UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Curso: Bacharelado em Ciência da Computação
Disciplina: Introdução aos Sistemas Lógicos
2o. Semestre de 2017
Professor: Luiz Filipe Menezes Vieira (lfvieira@dcc.ufmg.br)

Data da entrega: 25.09.2017

Lista de Exercícios 2

Exercícios Teóricos

1. Consider the function $f(A, B, C, D) = \sum m(0, 1, 2, 7, 8, 9, 10, 15)$.
   (a) Write this as a Boolean expression in canonical minterm form.
   (b) Rewrite the expression in canonical maxterm form.
   (c) Write the complement of f in "little m" notation and as a canonical minterm expression.
   (d) Write the complement of f in "big M" notation and as a canonical maxterm expression.

2. Given the following function in product of sums form, not necessarily minimized:
   $F(W, X, Y, Z) = (W + \overline{X} + \overline{Y})(\overline{W} + \overline{Z})(W + Y)$
   (a) Express the function in the canonical sum of products form. Use "little m" notation.
   (b) Reexpress the function in minimized sum of products form.
   (c) Express $\overline{F}$ in minimized sum of products form.
   (d) Reexpress $\overline{F}$ in minimized product of sums form.

3. Use Karnaugh maps (K-maps) to simplify the following functions in sum of products form. How many literals appear in your minimized solutions?
   (b) $f(W, X, Y, Z) = \prod M(1, 3, 7, 9, 11, 15)$
   (e) $f(A, B, C, D) = \sum m(0, 1, 4, 5, 12, 13)$

4. Determine the minimized realization of the following function in the sum of products form:
   $f(W, X, Y, Z) = \sum m(1, 7, 11, 13) + \sum d(0, 5, 10, 15)$

5. Show the following:
   (a) A positive logic AND is equivalent to a negative logic OR.
   (b) A positive logic NOR is equivalent to a negative logic NAND.
   (c) A positive logic XOR is equivalent to a negative logic XNOR.
   (d) A positive logic XNOR is equivalent to a negative logic XOR.

6. Consider a five-input Boolean function that is asserted whenever exactly two of its inputs are asserted.

   (a) Construct its truth table.

   (b) What is the function in sum of products form, using "little m" notation?

   (c) What is the function in product of sums form, using "big M" notation?

   (d) Use the Karnaugh map method to simplify the function in sum of products form.

7. Design a combinational circuit with three data inputs D_2, D_1, D_0, two control inputs C_1, C_0, and two outputs R_1, R_0. R_1 and R_0 should be the remainder after dividing the binary number formed from D_2, D_1, D_0 by the number formed by C_1, C_0. For example, if D_2, D_1, $D_0 = 111$ and C_1, $C_0 = 10$, then R_1, $R_0 = 01$ (that is, the remainder of 7 divided by 2 is 1). Note that division by zero will never be requested.

   (a) Fill in truth tables for the combinational logic functions R_1 and R_0.

   (b) Derive minimized sum of product realizations of these functions using the Karnaugh map method.

   (c) Draw a circuit schematic that implements R_1 and R_0 using NAND gates only. You may assume any fan-in gates that you need.

8. Given the following function in sum of products form (not necessarily minimized): $F(A, B, C, D) = \overline{A}BC + AD + AC$. Reexpress the function in:

   (a) Canonical product of sums form. Use $\prod M$ notation.

   (b) Minimized product of sums form.

   (c) $\overline{F}$ in minimized product of sums form.

   (d) $\overline{F}$ in minimized sum of product form.

   (e) Implement F and $\overline{F}$ using NAND gates only. You may assume that literals and their complements are available.

9. Factor the following sum of products expressions:

   (a) $ABCD + ABDE$

   (c) $AC + ADE + BC + BDE$

10. Given the following specification of Boolean functions, implement them in a hazard-free manner:

    (a) $F(A, B, C) = B\overline{C} + \overline{A}C$

    (d) $F(A, B, C, D) = \prod M(0, 1, 3, 5, 7, 8, 9, 13, 15)$

11. Implement the function $f(A, B, C, D, E) = A + \overline{C}D + B\overline{D} + \overline{B}D + \overline{B}CE$ using a multiplexer and no other logic. The constants logic 1, logic 0, and the variables, but not their complements, are available.

12. The truth table for a 1-bit combinational binary subtractor, analogous to the half adder, computing D(ifference) = A minus B, with BL (borrow-from-left), is

A	B	D	BL
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

(a) Design a 1-bit combinational binary subtractor, comparable to the full adder, with two data inputs (A, B), a borrow from the right input (BI), a borrow request to the left output (BL), and a difference output (D).

(b) Show how your design can be cascaded to form multibit-subtractors.

(c) Does the subtractor work correctly for negative twos complement numbers?

(d) How is a subtraction underflow condition indicated?

13. Implement to the gate level an ALU bit slice with three operation selection inputs, S_2, S_1, S_0, that implements the following eight functions of the two data inputs A and B (and carry-in C_n):

S_2	S_1	S_2	ALU operation
0	0	0	$F_i = 0$
0	0	1	$F_i = B$ minus A
0	1	0	$F_i = A$ minus B
0	1	1	$F_i = A$ plus B
1	0	0	$F_i = A$ XOR B
1	0	1	$F_i = A$ OR B
1	1	0	$F_i = A$ AND B
1	1	1	$F_i = 1$

Assume a simple ripple carry scheme between bit slices.

14. [Katz Capítulo 4, Exercício 4.24] (Word Problems)
You are to design a converter that maps a 4-bit binary code into a 4-bit Gray code. The 4-bit Gray code sequence is defined as follows: 0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000. Give the truth table, and show how to implement this code converter as a ROM circuit and as a PLA circuit.