[CS M51A Fall 18] Ch5-sample-homework

Practice Exercises

From the book: 5.5, 5.7, 5.15. Solutions are posted.

Homework Problems

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.
2. Which of these prime implicants are essential?
3. Write the minimal sum of products for f.
4. Find all the prime implicates.
5. Which of these prime implicates are essential?
6. Write the minimal product of sums for f.

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.

x	y	z	w	E
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
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-map for the table.
3. Identify the essential prime implicants. Is xzw essential? Why or why not?

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.

x_1	x_0	y_1	y_0	z_2	z_1	z_0
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
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 K-maps for each output bit. Find the prime implicants.

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

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

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 Gray code for decimal digits is shown it the following table.

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.

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

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