EE 5393 UMN
Circuits, Computation, and Biology Spring 2015

Homework # 4

Due May 13, 2015 (but accepted until May 18)

1. **Linear Threshold Circuits**

   A linear threshold (LT) $f(X)$ is a Boolean-valued function with Boolean inputs $X = (x_1, x_2, \ldots, x_n)$ such that

   $$f(X) = \mathrm{sgn}[F(X)] = \begin{cases} 1 & \text{for } F(X) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

   An LT gate implements an LT function.

   (a) Consider the ADDITION function of two n-bit numbers. Let $X = (x_0, x_1, \ldots, x_{n-1})$, $Y = (y_0, x_1, \ldots, y_{n-1}) \in \{0,1\}^n$. The integer values represented by X and Y are equal to $\sum_{i=0}^{n-1} x_i 2^i$ and $\sum_{i=0}^{n} y_i 2^i$, respectively. Let $Z = (z_0, \ldots, z_n) \in \{0,1\}^{n+1}$. The integer represented by Z is equal to $\sum_{i=0}^{n} z_i 2^i$. Design a circuit with LT gates that implements addition: $Z = X + Y$.

   (b) Consider the COMPARISON function of two n-bit numbers. Let $X_1 = (x_1, x_3, \ldots, x_{2n-1})$, $X_2 = (x_2, x_4, \ldots, x_{2n}) \in \{0,1\}^n$. The integer values represented by X_1 and X_2 are equal to $\sum_{i=1}^{n} x_{2i-1} 2^{i-1}$ and $\sum_{i=1}^{n} x_{2i} 2^{i-1}$, respectively. The COMPARISON function is defined as

   $$C(X_1, X_2) = \begin{cases} 1 & X_1 > X_2 \\ 0 & \text{otherwise.} \end{cases}$$

   In other words,

   $$\begin{aligned} C(X_1, X_2) &= \mathrm{sgn}[X_1 - X_2] \\ &= \mathrm{sgn}\left[\sum_{i=1}^{n} 2^{i-1}(x_{2i-1} - x_{2i})\right]. \end{aligned}$$

2. **The Complete Quadratic Function**

   The Complete Quadratic (CQ) function is the Boolean function that consists of the XOR of all the $\binom{n}{2}$ possible AND's between pairs of variables. Namely,

   $$CQ(X) = (x_1 \wedge x_2) \oplus (x_1 \wedge x_3) \oplus \cdots (x_{n-1} \wedge x_n).$$

   For example,

   $$CQ(x_1, x_2, x_3) = (x_1 \wedge x_2) \oplus (x_1 \wedge x_3) \oplus (x_2 \wedge x_3).$$

   (a) Prove that $CQ(X)$ is a symmetric function. Express $CQ(X)$ as a function of $|X|$.

   (b) Let $|f(X)|$ be the number of X's for which $f(X) = 1$. Calculate $|CQ(X)|$ for $X \in \{0, 1\}^n$.

   (c) Draw a Linear Threshold circuit that computes $CQ(x_1, x_2, x_3, x_4, x_5)$

3. **Cyclic Combinational Circuits**

   In class we discussed combinational versus sequential circuits. Combinational circuits are "memoryless", i.e., the outputs depend only on the present values of the inputs. Sequential circuits have "memory", i.e., the outputs may depend on the past as well as present input values.

   In class we analyzed the circuit shown in Figure 1. It has three inputs a, b and c; six gates, each with fan-in 2, arranged in a single cycle; and six outputs, one from each gate. We argued that the circuit is combinational and produces the output functions shown. Note that each function depends on all three input variables. We argued that any acyclic circuit of fan-in 2 gates that implements the same output functions must have at least seven gates.

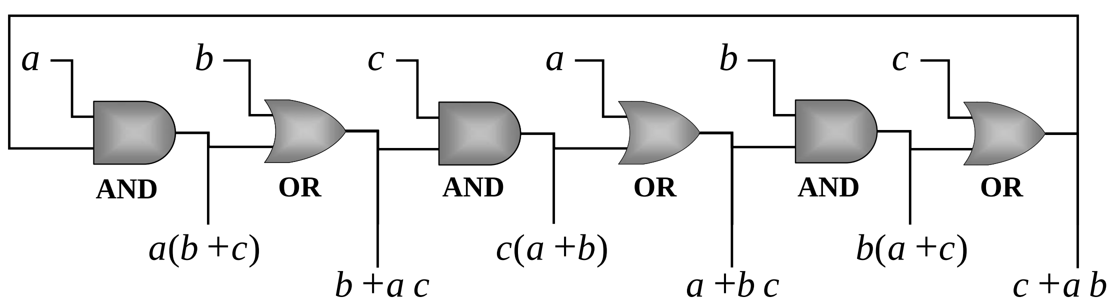

Figure 1: circuit

(a) Generalize the circuit in the following way.

Construct a circuit with six inputs and six fan-in 3 AND/OR gates such that each gate produces an output function that depends on all six input variables. What is the minimum number of fan-in 3 gates that would be required for an acyclic circuit that implements the same six functions? Justify your answer.

(b) Construct a circuit with $n(d-1)$ inputs and $2n$ fan-in d AND/OR gates, for $n \geq 3$, n odd, and $d \geq 2$, such that each gate produces an output function that depends on all $n(d-1)$ input variables. What is the minimum number of fan-in d gates that would be required for an acyclic circuit that implements the same $2n$ functions? Justify your answer.

(c) Construct a circuit consisting of six fan-in 2 AND/OR/NAND/NOR gates that implements six of the following eight functions (you choose which six!):

$$
\begin{aligned}
f_1 &= b(\bar{a} + \bar{c}) \\
f_2 &= \bar{a}bc \\
f_3 &= \bar{a} + \bar{b} + c \\
f_4 &= a + bc \\
f_5 &= c(a + \bar{b}) \\
f_6 &= c(a + b) \\
f_7 &= \bar{a} + b\bar{c} \\
f_8 &= \bar{a}\bar{c} + \bar{b}
\end{aligned}
$$

(Multiplication represents AND, addition OR, and a bar negation.)

(d) Prove that your circuit in (c) is combinational.

4. **Synthesis of Cyclic Combinational Circuits**

The goal in multilevel logic synthesis is to obtain the best multilevel, structured representation of a network. The process typically consists of an iterative application of minimization, decomposition, and restructuring operations. An important operation is **substitution**, in which node functions are expressed, or re-expressed, in terms of other node functions as well as of their original inputs. Consider the target functions in Figure 2.

$$\begin{aligned} f_1 &= \bar{x}_1 x_2 \bar{x}_3 + \bar{x}_2(x_1 + x_3) \\ f_2 &= \bar{x}_1 \bar{x}_2 \bar{x}_3 + x_1(x_2 + x_3) \\ f_3 &= \bar{x}_3(\bar{x}_1 + \bar{x}_2) + \bar{x}_1 \bar{x}_2 \end{aligned}$$

Figure 2: Target functions for synthesis.

Substituting f_3 into f_1, we get

$$f_1 = f_3(x_1 + x_2) + \bar{x}_2 x_3.$$

Substituting f_3 into f_2, we get

$$f_2 = \bar{x}_1 \bar{x}_2 \bar{x}_3 + x_1 \bar{f}_3.$$

Substituting f_2 and f_3 into f_1, we get

$$f_1 = \bar{x}_2 x_3 + \bar{f}_2 f_3.$$

For each target function, we can try substituting different sets of functions. Call such a set a **substitutional set**. Different substitutional sets yield alternative functions of varying cost. In general, augmenting the set of functions available for substitution leaves the cost of the resulting expression unchanged or lowers it. (Strictly speaking, this may not always be the case since the algorithms used are heuristic.)

In existing methodologies, a total ordering is enforced among the functions in the substitution phase to ensure that no cycles occur. This choice can influence the cost of the solution. Consider the ordering:

$$\begin{aligned} f_1 &= \bar{x}_2 x_3 + \bar{f}_2 f_3 \\ f_2 &= \bar{x}_1 \bar{x}_2 \bar{x}_3 + x_1 \bar{f}_3 \\ f_3 &= \bar{x}_3(\bar{x}_1 + \bar{x}_2) + \bar{x}_1 \bar{x}_2. \end{aligned}$$

This has a cost of 14. (As discussed in class, our **cost measure** for area is the number of literals in node expressions.)

Enforcing an ordering is limiting since functions near the top cannot be expressed in terms of very many others (the one at the very top cannot be expressed in terms of *any* others). Dropping this restriction can lower the cost. For instance, if we allow every function to be substituted into every other, we obtain:

$$\begin{aligned} f_1 &= \bar{x}_2 x_3 + \bar{f}_2 f_3 \\ f_2 &= x_1 \bar{f}_3 + \bar{x}_3 \bar{f}_1 \\ f_3 &= f_1 \bar{f}_2 + \bar{x}_2 \bar{x}_3 \end{aligned}$$

This has cost of 12. This network is cyclic and it is *not* combinational. As was discussed in class, an essential step in the synthesis process is the verify whether a circuit is combinational.

A cyclic solution that *is* combinational is:

$$\begin{aligned} f_1 &= \bar{x}_3 \bar{f}_2 + \bar{x}_2 x_3 \\ f_2 &= \bar{x}_1 \bar{x}_2 \bar{x}_3 + x_1 \bar{f}_3 \\ f_3 &= \bar{x}_1 f_1 + \bar{x}_2 \bar{x}_3 \end{aligned}$$

This has cost 13. So, introducing cycles can help reduce the cost, and yet still produce a valid solution.

Problem

Consider the functions:

$$\begin{aligned} f_1 &= ab\bar{c} + \bar{b}c \\ f_2 &= \bar{a}\bar{b}c \\ f_3 &= \bar{c}(\bar{b} + \bar{a}) + \bar{a}\bar{b} \end{aligned}$$

Design a combinational network, possibly cyclic, with minimal cost (where the cost is measured as the literal count.) For full points, find a network with cost 10.