Chapter 4

Examples of Solved Problems

This section presents some typical problems that the student may encounter, and shows how such problems can be solved.

Example 4.1

Problem: Determine the minimum-cost SOP and POS expressions for the function $f(x_1, x_2, x_3, x_4) = \sum m(4, 6, 8, 10, 11, 12, 15) + D(3, 5, 7, 9)$.

Solution: The function can be represented in the form of a Karnaugh map as shown in Figure 4.1*a*. Note that the location of minterms in the map is as indicated in Figure 4.6 (in the text book). To find the minimum-cost SOP expression, it is necessary to find the prime implicants that cover all 1s in the map. The don't-cares may be used as desired. Minterm m_6 is covered only by the prime implicant $\overline{x}_1 x_2$, hence this prime implicant is essential and it must be included in the final expression. Similarly, the prime implicants $x_1\overline{x}_2$ and $x_3 x_4$ are essential because they are the only ones that cover m_{10} and m_{15}, respectively. These three prime implicants cover all minterms for which $f = 1$ except m_{12}. This minterm can be covered in two ways, by choosing either $x_1\overline{x}_3\overline{x}_4$ or $x_2\overline{x}_3\overline{x}_4$. Since both of these prime implicants have the same cost, we can choose either of them. Choosing the former, the desired SOP expression is

$$f = \overline{x}_1 x_2 + x_1\overline{x}_2 + x_3 x_4 + x_1\overline{x}_3\overline{x}_4$$

These prime implicants are encircled in the map.

The desired POS expression can be found as indicated in Figure 4.1*b*. In this case, we have to find the sum terms that cover all 0s in the function. Note that we have written 0s explicitly in the map to emphasize this fact. The term $(x_1 + x_2)$ is essential to cover the 0s in squares 0 and 2, which correspond to maxterms M_0 and M_2. The terms $(x_3 + \overline{x}_4)$ and $(\overline{x}_1 + \overline{x}_2 + \overline{x}_3 + x_4)$ must be used to cover the 0s in squares 13 and 14, respectively. Since these three sum terms cover all 0s in the map, the POS expression is

$$f = (x_1 + x_2)(x_3 + \overline{x}_4)(\overline{x}_1 + \overline{x}_2 + \overline{x}_3 + x_4)$$

The chosen sum terms are encircled in the map.

Observe the use of don't-cares in this example. To get a minimum-cost SOP expression we assumed that all four don't-cares have the value 1. But, the minimum-cost POS expression becomes possible only if we assume that don't-cares 3, 5, and 9 have the value 0 while the don't-care 7 has the value 1. This means that the resulting SOP and POS expressions are not identical in terms of the functions they represent. They cover identically all valuations for which f is specified as 1 or 0, but they differ in the valuations 3, 5, and 9. Of course, this difference does not matter, because the don't-care valuations will never be applied as inputs to the implemented circuits.

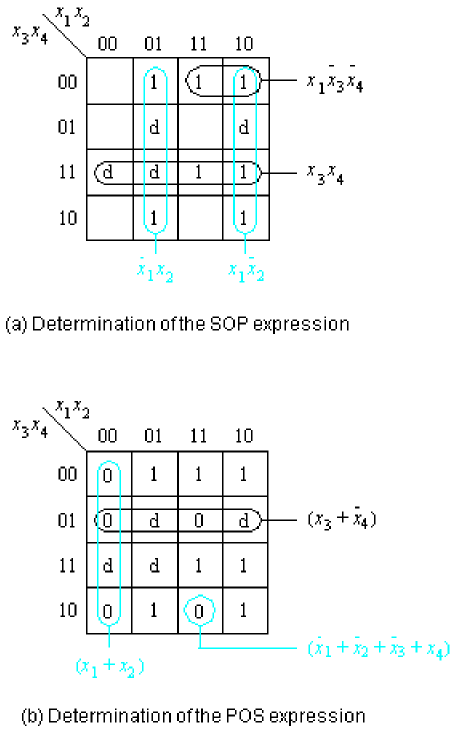


Figure 4.1. Karnaugh maps for Example 4.1.

Example 4.2

Problem: Use Karnaugh maps to find the minimum-cost SOP and POS expressions for the function

$$f(x_1, \ldots, x_4) = \overline{x}_1\overline{x}_3\overline{x}_4 + x_3x_4 + \overline{x}_1\overline{x}_2x_4 + x_1x_2\overline{x}_3x_4$$

assuming that there are also don't-cares defined as $D = \sum(9, 12, 14)$.

Solution: The Karnaugh map that represents this function is shown in Figure 4.2*a*. The map is derived by placing 1s that correspond to each product term in the expression used to specify f. The term $\overline{x}_1\overline{x}_3\overline{x}_4$ corresponds to minterms 0 and 4. The term x_3x_4 represents the third row in the map, comprising minterms 3, 7, 11, and 15. The term $\overline{x}_1\overline{x}_2x_4$ specifies minterms 1 and 3. The fourth product term represents the minterm 13. The map also includes the three don't-care conditions.

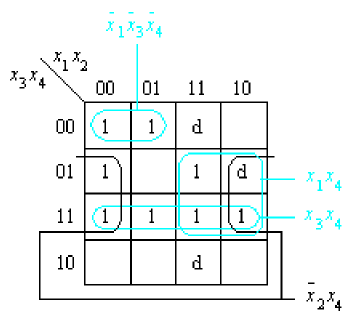
To find the desired SOP expression, we must find the least-expensive set of prime implicants that covers all 1s in the map. The term x_3x_4 is a prime implicant which must be included because it is the only prime implicant that covers the minterm 7; it also covers minterms 3, 11, and 15.

Minterm 4 can be covered with either $\overline{x}_1\overline{x}_3\overline{x}_4$ or $x_2\overline{x}_3\overline{x}_4$. Both of these terms have the same cost; we will choose $\overline{x}_1\overline{x}_3\overline{x}_4$ because it also covers the minterm 0. Minterm 1 may be covered with either $\overline{x}_1\overline{x}_2\overline{x}_3$ or $\overline{x}_2 x_4$; we should choose the latter because its cost is lower. This leaves only the minterm 13 to be covered, which can be done with either x_1x_4 or x_1x_2 at equal costs. Choosing x_1x_4, the minimum-cost SOP expression is

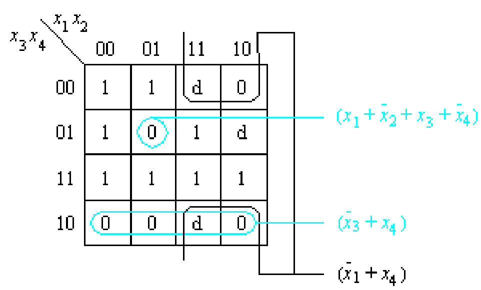
$$f = x_3x_4 + \overline{x}_1\overline{x}_3\overline{x}_4 + \overline{x}_2x_4 + x_1x_4$$

Figure 4.2*b* shows how we can find the POS expression. The sum term $(\overline{x}_3 + x_4)$ covers the 0s in the bottom row. To cover the 0 in square 8 we must include $(\overline{x}_1 + x_4)$. The remaining 0, in square 5, must be covered with $(x_1 + \overline{x}_2 + x_3 + \overline{x}_4)$. Thus, the minimum-cost POS expression is

$$f = (\overline{x}_3 + x_4)(\overline{x}_1 + x_4)(x_1 + \overline{x}_2 + x_3 + \overline{x}_4)$$



(a) Determination of the SOP expression



(b) Determination of the POS expression

Figure 4.2. Karnaugh maps for Example 4.2.

Example 4.3

Problem: Find the minimum-cost implementation for the function

$$f(x_1, \ldots, x_4) = \overline{x}_1\overline{x}_3\overline{x}_4 + x_3x_4 + \overline{x}_1\overline{x}_2x_4 + x_1x_2\overline{x}_3x_4$$

assuming that there are also don't-cares defined as $D = \sum(9, 12, 14)$.

Solution: This is the same function used in Examples 4.2, where we found that the minimum-cost SOP implementation is

$$f = x_3x_4 + \overline{x}_1\overline{x}_3\overline{x}_4 + \overline{x}_2x_4 + x_1x_4$$

which requires four AND gates, one OR gate, and 13 inputs to the gates, for a total cost of 18.

The minimum-cost POS implementation is

$$f = (\overline{x}_3 + x_4)(\overline{x}_1 + x_4)(x_1 + \overline{x}_2 + x_3 + \overline{x}_4)$$

which requires three OR gates, one AND gate, and 11 inputs to the gates, for a total cost of 15.

We can also consider a multilevel realization for the function. Applying the factoring concept to the above SOP expression yields

$$f = (x_1 + \overline{x}_2 + x_3)x_4 + \overline{x}_1\overline{x}_3\overline{x}_4$$

This implementation requires two AND gates, two OR gates, and 10 inputs to the gates, for a total cost of 14. Compared to the SOP and POS implementations, this has the lowest cost in terms of gates and inputs, but it results in a slower circuit because there are three levels of gates through which the signals must propagate.