

Homework # 6

Due Wednesday, March 15, 2006, 2:00 PM, at the TA office

Collaboration and/or discussion are not allowed on this homework set

1. LT Circuits for Symmetric Functions

In this problem you are asked to implement symmetric functions using circuits of LT gates. While in class we focused on depth-2 circuits, here the goal is to learn about depth size trade-offs.

Recall that a symmetric function $f(x_1, x_2, \dots, x_n)$ can be described by a vector of length $n + 1$ that we call the symmetric function table and denote it by $V(f) = (v_0, v_1, \dots, v_n)$, where

$$v_i = \begin{cases} 1 & \text{if for } |X| = i \quad f(X) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Let f be a 4-variable symmetric function with $V(f) = \{11010\}$.

Let g be a 16-variable symmetric function with $V(g) = \{11010110101101000\}$.

- Show a 4-gate, *layered* LT circuit that computes f .
- Show a 2-gate, *non-layered* LT circuit that computes f .
- Show a 4-gate, *non-layered* LT circuit that computes g .
- Show a 5-gate, depth-3, *non-layered* LT circuit that computes the *XOR* function of 15 variables.
- For $n = 4k^2 - 1$, where k is an integer greater than 1, show how to implement an n -variable *XOR* function using an LT *non-layered* circuit of depth 3 and a total of $3k - 1$ gates.

2. Reductions and Circuits

Reduction is a technique that helps in exploring relationships between circuits and the functions they compute. Let f and g be Boolean functions of n variables. Given a circuit C that computes f , if we show that g can be computed by C by moderately (at most polynomial in n) expanding the number of inputs to C , then we say that g can be reduced to f . Namely, g is easier to compute with circuits than f .

For example, let the Inner-Product-Mod 2 ($IP2$) be the following function that is defined for n even.

$$IP2(x_1, x_2, \dots, x_n) = (x_1 \wedge x_2) \oplus (x_3 \wedge x_4) \oplus \dots \oplus (x_{n-1} \wedge x_n)$$

For example, for $n = 4$,

$$IP2(x_1, x_2, x_3, x_4) = (x_1 \wedge x_2) \oplus (x_3 \wedge x_4)$$

It is easy to see that XOR can be reduced to $IP2$ by using a circuit for $IP2$ with $2n$ inputs, where the n bits of XOR are fed into the odd inputs and the even inputs are set to 1. In the following assume that:

- *Majority* is a function that outputs the majority of the inputs (in case of a tie it outputs 1).
- *Product* is the function that has $2n$ inputs and $2n$ outputs and computes the multiplication of two n -bit integers of n bits each.
- *Sorting* is the function of n^2 inputs and n^2 outputs that gets n n -bit integers as inputs and outputs the n integers in a sorted order.

Please prove the following (you can assume that the constant 0 and 1 and both the variables and their complements are available as inputs).

- (a) Every function in TH can be reduced to *Majority*.
- (b) $IP2$ can be reduced to *Product*.
- (c) *Extra Credit*: $IP2$ can be reduced to *Sorting*.

3. Stable States

In this problem we prove a couple of simple properties of stable states in feedback networks. Assume in this problem that $N = (W, T)$ is a feedback network operating in a serial mode with W a symmetric matrix with zero diagonal and T being the all-zero vector.

- (a) Prove that if V is a stable state then $-V$ is also a stable state.
- (b) Prove that if V is a stable state then all the n vectors that differ from V in exactly one entry are not stable states.
- (c) Let W be the matrix with all -1's (besides the diagonal which is all 0's). What are the stable states of the feedback network related to W ?