Chapter 3

Boolean Algebra And Combinational Logic

Key Terms

Logic Gate Network Two or more logic gates connected together.

Logic Diagram A diagram, similar to a schematic, showing the connection of logic gates.

Combinational Logic Digital circuitry in which an output is derived from the combination of inputs, independent of the order in which they are applied.

Combinatorial Logic Another name for combinational logic.

Sequential Logic Digital circuitry in which the output state of the circuit depends not only on the states of the inputs, but also on the sequence in which they reached their present states.

Logic Gate Network

- Two or more logic gates connected together
  （兩個或以上的邏輯閘連接在一起）

- Described by truth table, logic diagram, or Boolean expression
  （使用真值表，邏輯電路，或布林代數來表示）
- Boolean expression from a gate network

FIGURE 3.1 Boolean Expression from a Gate Network

Boolean Expression for Logic Gate Network

- Similar to finding the expression for a single gate.
- Inputs may be compound expressions that represent outputs from previous gates.

KEY TERMS

Bubble-to-Bubble Convention The practice of drawing gates in a logic diagram so that inverting outputs connect to inverting inputs and noninverting outputs connect to noninverting inputs.

Order of Precedence The sequence in which Boolean functions are performed, unless otherwise specified by parentheses.

Example 3.1

Derive the Boolean expression of the logic gate network shown in Figure 3.2a.

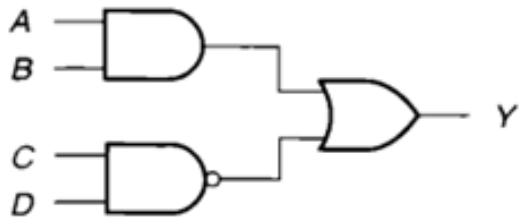


FIGURE 3.2 a. Logic gate network

Sol:

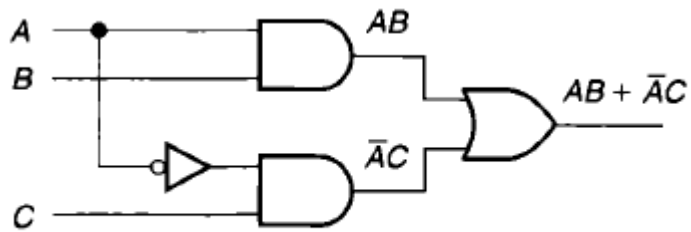
Example 3.2

Redraw the circuit in Figure 3.2 to conform to the bubble-to-bubble convention.
Write the Boolean expression of the new logic diagram.

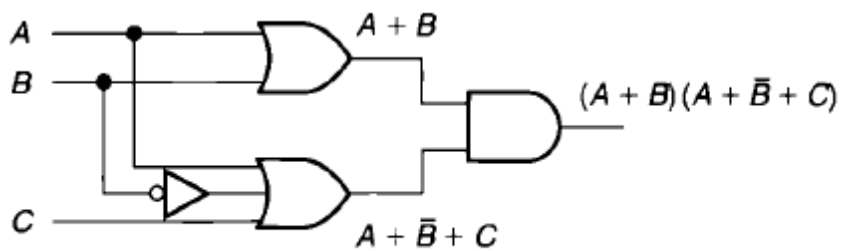
Sol:

Order of Precedence

- Unless otherwise specified, in Boolean expressions AND functions are performed first, followed by ORs. (AND比OR優先)
- To change the order of precedence, use parentheses.（可用小括號改變優先順序）



a. No parentheses required (AND, then OR)



b. Parentheses required (OR, then AND)

Example 3.3

Write the Boolean expression for the logic diagrams in Figure 3.5.

Sol:

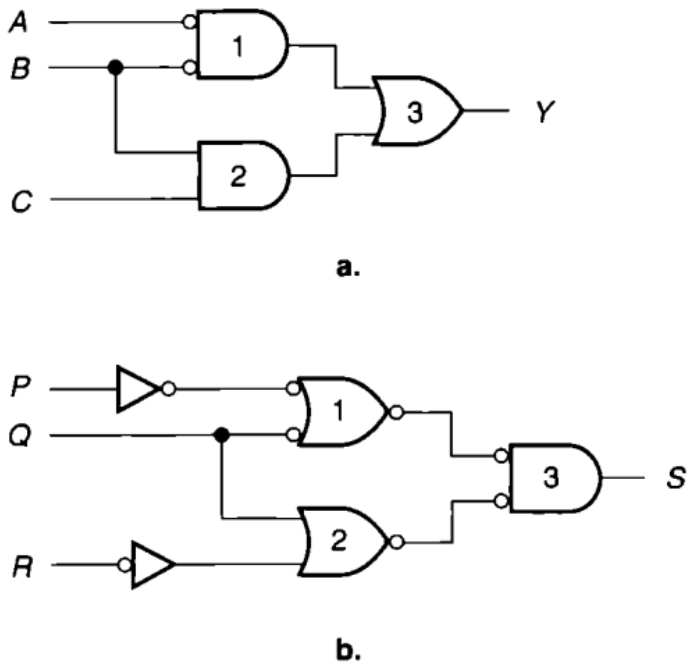


FIGURE 3.5 Example 3.3: Order of Precedence

Note: Simplification by Double Inversion

- When two bubbles touch, they cancel out.
- In Boolean expressions, bars of the same length cancel.

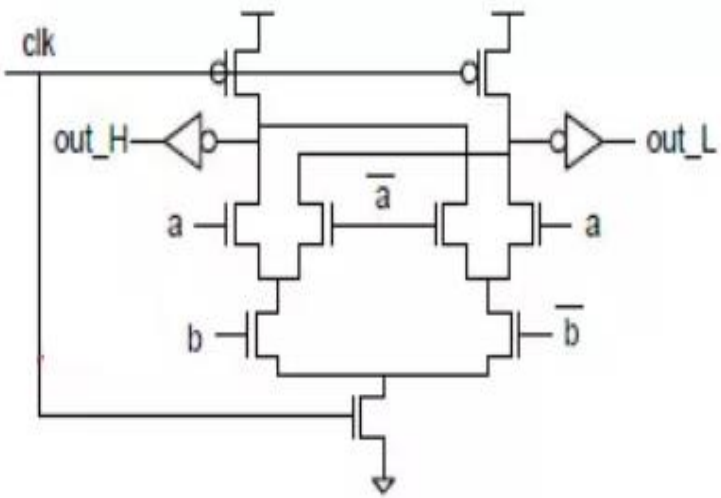
Logic Diagrams from Boolean Expressions

■ KEY TERMS

Levels of Gating The number of gates through which a signal must pass from input to output of a logic gate network.

Double-Rail Inputs Boolean input variables that are available to a circuit in both true and complement form.

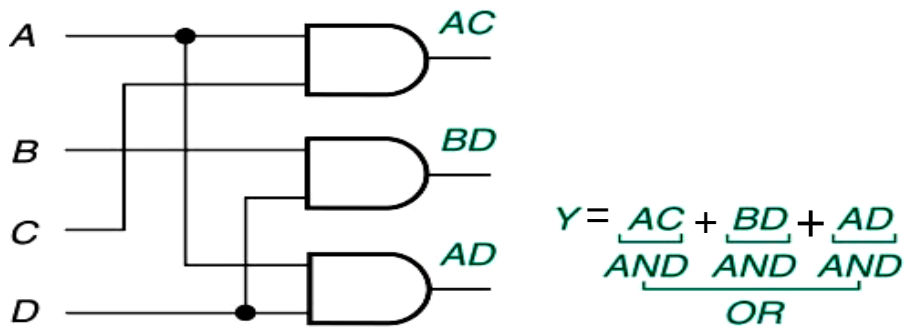
Dual-rail domino takes true and complementary inputs and producing true and complementary outputs.



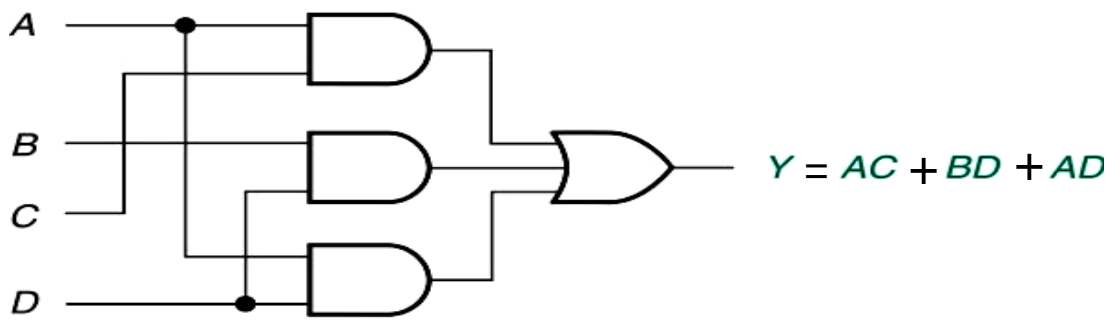
Synthesis The process of creating a logic circuit from a description such as a Boolean equation or truth table.

(Synthesize)
Logic Diagrams from Boolean Expressions

- Logic diagram for $Y=AC+BD+AD$
- Use order of precedence.



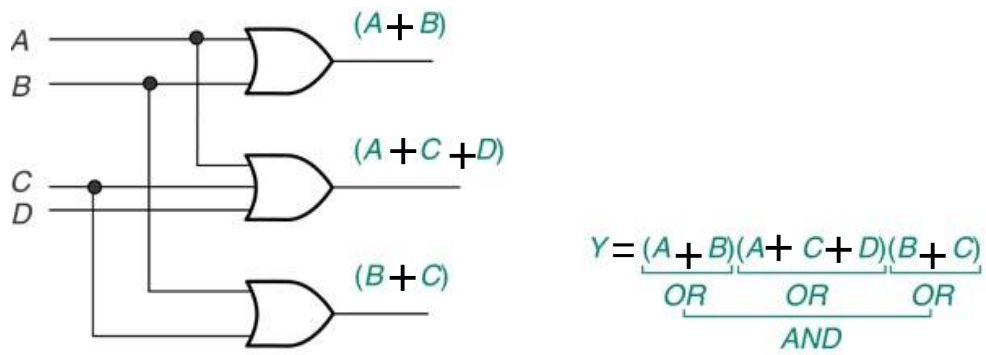
a. ANDs first



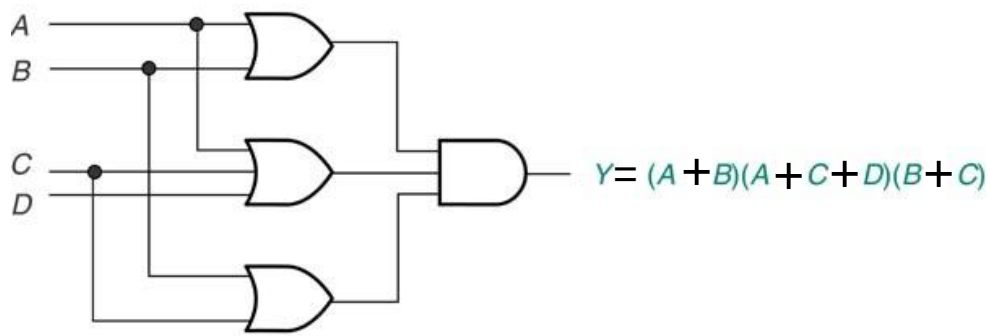
b. Combine ANDs in an OR gate

(Synthesize)
Logic Diagrams from Boolean Expressions

- Logic diagram for $Y=(A+B)(A+C+D)(B+C)$



a. ORs first



b. Combine ORs in an AND gate

Example 3.4 part 1

Synthesize the logic diagrams for the following Boolean expressions:

1. $P = \ P = Q\overline{RS} + \overline{S}T$

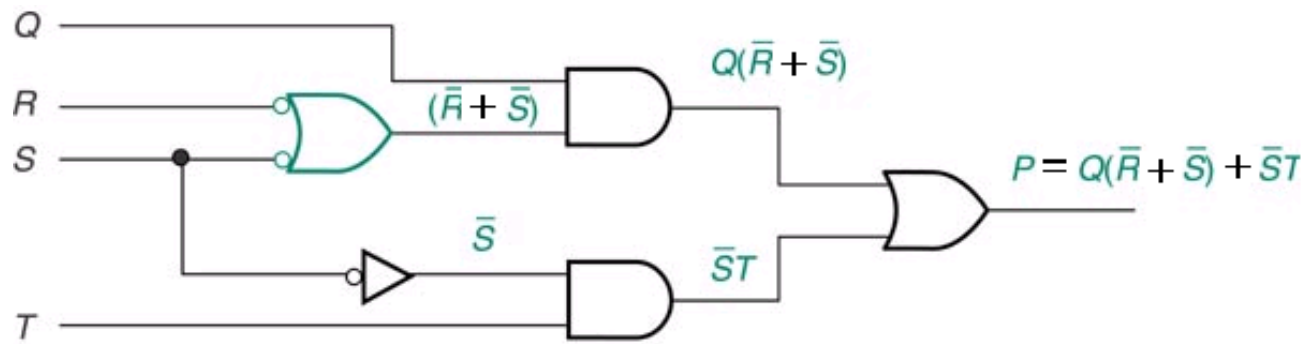
Example 3.4 part 2

Synthesize the logic diagrams for the following Boolean expressions:

2. $X = (W + Z + Y)\overline{V} + (\overline{W} + V)\overline{Y}$

Example 3.5 $\overline{RS}$

- Use DeMorgan's theorem to modify Boolean equation in part 1 of Example 3.4 so that there is no bar over any group of variables



a. Logic diagram of $P = Q(\bar{R}+\bar{S})+\bar{S}T$

Gating level can be further reduced from three to two (not counting inverters)

Truth Tables from Logic Diagrams or Boolean Expressions

- Two methods:
  - Combine individual truth tables from each gate into a final output truth table.
  - Develop a Boolean expression and use it to fill in the truth table.

Truth Tables from Logic Diagrams or Boolean Expressions

- Logic diagram for *AB*+*C*

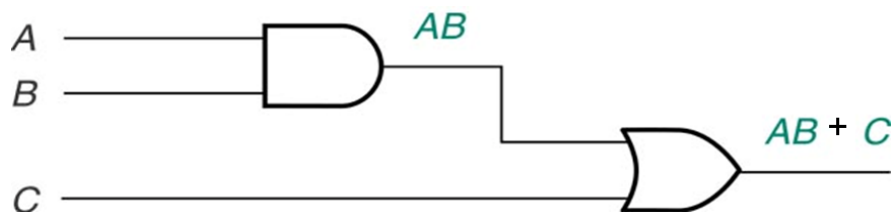


Fig. 3.12

TABLE 3.1 Truth Table for Figure 3.12

A	B	C	AB	$AB + C$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

Example 3.6

Derive the truth table for the logic diagram shown in Figure 3.13.

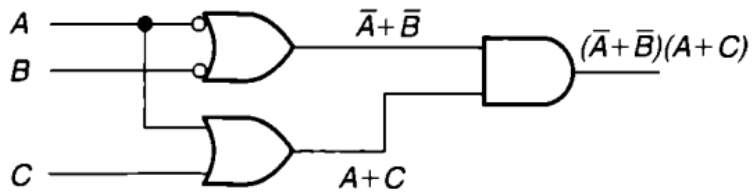


FIGURE 3.13 Example 3.6: Logic Diagram

■ **Solution**

TABLE 3.2 Truth Table for Figure 3.13

A	B	C	$(\bar{A}+\bar{B})$	$(A+C)$	$(\bar{A}+\bar{B})\ (A+C)$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

3.2 Sum-of-Product and Product-of-Sums Form

- Circuit Description Using Boolean Expressions

- Product term:
  - Part of a Boolean expression where one or more true or complement variables are ANDed (e.g., $\overline{A}\overline{C}$).

- Sum term:
  - Part of a Boolean expression where one or more true or complement variables are ORed (e.g., $A+C+D$).

Circuit Description Using Boolean Expressions

- Sum-of-products (SOP積之和):
  - A Boolean expression where several product terms are summed (ORed) together.

$$\text{SOP}: \; Y = AB + B\overline{C} + \overline{A}D$$

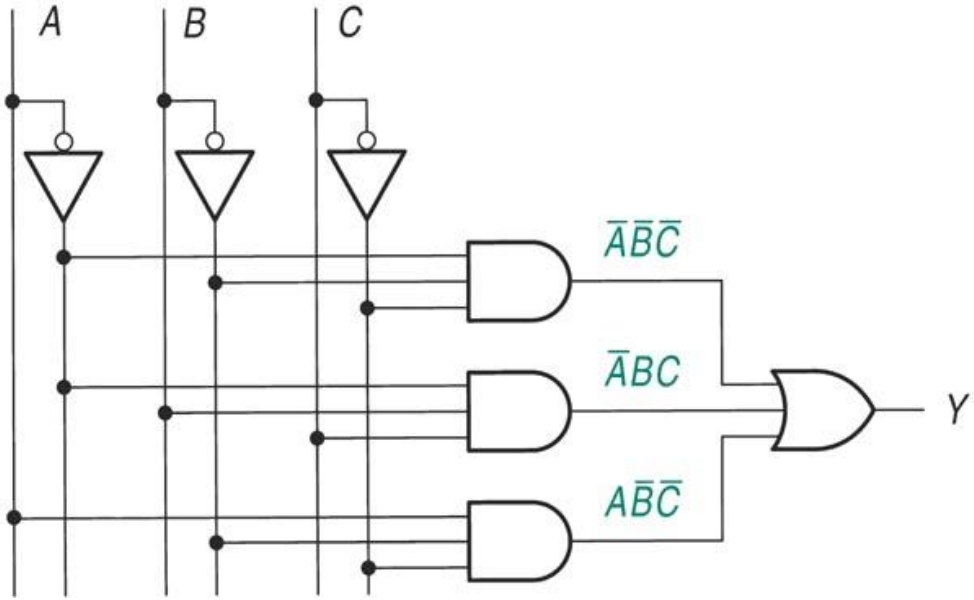
- Product-of-sum (POS和之積):
  - A Boolean expression where several sum terms are multiplied (ANDed) together.

$$\text{POS}: \; Y = (\overline{A} + B) \bullet (B + \overline{C}) \bullet (A + C)$$

- SOP and POS formats are used to present a summary of the circuit operation.

Bus Form

- A schematic convention(公約,協定) in which each variable is available, in true or complement form, at any point along a conductor.



Deriving a SOP Expression from a Truth Table

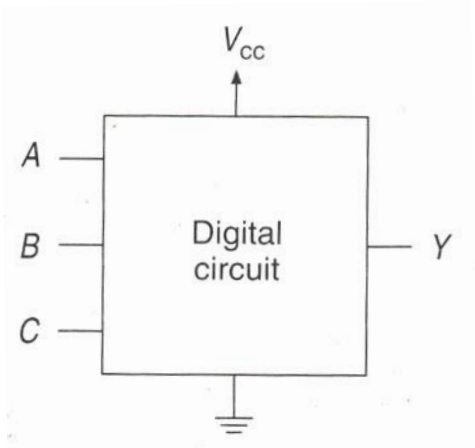
- Each line of the truth table with a 1 (HIGH) output represents a product term.

  （真值表中，每一個輸出為High的項就是一個乘積項）

- Each product term is summed (ORed).

  （將每一個乘積項加起來）

Sum-of-products (SOP)



(1)

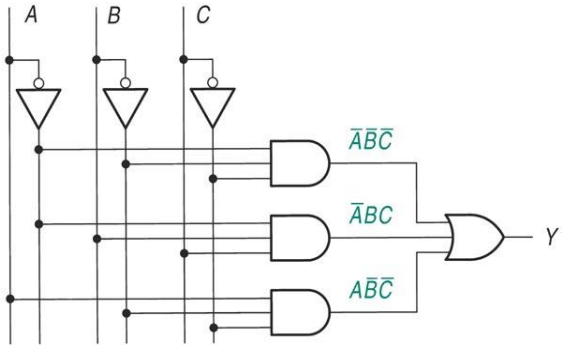
TABLE 3.4 Truth Table for Figure 3.16

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

(2)

$$Y = \bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,B\,C + A\,\bar{B}\,\bar{C}$$

(3)



(4)

Example 3.7 -1

Tables 3.5 and 3.6 show the truth tables for the Exclusive OR and the Exclusive NOR functions. Derive the sum-of-products expression for each of these functions and draw the logic diagram for each one.

TABLE 3.5 XOR Truth Table

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

TABLE 3.6 XNOR Truth Table

A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

■ **Solution**

Example 3.7 -2

XNOR: The product terms for this function are: $\bar{A}\bar{B}$ and AB. The SOP form of the XNOR function is $A \oplus B = \bar{A}\bar{B} + AB$. The logic diagram in Figure 3.19 represents the XNOR function.

- Plot the Logic Circuit:

Deriving a POS Expression from a Truth Table

- Each line of the truth table with a 0 (LOW) output represents a sum term.

  （真值表中，每一個輸出為Low的項就是一個和項）

- The sum terms are multiplied (ANDed).

  （將每一個和項乘起來）

Example 3.8-1 Deriving a SOP and POS Expression from a Truth Table

Find the Boolean expression, in both SOP and POS forms, for the logic function represented by Table 3.7. Draw the logic circuit for each form.

Table 3.7 Truth Table for Example 3.8 (with minterms and maxterms)

A	B	C	D	Y	Minterms	Maxterms
0	0	0	0	1	$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$	
0	0	0	1	1	$\overline{A}\,\overline{B}\,\overline{C}\,D$	
0	0	1	0	0		$A + B + \overline{C} + D$
0	0	1	1	1	$\overline{A}\,\overline{B}\,C\,D$	
0	1	0	0	0		$A + \overline{B} + C + D$
0	1	0	1	0		$A + \overline{B} + C + \overline{D}$
0	1	1	0	0		$A + \overline{B} + \overline{C} + D$
0	1	1	1	0		$A + \overline{B} + \overline{C} + \overline{D}$
1	0	0	0	1	$A\,\overline{B}\,\overline{C}\,\overline{D}$	
1	0	0	1	0		$\overline{A} + B + C + \overline{D}$
1	0	1	0	1	$A\,\overline{B}\,C\,\overline{D}$	
1	0	1	1	0		$\overline{A} + B + \overline{C} + \overline{D}$
1	1	0	0	1	$A\,B\,\overline{C}\,\overline{D}$	
1	1	0	1	1	$A\,B\,\overline{C}\,D$	
1	1	1	0	1	$A\,B\,C\,\overline{D}$	
1	1	1	1	0		$\overline{A} + \overline{B} + \overline{C} + \overline{D}$

Solution All minterms (for SOP form) and maxterms (for POS form) are shown in the last two columns of Table 3.5.

Boolean Expressions:

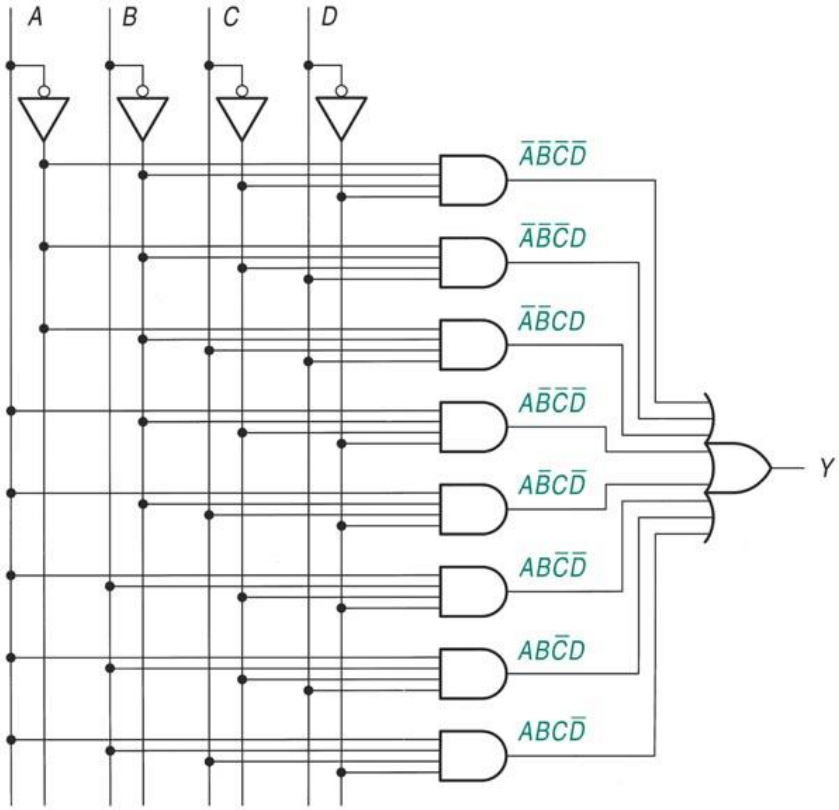
SOP form:

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

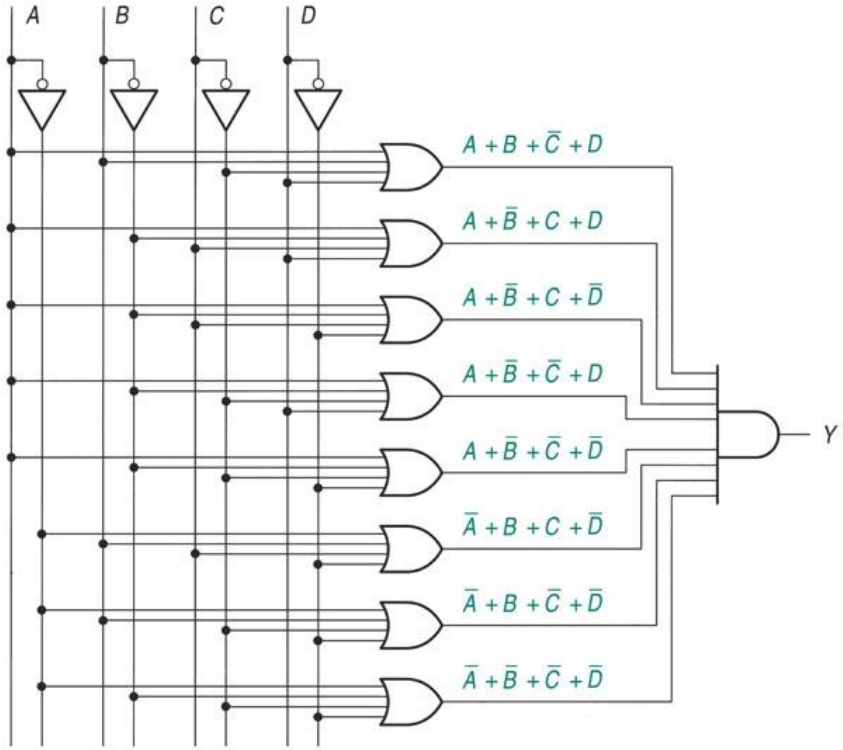
POS form:

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

Example 3.8-2



SOP



POS

3.3 Theorems of Boolean Algebra

- Used to minimize a Boolean expression to reduce the number of logic gates in a network.

- 24 theorems.

Commutative Property *(交換律)*

- The operation can be applied in any order with no effect on the result.
  - Theorem 1: $xy = yx$
  - Theorem 2: $x + y = y + x$

Associative Property *(結合律)*

- The operands can be grouped in any order with no effect on the result.
  - Theorem 3: $(xy)z = x(yz) = (xz)y$
  - Theorem 4: $(x + y) + z = x + (y + z) = (x + z) + y$

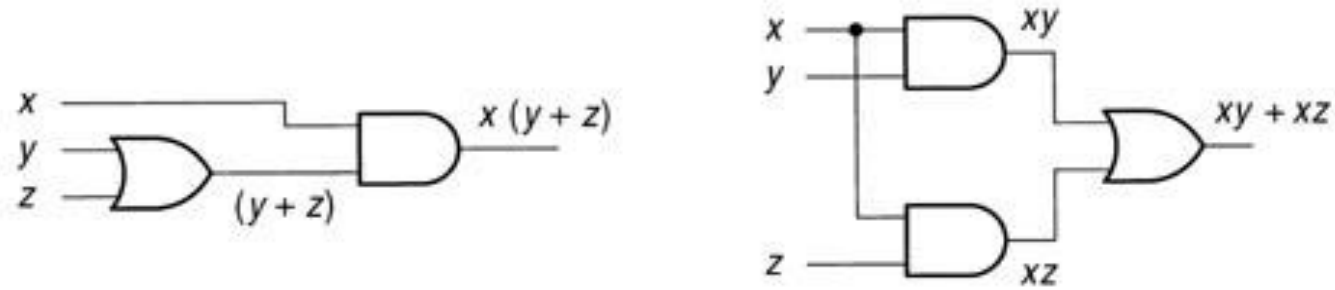
Distributive Property *(分配律)*

- Allows distribution (multiplying through) of AND across several OR functions.

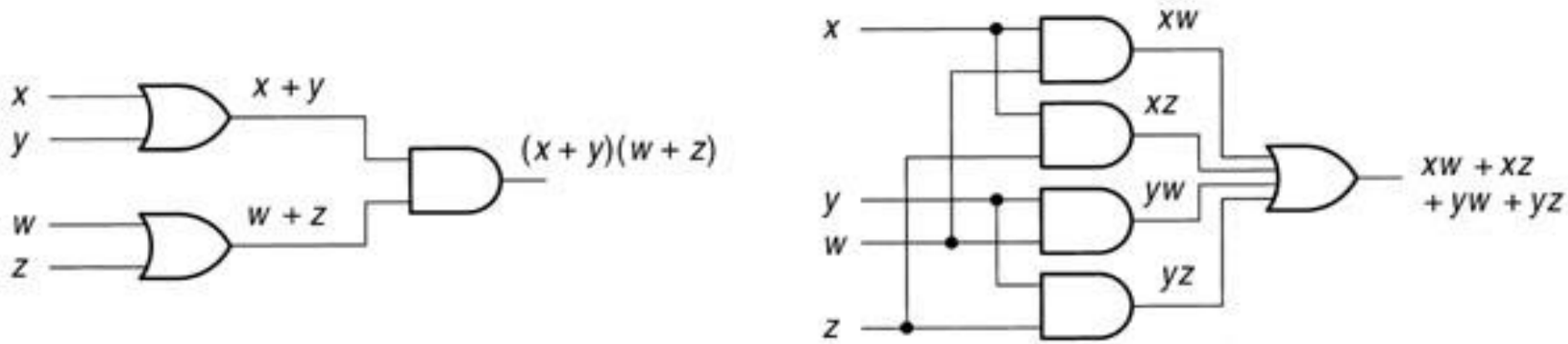
- Theorem 5: $x(y + z) = xy + xz$

- Theorem 6: $(x + y)(w + z) = xw + xz + yw + yz$

Distributive Property



a. $x(y + z) = xy + xz$ (Theorem 5)

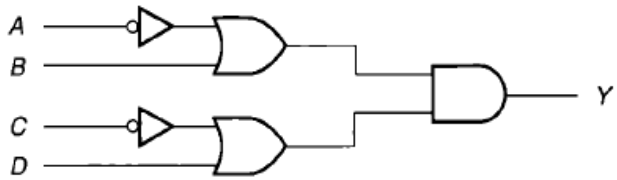


b. $(x + y)(w + z) = xw + xz + yw + yz$ (Theorem 6)

Example 3.9

Find the Boolean expression of the POS circuit in Figure 3.24a. Apply the distributive property to transform the circuit to an SOP form.

Sol:



a. POS form

FIGURE 3.24 Example 3.9: Distributive Property

x AND/OR/XOR 0/1
Operations with Logic 0 ⟵ (加法單位元素)

- Theorem 7: $x \bullet 0 = 0$

A	x	Y
0	0	0
0	1	0
1	0	0
1	1	1

- Theorem 8: $x + 0 = x$

A	x	Y
0	0	0
0	1	1
1	0	1
1	1	1

- Theorem 9: $x \otimes 0 = x$

A	x	Y
0	0	0
0	1	1
1	0	1
1	1	0

x AND/OR/XOR 0/1
Operations with Logic 1 ⟵ (乘法單位元素)

- Theorem 10: $x \bullet 1 = x$

A	x	Y
0	0	0
0	1	0
1	0	0
1	1	1

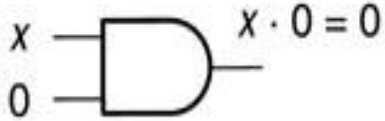
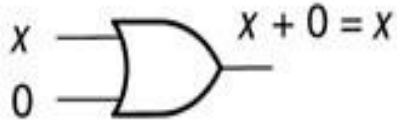
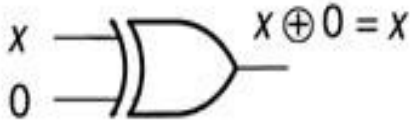
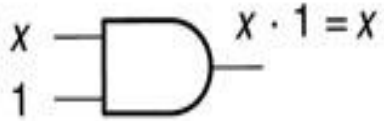
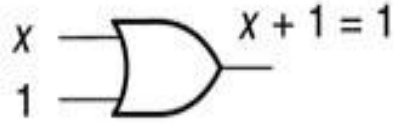
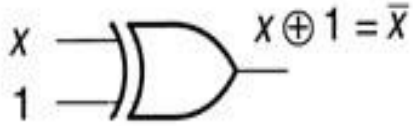
- Theorem 11: $x + 1 = 1$

A	x	Y
0	0	0
0	1	1
1	0	1
1	1	1

- Theorem 12: $x \otimes 1 = \overline{x}$

A	x	Y
0	0	0
0	1	1
1	0	1
1	1	0

x AND/OR/XOR 0/1 Summary

	AND	OR	XOR
0	 $x \cdot 0 = 0$	 $x + 0 = x$	 $x \oplus 0 = x$
1	 $x \cdot 1 = x$	 $x + 1 = 1$	 $x \oplus 1 = \overline{x}$

x AND/OR/XOR x / $\overline{x}$
Operations with its Complement（反元素）

- Theorem 13: $x \bullet x = x$

A	x	Y
0	0	0
0	1	0
1	0	0
1	1	1

- Theorem 14: $x + x = x$

A	x	Y
0	0	0
0	1	1
1	0	1
1	1	1

- Theorem 15: $x \otimes x = 0$

A	x	Y
0	0	0
0	1	1
1	0	1
1	1	0

x AND/OR/XOR $x / \overline{x}$
Operations with its Complement（反元素）

- Theorem 16 : $x \bullet \overline{x} = 0$

A	x	Y
0	0	0
0	1	0
1	0	0
1	1	1

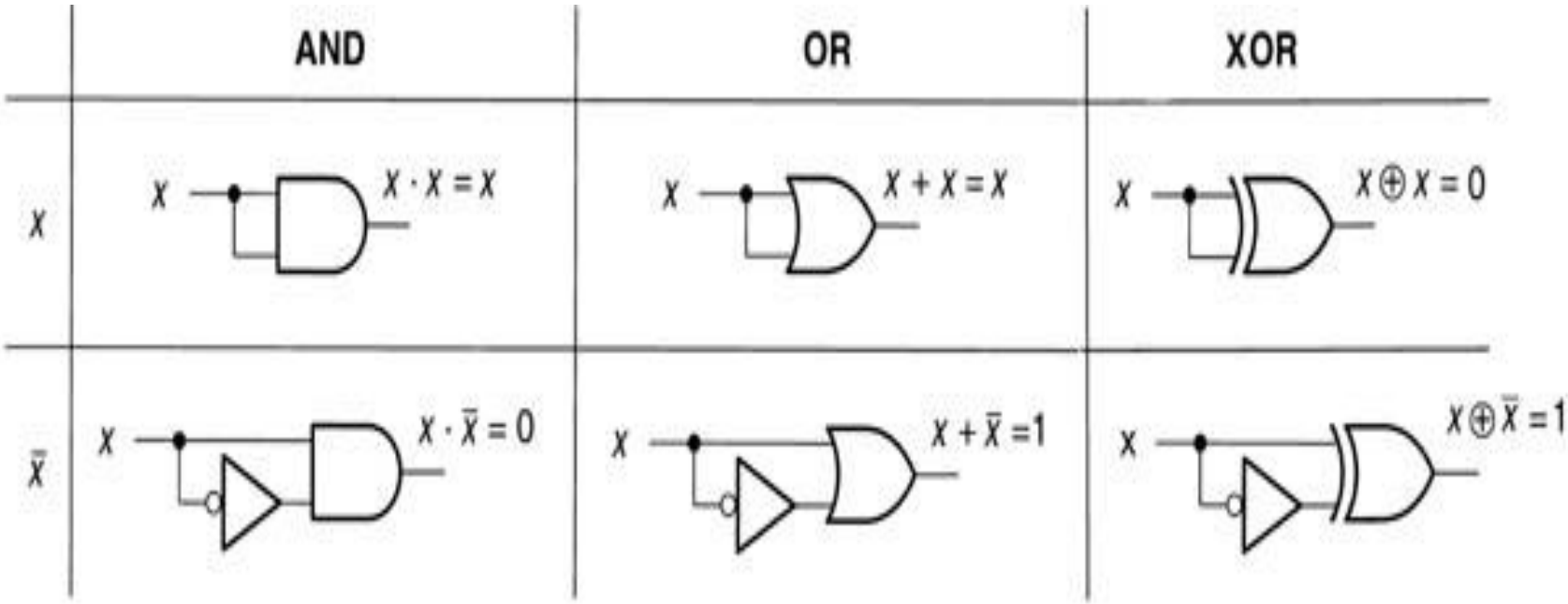
- Theorem 17 : $x + \overline{x} = 1$

A	x	Y
0	0	0
0	1	1
1	0	1
1	1	1

- Theorem 18 : $x \otimes \overline{x} = 1$

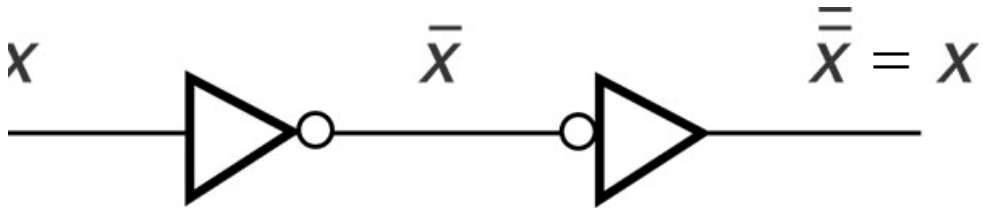
A	x	Y
0	0	0
0	1	1
1	0	1
1	1	0

x AND/OR/XOR $x / \bar{x}$
Operations with its Complement（反元素）
Summary



Double Inversion

- Theorem19: $\overline{\overline{x}} = x$



DeMorgan's Theorem

- Theorem 20: $\overline{x \bullet y} = \overline{x} + \overline{y}$
- Theorem 21: $\overline{x + y} = \overline{x} \bullet \overline{y}$

Multivariable Theorems -1

- Theorem 22: $x + xy = x$

Theorem 22 $x + xy = x$

Proof

分配律

$$
\begin{aligned}
x + xy &= x\,(1 + y) && \text{(Distributive property)} \\
&= x \cdot 1 && (1 + y = 1;\ \text{Theorem 11}) \\
&= x && (x \cdot 1 = x;\ \text{Theorem 10})
\end{aligned}
$$

Figure 3.29 illustrates the circuit in this theorem. Note that the equivalent is not a circuit at all, but a single, unmodified variable. Thus, the circuit shown need never be built.

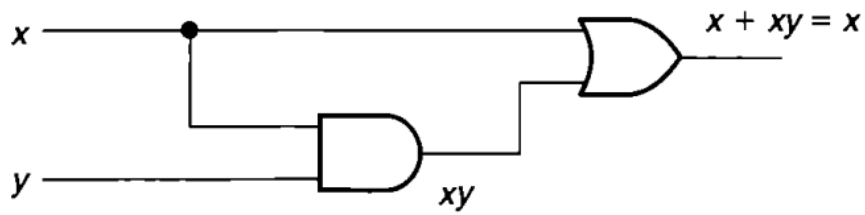


FIGURE 3.29 Theorem 22

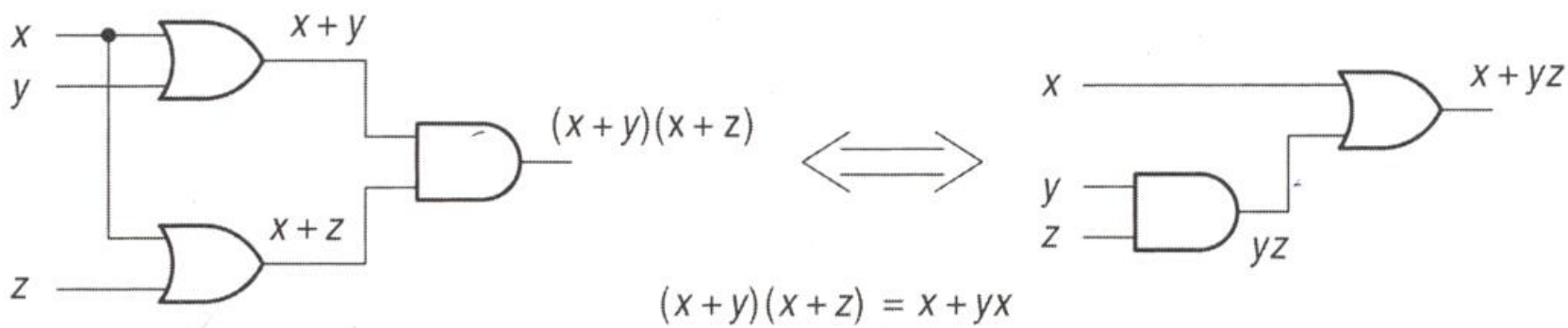
Multivariable Theorems -2

- Theorem 23: $(x + y)(x + z) = x + yz$

Theorem 23 $(x + y)(x + z) = x + yz$

Proof

$$
\begin{aligned}
(x + y)(x + z) &= xx + xz + xy + yz && \text{(distributive property)} \\
&= (x + xy) + xz + yz && (xx = x;\ \text{associative property}) \\
&= x + xz + yz && (x + xy = x\ \text{(Theorem 22)}) \\
&= (x + xz) + yz && \text{(associative property)} \\
&= x + yz && \text{(Theorem 22)}
\end{aligned}
$$



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

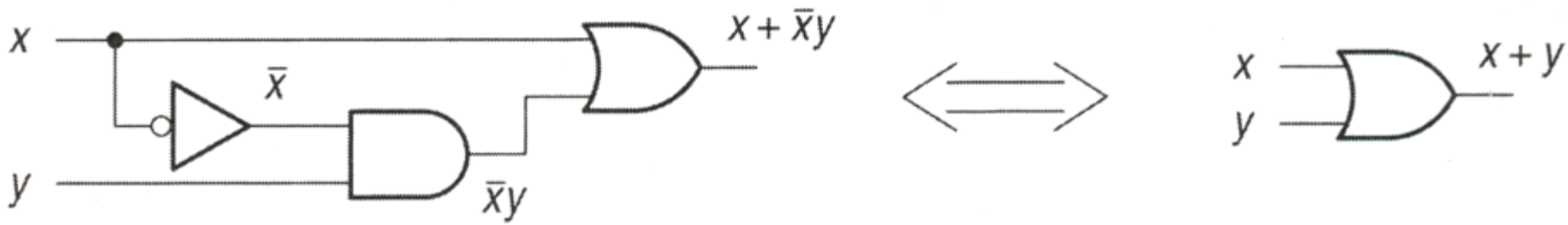
Multivariable Theorems -3

- Theorem 24: $x + \bar{x}y = x + y$

Theorem 24 $x + \bar{x}y = x + y$

Proof Since $(x + y)(x + z) = x + yz$, then for $y = \bar{x}$:

$$\begin{aligned} x + \bar{x}y &= (x + \bar{x})(x + y) \\ &= 1 \cdot (x + y) \qquad (x + \bar{x} = 1) \\ &= x + y \end{aligned}$$



3.4 Simplifying SOP and POS Expressions

- A Boolean expression can be simplified by:
  - Applying the Boolean Theorems to the expression
  - Application of graphical tool called a Karnaugh map (K-map卡諾圖) to the expression

■ KEY TERMS

Maximum SOP Simplification The form of an SOP Boolean expression that cannot be further simplified by canceling variables in the product terms. It may be possible to get a POS form of the expression with fewer terms or variables.

Maximum POS Simplification The form of a POS Boolean expression that cannot be further simplified by canceling variables in the sum terms. It may be possible to get an SOP form of the expression with fewer terms or variables.

Simplifying an Expression

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

factoring out the common term $A\overline{B}$

$$Y = A\overline{B}\,(1 + C)$$

applying theorem 11: $(1 + x = 1)$

$$Y = A\overline{B} \bullet 1$$

applying theorem 10 : $(x \bullet 1 = x)$

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

Simplifying an Expression

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

$$Y = \overline{A}\,\overline{\overline{B}} + \overline{(C+A)} \quad \textit{applying DeMorgan's theorem } 20$$

$$Y = \overline{A} + \overline{\overline{B}} + \overline{C}\,\overline{A} \quad \textit{applying DeMorgan's theorem } 20$$

$$Y = \overline{A}(1+\overline{C}) + B \quad \textit{factoring out } \overline{A}$$

$$Y = \overline{A} + B \quad \textit{applying theorem } 11: x+1=1$$

Maximum SOP/POS Simplification -1

Table 3.9 Truth Table for the SOP and POS Networks in Figure 3.35

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

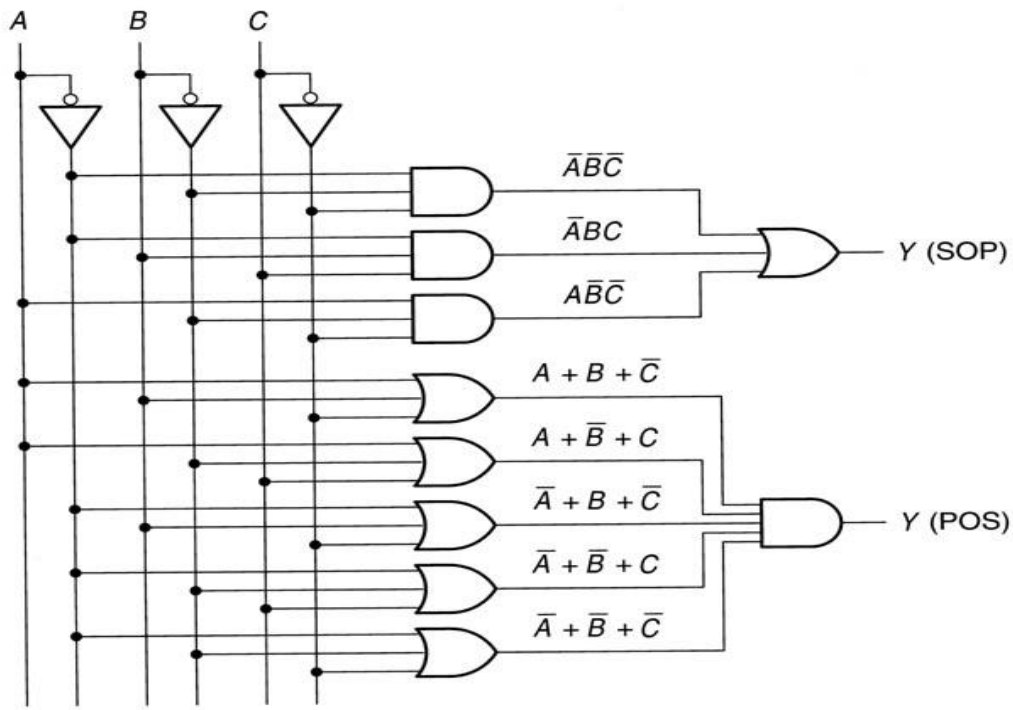


FIGURE 3.35
Unsimplified SOP and POS Networks

$$Y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,C + A\,\overline{B}\,\overline{C} \quad \text{(SOP)}$$

$$Y = (A + B + \overline{C})(A + \overline{B} + C)(\overline{A} + B + \overline{C})(\overline{A} + \overline{B} + C)(\overline{A} + \overline{B} + \overline{C}) \quad \text{(POS)}$$

Maximum SOP/POS Simplification -2

The SOP form is fairly easy to simplify:

$$
\begin{aligned}
Y &= \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,C + A\,\overline{B}\,\overline{C} \\
&= (\overline{A} + A)\,\overline{B}\,\overline{C} + \overline{A}\,B\,C && \text{(Distributive property)} \\
&= 1 \cdot \overline{B}\,\overline{C} + \overline{A}\,B\,C && (x + \overline{x} = 1) \\
&= \overline{B}\,\overline{C} + \overline{A}\,B\,C && (x \cdot 1 = x)
\end{aligned}
$$

Since we cannot cancel any more SOP terms, we can call this final form the **maximum SOP simplification.** The logic diagram for the simplified expression is shown in Figure 3.36.

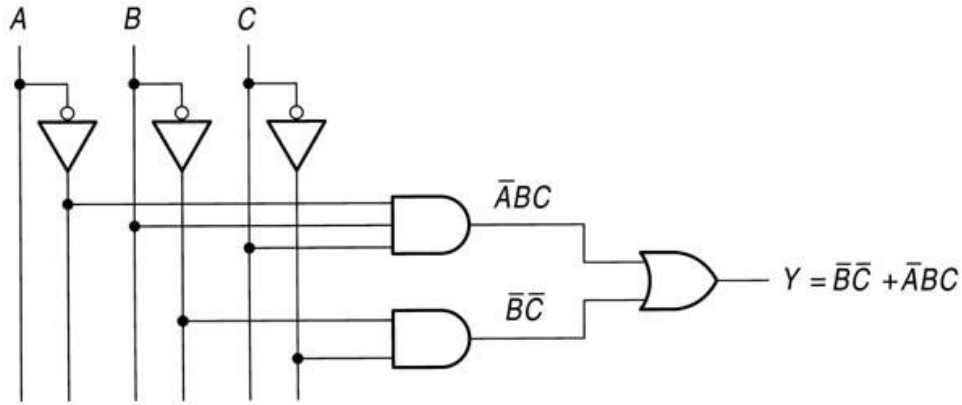


FIGURE 3.36
Simplified SOP Circuit

3.5 Simplification By Karnaugh Mapping

■ KEY TERMS

Karnaugh Map A graphical tool for finding the maximum SOP or POS simplification of a Boolean expression. A Karnaugh map (or K-map) works by arranging the terms of an expression in such a way that variables can be canceled by grouping minterms or maxterms.

Cell The smallest unit of a Karnaugh map, corresponding to one line of a truth table. The input variables are the cell's coordinates, and the output variable is the cell's contents.

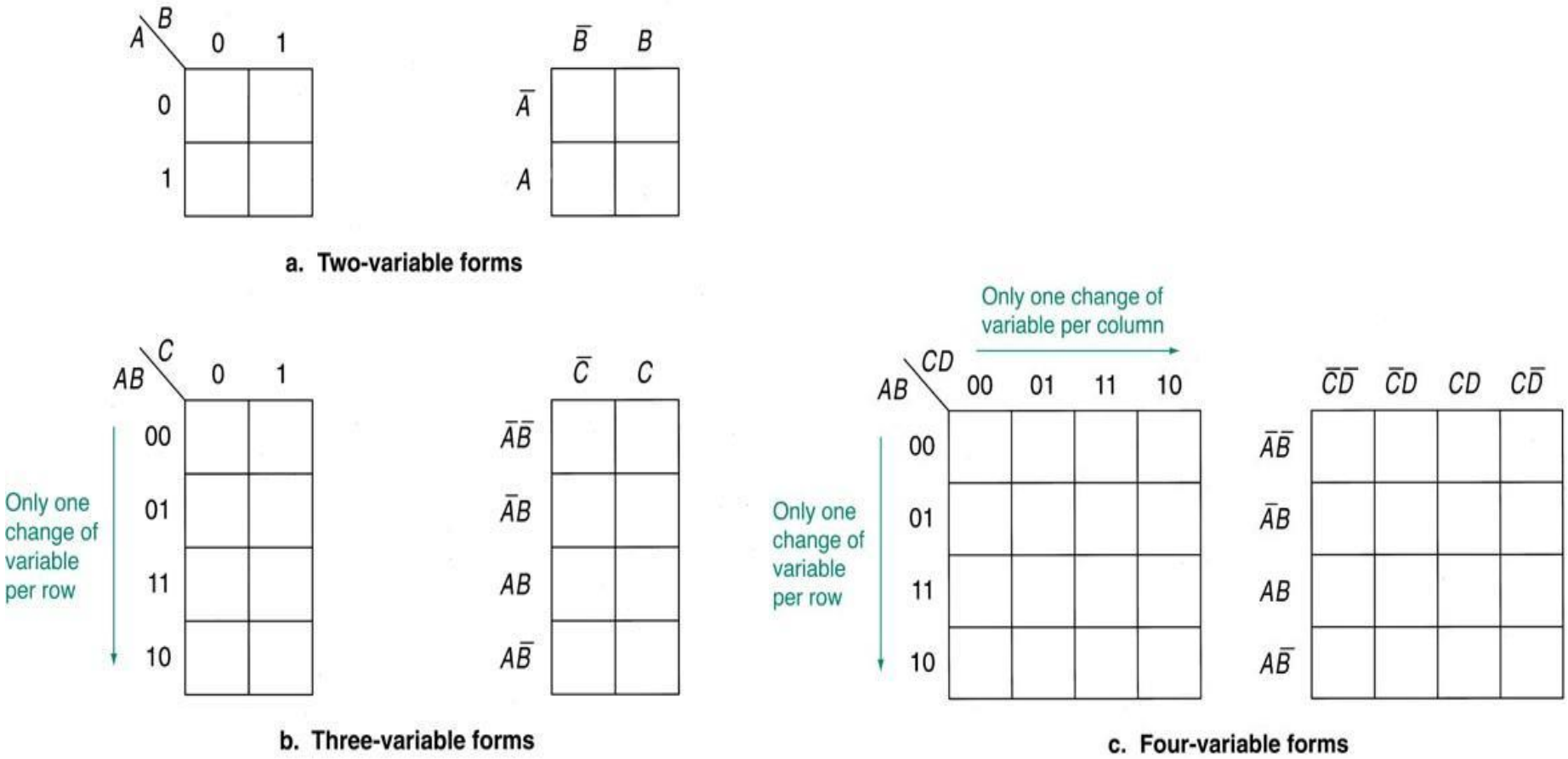
Adjacent Cell Two cells are adjacent if there is only one variable that is different between the coordinates of the two cells. For example, the cells for minterms ABC and $\bar{A}BC$ are adjacent.

Pair A group of two adjacent cells in a Karnaugh map. A pair cancels one variable in a K-map simplification.

Quad A group of four adjacent cells in a Karnaugh map. A quad cancels two variables in a K-map simplification.

Octet A group of eight adjacent cells in a Karnaugh map. An octet cancels three variables in a K-map simplification.

Construction of a Karnaugh Map



Grouping Cells

A \ B	0	1
0	1	1
1	0	0

FIGURE 3.41 Grouping a Pair of Adjacent Cells

AB \ C	0	1
00	0	1
01	0	1
11	0	1
10	0	1

FIGURE 3.42 Quad

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	0	0	0	0

FIGURE 3.43 Octet

Construction of a Karnaugh Map -1

The SOP expression of the truth table is

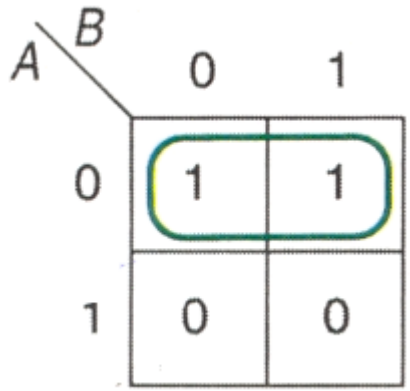
$$Y = \bar{A}\,\bar{B} + \bar{A}\,B$$

which can be simplified as follows:

$$\begin{aligned} Y &= \bar{A}\,(\bar{B} + B) \\ &= \bar{A} \end{aligned}$$

TABLE 3.13 Truth Table for a Two-Variable Boolean Expression

A	B	Y
0	0	1
0	1	1
1	0	0
1	1	0



$\bar{A}$ is a coordinate of both cells of the circled pair. (Keep $\bar{A}$.)

$\bar{B}$ is a coordinate of one cell of the circled pair, and B is a coordinate of the
(Discard $B/\bar{B}$.)

$$Y = \bar{A}$$

Construction of a Karnaugh Map -2

AB \ C	0	1
00	0	1
01	0	1
11	0	1
10	0	1

Quad

A group of four adjacent cells is called a **quad.** Figure 3.42 shows a Karnaugh map for a Boolean function whose terms can be grouped in a quad. The Boolean expression displayed in the K-map is:

$$Y = \bar{A}\,\bar{B}\,C + \bar{A}\,B\,C + A\,B\,C + A\,\bar{B}\,C = \bar{A}(\bar{B}C + BC) + A(BC + \bar{B}C)$$

$$= (\bar{B}C + BC)(\bar{A} + A)$$

$$= (\bar{B} + B)C \quad 1$$

$$= C.$$

A and *B* are both part of the quad coordinates in true and complement form. (Discard *A* and *B*.)

C is a coordinate of *each cell* in the quad. (Keep *C*.)

$$Y = C$$

Construction of a Karnaugh Map -3

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	0	0	0	0

Octet

Boolean expression:

$$Y = \bar{A}\,B\,\bar{C}\,\bar{D} + \bar{A}\,B\,\bar{C}\,D + \bar{A}\,B\,C\,D + \bar{A}\,B\,C\,\bar{D}$$
$$+ A\,B\,\bar{C}\,\bar{D} + A\,B\,\bar{C}\,D + A\,B\,C\,D + A\,B\,C\,\bar{D}$$

Variables A, C, and D are all coordinates of the octet cells in true and compleme form. (Discard A, C, and D.)

B is a coordinate of *each* cell. (Keep B.)

$$Y = B$$

Example 3.16
Grouping Cells along the Outside Edge

Use Karnaugh maps to simplify the following Boolean expressions:

a. $Y = \bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}\,C + A\,\bar{B}\,\bar{C} + A\,\bar{B}\,C$

b. $Y = \bar{A}\,\bar{B}\,\bar{C}\,\bar{D} + \bar{A}\,\bar{B}\,C\,\bar{D} + A\,\bar{B}\,\bar{C}\,\bar{D} + A\,\bar{B}\,C\,\bar{D}$

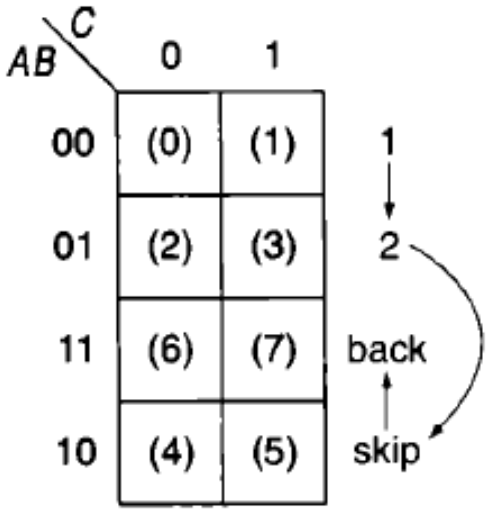
Sol:

Loading a K-Map from a Truth Table -1

- Each cell of the K-map represents one line from the truth table.
- The K-map is not laid out in the same order as the truth table.

A	B	C	Y
0	0	0	(0)
0	0	1	(1)
0	1	0	(2)
0	1	1	(3)
1	0	0	(4)
1	0	1	(5)
1	1	0	(6)
1	1	1	(7)

a. Truth table



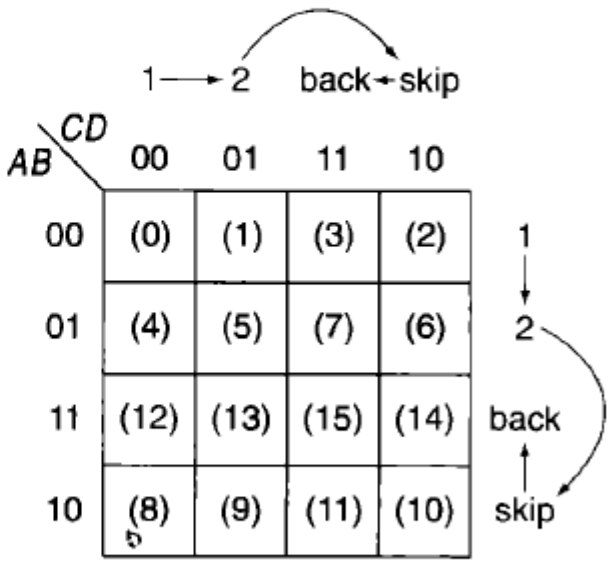
b. K-map

FIGURE 3.45 Order of Terms (Three-Variable Function)

Loading a K-Map from a Truth Table -2

A	B	C	D	Y
0	0	0	0	(0)
0	0	0	1	(1)
0	0	1	0	(2)
0	0	1	1	(3)
0	1	0	0	(4)
0	1	0	1	(5)
0	1	1	0	(6)
0	1	1	1	(7)
1	0	0	0	(8)
1	0	0	1	(9)
1	0	1	0	(10)
1	0	1	1	(11)
1	1	0	0	(12)
1	1	0	1	(13)
1	1	1	0	(14)
1	1	1	1	(15)

a. Truth table



b. K-map

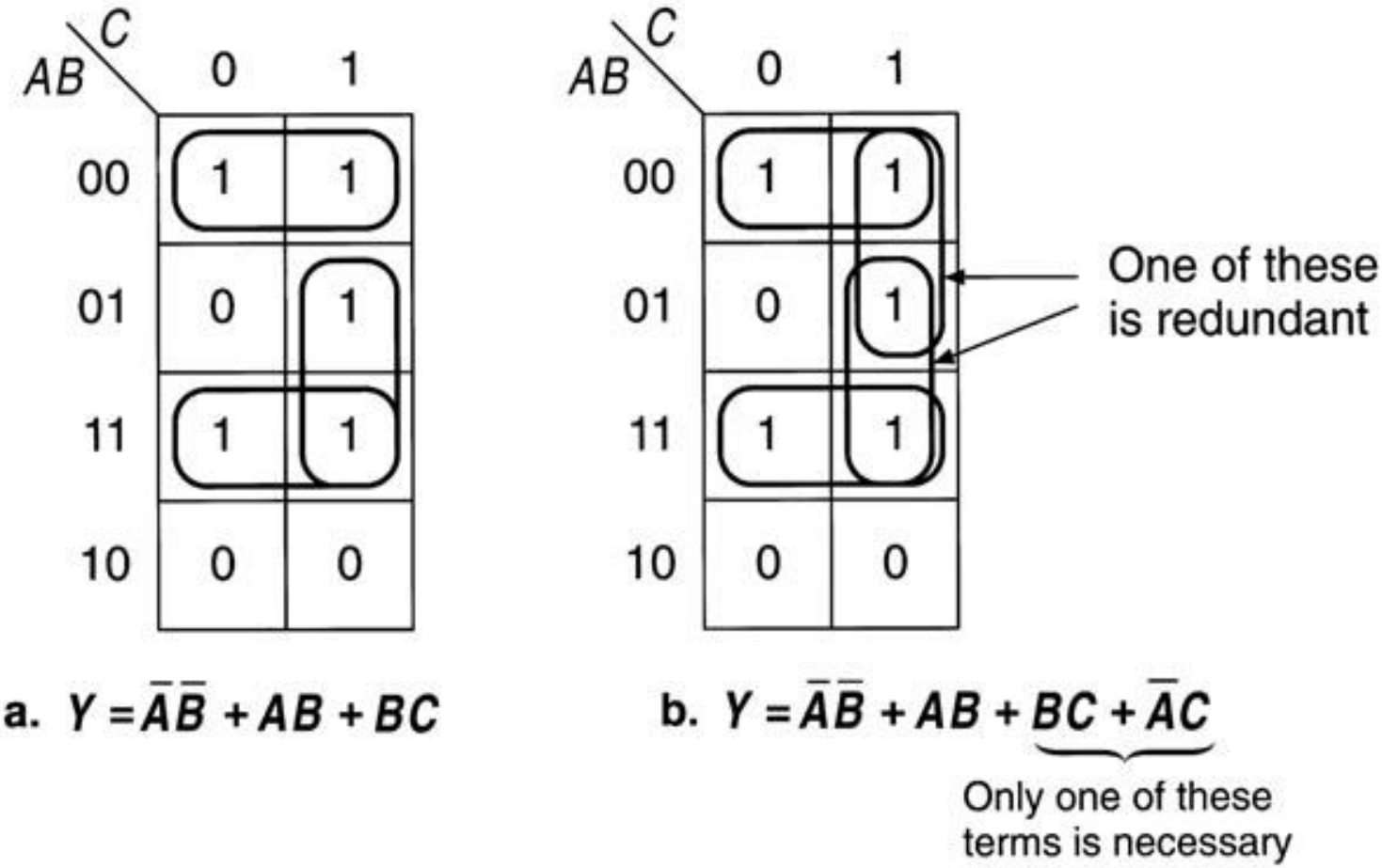
FIGURE 3.46 Order of Terms (Four-Variable Function)

Multiple Groups

- Each group is a term in the maximum SOP expression.

- A cell may be grouped more than once as long as every group has at least one cell that does not belong to any other group. Otherwise, redundant terms will result.

Multiple Groups



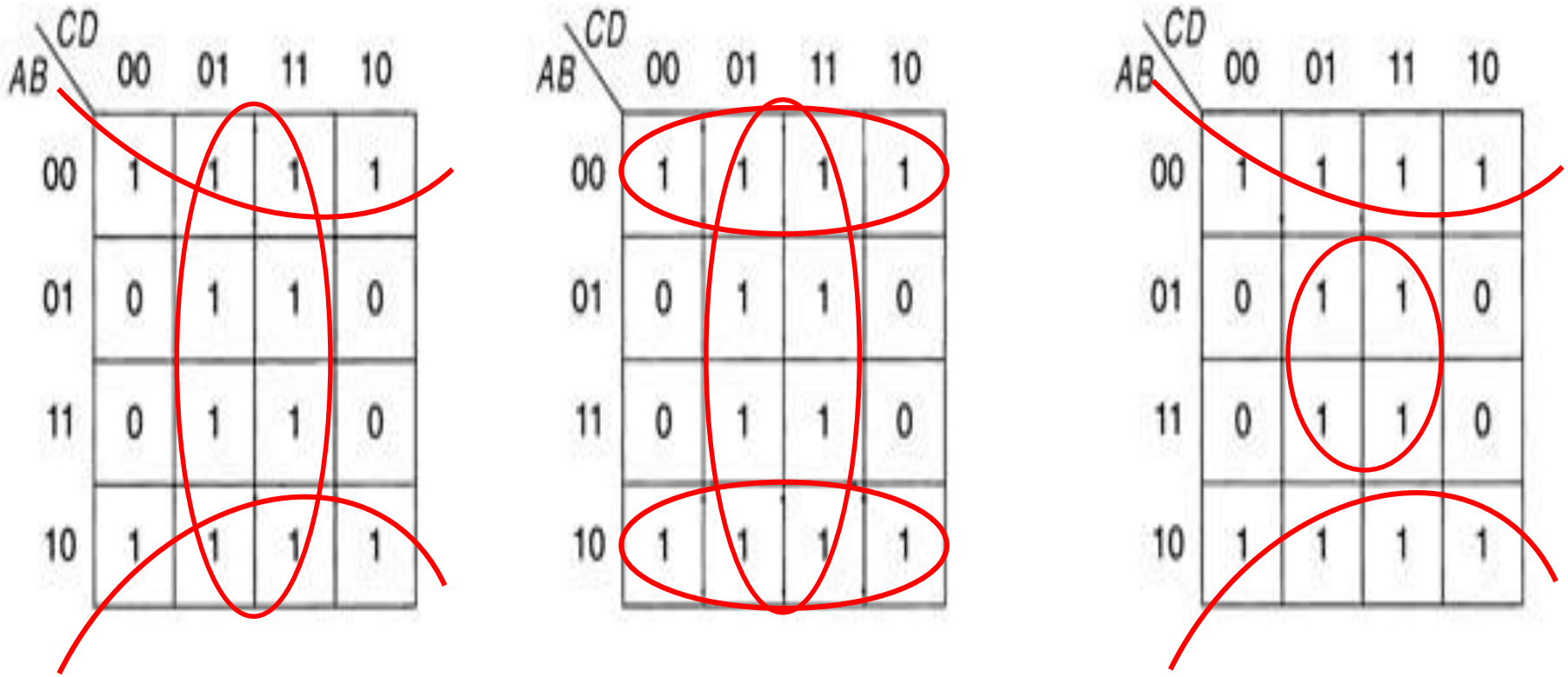
Maximum Simplification

- Achieved if the circled group of cells on the K-map are as large as possible.

- There are as few groups as possible.

圈圈越大越好, 圈圈越少越好

Maximum Simplification



a. Maximum simplification
$Y = \overline{B} + D$

b. Less than maximum simplification
$Y = \overline{A}\,\overline{B} + A\overline{B} + D$

c. Less than maximum simplification
$Y = \overline{B} + BD$

Using K-Maps for Partially Simplified Circuits -1

- Logic diagram can be further simplified

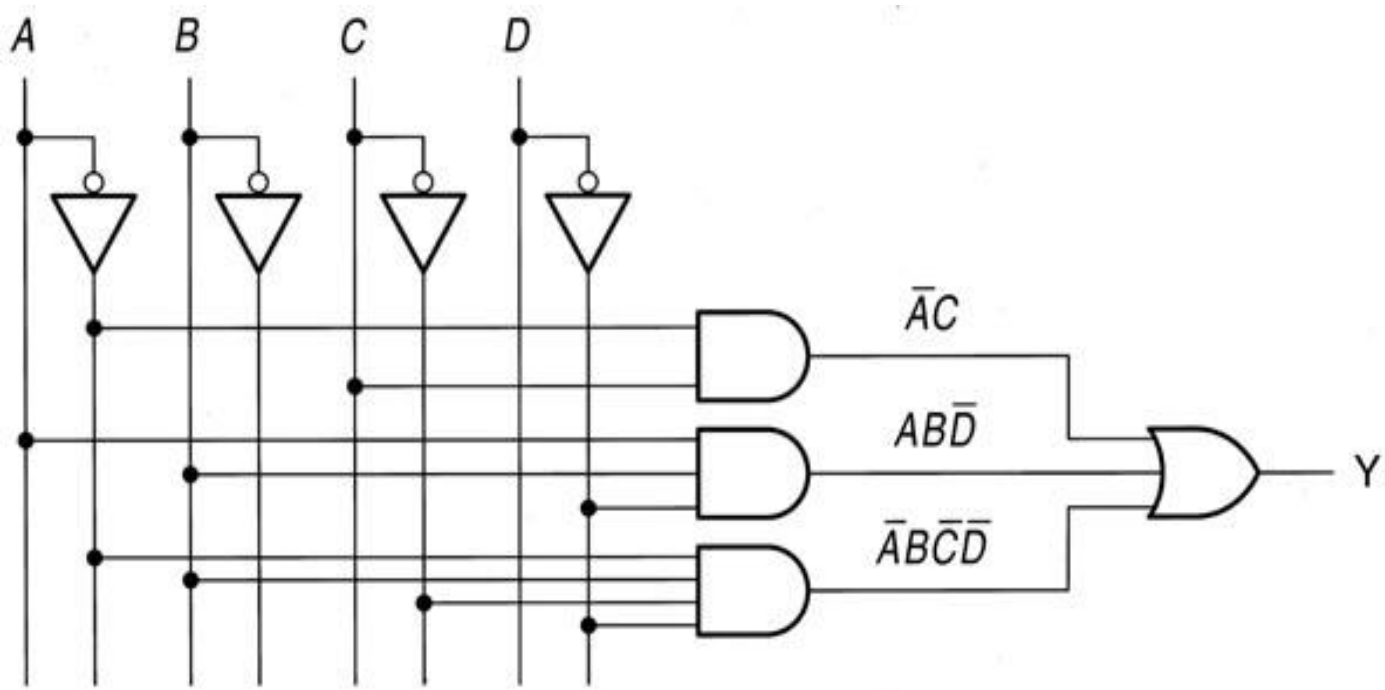
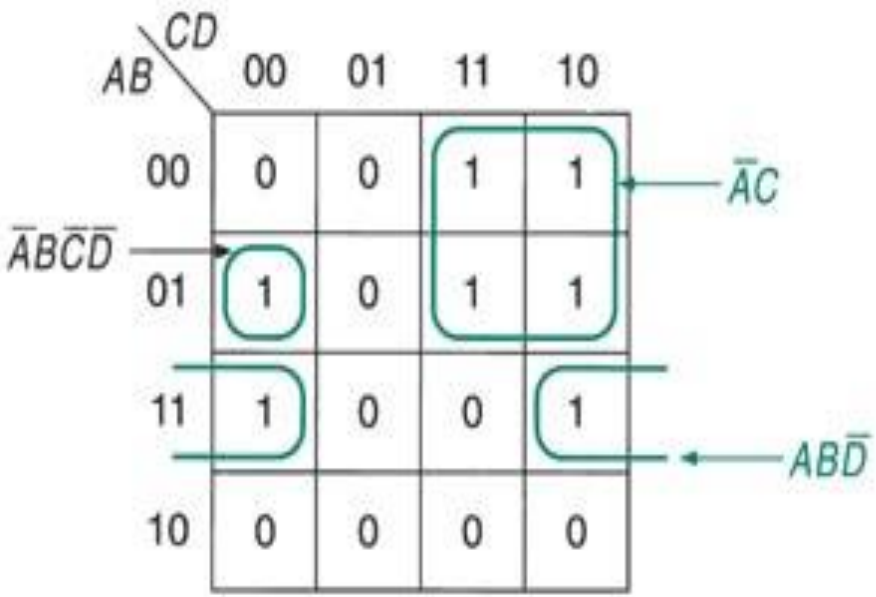
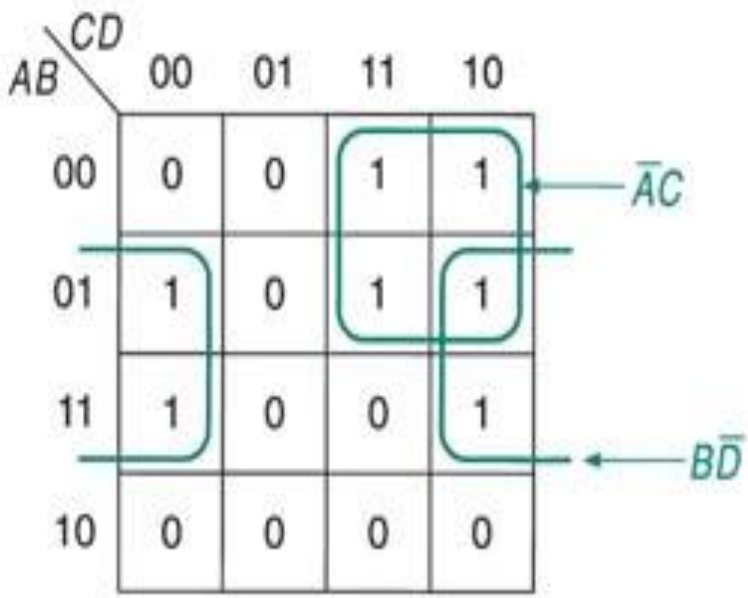


Fig. 3.50

Using K-Maps for Partially Simplified Circuits -2



a. K-map from logic diagram (Figure 3.50)

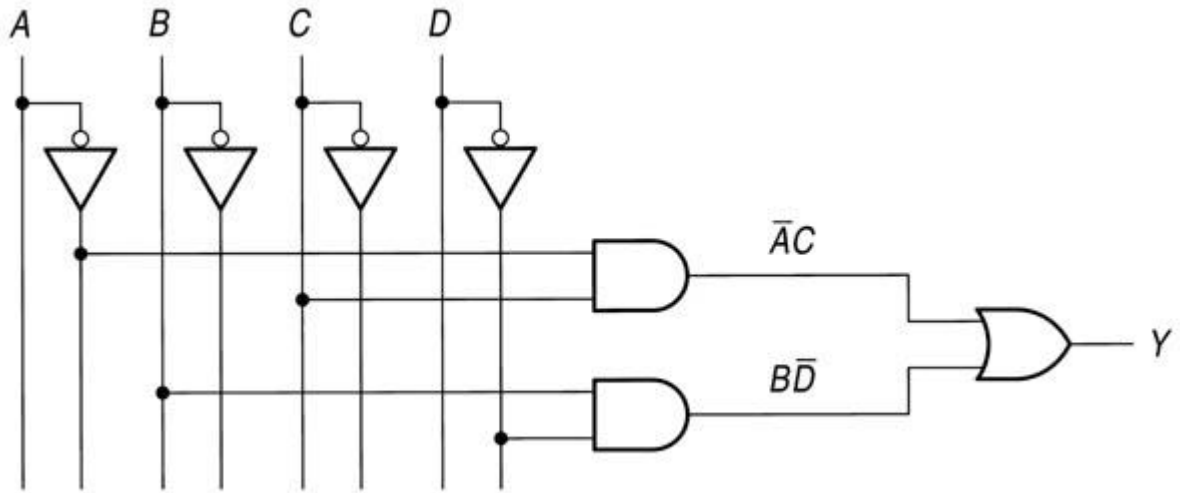


b. Maximum simplification

每一個Product項可能不只佔用一個Cell,
填完後重新Grouping化簡

Using K-Maps for Partially Simplified Circuits -3

- Simplified circuit



Don't Care States

- The output state of a circuit for a combination of inputs that will never occur.
  （不可能出現的輸出狀態）

- Shown in a K-map as an "x".

Value of Don't Care States

- In a K-map, set "x" to a 0 or a 1, depending on which case will yield the maximum simplification.

Example 3.21

Output is High: 5, 7, 9
Output is Low: not 5, 7, 9
Output is not defined: x

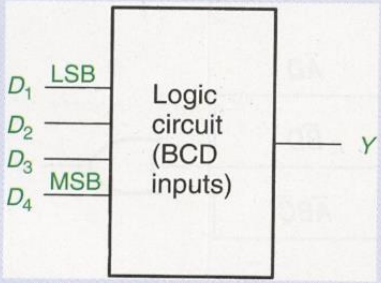


FIGURE 3.56 Example 3.21: Circuit to Be Simplified

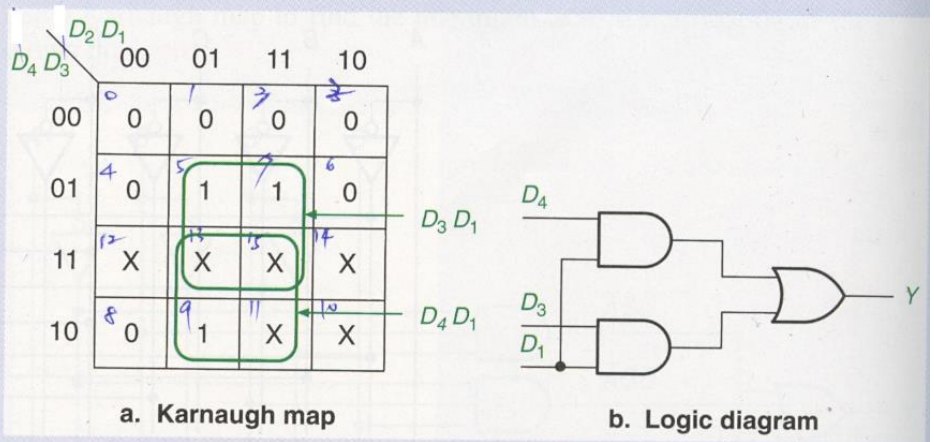


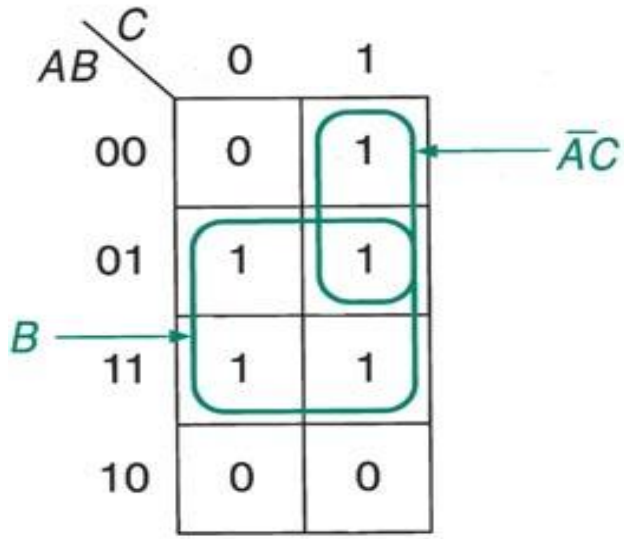
FIGURE 3.57 Example 3.21: Karnaugh Map and Logic Diagram

We can designate three of the don't care cells as 1s—those corresponding to input states 1011, 1101, and 1111. This allows us to group the 1s into two overlapping quads, which yield the following simplification.

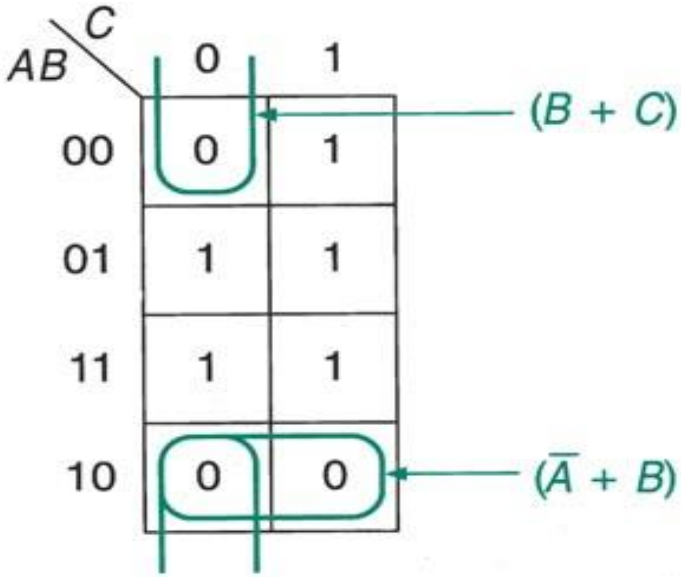
$$Y = D_4 D_1 + D_3 D_1$$

POS Simplification Using Karnaugh Mapping

- **Group those cells with values of 0**.
- Use the complements of the cell coordinates as the sum term.



a. SOP simplification
$Y = \bar{A}C + B$



b. POS simplification
$Y = (\bar{A} + B)(B + C)$

Example 3.23

Find the maximum POS simplification of the logic function represented by Table 3.18.

Sol:

TABLE 3.18 Truth Table for Example 3.23

A	*B*	*C*	*D*	*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	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

3.6 Simplification by DeMorgan Equivalent Gates Bubble-to-Bubble Convention –1

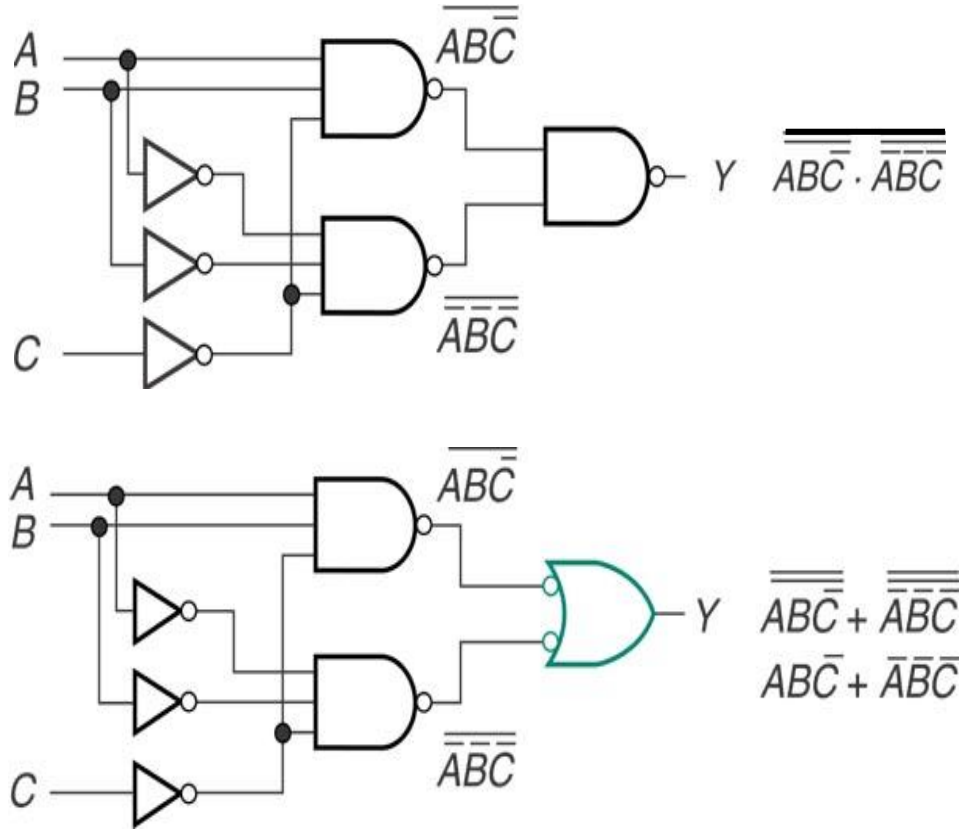
- Start at the output and work towards the input.
- Select the OR gate as the last gate for an SOP solution.
- Select the AND gate as the last gate for an POS solution.

- Choose the active level of the output if necessary.
- Go back to the circuits inputs to the next level of gating.

- Match the output of these gates to the input of the final gate. This may require converting the gate to its DeMorgan's equivalent.
- Repeat Step 2 until you reach the circuits.

Bubble-to-Bubble Convention – 2

- Simplification by DeMorgan Equivalent Gates



3.7 Universality (普遍性) of NAND/NOR Gates

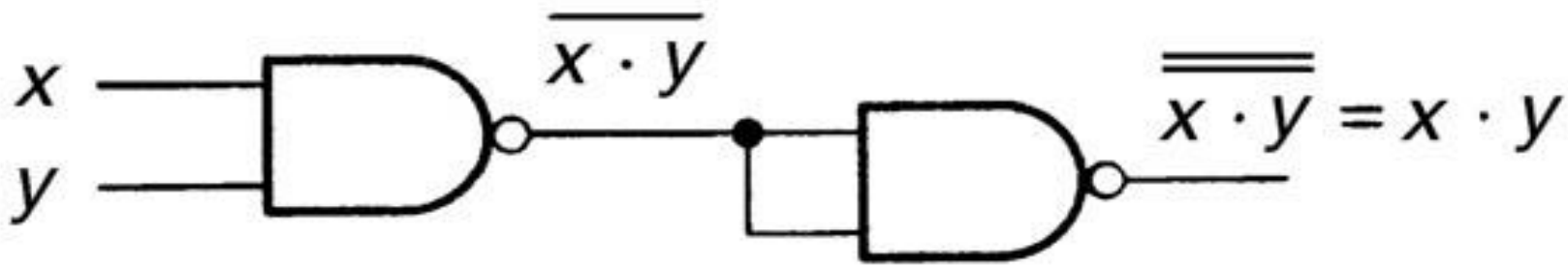
- Any logic gate can be implemented using only NAND or only NOR gates.

NOT from NAND

AND from NAND

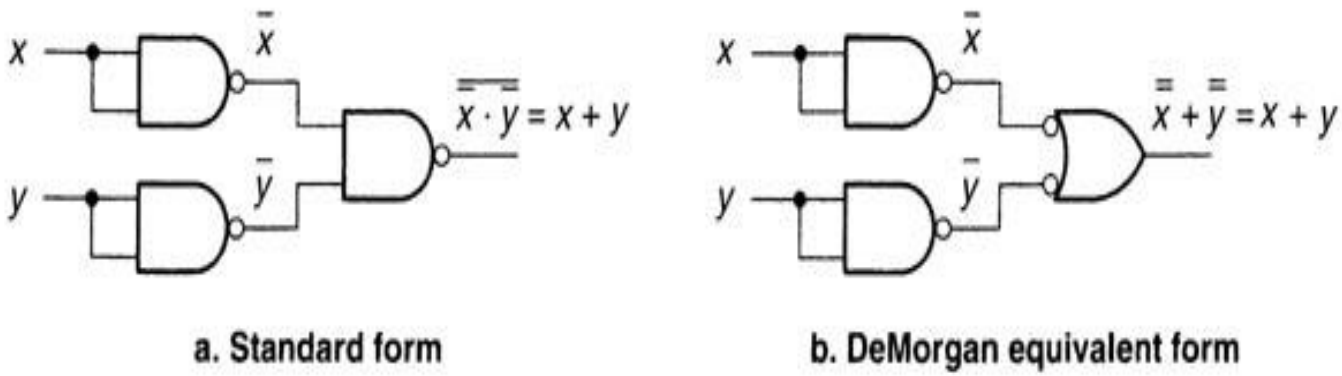
- The AND gate is created by inverting the output of the NAND gate.

$$Y = \overline{\overline{A \bullet B}} = A \bullet B$$



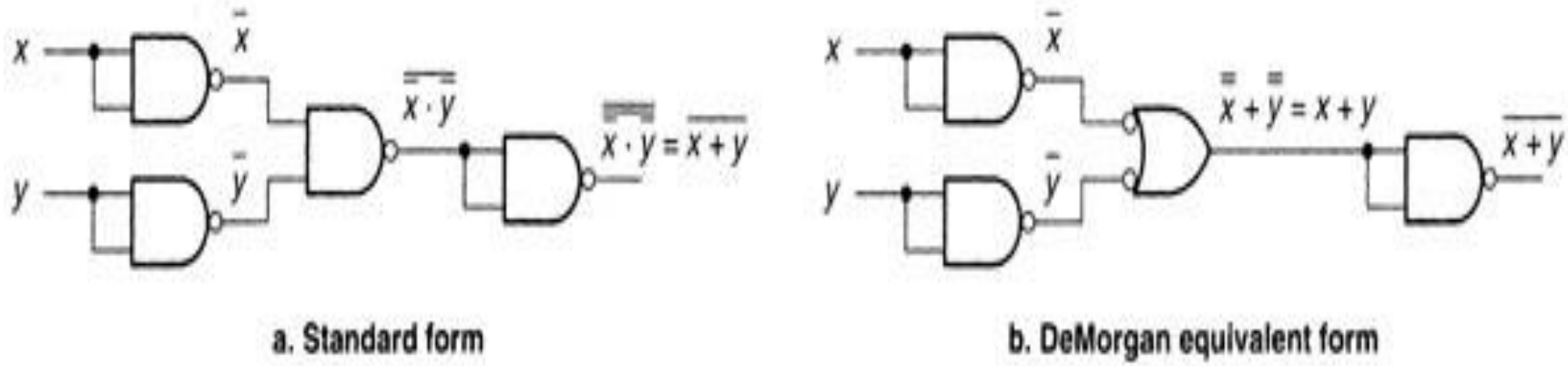
OR from NAND

$$\text{OR} = \overline{\overline{X} \bullet \overline{Y}} = X + Y$$



NOR from NAND

$$\text{NOR} = \overline{\overline{\overline{X} \bullet \overline{Y}}} = \overline{X + Y}$$

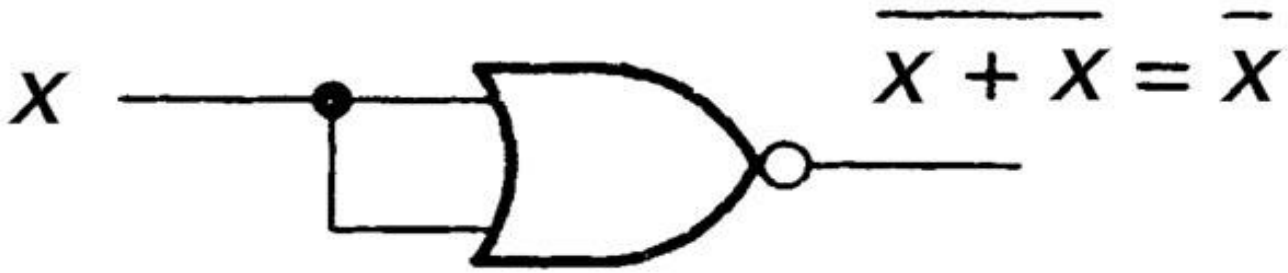


a. Standard form

b. DeMorgan equivalent form

NOT from NOR

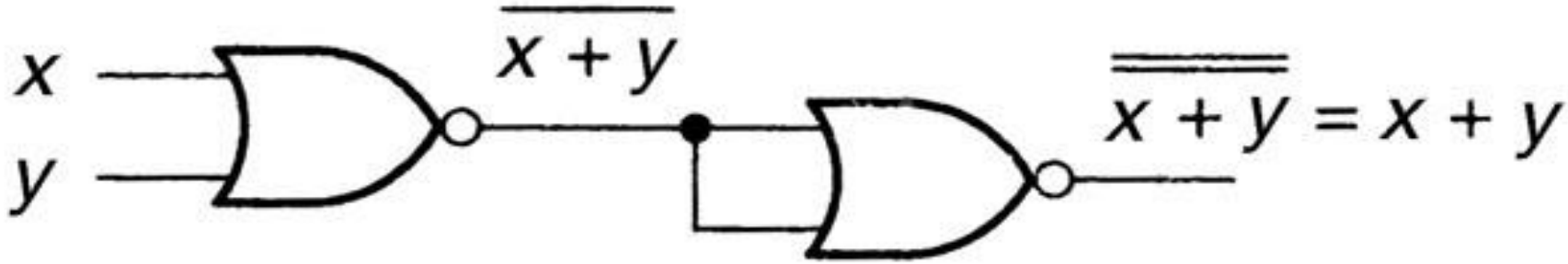
- An inverter can be constructed from a single NOR gate by connecting both inputs together.



OR from NOR

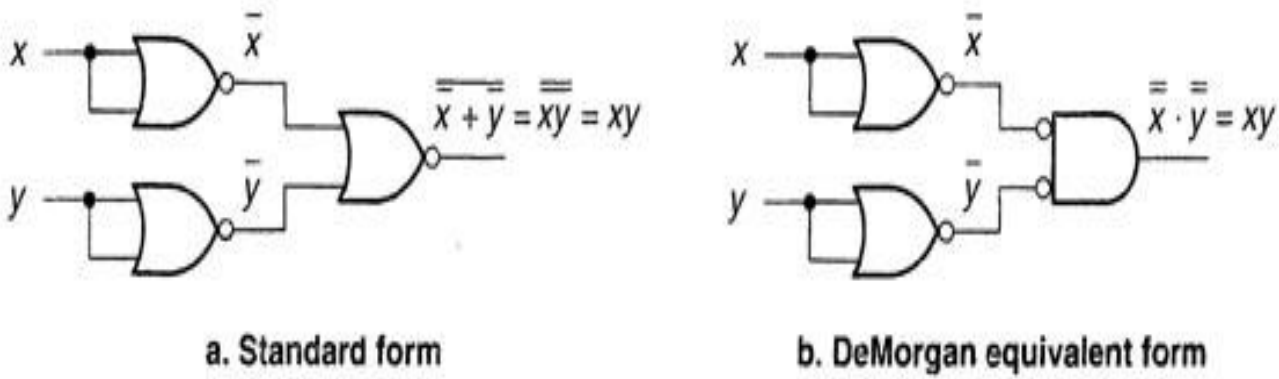
- The OR gate is created by inverting the output of the NOR gate.

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



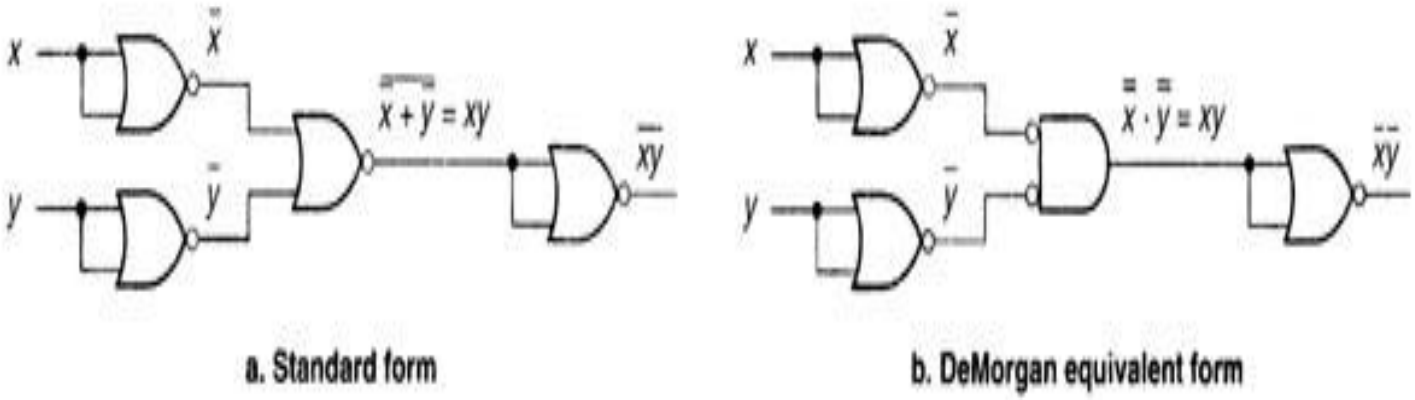
AND from NOR

$$\text{AND} = \overline{\overline{X} + \overline{Y}} = X \bullet Y$$



NAND from NOR

$$\text{NAND} = \overline{\overline{\overline{X} + \overline{Y}}} = \overline{X \bullet Y}$$



a. Standard form

b. DeMorgan equivalent form

3.8 Practical Circuit Implementation in Small-Scale Integration (SSI) Logic

Key Terms:

Integrated circuit (IC) An electronic circuit having many components, such as transistors, diodes, resistors, and capacitors, in a single package.

Small scale integration (SSI) An integrated circuit having 12 or fewer gates in one package.

Medium scale integration (MSI) An integrated circuit having the equivalent of 12 to 100 gates in one package.

Large scale integration (LSI) An integrated circuit having from 100 to 10,000 equivalent gates.

Very large scale integration (VLSI) An integrated circuit having more than 10,000 equivalent gates.

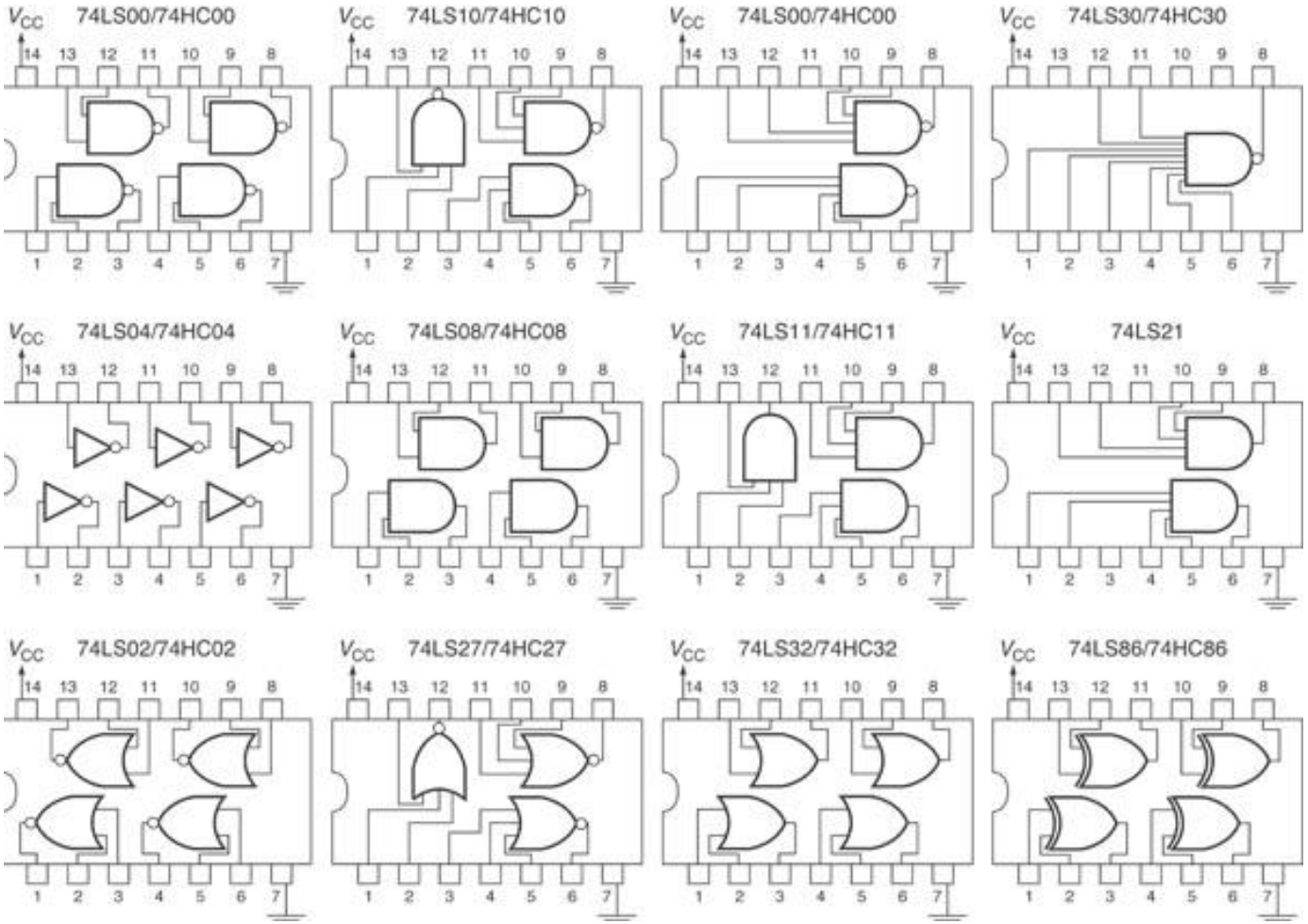
Transistor-transistor logic (TTL) A family of digital logic devices whose basic element is the bipolar junction transistor.

Complementary metal-oxide-semiconductor (CMOS) A family of digital logic devices whose basic element is the metal-oxide-semiconductor field effect transistor (MOSFET).

3.8 Practical Circuit Implementation in Small-Scale Integration (SSI) Logic

- Small Scale Integration (SSI): An integrated circuit having 12 or fewer gates in one package.
- Not all gates are available in TTL.
- TTL components are becoming more difficult to find.
- In a circuit design, it may be necessary to replace gates with other types of gates in order to achieve the final design.

Pinouts for TTL and High-Speed CMOS Gate



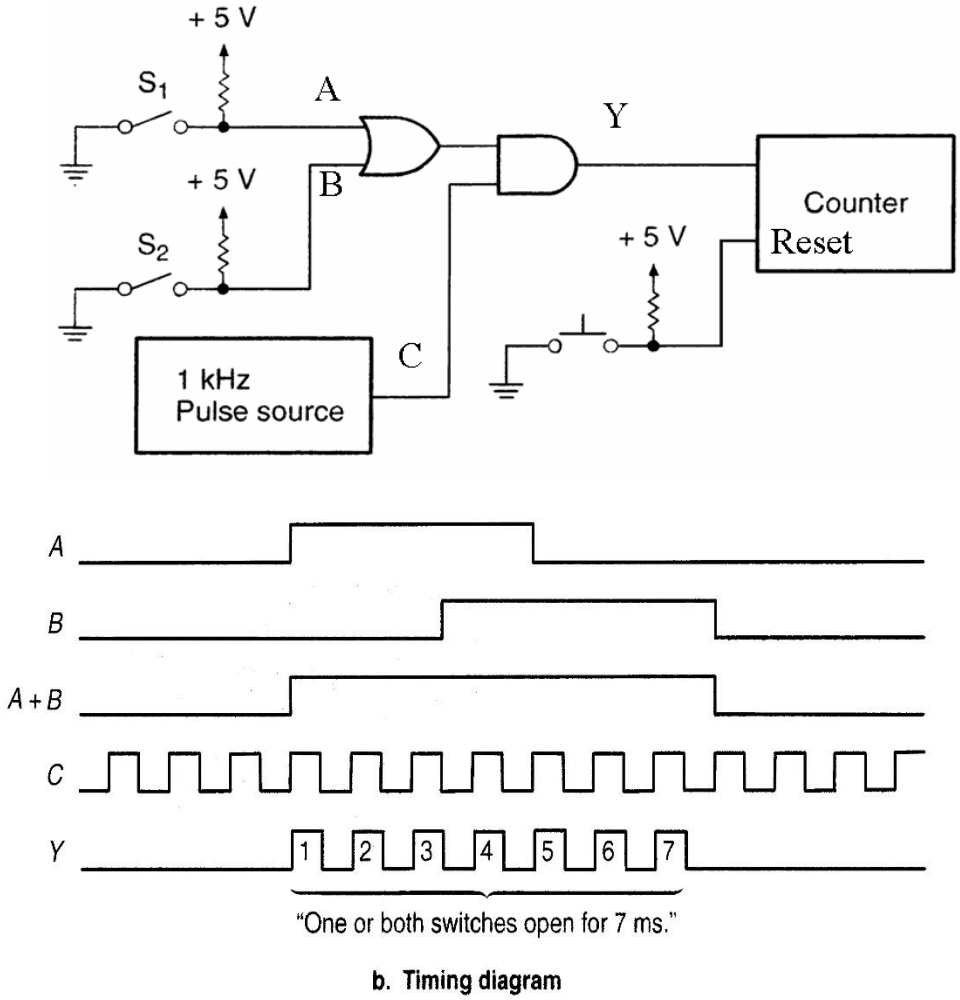
3.9 Pulsed Operation

- The enabling and inhibiting properties of the basic gates are used to pass or block pulsed digital signals.

- The pulsed signal is applied to one input.

- One input is used to control (enable/inhibit) the pulsed digital signal.

Pulsed Operation



b. Timing diagram

3.10 General Approach to Logic Circuit Design – 1

- Have an accurate description of the problem.
- Understand the effects of all inputs on all outputs.
- Make sure all combinations have been accounted for.

General Approach to Logic Circuit Design – 2

- Active levels as well as the constraints on all inputs and outputs should be specified.

- Each output of the circuit should be described either verbally or with a truth table.

General Approach to Logic Circuit Design – 3

- Look for keywords AND, OR, NOT that can be translated into a Boolean expression.

- Use Boolean algebra or K-maps to simplify expressions or truth tables.

Homework