NANO-SCALE RECONFIGURABLE CHIPS FOR IMAGE PROCESSING[1]

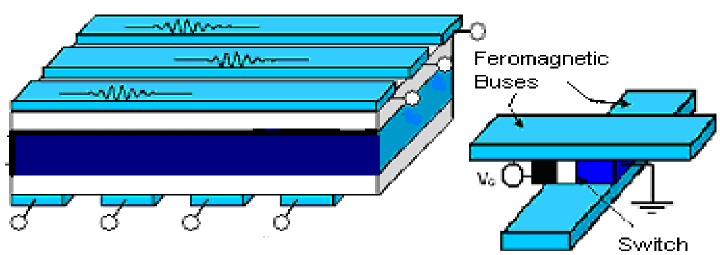
M. M. Eshaghian-Wilner, Alex Khitun*, Shiva Navab, Kang Wang*

***MARCO Focus Center on Functional Engineered Nano Architectonics (FENA)**

Department of Electrical Engineering
University of California at Los Angeles

Over the past few decades, several mesh-based parallel architectures such as mesh-connected computer, mesh-of-trees, pyramid, reconfigurable meshes, systolic meshes, and optical meshes have been considered for performing low and intermediate-level computer vision tasks. This is due the fact that a two dimensional image can be mapped in a straightforward fashion onto a two-dimensional mesh. In particular, reconfigurable meshes have been shown to be attractive computational engines due to the flexibility that the reconfigurable bus offers.

Here, we study a nano-scale reconfigurable mesh that is interconnected with ferromagnetic spin-wave buses for image processing. In this architecture, as shown in the figure below, a set of column spin-wave buses at the bottom and a set of row spin-wave buses on the top are connected via the spin-wave switches. Each switch is placed at the grid point of the mesh. Basically, except for the spin-wave buses, the nano-scale reconfigurable mesh with spin-wave buses is similar to the standard reconfigurable mesh. However, in a spin-wave architectures data is transmitted over waves, and logic can be performed using the phase of waves.



Our nano-scale spin-wave-based architecture, while requiring the same number of switches and buses as regular reconfigurable meshes, is capable of simultaneously transmitting multiple waves on each of the spin-wave buses. Furthermore, this design is fault tolerant because if there is a failure in one of the channels, any of the other channels can be reconfigured to transmit the data. Because of these feature, very fast and fault-tolerant algorithms can be designed for this architecture. The actual physical implementation of spin-wave reconfigurable mesh is currently underway at UCLA, and we have a preliminary prototype version in which we are able to detect spin-waves at nano-scale.

One of the most attractive properties of spin-waves is superposition. This property suggests that the resultant disturbance at any point in a medium is the algebraic sum of the separate constituent waves. Hence, it enables many optical signals to pass through the same point in space at the same time without causing mutual interference or crosstalk. We employ this characteristic to allow concurrent writes if all the requesting processors want to write a *"1"*. This leads to constant running time of the following geometric algorithm, under the assumption that broadcasting of O(log N) bits can be done in unit time:

[1] Authors are listed Alphabetically

Given an $N^{1/2} \times N^{1/2}$ image, using a $N \times N$ Spin-Wave reconfigurable mesh, in O(1) time, for a single figure, its nearest neighbor, convex hull and a smallest enclosing box can be found. For multiple figures, we present the following image processing algorithms illustrating the potential of our proposed spin-wave reconfigurable mesh.

Labeling of Multiple Figures on Spin-Wave Mesh

Identifying figures in an image is an early step in intermediate-level image processing. In a *0/1* picture, the connected 1's are said to form a figure. Thus, associated with each PE is a label, which is the unique *ID* of the figure to which the PE belongs. An $N \times N$ digitized picture may contain more than one connected region of black pixels. The problem is to identify to which figure (label) each *"1"* belongs.

We show that even given an $N \times N$ image, using a $N \times N$ Reconfigurable Mesh with Spin-Wave Buses, in a systolic fashion with inputs coming from left and leaving from the right, each figure can be labeled with O(1) steps per systolic cycle. This result is also an O (log N) time faster than the best known solution for systolic reconfigurable mesh. Basically the improvement is possible because of the fact that multiple labels can be simultaneously transmitted over the buses using multiple waves at different frequencies. Therefore there is no need to rotate the figure or slow down the entire process to avoid conflicts.

Furthermore, in a non-systolic fashion the same algorithm can be solved in O (log N) time. This is done by each node immediately setting the buses over the connected region and then the minimum label over each connected region to be computed by O (log N) bits. Once the figures are labeled then we can associate a different frequency to each of the figures to continue with other image computations such as the convexity problem shown below.

Convex Hull of Multiple Figures on Spin-Wave Mesh

To find the convex hull of multiple O(N) figures in an image size of O(N) using a spin-wave reconfigurable mesh of size NxN, we will find the extreme points on each of the figures. To do this, we basically find the convex hull of each figure using a different frequency on the spin-wave mesh. Therefore, before starting this algorithm, we first do a labeling of figures as discussed above and then associate to each label a distinct frequency. Now using each frequency we find the right most and left most points of each figure, and then find if they would have to be eliminated or not. A left most or a right most figure is eliminated if it is not an extreme point. An extreme point is one that makes an angle less than 180 degrees with respect to all the right most points and left most points above it and below it. This can be done in O(1) time for N figures in parallel. This is possible because the computations are done in parallel for N figures on each column using a different frequency for each of the N figures. Also, previously on the Mesh-of-Trees this was possible in O (log N) time, but only for one figure at a time. Here using the spin-wave mesh it is possible to process N figures at the same time.

Nearest Neighbors of Multiple Figures on Spin-Wave Mesh

Finding the nearest neighbor to each of the figures is relatively a simpler problem because the nearest neighbor to each point can first be found, and then the result can be propagated to each of the figures, but all in parallel. This can be done in O (log N) time, but due to space limitations the details are not shown. Similarly, for a smaller image size of N, this problem can be solved in O (1) time.