cs281: Computer Organization

Homework 1

1 Boolean Identities

1. (a) $A + 0 = A$
   (b) $A \cdot 1 = A$
2. (a) $A + 1 = 1$
   (b) $A \cdot 0 = 0$
3. (a) $A \cdot A = A$
   (b) $A + A = A$
4. (a) $A + \overline{A} = 1$
   (b) $A \cdot \overline{A} = 0$
5. $(\overline{\overline{A}}) = A$
6. Commutative Law
   (a) $A + B = B + A$
   (b) $A \cdot B = B \cdot A$
7. Associative Law
   (a) $A + (B + C) = (A + B) + C$
   (b) $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
8. Distributive Law
   (a) $A \cdot (B + C) = A \cdot B + A \cdot C$
   (b) $A + B \cdot C = (A + B) \cdot (A + C)$
9. DeMorgan's Theorem
   (a) $\overline{A + B} = \overline{A} \cdot \overline{B}$
   (b) $\overline{A \cdot B} = \overline{A} + \overline{B}$

2 Problems

1. Prove each of the following through algebraic simplification. Be sure and apply no more than 1 identity in each step and give the identity that allows you to simplify from one expression to the next. Use the numbering of the identities above, so that if the proof step uses the associative law on OR, you would label the line 7(a).

   (a) $A \cdot B + A \cdot \overline{B} = A$

   (b) $(A + B) \cdot (A + \overline{B}) = A$

   (c) $A + A \cdot B = A$

   (d) $A \cdot (A + B) = A$

   (e) $A + \overline{A} \cdot B = A + B$

   (f) $(A + \overline{B} + \overline{C}) \cdot (A + \overline{B} \cdot C) = A + \overline{B} \cdot C$

2. The following problems each present a "word problem" specifiying some desired boolean function. In each, you need to identify the inputs, identify the output(s), and then construct a truth table so that the outputs have the correct values for the input combinations.

   (a) Given a 4-bit word as input, the boolean function f should yield 1 if all bits are zero and 0 otherwise.

   (b) Given a 3-bit word as input, the boolean function f should yield 1 when two or more of the bits are 1 and should yield 0 otherwise.

   (c) Given two 2-bit words A and B (consisting of a_1a_0 and b_1b_0) as input, the boolean function f should yield 1 whenever A is greater than or equal to B when the words are interpreted as unsigned binary integers and should yield 0 otherwise. Include a second output g that has value 1 whenever the 2-bit words A and B are equal, and should yield 0 otherwise.

   (d) An n-bit word has *even parity* if, when we count the number of bits that are value 1, the result is an even number. So an 8-bit number would have even parity when the count of the number of 1 bits in the word are 0, 2, 4, 6, or 8. Define a boolean function f on a 4-bit input word A ($a_3a_2a_1a_0$) that yields 1 when A has even parity and 0 otherwise.

3. For each of the boolean functions defined in the previous question, do the following:

   (a) Give the canonical sum-of-products equation for each output.

   (b) Where possible, use boolean algebra to simplify the equation for each output.

   (c) Write a realization in logic gates for each output equation.