Homework One

EECS 303: Advanced Digital Logic Design
Assigned 25 September
Due date to be announced in class, definitely not due before 2 October

You may discuss the assignment with your classmates. However, you need to understand and write the solutions independently.

1. **(10 points)** The Boolean function $f(a_1, a_0, b_1, b_0)$ has two two-bit inputs, for a total of four input bits. $a = (a_1, a_0)$ and $b = (b_1, b_0)$ are two-bit binary numbers. $f(a_1, a_0, b_1, b_0)$ is true if and only if $a = b$.

   (a) Write a Boolean formula for f.

   (b) Write the truth table for f.

   (c) Draw a Karnaugh map for f.

   (d) a and b are each two-bit numbers. If a and b were each n-bit numbers, what is the minimal number of AND gates that would be required for a SOP implementation of

   $$f(a_{n-1}, \ldots, a_1, a_0, b_{n-1}, \ldots, b_1, b_0)$$

   The AND gates may have more than two inputs. Explain your answer.

2. **(10 points)** Simplify the following Boolean functions using algebraic manipulation, K-Maps, and the Quine-McCluskey algorithm, i.e., solve each problem using each technique. When doing algebraic manipulation, indicate the law you use with each step. For example, if you would like to convert from $a + \overline{b\overline{c} + d}$ to $a + \overline{(b\overline{c})}\,\overline{d}$, you can indicate that you're using the following law: $\overline{(a+b)} = \overline{a}\,\overline{b}$. You don't need to state the name of the law as long as you give its canonical form/definition. Treat don't-cares as zeros when doing algebraic minimization.

   Do any of these problems have special properties? Which one(s)? What is the property called? What impact does it have on minimization?

   (a) $f(a, b, c, d) = \sum(0, 5, 6, 10, 11, 13) + d(4, 8, 14)$

   (b) $f(a, b, c) = \sum(0, 2, 3, 4, 5, 7)$

   (c) $f(a, b, c, d) = \sum(1, 7, 8, 11) + d(0, 3, 14)$

3. **(10 points)** Minimize the following function using whatever manual technique you prefer

   $$f(a, b, c, d, e) = \sum(0, 1, 2, 5, 6, 8, 10, 11, 15, 17) + d(4, 9, 12, 13, 16, 21, 26, 29, 30, 31)$$

4. **(5 points)** Minimize the following functions using K-maps. Give your answer in POS form. In this notation, the zeros and don't-cares are specified instead of the ones and don't-cares. You can use either of the techniques presented in class.

   (a) $f = \prod(1, 4, 5, 9, 11, 14) + d(0, 2, 7, 8, 13)$

   (b) $f = \prod(2, 7, 10, 11, 14) + d(1, 4, 6)$

5. **(5 points)** Define the following terms:
   (a) Observability Don't-care
   (b) Satisfiability Don't-care

6. **(10 points)** What techniques are available for two-level logic minimization? Briefly (a word or two for each), what are the advantages and disadvantages of each?

7. **(5 points)** Cube $f(a, b) = a$. Cube $g(a, b) = a\bar{b}$. Does f cover g? Does g cover f?

8. **(5 points)** In two or three sentences, describe how Espresso searches for good implicants? Does it find all primes?

9. **(5 points)** Cofactor $f(a, b, c, d) = \bar{a}\,b + ac + ab\bar{d}$ by the cube $a\bar{b}$.

10. **(10 points)** Consider the following set of cubes:

0	0	X	0	1
X	1	0	X	1
1	X	0	1	1
1	1	1	X	1

You would like to expand the 111X|1 cube to 11XX|1. However, you need to confirm that this expansion is valid.

   (a) Write down new cube that must be checked for containment.
   (b) Write down the set of cubes within which it must be contained.
   (c) Cofactor the set of cubes by the new cube and use the result to determine whether the cube is contained.
   (d) Repeatedly cofactor on unate variables of the cover until it is clear that the expansion is valid or invalid.
   (e) Explain why cofactoring on a unate variables (instead of non-unate variables) makes validity checking faster.