CSE 311: Foundations of Computing I
Assignment #7
May 16, 2014
due: Friday, May 23, 1:30 p.m., *before lecture begins*

Bundles: The problems in each homework assignment will be divided into 2 groups (to facilitate distribution to grading TAs). You will turn in 2 corresponding bundles. Write your name in the *upper left corner* of each bundle's top page, with your last name printed clearly in CAPITAL LETTERS. Each bundle should be stapled separately. We don't supply the stapler.

This week's turnin bundles: (A) problems 1–2, (B) problems 3–5.

Textbook numbering labeled "6th edition" refers to the textbook's Sixth Edition. Numbering that is unlabeled refers to the Seventh Edition.

In problems 1-3, use the Boolean expression notation of Chapter 12 [6th edition: Chapter 11], not the propositional logic notation of Chapter 1. For the complement of p, you may use either p' as was done in lecture or $\bar{p}$ as is done in the text.

1. Let $F(x, y, z) = x \oplus y \oplus z$. Be sure to show your work.
   (a) Express $F(x, y, z)$ in sum-of-products canonical form.
   (b) Express $F(x, y, z)$ in product-of-sums canonical form.

2. Section 12.2 [6th edition: Section 11.2], exercise 14. Explain how this exercise shows that the single operator NAND by itself can be used to express any Boolean function. (Exercise 15 similarly shows that NOR by itself can be used to express any Boolean function.)

3. Figure 9 on page 827 [6th edition: page 765] shows a very different implementation of a full adder than was presented in lecture. For this problem, assume that the half adders are implemented as presented in lecture, and not as shown in Figure 8. That is, the top output of each half adder is the XOR of its two inputs, and the bottom output of each half adder is the AND of its two inputs. Each half adder thus consists of just 2 gates, for a total of 5 gates in the full adder of Figure 9, one fewer gate than the full adder presented in lecture.

   Show that the resulting full adder computes the correct outputs. There is a subtlety to work out, because if you look carefully you will see that the carry out computed by this circuit does not come out exactly $xy + c_i x + c_i y$ as it should.

4. Section 13.2 [6th edition: Section 12.2], exercise 4. For each part, in addition to the output string, give the sequence of states (including the start state) determined by the input string.

5. Construct the state diagram for a finite-state machine *with output* that outputs 1 if the input string read so far contains the substring *aba*, and outputs 0 otherwise. The input alphabet is $\{a, b\}$ and the output alphabet is $\{0, 1\}$.