FastTrack: Towards Nanoscale Fault Masking with High Performance

Md Muwyid U. Khan, Pritish Narayanan, Prachi Joshi, Pavan Panchapakeshan, and Csaba Andras Moritz

Abstract – High defect rates are associated with novel nanodevice-based systems owing to unconventional and self-assembly based manufacturing processes. Furthermore, in emerging nanosystems, fault mechanisms and distributions may be very different from CMOS due to unique physical layer aspects, and emerging circuits and logic styles. Development of analytical fault models for nanosystems is necessary to explore the design of novel fault tolerance schemes that could be more effective than conventional schemes. In this paper, we first develop a detailed analytical fault model for the Nanoscale Application Specific Integrated Circuits (NASIC) computing fabric and show that the probability of 0-to-1 faults is much higher than of 1-to-0 faults. We then show that in fabrics with unequal fault probabilities, using biased voting schemes, as opposed to conventional majority voting, could provide better yield. However, due to the high defect rates, voting will need to be combined with more fine-grained structural redundancy for acceptable yield. This entails degradation in performance (operating frequency) due to an increase in circuit fan-in and fan-out. We, therefore, introduce a new class of redundancy schemes called FastTrack that combine non-uniform structural redundancy with uniquely-biased nanoscale voters to achieve greater yield without a commensurate loss in performance. A variety of such techniques are employed on a Wire Streaming Processor (WISP-0) implemented on the NASIC fabric. We show that FastTrack schemes can provide 23% higher effective yield than conventional redundancy schemes even at 10% defect rates along with 79% lesser performance degradation.

***Index Terms*—nanofabric, NASIC, defect tolerance, FastTrack, effective yield, performance.**

I. Introduction

Manufacturing techniques for nanoscale systems whether based on bottom-up self-assembly or a purely top-down lithographic approach [1]-[4] are likely to produce defects on the order of billions per square centimeter. With such high defect rates, redundancy must be introduced at multiple system levels in order to build functional systems with acceptable yield.

Prior work on fault tolerance includes techniques such as reconfiguration and built-in redundancy. Techniques for reconfigurable fabrics include mapping of logic functions onto defective circuits or reconfiguring around defective blocks [5] - [8] and built-in self test techniques for testing and diagnosis [9][10]. In hardware redundancy techniques for non-reconfigurable crossbar architectures, previous work includes the efficient mapping of logic functions onto defective crossbars [11]-[13]. Both these approaches lead to technical challenges such as the need for special reconfigurable devices or the complex interfacing between micro and nano circuits to extract defect maps.

Modular redundancy techniques have been widely researched in the past few decades and include Triple modular redundancy (TMR) [14]-[17] and N-tuple modular redundancy [18]. Designs of new voter circuits [19][20] and even NMR systems without a centralized voter [21] have been proposed. A more fine-grained built-in fault tolerance technique is the structural redundancy [22]. As the focus shifts to nanoscale devices, hardware redundancy techniques still hold promise [23]-[25].

However, nanoscale computational fabrics may have very different fault models compared to conventional CMOS due to novel circuit and logic styles and different defect scenarios owing to unconventional manufacturing. It is thus necessary to first analyze the fault mechanisms and distributions in nanoscale fabrics to facilitate the design of fault-tolerance schemes that are better tailored to these unique fault models. In the analytical fault model for the Nanoscale Application Specific Integrated Circuits (NASIC) computational fabric presented in this paper, we find that '0' to '1' faults (referred to as faulty '1' henceforth for convenience) are more likely than '1' to '0' faults (faulty '0'). This provides an opportunity to potentially achieve higher yields by using biased voters which offer greater protection against the most likely faults and less against the others.

On the other hand, at very high defect rates, modular redundancy alone may not be sufficient since many or all redundant modules could be faulty. It is therefore necessary to incorporate some level of structural redundancy [22][26] within individual modules in conjunction with modular redundancy. However, while these techniques have proven effective for yield purposes, they cause degradation in performance (mean operating frequency) due to the increase in fan-in and fan-out that makes circuits slower.

To achieve the twin targets of yield and performance, a new suite of fault tolerance techniques called FastTrack is proposed. In this approach, input modules to a given biased voter have different levels of structural redundancy. Signals from modules

Manuscript received July 23, 2010; revised July 27, 2011 and November 8, 2011; accepted February 29, 2012. This work was supported in part by the Center for Hierarchical Manufacturing (CHM) at UMass Amherst, and NSF Awards CCR:0105516, NER:0508382, and CCR:051066. This work is partially based on a conference submission to IEEE DFT 2011[31]

The authors are with the University of Massachusetts, Amherst, MA 01002 USA. (email: mukhan@ecs.umass.edu; andras@ecs.umass.edu)

with higher redundancy have high reliability but are slower and vice versa. This imbalance in structural redundancy levels of input modules, in conjunction with biased voting, may be optimized/configured to meet yield and performance goals.

In this paper, we first analyze the fault model of the Nanoscale Application Specific Integrated Circuits (NASIC) computational fabric [22],[26]-[32] and show that unequal fault probabilities exist. Using the theoretical modeling approach, we show that biased voting schemes can potentially lead to higher yields compared to more conventional majority voting schemes in fabrics exhibiting such fault models. For further improvement of the yield without considerable degradation in performance, new FastTrack schemes are proposed that combine biased voters with non-uniform levels of structural redundancy across input modules. A variety of such schemes are introduced and evaluated. Simulations performed on the Wire Streaming Processor 0 (WISP-0) [35], built on the NASIC computational fabric demonstrate a 23% improvement in the effective yield (yield per unit area) for FastTrack schemes along with 79% lower degradation in mean operating frequency compared to conventional techniques at 10% defect rate. FastTrack schemes perform well across both the effective yield and performance metrics, and show an improvement of 122% - 268% in effective yield-performance products at defect rates of 8% - 12% compared to conventional schemes performing best at the same defect rates. While the approach is presented in the context of NASICs, where precise physical and circuit models are available, it can be easily extended to other nanofabrics relying on redundancy for fault masking.

The rest of the paper is organized as follows: Section II provides an analytical fault model of the NASIC fabric showing unequal probability of faulty '0' and '1's. Section III demonstrates the possibility of gaining higher yield using biased voters. Section IV describes FastTrack schemes for NASICs and the potential for improved yield and performance. Section V describes the simulation framework and the processor architecture used as the simulation test-case. Section VI provides simulation results for various FastTrack schemes. Section VII concludes the paper.

II. Analytical Fault Modeling For NASICs

In this section we present an overview of the NASIC fabric and use an analytical fault modeling approach [31] to demonstrate the unequal fault probabilities of faulty '0's and '1's. This motivates the use of biased voting schemes to improve yield followed by new directions for nanoscale voters.

A. *Fabric overview*

NASICs [22],[26]-[32] is a computational fabric based on a 2-D grid of semiconductor nanowires [2][33][34] with external dynamic control for data streaming and cascading. Fig. 1 shows a single NASIC tile (consisting of 2 dynamic NAND stages) implementing a 1-bit full adder. Many such tiles can be cascaded together to build a large-scale system such as a processor [35] or an image processing architecture [36].

Cross-nanowire transistors (xnwFETs) are formed at selected cross-points to implement the logic function. In NASICs, the 2-stage dynamic NAND-NAND logic style is one of the logic families used [29]. The output signals from the first stage NAND gates become the input signals for the nanowire transistors in the second stage NAND gate as shown in Fig. 1. In Fig. 1, *hpre* and *heva* are the precharge and evaluate control transistors that enable dynamic circuit evaluation.

In NASICs, high fan-in circuits are possible since delay scales linearly with respect to fan-in as opposed to conventional CMOS where the trend is typically quadratic. This is due to the unique dynamic control schemes used, where successive cascaded stages are evaluated using different control signals. The series stack resistance of a given stage is overcome during and after the pre-charge of the previous stage. This implies that during the evaluation of the current stage, only the linear impact of capacitance affects the performance with increasing fan-in. This behavior has been verified through detailed simulations of device behavior and circuit characteristics. Additional details can be found in [37].

B. *Defect model*

Defects in the NASICs fabric depend on the manufacturing pathway used. One possible manufacturing pathway has been described in [38]. Reliable manufacturing of nanowires up to a few microns in length has been demonstrated in [1][39], so the

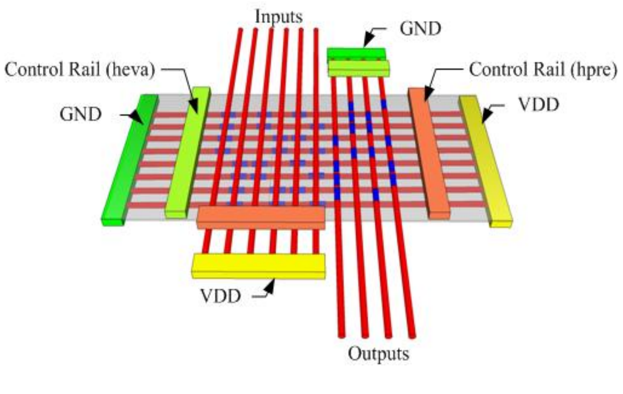

Fig 1. A full adder implemented in NASICs.

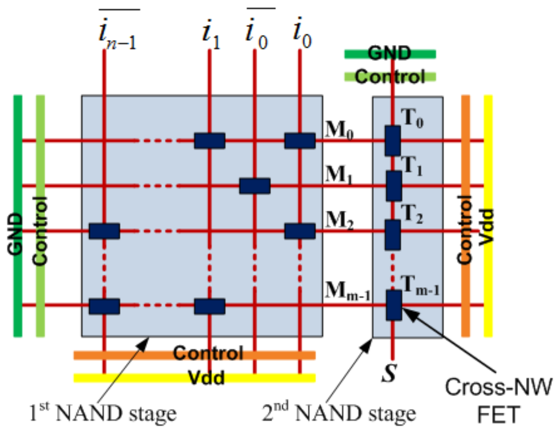

Fig 2. An n-input NASIC tile built with 2 NAND stages.

frequency of broken nanowires is assumed to be negligible. Stuck-on transistors are the most prevalent in this pathway due to the ion implantation and metallization processes involved. The probability of a transistor being stuck-on, denoted by P_d, thus represents the defect rate of the NASIC fabric. These defects are considered to occur independently of each other, since they are caused by local effects (e.g. lateral diffusion after ion implant).

A defect rate of up to 12% is considered because, according to our initial work [22], at defect rates higher than this, any density advantage over projected CMOS would likely be eliminated in the context of microprocessor designs. It must be noted that this is a device-level defect rate and is 10 orders of magnitude higher than in scaled CMOS. For instance, CMOS defect rates are 0.4 defects/cm^2 [40] whereas 1-12% defect rate in NASICs translates to billions of defects/cm^2.

C. *Notations used for analytical fault model*

Fig. 2 shows the two NAND stages in a single n-input NASIC tile. Here, input signals are denoted by *i*. M_0 to M_{m-1} are the minterms generated by the first NAND stage and T_0 to T_{m-1} denote the transistors in the 2nd stage NAND gate.

The notations that have been used in this analytical model are:

- n Number of inputs for the logic function implemented
- m Number of minterms generated by 1st NAND stage
- S^C Correct output of logic function
- S^A Actual output from a defective NASIC circuit implementing the logic function
- M_i^C Correct i^{th} minterm expected from a defect-free circuit, $0 \leq i \leq m\text{-}1$
- M_i^A Actual i^{th} minterm in a defective circuit
- T_i Transistor gated by minterm M_i in the *second* stage NAND gate
- $P_{i/j}$ Conditional probability of getting a module output of *i* given that the correct module output should have been *j* (*i,j=0,1*). Clearly, $P_{0/j}+P_{1/j}=1$, for $j=0,1$.
- $P_{1/0} = P(S^A=1|S^C=0)$ Probability of a faulty ‘1’ at the output of a defective circuit (module)
- $P_{0/1} = P(S^A=0|S^C=1)$ Probability of a faulty ‘0’ at the output of a defective circuit (module)
- $P_{Mf0} = P(M_i^A=0|M_i^C=1)$ Probability of the minterm, M_i, being a faulty ‘0’

D. *Occurrence of a faulty ‘1’*

In a 2-stage NAND-NAND logic implementation, logic ‘0’ can be produced at the output only if all of the input signals (minterms) to the second stage NAND gate are ‘1’s. Thus, in a NASIC tile, all of the xnwFETs in the second stage dynamic NAND gate must be correctly switched on to allow the output to evaluate to ‘0’. A single incorrectly switched off transistor in the series stack is sufficient to cause a faulty ‘1’ at the output. If T_i^C is the correct state of transistor T_i that is expected in a defect-free circuit and T_i^A is the actual state in a defective circuit, the probability of a faulty ‘1’ at the output can be written as,

$$P(S^A=1|S^C=0) = 1\text{-}P(S^A=0|S^C=0)$$
$$= 1\text{-}P(\textstyle\bigcap_{i=0}^{m-1} T_i^A = ON | T_i^C = ON) \qquad (1)$$

The transistor T_i will be correctly switched on if it is functional (not defective) and its gate signal, M_i, is logic high or if it is stuck-on.

$$P(T_i^A=ON|T_i^C=ON)$$
$$=\mathrm{P}(\mathrm{T_i}=\text{stuck-on} \cup (\mathrm{T_i} = \text{not stuck on } \cap \ \mathrm{M_i^A} = 1 |\ \mathrm{M_i^C} = 1))(2)$$

Since $P_{Mf0} = P(M_i^A=0|M_i^C=1)$ and $P(T_i=\text{stuck-on}) = P_d$,

$$P(T_i^A=ON|T_i^C=ON) = P_d+(1\text{-}P_d)(1\text{-}P_{Mf0}) \qquad (3)$$

$$P(S^A=1|S^C=0) = 1\text{-}[P_d+(1\text{-}P_d)(1\text{-}P_{Mf0})]^m$$
$$= 1\text{-}(1\text{-}P_{Mf0}+P_dP_{Mf0})^m \qquad (4)$$

The expression for P_{Mf0} is derived in the next sub-section.

E. *Occurrence of a faulty ‘0’*

In a defect-free and structurally non-redundant NASIC circuit implementing the NAND-NAND logic style, a single transistor in the 2nd NAND stage (denoted by T_x in this model) will be switched off to cause the output to produce a logic '1'. Hence, for a faulty ‘0’ to occur in a defective circuit, transistor T_x should be incorrectly switched on while all the other transistors remain correctly switched on, enabling evaluation to faulty ‘0’. Since incorrect ‘1’s at the minterms are not possible due to the defect model considered, transistor, T_x, would have to be defective (stuck-on) in order to be incorrectly switched on. Thus, $P(T_x^A = ON|T_x^C = OFF) = P_d$. The expression for the probability of a faulty ‘0’ will be,

$$\mathrm{P(S^A=0|S^C=1)} \quad = \quad \mathrm{P}(\{\textstyle\bigcap_{i=0;i\neq x}^{m-1} T_i^A = ON | T_i^C = ON\} \cap \{T_x^A = OM | TxC = OFF\}) \qquad (5)$$

Substituting (3) into (5), we get,

$$P(S^A=0|S^C=1) = P_d(1\text{-}P_{Mf0}+P_dP_{Mf0})^{m\text{-}1} \qquad (6)$$

In this model, the probability of a minterm being a faulty ‘0’, P_{Mf0}, is dependent on the transistors of the first stage NAND gates being defective. In the development of an expression for P_{Mf0}, we assume that logic functions are implemented in a 2-level Sum of Products form. Hence, for a logic function with *n* inputs, every NAND gate in the first stage has *n* transistors. For a minterm to be a faulty ‘0’, one or more transistors needs to be stuck on depending on the input pattern. For instance, in a 3-input logic function implemented in the NASIC fabric, at any one of the first stage NAND gates, there are 7 input patterns that should produce an output of ‘1’ at that NAND gate. 1 out of those 7 input patterns will require all 3 transistors in the gate to be stuck-on to produce a faulty ‘0’. 3 of the input patterns will require 2 transistors to be stuck-on and the rest 3 input patterns will require only one transistor to be stuck-on to produce a faulty ‘0’. Assuming all input patterns are equally probable, P_{Mf0} for this 3-input logic function would be,

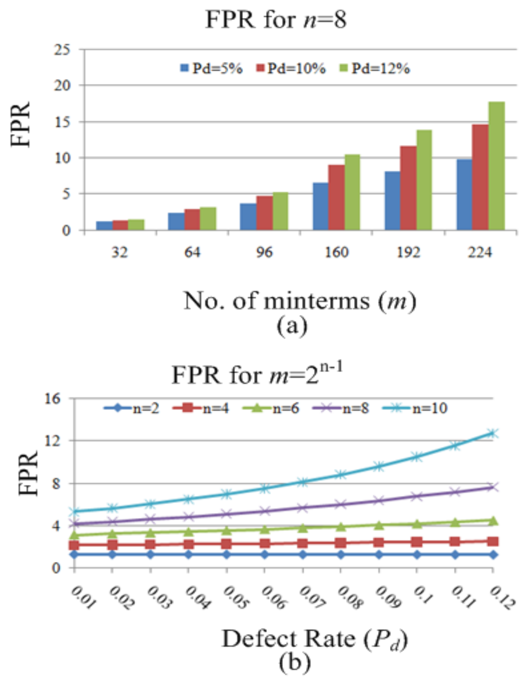

Fig 3. Fault Probability Ratio as a function of (a) number of minterms for n=8 at defect rates of 5%, 10% and 12%, (b) defect rate for a range of fan-in (n) when m=2^{n-1}

$$P_{Mf0} = \frac{1}{7}\,(P_d^{\,3}+3P_d^{\,2}+3P_d) \tag{7}$$

Using the binomial expansion, an expression for P_{Mf0} for an n-input logic function can be obtained:

$$P_{Mf0} = \frac{1}{2^n-1}[(1+P_d)^n-1] \tag{8}$$

By substituting (8) in (4) and (6), probabilities of faulty ‘1’s and faulty ‘0’s at the output of a NASIC tile with no structural redundancy can be found. The Fault Probability Ratios (*FPR*) is the ratio between the probability of a faulty ‘1’ to the probability of a faulty ‘0’, i.e., FPR = $P_{1/0}/P_{0/1}$. Thus, the FPR is a function of the fault probability P_d, the number of inputs of the logic function, n, and the number of minterms, m.

F. Analysis of the Fault Probability Ratio (FPR)

Fig. 3(a) shows the FPR of a NASIC tile with a fan-in of n=8 for an increasing number of minterms (m) at three different defect rates. It is observed that the FPR rises with increasing number of minterms. This is expected since an increase in the number of transistors in the 2nd stage NAND gate makes faulty ‘0’s less likely because any transistor in the stack being switched off is enough to keep the output from evaluating to ‘0’.

Furthermore, for a certain NASIC tile, the FPR is greater for a higher value of the defect rate, P_d. For instance, the FPR is 8.1 for a tile with n=8 and m=160 at a defect rate of 5% and 18.22 for a defect rate of 12%, suggesting that the reliability improvement is greater if biased voters are used where the defect rates are high. This can be better observed from Fig. 3(b) that shows the FPR for a range of defect rates up to 12%.

Fig. 3(b) shows that although the FPR rises with the defect rate, the rate of increase is more marked for circuits with higher fan-in (n). Moreover, for a particular defect rate, the FPR is greater for circuits with higher fan-in. For instance, at a defect rate of 12%, a tile with a fan-in of 6 had an FPR of 4.5 whereas a tile with a fan-in of 10 had an FPR of 12.7. This implies that using biased voters will provide increased yield for circuits with a high number of inputs (fan-in) and minterms at the high defect rates that are characteristic of nanoscale fabrics.

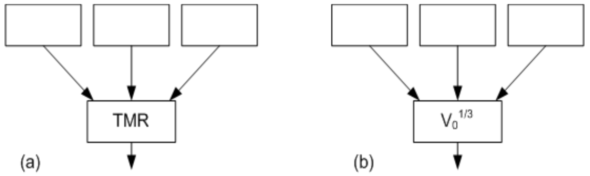

Fig 4. Voting configurations: (a) TMR majority voting, (b) V$_0^{1/3}$ biased voting

III. Biased Voting in NASICs

In this section, we evaluate the expected yield from a majority voting scheme, the TMR [18], and its biased voting counterpart applied in the NASICs fabric.

Fig. 4 shows the two voting configurations. The key difference between the two is the voting decision: in the first case (TMR) a majority voter is used to vote on three structurally non-redundant input modules whereas in the second a voter biased towards a single logic ‘0’ is used. The notation used for this biased voter is V$_0^{1/3}$ (voter biased towards logic ‘0’ that will produce logic ‘0’ even if only 1 out of the 3 inputs is ‘0’). A biased voter has a lower (higher) tolerance for the less (more) prevalent fault type. For instance, the V$_0^{1/3}$ biased voter will not be able to tolerate any faulty ‘0’s but can tolerate up to two faulty ‘1’s.

Our objective is to find out whether the use of a biased voter instead of a conventional majority voter can enhance the expected yield of the voting configuration. Yield is the probability that the circuit will produce a correct output in the presence of manufacturing defects. The two voting configurations (shown in Fig. 4) use the same three structurally non-redundant modules. Thus, the difference in the probability of producing a correct output by the two circuits will be due to the different voting styles. Such an accurate comparison between the two voting schemes and their corresponding voter designs can be achieved by using the Signal Reliability metric [41][42].

In the analytical model presented in this section, faults in voters are not considered. However, voters were allowed to fail in the simulation results presented in Section VI.

In addition to the notations defined in Section II, we use the following when deriving the probabilities of producing correct ‘0’ and ‘1’ outputs by the two voting configurations:

- V^C (V^A) Correct (Actual) output from the voter if it were to take inputs from defect-free input modules

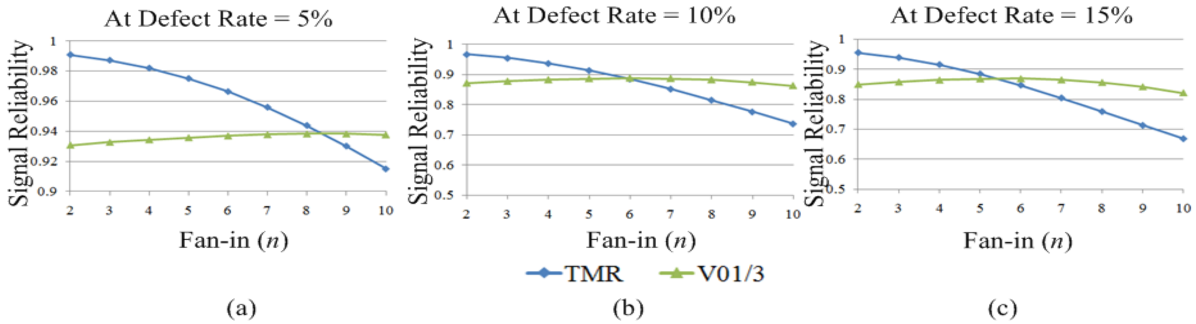

Fig 5. Signal Reliability comparison for biased voting and majority voting schemes at: (a) 5%, (b) 10%, and (c) 15% defect rates.

- *SR* Signal Reliability: Probability of a correct output from voter taking inputs from defective modules

A correct ‘0’ (‘1’) will be produced by a TMR if all three inputs to the voter from the modules are correct ‘0’s (‘1’s) or if any two of the inputs are correct ‘0’s (‘1’s). The probabilities that a TMR generates correct ‘0’s and ‘1’s are

$$P(V^A=0|V^C=0)_{TMR} = (1-P_{1/0})^3+3P_{1/0}(1-P_{1/0})^2 \\ = (P_{0/0})^3+3P_{1/0}(P_{0/0})^2 \qquad (9)$$

$$P(V^A=1|V^C=1)_{TMR} = (1-P_{0/1})^3+ 3P_{0/1}(1-P_{0/1})^2 \\ =(P_{1/1})^3+3P_{0/1}(P_{1/1})^2 \qquad (10)$$

Similarly, probability expressions for the $V_0^{1/3}$ biased voter can be derived,

$$P(V^A=0|V^C=0)_{V01/3}=(1-P_{1/0})^3 +3P_{1/0}(1-P_{1/0})^2 + 3P_{1/0}{}^2(1-P_{1/0}) \\ =(P_{0/0})^3+3P_{1/0}(P_{0/0})^2+3P_{1/0}{}^2P_{0/0} \qquad (11)$$

$$P(V^A=1|V^C=1)_{V01/3} = (1-P_{0/1})^3 = P_{1/1}{}^3 \qquad (12)$$

The signal reliability of a circuit can be expressed as:

$$SR = [P(V^A=0|V^C=0)]\ P(V^C=0) + [P(V^A=1|V^C=1)]\ P(V^C=1) \qquad (13)$$

The signal reliability expressions for the TMR and $V_0^{1/3}$ can be obtained by substituting equations (5) and (6) from the previous section into (9), (10), (11), (12) and subsequently substituting into (13). In a similar manner, the signal reliability for other majority and biased voting configurations can be found.

Fig. 5 shows the reliabilities for n-input NASIC circuits for the two voting styles with $P(V^C=0)= P(V^C=1)$=0.5. As can be seen, for a particular defect rate, the biased voting schemes give a higher reliability as the circuit fan-in increases due to the increasing FPR discussed in the previous section. The $V_0^{1/3}$ biased scheme is shown to have up to 27% greater reliability compared to the TMR scheme. This implies that biased voters could be employed at key architectural points in the design, specifically at the outputs of high fan-in stages, for greater yield, while carefully managing yield-area tradeoffs.

IV. FastTrack Schemes

Even though comparatively higher yields are achieved using biased voting schemes as opposed to majority voting schemes, these yields may still be very low at high defect rates since many, if not all input modules, could be faulty. Therefore, incorporating structural redundancy within the modules and the voter itself is necessary.

However, structural redundancy increases circuit fan-in and fan-out as illustrated by Fig. 6 (a) and (b). Fig. 6 (a) shows a single non-redundant NASIC tile and Fig. 6 (b) shows its 2-way redundant implementation. In the 2-way redundant tile, every nanowire is duplicated, each containing twice as many xnwFETs. The redundant signals are merged within the logic plane itself. In Fig. 6 (a) and (b), $\overline{a_0}$ and $a_0^{'}$ is the complement and the redundant signal of a_0, respectively.

Thus, although structural redundancy may improve the yield, it leads to slower circuits due to the increased fan-in and fan-out. Simulations have verified this degradation in performance; results are shown in Section VI. Detailed description of structural redundancy can be found in [22].

FastTrack schemes are able to achieve both improved yield per unit area and lesser degradation in performance. In FastTrack, biased voters vote on output signals of input modules that have *different levels of structural redundancy* (but implement the same logic function) allowing data propagation without all inputs to the voter being fully evaluated.

In NASICs, outputs will initially have value ‘1’ due to the precharge phase of its dynamic NAND logic style [30], which is one of the logic styles used in NASICs. During the evaluate phase, each output will either discharge to ‘0’ or stay at logic ‘1’ depending on the state of the xnwFETs on that nanowire. Clearly, this suggests logic ‘1’ outputs being at least twice as fast as logic ‘0’s since the logic value is already present at the beginning of the evaluate phases. Thus, the performance of a design is dominated by the evaluation to ‘0’ of high fan-in stages. Consequently, voters biased towards a smaller number of ‘0’s and voting on input modules with varying levels of structural redundancy, will be able to operate at higher frequencies compared to majority voters due to the faster input signals from modules with lower structural redundancy. This advantage in performance over conventional voting schemes is

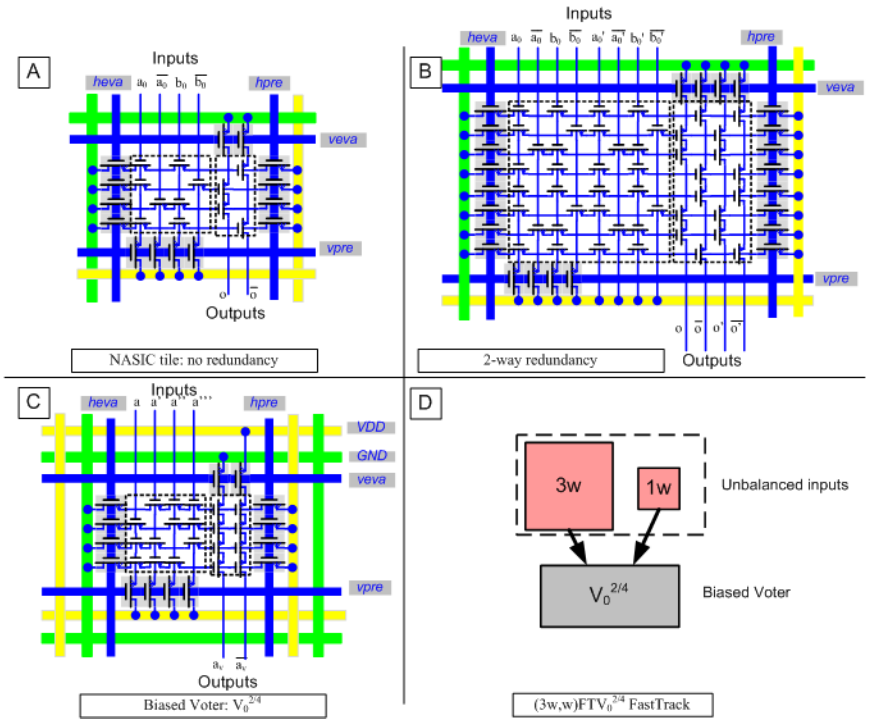

Fig 6. NASIC fault tolerance schemes: (a) NASIC tile with no redundancy incorporated. (b) 2-way redundant implementation of the NASIC tile in (a). (c) Biased Voter: $V_0^{2/4}$ (d) Block diagram showing FastTrack scheme $(3w,w)FTV_0^{2/4}$.

in addition to the yield benefits that were shown for biased voting in Section III.

Although input signals from low-redundancy modules are faster, they are more likely to be faulty. However, if faulty ‘1’s (which are the more prevalent fault type in NASICs) are obtained on low-redundancy copies, the system still functions at the speed of the high redundancy copy maintaining output integrity. The fast ‘1’s can always be obtained from high redundancy modules since a result is available after precharge, less than half the time otherwise it would take for a logic ‘0’.

The combination of unbalanced input blocks and biasing define a variety of new techniques: *the input configuration, redundancy levels for each input, voting type, and biasing applied are the configuration knobs*.

Fig. 6 (c) shows a NASIC implementation of a biased voter. The voter is biased towards two ‘0’s out of the four inputs, which is shown with $V_0^{2/4}$. Fig. 6 (d) shows a FastTrack scheme using the biased voter and non-uniform structural redundancy across the two input modules, one being 3-way redundant and another having no structural redundancy. This FastTrack technique is denoted as (3w, w) $FTV_0^{2/4}$. This notation can be extended to voting schemes where the input blocks have unbalanced redundancy or biasing in general. For example, if we have conventional modular redundancy where all the three input blocks are 2-way redundant and the voter is biased towards 3 inputs needing to be ‘0’ to produce a ‘0’, it can be represented as (2w, 2w, 2w)$V_0^{3/6}$. The ‘FT’ has been dropped when inputs are not unbalanced. If the voter is not biased, there is no need to have subscript/superscript.

V. Simulation Methodology and Circuit

In order to evaluate the effectiveness of the FastTrack schemes in comparison to conventional majority voting schemes, simulations were performed on the Wire Streaming Processor built on the NASIC fabric.

A. Overview of the WISP-0 Processor

WISP-0 is a streaming nanoprocessor design that is implemented on 5 NASIC nanotiles: a program counter, ROM, instruction decoder, register file, and ALU. Communication takes place between adjacent tiles on nanowires. Each nanotile is surrounded by microwires that carry ground, power supply voltage, and some control signals needed for dynamic data streaming. A block diagram of the WISP-0 nanoprocessor

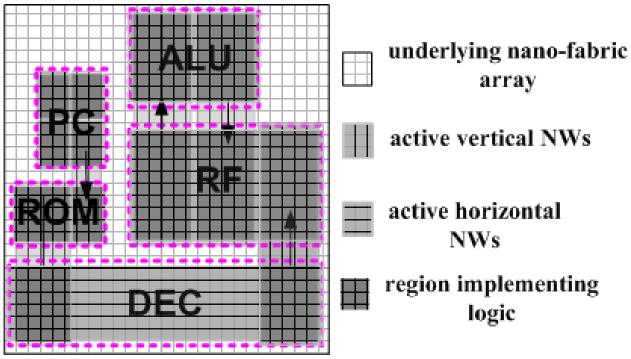

Fig 7. Floor-plan of WISP-0

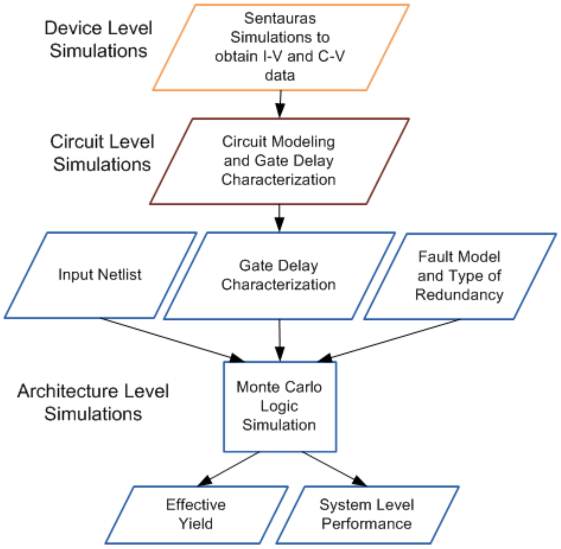
Device Level Simulations
Sentauras Simulations to obtain I-V and C-V data
Circuit Level Simulations
Circuit Modeling and Gate Delay Characterization
Input Netlist
Gate Delay Characterization
Fault Model and Type of Redundancy
Architecture Level Simulations
Monte Carlo Logic Simulation
Effective Yield
System Level Performance

Fig 8. Flowchart for cross-layer simulation methodology

pipeline is shown in Fig. 7. Details of the WISP-0 can be found in [29][30][35].

B. Simulation Framework

The generic device-circuit exploration methodology for nanofabrics detailed in [32] was used. Briefly, xnwFETs are fully characterized in 3-D physics (including materials and geometry) using Synopsys Sentaurus [43]. These physical models were calibrated based on experimental xnwFET data [32] to account for surface roughness and doping distribution effects at this scale. Mathematical expressions are derived from current-voltage and capacitance-voltage data extracted from Sentaurus using standard curve-fit tools. These are then incorporated into a unified behavioral model of the xnwFET device for use in a circuit simulator (e.g., HSPICE). Extensive delay characterization for NASIC circuits is then carried out and data for delay as a function of circuit fan-in is extracted. This information is then used in a higher-level architectural simulator called FTSIM.

This is a custom designed logic and timing simulator that is used to evaluate the yield and performance impact of redundancy on complete nano-systems and that in essence relies on detailed physical models. FTSIM incorporates the above-mentioned device and defect models, and physical design parameters for nanowire interconnect and is capable of handling both parameter variation and device faults, the latter being the focus of this paper. Inputs to the simulator in our context include: (i) a design netlist, (ii) defect model and rates, and (iii) gate delay characterizations. The simulator uses Monte Carlo methods to generate 1,000 defect maps for the system according to the configuration specified and simulates the final output for each case.

Yield is quantified as the number of trials with correct outputs/ *total trials*. Additionally, for each trial with correct outputs, the maximum frequency at which correct outputs are obtained is determined. The overall methodology for integrated exploration is summarized in the flowchart of Fig. 8. To the best of our knowledge this is the most detailed cross-layer nanofabric simulation methodology used to date.

C. Defect Model

The same defect model that had been described in Section II has been used in the simulations. Stuck-on types of defects are the most common in xnwFETs due to the ion implantation and metallization processes in the manufacturing pathway. The Pre-charge and Evaluate control transistors, however, are unlikely to be stuck-on because they are gated by microwires and so have longer channels [29].

The voters are built on the NASIC fabric just like the rest of the WISP-0 circuit. Thus, in this simulation, voters are subject to the same defect rates as the rest of the circuit. An example of such a voter was shown in Fig. 6 (c). Since voters are critical components of the fault tolerance schemes, they incorporate a level of structural redundancy commensurate with the module that has maximum level of structural redundancy within the same FastTrack technique. For instance, in a (3w, 2w) $FTV_0^{2/5}$ technique, all the voters are 3-way redundant.

D. Metrics

Metrics to determine the impact of built-in fault tolerance include: effective yield, normalized performance and the Normalized Performance × Effective Yield product. The Effective yield is the overall yield per unit area for a given technique. The area increase is relative to a design with no redundancy. The area increase also incorporates the area of voters. Thus, for example if a certain fault-tolerance scheme provides a yield of 40% but requires 4 times the area of a circuit without any fault-tolerance scheme, its effective yield would be 0.1. The Normalized performance is defined as the mean operating frequency achieved for a given built-in fault tolerance technique normalized against the mean operating frequency for the slowest technique amongst those considered. Normalized performance × Effective yield reflects the tradeoff between performance and yield.

VI. Results and Analysis

The traditional built-in fault tolerance techniques used in this paper include structural redundancy (2-way (2w), 3-way (3w) or 4-way (4w), etc) along with majority voting, e.g., (2w, 2w, 2w)6MR. In all of the results presented in this section, the WISP-0 processor was the circuit simulated using the simulation framework and defect model presented in section V. For instance in a (2w, 2w, 2w)6MR scheme, each tile of the WISP-0 is 2-way redundant with a voter voting on three identical 2-way redundant tiles.

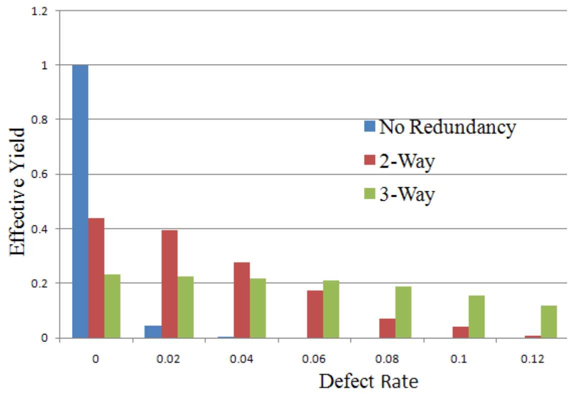

Fig 9. Effective Yield for the WISP-0 processor with 2-way, 3-way, and without using redundancy

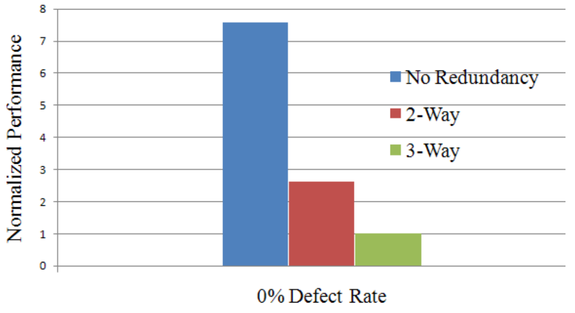

Fig 10. Impact of structural redundancy on the performance of the WISP-0 nanoprocessor with 2-way and 3-way redundancy, and without redundancy

A. *Performance Impact of Structural Redundancy*

Fig. 9 and 10 show the effective yield and the normalized performance (normalized to the slowest i.e., 3-way here), respectively, of the WISP-0 processor with and without structural redundancy. From Fig. 9 we can note that while 2-way redundancy improves the effective yield by 707% at 4% defect rates, 3-way redundancy is required for higher defect rates -- 6% or higher in this design. However, redundancy carries a significant performance penalty. From Fig. 10 we can see that 2-way operates 3X slower and 3-way at 7.5X slower versus the no-redundancy version owing to the increased fan-in/fan-out. Even though 3-way redundancy has significantly better effective yield for higher defect rates, the performance penalty may be unacceptable. These results clearly indicate that while it may be possible to increase effective yield further with higher levels of redundancy, performance will dramatically deteriorate.

B. *Effective Yield – FastTrack Schemes*

Fig. 11 shows the effective yield for the FastTrack and traditional techniques considered. As shown, (w, w, w)TMR performs poorly and has very low effective yield. This is because in the absence of structural redundancy, the voters have insufficient outputs to vote upon in order to mask faulty '0's. Hence the voter propagates the faults from the non-redundant versions and there are no other fall-back mechanisms to mask them. Therefore, clearly, input redundancy is required in input modules.

The Effective yield of (2w, 2w, w)FTV$_0^{2/5}$ is 18% higher at 2% defect rate compared to (2w, 2w, 2w)6MR because of the 20% reduced area overhead (due to the input module with no-redundancy). At 4% or higher defect rates, the effective yield of (2w, 2w, 2w)6MR is 3.4% higher than (2w, 2w, w)FTV$_0^{2/5}$; this is because in the latter case, the voter is dependent on 2 input signals to be '0' to output a '0'vs two 4 input signals to be '0' in the former case. (3w, 3w, 3w)9MR has 33% less effective yield than (2w, 2w, 2w)6MR at the 2% defect rate due to its area penalty.

FastTrack schemes impose less area overhead penalty and if applied carefully, can show better effective yield. Consider for example (3w, 2w)FTV$_0^{2/5}$: the scheme has 18% lesser area compared to (2w, 2w, 2w)6MR. Also, at 4% defect rate, where the 2-way module in FastTrack scheme (2w, 2w, w)FTV$_0^{2/5}$ produces faulty '0's, (3w, 2w)FTV$_0^{2/5}$ still operates correctly because of the 3-way redundant input module. It has 41% higher effective yield than (2w, 2w, 2w)6MR.

Beyond 8% defect rate, it can be seen that the effective yield

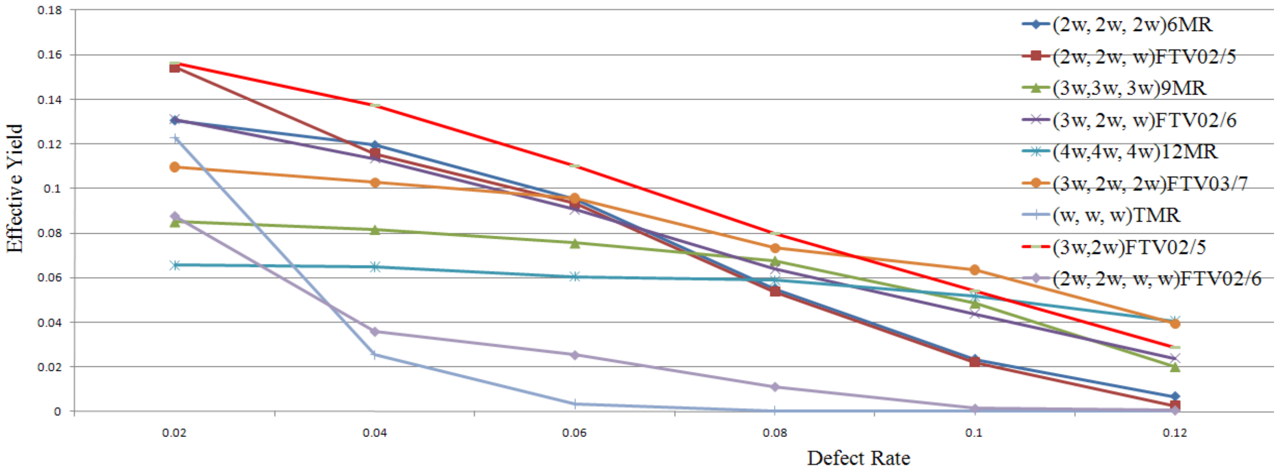

Fig 11. Effective yield vs. defect rate for FastTrack and conventional built-in fault tolerance schemes

of (3w, 2w, 2w)FTV$_0^{3/7}$ is higher than (3w, 2w)FTV$_0^{2/5}$. The reason behind this is that in (3w, 2w, 2w)FTV$_0^{3/7}$ the voting decision is based on more input signals and faulty '0's are thus much less likely (in two 2-way and one 3-way redundancy modules compared to one 2-way and one 3-way in (3w, 2w)FTV$_0^{2/5}$). Overall, the FastTrack scheme performing best at the 10% defect rate is (3w, 2w, 2w)FTV$_0^{3/7}$. This scheme is 23% better in effective yield at this defect rate than (4w, 4w, 4w)12MR, the conventional technique performing best in our experiments at this rate.

C. *Normalized Performance – FastTrack Schemes*

Fig. 12 shows the normalized performance for the techniques evaluated at no defect rate – shown to better highlight the degradation in performance due to redundancy. Techniques are normalized to the slowest scheme, (4w, 4w, 4w)12MR in this case. As can be interpreted from the graph, the (w, w, w)TMR is the fastest scheme as expected because it uses no-redundancy input modules. The FastTrack scheme (2w, 2w, w)FTV$_0^{2/5}$ has normalized speed equivalent to the conventional fault tolerance technique (2w, 2w, 2w)6MR because at zero defect rate, (2w, 2w, w)FTV$_0^{2/5}$ is dependent on the 2-way redundant input module to output one of the two correct ‘0’s. Hence this has the same performance penalty as (2w, 2w, 2w)6MR, i.e. 2.5X slower than the no-redundancy.

In contrast, the other conventional built-in fault tolerance techniques considered, i.e., (3w, 3w, 3w)9MR and (4w, 4w, 4w)12MR show a considerable performance degradation, 8X and 28X slowdown respectively; this is because their voter inputs are dependent on high redundancy input modules and higher number of input signals for correct decision making. However, FastTrack schemes considered i.e., (3w, 2w, w)FTV$_0^{2/6}$, (3w, 2w, 2w)FTV$_0^{3/7}$ and (3w, 2w)FTV$_0^{2/5}$ show lesser performance degradation: they are only 3.76X, 3.5X and 3.8X slower than no-redundancy respectively. This is because, the performance in FastTrack techniques is expected to be determined by the evaluation of correct '0's from the lesser redundant input modules and the time it takes to precharge the higher redundant input modules (in these examples, 3-way). This impacts performance to a lesser extent compared to the conventional techniques depending on inputs only from the symmetric but higher redundant blocks.

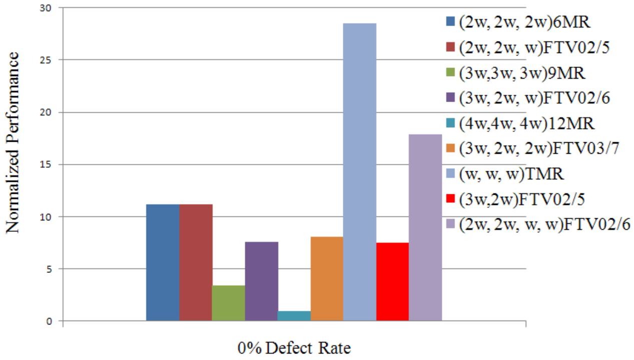

Fig 12. Normalized performance for built-in fault tolerance techniques evaluated at 0% defect rate (the higher the better)

FastTrack schemes can be added with much lower performance penalty compared to conventional ones because the voters in their design are dependent on less redundant input modules. In contrast, when the defect rate increases the speed of conventional techniques remain the same as at the zero defect rate because there is no variation of redundancy in the input modules used in voting decisions.

As can be seen in Fig. 13 the conventional schemes with no redundancy in their voting blocks do not achieve a non-zero yield beyond certain defect rates and therefore have no performance bar associated. The other remaining conventional schemes start losing in performance vs. FastTrack as the defect rates increase despite the speed of FastTrack schemes gradually decreasing; except for (2w, 2w, w)FTV$_0^{2/5}$ (because the voters are always dependent on inputs from 2-way redundant blocks). This is because they have to increasingly rely on outputs from the copies with higher levels of defect tolerance/redundancy. For instance, for (3w, 2w)FTV$_0^{2/5}$, more circuits must depend on the outputs from the 3w copy rather than using solely the outputs from the 2w copy. This results in the mean frequency gradually decreasing. The distribution is not bimodal, however: e.g., this circuit would run at a slower speed than one, which can use solely outputs from the 2w copy, but sometimes faster than one which uses solely outputs from 3w copies. This would be especially visible in FastTrack schemes that have multiple high redundancy copies wherein the faster high redundancy

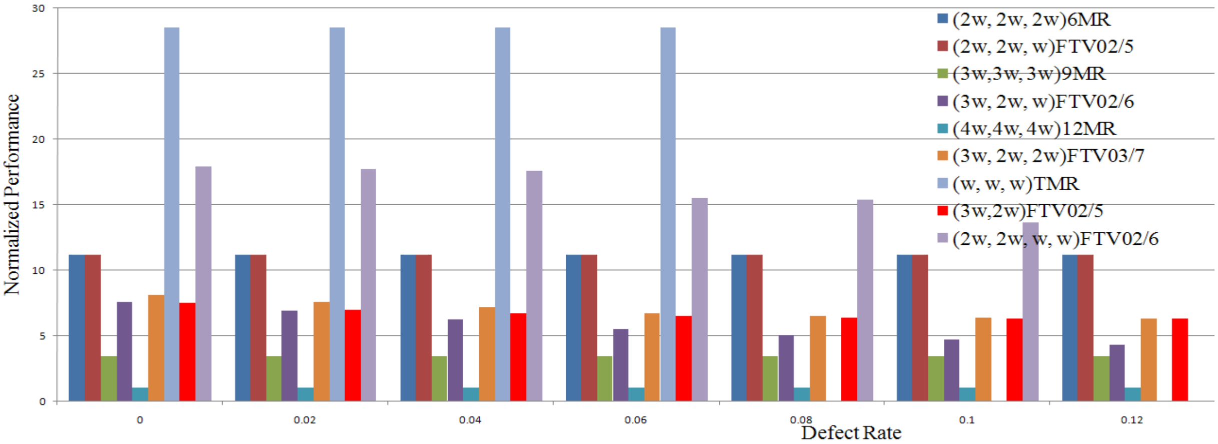

Fig 13. Normalized performance vs. defect rate for FastTrack built-in fault tolerance schemes

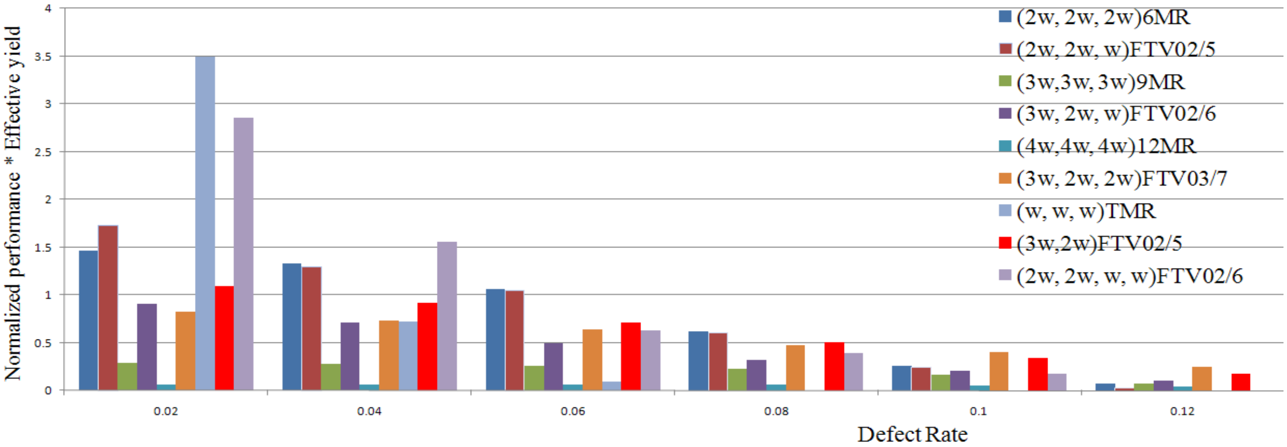

Fig 14. Normalized performance × effective yield products vs. defect rate for FastTrack schemes

copy can vote with lower redundancy ones. This would especially be useful when considering parameter variation (beyond the scope of this paper).

It is interesting to note that FastTrack schemes such as (3w, 2w)FTV$_0^{2/5}$ and (3w, 2w, 2w)FTV$_0^{3/7}$ operate with 77% and 79%, respectively, lesser performance penalty at 10% defect rate compared to conventional fault tolerance technique (3w, 3w, 3w)9MR, while even achieving much better effective yield as can be seen from Fig. 12. The only conventional scheme that does well in performance is (2w, 2w, 2w)6MR, has same performance at 10% as the (2w, 2w, w)FTV$_0^{2/5}$, but has worse effective yield. To better highlight the tradeoff between performance and effective yield we next evaluate their product term in the following section.

D. Normalized Performance × Effective Yield (PEY)-FastTrack Schemes

PEY product results for the various techniques are shown in Fig. 14. At 2% defect rate, (w, w, w)TMR exhibits the highest PEY owing to its large performance advantage (28X faster than the slowest (4w, 4w, 4w)TMR scheme). However, its PEY product falls off rapidly with increasing defect rates owing to high deterioration in yield.

The (2w, 2w, 2w)6MR and (2w, 2w, w)FTV$_0^{2/5}$ have very good PEY products for lower defect rates (up to 6%), since these schemes have good effective yield as well as good normalized performance in this range as discussed previously. However, at higher defect rates the benefits from these schemes drop off due to the reduced effective yield.

The (3w, 2w)FTV$_0^{2/5}$ has the best PEY for up to 8% defect rate because of both its high effective yield owing to its reduced area vs. schemes with 3w blocks alone, as well as, its high normalized performance. At higher defect rates than 10% the PEY product of (3w, 2w, 2w)FTV$_0^{3/7}$ is the best. This scheme is consistently good across all defect rates; it is also the clear winner scheme for defect rates higher than 8%.

The PEY product of the FastTrack scheme performing best at the defect rate of 12%, i.e. (3w, 2w, 2w)FTV$_0^{3/7}$, is 268% higher than the best performing conventional technique, i.e.(3w, 3w, 3w)9MR.

VII. CONCLUSIONS

A step-by-step exploration of redundancy and built-in voting styles at nanoscale for fault masking in nanoscale fabrics is explored. First, analytical fault models of the NASIC fabric demonstrating the unequal fault probabilities of faulty ‘0’s and ‘1’s are introduced. This serves as the theoretical foundation for demonstrating that biased voting not only increases the yield in nano fabrics with unequal fault probabilities, but also lowers the degradation in performance when it is combined with non-uniform structural redundancy across input modules. An ensemble of such new redundancy schemes called FastTrack are proposed to improve performance during fault masking versus conventional redundancy was developed. Then, these schemes were evaluated with a comprehensive cross-layer simulation framework with detailed physical device and circuit models. Results were shown for effective yield and their performance and were compared with conventional built-in fault tolerance schemes for a processor design. FastTrack schemes cause 79% less performance penalty at 10% defect rate compared to conventional fault tolerance techniques. The scheme (3w, 2w)FTV$_0^{2/5}$ shows the highest effective yield for up to 8% defects for the WISP-0 processor design. At higher defect rates, (3w, 2w, 2w)FTV$_0^{3/7}$ achieves the highest effective yield. The normalized performance - effective yield product for this scheme was found to be 2.5X better than any other traditional redundancy scheme considered, even at 12% defect rate. Given a variety of schemes, FastTrack schemes not only provide higher performance but also outperform on effective yield due to their lower area penalty. While the approach is shown for NASICs, where detailed physical models are available, the overall methodology, analytical models and the FastTrack redundancy schemes are applicable in other nanoscale fabrics with an asymmetric occurrence of faulty ‘1’s vs. faulty ‘0’s.

REFERENCES

[1] D. Whang, S. Jin, and C. M. Lieber, "Nanolithography Using Hierarchically Assembled Nanowire Masks," *Nano Letters*, vol. 3, pp. 951-954, Jun. 2003.

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

[3] X. Xiong, L. Jaberansari, M. G. Hahm, A. Busnaina, and Y. J. Jung, "Building Highly Organized Single-Walled-Carbon-Nanotube Networks Using Template-Guided Fluidic Assembly," *Small*, vol. 3, pp. 2006- 2010, Dec. 2007.

[4] Y. Shan and S. J. Fonash, "Self-Assembly Silicon Nanowires for Device Applications Using the Nanochannel-Guided "Grow-in-Place" Approach," *ACS Nano*, vol. 2, pp. 429-434, Mar. 2008.

[5] Y. Yellambalase, M. Choi, Y. Kim, "Inherited Redundancy and Configurability Utilization for Repairing Nanowire Crossbars with Clustered Defects," in *Proc. IEEE 25th Int. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT),* Arlington, 2006, pp. 98-106.

[6] Y. Dotan, N. Levison, R. Avidan, D. K. Lilja, "History Index of Correct Computation for Fault-Tolerant Nano-Computing," *IEEE Trans. Very Large Scale Integration (VLSI) Systems,* pp. 943-952, July 2009.

[7] S. C. Goldstein, M. Budiu, "NanoFabrics: spatial computing using molecular electronics," *28th Annual International Symposium on Computer Architecture,* Goteburg, 2001, pp. 178-189.

[8] A. KleinOsowski et al, "Exploring Fine-Grained Fault Tolerance for Nanotechnology Devices With the Recursive NanoBox Processor Grid," *IEEE Transactions on Nanotechnology*, pp. 575-586, Sep. 2006.

[9] M. Tehranipoor, "Defect tolerance for molecular electronics-based nanofabrics using built-in self-test procedure," in *Proc. IEEE 20th Int. Symp. Defect and Fault Tolerance (DFT),* Monterey, 2005, pp. 305-313.

[10] J. G. Brown, R. D. Blanton, "CAEN-BIST: testing the nanofabric," *International Test Conference (ITC)*, Charlotte, 2004, pp. 462-471.

[11] W. Rao, A. Orailoglu, R. Karri, "Logic Mapping in Crossbar-Based Nanoarchitectures," *IEEE Design & Test of Computers,* pp 68-77, Feb 2009

[12] T. Hogg, G. S. Snider, "Defect-tolerant adder circuits with nanoscale crossbars," *IEEE Transactions on Nanotechnology,* pp 97-100, Mar. 2006

[13] A. DeHon, H. Naeimi, "Seven strategies for tolerating highly defective fabrication," *IEEE Design & Test of Computers,* pp 306-315, Aug 2005

[14] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata Studies, C. E. Shannon and J. McCarthy*. Princeton, NJ: Princeton Univ. Press, 1956, pp. 43–98.

[15] J. Sklaroff, "Redundancy management technique for space shuttle computers," *IBM J. Res. Dev.*, vol. 20, no. 1, pp. 20–28, Jan. 1976.

[16] V. Julien, B. Alberto and G. Patrick. "Using TMR architectures for yield improvement," in *Proc. IEEE Int. Symp. on Defect and Fault Tolerance in VLSI systems, (DFTVS),* Boston, 2008. pp. 7–15.

[17] M. Hunger and S. Hellebrand, "The impact of manufacturing defects on the fault tolerance of TMR-systems," in *Proc. IEEE 25th Int. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT)*, Kyoto, 2010, pp 101-108.

[18] I. Koren and C. M. Krishna, *Fault-Tolerant Systems*, San Fransisco, CA: Morgan Kaufmann Publishers, 2007.

[19] T. Ban, L. A. de Barros Naviner, "A simple fault-tolerant digital voter circuit in TMR nanoarchitectures," in *Proc. of 8th IEEE International NEWCAS Conference*, Montreal, 2010, pp 269-272.

[20] K. Granhaug and S. Aunet, "Improving yield and defect tolerance in subthreshold CMOS through output-wired redundancy," *J. Electron. Testing*, vol. 24, no. 1–3, pp. 157–163, Jun. 2008.

[21] A. Namazi, M. Nourani and M. Saquib, "A Fault-Tolerant Interconnect Mechanism for NMR nanoarchitectures," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, pp 1433-1446, Oct. 2010.

[22] 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, no. 11, pp. 2422 –2437, Nov. 2007.

[23] S. Roy and V. Beiu, "Majority multiplexing-economical redundant fault-tolerant designs for nanoarchitectures," *IEEE Transactions of Nanotechnology*, vol. 4, no. 4, pp. 441–451, Jul. 2005.

[24] J. Han, J. Gao, P. Jonker, Y. Qi, and J. A. B. Fortes, "Toward hardware-redundant, fault-tolerant logic for nanoelectronics," *IEEE Design & Test Computers*, vol. 22, no. 4, pp. 328–339, Jul./Aug. 2005.

[25] J. Han and P. Jonker, "A system architecture solution for unreliable nanoelectronic devices," *IEEE Transactions on Nanotechnology,* vol. 1, no. 4, pp. 201-208, Dec. 2002.

[26] C. A. Moritz, P. Narayanan, and C. O. Chui, "Nanoscale application specific integrated circuits," in *Nanoelectronic Circuit Design*, Editors: N. K. Jha and D. Chen, New York: Springer, 2011, pp. 215–275.

[27] P. Vijayakumar, P. Narayanan, I. Koren, C.M. Krishna, C.A. Moritz, "Incorporating Heterogeneous Redundancy in a Nanoprocessor for Improved Yield and Performance," in *Proc. of the 25th IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems*, Kyoto, 2010, pp. 273-279.

[28] T. Wang, P. Narayanan, and C. A. Moritz, "Heterogeneous Two-Level Logic and Its Density and Fault Tolerance Implications in Nanoscale Fabrics," *IEEE Transactions on Nanotechnology*, vol. 8, no. 1, pp. 22-30, Jan. 2009.

[29] P. Narayanan, M. Leuchtenburg, T. Wang, and C. A. Moritz, "CMOS control enabled Single-Type FET NASIC," in *Proc. of the 2008 IEEE Computer Society Annual Symposium on VLSI*, Montpellier, 2008, pp. 191–196.

[30] T. Wang, P. Narayanan, and C. A. Moritz, "Combining 2-level logic families in grid-based nanoscale fabrics," in *Proc. of the 2007 IEEE Int. Symp. on Nanoscale Architectures*, San Jose, 2007, pp. 101-108.

[31] M. M. Khan, P. Narayanan, P. Vijayakumar, I. Koren, C. M. Krishna, C. A. Moritz, "Biased Voting for Improved Yield in Nanoscale Fabrics," in *Proc. of IEEE Int. Symp. on Defect and Fault Tolerance (DFT 2011),* pp. 79-85, Oct 2011.

[32] P. Narayanan, C. A. Moritz, K. W. Park, C. O. Chui, "Validating cascading of crossbar circuits with an integrated device-circuit exploration," in *Proc. of the 2009 IEEE Int. Symp. on Nanoscale Architectures*, San Francisco, 2009, pp. 37-42.

[33] A. B. Greytak, L. J. Lauhon, M. S. Gudiksen, and C. M. Lieber, "Growth and transport properties of complementary germanium nanowire field-effect transistors," *Appl. Phys. Lett.,* vol. 84, pp. 4176- 4178, May 2004.

[34] M. I. Khan, X. Wang, K. N. Bozhilov, and C. S. Ozkan, "Templated fabrication of InSb nanowires for nanoelectronics," *J. Nanomaterials*, vol. 2008, pp. 1-5, Feb. 2008.

[35] T. Wang, M. Bennaser, Y. Guo and C. A. Moritz, "Wire-Streaming Processors on 2-D Nanowire Fabrics", in *Proc. of Nanotech 2005, Nano Science and Technology Institute*, Anaheim, 2005.

[36] P. Narayanan, T. Wang, M. Leuchtenburg, C. A. Moritz, "Image Processing Architecture for Semiconductor Nanowire based Fabrics," in *Proc. of IEEE 8th International Conference on Nanotechnology*, Arlington, 2008, pp. 677-680.

[37] P. Narayanan, J. Kina, P. Panchapakeshan, C. O. Chui and C. A. Moritz, "Integrated device-fabric exploration and noise mitigation in nanoscale fabrics," *IEEE Transactions on Nanotechnology*, accepted for publication.

[38] P. Narayanan, K. W. Park, C. O. Chui, and C. Moritz, "Manufacturing pathway and associated challenges for nanoscale computational systems," in *Proc. of the 9th IEEE Conference on Nanotechnology*, Genoa, 2009, pp. 119 –122.

[39] Y. Cui, X. Duan, J. Hu, and C. M. Lieber, "Doping and electrical transport in silicon nanowires," *The Journal of Physical Chemistry B*, vol. 104, no. 22, pp. 5213–5216, Jun. 2000.

[40] E. R. Hnatek, *Integrated Circuit Quality and Reliability*, New York: Marcel Dekker Inc., 1995.

[41] I. Koren, "Analysis of the Signal Reliability Measure and an Evaluation Procedure," *IEEE Transactions on Computers*, pp. 244-249, Mar. 1979.

[42] D. T. Franco, M. C. Vasconcelos, L. Naviner, J. F. Naviner, "Reliability analysis of logic circuits based on signal probability," in *Proc. of the 2008 IEEE International conference on electronics, circuits and systems*, St. Julien's, 2008, pp. 670-673.

[43] Sentaurus Device User Guide, *Synopsys, Inc.*, 2007.