K-Map

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

A different way to draw a truth table !

Take advantage of adjacency

A \ B	0	1
0	0 A’ B’	1 A’ B
1	0 A B’	1 A B

$F = A'B + AB = B$

Keep common literal only!

Minimization with K-maps

1. Draw a K-map
2. Combine maximum number of 1's following rules:
    1. Only adjacent squares can be combined
    2. All 1's must be covered
    3. Covering rectangles must be of size 1,2,4,8, … 2^n
3. Check if all covering are really needed
4. Read off the SOP expression

2-variable K-map

Given a function with 2 variables: F(X,Y), the total number of minterms are equal to 4:

m_0, m_1, m_2, m_3

The size of the k-map is always equal to the total number of minterms.

Each entry of the k-map corresponds to one minterm for the function:

Row 0 represents: X'Y', X'Y

Row 1 represents: XY', XY'

X \ Y	0	1
0	m_0	m_1
1	m_2	m_3

X \ Y	0	1
0	*0*	*1*
1	*2*	*3*

Example 1

For a given function F(X,Y) with the following truth table, minimize it using k-maps

X	Y	F
0	0	0
0	1	0
1	0	1
1	1	1

X \ Y	0	1
0	0	0
1	1	1

Combining all the 1's in only the adjacent squares

The final reduced expression is given by the common literals from the combination:

Therefore, since for the combination, Y has different values (0, 1), and X has a fixed value of 1,

The reduced function is: F(X,Y) = X

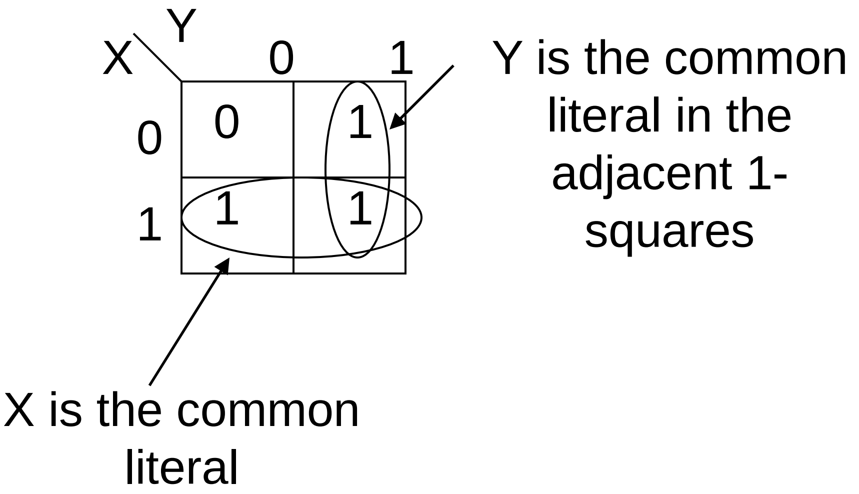
Example 2

Q. Simplify the function $F(X,Y) = \sum m(1,2,3)$

Sol. This function has 2 variables, and three 1-squares (three minterms where function is 1)

$$F = m_1 + m_2 + m_3$$

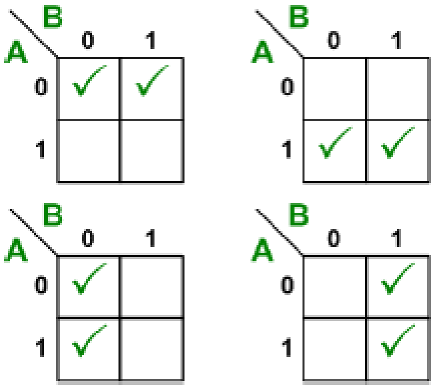
Note: The 1-squares can be combined more than once

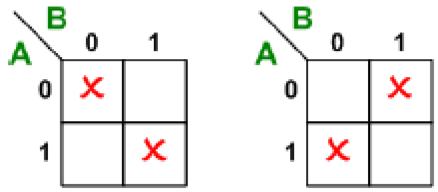


Minimized expression: F = X + Y

2 variable K-Maps (Adjacency)

In an n-variable k-map, each square is adjacent to exactly *n* other squares





Q: What if you have 1 in all squares?

3-variable K-maps

For 3-variable functions, the k-maps are larger and look different.

Total number of minterms that need to be accommodated in the k-map = 8

To maintain adjacency neighbors don't have more than 1 different bit

	A \ BC	00	01	11	10
				B	B
	0	m_0	m_1	m_3	m_2
A	1	m_4	m_5	m_7	m_6
			C	C	

3-variable K-maps

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

A \ BC	00	01	11	10
0	$\mathbf{m_0}$	m_1	m_3	$\mathbf{m_2}$
1	$\mathbf{m_4}$	m_5	m_7	$\mathbf{m_6}$

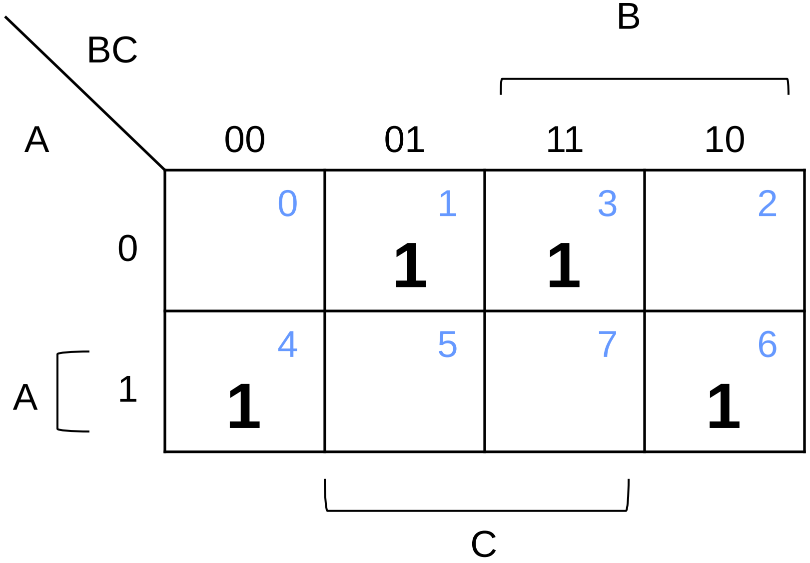
Note: You can only combine a power of 2 adjacent 1-squares. For e.g. 2, 4, 8, 16 squares. You cannot combine 3, 7 or 5 squares

Minterms m_o, m_2, m_4, m_6 can be combined as m_0 and m_2 are adjacent to each other, m_4 and m_6 are adjacent to each other

m_o and m_4 are also adjacent to each other, m_2 and m_6 are also adjacent to each other

Example 1

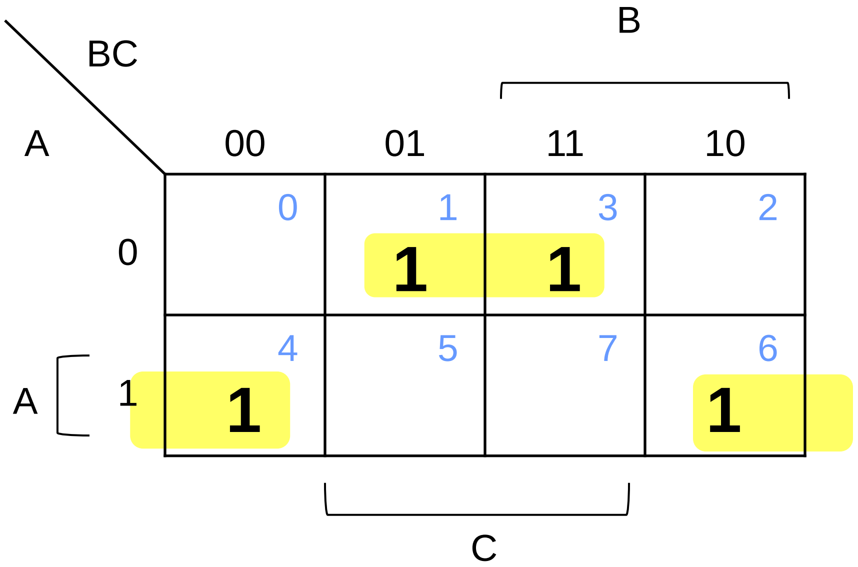
Simplify F = $\sum m(1, 3, 4, 6)$ using K-map



Example 1

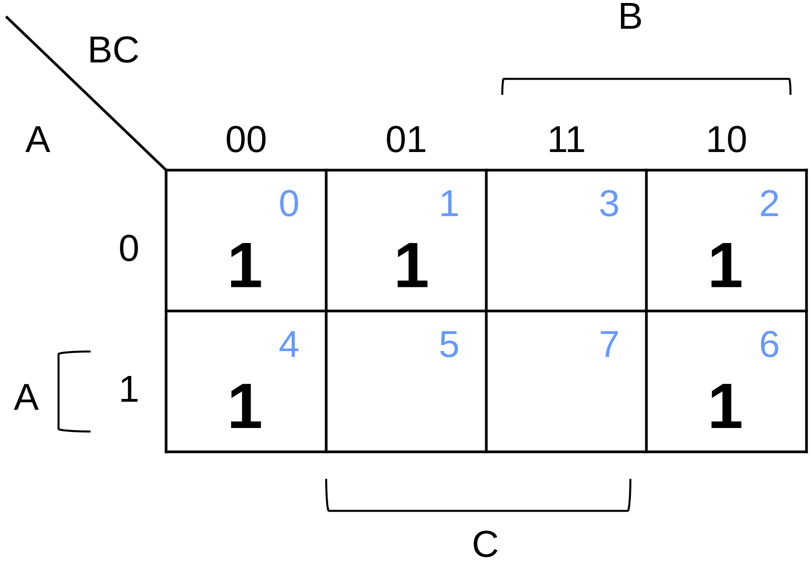
Simplify F = $\sum m(1, 3, 4, 6)$ using K-map

$F = A'C + AC'$



Example 2

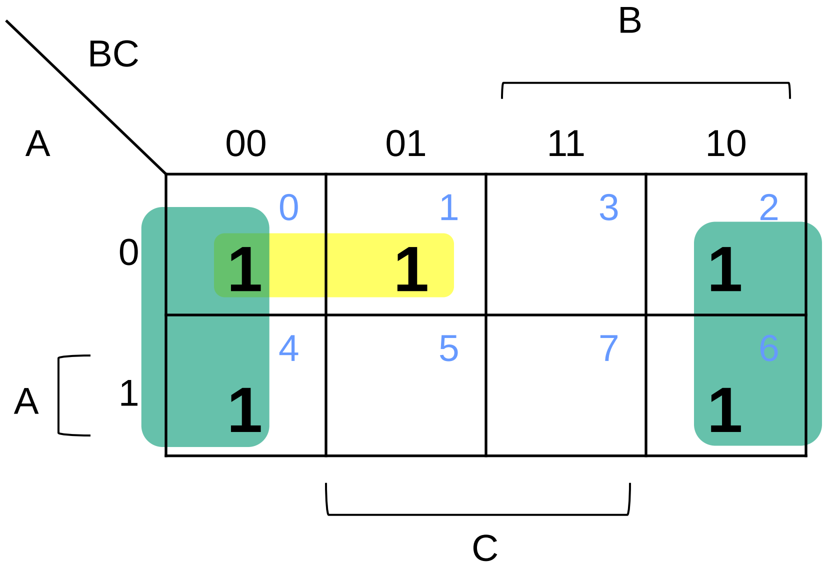
Simplify F = $\sum m(0,1, 2, 4, 6)$ using K-map



Example 2

Simplify F = $\sum m(0,1, 2, 4, 6)$ using K-map

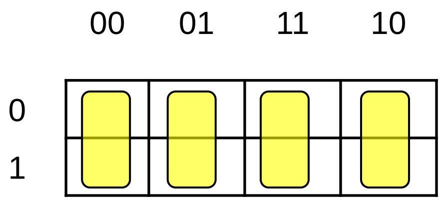
F = A' B' + C'

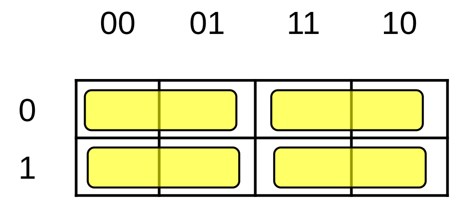


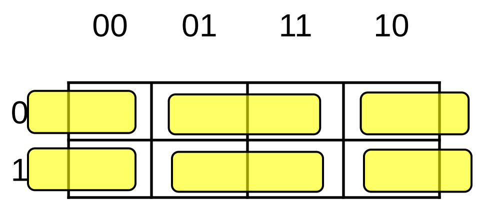
3 variable K-Maps (Adjacency)

A 3-variable map has 12 possible groups of 2 minterms

They become product terms with 2 literals



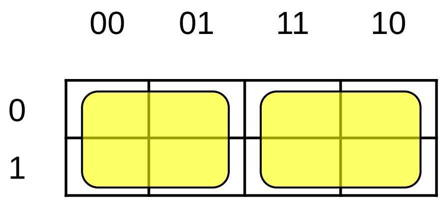


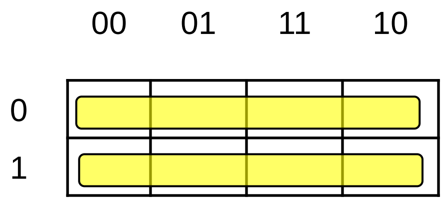


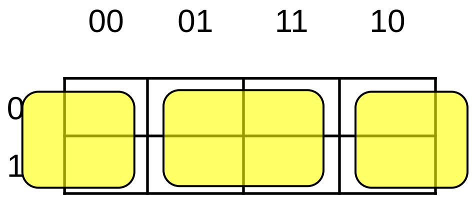
3 variable K-Maps (Adjacency)

A 3-variable map has 6 possible groups of 4 minterms

They become product terms with 1 literals







4-variable K-maps

A 4-variable function will consist of 16 minterms and therefore a size 16 k-map is needed

Each square is adjacent to 4 other squares

A square by itself will represent a minterm with 4 literals

Combining 2 squares will generate a 3-literal output

Combining 4 squares will generate a 2-literal output

Combining 8 squares will generate a 1-literal output

				C	C	
	AB \ CD	00	01	11	10	
	00	m_0	m_1	m_3	m_2	
	01	m_4	m_5	m_7	m_6	B
A	11	m_{12}	m_{13}	m_{15}	m_{14}	B
A	10	m_8	m_9	m_{11}	m_{10}	
			D	D		

4-variable K-maps (Adjacency)

AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

Note: You can only combine a power of 2 adjacent 1-squares. For e.g. 2, 4, 8, 16 squares. You cannot combine 3, 7 or 5 squares

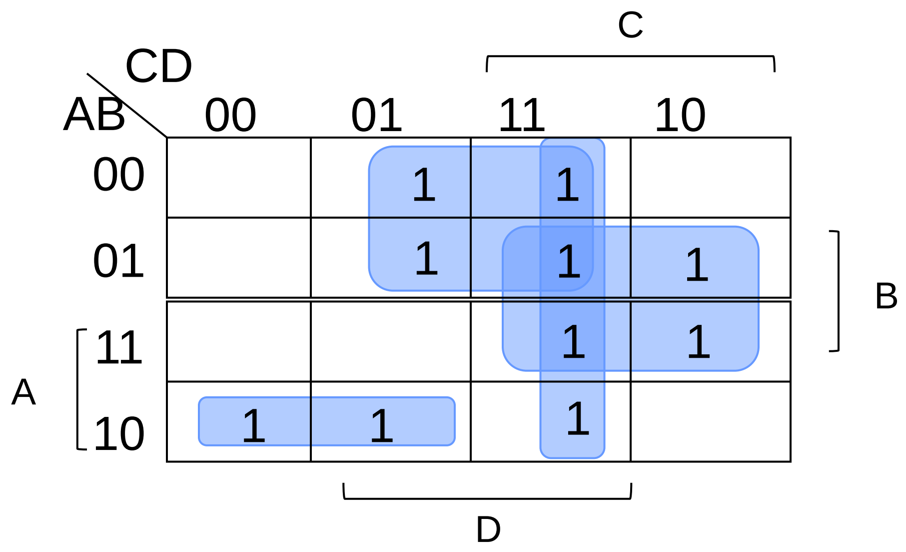
Right column and left column are adjacent; can be combined

Top row and bottom column are adjacent; can be combined

Many possible 2, 4, 8 groupings

Example

Minimize the function $F(A,B,C,D)=\sum m(1,3,5,6,7,8,9,11,14,15)$



F = CD + A'D + BC + AB'C'

Example

$F(A,B,C,D) = \Sigma m(0,1,2,5,8,9,10)$

C=1

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

B=1

A=1

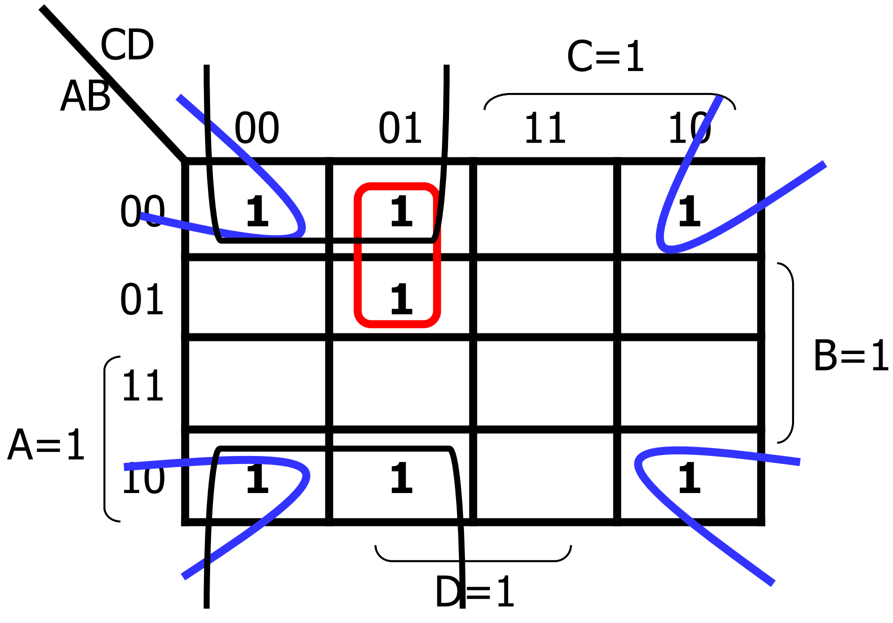
D=1

Example

F(A,B,C,D) = Σm(0,1,2,5,8,9,10)

Solution:

F = B' D' + B' C' + A' C' D



Example (POS)

$F(A,B,C,D) = \Sigma m(0,1,2,5,8,9,10)$

Write F in the simplified product of sums (POS)

Two methods?

You already know one!

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

C=1

B=1

A=1

D=1

Example (POS)

F(A,B,C,D) = Σm(0,1,2,5,8,9,10)

Write F in the simplified product of sums (POS)

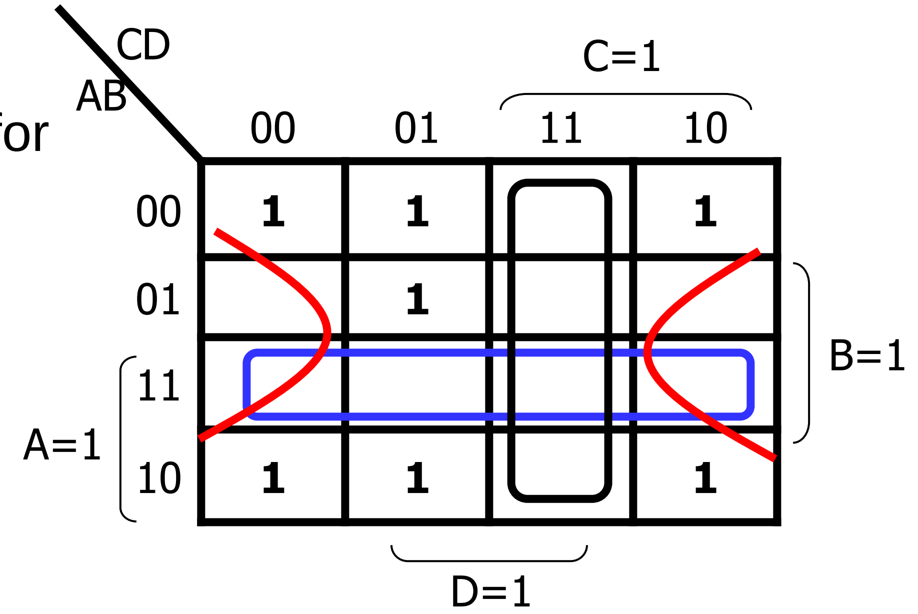
Method 2:

Follow same rule as before but for the ZEROs

F' = AB + CD + BD'

Therefore,

F'' = F = (A'+B')(C'+D')(B'+D)



Don’t Cares

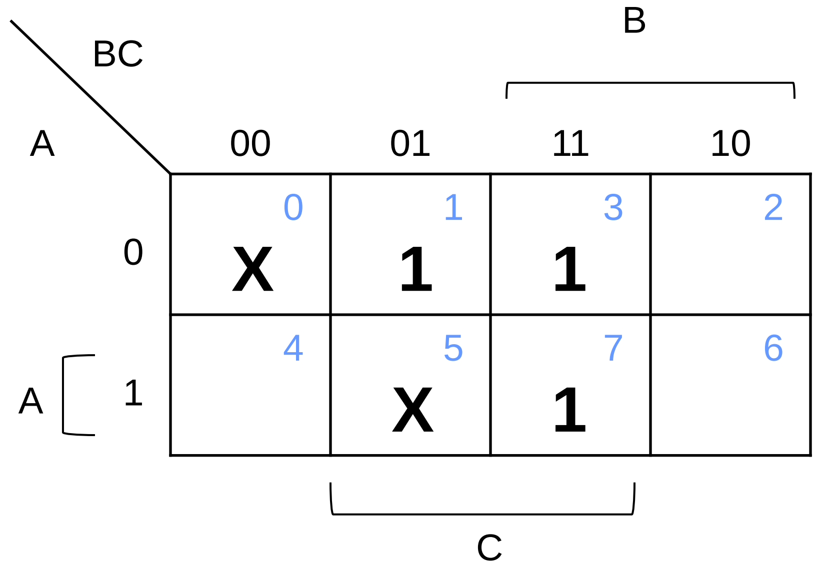
- In some cases, the output of the function (1 or 0) is not specified for certain input combinations either because
  - The input combination never occurs (Example BCD codes), or
  - We don’t care about the output of this particular combination
- Such functions are called incompletely specified functions
- Unspecified minterms for these functions are called don’t cares
- While minimizing a k-map with don’t care minterms, their values can be selected to be either 1 or 0 depending on what is needed for achieving a minimized output.

Example

$$F = \sum m(1, 3, 7) + \sum d(0, 5)$$

Circle the x's that help get bigger groups of 1's (or 0's if POS).

Don't circle the x's that don't help.

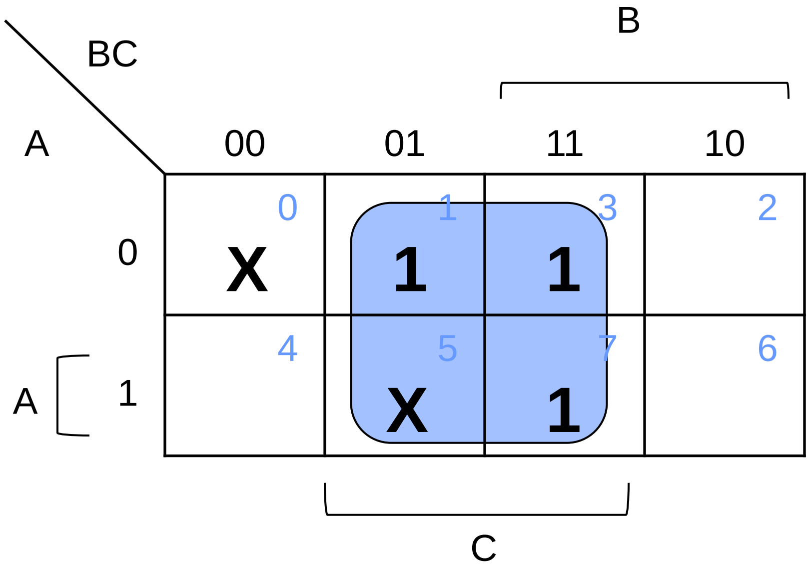


Example

$F = \sum m(1, 3, 7) + \sum d(0, 5)$

Circle the x's that help get bigger groups of 1's (or 0's if POS).

Don't circle the x's that don't help.



F = C

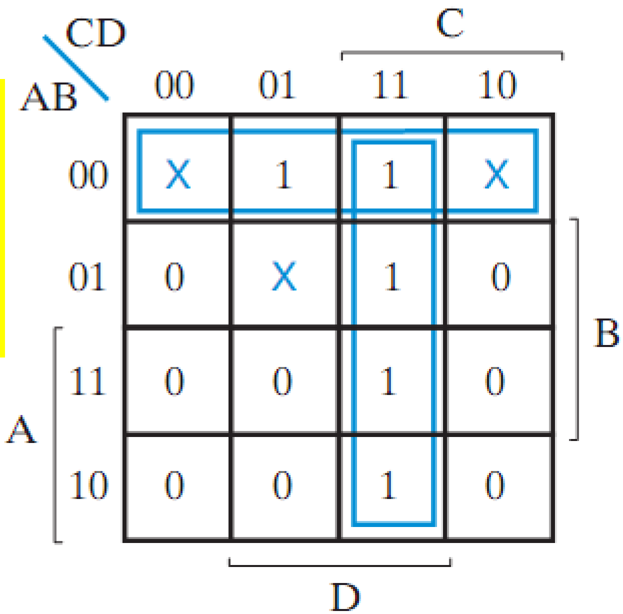
Example 2

$F(A, B, C, D) = \sum m(1, 3, 7, 11, 15) + \sum d(0, 2, 5)$

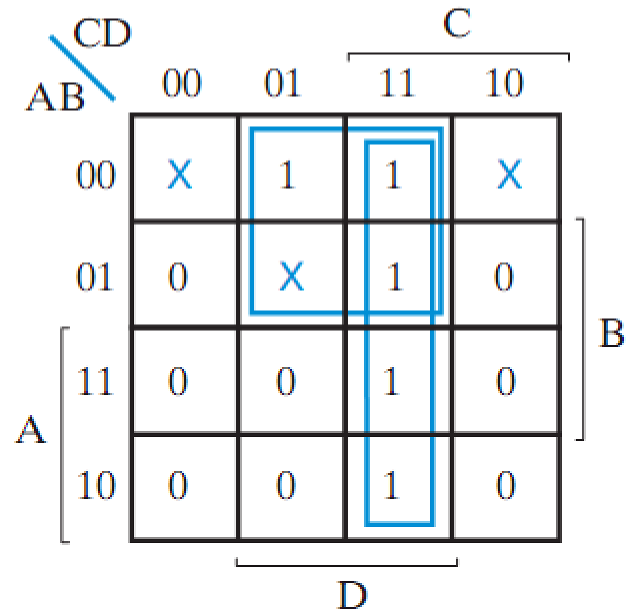
-Two possible solutions!

-Both acceptable.

-All 1's covered



(a) $F = CD + \overline{A}\,\overline{B}$



(b) $F = CD + \overline{A}D$

Src: Mano's Textbook