EEE130 Digital Electronics I
Lecture #4_3

- Karnaugh Map – POS -

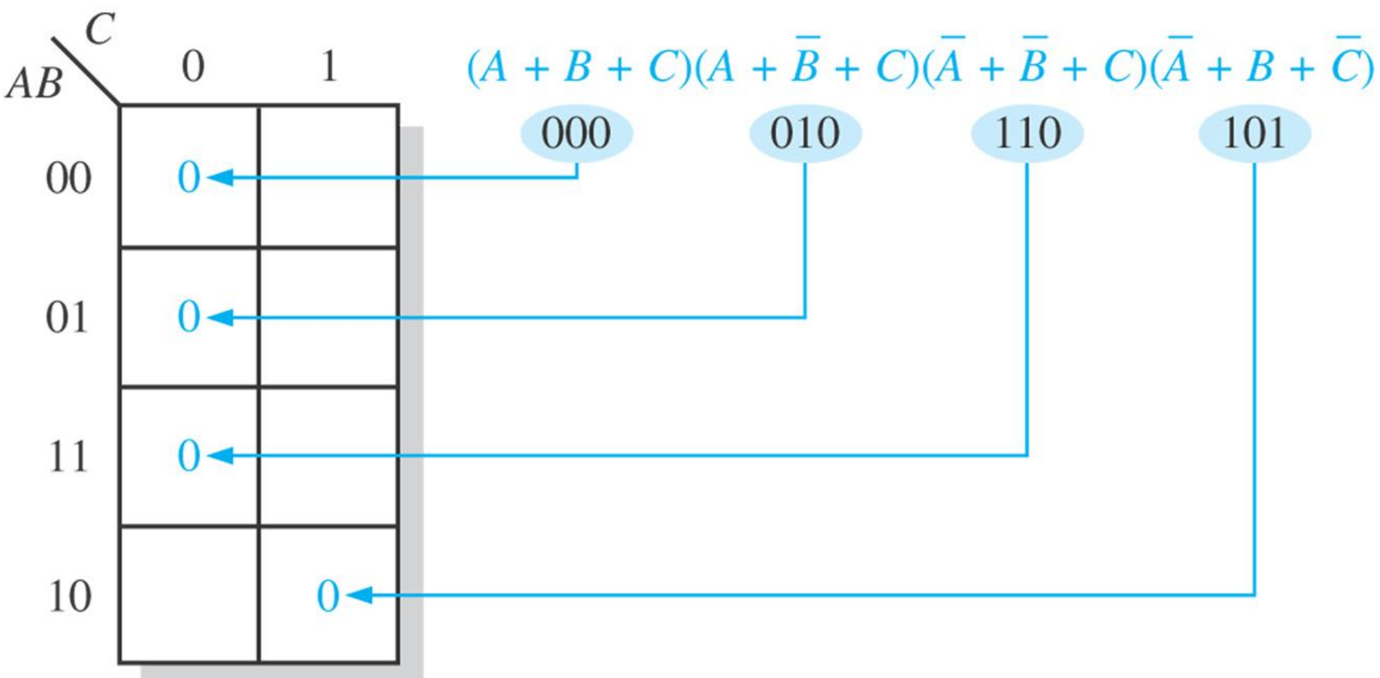
By Dr. Shahrel A. Suandi

4-10 Karnaugh Map POS Minimization

- In contrast to Karnaugh Map SOP Minimization, K-Map POS minimization will use 0s to represent the standard sum terms (instead of 1s)
  - Each 0 is placed in a cell corresponding to the value of a sum term
  - Example: $A + \bar{B} + C \longrightarrow 010$
- When a POS expression is completely mapped, there will be a number of 0s on the K-Map equal to the number of sum terms in the standard POS expression
  - The cells do not have a 0 → the expression is 1 (will be left off)

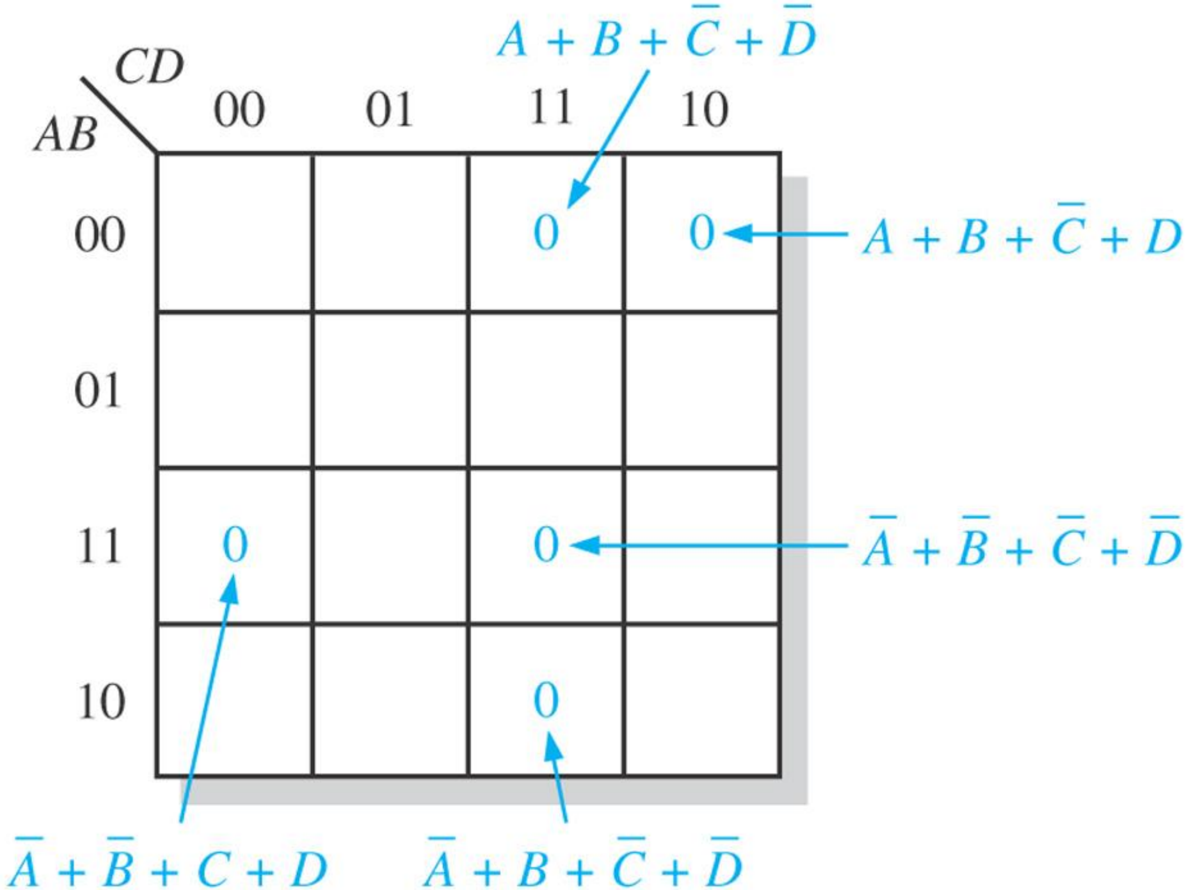
Steps to be taken to map POS expression

- Determine the binary value of each sum term in the standard POS expression. This is the binary value that makes the term equal to 0
- As each sum term is evaluated, place a 0 on the K-Map in the corresponding cell



Example 4-30

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



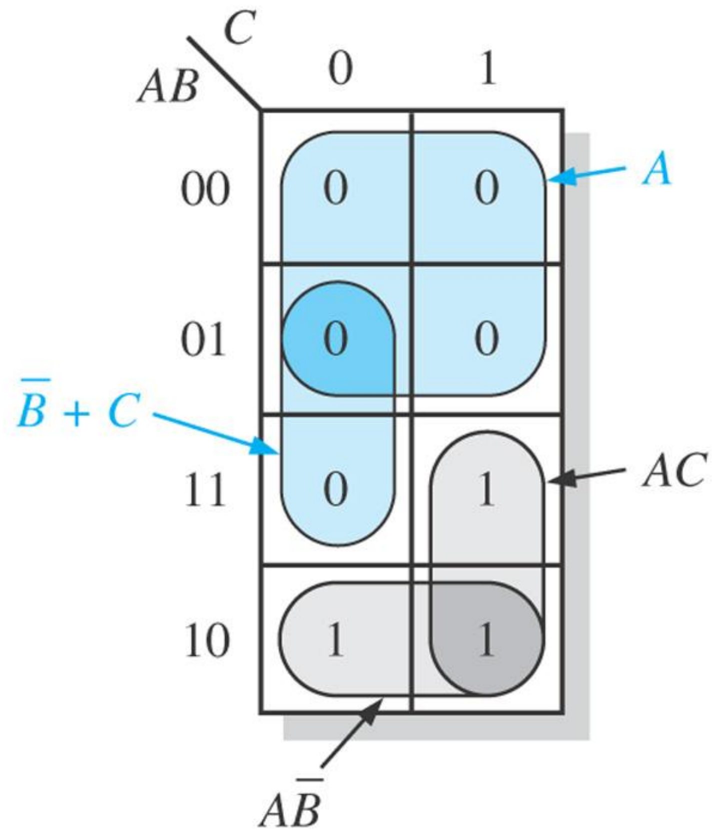
Example 4-30 (Related Problem)

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

$AB \backslash CD$	00	01	11	10
00	0	0		
01				0
11				
10				0

K-Map Simplification of POS Expressions

- Similar to minimizing SOP method, we group the 0s instead of 1s with the same condition
  - Maximizing the cells in a group
- Let's discuss Example 4-31



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

Important information from the K map

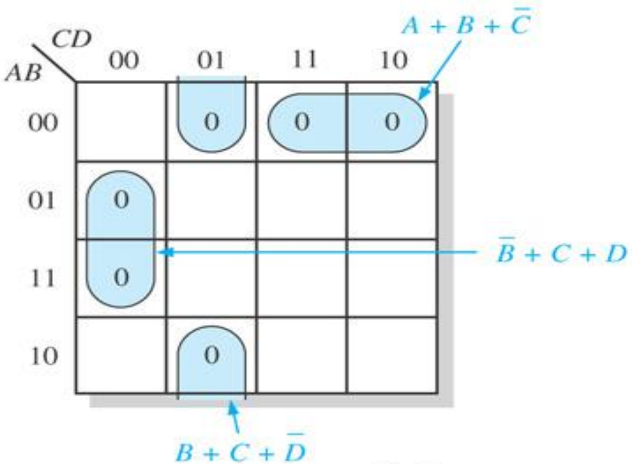
- If we know the POS, then we may know the SOP
- As in the example (Ex. 4-31), we may notice that

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

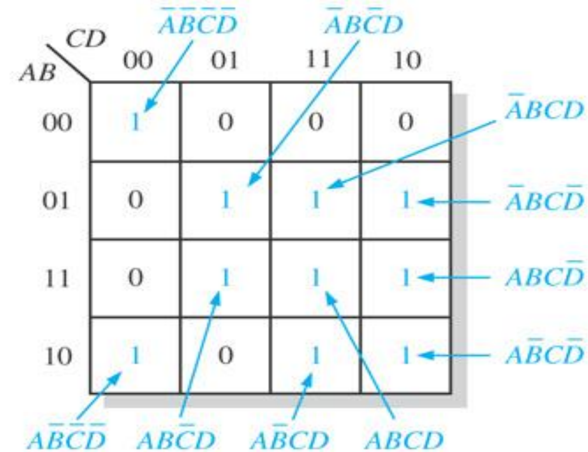
Converting between POS and SOP using the K-Map

- Advantage of using the results of this conversion:
  - Seeking which one is simpler (minimum forms so that fewer gates can be used)
- If we have a POS K-Map, then we may get/know the SOP expressions
  - Vice versa, if we have a SOP K-Map
- Let’s do example 4-33

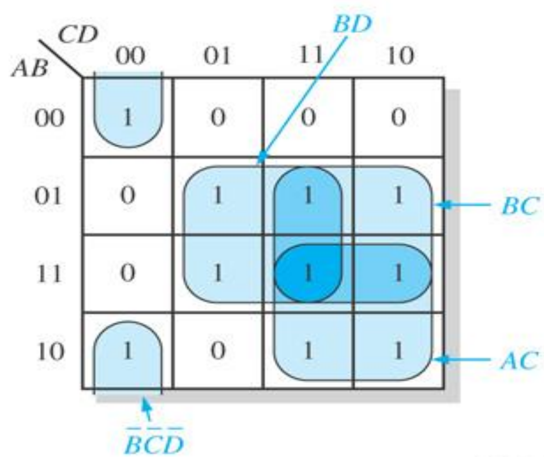
Example 4-33



(a) Minimum POS: $(A + B + C)(\overline{B} + C + D)(B + C + \overline{D})$



(b) Standard SOP:
$\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BCD + \overline{A}BC\overline{D} + ABC\overline{D} + A\overline{B}C\overline{D} +$
$A\overline{B}\overline{C}\overline{D} + AB\overline{C}D + A\overline{B}CD + ABCD$



(c) Minimum SOP: $AC + BC + BD + \overline{B}\overline{C}\overline{D}$

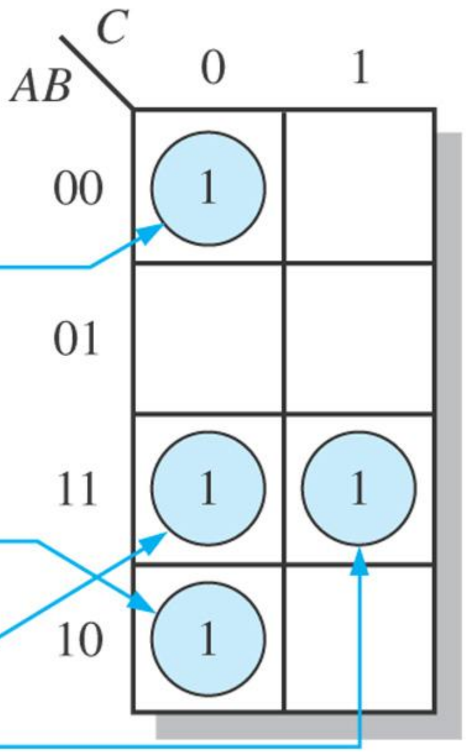
Other important things to know

- Mapping directly from a truth table
  - Please refer to Figure 4-35 for the illustration of doing this
  - we can map the 1s into the correct cell in K-Map by looking at the inputs
- "Don't care" conditions
  - Situations where some input variable combinations are not allowed (example: BCD has only 10 combinations… NOT 16)
  - Don't care means we can use either 1 or 0
  - Look at Figure 4-36 – "don't care" conditions are used to advantage on the K-Map

Mapping Directly From Truth Table (Figure 4-35)

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

Inputs			Output
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

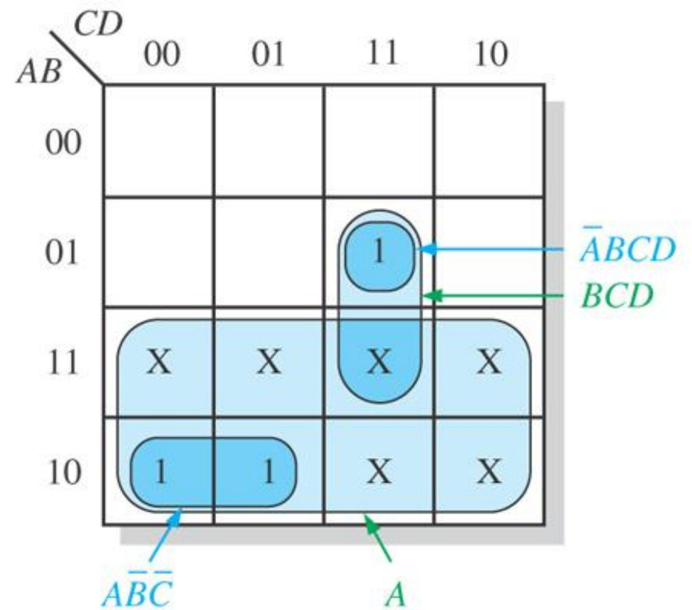


"Don't care" conditions (Figure 4-36)

Inputs $ABCD$	Output 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



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

Summary

- Why do we use K-Map (Karnaugh Map)?
  - To simplify Boolean expressions
  - When we have simplified the expressions, we can minimize the logic gates used
  - There are two expressions concerned here, SOP and POS
- Why we need to know SOP and POS minimization technique?
  - As mentioned earlier, we want to simplify the Booelan expression
  - Advantage of using K-Map in terms of minimization technique:
    - When we know SOP, we will also know POS, and vice versa