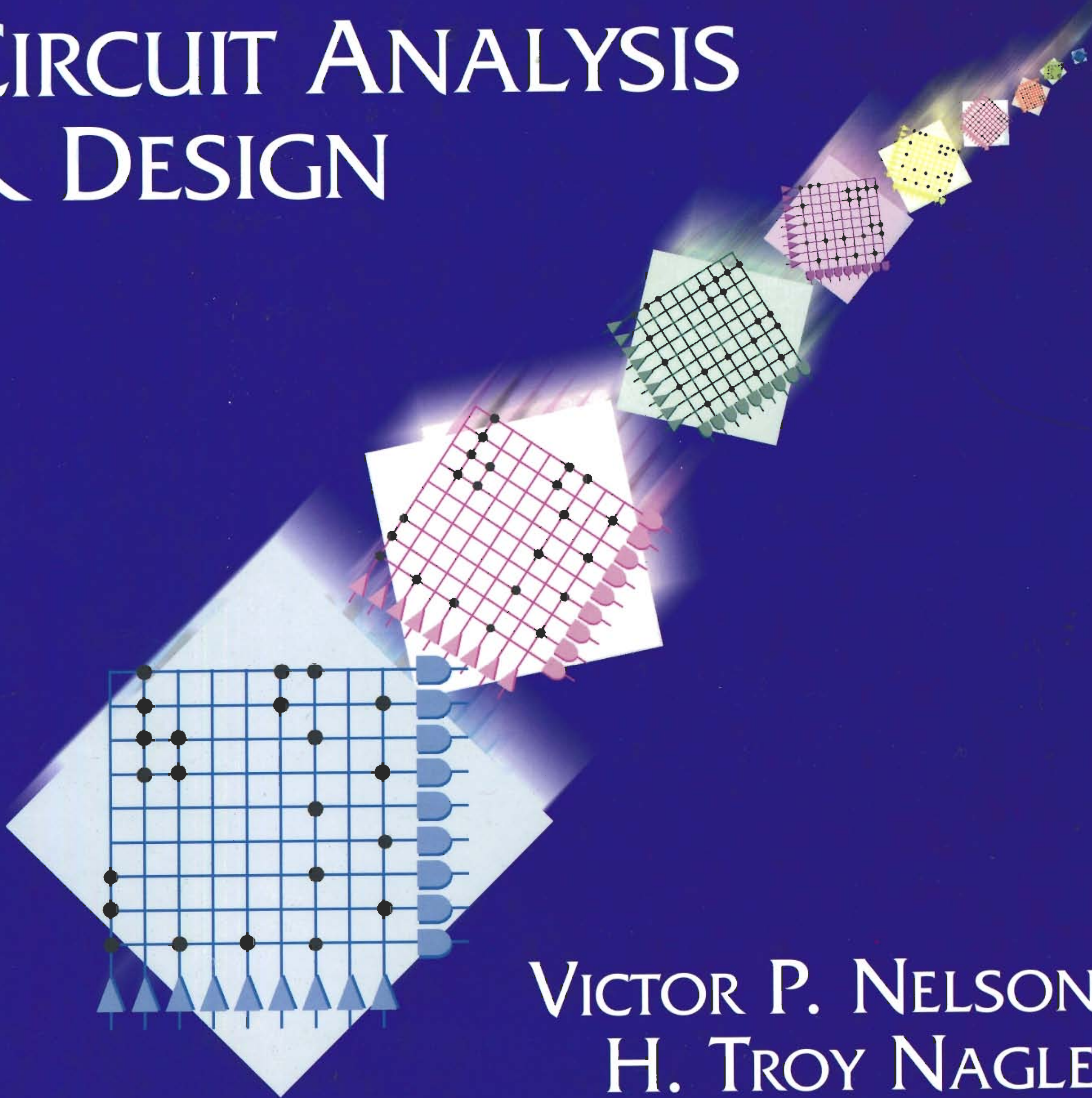
DIGITAL LOGIC CIRCUIT ANALYSIS & DESIGN



VICTOR P. NELSON
H. TROY NAGLE
J. DAVID IRWIN
BILL D. CARROLL

3.9 Quine–McCluskey Tabular Minimization Method

The *Quine–McCluskey (Q–M) method* is a tabular approach to Boolean function minimization [5, 6, 7]. Basically, the Q–M method has two advantages over the K-map. First, it is a straightforward, systematic method for producing a minimal function that is less dependent on the designer's ability to recognize patterns than the K-map method. Second, the method is a viable scheme for handling a large number of variables as opposed to the K-map, which, practically, is limited to about five or six variables. In general, the Q–M method performs an ordered linear search over the minterms in the function to find all combinations of logically adjacent minterms. As will be shown, the method can also be extended to functions with multiple outputs.

The Quine–McCluskey method begins with a list of the n-variable minterms of the function and successively derives all implicants with $n - 1$ variables, implicants with $n - 2$ variables, and so on, until all prime implicants are identified. A minimal covering of the function is then derived from the set of prime implicants. The four steps of the process are listed next. The exact meaning of each step will be illustrated by the examples that follow.

Step 1. List in a column all the minterms of the function to be minimized in their binary representation. Partition them into groups according to the number of 1 bits in their binary representations. This partitioning simplifies identification of logically adjacent minterms since, to be logically adjacent, two minterms must differ in exactly one literal, and therefore the binary representation of one minterm must have either one more or one fewer 1 bit than the other.

Step 2. Perform an exhaustive search between neighboring groups for adjacent minterms and combine them into a column of $(n - 1)$-variable implicants, checking off each minterm that is combined. The binary representation of each new implicant contains a dash in the position of the eliminated variable. Repeat for each column, combining $(n - 1)$-varible implicants into $(n - 2)$-variable implicants, and so on, until no further implicants can be combined. Any term not checked off represents a prime implicant of the function, since it is not covered by a larger implicant. The final result is a list of prime implicants of the switching function.

Step 3. Construct a prime implicant chart that lists minterms along the horizontal and prime implicants along the vertical, with an $\times$ entry placed wherever a certain prime implicant (row) covers a given minterm (column).

Step 4. Select a minimum number of prime implicants that cover all the minterms of the switching function.

A complete example will now be presented that demonstrates these four steps.

EXAMPLE 3.24

Let us use the Q–M technique to minimize the function

$$f(A,B,C,D) = \sum m(2,4,6,8,9,10,12,13,15)$$

The K-map for this example is shown in Fig. 3.32, and the reader is encouraged to try his or her hand at obtaining a minimal function via the map method.

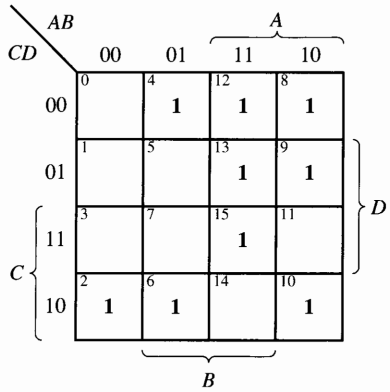


Figure 3.32 K-map for Example 3.30.

Step 1. To begin the Q–M minimization technique, the minterms are grouped according to the number of ones in the binary representation of the minterm number. This grouping of terms is illustrated in the following table:

Minterms	$ABCD$	
2	0010	
4	0100	Group 1 (a single 1)
8	1000	
6	0110	
9	1001	Group 2 (two 1's)
10	1010	
12	1100	
13	1101	Group 3 (three 1's)
15	1111	Group 4 (four 1's)

Step 2. Once this table has been formed, an exhaustive search for all combinations of logically adjacent terms is initiated. The method of performing this functional reduction is summarized here and explained in detail later. Consider the minimizing table shown next containing the three minterm lists. The two terms can be combined if and only if they differ in a single literal. Hence, in list 1 we can combine terms in group 1 only with those in group 2. When all the combinations between these two groups have been made and they have been entered in list 2, a line is drawn under these combinations, and we begin combining the terms in group 2 with those in group 3. This simple procedure is repeated from one list to another in order to generate the entire minimizing table.

List 1			List 2			List 3		
Minterm	$ABCD$		Minterms	$ABCD$		Minterms	$ABCD$	
2	0010	$\surd$	2, 6	0-10	PI_2	8, 9, 12, 13	1-0-	PI_1
4	0100	$\surd$	2, 10	-010	PI_3			
8	1000	$\surd$	4, 6	01-0	PI_4			
6	0110	$\surd$	4, 12	-100	PI_5			
9	1001	$\surd$	8, 9	100-	$\surd$			
10	1010	$\surd$	8, 10	10-0	PI_6			
12	1100	$\surd$	8, 12	1-00	$\surd$			
13	1101	$\surd$	9, 13	1-01	$\surd$			
15	1111	$\surd$	12, 13	110-	$\surd$			
			13, 15	11-1	PI_7			

There are a number of items in the table that beg for explanation. Note that the first element in list 2 indicates that minterms 2 and 6 have been combined since they differ in only a single literal. The terms differed in the variable B and hence a dash appears in that position in the combination 2, 6, indicating that variable B was eliminated when the two minterms were combined. This combination can easily be checked by Boolean algebra:

$$\text{minterm } 2 = \bar{A}\bar{B}C\bar{D}, \qquad \text{minterm } 6 = \bar{A}BC\bar{D}$$

and

$$\bar{A}\bar{B}C\bar{D} + \bar{A}BC\bar{D} = \bar{A}C\bar{D} \Rightarrow 0-10$$

Each minterm in list 1 that is combined with another is checked off with a $\surd$, indicating that it has been included in a larger set. Although a term may be combined more than once, it is only checked off once.

Once list 2 has been generated from list 1, an exhaustive search is made to combine the terms in list 2 to generate list 3. It is at this point that it becomes evident why it is important to indicate which of the variables has been eliminated. Since, as before, two terms in list 2 can be combined only if they differ in a single literal, only terms that have the same missing literal (a dash in the same position) can possibly be combined. Note that in list 2 minterm combinations 8, 12 and 9, 13 and also 8, 9 and 12, 13 can be combined to yield the combination 8, 9, 12, 13 in list 3.Inspection of list 2 shows that minterm combinations 8, 12 and 9, 13 both have the same missing literal and differ by one other literal. The same is true for the other combination. Hence all four terms are checked off in list 2 in the table. No other terms in list 2 in the table can be combined. Hence, all the terms that are not checked off in the entire table are prime implicants and are labeled $PI_1 \ldots PI_7$. The function could now be realized as a sum of all the prime implicants; however, we are looking for a minimal realization, and hence we want to use only the smallest number that is actually required.

A convenient way to check for errors in lists 2, 3, 4, and so on, is to perform the following test on each entry: subtract the minterm numbers to verify that the proper variables have been omitted. For example, the entry (4, 6 01-0) in list 2 indicates that the variable with weight $6 - 4 = 2$ should be eliminated. In this example, the possible weights are 8, 4, 2, and 1. For the entry in list 3 (8, 9, 12, 13 $\Rightarrow$ 1-0-):

$$9 - 8 = 1 \qquad\qquad 12 - 8 = 4$$
$$13 - 12 = 1 \qquad\qquad 13 - 8 = 4$$

so variables with weights 1 and 4 should be eliminated.

Step 3. To determine the smallest number of prime implicants required to realize the function we form a prime implicant chart as follows:

	2	4	6	√ 8	√ 9	10	√ 12	√ 13	√ 15
**PI_1				×	⊗		×	×	
PI_2	×		×						
PI_3	×					×			
PI_4		×	×						
PI_5		×					×		
PI_6				×		×			
**PI_7								×	⊗

The double horizontal line through the chart between PI_1 and PI_2 is used to separate prime implicants that contain different numbers of literals.

Step 4. An examination of the minterm columns in the prime implicant chart indicates that minterms 9 and 15 are each covered by only one prime implicant (shown circled). Therefore, prime implicants 1 and 7 must be chosen, and hence they are essential prime implicants (as indicated by the double asterisks). Note that in choosing these two prime implicants we have *also* covered minterms 8, 12, and 13. All five of the covered minterms are checked in the table; the checks are placed above the minterm numbers.

The problem now remaining is that of selecting as few additional (nonessential) prime implicants as are necessary to cover the minterms 2, 4, 6, and 10. In general, this is accomplished by forming a reduced prime implicant chart. This reduced chart is shown next; note that the chart contains only the minterms that remain to be covered and the remaining prime implicant candidates for inclusion in the cover.

	√ 2	√ 4	√ 6	√ 10
PI_2	×		×	
*PI_3	×			×
*PI_4		×	×	
PI_5		×		
PI_6				×

Which PIs should we select? Prime implicants PI_5 and PI_6 are obviously bad choices because they cover only one minterm, and that minterm is also covered by another PI that covers two minterms. Notice that the minterms 2, 4, 6, and 10 can be most efficiently covered (with the minimum number of prime implicants) by choosing PI_3 and PI_4. The single asterisk indicates our selection, and the checks above all the remaining minterms mean we have generated a complete cover. Therefore, a minimal realization of the original function would be

$$\begin{aligned} f(A, B, C, D) &= PI_1 + PI_3 + PI_4 + PI_7 \\ &= 1\text{-}0\text{-} + \text{-}010 + 01\text{-}0 + 11\text{-}1 \\ &= A\bar{C} + \bar{B}C\bar{D} + \bar{A}B\bar{D} + ABD \end{aligned}$$

The corresponding groupings of the minterms on the K-map are shown in Fig. 3.33.

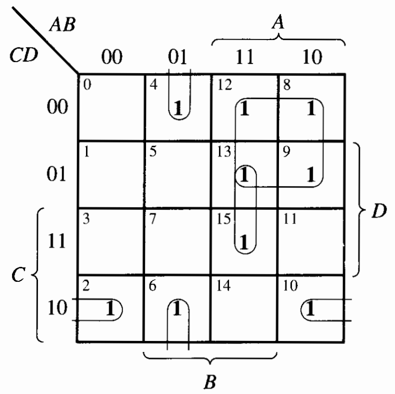


Figure 3.33 Grouping of terms.

3.9.1 Covering Procedure

The problem of selecting a minimum number of prime implicants to realize a switching function is sometimes called the *covering problem.* The following procedure may be employed to systematically choose a minimum number of nonessential prime implicants from the prime implicant chart.

The first step is to remove all essential prime implicant rows, as well as the minterm columns that they cover, from the chart, as in the last example. Then this reduced chart is further simplified as described next.

A row (column) i of a PI chart *covers* row (column) j if row (column) i contains an $\times$ in each column (row) in which j contains an $\times$. Each row represents a nonessential prime implicant PI_i, while each column represents a minterm m_i of the switching function. For example, consider the following PI chart for the switching function

$$f(A, B, C, D) = \sum m(0, 1, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15)$$

	√	√		√	√	√	√				√	√
	0	1	5	6	7	8	9	10	11	13	14	15
**PI_1	⊗	×				×	×					
PI_2		×	×				×			×		
PI_3			×		×					×		×
PI_4						×	×	×	×			
PI_5							×		×	×		×
PI_6								×	×		×	×
**PI_7				⊗	×						×	×

For this PI chart, PI_1 and PI_7 are essential PI and are marked with double asterisks. Now we remove these two rows as well as all columns in which the rows have × entries. The following reduced PI chart is generated:

	5	10	11	13
PI_2	×			×
PI_3	×			×
PI_4		×	×	
PI_5			×	×
PI_6		×	×	

According to the definition of row and column covering stated earlier, row PI_2 covers row PI_3 (and vice versa), row PI_4 covers row PI_6 (and vice versa), column 11 covers column 10, and column 13 covers column 5.

In view of the previous discussion, the rules for PI chart reduction can be stated as follows:

Rule 1. A row that *is covered* by another row may be eliminated from the chart. When identical rows are present, all but one of the rows may be eliminated. In the example, rows PI_3 and PI_6 may be eliminated.

Rule 2. A column that *covers* another column may be eliminated. All but one column from a set of identical columns may be eliminated. In the example, columns 11 and 13 can be eliminated.

If we apply these rules to the previous PI chart, the following reduced PI chart is obtained:

	√	√
	5	10
*PI_2	×	
*PI_4		×

Hence we may choose PI_2 and PI_4 along with the essential PI_1 and PI_7 to obtain a minimum cover for the switching function.

A type of PI chart that requires a special approach to accomplish reduction will now be discussed. A *cyclic* PI chart is a chart that contains no essential PI and that cannot be reduced by rules 1 and 2. An example of a cyclic chart is shown next for the switching function

$$f(A, B, C) = \sum m(1, 2, 3, 4, 5, 6)$$

	√		√			
	1	2	3	4	5	6
$*PI_1$	×		×			
PI_2		×	×			
PI_3		×				×
PI_4				×		×
PI_5				×	×	
PI_6	×				×	

Verify that no row or column covers another row or column. The procedure to follow for cyclic chart reduction is to arbitrarily select one PI from the chart. The row corresponding to this PI and the columns corresponding to the minterms covered by the PI are then removed from the chart. If the resulting reduced chart is not cyclic, then rules 1 and 2 may be applied. However, if another cyclic chart is produced, the procedure for a cyclic chart is repeated and another arbitrary choice is made. For example, arbitrarily choose PI_1 in the preceding cyclic chart. The following noncyclic chart is obtained by removing row PI_1 and columns 1 and 3.

	2	4	5	6
PI_2	×			
PI_3	×			×
PI_4		×		×
PI_5		×	×	
PI_6			×	

Rules 1 and 2 may now be applied to further reduce this chart. PI_3 covers row PI_2; hence row PI_2 may be removed. Row PI_5 covers row PI_6, so we can eliminate PI_6. The resulting reduced chart is

	√	√	√	√
	2	4	5	6
$*PI_3$	×			×
PI_4		×		×
$*PI_5$		×	×	

PI_3 and PI_5 must be chosen to cover the chart.

A minimum cover for the switching function is PI_1, PI_3, and PI_5. Other minimal covers also exist. The previous discussion can be summarized as follows:

Step 1. Identify any minterms covered by only one PI in the chart. Select these PIs for the cover. Note that this step identifies essential PIs on the first pass and nonessential PIs on subsequent passes (from step 4).

Step 2. Remove rows corresponding to the identified essential and nonessential PIs. Remove columns corresponding to minterms covered by the removed rows.

Step 3. If a cyclic chart results after completing step 2, go to step 5. Otherwise, apply the reduction procedure of rules 1 and 2.

Step 4. If a cyclic chart results from step 3, go to step 5. Otherwise, return to step 1.

Step 5. Apply the cyclic chart procedure. Repeat step 5 until a void chart occurs or until a noncyclic chart is produced. In the latter case, return to step 1.

The procedure terminates when step 2 or 5 produces a void chart. A void chart contains no rows or columns. On the first application of step 1, prime implicants are found that must be identified to cover minterms for which only one $\times$ appears in this column. They are identified by a double asterisk and are essential PIs. On the second and succeeding applications of step 1 (determined by step 4), nonessential prime implicants are identified by an asterisk from reduced PI charts.

3.9.2 Incompletely Specified Functions

The minimization of functions involving don't-cares proceeds exactly as shown in the preceding example with one important exception, which will be demonstrated by the next example.

EXAMPLE 3.25

We want to use the Q–M approach to minimize the function

$$f(A,B,C,D,E) = m(2,3,7,10,12,15,27) + d(5,18,19,21,23)$$

Following the procedure demonstrated in the preceding example, all the minterms *and don't-cares* are listed in the minimizing table and combined in the manner previously illustrated. The results of this procedure are shown in the following table:

List 1			List 2			List 3		
Minterm	*ABCDE*		Minterms	*ABCDE*		Minterms	*ABCDE*	
2	00010	√	2, 3	0001-	√	2, 3, 18, 19	-001-	PI_1
3	00011	√	2, 10	0-010	PI_4	3, 7, 19, 23	-0-11	PI_2
5	00101	√	2, 18	-0010	√	5, 7, 21, 23	-01-1	PI_3
10	01010	√	3, 7	00-11	√			
12	01100	PI_7	3, 19	-0011	√			
18	10010	√	5, 7	001-1	√			
7	00111	√	5, 21	-0101	√			
19	10011	√	18, 19	1001-	√			
21	10101	√	7, 15	0-111	PI_5			
15	01111	√	7, 23	-0111	√			
23	10111	√	19, 23	10-11	√			
27	11011	√	19, 27	1-011	PI_6			
			21, 23	101-1	√			

A prime implicant chart for the example must now be obtained. It is at this point that the method differs from that described earlier. Since some of the terms in list 1 are don't-cares, there is no need to cover them. Only the specified minterms must be covered, and thus they are the only minterms that appear in the prime implicant chart shown next. Do *not* list don't-cares in the PI chart.

It can be seen from the chart that the essential prime implicants are PI_4, PI_5, PI_6, and PI_7. Since only minterm 3 is not covered by the essential prime implicants, a reduced prime implicant chart is not necessary. Minterm 3 can be covered using PI_1 or PI_2, so there are two minimal covers for this function. The minimal realizations for the function are

$$f(A, B, C, D, E) = PI_1 + PI_4 + PI_5 + PI_6 + PI_7$$

or

$$f(A, B, C, D, E) = PI_2 + PI_4 + PI_5 + PI_6 + PI_7$$

	√		√	√	√	√	√
	2	3	7	10	12	15	27
PI_1	×	×					
PI_2		×	×				
PI_3			×				
$\ast\ast\mathrm{PI}_4$	×			⊗			
$\ast\ast\mathrm{PI}_5$			×			⊗	
$\ast\ast\mathrm{PI}_6$							⊗
$\ast\ast\mathrm{PI}_7$					⊗		

In terms of the variables,

$$f(A, B, C, D, E) = \bar{B}\bar{C}D + \bar{A}\bar{C}D\bar{E} + \bar{A}CDE + A\bar{C}DE + \bar{A}BC\bar{D}\bar{E}$$

or

$$f(A, B, C, D, E) = \bar{B}DE + \bar{A}\bar{C}D\bar{E} + \bar{A}CDE + A\bar{C}DE + \bar{A}BC\bar{D}\bar{E}$$

3.9.3 Systems with Multiple Outputs

In the design of digital systems, it is often necessary to implement more than one output function with some given set of input variables. Using the techniques developed thus far, the problem can be solved by treating each function individually. However, there exists a potential for sharing gates and thus obtaining a simpler and less expensive design.

The extension of the Q–M tabular method to the multiple-output case is performed like the singular case with the following exceptions:

1. To each minterm we must affix a flag to identify the function in which it appears.
2. Two terms (or minterms) can be combined only if they both possess one or more common flags and the term that results from the combination carries only flags that are common to both minterms.
3. Each term in the minimizing table can be checked off only if all the flags that the term possesses appear in the term resulting from the combination.

EXAMPLE 3.26

Let us use the tabular method to obtain a minimum realization for the functions

$$f_\alpha(A,B,C,D) = \sum m(0,2,7,10) + d(12,15)$$

$$f_\beta(A,B,C,D) = \sum m(2,4,5) + d(6,7,8,10)$$

$$f_\gamma(A,B,C,D) = \sum m(2,7,8) + d(0,5,13)$$

Note that this example will also demonstrate a minimization with don't-cares present. The minimizing table is shown next.

Min term	List 1 $ABCD$	Flags		Min terms	List 2 $ABCD$	Flags		Min terms	List 3 $ABCD$	Flags	
0	0000	$\alpha\gamma$	√	0, 2	00-0	$\alpha\gamma$	PI_2	4, 5, 6, 7	01--	β	PI_1
2	0010	$\alpha\beta\gamma$	PI_{10}	0, 8	-000	γ	PI_3				
4	0100	β	√	2, 6	0-10	β	PI_4				
8	1000	$\beta\gamma$	PI_{11}	2, 10	-010	$\alpha\beta$	PI_5				
5	0101	$\beta\gamma$	√	4, 5	010-	β	√				
6	0110	β	√	4, 6	01-0	β	√				
10	1010	$\alpha\beta$	√	8, 10	10-0	β	PI_6				
12	1100	α	PI_{12}	5, 7	01-1	$\beta\gamma$	PI_7				
7	0111	$\alpha\beta\gamma$	PI_{13}	5, 13	-101	γ	PI_8				
13	1101	γ	√	6, 7	011-	β	√				
15	1111	α	√	7, 15	-111	α	PI_9				

Consider the combination 0, 8 in List 2. This term is generated for function $f_{\gamma}(A, B, C, D)$ from minterms 0 and 8 in list 1. Minterm 8 cannot be checked because its entire label $\beta\gamma$ is not included in the label for minterm 0. Minterm 0 has a check due to the term 0, 2 in List 2.

It is important to note at this point that although our minimizing tables thus far have had three lists, in general, the number of lists can be any integer less than or equal to $n + 1$, where n is the number of input variables for the switching function, or functions in the multiple-output case. The prime implicant chart for the minimizing table is shown next (remember, *no* don't-cares across the top):

	f_α				f_β			f_γ		
	√	√		√	√	√	√	√		
	0	2	7	10	2	4	5	2	7	8
**PI$_1$ β						⊗	×			
**PI$_2$ αγ	⊗	×						×		
PI$_3$ γ										×
PI$_4$ β					×					
**PI$_5$ αβ		×		⊗	×					
PI$_6$ β										
PI$_7$ βγ							×		×	
PI$_8$ γ										
PI$_9$ α			×							
PI$_{10}$ αβγ		×			×			×		
PI$_{11}$ βγ										×
PI$_{12}$ α										
PI$_{13}$ αβγ			×						×	

The chart illustrates that PI$_1$, PI$_2$, and PI$_5$ are essential prime implicants. The reduced prime implicant chart is shown next; note that all prime implicants covering only don't-cares have been omitted.

	f_α	f_γ	
	√	√	√
	7	7	8
*PI$_3$ γ			×
PI$_7$ βγ		×	
PI$_9$ α	×		
PI$_{11}$ βγ			×
*PI$_{13}$ αβγ	×	×	

It is obvious that the best set of remaining prime implicants is PI_3 and PI_{13}. We choose PI_3 rather than PI_{11} because it has fewer literals. Hence the minimum realizations for the three functions are

$$f_\alpha = PI_2 + PI_5 + PI_{13}$$
$$f_\beta = PI_1 + PI_5$$
$$f_\gamma = PI_2 + PI_3 + PI_{13}$$

or

$$f_\alpha = \bar{A}\bar{B}\bar{D} + \bar{B}C\bar{D} + \bar{A}BCD$$
$$f_\beta = \bar{A}B + \bar{B}C\bar{D}$$
$$f_\gamma = \bar{A}\bar{B}\bar{D} + \bar{B}\bar{C}\bar{D} + \bar{A}BCD$$

It is important to note that PI_2, PI_5, and PI_{13} are generated only once, but are used to implement two of the functions, as shown in Fig. 3.34.

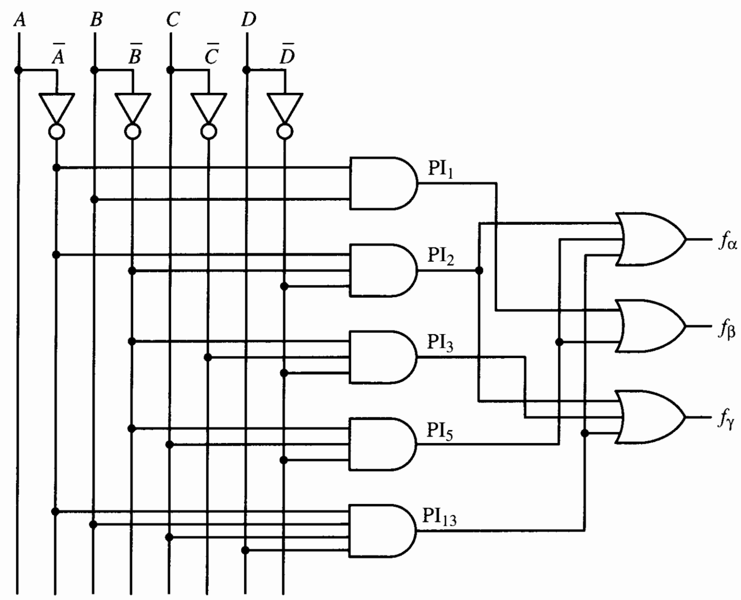


Figure 3.34 Reduced multiple-output circuit.

3.10 Petrick's Algorithm

As stated previously, the methods presented in the previous section for selecting a minimum cover are heuristic and therefore not guaranteed to find an optimum solution. In particular, the final steps of K-map Algorithms 3.2 and 3.4 and the final step of the Quine–McCluskey method all rely on heuristics and the talent of the designer to identify a minimum set of prime implicants to complete a cover after the essential prime implicants have been found. Often trial and error is used to identify and evaluate multiple possible covers.