[CS M51A WINTER 18]
HOMEWORK 3

Due: 02/09/18

Rules of Engagement: **Homework problems** must be submitted on the specified due date before discussion starts. Once discussion starts, a homework is considered late and will **not** be accepted. Please write legibly and follow directions.

1 Homework Problems (80 points total)

Problem 1 (15 points)

We are given a module with four binary inputs, w, x, y, z and one binary output , f, which are:

$$f = \sum m(0, 1, 2, 3, 5, 7, 8, 10, 11, 15)$$

1. (5 points) Draw K-maps for each output. Find the prime implicants.
2. (5 points) Find the essential prime implicants.
3. (5 points) Write the minimal sum of products for f.

Problem 2 (20 points)

For $f(x, y, z, w) = \Pi M(0, 1, 2, 3, 6, 7, 11, 14, 15)$.

1. (5 points) Write the above expression as the sum of minterms.
2. (5 points) Draw K-maps for f. Find the prime implicates.
3. (5 points) Find the essential prime implicates.
4. (5 points) Write the minimal product of sum for f.

Problem 3 (25 points)

We would like to design a BCD-to-Excess-5 code converter. The Excess-5 Code is obtained by adding 5 to each BCD number. 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	1	0	1
1	0	1	1	0
2	0	1	1	1
3	1	0	0	0
4	1	0	0	1
5	1	0	1	0
6	1	0	1	1
7	1	1	0	0
8	1	1	0	1
9	1	1	1	0

1. (3 points) How many bits do we need to encode the input?
2. (5 points) Write the truth table for the converter. Do not forget the don't care cases.
3. (12 points) Draw k-maps for each z bit. Find the prime implicants.
4. (5 points) Write the minimal sum of products for each z bit.

2 LogiSim Design Problem (20 points)

2.1 Introduction

For this assignment you will be designing a 4-bit comparator, which compares the magnitudes of two 4-bit inputs. You have been provided with a **comparator_4bit_skeleton.circ** file in which you must implement the required modules.

You will first implement the **comparator_1bit** module. This module takes two 1-bit inputs **A** and **B** as well as the inputs **Lin** and **Gin**, where Lin denotes whether the result of the previous 1-bit comparison was that $A < B$ and Gin denotes whether the result of the previous 1-bit comparison was that $A > B$.

Once this is done, use the **comparator_1bit** module as a component in the **TOP_comparator_4bit** module to build the full 4-bit comparator.

2.2 Design Guidelines

2.2.1 Allowed Components

In completing the 4 bit comparator, you are only allowed to use the gates NOT, AND, OR, NAND, NOR, XOR, XNOR, the comparator_1bit module that you design, and the Tunnel and Constant components under the Wiring category. Refrain from using long wires to connect everything, instead utilize the tunnel component to improve readability of the circuit. Please do not directly use the components under the arithmetic category for any of the modules.

2.2.2 Bit vectors

All bit vectors used in the design are indexed in little-endian form; the most significant bit has the largest index, all the way down to the least significant bit, which is always index 0. Take note of this fact when deciding the direction of the Lin, Gin and Lout, Gout signals.

2.2.3 Skeleton module

DO NOT change the location of the input/output pins in each module or add any new pins to the skeleton module. Moving the positions of the pins can cause the signals to connect to different pins from the original setup. Points will be deducted for any errors caused by this.

2.3 Submission

Print out your design and attach it to your homework submission. Make sure you have printed both the modules.