

Towards Defect-Tolerant Nanoscale Architectures

Csaba Andras Moritz

Department of Electrical and Computer Engineering
University of Massachusetts in Amherst
Amherst, Ma 01002
E-mail: andras@ecs.umass.edu

Teng Wang

Department of Electrical and Computer Engineering
University of Massachusetts in Amherst
Amherst, Ma 01002
Email: twang@ecs.umass.edu

Abstract—Nanoscale computing systems show great potential but at the same time introduce new challenges not encountered in the world of conventional CMOS designs and manufacturing. For example, these systems need to work around layout and doping constraints resulting from unconventional bottom-up self-assembly, and need to cope with high manufacturing defect rates and transient faults. Unfortunately, most conventional defect-tolerance techniques are not directly applicable in nanoscale systems because they have been designed for very small defect rates. In this paper, we explore built-in defect-tolerance techniques on 2-D semiconductor nanowire (NW) arrays to make designs self-healing. Our approach combines circuit and system-level techniques and it does not require defect map extraction, reconfigurable devices, or addressing each cross-point similar to reconfigurable approaches. We show that a defect-tolerant simple processor based on our approach would be still around 3X denser than an 18-nm CMOS version with equivalent functionality; a yield greater than 30% is achieved despite a fabric with 14% defective FETs.

Keywords-semiconductor nanowire; defect tolerance, processor

I. INTRODUCTION

There are many novel devices under development such as based on unique molecular structures, carbon nanotubes (CNT), and semiconductor nanowires, and arrays of crossed NWs. Researchers have already built FETs and diodes out of NWs [7]. Complementary depletion-mode FETs in the same material have been demonstrated with Germanium [6] and Silicon [1]. Considerable progress has been made on assembling arrays with such devices using either unconventional lithographic techniques or bottom-up self-assembly [8]. This rapid progress is driving researchers to explore possible new nanoscale architectures. Examples of proposed architectures include [2], [5], [9], [15], [16].

This paper focuses on defect-tolerance techniques on 2-D nanowire array based fabrics and explores a defect-tolerant nanoscale processor. It is extending ideas presented in [15] and adds system-level techniques in CMOS.

Most nanoscale defect-tolerance techniques proposed are based on reconfiguration [3], [5], [9]. By contrast, our solution for defect-tolerance is based primarily on built-in circuit-level redundancy in a cascaded AND-OR logic family [12], [15]. Additionally, we combine these circuit-level techniques with system-level CMOS voting using TMR [10] to further improve the yield.

We base our work on Nanoscale Application-Specific IC (NASIC) fabrics [13], [16]. To explore the benefits of the proposed techniques, we develop and evaluate a defect-tolerant Wire Streaming Processor [14]. WISP-0 is a simple but complete stream processor that exercises many different NASIC circuit styles and optimizations.

Compared with reconfiguration-based approaches, our self-healing techniques eliminate the need for defect map extraction, do not require reconfigurable devices, and dispense with the complex nano-micro interfacing/decoder required to address each crosspoint in a reconfigurable fabric. Our preliminary results show that a WISP-0 processor with defect tolerance has a 3X density advantage compared to equivalent 18-nm CMOS implementation. The CMOS version is synthesized with modern CAD tools and scaled to 18-nm. The resulting yield for WISP-0 is 30% even in the presence of 14% defective transistors.

The rest of this paper is organized as follows: Section 2 provides a brief overview of NASICs and the architecture of the WISP-0 processor. Defect-tolerance techniques are introduced and discussed in Section 3. Results are presented in Section 4. Section 5 concludes the paper.

II. NASIC FABRICS AND WISP-0 PROCESSORS

NASIC designs use FETs on 2-D semiconductor NW arrays to implement logic functions. Various optimizations are applied to work around layout and manufacturing constraints [14], [16]. While based on 2-level AND-OR logic style, NASIC designs are optimized according to specific applications to achieve high density. Figure 1 demonstrates the design of a 1-bit full adder in dynamic style. By using dynamic circuits and pipelining on the wires, NASICs eliminate the need for explicit flip-flops and therefore can improve the density considerably [13].

WISP-0 is a stream processor that implements a 5-stage pipelined streaming architecture. Each stage is implemented in its own tile. NWs are used to provide communication between adjacent nanotiles. Each nanotile is surrounded by microwires (MWs) which carry ground, power supply voltage, and some control signals. Additionally, in order to preserve the density advantages of nanodevices, data is streamed through with minimal control/feedback paths. With the help of dynamic Nano-latches [13], intermediate values during processing are stored on the wire without requiring explicit latching. Support is assumed in the compiler to avoid hazards.

Figure 2 shows the layout. A nanotile is shown as a box surrounded by dashed lines. More details about the various circuits used can be found in [14]. In this paper, we use WISP-0 to evaluate the efficiency of our defect-tolerance techniques.

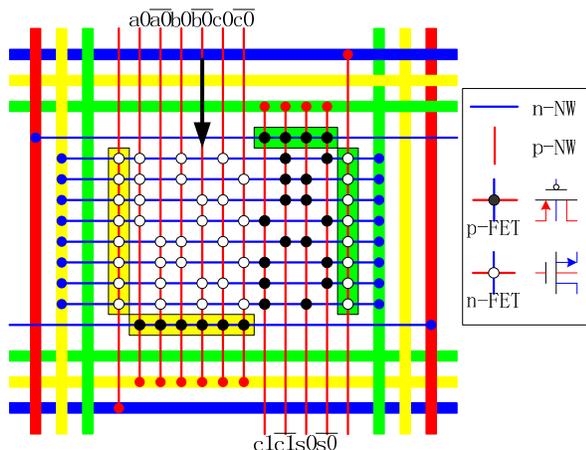


Figure 1 Dynamic NASIC implementation of a 1-bit full adder. The thicker wires represent microwires (MWs), the thin ones are NWs. The black and white dots, at NW crosspoints, represent p-FETs and n-FETs respectively.

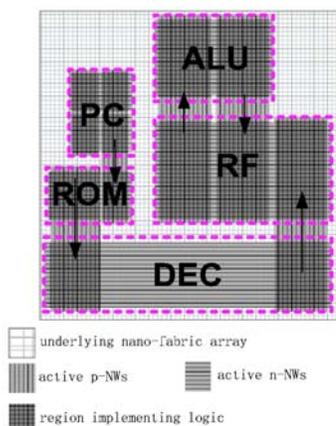


Figure 2 Floorplan of the WISP-0 Processor.

III. NASIC DEFECT-TOLERANCE APPROACH

Although the defect rates of nanoscale fabrics will likely improve with time, defect levels of nanodevices are expected to remain in the few percent range [7]. Larger-scale systems would likely have greater than 5% defects. We are not considering defect rates greater than 15% as we believe such fabrics would unlikely become practical.

A. Defect Model Assumed

There are two main types of defects while building nanoscale systems: NWs may be broken and the FETs at the NW crosspoints can be defective. FETs may be stuck-short (channel is always open) or stuck-open (channel is always off).

A stuck-open transistor can be treated as a broken NW; a stuck-short transistor means no active transistor at the crosspoint.

B. Possible Directions for Defect Tolerance

Basically there are two main approaches that can be followed. First, if reconfigurable devices are available, we could devise techniques to work around defects in a fabric. Reconfigurable solutions need to address several challenges. One key challenge is accessing crosspoints in the fabric for the purpose of reconfiguration. That requires a special interface between the micro and the nanodevices. Such an interface involves a large number of extra MWs - a high area overhead and a major manufacturing challenge due to the required alignment between the NWs and the MWs. No proposals with exception of perhaps CMOL [9] address this issue in a practical way as yet. Additional fundamental issues include extracting defect maps, reconfiguration algorithms, and the availability of reconfigurable devices.

Alternatively, as proposed here, we can make the circuits and the architectures self-healing by adding redundancy and by modifying a design such that it becomes more tolerant to defects and faults. We classify our self-healing approach into four techniques: circuit-level built-in redundancy, NW interleaving, weak pull-up/down NWs, and system-level Triple Modular Redundancy (TMR) [10].

C. Circuit-Level Built-In Redundancy

Figure 3 shows a simple example of a NASIC circuit implementing an *AND-OR* logic function with built-in redundancy. To make the masking mechanism work, we modify the dynamic circuit style reported in our prior work [13]. We use different schemes for horizontal and vertical NWs. As shown in the figure, horizontal NWs are precharged to “0” and then evaluated. For vertical NWs, they are instead precharged to “1” and then evaluated. The circuit implements the logic function $o = ab+c$; a' is the redundant copy of a and so on.

A NASIC design is effectively a connected chain of *AND-OR* logic planes. Our objective is to mask defects either in the logic stage where they occur or following ones. For example, a break on a horizontal NW in the *AND* plane (see for example position “A” in the figure) causes the signal on the NW to be “0”. This is because the NW is disconnected from V_{dd} . The faulty “0” signal can, however, be masked by the following logic *OR* plane if the corresponding duplicated/redundant NW is not defective.

A NW break at position “B” can be masked by the *AND* plane in the next stage. Similar masking can be achieved for breaks on vertical NWs. Stuck-open FETs can be modeled with broken nanowires; the defect tolerance would work as described above. For stuck-short FETs, the situation is relatively simpler as each FET has its redundant copy: if one of the two transistors is stuck-short (no active transistor at crosspoint), the circuit still works.

D. Improving Defect-Tolerance by Interleaving NWs

While the previous technique can mask many types of defects, faults at certain positions are difficult to mask. For

example, if there is a break at position “C” in Figure 3, the bottom horizontal NW is disconnected from ground. The signal on this NW will be set to logic “1”. Because of *OR* logic on the vertical NWs, the two vertical NWs would be always set to logic “1”.

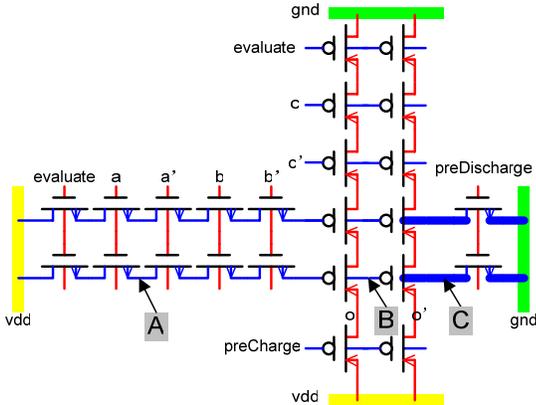


Figure 3 Simple NASIC circuit with built-in redundancy.

Clearly, if the NW break is in a specific region, it cannot be masked easily in subsequent *AND* or *OR* planes. We call these regions *hard-to-mask* (shown as the thicker segments in Figure 3).

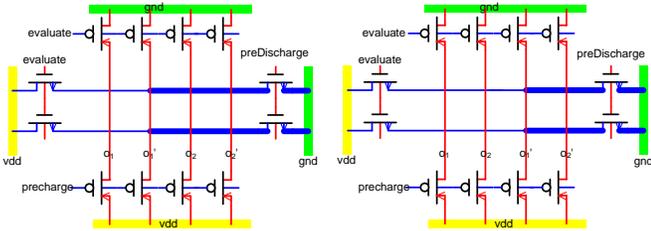


Figure 4 Interleaving NWs to reduce hard-to-mask regions. The right circuit has interleaved vertical NWs.

By carefully interleaving NWs, however, we can reduce hard-to-mask regions to minimum. Assuming we have two outputs (o_1 and o_2) in a nanotile, Figure 4 illustrates how interleaving shrinks these regions. Note that the order of NWs is changed from o_1, o_1', o_2, o_2' to o_1, o_2, o_1', o_2' . (The input NWs and some transistors are not shown in the figure for better clarity.) Interleaving is also helpful in masking clustered defects because duplicated NWs are set apart.

E. Adding Weak Pull-UP/Down NWs

Even after built-in redundancy and careful interleaving, there are still hard-to-mask regions remaining: see for example the regions in the right circuit in Figure 4. A possible solution to mitigate this problem is to insert weak pull-down vertical NWs between the *AND* and *OR* planes. The idea is to pull down (or up depending on logic plane) floating inputs, due to broken NWs, that would cause logic faults: e.g., a floating “1” input before an *OR* plane would make the *OR* logic always compute “1”. Modifying floating signals to a preferred logic level would allow masking in following logic planes.

A weak pull-down NW does not change correct operations if there are no defects, but introduces a performance tradeoff when there are defects by slowing the circuit down somewhat. Additionally it adds leakage power. At each crosspoint between a vertical pull-down NW and horizontal NWs there is a resistance created. This resistance has to be made larger than the switch-on resistance (estimated to be smaller than 10M Ω according to [10]) of a depletion-mode FET and smaller than the switch-off resistance (over 100 Ω). We are currently building a detailed Spice simulator that would enable us to explore the performance tradeoffs due to these added NWs. To ease manufacturing we could also use MWs instead of the NWs implementing weak pull-up/down wires.

F. Adding CMOS TMR

Voting based techniques such as TMR have been used extensively before. To be efficient, voting requires that the probability of a defect in the voting circuit is much smaller than in the design it is applied to. This is clearly the case in conventional technology. TMR is not applicable as is in NASIC designs because at 10-15% fabric defect rates the TMR circuits themselves would be likely defective.

Nevertheless, in pipelined processor designs one could add TMR at certain points in a design in CMOS, without affecting throughput significantly. If each nanotile has two extra identical replicas, we could vote either at each stage or on the final outputs. Voting helps where the other techniques leave faulty outputs. In the following section, we show results for each of the techniques presented applied to the WISP-0 processor design.

IV. RESULTS

By simulating WISP-0 with randomly generated defects and comparing the outputs with a defect-free design, we evaluate the efficiency of our techniques for tolerating defective FETs and broken NWs. We develop an equivalent CMOS WISP-0 version in Verilog and compare the area of the scaled CMOS WISP-0 with the nanoscale WISP-0 after defect tolerance techniques are included.

A. Defect Tolerance and Yield Results

An assumption we make in the simulation is that defects are evenly distributed along NWs and among transistors. We do not consider clustered defects (which are mitigated somewhat by our NW interleaving but exploring that is beyond the scope of this paper).

Figure 5 shows the yield of WISP-0 assuming some defective transistors and Figure 6 shows the yield of WISP-0 with broken NWs. There are 8 curves in each figure, each of them representing one configuration of WISP-0 with a combination of defect-tolerance techniques applied. The figures show that the defect-tolerance techniques considerably improve yield. Even if the defect rate of FETs reaches 14%, the yield still remains greater than 30%. If the defect rate of broken NWs is 10%, the yield is over 19%.

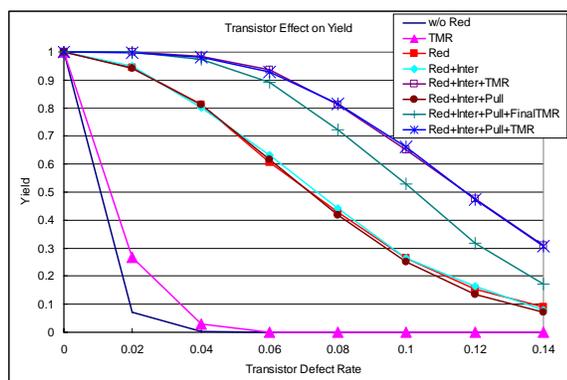


Figure 5 Yield with different defect-tolerance techniques and assuming various rates of defective FETs. Notation: Red shows WISP-0 with built-in redundancy; Inter means interleaving of NWs; Pull means applying weak pull-up/down NWs; and TMR refers to CMOS TMR added.

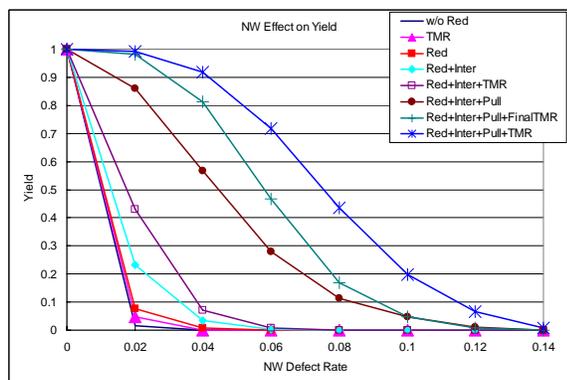


Figure 6 The yield achieved with different techniques when considering broken NWs.

Interleaving and weak pull-up/down NWs (or MWs) do not improve the yield of WISP-0 with defective transistors considerably, but significantly improve it on fabrics with broken NWs.

B. Comparison with Equivalent CMOS Processor

Figure 7 shows the normalized density of WISP-0 with different defect-tolerance techniques. The baseline is the equivalent CMOS design of WISP-0 that we have implemented in Verilog-HDL and synthesized/scaled to various process technology nodes. Roughly speaking, our self-healing techniques increase the area of the original WISP-0 by around 3X. System-level TMR related copies increase the area by another 3X. If a 3-D layout of CMOS circuits and nanoarrays described in [11] turns out to be possible, the voting circuits can be overlapped with the nanoarray. The density of WISP-0 with redundancy (without TMR) is in fact higher than the density without redundancy but with a micro-nano decoder required in a reconfigurable solution. Compared with CMOS implementations, the nanoscale WISP-0 preserves its density advantage. Even at 18-nm CMOS, available in 12 years according to ITRS 2005, the self-healing WISP-0 design

combined with system-level TMR would be still around 3X denser than the equivalent CMOS version.

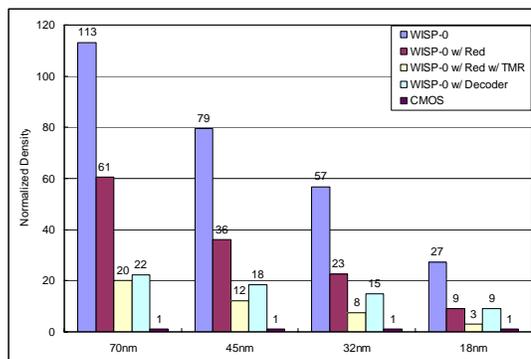


Figure 7 Density comparison NASIC and CMOS WISP-0.

REFERENCES

- [1] Y. Cui and C. M. Lieber, "Functional nanoscale electronic devices assembled using silicon nanowire building blocks, *Science*, 2001.
- [2] A. DeHon, "Array-based architecture for FET-based, nanoscale electronics", *IEEE Transactions on Nanotechnology*, 92(1), 2003.
- [3] A. DeHon, "Nanowire-based programmable architectures", *ACM Journal on Emerging Technologies in Computing Systems*, 1(2), 2005.
- [4] A. DeHon and M. J. Wilson, "Sublithographic programmable logic arrays", *Proceedings of the International Symposium on Field Programmable Gate Arrays, FPGA'04*, 2004.
- [5] S. C. Goldstein and M. Budiu, "Nanofabrics: Spatial computing using molecular electronics", *The 28th Annual International Symposium on Computer Architecture, ISCA'01*, 2001.
- [6] A. B. Greytak, L. J. Lauhon, M. S. Gudiksen, and C. M. Lieber, "Growth and transport properties of complementary germanium nanowire field-effect transistors, *Applied Physics Letters*, 84(21), May 2004.
- [7] Y. Huang, X. Duan, Y. Cui, L. Lauhon, K.-Y. Kim, and C. Lieber, "Logic gates and computation from assembled nanowire building blocks, *Science*, 1313(294), 2001.
- [8] Y. Huang, X. Duan, Q. Wei, and C. Lieber, "Directed assembly of one-dimensional nanostructures into functional networks", *Science*, 2001.
- [9] K. K. Likharev, "CMOL: Devices, circuits, and architectures", *Introducing Molecular Electronics*, 2004.
- [10] R. E. Lyons and W. Vanderkulk, "The use of triple modular redundancy to improve computer reliability", *IBM Journal of Research and Development*, 6(2), 1962.
- [11] X. Ma, D. Strukov, J. H. Lee, and K. Likharev, "Afterlife for silicon: CMOL circuit architectures", *5th IEEE Conference on Nanotechnology*, 2005.
- [12] C. A. Moritz, "Exploring NASICs and a comparison with CMOL: an architect's perspective", *Third Advanced Research and Development Agency, workshop*, 2006.
- [13] C. A. Moritz and T. Wang, "Latching on the wire and pipelining in nanoscale designs", *Non-Silicon Computing Workshop, NSC-3*, 2004.
- [14] T. Wang, M. Bennaser, Y. Guo, and C. A. Moritz, "Wire-streaming processors on 2-D nanowire fabrics", *Nanotech 2005, Nano Science and Technology Institute*, May 2005.
- [15] T. Wang, M. Bennaser, Y. Guo, and C. A. Moritz, "Self-healing wire-streaming processors on 2-d semiconductor nanowire fabrics", *Nanotech 2006. Nano Science and Technology Institute*, May 2006.
- [16] T. Wang, Z. Qi, and C. A. Moritz, "Opportunities and challenges in application-tuned circuits and architectures based on nanodevices", *1st ACM Conference on Computing Frontiers*, pp 503--511, New York, USA, 2004