2.3 Karnaugh Maps

Algebraic minimization presented in the previous section becomes tedious when large logical expressions are encountered. The *Karnaugh map* provides a simplified method for minimizing Boolean functions. A Karnaugh map provides a geometrical representation of a Boolean function.

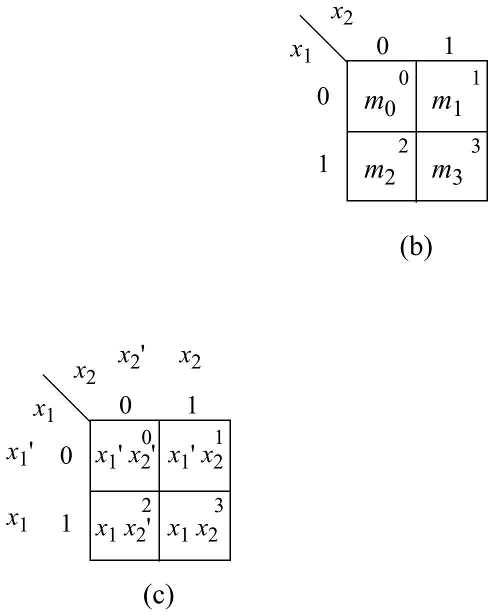
The Karnaugh map is arranged as an array of squares (or cells) in which each square represents a binary value of the input variables. The map is a convenient method of obtaining a minimal number of terms with a minimal number of variables per term for a Boolean function. A Karnaugh map presents a clear indication of function minimization without recourse to Boolean algebra and will generate a minimized expression in either a sum-of-products form or a product-of-sums form.

The Karnaugh map is a diagram consisting of squares, where each square represents a minterm. Since any Boolean function can be expressed as a sum of minterms, a Karnaugh map presents a graphical representation of the Boolean function and is an ideal tool for minimization.

Figure 2.1 illustrates a two-variable Karnaugh map and the corresponding truth table. Since there are four minterms for two variables, the map consists of four squares, one for each minterm. Figure 2.1 (a) shows the truth table for the four minterms that represent the two variables; Figure 2.1 (b) shows the position of the minterms in the map; and Figure 2.1 (c) shows the relationship between the squares and the two variables.

x_1	x_2	Minterm
0	0	m_0
0	1	m_1
1	0	m_2
1	1	m_3

(a)



(b)

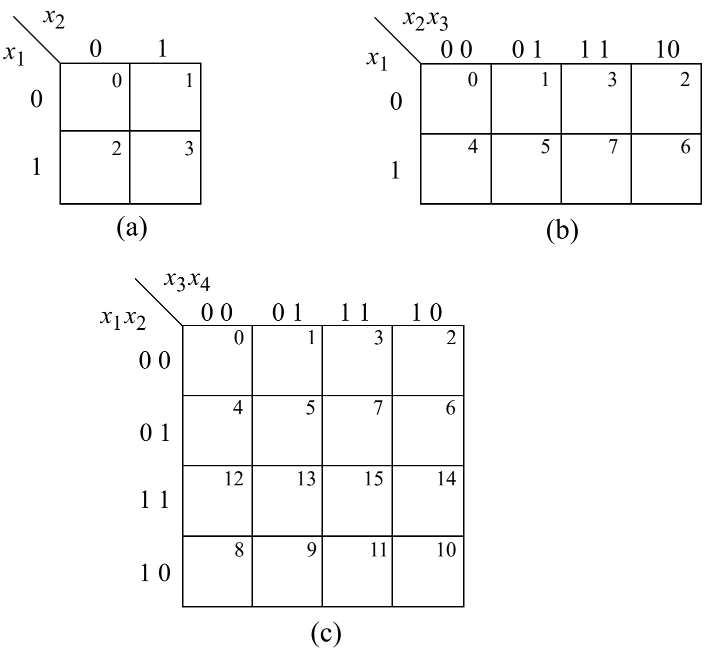
(c)

Figure 2.1 Two-variable Karnaugh map: (a) truth table; (b) minterm placement; and (c) minterms.

Figure 2.2 shows Karnaugh maps for two, three, four, and five variables. Each square in the maps corresponds to a unique minterm. The maps for three or more variables contain column headings that are represented in the *Gray code* format; the maps for four or more variables contain column and row headings that are represented in Gray code. Using the Gray code to designate column and row headings permits physically adjacent squares to be also logically adjacent; that is, to differ by only one variable. Map entries that are adjacent can be combined into a single term. For example, the expression $z_1 = x_1x_2'x_3 + x_1x_2x_3$, which corresponds to minterms 5 and 7 in Figure 2.2 (b), reduces to $z_1 = x_1x_3(x_2' + x_2) = x_1x_3$ using the distributive and complementation laws. Thus, if 1s are entered in minterm locations 5 and 7, then the two minterms can be combined into the single term x_1x_3.

Similarly, in Figure 2.2 (c), if 1s are entered in minterm locations 4, 6, 12, and 14, then the four minterms combine as x_2x_4'. That is, only variables x_2 and x_4' are common to all four squares — variables x_1 and x_3 are discarded by the complementation law. The minimized expression obtained from the Karnaugh map can be verified algebraically by listing the four minterms as a sum-of-minterms expression, then applying the appropriate laws of Boolean algebra as shown below.

$$\begin{aligned} & x_1'x_2x_3'x_4' + x_1'x_2x_3x_4' + x_1x_2x_3'x_4' + x_1x_2x_3x_4' \\ & = x_2x_4'(x_1'x_3' + x_1'x_3 + x_1x_3' + x_1x_3) \\ & = x_2x_4' \end{aligned}$$

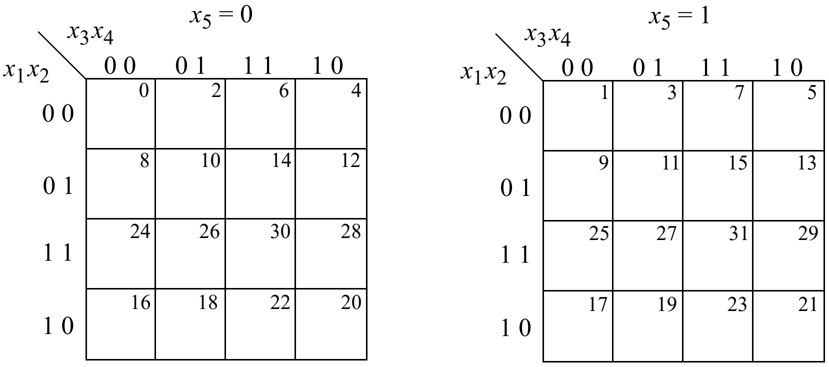


(a)

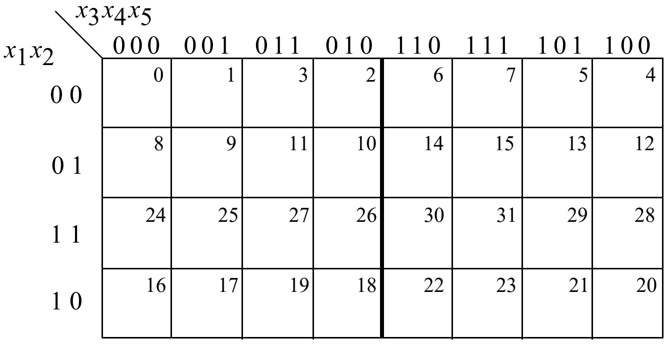
(b)

(c)

Figure 2.2 Karnaugh maps showing minterm locations: (a) two variables; (b) three variables; (c) four variables; (d) five variables; and (e) alternative map for five variables.



(d)



(e)

Figure 2.2 (Continued)

When minimizing a Boolean expression by grouping the 1s in a Karnaugh map, the result will be in a sum-of-products form; grouping the 0s results in a product-of-sums form. Each product term in a sum-of-products expression is specified as an *implicant* of the function, since the product term implies the function. That is, if the product term is equal to 1, then the function is also equal to 1. Specifically, a *prime implicant* is a unique grouping of 1s (an implicant) that does not imply any other grouping of 1s (other implicants).

Example 2.9 The following function will be minimized using a 4-variable Karnaugh map:

$$z_1(x_1,x_2,x_3,x_4) = x_2x_3' + x_2x_3x_4' + x_1x_2'x_3 + x_1x_3x_4'$$

The minimized result will be obtained in both a sum-of-products form and a product-of-sums form. To plot the function in the Karnaugh map, 1s are entered in the minterm locations that represent the product terms. For example, the term x_2x_3' is represented by the 1s in minterm locations 4, 5, 12, and 13. Only variables x_2 and x_3' are common to these four minterm locations. The term $x_2x_3x_4'$ is entered in minterm locations 6 and 14. When the function has been plotted, a minimal set of prime implicants can be obtained to represent the function. The largest grouping of 1s should always be combined, where the number of 1s in a group is a power of 2. The grouping of 1s is shown in Figure 2.3 and the resulting equation in Equation 2.1 in a sum-of-products notation.

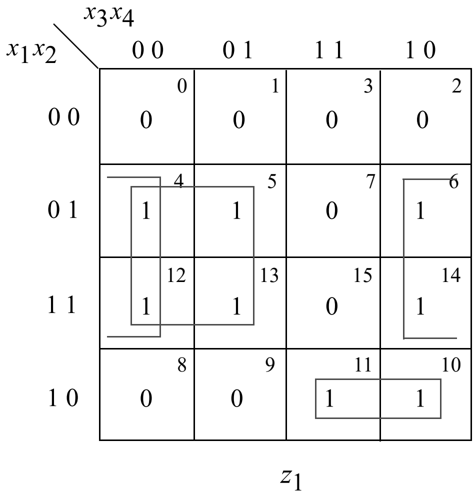


Figure 2.3 Karnaugh map representation of the function $z_1(x_1,x_2,x_3,x_4) = x_2x_3' + x_2x_3x_4' + x_1x_2'x_3 + x_1x_3x_4'$.

$$z_1(x_1,x_2,x_3,x_4) = x_2x_3' + x_2x_4' + x_1x_2'x_3 \tag{2.1}$$

The minimal product-of-sums expression can be obtained by combining the 0s in Figure 2.3 to form sum terms in the same manner as the 1s were combined to form product terms. However, since 0s are being combined, each sum term must equal 0. Thus, the four 0s in row $x_1x_2 = 00$ in Figure 1.2 combine to yield the sum term $(x_1 + x_2)$. In a similar manner, the remaining 0s are combined to yield the product-of-sums expression shown in Equation 2.2. When combining 0s to obtain sum terms, treat a variable value of 1 as false and a variable value of 0 as true. Thus, maxterm locations 7 and 15 have variables $x_2x_3x_4 = 111$, providing a sum term of $(x_2' + x_3' + x_4')$.

$$z_1(x_1,x_2,x_3,x_4) = (x_1 + x_2)(x_2 + x_3)(x_2' + x_3' + x_4') \quad (2.2)$$

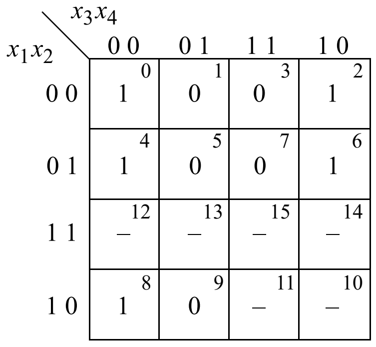
Equation 2.1 and Equation 2.2 both specify the conditions where z_1 is equal to 1. For example, consider the first term of Equation 2.1. If $x_2\,x_3 = 10$, then Equation 2.1 yields $z_1 = 1 + \cdots + 0$, which generates a value of 1 for z_1. Applying $x_2\,x_3 = 10$ to Equation 2.2 will cause every term to be equal to 1, such that $z_1 = (1)\,(1)\,(1) = 1$.

Figure 2.2 (d) illustrates a 5-variable Karnaugh map. To determine adjacency, the left map is superimposed on the right map. Any cells that are then physically adjacent are also logically adjacent and can be combined. Since x_5 is the low-order variable, the left map contains only even-numbered minterms; the right map is characterized by odd-numbered minterms. If 1s are entered in minterm locations 28, 29, 30, and 31, the four cells combine to yield the term $x_1\,x_2\,x_3$.

Figure 2.2 (e) illustrates an alternative configuration for a Karnaugh map for five variables. The map hinges along the vertical centerline and folds like a book. Any squares that are then physically adjacent are also logically adjacent. For example, if 1s are entered in minterm locations 24, 25, 28, and 29, then the four squares combine to yield the term $x_1\,x_2\,x_4'$.

Some minterm locations in a Karnaugh map may contain unspecified entries which can be used as either 1s or 0s when minimizing the function. These "*don't care*" entries are indicated by a dash (–) in the map. A typical situation which includes "don't care" entries is a Karnaugh map used to represent the BCD numbers. This requires a 4-variable map in which minterm locations 10 through 15 contain unspecified entries, since digits 10 through 15 are invalid for BCD.

Example 2.10 A minimized equation will be derived which is asserted whenever a BCD digit is even. All even BCD digits are plotted on a Karnaugh map as shown in Figure 2.4 for function z_1. The unspecified entries in minterm locations 10, 12, and 14 are assigned a value of 1; all remaining unspecified entries are assigned a value of 0. The equation for z_1 is shown in Equation 2.3.



x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0 1	1 0	3 0	2 1
0 1	4 1	5 0	7 0	6 1
1 1	12 –	13 –	15 –	14 –
1 0	8 1	9 0	11 –	10 –

z_1

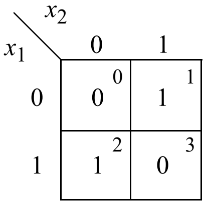
Figure 2.4 Karnaugh map to represent even-numbered BCD digits.

$$z_1 = x_4' \tag{2.3}$$

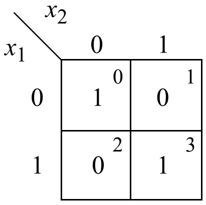
To obtain the equation which specifies BCD digits that are evenly divisible by 3, 1s are entered in minterm locations 0, 3, 6, and 9 to yield Equation 2.4.

$$z_1 = x_1x_4 + x_2'x_3x_4 + x_2x_3x_4' + x_1'x_2'x_3'x_4' \tag{2.4}$$

Example 2.11 The exclusive-OR function and the exclusive-NOR function will be plotted on a 2-variable Karnaugh map. The exclusive-OR function is $f = x_1x_2' + x_1'x_2$; the exclusive-NOR function is $f = (x_1x_2' + x_1'x_2)' = x_1'x_2' + x_1x_2$.



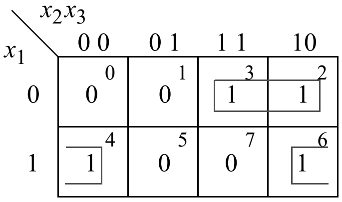
$f = x_1x_2' + x_1'x_2$



$f = x_1'x_2' + x_1x_2$

Example 2.12 Given the truth table shown below for the function *f*, the function will be plotted on a 3-variable Karnaugh map to obtain the minimized expression for *f* in a sum-of-products notation.

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



x_1 \ x_2x_3	0 0	0 1	1 1	10
0	0 0	1 0	3 1	2 1
1	4 1	5 0	7 0	6 1

f

$f = x_1x_3' + x_1'x_2$

Example 2.13 The function shown below will be minimized using a Karnaugh map and the solution verified using Boolean algebra.

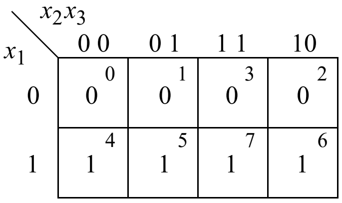
$$f = x_1x_2x_3' + x_1x_2'x_3' + x_1x_2x_3 + x_1x_2'x_3$$

The minterms in the function correspond to minterms 6, 4, 7, and 5, respectively. Thus, the function can also be written in disjunctive normal form using decimal notation as

$$f(x_1, x_2, x_3) = \Sigma_m(4, 5, 6, 7), \text{ where the function } f = 1$$

or as

$$f'(x_1, x_2, x_3) = \Sigma_m(0, 1, 2, 3), \text{ where the function } f = 0$$



x_1 \ x_2x_3	0 0	0 1	1 1	10
0	0 0	1 0	3 0	2 0
1	4 1	5 1	7 1	6 1

f

The minimized expression for the function *f* as obtained from the Karnaugh map is

$$f = x_1$$

Verify using Boolean algebra.

$$\begin{aligned} f = \; & f = x_1x_2x_3' + x_1x_2'x_3' + x_1x_2x_3 + x_1x_2'x_3 \\ = \; & x_1(x_2x_3' + x_2'x_3' + x_2x_3 + x_2'x_3) \\ = \; & x_1[x_2(x_3' + x_3) + x_2'(x_3' + x_3)] \\ = \; & x_1(x_2 + x_2') \\ = \; & x_1 \end{aligned}$$

Example 2.14 The Karnaugh map of Figure 2.5 shows a redundant grouping of 1s that combines minterm locations 4 and 6. If all the 1s are covered, then any other grouping of 1s is redundant, except for hazards, which are covered in Chapter 3. The minimized equation for the function *f* is

$$f = x_1x_2 + x_2'x_3'$$

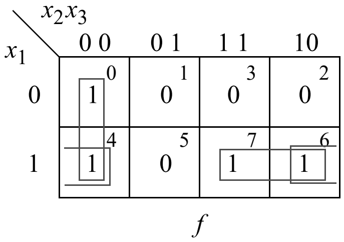


Figure 2.5 Karnaugh map showing a redundant product term of x_1x_3'.

To illustrate that x_1x_3' is redundant, the term will be included in the equation for *f*, then minimized using Boolean algebra, as shown below.

$$\begin{aligned}
f &= x_1x_2 + x_2'x_3' + x_1x_3' && \text{Expand to sum of minterms}\\
&= x_1x_2(x_3 + x_3') + x_2'x_3'(x_1 + x_1') + x_1x_3'(x_2 + x_2')\\
&= x_1x_2x_3 + x_1x_2x_3' + x_1x_2'x_3' + x_1'x_2'x_3' + x_1x_2x_3'\\
&\quad + x_1x_2'x_3'\\
&= x_1x_2x_3 + x_1x_2x_3' + x_1x_2'x_3' + x_1'x_2'x_3'\\
&= x_1x_2(x_3 + x_3') + x_2'x_3'(x_1 + x_1')\\
&= x_1x_2 + x_2'x_3'
\end{aligned}$$

Example 2.15 Always group the largest number of 1s; that is, a grouping a 1, 2, 4, 8, or 16 1s, or a any number of 1s that is a power of two. In Figure 2.6 (a), if minterm locations 3 and 7 were combined in a group and minterm locations 2 and 6 were combined in a separate group, then this would result in an incorrect grouping, resulting in an equation for the function *f* as follows:

$$f = x_2x_3 + x_2x_3'$$

The above equation will function correctly, but it is not minimized. The correct grouping of minterm locations should be minterm locations 2, 3, 6, and 7 as a single grouping. This can be verified by Boolean algebra, as follows:

$$\begin{aligned} f &= x_2x_3 + x_2x_3' \\ &= x_2(x_3 + x_3') \\ &= x_2 \end{aligned}$$

Similarly, in Figure 2.6 (b), if minterm locations 2, 3, 6, and 7 are one grouping and minterm locations 1 and 5 are a second grouping, then the second grouping is incorrect to obtain a minimized equation for the function *f*. The correct groupings are as follows:

- Group minterm locations 2, 3, 6, and 7 to obtain the term x_2
- Group minterm locations 1, 3, 5, and 7 to obtain the term x_3 using minterm locations 3 and 7 a second time.

These groupings provide a minimized equation for the function *f* as follows:

$$f = x_2 + x_3$$

The non-minimized equation for *f* can be reduced by Boolean algebra as shown below.

$$\begin{aligned} f &= x_2 + x_2'x_3 \\ &= x_2 + x_3 \qquad \text{Absorption law} \end{aligned}$$

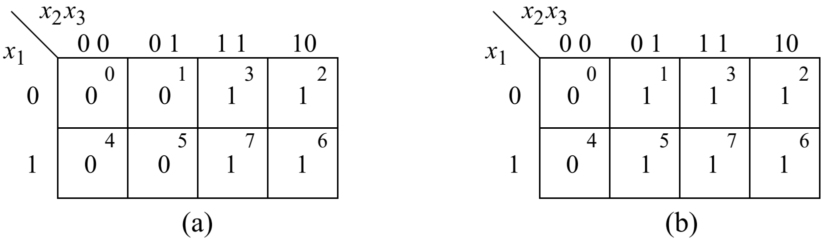


Figure 2.6 Karnaugh maps for Example 2.15.

Example 2.16 The following equation will be minimized as a sum of products and also as a product of sums:

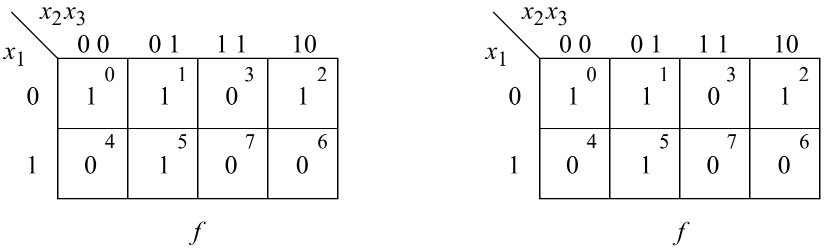
$$f(x_1, x_2, x_3) = \Sigma_m(0, 1, 2, 5).$$

Figure 2.7 (a) shows the equation plotted on a 3-variable Karnaugh map as a function of x_1, x_2, and x_3, where the 1s are grouped to obtain a minimum sum of products, as shown in Equation 2.5. The value of *f* in Equation 2.5 is expressed as a logic 1.

$$f = x_1'x_3' + x_2'x_3 \tag{2.5}$$

Figure 2.7 (b) shows the same map where the 0s are grouped to obtain a minimum product of sums, as shown in Equation 2.6. The value of *f* in Equation 2.6 is expressed as a logic 0.

$$f = (x_1' + x_3)(x_2' + x_3') \tag{2.6}$$



$x_1 \backslash x_2x_3$	0 0	0 1	1 1	10
0	0 1	1 1	3 0	2 1
1	4 0	5 1	7 0	6 0

f

(a)

$x_1 \backslash x_2x_3$	0 0	0 1	1 1	10
0	0 1	1 1	3 0	2 1
1	4 0	5 1	7 0	6 0

f

(b)

Figure 2.7 Karnaugh maps for Example 2.16: (a) combine 1s to obtain a sum of products; and (b) combine 0s to obtain a product of sums.

The complement of *f* is obtained by combining 0s, as shown in Equation 2.7. The product-of-sums expression can also be obtained by complementing the sum-of-products expression for *f*', which has a value of 0, as shown below.

$$f' = x_1x_3' + x_2x_3 \tag{2.7}$$

$$\begin{aligned} f' &= x_1x_3' + x_2x_3 \\ f'' = \quad f &= (x_1x_3' + x_2x_3)' \\ &= (x_1' + x_3)(x_2' + x_3') \end{aligned}$$

Alternatively, the product of sums can be obtained directly from the map by combining the 0s, where the true value of a variable is treated as a 0 and the false value of a variable is treated as a 1, as shown below.

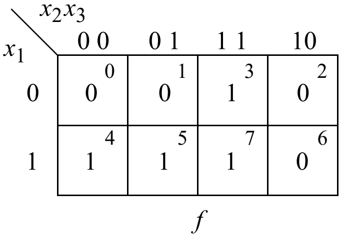
$$f = (x_1' + x_3)(x_2' + x_3') = 1$$

where $x_1' = 1, x_3 = 0, x_2' = 1$, and $x_3' = 1$. Thus, the equation for the function *f* can be expressed in two different ways, resulting in a value of logic 1, as shown in Equation 2.8.

$$f = x_1'x_3' + x_2'x_3 = (x_1' + x_3)(x_2' + x_3') = 1 \tag{2.8}$$

Example 2.17 Karnaugh maps help to eliminate redundant terms in a minimized expression. The following equation will be minimized using the Karnaugh map of Figure 2.8 to obtain a product of sums:

$$f = x_1x_2' + x_2x_3$$



$x_1 \backslash x_2x_3$	0 0	0 1	1 1	10
0	0 0	1 0	3 1	2 0
1	4 1	5 1	7 1	6 0

Figure 2.8 Karnaugh map for Example 2.17.

A Karnaugh map provides a clear indication of the function, making it easy to obtain either a sum-of-products expression or a product-of-sums expression. It is readily apparent that the minimized product-of-sums expression will contain two terms by combining two sets of adjacent 0s — maxterm locations 0 and 1; and maxterm locations 2 and 6, as shown in Equation 2.9.

$$f = (x_1 + x_2)(x_2' + x_3) \tag{2.9}$$

Using Boolean algebra to obtain a minimized product-of-sums expression may introduce redundant terms, since it may not be immediately apparent that a term is redundant. Boolean algebra will now be used to obtain a minimized product-of-sums expression from the original equation $f = x_1x_2' + x_2x_3$.

$$\begin{aligned} f &= x_1x_2' + x_2x_3 \\ &= (x_1x_2' + x_2)(x_1x_2' + x_3) && \text{Distributive law} \\ &= (x_1 + x_2)(x_1 + x_3)(x_2' + x_3) && \text{Absorption law, distributive law} \end{aligned}$$

The term $(x_1 + x_3)$ is redundant, because all the 0s are covered. This term represents maxterm locations 0 and 2, which are included in the other two terms.

Example 2.18 In the truth table shown in Table 2.4 under the column labeled *f*, the 1 entries indicate minterms; the 0 entries indicate maxterms. The function *f* can be expressed as a sum of minterms by combining the 1 entries in a disjunctive normal form.

$$f(x_1, x_2, x_3) = \Sigma_m(1, 3, 4, 6)$$
$$f(x_1, x_2, x_3) = x_1'x_2'x_3 + x_1'x_2x_3 + x_1x_2'x_3' + x_1x_2x_3'$$

or as a product of maxterms by combining the 0s in a conjunctive normal form.

$$f(x_1, x_2, x_3) = \Pi_M(0, 2, 5, 7)$$
$$f(x_1, x_2, x_3) = (x_1 + x_2 + x_3)(x_1 + x_2' + x_3)(x_1' + x_2 + x_3')(x_1' + x_2' + x_3')$$

where the uncomplemented variables represent 0s in Table 2.4 and the complemented variables represent 1s in Table 2.4.

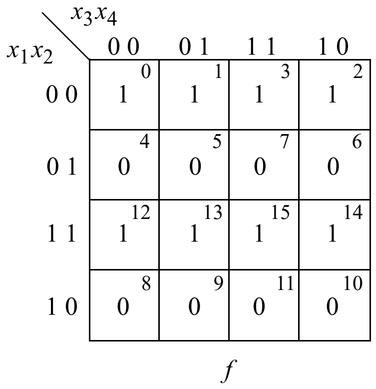
Table 2.4 Truth Table for the Function *f* of Example 2.18

x_1	x_2	x_3	Function *f*	
0	0	0	0	Maxterm
0	0	1	1	Minterm
0	1	0	0	Maxterm
0	1	1	1	Minterm
1	0	0	1	Minterm
1	0	1	0	Maxterm
1	1	0	1	Minterm
1	1	1	0	Maxterm

Example 2.19 The function shown in Table 2.5 will be plotted on the 4-variable Karnaugh map shown in Figure 2.9 and then minimized as a sum of products.

Table 2.5 Truth Table for the Function *f* of Example 2.19

x_1	x_2	x_3	x_4	Function *f*	x_1	x_2	x_3	x_4	Function *f*
0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	0	1	1	1	1	0	1	1	0
0	1	0	0	0	1	1	0	0	1
0	1	0	1	0	1	1	0	1	1
0	1	1	0	0	1	1	1	0	1
0	1	1	1	0	1	1	1	1	1



x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0 1	1 1	3 1	2 1
0 1	4 0	5 0	7 0	6 0
1 1	12 1	13 1	15 1	14 1
1 0	8 0	9 0	11 0	10 0

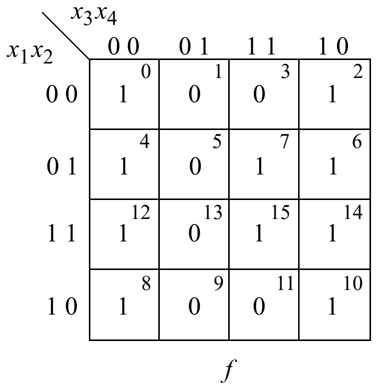
f

Figure 2.9 Karnaugh map for Example 2.19.

The function $f = x_1'x_2' + x_1x_2$, which is the exclusive-NOR function in which the function is a logic 1 only when the variables are equal. The exclusive-NOR function is also known as the *equality function*.

Example 2.20 The following equation will be minimized utilizing the Karnaugh map of Figure 2.10 and the result will be verified using Boolean algebra:

$$f = x_1'x_3'x_4' + x_1'x_3x_4' + x_1x_3'x_4' + x_2x_3x_4 + x_1x_3x_4'$$



x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0 1	1 0	3 0	2 1
0 1	4 1	5 0	7 1	6 1
1 1	12 1	13 0	15 1	14 1
1 0	8 1	9 0	11 0	10 1

f

Figure 2.10 Karnaugh map for Example 2.20.

The minimized function is $f = x_4' + x_2x_3$.

Verify using Boolean algebra.

$$\begin{aligned} f &= x_1'x_3'x_4' + x_1'x_3x_4' + x_1x_3'x_4' + x_2x_3x_4 + x_1x_3x_4' \\ &= x_3'x_4' + x_3x_4' + x_2x_3x_4 \\ &= x_4' + (x_2x_3)x_4 \\ &= x_4' + x_2x_3 \end{aligned}$$

Example 2.21 Given the Karnaugh map in Figure 2.11, the function *f* will be minimized as a sum-of-products expression and also as a product-of-sums expression.

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0 0	1 0	3 1	2 0
0 1	4 0	5 1	7 1	6 0
1 1	12 0	13 1	15 1	14 0
1 0	8 1	9 1	11 1	10 1

f

Figure 2.11 Karnaugh map for Example 2.21.

Minimum sum of products: $f = x_1x_2' + x_2x_4 + x_3x_4$

Minimum product of sums: $f = (x_2' + x_4)(x_1 + x_4)(x_1 + x_2 + x_3)$

Example 2.22 This example will obtain an equation which generates an output of logic 1 whenever a 4-bit unsigned binary number $f = x_1x_2x_3x_4$ is greater than five, but less than ten, where x_4 is the low-order bit. The equation will be in a minimum sum-of-products form.

The Karnaugh map of Figure 2.12 is used to portray the specifications of Example 2.22. Thus, entries of 1 are placed in minterm locations 6, 7, 8, and 9, which satisfies the requirement of the number being greater than five, but less than ten.

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0 0	1 0	3 0	2 0
0 1	4 0	5 0	7 1	6 1
1 1	12 0	13 0	15 0	14 0
1 0	8 1	9 1	11 0	10 0

f

Figure 2.12 Karnaugh map for Example 2.22 to indicate when a number is greater than five, but less than ten.

The following function meets the requirements of Example 2.22:

$$f = x_1'x_2x_3 + x_1x_2'x_3'$$

Example 2.23 This example will obtain the minimum product-of-sums expression for the function *f* represented by the Karnaugh map of Figure 2.13.

$x_5 = 0$

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0 0	2 1	6 1	4 0
0 1	8 1	10 1	14 1	12 1
1 1	24 0	26 1	30 0	28 0
1 0	16 0	18 1	22 1	20 1

$x_5 = 1$

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	1 0	3 1	7 1	5 0
0 1	9 1	11 1	15 1	13 1
1 1	25 1	27 1	31 0	29 0
1 0	17 1	19 1	23 1	21 0

f

Figure 2.13 Karnaugh map for Example 2.23.

Maxterm locations 30, 28, 31, and 29 are all adjacent and combine to yield the sum term of $(x_1' + x_2' + x_3')$ as follows:

$(x_1' + x_2' + x_3' + x_4' + x_5)\ (x_1' + x_2' + x_3' + x_4 + x_5)\ (x_1' + x_2' + x_3' + x_4' + x_5')$
$(x_1' + x_2' + x_3' + x_4 + x_5')$

Using the first two maxterms:
$(x_1' + x_2' + x_3' + x_4' + x_5)\ (x_1' + x_2' + x_3' + x_4 + x_5)$
$= (x_1' + x_2' + x_3') + [(x_4' + x_5)\ (x_4 + x_5)]$
$= (x_1' + x_2' + x_3') + [(x_4' + x_5)x_4 + (x_4' + x_5)x_5]$
$= (x_1' + x_2' + x_3') + [x_4x_5 + x_4'x_5 + x_5]$
$= (x_1' + x_2' + x_3') + x_5 + x_5$
$= (x_1' + x_2' + x_3') + x_5$

Using the last two maxterms:
$(x_1' + x_2' + x_3' + x_4' + x_5')\ (x_1' + x_2' + x_3' + x_4 + x_5')$
$= (x_1' + x_2' + x_3') + [(x_4' + x_5')\ (x_4 + x_5')]$
$= (x_1' + x_2' + x_3' + [(x_4' + x_5')x_4 + (x_4' + x_5')x_5']$
$= (x_1' + x_2' + x_3') + (x_4x_5' + x_4'x_5' + x_5')$
$= (x_1' + x_2' + x_3') + x_5' + x_5'$
$= (x_1' + x_2' + x_3') + x_5'$

Combine all four maxterms:
$(x_1' + x_2' + x_3' + x_4' + x_5)\ (x_1' + x_2' + x_3' + x_4 + x_5)\ (x_1' + x_2' + x_3' + x_4' + x_5')$
$(x_1' + x_2' + x_3' + x_4 + x_5')$
$= [(x_1' + x_2' + x_3') + x_5]\ [(x_1' + x_2' + x_3') + x_5']$
$= [(x_1' + x_2' + x_3') + x_5]\ [(x_1' + x_2' + x_3')] + [(x_1' + x_2' + x_3') + x_5]x_5'$
$= (x_1' + x_2' + x_3') + (x_1' + x_2' + x_3')x_5 + (x_1' + x_2' + x_3')x_5'$
$= [(x_1' + x_2' + x_3')]\ (1 + x_5 + x_5')$
$= (x_1' + x_2' + x_3')$

In a similar manner, maxterms 0, 4, 1, and 5 combine to yield the sum term of $(x_1 + x_2 + x_4)$; maxterms 24 and 16 combine to yield $(x_1' + x_3 + x_4 + x_5)$; and maxterms 29 and 21 combine to yield $(x_1' + x_3' + x_4 + x_5')$. Therefore, the minimized function f as a product of sums is

$$f = (x_1' + x_2' + x_3')\ (x_1 + x_2 + x_4)\ (x_1' + x_3 + x_4 + x_5)\ (x_1' + x_3' + x_4 + x_5')$$

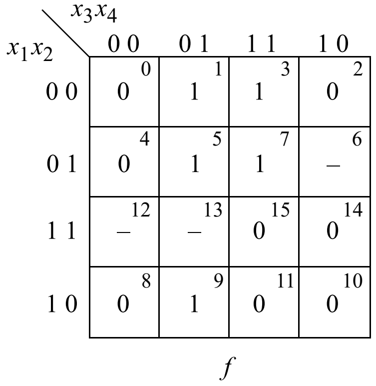
An equivalent product of sums can be obtained by combining maxterms 5 and 21 for the last term to yield

$$f = (x_1' + x_2' + x_3')\ (x_1 + x_2 + x_4)\ (x_1' + x_3 + x_4 + x_5)\ (x_2 + x_3' + x_4 + x_5')$$

Example 2.24 The following function will be plotted on the 4-variable Karnaugh map of Figure 2.14 and then minimized to obtain a sum-of-products expression:

$$f(x_1, x_2, x_3, x_4) = \Sigma_m(1, 3, 5, 7, 9) + \Sigma_d(6, 12, 13)$$

where $\Sigma_d(6, 12, 13)$ represents "don't care" entries for minterm locations 6, 12, and 13.



x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0 0	1 1	3 1	2 0
0 1	4 0	5 1	7 1	6 –
1 1	12 –	13 –	15 0	14 0
1 0	8 0	9 1	11 0	10 0

f

Figure 2.14 Karnaugh map for Example 2.24.

Minterm locations 1, 5, 13, and 9 combine to yield the product term $x_3'x_4$. Minterm locations 1, 3, 5, and 7 combine to yield the product term $x_1'x_4$. The minimized sum-of-products expression is

$$f = x_3'x_4 + x_1'x_4$$

This can be further minimized by using the distributive law of Boolean algebra to obtain the following expression, which is in a product-of-sums form:

$$f = x_4(x_3' + x_1')$$

The product-of-sums form can also be obtained by combining the 0s and "don't cares" in the Karnaugh map of Figure 2.14.

Example 2.25 The Karnaugh map of Figure 2.15 depicts a function with unspecified entries that will be minimized as a sum of products and as a product of sums. Minterm locations 12, 8, 14, and 10 combine to yield the product term x_1x_4'; minterm locations 7, 6, 15, and 14 combine to yield x_2x_3; minterm locations 2, 6, 14, and 10 combine to yield x_3x_4'; and minterm location 1 yields $x_1'x_2'x_3'x_4$.

Using the same map, 0s are combined with unspecified entries to form sum terms. Maxterm locations 13, 15, 9, and 11 combine to yield the sum term $(x_1' + x_4')$; maxterm locations 4, 5, 12, and 13 combine to yield $(x_2' + x_3)$; maxterm locations 3 and 11 combine to yield $(x_2 + x_3' + x_4')$; and maxterm locations 0 and 4 combine to yield $(x_1 + x_3 + x_4)$.

x_1x_2 \ x_3x_4	00	01	11	10
00	0 0	1 1	3 0	2 1
01	4 0	5 0	7 1	6 1
11	12 –	13 –	15 –	14 –
10	8 1	9 0	11 –	10 –

f

Figure 2.15 Karnaugh map for Example 2.25.

Sum of products: $f = x_1x_4' + x_2x_3 + x_3x_4' + x_1'x_2'x_3'x_4$
Product of sums: $f = (x_1' + x_4')\,(x_2' + x_3)\,(x_2 + x_3' + x_4')\,(x_1 + x_3 + x_4)$

Example 2.26 Given the four variables $x_1x_2x_3x_4$, an expression will be obtained to satisfy the following requirement: a logic 1 will be generated whenever $x_1x_2 \geq x_3x_4$. This can be solved by means of a truth table; however, a faster approach is to use a 4-variable Karnaugh map, as shown in Figure 2.16, where x_1x_2 and x_3x_4 are two binary numbers with four values for each number — 00, 01, 10, and 11.

A value of 1 is entered in the map whenever the condition $x_1x_2 \geq x_3x_4$ is satisfied. In row x_1x_2 = 00, $x_1x_2 = x_3x_4$ in minterm 0; therefore, a 1 is entered in minterm 0 location. In row x_1x_2 = 01, $x_1x_2 \geq x_3x_4$ in minterm locations 4 and 5. In a similar manner, 1s are entered in appropriate locations for rows x_1x_2 = 11 and x_1x_2 = 10. The equation to satisfy the requirement of $x_1x_2 \geq x_3x_4$ is shown in Equation 2.10.

x_1x_2 \ x_3x_4	00	01	11	10
00	0 1	1 0	3 0	2 0
01	4 1	5 1	7 0	6 0
11	12 1	13 1	15 1	14 1
10	8 1	9 1	11 0	10 1

f

Figure 2.16 Karnaugh map to indicate when $x_1x_2 \geq x_3x_4$.

$$f = x_3'x_4' + x_2x_3' + x_1x_3' + x_1x_2 + x_1x_4' \qquad (2.10)$$

2.3.1 Map-Entered Variables

Variables may also be entered in a Karnaugh map as map-entered variables, together with 1s and 0s. A map of this type is more compact than a standard Karnaugh map, but contains the same information. A map containing map-entered variables is particularly useful in analyzing and synthesizing synchronous sequential machines. When variables are entered in a Karnaugh map, two or more squares can be combined only if the squares are adjacent and contain the same variable(s).

Example 2.27 The following Boolean equation will be minimized using a 3-variable Karnaugh map with x_4 as a map-entered variable:

$$z_1(x_1,x_2,x_3,x_4) = x_1x_2'x_3x_4' + x_1x_2 + x_1'x_2'x_3'x_4' + x_1'x_2'x_3'x_4$$

Note that instead of $2^4 = 16$ squares, the map of Figure 2.17 contains only $2^3 = 8$ squares, since only three variables are used in constructing the map. To facilitate plotting the equation in the map, the variable that is to be entered is shown in parentheses as follows:

$$z_1(x_1,x_2,x_3,x_4) = x_1x_2'x_3(x_4') + x_1x_2 + x_1'x_2'x_3'(x_4') + x_1'x_2'x_3'(x_4)$$

x_1 \ x_2x_3	0 0	0 1	1 1	1 0
0	0 $x_4' + x_4$	1 0	3 0	2 0
1	4 0	5 x_4'	7 1	6 1

z_1

Figure 2.17 Karnaugh map using x_4 as a map-entered variable for Example 2.27.

The first term in the equation for z_1 is $x_1x_2'x_3$ (x_4') and indicates that the variable x_4' is entered in minterm location 5 $(x_1x_2'x_3)$. The second term x_1x_2 is plotted in the usual manner: 1s are entered in minterm locations 6 and 7. The third term specifies that the variable x_4' is entered in minterm location 0 $(x_1'x_2'x_3')$. The fourth term also applies to minterm 0, where x_4 is entered. The expression in minterm location 0, therefore, is $x_4' + x_4$.

To obtain the minimized equation for z_1 in a sum-of-products form, 1s are combined in the usual manner; variables are combined only if the minterm locations containing the variables are adjacent and the variables are identical. Consider the expression $x_4' + x_4$ in minterm location 0. Since $x_4' + x_4 = 1$, minterm 0 equates to $x_1' x_2' x_3'$. The entry of 1 in minterm location 7 can be restated as $1 + x_4'$ without changing the value of the entry (Theorem 1). This allows minterm locations 5 and 7 to be combined as $x_1 x_3 x_4'$. Finally, minterms 6 and 7 combine to yield the term $x_1 x_2$. The minimized equation for z_1 is shown in Equation 2.11.

$$z_1 = x_1'x_2'x_3' + x_1x_3x_4' + x_1x_2 \tag{2.11}$$

Example 2.28 A Karnaugh map will be used to minimize the following Boolean function where x_2 is a map-entered variable:

$$f = x_1x_2' + x_1x_2'x_3'x_4 + x_3x_4 + x_2x_3'x_4 + x_1x_2x_3x_4$$

The equation is rewritten as Equation 2.12 with variable x_2 in parentheses for ease of use. The map will have $2^3 = 8$ squares to cover variables x_1, x_3, and x_4 as shown in Figure 2.18.

$$f = x_1(x_2') + x_1(x_2')x_3'x_4 + x_3x_4 + (x_2)x_3'x_4 + x_1(x_2)x_3x_4 \tag{2.12}$$

x_1 \ x_3x_4	0 0	0 1	1 1	1 0
0	0 0	1 x_2	3 1	2 0
1	4 x_2'	5 $x_2' + x_2' + x_2$	7 $x_2' + 1 + x_2$	6 x_2'

f

Figure 2.18 Karnaugh map using variable x_2 as a map-entered variable.

The first term in Equation 2.12 specifies that the entire row of x_1 contains the variable x_2'. The second term inserts x_2' in minterm 5 location. The third term does not contain a version of x_2; therefore, 1s are entered in minterm locations 3 and 7. The fourth term places x_2 in minterm locations 1 and 5. The fifth term causes x_2 to be inserted in minterm location 7.

The equation, obtained from the map, is shown in Equation 2.13. The first term in Equation 2.13 is obtained from row x_1 in the Karnaugh map in which each square contains x_2'. Minterm location 5 contains $x_2' + x_2' + x_2 = 1$; minterm location 7 contains $x_2' + 1 + x_2 = 1$. Minterm location 3 contains an entry of 1, which can be restated as $1 + x_2 = 1$. Therefore, the second term in Equation 2.13 combines the x_2 variable in minterm locations 1, 3, 5, and 7 with common variable x_4 — the x_3 variable is not a contributing factor to the x_2x_4 term due to the distributive law and the complementation law. The third term combines the 1 entries in minterm locations 3 and 7.

$$f = x_1x_2' + x_2x_4 + x_3x_4 \tag{2.13}$$

Example 2.29 The following Boolean equation will be minimized using x_4 and x_5 as map-entered variables:

$$\begin{aligned} z_1 = {} & x_1'x_2'x_3'(x_4x_5') + x_1'x_2 + x_1'x_2'x_3'(x_4x_5) + x_1x_2'x_3'(x_4x_5) \\ & + x_1x_2'x_3 + x_1x_2'x_3'(x_4') + x_1x_2'x_3'(x_5') \end{aligned}$$

Figure 2.19 shows the map entries for Example 2.29. The expression $x_4x_5' + x_4x_5$ in minterm location 0 reduces to x_4; the 1 entry in minterm location 2 can be expanded to $1 + x_4$ without changing the value in location 2. Therefore, locations 0 and 2 combine as $x_1'x_3'x_4$. The expression $x_4x_5 + x_4' + x_5'$ in minterm location 4 reduces to 1. Thus, the 1 entries in the map combine in the usual manner to yield Equation 2.14.

$x_1 \backslash x_2x_3$	0 0	0 1	1 1	1 0
0	0 $x_4x_5' + x_4x_5$	1 0	3 1	2 1
1	4 $x_4x_5 + x_4' + x_5'$	5 1	7 0	6 0

z_1

Figure 2.19 Karnaugh map for Example 2.29 using x_4 and x_5 as map-entered variables.

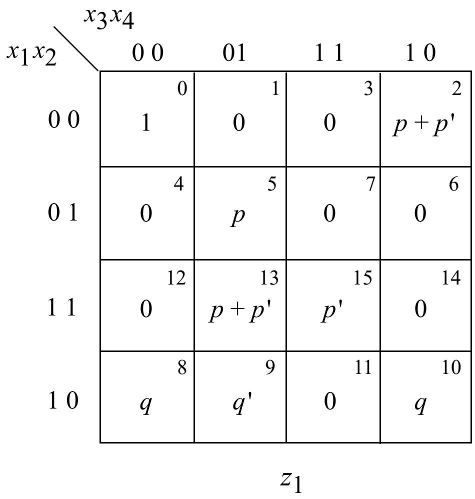
$$z_1 = x_1'x_3'x_4 + x_1'x_2 + x_1x_2' \tag{2.14}$$

Example 2.30 The Karnaugh map of Figure 2.20 contains two map-entered variables p and q. The entry in minterm locations 2 and 13 is $p + p' = 1$. Since minterm location 2 is equivalent to 1, then that entry can be changed to $p + p' + q = 1$ without changing the value of the location (Theorem 1). Likewise, minterm location 13 can be changed to $p + p' + q' = 1$. These changes will be used to minimize the function. Minterm locations 0, 2, 8, and 10 combine to yield $x_2'x_4'q$ because every location can contain the variable q.

Minterm location 0 can be changed to $1 + q$
Minterm location 2 can be changed to $p + p' + q$

In a similar manner, the following minterm locations combine:

0 and 2 combine to yield $x_1'x_2'x_4'$
5 and 13 combine to yield $x_2x_3'x_4p$
13 and 15 combine to yield $x_1x_2x_4p'$
13 and 9 combine to yield $x_1x_3'x_4q'$



$x_1x_2 \backslash x_3x_4$	0 0	01	1 1	1 0
0 0	0 1	1 0	3 0	2 $p+p'$
0 1	4 0	5 p	7 0	6 0
1 1	12 0	13 $p+p'$	15 p'	14 0
1 0	8 q	9 q'	11 0	10 q

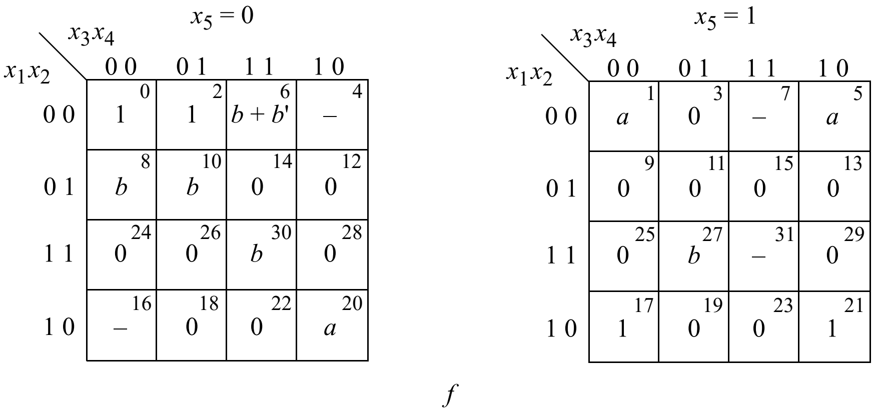
z_1

Figure 2.20 Karnaugh map for Example 2.30 using two map-entered variables.

Equation 2.15 shows the result of minimizing the Karnaugh map of Figure 2.19.

$$f = x_2'x_4'q + x_1'x_2'x_4' + x_2x_3'x_4p + x_1x_2x_4p' + x_1x_3'x_4q' \qquad (2.15)$$

Example 2.31 Given the Karnaugh map shown in Figure 2.21 for five variables where *a* and *b* are map-entered variables, the function *f* will be minimized in a sum-of-products form.



$x_5 = 0$

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0 1	2 1	6 $b + b'$	4 –
0 1	8 b	10 b	14 0	12 0
1 1	24 0	26 0	30 b	28 0
1 0	16 –	18 0	22 0	20 a

$x_5 = 1$

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	1 a	3 0	7 –	5 a
0 1	9 0	11 0	15 0	13 0
1 1	25 0	27 b	31 –	29 0
1 0	17 1	19 0	23 0	21 1

Figure 2.21 Five-variable Karnaugh map with *a* and *b* as map-entered variables.

The following minterm locations can be combined to generate the sum-of-minterm expression of Equation 2.16:

0, 2, 8, 10 combine to yield $x_1'x_3'x_5'b$;
    0 and 2 are equivalent to 1 + b
0, 2, 6, 4 combine to yield $x_1'x_2'x_5'$
30 and 31 combine to yield $x_1x_2x_3x_4b$
27 and 31 combine to yield $x_1x_2x_4x_5b$
17 and 21 combine to yield $x_1x_2'x_4'x_5$
0, 4, 16, 20, 1, 5, 17, and 21 combine to yield $x_2'x_4'a$;
    1, 17, and 21 are equivalent to 1 + a

$$f = x_1'x_3'x_5'b + x_1'x_2'x_5' + x_1x_2x_3x_4b + x_1x_2x_4x_5b + x_1x_2'x_4'x_5 + x_2'x_4'a \tag{2.16}$$

Example 2.32 As a final example of map-entered variables, consider the Karnaugh map of Figure 2.22 in which x_4 is a map-entered variable. The equation for the function *f* will be obtained in a minimum sum-of-products form. The following minterm locations combine to yield the minimized equation of Equation 2.17:

0 and 2 combine to yield $x_1'x_3'x_4$
2 yields $x_1'x_2x_3'$
5 and 7 combine to yield $x_1x_3x_4'$
4 and 5 combine to yield $x_1x_2'x_4'$
5 yields $x_1x_2'x_3$

x_1 \ x_2x_3	0 0	0 1	1 1	1 0
0	0 x_4	1 0	3 0	2 $x_4 + x_4'$
1	4 x_4'	5 $x_4 + x_4'$	7 x_4'	6 0

f

Figure 2.22 Karnaugh map for Example 2.32 using x_4 as a map-entered variable.

$$f = x_1'x_3'x_4 + x_1'x_2x_3' + x_1x_3x_4' + x_1x_2'x_4' + x_1x_2'x_3 \tag{2.17}$$

2.4 Quine-McCluskey Algorithm

If the number of variables in a function is greater than seven, then the number of squares in the Karnaugh map becomes excess, which makes the selection of adjacent squares tedious. The Quine-McCluskey algorithm is a tabular method of obtaining a minimal set of prime implicants that represents the Boolean function. Because the process is inherently algorithmic, the technique is easily implemented with a computer program. The method consists of two steps: first obtain a set of prime implicants for the function; then obtain a minimal set of prime implicants that represents the function.

The rationale for the Quine-McCluskey method relies on the repeated application of the distributive and complementation laws. For example, for a 4-variable function, minterms $x_1x_2x_3'x_4$ and $x_1x_2x_3'x_4'$ are adjacent because they differ by only one variable. The two minterms can be combined, therefore, into a single product term as follows:

$$\begin{aligned} & x_1x_2x_3'x_4 + x_1x_2x_3'x_4' \\ &= x_1x_2x_3'(x_4 + x_4') \\ &= x_1x_2x_3' \end{aligned}$$

The resulting product term is specified as $x_1x_2x_3x_4$ = 110–, where the dash (–) represents the variable that has been removed. The process repeats for all minterms in the function. Two product terms with dashes in the same position can be further combined into a single term if they differ by only one variable. Thus, the terms $x_1x_2x_3x_4$ = 110– and $x_1x_2x_3x_4$ = 100– combine to yield the term $x_1x_2x_3x_4$ = 1–0–, which corresponds to x_1x_3'.

The minterms are initially grouped according to the number of 1s in the binary representation of the minterm number. Comparison of minterms then occurs only between adjacent groups of minterms in which the number of 1s in each group differs by one. Minterms in adjacent groups that differ by only one variable can then be combined.

Example 2.33 The following function will be minimized using the Quine-McCluskey method: $f(x_1, x_2, x_3, x_4) = \Sigma_m(0, 1, 3, 6, 7, 8, 9, 14)$. The first step is to list the minterms according to the number of 1s in the binary representation of the minterm number. Table 2.6 shows the listing of the various groups. Minterms that combine cannot be prime implicants; therefore, a check (✓) symbol is placed beside each minterm that combines with another minterm. When all lists in the table have been processed, the terms that have no check marks are prime implicants.

Consider List 1 in Table 2.6. Minterm 0 differs by only one variable with each minterm in Group 1. Therefore, minterms 0 and 1 combine as 000–, as indicated in the first entry in List 2 and minterms 0 and 8 combine to yield –000, as shown in the second row of List 2. Next, compare minterms in List 1, Group 1 with those in List 1, Group 2. It is apparent that the following pairs of minterms combine because they differ by only one variable: (1,3), (1,9), and (8,9) as shown in List 2, Group 1. Minterms 1 and 3 are in adjacent groups and can combine because they differ by only one variable. The resulting term is 00–1. Minterms 1 and 6 cannot combine because they differ by more than one variable. Minterms 1 and 9 combine as –001 and minterms 8 and 9 combine to yield 100–.

Table 2.6 Minterms Listed in Groups for Example 2.33

List 1			List 2			List 3		
Group	Minterms	$x_1x_2x_3x_4$	Group	Minterms	$x_1x_2x_3x_4$	Group	Minterms	$x_1x_2x_3x_4$
0	0	0 0 0 0 ✓	0	0,1	0 0 0 – ✓	0	0,1,8,9	– 0 0 –
				0,8	– 0 0 0 ✓			
1	1	0 0 0 1 ✓	1	1,3	0 0 – 1			
	8	1 0 0 0 ✓		1,9	– 0 0 1 ✓			
				8,9	1 0 0 – ✓			
2	3	0 0 1 1 ✓	2	3,7	0 – 1 1			
	6	0 1 1 0 ✓		6,7	0 1 1 –			
	9	1 0 0 1 ✓		6,14	– 1 1 0			
3	7	0 1 1 1 ✓						
	14	1 1 1 0 ✓						

In a similar manner, minterms in the remaining groups are compared for possible adjacency. Note that those minterms that combine differ by a power of 2 in the decimal value of their minterm number. For example, minterms 6 and 14 combine as –110, because they differ by a power of 2 ($2^3 = 8$). Note also that the variable x_1 which is removed is located in column 2^3, where the binary weights of the four variables are $x_1x_2x_3x_4 = 2^3\ 2^2\ 2^1\ 2^0$.

List 3 is derived in a similar manner to that of List 2. However, only those terms that are in adjacent groups and have dashes in the same column can be compared. For example, the terms 0,1 (000–) and 8,9 (100–) both contain dashes in column x_4 and differ by only one variable. Thus, the two terms can combine into a single product term as $x_1x_2x_3x_4$ = –00– ($x_2'x_3'$). If the dashes are in different columns, then the two terms do not represent product terms of the same variables and thus, cannot combine into a single product term.

When all comparisons have been completed, some terms will not combine with any other term. These terms are indicated by the absence of a check symbol and are designated as prime implicants. For example, the term $x_1x_2x_3x_4$ = 00–1 ($x_1'x_2'x_4$) in List 2 cannot combine with any term in either the previous group or the following group. Thus, $x_1'x_2'x_4$ is a prime implicant. The following terms represent prime implicants: $x_1'x_2'x_4$, $x_1'x_3x_4$, $x_1'x_2x_3$, $x_2x_3x_4'$ and $x_2'x_3'$.

Some of the prime implicants may be redundant, since the minterms covered by a prime implicant may also be covered by one or more other prime implicants. Therefore, the second step in the algorithm is to obtain a minimal set of prime implicants that covers the function. This is accomplished by means of a *prime implicant chart* as shown in Figure 2.23 (a). Each column of the chart represents a minterm; each row of the chart represents a prime implicant. The first row of Figure 2.23 (a) is specified by the minterm grouping of (1, 3), which corresponds to prime implicant $x_1'x_2'x_4$ (00–1). Since prime implicant $x_1'x_2'x_4$ covers minterms 1 and 3, an × is placed in columns 1 and 3 in the corresponding prime implicant row. The remaining rows are completed in a similar manner. Consider the last row which corresponds to prime implicant $x_2'x_3'$. Since prime implicant $x_2'x_3'$ covers minterms 0, 1, 8, and 9, an × is placed in the minterm columns 0, 1, 8, and 9.

A single × appearing in a column indicates that only one prime implicant covers the minterm. The prime implicant, therefore, is an *essential prime implicant*. In Figure 2.23 (a), there are two essential prime implicants: $x_2x_3x_4'$ and $x_2'x_3'$. A horizontal line is drawn through all ×s in each essential prime implicant row. Since prime implicant $x_2x_3x_4'$ covers minterm 6, there is no need to have prime implicant $x_1'x_2x_3$ also cover minterm 6. Therefore, a vertical line is drawn through all ×s in column 6, as shown in Figure 2.23 (a). For the same reason, a vertical line is drawn through all ×s in column 1 for the second essential prime implicant $x_2'x_3'$.

The only remaining minterms not covered by a prime implicant are minterms 3 and 7. Minterm 3 is covered by prime implicants $x_1'x_2'x_4$ and $x_1'x_3x_4$; minterm 7 is covered by prime implicants $x_1'x_3x_4$ and $x_1'x_2x_3$, as shown in Figure 2.23 (b). Therefore, a minimal cover for minterms 3 and 7 consists of the *secondary essential prime implicant* $x_1'x_3x_4$. The complete minimal set of prime implicants for the function z_1 is shown in Equation 2.18. The minimized expression for z_1 can be verified by plotting the function on a Karnaugh map, as shown in Figure 2.24.

	Prime implicants		Minterms							
			0	1	3	6	7	8	9	14
	1,3	$(x_1'x_2'x_4)$		×	×					
	3,7	$(x_1'x_3x_4)$			×		×			
	6,7	$(x_1'x_2x_3)$				×	×			
*	6,14	$(x_2x_3x_4')$				×				×
*	0,1,8,9	$(x_2'x_3')$	⊗	×				×	×	

(a)

Prime implicants		Minterms							
		0	1	3	6	7	8	9	14
1,3	$(x_1'x_2'x_4)$			×					
3,7	$(x_1'x_3x_4)$			×		×			
6,7	$(x_1'x_2x_3)$					×			

(b)

Figure 2.23 Prime implicant chart for Example 2.33: (a) essential and nonessential prime implicants; and (b) secondary essential prime implicants with minimal cover for remaining prime implicants.

$$f(x_1,x_2,x_3,x_4) = x_2x_3x_4' + x_2'x_3' + x_1'x_3x_4 \tag{2.18}$$

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0 1	1 1	3 1	2 0
0 1	4 0	5 0	7 1	6 1
1 1	12 0	13 0	15 0	14 1
1 0	8 1	9 1	11 0	10 0

z_1

Figure 2.24 Karnaugh map for Example 2.33.

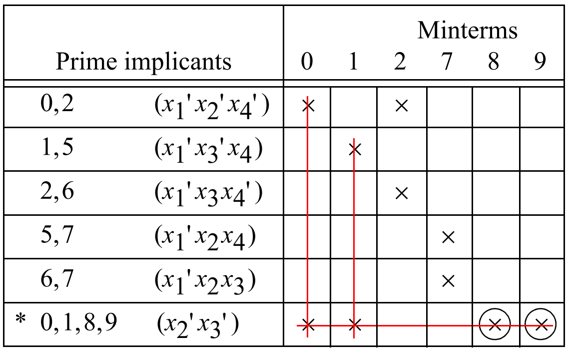
Example 2.34 Functions which include unspecified entries ("don't cares") are handled in a similar manner. The tabular representation of step 1 lists all the minterms, including "don't cares." The "don't care" conditions are then utilized when comparing minterms in adjacent groups. In step 2 of the algorithm, only the minterms containing specified entries are listed — the "don't care" minterms are not used. Then the minimal set of prime implicants is found as described in Example 2.33. The following function will be minimized using the Quine-McCluskey algorithm, as shown in Table 2.7:

$$f(x_1, x_2, x_3, x_4) = \Sigma_m(0, 1, 2, 7, 8, 9) + \Sigma_d(5, 6)$$

where minterm locations 5 and 6 are "don't cares." The prime implicant chart is shown in Figure 2.25. Notice that there are four solutions to this example, all of which are shown in Equation 2.19. The Karnaugh map of Figure 2.26 verifies the minimized sum-of-products solutions.

Table 2.7 Minterms Listed in Groups for Example 2.34

List 1			List 2			List 3		
Group	Minterms	$x_1x_2x_3x_4$	Group	Minterms	$x_1x_2x_3x_4$	Group	Minterms	$x_1x_2x_3x_4$
0	0	0 0 0 0 ✓	0	0,1	0 0 0 – ✓	0	0,1,8,9	– 0 0 –
				0,2	0 0 – 0			
				0,8	– 0 0 0 ✓			
1	1	0 0 0 1 ✓	1	1,5	0 – 0 1			
	2	0 0 1 0 ✓		1,9	– 0 0 1 ✓			
	8	1 0 0 0 ✓		2,6	0 – 1 0			
				8,9	1 0 0 – ✓			
2	5 (–)	0 1 0 1 ✓	2	5,7	0 1 – 1			
	6 (–)	0 1 1 0 ✓		6,7	0 1 1 –			
	9	1 0 0 1 ✓						
3	7	0 1 1 1 ✓						



Prime implicants		Minterms 0	1	2	7	8	9
0,2	$(x_1'x_2'x_4')$	×		×			
1,5	$(x_1'x_3'x_4)$		×				
2,6	$(x_1'x_3x_4')$			×			
5,7	$(x_1'x_2x_4)$				×		
6,7	$(x_1'x_2x_3)$				×		
* 0,1,8,9	$(x_2'x_3')$	×	×			⊗	⊗

Figure 2.25 Prime implicant chart for Example 2.34.

$$f = x_2'x_3' + x_1'x_2'x_4' + x_1'x_2x_4$$

$$f = x_2'x_3' + x_1'x_2'x_4' + x_1'x_2x_3$$

$$f = x_2'x_3' + x_1'x_3x_4' + x_1'x_2x_4$$

$$f = x_2'x_3' + x_1'x_3x_4' + x_1'x_2x_3 \tag{2.19}$$

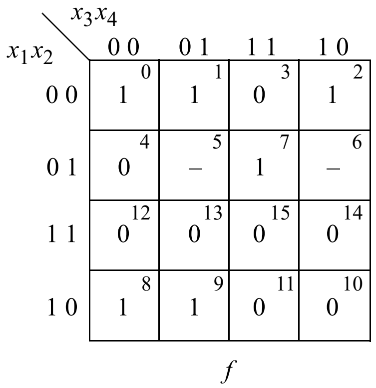


Figure 2.26 Karnaugh map for Example 2.34.

2.4.1 Petrick Algorithm

The function may not always contain an essential prime implicant, or the secondary essential prime implicants may not be intuitively obvious, as they were in previous examples. The technique for obtaining a minimal cover of secondary prime implicants is called the *Petrick algorithm* and can best be illustrated by an example.

Example 2.35 Given the prime implicant chart of Figure 2.27 for function z_1, it is obvious that there are no essential prime implicants, since no minterm column contains a single ×.

	Minterms				
Prime implicants	m_i	m_j	m_k	m_l	m m
pi_1	×	×			×
pi_2	×		×	×	
pi_3		×		×	
pi_4			×		×

Figure 2.27 Prime implicant chart for Example 2.35.

It is observed that minterm m_i is covered by prime implicants pi_1 or pi_2; m_j is covered by pi_1 or pi_3; m_k is covered by pi_2 or pi_4; m_l is covered by pi_2 or pi_3; and m_m is covered by pi_1 or pi_4. Since the function is covered only if all minterms are covered, Equation 2.20 represents this requirement.

$$\begin{aligned}\text{Function is covered} = {} & (pi_1 + pi_2)\,(pi_1 + pi_3)\,(pi_2 + pi_4) \\ & (pi_2 + pi_3)\,(pi_1 + pi_4)\end{aligned} \tag{2.20}$$

Equation 2.20 can be reduced by Boolean algebra or by a Karnaugh map to obtain a minimal set of prime implicants that represents the function. Figure 2.28 illustrates the Karnaugh map in which the sum terms of Equation 2.20 are plotted. The map is then used to obtain a minimized expression that represents the different combinations of prime implicants in which all minterms are covered. Equation 2.21 lists the product terms specified as prime implicants in a sum-of-products notation.

$pi_1 pi_2$ \ $pi_3 pi_4$	0 0	0 1	1 1	1 0
0 0	0	0	0	0
0 1	0	0	1	0
1 1	1	1	1	1
1 0	0	0	1	0

Figure 2.28 Karnaugh map in which the sum terms of Equation 2.20 are entered as 0s.

$$\text{All minterms are covered} = pi_1\,pi_2 + pi_2\,pi_3\,pi_4 + pi_1\,pi_3\,pi_4 \tag{2.21}$$

The first term of Equation 2.21 represents the fewest number of prime implicants to cover the function. Thus, function z_1 will be completely specified by the expression $z_1 = pi_1\,pi_2$. From any covering equation, the term with the fewest number of variables is chosen to provide a minimal set of prime implicants. Assume, for example, that prime implicant $pi_1 = x_i x_j' x_k$ and that $pi_2 = x_l' x_m x_n$. Thus, the sum-of-products expression is $z_1 = x_i x_j' x_k + x_l' x_m x_n$.

Example 2.36 Using the prime implicant chart of Figure 2.29 for the function *f*, the minimal sum-of-products expression will be obtained. Any "don't care" conditions are not shown in the chart. There are no essential prime implicants because there is no column with a single ×. Prime implicants are selected such that all columns (minterms) are covered by the least number of prime implicants.

	Minterms			
Prime implicants	1	2	3	4
A	×			×
B			×	×
C	×			
D		×	×	
E		×		

Figure 2.29 Prime implicant chart for the function *f* of Example 2.36.

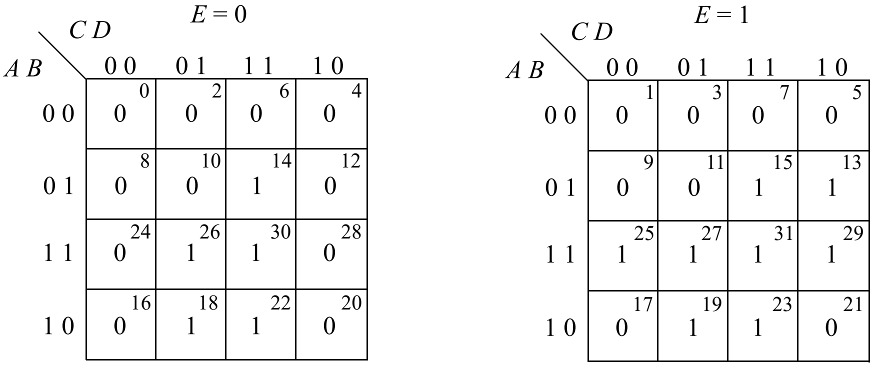
The following minterm coverage exists:

Minterm 1 is covered by prime implicants $(A + C)$
Minterm 2 is covered by prime implicants $(D + E)$
Minterm 3 is covered by prime implicants $(B + D)$
Minterm 4 is covered by prime implicants $(A + B)$

Therefore, all minterms are covered by Equation 2.22.

$$\text{All minterms are covered} = (A + C)\ (D + E)\ (B + D)\ (A + B) \tag{2.22}$$

Equation 2.22 is plotted on the Karnaugh map of Figure 2.30 as sum terms, then minimized as a sum of products. This provides a list of all covers for the function *f*, as shown in Equation 2.23. The product term with the fewest number of variables is chosen to be the minimal cover. The product term AD requires two prime implicants; all others require three prime implicants. Therefore, the minimal cover for the function *f* is the sum-of-products implementation of $A + D$. If there are two or more prime implicants with the fewest number of variables, then any one can be chosen.



$E = 0$

$AB \backslash CD$	0 0	0 1	1 1	1 0
0 0	0 0	2 0	6 0	4 0
0 1	8 0	10 0	14 1	12 0
1 1	24 0	26 1	30 1	28 0
1 0	16 0	18 1	22 1	20 0

$E = 1$

$AB \backslash CD$	0 0	0 1	1 1	1 0
0 0	1 0	3 0	7 0	5 0
0 1	9 0	11 0	15 1	13 1
1 1	25 1	27 1	31 1	29 1
1 0	17 0	19 1	23 1	21 0

Figure 2.30 Karnaugh map for the Petrick algorithm of Example 2.36.

$$f = AD + ABE + BCE + BCD \tag{2.23}$$

Example 2.37 As a final example to illustrate the application of the Petrick algorithm when using the Quine-McCluskey minimization technique, Equation 2.24 will be minimized. Table 2.8 shows the minterms partitioned into groups containing identical number of 1s.

$$f(x_1, x_2, x_3, x_4) = \Sigma_{\text{m}}\,(1, 4, 5, 6, 13, 14, 15) + \Sigma_{\text{d}}\,(8, 9) \tag{2.24}$$

Table 2.8 Minterms Listed in Groups for Example 2.37

List 1			List 2			List 3		
Group	Minterms	$x_1x_2x_3x_4$	Group	Minterms	$x_1x_2x_3x_4$	Group	Minterms	$x_1x_2x_3x_4$
1	1	0 0 0 1 ✓	1	1,5	0 – 0 1 ✓	1	1,5,9,13	– – 0 1
	4	0 1 0 0 ✓		1,9	– 0 0 1 ✓			
	8 (–)	1 0 0 0 ✓		4,5	0 1 0 –			
				4,6	0 1 – 0			
				8,9	1 0 0 –			
2	5	0 1 0 1 ✓	2	5,13	– 1 0 1 ✓			
	6	0 1 1 0 ✓		6,14	– 1 1 0			
	9 (–)	1 0 0 1 ✓		9,13	1 – 0 1 ✓			
3	13	1 1 0 1 ✓	3	13,15	1 1 – 1			
	14	1 1 1 0 ✓		14,15	1 1 1 –			
4	15	1 1 1 1 ✓						

Figure 2.31 shows the prime implicant chart for Example 2.37 in which there is only one essential prime implicant [1, 5, 9, 13 ($x_3'x_4$)] — the remaining prime implicants are secondary essential prime implicants. Since minterms 8 and 9 are "don't care" conditions, there are no columns labeled 8 and 9. Figure 2.32 shows the chart for the secondary essential prime implicants.

		Minterms						
Prime implicants		1	4	5	6	13	14	15
4,5	($x_1'x_2x_3'$)		×	×				
4,6	($x_1'x_2x_4'$)		×		×			
6,14	($x_2x_3x_4'$)				×		×	
13,15	($x_1x_2x_4$)					×		×
14,15	($x_1x_2x_3$)						×	×
* 1,5,9,13	($x_3'x_4$)	⊗		×		×		

Figure 2.31 Prime implicant chart for Example 2.37.

			Minterms			
	Prime implicants		4	6	14	15
A	4,5	($x_1'x_2x_3'$)	×			
B	4,6	($x_1'x_2x_4'$)	×	×		
C	6,14	($x_2x_3x_4'$)		×	×	
D	13,15	($x_1x_2x_4$)				×
E	14,15	($x_1x_2x_3$)			×	×

Figure 2.32 Chart for secondary essential prime implicants.

The following minterm coverage exists:

Minterm 4 is covered by prime implicants $(A + B)$
Minterm 6 is covered by prime implicants $(B + C)$
Minterm 14 is covered by prime implicants $(C + E)$
Minterm 15 is covered by prime implicants $(D + E)$

Therefore, all minterms are covered by Equation 2.25.

$$\text{All minterms are covered} = (A + B)\,(B + C)\,(C + E)\,(D + E) \tag{2.25}$$

Equation 2.25 is plotted on the Karnaugh map of Figure 2.33 as sum terms, then minimized as a sum of products. This provides a list of all covers for the function *f*, as shown in Equation 2.26. The product term with the fewest number of variables is chosen to be the minimal cover. The product term *BE* requires two prime implicants; all others require three prime implicants. Therefore, the minimal cover for the function *f* is the sum-of-products implementation of $B + E$ plus the essential prime implicant, as shown in Equation 2.27.

$E = 0$

AB \ *CD*	0 0	0 1	1 1	1 0
0 0	0 0	2 0	6 0	4 0
0 1	8 0	10 0	14 1	12 0
1 1	24 0	26 0	30 1	28 0
1 0	16 0	18 0	22 1	20 0

$E = 1$

AB \ *CD*	0 0	0 1	1 1	1 0
0 0	1 0	3 0	7 0	5 0
0 1	9 1	11 1	15 1	13 1
1 1	25 1	27 1	31 1	29 1
1 0	17 0	19 0	23 1	21 1

Figure 2.33 Karnaugh map for the Petrick algorithm of Example 2.37.

$$f = BE + ACE + ACD + BCD \tag{2.26}$$

$$f = x_3'x_4 + x_1'x_2x_4' + x_1x_2x_3 \tag{2.27}$$

2.5 Problems

2.1 Minimize the following function using Boolean algebra:

$$f = x_1'x_3'x_1 + x_1x_2x_3x_3' + x_1x_2x_3x_3'$$

2.2 Write the dual for the following statement:

$$z_1 = x_1x_2' + x_1'x_2$$

2.3 Write a Boolean equation using three terms for $f(x_1, x_2, x_3, x_4)$ in

(a) Disjunctive normal form.
(b) Canonical sum-of-products form.

2.4 Indicate which of the equations shown below will always generate a logic 1.

(a) $z_1 = x_1 + x_2x_3x_4' + x_1'x_3 + x_3'$
(b) $z_1 = x_1x_2 + x_1x_2' + x_2'x_4' + x_1'x_2x_4$
(c) $z_1 = x_1x_2'x_4 + x_1'x_2x_3' + x_1'x_2x_4 + x_1'x_2'x_3 + x_1'x_2x_4'$
(d) $z_1 = x_4' + x_1'x_4 + x_1x_4 + x_2'x_4'$

2.5 Minimize the following Boolean expression:

$$f = x_1'x_2(x_3'x_4' + x_3'x_4) + x_1x_2(x_3'x_4' + x_3'x_4) + x_1x_2'x_3'x_4$$

2.6 Indicate whether the following statement is true or false:

$$x_1'x_2'x_3' + x_1x_2x_3' = x_3'$$

2.7 Minimize the following equation using Boolean algebra:

$$z_1 = x_1'x_3'x_4' + x_1'x_3x_4' + x_1x_3'x_4' + x_2x_3x_4 + x_1x_3x_4'$$

2.8 Obtain the canonical product-of-sums form for the following function using Boolean algebra:

$$f(x_1, x_2, x_3, x_4) = x_3$$

2.9 Minimize the following equation to obtain a sum-of-products expression using Boolean algebra:

$$f = (x_1x_2' + x_1'x_2)x_3' + (x_1x_2' + x_1'x_2)'x_3 + x_1'x_3 + x_2'x_3$$

2.10 Determine if the following Boolean equation is valid using the axioms and theorems of Boolean algebra:

$$x_1x_2 = (x_1 + x_3')(x_1' + x_2')(x_1' + x_2)$$

2.11 Prove that $x_1 + 1 = 1$.

2.12 Prove that $x_1'' = x_1$.

2.13 Use DeMorgan's theorem to minimize the following Boolean expression: $(x_1 + x_2' + x_3 + x_4')' + (x_1x_2x_3x_4')'$

2.14 Given the equation shown below, use x_4 as a map-entered variable in a Karnaugh map and obtain the minimum sum-of-products expression. Then use the original equation using x_2 as a map-entered variable and compare the results. If possible, further minimize both answers using Boolean algebra.

$$f = x_1'x_3x_4 + x_1x_2x_3 + x_1x_2'x_3x_4 + x_1x_2'x_3x_4'$$

2.15 Given the Karnaugh map shown below, obtain the minimum sum-of-products expression and the minimum product-of-sums expression for the function f.

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0	0	1	0
0 1	0	1	1	0
1 1	0	1	1	0
1 0	1	1	1	1

f

2.16 Plot the following Boolean expression on a Karnaugh map:

$$f = x_1'x_2(x_3'x_4' + x_3'x_4) + x_1x_2(x_3'x_4' + x_3'x_4) + x_1x_2'x_3'x_4$$

2.17 Obtain the minimized sum-of-products expression for the function z_1 represented by the Karnaugh map shown below.

$x_5 = 0$

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0 0	2 1	6 1	4 0
0 1	8 1	10 1	14 1	12 1
1 1	24 0	26 1	30 0	28 0
1 0	16 0	18 1	22 1	20 1

$x_5 = 1$

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	1 0	3 1	7 1	5 0
0 1	9 1	11 1	15 1	13 1
1 1	25 1	27 1	31 0	29 0
1 0	17 1	19 1	23 1	21 0

z_1

2.18 Obtain the minimized expression for the function *f* in a sum-of-products form from the Karnaugh map shown below.

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	z	0	–	1
0 1	y	–	1	0
1 1	1	–	1	0
1 0	1	–	0	z

z_1

2.19 Write the equation that will generate a logic 1 whenever a 4-bit unsigned binary number $z_1 = x_1x_2x_3x_4$ is greater than six. The equation is to be in a minimum sum-of-products form.

2.20 Given the following Karnaugh map, obtain the minimized expression for z_1 in a sum-of-products form and a product-of-sums form.

x_1x_2 \ x_3x_4	0 0	0 1	1 1	1 0
0 0	0 0	1 0	3 1	2 0
0 1	4 0	5 1	7 0	6 0
1 1	12 0	13 1	15 0	14 0
1 0	8 0	9 1	11 1	10 0

z_1

2.21 Given the Karnaugh map shown below, obtain the minimized expression for z_1 in a sum-of-products form.

x_1 \ x_2x_3	0 0	0 1	1 1	1 0
0	0 $A + B'$	1 –	3 0	2 A
1	4 B'	5 1	7 $AB + A' + B'$	6 0

z_1

2.22 Minimize the following expression using a Karnaugh map with x_3 and x_4 as map-entered variables:

$$z_1 = x_1'x_2x_3'x_4' + x_1'x_2x_3'x_4 + x_1'x_2x_3x_4' + x_1x_2'x_3x_4 + x_1x_2x_3'x_4$$

2.23 Plot the following expression on a Karnaugh map using x_4 and x_5 as map-entered variables, then obtain the minimized sum of products:

$$f(x_1, x_2, x_3, x_4, x_5) = x_1'x_2x_3'x_4x_5 + x_1x_2x_3' + x_1x_2'x_3'x_4x_5'$$

2.24 Plot the following function on a Karnaugh map, then obtain the minimum sum-of-products expression and the minimum product-of-sums expression:

$$f(x_1, x_2, x_3, x_4) = \Pi_{\mathrm{M}}\ (0, 1, 2, 8, 9, 12)$$

2.25 Plot the following function on a Karnaugh map, then obtain the minimum sum-of-products expression and the minimum product-of-sums expression:

$$f(x_1, x_2, x_3, x_4) = \Sigma_{\mathrm{m}}\ (0, 1, 2, 3, 5, 7, 10, 12, 15)$$

2.26 Indicate whether the following equation is true or false:

$$x_1'x_2 + x_1'x_2'x_3x_4' + x_1x_2x_3'x_4 = x_1'x_2x_3' + x_1'x_3x_4' + x_1'x_2x_3 + x_2x_3'x_4$$

2.27 Plot the following Boolean expression on a Karnaugh map, then obtain the minimum product of sums:

$$f(x_1, x_2, x_3, x_4, x_5) = x_1x_3x_4(x_2 + x_4x_5') + (x_2' + x_4)\ (x_1x_3' + x_5)$$

2.28 Plot the following equation on a Karnaugh map, then obtain the minimum sum of products:

$$f\ = \{[x_1' + (x_1x_2)''] \ [x_2' + (x_1x_2)'']\}'$$

2.29 Obtain the minimum sum-of-products equation that will generate a logic 1 whenever the binary number N shown below satisfies the following criteria:

$$N = x_1 x_2 x_3 x_4 \text{, where } x_4 \text{ is the low-order bit}$$

$$2 < N \leq 6 \text{ and } 11 \leq N < 14$$

2.30 Obtain the minimum sum-of-products expression for the Quine-McCluskey prime implicant table shown below, where $f(x_1, x_2, x_3, x_4, x_5)$.

	Minterms									
Prime implicants	0	1	3	7	15	16	18	19	23	31
0 0 0 0 – $(x_1'x_2'x_3'x_4')$	×	×								
0 0 0 – 1 $(x_1'x_2'x_3'x_5)$		×	×							
– 0 – 1 1 $(x_2'x_4x_5)$			×	×				×	×	
– – 1 1 1 $(x_3x_4x_5)$				×	×				×	×
1 0 0 1 – $(x_1x_2'x_3'x_4)$							×	×		
1 0 0 – 0 $(x_1x_2'x_3'x_5')$						×	×			
– 0 0 0 0 $(x_2'x_3'x_4'x_5')$	×					×				

2.31 Minimize the following equation using the Quine-McCluskey algorithm, then verify the result by a Karnaugh map:

$$f(x_1, x_2, x_3) = \Sigma_{\mathrm{m}} (1, 2, 5, 6) + \Sigma_{\mathrm{d}} (0, 3)$$

2.32 Use the Quine-McCluskey algorithm to obtain the minimum sum-of-products form for the following equation:

$$f(x_1, x_2, x_3, x_4) = \Sigma_{\mathrm{m}} (0, 1, 3, 7, 9, 11, 14, 15)$$

2.33 Obtain the minimum sum-of-products expression for the Quine-McCluskey prime implicant table shown below, where $f(x_1, x_2, x_3, x_4, x_5)$.

	Minterms									
Prime implicants	0	1	3	7	15	16	18	19	23	31
0,1 $(x_1'x_2'x_3'x_4')$	×	×								
1,3 $(x_1'x_2'x_3'x_5)$		×	×							
3,7,19,23 $(x_2'x_4x_5)$			×	×				×	×	
7,15,23,31 $(x_3x_4x_5)$				×	×				×	×
18,19 $(x_1x_2'x_3'x_4)$							×	×		
16,18 $(x_1x_2'x_3'x_5')$						×	×			
0,16 $(x_2'x_3'x_4'x_5')$	×					×				

2.34 Using any method, list all of the prime implicants (both essential and nonessential) for the following expression:

$$z_1(x_1, x_2, x_3, x_4) = \Sigma_{\mathrm{m}}\,(0, 2, 3, 5, 7, 8, 10, 11, 13, 15)$$

2.35 Given the prime implicant chart shown below, use the Petrick algorithm to find a minimal set of prime implicants for the function.

	Minterms						
Prime implicants	m1	m2	m3	m4	m5	m6	m7
A	×	×	×			×	×
B	×		×	×	×		
C		×				×	
D				×	×		×

2.36 Use the Quine-McCluskey algorithm to find the minimum sum-of-products expression for the following function, then verify the result by means of a Karnaugh map:

$$f(x_1, x_2, x_3, x_4) = \Sigma_{\mathrm{m}}\,(0, 1, 7, 8, 10, 12, 14, 15) + \Sigma_{\mathrm{d}}\,(2, 5)$$

2.37 Use the Quine-McCluskey algorithm and the Petrick algorithm to obtain the minimum sum-of-products expression for the function shown below. Verify the result by means of a Karnaugh map.

$$f(x_1, x_2, x_3, x_4) = \Sigma_{\text{m}}\ (3, 4, 6, 7, 11, 12, 13, 15)$$