2

Quine-McCluskey Minimization

Chapter Outline

Minimization of Boolean Expression using Karnaugh Maps has been exhaustively dealt with in Chapter 1. Karnaugh maps are effective in simplifying expression with a small number of variables, as evident by that complexities involved in five and six variable maps. An algorithmic approach suitable for implementation on a digital computer is desirable when large number of variables and multiple functions are involved. The Quine-McCluskey method presented in the chapter provides one such algorithmic approach. For convenience, m is used to represent minterms and M for maxterms.

2.1 Introduction

The Quine-McCluskey method of simplification applies to expressions with any number of variables and is algorithmic in nature as we shall soon see.

The Quine-McCluskey algorithm consists of two main steps. All the prime implicants and implicates as the case may be are obtained in the first step. As many literals as possible are eliminated systematically by application of the logical adjacency theorem $XY + \overline{X}\,Y = Y$ which to obtain the prime implicants.

In the second step all possible irrelevant expressions of the function is obtained out of which the minimal set is picked. The algorithm is explained with plenty of illustrative examples of increasing complexity.

2.2 The Quine-McCluskey Method for obtaining Prime Implicants

The logical adjacency theorem $XY + \overline{X}\,Y = Y$ can be repeatedly applied to obtain all the implicants of a Boolean function from its minterms. Let X represent a single variable and let Y represent a product of variables. The logical adjacency theorem implies that if two product terms differ in only one variable, where this variable appears complemented in one and uncomplemented in the other, then these two product terms can be replaced by a single term without the variable which has changed *i.e.*, XY and $\overline{X}\,Y$ can be replaced by Y. For example, consider the product terms $a\overline{b}c\overline{d}$ and $\overline{a}\,\overline{b}c\overline{d}$. Here $X = a$ and $Y = \overline{b}c\overline{d}$. Hence $a\overline{b}c\overline{d}$ and $\overline{a}\,\overline{b}c\overline{d}$ can be replaced by $\overline{b}c\overline{d}$.

Broadly speaking, the Quine-McCluskey process begins with the listing of all the minterms in one column of a table. Every pair of minterms is examined to see if, $XY + \overline{X}\,Y = Y$ can be applied if yes, a '✓' mark is placed besides these pairs of minterms and the extracted terms Y are placed in the second column. This process is then applied to the second column by examining every pair of product terms in the second column to see if, $XY + \overline{X}\,Y = Y$ is applicable. If yes, again a '✓' mark is placed besides these pairs and the extracted terms Y are placed in a third column. This process is repeated until there are no pairs left in the last column. Those terms which remain unchecked are the prime implicants. This process is illustrated with the following examples.

EXAMPLE 2.1

Obtain all the prime implicants of the given Boolean functions using Quine-McCluskey method.

a. $f(a, b, c) = \Sigma\,(0, 2, 3, 4)$
b. $f(a, b, c) = \Sigma\,(0, 1, 2, 3, 4, 5, 6)$

Solution:

a. $f = \Sigma\,(0, 2, 3, 4)$

The minterms can be picked up from the tabulation shown.

Cell no.	*a*	*b*	*c*	*f*	*Minterm*
0	0	0	0	1	$\overline{a}\overline{b}\overline{c}$
1	0	0	1	0	
2	0	1	0	1	$\overline{a}b\overline{c}$
3	0	1	1	1	$\overline{a}bc$
4	1	0	0	1	$a\overline{b}\overline{c}$
5	1	0	1	0	
6	1	1	0	0	
7	1	1	1	0	

Let us now enter the minterms in the first column of the Quine–McCluskey table. Then starting with the first entry in the column 1, look at all pairs of terms to see if $XY + \overline{X}\,Y = Y$ can be applied. If 'yes' place a '✓' besides such pairs and enter the extracted Ys in column 2. No pairs in column 2 pair as $XY + X\overline{Y}$.

Cell no.	*Column 1 Minterm*	*Column 2*
0	$\overline{a}\overline{b}\overline{c}$ ✓	(0, 4) $\overline{b}\overline{c}$
2	$\overline{a}b\overline{c}$ ✓	(0, 2) $\overline{a}\,\overline{c}$
3	$\overline{a}bc$ ✓	(2, 3) $\overline{a}\overline{b}$
4	$a\overline{b}\overline{c}$ ✓	

∴ The prime implicants are $\overline{b}\,\overline{c}$, $\overline{a}\,\overline{c}$ and $\overline{a}\,b$.

b. $f = \Sigma\,(0, 1, 2, 3, 4, 5, 6)$

The Quine–McCluskey tabulation is shown below.

Cell no.	*Column 1 Minterm*	*Column 2*	*Column 3*
0	$\overline{a}\overline{b}\overline{c}$ ✓	(0, 1) $\overline{a}\overline{b}$ ✓	(0, 1), (2, 3) $\overline{a}$
1	$\overline{a}\overline{b}c$ ✓	(0, 2) $\overline{a}\,\overline{c}$ ✓	(0, 2), (1, 3)
2	$\overline{a}b\overline{c}$ ✓	(0, 4) $\overline{b}\overline{c}$ ✓	(0, 4), (1, 5) $\overline{b}$
3	$\overline{a}bc$ ✓	(1, 3) $\overline{a}c$ ✓	(0, 1), (4, 5)
4	$a\overline{b}\overline{c}$ ✓	(1, 5) $\overline{b}c$ ✓	(0, 2), (4, 6) $\overline{c}$
5	$a\overline{b}c$ ✓	(2, 3) $\overline{a}b$ ✓	(0, 4), (2, 6)
6	$ab\overline{c}$ ✓	(2, 6) $b\overline{c}$ ✓	
		(4, 5) $a\overline{b}$ ✓	
		(4, 6) $a\overline{c}$ ✓	

The prime implicants are $\overline{a}$, $\overline{b}$ and $\overline{c}$.

EXAMPLE 2.2

Obtain all the prime implicants of the following Boolean function using Quine–McCluskey method

$$f(a, b, c, d) = \Sigma\,(0, 2, 3, 5, 8, 10, 11)$$

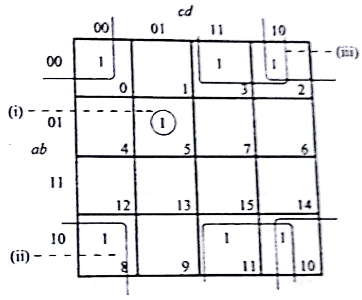
***Solution*:**

ab \ cd	00	01	11	10
00	1 (0)	(1)	1 (3)	1 (2)
01	(4)	1 (5)	(7)	(6)
11	(12)	(13)	(15)	(14)
10	1 (8)	(9)	1 (11)	1 (10)

The Quine–McCluskey tabulation is shown below.

Cell no.	*Column 1 Minterm*	*Column 2*	*Column 3*
0	$\bar{a}\bar{b}\bar{c}\bar{d}$ ✓	(0, 2) $\bar{a}\bar{b}\bar{d}$ ✓	(0, 2),(8, 10) $\bar{b}\bar{d}$
2	$\bar{a}\bar{b}c\bar{d}$ ✓	(0, 8) $\bar{b}\bar{c}\bar{d}$ ✓	(0, 8),(2, 10)
3	$\bar{a}\bar{b}cd$ ✓	(2, 3) $\bar{a}\bar{b}c$ ✓	(2, 3) (10, 11) $\bar{b}c$
5	$\bar{a}b\bar{c}d$	(8, 10) $a\bar{b}\bar{d}$ ✓	(2,10) (3, 11)
8	$a\bar{b}\bar{c}\bar{d}$ ✓	(10, 11) $a\bar{b}c$ ✓	
10	$a\bar{b}c\bar{d}$ ✓	(2, 10) $\bar{b}c\bar{d}$ ✓	
11	$a\bar{b}cd$ ✓	(3, 11) $\bar{b}cd$ ✓	

∴ The prime implicants are those against which there are no '✓' marks *i.e.*, $\bar{a}b\bar{c}d$, $\bar{b}\bar{d}$ and $\bar{b}c$
Let us verify using Karnaugh map.



The prime implicants are given by $\bar{a}b\bar{c}d$ (subcube(i)), $\bar{b}\bar{d}$ (subcube (ii)) and $\bar{b}c$ (subcube (iii)).

The Quine-McCluskey process can be written down in algorithmic form. We need to introduce some notations to make the algorithm simpler. 1s, 0s and dashes (–) are used to represent any product term of an n–variable Boolean function. An uncomplemented variable is represented by 1, a complemented variable by 0 and a variable that is not present by a '–'.

Fig. 2.1 gives some examples of product terms represented using this convention.

Function	*Product term*	*1,0, – representation*	*Index*
$f(a,b,c,d)$	$\bar{a}\bar{b}cd$	0 0 1 1	2
$f(a,b,c,d)$	$a\bar{b}\bar{d}$	1 0 – 0	1
$f(a,b,c,d,e)$	$\bar{c}$	– – 0 – –	0
$f(a,b,c,d,e)$	$\bar{a}b\bar{e}$	0 1 – – 0	1
$f(a,b,c,d)$	bd	– 1 – 1	2
$f(a,b,c,d,e)$	$ab\bar{c}e$	1 1 0 – 1	3

Fig. 2.1 Examples of 1, 0, – representation

The number of 1s in the 1, 0, – representation is called the *index* of a term. This is illustrated in Table 2.1. For two terms to qualify to be grouped under $XY + \bar{X}Y$, a 0 in one term should be a 1 in the other or vice versa. Thus the difference in the indices of such terms will be precisely 1. We can also observe that the other literals in these two minterms will be the same and hence will have the 1s, 0s and dashes at the same locations except for the variable which is complemented in one of the terms Examples are shown in Fig. 2.2.

Function	XY	XY in 10– Notation	$\bar{X}Y$	$\bar{X}Y$ in 10– Notation	X	Y	Y in 10– Notation
$f(a,b,c,d)$	$\bar{a}bd$	01–1	$\bar{a}\bar{b}d$	00–1	b	$\bar{a}d$	0––1
$f(a,b,c,d,e)$	bde	–1–11	$bd\bar{e}$	––10	e	bd	–1–1–
$f(a,b,c,d)$	$\bar{b}c$	–01–	$\bar{b}\bar{c}$	–00–	c	$\bar{b}$	–0––
$f(a,b,c,d)$	$ac\bar{d}$	1–10	acd	1–11	$\bar{d}$	ac	1–1–

Fig. 2.2 Example of adjacent terms represented in 1, 0, – notation

Observe in Fig. 2.2 that the single term Y which replaces two terms $XY + \bar{X}Y$ has the 0s and 1s in the same locations as the individual term XY and $\bar{X}Y$, with the exception that term Y has – position where the variable differed.

The terms in each column of the Quine–McCluskey tabulation are grouped into sets according to their indices. It is now necessary only to look for pairs of terms that combine under $XY + \bar{X}Y$ among terms in sets which differ by 1 in their index. Pairs of terms will never combine under $XY + \bar{X}Y$ to be replaced by a single term if their indices are either same or differ by more than 1.

2.3 Quine–McCluskey Algorithm for obtaining Prime Implicants

The Quine-McCluskey procedure for obtaining all the prime implicants can be algorithmically stated as follows:

1. Represent each minterm in its 1/0 notation.
2. Write down the minterm in increasing order of their index in one column.
3. Draw a line after each set of minterms with same index value.
4. Set $i = 0$.
5. Pick up each term with index i and $i + 1$ to see if they differ in exactly one position. If yes write the single term which results from the combination in 1, 0, – notation in a new column and place a '✓' besides the two terms that combined.
   If no, proceed with other pairs until all the pairs with indices i and $i + 1$ have been compared.
   Now draw a line under the last term in the new list.
6. Set $i = i + 1$. Repeat step 5 until all the terms have been covered.
7. Repeat steps 4, 5 and 6 on the new list to form another list.
8. Terminate the process when no new lists are formed.
9. The prime implicants of the function are all those terms without a '✓' besides them.
   This algorithm is illustrated with the following examples.

Find all the prime implicants of the function $f(a, b, c, d) = \Sigma(0, 2, 3, 4, 8, 10, 12, 13, 14)$ using the Quine– McCluskey algorithm.

Solution:

$$f(a, b, c, d) = \Sigma(0, 2, 3, 4, 8, 10, 12, 13, 14)$$

Step 1: Represent each minterm in its 1/0 notation

No.	*Minterm*	1/0 *notation*	*Index*
0	$\bar{a}\bar{b}\bar{c}\bar{d}$	0000	0
2	$\bar{a}\bar{b}c\bar{d}$	0010	1
3	$\bar{a}\bar{b}cd$	0011	2
4	$\bar{a}b\bar{c}\bar{d}$	0100	1
8	$a\bar{b}\bar{c}\bar{d}$	1000	1
10	$a\bar{b}c\bar{d}$	1010	2
12	$ab\bar{c}\bar{d}$	1100	2
13	$ab\bar{c}d$	1101	3
14	$abc\bar{d}$	1110	3

Step 2: List the minterms in increasing order of their index.

	a	b	c	d	
0	0	0	0	0	index 0
2	0	0	1	0	
4	0	1	0	0	index 1
8	1	0	0	0	
3	0	0	1	1	
10	1	0	1	0	index 2
12	1	1	0	0	
13	1	1	0	1	index 3
14	1	1	1	0	

Fig. 2.3 Steps 2 and 3 of Quine– McCluskey algorithm

Step 3: Draw a line after each set of minterms with the same index value done in Fig. 2.3.

Step 4: Set $i = 0$.

Step 5: Pick up each term with index 0 and 1 and see if they differ in one bit position. If place a '✓' besides those terms and place the new single term in a new list.

	a	b	c	d	
0	0	0	0	0	✓
2	0	0	1	0	✓
4	0	1	0	0	✓
8	1	0	0	0	✓
3	0	0	1	1	
10	1	0	1	0	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	

	a	b	c	d
(0,2)	0	0	–	0
(0,4)	0	–	0	0
(0,8)	–	0	0	0

Fig. 2.4 After iteration $i = 0$

Step 6: Set $i = 1$ pick up each term with index 1 and 2 and repeat step 5.

	a	b	c	d	
0	0	0	0	0	✓
2	0	0	1	0	✓
4	0	1	0	0	✓
8	1	0	0	0	✓
3	0	0	1	1	✓
10	1	0	1	0	✓
12	1	1	0	0	✓
13	1	1	0	1	
14	1	1	1	0	

	a	b	c	d
(0, 2)	0	0	–	0
(0, 4)	0	–	0	0
(0, 8)	–	0	0	0
(2, 3)	0	0	1	–
(2, 10)	–	0	1	0
(4, 12)	–	1	0	0
(8, 10)	1	0	–	0
(8, 12)	1	–	0	0

Fig. 2.5 After iteration $i = 1$

Set $i = 2$ Pick up each term with index 2 and 3 and repeat step 5

	a	b	c	d	
0	0	0	0	0	✓
2	0	0	1	0	✓
4	0	1	0	0	✓
8	1	0	0	0	✓
3	0	0	1	1	✓
10	1	0	1	0	✓
12	1	1	0	0	✓
13	1	1	0	1	✓
14	1	1	1	0	✓

	a	b	c	d	
(0, 2)	0	0	–	0	index = 0
(0, 4)	0	–	0	0	
(0, 8)	–	0	0	0	
(2, 3)	0	0	1	–	index = 1
(2, 10)	–	0	1	0	
(4, 12)	–	1	0	0	
(8, 10)	1	0	–	0	
(8, 12)	1	–	0	0	
(10, 14)	1	–	1	0	index = 2
(12, 13)	1	1	0	–	
(12, 14)	1	1	–	0	

Fig. 2.6 After iteration $i = 2$

Step 7: Repeat steps 4, 5 and 6 on the new list. Set $i = 0$. Pick up each term with index 0 and 1 in the second list and repeat step 5.

	a	b	c	d	
0	0	0	0	0	✓
2	0	0	1	0	✓
4	0	1	0	0	✓
8	1	0	0	0	✓
3	0	0	1	1	✓
10	1	0	1	0	✓
12	1	1	0	0	✓
13	1	1	0	1	✓
14	1	1	1	0	✓

	a	b	c	d	
(0, 2)	0	0	–	0	✓
(0, 4)	0	–	0	0	✓
(0, 8)	–	0	0	0	✓
(2, 3)	0	0	1	–	✓
(2, 10)	–	0	1	0	✓
(4, 12)	–	1	0	0	✓
(8, 10)	1	0	–	0	✓
(8, 12)	1	–	0	0	✓
(10, 14)	1	–	1	0	
(12, 13)	1	1	0	–	
(12, 14)	1	1	–	0	

	a	b	c	d
(0, 2) (8, 10)	–	0	–	0
(0, 4) (8, 12)	–	–	0	0
(8, 10) (12, 14)	1	–	–	0

Fig. 2.7 After iteration $i = 0$ on the second list

Note that combining (0, 2) (8, 10) or (0, 8) (2, 10) results in the same single term – 0 – 0. Combining (0, 4) (8, 12) or (0, 8) (4, 12) results in the same single term – – 0 0. 0 0 1 – does not pair with any other term and is left without a '✓' beside it.

Set $i = 1$. Pick up each term with index 1 and 2 in the second list and repeat step 5.

	a	b	c	d	
0	0	0	0	0	✓
2	0	0	1	0	✓
4	0	1	0	0	✓
8	1	0	0	0	✓
3	0	0	1	1	✓
10	1	0	1	0	✓
12	1	1	0	0	✓
13	1	1	0	1	✓
14	1	1	1	0	✓

	a	b	c	d	
(0, 2)	0	0	–	0	✓
(0, 4)	0	–	0	0	✓
(0, 8)	–	0	0	0	✓
(2, 3)	0	0	1	–	
(2, 10)	–	0	1	0	✓
(4, 12)	–	1	0	0	✓
(8, 10)	1	0	–	0	✓
(8, 12)	1	–	0	0	✓
(10, 14)	1	–	1	0	✓
(12, 13)	1	1	0	–	
(12, 14)	1	1	–	0	✓

	a	b	c	d	
(0, 2) (8, 10)	–	0	–	0	index =
(0, 4) (8, 12)	–	–	0	0	
(8, 10) (12, 14)	1	–	–	0	index =

Fig. 2.8 After iteration $i = 1$ on the second list

Note that combining (8, 10) (12, 14) or (8, 12) (10, 14) results in the same single term 1 – –
Repeat steps 4, 5 and 6 on the third list. Set $i = 0$.
Pick up each term with index 0 and 1 in the third list and repeat step 5.
We see that no new list can be generated as no pairs of terms combine to form single term

Step 8: Terminate the process as no new lists are formed.

Step 9: The prime implicants are the terms without a '✓' mark beside them. $\bar{a}\bar{b}c$, $ab\bar{c}$ from second list and $\bar{b}\,\bar{d}$, $\bar{c}\,\bar{d}$ and $a\bar{d}$ from the third list.

Find all the prime implicants of the function

$$f(a, b, c, d) = \Sigma\,(7, 9, 12, 13, 14, 15) + \Sigma d\,(4, 11)$$

using Quine–McCluskey algorithm

Solution: We saw in Section 1.5.5 that don't care cells are considered as 1-cells to determine prime implicants. The given function thus becomes

$$f(a, b, c, d) = \Sigma\,(4, 7, 9, 11, 12, 13, 14, 15)$$

Step 1: Represent each minterms in its 1/0 notation.

No.	Minterm	1/0 notation	Index
7	$\bar{a}bcd$	0111	3
9	$a\bar{b}\bar{c}d$	1001	2
12	$ab\bar{c}\bar{d}$	1100	2
13	$ab\bar{c}d$	1101	3
14	$abc\bar{d}$	1110	3
15	$abcd$	1111	4
4	$\bar{a}b\bar{c}\bar{d}$	0100	1
11	$a\bar{b}cd$	1011	3

Step 2: List the minterms in increasing order of their index.

	a	b	c	d	
4	0	1	0	0	index 1
9	1	0	0	1	index 2
12	1	1	0	0	
7	0	1	1	1	index 3
11	1	0	1	1	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	index 4

Fig. 2.9 Steps 2 and 3 of Quine– McCluskey algorithm

All further steps are shown in Fig. 2.10. The minterms combining at various iterations are shown with an offset in the '✓' marks

	a	b	c	d			
4	0	1	0	0	✓		
9	1	0	0	1		✓	
12	1	1	0	0	✓	✓	
7	0	1	1	1			✓
11	1	0	1	1		✓	✓
13	1	1	0	1		✓	✓
14	1	1	1	0		✓	✓
15	1	1	1	1			✓

	a	b	c	d	
(4, 12)	–	1	0	0	
(9, 11)	1	0	–	1	✓
(9, 13)	1	–	0	1	✓
(12, 13)	1	1	0	–	✓
(12, 14)	1	1	–	0	✓
(7, 15)	–	1	1	1	
(11, 15)	1	–	1	1	✓
(13, 15)	1	1	–	1	✓
(14, 15)	1	1	1	–	✓

	a	b	c	d
(9, 11) (13, 15)	1	–	–	1
(12, 13) (14, 15)	1	1	–	–

Fig. 2.10 The Quine–McCluskey algorithm

The terms without a ✓ besides them constitute the prime implicants.
The prime implicants are

$$b\bar{c}\,\bar{d},\ bcd,\ ad \text{ and } ab$$

◆ **Exercise 2.1**

Find all the prime implicants of the following functions using Quine–McCluskey method.

a. $f(a, b, c, d) = \Sigma\,(0, 5, 6, 7, 9, 10, 12, 15)$

b. $f(a, b, c, d) = \Sigma\,(0, 1, 2, 6, 7, 9, 10, 12) + \Sigma d\,(3, 5)$

2.4 The Quine–McCluskey Method for obtaining Prime Implicates

The simple method of applying Quine–McCluskey method to find prime implicates given the function in maxterm canonical form is to apply the algorithm on the complement of the function f, which will be in minterm canonical form. This will give us the prime implicants of $\bar{f}$. The prime implicates of f are then found by inverting each prime implicant of f using De Morgan theorem. In case of incomplete Boolean functions, the don't care cells are taken as 0–cells.

The procedure is illustrated in the following example.

Find all the prime implicates of the function

$$f(a, b, c, d) = \Pi\,(0, 2, 3, 4, 5, 12, 13) + \Pi d\,(8, 10)$$

using the Quine–McCluskey method.

Solution:

$$f(a, b, c, d) = \Pi\,(0, 2, 3, 4, 5, 12, 13) + \Pi d\,(8, 10)$$

To determine prime implicates don't care–cells are taken as 0–cells

$$\therefore\quad f(a, b, c, d) = \Pi\,(0, 2, 3, 4, 5, 8, 10, 12, 13)$$

We need to apply Quine–McCluskey method to $\bar{f}$

$$\bar{f}(a, b, c, d) = \overline{M_0 \cdot M_2 \cdot M_3 \cdot M_4 \cdot M_5 \cdot M_8 \cdot M_{10} \cdot M_{12} \cdot M_{13}}$$

$$= \overline{M_0} + \overline{M_5} + \overline{M_6} + \overline{M_7} + \overline{M_8} + \overline{M_9} + \overline{M_{13}} + \overline{M_{15}}$$

$$= m_0 + m_2 + m_3 + m_4 + m_5 + m_8 + m_{10} + m_{12} + m_{13}$$

$$\therefore\quad \bar{f}(a, b, c, d) = \Sigma\,(0, 2, 3, 4, 5, 8, 10, 12, 13)$$

Now let us apply the Quine–McCluskey method to find the prime implicates of $\bar{f}$

Step 1: Represent each term in its 1/0 notation

No.	*Minterm*	*1/0 notation*	*Index*
0	$\bar{a}\bar{b}\bar{c}\bar{d}$	0000	0
2	$\bar{a}\bar{b}c\bar{d}$	0010	1
3	$\bar{a}\bar{b}cd$	0011	2
4	$\bar{a}b\bar{c}\bar{d}$	0100	1
5	$\bar{a}b\bar{c}d$	0101	1
8	$a\bar{b}\bar{c}\bar{d}$	1000	2
10	$a\bar{b}c\bar{d}$	1010	2
12	$ab\bar{c}\bar{d}$	1100	3
13	$ab\bar{c}d$	1101	3

Step 2: List the minterms in increasing order of their index

	a b c d	
0	0 0 0 0	index 0
2	0 0 1 0	index 1
4	0 1 0 0	
8	1 0 0 0	
3	0 0 1 1	index 2
5	0 1 0 1	
10	1 0 1 0	
12	1 1 0 0	
13	1 1 0 1	index 3

Fig. 2.11 Steps 2 and 3 of Quine–McCluskey algorithm

All further steps are shown in Fig. 2.12.

	a	b	c	d			
0	0	0	0	0	✓		
2	0	0	1	0	✓	✓	
4	0	1	0	0	✓	✓	
8	1	0	0	0	✓	✓	
3	0	0	1	1		✓	
5	0	1	0	1		✓	✓
10	1	0	1	0		✓	
12	1	1	0	0		✓	✓
13	1	1	0	1		✓	✓

	a	b	c	d	✓	
(0, 2)	0	0	–	0	✓	
(0, 4)	0	–	0	0	✓	
0, 8	–	0	0	0	✓	
2, 3	0	0	1	–		
2, 10	–	0	1	0	✓	
4, 5	0	1	0	–		✓
4, 12	–	1	0	0		✓
8, 10	1	0	–	0	✓	
8, 12	1	–	0	0	✓	
5, 13	–	1	0	1		✓
12, 13	1	1	0	–		✓

		a	b	c	d
(0, 2)	(8, 10)	–	0		
(0, 4)	(8, 12)	–	–	0	
(4, 5)	(12, 13)		1	0	

Fig. 2.12 The Quine–McCluskey algorithm

The prime implicants of $\bar{f}$ are the terms without ✓ beside them. Prime implicates of f are the complements of these prime implicants found by applying De Morgan's theorem.

Prime Implicants of $\bar{f}$	*Prime Implicates of* f
$\bar{a}\bar{b}c$	$a+b+\bar{c}$
$\bar{b}\bar{d}$	$b+d$
$\bar{c}\bar{d}$	$c+d$
$b\bar{c}$	$\bar{b}+c$

◆ **Exercise 2.2**

Find all the prime implicates of the function $f(a, b, c, d) = \Pi\,(1, 3, 6, 7, 9, 11, 13, 14)$ by applying Quine McCluskey algorithm on $\bar{f}$.

2.5 Prime Implicant Tables for obtaining Irredundant Expressions

All the prime implicants are not always required to form irredundant minimal expressions based on some cost to be minimized. The Quine–McCluskey method was a simple procedure to obtain the prime implicants and prime implicates of any Boolen function expressed in its minterm or maxterm canonical form. We need to develop a procedure by which we could choose those prime implicants which constitute minimal expressions.

The minimal expression with reference to some cost criterion will contain a set of prime implicants so that each minterm of the function subsumes atleast one prime implicant in this set.

A tabulation which express the relationships between the prime implicants of a complete Boolean function and the minterms of a function is called the prime implicant table. Consider the function in Example 1.22(b) in Section 1.5.3, $f = \Sigma\,(2, 3, 4, 5, 7)$. The prime implicants were found to be $\bar{a}b$, $a\bar{b}$, ac and bc. While writing the prime implicant table, the minterms are placed along the horizontal axis and the prime implicants along the vertical axis, as shown for this example in Fig. 2.13.

	m_2 $\bar{a}b\bar{c}$	m_3 $\bar{a}bc$	m_4 $a\bar{b}\bar{c}$	m_5 $a\bar{b}c$	m_7 abc
$\bar{a}b$	×	×			
$a\bar{b}$			×	×	
ac				×	×
bc					×

Fig. 2.13 Prime implicant table for $f = \Sigma\,(2, 3, 4, 5, 7)$

Begin with the first minterm and place a '×' mark in all rows where the minterm subsumes the prime implicant. Both the minterms $\bar{a}b\bar{c}$ and $\bar{a}bc$ subsume prime implicant $\bar{a}b$. Hence, a '×' is placed under columns m_2 and m_3 corresponding to the $\bar{a}b$ row. Similarly, both the minterms $a\bar{b}\bar{c}$ and $a\bar{b}c$ subsume prime implicant $a\bar{b}$. Hence a '×' is placed under columns m_4 and m_5 corresponding to the $a\bar{b}$ row. Minterm abc subsumes both prime implicants ac as well as bc. Hence, a '×' is placed under column m_7 corresponding to both ac row and bc row. In the case of incomplete Boolean function minterms corresponding to don't care states are not placed along the horizontal axis. Only minterms for which the function equals 1 are placed along the horizontal axis in the prime implicant table.

Write the prime implicant table of the function $f(a, b, c, d) = \Sigma\,(0, 1, 2, 5, 10, 11, 14, 15)$ of Example 1.15 in Section 1.5.1.

Solution: The prime implicants obtained in Example 1.15 were

$$\bar{a}\bar{b}\bar{d}, \bar{a}b\bar{c}, \bar{a}\bar{c}d, \bar{b}c\bar{d} \text{ and } ac$$

The prime implicant table is generated now.

	m_0 $\bar a\bar b\bar c\bar d$	m_1 $\bar a\bar b\bar c d$	m_2 $\bar a\bar b c\bar d$	m_5 $\bar a b\bar c d$	m_{10} $a\bar b c\bar d$	m_{11} $a\bar b c d$	m_{14} $a b c\bar d$	m_{15} $abcd$
$\bar a\bar b\bar d$	×		×					
$\bar a\bar b\bar c$	×	×						
$\bar a\bar c d$		×		×				
$\bar b c\bar d$			×		×			
ac					×	×	×	×

Fig. 2.14 Prime implicant table for $f(a, b, c, d) = \Sigma(0, 1, 2, 5, 10, 11, 14, 15)$

If a given Boolean expression describing a complete function in the SOP from (a) every product term is a prime implicant and (b) no product term can be dropped from the expression without changing the function described by the expression, then such a Boolean expression is said to be in an Irredundant Disjunctive Normal Form.

This means that given an SOP expression one or more literal can be removed such that the resulting expression has (a) only prime implicants as its terms and (b) no terms can be dropped without altering the function description.

The minimal set of prime implicants without altering the function definition can be obtained from the prime implicant table. A set of prime implicants are so chosen that removing any prime implicant from this set would render one or more columns in the table without '×' marks along the rows of the remaining prime implicants in the set. This means that some of the minterms would cease to imply the function.

In Fig. 2.13, we see that together with $\bar a b$ and $a\bar b$, by either choosing ac or bc, all the minterms would be covered.

Thus, the irredundant disjunctive normal expressions are

1. $f(a, b, c) = \bar a b + a\bar b + ac$ and
2. $f(a, b, c) = \bar a b + a\bar b + bc$

The minimal form can be identified by computing the cost of (1) and (2) based on some cost criterion. If the number of inputs in a two level gate implementation is the cost criterion, the cost of both the irredundant forms is 9.

2.6 Petrick's Method of Determining Irredundant Expressions from Prime Implicant Table

If we try to apply the procedure adopted to find irredundant expressions from Fig. 2.13 to Fig. 2.14 the process appears complicated. As the number of minterms increase, reading irredundant prime implicants directly from the prime implicant table becomes difficult. Petrick's method is a generalized procedure to find irredundant expressions from the prime implicant table.

Let us study the procedure with the help of Fig. 2.13 where the prime implicants have been labeled with letters W, X, Y and Z as shown in Fig. 2.15.

		m_2 $\bar a b\bar c$	m_3 $\bar a bc$	m_4 $a\bar b\bar c$	m_5 $a\bar b c$	m_7 abc
W	$\bar a b$	×	×			
X	$a\bar b$			×	×	
Y	ac				×	×
Z	bc					×

Fig. 2.15 Prime implicant table to illustrater Petrick's method on Fig. 2.13

Every prime implicant along the horizontal axis must evaluate to 1 in order to preserve the function definition. We can write an algebraic expression called the p–expression which states this requirement that every minterm along the horizontal axis must evaluate to 1 as

$$p = m_2 \cdot m_3 \cdot m_4 \cdot m_5 \cdot m_7$$

m_2 is implied by prime implicants W
m_3 is implied by prime implicant W
m_4 is implied by prime implicant X
m_5 is implied by prime implicant X or Y
m_7 is implied by prime implicant Y or Z

$$\therefore \quad p = (W)(W)(X)(X+Y)(Y+Z)$$
$$= W(X + XY)(Y + Z)$$

Here XY subsumes X and can be dropped

$$p = WX(Y+Z) = WXZ + WXZ$$

Observe that the expression for p is also a Boolean expression. Hence, the two irredundant normal expressions are

1. $f = WXY = \bar a b + a\bar b + ac$ and
2. $f = WXZ = \bar a b + a\bar b + bc$

which are the same expressions obtained in Section 2.5.

Let us now apply Petrick's method to prime implicant table in Fig. 2.14 of Example 2.6. Table in Fig. 2.16 shows the prime implicant table along with the labels for the prime implicants.

		m_0	m_1	m_2	m_5	m_{10}	m_{11}	m_{14}	m_{15}
V	$\bar{a}\bar{b}\bar{d}$	×		×					
W	$\bar{a}\bar{b}\bar{c}$	×	×						
X	$\bar{a}\bar{c}d$		×		×				
Y	$\bar{b}c\bar{d}$			×		×			
Z	ac					×	×	×	×

Fig. 2.16 Petrick's method applied to Table in Fig. 2.14

Let us now write the *p*-expression

$$p = m_0 \cdot m_1 \cdot m_2 \cdot m_5 \cdot m_{10} \cdot m_{11} \cdot m_{14} \cdot m_{15}$$

m_0 is implied by prime implicants V or W
m_1 is implied by prime implicant W or X
m_2 is implied by prime implicant V or Y
m_5 is implied by prime implicant X
m_{10} is implied by prime implicant Y or Z
m_{11} is implied by prime implicant Z
m_{14} is implied by prime implicant Z
m_{15} is implied by prime implicant Z

$$\begin{aligned} p &= (V+W)(W+X)(V+Y)(X)(Y+Z)(Z)(Z)(Z) \\ &= (V+W)(W+X)(V+Y)(X)(Y+Z)(Z) \\ &= (V+W)(W+X)(V+Y)(X)(YZ+Z) \end{aligned}$$

YZ subsumes Z and can be dropped

$$\therefore \quad \begin{aligned} p &= (V+W)(W+X)(V+Y)(X)(Z) \\ &= (V+W)(V+Y)(WXZ+XZ) \end{aligned}$$

WXZ subsumes XZ and can be dropped

$$\therefore \quad \begin{aligned} p &= (V+W)(V+Y)(XZ) \\ &= (V+WV+VY+WY)(XZ) \end{aligned}$$

WV subsumes V and can be dropped
VY subsumes V and can be dropped

$$\therefore \quad \begin{aligned} p &= (V+WY)(XZ) \\ &= VXZ + WXYZ \end{aligned}$$

The two irredundant normal expressions are

1. $f = VXZ = \bar{a}\bar{b}\bar{d} + \bar{a}\bar{c}\bar{d} + ac$
2. $f = WXYZ = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{c}\bar{d} + \bar{b}c\bar{d} + ac$

Let us verify this by minimizing the function using Karnaugh map.

$$f = \Sigma(0, 1, 2, 5, 10, 11, 14, 15)$$

The Karnaugh map for this function is shown in Fig. 2.17.

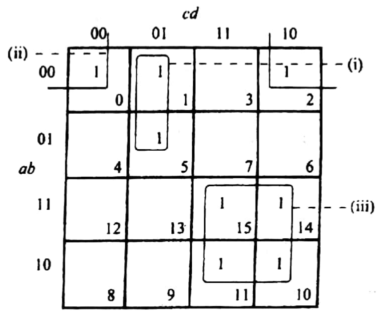


Fig. 2.17 Karnaugh map of $f = \Sigma(0, 1, 2, 5, 10, 11, 14, 15)$

Essential cell 5 groups to subcube (i)
Cell 0 groups to subcube (ii)
The remaining 1s form the largest subcube (iii)
All 1s have been covered in atleast one subcube. ∴ The irredundant sum is

$$f = \bar{a}\bar{b}\bar{d} + \bar{a}\bar{c}d + ac$$

[illegible]

For the following Boolean function use the Quine–McCluskey method to obtain all the prime implicants and apply Petrick's method to find the irredundant disjunctive normal expressions and identify the minimal sums.

$$f(a, b, c, d) = \Sigma(4, 5, 7, 12, 14, 15)$$

Solution: Let us first obtain all the prime implicants by Quine–McCluskey algorithm.

Step 1: Represent each minterm in its 1/0 notation

No.	*Minterm*	*1/0 notation*	*Index*
4	$\bar a b \bar c \bar d$	0100	1
5	$\bar a b \bar c d$	0101	2
7	$\bar a b c d$	0111	3
12	$a b \bar c \bar d$	1100	2
14	$a b c \bar d$	1110	3
15	$abcd$	1111	4

Step 2: List the minterms in increasing order of their index.

Step 3: Draw a line after each set of terms with the same index value. The results of steps 2 and 3 are shown below

	a	*b*	*c*	*d*	
4	0	1	0	0	index 1
5	0	1	0	1	index 2
12	1	1	0	0	
7	0	1	1	1	index 3
14	1	1	1	0	
15	1	1	1	1	index 4

Step 4: Set $i = 1$ (There are no minterms with index = 0)

Step 5: Pick up each term with index 1 and 2 and see if they differ in one bit position. If 'yes' place a '✓' beside those terms and place the new single term in the new list.

	a	*b*	*c*	*d*	
4	0	1	0	0	✓
5	0	1	0	1	✓
12	1	1	0	0	✓
7	0	1	1	1	
14	1	1	1	0	
15	1	1	1	1	

	a	*b*	*c*	*d*
(4, 5)	0	1	0	–
(4, 12)	–	1	0	0

Step 6: Set $i = 2$ pick up each term with index $i = 2$ and 3 and repeat step 5.

	a	*b*	*c*	*d*	
4	0	1	0	0	✓
5	0	1	0	1	✓
12	1	1	0	0	✓
7	0	1	1	1	✓
14	1	1	1	0	✓
15	1	1	1	1	

	a	*b*	*c*	*d*
(4, 5)	0	1	0	–
(4, 12)	–	1	0	0
(5, 7)	0	1	–	1
(12, 14)	1	1	–	0

Set $i = 3$. Pick up each term with index $i = 3$ and 4 and repeat step 5.

	a	*b*	*c*	*d*	
4	0	1	0	0	✓
5	0	1	0	1	✓
12	1	1	0	0	✓
7	0	1	1	1	✓
14	1	1	1	0	✓
15	1	1	1	1	✓

	a	*b*	*c*	*d*	
(4, 5)	0	1	0	–	index = 1
(4, 12)	–	1	0	0	
(5, 7)	0	1	–	1	index = 2
(12, 14)	1	1	–	0	
(7, 15)	–	1	1	1	index = 3
(14, 15)	1	1	1	–	

Step 7: Repeat steps 4, 5 and 6 on the new list. Set $i = 1$. Pick up each terms with index 1 and 2 in the second list and repeat step 5. There are no two terms which differ in one bit position. This is true for all value of i. Thus, no new list can be generated.

Step 8: Terminate the process as no new lists are formed.

Step 9: The prime implicants are those terms without a '✓' mark besides them.

∴ The prime implicants are

$$\bar a b \bar c,\ b \bar c \bar d,\ \bar a bd,\ ab\bar d,\ bcd,\ abc$$

Let us now construct the prime implicant table with the minterms along the horizontal axis and prime implicants along the vertical axis.

	m_4 $\bar{a}b\bar{c}\bar{d}$	m_5 $\bar{a}b\bar{c}d$	m_7 $\bar{a}bcd$	m_8 $ab\bar{c}\bar{d}$	m_9 $abc\bar{d}$	m_{13} $abcd$
$\bar{a}b\bar{c}$	×	×				
$b\bar{c}\bar{d}$	×			×		
$\bar{a}bd$		×	×			
$ab\bar{d}$				×	×	
bcd			×			×
abc					×	×

Place a '×' mark below each minterm against every row whose prime implicant is subsumed by the minterm.

Let us now apply Petrick's method to pick up just enough prime implicants to form irredundant disjunctive normal expressions without altering the function definition. Let us first label the prime implicants as shown below.

		m_4	m_5	m_7	m_{12}	m_{14}	m_{15}
U	$\bar{a}b\bar{c}$	×	×				
V	$b\bar{c}\bar{d}$	×			×		
W	$\bar{a}bd$		×	×			
X	$ab\bar{d}$				×	×	
Y	bcd			×			×
Z	abc					×	×

Write the p-expression

$$p = m_4 \cdot m_5 \cdot m_7 \cdot m_{12} \cdot m_{14} \cdot m_{15}$$

m_4 subsumes prime implicants U and V
m_5 subsumes prime implicants U and W
m_7 subsumes prime implicants W and Y
m_{12} subsumes prime implicants V and X
m_{14} subsumes prime implicants X and Z
m_{15} subsumes prime implicants Y and Z

$$\therefore \quad p = (U+V)(U+W)(W+Y)(V+X)(X+Z)(Y+Z)$$
$$= (U + VU + UW + VW)(W+Y)(V+X)(X+Z)(Y+Z)$$
$$= (UW + VUW + UW + VW + UY + VUY + UWY + VWY)(V+X)(X+Z)(Y+Z)$$

VUW subsumes UW and is dropped
VUY subsumes UY and is dropped
UWY subsumes UW and is dropped
VWY subsumes VW and is dropped

$$\therefore \quad p = (UW + VW + UY)(V+X)(X+Z)(Y+Z)$$
$$= (UVW + VW + UVY + UWX + VWX + UXY)(X+Z)(Y+Z)$$

UVW subsumes VW and is dropped
VWX subsumes VW and is dropped

$$p = (VW + UVY + UWX + UXY)(x+Z)(Y+Z)$$
$$= (VWX + UVXY + UWX + UXY + VWZ + UVYZ + UWXZ + UXYZ)(Y+Z)$$

UVXY subsumes UXY and is dropped
UWXZ subsumes UWX and is dropped
UXYZ subsumes UXY and is dropped

$$\therefore \quad p = (VWX + UWX + UXY + VWZ + UVYZ)(Y+Z)$$
$$= VWXY + UWXY + UXY + VWYZ + UVYZ + VWXZ + UWXZ + UXYZ + VWZ + UVYZ$$

UWXY subsumes UXY and is dropped
UXYZ subsumes UXY and is dropped
VWXZ subsumes VWZ and is dropped
VWYZ subsumes VWZ and is dropped

$$\therefore \quad p = VWXY + UXY + UVYZ + UWXZ + VWZ$$

Hence there are five irredundant disjunctive normal expressions. These expressions along with their costs are shown below. The number of gate inputs in a two level gate circuit is taken as the cost criterion.

$f_1 = VWXY = b\bar{c}\bar{d} + \bar{a}bd + ab\bar{d} + bcd$ (cost = 16)
$f_2 = UXY = \bar{a}b\bar{c} + ab\bar{d} + bcd$ (cost = 12)
$f_3 = UVYZ = \bar{a}b\bar{c} + b\bar{c}\bar{d} + bcd + abc$ (cost = 16)
$f_4 = UWXZ = \bar{a}b\bar{c} + \bar{a}bd + ab\bar{d} + abc$ (cost = 16)
$f_5 = VWZ = b\bar{c}\bar{d} + \bar{a}bd + abc$ (cost = 12)

f_2 and f_5 would give a minimal sum for this cost criterion.

2.7 Prime-Implicate Tables for Obtaining Irredundant Expressions

One way of obtaining irredundant conjunctive expressions is to find all the prime implicates of the given Boolean function in maxterm canonical form using Quine–McCluskey algorithm. A prime implicate table is then written listing all the prime implicates along the vertical axis and maxterms along the horizontal axis. Petrick's method is then applied to obtain irredundant conjunctive normal expressions.

The easier approach is to find the prime implicants of complement of the function expressed in SOP form, using Quine–McCluskey algorithm, write the prime implicant table and obtain the expressions as in Section 2.6. The p–experssion is then simplified and the irredundant conjunctive expressions are obtained for the complement of the function. The irredundant normal expressions of the original function are obtained by inverting the irredundant disjunctive or *SOP* expressions of the complemented function. The cost of all the irredundant conjunctive expressions are calculated based on the given cost criterion to arrive at the minimal products. This procedure is illustrated with the following example.

For the following Boolean function use the Quine–McCluskey method and Petrick's method to obtain all the irredundant conjunctive normal expressions. Which of these form the minimal products if the number of inputs is the cost criterion.

$$f(a, b, c, d) = \Pi(0, 6, 7, 8, 9, 13) + \Pi d(5, 15)$$

Solution: Don't care terms are considered to imply 0s for obtaining all the prime implicants of a function.

$$\therefore \quad f(a, b, c, d) = \Pi(0, 5, 6, 7, 8, 9, 13, 15)$$

$$f = M_0 \cdot M_5 \cdot M_6 \cdot M_7 \cdot M_8 \cdot M_9 \cdot M_{13} \cdot M_{15}$$

$$\bar{f} = \overline{M_0 . M_5 . M_6 . M_7 . M_8 . M_9 . M_{13} . M_{15}}$$

$$= \overline{M_0} + \overline{M_5} + \overline{M_6} + \overline{M_7} + \overline{M_8} + \overline{M_9} + \overline{M_{13}} + \overline{M_{15}}$$

$$\bar{f} = m_0 + m_5 + m_6 + m_7 + m_8 + m_9 + m_{13} + m_{15}$$

Let us now find all the prime implicants of $\bar{f}$ using Quine McCluskey algorithm.

Step 1: Represent each minterm in its 1/0 notation.

No.	Minterm	1/0 notation	Index
0	$\bar{a}\bar{b}\bar{c}\bar{d}$	0000	0
5	$\bar{a}b\bar{c}d$	0101	2
6	$\bar{a}bc\bar{d}$	0110	2
7	$\bar{a}bcd$	0111	3
8	$a\bar{b}\bar{c}\bar{d}$	1000	1
9	$a\bar{b}\bar{c}d$	1001	2
13	$ab\bar{c}d$	1101	3
15	$abcd$	1111	4

Step 2 and Step 3:

List the minterms in increasing order of their index and draw a line after each set of minterms with the same index value.

	a	b	c	d	
0	0	0	0	0	index 0
8	1	0	0	0	index 1
5	0	1	0	1	index 2
6	0	1	1	0	
9	1	0	0	1	
7	0	1	1	1	index 3
13	1	1	0	1	
15	1	1	1	1	index 4

Step 4: Set $i = 0$.

Step 5: Pick up each term with index 0 and 1 and see if they combine to form a single term, *i.e.*, see if they differ in only one bit position. Place a '✓' against these minterms and write the single term they combine into in a new list. This is shown below.

	a	b	c	d	
0	0	0	0	0	✓
8	1	0	0	0	✓
5	0	1	0	1	
6	0	1	1	0	
9	1	0	0	1	
7	0	1	1	1	
13	1	1	0	1	
15	1	1	1	1	

	a	b	c	d
(0, 8)	-	0	0	0

Step 6: Set $i = 1$. Pick up each term with index 1 and 2 and repeat step 5.

	a	b	c	d	
0	0	0	0	0	✓
8	1	0	0	0	✓
5	0	1	0	1	
6	0	1	1	0	
9	1	0	0	1	✓
7	0	1	1	1	
13	1	1	0	1	
15	1	1	1	1	

	a	b	c	d
(0, 8)	-	0	0	0
(8, 9)	1	0	0	-

Set $i = 2$. Pick up each term with index 2 and 3 and repeat step 5.

	a	b	c	d	
0	0	0	0	0	✓
8	1	0	0	0	✓
5	0	1	0	1	✓
6	0	1	1	0	✓
9	1	0	0	1	✓
7	0	1	1	1	✓
13	1	1	0	1	✓
15	1	1	1	1	

	a	b	c	d
(0, 8)	–	0	0	0
(8, 9)	1	0	0	–
(5, 7)	0	1	–	1
(5, 13)	–	1	0	1
(6, 7)	0	1	1	–
(9, 13)	1	–	0	1

Set $i = 3$. Pick up each term with index 3 and 4 and repeat step 5.

	a	b	c	d	
0	0	0	0	0	✓
8	1	0	0	0	✓
5	0	1	0	1	✓
6	0	1	1	0	✓
9	1	0	0	1	✓
7	0	1	1	1	✓
13	1	1	0	1	✓
15	1	1	1	1	✓

	a	b	c	d	
(0, 8)	–	0	0	0	index 0
(8, 9)	1	0	0	–	index 1
(5, 7)	0	1	–	1	index 2
(5, 13)	–	1	0	1	
(6, 7)	0	1	1	–	
(9, 13)	1	–	0	1	
(7, 15)	–	1	1	1	index 3
(13, 15)	1	1	–	1	

Step 7: Repeat steps 4, 5 and 6 on the new list.
Set $i = 0$.
No terms with index 0 and 1 differ in one bit position in the second list.
Set $i = 1$.
No terms with index 1 and 2 differ in one bit position in the second list.
Set $i = 2$. Pick up each term with index 2 and 3 in the second list and repeat step 5.

	a	b	c	d	
0	0	0	0	0	✓
8	1	0	0	0	✓
5	0	1	0	1	✓
6	0	1	1	0	✓
9	1	0	0	1	✓
7	0	1	1	1	✓
13	1	1	0	1	✓
15	1	1	1	1	✓

	a	b	c	d	
(0, 8)	–	0	0	0	
(8, 9)	1	0	0	–	
(5, 7)	0	1	–	1	✓
(5, 13)	–	1	0	1	✓
(6, 7)	0	1	1	–	
(9, 13)	1	–	0	1	
7, 15	–	1	1	1	✓
13, 15	1	1	–	1	✓

	a	b	c	d
(5, 7) (13, 15)	–	1	–	1

$\bar{b}\bar{c}d$
$a\bar{b}\bar{c}$
$\bar{a}bc$
$a\bar{c}d$
bd

Note that combining (5, 7) (13, 15) or (5, 13) (7, 15) results in the same single term –1–1. There is only one term in the third list.

Step 8: Terminate the process as no new list is formed.
Step 9: The prime implicants are those terms without a '✓' mark.

The prime implicants of $\bar{f}$ are

$$\bar{b}\,\bar{c}\,\bar{d},\ a\,\bar{b}\,\bar{c},\ \bar{a}\,b\,c,\ a\,\bar{c}\,d \text{ and } b\,d$$

Construct the prime implicant table with the minterms along the horizontal axis and the prime implicants along the vertical axis. Consider only the actual 0-terms and not the don't cares.

	m_0 $\bar{a}\bar{b}\bar{c}\bar{d}$	m_6 $\bar{a}bc\bar{d}$	m_7 $\bar{a}bcd$	m_8 $a\bar{b}\bar{c}\bar{d}$	m_9 $a\bar{b}\bar{c}d$	m_{13} $ab\bar{c}d$
$\bar{b}\bar{c}\bar{d}$	×			×		
$a\bar{b}\bar{c}$				×	×	
$\bar{a}bc$		×	×			×
$a\bar{c}d$					×	×
bd			×			×

Place a '×' mark below each minterm against every row whose prime implicants are subsumed by the minterm.

Now apply Petrick's procedure to pick up just enough prime implicants to form irredundant disjunctive normal expressions of $\bar{f}$. Let us first label the prime implicants as shown below.

		m_0	m_6	m_7	m_8	m_9	m_{13}
V	$\bar{b}\bar{c}\bar{d}$	×			×		
W	$a\bar{b}\bar{c}$				×	×	
X	$\bar{a}bc$		×	×			
Y	$a\bar{c}d$					×	×
Z	bd			×			×

Write the p-expression

$$p = m_0 \cdot m_6 \cdot m_7 \cdot m_8 \cdot m_9 \cdot m_{13}$$
$$p = (V)(X)(X+Z)(V+W)(W+Y)(Y+Z)$$
$$= (VX + VZ + Wx + WZ)(W+Y)(Y+Z)(VX)$$
$$= (VX + VXZ + VWX + VWXZ)(W+Y)(Y+Z)$$

VXZ subsumes VX and is dropped

VWX subsumes VX and is dropped

$VWXZ$ subsumes VX and is dropped

$$\therefore \quad p = (VX(W+Y))(Y+Z)$$
$$= (VWX + VXY)(Y+Z)$$
$$= VWXY + VXY + VWXZ + VXYZ$$

$VWXY$ subsumes VXY and is dropped

$VXYZ$ subsumes VXY and is dropped

$$\therefore \quad p = VXY + VWXZ$$

The two irredundant disjunctive forms of $\bar{f}$ are

$$\bar{f} = VXY = \bar{b}\bar{c}\bar{d} + \bar{a}bc + a\bar{c}d$$
$$\bar{f} = VWXZ = \bar{b}\bar{c}\bar{d} + a\bar{b}\bar{c} + \bar{a}bc + bd$$

$\therefore$ Applying De Morgan's theorem to $\bar{f}_1$ and $\bar{f}_2$ we have

$$f_1 = \overline{\overline{f_1}} = \overline{\bar{b}\bar{c}\bar{d} + \bar{a}bc + a\bar{c}d}$$
$$= \overline{\bar{b}\bar{c}\bar{d}} \cdot \overline{\bar{a}bc} \cdot \overline{a\bar{c}d}$$
$$f_1 = (b+c+d)(a+\bar{b}+\bar{c})(\bar{a}+c+\bar{d}) \text{ (cost = 12)}$$
$$f_2 = \overline{\overline{f_2}} = \overline{\bar{b}\bar{c}\bar{d} + a\bar{b}\bar{c} + \bar{a}bc + bd}$$
$$= \overline{\bar{b}\bar{c}\bar{d}} \cdot \overline{a\bar{b}\bar{c}} \cdot \overline{\bar{a}bc} \cdot \overline{bd}$$
$$f_2 = (b+c+d)(\bar{a}+b+c)(a+\bar{b}+\bar{c})(\bar{b}+\bar{d}) \text{ (cost = 15)}$$

$\therefore$ The minimal product is

$$f = (b+c+d)(a+\bar{b}+\bar{c})(\bar{a}+c+\bar{d})$$

◆ **Exercise 2.3**

For the following Boolean function use the Quine–McCluskey method and Petrick's method to obtain all the irredundant conjunctive normal expressions. Which of these form minimal products.

$$f(a\,b\,c\,d) = \Pi\,(3, 4, 5, 8, 9, 11, 13, 14, 15)$$

◆ **Exercise 2.4**

For the following Boolean function use the Quine–McCluskey method and Petrick's method to obtain all the irredundant disjunctive normal expressions. Which of these form minimal sums.

$$f(a, b, c, d) = \Sigma\,(0, 4, 7, 8, 11, 12, 14, 15)$$

2.8 Prime Implicant and Prime Implicate Table Reductions

As we observed in Examples 2.7 and 2.8 Petrick's method of extracting irredundant expressions is more laborious if the number of columns and rows are large. Obtaining irredundant disjunctive normal form can be easier, if the number of rows and columns in the prime implicant table can be reduced and this is often possible.

2.8.1 Essential Prime Implicants

In the prime implicant table if the column corresponding to any minterm has only one '×' entry, then the corresponding prime implicant along the row is an essential prime implicant and the row in which '×' entry appears is referred to as an essential row. These essential implicants must appear in all irredundant disjunctive expressions and hence must appear in the minimal sum, because, the minimal sum is simply one of the irredundant disjunctive expressions selected, based on some cost criterion.

Once the essential implicants are identified, the other prime implicants required to obtain irredundant disjunctive expressions without altering the function definition can be found by reducing the prime implicant table as follows.

1. Delete all columns in which an '×' mark appears along an essential row.
2. Delete all essential rows.

Now obtain the irredundant disjunctive expression of the reduced table by applying Petrick's method. Augment the essential prime implicants to this irredundant disjunctive expression obtained from the reduced table to obtain the irredundant disjunctive expressions of the original function.

Let us illustrate prime implicant table reduction using the following examples.

Consider the prime implicant table of the function $f(a, b, c) = \Sigma\,(2, 3, 4, 5, 7)$ shown in Fig. 2.15 reproduced in Fig. 2.18 for convenience.

		m_2	m_3	m_4	m_5	m_7
W	$\bar{a}b$	×	×			
X	$a\bar{b}$			×	×	
Y	ac				×	×
Z	bc					×

Fig. 2.18 Prime implicant table for $f(a, b, c) = \Sigma\,(2, 3, 4, 5, 7)$

W row is an essential row as there is only one '×' in column m_2. Hence, delete this column and all other columns where × appears corresponding to row W, *i.e.*, delete column m_3 too. Delete the entire W row, X row is another essential row as there is only one '×' in column m_4 hence delete this column as well as column m_5 where an '×' appears along row X. Now delete the entire X row. The essential prime implicants are $\bar{a}b$ and $a\bar{b}$ which correspond to the essential rows. The reduced prime implicant table is shown in Fig. 2.19.

		m_7
Y	ac	×
Z	bc	×

Fig. 2.19 Reduction of table in Fig. 2.18

Applying Petrick's method the irredundant disjunctive expression of the reduced table is $Y + Z$ or $(ac + bc)$.

Hence the irredundant normal expression of the function is obtained by including the essential implicants to get

$$f_1 = \bar{a}b + a\bar{b} + ac \quad \text{or}$$
$$f_2 = \bar{a}b + a\bar{b} + bc$$

2.8.2 Column and Row Reductions

Columns and rows of prime implicant table can be deleted subject to certain conditions. One of the criteria for minimality was to obtain an expression with minimum number of terms containing more than one literal *e.g.*, $f = \bar{a}\,b + c + \bar{d}$.

Only the term with more than one literal require gating at the first level and can be assigned a cost equal to 1 and those terms with one literal be assigned a cost equal to 0.

The total number of gate inputs for a two level gating circuit was taken as another criterion for minimality. Consider the function

$$f(a, b, c, d) = a\bar{b}\,c + ab\bar{c}d + \bar{a}\bar{d}$$

The cost of each term is one more than the number of literals to account for the OR input at the second level. Hence cost of the above expression is 4 + 5 + 3 = 12. We will generally adopt the second cost criterion and add it to the prime implicant table.

Table in Fig. 2.20 is drawn with the cost of each prime implicant included according to the second cost criterion.

	m_0	m_1	m_2	m_5	m_{10}	m_{11}	m_{14}	m_{15}	Cost
A	×		×						4
B	×	×							4
C		×		×					4
D							×		4
E					×	×	×	×	3

Fig. 2.20 A prime implicant table to illustrate column reduction

We need to look at some definitions. If there are '×'s in the same rows in two columns in a prime implicant table, the two columns are said to be equal. In Fig. 2.20 columns m_{10}, m_{11} and m_{15} are equal.

A column is said to dominate another, if in addition to having '×'s in the same rows as that of the other column, it has '×'s in rows in which the other column does not have '×'s. In Fig. 2.20 column m_{14} dominates columns m_{10}, m_{11} and m_{15}; column m_0 dominates column m_2 and column m_1 dominates column m_5. Now we can state the condition for removal of a column as follows:

A column C_i in a prime implicant table can be deleted if

a. C_i equals C_j.
b. C_i dominates C_j

where C_j is some other column in the table. In other words, equal and dominating columns can be removed.

Applying this column reduction rule to Fig. 2.20, we find that

Column m_0 is deleted as it dominates column m_2
Column m_1 is deleted as it dominates column m_5
Column m_{10} and m_{11} are deleted as they equal column m_{15}
Column m_{14} is deleted as it dominates column m_{15}

After column reduction table in Fig. 2.20 will be as shown in Fig. 2.21.

	m_2	m_5	m_{15}	Cost
A	×			4
B				
C		×		4
D				
E			×	3

Fig. 2.21 Table in Fig. 2.20 after column reduction

Since the remaining columns have only one '×' each, these correspond to the essential prime implicants.

∴ The irredundant disjunctive normal expression which is also the minimum sum is

$$f = A + C + E$$

Similar rules could also be applied to delete rows in a prime implicant table.

If two rows have '×'s at exactly the same columns, they are said to be equal.

A row is said to dominate another if in addition to having '×'s in the same columns as that of the other row, it has '×'s in columns in which the other row has no '×'s.

	m_1	m_5	m_7	m_9	m_{10}	m_{11}	*Cost*
A	×		×	×			3
B	×		×				4
C		×			×		4
D						×	3
E		×			×		3

Fig. 2.22 Table to illustrate equal rows and dominating rows

In Fig. 2.22, rows *C* and *E* are equal in which row *A* dominates row *B*.
The conditions for the removal of a row is as follows.

a. When two rows are equal remove the row with higher or equal cost
b. Remove a dominated row if it costs more or equal to the dominating row.

In other words equal and dominated rows are eligible to be deleted subject to their costs. But the irredundant expressions cannot be obtained if rows are deleted. However, one minimal sum generated. In Fig. 2.22 rows *C* and *E* are equal, but *C* costs more and hence row *C* must be deleted. Row *B* is dominated by row *A* and costs more than row *A* and hence row *B* must be deleted.

We arrive at a cyclic table if row and columns can no longer be deleted based on equality or dominance and no column has only '×', meaning there are no essential prime implicants. Petrick's method can be applied to the cyclic table to obtain the terms which together with the essential prime implicants from the irredundant expressions.

2.8.3 Procedure to Select Prime Implicants to obtain a Single Minimal Sum

We can adopt the following procedure to obtain a single minimal sum from the prime implicant table.

1. Identify all the essential prime implicants, *i.e.*, look for columns with only one '×' and the essential prime implicant can be found in the corresponding row called the essential row. Score out all the rows and all the columns which have '×' in the essential row.
2. Score out all dominating columns and dominated rows. Dominated rows are deleted subject to the cost conditions stated in Section 2.8.2.
3. If any column that is left has a angle ×, identify the corresponding essential prime implicant along the row called the secondary essential row. Score out the secondary essential row and all columns where '×' appear in the secondary essential row.
   If there are no columns with single '×', but more than one '×', then we have a cyclic table.
4. At this point if all the columns have been scored out, all the essential prime implicants constitute the minimal sum.
   If all the columns are not scored out, repeat steps 2 and 3 until either all the columns are scored out or a cyclic table is obtained.
5. Use Petrick's method on the cyclic table to arrive at the irredundant expressions. The essential prime implicants together with those obtained with Petrick's method constitute the minimal sum

Find the minimal cover for the prime implicant table shown below.

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	*Cost*
r_1			×		×		×	3
r_2				×		×		4
r_3		×	×			×		4
r_4	×	×				×		4
r_5			×		×		×	4
r_6	×	×			×	×		6
r_7			×	×	×			6

Solution: There are no columns with a single ×, hence there are no essential prime implicants.

Delete c_2 as c_2 dominates c_1
Delete c_6 as c_6 dominates c_1 and c_2
Delete c_3 as c_3 dominates c_7
Delete c_5 as c_5 dominates c_7
The table reduces to

	c_1	c_4	c_7	*Cost*
r_1			×	3
r_2		×		4
r_4	×			4
r_5			×	4
r_6	×			6
r_7		×		6

Delete r_5 as $r_1 = r_5$ and cost $(r_5) >$ cost (r_1)
Delete r_6 as $r_4 = r_6$ and cost $(r_6) >$ cost (r_4)
Delete r_7 as $r_2 = r_7$ and cost $(r_7) >$ cost (r_2)
The table now reduces to

	c_1	c_4	c_7	*Cost*
r_1			×	
r_2		×		
r_4	×			

Now the table has only one '×' in each column. Hence the minimal cover is row 1, row 2, row 4 or minimal cover = $\{r_1, r_2, r_4\}$ and the minimal sum is

$$f = r_1 + r_2 + r_4$$

[redacted]

Find the minimal cover for the prime implicant table shown below.

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	*Cost*
r_1		×		×	×		×					3
r_2						×				×		3
r_3											×	4
r_4	×	×						×	×			4
r_5	×			×			×					4
r_6				×					×	×		5
r_7	×						×	×				6
r_8			×			×			×	×		7
r_9		×	×		×				×			7

Solution: c_{11} has only one '×' and hence r_3 is an essential prime implicant. Delete r_3 and
There is no '×' along r_3 in any other column.

Delete c_1 as c_1 dominates c_8
Delete c_2 as c_2 dominates c_5
Delete c_9 as c_9 dominates c_3
Delete c_{10} as c_{10} dominates c_6
Minimal cover so far is $\{r_3 \ldots\}$. The table becomes

	c_3	c_4	c_5	c_6	c_7	c_8	*Cost*
r_1		×	×		×		3
r_2				×			3
r_4						×	4
r_5		×			×		4
r_6		×					5
r_7					×	×	6
r_8	×			×			7
r_9	×		×				7

Delete r_5 as r_5 is dominated by r_1 and cost $(r_5) >$ cost (r_1)
Delete r_6 as r_6 is dominated by r_1 and cost $(r_6) >$ cost (r_1)
r_2 is dominated by r_8, but cost $(r_2) <$ cost (r_8)
r_4 is dominated by r_7, but cost $(r_4) <$ cost (r_7)
∴ The table becomes

	c_3	c_4	c_5	c_6	c_7	c_8	*Cost*
r_1	×		×		×		3
r_2				×			3
r_4						×	4
r_7					×	×	6
r_8	×			×			7
r_9	×		×				7

c_4 has only one '×' and hence r_1 figures in minimal cover. Delete r_1 and all columns where '×' appears along r_1, *i.e.*, delete c_5 and c_7.
∴ Minimal cover so far is $\{r_3, r_1 \ldots\}$

	c_3	c_6	c_8	*Cost*
r_2		×		3
r_4			×	4
r_7			×	6
r_8	×	×		7
r_9	×			7

Delete r_7 as $r_4 = r_7$ and cost $(r_7) >$ cost (r_4)
Delete r_9 as r_9 is dominated by r_8 and cost $(r_9) =$ cost (r_8)
∴ The table becomes

	c_3	c_6	c_8	*Cost*
r_2		×		
r_4			×	
r_8	×	×		

c_3 has single '×' along r_8 and c_8 has single '×' along r_4. Hence, r_4 and r_8 figure in the minimal cover.

∴ The minimal cover so far is $\{r_1, r_3, r_4, r_8\}$

∴ c_3 and c_8 are deleted. c_6 is also deleted as it has an '×' along row r_8.

∴ The minimal cover is $\{r_1, r_3, r_4, r_8\}$.

[redacted]

Find a minimal sum for the following incomplete Boolean function using Quine–McCluskey method and prime–implicant table reduction.

$$f(a, b, c, d) = \Sigma\,(3, 4, 5, 7, 10, 12, 14, 15) + \Sigma\, d\,(2)$$

Solution: For finding all the prime implicants don't cares are taken as 1s. Hence, the function becomes

$$f(a, b, c, d) = \Sigma\,(2, 3, 4, 5, 7, 10, 12, 14, 15)$$

Let us first apply the Quine–McCluskey method.

Step 1: Represent each minterm in its 1/0 notation

No.	*Minterm*	*1/0 Notation*	*Index*
2	$\bar{a}\bar{b}c\bar{d}$	0010	1
3	$\bar{a}\bar{b}cd$	0011	2
4	$\bar{a}b\bar{c}\bar{d}$	0100	1
5	$\bar{a}b\bar{c}d$	0101	2
7	$\bar{a}bcd$	0111	3
10	$a\bar{b}c\bar{d}$	1010	2
12	$ab\bar{c}\bar{d}$	1100	2
14	$abc\bar{d}$	1101	3
15	$abcd$	1111	4

Step 2 and Step 3:

List the minterms in increasing order of their index and draw a line after each set of terms with the same index value.

	a	*b*	*c*	*d*	
2	0	0	1	0	index 1
4	0	1	0	0	
3	0	0	1	1	index 2
5	0	1	0	1	
10	1	0	1	0	
12	1	1	0	0	
7	0	1	1	1	index 3
14	1	1	1	0	
15	1	1	1	1	index 4

Step 4: Set $i = 1$. Pick up each term with index 1 and 2 and see if they differ in one bit position.

	a	*b*	*c*	*d*	
2	0	0	1	0	✓
4	0	1	0	0	✓
3	0	0	1	1	✓
5	0	1	0	1	✓
10	1	0	1	0	✓
12	1	1	0	0	✓
7	0	1	1	1	
14	1	1	1	0	
15	1	1	1	1	

	a	*b*	*c*	*d*
(2, 3)	0	0	1	–
(2, 10)	–	0	1	0
(4, 5)	0	1	0	–
(4, 12)	–	1	0	0

Step 5: Set $i = 2$. Pick up each term with index 2 and 3 and repeat step 5.

	a	*b*	*c*	*d*	
2	0	0	1	0	✓
4	0	1	0	0	✓
3	0	0	1	1	✓
5	0	1	0	1	✓
10	1	0	1	0	✓
12	1	1	0	0	✓
7	0	1	1	1	✓
14	1	1	1	0	✓
15	1	1	1	1	

	a	*b*	*c*	*d*
(2, 3)	0	0	1	–
(2, 10)	–	0	1	0
(4, 5)	0	1	0	–
(4, 12)	–	1	0	0
(3, 7)	0	–	1	1
(5, 7)	0	1	–	1
(10, 14)	1	–	1	0
(12, 14)	1	1	–	0

Set $i = 3$. Pick up terms with index 3 and 4 and repeat step 5.

	a	b	c	d	
2	0	0	1	0	✓
4	0	1	0	0	✓
3	0	0	1	1	✓
5	0	1	0	1	✓
10	1	0	1	0	✓
12	1	1	0	0	✓
7	0	1	1	1	✓
14	1	1	1	0	✓
15	1	1	1	1	

	a	b	c	d	
(2, 3)	0	0	1	–	Q
(2, 10)	–	0	1	0	R
(4, 5)	0	1	0	–	S
(4, 12)	–	1	0	0	T
(3, 7)	0	–	1	1	U
(5, 7)	0	1	–	1	V
(10, 14)	1	–	1	0	W
(12, 14)	1	1	–	0	X
(7, 15)	–	1	1	1	Y
(14,15)	1	1	1	–	Z

Step 6: Repeat steps 4 and 5 on the new list. We see that no new list can be generated.

Step 7: Terminate the process as no new lists are formed.

Step 8: The prime implicants are the terms without a '✓'.

Let us now construct the prime implicant table with minterms along the horizontal axis and prime implicants labeled Q to Z along the vertical axis. Observe that the cost of every prime implicant is 4 as there are three variables in each term for a two level gate implementation.

Let us now construct the prime implicant table.

		m_3 $\bar{a}\bar{b}cd$	m_4 $\bar{a}b\bar{c}\bar{d}$	m_5 $\bar{a}b\bar{c}d$	m_7 $\bar{a}bcd$	m_{10} $a\bar{b}c\bar{d}$	m_{12} $ab\bar{c}\bar{d}$	m_{14} $abc\bar{d}$	m_{15} $abcd$	Cost
Q	$\bar{a}\bar{b}c$	×								4
R	$\bar{b}c\bar{d}$					×				4
S	$\bar{a}b\bar{c}$		×	×						4
T	$b\bar{c}\bar{d}$		×				×			4
U	$\bar{a}cd$	×			×					4
V	$\bar{a}bd$			×	×					4
W	$ac\bar{d}$					×		×		4
X	$ab\bar{d}$						×	×		4
Y	bcd				×				×	4
Z	abc							×	×	4

Note that m_2 does not appear in the prime implicant table as it is a don't care term. Place a '×' mark below each minterm against every row whose prime implicant is subsumed by the minterm.

Let us now apply column–row reduction to this table by looking of equals and dominations.

Row U dominates Row Q – delete Row Q as costs are equal.

Row W dominates Row R – delete Row R as costs are equal.

The table reduces to the following form

	m_3	m_4	m_5	m_7	m_{10}	m_{12}	m_{14}	m_{15}	Cost
S		×	×						4
T		×				×			4
U	×			×					4
V			×	×					4
W					×		×		4
X						×	×		4
Y				×				×	4
Z							×	×	4

Columns m_3 and m_{10} have only one '×' each. The corresponding rows row U and row W are essential rows and prime implicants U and W figure in the minimal sum.

Minimal sum = $U + W + \ldots\ldots$

Delete row U and row W and column m_3 and m_{10}.

Delete column m_7 as it has a '×' in row U.

Delete column m_{14} as it has a '×' in row W.

The table now becomes

	m_4	m_5	m_{12}	m_{15}	Cost
S	×	×			4
T	×		×		4
V		×			4
X			×		4
Y				×	4
Z				×	4

Rows Y and Z are equal and hence delete row Z costs are also equal.

Row S dominates row V – delete row V as costs are same.

Row T dominates row X – delete row X as costs are same.

∴ The table becomes

	m_4	m_5	m_{12}	m_{15}	Cost
S	×	×			4
T	×		×		4
Y				×	4

Columns m_5, m_{12} and m_{15} have a single '×'. Hence rows S, T and Y are selected for minimal sum.
∴ Minimal sum = $U + W + S + T + Y + \ldots$
Delete columns m_5, m_{12} and m_{15}. Delete column m_4 as it has a × along row S.
∴ The minimal sum is

$$f = S + T + U + W + Y$$

$$f = \bar{a}b\bar{c} + b\bar{c}\bar{d} + \bar{a}cd + ac\bar{d} + bcd$$

◆ **Exercise 2.5**

Find a minimal sum for the following incomplete Boolean functions using Quine–McCluskey method and prime implicant table reduction.

1. $f(a, b, c, d) = \Sigma(0, 1, 2, 4, 6, 7, 9, 11, 12, 13, 15)$
2. $f(a, b, c, d) = a\bar{b} + c\bar{d} + \bar{c}$

2.9 Decimal Notation for obtaining Prime Implicants

Decimal notation can be used all through the prime implicant table. The terms which figure in the irredundant expression can be obtained in algebraic form finally from the results in decimal notation.

Let us take the function in Example 2.33 to illustrate this procedure

$$f(a, b, c, d) = \Sigma(7, 9, 12, 13, 14, 15) + \Sigma d(4, 11)$$

Since don't care entries are taken as 1s in determining prime implicants, for this purpose the function becomes.

$$f(a, b, c, d) = \Sigma(4, 7, 9, 11, 12, 13, 14, 15)$$

The decimal numbers of minterms are placed in groups depending on their index value. The index values of the minterms are shown in Fig. 2.23.

Minterm	Index
4	1
7	3
9	2
11	3
12	2
13	3
14	3
15	4

Fig. 2.23 Index value of minterms

Minterms arranged according to index value are shown in Fig. 2.24.

Minterm	Index
4	index 1
9	index 2
12	
7	
11	
13	index 3
14	
15	index 4

Fig. 2.24 Minterms grouped by index values

These minterms grouped according to their index values are placed in the first list as shown in Fig. 2.25. A line is drawn after each group.

4	✓	4, 12 (8)
9		
12	✓	
7		
11		
13		
14		
15		

Fig. 2.25 Decimal numbers grouped by their index

Now pick up two decimal numbers in adjacent groups. They combine to form a single term if the two decimal numbers differ by powers of 2 *i.e.*, if they differ by 1, 2, 4, 8 etc. If two decimal numbers differ by powers of 2, place a '✓' against these numbers and place the combination in a new list with the amount they differ by in parenthesis. Consider index 1 and index 2. For example, 4 and 12 differ by 2^3 or 8, so place 4, 12(8) in the new list as shown in Fig. 2.24 and place a '✓' beside 4 and 12 in the first list.

Now pick up decimal numbers with index values 2 and 3 and check if they differ by powers of 2. The results are shown in Fig. 2.26.

4	✓	4, 12 (8)
9	✓	9, 11 (2)
12	✓	9, 13 (4)
7		12, 13 (1)
11	✓	12, 14 (2)
13	✓	
14	✓	
15		

Fig. 2.26 Decimal table after index = 2 and 3

Now pick up numbers of index 3 and 4 check if they differ by powers of 2.

4	✓	4, 12 (8)
9	✓	9, 11 (2)
12	✓	9, 13 (4)
7	✓	12, 13 (1)
11	✓	12, 14 (2)
13	✓	7, 15 (8)
14	✓	11, 15 (4)
15	✓	13, 15 (2)
		14,15 (1)

Fig. 2.27 Decimal table after index = 3 and 4

In the second list two terms in adjacent groups are eligible to combine only if the number in the parenthesis matches. Let the two terms be a, b (c) and d, e (f). These two terms can combine only if.

1. $c = f$
2. $d - a$ and $e - b$ are positive and powers of 2.

Thus, 4, 12 (8) in the first group in the second list cannot pair with any term in the second group of the second list. Among the second and third group 9, 11(2) pairs with 13, 15(2). Place this single term in the third list as shown in Fig. 2.28.

4	✓	4, 12 (8)		9,11,13,15 (2, 4)
9	✓	9, 11 (2)	✓	12,13,14,15 (1, 2)
12	✓	9, 13 (4)	✓	
7	✓	12, 13 (1)	✓	
11	✓	12, 14 (2)	✓	
13	✓	7, 15 (8)		
14	✓	11, 15 (4)	✓	
15	✓	13, 15 (2)	✓	
		14, 15 (1)	✓	

Fig. 2.28 Generation of third list

The 4 next to 2 within the parenthesis in the first term in the third list is the amount by which the lower decimals of the two terms differ. The lower decimal of 9, 11 is 9 and the lower decimal of 13, 15 is 13. Hence, the lower decimals differ by 13 − 9 = 4.

9, 13 (4) combines with 11, 15 (4) to give 9, 11, 13, 15 (2, 4) which is already in the third list. Simply place a '✓' beside 9, 13(4) and 11, 15 (4).

12, 13 (1) and 14, 15 (1) combine to give 12, 13, 14, 15 (1, 2). 12, 14 (2) and 13, 15 (2) combine to give 12, 13, 14, 15 (1, 2) which is already entered in the third list.

Now we have to obtain the prime implicants from the list in Fig. 2.29 in algebraic form. Prime implicants correspond to terms which do not have a '✓' beside them. Consider 4, 12 (8).

The rules for transforming this to algebraic form is shown in Fig. 2.29.

Description	*a*	*b*	*c*	*d*
Binary weights	8	4	2	1
For position in paranthesis place dash	–			
For position of smallest minterm place 1	–	1		
For remaining positions place 0	–	1	0	0
Binary weights	8	4	2	1
For position in paranthesis place dash	–			
For position of smallest minterm place 1	–	1	1	1

Fig. 2.29 Algebraic conversion of 4, 12 (8) and 7, 15 (8)

Now consider 9, 11, 13, 15 (2, 4).

The rules for transforming this to algebraic form is shown in Fig. 2.30.

Description	a	b	c	d
Binary weights	8	4	2	1
Place dashes in position 2 and 4		–	–	
Place 1s in the smallest minterm (*i.e.*, 9) positions	1	–	–	1
Fill any blank spaces with 0s				

Fig. 2.30 Algebraic conversion of 9, 11, 13, 15 (2, 4)

Now consider 12, 13, 14 15 (1, 2)
The application of rules to transform this to algebraic form is shown in Fig. 2.31.

Description	a	b	c	d
Binary weights	8	4	2	1
Place dashes in position 1 and 2			–	–
Place 1s in the smallest minterm (*i.e.*, 12) positions	1	1	–	–

Fig. 2.31 Algebraic conversion of 12, 13, 14, 15 (1, 2)

Observe that these conversions match with the results obtained in Example 2.4.

EXAMPLE 2.12

Find a minimal sum for the following incomplete Boolean function using decimal Quine-McCluskey method and prime-implicant table reduction.

$$f = \Sigma(1, 3, 6, 8, 9, 10, 12, 14) + \Sigma d(7, 13)$$

***Solution*:** Don't care terms are taken as 1s while determining all the prime implicants.

$$f = \Sigma(1, 3, 6, 7, 8, 9, 10, 12, 13, 14)$$

First let us write the index of all the minterms.

Minterm	*Index*
1	1
3	2
6	2
7	3
8	1
9	2
10	2
12	2
13	3
14	3

Let us now write the Quine-McCluskey table in decimal notation.

1 ✓	1, 3 (2)
8 ✓	1, 9 (8)
3 ✓	8, 9 (1)
6	8, 10 (2)
9 ✓	8, 12 (4)
10 ✓	
12 ✓	
7	
13	
14	

The next stage is shown below

1 ✓	1, 3 (2)
8 ✓	1, 9 (8)
3 ✓	8, 9 (1)
6 ✓	8, 10 (2)
9 ✓	8, 12 (4)
10 ✓	3, 7 (4)
12 ✓	6, 7 (1)
7 ✓	6, 14 (8)
13 ✓	9, 13 (4)
14 ✓	10, 14 (4)
	12, 13 (1)
	12, 14 (2)

The next stage is shown below.

1 ✓	1, 3 (2)	*V*	8, 9, 12, 13 (1,4)	*T*
8 ✓	1, 9 (8)	*W*	8, 10, 12, 14 (2,4)	*U*
3 ✓	8, 9 (1)	✓		
6 ✓	8, 10 (2)	✓		
9 ✓	8, 12 (4)	✓		
10 ✓	3, 7 (4)	*X*		
12 ✓	6, 7 (1)	*Y*		
7 ✓	6, 14 (8)	*Z*		
13 ✓	9, 13 (4)	✓		
14 ✓	10, 14 (4)	✓		
	12, 13 (1)	✓		
	12, 14 (2)	✓		

The prime implicants are

	Decimal form	*Algebraic form* a b c d / 8 4 2 1				
		a	*b*	*c*	*d*	
		8	4	2	1	
T	8, 9, 12, 13 (1,4)	1	–	0	–	$a\bar{c}$
U	8, 19, 12, 14 (2,4)	1	–	–	0	$a\bar{d}$
V	1, 3 (2)	0	0	–	1	$\bar{a}\bar{b}d$
W	1, 9 (8)	–	0	0	1	$\bar{b}\bar{c}d$
X	3, 7 (4)	0	–	1	1	$\bar{a}cd$
Y	6, 7 (8)	0	1	1	–	$\bar{a}bc$
Z	6, 14 (8)	–	1	1	0	$bc\bar{d}$

Let us now write the prime implicant table, dropping the don't care terms.

	m_1 $\bar{a}\bar{b}\bar{c}d$	m_3 $\bar{a}\bar{b}cd$	m_6 $\bar{a}bc\bar{d}$	m_8 $a\bar{b}\bar{c}\bar{d}$	m_9 $a\bar{b}\bar{c}d$	m_{10} $a\bar{b}c\bar{d}$	m_{12} $ab\bar{c}\bar{d}$	m_{14} $abc\bar{d}$	*Cost*
T $a\bar{c}$				×	×		×		3
U $a\bar{d}$				×		×	×	×	3
V $\bar{a}\bar{b}d$	×	×							4
W $\bar{b}\bar{c}d$	×				×				4
X $\bar{a}cd$		×							4
Y $\bar{a}bc$			×						4
Z $bc\bar{d}$			×					×	4

There is only one '×' in column m_{10}. Hence $a\bar{d}$ or U is an essential implicant.
∴ Minimal sum = $\{U + \ldots\}$
Delete row U and column m_{10} as well as columns m_8, m_{12} and m_{14} which have a '×' along row U.
The table now reduces to

	m_1	m_3	m_6	m_9	*Cost*
T				×	3
V	×	×			4
W	×			×	4
X		×			4
Y			×		4
Z			×		4

Row W dominates row T, but row T cannot be removed because cost (T) < cost (W).
Rows Y and Z are equal and of same cost. Hence delete row Z.
Row V dominates row X and both are of same cost. Hence delete row X.
The table now becomes

	m_1	m_3	m_6	m_9	*Cost*
T				×	3
V	×	×			4
W	×			×	4
Y			×		4

There is one '×' in column m_3 and m_6. Hence, V and Y figure in the minimum sum.
Minimal sum = $\{U + V + Y + \ldots\}$
Delete columns m_6 and m_3. Delete column m_1 too as it has a '×' along row V.
∴ The table now becomes

	m_9	*Cost*
T	×	3
W	×	4

row T and row W are equal. Delete row W as it costs more.
∴ Minimal sum is

$$f = T + U + V + Y$$

or

$$f = a\bar{c} + a\bar{d} + \bar{a}\bar{b}d + \bar{a}bc$$

◆ **Exercise 2.6**

Find a minimal sum for the following Boolean functions using decimal Quine-McCluskey method and prime implicants table reduction.

1. $f(a, b, c, d) = \Sigma\,(1, 3, 5, 8, 9, 11, 12, 15)$
2. $f(a, b, c, d) = \Sigma\,(0, 3, 5, 7, 10, 11) + \Sigma d\,(4, 14)$

2.10 Map Entered Variables or *MEV*

We can see that Karnaugh maps are not easy to handle when the number of variables go beyond four. Map entered variables technique makes it possible to use smaller maps to handle a larger number of variables. For example, a four variable map can be used to solve a six variable minterm expression.

In the usual Karnaugh maps the cells either contained a minterm, maxterm or a don't care term. In a *MEV* Karnaugh map, the cells are also allowed to contain single variables or expressions. Let us illustrate this technique with an example. Consider the function $f(a, b, c) = \Sigma\,(0, 1, 4, 5, 7)$. Let us solve this using a two variable map and let c be the map entered variable.