Image Processing Architecture for Semiconductor Nanowire Fabrics

Pritish Narayanan, Teng Wang, Michael Leuchtenburg, Csaba Andras Moritz
Dept. of Electrical and Computer Engineering
University of Massachusetts
Amherst MA USA
{pnarayan, andras}@ecs.umass.edu

Abstract – **A new processing architecture for semiconductor nanowire grid fabrics is presented. The system consists of a large number of functionally identical units called cells. Cells are locally interconnected with nearest neighbors, with a limited number of global signals routed from supporting CMOS circuitry. One possible implementation of a digital Cellular Neural Network (CNN) using this architecture is shown. The digital cellular design may be up to 27X denser than an equivalent 18nm CMOS implementation. The system, based on a collective computation model, may also significantly alleviate manufacturing, since 100% fault-free components may not be necessary.**

I. Introduction

Semiconductor nanowire (NW) based architectures [1][2] are promising alternatives to end-of-the-line CMOS given advances in manufacturing and assembly of these materials [5][6]. A fabric based on NWs is NASIC (Nanoscale Application Specific Integrated Circuits) [1][3][4], which utilizes 2D grids of nanowires for logic computation. Processor designs using NASIC compare substantially well against equivalent CMOS designs at aggressively scaled technology nodes [1][3].

In this paper we explore a new architecture for image processing based on the NASIC fabric. The system is highly parallelized, with a large number of identical cells performing a given task. Instructions to each cell are transmitted on a limited number of global signals controlled from reliable peripheral CMOS circuitry. The cells themselves are composed of a small number of nanodevices that perform simple computations and are locally interconnected. The collective behavior of a large number of interconnected cells achieves information processing.

These types of collective computation systems with parallel processing of image pixels may be a 'natural fit' for nanoscale architectures because:

i) High densities are achievable for simultaneous processing of large pixel arrays for real time image processing tasks

ii) Non-conventional manufacturing methods are more amenable toward the formation of regular structures

iii) Local interconnectivity of cells with minimal global routing and

iv) The nature of the application is such that achieving 100% fault free components is not a requirement, in contrast to general purpose computing. However, given the high defect rates in nanomanufacturing, NASIC built-in fault tolerance techniques [1][3] will need to be integrated at different levels to ensure that a sufficiently large number of cells are functioning correctly.

These systems built using the NASIC fabric can be adapted for implementing a variety of image processing paradigms. For example, a system based on discrete-time cellular nonlinear network (DT-CNN) [9] with programmable templates is discussed in detail in this paper. Other models, such as based on time-domain image convolution [7], are also possible.

While the original Cellular Nonlinear Network of Chua and Yang is an analog system [8], many digital implementations have since been proposed [9][11]. We believe that analog systems based on charge based devices may not be feasible at the nanoscale given the requirements for arbitrary placement and sizing of devices and precise control of transistor operating points. However, some implementations utilizing novel devices such as the resonant tunneling diode and spin wave bus [10] may be possible in the long term. In this paper we discuss a fully digital, discrete time CNN with crossed nanowire field-effect transistors (FETs) that we feel is realizable in the near term. A detailed study of different CNN implementations at the nanoscale is presented in [12].

The density of the NASIC cellular architecture is compared against equivalent CMOS designs scaled to aggressive technology nodes. It is seen that the NASIC cellular architecture has significant density benefits as compared to an 18nm CMOS implementation (up to 27X).

The rest of the paper is organized as follows. Section II provides an overview of the NASIC fabric. Section III discusses the NASIC cellular architecture in detail. Density and manufacturing implications are discussed in Section 2.

II. Overview of NASIC

NASIC (Nanoscale Application Specific Integrated Circuits) is a nanoscale fabric proposed for semiconductor nanowires and targeting datapaths. NASIC designs use Field Effect Transistors (FETs) on 2-D semiconductor NW grids to implement logic functions. NASICs are based on cascaded 2-level logic style, e.g., AND-OR, NAND-NAND. Microwires provide control signals generated from CMOS circuitry, with a dynamic control style that channels the flow of data through the nanowire tiles. NASIC designs are optimized according to

978-1-4244-2104-6/08/$25.00 ©2008 IEEE.

specific applications to achieve higher density and defect/fault-masking.

0 shows a 1-bit full adder implemented on the NASIC fabric. This tile uses a 2-level cascaded NAND-NAND scheme for logic implementation. The tile comprises of n-type nanowires in horizontal and vertical directions surrounded by a small number of microwires carrying power supplies (V_{dd} and V_{ss}) and control. *hpre, heva, vpre* and *veva* are signals used to dynamically control the flow of data through the tile.

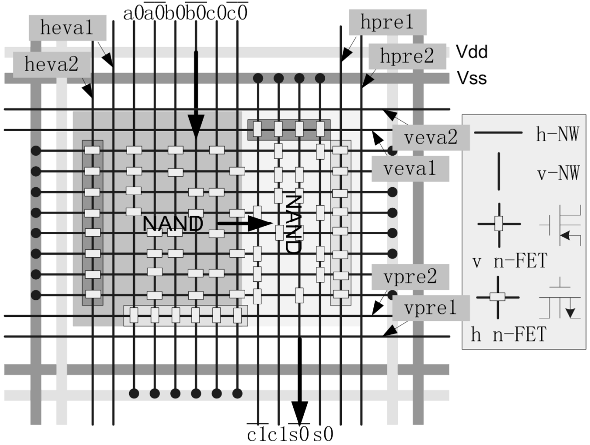

Fig.1. NAND-NAND based implementation of a NASIC 1-bit full adder. White boxes are n-type transistors.

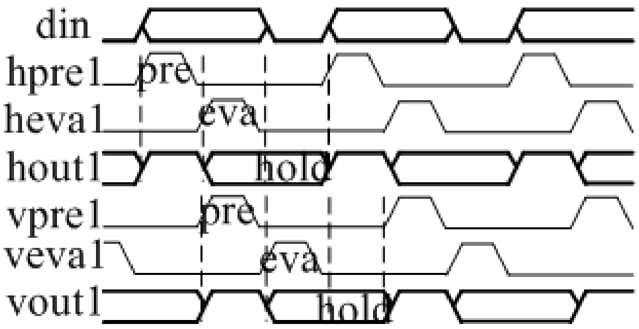

Fig. 2. Timing diagram for NAND-NAND logic showing precharge and evaluate of horizontal and vertical NAND planes

The timing diagrams for dynamic control and data propagation are shown in 0. The outputs of the horizontal plane are first precharged by asserting *hpre*. *hpre* is then switched off and *heva* is switched on to evaluate the output of the horizontal stage based on the inputs. During this time the output vertical nanowires are precharged (*vpre*). The horizontal nanowires then go into hold phase with both *hpre* and *heva* switched off, so that the vertical plane can be evaluated (*veva*). This 3-phase progression of precharge-evaluate-hold is implemented on all NASIC tiles and enables streaming of data through a large multi-tile system. More information about NASIC dynamic control schemes is available in [3][3][4].

III. The NASIC Cellular Architecture

In this section we explore a new cellular architecture based on the NASIC fabric for tasks such as image processing. The system (Fig. 3) consists of an array of simple, locally interconnected cells, along with a limited number of global signals propagated to every cell in the design. In the specific case of a digital cellular architecture on the NASIC fabric, these global signals would satisfy the role of templates. These signals are driven from peripheral CMOS circuitry, making fully programmable templates possible. In addition, there is a CMOS Input-Output Controller that serially loads the inputs and reads out outputs. Alternatively, input-output operations may be performed by a focal plane array of nanosensors placed directly above the cell array, with local processing of individual pixels. This is one motivation for creating nanodevice based cells, since with CMOS technology; the area required to process a single pixel may be much larger than the size of the pixel itself [7].

Each cell, or basic functional unit, is a collection of NASIC tiles. Nanowires provide local interconnection between tiles in a cell as well as between neighboring cells. The regular nature of the fabric and periodic arrangement of cells is specifically chosen. This implies that precise nano-micro alignment for any lithographic steps in the manufacturing process is removed.

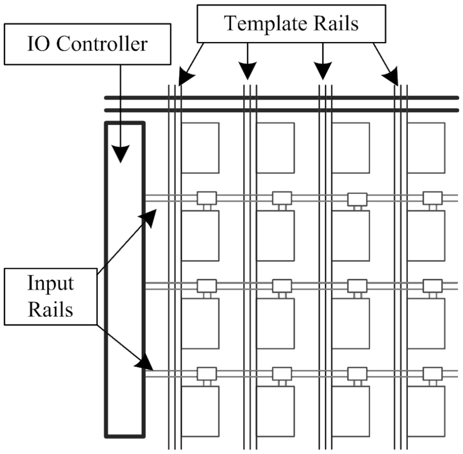

Fig. 3. Generalized architecture for Image processing using the NASIC fabric

Each cell performs a relatively simple computation task, and the collective behavior of a large number of cells accomplishes information processing. In the case of a digital Cellular Neural Network, the task is state evolution; through the generation of partial sums, followed by the accumulation of contributions from neighbors. The expression for state evolution of any cell is given by the expression:

$$x(n+1)=\sum_{i\in N} a_i\, y_i(n)+b_i\, u_i+C$$

where each a_i and b_i are instantaneous template values, $y_i(n)$ and u_i are the outputs and inputs to a particular cell, $x(n+1)$ is the state and C is some constant term. The term $a_i\, y_i(n)\, +\, b_i\, u_i$ represents the i^{th} partial sum. The system is based on the computation model for digital CNN designs proposed in [9]. However, it achieves this model on a 2-D fabric with programmable templates and built-in fault tolerance. The state, templates and input may all be multi-bit values. The output of every iteration is simply the most significant bit (MSB) of the state term. This relation achieves the nonlinearity for the digital CNN.

The layout of a single cell of this CNN is shown in Fig. 4. Input and template values are received from the respective rails. Since templates are externally controlled, the CNN is programmable and can perform a multitude of image processing tasks. Nanowire girds perform logic computations. Arrows show the datapaths in the design. A broadcast network (solid arrows) sends partial sums from the cell to the nearest neighbors. Similarly, dotted lines in the periphery are partial sums received from neighbors. The dashed lines represent the state accumulation datapath. There is a NASIC tile at the center for generating partial sums and state accumulator tiles at the corners of each cell. The execution inside a cell progresses through the following stages:

1. Generation of Partial Sums: Initially, the control circuitry for the state accumulators is off and the partial sum generator is on. Each cell then generates 5 partial sums in sequence, one per nearest neighbor, and one for itself. These are transmitted on the broadcast network (solid lines - clockwise loop).
2. State Accumulation: Once all partial sums are available, state accumulation datapaths are activated (dashed lines - anti-clockwise loop). This is enabled by switching off the control circuitry for the partial sum and switching on the control for the state accumulator tiles. During this stage, the broadcast networks may be envisioned to be in an extended hold phase, with no precharge-evaluate

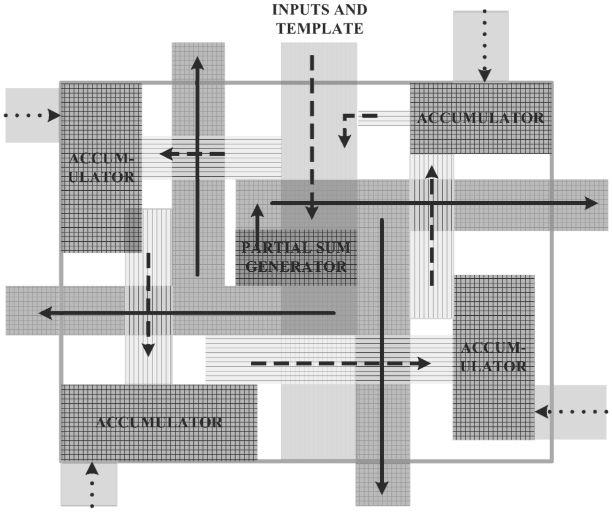

Fig. 4. Single Cell of a NASIC based Digital Cell. Solid lines represent partial sum broadcast to neighbors. State accumulation datapath shown using dashed lines. Dotted lines in the periphery are components received from neighbors.

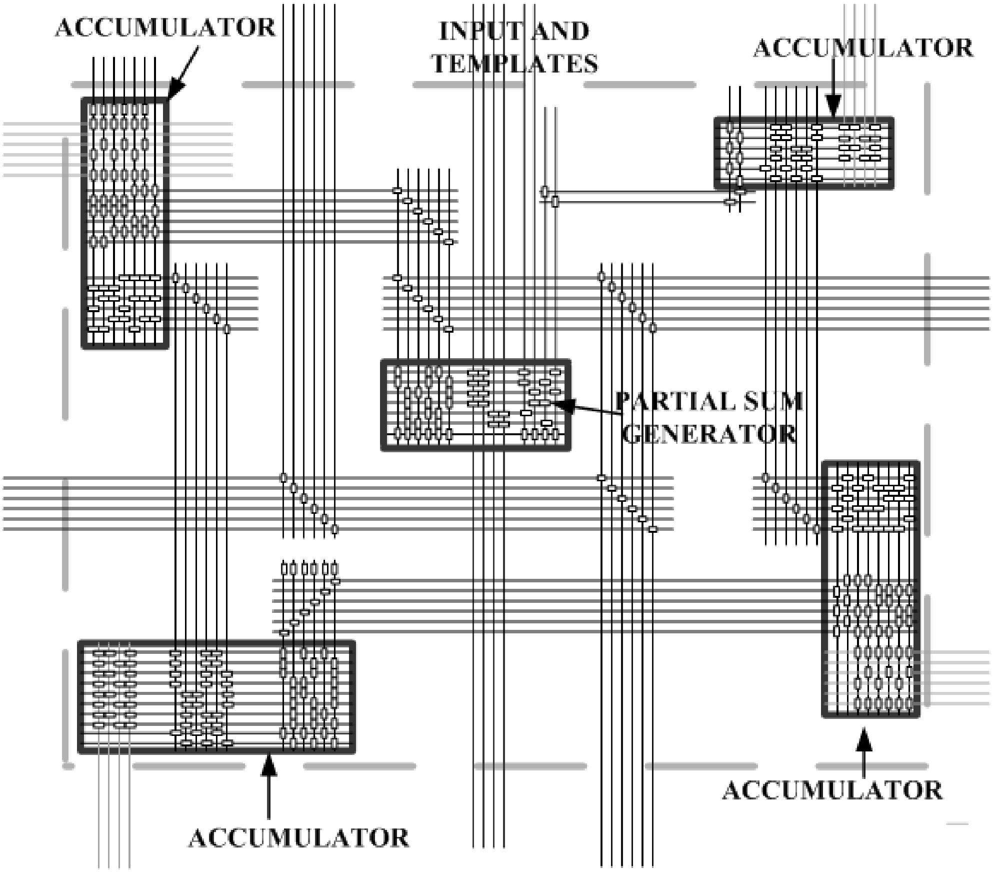

Fig. 5. Circuit level implementation of a NASIC digital cell: White boxes are transistors

operations. Thus, by suitably triggering the dynamic control circuitry, selective activation of datapaths and propagation of signals is achieved on a NASIC fabric without the need for arbitrary routing or additional multiplexers.

3. After all partial sums from neighbors have been accumulated, the new value of the output is calculated and the next partial sum generation cycle will begin. The CNN is mathematically proven to converge; after a specified number of iterations, the final output may be readout using the IO controller.

It is important to note here that this approach is fundamentally different from published CNN implementations using CMOS technology such as [9][11]. In these implementations, signals from the neighborhood are gathered together and all partial sums and states are computed locally. However, the unique manufacturing and device constraints of a 2D nanowire grid based fabric utilizing 2-level logic is that i) such arbitrary routing of signals is not possible ii) 2-level routing on the grid itself is extremely area inefficient [4].

Fig . 5 shows the circuit level implementation of the CNN cell. White boxes at certain crosspoints represent n-type transistors. The long nanowires through the center of the cell are for templates programmable from outside the cell array. While 2-level logic implementation (NAND-NAND) is used in these circuits, at some points along the state accumulation datapath, an additional routing stage is added for turning a signal in a perpendicular direction. This is required since arbitrary routing is not possible on a semiconductor nanowire grid based fabric.

IV. Discussion

Density comparisons of the NASIC CNN design were done against equivalent CMOS implementations at aggressively scaled technology nodes (Fig. 6). Since nanomanufacturing is expected to have high defect rates, NASIC designs use built-in defect tolerance schemes at various levels to improve yields. The introduction of additional components however leads to loss of density. In Fig. 6, densities of two simple schemes, 2-way redundancy and TMR are also compared against CMOS, which is assumed to be non-redundant and fault free. While a non-redundant scheme may have up to 2 orders of magnitude density benefit over CMOS, its yield is likely to be very low whereas a 2way + TMR scheme greatly reduces density benefits. More sophisticated techniques, such as based on error correction [1] are currently being explored for our design.

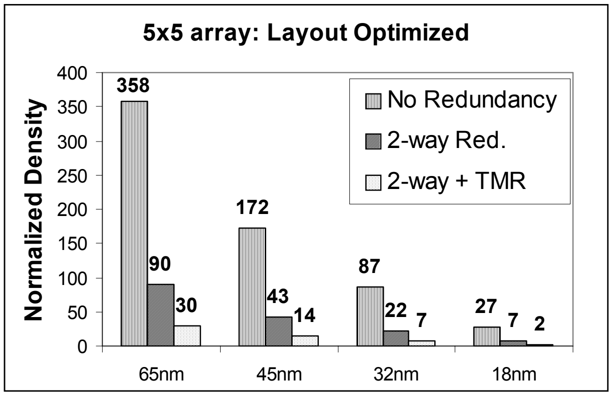

Fig. 6. Density comparison: NASIC CNN vs CMOS

The system does not impose any additional constraints on previously proposed manufacturing routes for the NASIC fabric [1]. Collective computation implies that output integrity may be preserved even in the presence of a few faulty components. Moreover, given its regular nature, precise alignment of masks for any lithographic steps are not necessary. It is hoped that improvements in manufacturing in conjunction with the NASIC fabric and built-in defect tolerance schemes could lead to reliable, high density image processing systems in the near future.

References

[1] C. A. Moritz, T. Wang, P. Narayanan, M. Leuchtenburg, Y. Guo, C. Dezan, and M. Bennaser, “Fault-Tolerant Nanoscale Processors on Semiconductor Nanowire Grids”, *IEEE Transactions on Circuits and Systems I,* vol 54, pp. 2422 -2437, November 2007.

[2] A. DeHon. “Nanowire-based programmable architectures”, *ACM Journal on Emerging Technologies in Compu1ting Systems,* vol 1, pp. 109-162, 2005.

[3] C. A. Moritz and T. Wang, “Towards Defect-Tolerant Nanoscale Architectures”, *Sixth IEEE Conference on Nanotechnology, IEEE Nano2006,* vol 1, pp. 331-334, 2006.

[4] C. A. Moritz and T. Wang, “Latching on the wire and pipelining in nanoscale designs”, *Non-Silicon Computing Workshop, NSC-3*, 2004.

[5] W. Lu and C.M. Lieber, "Semiconductor Nanowires," *J. Phys. D: Appl. Physics,* vol. 39, pp. R387-R406, October 2006.

[6] Y. Cui, X. Duan, J. Hu, and C. M. Lieber, “Doping and Electrical Transport in Silicon Nanowires”, *Journal of Physical Chemistry B,* vol. 104, pp. 5213-5216, May 2000.

[7] D. B. Strukov and K. K. Likharev, “Reconfigurable Hybrid CMOS Devices for Image Processing”, *IEEE Transactions on Nanotechnology,* vol 6, pp. 696-710, November 2007.

[8] L. O. Chua and L. Yang, “Cellular Neural Networks: Theory”, *IEEE Transactions on Circuits and Systems I,* vol. 35, pp.1257-1272, October 1988.

[9] M. Sindhwani, T. Srikanthan and K. Vijayan Asari, “VLSI efficient discrete-time cellular neural network processor”, *IEE Proceedings on Circuits, Devices and Systems,* vol. 149, pp. 167-171, June 2002.

[10] A. Khitun and K. L. Wang, “Cellular Nonlinear Network based on semiconductor tunneling nanostructure”, *IEEE Transactions on Electron Devices,* vol 52, pp. 183-189, Feb. 2005.

[11] M.-D. Doan, M. Glesner, R. Chakrabaty, M. Hidenreich and S. Cheung, “Realization of a cellular neural network for image processing”, *Proc. of the 3rd IEEE Intl. Workshop on Cellular Neural Networks and their Applications,* pp. 85-90, December 1994.

[12] P. Narayanan, T. Wang, M. Leuchtenburg, C. A. Moritz, “Comparison of Analog and Digital Naroscale Systems: Issues for the Nano-architect”, *IEEE Nanoelectronics Conference,* March 2008. `