Logic Design Laboratory EE 2121

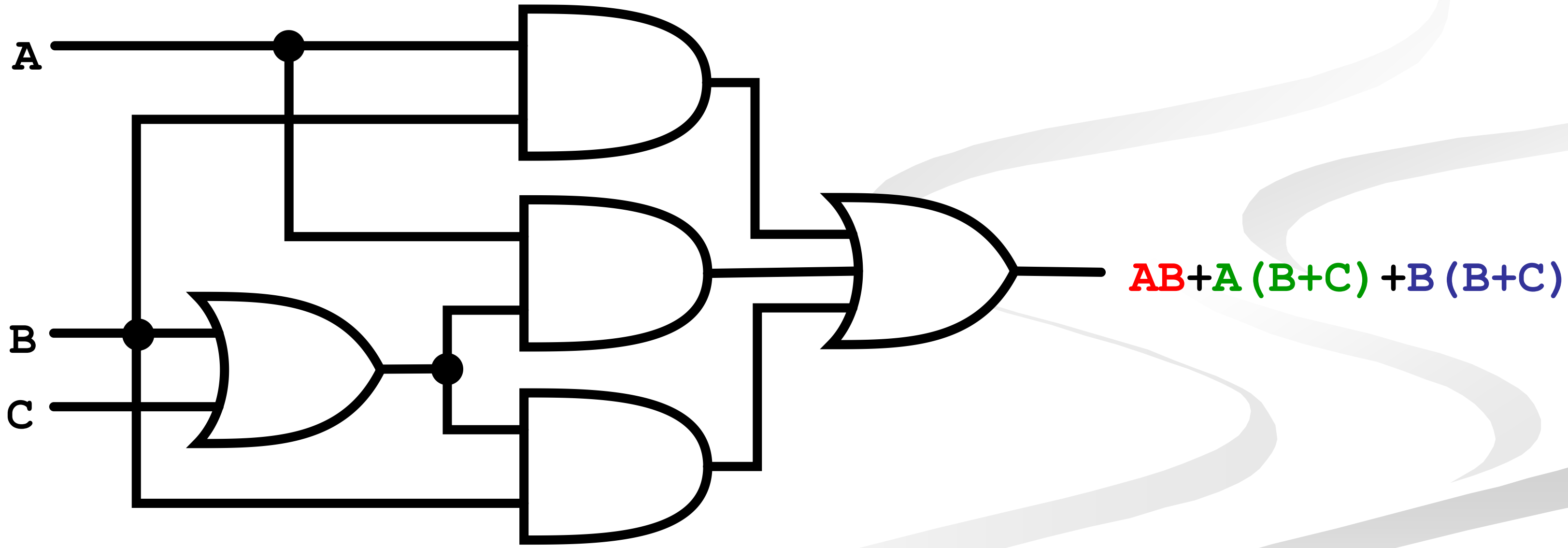
Logic Simplification-Using K-Maps

Lectures By
Manesh T
EE2121-In Charge
m.thankappan@psau.edu.sa

Contents

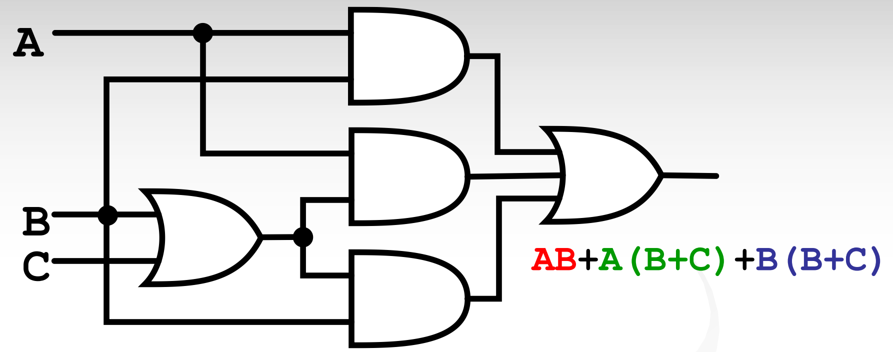
Simplification Using Boolean Algebra

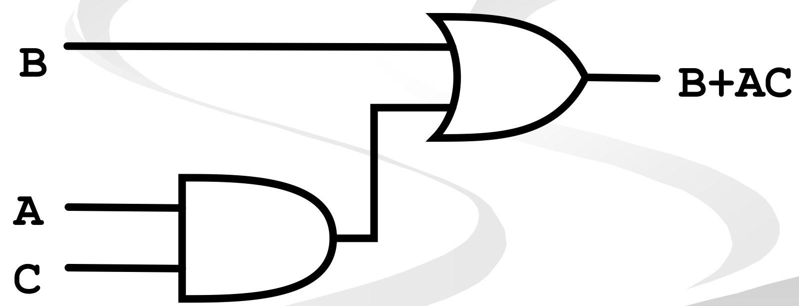
- A simplified Boolean expression uses the fewest gates possible to implement a given expression.



Simplification Using Boolean Algebra

- AB+A(B+C)+B(B+C)
  - (distributive law)
    - AB+AB+AC+BB+BC
  - (rule 7; BB=B)
    - AB+AB+AC+**B**+BC
  - (rule 5; AB+AB=AB)
    - **AB**+AC+B+BC
  - (rule 10; B+BC=B)
    - AB+AC+**B**
  - (rule 10; AB+B=B)
    - **B**+AC





Simplification Using Boolean Algebra

- Try these:

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

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

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

Standard Forms of Boolean Expressions

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

Sum-of-Products (SOP)

Canonical Forms-Minterms and Maxterms

x	y	z	minterm	designation	maxterm	designation
0	0	0	$\bar{x}\,\bar{y}\,\bar{z}$	m_0	$x+y+z$	M_0
0	0	1	$\bar{x}\,\bar{y}\,z$	m_1	$x+y+\bar{z}$	M_1
0	1	0	$\bar{x}\,y\,\bar{z}$	m_2	$x+\bar{y}+z$	M_2
0	1	1	$\bar{x}\,y\,z$	m_3	$x+\bar{y}+\bar{z}$	M_3
1	0	0	$x\,\bar{y}\,\bar{z}$	m_4	$\bar{x}+y+z$	M_4
1	0	1	$x\,\bar{y}\,z$	m_5	$\bar{x}+y+\bar{z}$	M_5
1	1	0	$x\,y\,\bar{z}$	m_6	$\bar{x}+\bar{y}+z$	M_6
1	1	1	$x\,y\,z$	m_7	$\bar{x}+\bar{y}+\bar{z}$	M_7
			(AND terms)		(OR terms)	

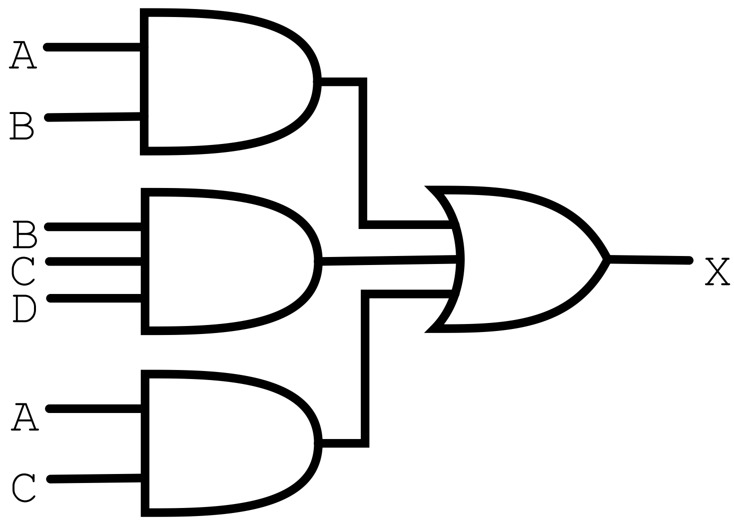
The Sum-of-Products (SOP) Form

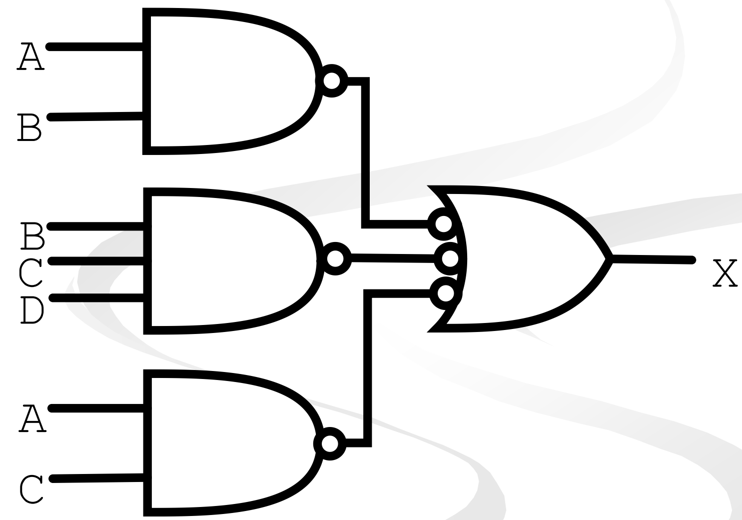
- An SOP expression → when two or more product terms are summed by Boolean addition.
  - Examples:
    $$AB + ABC$$
    $$ABC + CDE + \overline{B}C\overline{D}$$
    $$\overline{A}B + \overline{A}B\overline{C} + AC$$
  - Also:
    $$A + \overline{A}\overline{B}C + BC\overline{D}$$
- In an SOP form, a single overbar cannot extend over more than one variable; however, more than one variable <u>in a term</u> can have an overbar:
  - example: $\overline{A}\overline{B}\overline{C}$ is OK!
  - **But not:** $\overline{ABC}$

Implementation of an SOP

$X = AB + BCD + AC$

- AND/OR implementation
- NAND/NAND implementation





General Expression → SOP

- Any logic expression can be changed into SOP form by applying Boolean algebra techniques.

  ex:

$$A(B+CD) = AB + ACD$$

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

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

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

The Standard SOP Form

- A standard SOP expression is one in which *all* the variables in the domain appear <u>in each product term</u> in the expression.
  - Example:

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

- Standard SOP expressions are important in:
  - Constructing truth tables
  - The Karnaugh map simplification method

Converting Product Terms to Standard SOP

- **Step 1:** Multiply each nonstandard product term by a term made up of the sum of a missing variable and its complement. This results in two product terms.
  - As you know, you can multiply anything by 1 without changing its value.
- **Step 2:** Repeat step 1 until all resulting product term contains 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.

Converting Product Terms to Standard SOP (example)

- Convert the following Boolean expression into standard SOP form:

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

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

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

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

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

Binary Representation of a Standard Product Term

- A standard product term is equal to 1 for only one combination of variable values.
  - Example: $A\overline{B}C\overline{D}$ is equal to 1 when A=1, B=0, C=1, and D=0 as shown below

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

  - And this term is 0 for all other combinations of values for the variables.

Product-of-Sums (POS)

The Product-of-Sums (POS) Form

- When two or more sum terms are multiplied, the result expression is a product-of-sums (POS):
  - Examples:

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

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

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

  - Also:

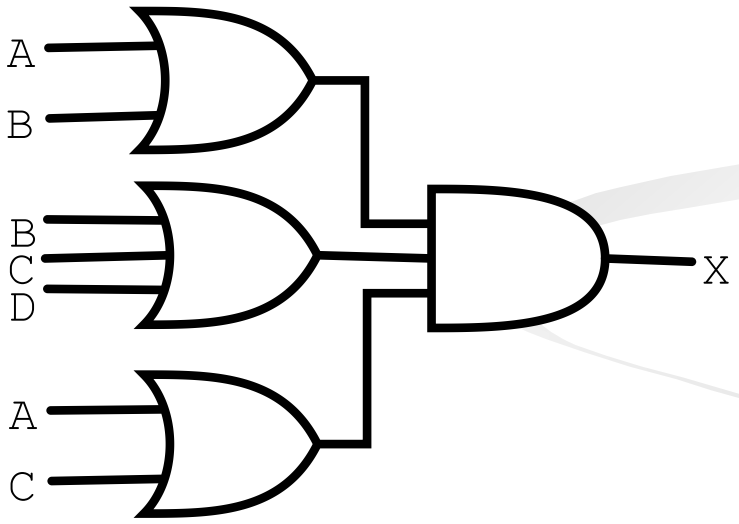
$$\overline{A}(\overline{A}+\overline{B}+C)(B+C+\overline{D})$$

- In a POS form, a single overbar cannot extend over more than one variable; however, more than one variable <u>in a term</u> can have an overbar:
  - example: $\overline{A}+\overline{B}+\overline{C}$ is OK!
  - **But not**: $\overline{A+B+C}$

Implementation of a POS

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

- OR/AND implementation



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.
  - Example: $(\overline{A}+\overline{B}+\overline{C}+\overline{D})(A+\overline{B}+C+D)(A+B+\overline{C}+D)$

- Standard POS expressions are important in:
  - Constructing truth tables
  - The Karnaugh map simplification method

Converting a Sum Term to Standard POS

- **Step 1:** Add to each nonstandard product term a term made up of the product of the missing variable and its complement. This results in two sum terms.
  - As you know, you can add 0 to anything without changing its value.
- **Step 2:** Apply rule 12 → $A+BC=(A+B)(A+C)$.
- **Step 3:** Repeat step 1 until all resulting sum terms contain all variable 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:

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

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

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

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

Binary Representation of a Standard Sum Term

- A standard sum term is equal to 0 for only one combination of variable values.
  - Example: $A+\overline{B}+C+\overline{D}$ is equal to 0 when A=0, B=1, C=0, and D=1 as shown below

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

  - And this term is 1 for all other combinations of values for the variables.

SOP/POS

Converting Standard SOP to Standard POS

- The Facts:
  - The binary values of the product terms in a given standard SOP expression are not present in the equivalent standard POS expression.
  - The binary values that <u>are not</u> represented in the SOP expression are present in the equivalent POS expression.

Converting Standard SOP to Standard POS

- What can you use the facts?
  - Convert from standard SOP to standard POS.
- How?
  - **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.

Converting Standard SOP to Standard POS (example)

- Convert the SOP expression to an equivalent POS expression:

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

  - The evaluation is as follows:

$$000 + 010 + 011 + 101 + 111$$

  - There are 8 possible combinations. The SOP expression contains five of these, so the POS must contain the other 3 which are: 001, 100, and 110.

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

Boolean Expressions & Truth Tables

- All standard Boolean expression can be easily converted into truth table format using binary values for each term in the expression.
- Also, standard SOP or POS expression can be determined from the truth table.

Converting SOP Expressions to Truth Table Format

- Recall the fact:
  - An SOP expression is equal to 1 only if at least one of the product term is equal to 1.
- Constructing a truth table:
  - **Step 1:** List all possible combinations of binary values of the variables in the expression.
  - **Step 2:** Convert the SOP expression to standard form if it is not already.
  - **Step 3:** Place a 1 in the output column (X) for each binary value that makes the standard SOP expression a 1 and place 0 for all the remaining binary values.

Converting SOP Expressions to Truth Table Format (example)

- Develop a truth table for the standard SOP expression

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

Inputs			Output	Product Term
A	B	C	X	
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

Converting POS Expressions to Truth Table Format

- Recall the fact:
  - A POS expression is equal to 0 only if at least one of the product term is equal to 0.
- Constructing a truth table:
  - **Step 1:** List all possible combinations of binary values of the variables in the expression.
  - **Step 2:** Convert the POS expression to standard form if it is not already.
  - **Step 3:** Place a 0 in the output column (X) for each binary value that makes the standard POS expression a 0 and place 1 for all the remaining binary values.

Converting POS Expressions to Truth Table Format (example)

- Develop a truth table for the standard SOP expression

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

Inputs			Output	Product Term
A	B	C	X	
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 Expression from a Truth Table

- To determine the standard **SOP expression** represented by a truth table.
- Instructions:
  - **Step 1:** List the binary values of the input variables for which the output is 1.
  - **Step 2:** 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.
- Example: 1010 → $A\overline{B}C\overline{D}$

Determining Standard Expression from a Truth Table

- To determine the standard **POS expression** represented by a truth table.
- Instructions:
  - **Step 1:** List the binary values of the input variables for which the output is 0.
  - **Step 2:** Convert each binary value to the corresponding product term by replacing:
    - each 1 with the corresponding variable complement, and
    - each 0 with the corresponding variable.
- Example: 1001 → $\overline{A}+B+C+\overline{D}$

Determining Standard Expression from a Truth Table (example)

I/P			O/P
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

SOP

- There are <u>four 1s</u> in the output and the corresponding binary value are 011, 100, 110, and 111.

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

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

POS

- There are <u>four 0s</u> in the output and the corresponding binary value are 000, 001, 010, and 101.

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

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

The Karnaugh Map

The Karnaugh Map

- Feel a little difficult using Boolean algebra laws, rules, and theorems to simplify logic?
- A K-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.

What is K-Map

- It's similar to truth table; instead of being organized (i/p and o/p) into columns and rows, the K-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.
- K-maps can be used for expressions with 2, 3, 4, and 5 variables.
  - 3 and 4 variables will be discussed to illustrate the principles.

The 3 Variable K-Map

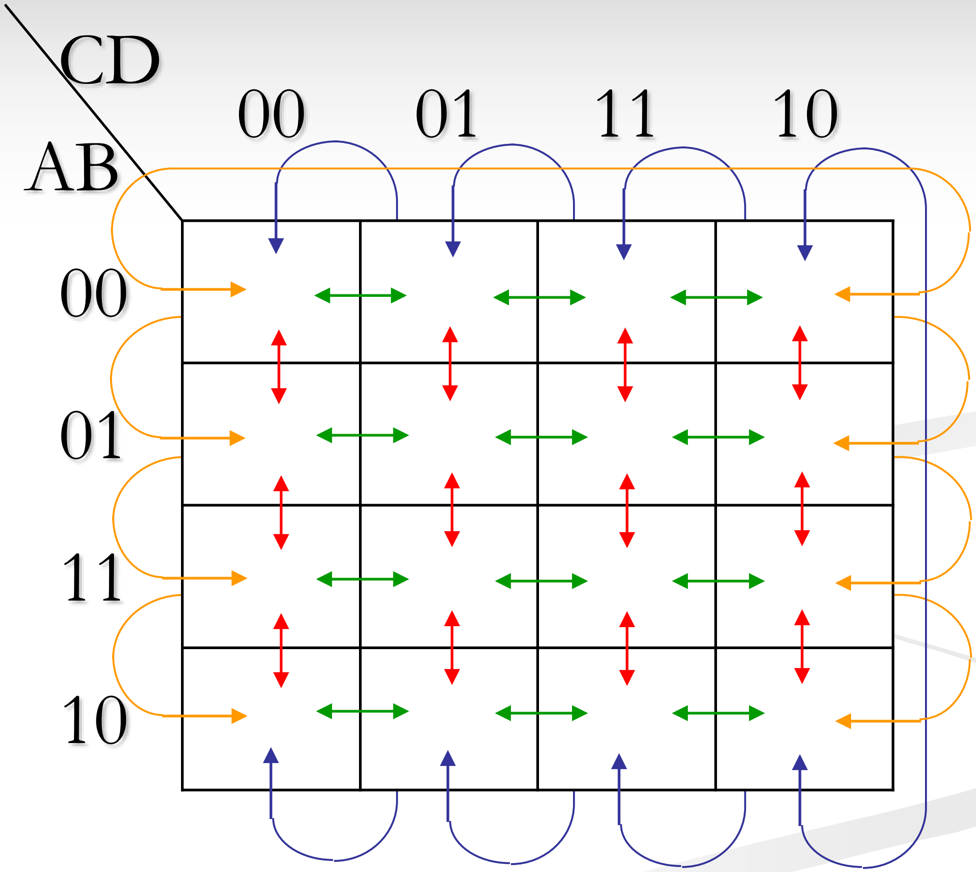
- There are 8 cells as shown:

AB \ C	0	1
00	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$
01	$\bar{A}B\bar{C}$	$\bar{A}BC$
11	$AB\bar{C}$	ABC
10	$A\bar{B}\bar{C}$	$A\bar{B}C$

The 4-Variable K-Map

AB \ CD	00	01	11	10
00	$\overline{A}\overline{B}\overline{C}\overline{D}$	$\overline{A}\overline{B}\overline{C}D$	$\overline{A}\overline{B}CD$	$\overline{A}\overline{B}C\overline{D}$
01	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}BCD$	$\overline{A}BC\overline{D}$
11	$AB\overline{C}\overline{D}$	$AB\overline{C}D$	$ABCD$	$ABC\overline{D}$
10	$A\overline{B}\overline{C}\overline{D}$	$A\overline{B}\overline{C}D$	$A\overline{B}CD$	$A\overline{B}C\overline{D}$

Cell Adjacency



K-Map SOP Minimization

- The K-Map is used for simplifying Boolean expressions to their minimal form.
- A minimized SOP expression contains the fewest possible terms with fewest possible variables per term.
- Generally, a minimum SOP expression can be implemented with fewer logic gates than a standard expression.

Mapping a Standard SOP Expression

- For an SOP expression in standard form:
  - A 1 is placed on the K-map for each product term in the expression.
  - Each 1 is placed in a cell corresponding to the value of a product term.
  - Example: for the product term $A\overline{B}C$, a 1 goes in the 101 cell on a 3-variable map.

AB \ C	0	1
00	$\overline{A}\,\overline{B}\,\overline{C}$	$\overline{A}\,\overline{B}C$
01	$\overline{A}B\overline{C}$	$\overline{A}BC$
11	$AB\overline{C}$	ABC
10	$A\overline{B}\,\overline{C}$	$A\overline{B}C$ 1

Mapping a Standard SOP Expression (full example)

The expression:

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

000	001	110	100

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

Practice:

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

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

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

Mapping a Nonstandard SOP Expression

- A Boolean expression must be in standard form before you use a K-map.
  - If one is not in standard form, it must be converted.
- You may use the procedure mentioned earlier or use numerical expansion.

Mapping a Nonstandard SOP Expression

- Numerical Expansion of a Nonstandard product term
  - Assume that one of the product terms in a certain 3-variable SOP expression is $A\overline{B}$.
  - It can be expanded numerically to standard form as follows:
    - **Step 1:** Write the binary value of the two variables and attach a 0 for the missing variable $\overline{C}$: 100.
    - **Step 2:** Write the binary value of the two variables and attach a 1 for the missing variable C: 100.
  - The two resulting binary numbers are the values of the standard SOP terms → $A\overline{B}\overline{C}$ and $A\overline{B}C$.
- If the assumption that one of the product term in a 3-variable expression is B. How can we do this?

Mapping a Nonstandard SOP Expression

- Map the following SOP expressions on K-maps:

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

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

K-Map Simplification of SOP Expressions

- After an SOP expression has been mapped, we can do the process of *minimization*:
  - Grouping the 1s
  - Determining the minimum SOP expression from the map

Grouping the 1s

- You can group 1s on the K-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.

Grouping the 1s (rules)

1. A group must contain either 1,2,4,8,or 16 cells (depending on number of variables in the expression)
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 the 1s (example)

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

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

Grouping the 1s (example)

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

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

Determining the Minimum SOP Expression from the Map

- The following rules are applied to find the minimum product terms and the minimum SOP expression:
  1. Group the cells that have 1s. Each group of cell containing 1s creates one product term composed of all variables that occur in only one form (either complemented or complemented) within the group. Variables that occur both complemented and uncomplemented within the group are eliminated → called *contradictory variables*.

Determining the Minimum SOP Expression from the Map

2. Determine the minimum product term for each group.
   - For a 3-variable map:
     1. A 1-cell group yields a 3-variable product term
     2. A 2-cell group yields a 2-variable product term
     3. A 4-cell group yields a 1-variable product term
     4. An 8-cell group yields a value of 1 for the expression.
   - For a 4-variable map:
     1. A 1-cell group yields a 4-variable product term
     2. A 2-cell group yields a 3-variable product term
     3. A 4-cell group yields a 2-variable product term
     4. An 8-cell group yields a a 1-variable product term
     5. A 16-cell group yields a value of 1 for the expression.

Determining the Minimum SOP Expression from the Map

3. When all the minimum product terms are derived from the K-map, they are summed to form the minimum SOP expression.

Determining the Minimum SOP Expression from the Map (example)

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

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

$\overline{A}C$

B

$A\overline{C}D$

Determining the Minimum SOP Expression from the Map (exercises)

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

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

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

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

Determining the Minimum SOP Expression from the Map (exercises)

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

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

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

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

Practicing K-Map (SOP)

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

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

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

$$\overline{D} + \overline{B}C$$

Mapping Directly from a Truth Table

I/P			O/P
A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

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

"Don't Care" Conditions

- Sometimes a situation arises in which some input variable combinations are not allowed, i.e. BCD code:
  - There are six invalid combinations: 1010, 1011, 1100, 1101, 1110, and 1111.
- Since these unallowed states will never occur in an application involving the BCD code → they can be treated as "don't care" terms with respect to their effect on the output.
- The "don't care" terms can be used to advantage on the K-map (how? see the next slide).

"Don't Care" Conditions

INPUTS				O/P
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	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

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

Without "don't care"
$$Y = A\overline{B}\,\overline{C} + \overline{A}BCD$$

With "don't care"
$$Y = A + BCD$$

K-Map POS Minimization

- The approaches are much the same (as SOP) except that with POS expression, 0s representing the standard sum terms are placed on the K-map instead of 1s.

Mapping a Standard POS Expression (full example)

The expression:

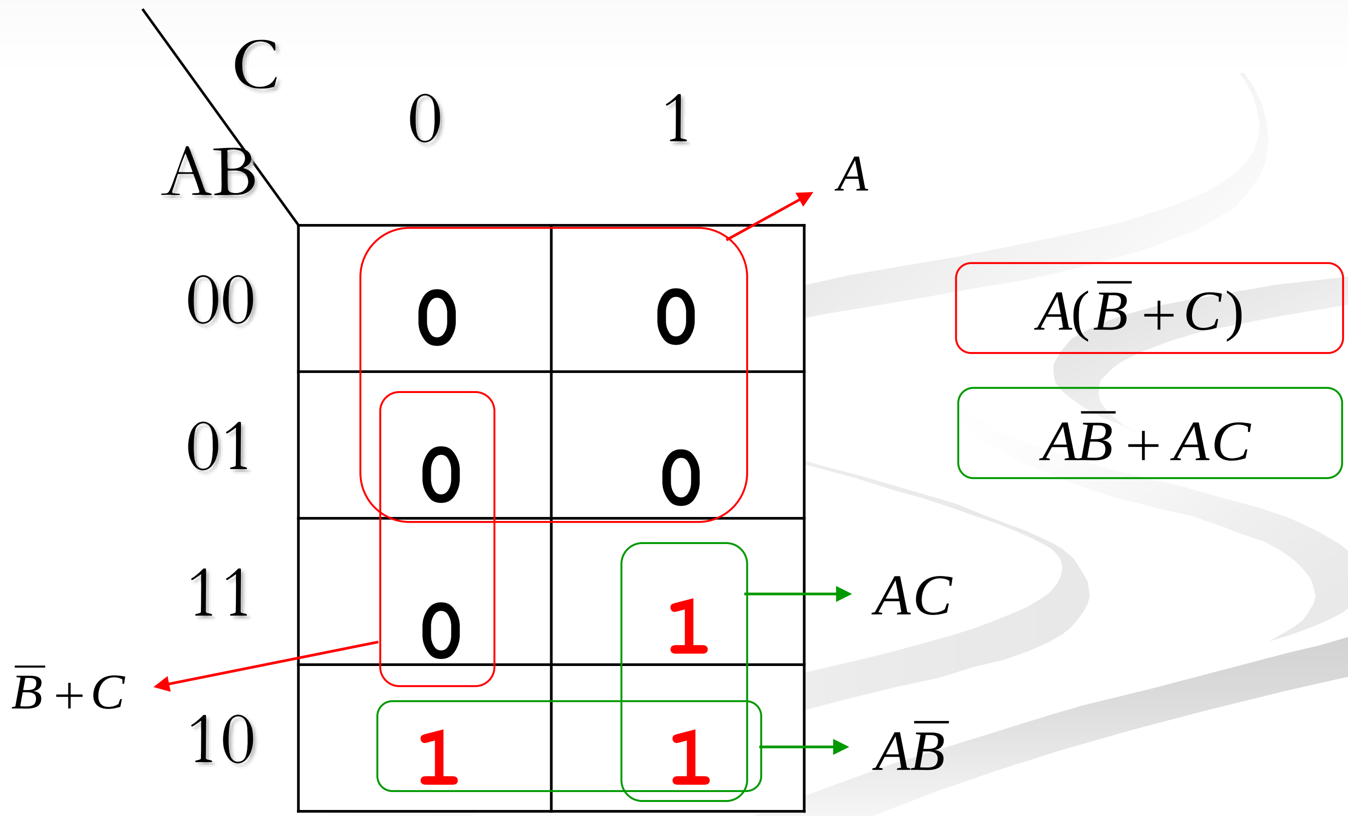
$$(A+B+C)(A+\overline{B}+C)(\overline{A}+\overline{B}+C)(\overline{A}+B+\overline{C})$$

000 010 110 101

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

K-map Simplification of POS Expression

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



Rules of Boolean Algebra

$$1. A + 0 = A$$

$$2. A + 1 = 1$$

$$3. A \bullet 0 = 0$$

$$4. A \bullet 1 = A$$

$$5. A + A = A$$

$$6. A + \overline{A} = 1$$

$$7. A \bullet A = A$$

$$8. A \bullet \overline{A} = 0$$

$$9. \overline{\overline{A}} = A$$

$$10. A + AB = A$$

$$11. A + \overline{A}B = A + B$$

$$12. (A + B)(A + C) = A + BC$$

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