EECS 140/141 Introduction to Digital Logic Design
Spring Semester 2020
Assignment **#2** Due **4 February 2020**

Reading: Sections 2.5, 2.6, 2.8 in Brown/Vranesic

All logic networks on this (and every other assignment) *must* be drawn using the logic template specified in the course syllabus (or similar). Points will be deducted for failure to do this!

1. Problem 2.1, p. 69, in the Brown and Vranesic text.
2. Problem 2.5, Brown & Vranesic, p. 70. Do only the 15a version of DeMorgan's theorem. Venn diagrams.
3. Problem 2.2, p. 69.
4. Problem 2.3, p. 69.
5. Problem 2.17, p. 70, but you do not need to find the minimal sum-of-products form. In constructing the Venn diagram, first do a separate Venn diagram for each of the four $f = 1$ minterms (that is, for each row of the Truth Table that has $f = 1$). Then combine those 4 Venn diagrams together appropriately to get a Venn diagram for f itself.
6. Repeat problem 2.1, but this time you *must* prove the equality by expanding each side of the equation into canonical sum-of-products (CSoP) form using Boolean Algebra properties, then comparing the two sides. A sum of products (SoP) form has only "product" (AND) terms that are "summed" (OR'd) together. So, the left-hand side of this identity is in SoP form, but the right-hand side is not. You can get the right-hand side in SoP form using property 12a (distributive property), which behaves exactly like the distributive property in arithmetic (regular) algebra. In a *C*SoP form, *every* product term contains *every* input variable exactly once, either in its natural (no bar) or complement (bar) form. Such a product term is called a *minterm*. Product terms that do not contain every input variable (that is, product terms that are not minterms) can be expanded using property 14a. Along the way, use properties 7 and 8 as needed to simplify.
7. Problem 2.7, part (b) only, p. 70. For this problem, you *must* prove or disprove the equality by constructing a truth table for each side, then comparing the truth tables. To construct the truth tables, it will probably be helpful to insert some columns for "intermediate" values. For example, for the right-hand side, it will be helpful to construct columns for each of the three parenthetical expressions that are ANDed together, such as $(x_1 + \bar{x}_2 + x_3)$. The right-hand side result will then be the logical AND of those three intermediate columns.
8. Problem 2.10, p. 70, BUT do not use algebraic manipulation. Instead, note that the left side is expressed in minterm form (a minterm is a logical AND containing *every* input variable exactly once, in either its natural (no bar) or complement (bar) form), and the minterm numbers correspond to row numbers in the truth table (with variables given in order x_1 then x_2 then x_3), where rows are numbered starting with 0. To prove the equality, construct a truth table for the right-hand side (putting the input variables in order) and verify that only rows 1 through 7 contain a logical 1 in the output column, where rows are numbered starting with 0.
9. A system has three inputs, x_1, x_2, and x_3. The output of this system (z) should be a 1 if exactly two of the input variables has the value 1; otherwise, the output is 0.

a. Construct the truth table for this circuit.

b. Express the output variable in *canonical sum-of-products (CSoP)* form (see above for the definition of CSoP form). Note that there is one product term (a minterm) for each row of the truth table that has a logical 1 in the output column.

c. Draw the CSoP logic network for this circuit. See Figure 2.27(a) for an example of a logic network in canonical sum-of-products form.

d. Determine the "cost" (as defined in lecture) for this CSoP implementation. Remember that inverting any input variable is "free" (i.e., does not contribute to the cost).

10. For the the truth table in the next-to-last problem of Assignment 1 *as given in the solutions:*

a. Express the corresponding function in *canonical sum-of-products (CSoP)* form.

b. Draw the canonical sum-of-products logic network. See Figure 2.27(a) for an example of a logic network in canonical sum-of-products form.

c. Compare the *cost* (as defined in lecture) of this CSoP implementation with the cost of the logic network from the last problem of Assignment 1 (as given in the solutions). Note that the two logic networks are equivalent in that they implement the same truth table, but they have different costs.

