

		z				
		1	0	0	-	
		1	1	0	0	
		0	0	1	1	
w		-	1	1	0	x
		y				

(c) $E(x, y, z) = \sum m(0, 1, 4, 6) = \prod M(2, 3, 5, 7)$

		z				
		1	1	0	0	
x		1	0	0	1	
		y				

Exercise 5.4 $f(w, x, y, z) = \sum m(0, 1, 2, 3, 5, 7, 8, 10, 11, 15)$

(a) Kmap and prime implicants

		z				
		1	1	1	1	
		0	1	1	0	
		0	0	1	0	
w		1	0	1	1	x
		y				

prime implicants: $w'x', w'z, x'y, yz, x'z'$

(b) essential prime implicants: $w'z, yz, x'z'$

(c) The minimal sum of products for this function is unique:

$$f(w, x, y, z) = w'z + yz + x'z'$$

Exercise 5.5

(a) More implicants than minterms

		z				
		0	1	1	0	
x		0	0	0	0	
		y				

implicants: $x'yz, x'y'z$, and $x'z$

minterms: $x'yz$ and $x'y'z$

(b) Equal number of implicants and minterms

(b) $E(w, x, y, z) = \sum m(0, 4, 5, 9, 11, 14, 15)$, $dc(w, x, y, z) = \sum m(2, 8)$

5.6b

				z																							
w				<table><tr><td>1</td><td>0</td><td>0</td><td>-</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>-</td><td>1</td><td>1</td><td>0</td></tr></table>				1	0	0	-	1	1	0	0	0	0	1	1	-	1	1	0	x			
								1	0	0	-																
								1	1	0	0																
								0	0	1	1																
-	1	1	0																								
				y																							

minimal SP: $w'y'z' + w'y'x + wx'z + wxy$

				z							
w	1	0	0	-					x		
	1	1	0	0							
	0	0	1	1							
	-	1	1	0							
				y							

minimal PS: $(w+x+z')(w+y')(x+y'+z)(w'+x'+y)$

(c) $E(x, y, z) = \sum m(0, 1, 4, 6) = \prod M(2, 3, 5, 7)$

z				

x	1	1	0	0
	1	0	0	1
y				

minimal sum of products: $x'y' + xz'$

z				
x	1	1	0	0
	1	0	0	1
y				

minimal product of sums: $(x + y')(x' + z')$

Exercise 5.7

$$f(w, x, y, z) = one_set(1, 5, 7, 8, 9, 10, 14)$$

- (a) prime implicants are:
 $(w + z), (w + x + y'), (x + y' + z'), (w' + y' + z'), (w' + x' + z'), (w' + x' + y), (x' + y + z)$
 (b) essential prime implicate is: $(w + z)$
 (c) a minimal product of sums expression that implements $f(w, x, y, z)$ is:

$$E(w, x, y, z) = (w + z)(x + y' + z')(w' + x' + z')(x' + y + z)$$

the solution is not unique because there are other ways to cover the 0-cells (not covered by the essential prime implicate) with the same number of terms.

Exercise 5.8

Input: (a, b, c, d) , with $a, b, c, d \in \{0, 1\}$

Output: $y \in \{0, 1\}$

Function:

$$y = \begin{cases} 1 & \text{if } (8a + 4b + 2c + d) \text{ is prime} \\ 0 & \text{otherwise} \end{cases}$$

input value	abcd	y
0	0000	0
1	0001	0
2	0010	1
3	0011	1
4	0100	0
5	0101	1
6	0110	0
7	0111	1
8	1000	0
9	1001	0
10	1010	0
11	1011	1
12	1100	0
13	1101	1
14	1110	0
15	1111	0

		d			
		0	0	1	1
		0	1	1	0
		0	1	0	0
		0	0	1	0
		c			
a	b				

From the Kmap we get the following prime implicants: $a'bd$, $b'cd$, $a'b'c$, $a'cd$, and $bc'd$

The essential prime implicants are: $b'cd$, $a'b'c$, and $bc'd$

A minimal sum of products for function y is:

$$y = b'cd + a'b'c + bc'd + a'cd$$

and the gate network that implements this expression is shown in Figure 5.1.

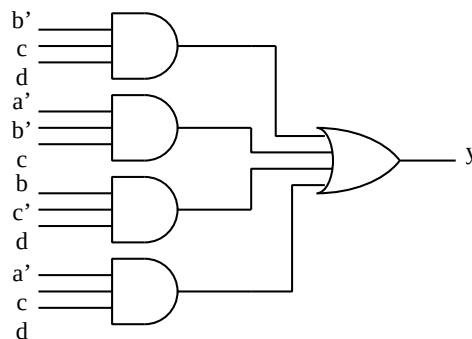


Figure 5.1: AND-OR gate network for “prime detector” (Exercise 5.8)

		d				
		0	0	1	1	
		0	1	1	0	
		0	1	0	0	
		0	0	1	0	
		c				
a	b					

From the Kmap we get the following prime implicants: $(b' + d)$, $(a' + d)$, $(b + c)$, $(c + d)$ and $(a' + b' + c')$. Only the $(c + d)$ prime implicate is not essential. So, the minimal product of sums in this case is:

$$y = (b' + d)(a' + d)(b + c)(a' + b' + c')$$

and the gate network that implements this expression is shown in Figure 5.2. Notice that the cost of the product of sums is lower.

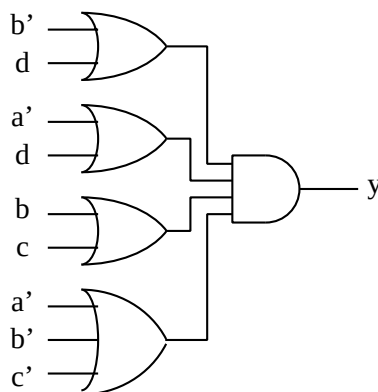


Figure 5.2: OR-AND gate network for “prime detector” (Exercise 5.8)

Exercise 5.9

To repeat the Exercise 5.8 using the Quine-McCluskey minimization method we make use of two tables. The first table is used to obtain the prime implicants and the second to select the minimum cover.

The one_set for this function is one_set = {2, 3, 5, 7, 11, 13}

Minterms	3-literal Prods	2-literal Prods	1-literal Prods
0010 N	001-		
0011 N	-101		
0101 N	-011		
	0-11		
0111 N	01-1		
1011 N			
1101 N			

The second table consider the prime-implicants obtained in the previous table:

Prime Implicants	2	3	5	7	11	13	Essential
001-	x	x					•
-101			x			x	•
-011		x			x		•
0-11		x		x			
01-1			x	x			

Based on the table, the function is represented by 3 essential terms and the minterm 7 must be covered with either 0 – 11 or 01 – 1. The minimal SP expressions are:

$$y = a'b'c + bc'd + b'cd + a'cd$$

or

$$y = a'b'c + bc'd + b'cd + a'bd$$

Exercise 5.10 A high-level description for this system is:

Input: $a, b, c, d, e \in \{0, 1\}$

Output: $f \in \{0, 1\}$

Function:

$$f = \begin{cases} 1 & \text{if } (a + b + c + d + e) \geq 3 \\ 0 & \text{otherwise} \end{cases}$$

Using Quine-McCluskey minimization method we obtain the following table:

Minterms	4-literal products	3-literal products
00111 N	0-111 N	--111
01011 N	-0111 N	-1-11
10011 N	01-11 N	1--11
01101 N	10-11 N	-11-1
10101 N	1-011 N	1-1-1
11001 N	011-1 N	11--1
01110 N	-1101 N	-111-
10110 N	101-1 N	1-11-
11010 N	1-101 N	11-1-
11100 N	110-1 N	111--
	11-01 N	
01111 N	0111- N	
10111 N	-1110 N	
11011 N	1011- N	
11101 N	1-110 N	
11110 N	1101- N	
	11-10 N	
11111 N	1110- N	
	111-0 N	
	-1111 N	
	1-111 N	
	11-11 N	
	111-1 N	
	1111- N	

To obtain a cover for the minterms we use a second table to identify the essential prime implicants, as shown below:

	7	11	13	14	15	19	21	22	23	25	26	27	28	29	30	31	Essential Prime
--111	x				x				x							x	•
-1-11		x			x							x				x	•
1--11						x			x			x				x	•
-11-1			x		x									x		x	•
1-1-1							x		x					x		x	•
11--1										x		x		x		x	•
-111-				x	x										x	x	•
1-11-								x	x						x	x	•
11-1-											x	x			x	x	•
111--													x	x	x	x	•

All the terms obtained in the first table are essential prime implicants. The minimal expression is:

$$f = abc + abd + abe + acd + ace + ade + bcd + bce + bde + cde$$

and the corresponding two-level gate network is shown in Figure 5.3 on page 63. Note that the OR gate has 10 inputs, which might make a two-level implementation impractical.

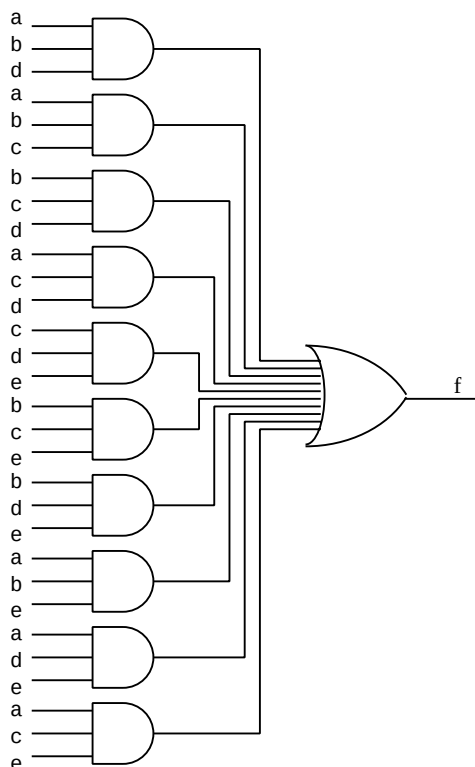


Figure 5.3: Majority function - Exercise 5.10

Exercise 5.11 A high-level specification for the error detector is:

Input: x is a 2-out-of-5 code represented as $\underline{x} = (a, b, c, d, e)$, where $a, b, c, d, e \in \{0, 1\}$.

Output: $f \in \{0, 1\}$.

Function:

$$f = \begin{cases} 0 & \text{if the number of 1s in the input is 2} \\ 1 & \text{otherwise} \end{cases}$$

The synthesis of this function using Quine-McCluskey minimization method is shown in Table 5.1. The obtained 3-literal products generate a Prime-implicant chart similar to the one shown in Exercise 5.10, where it was concluded that all products are essential. The minimal sum of products is:

$$\begin{aligned} f = & a'b'c'd' + abc + abd + acd + bcd + abe + ace + ade + bce + bde + cde \\ & + a'b'c'e' + a'b'd'e' + a'c'd'e' + b'c'd'e' \end{aligned}$$

The gate network that implements the function f is shown in Figure 5.4. Note that the OR gate has 15 inputs, which might make a two-level implementation impractical.

Exercise 5.13Input: x in the range $[0, 15]$ Output: y in the range $[0, 7]$ Function: $y = x \bmod 7$

The switching functions are shown in the table

$x = (x_3x_2x_1x_0)$	$y = (y_2y_1y_0)$
0000	000
0001	001
0010	010
0011	011
0100	100
0101	101
0110	110
0111	000
1000	001
1001	010
1010	011
1011	100
1100	101
1101	110
1110	000
1111	001

From K-maps (not shown) we obtain the following minimal switching expressions

$$\begin{aligned}
y_2 &= (x_3 + x_2)(x'_2 + x'_1 + x'_0)(x'_3 + x'_1 + x_0)(x_2 + x_1) \\
&= x_2x'_1 + x_3x'_2x_1x_0 + x'_3x_2x'_0 \\
y_1 &= (x_3 + x_1)(x_1 + x_0)(x_3 + x'_2 + x'_0)(x'_3 + x'_2 + x'_1)(x'_3 + x'_1 + x'_0) \\
&= x_3x'_1x_0 + x'_3x'_2x_1 + x'_3x_1x'_0 + x'_2x_1x'_0 \\
y_0 &= (x_3 + x_0)(x'_3 + x_2 + x'_0)(x'_3 + x_1 + x'_0)(x'_2 + x'_1 + x_0)(x_3 + x'_2 + x'_1) \\
&= x_3x'_1x'_0 + x_3x'_2x'_0 + x_3x_2x_1x_0 + x'_3x'_1x_0 + x'_3x'_2x_0
\end{aligned}$$

Using the lowest cost version of the network that produces each output, we obtain the gate network shown in Figure 5.6.

Exercise 5.14

The multiplier is specified as follows:

Inputs: x, y where $x, y \in \{0, 1, 2, 3\}$ Output: $z \in \{0, 1, 2, 3, 4, 6, 9\}$ Function: $z = x \cdot y$

Coding the inputs and outputs in a binary code, produces the switching function of the following table: