

CSE 140 Homework Two

July 8, 2010

*Only **Problem Set Part B** will be graded. Turn in only **Problem Set Part B** which will be due on July 19, 2010 (Monday) at 4:00pm.*

1 Problem Set Part A

- textbook 2.24
- textbook 2.25
- textbook 2.29
- textbook 3.1(a)(d)
- textbook 3.3
- textbook 3.6(a)(c)
- textbook 3.9
- textbook 3.10(c)(d)
- textbook 3.12(a)(d)
- textbook 3.14(a)(b)
- textbook 4.4(d)
- Using the map method, determine the prime implicants of 4.4(d)
- textbook 4.5(a)(d)
- textbook 4.6(a)(d)
- textbook 4.7(b)
- textbook 4.9(b)
- textbook 4.17(a)
- textbook 4.18

2 Problem Set Part B

1 (Hamming Code & Excess-3 Code)

Hamming codes are used for error detection and correction purposes. A 7-bit Hamming code consists of 3 parity bits and 4 data bits, as shown in the table below.

Position	7	6	5	4	3	2	1
Bit	d_3	d_2	d_1	p_2	d_0	p_1	p_0

In this question, we consider the use of a Hamming code to detect and correct errors in Excess-3 codes. The 4 data bits constitute a valid Excess-3 code, while the 3 parity bits are determined by the following equations.

$$p_2 = d_3 \oplus d_2 \oplus d_1$$

$$p_1 = d_3 \oplus d_2 \oplus d_0$$

$$p_0 = d_3 \oplus d_1 \oplus d_0$$

(Part A) You are asked to derive the minimum **sum-of-products** expression for p_1 using the following Karnaugh map. The positions of the *don't cares* and one of the minterms have already been filled in for you. Please complete the map, circle the prime implicants, and then derive the corresponding expression. (*Hint: perhaps thinking about some of the symmetry issues and the function form for p_1 can help you fill in the table quicker than evaluating each entry individually.*)

The map of p_1

$d_3d_2 \backslash d_1d_0$	00	01	11	10
00	X	X	1	X
01				
11		X	X	X
10				

$p_1 =$ _____

(Part B) As we have examined in class, Excess-3 codes exhibit a *complementation property*. Given the codeword k , by flipping every bit, we can get the codeword $9 - k$.

When the 3 parity bits are embedded into the 4-bit Excess-3 codes, does the complementation property still hold for the resulting augmented 7-bit codewords? If your answer to this question is yes, please give a brief reasoning. If your answer is no, please provide a counterexample that violates this property.

The complementation property holds for the full 7-bit code.

True / False

Your justification:

(Part C) One student has been working on the generation of the minimum **sum-of-products** expression for one of the three parity bits. (Furthermore, his notes are a bit blurry so we don't quite know which parity bit he seemed to have made the mistake on.) In a moment of bleary eyed, need more coffee time, he made a mistake and filled one entry of the Karnaugh map incorrectly. The map he got is shown below, together with his final answer for the expression.

The map with a single-bit error

$d_3d_2 \backslash d_1d_0$	00	01	11	10
00	X	X	1	X
01	1	1	0	0
11	1	X	X	X
10	1	1	0	0

$$p_i = \underline{d'_1 + d'_3d'_2}$$

We would like you to detect his mistake and fix the problem. Please identify the parity bit that he was working on and the entry that was incorrectly filled, and give a brief reasoning for your answer.

The map is for parity bit _____, and the incorrect entry is _____

The reason is:

2 (Boolean Algebra)

This question concerns a Boolean Algebra of 8 variables, 0, A, B, C, D, E, F, and 1. We are given two operations “+” and “.”, with identity elements, 0 and 1, respectively, over the set {0, A, B, C, D, E, F, 1}. Two partially filled tables provide us some information about this particular Boolean Algebra, but not enough. You need to complete these two tables step by step, by applying the Boolean axioms and theorems we have discussed in class.

•	0	A	B	C	D	E	F	1
0								
A					?	?		
B		0			?		?	
C		0	0			?	?	
D		?	?	0				
E		?	0	?	A			
F		0	?	?	B	C		
1	0	A	B	C	D	E	F	1

+	0	A	B	C	D	E	F	1
0	0	A	B	C	D	E	F	1
A					?	?		
B					?		?	
C						?	?	
D		?	?	1				
E		?	1	?				
F		1	?	?				
1	1							

1) At the first step, you are asked to fill in the entries in the “.” table that are left blank, by applying **identity** and **commutativity**.

2) Based on the information you get from the first step, you are asked to fill in the entries in the “+” table that are left blank, by applying **De Morgan’s law**.

3) Again, based on the information you get from the first step, you can fill in the entries in the “+” table that are marked with “?”, by applying **absorption**.

4) At the last step, based on the information you get from the third step, you are asked to fill in the entries in the “.” table that are marked with “?”, by applying **De Morgan’s law**.

3 (Karnaugh Map Representations)

Karnaugh maps are employed for the simplification of Boolean functions. Remember that in Karnaugh maps the minterms (or maxterms) that can be combined together to generate a larger subcube are the ones that are placed adjacent. Two (min/max)terms are considered to be **adjacent** if they are next to each other either (1) directly or (2) in a wrap-around sense. Furthermore, to ensure the correctness and the optimality of the cube-merging process, a **valid** Karnaugh map should obey the following fundamental rule:

Rule: A pair of (min/max)terms should be placed at *adjacent* map positions **if and only if** they differ in exactly one variable.

(Part A) The following figure shows an incomplete 4-variable Karnaugh Map. The variables x, y and z, w are placed in the vertical and the horizontal axes, respectively. Yet the labels on the axes are missing.

$xy \backslash zw$	?	?	?	?
?				
?				
?				
?				

Given the variable placement shown in the figure above, what is the total number of distinct **valid** Karnaugh maps one can draw? In other words, how many distinct ways can one specify the labels on the horizontal and vertical axes that fulfill the rule outlined above? Please give a brief reasoning for your answer.

The total number of distinct **valid** Karnaugh Maps are _____.

Your reasoning:

(Part B) One crucial shortcoming of Karnaugh maps is that the map representation is not easily extendable to functions of more than 4 inputs, and the only representations that we have seen in class are admittedly clumsy and necessitate following relationships across multiple Karnaugh *submaps*.

One among the many bright CSE 140 students this year though noticed that perhaps if certain constraints were to be imposed regarding (min/max)term distribution, simple Karnaugh maps without relying on the subterfuge of trying to follow adjacency relationship across many Karnaugh maps could be derived, which will in turn enable correct derivation of cubes. Specifically, (s)he suggests that were the following constraints imposed on the term distribution of a 5-input function $F(a, b, c, d, e)$, a simple Karnaugh map could be attained.

$$F(*_1, *_2, 0, 0, *_3) \cdot F(*_1, *_2, 1, 0, *_3) = 0, \quad \forall *_1, *_2, *_3 \in \{0, 1\} \quad (1)$$

$$F(*_1, 0, *_2, 1, *_3) \cdot F(*_1, 1, *_2, 1, *_3) = 0, \quad \forall *_1, *_2, *_3 \in \{0, 1\} \quad (2)$$

The first equation precludes 8 pairs of *adjacent* minterms that differ only at c from being “1” at the same time, while the second equation additionally precludes 8 pairs of *adjacent* minterms that differ only at b from being “1” at the same time. Given these constraints, the student suggests to develop a simple map representation for F by putting 3 variables on one axis and the remaining 2 on the other, as shown in the following figure. The way of using this map representation to find prime implicants is identical to the way of using the conventional Karnaugh maps.

$a? \setminus ?b?$	000	001	?	?	?	?	?	?
00								
01								
11								
10								

As you may have noticed, the map representation shown above is incomplete. Could you help this student complete this map representation by 1) figuring out the variable placement 2) figuring out the labels on the horizontal axis.

The variable placement is a $\setminus$ b .

The labels on the horizontal axis are 000 001 .

(Part C) Now that you have managed to help this bright student generate a valid Karnaugh map for this 5-variable Boolean function F , flush with the zeal of victory, a number of students in the class decide to tackle bigger and better things. They start to derive a minimum **product-of-sums** expression for F (the function of course displays the same set of constraints as outlined in Part B) using the **maxterms**, based on the Karnaugh map you have just derived, by finding prime *implicates* in a manner identical to the way of using the conventional Karnaugh maps. These students derive an expression which of course they check several times, thus ensuring the correctness of at least the application of the cube-merging procedure.

Please justify the answer of these students and suggest to the TA whether

- (A) The simplified function generated using this map representation is not guaranteed to be correct.
- (B) The simplified function generated using this map representation is always correct but not guaranteed to be optimal.
- (C) The simplified function generated using this map representation is always correct and optimal.

Please clearly mark your choice. Moreover, if you think that the expression these students obtained is correct and optimal (i.e., you marked **C**), please give your reasoning. If you think the simplified function generated by them is incorrect or non-optimal (i.e., you marked **A** or **B**), please provide a counterexample to illustrate the problem(s) you identified.

Your selection is _____.

Your reasoning/example: