**Introduction to Digital Computers - Spring 1999**

Assignment No. 4

Course homepage: http://www.eng.tau.ac.il/~guy/Digital_Computers99/dc_home.html

**Messages:**

1. Firm Deadline: May 12th - before the beginning of the lecture.
2. Course's homepage provides instructions on how to join the course's mailing list.

**Questions:**

1. A cyclic shift (to the left) by one position is the function $c^1 : \{0,1\}^n \to \{0,1\}^n$ defined by

$$c^1(a[n-1:0]) = a[n-2:0] \cdot a[n-1].$$

A cyclic shift by $i+1$ positions is defined recursively by:

$$c^{i+1}(a[n-1:0]) = c(c^i(a[n-1:0])).$$

   (a) Design a circuit $CLS(n,i)$ (cycle left shift by $i$ positions) defined as follows:
   **input:** $a \in \{0,1\}^n$ and $s \in \{0,1\}$.
   **output:** $x \in \{0,1\}^n$ defined by

$$x[n-1:0] = \begin{cases} a[n-1:0] & \text{if } s=0 \\ c^i(a[n-1:0]) & \text{if } s=1 \end{cases}$$

   (b) Design a cyclic shifter $CLS(n)$ defined as follows:
   **input:** $a \in \{0,1\}^n$ - the string to be shifted. $b \in \{0,1\}^{\lceil \log_2 n \rceil}$ - the shift amount.
   **output:** $c^{\langle b \rangle}(a[n-1:0])$.
   hint: Use $CLS(n,2^i)$ for $i = 0,1,\ldots,k$ (where $k = \lceil \log_2 n \rceil - 1$).

   (c) Analyze the cost and delay of your $CLS(n)$ design.

2. A bi-directional cyclic shift can be either to the left or to the right. We denote a right shift by one position as a left shift by $(-1)$ positions, as follows:

$$c^{-1}(a[n-1]:0) = a[0] \cdot a[n-1:1].$$

A cyclic right shift by $i+1$ positions is defined by:

$$c^{-(i+1)}(a[n-1:0]) = c^{-1}(c^{-i}(a[n-1:0])).$$

   (a) Prove that if $x = y \pmod{n}$, then

$$c^x(a[n-1:0]) = c^y(a[n-1:0])$$

   (b) Prove that if $s \in \{0,1\}^{m+1}$ and $n = 2^m$, then

$$c^{[s[m:0]]}(a[n-1:0]) = c^{\langle s[m-1:0] \rangle}(a[n-1:0])$$

(c) What does the previous claim imply about implementations of bi-directional cyclic shifters when $n$ is a power of 2? How would you suggest to implement a bi-directional cyclic shifter?

(d) Consider a processor that has two types of cyclic shift instructions: "shift left by $\langle s \rangle$ positions" and "shift right by $\langle s \rangle$ positions". Assume $n = 2^m$ (length of string to be shifted). Suggest an implementation for these shifts? Can you do better than design two separate cyclic shifters?

3. Consider the following notation for a fast adder:

$$\begin{aligned} \sigma[i] &= a[i] + b[i] \quad \text{for } 0 \leq i \leq n-1 \\ \pi'[i] &= \sigma[i] * \cdots \sigma[1] * \sigma[0] \quad \text{for } 0 \leq i \leq n-1 \end{aligned}$$

Note that in class $\pi[i] = \sigma[i] * \cdots \sigma[0] * \sigma[-1]$.

(a) Suppose that we have computed $\pi'[n-1:0]$. Show the fastest and cheapest possible way to compute both $\langle a[n-1:0] \rangle + \langle b[n-1:0] \rangle$ and $\langle a[n-1:0] \rangle + \langle b[n-1:0] \rangle + 1$ from $\pi'[n-1:0]$, $a[n-1:0]$, and $b[n-1:0]$.

(b) Prove the correctness of your suggestion.

(c) Compare this design with the design $ST\_Add(n)$.