ECE199JL: Introduction to Computer Engineering **Fall 2012**
Homework 6 **Due: at start of lecture on Friday 5 October**

Logic Structures and Simplification

Study and learn the Lecture Notes, Set 2.1 "Optimizing Logic Expressions" and Set 2.2 "Boolean Properties and Don't Care Simplification to prepare for this homework.

Please do problems 3.15 from the textbook.

Here are six additional problems for this week:

1. Logic Simplification

A. The function $F(A, B, C, D)$ is specified by the Karnaugh map below.

CD \ AB	**00**	**01**	**11**	**10**
00	1	1	0	0
01	0	1	1	0
11	0	1	1	1
10	1	1	0	1

Express $F(A, B, C, D)$ as a canonical sum-of-products (SOP) expression.

B. List all prime implicants of the K-map.

C. Find a SOP expression using prime implicants from this K-map.

D. Find a SOP expression involving prime implicants for the following K-map.

CD \ AB	**00**	**01**	**11**	**10**
00	1	1	1	1
01	0	0	1	0
11	0	1	1	0
10	1	1	0	1

2. Logic Simplification

A. Write the truth-table for the middle (horizontal) bar of a seven-segment numerical display, with zero entries for the non-decimal-digit entries.

B. Make the K-map corresponding to this truth-table.

C. Find a minimal sum-of-products expression for this function.

D. Now assume that we don't care what value are taken on by the non-decimal-digit entries. Write the appropriate K-map including the don't-care entries.

E. Find a sum-of-products expression based on prime implicants for this function.

3. Logic Simplification

Consider the following K-map

CD \ AB	**00**	**01**	**11**	**10**
00	1	1	1	1
01	1	0	0	1
11	1	0	0	1
10	1	0	1	1

A. Find a sum-of-products expression from prime implicants for this function.

B. Draw a K-map for $\overline{F}$, the inverse of this K-map.

C. Find a minimal sum-of-products expression for $\overline{F}$.

D. Using De Morgan's Laws and your result in Part C, find a product-of-sums expression for the original K-map function F.

4. Logic Design
Redesign the ice cream dispenser from Notes Set 2.2 so that it uses only two buttons, L and M. Pressing both should produce a blend as described in the notes. The output bits must match the original specification exactly (you're just changing the interface logic). Give a schematic logic diagram similar to those on Page 15 as your answer, along with your derivation and explanation.

5. Logic Design Word Problem

So you notice an advertisement in a campus newspaper seeking an ECE student for a part-time job working in Psychology Professor Zapper's laboratory, and you apply for and get the job! Professor Zapper assigns you the task of building logic circuitry for the following experiment studying the learning ability of rats. The experimenter enters a number 0,1,2, or 3 on a keypad, which encodes the number as a two-bit unsigned integer as the binary inputs A and B to your circuit; A corresponds to the most significant bit, and B corresponds to the least significant bit. The experiment cage has two levers, one each to which the left and right paws of the rat are strapped. The output of the levers, L and R respectively, which will also be inputs to your circuit, are 1 if depressed by the rat, and 0 otherwise.

There are three lights in a row in the cage, corresponding to the input numbers 1, 2, and 3; call these F1, F2, and F3; the corresponding light should be on (output 1) only for the corresponding input number. There is also an output Z, which when activated ($Z = 1$) gives the rat a mild electric shock. Professor Zapper wants the experiment to work as follows: When the experimenter's input number is zero, the rat is never shocked ($Z = 0$), and all of the lights are off. When the input number is 1, the light F1 alone is activated, and the rat receives a shock unless it depresses only the left lever. When the input number is 2, the light F2 alone is activated, and the rat receives a shock unless it depresses only the right lever. When the input number is 3, the light F3 alone is activated, and the rat receives a shock if it depresses either or both levers.

A. Based on this description, design the desired truth table (inputs A, B, L, R) for the output Z.

B. Use a Karnaugh map to reduce the logical expression for this function Z to the simplest form that you can.

C. Draw the complete logic schematic circuit diagram for your system, for inputs A, B, L, and R entering on the left side, and outputs F1, F2, F3, and Z on the right side.

6. Generating Truth Tables
The C program `truthtable.c`, found on the course website or in the directory `/class/ece199/hw5`, generates the truthtable for a three-input AND function.

A. Modify your program from Homework 5 for generating and printing truthtables to check for *logical equivalence* (that is, whether two Boolean expressions produce the same values for all inputs). Your program should print the text "The expressions are equvialent" or "The expressions are not equivalent" accordingly. (If you have extra time, we encourage you to print a line for each element of the truthtable where they differ, giving the corresponding values of A,B,C, and D. This will be very useful for helping you to debug your logic.) Test your program with a few expressions that you know to be equivalent (for example, by De Morgan's Laws) and a few that you know are not. Using this program, check your answers to Problem 3,A and D, above. Turn in a printout of your program and a description of your tests of it as your answer.