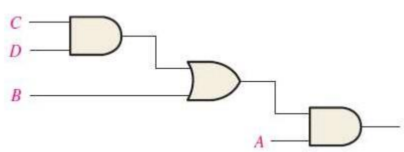
اسم المادة : تقنيات رقمية
اسم التدريسي :م.م علياء محمد جواد
المرحلة : الثانية
السنة الدراسية: 2023_2024
عنوان المحاضرة: Simplifying logic circuit

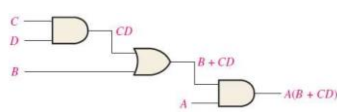
Boolean Expression for a Logic Circuit

- To derive the Boolean expression for a given combinational logic circuit, begin at the left-most inputs and work toward the final output, writing the expression for each gate.

- Example: Determine the Boolean expression for the following logic circuit



- Solution: $A(B + CD)$



Constructing a Truth Table for a Logic Circuit

- Once the Boolean expression for a given logic circuit has been determined, a truth table that shows the output for all possible values of the input variables can be developed.

Step1: Evaluating the Expression

- For example, to evaluate the expression $A(B + CD)$, find the values of the variables that make the expression equal to 1 using the rules for Boolean addition and multiplication.
- Thus, the expression $A(B + CD)$ equals 1 only if $A = 1$ and $B + CD = 1$.
- Now determine when the $B + CD$ term equals 1. The term $B + CD = 1$ if either $B = 1$ or $CD = 1$ or if both B and CD equal 1.
- The term $CD = 1$ only if $C = 1$ and $D = 1$.
- Therefore, the expression $A(B + CD) = 1$ when $A = 1$ and $B = 1$ regardless of the values of C and D or when $A = 1$ and $C = 1$ and $D = 1$ regardless of the value of B. And, the expression $A(B + CD) = 0$ for all other value combinations of the variables.

Constructing a Truth Table for a Logic Circuit

- **Step2: Putting the Results in Truth Table Format**
- First, list the sixteen input variable combinations of 1s and 0s in a binary sequence. Next, place a 1 in the output column for each combination of input variables that was determined in the evaluation. Finally, place a 0 in the output column for all other combinations of input variables. These results are shown in the truth table below

Inputs				Output
A	B	C	D	$A(B + CD)$
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	0
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	1
1	1	1	0	1
1	1	1	1	1

Logic Simplification Using Boolean Algebra

Example: Using Boolean algebra techniques, simplify this expression:

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

Solution:

Note: The following is not necessarily the only approach.

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

$= AB + AB + AC + B + BC$

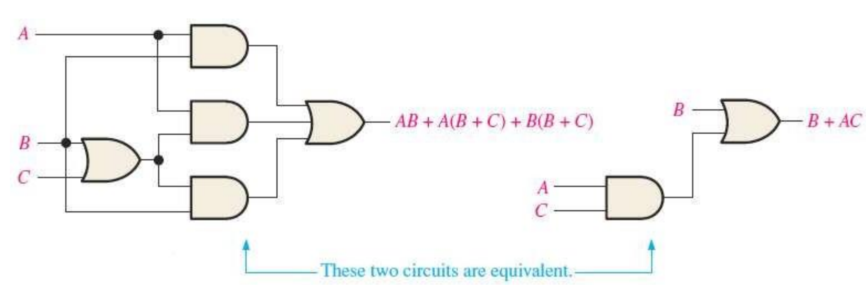
$= AB + AC + B + BC$

$= AB + AC + B$

$= B + AC$

Simplification reduces gates for the same function

- The figure below shows that five gates are required to implement the expression $AB + A(B + C) + B(B + C)$ in its original form; however, only two gates are needed for the simplified expression ($B + AC$).



Simplification reduces gates for the same function

Example: Simplify the following Boolean expression $[A\overline{B}(C + BD) + \overline{A}\,\overline{B}]C$

Solution:

$$
\begin{aligned}
[A\overline{B}(C + BD) + \overline{A}\,\overline{B}]C &= (A\overline{B}C + A\overline{B}BD + \overline{A}\,\overline{B})C \\
&= (A\overline{B}C + A \cdot 0 \cdot D + \overline{A}\,\overline{B})C \\
&= (A\overline{B}C + 0 + \overline{A}\,\overline{B})C \\
&= (A\overline{B}C + \overline{A}\,\overline{B})C \\
&= A\overline{B}CC + \overline{A}\,\overline{B}C \\
&= A\overline{B}C + \overline{A}\,\overline{B}C \\
&= \overline{B}C(A + \overline{A}) \\
&= \overline{B}C \cdot 1 = \overline{B}C
\end{aligned}
$$

Simplification reduces gates for the same function

Example: Simplify the following Boolean expression:

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

Solution:

$$
\begin{aligned}
\overline{A}BC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + ABC &= BC(\overline{A} + A) + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C \\
&= BC \cdot 1 + A\overline{B}(\overline{C} + C) + \overline{A}\,\overline{B}\,\overline{C} \\
&= BC + A\overline{B} \cdot 1 + \overline{A}\,\overline{B}\,\overline{C} \\
&= BC + A\overline{B} + \overline{A}\,\overline{B}\,\overline{C} \\
&= BC + \overline{B}(A + \overline{A}\,\overline{C}) \\
&= BC + \overline{B}(A + \overline{C}) \qquad \text{(using rule 11 } (A + \overline{A}\,\overline{C} = A + \overline{C})\text{)} \\
&= BC + A\overline{B} + \overline{B}\,\overline{C}
\end{aligned}
$$

Simplification reduces gates for the same function

Example: Simplify the following Boolean expression: $\overline{AB + AC} + \bar{A}\bar{B}C$

Solution:

$\overline{AB + AC} + \bar{A}\bar{B}C = (\overline{AB})(\overline{AC}) + \bar{A}\bar{B}C$ (By applying DeMorgan’s theorem)

$= (\bar{A} + \bar{B})(\bar{A} + \bar{C}) + \bar{A}\bar{B}C$ (By applying DeMorgan’s theorem to each term in the parentheses)

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

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

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

9

Standard Forms of Boolean Expressions

- All Boolean expressions can be converted into either of two standard forms: the sum-of-products form or the product-of-sums form.
- Standardization makes the evaluation, simplification, and implementation of Boolean expressions much more systematic and easier.

10

The Sum-of-Products (SOP) Form

- When two or more product terms are summed by Boolean addition, the resulting expression is a **sum-of-products (SOP)**. Some examples are:

$$AB + ABC$$
$$ABC + CDE + \overline{B}C\overline{D}$$
$$\overline{A}B + \overline{A}B\overline{C} + AC$$

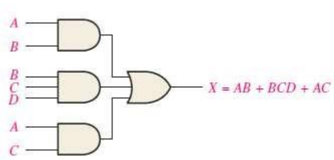
- Also, an SOP expression can contain a single-variable term, as in $A + \overline{A}\,\overline{B}C + BC\overline{D}$.

Note:

- In an SOP expression, a single overbar cannot extend over more than one variable; however, more than one variable in a term can have an overbar.
- For example, an SOP expression can have the term $\overline{A}\,\overline{B}\,\overline{C}$ but not $\overline{ABC}$.

AND/OR Implementation of an SOP Expression

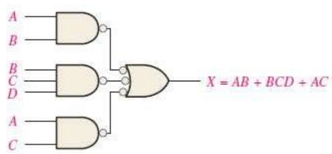
- Implementing an SOP expression simply requires ORing the outputs of two or more AND gates.
- A product term is produced by an AND operation, and the sum (addition) of two or more product terms is produced by an OR operation.
- Therefore, an SOP expression can be implemented by AND-OR logic in which the outputs of a number (equal to the number of product terms in the expression) of AND gates connect to the inputs of an OR gate, as shown in the figure below for the expression $AB + BCD + AC$



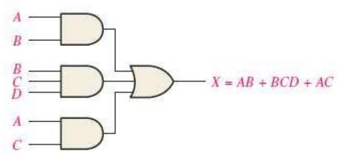
[اكتب نصاً]

NAND/NAND Implementation of an SOP Expression

- NAND gates can be used to implement an SOP expression.
- By using only NAND gates, an AND/OR function can be accomplished, as illustrated in figure (a) below to implement $AB + BCD + AC$.



(a)



(b)

- The first level of NAND gates feed into a NAND gate that acts as a negative-OR gate. The NAND and negative-OR inversions cancel and the result is effectively an AND/OR circuit.

13

Conversion of a General Expression to SOP Form

- Any logic expression can be changed into SOP form by applying Boolean algebra techniques.
- For example, the expression $A(B + CD)$ can be converted to SOP form by applying the distributive law: $A(B + CD) = AB + ACD$
- Example: Convert each of the following Boolean expressions to SOP form: (a) $AB + B(CD + EF)$ (b) $(A + B)(B + C + D)$ (c) $\overline{\overline{(A + B)} + C}$

Solution

(a) $AB + B(CD + EF) = AB + BCD + BEF$

(b) $(A + B)(B + C + D) = AB + AC + AD + BB + BC + BD$

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

14

The **Standard** SOP Form

- A *standard SOP expression* is one in which *all* the variables in the **domain** appear in each product term in the expression.
- For example, the expression $\overline{A}B\overline{C} + A\overline{B}D + \overline{A}B\overline{C}D$ is not in the standard SOP while $A\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + AB\overline{C}\,\overline{D}$ is in a standard SOP expression.

- Note:

$\overline{A}B\overline{C} + A\overline{B}D + \overline{A}B\overline{C}D$ has a **domain** made up of the variables *A*, *B*, *C*, and *D.* However, the complete set of variables in the domain is not represented in the first two terms of the expression; that is, *D* or $\overline{D}$ is missing from the first term and *C* or *C* is missing from the second term.

Converting Product Terms to Standard SOP

- Each product term in an SOP expression that does not contain all the variables in the domain can be expanded to standard form to include all variables in the domain and their complements.
- As stated in the following steps, a nonstandard SOP expression is converted into standard form using Boolean algebra rule 6 ($A + A = 1$).
- **Step 1:** Multiply each nonstandard product term by a term made up of the sum of a missing variable and its complement (you can multiply anything by 1 without changing its value).
- **Step 2:** Repeat Step 1 until all resulting product terms contain all variables in the domain in either complemented or uncomplemented form.
- In converting a product term to standard form, the number of product terms is doubled for each missing variable.

[اكتب نصاً]

Converting Product Terms to Standard SOP

- Example: Convert the following Boolean expression into standard SOP form: $A\overline{B}C + \overline{A}\,\overline{B} + AB\overline{C}D$
- Solution:

$$
\begin{aligned}
A\overline{B}C + \overline{A}\,\overline{B} + AB\overline{C}D &= A\overline{B}C(D + \overline{D}) + \overline{A}\,\overline{B}(C + \overline{C}) + AB\overline{C}D \\
&= A\overline{B}CD + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\,\overline{C} + AB\overline{C}D \\
&= A\overline{B}CD + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}C(D + \overline{D}) + \overline{A}\,\overline{B}\,\overline{C}(D + \overline{D}) + AB\overline{C}D \\
&= A\overline{B}CD + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + AB\overline{C}D
\end{aligned}
$$

The Product-of-Sums (POS) Form

- When two or more sum terms are multiplied, the resulting expression is a **product-of-sums (POS)**. Some examples are

$$
\begin{aligned}
&(\overline{A} + B)(A + \overline{B} + C) \\
&(\overline{A} + \overline{B} + \overline{C})(C + \overline{D} + E)(\overline{B} + C + D) \\
&(A + B)(A + \overline{B} + C)(\overline{A} + C)
\end{aligned}
$$

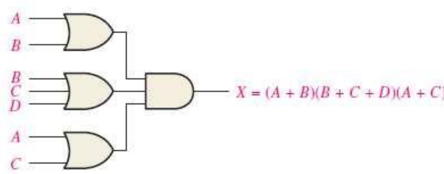
- A POS expression can contain a single-variable term, as in $\overline{A}(A + \overline{B} + C)(\overline{B} + \overline{C} + D)$

In a POS expression, a single overbar cannot extend over more than one variable; however, more than one variable in a term can have an overbar.

- For example, a POS expression can have the term $\overline{A} + \overline{B} + \overline{C}$ but not $\overline{A + B + C}$

Implementation of a POS Expression

- Implementing a POS expression simply requires ANDing the outputs of two or more OR gates. A sum term is produced by an OR operation, and the product of two or more sum terms is produced by an AND operation.
- For example, the implementation of the POS expression $(A + B)(B + C + D)(A + C)$ is shown in the figure below:



The Standard POS Form

- A *standard POS expression* is one in which *all* the variables in the domain appear in each sum term in the expression. For example, $(\overline{A} + \overline{B} + \overline{C} + \overline{D})(A + \overline{B} + C + D)(A + B + \overline{C} + D)$ is a standard POS expression.
- **Converting a Sum Term to Standard POS**
- A nonstandard POS expression is converted into standard form by using the following steps:
- Step1: Add to each nonstandard product term a term made up of the product of the missing variable and its complement $(A \cdot \overline{A} = 0)$.
- Step2: Apply rule 12: $A + BC = (A + B)(A + C)$
- Step3: Repeat Step 1 until all resulting sum terms contain all variables in the domain in either complemented or uncomplemented form.

Converting a Sum Term to Standard POS

- Example: Convert the following Boolean expression into standard POS form: $(A + \overline{B} + C)(\overline{B} + C + \overline{D})(A + \overline{B} + \overline{C} + D)$
- Solution

The domain of this POS expression is A, B, C, D. Take one term at a time. The first term, $A + \overline{B} + C$, is missing variable D or $\overline{D}$, so add $D\overline{D}$ and apply rule 12 as follows:

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

The second term, $\overline{B} + C + \overline{D}$, is missing variable A or $\overline{A}$, so add $A\overline{A}$ and apply rule 12 as follows:

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

The third term, $A + \overline{B} + \overline{C} + D$, is already in standard form. The standard POS form of the original expression is as follows:

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

Converting Standard SOP to Standard POS

- **Step 1:** Evaluate each product term in the SOP expression. That is, determine the binary numbers that represent the product terms.
- **Step 2:** Determine all of the binary numbers not included in the evaluation in Step 1.
- **Step 3:** Write the equivalent sum term for each binary number from Step 2 and express in POS form.

Note: Using a similar procedure, you can go from POS to SOP.

Converting Standard SOP to Standard POS

- Example: Convert the following SOP expression to an equivalent POS expression: $\overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + ABC$
- Solution:

The evaluation is as follows:

000 + 010 + 011 + 101 + 111

- Since there are three variables in the domain of this expression, there are a total of eight (2^3) possible combinations. The SOP expression contains five of these combinations, so the POS must contain the other three which are 001, 100, and 110.
- Remember, these are the binary values that make the sum term 0. The equivalent POS expression is

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

Boolean Expressions and Truth Tables

- A truth table is simply a list of the possible combinations of input variable values and the corresponding output values (1 or 0).
- The truth table is a common way of presenting the logical operation of a circuit.
- Standard SOP or POS expressions can be determined from a truth table.

Converting SOP Expressions to Truth Table Format

Step1: construct a truth table by listing all possible combinations of binary values of the variables in the expression.

Step2: convert the SOP expression to standard form if it is not already.

Step3: place a 1 in the output column (*X*) for each binary value that makes the standard SOP expression a 1 and place a 0 for all the remaining binary values.

- Note: an SOP expression is equal to 1 only if at least one of the product terms is equal to 1

Converting SOP Expressions to Truth Table Format

Example: Develop a truth table for the standard SOP expression

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

Solution:

Inputs A	B	C	Output X	Product Term
0	0	0	0	
0	0	1	1	$\overline{A}\,\overline{B}C$
0	1	0	0	
0	1	1	0	
1	0	0	1	$A\overline{B}\,\overline{C}$
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

Boolean Expressions and Truth Tables

Converting POS Expressions to Truth Table Format

Step1: list all the possible combinations of binary values of the variables just as was done for the SOP expression.

Step2: convert the POS expression to standard form if it is not already.

Step3: place a 0 in the output column (*X*) for each binary value that makes the expression a 0 and place a 1 for all the remaining binary values.

- Note: a POS expression is equal to 0 only if at least one of the sum terms is equal to 0.

Converting POS Expressions to Truth Table Format

- Example: Determine the truth table for the following standard POS expression: $(A + B + C)(A + \overline{B} + C)(A + \overline{B} + \overline{C})(\overline{A} + B + \overline{C})(\overline{A} + \overline{B} + C)$

- Solution

Inputs			Output	
A	B	C	X	Sum Term
0	0	0	0	$(A + B + C)$
0	0	1	1	
0	1	0	0	$(A + \overline{B} + C)$
0	1	1	0	$(A + \overline{B} + \overline{C})$
1	0	0	1	
1	0	1	0	$(\overline{A} + B + \overline{C})$
1	1	0	0	$(\overline{A} + \overline{B} + C)$
1	1	1	1	

Determining Standard Expressions from a Truth Table

Determining the standard SOP expression represented by a truth table

- Step1: list the binary values of the input variables for which the output is 1.
- Step2: Convert each binary value to the corresponding product term by replacing each 1 with the corresponding variable and each 0 with the corresponding variable complement. For example, the binary value 1010 is converted to a product term as follows:

$$1010 \longrightarrow A\overline{B}C\overline{D}$$

If you substitute, you can see that the product term is 1:

$$A\overline{B}C\overline{D} = 1 \cdot \overline{0} \cdot 1 \cdot \overline{0} = 1 \cdot 1 \cdot 1 \cdot 1 = 1$$

Determining Standard Expressions from a Truth Table

- Determining the standard POS expression represented by a truth table

Step1: list the binary values for which the output is 0.

Step2: Convert each binary value to the corresponding sum term by replacing each 1 with the corresponding variable complement and each 0 with the corresponding variable. For example, the binary value 1001 is converted to a sum term as follows:

$$1001 \longrightarrow \overline{A} + B + C + \overline{D}$$

If you substitute, you can see that the sum term is 0:

$$\overline{A} + B + C + \overline{D} = \overline{1} + 0 + 0 + \overline{1} = 0 + 0 + 0 + 0 = 0$$

Determining Standard Expressions from a Truth Table

- Example: From the truth table below, determine the standard SOP expression and the equivalent standard POS expression.

Inputs			Output
A	*B*	*C*	*X*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Solution:

There are four 1s in the output column and the corresponding binary values are 011, 100, 110, and 111. Convert these binary values to product terms as follows:

$$011 \longrightarrow \overline{A}BC$$
$$100 \longrightarrow A\overline{B}\,\overline{C}$$
$$110 \longrightarrow AB\overline{C}$$
$$111 \longrightarrow ABC$$

The resulting standard SOP expression for the output X is

$$X = \overline{A}BC + A\overline{B}\,\overline{C} + AB\overline{C} + ABC$$

For the POS expression, the output is 0 for binary values 000, 001, 010, and 101. Convert these binary values to sum terms as follows:

$$000 \longrightarrow A + B + C$$
$$001 \longrightarrow A + B + \overline{C}$$
$$010 \longrightarrow A + \overline{B} + C$$
$$101 \longrightarrow \overline{A} + B + \overline{C}$$

The resulting standard POS expression for the output X is

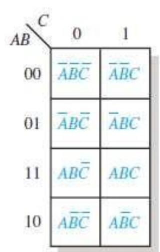
$$X = (A + B + C)(A + B + \overline{C})(A + \overline{B} + C)(\overline{A} + B + \overline{C})$$

Karnaugh map

- A Karnaugh map provides a systematic method for simplifying Boolean expressions and, if properly used, will produce the simplest SOP or POS expression possible, known as the minimum expression.
- A **Karnaugh map** is similar to a truth table because it presents all of the possible values of input variables and the resulting output for each value.
- The Karnaugh map is an array of **cells** in which each cell represents a binary value of the input variables.
- The cells are arranged in a way so that simplification of a given expression is simply a matter of properly grouping the cells.

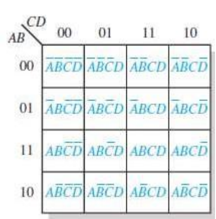
Karnaugh Map

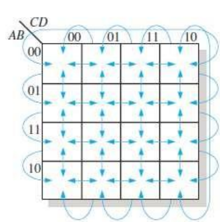
- The 3-variable Karnaugh map is an array of eight cells, as shown in the figure below:



010 cell is adjacent to the 000 cell, the 011 cell, and the 110 cell

- The 4-variable Karnaugh map is an array of sixteen cells, as shown in the figure





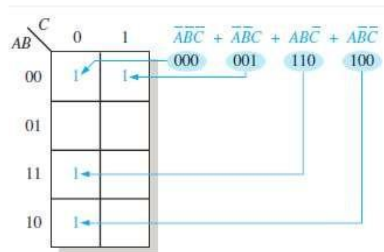
In “wrap-around” adjacency, the cells in the top row are adjacent to the corresponding cells in the bottom row and the cells in the outer left column are adjacent to the corresponding cells in the outer right column.

[اكتب نصاً]

Karnaugh Map SOP Minimization

Mapping a Standard SOP Expression

- For an SOP expression in standard form, a 1 is placed on the Karnaugh map for each product term in the expression.
- Each 1 is placed in a cell corresponding to the value of a product term. For example, for the product term *ABC*, a 1 goes in the 101 cell on a 3-variable map.
- When an SOP expression is completely mapped, there will be a number of 1s on the Karnaugh map equal to the number of product terms in the standard SOP expression as shown in the figure below.



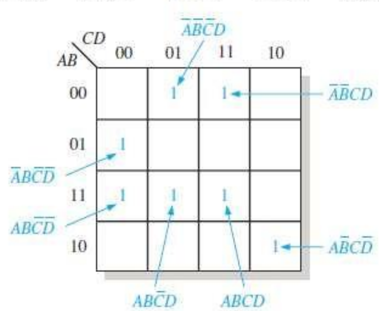
33

Karnaugh Map SOP Minimization

- Example: Map the following standard SOP expression on a Karnaugh map: $\bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + AB\bar{C}D + ABCD + AB\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + A\bar{B}C\bar{D}$
- Solution:

Evaluate the expression as shown below. Place a 1 on the 4-variable Karnaugh map in Figure 4–30 for each standard product term in the expression.

$$\begin{array}{ccccccccccccc} \bar{A}\bar{B}CD & + & \bar{A}B\bar{C}\bar{D} & + & AB\bar{C}D & + & ABCD & + & AB\bar{C}\bar{D} & + & \bar{A}\bar{B}\bar{C}D & + & A\bar{B}C\bar{D} \\ 0011 & & 0100 & & 1101 & & 1111 & & 1100 & & 0001 & & 1010 \end{array}$$



34

Mapping a Nonstandard SOP Expression

- Note: A Boolean expression must first be in standard form before you use a Karnaugh map. If an expression is not in standard form, then it must be converted to standard form.

- **Numerical Expansion of a Nonstandard Product Term**
- Assume that one of the product terms in a 3-variable expression is *B,* then it can be expanded numerically to standard form as follow:

Write the binary value of the variable; then attach all possible values for the missing variables *A* and *C* as follows:

B
010
011
110
111

35

Mapping a Nonstandard SOP Expression

- Example: Map the following SOP expression on a Karnaugh map:

$$\overline{B}\,\overline{C} + A\overline{B} + AB\overline{C} + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + A\overline{B}CD$$

Solution:

$\overline{B}\,\overline{C}$	$+ A\overline{B}$	$+ AB\overline{C}$	$+ A\overline{B}C\overline{D}$	$+ \overline{A}\,\overline{B}\,\overline{C}D$	$+ A\overline{B}CD$
0000	1000	1100	1010	0001	1011
0001	1001	1101			
1000	1010				
1001	1011				

- Notice that some of the values in the expanded expression are redundant.

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

36

Karnaugh Map Simplification of SOP Expressions

- The process that results in an expression containing the fewest possible terms with the fewest possible variables is called **minimization**.

- After an SOP expression has been mapped, a minimum SOP expression is obtained by grouping the 1s and determining the minimum SOP expression from the map.

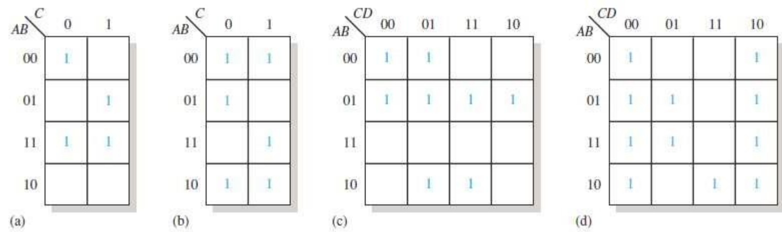
Karnaugh Map Simplification of SOP Expressions

- **Grouping the 1s**
- You can group 1s on the Karnaugh map according to the following rules by enclosing those adjacent cells containing 1s. The goal is to maximize the size of the groups and to minimize the number of groups.

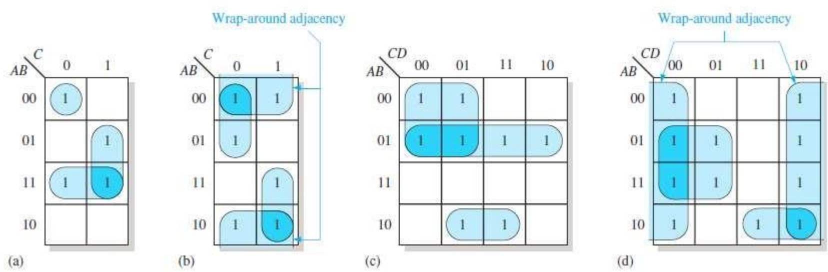
1. A group must contain either 1, 2, 4, 8, or 16 cells, which are all powers of two. In the case of a 3-variable map, $2^3 = 8$ cells is the maximum group.
2. Each cell in a group must be adjacent to one or more cells in that same group, but all cells in the group do not have to be adjacent to each other.
3. Always include the largest possible number of 1s in a group in accordance with rule 1.
4. Each 1 on the map must be included in at least one group. The 1s already in a group can be included in another group as long as the overlapping groups include noncommon 1s.

Grouping 1s on the Karnaugh map

- Example: Group the 1s in each of the Karnaugh maps



- Solution

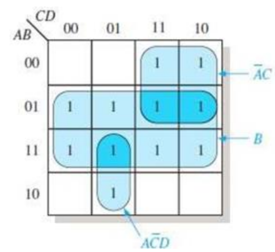


Determining the Minimum SOP Expression from the Map

1. Group the cells that have 1s. Each group of cells containing 1s creates one product term composed of all variables that occur in only one form (either uncomplemented or complemented) within the group. Variables that occur both uncomplemented and complemented within the group are eliminated. These are called *contradictory variables.*
2. Determine the minimum product term for each group.
3. When all the minimum product terms are derived from the Karnaugh map, they are summed to form the minimum SOP expression.

Determining the Minimum SOP Expression from the Map

Example: Determine the product terms for the Karnaugh map in the figure below and write the resulting minimum SOP expression.

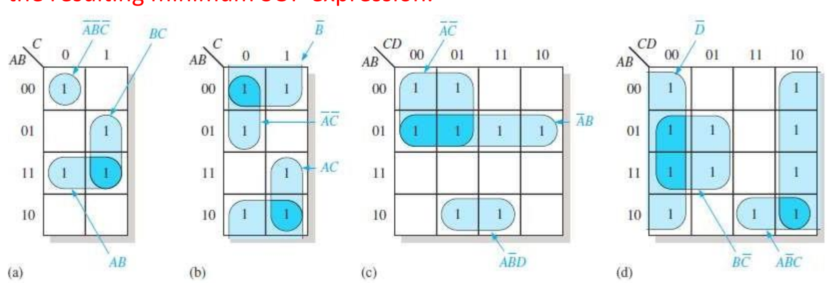


Solution:

- The resulting minimum SOP expression is the sum of these product terms: $B + \overline{A}C + A\overline{C}D$

Determining the Minimum SOP Expression from the Map

- Example: Determine the product terms for each of the Karnaugh maps in the figures below and write the resulting minimum SOP expression.



- Solution: The minimum SOP expressions for each of the Karnaugh maps in the figures are:

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

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

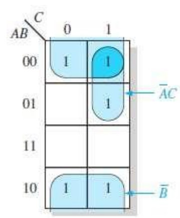
(c) $\overline{A}B + \overline{A}\,\overline{C} + A\overline{B}D$

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

Karnaugh map

- Example: Use a Karnaugh map to minimize the following standard SOP expression: $A\overline{B}C + \overline{A}BC + \overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}\,\overline{C}$

- **Solution:**
- The binary values of the expression are 101 + 011 + 001 + 000 + 100
- Map the standard SOP expression and group the cells as shown in the figure below:

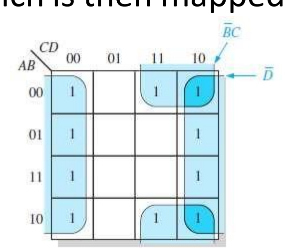


- Then, the resulting minimum SOP expression is $\overline{B} + \overline{A}C$

Karnaugh map

- Example: Use a Karnaugh map to minimize the following SOP expression: $\overline{B}\,\overline{C}\,\overline{D} + \overline{A}B\overline{C}\,\overline{D} + AB\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + A\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}BC\overline{D} + ABC\overline{D} + A\overline{B}C\overline{D}$

- Solution:
- The first term $\overline{B}\,\overline{C}\,\overline{D}$ must be expanded into $A\overline{B}\,\overline{C}\,\overline{D}$ and $\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$ to get the standard SOP expression, which is then mapped; the cells are grouped as shown below:



- The resulting minimum SOP expression is $\overline{D} + \overline{B}C$