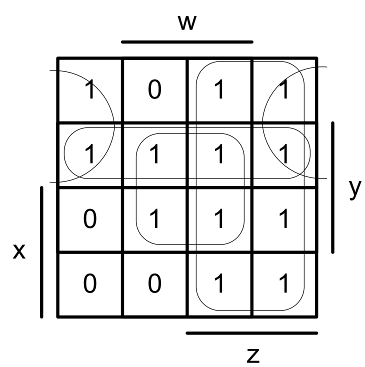
[CS M51A] Solution to sample chapter 5 hmw

Problem 1 (20 points)

For $f(x, y, z, w) = \prod M(1, 8, 9, 12)$

1. Using K-maps, find all the prime implicants.

   Solution The prime implicants are shown in the K-map.

   The equivalent product terms are z, yw, $x'w'$ and $x'y$.

2. Which of these prime implicants are essential?

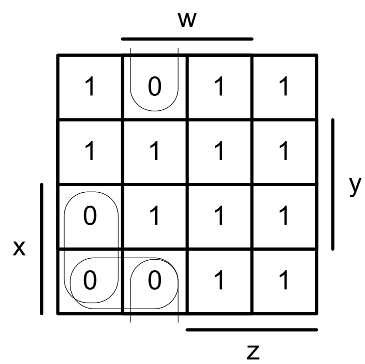
   Solution $x'y$ does not have any squares that it covers alone, therefore it is not essential.

3. Write the minimal sum of products for f.

   Solution Excluding the non-essential prime, we get $z + yw + x'w'$.

4. Find all the prime implicates.

   Solution



The equivalent sum terms are $(x' + z + w)$, $(y + z + w')$ and $(x' + y + z)$.

5. Which of these prime implicates are essential?

   Solution $(x' + y + z)$ does not have any squares that it covers alone, therefore it is not essential.

6. Write the minimal product of sums for f.

   Solution Excluding the non-essential prime, we get $(x' + z + w)(y + z + w')$.

Problem 2 (10 points)

We would like to examine how K-maps can be used to obtain minimal sum-of-product expressions. We are given the following expression.

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

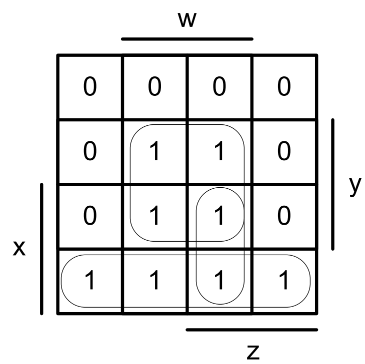
1. Using the given expression, fill in the table below.

   Solution

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

2. Draw the K-map for the table.

   Solution The K-map is shown here.



3. Identify the essential prime implicants. Is xzw essential? Why or why not?

   Solution From the above K-map, we can see that the two essential prime implicants are xy' and yw. xzw covers two squares, and since each one is also covered by the two prime implicants, it is not essential.

Problem 3 (20 points)

We are given a module with four input bits, x_1, x_0, y_1, y_0 and three output bits, z_2, z_1, z_0, which are:

$$\begin{aligned} z_2 &= \sum m(7, 10, 11, 13, 14, 15) \\ z_1 &= \sum m(2, 3, 5, 6, 8, 9, 12, 15) \\ z_0 &= \sum m(1, 3, 4, 6, 9, 11, 12, 14) \end{aligned}$$

1. Fill in the table.

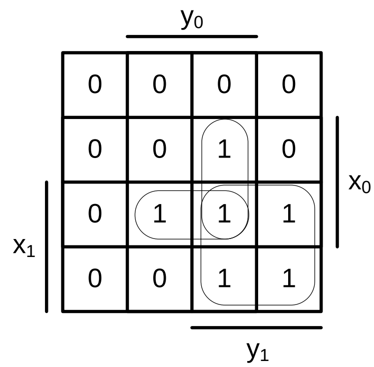
   Solution

x_1	x_0	y_1	y_0	z_2	z_1	z_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

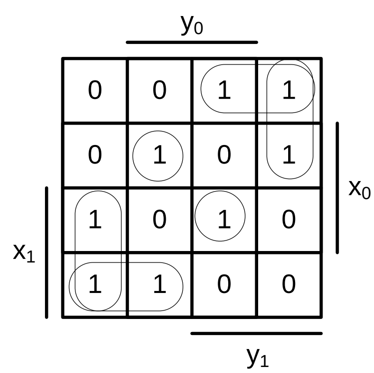
2. Draw K-maps for each output bit. Find the prime implicants.

   Solution The input bits are x_1, x_0, y_1, y_0 with x_1 being the most significant bit. The K-maps for each bit is as shown below.

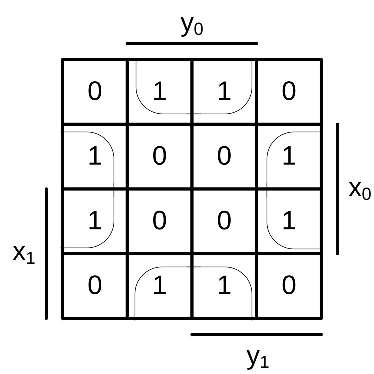
Prime implicants for z_2: x_1y_1, $x_1x_0y_0$, $x_0y_1y_0$



Prime implicants for z_1: $x_1y_1'y_0'$, $x_1x_0'y_1'$, $x_1'x_0y_1'y_0$, $x_1x_0y_1y_0$, $x_1'x_0'y_1$, $x_1'y_1y_0'$



Prime implicants for z_0: x_0y_0', $x_0'y_0$



3. Write the minimal sum of products expression for each z bit.

Solution All prime implicants for the outputs are essential here. Therefore the minimal SoPs are:

$$\begin{aligned} z_2 &= x_1y_1 + x_1x_0y_0 + x_0y_1y_0 \\ z_1 &= x_1y_1'y_0' + x_1x_0'y_1' + x_1'x_0y_1'y_0 + x_1x_0y_1y_0 + x_1'x_0'y_1 + x_1'y_1y_0' \\ z_0 &= x_0y_0' + x_0'y_0 \end{aligned}$$

4. Looking back at the table, can you identify the high-level function of the module?

   Solution Looking at the inputs as separate binary values of x and y, we can see that the resulting z is a sum of x and y. The module is an adder which adds the two inputs x and y.

Problem 4 (20 points)

We would like to design a BCD-to-Gray-Code converter. The Gray Code is a coding scheme where each number sequence differs from its predecessor by exactly one bit. The variation we use for this problem is for encoding a decimal digit. The actual code values used in this problem are shown in the table below.

Coded value	z_3	z_2	z_1	z_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	1	1	1	0
6	1	0	1	0
7	1	0	1	1
8	1	0	0	1
9	1	0	0	0

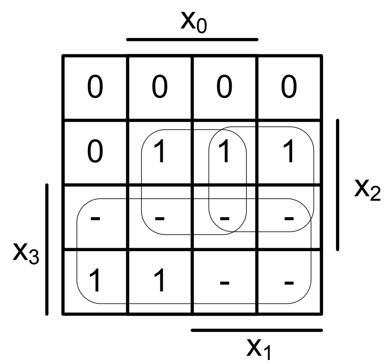
1. Write the truth table for the converter. How many input bits do we need? Do not forget don't-care cases.

   Solution To accomodate for digits from 0 to 9, we need four binary bits, which we will label x_3 downto x_0. The truth table is as shown:

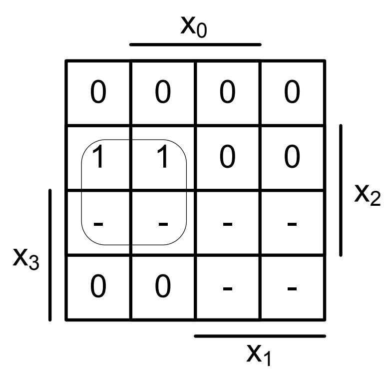
x_3	x_2	x_1	x_0	z_3	z_2	z_1	z_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	1	1	1	0
0	1	1	0	1	0	1	0
0	1	1	1	1	0	1	1
1	0	0	0	1	0	0	1
1	0	0	1	1	0	0	0
1	0	1	0	-	-	-	-
1	0	1	1	-	-	-	-
1	1	0	0	-	-	-	-
1	1	0	1	-	-	-	-
1	1	1	0	-	-	-	-
1	1	1	1	-	-	-	-

2. Draw the K-maps for each z bit. Find the prime implicants.

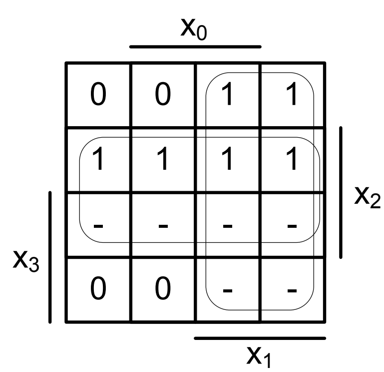
   Solution Prime implicants for z_3: x_3, x_2x_0, x_2x_1



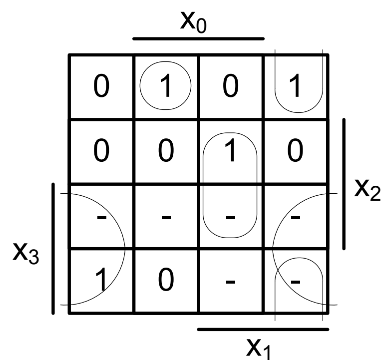
Prime implicants for z_2: x_2x_1'



Prime implicants for z_1: x_2, x_1



Prime implicants for z_0: $x_3'x_2'x_1'x_0$, x_3x_0', $x_2x_1x_0$, $x_2'x_1x_0'$



3. Write the minimal sum of products for each z bit. How many gates would we need for an AND-OR network implementation?

Solution

$$
\begin{aligned}
z_3 &= x_3 + x_2x_0 + x_2x_1 \\
z_2 &= x_2x_1' \\
z_1 &= x_2 + x_1 \\
z_0 &= x_3'x_2'x_1'x_0 + x_3x_0' + x_2x_1x_0 + x_2'x_1x_0'
\end{aligned}
$$

Assuming that the inverse of the input is available, the total number of gates needed is:

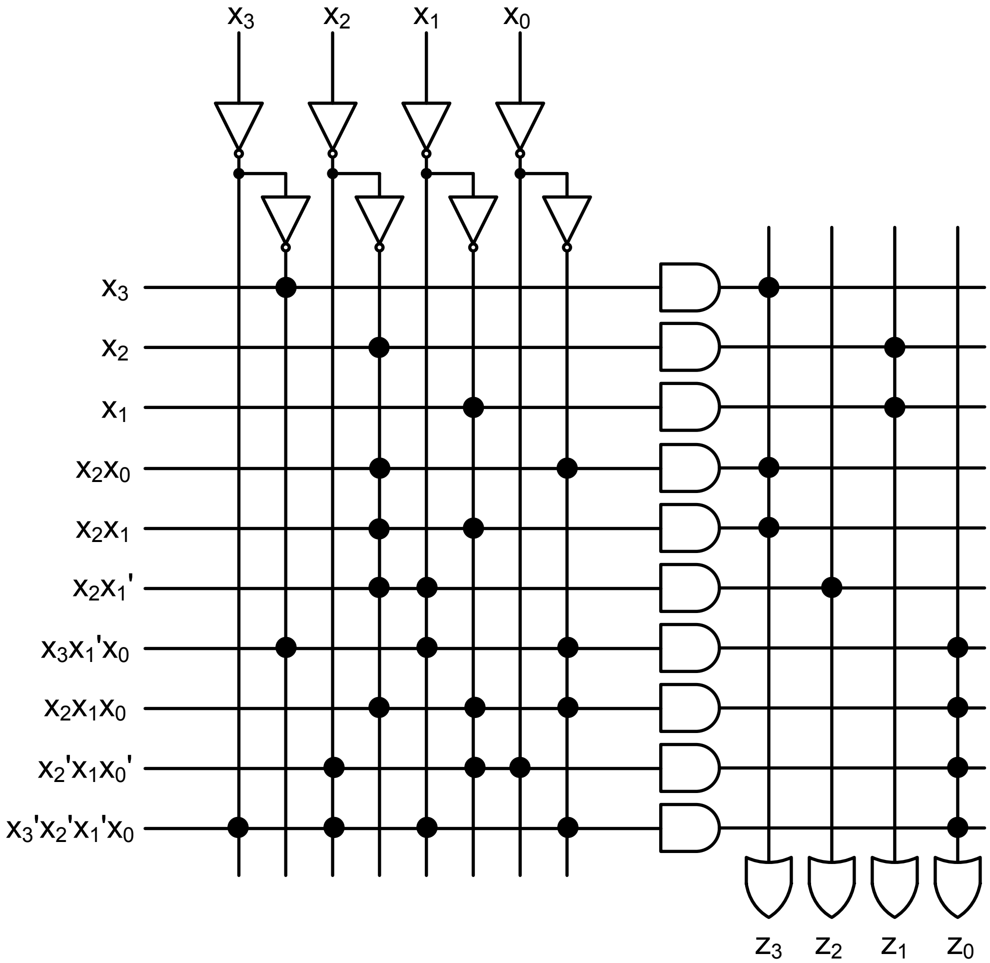
z_3: 1 OR3, 2 AND2

z_2: 1 AND2

z_1: 1 OR2

z_0: 1 OR4, 1 AND4, 2 AND3, 1 AND2

Total: 10 gates



x3
x2
x1
x0
x3
x2
x1
x2x0
x2x1
x2x1'
x3x1'x0
x2x1x0
x2'x1x0'
x3'x2'x1'x0
z3
z2
z1
z0