Toward Hardware-Redundant, Fault-Tolerant Logic for Nanoelectronics

Jie Han and Jianbo Gao
University of Florida

Pieter Jonker
Delft University of Technology

Yan Qi
Johns Hopkins University

José A.B. Fortes
University of Florida

Editor's note:
This article provides an overview of several logic redundancy schemes, including von Neumann's multiplexing logic, *N*-tuple modular redundancy, and interwoven redundant logic. The authors use Markov chain models and bifurcation analysis to compare the degree of redundancy and system reliability in these classical fault-tolerant approaches.
—Mehdi B. Tahoori, Northeastern University

THERE IS RENEWED INTEREST in using hardware redundancy to mask faulty behavior in nanoelectronic components. In this article, we go back to the early ideas of von Neumann and review the key concepts behind *N*-tuple modular redundancy (NMR), hardware multiplexing, and interwoven redundant logic. We discuss several important concepts for redundant nanoelectronic system designs based on recent results. First, we use Markov chain models to describe the error-correcting and stationary characteristics of multiple-stage multiplexing systems. Second, we show how to obtain the fundamental error bounds by using bifurcation analysis based on probabilistic models of unreliable gates. Third, we describe the notion of random interwoven redundancy. Finally, we compare the reliabilities of quadded and random interwoven structures by using a simulation-based approach. We observe that the deeper a circuit's logical depth, the more fault-tolerant the circuit tends to be for a fixed number of faults. For a constant gate failure rate, a circuit's reliability tends to reach a stationary state as its logical depth increases.

Two widely studied fault tolerance techniques that use hardware redundancy to mask faults are NMR and the multiplexed logic approach. The multiplexed logic approach, motivated by the pioneering work of John von Neumann, began as an attempt to build early digital computers out of unreliable components.[1] This approach and subsequent derivatives[2-6] have provided insight on how to design reliable nanoelectronic systems out of components that might fundamentally be less reliable than those of currently available technologies.[7,8] Quantum effects, increased sensitivity to noise, and decreased fabrication tolerances inherent in nanoelectronics will all contribute to reliability losses. Hence, questions arise as to what are the error behaviors of a fault-tolerant system, and what are the maximal error rates beyond which no reliable designs are possible. With these understandings, new questions arise: Are there new fault-tolerant designs that are well-suited for nanoelectronics? If so, what are the characteristics of these designs in contrast with theoretical models?

To address the first set of questions, we use a Markov chain model to analyze a multiplexing system to infer how system reliability depends on individual gate reliabilities. We then use bifurcation analysis of expressions describing logic circuit behavior that account for the probability of gate errors. To address the latter questions, we use interwoven redundant logic with randomized connectivity to cope with, and even leverage, the potential randomness of nanoscale interconnects resulting from either failures or the random characteristics of self-assembly. We then use simulations to experimentally investigate the system behaviors of random

0740-7475/05/$20.00 © 2005 IEEE Copublished by the IEEE CS and the IEEE CASS

interwoven redundancy and quadded structures with multiple logical depths.

Conventional fault-tolerant techniques based on hardware redundancy

To provide a background for our study, we briefly review classical fault-tolerant techniques derived from von Neumann's multiplexing scheme.

Von Neumann's multiplexing technique

In the 1950s, John von Neumann initiated the study of techniques for the design of reliable systems using redundant unreliable components.[1] In his multiplexing structure, von Neumann considered two types of basic logic, namely majority-voting and NAND logic. He duplicated each logic gate N times and replaced each input with a bundle of N lines; thus, each output bundle also had N lines. For NAND logic, the inputs from the first bundle randomly pair with those from the second bundle to form the input pairs of the duplicated NANDs (as illustrated in Figure 1). Instead of requiring all or none of the output bundle's lines to produce correct answers, von Neumann set a certain critical (or threshold) level Δ such that $0 < \Delta < 1/2$. If the number of lines carrying the correct signal was larger than $(1 - \Delta)N$, he interpreted it as a positive state of the bundle; if it was less than ΔN, he considered it a negative state. By using a massive duplication of unreliable components, von Neumann concluded that the construction can be reliable with a high probability if the failure probability of the gates is sufficiently low (for example, lower than approximately 10^{-2}).[1]

In general, von Neumann's construction requires a large amount of redundancy ($N > 10^3$) and a low error rate for individual gates. These features motivated extensive research efforts in later decades to find the complexity of redundancy required to cope with errors. Pippenger offers a review of these efforts.[5] Because CMOS devices became dominant in industry and showed an amazing performance in terms of reliability and scalability, chip designers never used von Neumann's multiplexing technique in practice. However, researchers have implemented many redundancy techniques derived from von Neumann's proposal, such as triple modular redundancy (TMR) and error-correcting codes (ECC), in high-reliability applications and in memory circuits.

N-tuple modular redundancy

As implied in von Neumann's theory, *N*-tuple modular redundancy (NMR) designs—of which TMR is the most-used particular case—have been used as benchmarks for evaluating fault-tolerant approaches and have been implemented in VLSI for high-reliability applications. NMR techniques, generally implemented at the modular rather than gate level, use redundant components to mask fault effects. In TMR, three identical modules perform the same operation, and a voter accepts outputs from all three modules, producing a majority vote at its output. In TMR, however, the reliability of a module imposes a demanding requirement on a module's size—the modules involved in TMR should be modest in size in relation to the error rate of an individual component in the circuit; in other words, a module with many components will present a serious limit on the upper bound of the device error rate that TMR can tolerate.

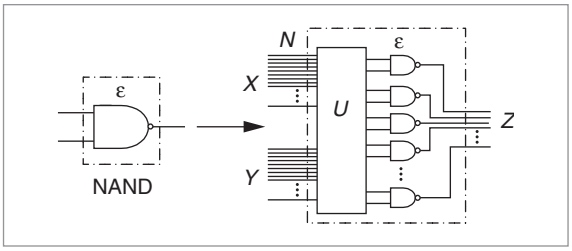


Figure 1. Example NAND multiplexing unit.

A TMR circuit can be further triplicated. The obtained circuit thus has nine copies of the original module and requires two layers of majority gates to collect information at outputs. This process can be repeated if necessary, resulting in a technique called cascaded triple modular redundancy (CTMR). Spagocci and Fountain have shown that using CTMR in a nanochip with many (for example, 10^{11}or 10^{12}) nanoscale devices would require an extremely low device error rate.[6] However, the method might be effective in modest or small circuit modules. Another disadvantage of the CTMR scheme is that it introduces an exponential growth in redundancy as the cascaded layers increase.

Interwoven redundant logic and quadded logic

Pierce generalized von Neumann's and his contemporaries' ideas on fault-tolerant logic to a theory termed *interwoven redundant logic*.[2] This theory interprets the faults it considers as $0 \rightarrow 1$ and $1 \rightarrow 0$ faults. The error correction mechanism in interwoven redundant logic depends on asymmetries in the effects of these two types of binary errors. The effect of a fault depends on the value of the erroneous input and the type of gate. Consider a NAND gate, for instance. If the binary value of one of its inputs is 0 while it should be 1, possibly because of a

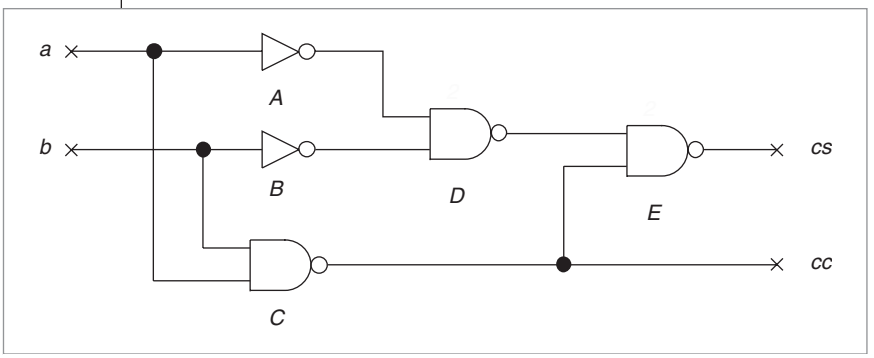


Figure 2. Schematic of a nonredundant complementary half adder implemented with NAND logic.

faulty gate or interconnection, the NAND's output value will remain a 1 regardless of the values of other inputs. If an input value is 1 while it should be 0, the output will not be stuck but will depend on other inputs. Thus, there are two types of faults for a NAND gate. One is critical in the sense that its occurrence on one of the inputs leads to a stuck output; the other is subcritical in the sense that its occurrence alone does not cause an output error. Hence, alternating layers of NAND (or NOR) gates can correct errors by switching them from critical to subcritical.

Quadded logic is an ad hoc configuration of the interwoven redundant logic. It requires four times as many circuits, interconnected in a systematic way, and it corrects errors and performs the desired computation at the same time. Researchers have studied quadded logic for use with AND, OR, and NOT logic,[3] and for use with NOR logic.[4] To illustrate quadded logic, we show the schematic of a complementary half adder (computing the complements of carry and sum, denoted as cc and cs) in Figure 2 and its quadded form in Figure 3, both implemented with NAND gates (including inverters, which we consider a special form of NAND gate).

The quadded implementation in Figure 3 replaces each NAND gate from Figure 2 with a group of four NAND gates, each of which has twice as many inputs as the one it replaces. The four outputs of each group are divided into two sets of two outputs, each providing inputs to two gates in a succeeding stage. The interconnections in a quadded circuit are hence eight times as many as those used in the nonredundant form.

In this pattern of interconnection, any single error introduced in the network is correctable by the network itself, provided that the network is large enough. To show this in Figure 3, assume that output $B1$ in stage B is wrongly in the 0 state when it should be in the 1 state (a critical $1 \rightarrow 0$ error for the NAND gate). Because of this error, outputs $D1$ and $D3$ of stage D must be 1; this can be erroneous, but it would be a subcritical $0 \rightarrow 1$ error. Since outputs $D2$ and $D4$ of stage D are not in error (thus in the correct 0 state), the subcritical errors at outputs $D1$ and $D3$ are masked at stage E, producing the expected (correct) 1 state at all the outputs of stage E. We observed that a subcritical $0 \rightarrow 1$ error is even more promptly corrected in the NAND network. In general, a single critical error in a quadded circuit will be eliminated after passing through two stages, and a single subcritical error will be corrected in the next stage after its occurrence.

The interconnect patterns in a quadded network are important to the network's capability of error correction, yet the rules are simple. The outputs of four gates, numbered 1 to 4 in Figure 3, are divided into two sets. Each set forms a pair of inputs and each pair feeds the two gates with the same numbers as the set in succeeding stages. If the four outputs are divided into two sets of (1,3) and (2,4), for instance, set (1,3) will provide inputs to gates 1 and 3 in the next stage and set (2,4) will provide inputs to gates 2 and 4. There are three possible ways to break four inputs into two sets to form an interconnect pattern: (1,2) and (3,4); (1,3) and (2,4); and (1,4) and (2,3). The rule to arrange these patterns is that the interconnect pattern at the outputs of a stage must differ from the interconnect patterns of any of its input variables.

The error correction property of a quadded NAND network is in fact a result of its logical characteristics. Let us take a look at the outputs of stage B in Figure 3: $B1$, $B2$, $B3$, and $B4$. After passing through two NAND stages, the outputs of stage B can be represented at stage E by the following Boolean function:

$$B1B3 + B2B4$$

All Bs in this function should be the same in the absence of errors, but any single error in the Bs will not affect the function's correct value.

In a quadded circuit, a single error is correctable in at most two logic layers. Errors occurring on the circuit's edge, however, might not be eliminated at outputs (more specifically, a critical error within the last two layers or a subcritical error in the last layer is not correctable at outputs). Therefore, the gates on the edge are critical in the sense that the failure of any critical

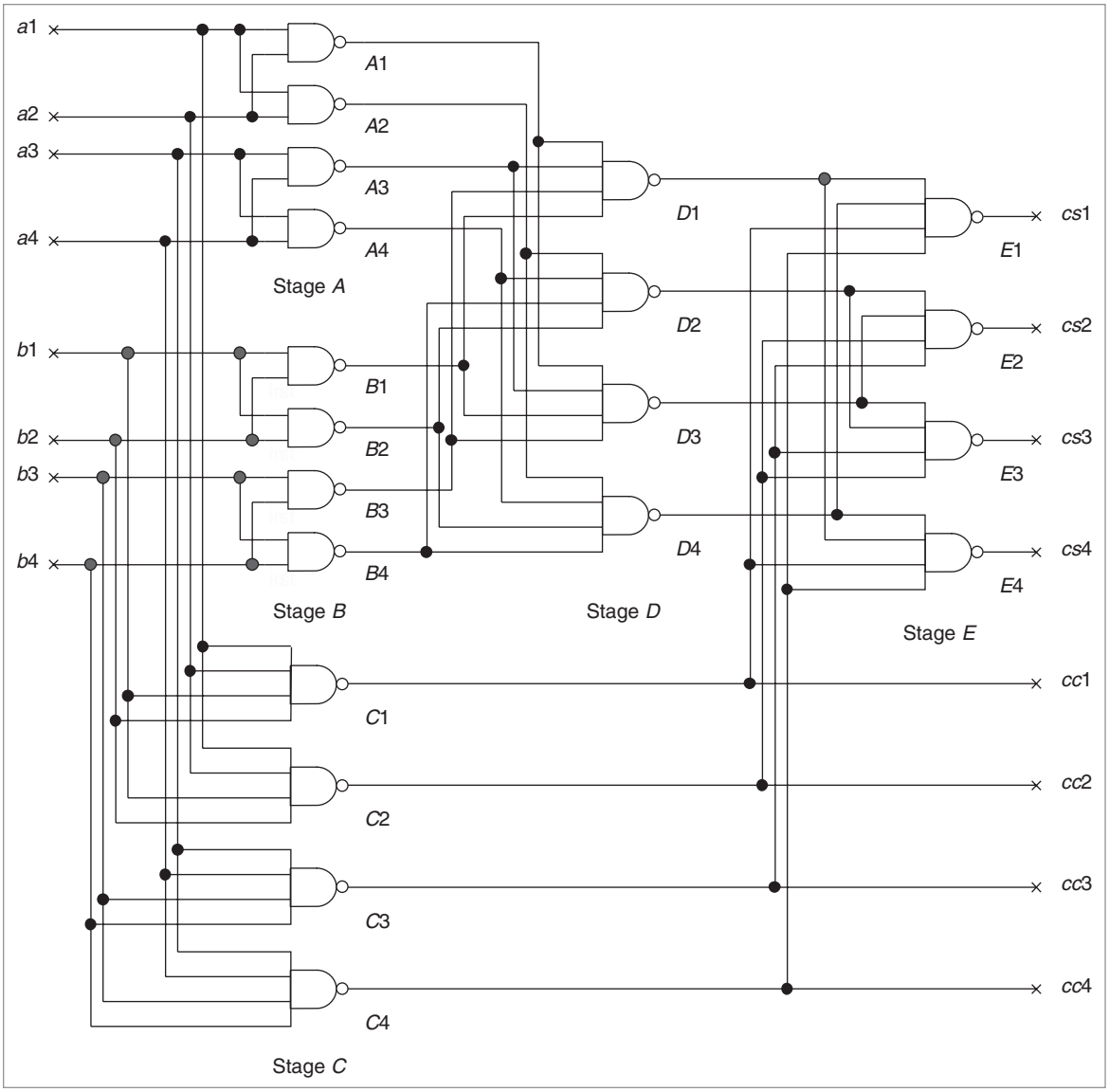


Figure 3. Quadded implementation of the complementary half adder.

gate will cause a high probability of failure for the whole circuit. Because a single error is corrected within a rather short logical path, many multiple errors do not interact. Hence, multiple errors are also correctable in many cases. This is a particular merit of quadded logic.

Markov chain models, error bounds, and random interwoven redundancy

To build reliable systems out of unreliable components, it is critical to consider the interplay of noise in the individual gates, the degree of redundancy, and system reliability. We integrate these aspects to present a coherent picture, using Markov chain models and bifurcation analysis. We also present random interwoven redundancy, which sprouted from the multiplexing technique, as a generalization of *N*-tuple modular redundancy with random interconnections.

Markov chain models of multiplexing systems

As CMOS technology enters the deep-submicron realm and novel electronic devices that target nanoscale applications emerge, reliability has become a crucial issue for nanoelectronics. Nikolić, Sadek, and Forshaw recently proposed von Neumann's multiplexing technique for implementing highly redundant, fault-tolerant nanoelectronic systems.[9] Han and Jonker extended the study of NAND multiplexing to consider a fairly low degree of redundancy and first used Markov

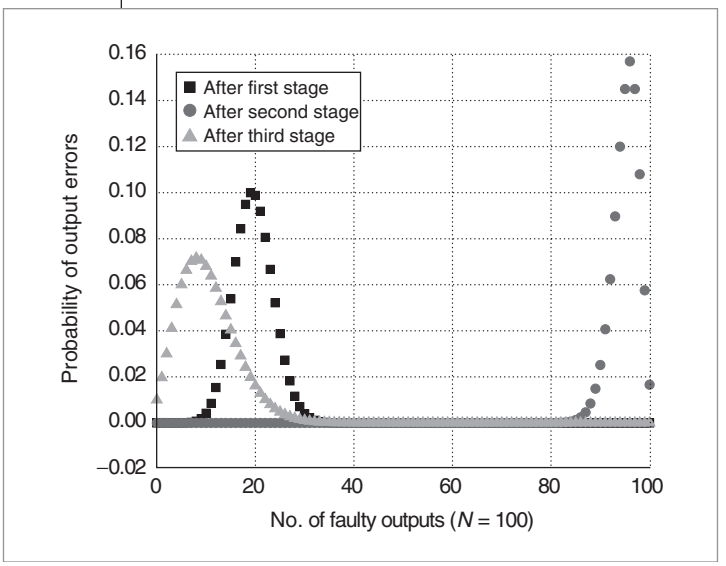


Figure 4. Output error distributions in a three-stage multiplexing system (ε = 0.01).

chain models to study multiple-stage multiplexing systems.[10] In von Neumann's original proposal, he duplicated the multiplexing unit to form a restorative unit, which he used to annul the possible degradation caused by the so-called executive unit. For large bundle size N, von Neumann concluded that the output error rate is a stochastic variable with an approximately normal distribution. For modest N, Han and Jonker used binomial distributions with the Markov chain models to explore the error behaviors in multiplexing systems.[10] Given imperfect inputs (with 90% of the input lines stimulated), we show the output error distributions of a three-stage multiplexing system by using a Markov chain model for $N = 100$ and $\varepsilon = 0.01$, where ε is the gate error rate. As Figure 4 illustrates, the error level becomes amplified after the first stage of multiplexing and then, by the second stage, shifts to the other side of the diagram (performing an imperfect AND function). After the third stage, the output error distribution shifts back and decreases to a lower level. This behavior shows the error-correcting mechanism of the multiplexing structure. The "Probabilistic models of multiplexing systems" sidebar briefly surveys recent work on the modeling of multiplexing systems.

Error bounds for logic gates

Noisy NAND gates are the building blocks of von Neumann's multiplexing scheme. For the basic gate error, the simple von Neumann model assumes that the gate flips the output with a probability of $\varepsilon \leq 1/2$, while the input and output lines function reliably. For a single NAND gate (as in Figure 1), let X, Y, and Z denote the probabilities of the two inputs and output being a 1. The two inputs can be treated as independent in circuits without closed loops or fan-out. Thus, by assuming first that the NAND gate is fault free, the probability Z of the output being a 1 (by at least one of the inputs being a 0) is $1 - XY$. If the gate has a probability ε of making a von Neumann error, the probability of its output being a 1 is[10]

$$Z = (1-\varepsilon)(1-XY) + \varepsilon XY = (1-\varepsilon) + (2\varepsilon - 1)XY \quad (1)$$

We can see that $X = Y$ constitutes the worst-case scenario, so we consider $X = Y$ next.

Let us consider a network of NAND gates and label a sequence of the gates by index i, such that $i = 1$ to n, where the output of gate i becomes an input to gate $i + 1$. If the two inputs of any gate are equally probable to be a 1, Equation 1 reduces to a simple nonlinear map:

$$X_{n+1} = (1-\varepsilon) + (2\varepsilon - 1)X_n^2 \quad (2)$$

In such a map, ε is called a bifurcation (or controlling) parameter. Equation 2 corresponds to a circuit with a binary tree structure where each node is a NAND gate. We can examine the dynamic behavior of the map through bifurcation analysis. Computationally, this involves the iterative execution of Equation 2. For any fixed ε ($0 < \varepsilon < 1/2$), we arbitrarily choose an initial value for X_0 and then iteratively calculate Equation 2. After a sufficiently large number of iterations such that the map's solution has converged to some attractors, we retain, say, the last 100 iterations, and plot those 100 points against each ε. Figure 5 shows this diagram (p. 334). Such a bifurcation analysis of Equation 2 reveals that a period-doubling bifurcation occurs at[11]

$$\varepsilon_* = 0.08856$$

When n is large, the system has a stable fixed-point solution x_0 for any ε when $\varepsilon_* < \varepsilon < 1/2$. When $0 \leq \varepsilon < \varepsilon_*$, x_0 loses stability, and the motion is periodic with a period of 2 (corresponding to the odd and even layers of NAND gates). We label these two periodic points as x_+ and x_- in Figure 5. For a NAND gate to function reliably, two identical inputs of 1 or 0 should output a 0 or 1, respectively. Thus, we see that $0 \leq \varepsilon < \varepsilon_*$ is the parameter interval at which the NAND gate can function. When $\varepsilon > \varepsilon_*$, we interpret output x_0 as neither 1 nor 0, hence, it is what von Neumann called a state of irrelevance.

Probabilistic models of multiplexing systems

Recently, Qi, Gao, and Fortes analyzed Markov-chain-based multiplexing systems[1] using bifurcation analysis, a method widely used for studying state transitions of dynamical systems that have multiple stationary states or oscillatory behavior. The analysis showed that the stationary distribution of a Markov chain characterizes a system's behavior. In a long run, the stationary distribution can degenerate into one peak (unimodal) or two peaks (bimodal), depending on whether the error probability of NAND gates is larger or smaller than a threshold value. Based on these understandings, Qi, Gao, and Fortes used a new scheme involving gate error rate ε to obtain adaptive threshold Δ for output bundles; they showed that the system reliability improves rapidly with increasing redundancy.[1]

However, the numerical analysis employing Markov chain models only gives an approximate evaluation of the output reliability of a multiplexing system. The larger the bundle size N, the better the approximation. When N becomes small, however, neither the binomial nor the normal distribution model used in Markov chains presents a good approximation because of the correlations among input errors. So the analytical approach using the multiplexing theory does not apply to systems with low redundancy ($N < 10$, for example). Norman et al. identified this limitation as an incorrect modeling of the random permutation of inputs using a "random choice with replacement."[2] To address this problem, Norman et al. and Bhaduri and Shukla proposed a CAD method based on probabilistic model checking, and used the idea of folding space into time to mitigate the state space explosion problem in modeling multiplexing systems.[2,3] Sadek, Nikolić, and Forshaw extended the Markov chain approach to consider the stochasticity of gate error probability, performing Monte Carlo simulations to study the error behavior in a multiplexing nanosystem.[4] Roy, Beiu, and Sulieman used an exact analysis using combinatorial arguments to model majority-based multiplexing systems that have small redundancy factors.[5] Han and Jonker equipped a Markov-chain-based multiplexing system with hierarchical reconfigurability for protection against both permanent and transient errors.[6] Bahar, Chen, and Mundy proposed fault-tolerant designs employing Markov random fields that relate thermal energy and entropy,[7] and Bhaduri and Shukla extended the study to consider interconnect noise.[3] Krishnaswamy et al. used an approach based on probabilistic transfer matrices of joint and conditional probabilities of signals to evaluate circuit reliability.[8]

References

1. Y. Qi, J.B. Gao, and J.A.B. Fortes, "Markov Chains and Probabilistic Computation: A General Framework for Multiplexed Nanoelectronic Systems," *IEEE Trans. Nanotechnology*, vol. 4, no. 2, Mar. 2005, pp. 194-205.
2. G. Norman et al., "Evaluating the Reliability of NAND Multiplexing with PRISM," to appear in *IEEE Trans. CAD*.
3. D. Bhaduri and S.K. Shukla, "Evaluating Reliability Trade-Offs for Nano-Architecture," *Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation*, S.K. Shukla and R.I. Bahar, eds., Springer, 2004, pp. 157-221.
4. A.S. Sadek, K. Nikolić, and M. Forshaw, "Parallel Information and Computation with Restitution for Noise-Tolerant Nanoscale Logic Networks," *Nanotechnology*, vol. 15, no. 1, Jan. 2004, pp. 192-210.
5. S. Roy, V. Beiu, and M. Sulieman, "Majority Multiplexing: Economical Redundant Fault-Tolerant Designs for Nano Architectures," to appear in *IEEE Trans. Nanotechnology*; http://www.eecs.wsu.edu/~vbeiu/Publications/2005%20T_NANO.pdf.
6. J. Han and P. Jonker, "A Defect- and Fault-Tolerant Architecture for Nanocomputers," *Nanotechnology*, vol. 14, no. 2, Feb. 2003, pp. 224-230.
7. R.I. Bahar, J. Chen, and J. Mundy, "A Probabilistic-Based Design for Nanoscale Computation," *Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation*, S.K. Shukla and R.I. Bahar, eds., Springer, 2004, pp. 133-156.
8. S. Krishnaswamy et al., "Accurate Reliability Evaluation and Enhancement via Probabilistic Transfer Matrices," *Proc. Design, Automation and Test in Europe* (DATE 05), IEEE CS Press, 2005, pp. 282-287.

We can readily extend bifurcation analysis for finding error threshold values to arbitrary K-input NAND gates, majority gates, and any combinations of logic gates.[11] The error threshold value for a noisy three-input NAND gate is 0.1186 and reaches 0.1330 for a five-input NAND gate, which is nearly 50% larger than that for $K = 2$. Because of the inclusion of inverters, however, the error threshold value for a combination of NAND gates and inverters decreases to 0.0107. As we discuss later, NAND gates and inverters are the basic building blocks of our new design of random interwoven redundant logic.

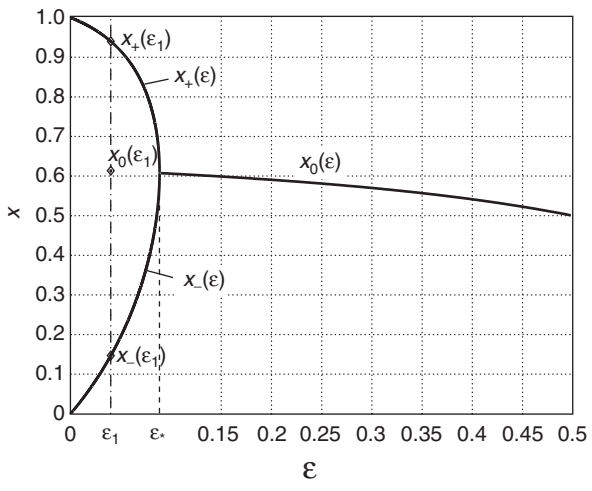


Figure 5. Bifurcation diagram for the faulty two-input NAND gate.

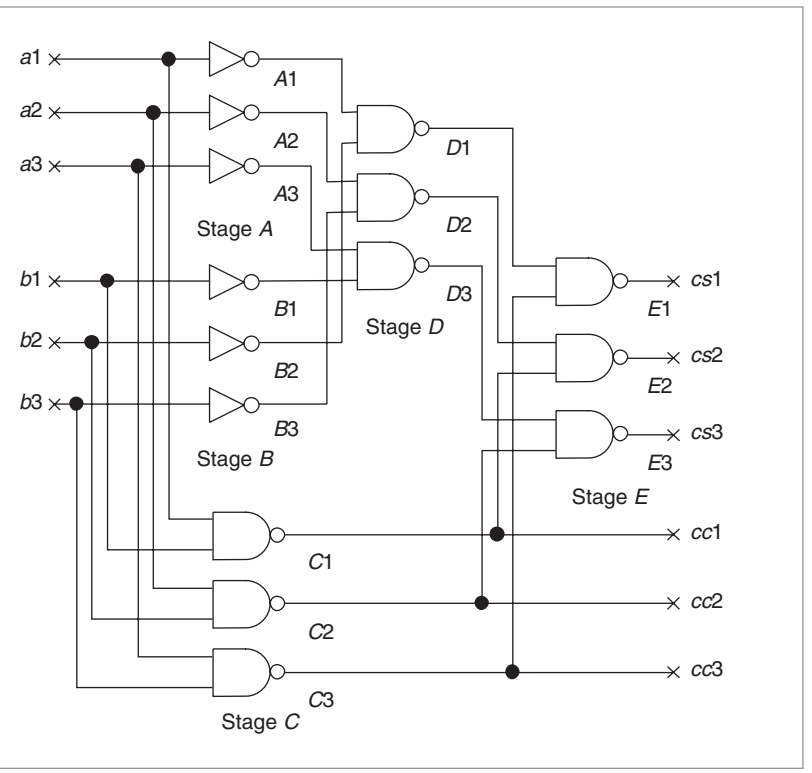


Figure 6. TIR implementation of the complementary half adder.

an ad hoc network to construct the bifurcation nonlinear map. Obviously, neither of the two structures represents a practical implementation. To improve upon this, we present a new design of interwoven redundant logic, called random interwoven redundancy, which can serve as the basis for building any realistic circuit.[12] We investigate the fault tolerance of random interwoven redundant circuits through a simulation-based experimental approach, which we present in the next section.

To explain the notion of random interwoven redundancy, we start with its simplest form, triplicated interwoven redundancy (TIR). Figure 6 shows the schematic of a TIR implementation for the complementary half adder in Figure 2. The TIR circuit triplicates each NAND gate in the nonredundant circuit, as well as all the interconnections. A TIR circuit thus has three times as many gates and interconnections as the corresponding nonredundant circuit.

The interconnections in a TIR circuit are, in principle, arranged randomly. In a practical implementation, it's possible to substitute the random interconnections with arbitrarily selected static ones that have specific routes. In a TIR circuit comprising two-input NAND gates, for instance, there are six possible pair connections: {(1,1), (2,2), (3,3)}, {(1,1), (2,3), (3,2)}, {(1,2), (2,3), (3,1)}, {(1,2), (2,1), (3,3)}, {(1,3), (2,1), (3,2)}, and {(1,3), (2,2), (3,1)}. Our notation (i,j) means that the output of gate i in a triplet of gates, pairs with the output of gate j in another triplet to form the inputs of a gate in the next stage. The total interconnect pattern becomes 36 (or 6×6) if we distinguish among the gate orders of a triplication in the next stage. One method of arranging the interconnections is to randomly adopt one of the 36 connection patterns for all connecting pairs in adjacent layers. As shown in Figure 6, the interconnect patterns used in the three layers from inputs to outputs of the circuit are {(1,1), (2,2), (3,3)}, {(1,2), (2,3), (3,1)}, and {(1,3), (2,1), (3,2)}, although the circuit can use any other interconnect pattern.

Notice that, if we use pattern {(1,1), (2,2), (3,3)} in all layers for all interconnections, the circuit in Figure 6 will perform a computation as three independent modules—it will actually work as a TMR circuit, as depicted in Figure 7. TIR is hence a generalization of TMR to allow for random interconnections. In our previ-

Random interwoven redundancy

To make a statistical analysis, Markov chain models adopted a highly abstract circuit structure consisting of chains of identical multiplexing stages. While in the process of searching for error bounds, we actually used

ous work, we showed that the reliability of a TIR circuit with random interconnections is in general comparable with that of its TMR equivalent, although for certain interconnect patterns the TIR structure can present an inferior performance to TMR because of its interwoven nature in gate interconnections (by which a single error's effect in a circuit is not confined to only one set of outputs).[12] The randomness in the TIR interconnections is particularly interesting in the physical implementation of molecular electronics, for which stochastic chemical assembly will most likely be the manufacturing method.

The principle of TIR is applicable to arbitrary logic circuits. A general procedure for constructing a TIR circuit is as follows:

1. Start with a nonredundant form of the circuit.
2. Triplicate each gate.
3. Following the interconnect pattern of the nonredundant circuit, randomly select a gate from a triplet to use as an input for a gate that has no other inputs from the same triplet.
4. Repeat Step 3 until all the gates are connected in the TIR circuit.

As in TMR, a TIR circuit requires a decision element (a voter) as a restoring device. TIR can be extended to higher orders, namely, *N*-tuple interwoven redundancy (NIR), similar to the extension of TMR to NMR. Hence, NIR is a generalization of NMR, but with random interconnections. We have previously shown that the design and implementation of voters are important for the NIR structure.[12] In general, using a simpler voter design allows for better reliability in an NIR circuit, whereas an increased complexity of voters in a higher-order NIR can possibly degrade system performance. Since NIR, unlike NMR, does not require systematic interconnections, it presents minimum precision requirements for manufacturing devices and interconnects. It is therefore favorable for implementing defect- and fault-tolerant nanoarchitectures.

Experimental studies

Traditionally, there have been two different methodologies in designing fault-tolerant systems.[5] One aims at constructing reliable systems based on unreliable components. In this approach, all components are subject to a certain error probability, and thus the number of faults scales up as the size of a system increases. The other approach focuses on building systems that can tolerate a fixed number of faults. We shall now show, by using fault injection simulation and our simulation-based reliability model, that it is possible to examine these two fault scenarios at the same time.

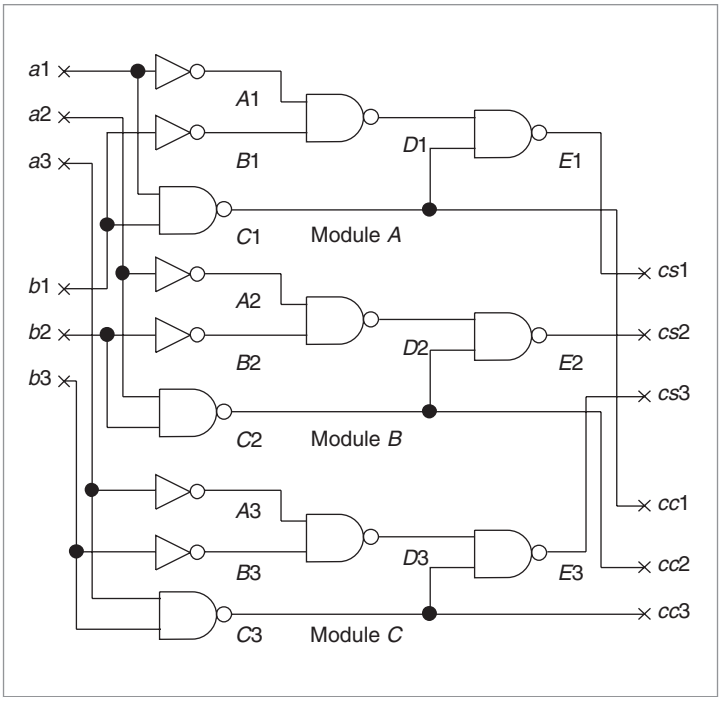


Figure 7. TMR configuration of the TIR complementary half adder.

Simulation-based reliability model

Researchers have proposed various analytical methods for reliability evaluation, but most are either extremely complex or present inaccurate predictions for practical designs. In this study, we perform a fault injection simulation and develop a simulation-based reliability model to investigate the effects of multiple component failures in quadded and TIR structures. The simulation procedure is as follows:

1. Starting from an initially fault-free circuit, randomly select m (initially equal to 1) faulty gates and emulate a von Neumann (inversion) fault as an erroneous gate output.
2. Apply a set of input patterns to the circuit, which contains injected faults. If the circuit provides the correct outputs, repeat this step until the complete set of input patterns has been tested. Otherwise, increase the number of failed simulations, k, by 1.
3. Increase the number of simulations performed thus far, i, by 1. If i is less than N, the total number of simulations to be performed (N = 1,000 here), go to

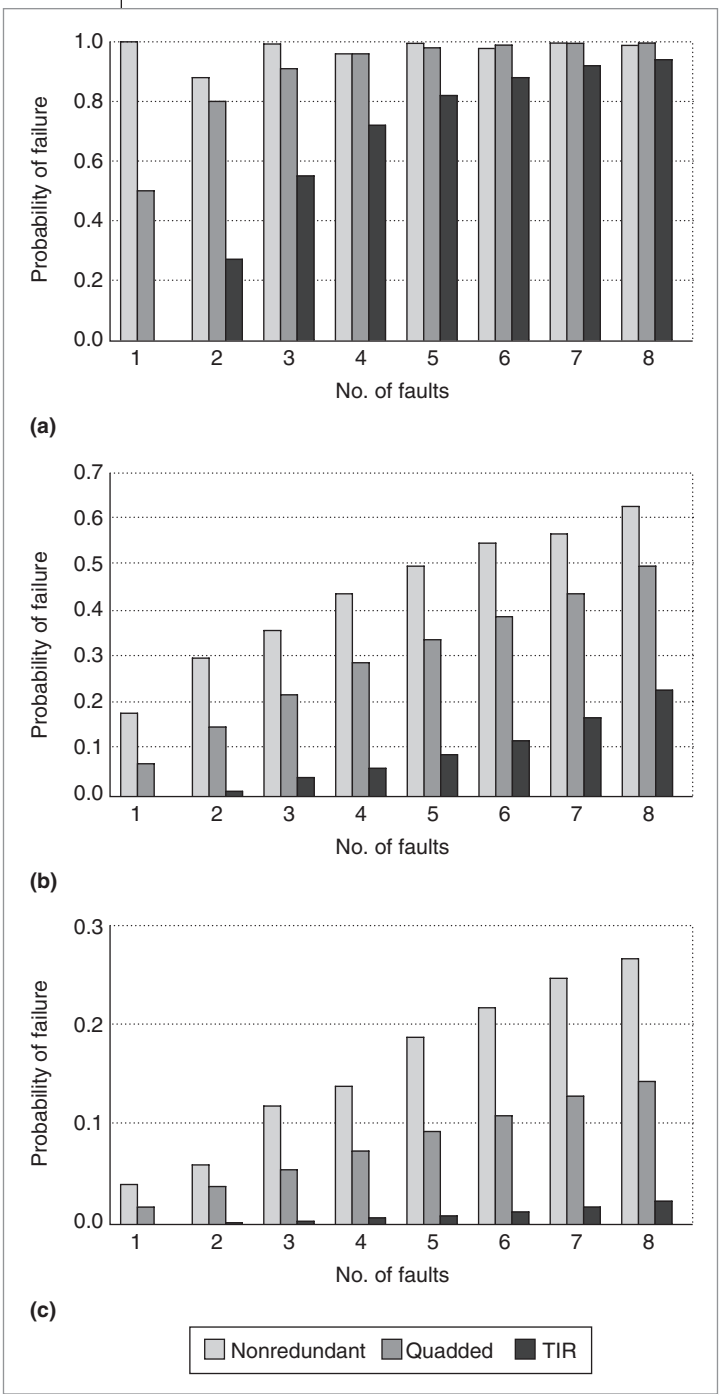


Figure 8. Failure probabilities of nonredundant, quadded, and TIR circuits with 2 (a), 8 (b), and 32 (c) stages of half adders, for a fixed number of faults.

Step 1. Otherwise, proceed to Step 4.

4. Compute the failure rate of the simulated circuit for the m injected faults: $F_m = k/N$. Increase the number of faults injected into the circuit, m, by 1. If m is no larger than the maximum number of faults, go to Step 1. Otherwise, end the simulation.

In this simulation, we assume that faults appear in logic gates, thus producing possibly faulty signals at the outputs of the erroneous gates. Interconnect faults, though not considered in this study, can readily be modeled in the simulation by accounting for possible errors at both inputs and outputs. This simulation procedure is, in principle, applicable to any fault-tolerant circuit.

Each simulation produces a failure rate, F_m, for the number of faults injected into the circuit. If every logic gate has the same error rate, and errors are randomly and independently distributed, the probability of a number m of faulty gates in a circuit follows the binomial distribution (given by the Poisson approximation)

$$P(m) = \frac{(np)^m e^{-np}}{m!}$$

where p is the error probability of a logic gate, and m and n are, respectively, the number of faulty gates and the total number of gates in the circuit.

If we assume the number of faults, m, to be a random variable, we can use failure rate F_m as a failure distribution in m. We can then obtain reliability distribution R_m, which gives the probability that a circuit will continue to properly operate in the presence of m faults: $R_m = 1 - F_m$. We therefore obtain the reliability of the fault-tolerant circuit by summing up all the conditional reliabilities with the presence of faults; that is,

$$R = \sum_{m=0}^{n} R_m P(m) \qquad (3)$$

Hence, we can obtain the reliability of a (fault-tolerant) circuit from the simulation-based formula (Equation 3).

Simulations and experimental results

We applied our simulation procedure to the nonredundant, quadded, and TIR forms of the complementary half adder. To investigate their long-term reliability, we cascaded the half adders to form multiple-stage systems; that is, we connected the two outputs of a half adder to the inputs of the half adder in the next stage. We can also describe this structure as a serial half adder working in a temporal domain by feeding back its own outputs to its inputs. Note that we assume errors to be transient in this case; they only affect one stage of the circuit.

In Figure 8, we show the failure rates we obtained for the nonredundant, quadded, and TIR circuits consisting of 2, 8, and 32 stages of half adders for a fixed num-

ber of up to 8 faults. TIR requires inserting triple voters for each stage as intermediate restoring devices, unlike the quadded structure, which does not require restoring. The voters are assumed to be realized using single-majority gates and thus have the same error rate as the NAND gates. Although other voter designs are possible, we use the single-gate structure here because of its implementability using several nanoscale devices and the relatively small size of the half adder. As a result of the voter inclusion, the gate redundancies involved in the TIR and quadded half adders become similar.

In general, as shown in Figure 8, the quadded structure presents an improved performance over the nonredundant form except for a few cases of multiple faults in the two-stage circuits (Figure 8a). In a small-scale circuit, the quadded structure is more vulnerable to errors because a relatively large percentage of its gates are critical ones. The TIR structure has the best overall performance for all circuits of different lengths. This is partly because of a high requirement we impose on the quadded structure—we consider a quadded circuit reliable only when all of its final outputs are correct, whereas a TIR circuit is reliable as long as a majority of its final outputs are correct.

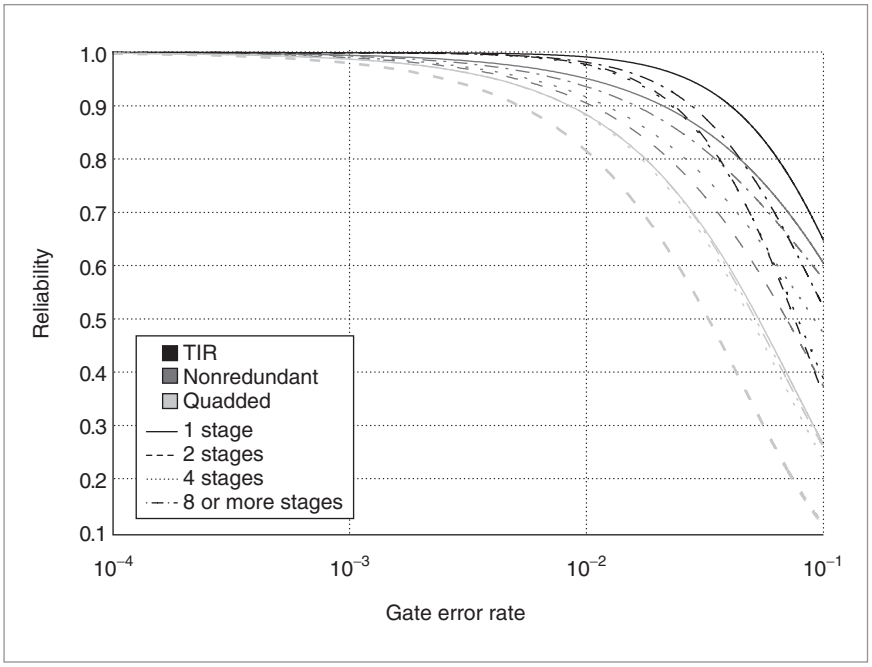


Figure 9. Reliabilities of nonredundant, quadded, and TIR circuits of multiple-stage half adders.

Comparing the same types of circuits with different lengths shows that a lower failure rate results in a larger circuit. In the presence of a single fault, for example, the nonredundant structure's failure rate falls from 100% for a two-stage circuit down to approximately 4% for a 32-stage circuit. We can draw similar conclusions for both the quadded and TIR structures. This indicates that, for a fixed number of faults, circuits with larger logical depth tend to have better fault tolerance. This characteristic is particularly interesting for a fault-tolerant design against transient errors, for which single-event upsets are the most common cause.

We present the results we obtained by using the simulation-based reliability model (which assumes a constant gate failure rate) to account for increases in the numbers of faults with increasing circuit size (that is, number of stages). In Figure 9, we plot the reliabilities for the nonredundant, quadded, and TIR circuits of different lengths against the gate error rate. For all three structures, as Figure 9 shows, the one-stage half adders perform the best. This is actually because of their specific structures: In a multiple-stage circuit, the fan-out appears to have a great impact on error propagation. The circuit's reliability improves as the number of stages increases—because of the compensating effects of multiple errors—until it reaches a stationary state. This is consistent with the analytical results from using the Markov-chain models.

The TIR structure indeed provides the best fault tolerance, as Figure 9 reveals, whereas the quadded structure actually shows a worse reliability than the nonredundant form. Figure 9 also shows that the reliabilities of the nonredundant circuits start to descend rapidly at a gate error rate of approximately a few multiples of 10^{-3}, which is bounded by the theoretical error bound obtained for combinations of NAND gates and inverters. However, using the TIR structure elevates this threshold value to approximately 10^{-2}. Though the TIR structure has better performance than the nonredundant one, this is true only when the gate error rate is strictly no larger than a threshold. For the TIR circuits in our study, this value is approximately 0.07 for an (infinitely) large logical depth; it differs for a circuit of lesser depth.

Finally, we consider random interwoven structures with different degrees of redundancy. Figure 10 shows the reliabilities obtained in our simulations for 3-tuple (TIR), 5-tuple (5IR), and 7-tuple (7IR) interwoven redundancy circuits. It is evident that the higher the redundancy

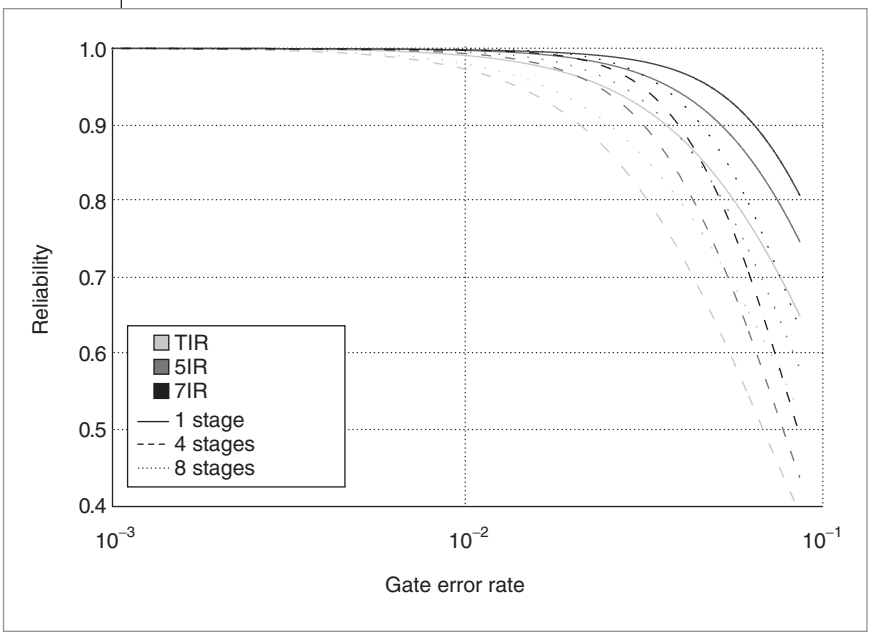


Figure 10. Reliabilities of NIR circuits of multiple-stage half adders.

degree, the better the reliability. Hence, higher threshold values for descending reliability require NIR circuits with higher degrees of redundancy. For NIR, we assume that each voter implementation uses a single-majority gate—the use of complex voter designs might degrade the system reliability of NIR as its degree of redundancy increases.[12] This indicates the significance of voters in an implementation of random interwoven redundancy.

WE HAVE SHOWN that the complexity and number of critical gates (that is, marginal gates in quadded circuits and voters in NIR) are important for the reliability of a fault-tolerant circuit. Therefore, a general idea for fault-tolerant design is to minimize the complexity and fraction of critical gates in a circuit. In our study of TIR and NIR, we inserted triple voters at each stage. This restoring process can in principle be applied to multiple stages to reduce redundancy. However, the issue of granularity in redundancy, as well as reliability-redundancy trade-offs, await further investigation. ■

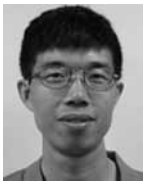
Acknowledgments

This work was supported by NASA award NCC 2-1363 and NSF ITR grant 0135946. It was partially supported by the Delft University of Technology's DIRC program, "Novel Computation Structures Based on Nanoelectronic and Quantum Devices."

References

1. J. von Neumann, "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components," *Automata Studies*, C.E. Shannon and J. McCarthy, eds., Princeton Univ. Press, 1956, pp. 43-98.
2. W.H. Pierce, *Failure-Tolerant Computer Design*, Academic Press, 1965.
3. J.G. Tryon, "Quadded Logic," *Redundancy Techniques for Computing Systems*, R.H. Wilcox and W.C. Mann, eds., Spartan Books, 1962, pp. 205-228.
4. P.A. Jensen, "Quadded NOR Logic," *IEEE Trans. Reliability*, vol. 12, no. 3, Sept. 1963, pp. 22-31.
5. N. Pippenger, "Developments in 'The Synthesis of Reliable Organisms from Unreliable Components,'" *Proc. Symposia in Mathematics*, American Mathematical Society, 1990, pp. 311-324.
6. S. Spagocci and T. Fountain, "Fault Rates in Nanochip Devices," *Proc. Electrochemical Society*, The Electrochemical Society Inc., 1999, vol. 98-19, pp. 582-593.
7. J. Han, "Fault-Tolerant Architectures for Nanoelectronic and Quantum Devices," doctoral dissertation, Delft University of Technology, 2004; http://www.qi.tnw.tudelft.nl/~jie.
8. P. Graham and M. Gokhale, "Nanocomputing in the Presence of Defects and Faults: A Survey," *Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation*, S.K. Shukla and R.I. Bahar, eds., Springer, 2004, pp. 39-72.
9. K. Nikolić, A. Sadek, and M. Forshaw, "Fault-Tolerant Techniques for Nanocomputers," *Nanotechnology*, vol. 13, no. 3, June 2002, pp. 357-362.
10. J. Han and P. Jonker, "A System Architecture Solution for Unreliable Nanoelectronic Devices," *IEEE Trans. Nanotechnology*, vol. 1, no. 4, Dec. 2002, pp. 201-208.
11. J.B. Gao, Y. Qi, and J.A.B. Fortes, "Bifurcations and Fundamental Error Bounds for Fault-Tolerant Computations," to appear in *IEEE Trans. Nanotechnology*.
12. J. Han and P. Jonker, "From Massively Parallel Image Processors to Fault-Tolerant Nanocomputers," *Proc. 17th Int'l Conf. Pattern Recognition* (ICPR 04), IEEE CS Press, 2004, pp. 2-7.

Jie Han is a postdoctoral fellow at the Advanced Computing and Information Systems (ACIS) Laboratory in the Department of Electrical and Computer Engineering, University of Florida. His research interests include fault-tolerant nanocomputing, fault tolerance issues in microarchitectures, and novel systems based on nanoscale and quantum devices. Han has a BSc in electronic engineering from Tsinghua University, China, and a PhD from the Delft University of Technology, the Netherlands. He is a member of the IEEE.



Jianbo Gao is an assistant professor in the Department of Electrical and Computer Engineering at the University of Florida. His research interests include nano- and fault-tolerant computing, nonlinear time series analysis, traffic modeling in communications networks, nonlinear dynamics of the Internet, characterization of non-Gaussian noise in radio communications channels, and bioinformatics. Gao has a PhD from the University of California, Los Angeles.

Yan Qi is pursuing a PhD in biomedical engineering at Johns Hopkins University. Her research interests include fault-tolerant computation, nanoelectronic system architectures, and statistical modeling and network dynamics analysis in computational biology and bioinformatics. Qi has a BSc in electrical engineering from Zhejiang University, China, and an MSc in electrical and computer engineering from the University of Florida.

Pieter Jonker is an associate professor in the Pattern Recognition Group (now Quantitative Imaging Group) of the Department of Applied Physics at the Delft University of Technology. His research interests include software and hardware architectures for embedded systems, including machine vision; autonomously soccer-playing robots; and fault-tolerant architectures for novel, nanoscale computer elements. Jonker has an MSc in electrical engineering from the Twente University of Technology and a PhD in applied physics from the Delft University of Technology. He is a member of the IEEE and a Fellow of the International Association for Pattern Recognition.

José A.B. Fortes is a professor and BellSouth Eminent Scholar in the Department of Electrical and Computer Engineering and the Department of Computer and Information Science and Engineering at the University of Florida. His research interests include network computing, distributed information processing systems, advanced computing architectures, and nanocomputing. Fortes has a PhD in electrical engineering from the University of Southern California. He is a Fellow of the IEEE and was a Distinguished Visitor of the IEEE Computer Society from 1991 to 1995.

■ Direct questions and comments about this article to Jie Han, Advanced Computing and Information Systems Lab, Dept. of Electrical and Computer Engineering, University of Florida, 339 Larsen Hall, PO Box 116200, Gainesville, FL 32611-6200; jiehan@acis.ufl.edu.

For further information on this or any other computing topic, visit our Digital Library at http://www.computer.org/publications/dlib.