Tishk International University
Science Faculty
IT Department

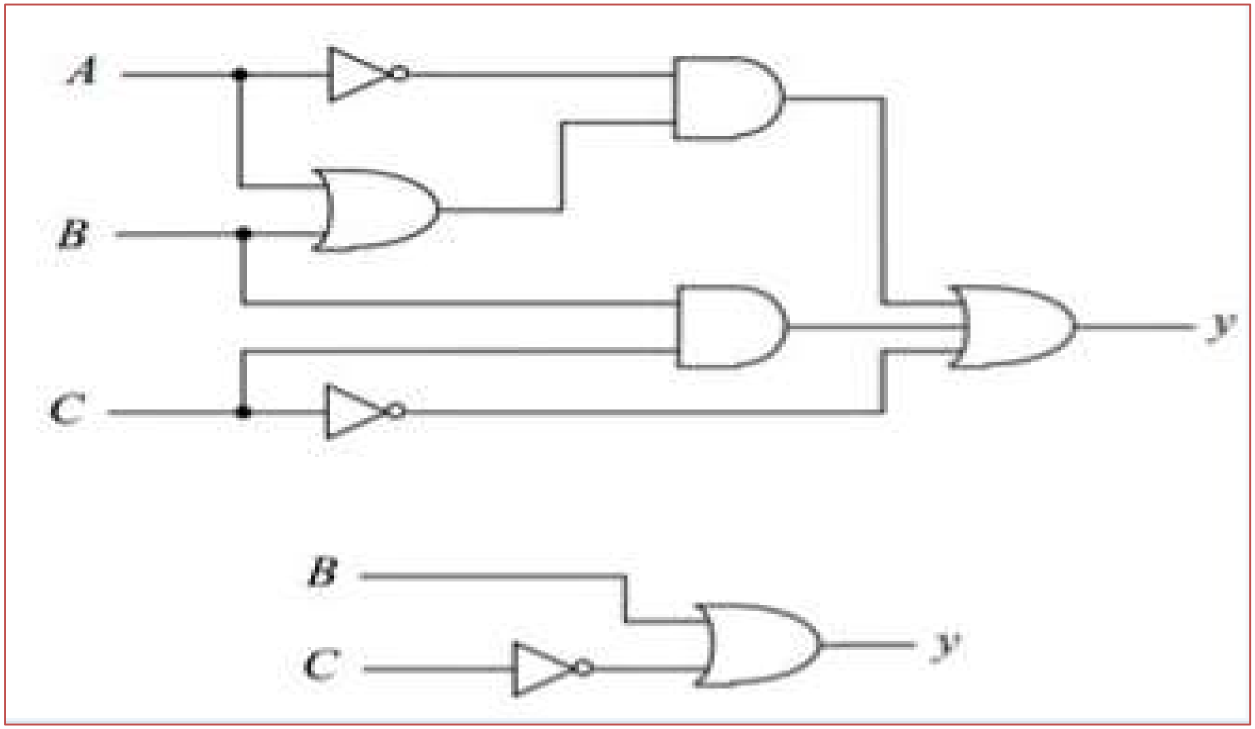
Logic Design

Lecture 3: Logic Simplification

2nd Grade

Instructor: Alaa Ghazi

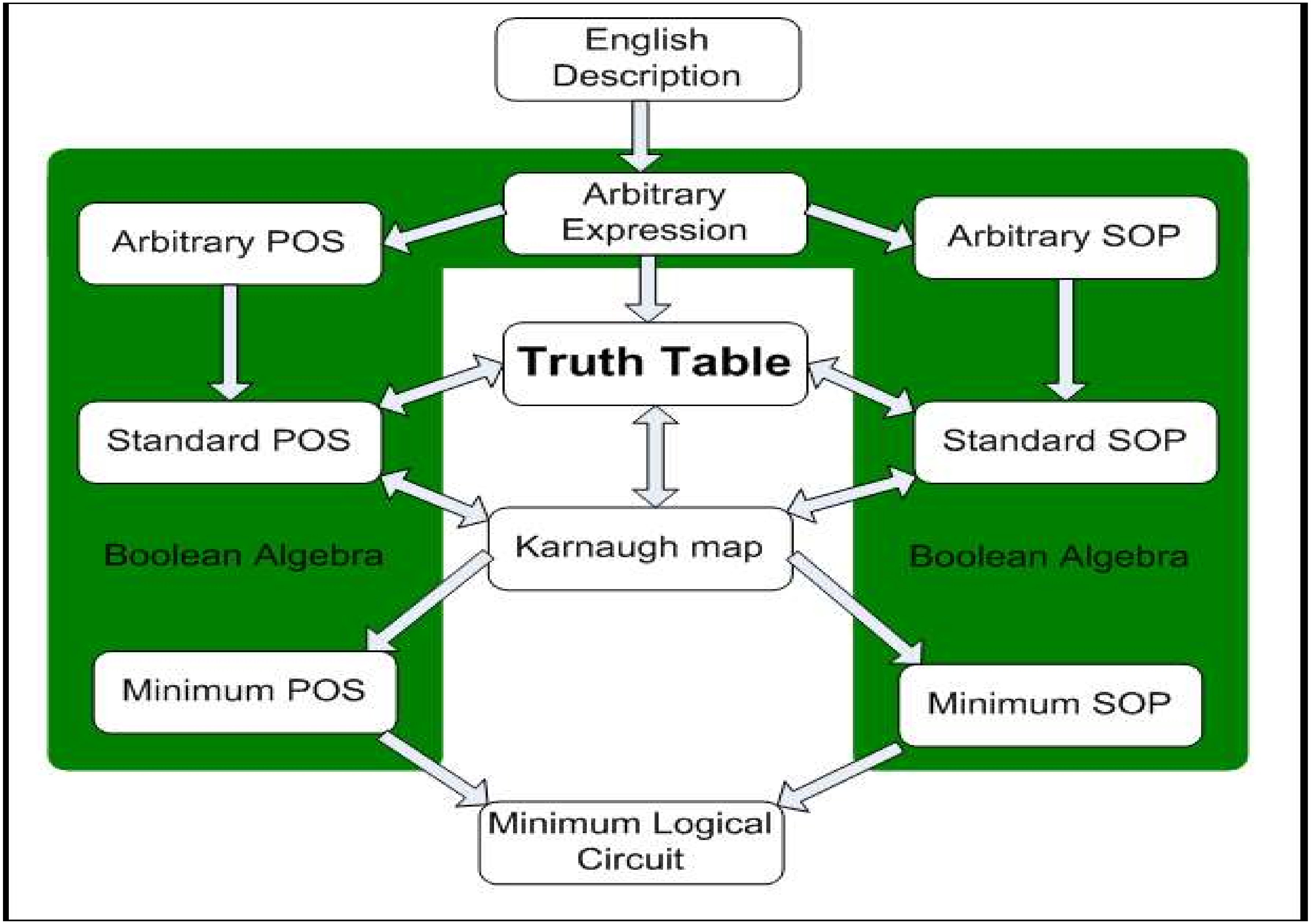
Lecture 3
Logic Simplification



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.

Logic Design Core Diagram



Logic Basics Review

- A variable AND'ed with 0 is always equal to 0
- A variable AND'ed with 1 is always equal to the variable
- A variable AND'ed with its complement is always equal to 0
- A variable OR'ed with 0 is always equal to the variable
- A variable OR'ed with 1 is always equal to 1
- A variable OR'ed with its complement is always equal to 1

$$A \cdot 0 = 0$$
$$A \cdot 1 = A$$
$$A \cdot \overline{A} = 0$$
$$A + 0 = A$$
$$A + 1 = 1$$
$$A + \overline{A} = 1$$

$A(B + C) = A.B + A.C$ (OR Distributive Law)

$A + (B.C) = (A + B).(A + C)$ (AND Distributive Law)

$$X = X.1 = X(A + \overline{A}) = XA + X\overline{A}$$

$$X = X + 0 = X + (A.\overline{A}) = (X + A)(X + \overline{A})$$

The Sum-of-Products (SOP) Form

- An SOP expression → when two or more product terms are summed by Boolean addition.
  - Examples:

$$AB + ABC$$

$$BC + CD \quad + \overline{B}C\overline{D}$$

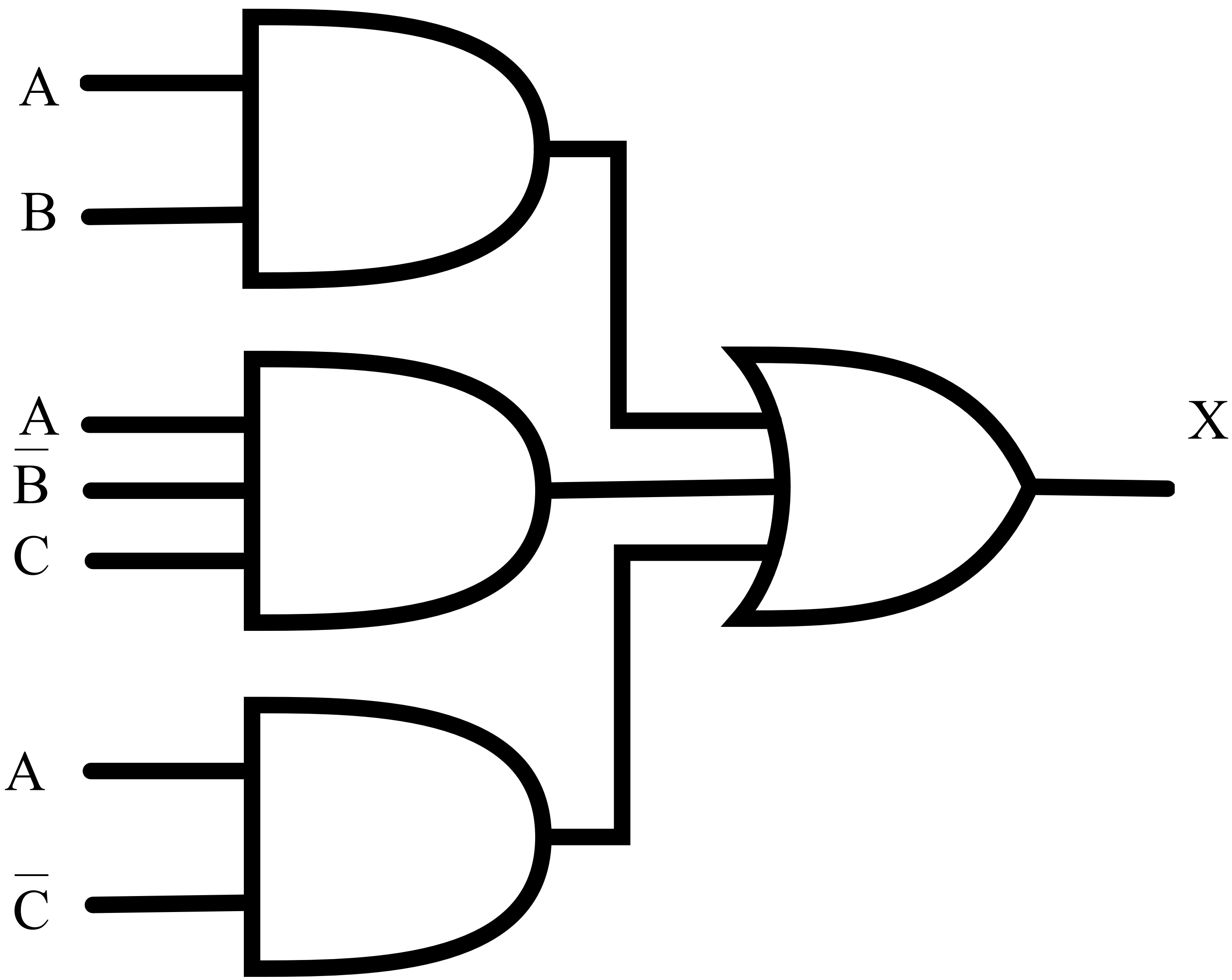
$$\overline{A}B + \overline{A}B\overline{C} + AC$$

- 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:** $\overline{ABC}$

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

AND/OR Implementation of SOP

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



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.
  - Example:

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

- 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. Then apply (OR Distributive Law)

$$X = X \cdot 1 = X(A + \overline{A}) = XA + X\overline{A}$$

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:

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

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

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

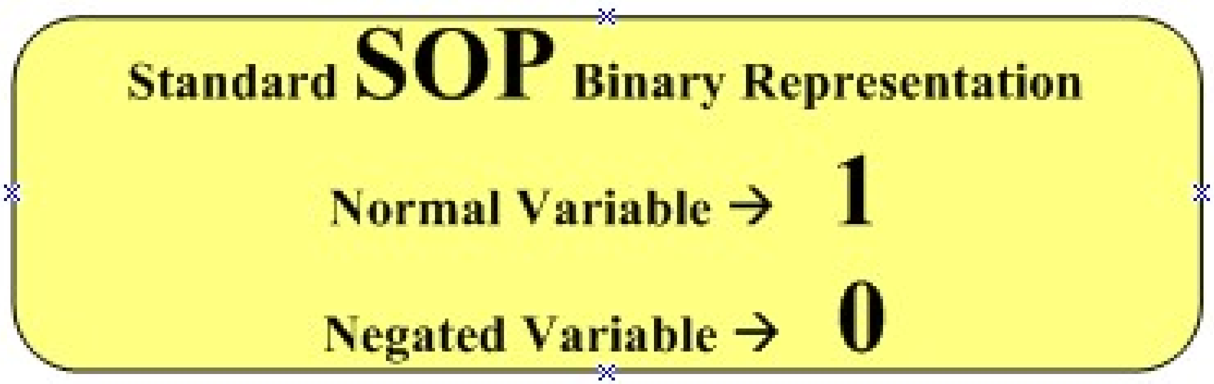
$$X = A\bar{B}C + A\bar{B}\,\bar{C} + \bar{A}B\bar{C} + \bar{A}\,\bar{B}\bar{C} + AB\bar{C}$$

Binary Representation of a Standard Product Term

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

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

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



The Product-of-Sums (POS) Form

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

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

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

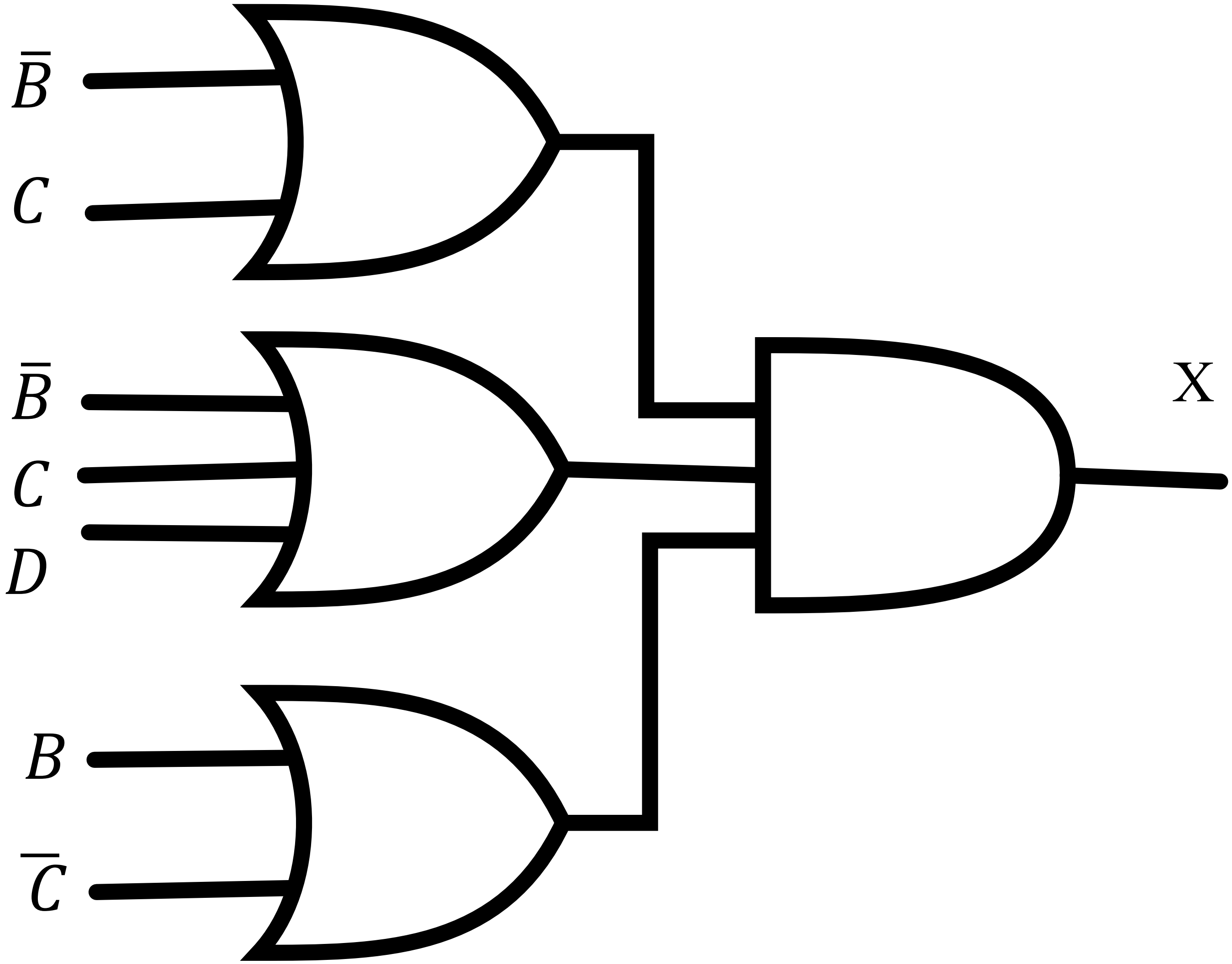
$$(A+B)(A+\overline{B}+C)(\overline{A}+C)$$

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

- 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**: $\overline{A+B+C}$

OR/AND Implementation of a POS

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



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

- 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 (AND Distributive Law)

$$X = X + 0 = X + (A.\overline{A}) = (X + A)(X + \overline{A})$$

- **Step 3:** Repeat step 1 until all resulting sum terms contain all variables in the domain in either normal or negated form.

Converting a Sum Term to Standard POS (example)

- Convert the following Boolean expression into standard POS form:

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

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

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

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

Binary Representation of a Standard Sum Term

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

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

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

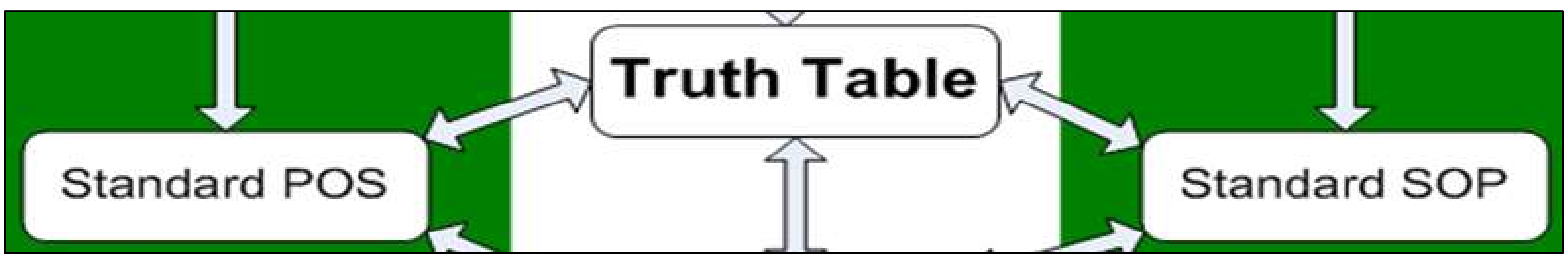
Standard POS Binary Representation

Normal Variable → 0

Negated Variable → 1

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.

Standard SOP → Truth Table
Put 1 for binary representation of each PRODUCT

Converting SOP Expressions to Truth Table Format (example)

- Develop a truth table for the standard SOP expression

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

Standard SOP → Truth Table

Each PRODUCT → 1

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.

Standard POS → Truth Table
Put 0 for binary representation of each SUM

Converting POS Expressions to Truth Table Format (example)

- Develop a truth table for the standard POS expression

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

Standard POS → Truth Table

Each SUM → 0

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: 010 → $\overline{B}C\overline{D}$

Truth Table →Standard SOP

Each 1 → PRODUCT

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: 100 → $\overline{A} + B + C$

Truth Table →Standard POS

Each 0 → SUM

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.

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

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

POS

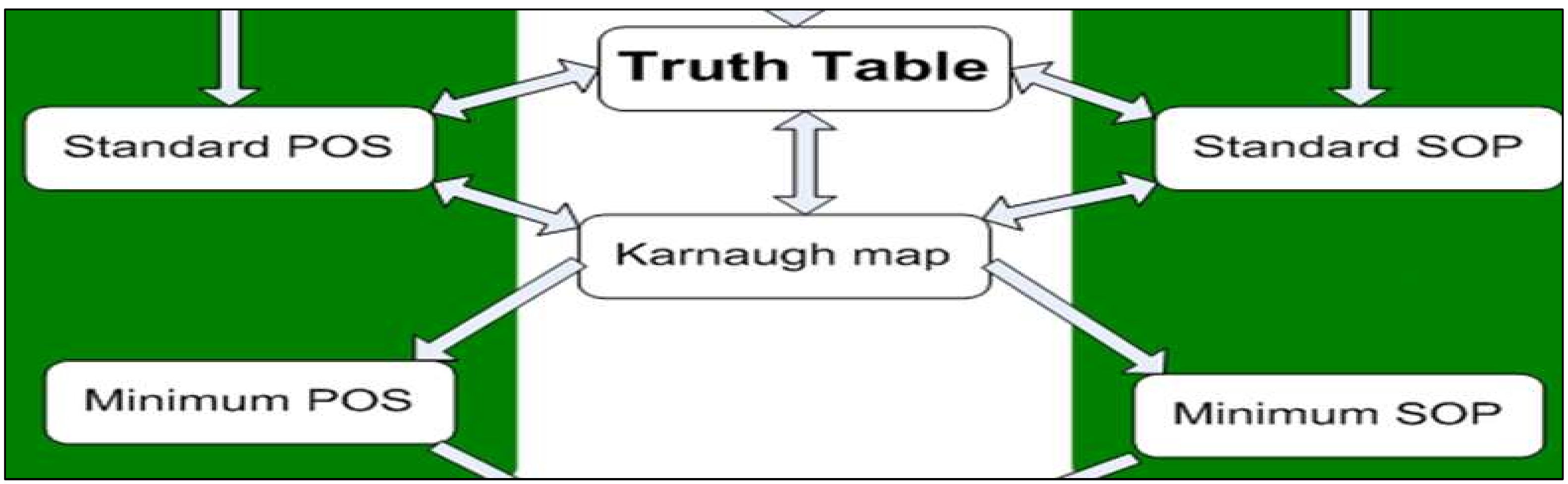
- There are <u>four 0s</u> in the output and the corresponding binary value are 000, 001, 010, and 101.

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

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

Standard SOP/POS To Minimum SOP/POS

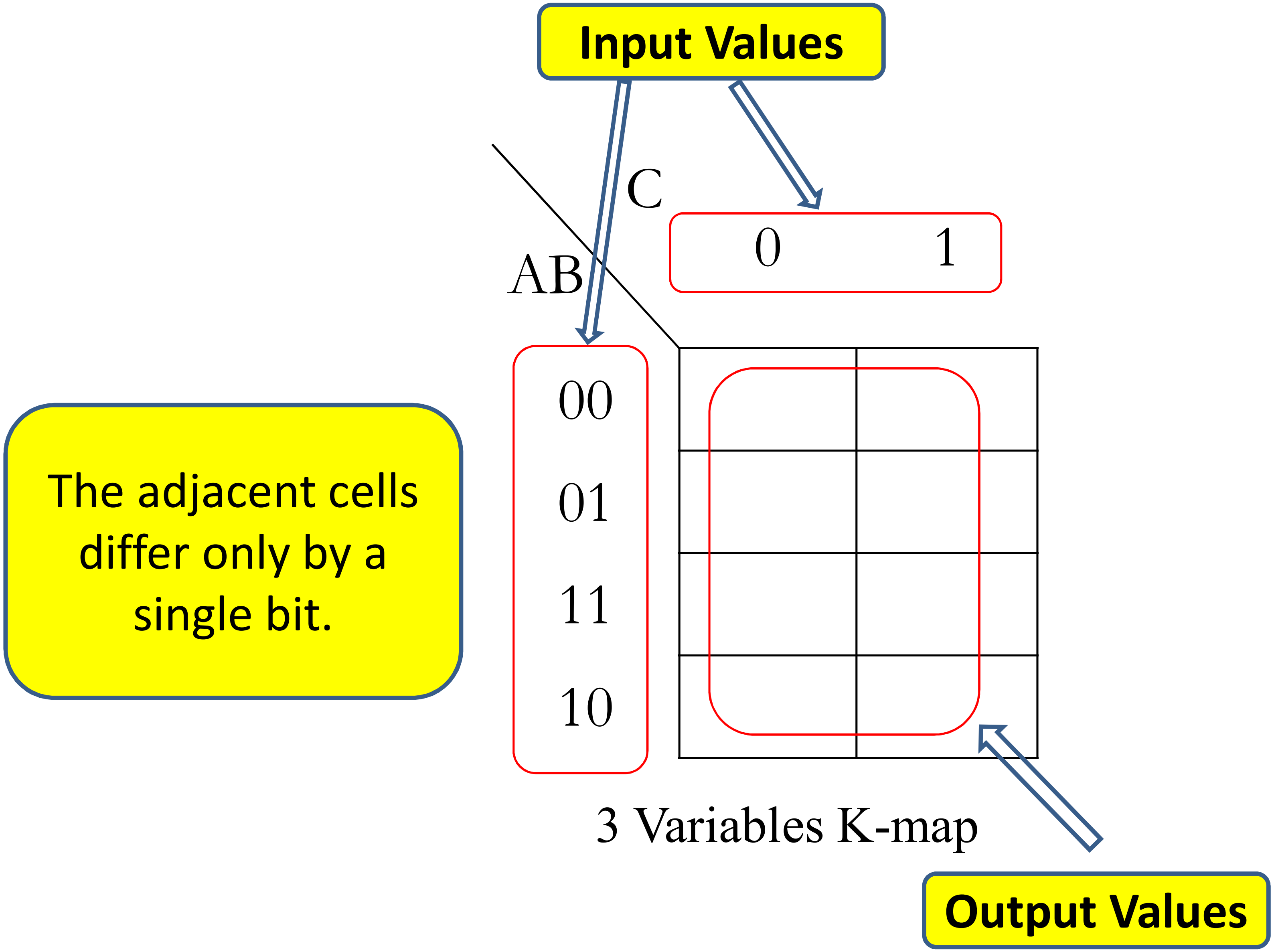
1. Convert Standard SOP/POS Expression to Truth Table (see previous slides)
2. Map the Truth Table to K-Map
3. Perform K-Map Grouping
4. Find the minimized SOP/POS expression from each group



The Karnaugh Map

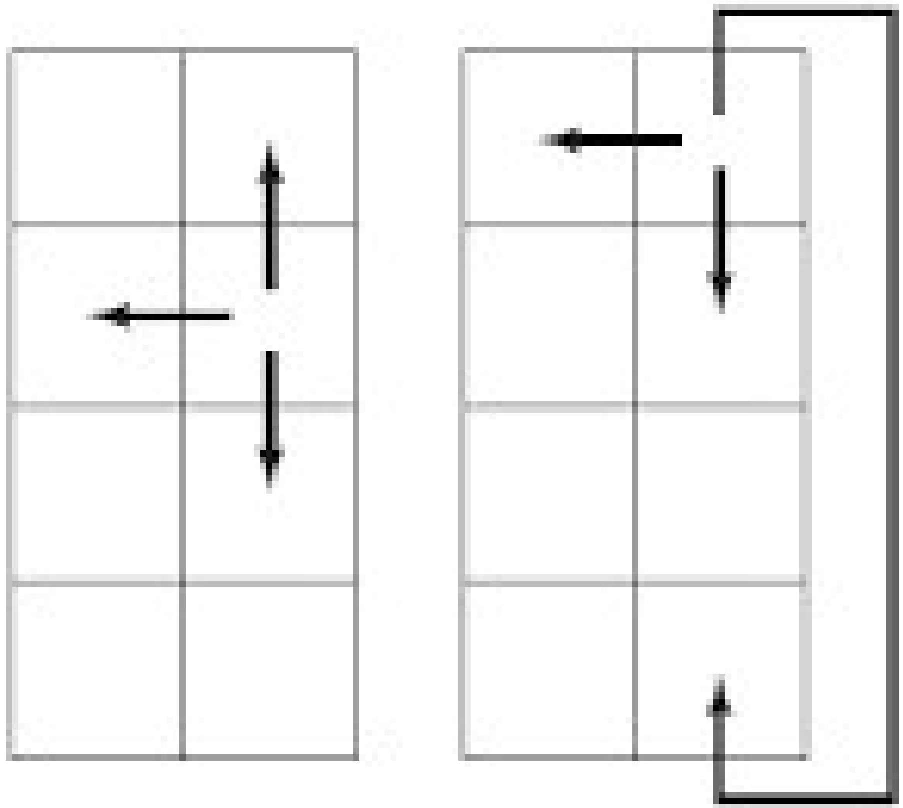
- A K-map provides a systematic method for simplifying Boolean expressions and, if properly used, will produce the simplest SOP or POS expression possible, known as the minimum expression.
- K-map can be thought of as a special version of a truth table. It is an array of cells in which each cell location represents a binary value of the input variables, and cell value represents the output variable. Visual Boolean function representation
- The main difference from Truth Table is that the adjacent cells differ only by a single bit. That is, if the given cell horizontal address is 01, then the previous and the next code-words can be 11 or 00, but cannot be 10 in any case. Mapping of Truth Table to K-Map should consider this point.
- The cells are arranged in a way so that simplification of a given expression is simply a matter of properly grouping the cells.
- **In this course we focus on 3 variables.**

K-Map Structure



Cell Adjacency

Groups can wrap around the K-map sides.



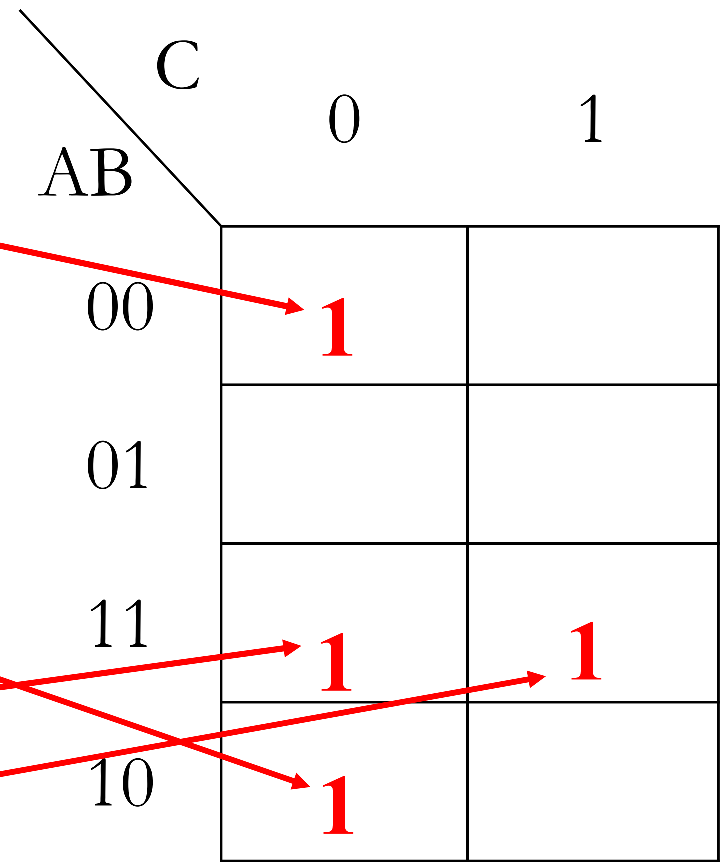
Truth Table to 3 Variables K-Map Mapping

	W	X	Y	F
0	0	0	0	**0**
1	0	0	1	**1**
2	0	1	0	**1**
3	0	1	1	**0**
4	1	0	0	**1**
5	1	0	1	**1**
6	1	1	0	**0**
7	1	1	1	**1**

WX \ Y	**0**	**1**
00	0 (0)	1 (1)
01	1 (2)	0 (3)
11	0 (6)	1 (7)
10	1 (4)	1 (5)

Mapping a Truth Table to K-Map (Example)

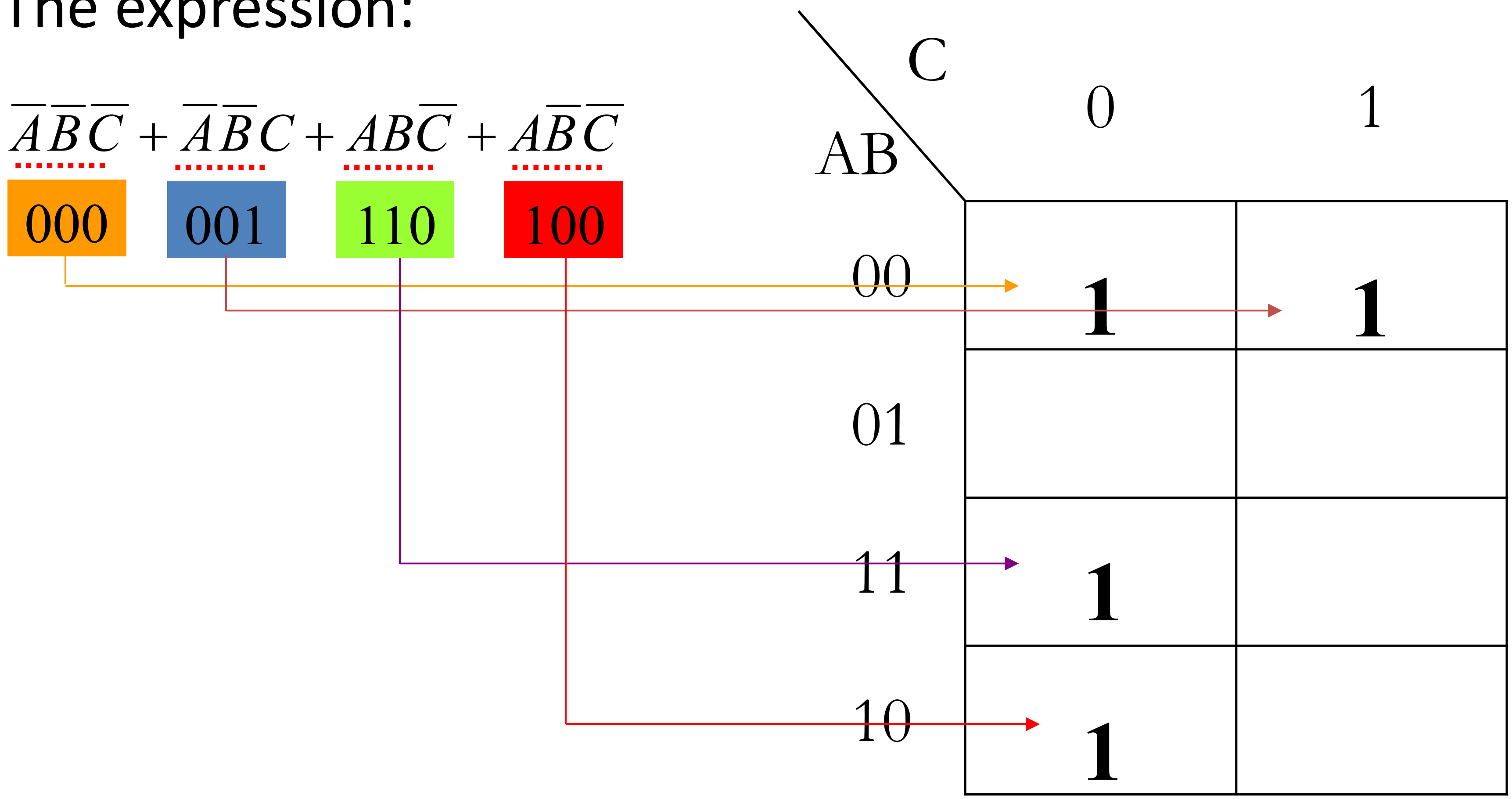
I/P			O/P
A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



Standard SOP to 3 Variables K-Map Mapping

The expression:

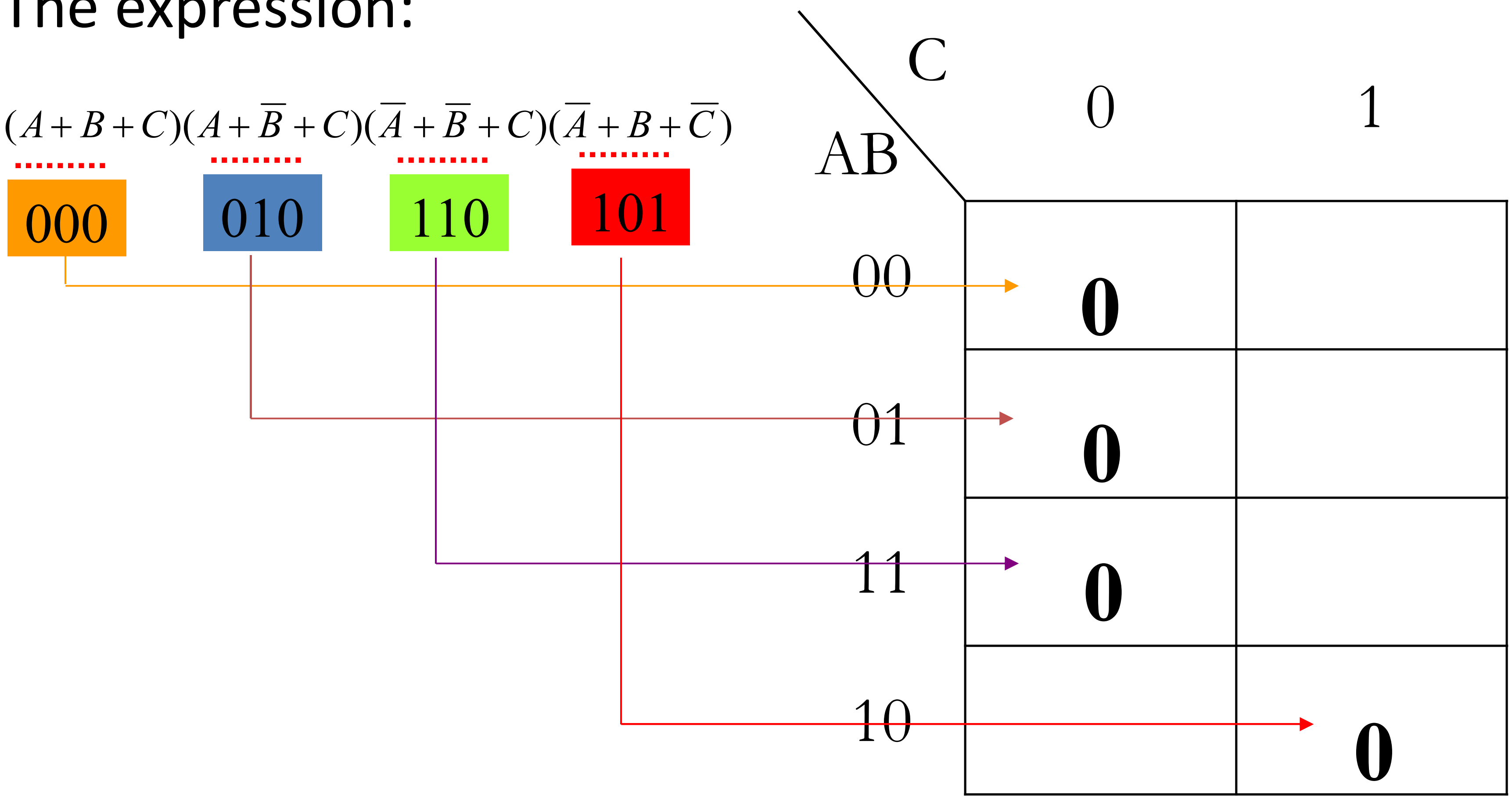
$$\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + AB\overline{C} + A\overline{B}\,\overline{C}$$



Standard POS to 3 Variables K-Map Mapping

The expression:

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



K-Map Grouping Rules

- **In SOP groups should include 1 only**
- **In POS groups should include 0 only**
- Groups may be doubled with **horizontal direction or vertical**, but not diagonal
- Number of cells are 1, 2, 4, or 8 in each group.
- Every cell must be in at least one group.
- Groups can **wrap around**. As the K-map is considered as spherical or folded, the squares at the corners (which are at the end of the column or row) should be considered as they adjacent cells
- Groups may **overlap** each other. The cells already in a group can be included in another group as long as the overlapping groups include non common cells.
- Each Group should be as **large** as possible.
- Minimum group size is **1 cell**
- Minimize number of groups.

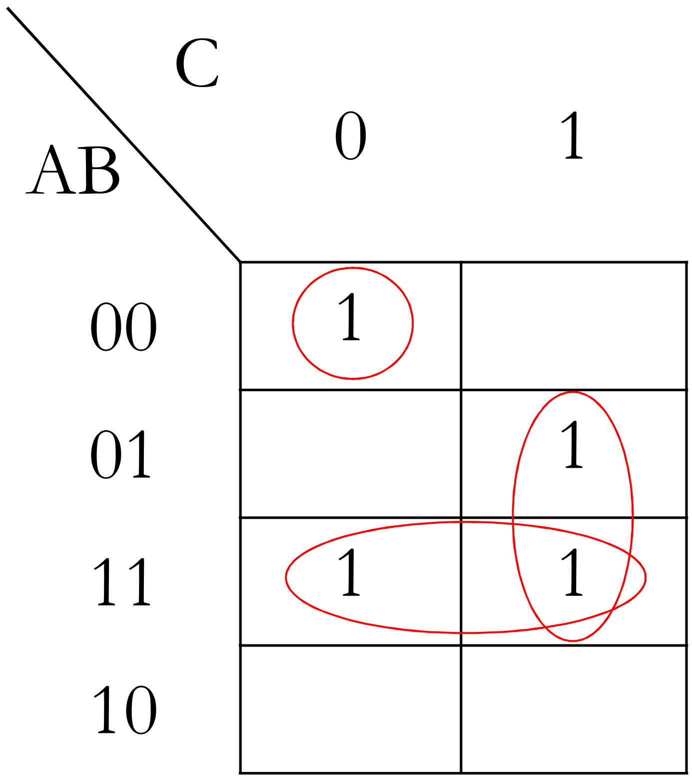
K-Map Grouping Procedures

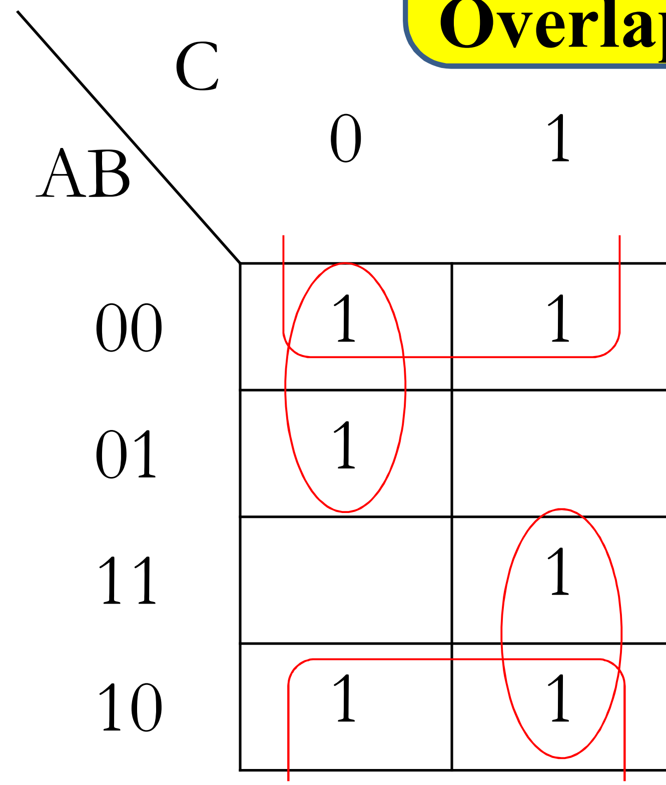
- Scan the map line by line
- **Find** ungrouped cell
- **Double** 1 cell with other ungrouped cells either horizontally or vertically, then double the 2, 4, 8 and so on. (Each time we double the size of a group, we remove changing variable from that group's)
- Double with **overlap** with other grouped cells either horizontally or vertically. Double the 2, 4, 8 and so on.

Find
Double
Overlap

Grouping SOP (3 variables example)

Find Double Overlap





Determining the Minimum SOP Expression from the Map

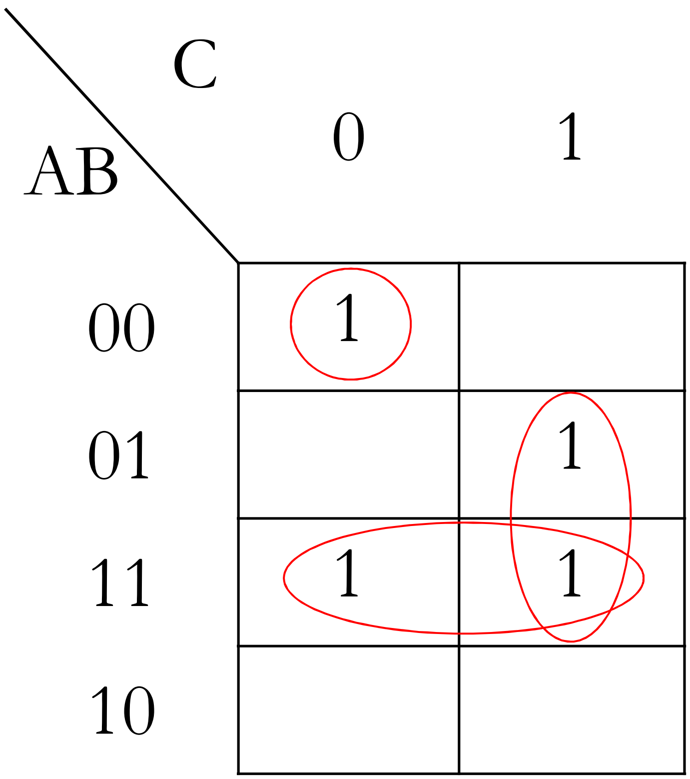
1. Group the cells that have 1s.
2. Each group of cell containing 1s creates one PRODUCT term
3. Remove repeating variables from the PRODUCT
4. When all the minimum product terms are derived from the K-map, they are summed to form the minimum SOP expression.

- The grouping of K-map variables can be done in many ways, so the obtained simplified expression need not to be unique always.

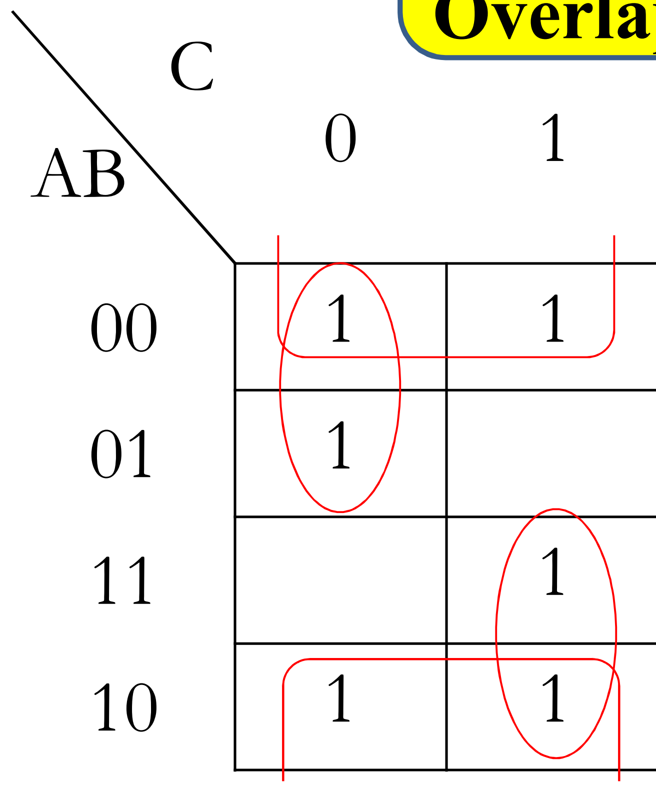
3 Variables K-Map	
cells	variables
1	3
2	2
4	1
8	Expression =1

Determining the Minimum SOP Expression from the Map (exercises)

Find Double Overlap



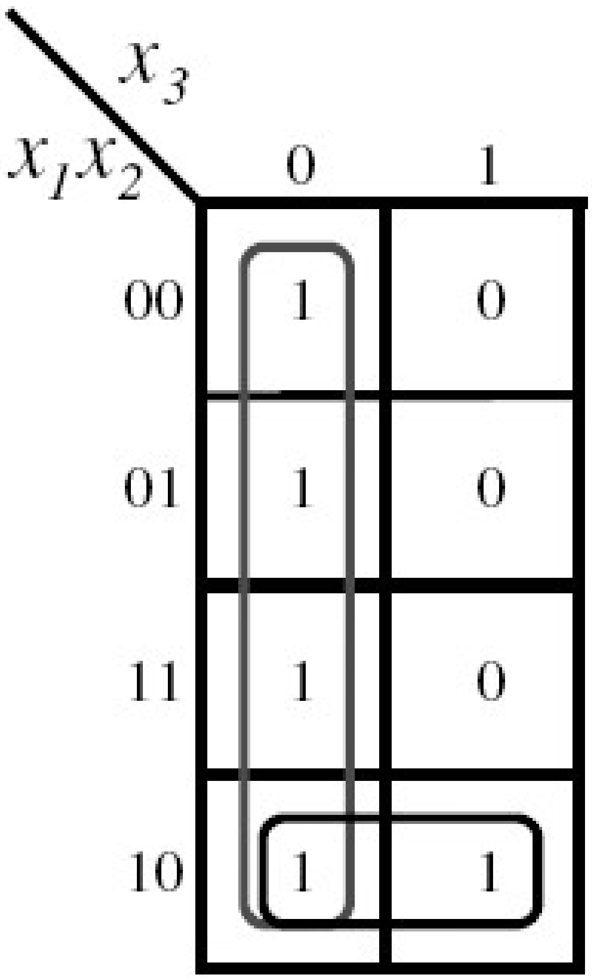
$$X = AB + BC + \overline{A}\,\overline{B}\,\overline{C}$$



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

K-map Simplification of SOP Examples

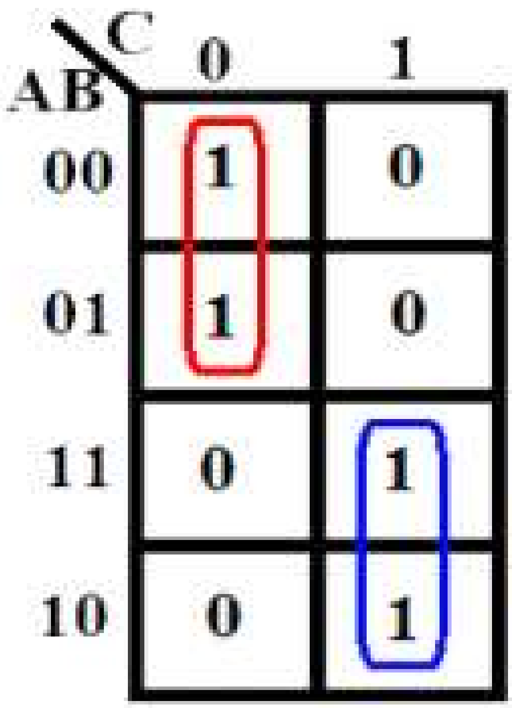
Find Double Overlap

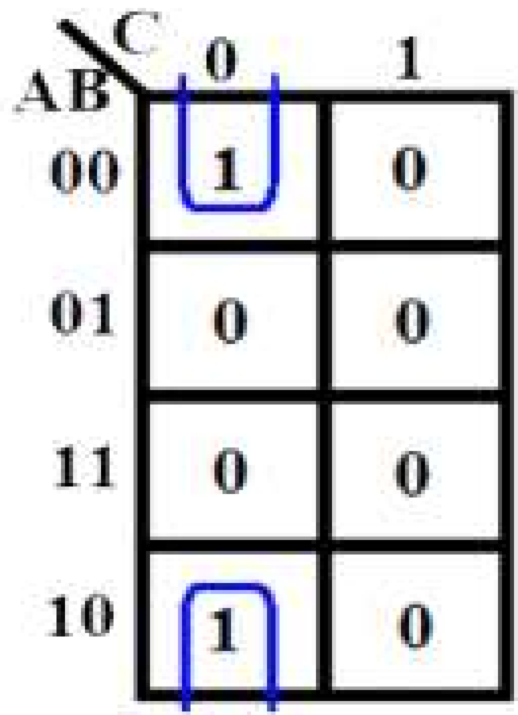


$$f = \overline{x}_3 + x_1\overline{x}_2$$

K-map Simplification of SOP Examples

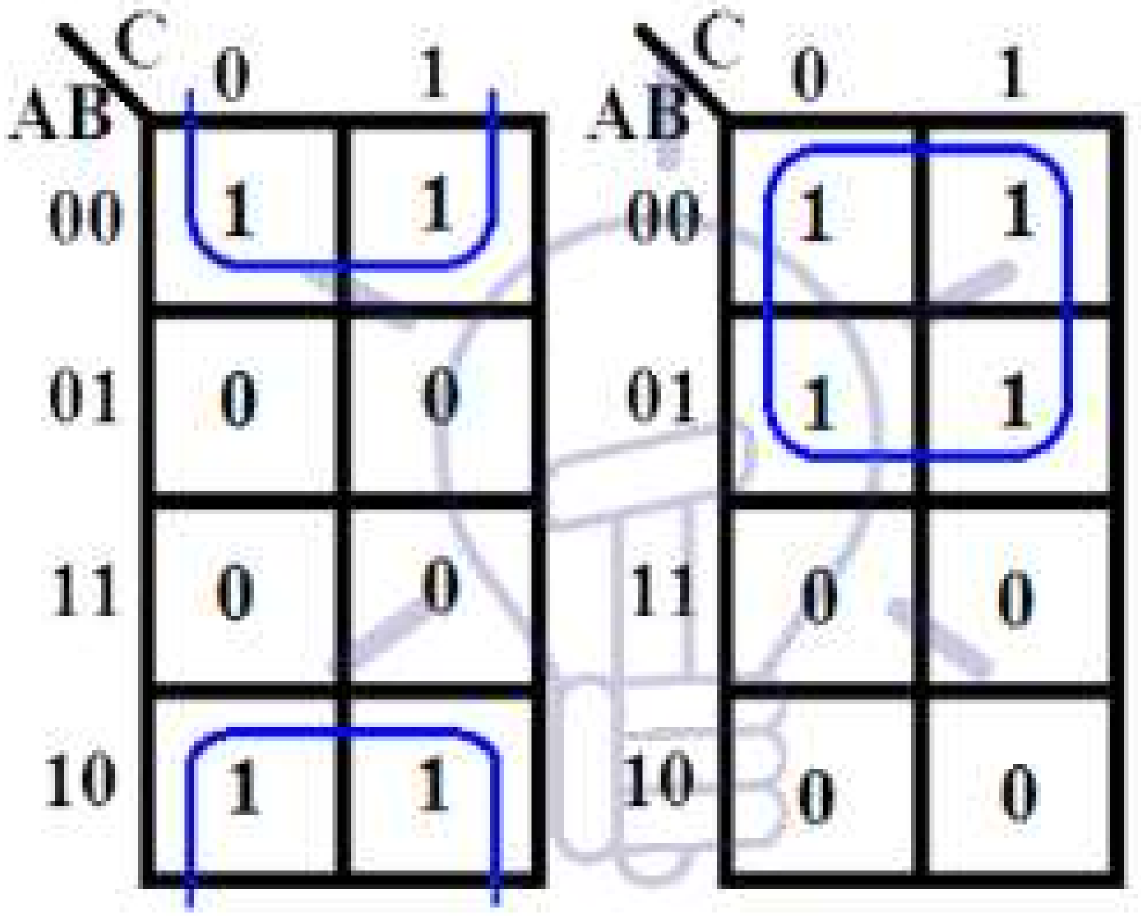
Find Double Overlap





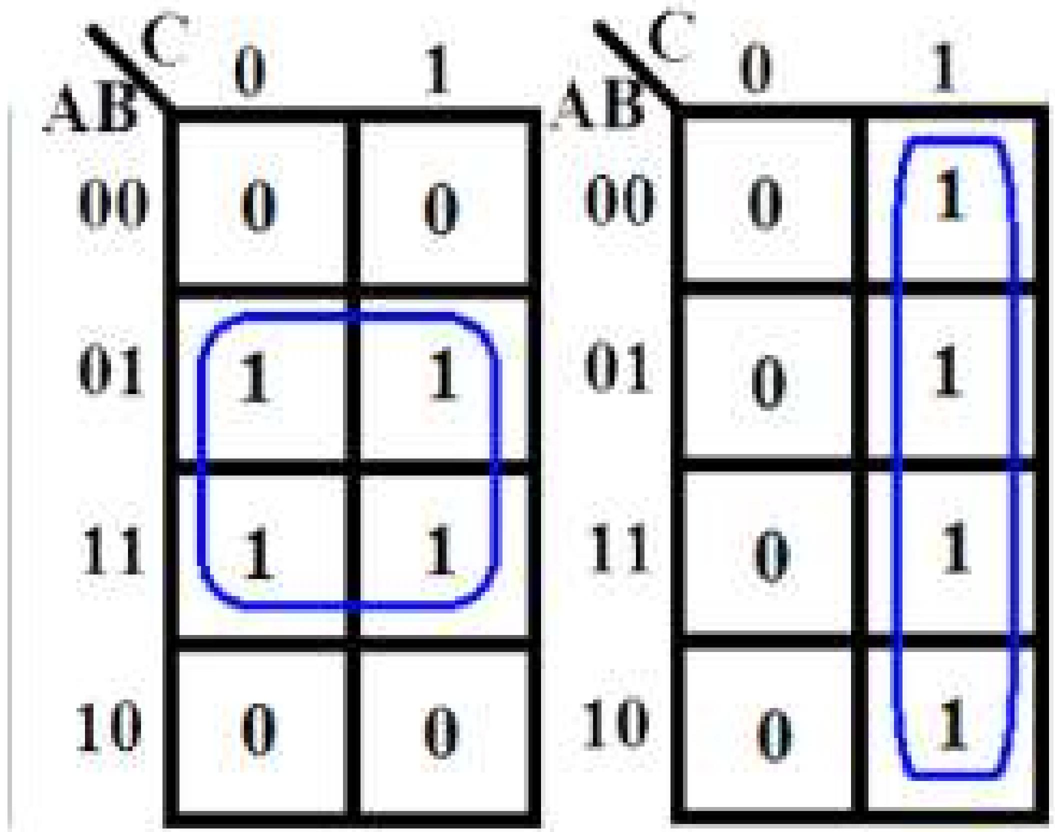
K-map Simplification of SOP Examples

Find Double Overlap



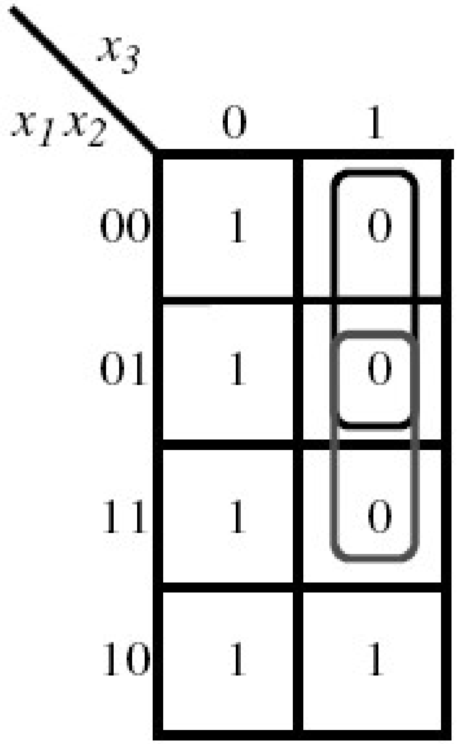
K-map Simplification of SOP Examples

Find Double Overlap



K-map Simplification of POS Examples

Find Double Overlap



$$f = (\bar{x}_2 + \bar{x}_3) \cdot (x_1 + \bar{x}_3)$$

K-Map POS Minimization

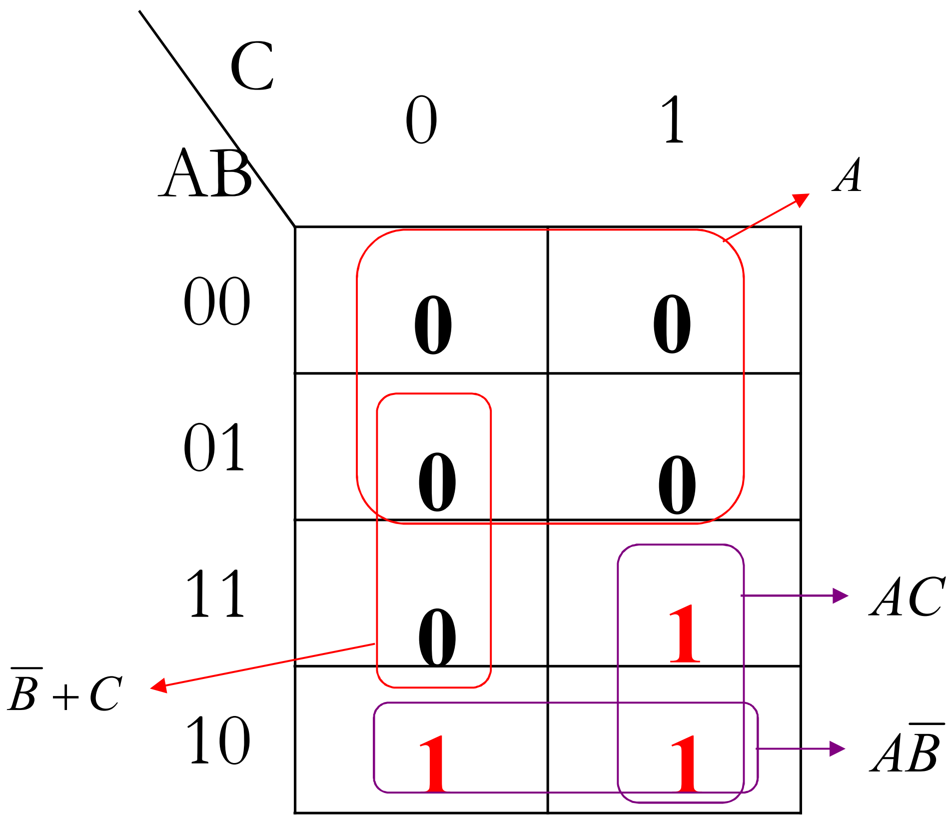
- The approaches are much the same (as SOP) except that with POS expression, 0s representing the standard SUM terms are placed and grouped on the K-map instead of 1s.

3 Variables K-Map	
cells	variables
1	3
2	2
4	1
8	Expression =0

K-map Simplification of POS Expression

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

Find Double Overlap

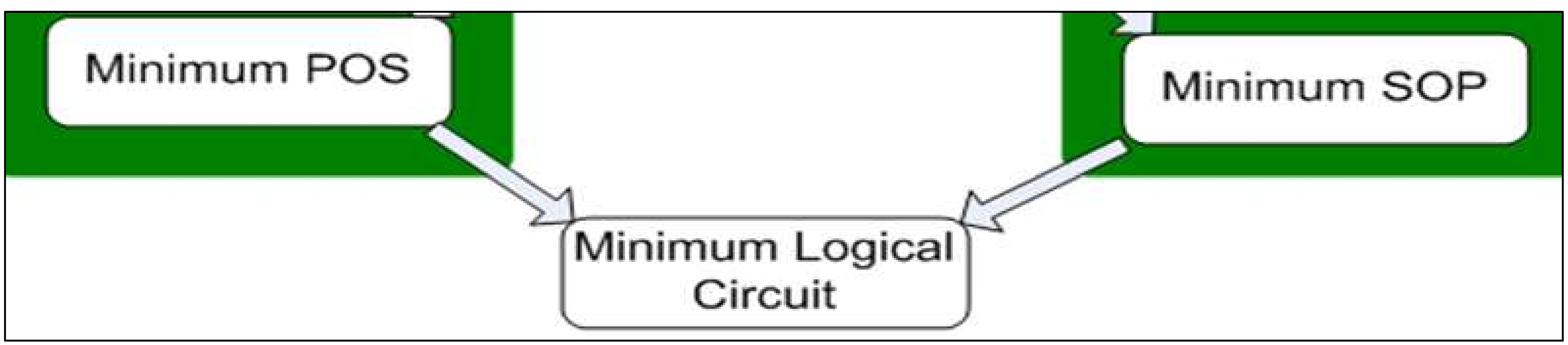


$A(\overline{B}+C)$

$A\overline{B}+AC$

Drawing Minimum Logical Circuit

- For minimum SOP expression combine each product term using AND gate and then mix all the outputs of the AND gates with big OR gate.
- For minimum POS expression combine each sum expression using OR gate and then mix all the outputs of the OR gates with big AND gate

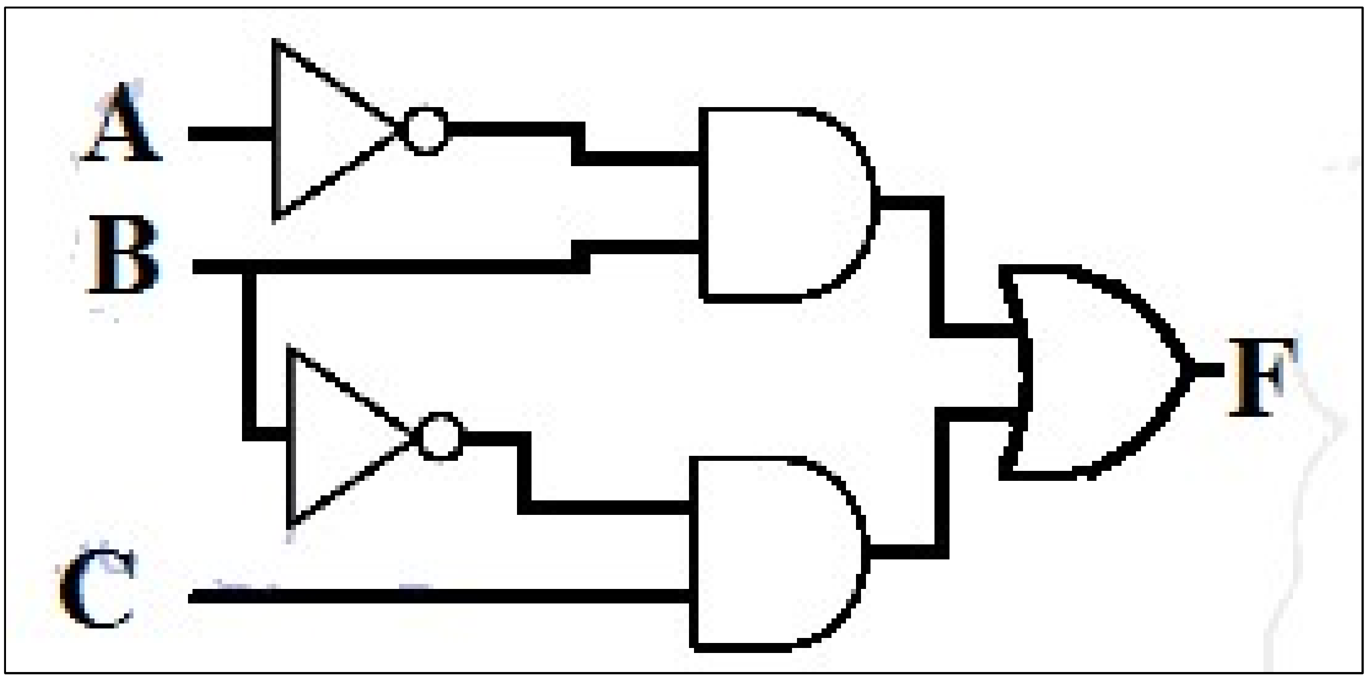


Drawing a circuit Examples

Example: Draw the logical circuit for below minimum SOP using AND-OR gates

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

Solution

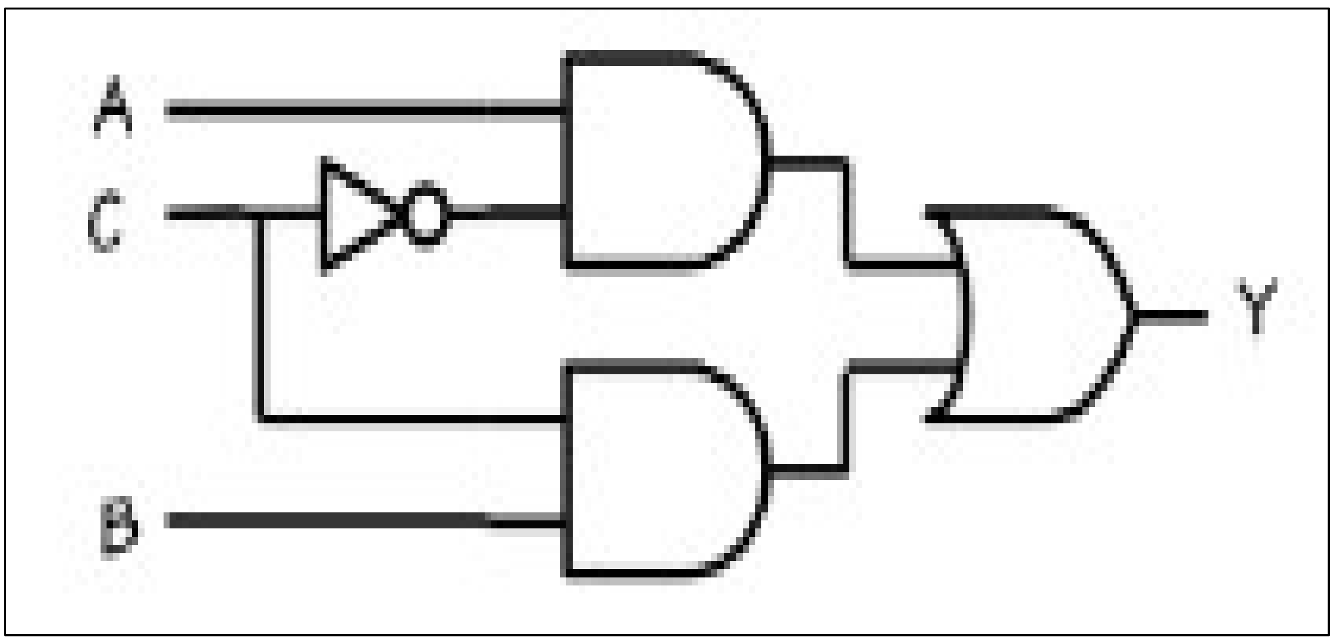


Drawing a circuit Examples

Example: Draw the logical circuit for below minimum SOP using AND-OR gates

$$Y = \bar{C}A + CB$$

Solution

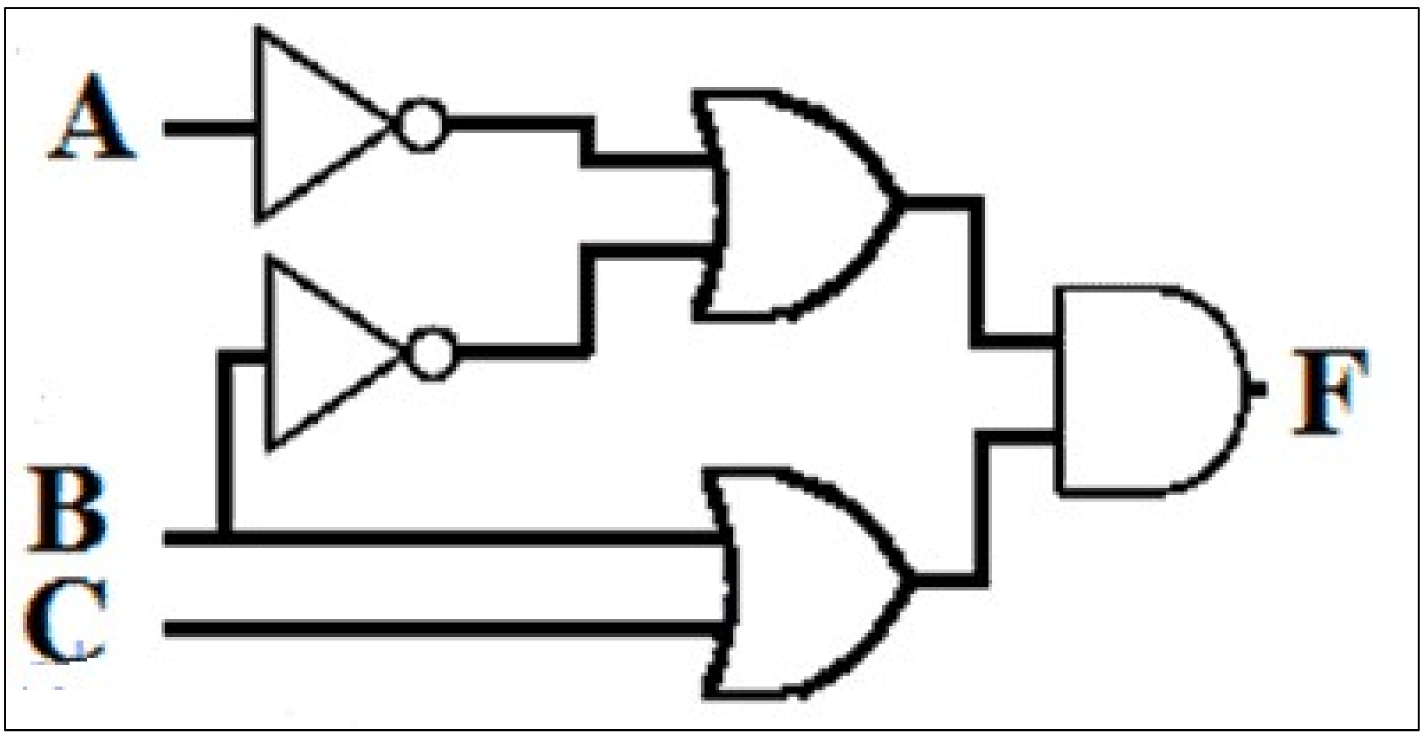


Drawing a circuit Examples

Example: Draw the logical circuit for below minimum POS using OR-AND gates

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

Soluti



Drawing a circuit Examples

Example: Draw the logical circuit for below minimum POS using OR-AND gates

$$F = (A + B) * (A + \overline{B})$$

Solution