## 5.3 KARNAUGH MAP REPRESENTATION OF LOGIC FUNCTIONS

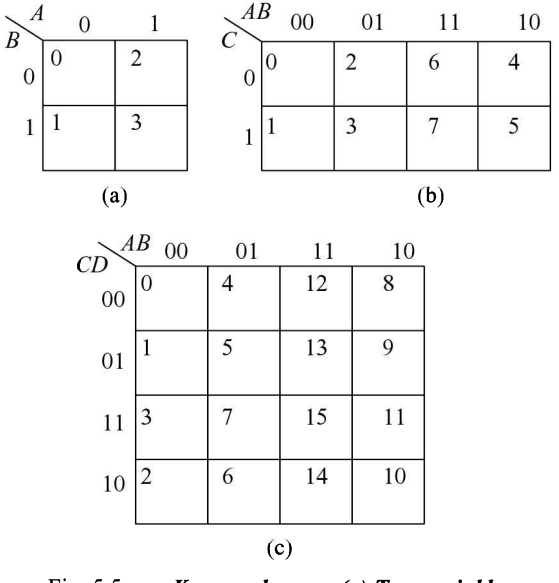


Fig. 5.5 ***Karnaugh-maps (a) Two-variable (b) Three-variable (c) Four-variable***

We have discussed the two standard forms of logic functions and their realisations using gates. We have also established the need for the simplification of the Boolean expressions and introduced the algebraic method of simplification using Boolean algebraic theorems. Sometimes it is difficult to be sure that a logic expression can be simplified. There is another technique, which is graphical, known as the Karnaugh map technique which provides a systematic method for simplifying and manipulating Boolean expressions. In this technique, the information contained in a truth table or available in POS or SOP form is represented on Karnaugh map (K-map). This is perhaps the most extensively used tool for simplification of Boolean functions. Although the technique may be used for any number of variables, it is generally used up to six variables beyond which it becomes very cumbersome.

Figure 5.5 shows the K-maps for two, three, and four variables. In an $n$-variable K-map there are $2^n$ cells. Each cell corresponds to one of the combinations of $n$ variables, since there are $2^n$ combinations of $n$ variables. Therefore, we see that for each

row of the truth table, for each minterm and for each maxterm there is one specific cell in the K-map. The variables have been designated as $A$, $B$, $C$, and $D$, and the binary numbers formed by them are taken as $AB$, $ABC$, and $ABCD$ for two, three, and four variables respectively. In each map the variables and all possible values of the variables are indicated (the first bit corresponds to the first variable and the second bit corresponds to the second variable) to identify the cells. Gray code has been used for the identification of cells. The reason for using Gray code will become clear when we discuss the application of K-map. You can verify the decimal number corresponding to each cell which is written in the top left corner of the cell as shown in Fig. 5.5.

Figure 5.6 shows the minterm/maxterm corresponding to each cell and the term is written inside the cell for clear understanding.

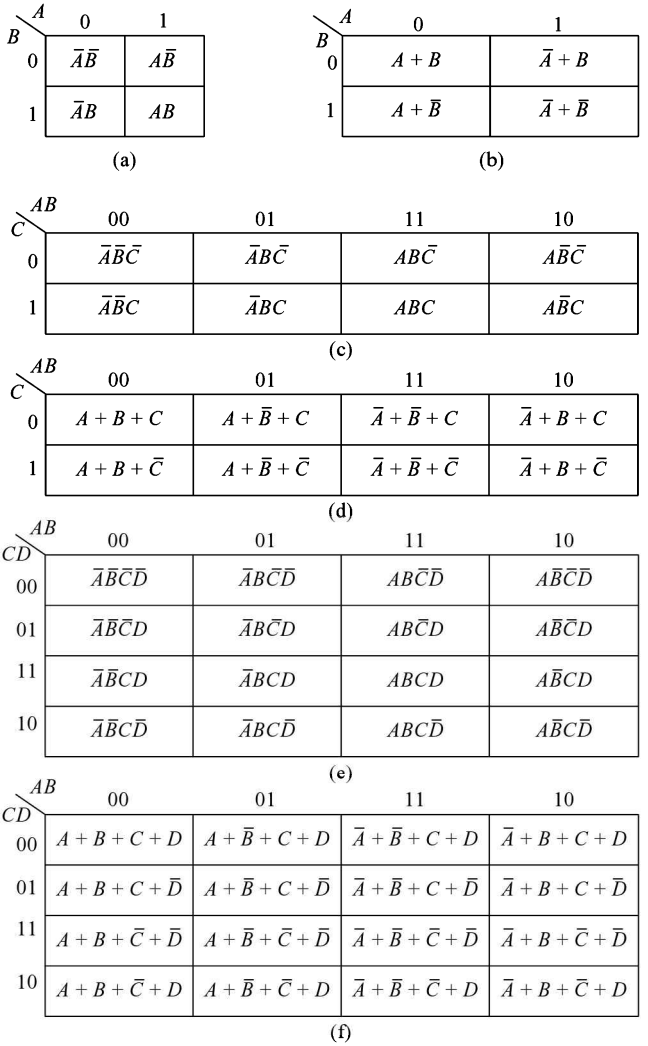


Fig. 5.6 ***Maxterm/Minterm Corresponding to Each Cell of K-maps***

### 5.3.1 Representation of Truth Table on K-Map

Consider the truth table of the 3-variable logic function given in Table 5.3. The output $Y$ is logic 1 corresponding to the rows 1, 2, 4, and 7. Corresponding to this we can write the equation in terms of canonical SOP as given below:

$$Y = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC \tag{5.14}$$

Equation (5.14) represents the complete truth table in canonical SOP form. Similarly, we note that the output $Y$ is logic 0 corresponding to the rows 0, 3, 5, and 6 and the output $Y$ can be represented in terms of canonical POS form as given below:

$$Y = (A + B + C)(A + \bar{B} + \bar{C})(\bar{A} + B + \bar{C})(\bar{A} + \bar{B} + C) \tag{5.15}$$

Equation (5.15) also represents the complete truth table, and Eqs (5.14) and (5.15) are equivalent. We shall make use of the 3-variable K-map of Fig. 5.5*b* and enter the value of the output variable $Y$ (0 or 1) in each cell corresponding to its decimal or minterm or maxterm identification. Figure 5.7 gives the complete K-map of the truth table given in Table 5.3.

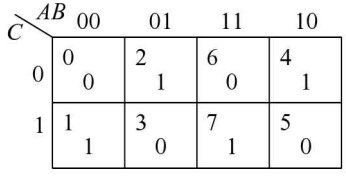


Fig. 5.7 ***K-map for Table 5.3***

The procedure used above is general and is used to represent a truth table on the K-map. On the other hand, if a K-map is given we can make the truth table corresponding to this by following the reverse process. That is, the output $Y$ is logic 1 corresponding to the decimal numbers/minterms represented by cells with entries 1. In all other rows, the output $Y$ is logic 0.

Table 5.3 ***Truth Table of a 3-variable Function***

| Row No. | Inputs | | | Output |
|---|---|---|---|---|
| | *A* | *B* | *C* | *Y* |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 |

**Example 5.4**

Prepare the truth table for K-map of Fig. 5.8.

***Solution***

The truth table is given in Table 5.4.

Note that the 0's are not written in the K-map of Fig. 5.8. Actually, we need enter either 0's or 1's only in the K-map. If only 1's are entered the empty cells are 0's and if only 0's are entered then the empty cells are 1's.

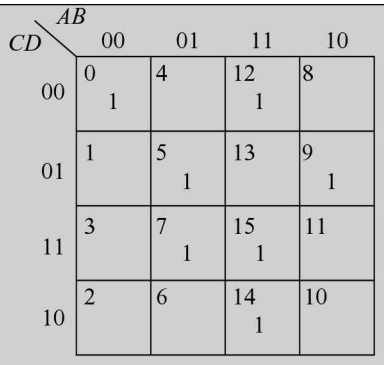


Fig. 5.8 ***K-map of Ex. 5.4***

Table 5.4 ***Truth Table for K-map of Fig. 5.8***

| | Inputs | | | | Output |
|---|---|---|---|---|---|
| **Row No.** | ***A*** | ***B*** | ***C*** | ***D*** | ***Y*** |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 1 |
| 13 | 1 | 1 | 0 | 1 | 0 |
| 14 | 1 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 1 |

## 5.3.2 Representation of Canonical SOP Form on K-Map

A logical equation in canonical SOP form can be represented on a K-map by simply entering 1's in the cells of the K-map corresponding to each minterm present in the equation.

**Example 5.5**

Represent Eq. (5.14) on K-map.

***Solution***

Corresponding to each minterm in the equation, there is a cell in the K-map and a 1 is entered in each one of these cells. The K-map will be as shown in Fig. 5.7.

Similarly, from the K-map, we can write the corresponding logic equation in canonical SOP form by ORing the minterms corresponding to each 1 entry in the K-map.

**Example 5.6**

Write the logic equation in the canonical SOP form for the K-map of Fig. 5.8.

***Solution***

$$\begin{aligned} Y &= \overline{A}\overline{B}\overline{C}\overline{D} + AB\overline{C}\overline{D} + \overline{A}B\overline{C}D + A\overline{B}\overline{C}D + \overline{A}BCD + ABCD + ABC\overline{D} \\ &= \Sigma m(0, 5, 7, 9, 12, 14, 15) \end{aligned} \tag{5.16}$$

If the equation is in SOP form, it can be converted to canonical SOP form by using the method discussed earlier and then it can be represented on K-map. Another method of representing SOP form on K-map without converting it to canonical SOP form will become clear from the discussions of Sections 5.4 and 5.6.

### 5.3.3 Representation of Canonical POS Form on K-Map

Logic equation in canonical POS form can be represented on K-map by entering 0's in the cells of K-map corresponding to each maxterm present in the equation.

**Example 5.7**

Represent Eq. (5.15) on K-map.

***Solution***

Corresponding to each maxterm in the equation, there is a cell in the K-map and a 0 is entered in each one of these cells. The K-map will be as shown in Fig. 5.7.

From a given K-map, we can write the logic equation in the canonical POS form by ANDing the maxterms corresponding to each 0 entry in the K-map.

**Example 5.8**

Write the logic equation in the canonical POS form for the K-map of Fig. 5.8.

***Solution***

$$\begin{aligned} Y = &(A+B+C+\bar{D})(A+B+\bar{C}+D)\,(A+B+\bar{C}+\bar{D}) \\ &(A+\bar{B}+C+D)(A+\bar{B}+\bar{C}+D)\,(\bar{A}+B+C+D) \\ &(\bar{A}+B+\bar{C}+D)(\bar{A}+B+\bar{C}+\bar{D})(\bar{A}+\bar{B}+C+\bar{D}) \\ &= \Pi M\,(1, 2, 3, 4, 6, 8, 10, 11, 13) \end{aligned} \tag{5.17}$$

If the equation is in POS form, it can be converted into canonical POS form by the method discussed earlier and then it can be represented on K-map. Another method of representing POS form on K-map without converting it to canonical POS form will become clear from the discussion of Sections 5.4 and 5.6.

Equation (5.16) represents the K-map of Fig. 5.8 in canonical SOP form and Eq. (5.17) represents the same K-map in standard POS form. Therefore, these two equations are equivalent. Alternatively stated, an equation in SOP form can be converted into an equivalent POS form and vice-versa.

## 5.4 SIMPLIFICATION OF LOGIC FUNCTIONS USING K-MAP

Simplification of logic functions with K-map is based on the principle of combining terms in adjacent cells. Two cells are said to be adjacent if they differ in only one variable. For example, in the two-variable K-maps of Figs 5.6*a* and *b*, the top two cells are adjacent and the bottom two cells are adjacent. Also, the left two cells and the right two cells are adjacent. It can be verified that in adjacent cells one of the literals is same, whereas the other literal appears in uncomplemented form in one and in the complemented form in the other cell.

Similarly, we observe adjacent cells in the 3-variable and 4-variable K-maps. Table 5.5 gives the adjacent cells of each cell in 2-, 3-, and 4-variable K-maps. From this it becomes clear that if the Gray code is used for the identification of cells in K-map, physically adjacent (horizontal and vertical but not diagonal) cells differ in only one variable. Also, the left-most cells are adjacent to their corresponding right-most cells and similarly the top cells are adjacent to their corresponding bottom cells. The simplification of logical function is achieved by grouping adjacent 1's or 0's in groups of $2^i$, where $i$ = 1, 2,…$n$ and $n$ is the number of variables.

The process of simplification involves grouping of minterms and identifying *prime-implicants* (PI) and *essential prime-implicants* (EPI).

A *prime-implicant* is a group of minterms that cannot be combined with any other minterm or groups. An *essential prime-implicant* is a *prime-implicant* in which one or more minterms are unique; i.e. it contains at least one minterm which is not contained in any other prime-implicant.

### 5.4.1 Grouping Two Adjacent Ones

If there are two adjacent ones on the map, these can be grouped together and the resulting term will have one less literal than the original two terms. It can be verified for each of the groupings of two ones as given in Table 5.5.

Table 5.5 ***Adjacent Cells in K-maps***

| Cell with decimal number | Decimal numbers of adjacent cells | | |
|---|---|---|---|
| | 2-variable | 3-variable | 4-variable |
| 0 | 1, 2 | 1, 2, 4 | 1, 2, 4, 8 |
| 1 | 0, 3 | 0, 3, 5 | 0, 3, 5, 9 |
| 2 | 0, 3 | 0, 3, 6 | 0, 3, 6, 10 |
| 3 | 1, 2 | 1, 2, 7 | 1, 2, 7, 11 |
| 4 | | 0, 5,6 | 0, 5, 6, 12 |
| 5 | | 1, 4, 7 | 1, 4, 7, 13 |
| 6 | | 2, 4, 7 | 2, 4, 7, 14 |
| 7 | | 3, 5, 6 | 3, 5, 6, 15 |
| 8 | | | 0, 9, 10,12 |
| 9 | | | 1, 8, 11,13 |
| 10 | | | 2, 8, 11,14 |
| 11 | | | 3, 9, 10,15 |
| 12 | | | 4, 8, 13,14 |
| 13 | | | 5, 9, 12,15 |
| 14 | | | 6,10, 12,15 |
| 15 | | | 7,11, 13,14 |

## Example 5.9

Simplify the K-map of Fig. 5.9.

### *Solution*

The canonical SOP form of equation can be written by inspection as

$$Y = \bar{A}\bar{B}\bar{C} + \bar{A}BC + ABC + A\bar{B}\bar{C} \tag{5.18}$$

If we combine the ones in adjacent cells (0, 4) and (3, 7), Eq. (5.18) can be written as

$$Y = (\bar{A} + A)\bar{B}\bar{C} + (\bar{A} + A)BC \tag{5.19}$$

$$= \bar{B}\bar{C} + BC \text{ (Theorems 1.7 and 1.2)} \tag{5.20}$$

Equation (5.20) can be directly obtained from the K-map by using the following procedure:

1. Identify adjacent ones, then see the values of the variables associated with these cells. Only one variable will be different and it gets eliminated. Other variables will appear in ANDed form in the term, it will be in the uncomplemented form if it is 1 and in the complemented form if it is 0.
2. Determine the term corresponding to each group of adjacent ones. These terms are ORed to get the simplified equation in SOP form.

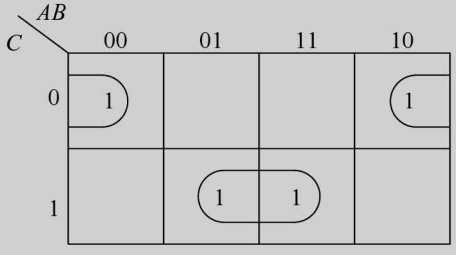


Fig. 5.9 ***K-map of Ex. 5.9***

Here, $\overline{B}\overline{C}$ and $BC$ are the two prime-implicants, since grouping of the two minterms $m_0$ and $m_4$ cannot be combined with anyone of the remaining minterms $m_3$ and $m_7$, or the combination of the minterms $m_3$ and $m_7$. Similarly, the grouping of $m_3$ and $m_7$ is a prime-implicant. We also observe that both the groups are also essential prime-implicants, since each one of them contains unique minterms which are not contained in the other group.

## 5.4.2 Grouping Four Adjacent Ones

Four cells form a group of four adjacent ones if two of the literals associated with the minterms/maxterms are not same and the other literals are same. Table 5.6 gives all possible groups of four adjacent ones for each cell in a 3-variable map. In case of 2-variable map, there is only one possibility corresponding to entry 1 in all the four cells, and the simplified expression will be $Y = 1$. That is, $Y$ always equals 1 (independent of the variables).

On the basis of groupings of 4 adjacent ones given in Table 5.6, we can find the groupings in K-maps of four or more variables. In the case of a four-variable K-map, there are six possible groupings of 4-variables involving any cell. It is left to the reader to verify this fact.

Table 5.6 ***Groups of Four Adjacent Ones in a 3-variable K-map***

| Cell with decimal number | Decimal numbers of cells forming groups of adjacent fours | | |
|---|---|---|---|
| 0 | (0, 2, 6, 4), | (0, 1, 2, 3), | (0, 1, 4, 5) |
| 1 | (1, 0, 2, 3), | (1, 3, 7, 5), | (1, 0, 4, 5) |
| 2 | (2, 0, 6, 4), | (2, 3, 1, 0), | (2, 3, 6, 7) |
| 3 | (3, 1, 7, 5), | (3, 2, 1, 0), | (3, 2, 6, 7) |
| 4 | (4, 6, 2, 0), | (4, 5, 6, 7), | (4, 5, 0, 1) |
| 5 | (5, 1, 3, 7), | (5, 4, 6, 7), | (5, 4, 0, 1) |
| 6 | (6, 0, 2, 4), | (6, 7, 4, 5), | (6, 7, 2, 3) |
| 7 | (7, 1, 3, 5), | (7, 6, 4, 5), | (7, 6, 2, 3) |

## Example 5.10

Simplify the K-map of Fig. 5.10.

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 1 | 4 | 12 | 8 1 |
| 01 | 1 1 | 5 | 13 | 9 1 |
| 11 | 3 1 | 7 1 | 15 1 | 11 1 |
| 10 | 2 | 6 | 14 | 10 |

Fig. 5.10 ***K-map of Ex. 5.10***

### *Solution*

The canonical SOP form of equation can be written by inspection as

$$\begin{aligned} Y &= m_0 + m_1 + m_3 + m_7 + m_8 + m_9 + m_{11} + m_{15} \\ &= (m_0 + m_1 + m_8 + m_9) + (m_3 + m_7 + m_{15} + m_{11}) \end{aligned} \tag{5.21}$$

In the K-map of Fig. 5.10, there are two groups of four adjacent ones. One corresponding to cells 0, 1, 8, and 9, and the other one corresponding to 3, 7, 15, and 11. In Eq. (5.21), the minterms corresponding to each group are

combined. The first term can be written as

$$\begin{aligned} m_0 + m_1 + m_8 + m_9 &= \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}D \\ &= \overline{B}\,\overline{C}(\overline{A}\,\overline{D} + \overline{A}D + A\overline{D} + AD) \\ &= \overline{B}\,\overline{C}\left[\overline{A}(\overline{D} + D) + A(\overline{D} + D)\right] \\ &= \overline{B}\,\overline{C}\left[\overline{A}\cdot 1 + A\cdot 1\right] \\ &= \overline{B}\,\overline{C}(\overline{A} + A) \\ &= \overline{B}\,\overline{C}\cdot 1 = \overline{B}\,\overline{C} \end{aligned}$$

In the first term of Eq. (5.21) we observe the following:

1. In this group of four minterms, two of the variables appear as $\overline{B}$ and $\overline{C}$ in all the four terms.
2. The variable $A$ appears as $A$ in two and as $\overline{A}$ in the other two minterms.
3. The variable $D$ appears as $D$ in two and as $\overline{D}$ in the other two minterms.
4. The combination of these four minterms results in one term with two literals which are present in all the four terms. Similarly, the second term of Eq. (5.21) is simplified to $CD$. Therefore, the K-map is simplified to

$$Y = \overline{B}\,\overline{C} + CD \tag{5.22}$$

## Example 5.11

In Fig. 5.10, show that the following groups of minterms are not prime-implicants:

(a) $m_0, m_1$ (d) $m_7, m_{15}$ (g) $m_9, m_{11}$ (i) $m_1, m_9$
(b) $m_1, m_3$ (e) $m_{11}, m_{15}$ (h) $m_0, m_8$ (j) $m_3, m_{11}$
(c) $m_3, m_7$ (f) $m_8, m_9$

### *Solution*

(a) $m_0, m_1$ group can be combined with $m_8, m_9$ group to form a group of four adjacent 1s. Hence, $m_0, m_1$ group is not a prime-implicant.
(b) $m_1, m_3$ group can be combined with $m_9, m_{11}$ group, therefore, it is not a prime-implicant.
(c) $m_3, m_7$ group can be combined with $m_{15}, m_{11}$ group.
(d) $m_7, m_{15}$ group can be combined with $m_3, m_{11}$ group.
(e) $m_{11}, m_{15}$ group can be combined with $m_3, m_7$ group.
(f) See (a)
(g) See (b)
(h) $m_0, m_8$ group can be combined with $m_1, m_9$ group
(i) See (h)
(j) See (d)

## Example 5.12

Show that the following groups of minterms in Fig. 5.10 are essential prime-implicants:

(a) $m$ (0, 1, 8, 9) (b) $m$ (3, 7, 11, 15)

*Solution*

(a) The group formed by the minterms $m_0$, $m_1$, $m_8$, $m_9$ is a prime-implicant, since it can not be combined with any other minterm or group. Also, it includes minterms which can not be included in the other group, therefore, it is an essential prime-implicant.

(b) The group formed by the minterms $m_3$, $m_7$, $m_{11}$, $m_{15}$ have $m_7$ and $m_{15}$ which can not be included in any other prime-implicant, hence this is an essential prime-implicant.

## 5.4.3 Grouping Eight Adjacent Ones

Eight cells form a group of eight adjacent ones if three of the literals associated with the minterms/maxterms are not same and the other literals are same. In case of 3-variable K-map, there is only one possibility of eight ones appearing in the K-map and this corresponds to output equal to 1, irrespective of the values of the input variables. Table 5.7 gives all possible groups of eight adjacent ones in a 4-variable K-map. From an understanding of this, we can easily find out such combinations for 5- and 6-variable K-maps. When eight adjacent ones are combined, the resulting equation will have only one term with the number of literals three less than the number of literals in the original minterms. Similar to the groupings of adjacent two and four ones, the literals which are common in all the eight minterms will be present and the literals which are not same get eliminated in the resulting term.

Table 5.7 ***Groups of Eight Adjacent Ones in 4-variable K-map***

| Decimal numbers of cells forming groups of adjacent eights in a 4-variable K-map |
|---|
| 0, 4, 12, 8, 1, 5, 13, 9, |
| 0, 4, 12, 8, 2, 6, 14, 10 |
| 0, 1, 3, 2, 4, 5, 7, 6 |
| 0, 1, 3, 2, 8, 9, 11, 10 |
| 1, 5, 13, 9, 3, 7, 15, 11 |
| 4, 5, 7, 6, 12, 13, 15, 14 |
| 12, 13, 15, 14, 8, 9, 11, 10 |
| 3, 7, 15, 11, 2, 6, 14, 10 |

The reader is advised to verify the simplification of eight adjacent ones into a single term with three variables eliminated. For example, let us take the first group of eight adjacent ones in Table 5.7. For all these eight cells, the variable $C$ appears as $\overline{C}$ in the minterms and the other three variables are not same. Therefore, the grouping of these eight cells results in a term $\overline{C}$. Figure 5.11 shows the simplified expression for each of the groupings of eight ones for a 4 variable K-map.

## 5.4.4 Grouping 2, 4, and 8 Adjacent Zeros

In the above discussion, we have considered groups of 2, 4, and 8 adjacent ones. Instead of making the groups of ones, we can also make groups of zeros. The procedure is similar to the one used above and is as follows:

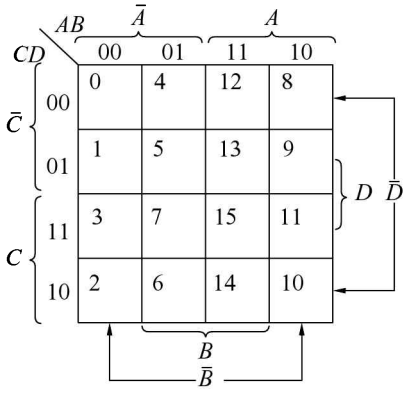


Fig. 5.11 ***Four-variable K-map Illustrating the Groupings of Eight Adjacent Ones***

1. Group of two adjacent zeros result in a term with one literal less than the number of variables. The literal which is not same in the two maxterms gets eliminated.
2. Group of four adjacent zeros result in a term with two literals less than the number of variables. The two literals which are not same in all the four maxterms get eliminated.
3. Group of eight adjacent zeros result in a term with three literals less than the number of variables. The three literals which are not same in all the eight maxterms get eliminated.

We have considered groups of 2, 4, and 8 adjacent ones and zeros. The same logic can be extended to 16, 32, and 64 adjacent ones and zeros which occur in K-maps with more than 4 variables.

## 5.5 MINIMISATION OF LOGIC FUNCTIONS SPECIFIED IN MINTERMS/MAXTERMS OR TRUTH TABLE

### 5.5.1 Minimisation of SOP Form

We have seen the advantages of simplifying a logical expression. If the expression is simplified to a stage beyond which it can not be further simplified, it will require minimum number of gates with minimum number of inputs to the gates. Such an expression is referred to as the *minimised expression.*

For minimising a given expression in SOP form or for a given truth table, we have to prepare the K-map first and then look for combinations of ones on the K-map. We have to combine the ones in such a way that the resulting expression is minimum. To achieve this, the following algorithm can be used which will definitely lead to minimised expression:

1. Identify the ones which can not be combined with any other ones and encircle them. These are *essential prime-implicants.*
2. Identify the ones that can be combined in groups of two in only one way. Encircle such groups of ones.
3. Identify the ones that can be combined with three other ones, to make a group of four adjacent ones, in only one way. Encircle such groups of ones.
4. Identify the ones that can be combined with seven other ones, to make a group of eight adjacent ones, in only one way. Encircle such groups of ones.
5. After identifying the essential groups of 2, 4, and 8 ones, if there still remains some ones which have not been encircled then these are to be combined with each other or with other already encircled ones. Of course, however, we should combine the left-over ones in largest possible groups and in as few groupings as possible. In this, the groupings may not be unique and we should make the groupings in an optimum manner. You can verify that any one can be included any number of times without affecting the expression.

The logic function consisting of the essential prime-implicants obtained in steps 1 to 4 and the prime-implicants obtained in step 5 will be the minimised function. The above algorithm will be used to minimise the logic functions in the examples given.

## Example 5.13

Minimise the four-variable logic function using K-map.

$$f(A, B, C, D) = \Sigma m(0, 1, 2, 3, 5, 7, 8, 9, 11, 14) \tag{5.23}$$

### *Solution*

The K-map of Eq. (5.23) is shown in Fig. 5.12. The equation is minimised in the following steps:

1. Encircle 1 in cell 14 which can not be combined with any other 1. The term corresponding to this is $ABC\overline{D}$.
2. There are at least two possible ways for every 1 forming groups of two adjacent ones. Therefore, we ignore it for the time being and go to the next step.
3. There is only one possible group of four adjacent ones involving each of the cells 8, 11, 5 or 7 and 2, and these are (8, 9, 0, 1), (11, 9, 1, 3), (5, 7, 3, 1) and (2, 3, 1, 0), respectively. Encircle these groups. The terms corresponding to these groups are $\overline{B}\overline{C}$, $\overline{B}D$, $\overline{A}D$, and , $\overline{A}\overline{B}$ respectively.

Since all the ones have been encircled, therefore, the minimised equation is

$$f(A,B,C,D) = ABC\overline{D} + \overline{B}\overline{C} + \overline{B}D + \overline{A}D + \overline{A}\overline{B} \tag{5.24}$$

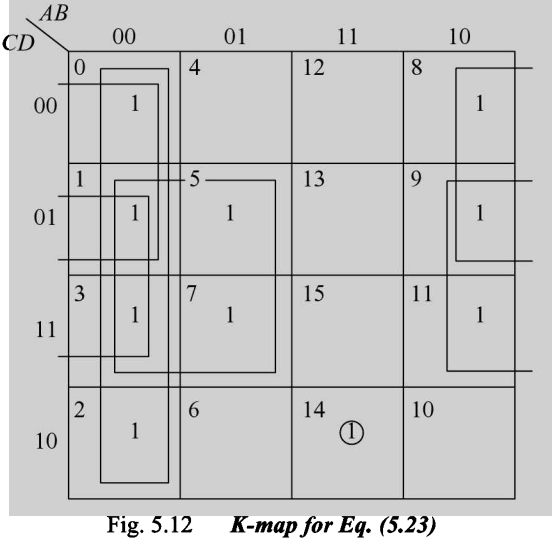


Fig. 5.12 ***K-map for Eq. (5.23)***

## Example 5.14

Determine the minimised expression in SOP form for the truth table given in Table 5.8.

### *Solution*

The K-map for the truth table of Table 5.8 is shown in Fig. 5.13. Using the minimisation steps, we obtain the minimised expression.

$$Y = \overline{B} + A\overline{C} + \overline{A}CD \tag{5.25}$$

Table 5.8

| Inputs | | | | Output |
|---|---|---|---|---|
| ***A*** | ***B*** | ***C*** | ***D*** | ***Y*** |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

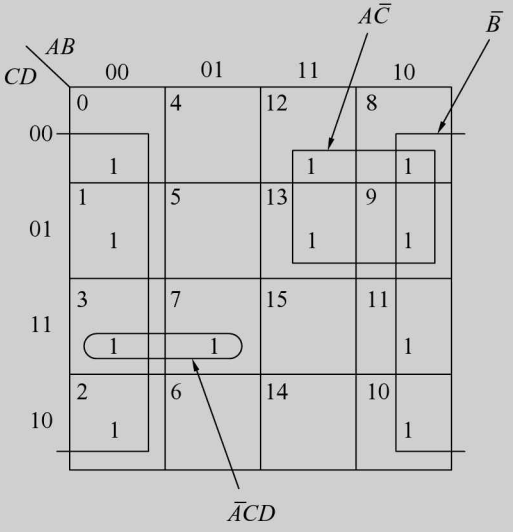


Fig. 5.13 ***K-map of Table 5.8***

## 5.5.2 Minimisation of POS Form

For minimising a given expression in POS form or for a given truth table we write zeros in the cells corresponding to maxterms for 0 outputs. The K-map is simplified by following the same procedure as used for SOP form with ones replaced by zeros. In this, groups of zeros are formed rather than groups of ones. We shall minimise the above two examples in POS form.

### Example 5.15

Minimise the logic function of Eq. (5.23) in POS form.

#### *Solution*

Equation (5.23) can be expressed in canonical POS form as

$$f(A, B, C, D) = \Pi M(4, 6, 10, 12, 13, 15) \tag{5.26}$$

The K-map corresponding to Eq. (5.26) is shown in Fig. 5.14. Note that the K-map can also be obtained directly from Eq. (5.23).

Using steps similar to those outlined for SOP form, we obtain the minimised expression,

$$f = (\bar{A} + B + \bar{C} + D) \cdot (\bar{A} + \bar{B} + C) \cdot (\bar{A} + \bar{B} + \bar{D}) \cdot (A + \bar{B} + D) \tag{5.27}$$

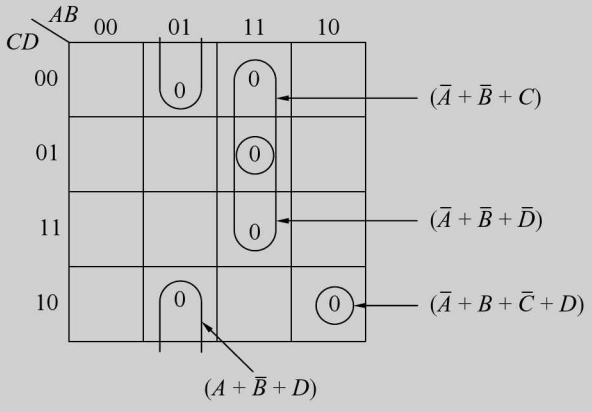


Fig. 5.14 ***K-map of Eq. (5.26)***

If we compare Eqs (5.24) and (5.27), we observe that the number of terms are not same in the two minimisations. In fact, in general the two minimisations will not have the same number of terms and will require different quantities of hardware. Therefore, one can obtain both minimisations and select the one which requires minimum hardware. In some situations there may not be any choice to the designer because of non-availability of certain ICs.

### Example 5.16

Minimise the truth table given in Table 5.8 using maxterms.

#### *Solution*

The K-map is given in Fig. 5.15.
The simplified expression is

$$Y = (A + \bar{B} + C)(\bar{A} + \bar{B} + \bar{C})(\bar{B} + \bar{C} + D) \tag{5.28}$$

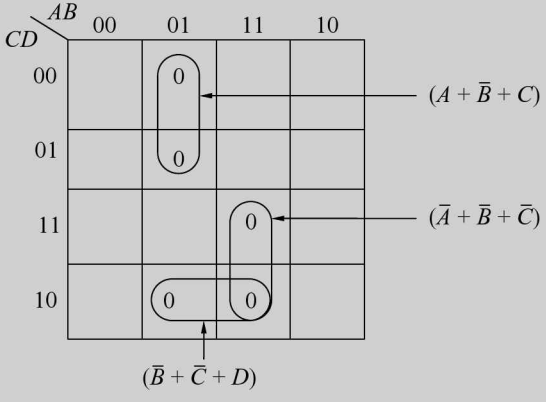


Fig. 5.15 ***K-map of Table 5.8***

Comparison of Eqs (5.25) and (5.28) confirms our generalizations made in Ex. 5.15 regarding the hardware requirements in the two methods.

## 5.6 MINIMISATION OF LOGIC FUNCTIONS NOT SPECIFIED IN MINTERMS/MAXTERMS

If the function is specified in one of the two canonical forms, its K-map can be prepared and the function can be minimised. Now we consider the cases where the functions are not specified in canonical forms. In such cases, the equations can be converted into canonical forms using the techniques given in Section 5.2, the K-maps obtained and minimised. Alternately, we can directly prepare K-map using the following algorithm:

1. Enter ones for minterms and zeros for maxterms.
2. Enter a pair of ones/zeros for each of the terms with one variable less than the total number of variables.
3. Enter four adjacent ones/zeros for terms with two variables less than the total number of variables.
4. Repeat for other terms in the similar way.

Once the K-map is prepared the minimisation procedure is same as discussed earlier. The following examples will help in understanding the above procedure:

## Example 5.17

Minimise the four variable logic function

$$f(A, B, C, D) = AB\bar{C}D + \bar{A}BCD + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{D} + A\bar{C} + A\bar{B}C + \bar{B} \tag{5.29}$$

### *Solution*

The method for obtaining K-map is

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

Fig. 5.16 ***K-map of Eq. (5.29)***

1. Enter 1 in the cell with $A = 1, B = 1, C = 0, D = 1$ corresponding to the minterm $AB\bar{C}D$.
2. Enter 1 in the cell with $A = 0, B = 1, C = 1, D = 1$ corresponding to the minterm $\bar{A}BCD$.
3. Enter 1's in the two cells with $A = 0, B = 0, C = 0$ corresponding to the term $\bar{A}\bar{B}\bar{C}$.
4. Enter 1's in the two cells with $A = 0, B = 0, D = 0$ (one of these is already entered) corresponding to the term $\bar{A}\ \bar{B}\ \bar{D}$
5. Enter 1's in the two cells with $A = 1, B = 0, C = 1$ corresponding to the term $A\bar{B}C$.
6. Enter 1's in the four cells with $A = 1, C = 0$ (one of them is already entered) corresponding to the term $A\bar{C}$.
7. Enter 1's in the eight cells with $B = 0$ (all of them except one have already been entered) corresponding to the term $\bar{B}$.

The K-map is given in Fig. 5.16 which is same as the K-map of Fig. 5.13 and the minimised expression is given by Eq. (5.25).

## Example 5.18

Minimise the four variable logic function

$$f(A, B, C, D) = (A + B + \bar{C} + \bar{D}) \cdot (\bar{A} + C + \bar{D}) \cdot (\bar{A} + B + \bar{C} + \bar{D}) \cdot (\bar{B} + C) \cdot (\bar{B} + \bar{C}) \cdot (A + \bar{B}) \cdot (\bar{B} + \bar{D}) \tag{5.30}$$

### *Solution*

The K-map cells in which 0's are to be entered corresponding to each term are given in Table 5.9. Even if a cell is involved in more than one terms, a 0 is to be entered only once.
The K-map is given in Fig. 5.17. The minimised expression is

Table 5.9 ***K-map Cells with 0 Entries***

| Term | Cell(s) with 0's |
|---|---|
| $A + B + \bar{C} + \bar{D}$ | $A = 0, B = 0, C = 1, D = 1$ |
| $\bar{A} + C + \bar{D}$ | $A = 1, C = 0, D = 1$ |

(*Continued*)

Table 5.9 **(*Continued*)**

| Term | Cell(s) with 0's |
|---|---|
| $\overline{A} + B + \overline{C} + \overline{D}$ | $A = 1,\ B = 0, C = 1, D = 1$ |
| $\overline{B} + C$ | $B = 1, C = 0$ |
| $\overline{B} + \overline{C}$ | $B = 1, C = 1$ |
| $A + \overline{B}$ | $A = 0, B = 1$ |
| $\overline{B} + \overline{D}$ | $B = 1, D = 1$ |

The K-map is given in Fig. 5.17. The minimised expression is

$$f(A, B, C, D) = \overline{B} \cdot (\overline{A} + \overline{D})(\overline{C} + \overline{D}) \tag{5.31}$$

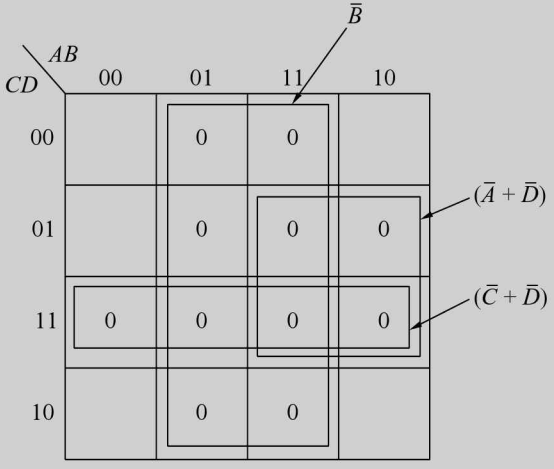


Fig. 5.17 ***K-map of Eq. (5.30)***

## 5.7 DON'T-CARE CONDITIONS

We enter 1's and 0's in the map corresponding to input variables that make the function equal to 1 or 0, respectively. The maps are simplified using either 1's or 0's. Therefore, we make the entries in the map for either 1's or 0's. The cells which do not contain 1 are assumed to contain 0 and vice-versa. This is not always true since there are cases in which certain combinations of input variables do not occur. Also, for some functions the outputs corresponding to certain combinations of input variables do not matter. In such situations the designer has a flexibility and it is left to him whether to assume a 0 or a 1 as output for each of these combinations. This condition is known as *don't-care* condition and can be represented on the K-map as a × mark in the corresponding cell. The × mark in a cell may be assumed to be a 1 or a 0 depending upon which one leads to a simpler expression. The function can be specified in one of the following ways:

1. In terms of minterms and don't-care conditions. For example,

$$f(A, B, C, D) = \Sigma m(1, 3, 7, 11, 15) + d(0, 2, 5) \tag{5.32}$$

Its K-map and the minimised expression are given in Fig. 5.18*a*.

2. In terms of maxterms and don't-care conditions. For example,

$$f(A, B, C, D) = \Pi M(4, 5, 6, 7, 8, 12) \cdot d(1, 2, 3, 9, 11, 14) \tag{5.33}$$

Its K-map and the minimised expression are given in Fig. 5.18*b*.

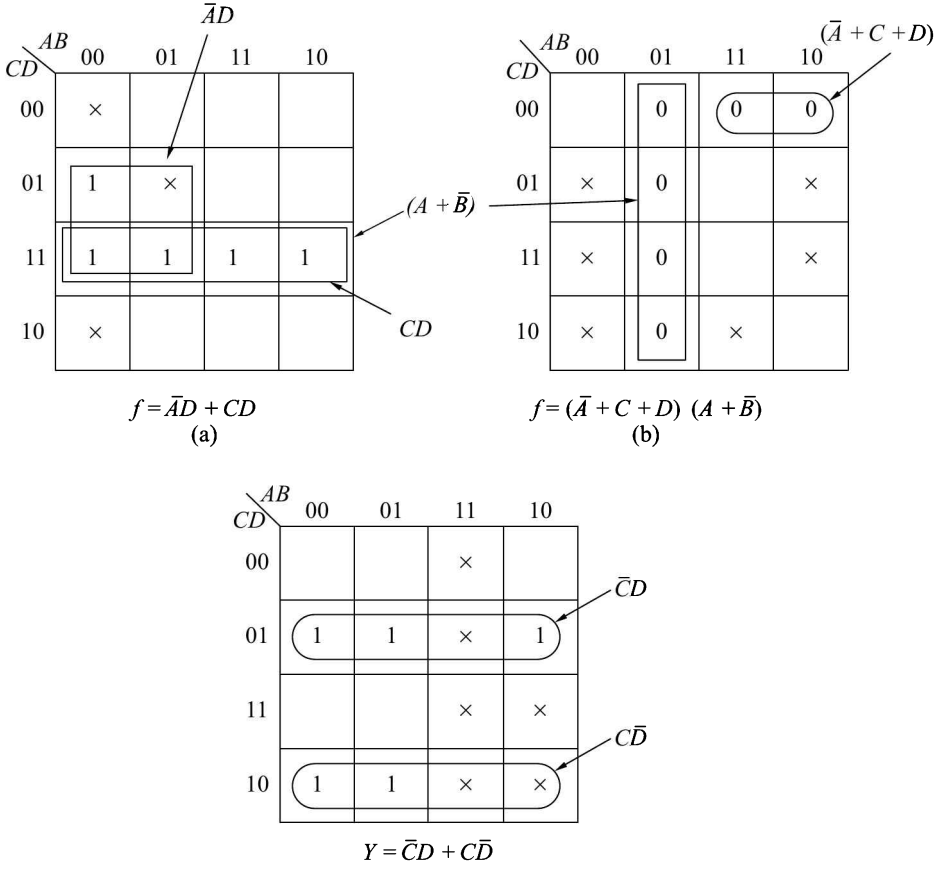


Fig. 5.18 ***K-maps with Don't-care Conditions***

3. In terms of truth table. For example, consider the truth table of Table 5.10.

Its K-map and the minimised expression in SOP form are given in Fig. 5.18*c*.

Table 5.10

| Inputs | | | | Output |
|---|---|---|---|---|
| *A* | *B* | *C* | *D* | *Y* |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | × |
| 1 | 0 | 1 | 1 | × |
| 1 | 1 | 0 | 0 | × |
| 1 | 1 | 0 | 1 | × |
| 1 | 1 | 1 | 0 | × |
| 1 | 1 | 1 | 1 | × |