

6.0 DESCRIPTION OF KMAPS AND TERMINOLOGY

NB: Karnaugh maps can be drawn for both Sum of product Boolean expressions or Product of Sums Boolean expressions. We shall start base our discussion mainly on Sum of Product expressions which uses minterms. But a few examples of Product of Sums Boolean expressions which uses maxterms are also given. Karnaugh map can be drawn directly from either minterm (sum-of-products) or maxterm (product-of-sums) Boolean expressions. Drawing a Karnaugh map from the truth table involves an additional step of writing the minterm or maxterm expression depending upon whether it is desired to have a minimized sum-of-products or a minimized product-of- sums expression. An n-variable Karnaugh map has 2^n squares, and each possible input is allotted a square.

Karnaugh maps, or Kmaps, are a graphical way to represent Boolean functions. A map is simply a table used to enumerate the values of a given Boolean expression for different input values. The rows and columns correspond to the possible values of the function's inputs. Each cell represents the outputs of the function for those possible inputs.

If a product term includes all of the variables exactly once, either complemented or not complemented, this product term is called a *minterm*. For example, if there are two input values, x and y , there are four minterms, $\bar{x}\bar{y}$, $\bar{x}y$, $x\bar{y}$, and xy , which represent all of the possible input combinations for the function. If the input variables are x , y , and z , then there are eight minterms: $\bar{x}\bar{y}\bar{z}$, $\bar{x}\bar{y}z$, $\bar{x}y\bar{z}$, $\bar{x}yz$, $x\bar{y}\bar{z}$, $x\bar{y}z$, $xy\bar{z}$, and xyz .

As an example, consider the Boolean function $F(x,y) = xy + \bar{x}y$. Possible inputs for x and y are shown in Figure 3A.1.

The minterm $\bar{x}\bar{y}$ represents the input pair (0,0). Similarly, the minterm $\bar{x}y$ represents (0,1), the minterm $x\bar{y}$ represents (1,0), and xy represents (1,1).

The minterms for three variables, along with the input values they represent, are shown in Figure 3A.2.

Minterm	x	y
$\bar{X}\bar{Y}$	0	0
$\bar{X}Y$	0	1
$X\bar{Y}$	1	0
XY	1	1

FIGURE 3A.1 Minterms for Two Variables

Minterm	x	y	z
$\bar{X}\bar{Y}\bar{Z}$	0	0	0
$\bar{X}\bar{Y}Z$	0	0	1
$\bar{X}Y\bar{Z}$	0	1	0
$\bar{X}YZ$	0	1	1
$X\bar{Y}\bar{Z}$	1	0	0
$X\bar{Y}Z$	1	0	1
$XY\bar{Z}$	1	1	0
XYZ	1	1	1

FIGURE 3A.2 Minterms for Three Variables

A similar discussion applies for maxterm Karnaugh maps.

Figure 6.10 shows the truth table, minterm Karnaugh map and maxterm Karnaugh map of the Boolean function of a two-input OR gate. The minterm and maxterm Boolean expressions for the two-input OR gate are as follows:

$$Y = A + B \text{ (maxterm or product-of-sums)} \quad (6.44)$$

$$Y = \bar{A}.B + A.\bar{B} + A.B \text{ (minterm or sum-of-products)} \quad (6.45)$$

Figure 6.11 shows the truth table, minterm Karnaugh map and maxterm Karnaugh map of the three-variable Boolean function

$$Y = \bar{A}.\bar{B}.\bar{C} + \bar{A}.B.\bar{C} + A.\bar{B}.\bar{C} + A.B.\bar{C} \quad (6.46)$$

$$Y = (\bar{A} + \bar{B} + \bar{C}).(\bar{A} + B + \bar{C}).(A + \bar{B} + \bar{C}).(A + B + \bar{C}) \quad (6.47)$$

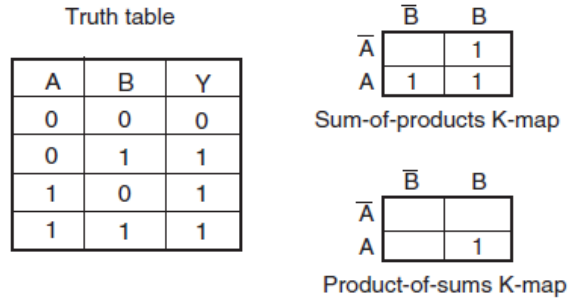


Figure 6.10 Two-variable Karnaugh maps.

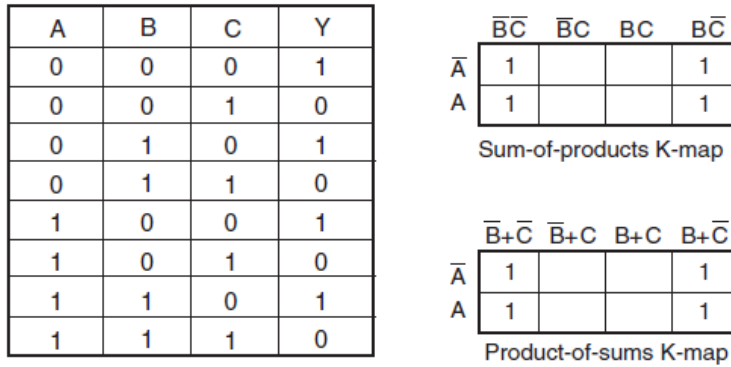


Figure 6.11 Three-variable Karnaugh maps.

Figure 6.12 shows the truth table, minterm Karnaugh map and maxterm Karnaugh map of the four-variable Boolean function

$$Y = \bar{A}.\bar{B}.\bar{C}.\bar{D} + \bar{A}.\bar{B}.\bar{C}.D + \bar{A}.B.\bar{C}.\bar{D} + \bar{A}.B.\bar{C}.D + A.\bar{B}.\bar{C}.\bar{D} + A.\bar{B}.\bar{C}.D + A.B.\bar{C}.\bar{D} + A.B.\bar{C}.D \quad (6.48)$$

$$Y = (A+B+\bar{C}+D).(A+B+\bar{C}+\bar{D}).(A+\bar{B}+\bar{C}+D).(A+\bar{B}+\bar{C}+\bar{D}).(\bar{A}+B+\bar{C}+D).(\bar{A}+B+\bar{C}+\bar{D}).(\bar{A}+\bar{B}+\bar{C}+D).(\bar{A}+\bar{B}+\bar{C}+\bar{D}) \quad (6.49)$$

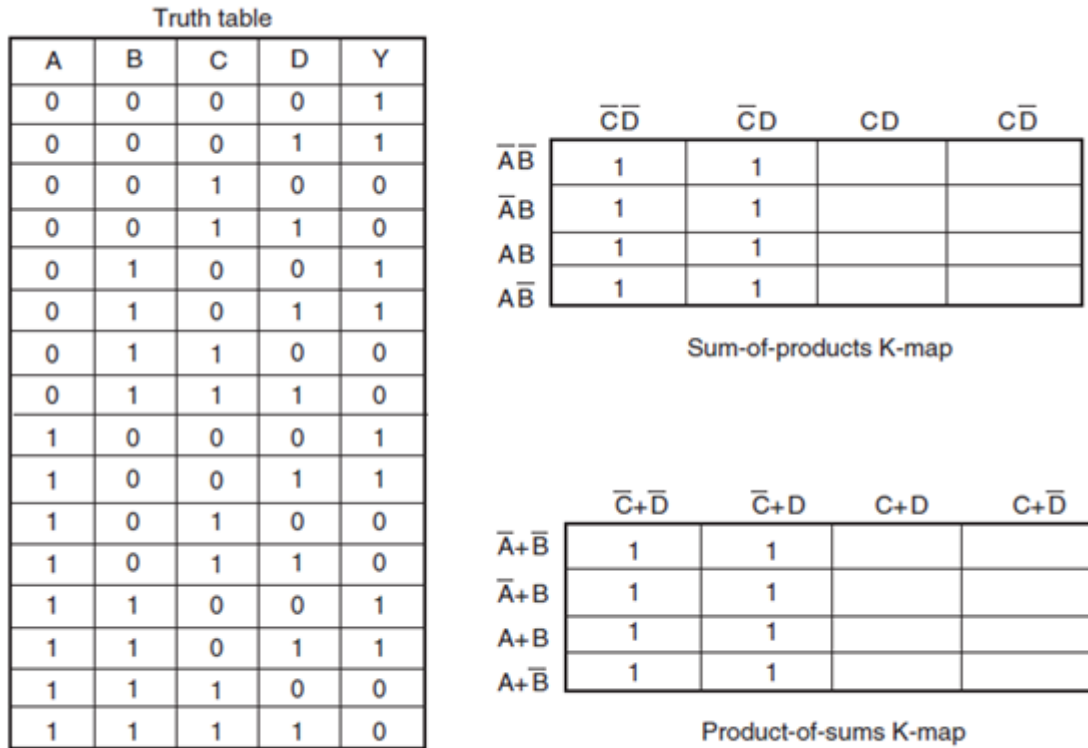


Figure 6.12 Four-variable Karnaugh maps.

How the simplification of Boolean Expressions is done using K-maps

A Kmap is a table with a cell for each minterm, which means it has a cell for each line of the truth table for the function. Consider the function $F(x,y) = xy$ and its truth table, as seen in Example 3A.1.

≡ EXAMPLE 3A.1 $F(x,y) = xy$

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

The corresponding Kmap is:

$x \backslash y$	0	1
0	0	0
1	0	1

Notice that the only cell in the map with a value of one occurs when $x = 1$ and $y = 1$, the same values for which $xy = 1$. Let's look at another example, $F(x,y) = x + y$.

≡ EXAMPLE 3A.2 $F(x,y) = x + y$

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

$x \backslash y$	0	1
0	0	1
1	1	1

Three of the minterms in Example 3A.2 have a value of 1, exactly the minterms for which the input to the function gives us a 1 for the output. To assign 1s in the Kmap, we simply place 1s where we find corresponding 1s in the truth table. We can express the function $F(x,y) = x + y$ as the logical OR of all minterms for which the minterm has a value of 1. Then $F(x,y)$ can be represented by the expression $\bar{x}y + x\bar{y} + xy$. Obviously, this expression is not minimized (we already know this function is simply $x + y$). We can minimize using Boolean identities:

$$\begin{aligned}
 F(x,y) &= \bar{x}y + x\bar{y} + xy \\
 &= \bar{x}y + xy + x\bar{y} + xy \quad (\text{remember, } xy + xy = xy) \\
 &= y(\bar{x} + x) + x(\bar{y} + y) \\
 &= y + x \\
 &= x + y
 \end{aligned}$$

How did we know to add in an extra xy term? Algebraic simplification using Boolean identities can be very tricky. This is where Kmaps can help.

3A.3 KMAP SIMPLIFICATION FOR TWO VARIABLES

In the previous reduction for the function $F(x,y)$, the goal was to group terms so we could factor out variables. We added the xy to give us a term to combine with the $\bar{x}y$. This allowed us to factor out the y , leaving $\bar{x} + x$, which reduces to $x + y$.

1. However, if we use Kmap simplification, we won't have to worry about which terms to add or which Boolean identity to use. The maps take care of that for us.

Let's look at the Kmap for $F(x,y) = x + y$ again in Figure 3A.3.

To use this map to reduce a Boolean function, we simply need to group ones. This grouping is very similar to how we grouped terms when we reduced using Boolean identities, except we must follow specific rules. First, we group only ones. Second, we can group ones in the Kmap if the ones are in the same row or in the same column, but they cannot be on the diagonal (i.e., they must be adjacent cells). Third, we can group ones if the total number in the group is a power of 2. The fourth rule specifies we must make the groups as large as possible. As a fifth and final rule, all ones must be in a group (even if some are in a group of one). Let's examine some correct and incorrect groupings, as shown in Figures 3A.4 through 3A.7.

Notice in Figure 3A.6(b) and 3A.7(b) that one 1 belongs to two groups. This is the map equivalent of adding the term xy to the Boolean function, as we did when we were performing simplification using identities. The xy term in the map will be used twice in the simplification procedure.

To simplify using Kmaps, first create the groups as specified by the rules above. After you have found all groups, examine each group and discard the variable that differs within each group. For example, Figure 3A.7(b) shows the correct grouping for $F(x,y) = x + y$. Let's begin with the group represented by the second row (where $x = 1$). The two minterms are $x\bar{y}$ and xy . This group represents the logical OR of these two terms, $x\bar{y}$ and xy . These terms differ in y , so y is discarded, leaving only x . (We can see that if we use Boolean identities, this would reduce to the same value. The Kmap allows us to take a shortcut, helping us to automatically discard the correct variable.) The second group represents $\bar{x}y + xy$. These differ in x , so x is discarded, leaving y . If we OR the results of the first group and the second group, we have $x + y$, which is the correct reduction of the original function, F .

		y	
x		0	1
	0	0	1
	1	1	1

FIGURE 3A.3 Kmap for $F(x,y) = x + y$

x \ y	0	1
0	0	1
1	1	1

a) Incorrect

FIGURE 3A.4 Groups Contain Only 1s

x \ y	0	1
0	0	1
1	1	1

b) Correct

x \ y	0	1
0	0	1
1	1	1

a) Incorrect

FIGURE 3A.5 Groups Cannot Be Diagonal

x \ y	0	1
0	0	1
1	1	1

b) Correct

x \ y	0	1
0	0	1
1	1	1

a) Incorrect

FIGURE 3A.6 Groups Must Be Powers of 2

x \ y	0	1
0	0	1
1	1	1

b) Correct

x \ y	0	1
0	0	1
1	1	1

a) Incorrect

FIGURE 3A.7 Groups Must Be as Large as Possible

x \ y	0	1
0	0	1
1	1	1

b) Correct

3A.4 KMAP SIMPLIFICATION FOR THREE VARIABLES

Kmaps can be applied to expressions of more than two variables. In this focus section, we show three variable and four-variable Kmaps. These can be extended for situations have five or more variables. You already know how to set up Kmaps for expressions involving two variables. We simply extend this idea to three variables, as indicated by Figure 3A.8. The first difference you should notice is that two variables, y and z, are grouped together in the table. The second difference is that the numbering for the columns is not sequential. Instead of labeling the columns as 00, 01, 10, 11 (a normal binary progression), we have labeled them 00, 01, 11, 10. The input values for the Kmap must be ordered so that each minterm differs in only one variable from each neighbor. By using this order (for example 01 followed by 11), the

x \ yz	00	01	11	10
0	$\overline{X}Y\overline{Z}$	$\overline{X}Y\overline{Z}$	$\overline{X}YZ$	$\overline{X}Y\overline{Z}$
1	$X\overline{Y}\overline{Z}$	$X\overline{Y}\overline{Z}$	XYZ	$XY\overline{Z}$

FIGURE 3A.8 Minterms and Kmap Format for Three Variables

corresponding minterms, $\bar{x}\bar{y}z$ and $\bar{x}yz$, differ only in the y variable. Remember, to reduce, we need to discard the variable that is different. Therefore, we must ensure that each group of two minterms differs in only one variable.

The largest groups we found in our two-variable examples were composed of two 1s. It is possible to have groups of four or even eight 1s, depending on the function. Let's look at a couple of examples of map simplification for expressions of three variables.

≡ **EXAMPLE 3A.3** $F(x, y, z) = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}z + xyz$

$x \backslash yz$	00	01	11	10
0	0	1	1	0
1	0	1	1	0

We again follow the rules for making groups. You should see that you can make groups of two in several ways. However, the rules stipulate we must create the largest groups whose sizes are powers of two. There is one group of four, so we group these as follows:

$x \backslash yz$	00	01	11	10
0	0	1	1	0
1	0	1	1	0

It is not necessary to create additional groups of two. The fewer groups you have, the fewer terms there will be. Remember, we want to simplify the expression, and all we have to do is guarantee that every 1 is in some group.

How, exactly, do we simplify when we have a group of four 1s? Two 1s in a group allowed us to discard one variable. Four 1s in a group allows us to discard two variables: The two variables in which all four terms differ. In the group of four from the preceding example, we have the following minterms: $\bar{x}\bar{y}z$, $\bar{x}yz$, $x\bar{y}z$ and xyz . These all have z in common, but the x and y variables differ. So we discard x and y , leaving us with $F(x,y,z) = z$ as the final reduction. To see how this parallels simplification using Boolean identities, consider the same reduction using identities. Note that the function is represented originally as the logical OR of the minterms with a value of 1.

$$\begin{aligned}
 F(x, y, z) &= \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}z + xyz \\
 &= \bar{x}(\bar{y}z + yz) + x(\bar{y}z + yz) \\
 &= (\bar{x} + x)(\bar{y}z + yz) \\
 &= \bar{y}z + yz \\
 &= (\bar{y} + y)z \\
 &= z
 \end{aligned}$$

The end result using Boolean identities is exactly the same as the result using map simplification. From time to time, the grouping process can be a little tricky. Let's look at an example that requires more scrutiny.

EXAMPLE 3A.4 $F(x, y, z) = \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + \bar{x}yz + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xy\bar{z}$

$x \backslash yz$	00	01	11	10
0	1	1	1	1
1	1	0	0	1

This is a tricky problem for two reasons: We have overlapping groups, and we have a group that “wraps around.” The leftmost 1s in the first column can be grouped with the rightmost 1s in the last column, because the first and last columns are logically adjacent (envision the map as being drawn on a cylinder). The first and last rows of a Kmap are also logically adjacent, which becomes apparent when we look at four-variable maps in the next section.

The correct groupings are as follows:

$x \backslash yz$	00	01	11	10
0	1	1	1	1
1	1	0	0	1

The first group reduces to \bar{x} (this is the only term the four have in common), and the second group reduces to \bar{z} , so the final minimized function is $F(x,y,z) = \bar{x} + \bar{z}$.

EXAMPLE 3A.5 A Kmap with all 1s

Suppose we have the following Kmap:

$x \backslash yz$	00	01	11	10
0	1	1	1	1
1	1	1	1	1

The largest group of 1s we can find is a group of eight, which puts all of the 1s in the same group. How do we simplify this? We follow the same rules we have been following. Remember, groups of two allowed us to discard one variable, and groups of four allowed us to discard two variables; therefore, groups of eight should allow us to discard three variables. But that's all we have! If we discard all the variables, we are left with $F(x,y,z) = 1$. If you examine the truth table for this function, you will see that we do indeed have a correct simplification.

yz \ wx	00	01	11	10
00	$\overline{w}x\overline{y}\overline{z}$	$\overline{w}x\overline{y}z$	$\overline{w}\overline{x}\overline{y}\overline{z}$	$\overline{w}\overline{x}\overline{y}z$
01	$\overline{w}x\overline{y}\overline{z}$	$\overline{w}x\overline{y}z$	$\overline{w}x\overline{y}\overline{z}$	$\overline{w}x\overline{y}z$
11	$w\overline{x}\overline{y}\overline{z}$	$w\overline{x}\overline{y}z$	$w\overline{x}\overline{y}\overline{z}$	$w\overline{x}\overline{y}z$
10	$w\overline{x}\overline{y}\overline{z}$	$w\overline{x}\overline{y}z$	$w\overline{x}\overline{y}\overline{z}$	$w\overline{x}\overline{y}z$

FIGURE 3A.8 Minterms and Kmap Format for Four Variables

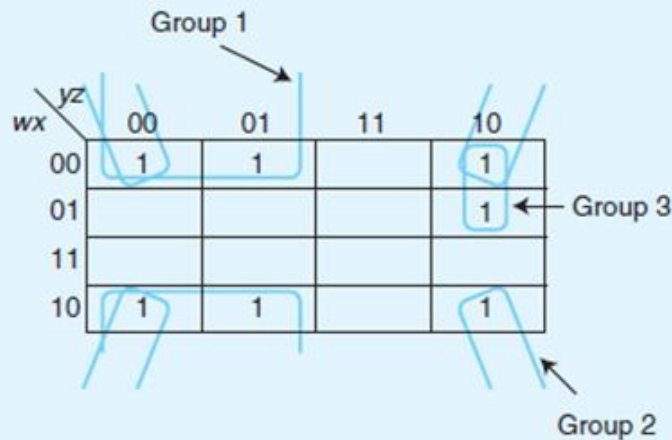
3A.5 KMAP SIMPLIFICATION FOR FOUR VARIABLES

We now extend the map simplification techniques to four variables. Four variables give us 16 minterms, as shown in Figure 3A.9. Notice the special order of 11 followed by 10 applies for the rows as well as the columns. Example 3A.6 illustrates the representation and simplification of a function with four variables. We are only concerned with the terms that are 1s, so we omit entering the 0s into the map.

$$F(w,x,y,z) = \overline{w}\overline{x}\overline{y}\overline{z} + \overline{w}\overline{x}\overline{y}z + \overline{w}x\overline{y}\overline{z} + \overline{w}x\overline{y}z + w\overline{x}\overline{y}\overline{z} + w\overline{x}\overline{y}z + w\overline{x}y\overline{z} + w\overline{x}yz$$

EXAMPLE 3A.6

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



Group 1 is a “wrap-around” group, as we saw previously. Group 3 is easy to find as well. Group 2 represents the ultimate wrap-around group: It consists of the 1s in the four corners. Remember, these corners are logically adjacent. The final result is that F reduces to three terms, one from each group: $\overline{x}\overline{y}$ (from Group 1), $\overline{x}\overline{z}$ (from Group 2), and $\overline{w}y\overline{z}$ (from Group 3). The final reduction for F is then $F(w, x, y, z) = \overline{x}\overline{y} + \overline{x}\overline{z} + \overline{w}y\overline{z}$.

Occasionally, there are choices to make when performing map simplification. Consider Example 3A.7.

EXAMPLE 3A.7 A Choice of Groups

wx \ yz	00	01	11	10
00	1		1	
01	1		1	1
11	1			
10	1			

The first column should clearly be grouped. Also, the $\bar{w}\bar{x}yz$ and $\bar{w}xyz$ terms should be grouped. However, we have a choice as to how to group the $\bar{w}xy\bar{z}$ term. It could be grouped with $\bar{w}xyz$ or with $\bar{w}x\bar{y}\bar{z}$ (as a wrap-around). These two solutions are indicated below.

wx \ yz	00	01	11	10
00	1		1	
01	1		1	1
11	1			
10	1			

wx \ yz	00	01	11	10
00	1		1	
01	1		1	1
11	1			
10	1			

The first map simplifies to $F(w, x, y, z) = F_1 = \bar{y}\bar{z} + \bar{w}yz + \bar{w}xy$. The second map simplifies to $F(w, x, y, z) = F_2 = \bar{y}\bar{z} + \bar{w}yz + \bar{w}x\bar{z}$. The last terms are different. F_1 and F_2 , however, are equivalent. We leave it up to you to produce the truth tables for F_1 and F_2 to check for equality. They both have the same number of terms and variables as well. If we follow the rules, Kmap minimization results in a minimized function (and thus a minimal circuit), but these minimized functions need not be unique in representation.

The rules for Kmap simplification can be summarized as follows:

1. The groups can only contain 1s; no 0s.
2. Only 1s in adjacent cells can be grouped; diagonal grouping is not allowed.
3. The number of 1s in a group must be a power of 2.
4. The groups must be as large as possible while still following all rules.
5. All 1s must belong to a group, even if it is a group of one.
6. Overlapping groups are allowed.
7. Wrap around is allowed.
8. Use the fewest number of groups possible.

Using these rules, let's complete one more example for a four-variable function. Example 3A.8 shows several applications of the various rules.

EXAMPLE 3A.8

$$F(w, x, y, z) = \bar{w}\bar{x}\bar{y}\bar{z} + \bar{w}\bar{x}yz + \bar{w}x\bar{y}z + \bar{w}xyz \\ + wx\bar{y}z + wxyz + w\bar{x}yz + w\bar{x}y\bar{z}$$

$wx \backslash yz$	00	01	11	10
00	1		1	
01		1	1	
11		1	1	
10			1	1

In this example, we have one group with a single element. Note there is no way to group this term with any others if we follow the rules. The function represented by this Kmap simplifies to $F(w, x, y, z) = yz + xz + \bar{w}\bar{x}\bar{y}\bar{z}$.

If you are given a function that is not written as a sum of minterms, you can still use Kmaps to help minimize the function. However, you have to use a procedure that is somewhat the reverse of what we have been doing to set up the Kmap before reduction can occur. Example 3A.9 illustrates this procedure.

EXAMPLE 3A.9 A Function Not Represented as a Sum of Minterms

Suppose you are given the function $F(w, x, y, z) = \bar{w}xy + \bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z}$. The last two terms are minterms, and we can easily place 1s in the appropriate positions in the Kmap. However the term $\bar{w}xy$ is not a minterm. Suppose this term were the *result* of a grouping you had performed on a Kmap. The term that was discarded was the z term, which means this term is equivalent to the two terms $\bar{w}xy\bar{z} + \bar{w}xyz$. You can now use these two terms in the Kmap, because they are both minterms. We now get the following Kmap:

		yz			
		00	01	11	10
wx	00			1	1
	01			1	1
	11				
	10				

So we know the function $F(w, x, y, z) = \bar{w}xy + \bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z}$ simplifies to $F(w, x, y, z) = \bar{w}y$.

EXAMPLE 3A.10

Figs 6.13(a) and (b) respectively show minterm and maxterm Karnaugh maps for the Boolean functions expressed by equations (6.50) and (6.51). The minimized expressions as deduced from Karnaugh maps in the two cases are given by Equation (6.52) in the case of the minterm Karnaugh map and Equation (6.53) in the case of the maxterm Karnaugh map:

$$Y = \bar{A}.\bar{B}.\bar{C}.\bar{D} + \bar{A}.\bar{B}.C.\bar{D} + \bar{A}.B.\bar{C}.D + \bar{A}.B.C.D + A.\bar{B}.\bar{C}.\bar{D} + A.\bar{B}.C.\bar{D} + A.B.\bar{C}.D + A.B.C.D \quad (6.50)$$

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

$$.(A + \bar{B} + \bar{C} + D).(\bar{A} + \bar{B} + C + \bar{D}).(\bar{A} + \bar{B} + \bar{C} + \bar{D}).(\bar{A} + B + C + \bar{D}).(\bar{A} + B + \bar{C} + \bar{D}) \quad (6.51)$$

$$Y = \bar{B}.\bar{D} + B.D \quad (6.52)$$

$$Y = \bar{D}.(A + \bar{B}) \quad (6.53)$$

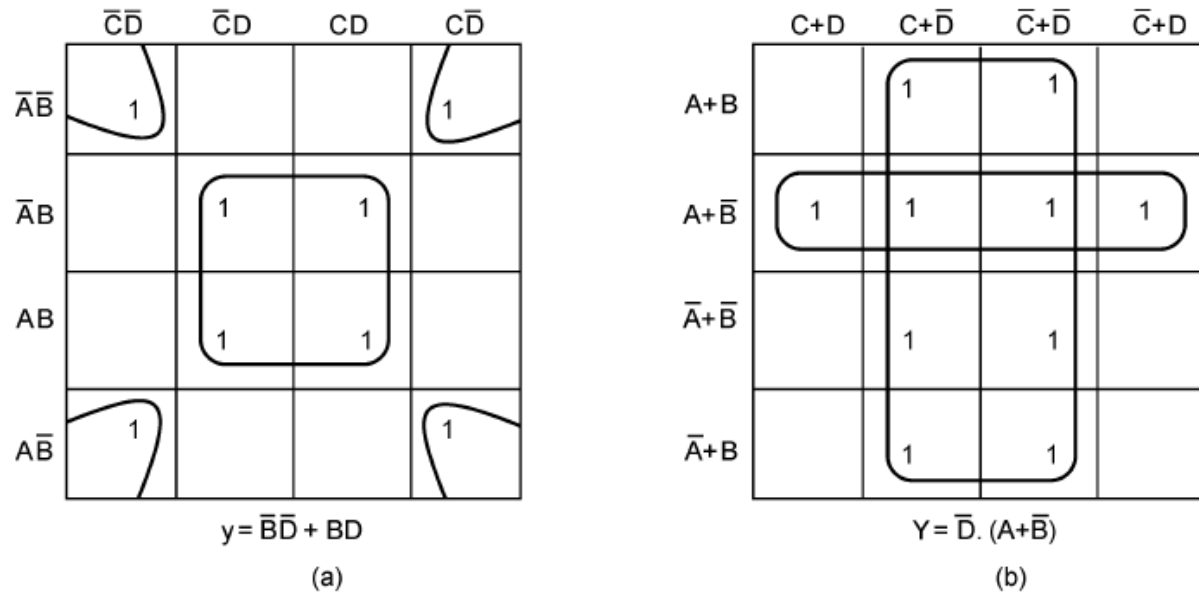


Figure 6.13

3A.6 DON'T CARE CONDITIONS

There are certain situations where a function may not be completely specified, meaning there may be some inputs that are undefined for the function. For example, consider a function with 4 inputs that act as bits to count, in binary, from 0 to 10 (decimal). We use the bit combinations 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, and 1010. However, we do not use the combinations 1011, 1100, 1101, 1110, and 1111. These latter inputs would be invalid, which means if we look at the truth table, these values wouldn't be either 0 or 1. They should not be in the truth table at all. We can use these *don't care* inputs to our advantage when simplifying Kmaps. Because they are input values that should not matter (and should never occur), we can let them have values of either 0 or 1, depending on which helps us the most. The basic idea is to set these don't care values in such a way that they either contribute to make a larger group, or they don't contribute at all. Example 3A.10 illustrates this concept.

EXAMPLE 3A.10 Don't Care Conditions

Don't care values are typically indicated with an "X" in the appropriate cell. The following Kmap shows how to use these values to help with minimization. We treat the don't care values in the first row as 1s to help form a group of four. The don't care values in rows 01 and 11 are treated as 0s. This reduces to

$$F_1(w, x, y, z) = \overline{w}\overline{x} + yz.$$

$wx \backslash yz$	00	01	11	10
00	X	1	1	X
01		X	1	
11	X		1	
10			1	

There is another way these values can be grouped:

$wx \backslash yz$	00	01	11	10
00	X	1	1	X
01		X	1	
11	X		1	
10			1	

Using the above groupings, we end up with a simplification of $F_2(w, x, y, z) = \overline{w}z + yz$. Notice that in this case, F_1 and F_2 are not equal. However, if you create the truth tables for both functions, you should see that they are not equal only in those values for which we “don’t care.”