Digital Circuits

ECS 371

Dr. Prapun Suksompong

prapun@siit.tu.ac.th

Lecture 8

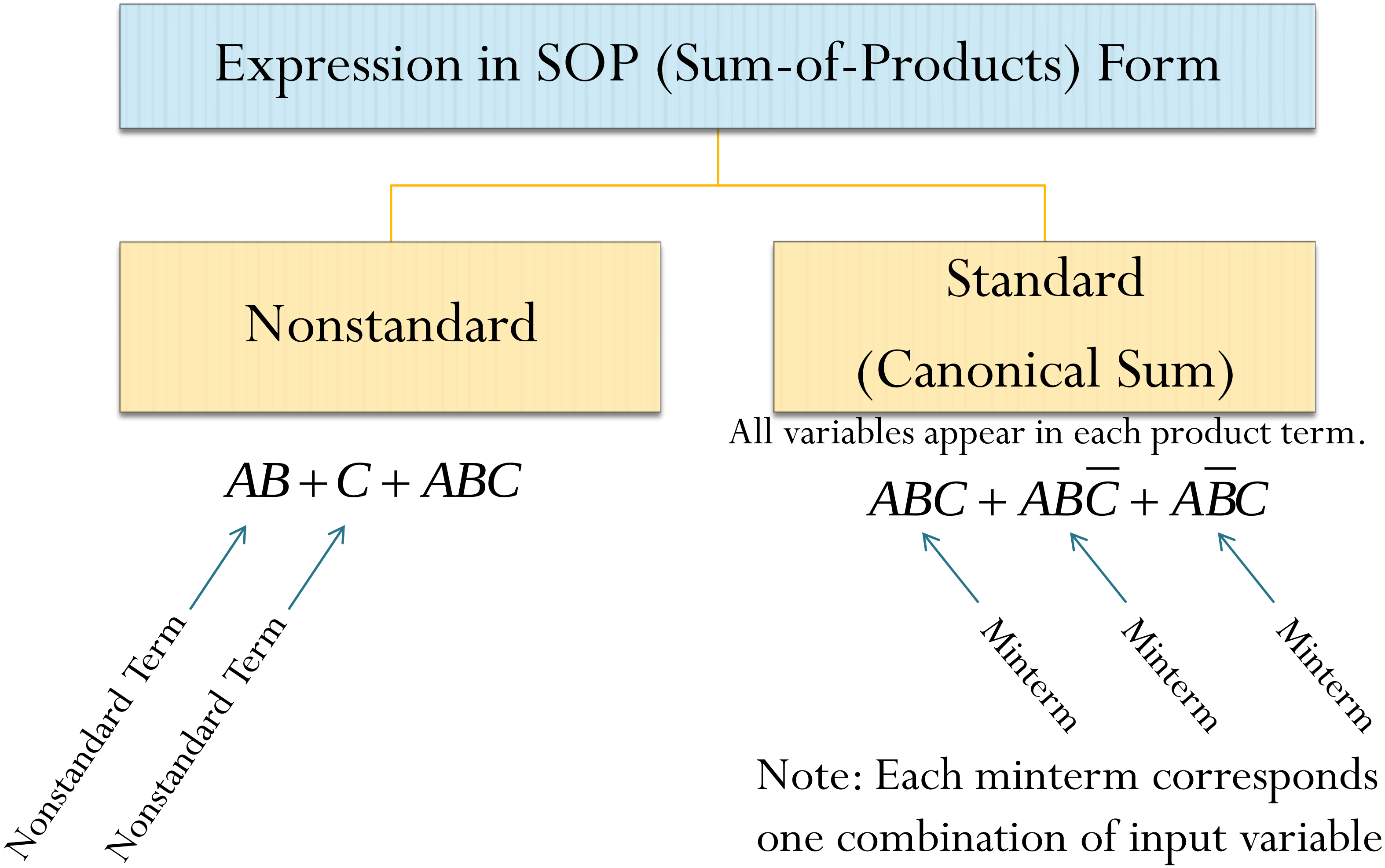
Office Hours:
BKD 3601-7
Monday 9:00-10:30, 1:30-3:30
Tuesday 10:30-11:30

ECS371.PRAPUN.COM

Announcement

- HW3 posted on the course web site
  - Chapter 4: 5(b,d), 26b, 30b, 32a, 34a, 44
  - **Write down all the steps** that you have done to obtain your answers.
  - Due date: July 2, 2009 (Thursday)
  - Please submit your HW to the instructor 3 minutes BEFORE your class starts.
  - Earlier submission is possible. There are two HW boxes in the EC department (6th floor) for ECS 371. (One for CS. Another one for IT.)

Review



Expression in SOP (Sum-of-Products) Form

Nonstandard

Standard (Canonical Sum)

All variables appear in each product term.

$$AB + C + ABC$$

Nonstandard Term

Nonstandard Term

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

Minterm

Minterm

Minterm

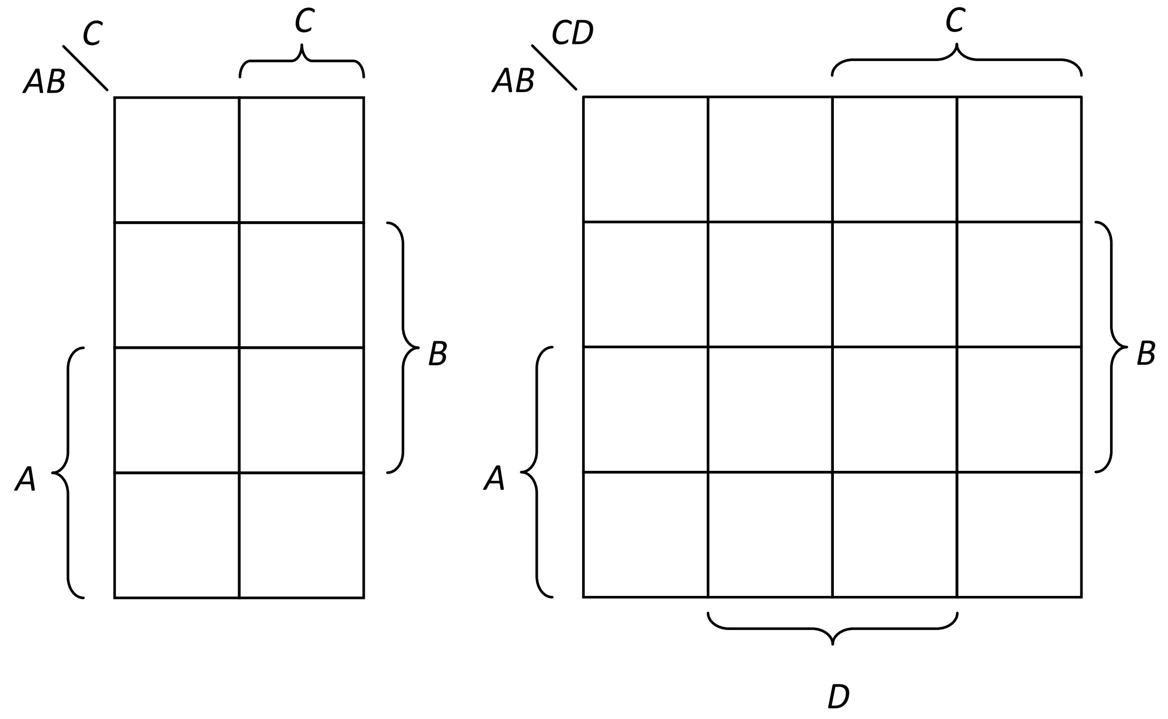
Note: Each minterm corresponds to one combination of input variables.

K-Map

- Used for simplifying Boolean expressions to their "minimum form." (We haven't talked about this.)
- We have done the following:
  - Construct K-map for three or four input variables.
  - Determine the minterm (standard product term) represented by each cell in a K-map.
  - Map a standard SOP expression (canonical sum) on a Karnaugh map.
- Today
  - New concept: Cell Adjacency

K-Map

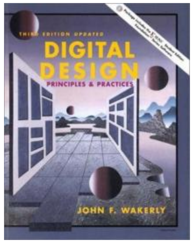
Convention: We use the following maps to define our K-maps

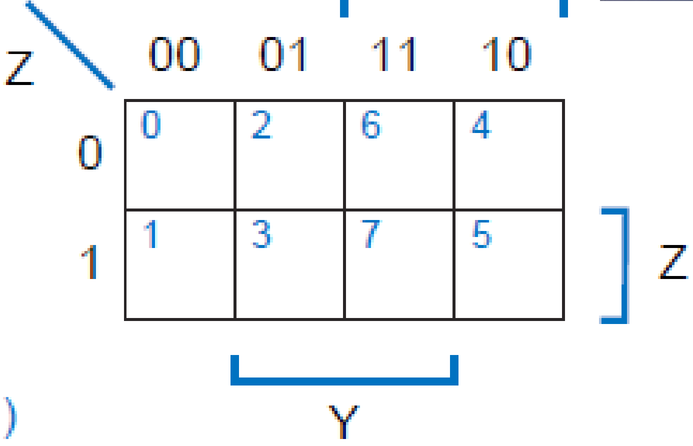


There are many ways to define the K-map. Once you are familiar with one convention, you may try to work with different convention.

Alternative K-Maps

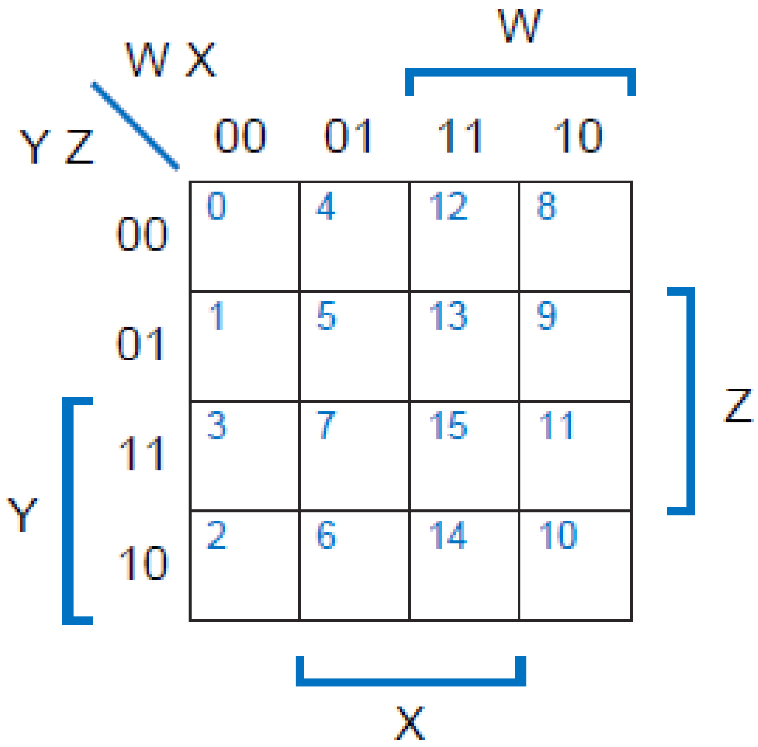
This convention is used in the textbook by Wakerly.





Z \ X Y	00	01	11	10
0	0	2	6	4
1	1	3	7	5

(b)



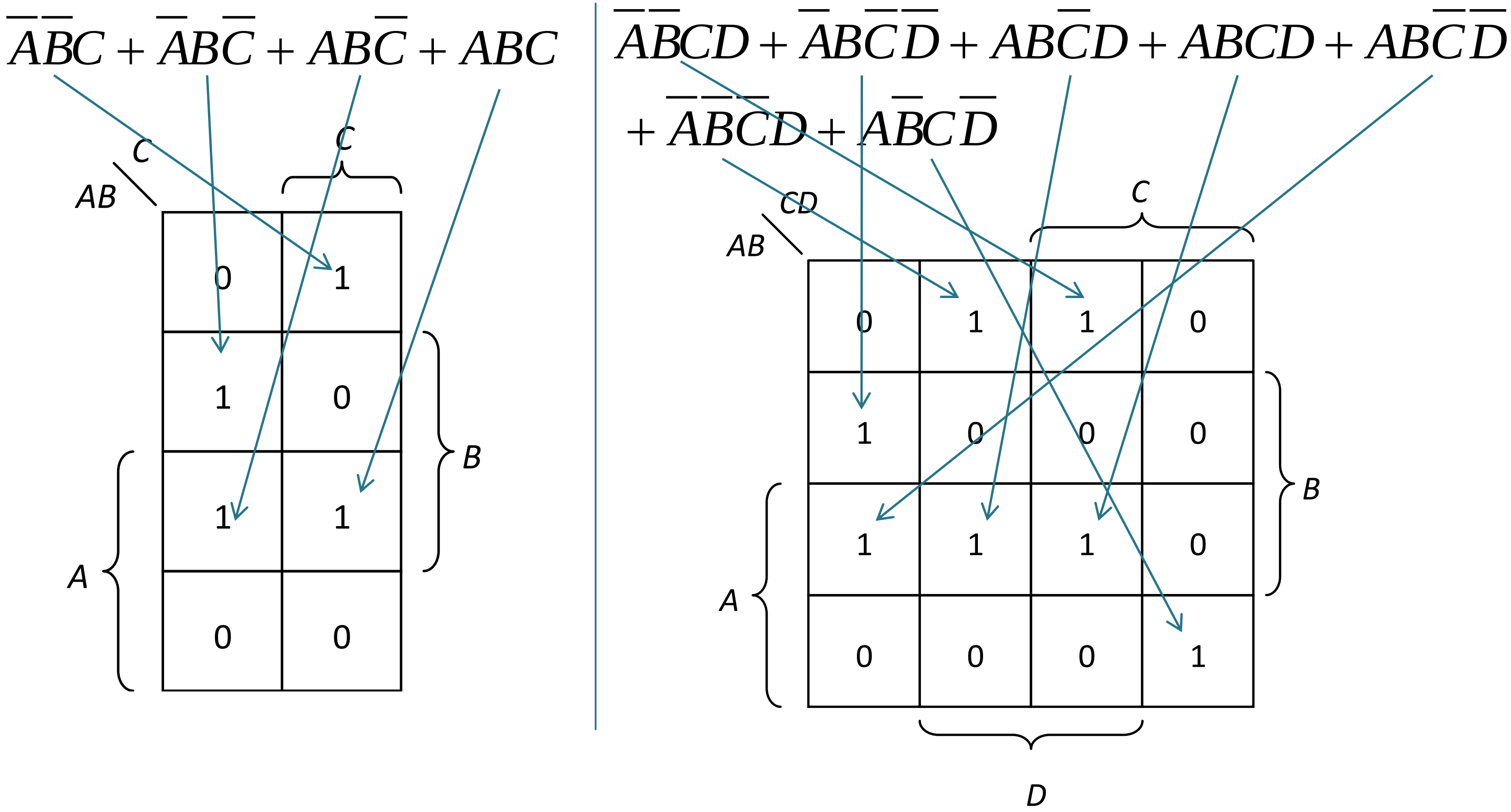
Y Z \ W X	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

(c)

We will NOT follow this convention.

Ex: Mapping (Canonical Sum)

Map the following expression on an appropriate K-map



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

0	1
1	0
1	1
0	0

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

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

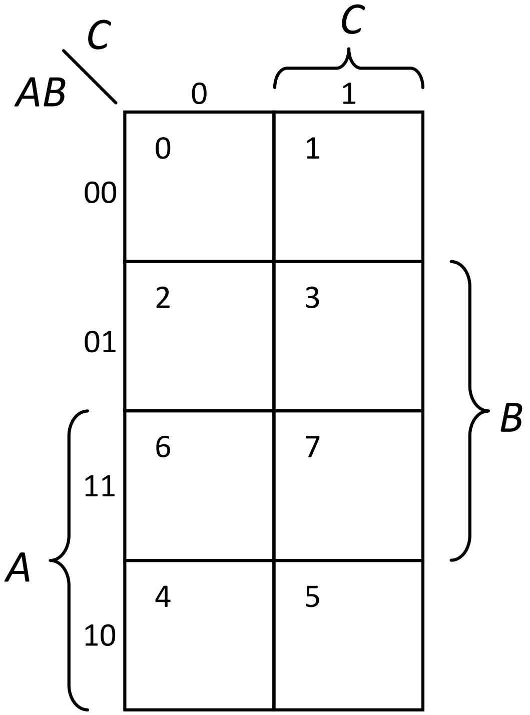
Summary

For a canonical sum (SOP expression in standard form):

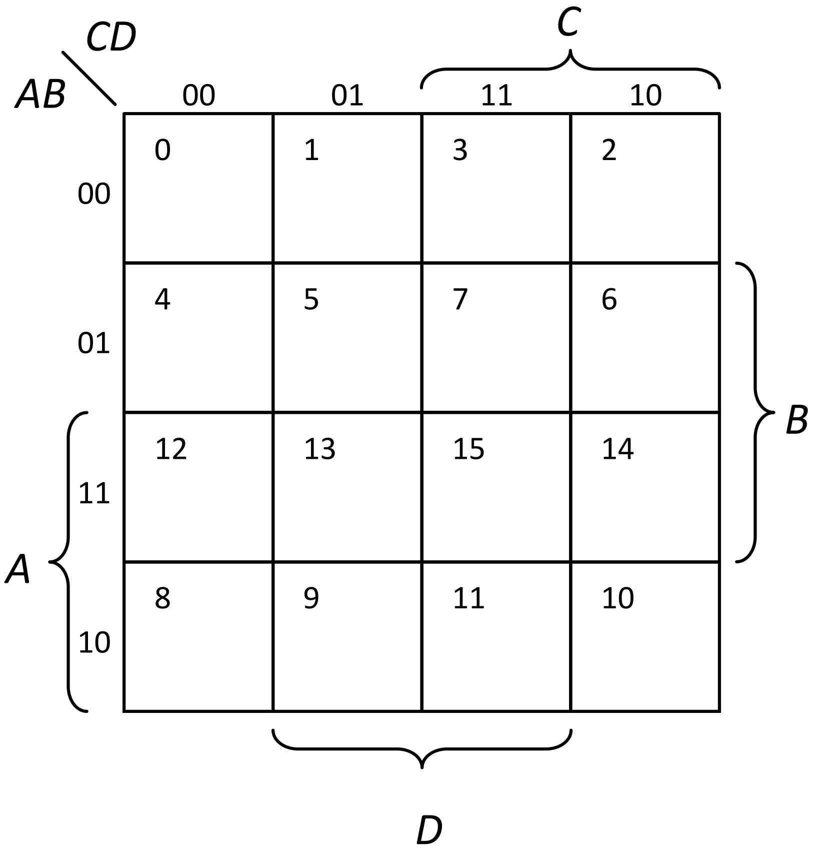
- A 1 is placed on the K-map for each minterm in the expression.
- Each 1 is placed in a cell corresponding to the minterm that produces it.
- When the canonical sum is completely mapped, there will be a number of 1s on the K-map equal to the number of minterms in the canonical sum.
- The cells that do not have a 1 are the cells for which the expression is 0.

Adjacency

Why do we arrange the cells in this “strange” ordering?



AB \ C	0	1
00	0	1
01	2	3
11	6	7
10	4	5



AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

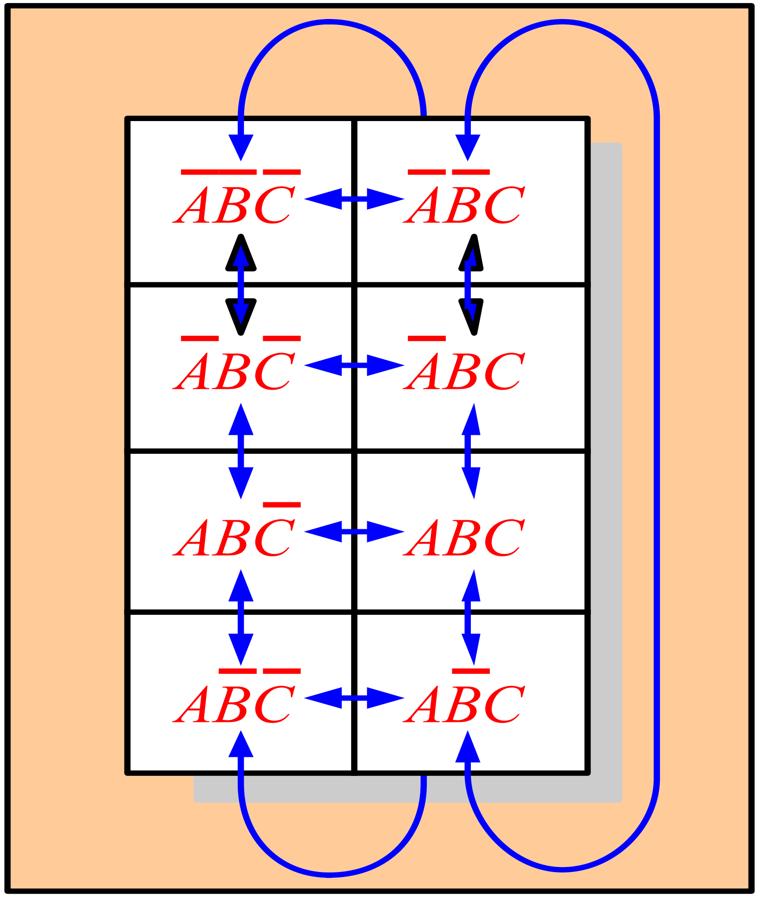
Adjacency

- The cells in a K-map are arranged so that there is only a single-variable change between adjacent cells.
- **Definition**: Cells that differ by only one variable are **adjacent**.
  - Physically, each cell is adjacent to the cells that are immediately next to it on any of its four sides.
- Note that a cell is not adjacent to the cells that diagonally touch any of its comers.

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

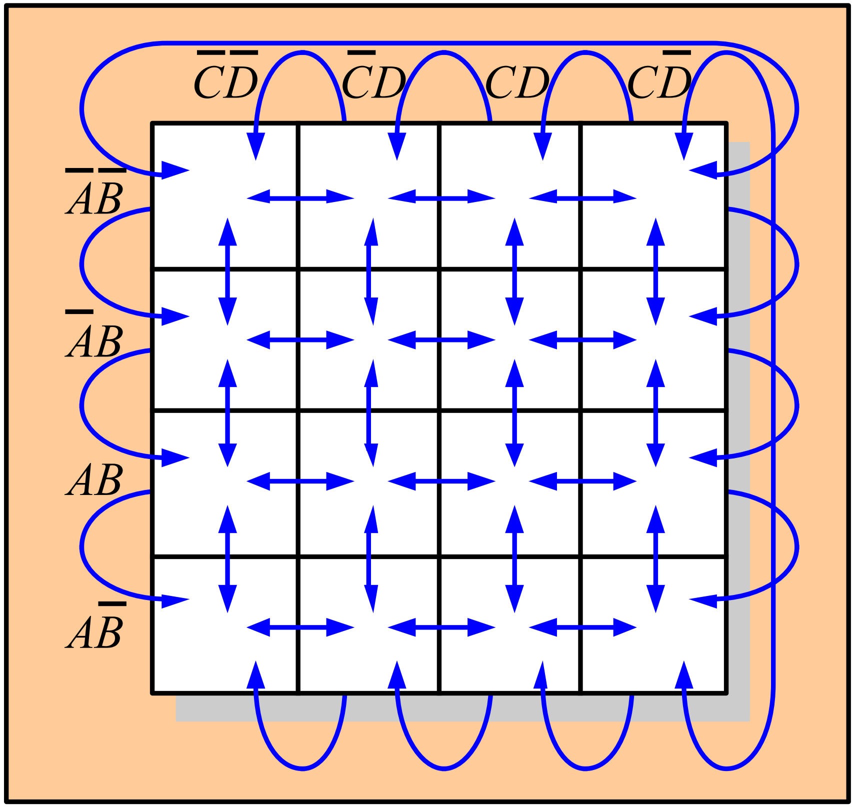
Wrap-around Adjacency (3 Variables)

The cells in the top row are adjacent to the corresponding cells in the bottom row.



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

Wrap-around Adjacency (4 Variables)

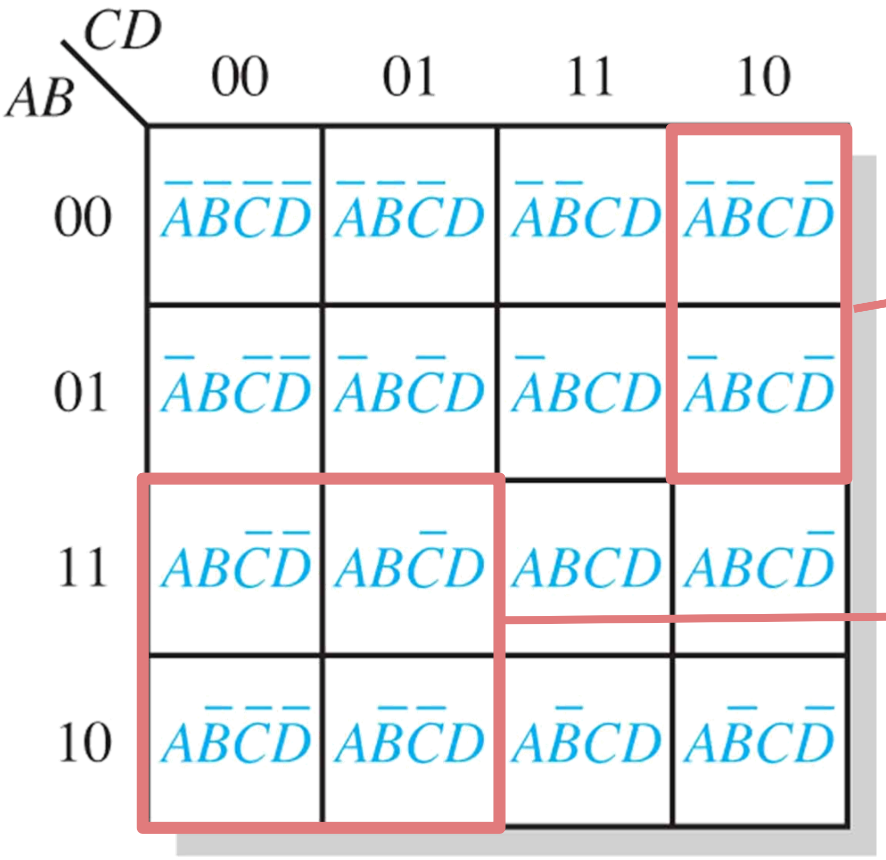


The cells in the top row are adjacent to the corresponding cells in the bottom row **and** the cells in the outer left column are adjacent to the corresponding cells in the outer right column.

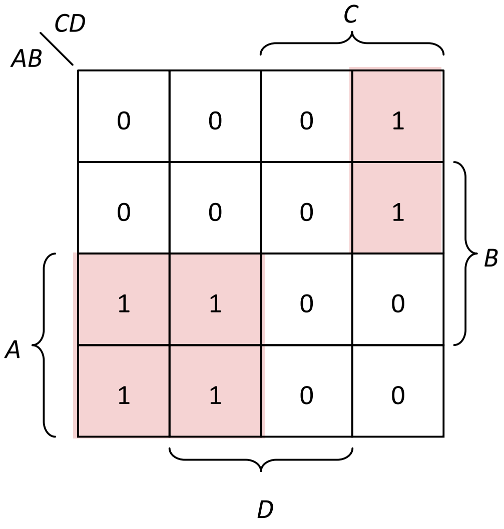
This is called "wrap-around" adjacency because you can think of the map as wrapping around from top to bottom to form a cylinder or from left to right to form a cylinder.

Why is this adjacency concept useful?

Grouping the 1s



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



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

$$\overline{A}\,\overline{B}C\overline{D} + \overline{A}BC\overline{D} = \overline{A}C\overline{D}\left(\overline{B} + B\right)$$

$$= \overline{A}C\overline{D}$$

Summation of these four minterms gives $A\overline{C}$

The K-map is used to “visualize” the combining theorem.

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

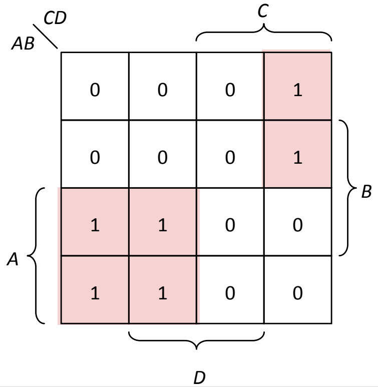
Rules: Grouping the 1s

You can group 1s on the K-map according to the following rules

- A group must contain 2^k cells.

Each time that we make a group larger, it will cover twice as many cells.

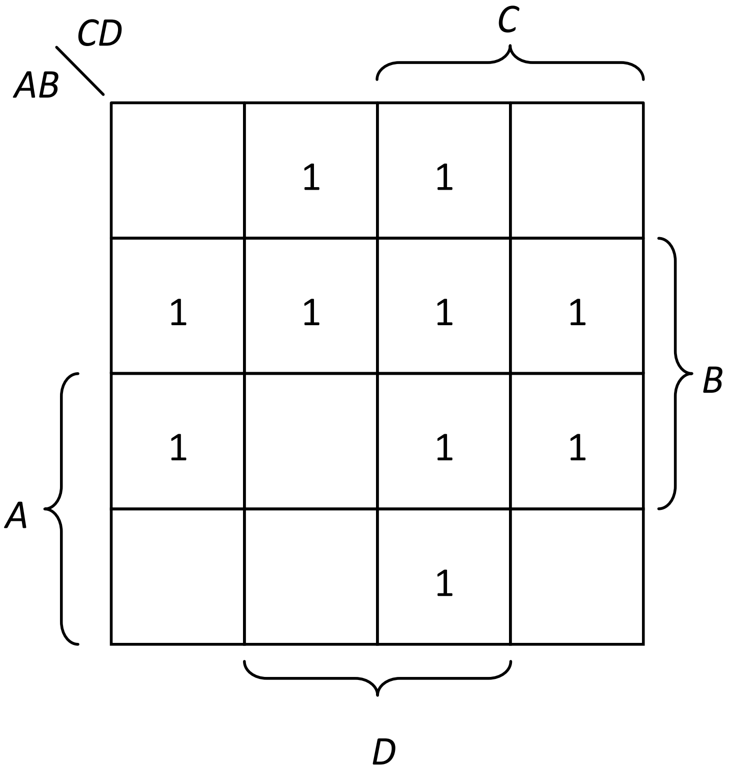
- Each cell in a group must be ***adjacent*** to one or more cells in that same group,
  - but all cells in the group do not have to be adjacent to each other.
- Always include the largest possible number of 1s in a group.
  - Each of this group is called "**prime implicant**"



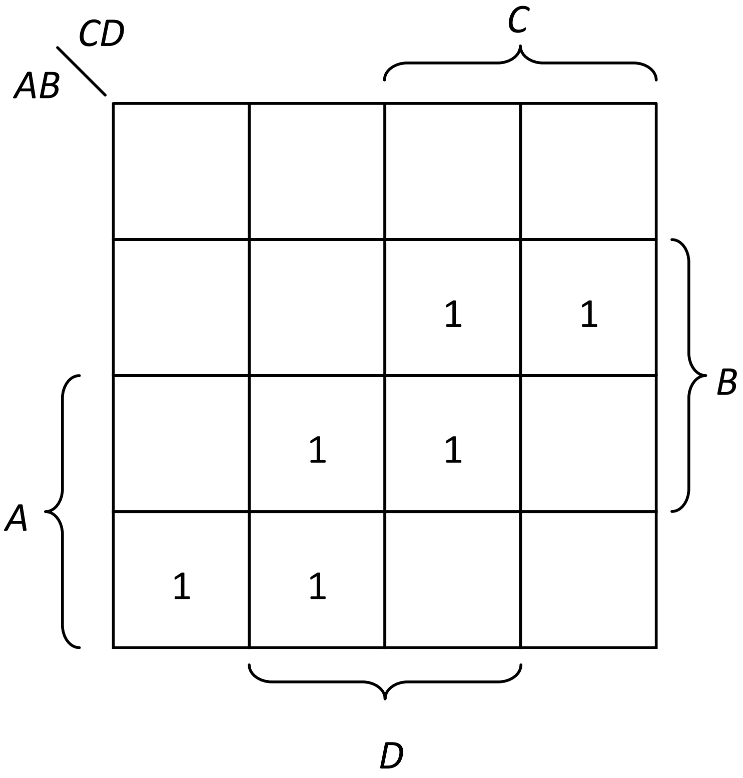
0	0	0	1
0	0	0	1
1	1	0	0
1	1	0	0

Example:

Find all prime implicants in each K-map.



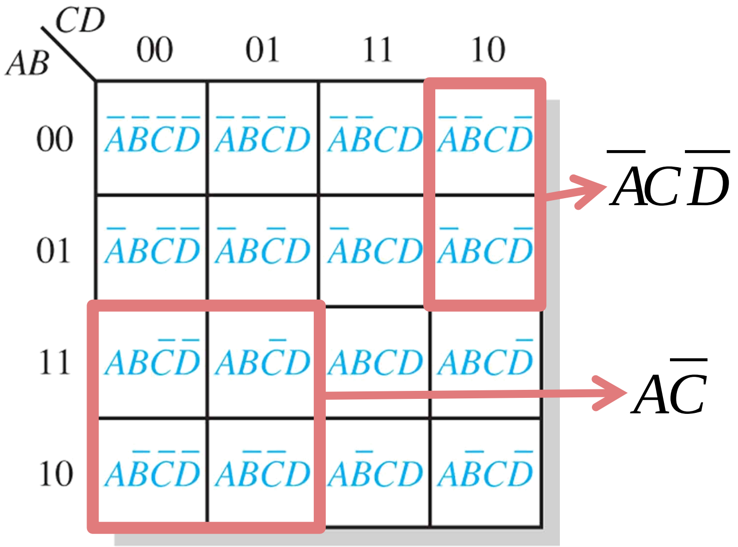
	1	1	
1	1	1	1
1		1	1
		1	



		1	1
	1	1	
1	1		

Prime Implicant to Product Term

- Each prime implicant is a product term
  - Composed of all variables that occur in only one form (either uncomplemented or complemented) within the group
  - Variables that occur both uncomplemented **and** complemented within the group are eliminated.
    - These are called **contradictory variables**.



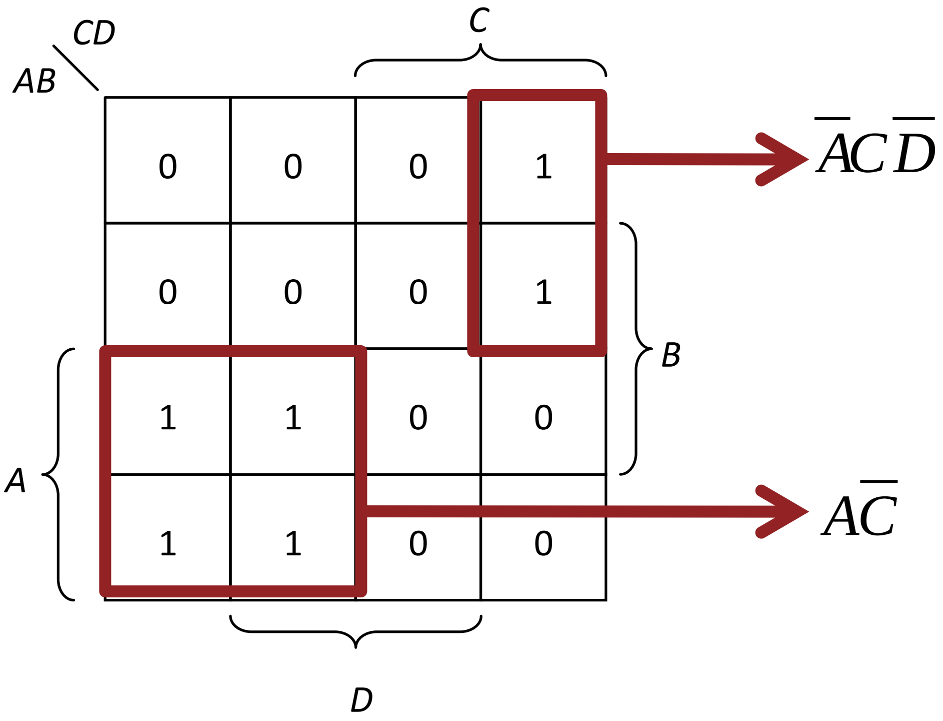
$AB \backslash CD$	00	01	11	10
00	$\overline{A}\overline{B}\overline{C}\overline{D}$	$\overline{A}\overline{B}\overline{C}D$	$\overline{A}\overline{B}CD$	$\overline{A}\overline{B}C\overline{D}$
01	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}BCD$	$\overline{A}BC\overline{D}$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

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

$A\overline{C}$

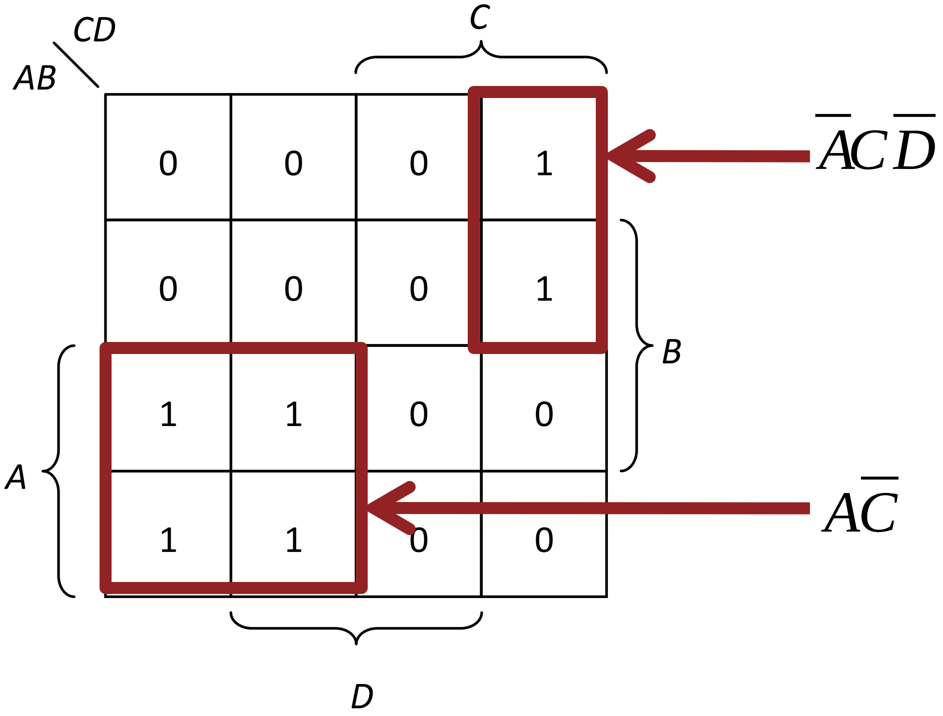
Prime Implicant to Product Term

Turn out that we can read the product term off the K-map directly



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

Product Term to K-Map

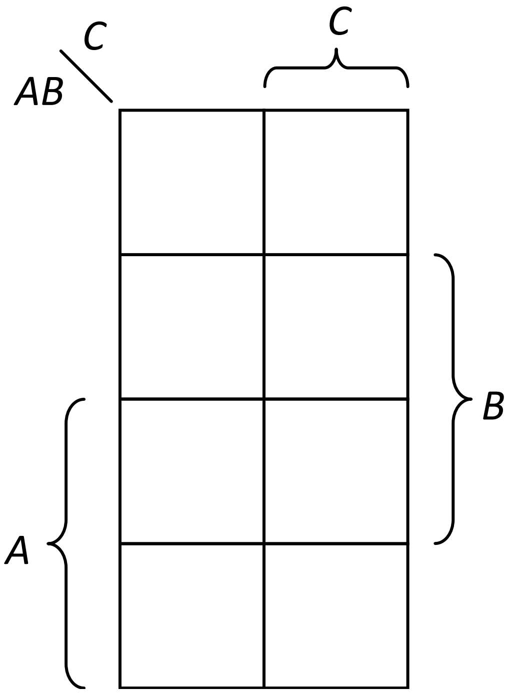


0	0	0	1
0	0	0	1
1	1	0	0
1	1	0	0

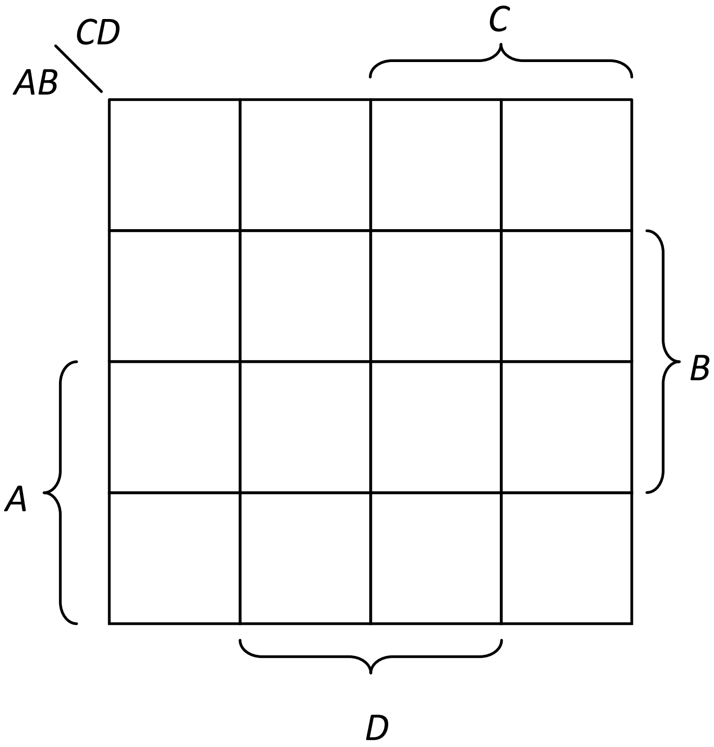
Mapping (SOP)

Map the following expression on a Karnaugh map

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



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



Minimal Sum

- **Definition**: A **minimal sum** is a SOP expression such that no equivalent SOP expression has fewer product terms, and any equivalent SOP expression with the same number of product terms has at least as many literals.
- **Prime-Implicant Theorem**:

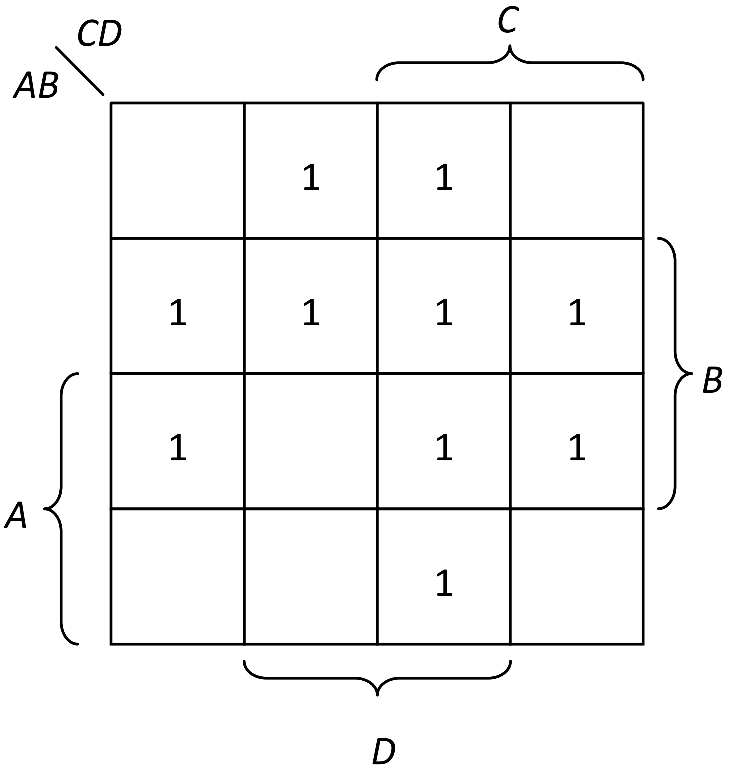
A minimum sum is a sum of prime implicants

K-Map to Minimal Sum

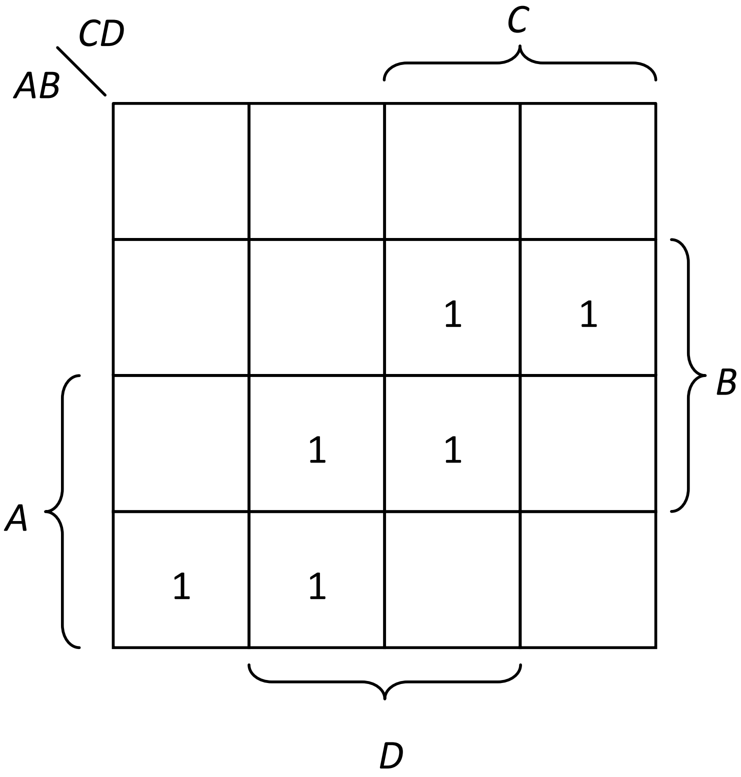
- Group the cells that have 1s according to the rules on earlier slide. This creates many prime implicants.
  - Each prime implicant creates one product term.
  - Each 1 on the map must be included in at least one prime implicant.
  - The sum of all the prime implicants of a logic function is called the **complete sum**.
    - It is a legitimate way to realize a logic function.
    - It's not always minimal.
- "Minimize the number of prime implicants."
- Add up all the "surviving" product terms

Example:

Find the minimal sum from each K-map.



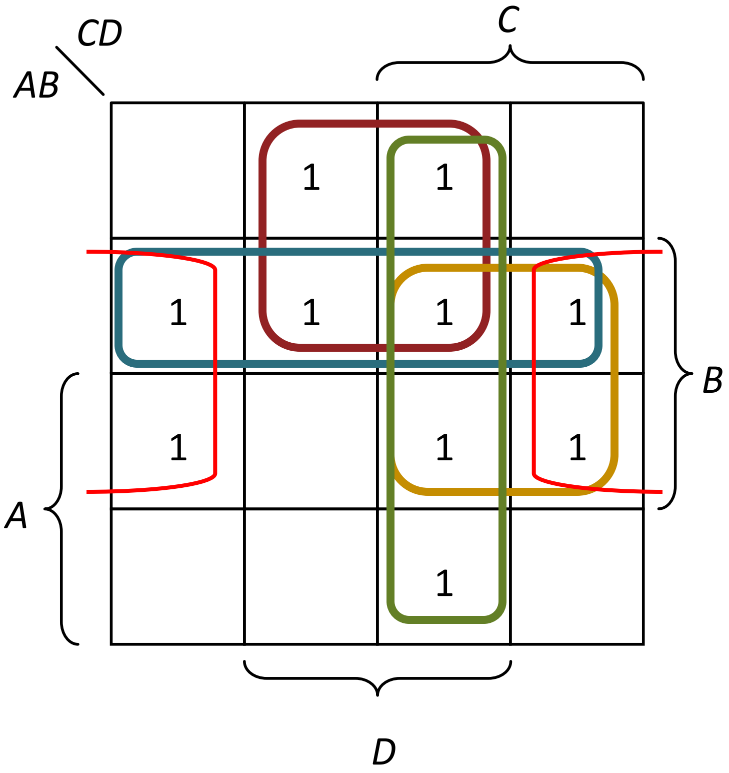
	1	1	
1	1	1	1
1		1	1
		1	



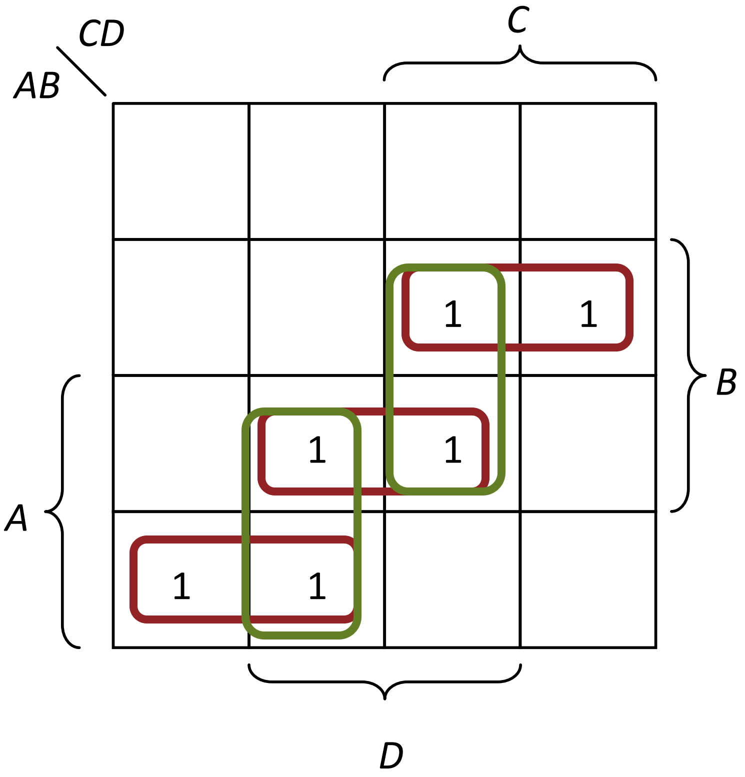
		1	1
	1	1	
1	1		

Example:

Here are all prime implicants.



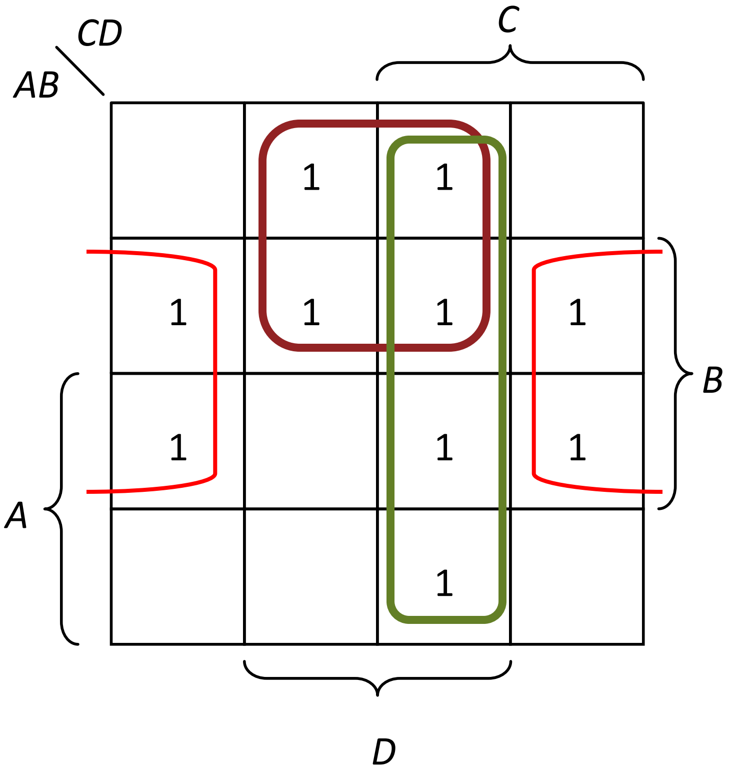
	1	1	
1	1	1	1
1		1	1
		1	



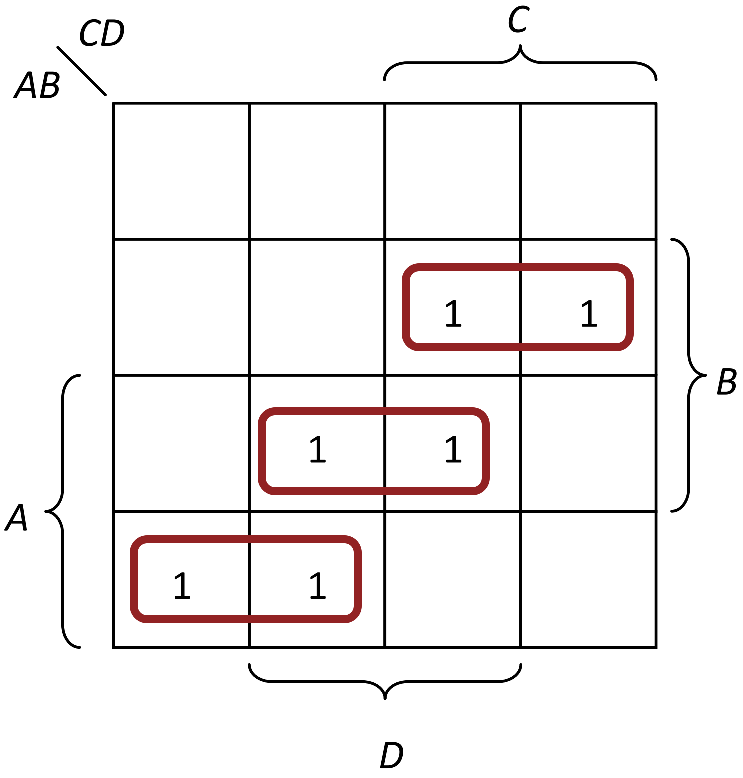
		1	1
	1	1	
1	1		

Example:

We need only these…

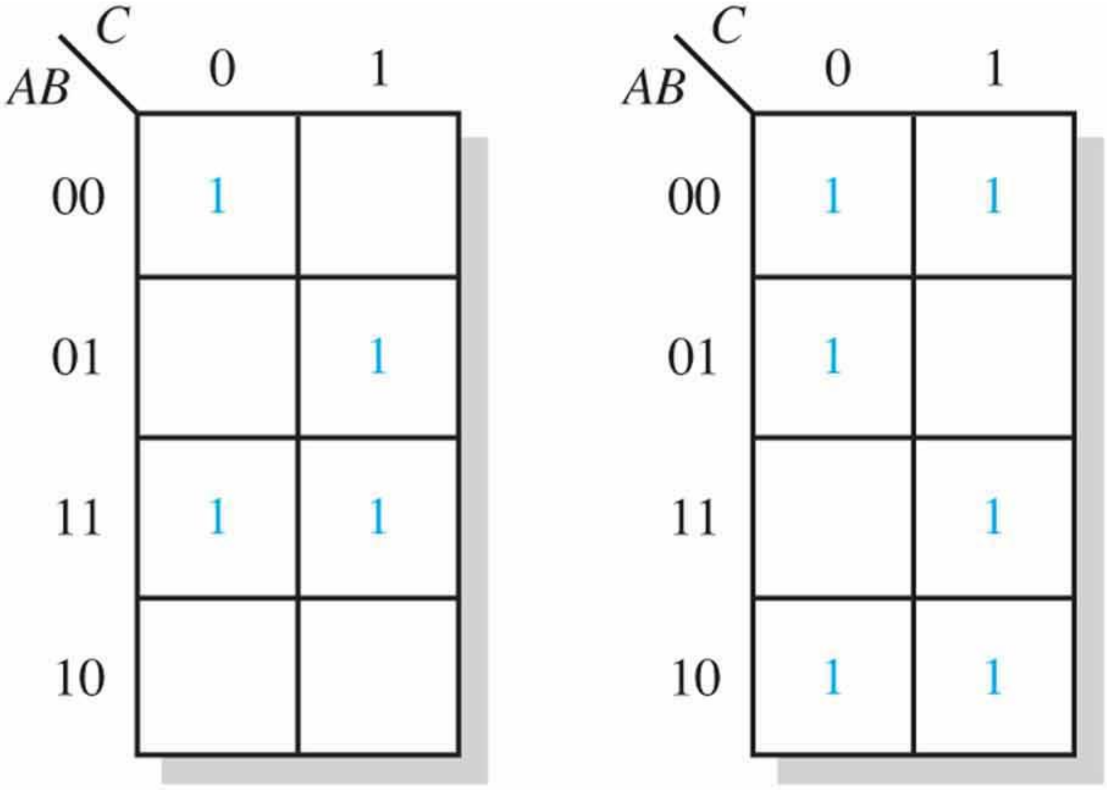


	1	1	
1	1	1	1
1		1	1
		1	



		1	1
	1	1	
1	1		

Example



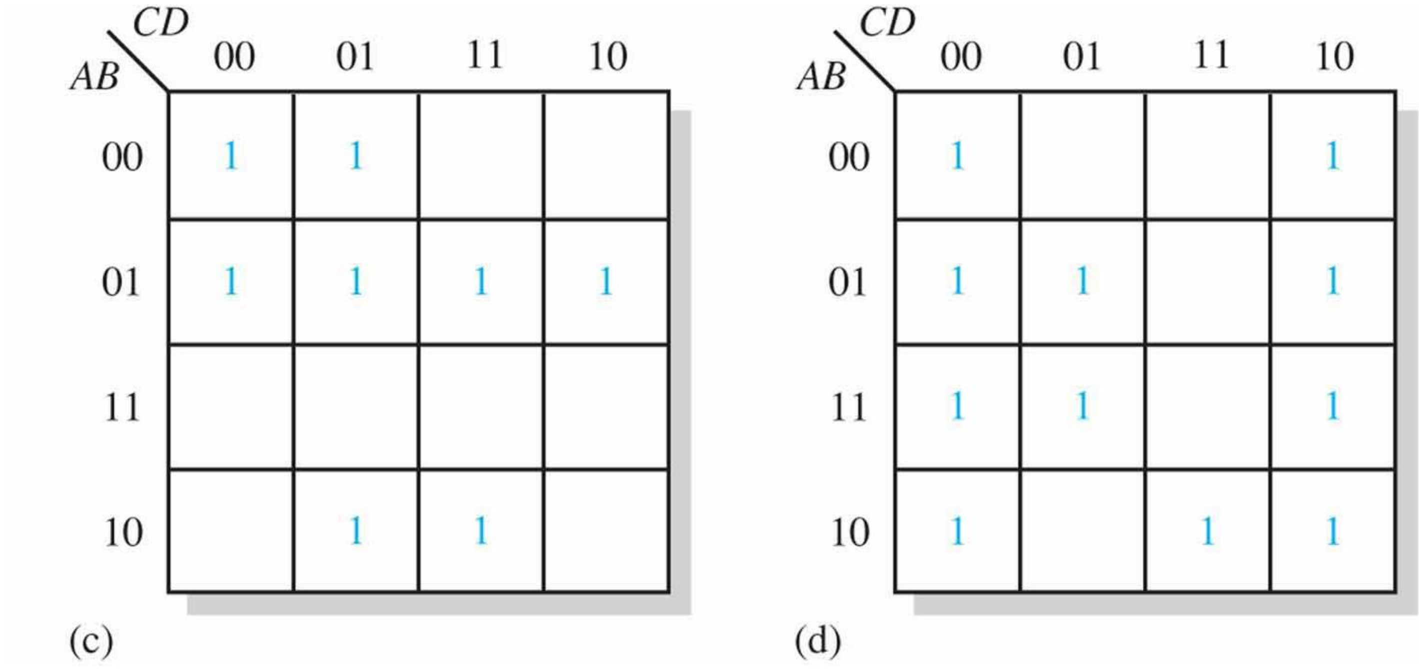
AB \ C	0	1
00	1	
01		1
11	1	1
10		

(a)

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

(b)

Example



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

(c)

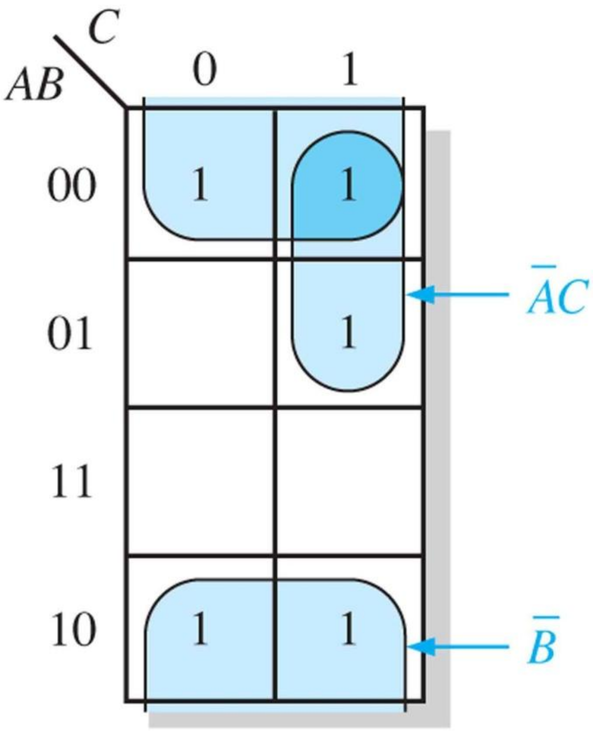
AB \ CD	00	01	11	10
00	1			1
01	1	1		1
11	1	1		1
10	1		1	1

(d)

Example

Use a K-map to minimize the following expression

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

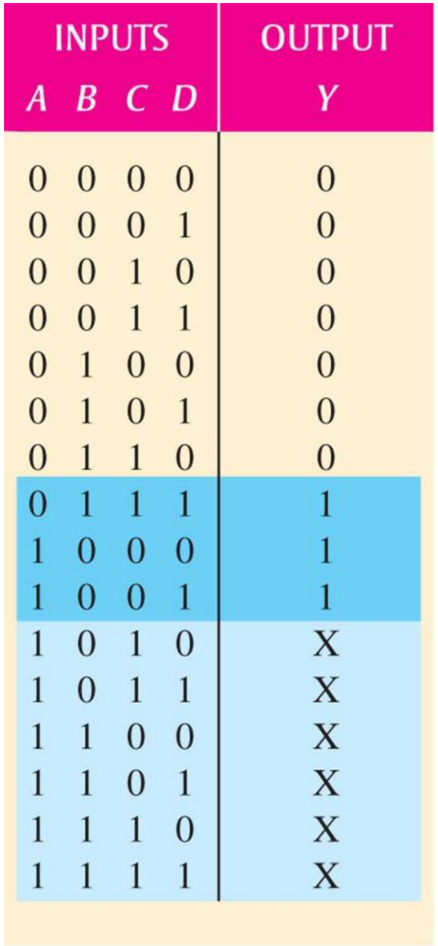


$$X = \overline{B} + AC$$

“Don’t Care” Input Combinations

- Sometimes the output doesn’t matter for certain input combinations.
  - For example, the combinations are not allowed in the first place.
- These combinations are called “don’t care”.
- The “don’t care” term can be used to advantage on K-map.
- For each “don’t care” term, place an X in the corresponding cell.
- When grouping the 1s,
  - the Xs can be treated as 1s to make a larger grouping
  - or as 0s if they cannot be used to advantage.

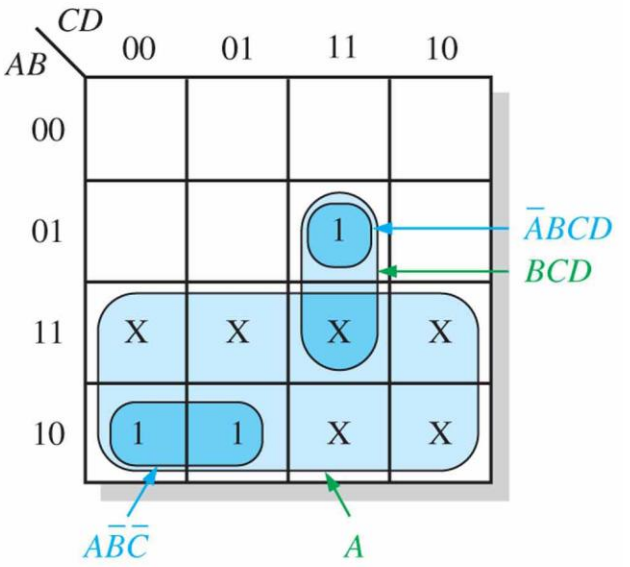
Example



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

Don't cares

(a) Truth table



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

(b) Without "don't cares" $Y = A\bar{B}\bar{C} + \bar{A}BCD$
With "don't cares" $Y = A + BCD$

Alternative Methods

- Disadvantages of using K-maps
  - Not applicable for more than five variables
  - Practical only for up to four variables
  - Difficult to automated in a computer program
- There are other ways to minimize Boolean functions.
  - More practical for more than four variables
  - Easily implemented with a computer

  1. Quine-McClusky method
     - Inefficient in terms of processing time and memory usage
  2. Espresso Algorithm
     - de facto standard

Canonical Product

- Product-of-Sums (POS) Form

  Example: $\left(A+\overline{B}\right)\cdot\left(A+B+C\right)$

- Standard POP Form (Canonical Product)

  Example: $\left(A+\overline{B}+C\right)\cdot\left(A+\overline{B}+\overline{C}\right)\cdot\left(A+B+C\right)$

- Convert expression in POS form into canonical product:

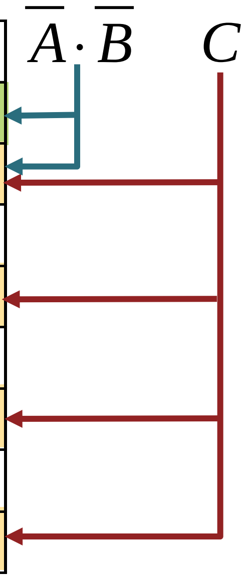
Example

Find the value of X for all possible values of the variables when

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

$$\begin{aligned} X &= \left(A+\overline{B}+C\right)\cdot\left(\overline{A}+B+C\right)\cdot\left(\overline{A}+\overline{B}+C\right) \\ &= \left(\left(A+\overline{B}\right)\cdot\left(\overline{A}+B\right)\cdot\left(\overline{A}+\overline{B}\right)\right)+C \\ &= \left(\left(A+\overline{B}\right)\cdot\left(\overline{A}+\left(B\cdot\overline{B}\right)\right)\right)+C \\ &= \left(\left(A+\overline{B}\right)\cdot\overline{A}\right)+C \\ &= \left(\overline{A}\cdot\overline{B}\right)+C \end{aligned}$$

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Example

Find the value of X for all possible values of the variables when

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

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

Maxterm

- A sum term in a canonical product is called a **maxterm**.
- A maxterm is equal to 0 for only one combination of variable values.

$$A+\overline{B}+C=0 \text{ iff } (A,B,C)=(0,1,0)$$

$$A+\overline{B}+\overline{C}=0 \text{ iff } (A,B,C)=(0,1,1)$$

$$A+B+C=0 \text{ iff } (A,B,C)=(0,0,0)$$

- We say that the max term $A+\overline{B}+C$ has a binary value of 010 (decimal 2)
- Maxterm list: $(A+\overline{B})\cdot(A+B+C)=\prod_{A,B,C}(0,2,3)$

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

Minterm/Maxterm & Truth Table

Row #	*A*	*B*	*C*	*Minterm*	*Maxterm*
0	0	0	0	$\overline{A}\cdot\overline{B}\cdot\overline{C}$	$A+B+C$
1	0	0	1	$\overline{A}\cdot\overline{B}\cdot C$	$A+B+\overline{C}$
2	0	1	0	$\overline{A}\cdot B\cdot\overline{C}$	$A+\overline{B}+C$
3	0	1	1	$\overline{A}\cdot B\cdot C$	$A+\overline{B}+\overline{C}$
4	1	0	0	$A\cdot\overline{B}\cdot\overline{C}$	$\overline{A}+B+C$
5	1	0	1	$A\cdot\overline{B}\cdot C$	$\overline{A}+B+\overline{C}$
6	1	1	0	$A\cdot B\cdot\overline{C}$	$\overline{A}+\overline{B}+C$
7	1	1	1	$A\cdot B\cdot C$	$\overline{A}+\overline{B}+\overline{C}$

Conversion

Row #	*A*	*B*	*C*	*Minterm*	*Maxterm*
0	0	0	0	$\overline{A}\cdot\overline{B}\cdot\overline{C}$	$A+B+C$
1	0	0	1	$\overline{A}\cdot\overline{B}\cdot C$	$A+B+\overline{C}$
2	0	1	0	$\overline{A}\cdot B\cdot\overline{C}$	$A+\overline{B}+C$
3	0	1	1	$\overline{A}\cdot B\cdot C$	$A+\overline{B}+\overline{C}$
4	1	0	0	$A\cdot\overline{B}\cdot\overline{C}$	$\overline{A}+B+C$
5	1	0	1	$A\cdot\overline{B}\cdot C$	$\overline{A}+B+\overline{C}$
6	1	1	0	$A\cdot B\cdot\overline{C}$	$\overline{A}+\overline{B}+C$
7	1	1	1	$A\cdot B\cdot C$	$\overline{A}+\overline{B}+\overline{C}$

$$\Sigma_{\mathrm{A,B,C}}(0,1,2,3) = \Pi_{\mathrm{A,B,C}}(4,5,6,7)$$

$$\Sigma_{\mathrm{X,Y}}(1) = \Pi_{\mathrm{X,Y}}(0,2,3)$$

$$\Sigma_{\mathrm{W,X,Y,Z}}(0,1,2,3,5,7,11,13) = \Pi_{\mathrm{W,X,Y,Z}}(4,6,8,9,10,12,14,15)$$