Parallel Algorithms - Spring 2001
Assignment No. 3- take-home exam
Course homepage: http://hyde.eng.tau.ac.il/~Parallel/

Deadline: July 15th, 2001.
Submit in printed format (drawings may be in hand).
You will need to defend your submission.

Questions:

1. **Comparison from MSB to LSB**. Design a combinational circuit consisting of n "gates" connected in a chain that computes the following.

   Input: Binary strings $x[n-1:0], y[n-1:0], z[n-1:0]$.

   Output: Does $\langle x[n-1:0]\rangle + \langle y[n-1:0]\rangle - \langle z[n-1:0]\rangle$ equal zero?

   The inputs of the ith gate are: $x[i], y[i], z[i]$ and $r[i]$. The output of the ith gate is $r[i-1]$. Remarks: (a) $r[i]$ is a symbol from a fixed alphabet that may have more than two symbols. (b) $r[n-1]$ is a fixed symbol. (c) $r[-1]$ is the output of the circuit and its value indicates if $\langle x\rangle + \langle y\rangle - \langle z\rangle = 0$.

   (a) Design the "gates".
   (b) Prove the correctness of your design.
   (c) Implement your design and simulate it. Submit simulations.

2. **Retiming of weighted addition trees**. A weighted addition tree is a tree T with non-negative integral edge weights $w(e)$. In every interior node v of T a full-adder is placed. The carry-out output is fed to a local register that feeds the carry-in input of the full-adder.

   Let r denote a retiming of (T, w). Let (T_r, w_r) denote the weighted addition tree obtained by retiming (T, w) according to the retiming r.

   (a) Prove that for every initial state S of (T, w), there exists an initial state S_r of (T_r, w_r) such that (T, w, S) is functionally equivalent to (T_r, w_r, S_r).
   (b) Suggest an algorithm that computes S_r.

3. **Multiplication of polynomials**. Polynomial multiplication is defined as follows. Let $a[n-1:0]$ denote the coefficients of a polynomial $a(x)$, namely:

$$a(x) = \sum_{i=0}^{n-1} a[i] \cdot x^i.$$

   The coefficients of the product $p(x)$ of two polynomials $a(x)$ and $b(x)$ of degree $n-1$ are defined by:

$$p[i] = \sum_{j,k:j+k=i} a[j] \cdot b[k], \quad \text{for every } i = 0, \ldots, 2n-1.$$

   If the coefficients are over $GF(2)$, then the above sum is simply the sum modulo 2 (i.e., xor).

(a) Design a systolic linear array polynomial multiplier over $GF(2)$. The complexity of the cell should be constant (i.e., independent of n).

(b) Prove the correctness of your design.

(c) Implement your design, and submit simulations.

4. **Retiming of Atrubin's multiplier**. Elimination of the broadcast mechanism in the semi-systolic version of Atrubin's multiplier was done by Even & Litman using the tiling technique. The retiming used is a retiming that satisfies $t_0 = \max_v r(v) = \Omega(n)$. The Retiming Lemma, however, guarantees that functionality is preserved only for states that are t_0-old. This means that if the multiplier is not properly initialized, then the Retiming Lemma does not guarantee the correctness of the systolic version of Atrubin's multiplier.

   The fact that the Retiming Lemma does not guarantee the correctness of the systolic version Atrubin's multiplier unless properly initialized is disturbing. Observe that the semi-systolic version of Atrubin's multiplier functions properly regardless of the initial state.

   (a) Formalize in what sense the semi-systolic version of Atrubin's multiplier functions properly regardless of the initial state.

   (b) Formalize in what sense elimination of broadcast by retiming preserves the functionality of Atrubin's multiplier (regardless of the initial state).

   (c) Prove the statements you have made in the previous two questions.

5. **Systolic counter with increment and decrement**. Design a systolic linear array that implements the following counter. Let a and b be fixed numbers such that $-2^n < a < 0 < b < 2^n$. All the registers are initially set to zero. In the ith clock cycle the input x_i is in the set $\{-1, 0, 1\}$. Let $S_k = \sum_{i=0}^{k} x_i$. The output EQa equals 1 in clock cycle $k+1$ iff $S_k = a$. Similarly, the output EQb equals 1 in clock cycle $k+1$ iff $S_k = b$. You may ignore what happens if $S_k \leq -2^n$ or $S_k \geq 2^n$.

   Hint: redundant arithmetic can be useful. See, for example, the following paper: Asger Munk Nielsen, David W. Matula, C. N. Lyu, and Guy Even, "An IEEE compliant floating-point adder that conforms with the pipelined packet-forwarding paradigm," IEEE Transactions on Computers, Vol. 49, No. 1, pp. 33-47, January 2000.

   (a) Design the requested counter.

   (b) Prove correctness.

   (c) Implement and submit simulations.