Computer Arithmetic - Spring 1999

Assignment No. 2

Course homepage: http://www.eng.tau.ac.il/~guy/arith99/arith_home.html

Deadline: July 11th 1p.m.

Questions:

1. Let x denote a number. Let $r_{ne}(x)$ denote the number x is rounded to in round-to-nearest mode.

   (a) Define the function $r_{ne}(x)$.

   (b) Let (s', e', f') denote the factoring defined by

   $$(s', e', f') = exp\_rnd(post\_norm(s, en, sig\_rnd(fn))),$$

   where (s, en, fn) denotes a normalized factoring of x. Prove that the value represented by (s', e', f') equals $r_{ne}(x)$.

2. Read the floating point addition algorithm in the paper "On the design of IEEE Compliant Floating Point Units". Prove the correctness of the addition algorithm.

3. Prove that injection based rounding is correct in floating point multiplication if the significands of the factors are normalized (i.e. in the range $[1, 2)$).

4. Show how to compute the sticky bit of a carry-save encoded binary string.

   Input: $c[0 : q-1], s[0 : q-1] \in \{0,1\}^q$.

   Output: $sticky\_bit(f, 0)$, where $f[1 : q-1]$ satisfies

   $$\langle f[1 : q-1] \rangle = \langle c[1 : q-1] \rangle + \langle s[1 : q-1] \rangle$$

   (a) (Hint) The following solution was suggested: Compute $x[i] = xor(s[i], c[i])$. Output $sticky\_bit(x, 0)$ using an OR-tree. Prove or disprove the correctness of this proposal.

   (b) Design the required circuit. Prove the correctness of your design. What is the overhead in delay and cost of your suggestion compared to an OR-tree? Can you reduce this overhead to a constant delay and linear cost?

5. Compound Adder. 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 $ST\_Add(n)$.