{D}CB\overline{A} + \overline{D}CBA + D\overline{C}\overline{B}\overline{A} + D\overline{C}\overline{B}A$




Minimum SOP expression: $D + B + CA + \overline{C}\overline{A}$

Figure 4–52

**The minimum logic implementation for segment a of the 7-segment display.**



# Summary

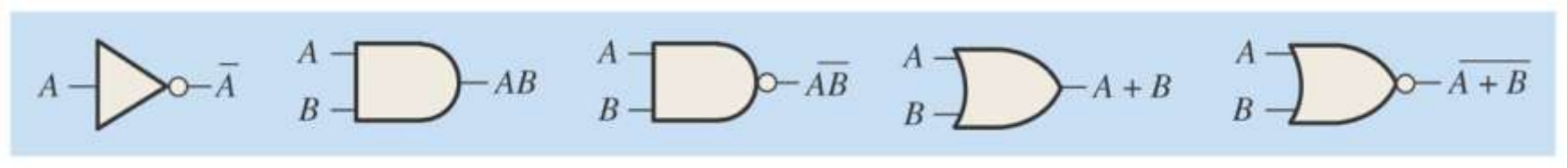


Figure 4–54

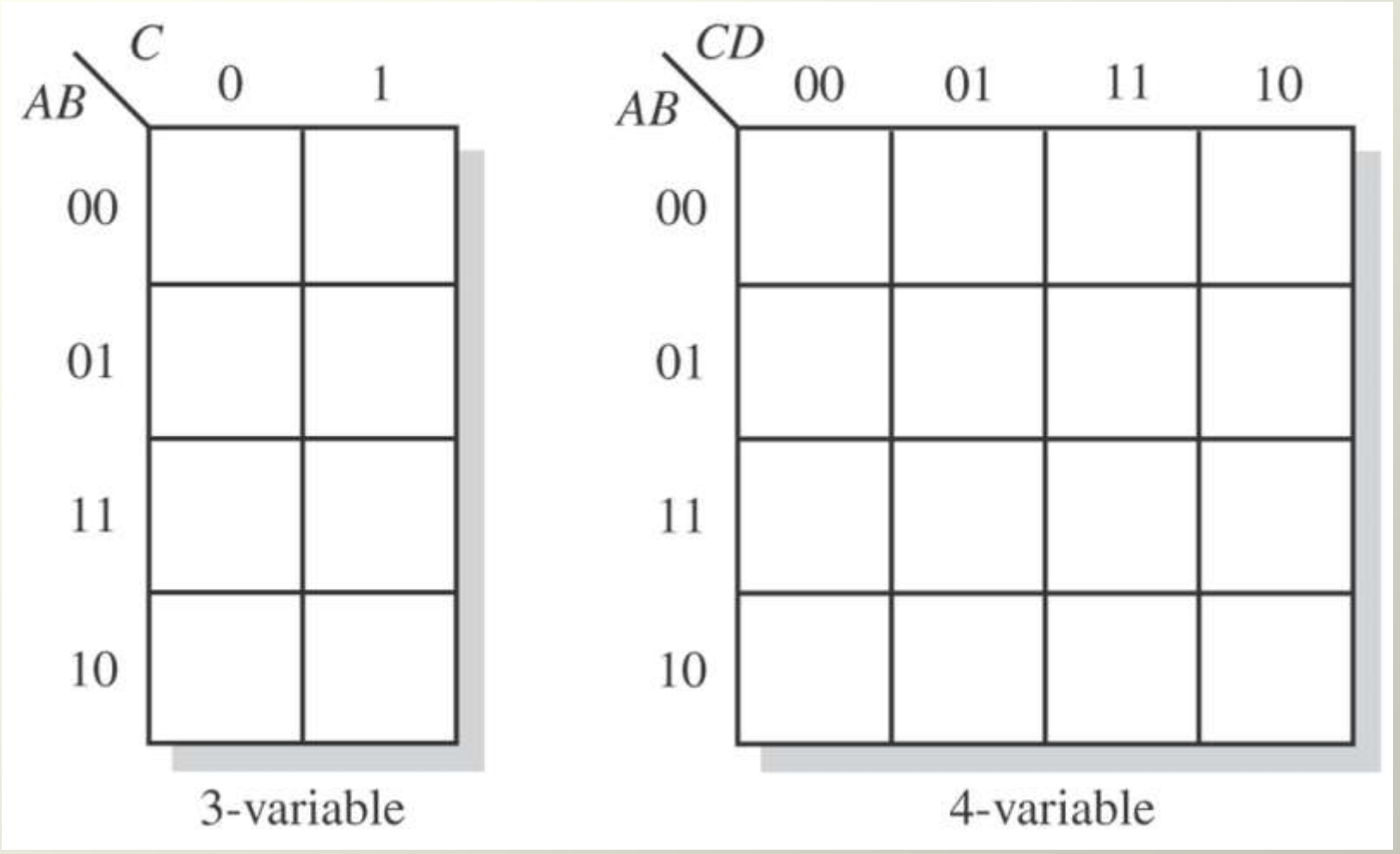

C
AB
0
1
00
01
11
10
3-variable
CD
AB
00
01
11
10
00
01
11
10
4-variable

Figure 4–55. What is the Boolean expression for each of the logic gates?

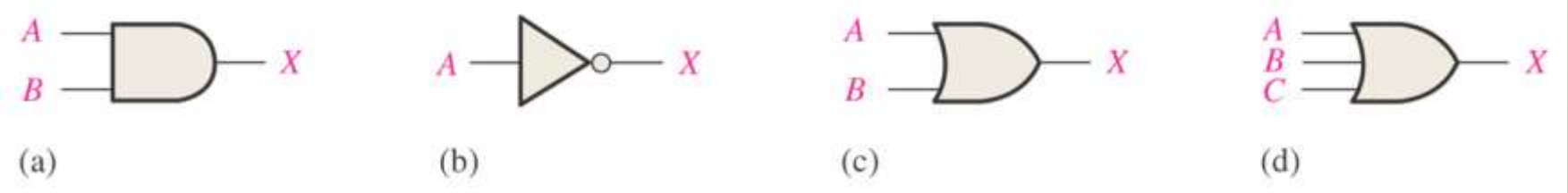


Figure 4–56. What is the Boolean expression for each of the logic gates?

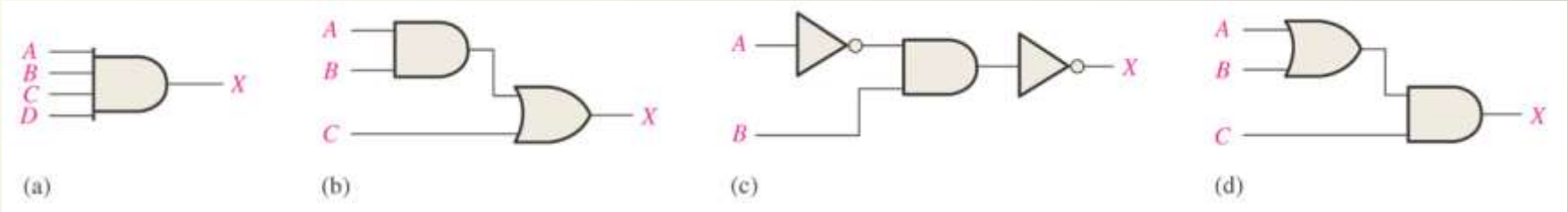
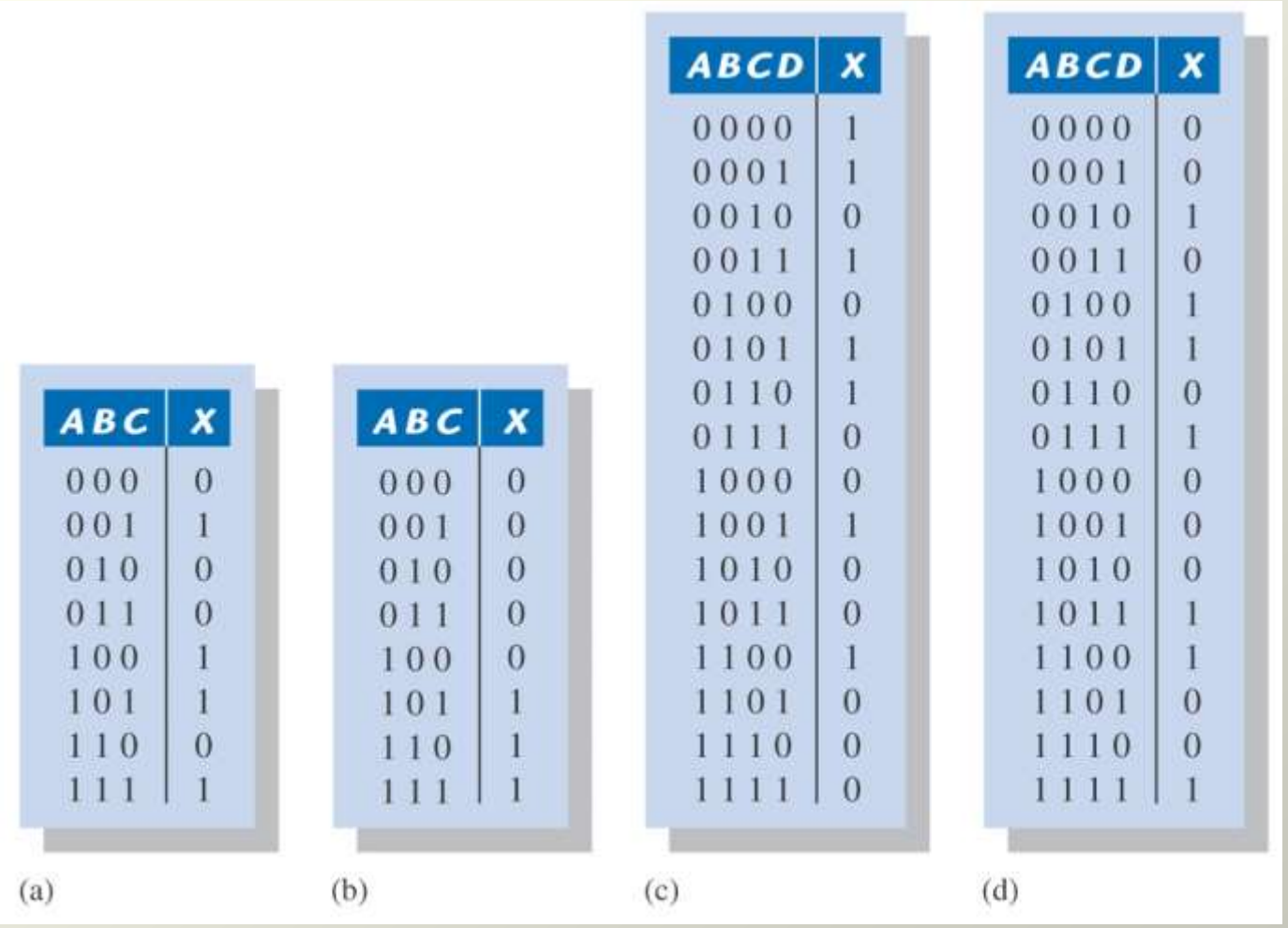


Figure 4–58. Derive a standard SOP and standard POS expression for each truth table.



(a)

| A B C | X |
|---|---|
| 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 | 0 |
| 1 1 1 | 1 |

(b)

| A B C | X |
|---|---|
| 0 0 0 | 0 |
| 0 0 1 | 0 |
| 0 1 0 | 0 |
| 0 1 1 | 0 |
| 1 0 0 | 0 |
| 1 0 1 | 1 |
| 1 1 0 | 1 |
| 1 1 1 | 1 |

(c)

| A B C D | X |
|---|---|
| 0 0 0 0 | 1 |
| 0 0 0 1 | 1 |
| 0 0 1 0 | 0 |
| 0 0 1 1 | 1 |
| 0 1 0 0 | 0 |
| 0 1 0 1 | 1 |
| 0 1 1 0 | 1 |
| 0 1 1 1 | 0 |
| 1 0 0 0 | 0 |
| 1 0 0 1 | 1 |
| 1 0 1 0 | 0 |
| 1 0 1 1 | 0 |
| 1 1 0 0 | 1 |
| 1 1 0 1 | 0 |
| 1 1 1 0 | 0 |
| 1 1 1 1 | 0 |

(d)

| A B C D | X |
|---|---|
| 0 0 0 0 | 0 |
| 0 0 0 1 | 0 |
| 0 0 1 0 | 1 |
| 0 0 1 1 | 0 |
| 0 1 0 0 | 1 |
| 0 1 0 1 | 1 |
| 0 1 1 0 | 0 |
| 0 1 1 1 | 1 |
| 1 0 0 0 | 0 |
| 1 0 0 1 | 0 |
| 1 0 1 0 | 0 |
| 1 0 1 1 | 1 |
| 1 1 0 0 | 1 |
| 1 1 0 1 | 0 |
| 1 1 1 0 | 0 |
| 1 1 1 1 | 1 |

Figure 4–59.
Reduce the function in the truth table to its minimum SOP form by using a Karnaugh map.

| Inputs | | | Output |
|---|---|---|---|
| A | B | C | X |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Figure 4–62. Example 4-21. Related Problem answer.

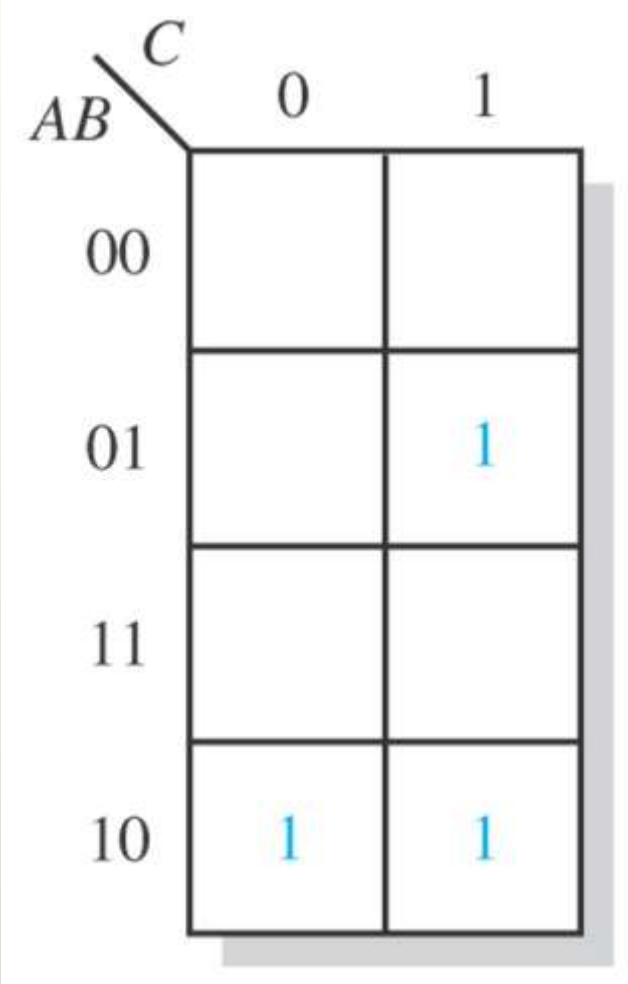
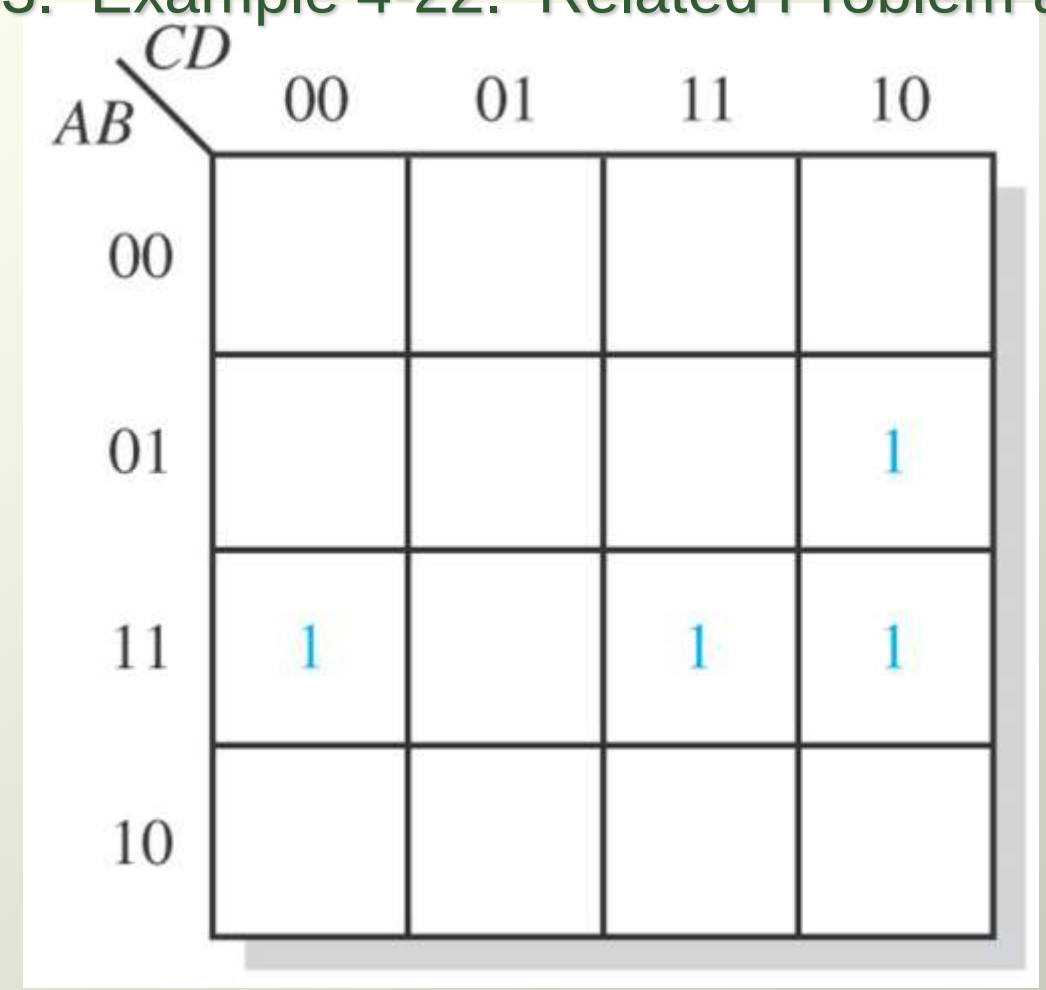
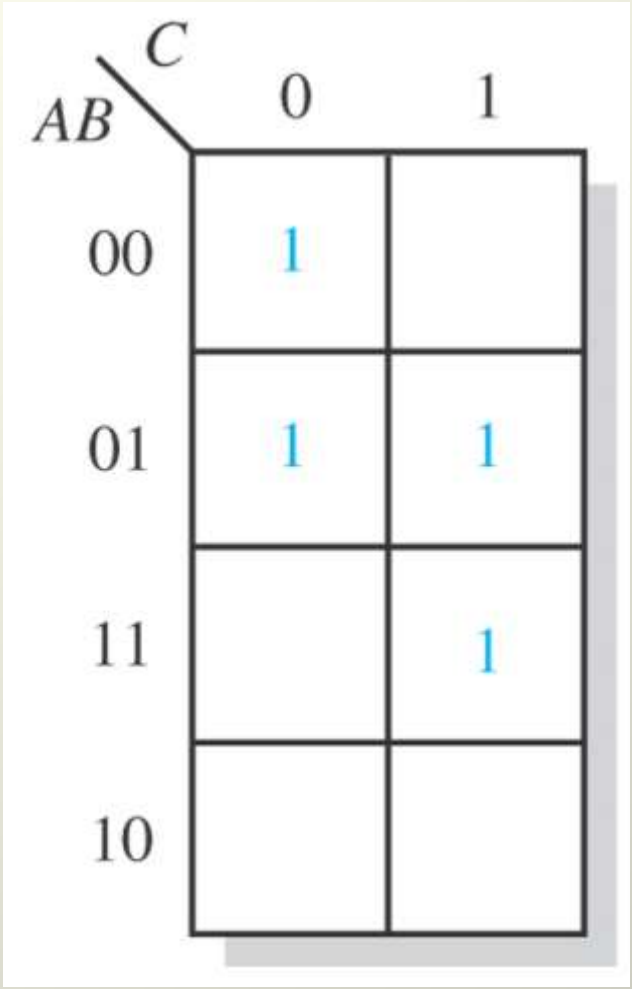


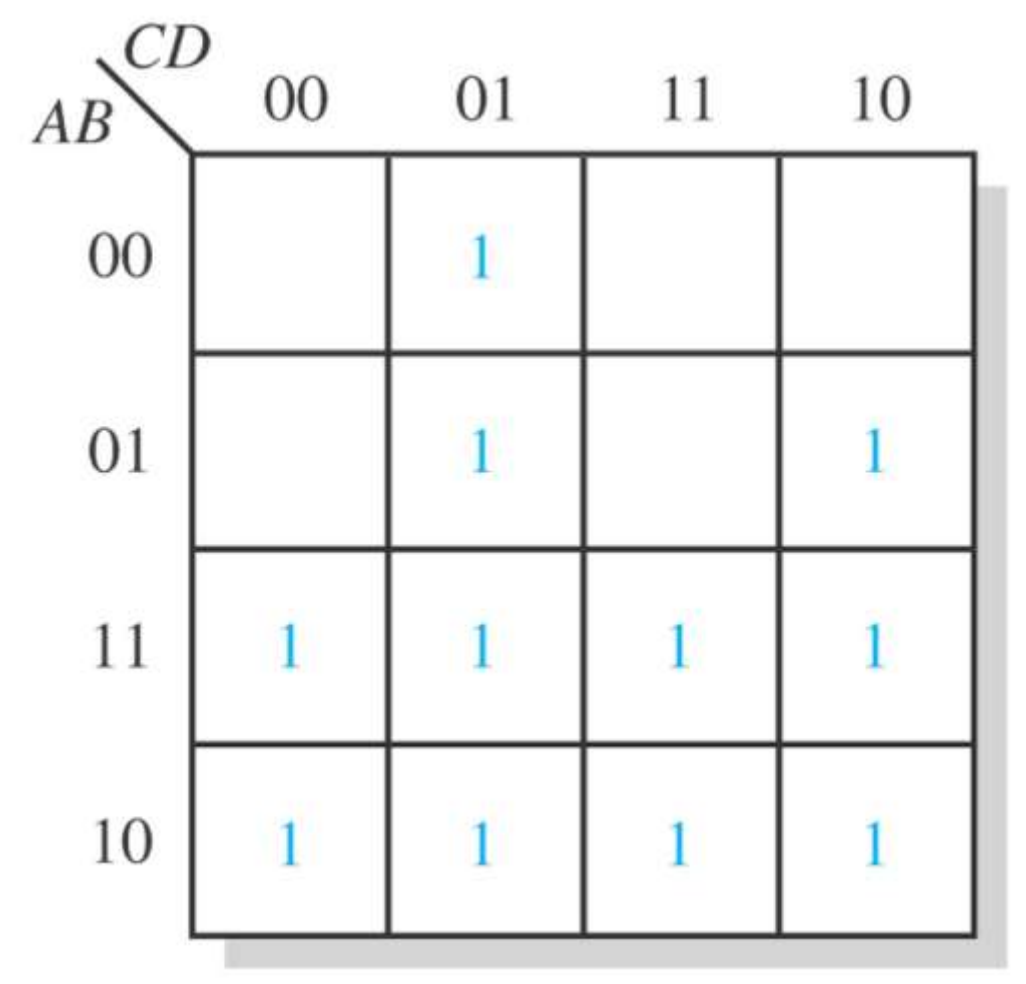
Figure 4–63. Example 4-22. Related Problem answer.



# Figure 4–64. Example 4-23. Related Problem answer.



# Figure 4–65. Example 4-24. Related Problem answer.



# Figure 4–66. Example 4-30. Related Problem answer.