



4b:

Logic Simplification



Simplification using Boolean Algebra

Simplification using Boolean algebra

- Simplification:
 - To reduce an expression to its simplest form
 - To change a form to a more convenient one for efficient implementation
- We use:-
 - basic laws, rules, theorems of Boolean algebra
- Practice makes perfect!

Example:

Simplify this expression

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

$$AB + \textcolor{red}{A(B + C)} + B(B + C)$$

(Page: 104)

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

Step 1: Apply distributive law to the red terms

$$AB + \textcolor{red}{AB} + AC + \textcolor{green}{BB} + BC$$

Step 2: Apply rule 7 ($BB = B$) to the green term

$$\textcolor{red}{AB} + AB + AC + B + BC$$

Step 3: Apply rule 5 ($AB + AB = AB$) to the red terms

$$AB + AC + \textcolor{green}{B} + BC$$

Step 4: Apply rule 10 ($B + BC = B$) to the green terms

$$\textcolor{red}{AB} + AC + B$$

Step 5: Apply rule 10 ($B + AB = B$) to the red terms

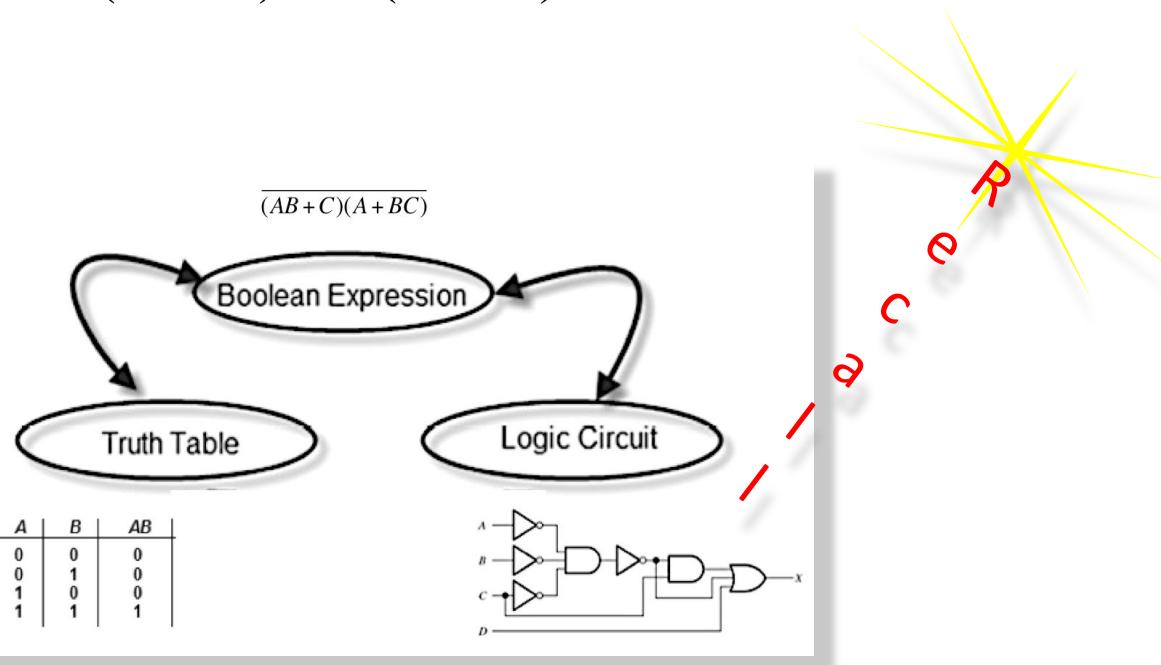
$$AC + B$$

Exercise 4b.1:

According to the example given before, draw the logic circuit for the original expression and the last expression simplified.

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

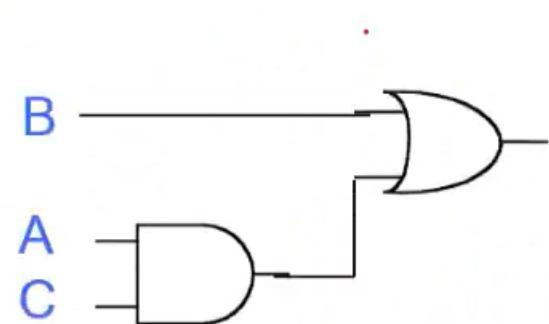
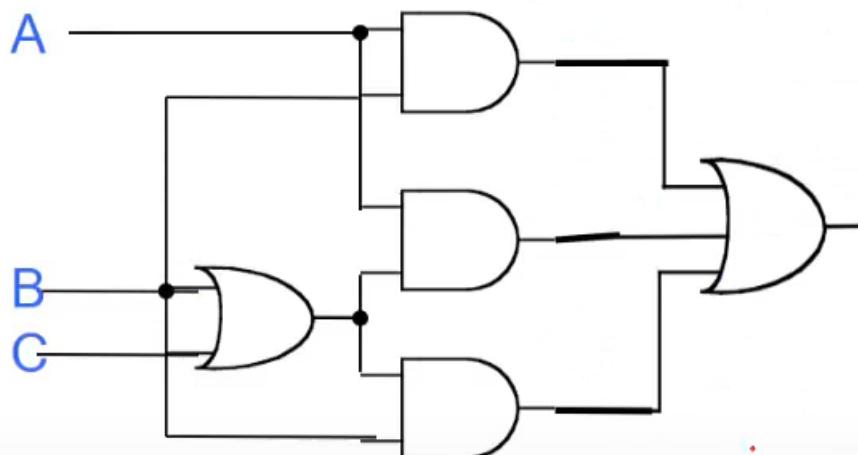
$$B + AC$$



Solution 4b.1:

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

After: $B + AC$



Simplify this expression

Example:

$$\bar{A}BC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + A\bar{B}C + ABC$$

$$\bar{A}BC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + A\bar{B}C + ABC$$

Step 1: Factor BC for the red terms

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

Step 2: Apply rule 6 to the green term and factor the blue term

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

Step 3: Apply rule 4 to the red term and rule 6 to the blue term

$$BC + AB \cdot 1 + \bar{A}\bar{B}\bar{C}$$

Step 4: Apply rule 4 to the green term

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

Step 5: Factor the red terms

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

Step 6: Apply rule 11 to the blue term

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

Step 7: Use the distributive and commutative laws to get the following expression

$$BC + AB + \bar{B}\bar{C}$$

(Page: 104)

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

 Image Upload



Home
work

Exercise 4b.2: According to previous example, draw the logic circuit for the original expression and the simplified expression.

Original expression:

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

Simplified expression:

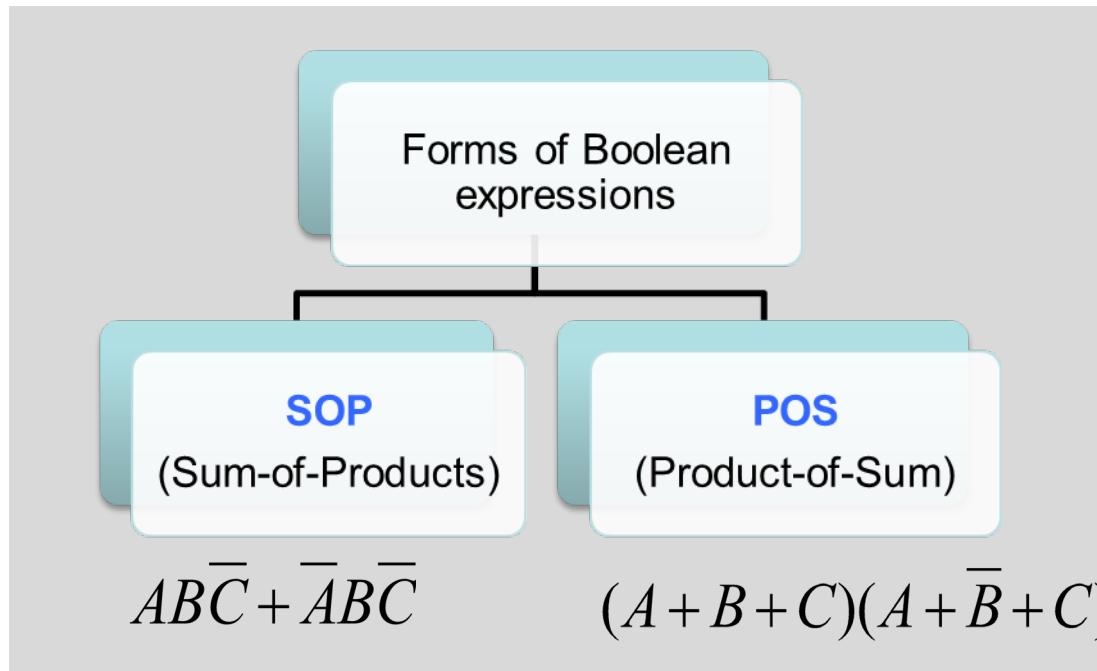
$$BC + A\overline{B} + \overline{B}\overline{C}$$



Forms of Boolean Expressions

Forms of Boolean expressions

- Boolean expression can be converted into one of 2 forms.



Product term = a term with the product (Boolean multiplication) of literals

Sum term = a term with the sum (Boolean addition) of literals

(a) The Sum-of-Product (SOP) form

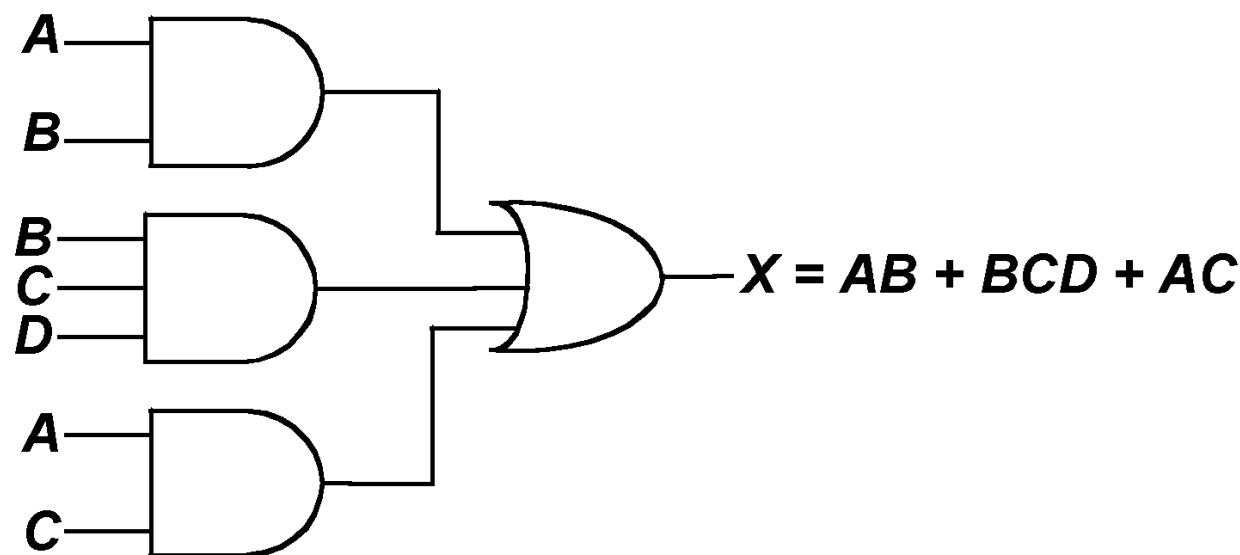
- **SOP** = when 2 or more product terms are summed.
- e.g. $AB_{P1} + ABC_{P2}$
 $ABC_{P1} + CDE_{P2} + BCD_{P3}$
- SOP can also contain a single variable term

- In SOP a single overbar cannot extend over more than 1 variable, but more than 1 variable can have an overbar.

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

\overline{ABC}

- Circuit implementation of the SOP expression
 $AB + BCD + AC$



Exercise 4b.3:

Convert each of the following Boolean expressions to SOP form:

$$(i) \quad AB + B(CD + EF)$$

$$(ii) \quad (A + B)(B + C + D)$$

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

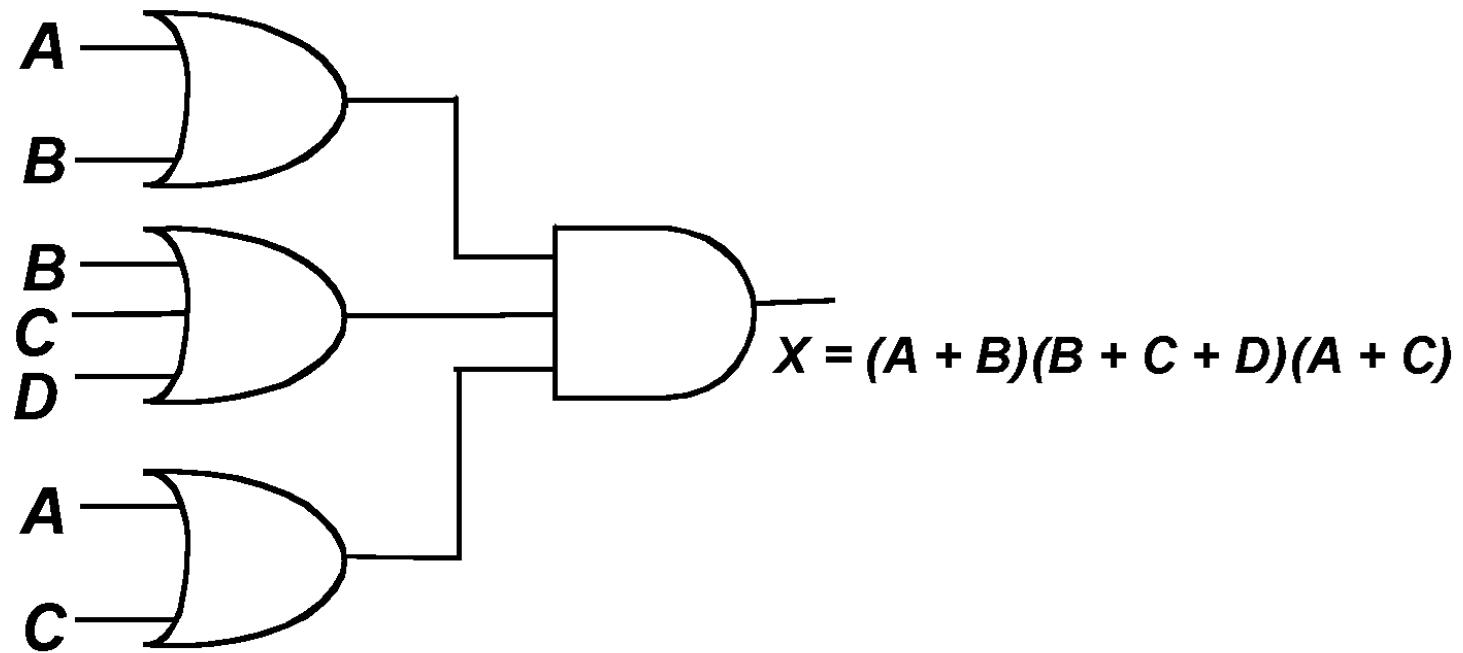
(b) The Product-of-Sum (POS) form

- **POS** = when 2 or more sum terms are multiplied.
 - $(A + B)_{S1}(A + B + C)_{S2}$
 - $(A + B + C)_{S1}(C + D + E)_{S2}(B + C + D)_{S3}$
- Like SOP, POS
 - can also contain a single variable term
 - a single overbar cannot extend over more than 1 variable, but more than 1 variable can have an overbar.

$$\overline{A} + \overline{B} + \overline{C} \quad \checkmark \quad \overline{\overline{A + B + C}} \quad \times$$

- Circuit implementation of the POS expression

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



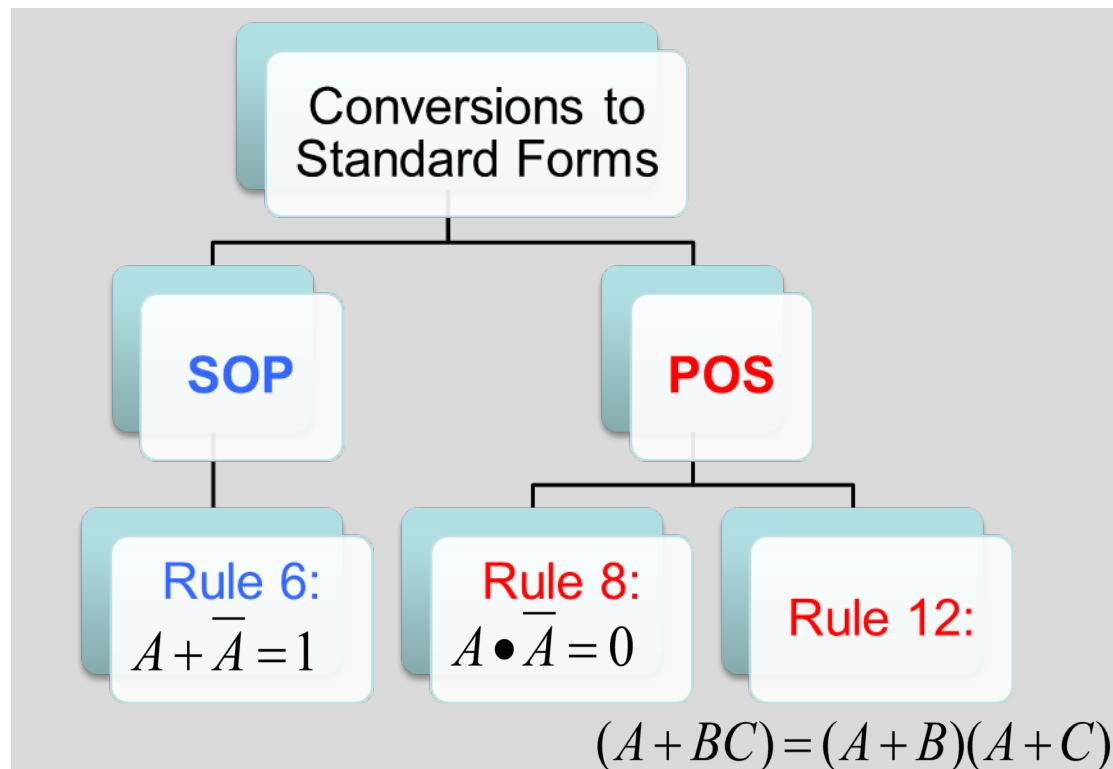


Standard of Boolean Expressions

Standard forms of Boolean expressions

- Standardization makes the evaluation, simplification, and implementation of Boolean expressions more systematic and easier.

- A standard **SOP** / **POS** form is derived when ALL the variables appear in each **product term** / **sum term** of the expression.



(a) Product term → Standard SOP

- A logic expression can be changed to SOP form using Boolean algebra techniques.
 - $A(B + CD) =$
 - $AB + B(CD + EF) =$
- Standard SOP form is when all the variables appear in each product term in the expression.

$$A\bar{B}CD + \bar{A}\bar{B}C\bar{D} + A\bar{B}C\bar{D}$$

- To convert product terms to standard SOP
 - Multiply each of the nonstandard term with the missing term using Boolean algebra **Rule 6: $(A + \bar{A})=1$**
 - Repeat until all variables appear in each product term.

Example:

- Convert this Boolean expression to standard SOP form:

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

Term 1 *Term 2* *Term 3*

Solution:

- Variables = A, B, C, D.
- What is missing?
 - Term 1: missing D
 - Term 2: missing C and D
- Complete these terms by applying Boolean rule 6

$$A\bar{B}C + \bar{A}\bar{B} + AB\bar{C}D$$

 Term 1 Term 2

Rule 6: $(A + \bar{A})=1$

Term 1: $A\bar{B}C = A\bar{B}C(D + \bar{D}) = A\bar{B}CD + A\bar{B}C\bar{D}$

Term 2 : $\bar{A}\bar{B} = AB(C + \bar{C}) = \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}$
 $= \bar{A}\bar{B}C(D + \bar{D}) + \bar{A}\bar{B}\bar{C}(D + \bar{D}) = \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D}$

- Now we have the standard of SOP expression:

$$\begin{aligned}
 & A\bar{B}C + \bar{A}\bar{B} + AB\bar{C}D \\
 &= \boxed{A\bar{B}CD + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}\bar{B}\bar{C}D + AB\bar{C}D}
 \end{aligned}$$

Exercise 4b.4:

Define the variables of SOP expression $A\bar{C} + B\bar{C}$ and convert the expression to standard SOP form.

Solution 4b.4:

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

$$\begin{aligned} A\bar{C} + B\bar{C} &= A\bar{C}(B + \bar{B}) + B\bar{C}(A + \bar{A}) && \text{(Apply rule 6)} \\ &= A\bar{B}\bar{C} + A\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} && \text{(Apply rule 5)} \\ &= A\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} \end{aligned}$$

Binary representation of a Standard SOP

- A **product = 1** only if ALL variables in the term is equal to **1**.

– Remember:

product = multiplication $\rightarrow 1 \cdot 1 = 1$

$$ABC + \bar{A}B\bar{C} = 1$$

- A **sum = 1** when ONE or ALL of the variables in the term is equal to **1**.

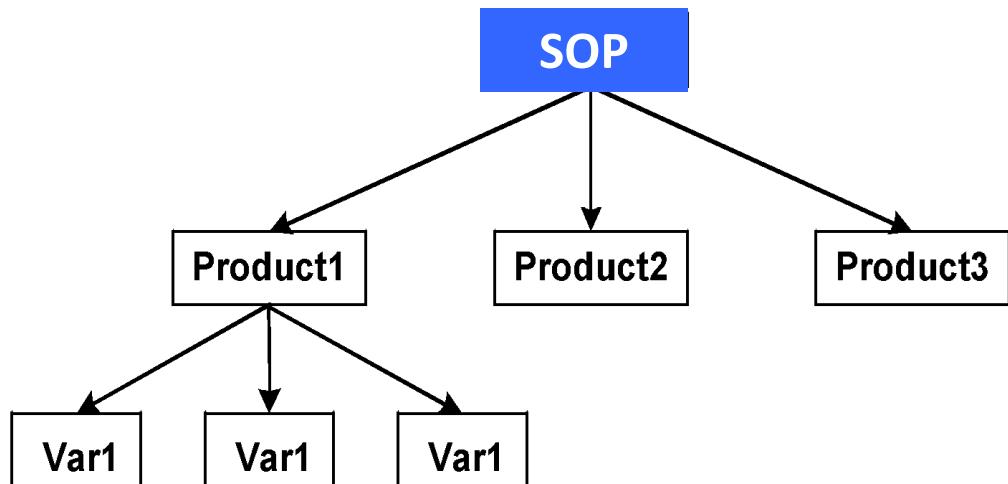
– Remember:

sum = addition

$$1 + 0 = 1;$$

$$0 + 1 = 1;$$

$$1 + 1 = 1$$



Example:

Determine the binary value for which the following standard SOP expression is equal to 1:

$$ABCD + A\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D}$$

The term $ABCD$ is equal to 1 when $A = 1$, $B = 1$, $C = 1$, and $D = 1$

$$ABCD = 1.1.1.1 = 1$$

The term $A\bar{B}\bar{C}D$ is equal to 1 when $A = 1$, $B = 0$, $C = 0$, and $D = 1$

$$A\bar{B}\bar{C}D = 1.\bar{0}.\bar{0}.1 = 1.1.1.1 = 1$$

The term $\bar{A}\bar{B}\bar{C}\bar{D}$ is equal to 1 when $A = 0$, $B = 0$, $C = 0$, and $D = 0$

$$\bar{A}\bar{B}\bar{C}\bar{D} = \bar{0}.\bar{0}.\bar{0}.\bar{0} = 1.1.1.1 = 1$$

The SOP expression equals 1 when any or all of the three product terms is 1.

From the standard SOP
it is easy to generate
the truth table.

$$ABCD + A\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D}$$

$$ABCD$$

1 1 1 1

$$\bar{A}\bar{B}\bar{C}D$$

1 0 0 1

$$\bar{A}\bar{B}\bar{C}\bar{D}$$

0 0 0 0

INPUT				OUTPUT
A	B	C	D	F
0	0	0	0	1
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	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(b) Sum term → Standard POS

- Standard POS form = where all the variables appear in each **sum term** in the expression.
- To **convert** product terms to **standard POS**
 - Multiply each of the nonstandard term with the missing term using Boolean algebra **Rule 8: $(A \cdot \bar{A}) = 0$**
 - Apply **Rule 12: $(A + BC) = (A + B)(A + C)$**
 - Repeat until all variables appear in each sum term.

Example:

- Convert this Boolean expression to standard POS form

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

Term 1 *Term 2* *Term 3*

The diagram shows the Boolean expression $(A + \overline{B} + C)(\overline{B} + C + \overline{D})(A + \overline{B} + \overline{C} + D)$. Three red brackets below the expression are labeled *Term 1*, *Term 2*, and *Term 3* from left to right. Red arrows point from these labels to the missing variables in each term: *Term 1* points to \overline{D} , *Term 2* points to A , and *Term 3* points to C .

Solution:

- Variables = A, B, C, D.
- What is missing?
 - Term 1: missing D
 - Term 2: missing A

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

Term 1 Term 2

- Apply rules 8 and 12

Rule 8: $(A \cdot \bar{A}) = 0$

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

$$\text{Term 1: } A + \overline{B} + C = A + \overline{B} + C + \boxed{D\overline{D}} = (A + \overline{B} + C + \boxed{D})(A + \overline{B} + C + \boxed{\overline{D}})$$

$$\begin{aligned}\text{Term 2: } \overline{B} + C + \overline{D} &= \overline{B} + C + \overline{D} + \boxed{A\overline{A}} \\ &= (\boxed{A} + \overline{B} + C + \overline{D})(\boxed{\overline{A}} + \overline{B} + C + \overline{D})\end{aligned}$$

- Now we have the standard of POS expression:

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

Exercise 4b.5:

Convert the following Boolean expression to standard POS form:

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

Rule 8: $(A \cdot \bar{A}) = 0$

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

Solution 4b.5:

$$\begin{aligned}(A + B)(\bar{B} + C) \\ &= (A + B + C \cdot \bar{C})(A \cdot \bar{A} + \bar{B} + C) \\ &= (A + B + C)(A + B + \bar{C})(A + \bar{B} + C)(\bar{A} + \bar{B} + C)\end{aligned}$$

Binary representation of a Standard POS

- A product = 0 only if ONE or MORE of the sum term is equal to 0.

– Remember:

product = multiplication $\rightarrow 1 \cdot 0 = 0$

- A sum = 1 when ONE or ALL of the variables in the term is equal to 1.

– Remember:

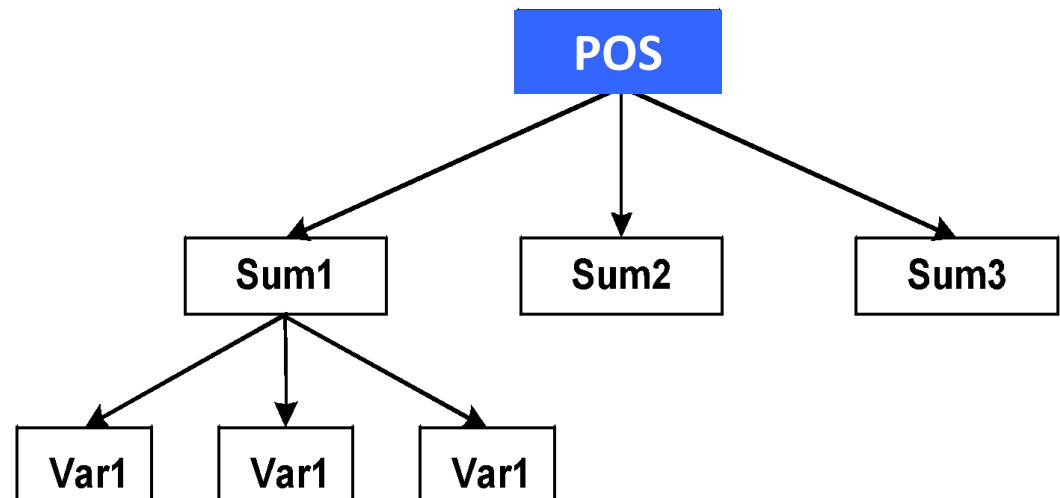
sum = addition

$$1 + 0 = 1;$$

$$0 + 1 = 1;$$

$$1 + 1 = 1$$

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



Example: Determine the binary value for which following standard POS expression is equal to **0**

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

Solution:

$$(A + B + C + D) = (0 + 0 + 0 + 0) = 0; \quad A=0, B=0, C=0, D=0$$

$$(A + \overline{B} + \overline{C} + D) = (0 + \overline{1} + \overline{1} + 0) = 0; \quad A=0, B=1, C=1, D=0$$

$$(\overline{A} + \overline{B} + \overline{C} + \overline{D}) = (\overline{1} + \overline{1} + \overline{1} + \overline{1}) = 0; \quad A=1, B=1, C=1, D=1$$

The **POS** expression equal **0** when **ALL of the terms** are **0**.

From the standard POS,
 $(A + B + C + D)(A + \bar{B} + \bar{C} + D)(\bar{A} + \bar{B} + \bar{C} + \bar{D})$
it is easy to generate the truth table.

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

0 0 0 0

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

0 1 1 0

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

1 1 1 1



INPUT				OUTPUT
A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0



Representation of Boolean Expressions



(Refer to module v5 page 122)

Boolean in SOP form can be represented as

1. Boolean expression , e.g. $X = \bar{A}\bar{B}\bar{C} + A\bar{B}C + ABC$
2. By using a sigma notation , e.g. $X = \sum_{ABC} (2,5,7)$

SOP

$$\begin{aligned} A &= 1 \\ \bar{A} &= 0 \end{aligned}$$

Determine the terms that 2, 5 and 7 represents?

$$\bar{A}\bar{B}\bar{C} = 010$$

$$A\bar{B}C = 101$$

$$ABC = 111$$



(Refer to module v5 page 122)

Boolean in POS form can be represented as

1. Boolean expression , e.g. $X = (A + B + \bar{C})(\bar{A} + B + C)(\bar{A} + \bar{B} + C)$
2. By using a PI notation , e.g. $X = \prod_{ABC} (1, 4, 6)$

POS

$$\begin{aligned}\bar{A} &= 1 \\ A &= 0\end{aligned}$$

Determine the terms that 1, 4 and 6 represent?

$$\begin{array}{lll}001 = (A + B + \bar{C}) & 100 = (\bar{A} + B + C) & 110 = (\bar{A} + \bar{B} + C)\end{array}$$

Exercise 4b.6: Represent the following Boolean expression:

(i) $A\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$ as a sigma notation

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

as a PI notation.

Solution 4b.6(i):

Expression:

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

110 100 010

Sigma notation:

$$\sum_{ABC} (6, 4, 2)$$

Exercise 4b.6: Represent the following Boolean expression:

(i) $A\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$ as a sigma notation

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

as a PI notation.

Solution 4b.6(ii):

Expression: $(A + B + C)(A + B + \bar{C})(A + \bar{B} + C)(\bar{A} + B + C)$

000 001 010 100

PI notation:

$$\prod_{ABC} (0, 1, 2, 4)$$

Exercise 4b.7:

A Boolean expression is written in sigma notation as $X = \sum_{ABC}(7, 4, 3)$
Determine the logic level (binary value) for each product term and write whole expression.

Solution 4b.7:

$$\sum_{ABC}(7, 4, 3)$$

Logic level: 111 100 011

Expression:
(SOP)

$$ABC + A\bar{B}\bar{C} + \bar{A}BC$$

Exercise 4b.8:

A Boolean expression is written in PI notation as $X =$

$$\prod_{ABC} (7, 4, 3)$$

Determine the logic level (binary value) for each sum term and write whole expression.

Solution 4b.8:

$$\prod_{ABC} (7, 4, 3)$$

Logic level:

111

100

011

Expression:
(POS)

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



Converting Standard of Boolean Expressions (SOP \leftrightarrow POS)



Converting Standard (SOP \leftrightarrow POS)

- **Step 1:** Evaluate each product term in the **SOP** expression → i.e., determine the **binary** numbers of the **product terms**.
 - **Step 2:** Determine all the binary numbers not included in Step 1.
 - **Step 3:** Write equivalent **sum term** for each binary number from Step 2 and express in **POS form**.
- ** Using a similar procedure, to go from POS to SOP

Example: Convert the following SOP expression to an equivalent POS expression.

$$\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C + ABC$$

Solution:

- Variables = (A, B, C) = 3. So, $2^3 = 8$ possible combinations.
- The SOP has 5 of 8 from the input combination

$$\begin{array}{ccccc} \bar{A}\bar{B}\bar{C} & + & \bar{A}B\bar{C} & + & \bar{A}BC & + & A\bar{B}C & + & ABC \\ 000 & & 010 & & 011 & & 101 & & 111 \\ 0 & & 2 & & 3 & & 5 & & 7 \end{array}$$

- Convert the above SOP expression to binary equivalent.

A	B	C	Output
0	0	0	1
0	0	1	
0	1	0	1
0	1	1	1
1	0	0	
1	0	1	1
1	1	0	
1	1	1	1

Example: Convert the following SOP expression to an equivalent POS expression.

$$\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C + ABC$$

Solution:

- The POS has the other 3 combinations which are 001, 100, 110; and this make sum term = 0
- These binaries combination (001, 100, 110) can create equivalent POS expression:

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

0	0	1	1	0	0	1	1	0
---	---	---	---	---	---	---	---	---

A	B	C	Output
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Exercise 4b.9:

Convert the following SOP expression to an equivalent POS expression:

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

Solution 4b.9:

- **Step 1:** Need to convert the expression into **standard SOP** (refer [Exercise 4b.4](#))

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

- **Step 2:** **Binary** number for each SOP term.

Variables = 3 (A, B, C); $2^3 = 8$ possible combinations.

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

1 1 0 1 0 0 0 1 0

(has 3 combinations)

... previous: $A\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}BC$
1 1 0 1 0 0 0 1 0

- Step 3: Equivalent sum term for each binary number (i.e., to get POS expression)

0 0 0	0 0 1	0 1 1	1 0 1	1 1 1	(has 5 combinations)
$(A + B + C) (A + B + \bar{C}) (A + \bar{B} + \bar{C}) (\bar{A} + B + \bar{C}) (\bar{A} + \bar{B} + \bar{C})$					the POS expression

\Updownarrow

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



Boolean Expressions and Truth Tables

Standard SOP → truth tables

Develop a truth table for the standard SOP expression

Example : $\bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$

Boolean expressions and truth tables

- **Step 1:** determine variables and combinations of binary values → input
- **Step 2:** convert expression to Standard SOP / POS.
- **Step 3:** find the binary values that make the
 - product terms = 1 (SOP) or
 - sum terms = 0 (POS)
- **Step 4:** the remaining combination will be
 - Equal to 0 (POS)
 - Equal to 1 (SOP)
- Fill in the truth table

Standard SOP → truth tables

Example : Develop a truth table for the standard SOP expression

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

Solution

- Variables = A, B, C. combinations = $2^3 = 8$
- What binary value makes the **product terms = 1?**

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

$$0\ 0\ 1 = 1\ 1\ 1 = 1$$

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

$$1\ 0\ 0 = 1\ 1\ 1 = 1$$

$$ABC$$

$$1\ 1\ 1 = 1\ 1\ 1 = 1$$

INPUTS			OUTPUT	PRODUCT TERM
A	B	C	X	
0	0	0	0	
0	0	1	1	$\bar{A}\bar{B}C$
0	1	0	0	
0	1	1	0	
1	0	0	1	$A\bar{B}\bar{C}$
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

- Fill the truth table

Standard POS → truth tables

Example: Develop a truth table for the standard POS expression

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

Solution

- Variables = A, B, C. combinations = $2^3 = 8$
- What binary value makes the sum terms = 0?

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

- Fill the truth table

INPUTS			OUTPUT	PRODUCT TERM
A	B	C	X	
0	0	0	0	$(A + B + C)$
0	0	1	1	
0	1	0	0	$(A + \bar{B} + C)$
0	1	1	0	$(A + \bar{B} + \bar{C})$
1	0	0	1	
1	0	1	0	$(\bar{A} + B + \bar{C})$
1	1	0	0	$(\bar{A} + \bar{B} + C)$
1	1	1	1	



Determining Standard expression from a truth table

- Replace each binary number with its corresponding variable.
SOP = 1 → var. & 0 → var. complement
POS = 0 → var. & 1 → var. complement
- Example:

SOP

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

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

$$= 1 \cdot 1 \cdot 1 \cdot 1 = 1$$

POS

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

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

$$= 0 + 0 + 0 + 0 = 0$$

Example:

From the truth table,
determine:

- (i) standard SOP expression
- (ii) equivalent standard POS expression

INPUT			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

Convert the binary values to product terms (SOP).

INPUT			OUTPUT	PRODUCT TERMS
A	B	C	X	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	→ $\bar{A}BC$
1	0	0	1	→ $A\bar{B}\bar{C}$
1	0	1	0	
1	1	0	1	→ $AB\bar{C}$
1	1	1	1	→ ABC

Standard SOP expression:

$$\bar{A}BC + A\bar{B}\bar{C} + AB\bar{C} + ABC$$

Solution

Convert the binary values to sum terms (POS).

INPUT			OUTPUT	SUM TERMS
A	B	C	X	
0	0	0	0	$(A + B + C)$
0	0	1	0	$(A + B + \bar{C})$
0	1	0	0	$(A + \bar{B} + C)$
0	1	1	1	
1	0	0	1	
1	0	1	0	$(\bar{A} + B + \bar{C})$
1	1	0	1	
1	1	1	1	

Standard POS expression:

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



- ✓ Introduction to K-Map (Karnaugh Map)
- ✓ K-Map → SOP expression Simplification
- ✓ K-Map → POS expression Simplification



- ✓ Introduction to K-Map (Karnaugh Map)
- ✓ K-Map → SOP expression Simplification
- ✓ K-Map → POS expression Simplification

What is Karnaugh Map (K-Map)?

- Special form of truth table which enables easier pattern recognition
- Pictorial method of simplifying Boolean expressions
- Good for circuit designs with up to 4 variables

Karnaugh Map (K-Map)

- K-Map is similar to the truth table, but it **presents all the possible values of input and output**.
- This is shown in an **array of cells**.
- K-Maps can be used for expressions with **2, 3, 4 or 5** variables.

C AB	0	1
	00	
11		
	10	

3 variables K-Map

CD AB	00	01	11	10
	00			
11				
	10			

4 variables K-Map

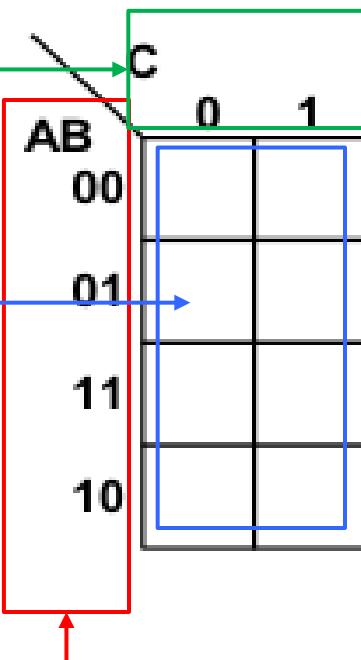
- The number of cells in a K-Map = total number of possible input variable combinations as in truth table

$$F = A + BC$$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Karnaugh Map (K-Map)

- The equivalent K-Map:

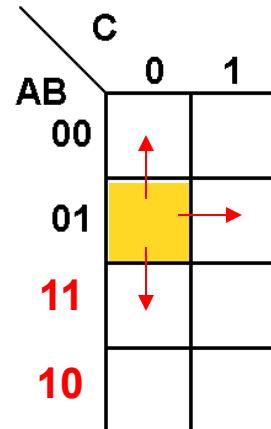


Example:

ABC, \rightarrow 3 variables $= 2^3 = 8$ cells.

- Physically, cells that share their walls are **adjacent**
- Cells that **differ by only one variable** are adjacent (*bersebelahan*)

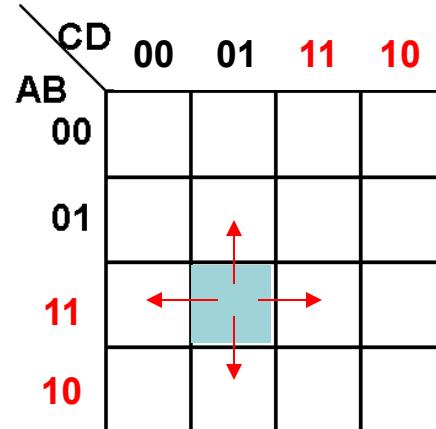
Cell **010** is adjacent to **000, 011** and **110**



	C	0	1
AB	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	$A\bar{B}\bar{C}$	$A\bar{B}C$

3-Variable Karnaugh Map

Cell **1101** is adjacent to **0101, 1011, 1100** and **1111**

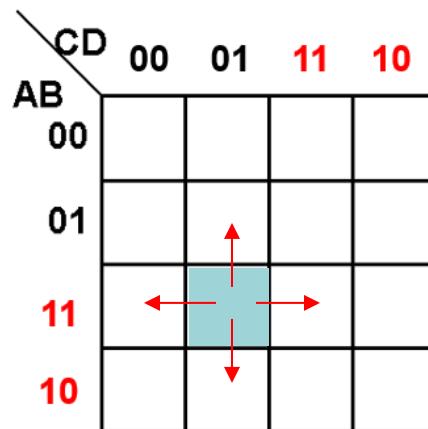


	CD	00	01	11	10
AB	00	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}CD$	$\bar{A}\bar{B}C\bar{D}$
	01	$\bar{A}B\bar{C}\bar{D}$	$\bar{A}B\bar{C}D$	$\bar{A}BC\bar{D}$	$\bar{A}BC\bar{D}$
	11	$AB\bar{C}\bar{D}$	$AB\bar{C}D$	$ABC\bar{D}$	$ABC\bar{D}$
	10	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}D$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

4-Variable Karnaugh Map

- Physically, cells that share their walls are **adjacent**
- Cells that **differ by only one variable** are adjacent (*bersebelahan*)

Cell **1101** is adjacent to **0101**, **1111**, **1001** and **1100**

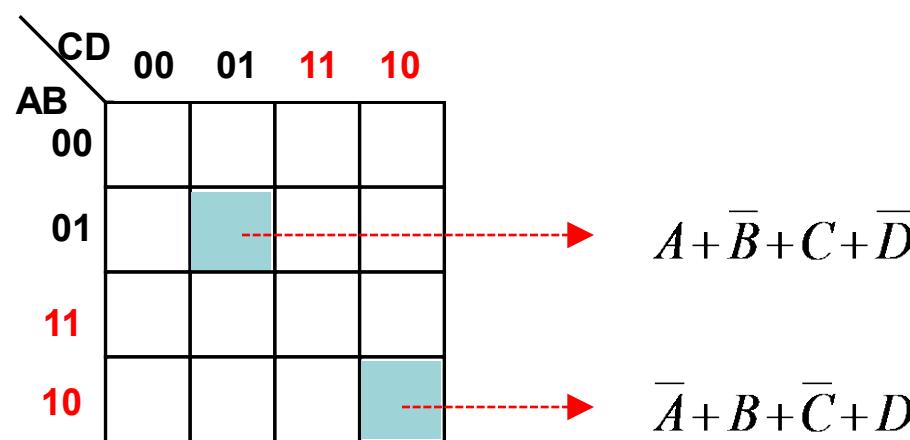
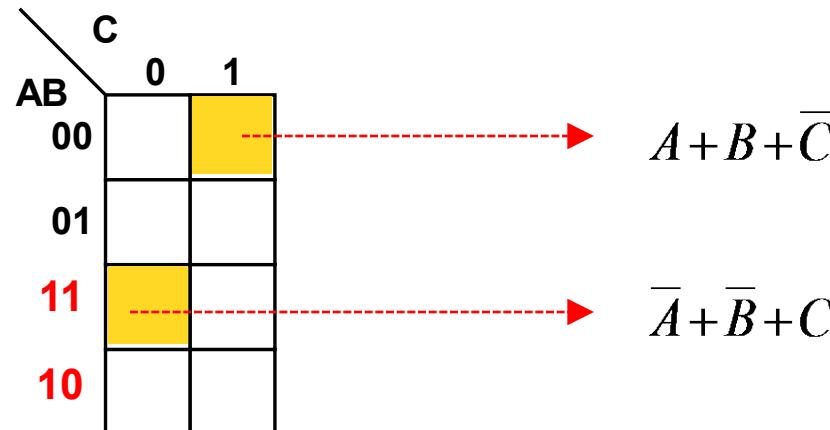


	CD \ AB	00	01	11	10
AB	00	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}CD$	$\bar{A}BC\bar{D}$
01	$\bar{A}B\bar{C}\bar{D}$	$\bar{A}BCD$	$\bar{A}BCD$	$\bar{A}BC\bar{D}$	
11	$A\bar{B}\bar{C}\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$	
10	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}CD$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

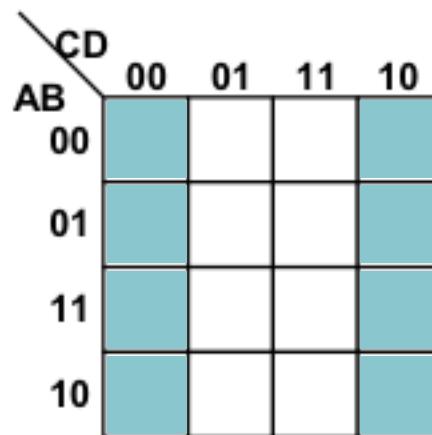
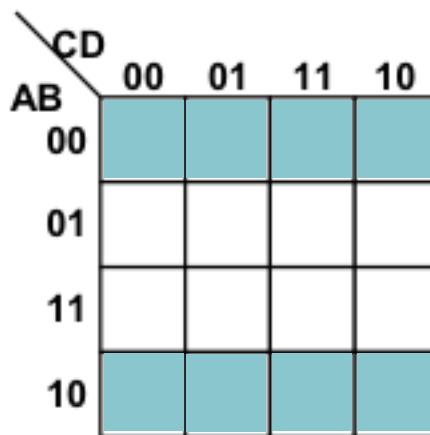
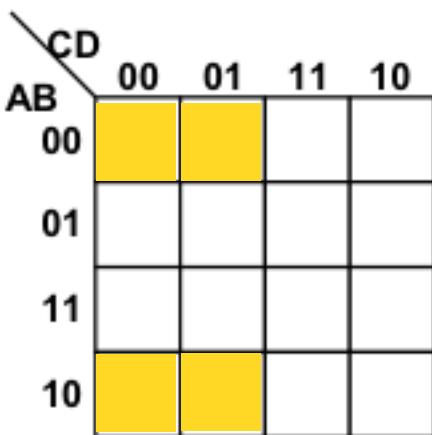
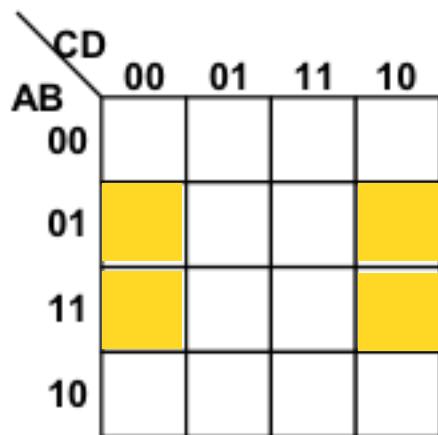
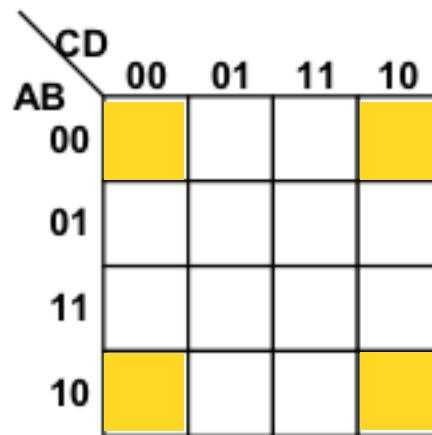
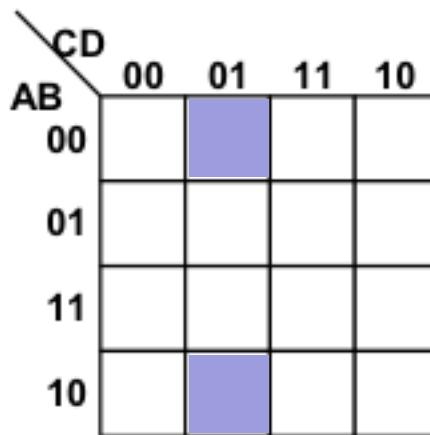
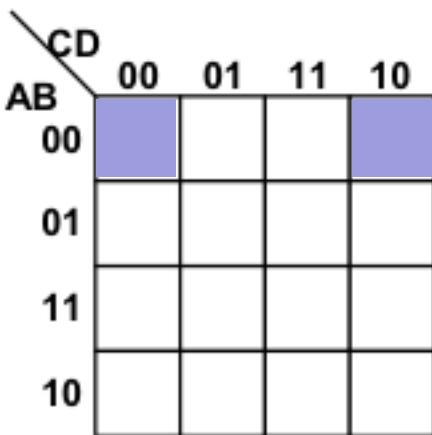
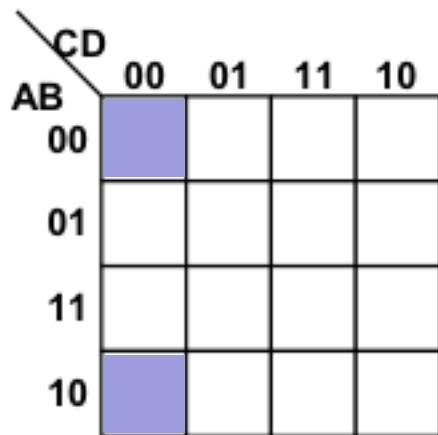
4-Variable Karnaugh Map

POS:

Get the **sum terms** for each colored cells for the following K-Map.



- In a K-map with 4-variable or more, the top-most & bottom-most cells of a column (and row) are adjacent.
 - (*recall → cells that differ by only one variable are adjacent*)



2 cells

4 cells

8 cells

Mapping Standard Expression on a K-Map

- K-Map is used to simplify Boolean expressions to their **minimum form**.
- A minimized expression :
 - has the **fewest** possible **term** with each term having **fewest** possible **variables**.
 - needs **fewer** logic **gates** than standard expression.
- To map an **SOP** / **POS** expression to a K-map:
 - **Step 1:** determine the binary value of each **product** / **sum term**
 - **Step 2:** Place a **1** / **0** in a cell that have the same value as the **product** / **sum term**

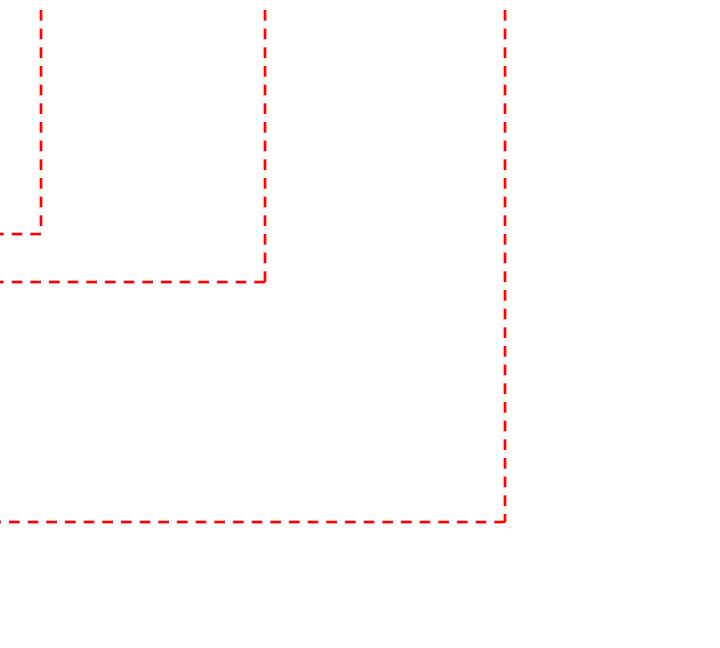
Example: 3 variables

- Step 1: determine the binary value of each product terms

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

0 0 0 0 0 1 1 1 0 1 0 0

AB	0	1
00	1	1
01		
11	1	
10	1	



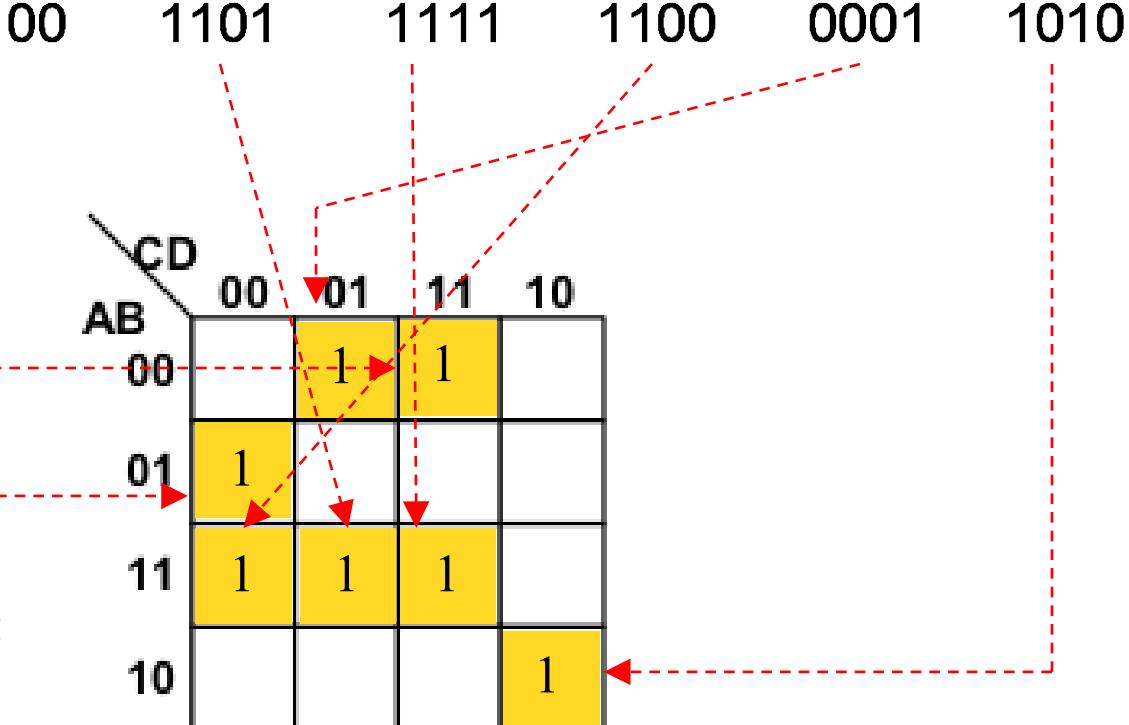
- Step 2: Place a 1 in a cell that have the same value as the binary product terms

Example: 4 variables

- Map the following expression

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

- Step 1:** determine the binary value of each product terms
- Step 2:** Place a 1 in a cell that have the same value as the product terms



Example: 3 variables

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

0	0	0	0	1	1	0	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

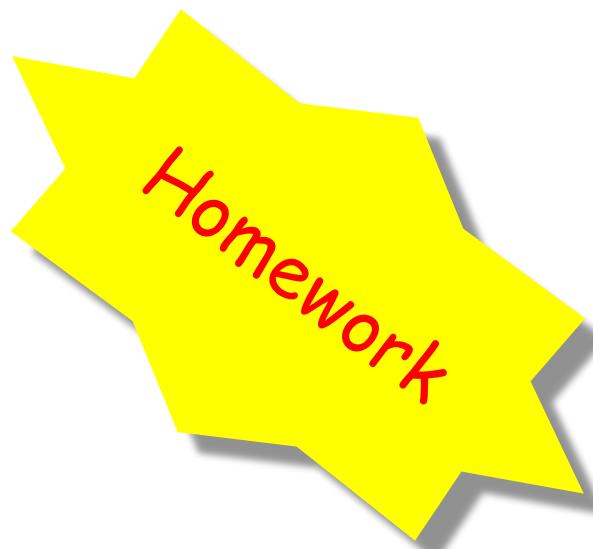
- **Step 1:** determine the binary value of each sum terms
- **Step 2:** Place a **0** in a cell that have the same value as the sum terms

A Karnaugh map for three variables A, B, and C. The columns represent the binary values of AB (00, 01, 10, 11) and the rows represent the binary values of C (0, 1). The map shows the minterms 000, 010, 110, and 111.

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

Exercise 4b.12: Map the following expression on a K-map.

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



Mapping Non-Standard Expression on a K-Map

- To use K-Maps, expressions must be in standard form.
- For expressions that are not standard, it must be converted to a standard form.
- Recall (SOP) : $A\bar{B} + ABC$

Expand numerically to standard form by writing the binary value of given variable & attach possible binary value for missing variables of the term.

$$A\bar{B} + ABC$$

Rule 6

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

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

1 0 1 1 0 0 1 1 1



1 0 0	1 1 1
1 0 1	

Mapping Non-Standard Expression on a K-Map

		C	
AB		0	1
00			
01			
11		1	
10	1	1	

Expand numerically to standard form by writing the binary value of given variable & attach possible binary value for missing variables the term.

$$A\bar{B} + ABC$$

$$\begin{array}{r} 100 \quad 111 \\ 101 \end{array}$$

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

Example:

Map the following SOP expression
on a K-Map:

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

Solution:

Expand the terms by including all
combinations of the missing variables
numerically as follow:

$$\begin{array}{ccccccc} \overline{B}\overline{C} & + & A\overline{B} & + & AB\overline{C} & + & A\overline{B}C\overline{D} + \overline{A}\overline{B}\overline{C}\overline{D} + A\overline{B}CD \\ 0000 & 1000 & 1100 & 1010 & 0001 & 1011 \\ \cancel{0001} & 1001 & 1101 \\ \cancel{1000} & \cancel{1010} \\ \cancel{1001} & \cancel{1011} & \text{(Discard all similar binary values)} \end{array}$$

Map each binary values by placing **1** in the corresponding cells of the 4-variables K-map

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

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

0000 1000 1100 1010 0001 1011
1001 1101

Example:

Map the following POS expression
on a K-Map:

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

Solution :

Expand the terms by including all combinations of the missing variables numerically for each sum term.

Example:

Map the following POS expression
on a K-Map:

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

Solution :

$$A + B + \textcolor{red}{C} + D$$

0 0 0 0
0 0 1 0

0 0 0 0
0 0 1 0

Example:

Map the following POS expression
on a K-Map:

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

Solution :

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

0 0 0 0

1 1 0 0

0 0 1 0

1 1 0 1

1 1 0 0

1 1 1 0

1 1 0 1

1 1 1 1

1 1 1 0

1 1 1 1

Example:

Map the following POS expression
on a K-Map:

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

Solution :

1 0 0 1

0 0 0 0

0 0 1 0

1 1 0 0

1 1 0 1

1 1 1 0

1 1 1 1

1 0 0 1

Example:

Map the following POS expression
on a K-Map:

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

Solution :

0 0 0 0	}
0 0 1 0	
1 1 0 0	}
1 1 0 1	
1 1 1 0	
1 1 1 1	}
1 0 0 1	

		CD	00	01	11	10
		AB	00	01	11	10
00	01	00	0			0
		01				
11	10	00	0	0	0	0
		01				



- ✓ Introduction to K-Map (Karnaugh Map)
- ✓ K-Map → SOP expression Simplification
- ✓ K-Map → POS expression Simplification

K-Map simplification of SOP expression

- There are 3 steps to obtain a minimum SOP expression from a K-Map.
 1. **Grouping** the binary **1s**
 2. Determine **product term** for each group
 3. **Summing** the resulting product terms

STEP 1: Grouping the binary 1s

- Group must contain cells in 2^x combination
(i.e., 1, 2, 4, 8, 16)
- Each cell must be adjacent to at least 1 other cell in the group, but all cells in a group need not be adjacent
- Try to have the **biggest possible group of 1s**
- Each binary **1** must be in **at least one group**.
- The 1s already in a group may be included in another group so long as the **overlapping** group includes non-common 1s

		c	0	1
AB		00	1	
		01	1	
11		1	1	
		1		

		c	0	1
AB		00		1
		01	1	
11		1		
		1	1	

- Here are some examples:

(a)

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

(b)

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

(c)

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

(d)

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

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

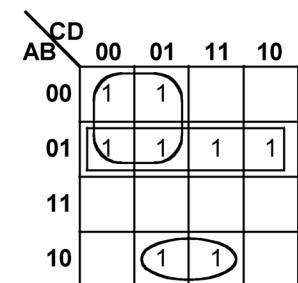
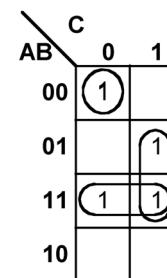
		C	0	1	
AB			00	1	1
00	1		01	1	
01			11		1
11			10	1	1
10	1	1			

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

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

STEP 2: Determining minimum SOP from a K-Map

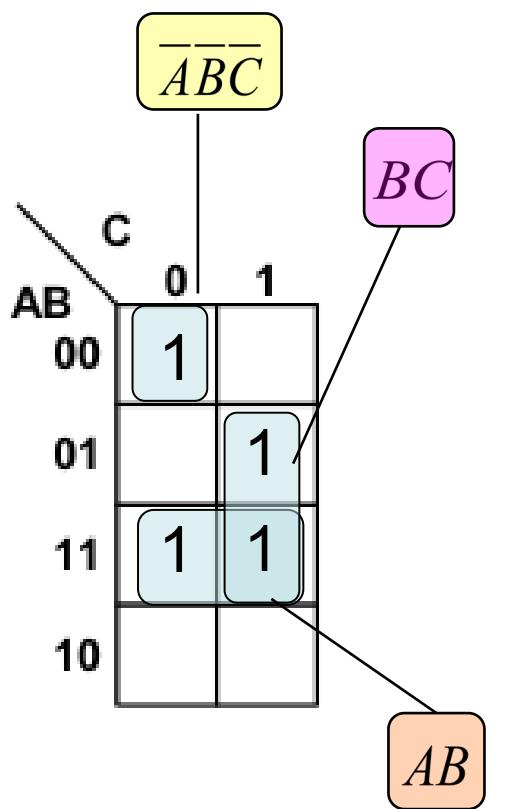
- ① Within each group made, only choose the variables that occur in one form (or **remain unchanged**): complemented or un-complemented. This is called *contradictory variables*.
- ② determine the minimum product term for each group with the *same variables*.
- ③ When the minimum product terms are derived, sum them to form the minimum SOP



Example:

(Module v5: Page 137)

- Determine the product terms for each of the K-maps and
- write the minimum SOP expression.

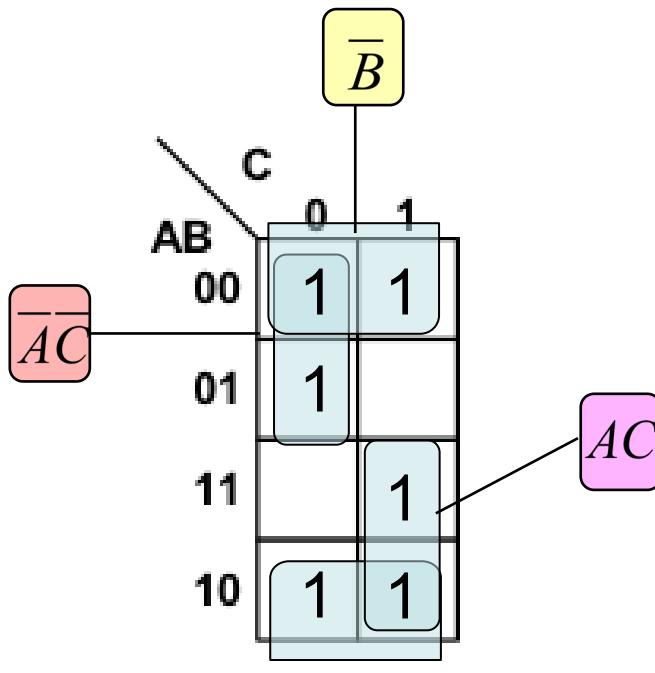


(a) $\overline{ABC} + AB + BC$

A	B	C

Example:

- Determine the product terms for each of the K-maps and
- write the minimum SOP expression.

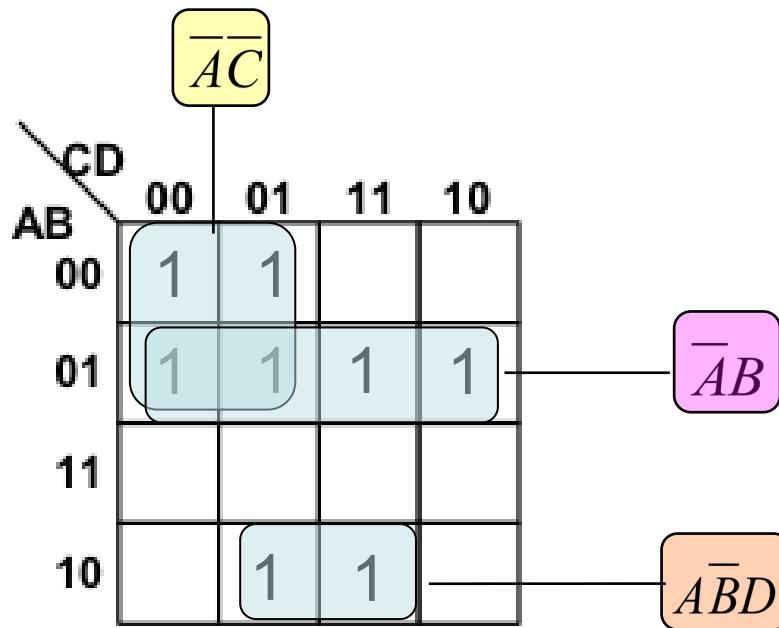


A	B	C

(b) $\overline{AC} + \overline{B} + AC$

Example:

- Determine the product terms for each of the K-maps and
- write the minimum SOP expression.



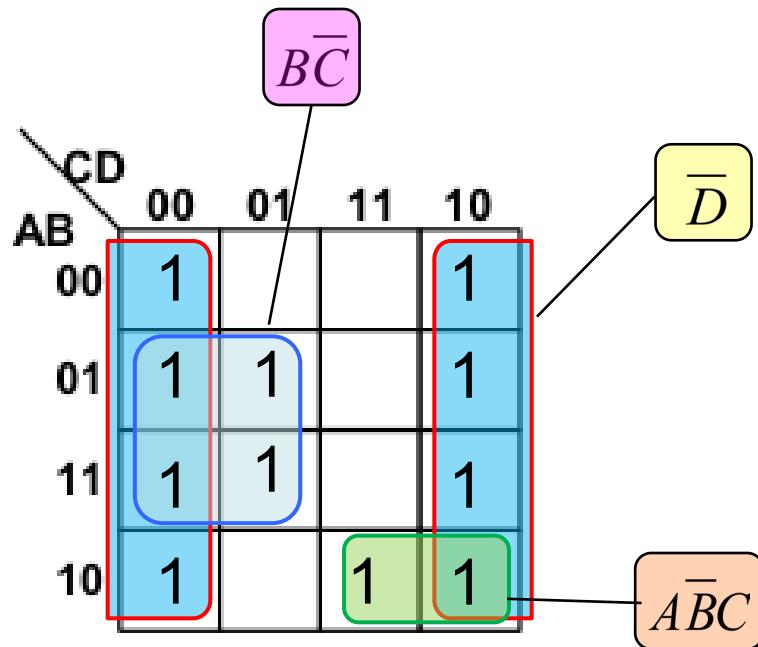
(c)

$$A\overline{BD} + \overline{AC} + \overline{AB}$$

A	B	C	D

Example:

- Determine the product terms for each of the K-maps and
- write the minimum SOP expression.



A	B	C	D

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

Exercise 4b.13:

Use a Karnaugh map to minimize the following standard SOP.

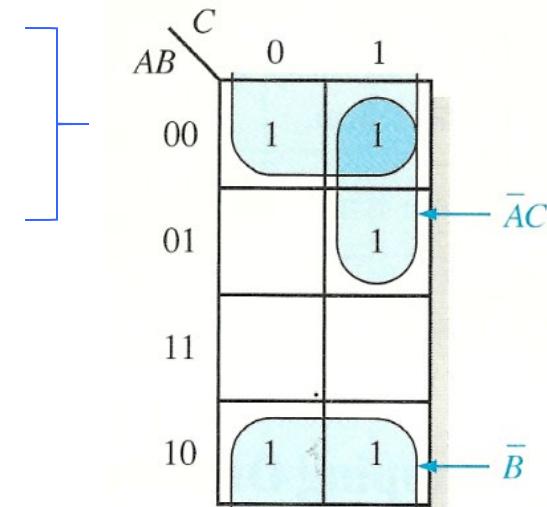
$$A\bar{B}C + \bar{A}BC + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C}$$

Solution 4b.13:

The binary values of the expression.

1 0 1 0 1 1 0 0 1 0 0 0 1 0 0

Map the expression and
group the cells.



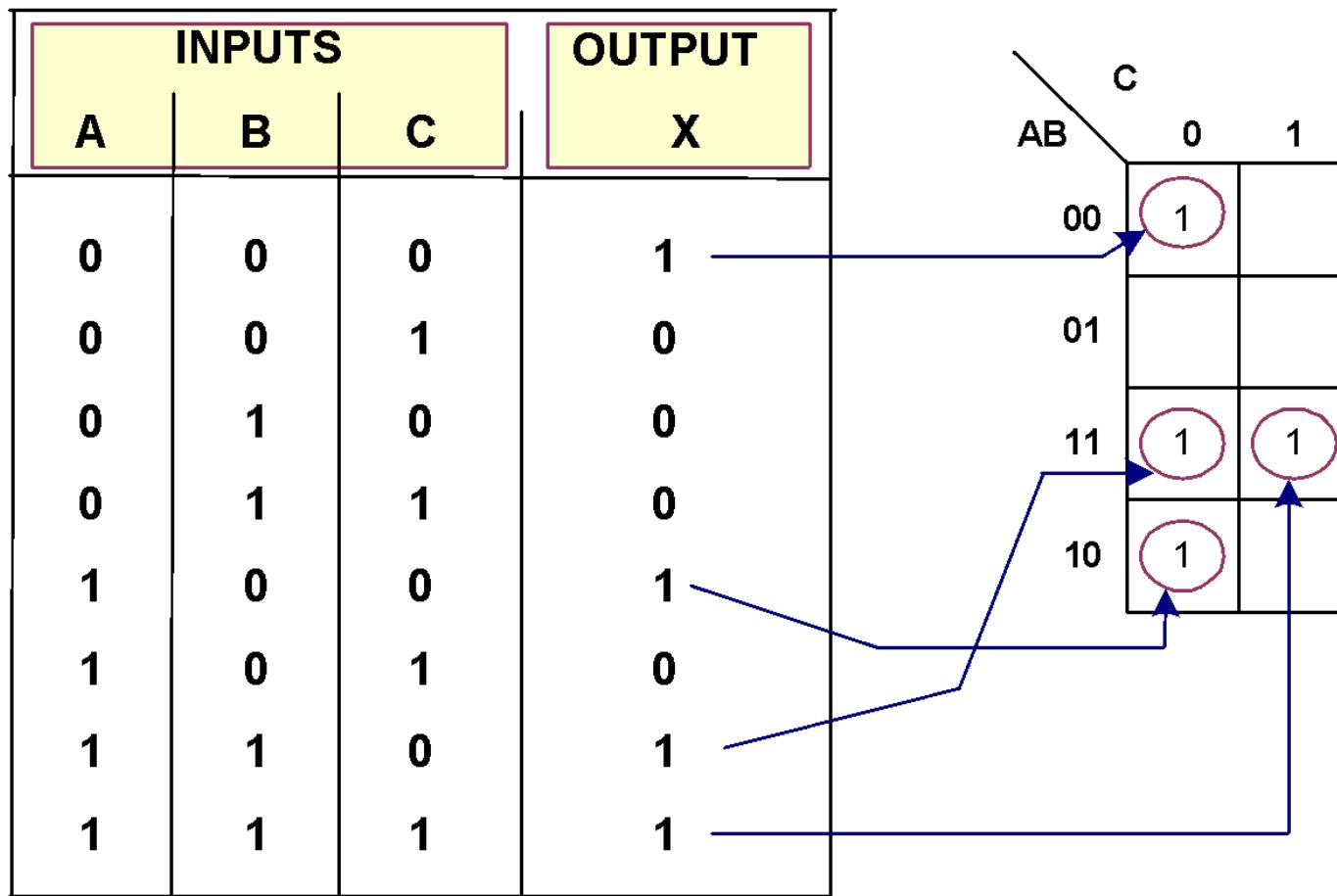
The resulting minimum SOP Expression is =

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

Example:

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

Mapping directly K-Map from
a truth table



'Don't Care' conditions

- Sometime there are certain combinations of input variable that is not allowed.
 - E.g. there are 6 invalid combinations in BCD code (1010, 1011, 1100, 1101, 1110, 1111)
- These states will never occur in an application using BCD code, hence they can be treated as “*don't care*” terms.
- “*don't care*” terms may be assigned either a 1 or a 0; it doesn't matter as it will never occur.

Recall
Module 2

Binary Coded Decimal (BCD)

- BCD is a way to express each of the decimal digits with a binary code.
- There are only 10 code groups in the BCD system, one for every digit (0000 – 1001)

Decimal	BCD	Decimal	BCD
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

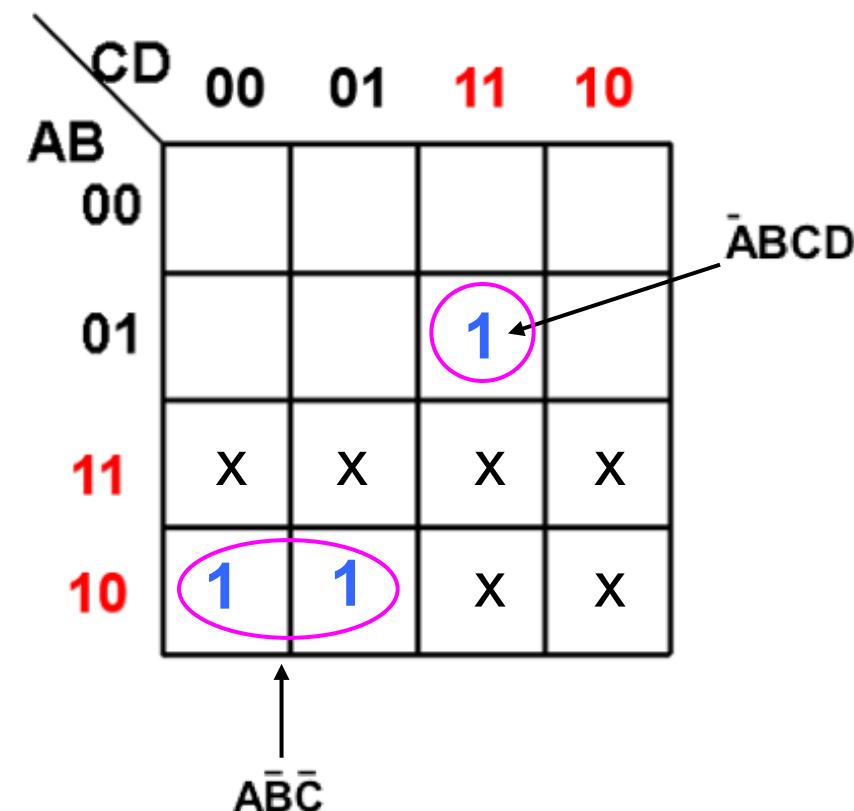
Invalid codes are **1010, 1011, 1100, 1101, 1110, 1111**

- “*don't care*” terms can be used with K-maps to produce better results
 - Used **as part of a group** → the larger the group, the simpler the resulting term will be
 - When grouping 1s (SOP), the **Xs** can be **treated as 1** to make the group larger
 - the **Xs** is **treated as 0s** if they cannot be used to advantage.

Example: BCD codes

INPUTS				OUTPUT
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

Invalid
input for
BCD

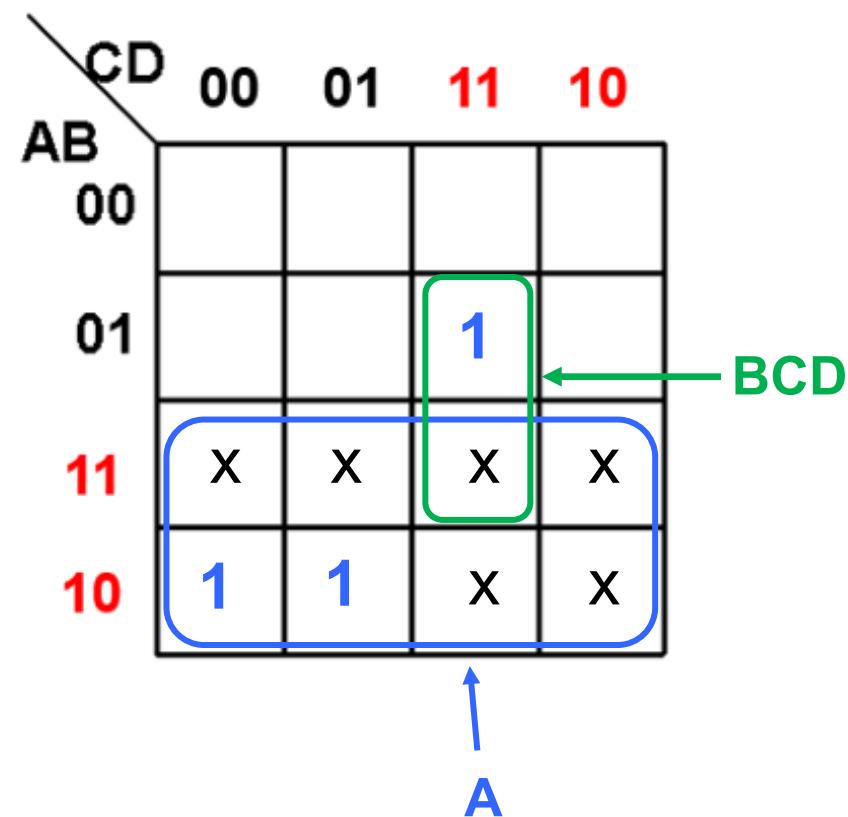


Without "don't cares" $Y = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{D}$

Example: BCD codes

INPUTS				OUTPUT
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

Invalid
input for
BCD



With “don’t care”

$$Y = A + BCD$$

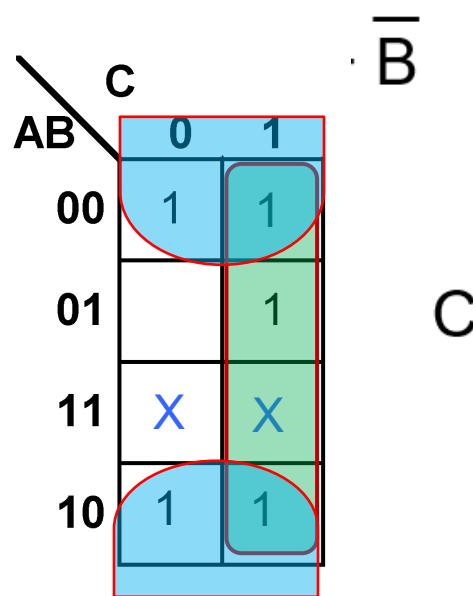
Use a Karnaugh map to minimize a standard SOP expression

Exercise 4b.14:

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

(Use “*don’t care*” terms, $d(6, 7)$ into the K-map to produce better results)

Solution 4b.14:



The binary values of the expression.

1 0 1 0 1 1 0 0 1 0 0 0 1 0 0

The binary value of “*don’t care*” notation.

1 1 0 1 1 1

Map the expression and group the cells.

Expression: $\bar{B} + C$

Exercise 4b.20: Use a Karnaugh map to minimize an expression

$$\sum_{ABCD} (2, 5, 8, 9, 11)$$

(Use “*don’t care*” terms, $d(0, 1, 10)$ into the K-maps to produce better results)



Exercise:

Derive minimum Boolean expression from the K-map below:

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

Homework



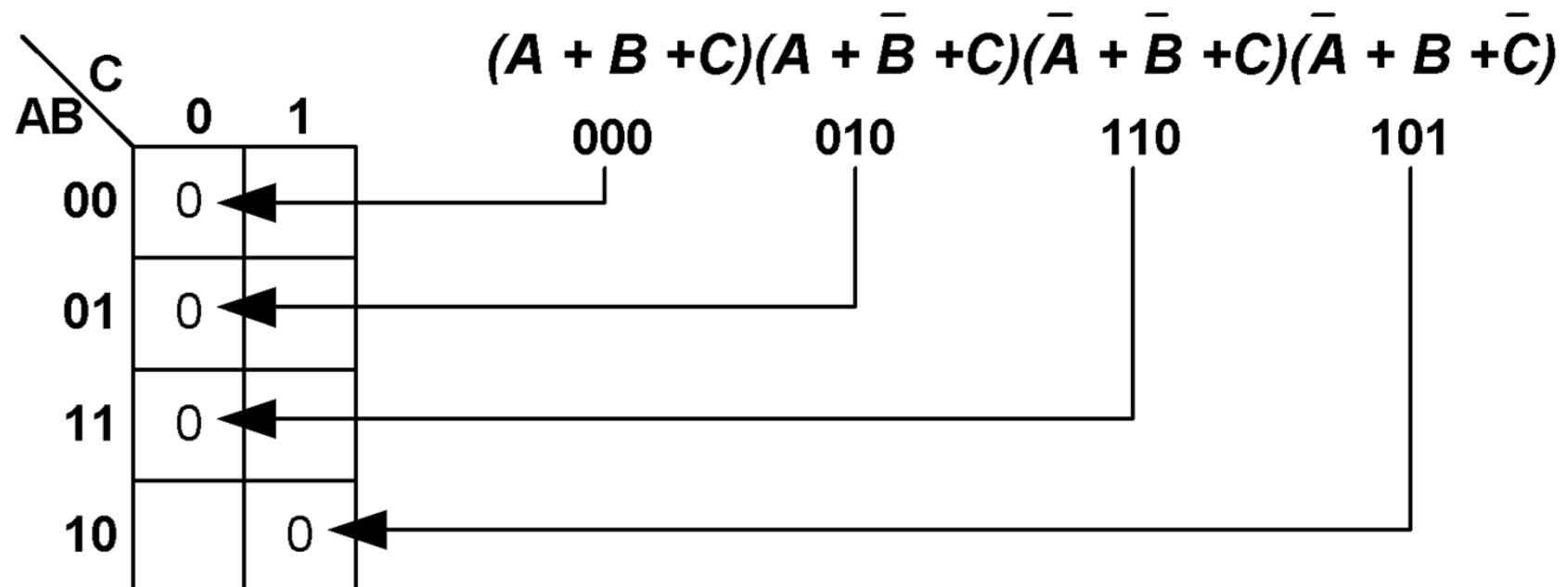
- ✓ Introduction to K-Map (Karnaugh Map)
- ✓ K-Map → SOP expression Simplification
- ✓ K-Map → POS expression Simplification

K-Map simplification of POS expression

- For POS expression in standard form, a **0** is put in the K-map for each **sum term**.
- The method is similar to SOP minimization, **except 0 is used**.
- To map a Standard **POS** expression:
 - **Step 1:** Determine the binary value of each sum term (i.e. that makes the sum term = 0)
 - **Step 2:** Check result and place a 0 on the corresponding cell in **K-map**

continue...

Example: Try mapping the following Standard POS expression into the K-Map:



(The method is similar to SOP minimization,
except **0** is used for POS)

Example: Try mapping the following Standard POS expression into the K-Map:

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

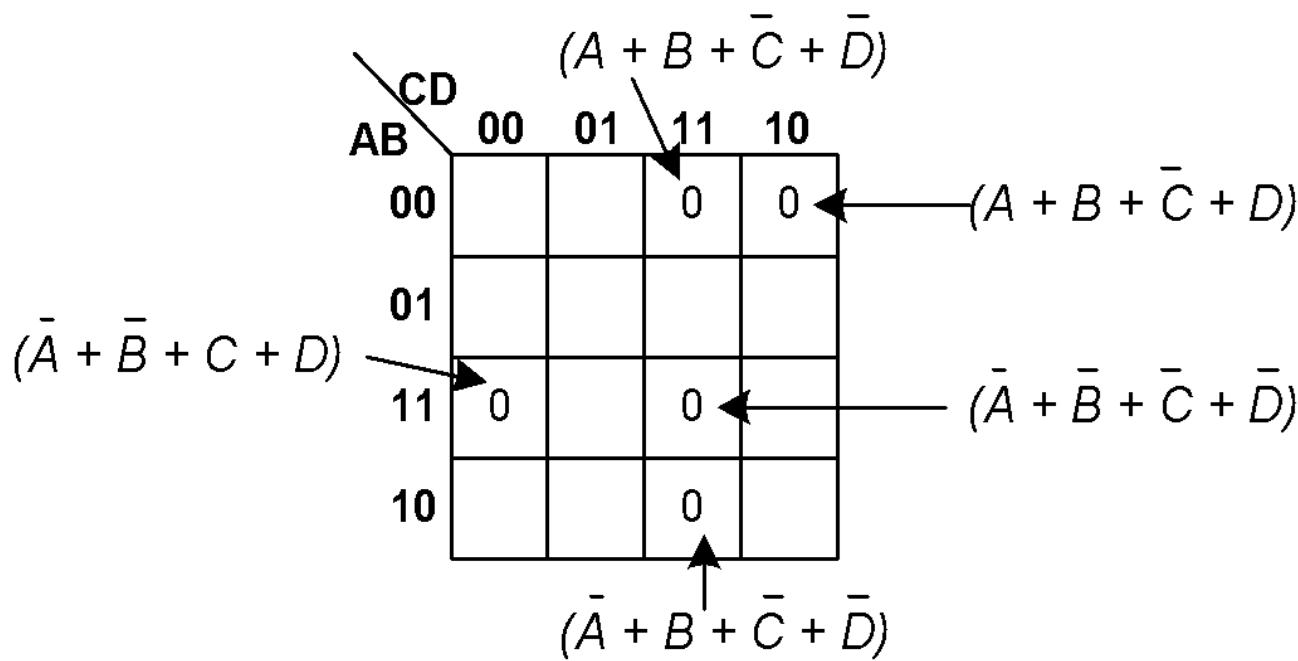
1100

1011

0010

1111

0011



K-Map simplification of POS expression

- The process is basically the same as with SOP expressions:
 - Group the 0s instead of 1s
 - The rules of grouping 0s are the same as those for 1s
 - Expression must be in Standard POS form

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

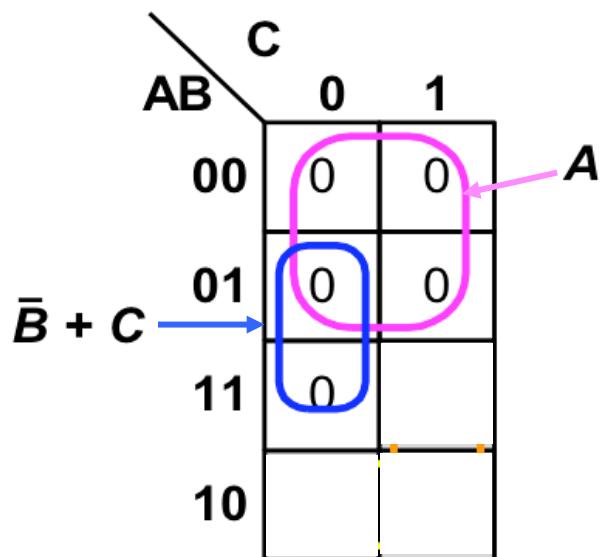
0 0 0

0 0 1

0 1 0

0 1 1

1 1 0



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

Example 2:

Minimize this POS expression using a K-Map:

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

- Change it to standard POS form

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

- Get the binary values

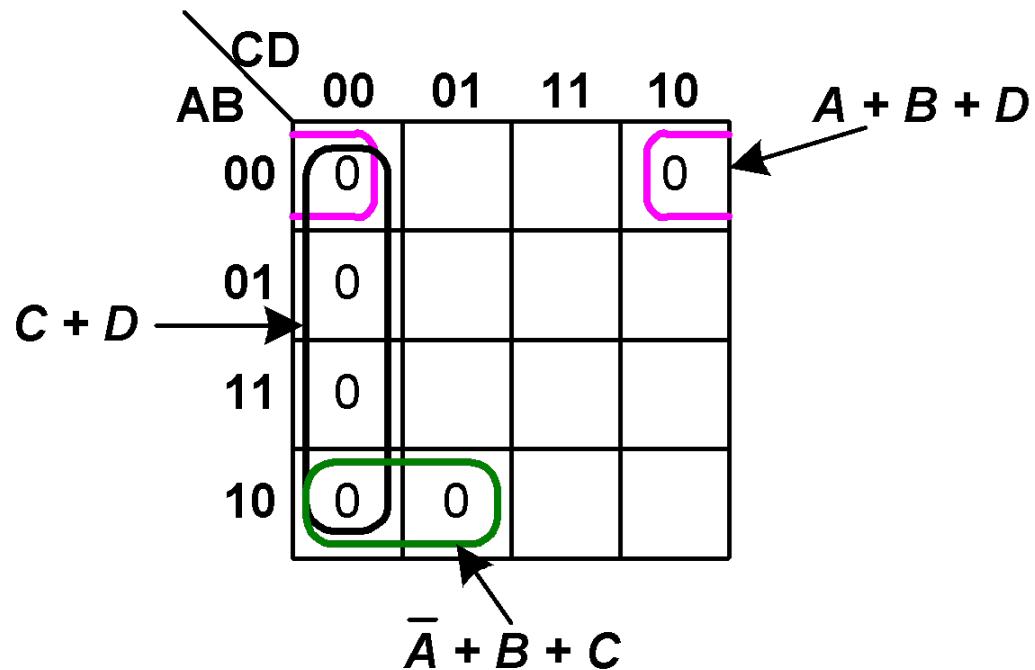
$$(0000)(1000)(0010)(1001)(0100)(1100)$$

$(0000)(1000)(0010)(1001)(0100)(1100)$

- Map it,

		CD	00	01	11	10
		AB	00	01	11	10
AB	00	0				0
	01	0				
	11	0				
	10	0	0			

then group the 0s



- Minimum POS expression

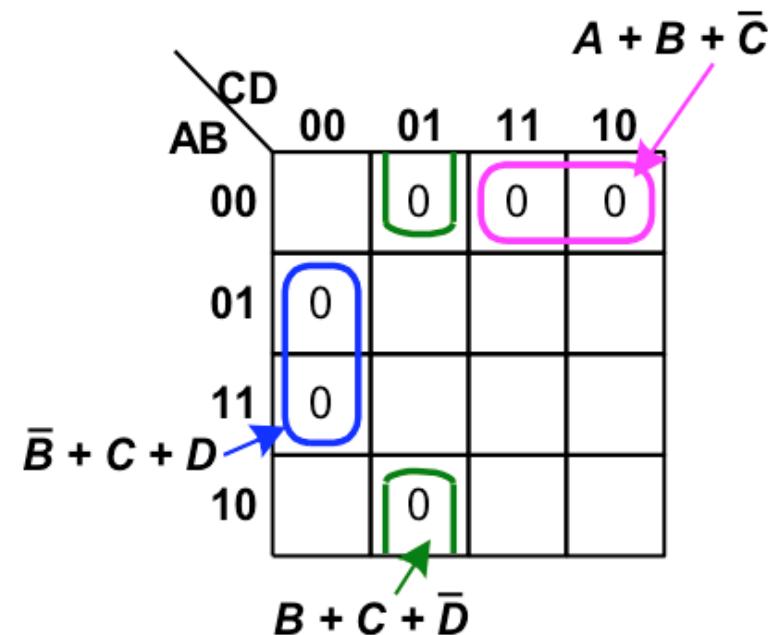
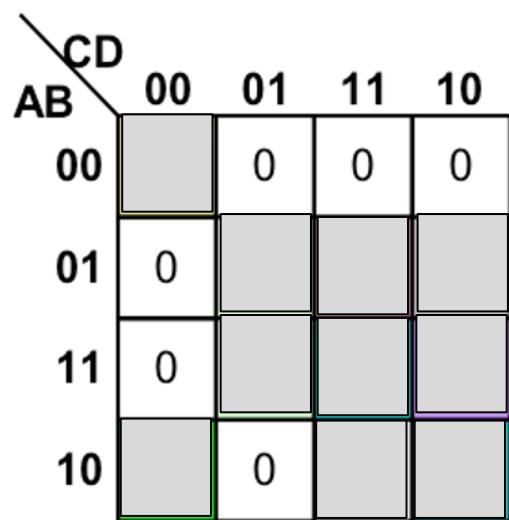
$$(C + D)(\bar{A} + B + C)(A + B + D)$$

Converting between POS and SOP using K-Map

- A mapped SOP expression **can be converted** to equivalent POS expression.
- This is a good way **to compare** which one can be implemented **using fewer gates**.
- Given a minimum POS map, the 1s will yield a **standard** SOP expression.
- This SOP expression can then be minimized by grouping the 1s.

Example: Converting between POS and SOP using K-Map

Get the **sum terms** for POS and write the **minimum POS** expression.

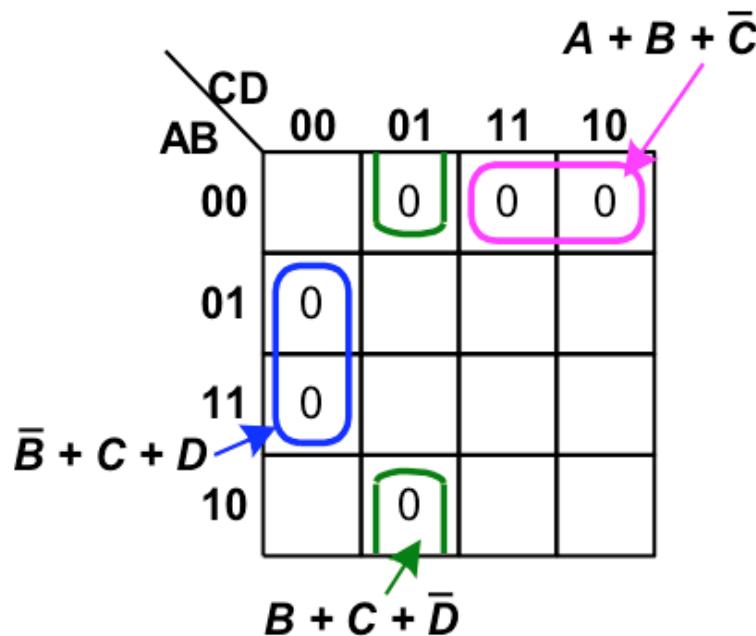


Minimum POS:-

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

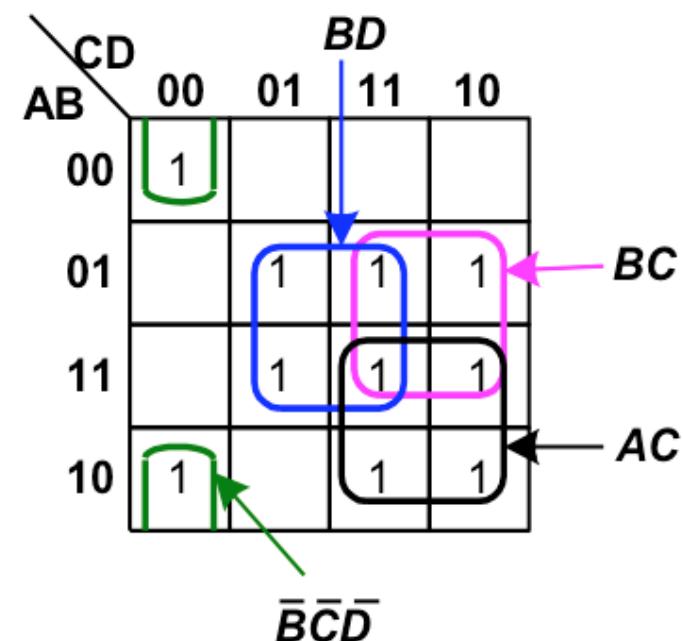
Example: Converting between POS and SOP using K-Map

Get the product terms for SOP and write the minimum SOP expression.



Minimum POS:-

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



$$\text{Minimum SOP: } AC + BC + BD + \bar{B}\bar{C}\bar{D}$$

Exercise 4b.19:

Assume $d(1,5,8,10)$ is a set of don't care used. Generate the new minimized expression from the K-Map.

		CD	00	01	11	10
		AB	00	01	11	10
00	00	0				0
01	01					
11	11	0	0	0	0	0
10	10		0			



 Image Upload

Exercise 4b.16:

Transform the following expression into the K-Map. Then, generate the simplified expression from the K-Map.

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



Exercise 4b.17:

Using the same expression in exercise 4b.16,
simplify the expression using the Boolean algebra
and laws.

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





Simplify a Combinational Logic Circuit

Simplify a Combinational Logic Circuit

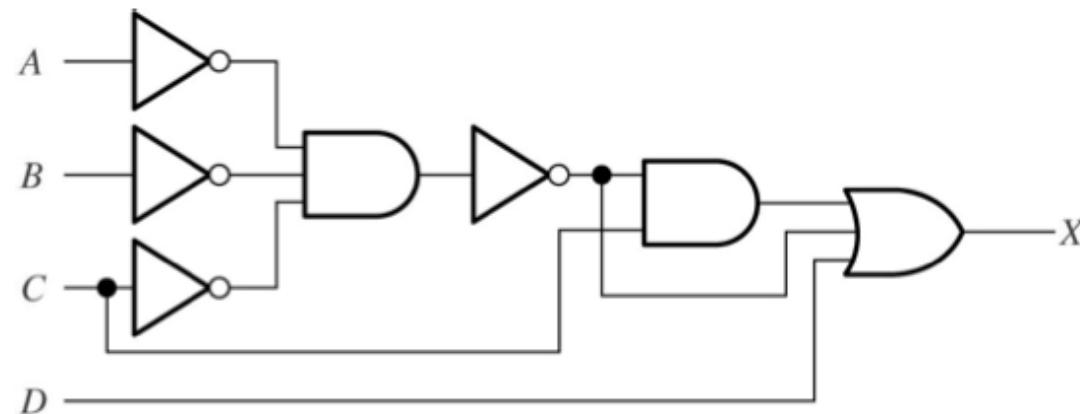
- Simplify a combinational logic circuit will result in lesser gates used
- How to do this?

Step 1 : Determine the input and output variables of each gates in the circuit

Step 2 : Get the final output expression

Step 3 : Apply De Morgan's theorem and Boolean algebra or K-map.

Example 1: Given the logic circuit below, simplify using boolean algebra and DeMorgan's theorem.

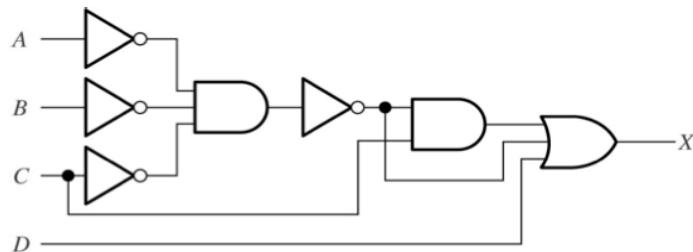


Step 1: input variables = A, B, C, D

output variables = X

Step 2: Get the final output expression

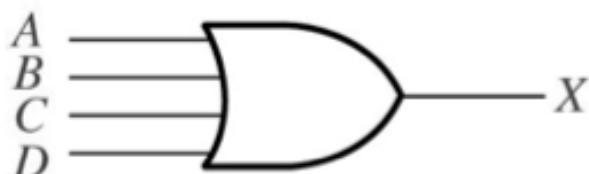
$$X = (\overline{\overline{A}}\overline{\overline{B}}\overline{\overline{C}})C + \overline{\overline{A}}\overline{\overline{B}}\overline{\overline{C}} + D$$



Step 3: Apply De Morgan's theorem and Boolean Algebra

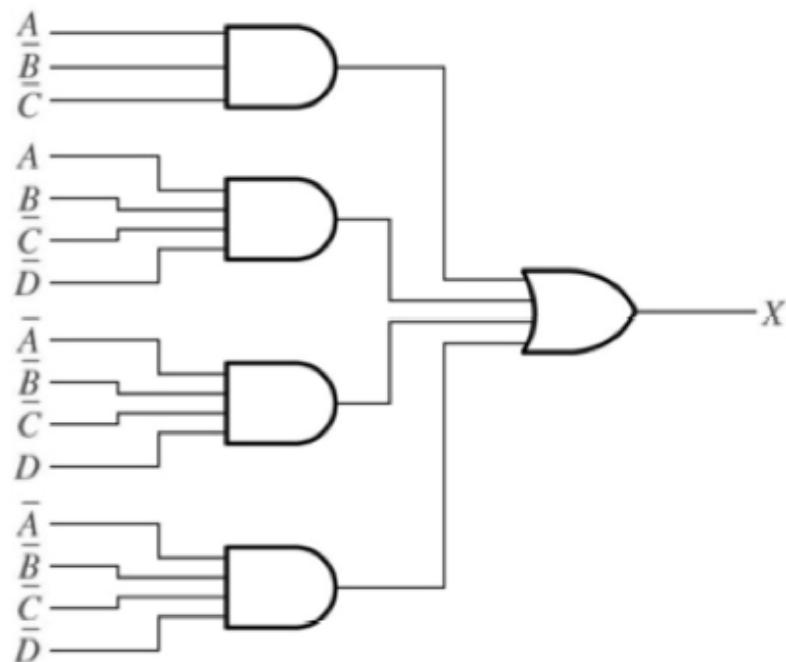
$$\begin{aligned}
 & (\overline{\overline{ABC}})C + \overline{\overline{ABC}} + D \\
 = & (\overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}})C + (\overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}}) + D & \overline{\overline{AB}} = \overline{A} + \overline{B} \\
 = & AC + BC + CC + A + B + C + D & \overline{\overline{A}} = A \\
 = & \cancel{(A + AC)} + \cancel{(B + BC)} + \cancel{(C + D)} & C.C = C \quad C + C = C \\
 = & A + B + C + D & A + AC = A \quad B + BC = B
 \end{aligned}$$

After simplification:



Example 2:

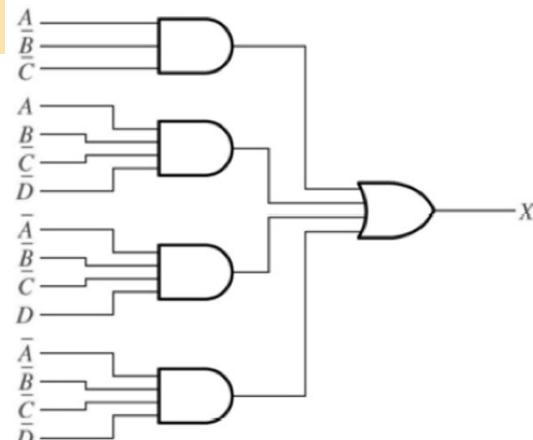
Simplify the combinational logic circuit using K-Map



Step 1 Input variable = A, B, C, D, Output variable = X

Step 2 Output Expression of the circuit is

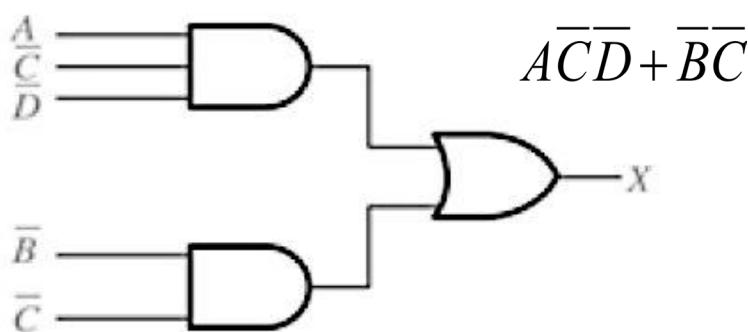
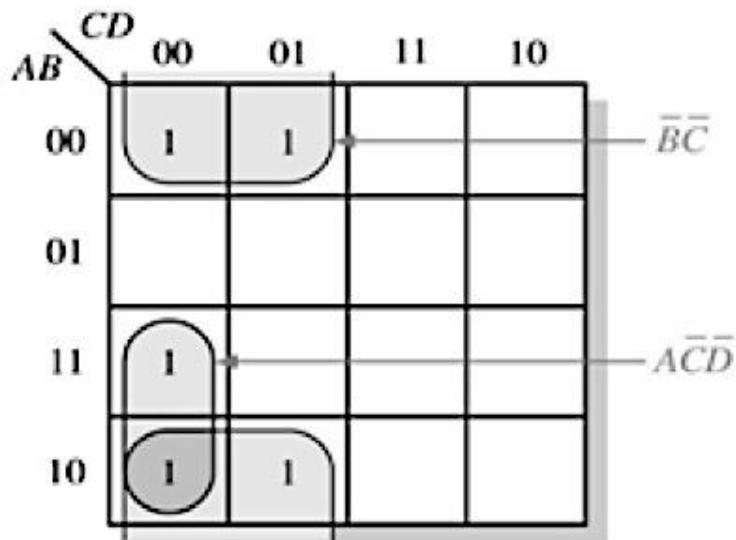
$$X = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D$$



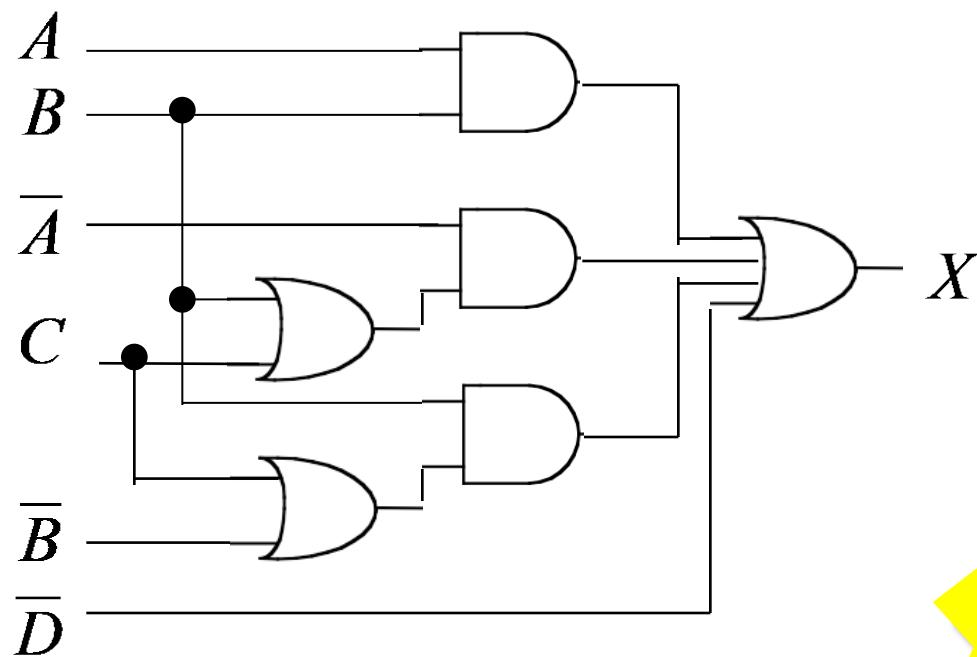
Change it to standard SOP form:

$$X = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D}$$

Step 3 Use K-map to simplify the circuit



Exercise 4b.21: Simplify the combinational logic circuit using K-map.



Homework

3. Simplify the function using Boolean Algebra

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

4. Develop a truth table for the following expression. From the truth table derive a standard product-of-sums (POS) expression.

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

5. Use a Karnaugh map to reduce the expression to a minimum sum-of-products (SOP) form

$$f = \bar{A}B(C\bar{D} + CD) + ACD$$

Homework

10. Using Karnaugh Map, find the minimum SOP expression for the following function

i)

$$Y = \overline{\overline{P} \overline{Q} \overline{R} \overline{S}} + \overline{\overline{P} \overline{Q} R \overline{S}} + \overline{\overline{P} \overline{Q} R \overline{S}} + \overline{\overline{P} Q \overline{R} \overline{S}} + \overline{P \overline{Q} \overline{R} \overline{S}} + \\ \overline{P \overline{Q} R \overline{S}} + P \overline{Q} \overline{R} \overline{S}$$

ii)

$$G(w, x, y, z) = \sum m(1, 3, 14, 15) + d(0, 2, 6, 8, 13)$$

Homework



Summary

(Example of 2 product terms only)

SOP

Form: $X = \overline{ABC} + ABC$

Binary: 0 1 0 1 1 0

Notation: $X = \sum_{ABC} (2,6)$

EXPRESSION

(Example of 2 sum terms only)

POS

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

Binary: 0 1 1 1 0 1

Notation: $X = \prod_{ABC} (3,5)$

Standard?

Rule 6: $A + \overline{A} = 1$

Truth Table

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

Product / Sum Terms

\overline{ABC}

$AB\overline{C}$

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

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

K-Map

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

(Example of some product / sum terms only)

Don't Care:

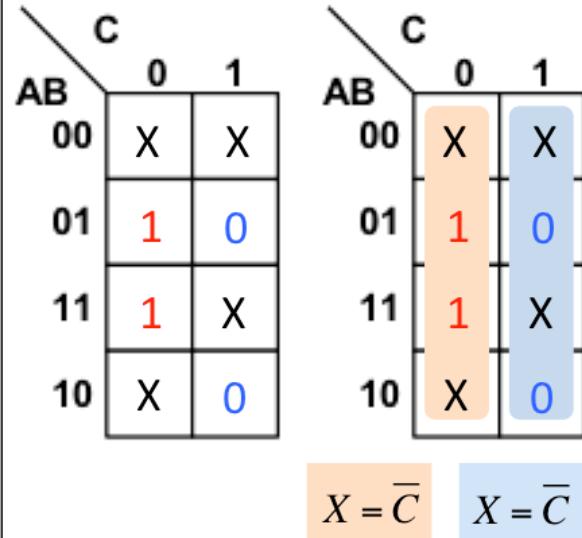
$d(0,1,4,7)$

Standard?

Rule 8: $A \cdot \overline{A} = 0$

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

K-Map





(Example of 2 product / sum terms only)

SOP

$$X = \overline{ABC} + ABC$$

0 1 0 1 1 0

POS

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

0 1 1 1 0 1

SOP

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

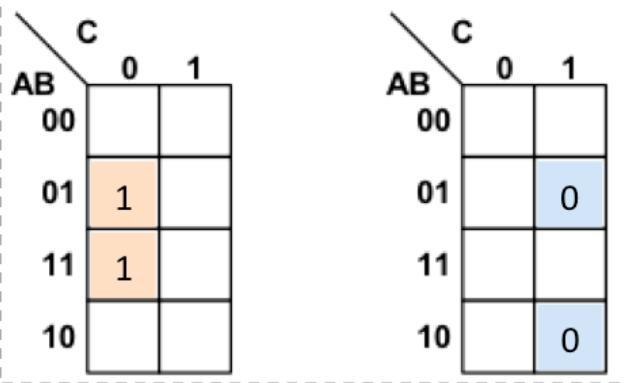
0 0 0 1 1 0 0 0 0
0 0 1 0 1 0
0 1 0 0 1 0
0 1 1 0 1 1

POS

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

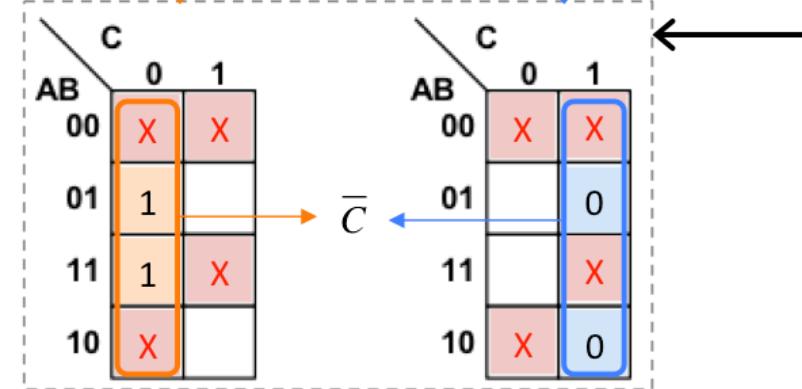
0 1 0 1 0 1 0 0 0
1 1 0 0 0 1
0 1 0 0 1 0
0 1 1 0 1 1

(Mapping on K-Map)

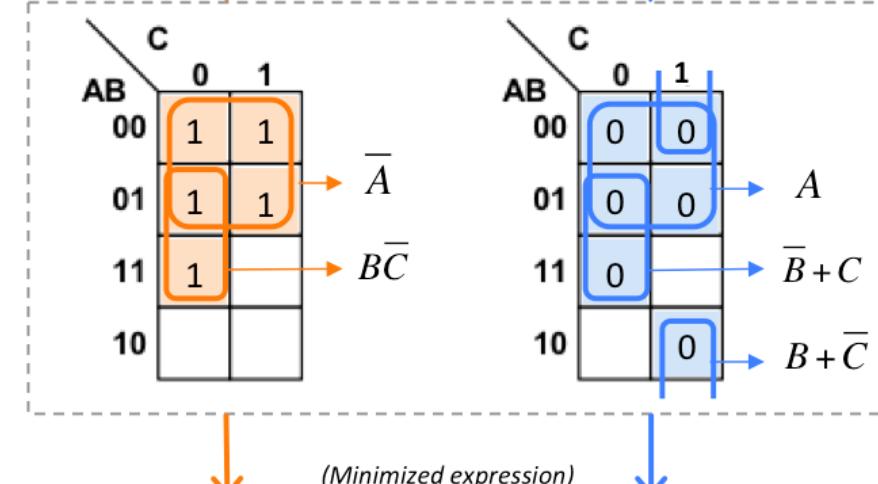


Don't Care:
 $d(0,1,4,7)$

(Grouping with don't care)



(Mapping and grouping on K-Map)



(Minimized expression)

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

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