CS 560 Spring 2008: Homework 4

due Tuesday, April 15

In this homework, you will explore the use of the MMAlpha system to perform many of the transformations that you have previously done by hand.

Problem 1 from HW3 [40 pts]: Redo this problem (repeated below for your convenience). However, there are two aspects. In the first instance, you should simply apply the transformation requested, normalize the program and save it. In the second instance, you should use MMAlpha's tools for manuipulating domains and work you way *step by step* through the same process that you used to do HW3 (i.e., grade your own work).

Consider the system of equations after uniformization/localization for the FIR as given in Equations (19-22) on page 9 of the [LQRR99] reference off the CS560 schedule page. Apply the CoB transformation $(i, j \rightarrow -j, i - j + 3)$ to the variable X, and construct the resulting SRE. Draw the new domain of X, the transformed dependences (your figure should not be overly cluttered, but should have enough detail to see all the different types of dependences). You should use the convention in the lecture notes: origin at the bottom left, i horizontal. [8 pts]

For the following three questions, you will start afresh with SURE (19-22). The transformations are cumulative: you apply a new transformation to the result of the previous one, so be careful since errors will cascade. After each transformation, write the resulting SRE, and draw out only the domain of the variable which is changed. As before, put the origin at bottom left.

- Apply the CoB transformation $(i, j \rightarrow j, i + j - 1)$ to the variable X. [8 pts]
- Apply the CoB transformation $(i, j \rightarrow j, i + j - 1)$ to the variable W. [8 pts]
- Apply the CoB transformation $(i, j \rightarrow j, i + j)$ to the variable Y. [8 pts]

After each of the four transformations above, was the resulting SRE an SARE or SURE? [8 pts]

Problem 2 [60 pts]: Redo problem 2 of the midterm to derive the convolution array as explained in class. Please note that your starting equation will have to be manipulated by hand. It may also be impossible to avoid the "mother of all tgransformations" (emacs) but please be sure to minimize its use.

Derive a systolic array that computes a $2m$-point convolution, $y_i = \sum_{j=0}^{2m-1} w_j x_{i-j}$ for the special case when the weights are symmetric, i.e., $w_j = w_{2m-j}$. Assume that the boundary conditions are as in the lecture notes (Chapter on Advanced Systolic Design).

The key constraint is that multipliers are expensive, so you should exploit symmetry and use only m multipliers (i.e., your architecture should have m PEs, each with a single multiplier). Describe the systematic design of your architecture, including:

- mathematical "preprocessing" of the equations;
- serialization (replacing reductions by accumulations);

- localization (replacing long dependences by propagations);
- scheduling;
- allocation;
- constructing the CoB and transforming the equations; and
- derivation (as far as possible) of the final hardware.