Self-Healing Wire-Streaming Processors on 2-D Semiconductor Nanowire Fabrics

Teng Wang, Mahmoud Bennaser, Yao Guo, Csaba Andras Moritz

Electrical and Computer Engineering Department
University of Massachusetts Amherst, MA, USA
{twang, mbennase, yaoguo, andras}@ecs.umass.edu

Abstract

With recent promising progress on nanoscale devices including semiconductor nanowires and nanowire crossbars, researchers are trying to explore the possibility of building nanoscale computing systems. We have designed a nanoscale application-specific architecture called NASIC, which is based on semiconductor nanowire grids and FETs at crosspoints. In this paper, we propose a built-in redundancy technique to tolerate the defects in our nanoscale architecture. Compared to other fault-tolerance techniques, our solution has significant advantages including self-healing, higher density. We evaluate the efficiency of self-healing technique and provide the density comparison with deep sub-micron CMOS technology.

Keywords: nanoscale circuits, architecture, NASIC, self-healing, fault-tolerance

1 Introduction

The most promising underlying nanodevice technologies today for nanoscale integrated circuits are semiconductor nanowires (NWs) and arrays of crossed NWs. Researchers have already built FETs and diodes out of NWs [5]. Complementary depletion-mode FETs in the same material have been demonstrated with Germanium [4] and Silicon [1]. There has also been a lot of progress made on assembling arrays with such devices using both unconventional lithographic techniques and bottom-up self-assembly. The rapid progress on devices is driving researchers to explore possible circuits/architectures out of them. Examples of proposed architectures include [2] [3] [6] [9].

The fabric architecture we proposed is called Nanoscale Application-Specific IC (NASIC) [8] [9] [10]. Wire Streaming Processor (WISP-0) is a simple but complete stream processor that exercises many of different NASIC circuit styles and optimizations. These previous efforts focused on circuit level optimizations that set apart our work from the other nanoscale proposals. In this paper we focus on another key distinguishing aspect: fault tolerance. As discussed by several researchers, fault tolerance is expected to be a key issue in nanoscale designs. Our solution for fault tolerance of NASICs is based on built-in circuit-level redundancy which makes the circuits in our designs self-healing.

2 Overview of NASIC Designs and WISP-0 Processor

NASIC designs use FETs on 2-D semiconductor NWs to implement logic functions and various optimizations to work around layout and manufacturing constraints as well as defects. While still based on 2-level *AND-OR* logic style, our designs are optimized according to specific applications to achieve higher density and self-healing. NASIC circuits are based on a new type of dynamic circuitry [8]. Figure 1 demonstrates the design of a 1-bit full adder in dynamic style. By using dynamic circuits, we eliminate the need of flip-flops and therefore improve the density considerably.

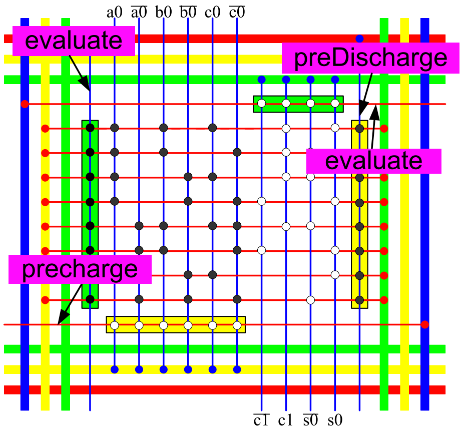

Figure 1: The dynamic implementations of 1-bit full adder. The thicker wires represent MWs and the thin ones are NWs. The doping type of the wires (p-type or n-type) along source-drain of a FET transistor determines the type of the transistor. The black and white dots at the crosspoints of NWs in the figure represent p-FET and n-FET respectively.

WISP-0 is a stream processor (based on self-healing NASIC circuits) that implements a 5-stage pipeline architecture including *fetch*, *decode*, *register file*, *execute* and *write back*. WISP-0 consists of five nanotiles, as shown in Figure 2. A nanotile is shown as a box surrounded by dashed lines. NWs are used to provide communication between adjacent nanotiles. Each nanotile is driven by the surrounding NWs, which are not shown in the figure. In WISP designs, in order to preserve the density advantages of nanodevices, data is streamed through the fabric with minimal control/feedback paths. With the help of dynamic Nano-latches [8], intermediate values during processing are often stored on the wire without requiring explicit latching. The compiler is responsible of generating code such that data hazards are avoided.

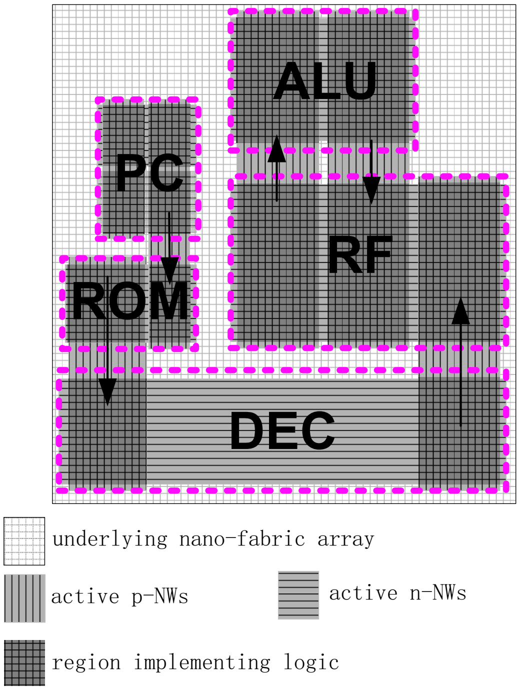

Figure 2: The floorplan of WISP-0.

3 Fault Tolerance Strategy

Nanoscale computing systems face challenges not encountered in the world of traditional microelectronic devices. Although the manufacturing process is improving rapidly, the defect levels in nano-fabrics are still close to a few percent range [5]. This fact makes fault tolerance a critical aspect in nanoscale systems. We have to build enough fault-tolerance to sustain functionality when a substantial fraction of circuits are faulty.

Their are two main kinds of faults while building nanoscale systems: NWs may be broken and the transistors at the crosspoints may be stuck-short or stuck-open. A stuck-open transistor can be treated as a broken NW and a stuck-short transistor means there is not transistor at this crosspoint. Another assumption we make in this paper is that faults are distributed evenly along NWs and among transistors: we do not therefore consider clustered faults.

Some traditional techniques for fault tolerance may not suitable at nanoscale. An example of such techniques is Triple Modular Redundancy (TMR) [7]. By replicating modules two times and voting on three versions of results, some faults could be masked. However, due to layout constraints on 2-D fabrics, voting at circuit level cause large overhead in area. In addition, voting circuits at nanoscale themselves could be faulty. Replicating voting circuits can solve this problem partly but this would increase area overhead further. Our simulation shows TMR at circuit level can not improve the yield of nanoscale computing systems.

Basically two main feasible approaches are applied to deal with faults at nanoscale. First, if reconfigurable devices are available, we could possibly devise techniques to work around faults. One key challenge in such solutions is accessing crosspoints in the fabric. That requires a special interface between the micro and the nanodevices and such an interface is not only presenting a high area overhead but it is also very difficult to do due to the required alignment between the nano and the micro wires. No proposals with exception of perhaps CMOL [6] (that has other challenges) address this issue in a credible way.

Alternatively, as proposed by this work, we can make the circuits and the architectures self-healing by replicating NWs and transistors in the same nanotile. As shown in Figure 3, this circuit implements a logic function $ab+c$ with redundancy for self-healing. Most faults can be automatically masked by the *AND-OR* logic either in the current stage or the next one. For example, the break on a horizontal NW in the *AND* plane (e.g. position “A” in Figure 3) causes the signal on this NW to be “0” because the NW is disconnected to V_{dd}. But this faulty “0” can be masked by the following *OR* plane if the corresponding duplicated NW is working correctly. A break at position “B” in Figure 3 can also be masked by the *AND* plane in the next stage. Similarly masking can be achieved for the breaks on the vertical NWs. Of course, if both NWs are broken, we can not get correct results. However, the possibility of this case is very low.

There are some faults that this circuit can not tolerate. For example, if there is a break at position “C” in Figure 3, the two vertical NWs are both set to “1” always. This case occurs only when the break happens in specific regions of the nanotile. By carefully aligning NWs in an interleaving way, we can reduce the area of these regions to minimum. Interleaving also protects our designs against clustered defects because such defects can not affect on two duplicated NWs at the same time.

However, even combined with interleaving, we still

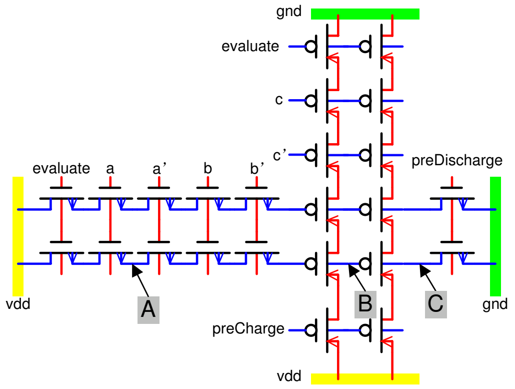

Figure 3: An example of self-healing circuit. "A", "B" and "C" indicate different break positions on a horizontal NW.

can not cover all the faults. To solve this problem, we propose to insert *weak* a pull-down vertical NW between *AND* and *OR* logic planes. At each crosspoint between this vertical NW and horizontal NW there is a resistance which can pull faulty "1" back to "0". As we discussed before, faulty "0" can be masked. Thus we improve the fault-tolerance of our self-healing circuits considerably with the cost of some speed reduction due to these resistances. Similarly, A pull-up NW is needed between two nanotiles.

Compared with other fault-tolerance techniques, our self-healing designs have the following advantages: First, the faults are automatically masked such that there is no need for defect map, which is very difficult to extract at nanoscale. Second, there is no additional requirements on manufacturing, such as special micro-nano interface to access each crosspoint required by reconfiguration approach.

4 Simulation Results

We apply the proposed self-healing techniques to WISP-0, including replicating, interleaving and weak pull-up/down NWs. By simulating WISP-0 with randomly generated defects and comparing the outputs with defect-free WISP-0, we evaluate the efficiency of our technique on tolerating both faulty transistors and broken NWs.

Figure 4 shows the yield of WISP-0 achieved when assuming faulty transistors and Figure 5 shows the yield of WISP-0 with broken NWs. Our built-in redundancy works very well and the combined techniques improve yield considerably. Without the self-healing mechanism, the yield goes down rapidly to 0 when the defect rate is more than 4%. With the exception of compensating faults, without fault tolerance the presence of faults cause incorrect execution. For self-healing NASICs, however, even if the defect rate of transistors reaches 10%, the yield remains over 30%. If the defect rate of broken NWs is 10%, the yield remains over 10%. Notice that although interleaving and weak pull-up/down NWs do not improve the yield of WISP-0 with faulty transistors, they help a lot to improve the yield of WISP-0 with broken NWs.

We can likely improve the yield further by combining self-healing techniques with system-level approaches such as reconfiguration and TMR. As we mentioned in the previous section, reconfiguration and TMR at the circuit level will introduce large area overhead. However, the overhead is expected to decrease at the system level.

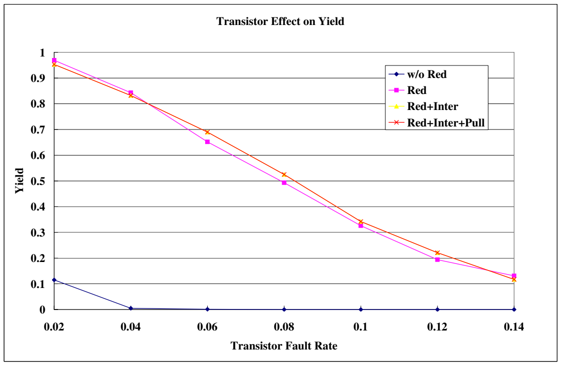

Figure 4: The yield achieved with different techniques (*Red* means WISP-0 with redundancy. *Inter* means interleaving of NWs. *Pull* means applying weak pull-up/down NWs) when considering faulty transistors.

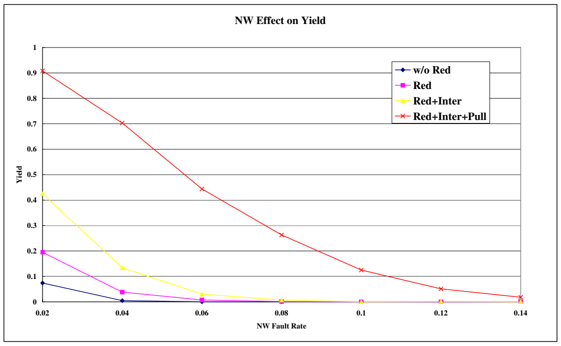

Figure 5: The yield achieved with different techniques when considering broken NWs.

Complex computation can be split into several simpler stages using cascading. We need to tell whether the cascading of nanotiles has impacts on self-healing mechanism. If the probability of correct outputs keeps going down when increasing the number of pipeline stages, our self-healing techniques would not be suitable for designs in large scale .

As an example, a carry-ripple adder is simulated. We increase the number of stages from 1 to 32. The results are shown in Figure 6. We can see that the probability of correct outputs drops fast at the early

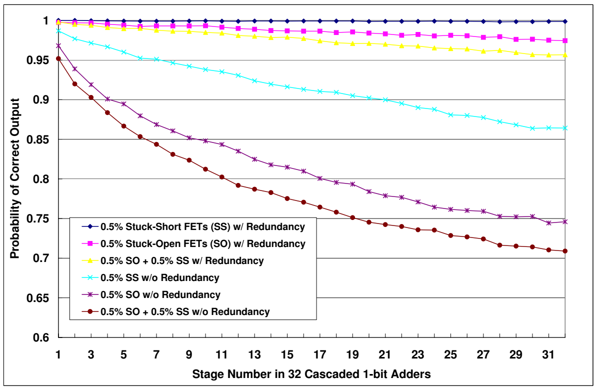

Figure 6: The probability of correct outputs for 32 cascaded adders. It converges to a certain point when increasing the cascading stages. Different defect parameters result in different converging points.

stages but the speed tends to slow down. The probability of correct outputs converges to a certain level in the end. This fact is due to the fault masking property of cascaded circuits. As mentioned in the previous section, faulty signals could be masked in the same or next stage. The probability of correct outputs converges when the masking effect and the fault generation along the pipeline reach an equilibrium. The converging point is determined by the defect parameters and the masking ability.

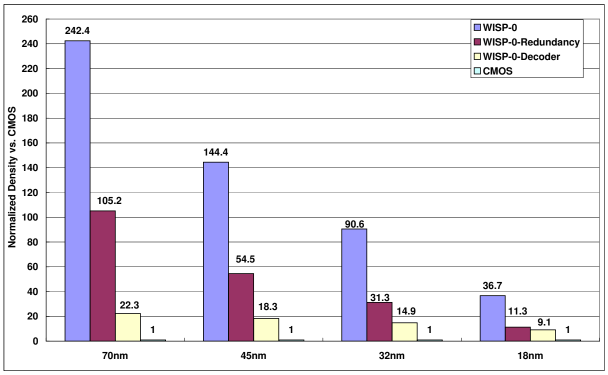

Figure 7: Normalized density of WISP-0 under different configurations compared with equivalent designs of 70, 45, 32, 18-nm CMOS technology.

We also estimate the impact of our self-healing techniques on density. As shown in Figure 7, the redundancy techniques cause around 2× area overhead for WISP-0 (see "WISP-0" and "WISP-0-Redundancy" in Figure 7). However, our solution eliminates the need of accessing each crosspoint for reconfiguration. The density of WISP-0 with redundancy is in fact better than the density without redundancy but with a micro-nano decoder. Compared with deep sub-micron CMOS technology, our self-healing design has great density advantage. Even at 18-nm CMOS, available in 12 years according to ITRS 2005, our self-healing design would still be over 10× denser than the equivalent processor design in 18-nm CMOS.

5 Conclusion

We proposed to introduce built-in redundancy into our NASIC designs to make them self-healing. The simulation results showed our self-healing techniques improve the yield considerably and are suitable for large scale designs. Compared to other fault-tolerance techniques, our self-healing solution is easier to implement on 2-D nano-fabric and has fewer requirements (e.g., complete defect map is not required).

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] S. C. Goldstein and M. Budiu. Nanofabrics: Spatial computing using molecular electronics. In *The 28th Annual International Symposium on Computer Architecture, ISCA'01*, 2001.

[4] 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.

[5] 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.

[6] K. K. Likharev. CMOL: Devices, circuits, and architectures. *Introducing Molecular Electronics*, 2004.

[7] R. E. Lyions and W. Vanderkulk. The use of triple modular redundancy to improve computer reliability. *IBM Journal of Research and Development*, 6(2), 1962.

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

[9] T. Wang, M. Bennaser, Y. Guo, and C. A. Moritz. Wire-streaming processors on 2-D nanowire fabrics. In *Nanotech 2005*. Nano Science and Technology Institute, May 2005.

[10] T. Wang, Z. Qi, and C. A. Moritz. Opportunities and challenges in application-tuned circuits and architectures based on nanodevices. In *CF '04: Proceedings of the 1st conference on Computing frontiers*, pages 503–511, New York, NY, USA, 2004. ACM Press.