Karnaugh Maps (K-Maps)

Karnaugh Maps (K-Maps)

- A visual way to simplify logic expressions
- It gives the most simplified form of the expression

Karnaugh Maps

- Algebraic procedures:
  - Difficult to apply in a systematic way.
  - Difficult to tell when you have arrived at a minimum solution.

Karnaugh Maps

- Karnaugh map (K-map) can be used to minimize functions of up to 6 variables.
    - K-map is directly applied to two-level networks composed of AND and OR gates.
        - Sum-of-products, (SOP)
        - Product-of-sum, (POS).

Rules to obtain the most simplified expression

- Simplification of logic expression using Boolean algebra is awkward because:
  - it lacks specific rules to predict the most suitable next step in the simplification process
  - it is difficult to determine whether the simplest form has been achieved.
- A Karnaugh map is a graphical method used to obtained the most simplified form of an expression in a standard form (Sum-of-Products or Product-of-Sums).

Rules to obtain the most simplified expression

- The simplest form of an expression is the one that has the minimum number of terms with the least number of literals (variables) in each term.
- By simplifying an expression to the one that uses the minimum number of terms, we ensure that the function will be implemented with the minimum number of gates.

Rules to obtain the most simplified expression

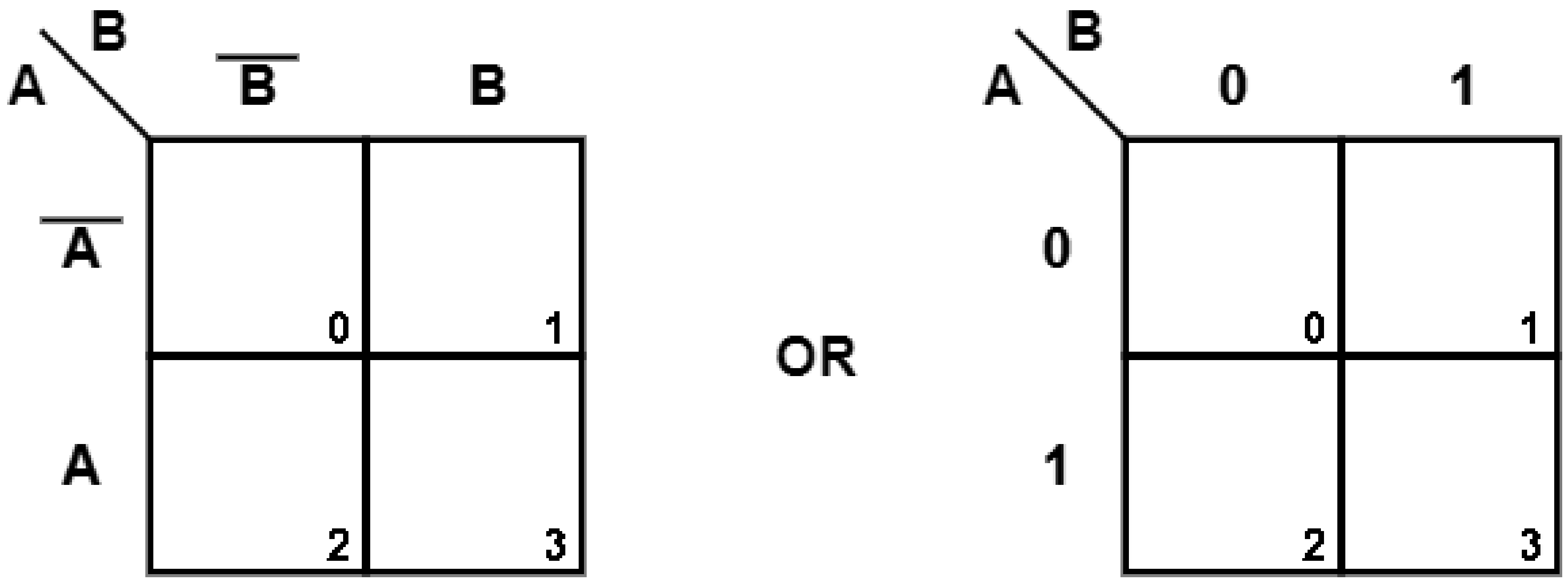
- By simplifying an expression to the one that uses the least number of literals for each terms, we ensure that the function will be implemented with gates that have the minimum number of inputs.

2-Variable K-map

- Place 1s and 0s from the truth table in the K-map.
- Each square of 1s = minterms.
- Minterms in adjacent squares can be combined since they differ in only one variable. Use XY’ + XY = X.

- Two variable K Map is drawn for a boolean expression consisting of two variables.
- The number of cells present in two variable K Map = 2^2 = 4 cells.

Two variable K Map may be represented as-



Two Variable K Map

K-map Simplification for Two Variables

The rules of Kmap simplification are:

- Groupings can contain only 1s; no 0s.
- Groups can be formed only at right angles; diagonal groups are not allowed.
- The number of 1s in a group must be a power of 2 – even if it contains a single 1.
- The groups must be made as large as possible.
- Groups can overlap and wrap around the sides of the Kmap.

Description of K-maps and Terminology

- A Kmap has a cell for each minterm.
- This means that it has a cell for each line for the truth table of a function.
- The truth table for the function $F(x,y) = xy$ is shown at the right along with its corresponding Kmap.

F(X,Y) = XY

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X \ Y	0	1
0	0	0
1	0	1

Description of Kmaps and Terminology

- As another example, we give the truth table and KMap for the function, $F(x,y) = x + y$ at the right.
- This function is equivalent to the OR of all of the minterms that have a value of 1. Thus:

$$F(x,y) = x + y = \overline{x}y + x\overline{y} + xy$$

F(X,Y) = X+Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

X \ Y	0	1
0	0	1
1	1	1

Kmap Simplification for Two Variables

- Of course, the minterm function that we derived from our Kmap was not in simplest terms.
  - That’s what we started with in this example.
- We can, however, reduce our complicated expression to its simplest terms by finding adjacent 1s in the Kmap that can be collected into groups that are powers of two.

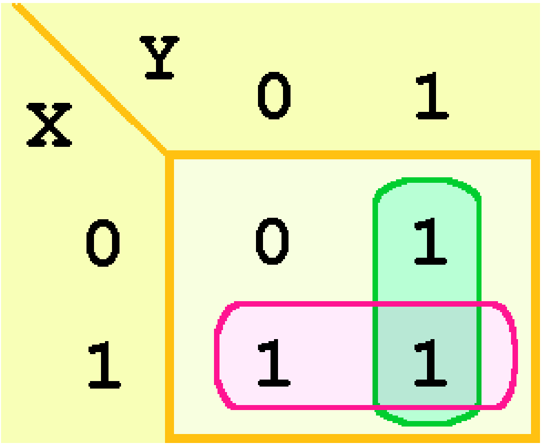
- In our example, we have two such groups.
  – Can you find them?

X \ Y	0	1
0	0	1
1	1	1

Kmap Simplification for Two Variables

- The best way of selecting two groups of 1s form our simple Kmap is shown below.
- We see that both groups are powers of two and that the groups overlap.
- The next slide gives guidance for selecting Kmap groups.

$$F(x,y) = x + y$$



3-Variable K-map

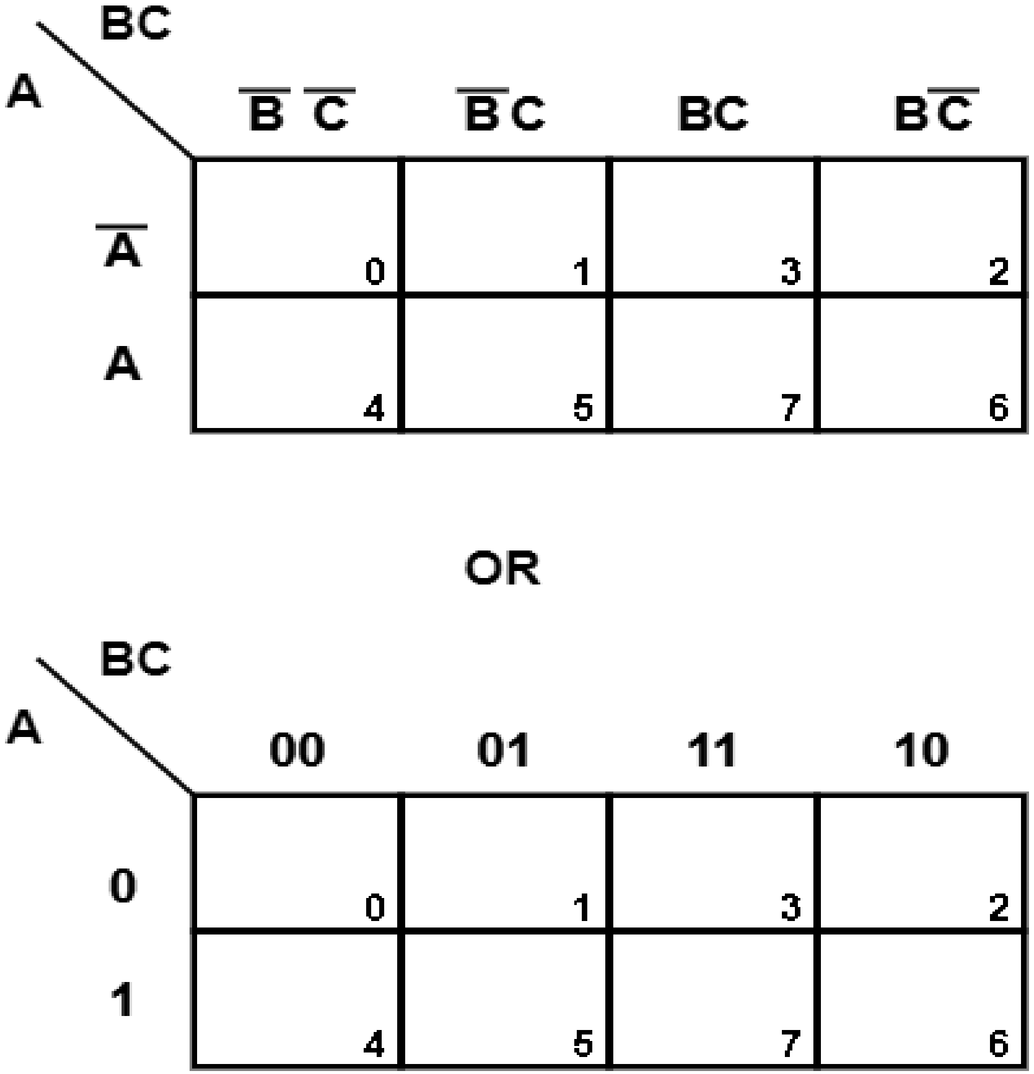
- Note BC is listed in the order of 00, 01, 11, 10. (Gray code)
- Minterms in adjacent squares that differ in only one variable can be combined using XY’ + XY = X.
- Three variable K Map is drawn for a Boolean expression consisting of three variables.
- The number of cells present in three variable K Map = 2^3 = 8 cells.

Kmap Simplification for Three Variables

- A Kmap for three variables is constructed as shown in the diagram below.
- We have placed each minterm in the cell that will hold its value.
  - Notice that the values for the *yz* combination at the top of the matrix form a pattern that is not a normal binary sequence.

x \ yz	00	01	11	10
0	$\overline{x}\,\overline{y}\,\overline{z}$	$\overline{x}\,\overline{y}z$	$\overline{x}yz$	$\overline{x}y\overline{z}$
1	$x\overline{y}\,\overline{z}$	$x\overline{y}z$	xyz	$xy\overline{z}$

- Three variable K Map may be represented as-



Three Variable K Map

Kmap Simplification for Three Variables

- Consider the function:

$$F(X,Y) = \overline{X}\,\overline{Y}Z + \overline{X}YZ + X\overline{Y}Z + XYZ$$

- Its Kmap is given below.
  - What is the largest group of 1s that is a power of 2?

X \ YZ	00	01	11	10
0	0	1	1	0
1	0	1	1	0

Three-Variable K-Maps

$$f = \sum(0,4) = \overline{B}\,\overline{C}$$

A \ BC	00	01	11	10
0	**1**	**0**	**0**	**0**
1	**1**	**0**	**0**	**0**

$$f = \sum(4,5) = A\,\overline{B}$$

A \ BC	00	01	11	10
0	**0**	**0**	**0**	**0**
1	**1**	**1**	**0**	**0**

$$f = \sum(0,1,4,5) = \overline{B}$$

A \ BC	00	01	11	10
0	**1**	**1**	**0**	**0**
1	**1**	**1**	**0**	**0**

$$f = \sum(0,1,2,3) = \overline{A}$$

A \ BC	00	01	11	10
0	**1**	**1**	**1**	**1**
1	**0**	**0**	**0**	**0**

$$f = \sum(0,4) = \overline{A}\,C$$

A \ BC	00	01	11	10
0	**0**	**1**	**1**	**0**
1	**0**	**0**	**0**	**0**

$$f = \sum(4,6) = A\,\overline{C}$$

A \ BC	00	01	11	10
0	**0**	**0**	**0**	**0**
1	**1**	**0**	**0**	**1**

$$f = \sum(0,2) = \overline{A}\,\overline{C}$$

A \ BC	00	01	11	10
0	**1**	**0**	**0**	**1**
1	**0**	**0**	**0**	**0**

$$f = \sum(0,2,4,6) = \overline{C}$$

A \ BC	00	01	11	10
0	**1**	**0**	**0**	**1**
1	**1**	**0**	**0**	**1**

Three-Variable K-Map Examples

A \ BC	00	01	11	10
0		1		
1	1		1	1

A \ BC	00	01	11	10
0	1		1	1
1	1			1

A \ BC	00	01	11	10
0			1	1
1	1	1		

A \ BC	00	01	11	10
0			1	
1	1		1	1

A \ BC	00	01	11	10
0		1	1	1
1		1	1	

A \ BC	00	01	11	10
0				
1				

Design of combinational digital circuits

- Steps to design a combinational digital circuit:
  - From the problem statement derive the truth table
  - From the truth table derive the unsimplified

    logic expression
  - Simplify the logic expression
  - From the simplified expression draw the logic circuit

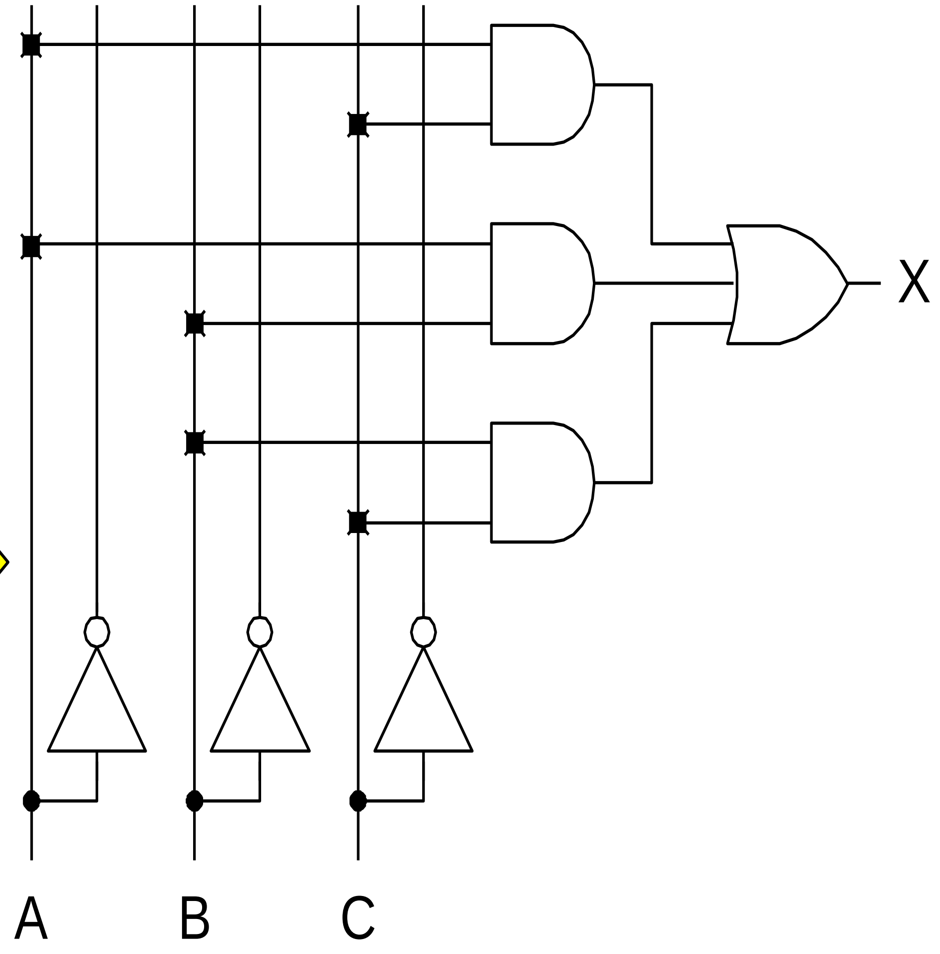
Example: Design a 3-input (A,B,C) digital circuit that will give at its output (X) a logic 1 only if the binary number formed at the input has more ones than zeros.

	Inputs			Output
	A	B	C	X
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

$$X = \sum(3, 5, 6, 7)$$

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$X = AC + AB + BC$$



Minimum SOP

- It has a minimum no. of terms.
  - That is, it has a minimum number of gates.
- It has a minimum no. of gate inputs.
  - That is, minimum no. of literals.
  - Each term in the minimum SOP is a prime implicant, i.e., it cannot be combined with others.
- It may not be unique.
  - Depend on the order in which terms are combined or eliminated.

Minimum POS

- It has a minimum no. factors.
- It has a minimum no. of literals.
- It may not be unique.
  - Use $(X+Y)(X+Y') = X$
  - Use $(X+C)(X'+D)(C+D) = (X+C)(X'+D)$ to eliminate term.

Minimum POS

- Example: Vertical input scheme

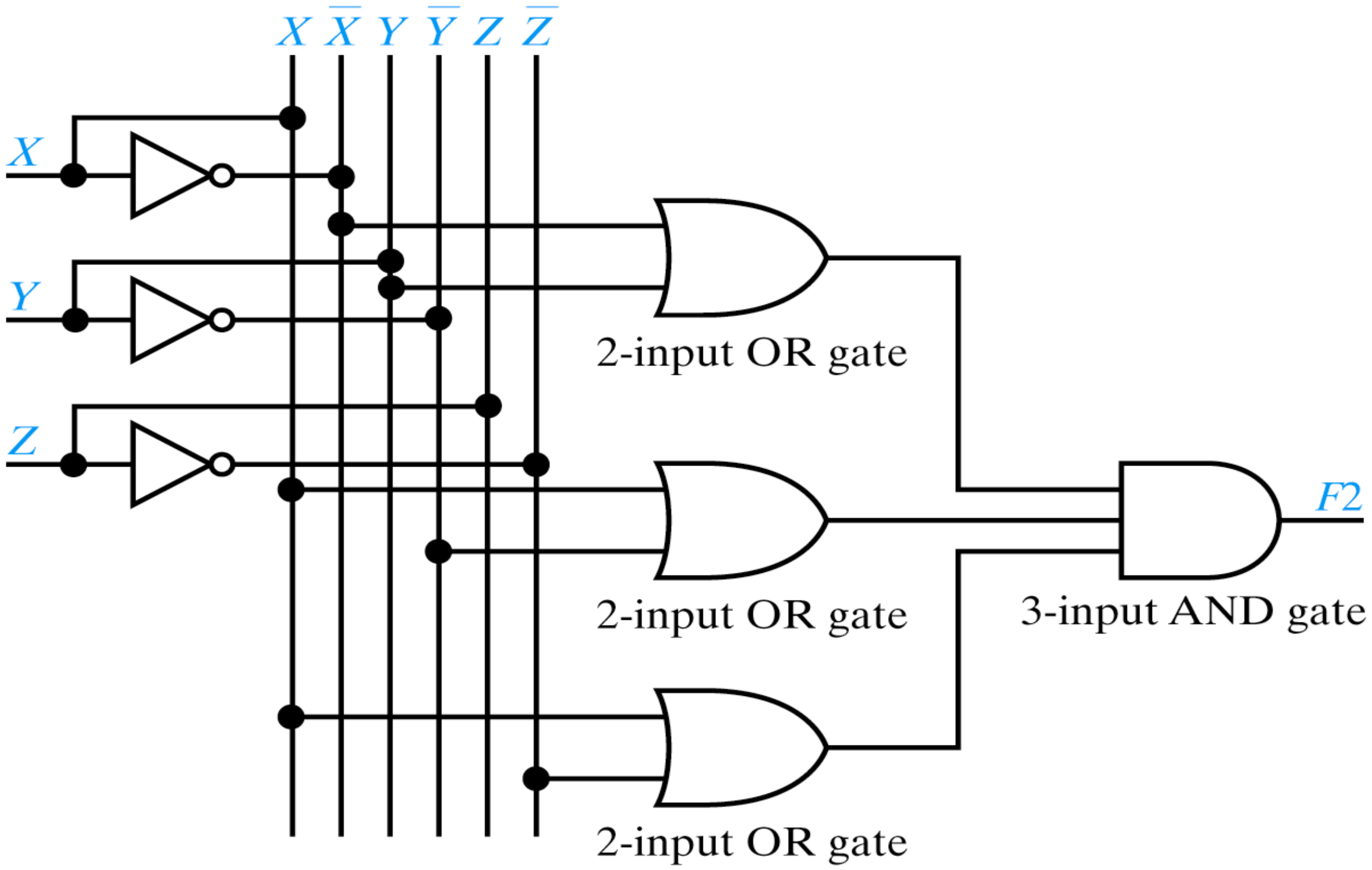


Figure 3.2.7 IC logic circuit design for a minimum POS form of a function using a vertical input scheme.

K Map Example

- K-map of $F(a,b,c) = \sum m(1,3,5) = \prod M(0,2,4,6,7)$

Place Product Terms on K Map

- Example
  - Place b, bc’ and ac’ in the 3-variable K map.

More Example

- Exercise. Plot $f(a, b, c) = abc' + b'c + a'$ into the K-map.

Simplication Example

- Exercise. Simplify: $F(a,b,c) = \sum m(1,3,5)$
  - Procedure: place minterms into map.
  - Select adjacent 1’s in group of two 1’s or four 1’s etc.
  - Kick off x and x’.

More Than Two Minimum Solutions

- $F = \sum m(0,1,2,5,6,7)$

Kmap Simplification for Three Variables

- This grouping tells us that changes in the variables *x* and *y* have no influence upon the value of the function: They are irrelevant.
- This means that the function,

$$F(X,Y) = \overline{X}\,\overline{Y}Z + \overline{X}YZ + X\overline{Y}Z + XYZ$$

reduces to $F(x) = z$.

You could verify this reduction with identities or a truth table.

X \ YZ	00	01	11	10
0	0	1	1	0
1	0	1	1	0

Kmap Simplification for Three Variables

- Now for a more complicated Kmap. Consider the function:

$$F(X,Y,Z) = \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}YZ + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XY\bar{Z}$$

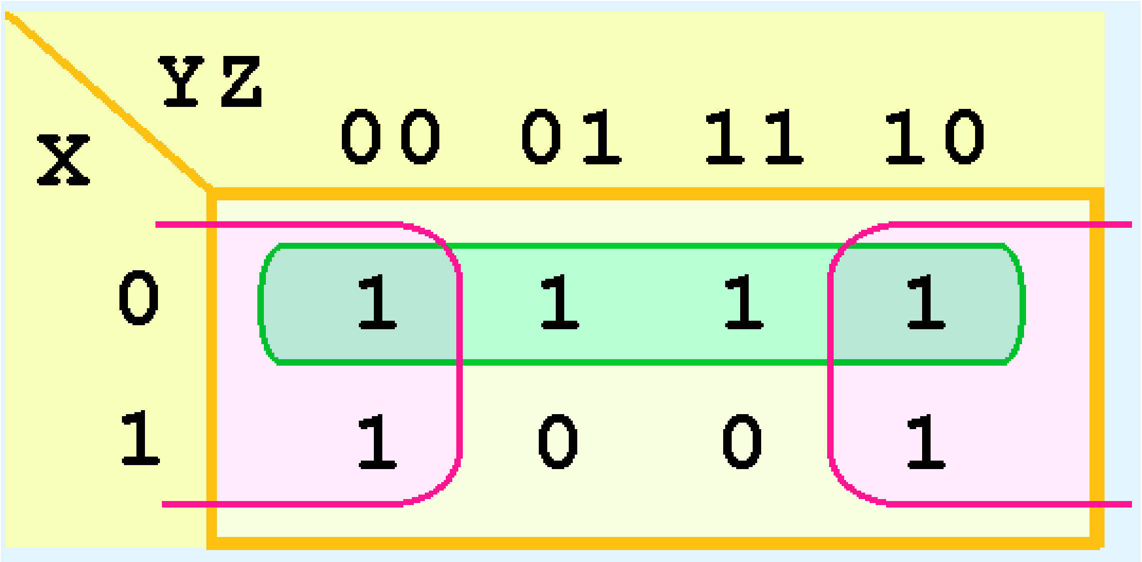
- Its Kmap is shown below. There are (only) two groupings of 1s.
  - Can you find them?

X \ YZ	00	01	11	10
0	1	1	1	1
1	1	0	0	1

Kmap Simplification for Three Variables

- In this Kmap, we see an example of a group that wraps around the sides of a Kmap.
- This group tells us that the values of *x* and *y* are not relevant to the term of the function that is encompassed by the group.
  - What does this tell us about this term of the function?

What about the green group in the top row?



K-map Simplification for Three Variables

- The green group in the top row tells us that only the value of *x* is significant in that group.
- We see that it is complemented in that row, so the other term of the reduced function is $\overline{X}$
- Our reduced function is: $F(X,Y,Z) = \overline{X} + \overline{Z}$

Recall that we had six minterms in our original function!

X \ YZ	00	01	11	10
0	1	1	1	1
1	1	0	0	1

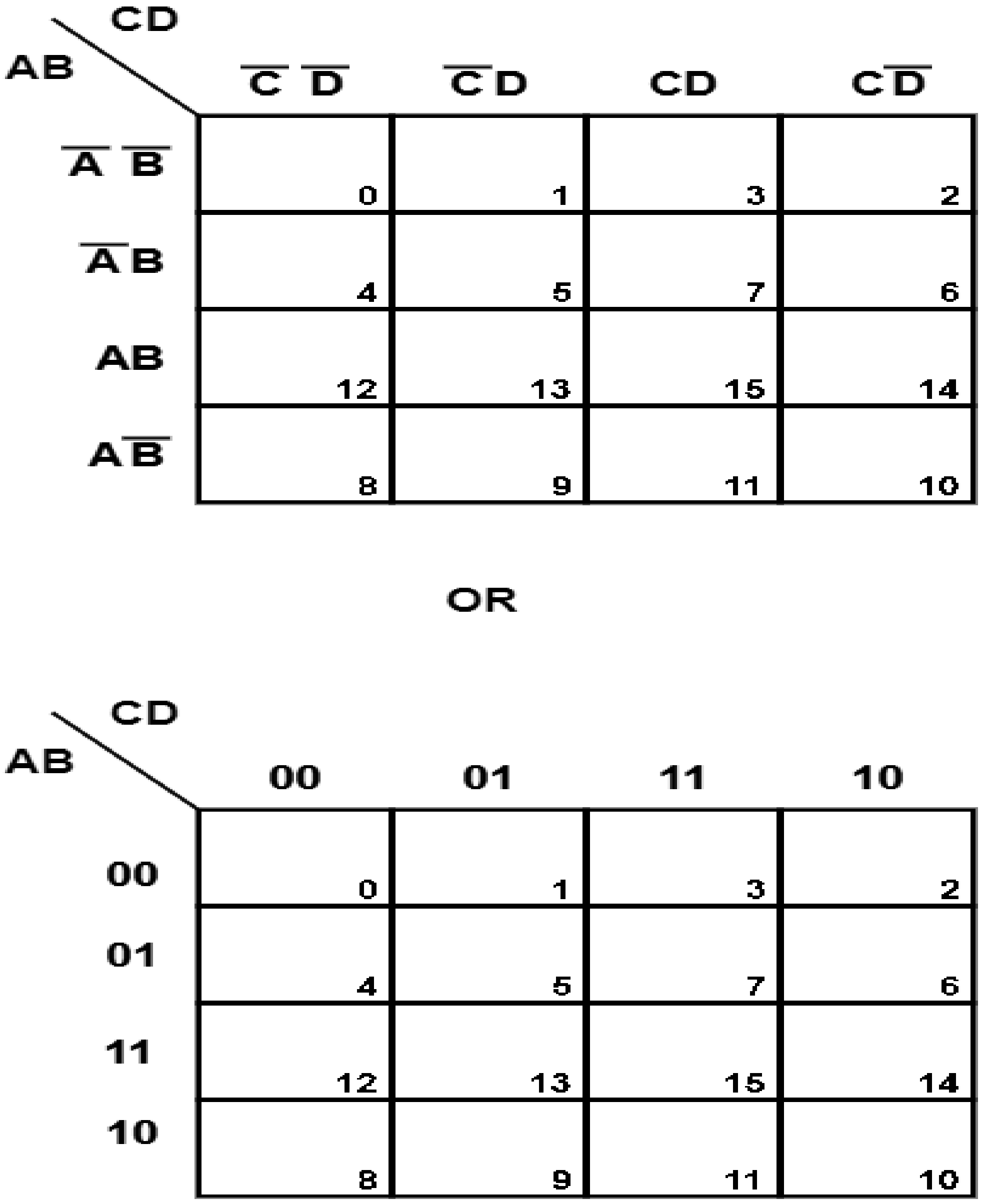
Four Variable K- Map

- Four variable K Map is drawn for a boolean expression consisting of four variables.
- The number of cells present in four variable K Map = 2^4 = 16 cells.

So, for a boolean function consisting of four variables,

we draw a 4 x 4 K Map.

- Four variable K Map may be represented as-



Four Variable K Map

4-Variable K Map

- Each minterm is adjacent to 4 terms with which it can combine.
  - 0, 8 are adjacent squares
  - 0, 2 are adjacent squares, etc.
  - 1, 4, 13, 7 are adjacent to 5.

Plot a 4-variable Expression

- F(a,b,c,d) = acd + a’b + d’

  acd = 1 if a=1, c=1, d=1

Simplification Example

- Minterms are combined in groups of 2, 4, or 8 to eliminate 1, 2, 3 variables.
- Corner terms.

Get a Minimum POS Using K Map

- Cover 0’s to get simplified POS.
  - We want 0 in each term.

$$f = x'z' + wyz + w'y'z' + x'y$$

First, the 1’s of f are plotted in Fig. 6–14. Then, from the 0’s,

$$f' = y'z + wxz' + w'xy$$

and the minimum product of sums for f is

$$f = (y + z')(w' + x' + z)(w + x' + y')$$

K-map Simplification for Four Variables

- Our model can be extended to accommodate the 16 minterms that are produced by a four-input function.
- This is the format for a 16-minterm Kmap.

WX \ YZ	00	01	11	10
00	$\overline{W}\overline{X}\overline{Y}\overline{Z}$	$\overline{W}\overline{X}\overline{Y}Z$	$\overline{W}\overline{X}YZ$	$\overline{W}\overline{X}Y\overline{Z}$
01	$\overline{W}X\overline{Y}\overline{Z}$	$\overline{W}X\overline{Y}Z$	$\overline{W}XYZ$	$\overline{W}XY\overline{Z}$
11	$WX\overline{Y}\overline{Z}$	$WX\overline{Y}Z$	$WXYZ$	$WXY\overline{Z}$
10	$W\overline{X}\overline{Y}\overline{Z}$	$W\overline{X}\overline{Y}Z$	$W\overline{X}YZ$	$W\overline{X}Y\overline{Z}$

Four-Variable K-Maps

AB \ CD	00	01	11	10
00	**1**	**0**	**0**	**0**
01	**0**	**0**	**0**	**0**
11	**0**	**0**	**0**	**0**
10	**1**	**0**	**0**	**0**

$$f = \sum(0,8) = \overline{B} \bullet \overline{C} \bullet \overline{D}$$

AB \ CD	00	01	11	10
00	**0**	**0**	**0**	**0**
01	**0**	**1**	**0**	**0**
11	**0**	**1**	**0**	**0**
10	**0**	**0**	**0**	**0**

$$f = \sum(5,13) = B \bullet \overline{C} \bullet D$$

AB \ CD	00	01	11	10
00	**0**	**0**	**0**	**0**
01	**0**	**0**	**0**	**0**
11	**0**	**1**	**1**	**0**
10	**0**	**0**	**0**	**0**

$$f = \sum(13,15) = A \bullet B \bullet D$$

AB \ CD	00	01	11	10
00	**0**	**0**	**0**	**0**
01	**1**	**0**	**0**	**1**
11	**0**	**0**	**0**	**0**
10	**0**	**0**	**0**	**0**

$$f = \sum(4,6) = \overline{A} \bullet B \bullet \overline{D}$$

AB \ CD	00	01	11	10
00	**0**	**0**	**1**	**1**
01	**0**	**0**	**1**	**1**
11	**0**	**0**	**0**	**0**
10	**0**	**0**	**0**	**0**

$$f = \sum(2,3,6,7) = \overline{A} \bullet C$$

AB \ CD	00	01	11	10
00	**0**	**0**	**0**	**0**
01	**1**	**0**	**0**	**1**
11	**1**	**0**	**0**	**1**
10	**0**	**0**	**0**	**0**

$$f = \sum(4,6,12,14) = B \bullet \overline{D}$$

AB \ CD	00	01	11	10
00	**0**	**0**	**1**	**1**
01	**0**	**0**	**0**	**0**
11	**0**	**0**	**0**	**0**
10	**0**	**0**	**1**	**1**

$$f = \sum(2,3,10,11) = \overline{B} \bullet C$$

AB \ CD	00	01	11	10
00	**1**	**0**	**0**	**1**
01	**0**	**0**	**0**	**0**
11	**0**	**0**	**0**	**0**
10	**1**	**0**	**0**	**1**

$$f = \sum(0,2,8,10) = \overline{B} \bullet \overline{D}$$

Four-Variable K-Maps

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

$f = \sum(4,5,6,7) = \overline{A} \bullet B$

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

$f = \sum(3,7,11,15) = C \bullet D$

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

$f = \sum(0,3,5,6,9,10,12,15)$

$f = A \otimes B \otimes C \otimes D$

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

$f = \sum(1,2,4,7,8,11,13,14)$

$f = A \oplus B \oplus C \oplus D$

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

$f = \sum(1,3,5,7,9,11,13,15)$

$f = D$

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

$f = \sum(0,2,4,6,8,10,12,14)$

$f = \overline{D}$

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

$f = \sum(4,5,6,7,12,13,14,15)$

$f = B$

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

$f = \sum(0,1,2,3,8,9,10,11)$

$f = \overline{B}$

Four-Variable K-Maps Examples

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

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

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

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

AB \ CD	00	01	11	10
00				
01				
11				
10				

AB \ CD	00	01	11	10
00				
01				
11				
10				

Design of combinational digital circuits (Cont.)

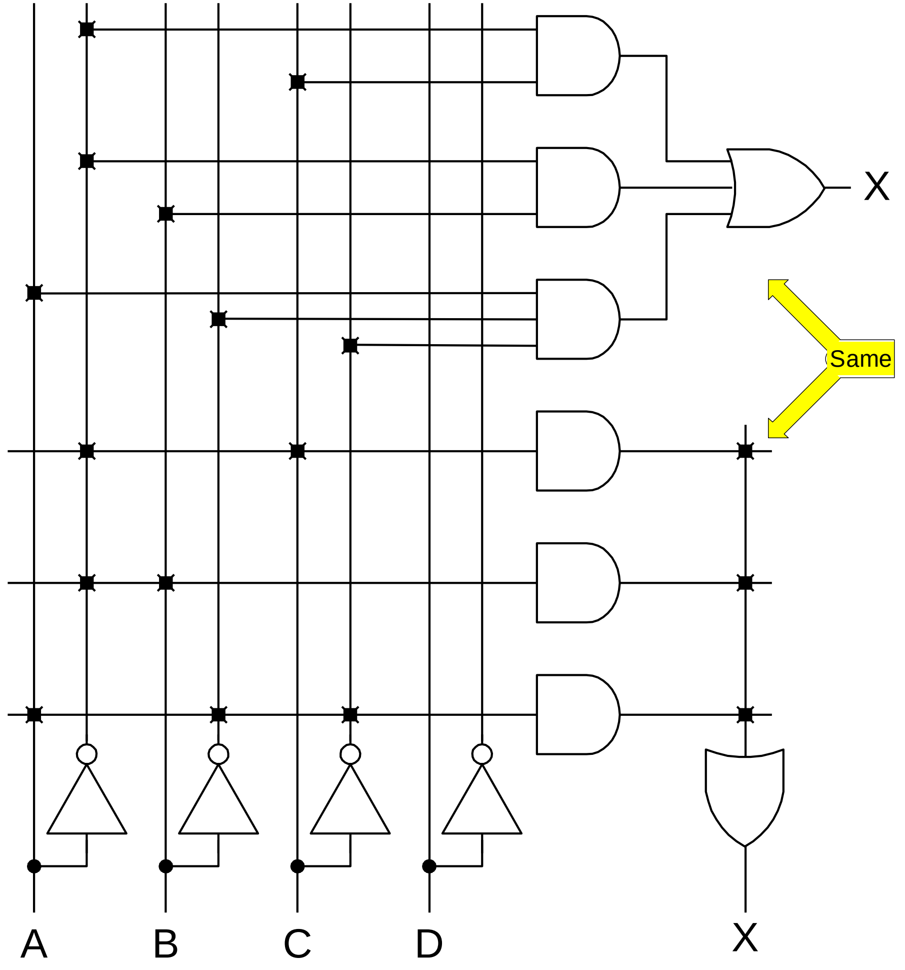
- Example: Design a 4-input (A,B,C,D) digital circuit that will give at its output (X) a logic 1 only if the binary number formed at the input is between 2 and 9 (including).

	Inputs				Output
	A	B	C	D	X
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

$$X = \sum(2,3,4,5,6,7,8,9)$$

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

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



Design of combinational digital circuits (Example)

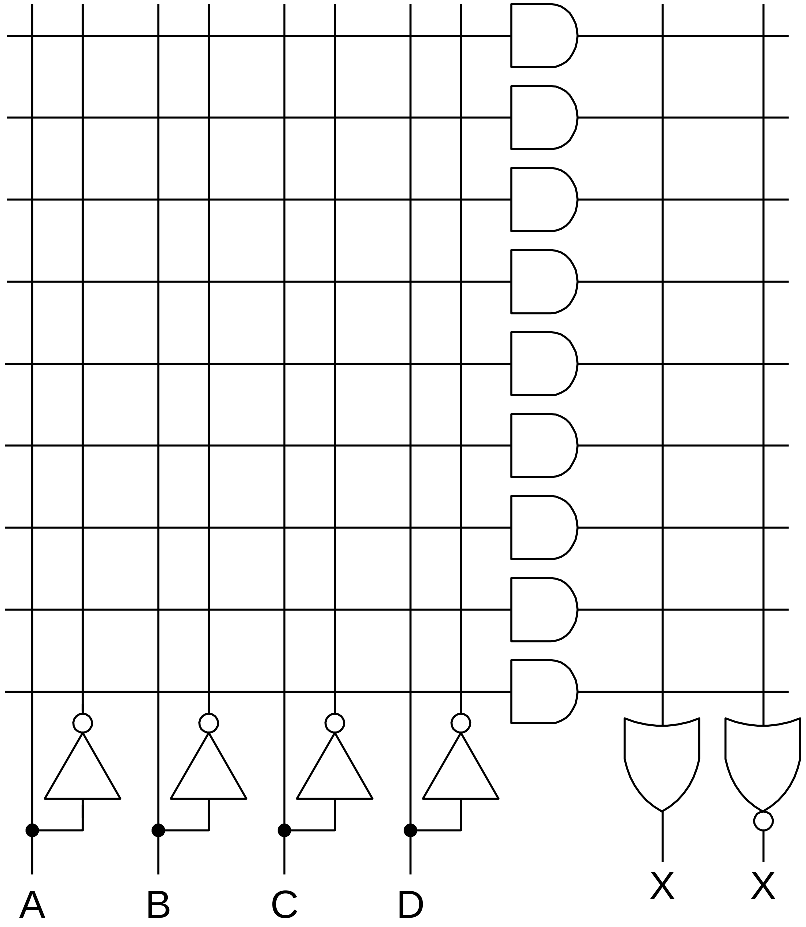
- Example: Design a 4-input (A,B,C,D) digital circuit that will give at its output (X) a logic 1 only if there more ones than zeros in the binary number formed at the input.

	Inputs				Output		
	A	B	C	D			
0	0	0	0	0			
1	0	0	0	1			
2	0	0	1	0			
3	0	0	1	1			
4	0	1	0	0			
5	0	1	0	1			
6	0	1	1	0			
7	0	1	1	1			
8	1	0	0	0			
9	1	0	0	1			
10	1	0	1	0			
11	1	0	1	1			
12	1	1	0	0			
13	1	1	0	1			
14	1	1	1	0			
15	1	1	1	1			

X =

AB \ CD	00	01	11	10
00				
01				
11				
10				

X =



Kmap Simplification for Four Variables

- We have populated the Kmap shown below with the nonzero minterms from the function:

$$
\begin{aligned}
F(W,X,Y,Z) = {} & \overline{W}\,\overline{X}\,\overline{Y}\,\overline{Z} + \overline{W}\,\overline{X}\,\overline{Y}Z + \overline{W}\,\overline{X}Y\overline{Z} \\
& + \overline{W}XY\overline{Z} + W\overline{X}\,\overline{Y}\,\overline{Z} + W\overline{X}\,\overline{Y}Z + W\overline{X}Y\overline{Z}
\end{aligned}
$$

- Can you identify (only) three groups in this Kmap?

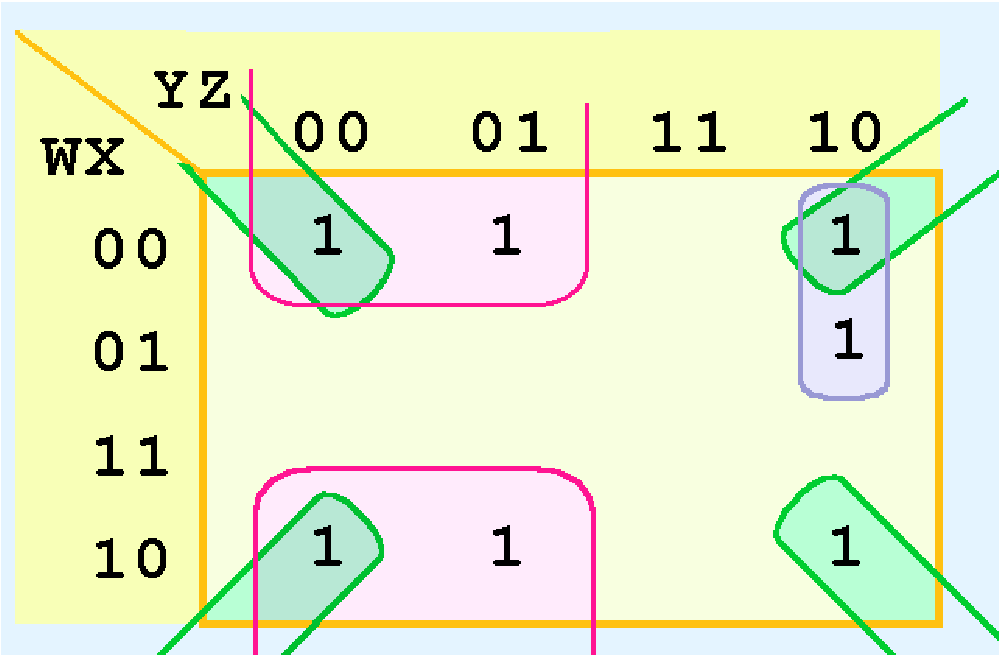
Recall that groups can overlap.

WX \ YZ	00	01	11	10
00	1	1		1
01				1
11				
10	1	1		1

Kmap Simplification for Four Variables

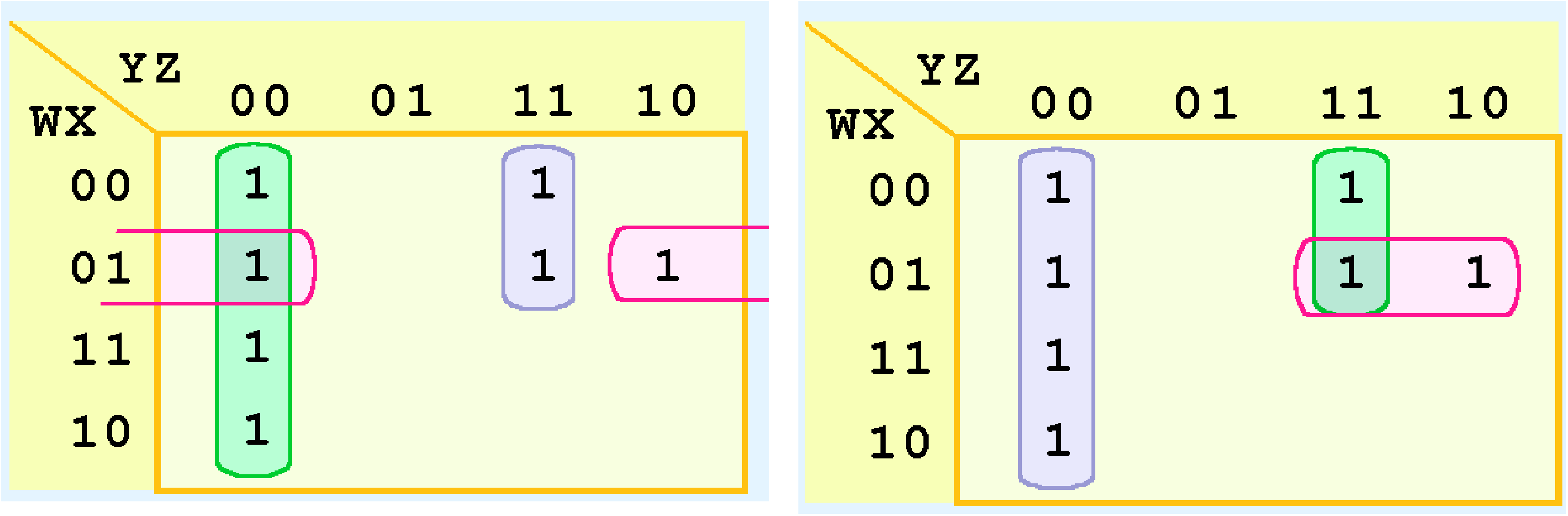
- Our three groups consist of:
  - A purple group entirely within the Kmap at the right.
  - A pink group that wraps the top and bottom.
  - A green group that spans the corners.
- Thus we have three terms in our final function:

$$F(W,X,Y,Z) = \overline{W}\,\overline{Y} + \overline{X}\,\overline{Z} + \overline{W}Y\overline{Z}$$



Kmap Simplification for Four Variables

- It is possible to have a choice as to how to pick groups within a Kmap, while keeping the groups as large as possible.
- The (different) functions that result from the groupings below are logically equivalent.



Don’t Care Conditions

- Real circuits don’t always need to have an output defined for every possible input.
    - For example, some calculator displays consist of 7-segment LEDs. These LEDs can display $2^7 - 1$ patterns, but only ten of them are useful.
- If a circuit is designed so that a particular set of inputs can never happen, we call this set of inputs a *don’t care* condition.
- They are very helpful to us in Kmap circuit simplification.

Don’t Care Conditions

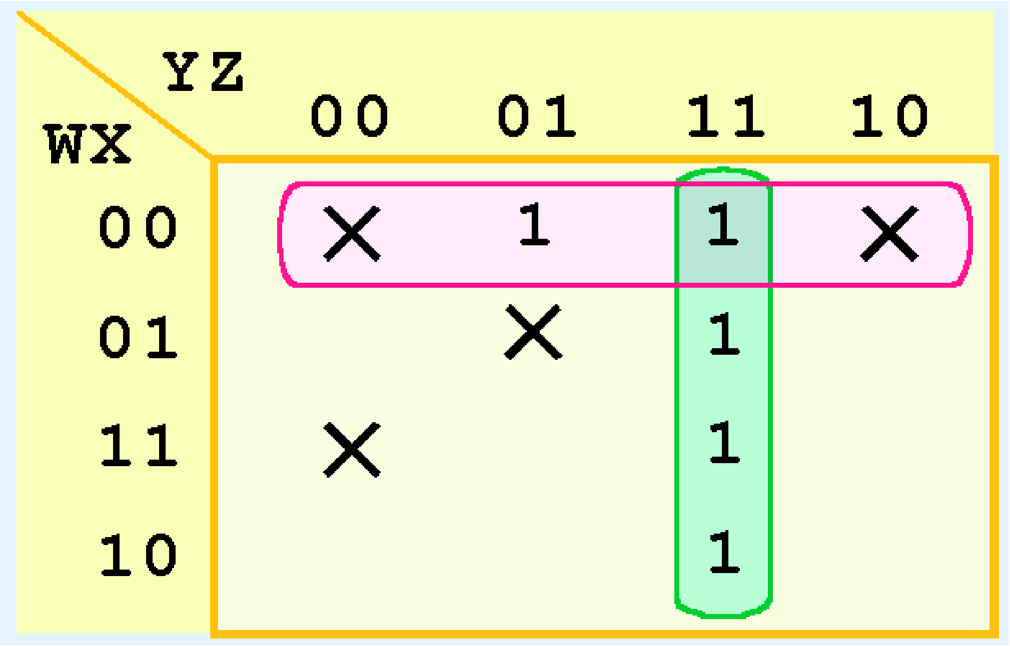
- In a Kmap, a don’t care condition is identified by an *X* in the cell of the minterm(s) for the don’t care inputs, as shown below.
- In performing the simplification, we are free to include or ignore the *X*’s when creating our groups.

WX \ YZ	00	01	11	10
00	X	1	1	X
01		X	1	
11	X		1	
10			1	

Don’t Care Conditions

- In one grouping in the Kmap below, we have the function:

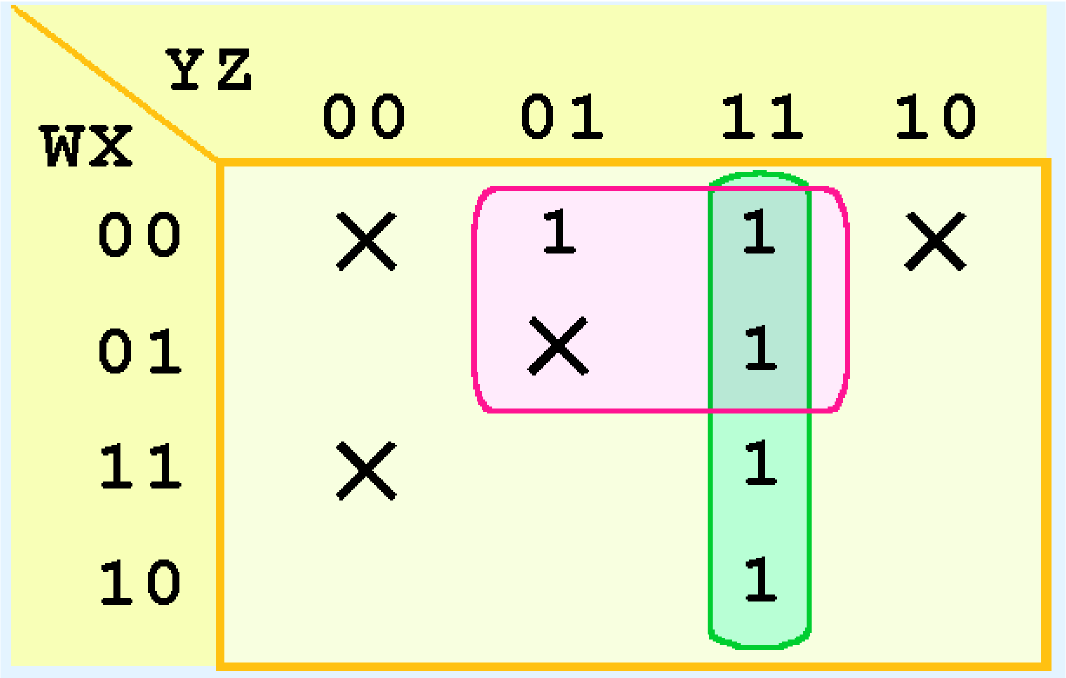
$$F(W,X,Y,Z) = \overline{W}\,\overline{Y} + YZ$$



Don’t Care Conditions

- A different grouping gives us the function:

$$F(W,X,Y,Z) = \bar{W}Z + YZ$$



Don’t Care Conditions

- The truth table of:

$$F(W,X,Y,Z) = \overline{W}\,\overline{Y} + YZ$$

  differs from the truth table of:

$$F(W,X,Y,Z) = \overline{W}Z + YZ$$

- However, the values for which they differ, are the inputs for which we have don’t care conditions.

WX \ YZ	00	01	11	10
00	X	1	1	X
01		X	1	
11	X		1	
10			1	

WX \ YZ	00	01	11	10
00	X	1	1	X
01		X	1	
11	X		1	
10			1	

Conclusion

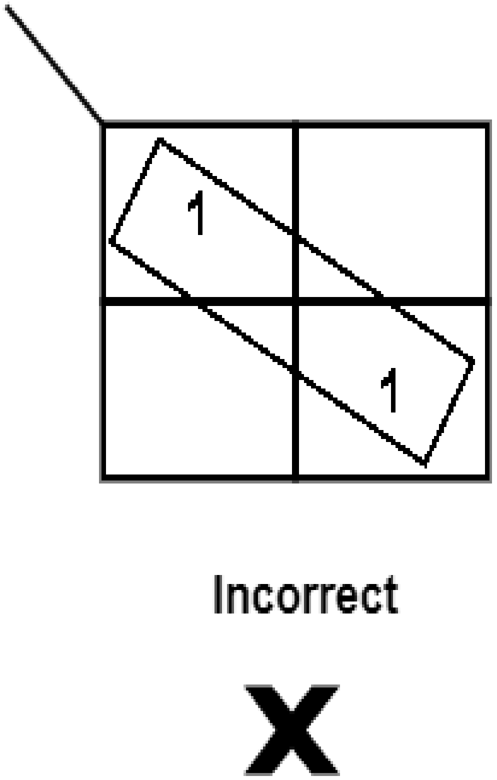
- Kmaps provide an easy graphical method of simplifying Boolean expressions.
- A Kmap is a matrix consisting of the outputs of the minterms of a Boolean function.
- In this section, we have discussed 2- 3- and 4-input Kmaps. This method can be extended to any number of inputs through the use of multiple tables.

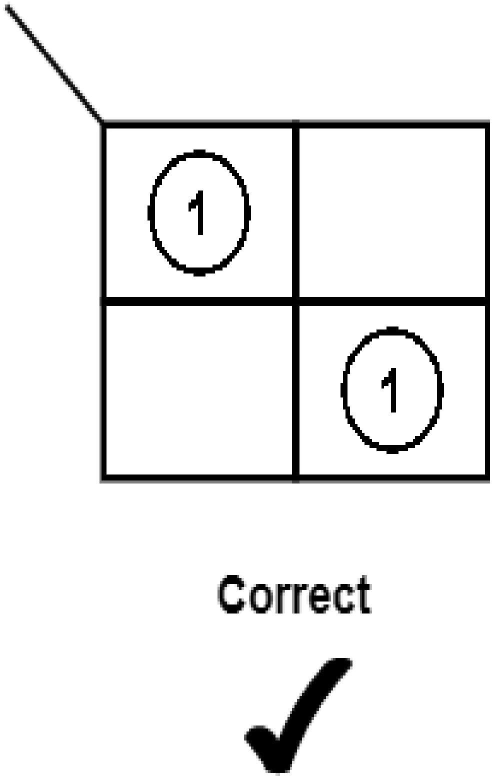
Conclusion

Recapping the rules of Kmap simplification:

- Groupings can contain only 1s; no 0s.
- Groups can be formed only at right angles; diagonal groups are not allowed.
- The number of 1s in a group must be a power of 2 – even if it contains a single 1.
- The groups must be made as large as possible.
- Groups can overlap and wrap around the sides of the Kmap.
- Use don’t care conditions when you can.

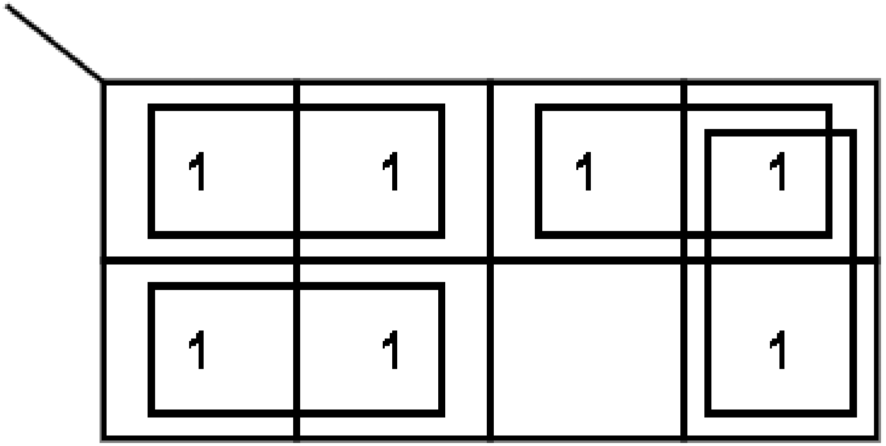
- Groups can be only either horizontal or vertical.
- We can not create groups of diagonal or any other shape.



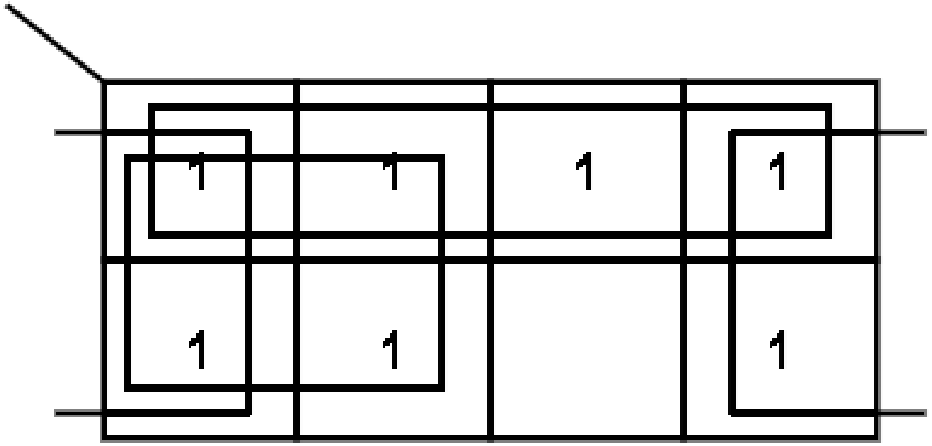


•Each group should be as large as possible.

Example-



Incorrect



Correct