On the Reliability of Computational Structures Using Majority Logic

Jie Han, *Member, IEEE*, Erin R. Boykin, Hao Chen, Jinghang Liang, and José A.B. Fortes, *Fellow, IEEE*

***Abstract*—The importance of the reliability of majority-based structures stems from their use in both conventional fault-tolerant architectures and emerging nanoelectronic systems. In this paper, analytical models are developed in order to gain a better understanding of the reliability of majority logic in these contexts. A minimally biased input scenario for N-input majority gates (N odd) occurs when only a minimal majority of the inputs are in consensus. In a tree of gates with these inputs, this paper determines 1) that any nonzero error rate of the majority gates and/or of its initial inputs will result in an unreliable output and 2) that the use of majority gates with a larger number of inputs leads to a less reliable structure. These results are extended to N-input minority gates for odd N. Although these findings are based on tree structures, their implications to circuit design are explored by investigating several fault-tolerant and nanoelectronic architectures. The simulation results show that the increased probability of error in nanoscale devices may impose serious constraints on the reliability of emerging nanoelectronic circuits, as well as their fault-tolerant counterparts. The worst case reliability must be accounted for in a fault-tolerant design to ensure reliable operation.**

***Index Terms*—Majority logic, nanoelectronics, quantum-dot cellular automata (QCA), reliability, triple/N-tuple modular redundancy (TMR/NMR).**

I. Introduction

As CMOS devices reach their fundamental physical limits, they will increasingly suffer from short channel effects, doping fluctuations, and other phenomena, which will negatively impact their reliability. In addition, their manufacture will require difficult and expensive lithographic methods. These future limitations of CMOS have led many to consider novel nanometer-scale devices that are hoped to have faster switching speeds, lower power consumption, and better scaling characteristics [1].

Manuscript received April 5, 2010; revised November 1, 2010; accepted January 24, 2011. Date of publication; date of current version.This work was supported by the Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC), by the startup fund from the Faculty of Engineering, University of Alberta, Edmonton, AB, Canada, in part by NASA under Award NCC 2-1363, in part by the National Science Foundation Information Technology Research (NSF ITR) under Grant 0135946, and by the BellSouth Foundation. The review of this paper was arranged by Associate Editor Sorin D. Cotofana.

J. Han, H. Chen, and J. Liang are with the Department of Electrical and Computer Engineering at the University of Alberta, Edmonton, AB, T6G 2V4, Canada (e-mail: jhan8@ualberta.ca; hc5@ualberta.ca; jinghang@ualberta.ca).

E. R. Boykin and J.A.B. Fortes are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32601 USA (e-mail: erinrt@gmail.com; fortes@ufl.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNANO.2011.2111460

Such novel devices include quantum-dot cellular automata (QCA) [2], [3], which utilize the electrostatic or magnetic interactions between neighboring cells to perform logic, single-electron tunneling (SET) devices [4], [5], which are based on quantum tunneling and single-electron operations, and spin-based devices [6], [7], which employ spins of electrons instead of charges as information carriers. In demonstrating the feasibility of any of these new technologies, it is important to show that the device can implement one of the universal logic functions and, thus, can perform any arbitrary function. For example, QCA have been shown to be capable of performing both the majority and inverting logic functions, which together constitute the universal minority function [2]. Magnetic QCA have been experimentally demonstrated for the implementation of majority gates at room temperature [3]. The minority function has been shown using SET devices [5]. Spintronic implementations of the majority function have recently been proposed as the building blocks of low-power circuits and systems [6], [7]. As new nanoelectronic devices emerge, many of which perform computation based on majority or minority logic, there will be an increased interest in the fundamental characteristics of these logic functions.

Also important to the success of any device (including scaled CMOS) is reliability. For devices at the nanometer scale, however, reliability will be adversely affected by background charges, dynamic variations in the operating environment, such as temperature fluctuations, and manufacturing defects because of the inability to precisely control the fabrication process. In the fault-tolerant architectures proposed for nanoelectronics, many are based on the use of majority logic. It is, therefore, pertinent to understand the fault-tolerant characteristics of a system built upon the majority logic function, as well as its universal derivative, the inverting majority or minority function.

The use of majority logic in reliable system design can be traced back to J. von Neumann's study on using unreliable components to synthesize reliable systems [8]. In his research, von Neumann proposed to study the probabilistic characteristics of a system in which each component can fail independently with probability ε ($\varepsilon \leq 1/2$). He concluded that a system built from unreliable components can compute reliably when ε is sufficiently small. In general, a reliable system is defined as one that performs computation with a probability of output error less than 1/2. As von Neumann stated, when the probability of output error reaches 1/2, the results from computation become irrelevant to the inputs and restoration of the outputs to their correct signal values is not possible.

In von Neumann's original work [8] and in subsequent research [9], [10], the issues of reliable computation were first

addressed as a problem of "remembering" a binary variable in a noisy circuit. The capability of remembering a bit is considered a prerequisite for reliably computing nontrivial functions of more variables. In the special (and favorable) case where inputs have identical nominal values, von Neumann showed that, if the error rate of each gate is equal or larger than 1/6, a bit of information cannot be remembered in a long computation using three-input majority gates. Hajek and Weller proved later that 1/6 is the error threshold for reliable computation using general three-input gates that include majority gates [9]. Evans and Schulman extended this result to general N-input gates for odd N and showed that the threshold increases as the number of inputs increases [10].

In these studies, the authors considered formulas built from gates, where a formula, in this context, is an interconnection of gates such that the output of each gate is an input to at most one other gate, and there are no loops or fanouts in the sequence of interconnections. Although it has not yet been shown that these results obtained for formulas also apply to networks, which are acyclic interconnections of gates that allow for fanouts and are, therefore, more representative of real circuits; it is believed that this is the case [11], [12].

Roy and Beiu [13] contributed to the study on majority-based architectures by analyzing a model of von Neumann's multiplexing scheme. They note that the worst case input scenario for majority gates intuitively occurs when the inputs are not in consensus. The error characteristics of several types of logic gates, including majority, are studied in [14] using bifurcation analysis. The reliability issues in a number of majority-based full-adder designs are discussed in [15] and [16], which have been recently extended to consider the effects of device failures and input vectors [17], [18]. Full adders are the basis of most logic processing units and have been proposed for implementations in several nanotechnology devices [2], [5], [7].

Two fault-tolerant system designs, triple modular redundancy (TMR) and N-tuple modular redundancy (NMR) have been serving as benchmark techniques and have been successfully implemented in many critical applications [19]. Recently, a study exploring the optimal design tradeoffs of reliability and redundancy was performed on a large-scale system consisting of highly unreliable nanoscale devices [20]. It was concluded that the use of TMR/NMR in such a system will incur a high cost in redundancy and thus in power and area as well. The TMR techniques are applied to QCA architectures in [21] with the aim of improving system reliability. However, it is found that TMR is not very effective, and that cascaded TMR may only be beneficial when the device error rate is very low. It is also shown that the reliability of QCA circuits is influenced by the physical implementations of wires and crossovers [22], [23], as well as that of the majority function [24].

In this paper, the limitations of TMR/NMR, as well as other computational structures using majority/minority logic, are investigated through a logic-level analysis. Particularly, the problem of remembering the information carried in the majority of bits is studied in the general case when the nominal inputs to majority/minority gates can have different values. When the inputs to three-input majority gates are not in consensus, it is shown that the information will be lost in a large tree structure when the gate or input error rate is not zero. It means that in this particular case, any nonzero error rate in the gates or of initial inputs will not be tolerated in a long computation. It is further shown that the use of majority gates with a larger number of inputs results in a less reliable structure in this case. These results are complementary to those in most previous studies and, thus, lead to a better understanding of the reliability of majority gates. The implications of these results are explored by evaluating several computational architectures that closely resemble this tree structure of majority/minority gates. While beyond the scope of this paper, the simulation results obtained in this paper can further be enhanced by incorporating noise in wires and devices in the analysis.

This paper is a significant extension of [25] and is organized as follows. In Section II, the error characteristics of three-input majority gates are described. In Section III, a tree structure of three-input majority gates is constructed for the analysis of its reliability. This analysis is extended to N-input majority and minority gates in Sections IV and V. Several case studies are presented in Section VI, and Section VII concludes the paper.

II. Error Characteristics of Three-Input Majority Gates

In this section, the error characteristics of three-input majority gates are investigated in the general case when their inputs are allowed to have different nominal values. To analyze the reliability of a gate, we use probabilistic gate models (PGMs), which relate the probabilistic value of a gate's output to its probabilistic inputs and error rate ε [26]. For the von Neumann fault model, which assumes that a gate flips its correct output with probability ε, the PGM for a three-input majority gate with output probability z and input probabilities x, y, and w is given by

$$z = \varepsilon + (1 - 2\varepsilon)(xy + xw + yw - 2xyw). \tag{1}$$

Here, the variables x, y, w, and z denote the probability of a signal being logical 1 (all results and reasoning would be similar, if instead using logical 0, due to the symmetry of the majority function). For convenience, the simple term "probability" is used to mean "probability of being logical 1" throughout the paper, unless it is otherwise noted.

In what von Neumann considered a "special case," the probabilities of errors in inputs are statistically independent and all inputs are expected to be in the same state of either logical 1 or logical 0. Let p and p' be the error probabilities of inputs and output, respectively; we obtain

$$p' = \varepsilon + (1 - 2\varepsilon)(3p^2 - 2p^3). \tag{2}$$

It has been shown that an improvement in reliability cannot be made through the use of majority gates when $\varepsilon \geq 1/6$ [8]. This is illustrated in Fig. 1. For $\varepsilon = 0.2$, as shown, the output of the majority gate becomes less reliable than its inputs, and therefore, the output probability slips into an indistinguishable state of 1/2 in a long computation. For $\varepsilon = 0.05$, however, the output error probability can be smaller than the input error probability. It

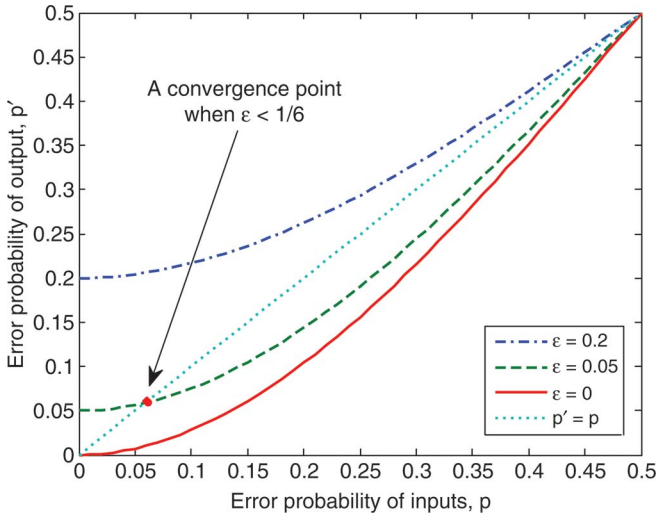


Fig. 1. Output error probability versus input error probability of three-input majority gates with gate error probability ε; inputs are expected to have identical nominal values.

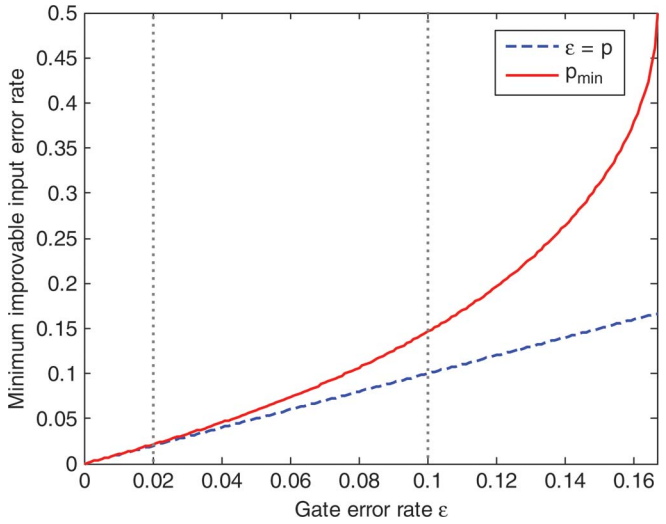


Fig. 2. Minimum input error probability that could be improved by a three-input majority gate is a function of the gate error rate ε. The figure is divided into three regions according to the relationships between $p_{\min}$ and ε.

converges into a stable point at the intersection of the output curve with the diagonal $p' = p$. For $\varepsilon = 0$, particularly, the output is more reliable than the inputs and in a long computation, it asymptotically approaches a perfectly reliable state for any imperfect inputs. This is the ideal case for many fault-tolerant applications, such as those using TMR/NMR techniques with highly reliable majority voters.

In the case of $\varepsilon < 1/6$, the output curve intersects with the diagonal $p' = p$. Whether an output is more reliable depends on whether the input error is larger than that indicated by the intersection. This intersection thus gives the minimum input error rate that can be improved by a majority gate. This minimum improvable input error rate is determined by the value of ε and can be found by solving (2) and $p' = p$. This leads to a solution of

$$p_{\min} = \left(\frac{1}{2}\right)\left(1 - \sqrt{\frac{1-6\varepsilon}{1-2\varepsilon}}\right) \tag{3}$$

as plotted in Fig. 2.

According to the values of ε and $p_{\min}$, Fig. 2 can be divided into three regions: 1) $0 \le \varepsilon \le 0.02$, where $p_{\min}$ is approximately equal to ε; 2) $0.02 < \varepsilon \le 0.1$, where $p_{\min}$ is noticeably larger than ε; and 3) $0.1 < \varepsilon < 1/6$, where $p_{\min}$ is significantly larger than ε. This indicates that, for a majority gate with noise ε, its input error probability has to be larger than the corresponding $p_{\min}$ in order to have a more reliable output.

We then consider the case when the inputs are expected to have different logic values. Assume two of the three inputs have a different nominal value than the third. If each input has the same error probability p, the output error probability p' is given by

$$p' = \varepsilon + (1-2\varepsilon)(2p - 3p^2 + 2p^3). \tag{4}$$

For any $p \in (0, 1/2)$, as shown in Fig. 3, p' is larger than p, regardless of the value of ε. In fact, the curve of p' only intersects

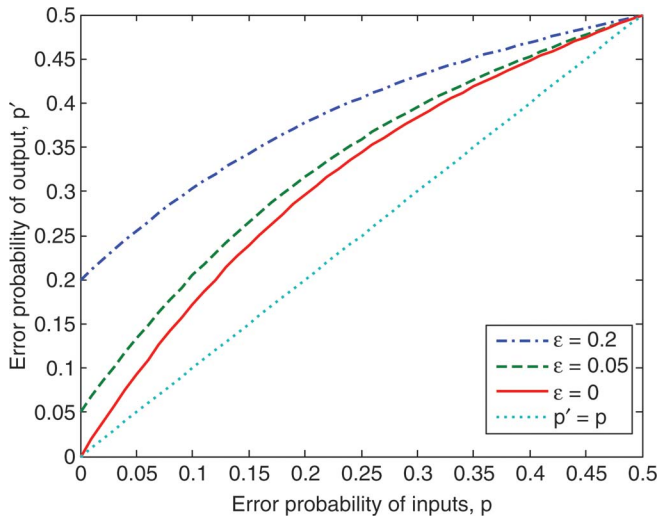


Fig. 3. Output error probability versus input error probability of the three-input majority gates with gate error probability ε; inputs are expected to have different nominal values.

with the diagonal $p' = p$ at (1/2, 1/2) for $\varepsilon > 0$, or at (1/2, 1/2) and (0, 0) for $\varepsilon = 0$. This can be seen as follows.

Since we have $p' = p$ at each intersection and (1/2, 1/2) is clearly an intersection point for any ε, taking $p' = p$ into (4) gives us

$$\left(p - \frac{1}{2}\right)((4\varepsilon - 2)p^2 + (2 - 4\varepsilon)p + 2\varepsilon) = 0. \tag{5}$$

The intersections other than the one at (1/2, 1/2) are given by the roots of

$$(4\varepsilon - 2)p^2 + (2 - 4\varepsilon)p + 2\varepsilon = 0 \tag{6}$$

as

$$p = \frac{1}{2}\left(1 \pm \sqrt{\frac{1+2\varepsilon}{1-2\varepsilon}}\right). \tag{7}$$

From (7), we obtain $p = 0$ and $p = 1$ for $\varepsilon = 0$, while for any $\varepsilon \in (0, 1/2)$, no valid solution exists for p in the interval [0, 1]. This indicates that when the nominal inputs to the majority gate are expected to be different, its output is less reliable than its inputs for any $\varepsilon \in (0, 1/2)$. It is interesting to note that even when the gates are fault free, i.e., when $\varepsilon = 0$, any uncertainty in the inputs to the majority gate results in a less reliable output, which in a long computation asymptotically reaches 1/2 and becomes irrelevant to the inputs.

III. Tree Structure of Three-Input Majority Gates With Minimally Biased Inputs

A. Construction

In this section, the case illustrated in Fig. 3 is considered and a computational structure is constructed. In contrast to the special case where a tree structure is used to remember a single bit of information, a similar tree structure is used here to remember a minimal majority of binary values. A minimal majority is defined as $K + 1$ for a majority gate with $2K + 1$ inputs ($K \geq 1$). In order to develop a model for analysis, we start with one three-input majority gate, and build up to an arbitrarily large network.

For a single gate, its inputs are assumed to be either (0,0,1) or (1,1,0), i.e., exactly two of the inputs are equal, and the third is their complement. Since the order of the inputs does not matter for a majority gate, this is equivalent to (0,1,0) and (1,0,0) or (1,0,1) and (0,1,1), respectively. Then, we add a second computation layer to the formula by inserting gates whose outputs serve as the inputs to the initial layer. At this second layer, there can be a maximum of nine (3^2) initial inputs, each of which are fed into one of the three majority gates. By continuing this process, the final result is a ternary tree of L layers ($L \geq 1$) with 3^L initial inputs and 3^{L-1} gates in the widest layer. The final structure is illustrated in Fig. 4.

Note that, for the ease of reference in later analysis, the widest layer of the tree is referred to as the first layer and the original single majority gate is considered to be the last layer L. A layer here denotes a group of gates that have the same distance to the last majority gate at the end of the tree. In the same layer, all gates' outputs have to go through the same number of majority gates to reach this last gate.

Let us assume that a majority gate at the ith layer has the nominal inputs (1,1,0). Then, at the $i - 1$ layer, two of the three gates connected to this gate must produce an output value of "1," while the third produces the value "0." If each gate in the tree is constrained to the inputs (1,1,0) and (0,0,1), the nominal inputs to the two gates producing "1" are (1,1,0), and the inputs to the gate producing "0" are (0,0,1). The inputs of the gates in any arbitrary layer can thus be determined by the inputs of the subsequent layer. Starting with the output of the gate at the last layer and traversing the tree with the input deduction rule explained earlier, we obtain a set of initial inputs indicated in Fig. 4. These initial values will be the only ideal values in the formula, since each gate has a probability ε of producing an incorrect output. In Fig. 4, therefore, X_0 and X_1 are used to represent probabilistic signals that, in the error-free case, would be "0" and "1," respectively.

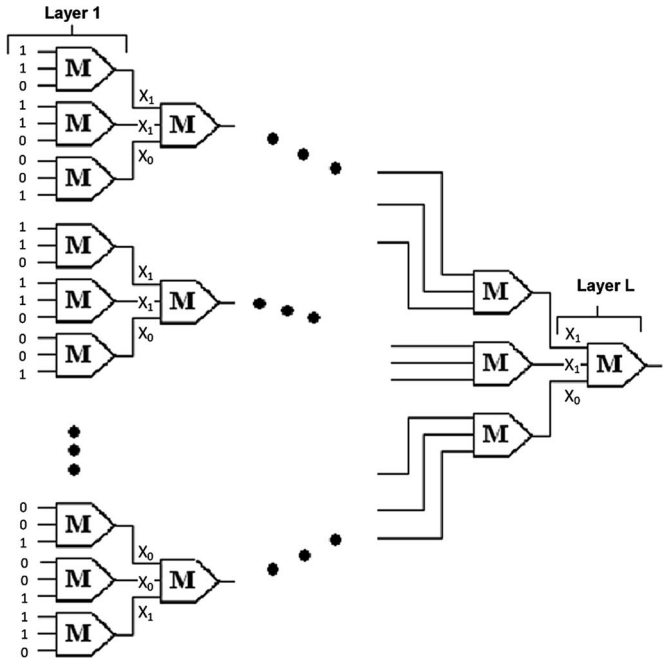


Fig. 4. Ternary tree of majority gates with the minimally biased inputs. It is shown that a majority of the input values are "1;" therefore, the final output of layer L, in the error-free case, is also "1."

At the inputs of an n-layer tree of the three-input majority gates, the minimum number of 1's (or 0's) needed to produce a "1" (or "0") is 2^n, which is actually in the minority of the total number of inputs, 3^n, for any $n > 1$ (these numbers are $(k + 1)^n$ and $(2k + 1)^n$ for majority gates with $2k + 1$ inputs). Since the signal that is carried in the minimal majority of the inputs is of our interest, the minimal-majority inputs, i.e., the nominal inputs that have a minimal majority to produce an expected output, are considered in this paper. If the inputs at each layer, as well as the inputs to each majority gate at this layer, are minimal-majority inputs, the inputs are referred to as the "minimally biased inputs." Any other inputs that are not minimally biased are not considered, unless otherwise noted.

B. Mathematical Model and Its Analysis

To analyze the output of a majority gate with inputs (X_1, X_1, and X_0), we first define the following function given by (1) for a majority gate with input probabilities (*a, a,* and b) when $\varepsilon = 0$:

$$f(a, b) = a^2 + 2ab - 2a^2b. \tag{8}$$

We further define $p_1 = \mathrm{Prob}(X_1 = 1)$ and $p_0 = \mathrm{Prob}(X_0 = 1)$. By appropriately substituting the values p_1 and p_0 for a and b, we obtain for a majority gate with inputs (X_1, X_1, and X_0) and nonzero ε

$$p_{1(1)} = \varepsilon + (1 - 2\varepsilon) f(p_1, p_0) \tag{9}$$

where $p_{1(1)}$ is the output probability of the majority gate with input (X_1, X_1, and X_0).

Similarly, the equation for the input combination (X_0, X_0, and X_1) is given by

$$p_{0(1)} = \varepsilon + (1-2\varepsilon)f(p_0, p_1). \tag{10}$$

Note that when the input combination is ($X_{1(i)}$, $X_{1(i)}$, and $X_{0(i)}$) or ($X_{0(i)}$, $X_{0(i)}$, $X_{1(i)}$), the output will be $X_{1(i+1)}$ or $X_{0(i+1)}$, respectively, where the subscript in parenthesis denotes the layer from which the value originates. Thus, a majority gate in the ith layer of the ternary tree can be modeled using one of the following equations:

$$p_{1(i+1)} = \varepsilon + (1-2\varepsilon)f(p_{1(i)}, p_{0(i)}) \tag{11}$$

$$p_{0(i+1)} = \varepsilon + (1-2\varepsilon)f(p_{0(i)}, p_{1(i)}). \tag{12}$$

Given independent and identical initial inputs, i.e., each initial input independently has the same probability to be "1" or "0," it is easy to see that the ternary tree of majority gates as described earlier and shown in Fig. 4 can be modeled by simultaneously iterating (11) and (12), and that at the ith layer, all the values of $p_{1(i)}$ are equal, as are all the values of $p_{0(i)}$.

In the case depicted in Fig. 4, one is the dominant value in the tree, since a majority of the gates have input (X_1, X_1, and X_0), including the majority gate at the last level. Because majority gates perform a symmetric function, the opposite case when zero is dominant can be modeled in a similar way.

Fig. 5 shows a plot of the final output of the tree versus ε, given perfectly reliable initial inputs. It was generated by iterating (11) and (12) for the 1-dominant tree and then plotting the output after 2, 5, and 15 iterations for various values of ε. This is repeated for the 0-dominant tree and the output is plotted on the same graph. An iteration here corresponds to the signal propagation through a layer of gates, or, in other words, an iteration of (11) or (12) produces an output of a layer from its inputs.

It can be seen that the plots for both trees converge asymptotically at $X = 1/2$, where X denotes the output. At this point, both the 1- and 0-dominant trees produce the identical output value, which means that the input value that was in the majority at the first layer of the tree was lost during computation. This is what von Neumann termed a state of irrelevance, since the output no longer holds any relevance to the initial inputs and there is no way to determine whether the computation began with a majority of 1's or 0's.

It is important to note that because the outputs of the two different trees converge asymptotically, they will reach the indistinguishable 1/2 in the limit as the number of iterations goes to infinity. However, the ability to distinguish between these two output values becomes impossible once they get reasonably close to one another. If 0.4 and 0.6 are the thresholds for deciding whether a signal is reliable, for instance, a perfectly reliable input becomes unreliable after five layers for $\varepsilon = 0.05$ and only after two layers for $\varepsilon = 0.25$, as shown in Fig. 5. Thus, in practical applications, only a few iterations can be used before the output becomes unreliable.

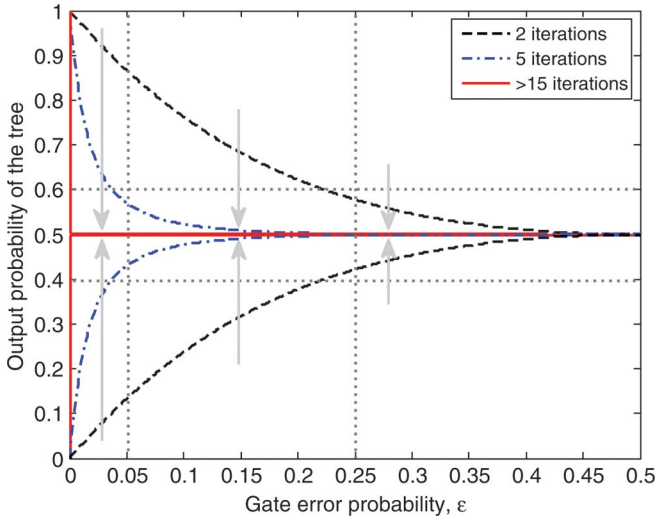


Fig. 5. Output X of the ternary tree of majority gates with minimally biased inputs after 2, 5, and 15 or more iterations versus the gate error rate ε. The arrows indicate the asymptotic convergence of X into the indistinguishable 1/2 for any $\varepsilon > 0$.

C. Simplified Model and the Convergence Rate

Given independent and identical inputs, i.e., each input independently has the same probability to be "1" or "0," the analysis can be simplified by condensing the model into a single equation. By using the values $p_{(i)}$ and $1 - p_{(i)}$ to represent the input probabilities of $X_{1(i)}$ and $X_{0(i)}$, respectively, in (11), we propose the following equation to describe the majority gate when its inputs are (X_1, X_1, and X_0) as well as when its inputs are (X_0, X_0, and X_1):

$$p_{(i+1)} = \varepsilon + (1-2\varepsilon)f(p_{(i)}, 1-p_{(i)}). \tag{13}$$

Theorem I : Given independent and identical initial inputs that are minimally biased, iteration of (13) correctly models the ternary tree of majority gates described by the simultaneous iterations of (11) and (12).

Proof: We first consider the output of a single iteration of equation (13) for the following two cases:

1) when inputs are initially (X_1, X_1, and X_0) with probability p_1 and p_0 for X_1 and X_0;
2) when inputs are initially (X_0, X_0, and X_1) with probability p_0 and p_1 for X_0 and X_1.

By definition, we have $p_0 = 1 - p_1$. In case 1), (13) and (11) simplify to the same expression; therefore, (13) is equivalent to (11) for the first iteration. In case 2), (13) and (12) simplify to the same expression; therefore, (13) is also equivalent to (12) for the first iteration.

To simplify this discussion, $p^*_{X(n)}$ will be used to denote the output of (13) after n iterations with the expected output X where $X \in \{0, 1\}$. From the aforementioned discussion, we see that $p^*_{0(1)} = p_{0(1)}$ and $p^*_{1(1)} = p_{1(1)}$, which mean (13) can be used to find the output of gates at the first layer. Further, it can be shown that $p_{0(1)} = 1 - p_{1(1)}$, which gives us $p^*_{0(1)} = 1 - p^*_{1(1)}$.

Now, we use induction to complete the proof. This is equivalent to showing that $p^*_{0(n+1)} = 1 - p^*_{1(n+1)}$ given that

$p^*_{0(n)} = p_{0(n)}, p^*_{1(n)} = p_{1(n)}$ and, $p^*_{0(n)} = 1 - p^*_{1(n)}$. First, we apply $1 - p^*_{1(n)}$ and $p^*_{1(n)}$ to the inputs of (13) and obtain the following:

$$p^*_{0(n+1)} = \varepsilon + (1-2\varepsilon) f(1 - p^*_{1(n)}, p^*_{1(n)}) \tag{14}$$

$$p^*_{1(n+1)} = \varepsilon + (1-2\varepsilon) f(p^*_{1(n)}, 1 - p^*_{1(n)}). \tag{15}$$

Simplifying the equations, we see that $p^*_{0(n+1)} = 1 - p^*_{1(n+1)}$, which completes the proof. □

With this simplified model, it is shown next that in an infinitely large ternary tree, the output probability of the tree indeed reaches 1/2 when either the gate error rate or the initial input error rate is not zero.

Theorem II: For a ternary tree described by (13), $\forall \delta > 0$, there exists i, which denotes the number of layers, such that $|p_{(i)} - 1/2| < \delta$, under one of the two following conditions:

1) $0 < \varepsilon <$1/2;
2) $\varepsilon = 0$ and $0 < |p_{(0)} - 1/2| <$ 1/2, where $p_{(0)}$ is the initial input probability at the first layer.

Proof: By (13), we have

$$\begin{aligned}
&\left| p_{(i)} - \frac{1}{2} \right| \\
&= \left| \varepsilon + (1-2\varepsilon)(2p_{(i-1)} - 3p^2_{(i-1)} + 2p^3_{(i-1)}) - \frac{1}{2} \right| \\
&= \left(\frac{1}{2} - \varepsilon \right) \cdot |1 - 4p_{(i-1)} + 6p^2_{(i-1)} - 4p^3_{(i-1)}| \\
&= (1-2\varepsilon) |p_{(i-1)} - \frac{1}{2}| (2p^2_{(i-1)} - 2p_{(i-1)} + 1).
\end{aligned} \tag{16}$$

Let $c_{(i)} = 2p^2_{(i)} - 2p_{(i)} + 1$. It can be shown that $1/2 \leq c_{(i)} \leq 1$ for any $p_{(i)} \in [0,1]$, where $c_{(i)} = 1$ when $p_{(i)} = 0$ or 1.

Hence,

$$\left| p_{(i)} - \frac{1}{2} \right| \leq (1-2\varepsilon) \left| p_{(i-1)} - \frac{1}{2} \right|. \tag{17}$$

By the recursive nature of (17), we obtain

$$\left| p_{(i)} - \frac{1}{2} \right| \leq (1-2\varepsilon)^i \left| p_{(0)} - \frac{1}{2} \right|. \tag{18}$$

To prove $|p_{(i)} - 1/2| < \delta$, it is sufficient to have

$$(1-2\varepsilon)^i \left| p_{(0)} - \frac{1}{2} \right| < \delta. \tag{19}$$

For $0 < \varepsilon <$ 1/2, (19) holds when $i > \log_{(1-2\varepsilon)} \left(\delta / (|p_{(0)} - 1/2|) \right)$, which proves the theorem under condition 1).

When $\varepsilon = 0$, (16) becomes

$$\left| p_{(i)} - \frac{1}{2} \right| = \left| p_{(i-1)} - \frac{1}{2} \right| c_{(i-1)}. \tag{20}$$

For $0 < |p_{(0)} - 1/2| <$ 1/2, we have $1/2 < c_{(0)} < 1$. It can then be obtained from (20) that $0 < |p_{(i)} - 1/2| <$ 1/2 for any $i \geq 1$. Therefore, we obtain 1/2$< c_{(i)} < 1$, for any $i \geq 1$.

Further, let $c_{\max} = \max_j (c_{(j)})$, where $j = 0,\ 1,\ 2,\ \ldots, i-1$. It can be shown that $c_{\max} = 2p^2_{(0)} - 2p_{(0)} + 1$.[1] (20) leads to

$$\left| p_{(i)} - \frac{1}{2} \right| \leq c^i_{\max} \cdot \left| p_{(0)} - \frac{1}{2} \right|. \tag{21}$$

Similarly, to prove $|p_{(i)} - 1/2| < \delta$, it is sufficient to have

$$c^i_{\max} \cdot |p_{(0)} - \frac{1}{2}| < \delta. \tag{22}$$

When $i > \log_{c_{\max}} (\delta / (|p_{(0)} - 1/2|))$, (22) holds, which proves the theorem under condition 2). □

Theorem II states that in a ternary tree with the minimally biased inputs as described earlier, any nonzero error rate will not be tolerated in a long computation. Even with the constituent majority gates being perfectly reliable, any disturbance to the initial inputs will result in an unreliable output in a large tree.

From the proof of Theorem II, the convergence rate of the signal probability to the indistinguishable 1/2 can be derived as follows. From (16), we obtain

$$\frac{|p_{(i)} - 1/2|}{|p_{(i-1)} - 1/2|} = (1-2\varepsilon) c_{(i-1)}, \tag{23}$$

which is bounded by $(1-2\varepsilon)$, i.e.,

$$\frac{|p_{(i)} - 1/2|}{|p_{(i-1)} - 1/2|} \leq (1-2\varepsilon) \tag{24}$$

or

$$\frac{|p_{(i)} - 1/2|}{|p_{(0)} - 1/2|} \leq (1-2\varepsilon)^i. \tag{25}$$

For $0 < \varepsilon <$ 1/2, the convergence rate is the largest at $1-2\varepsilon$ when $p_{(i-1)} = 1$ or 0, i.e., when the inputs are perfectly reliable. It approaches 1/2$-\ \varepsilon$ when $p_{(i-1)}$ is close to 1/2.

For $\varepsilon = 0$, the convergence rate is given by

$$\frac{|p_{(i)} - 1/2|}{|p_{(i-1)} - 1/2|} = c_{(i-1)} \tag{26}$$

where $c_{(i-1)}$ is dependent on $p_{(i-1)}$, the input probability, and $1/2 < c_{(i-1)} < 1$, for any $p_{(i-1)} \in (0,1)$. When $p_{(i-1)}$ is close to 1 or 0, the convergence rate is nearly 1, and when $p_{(i-1)}$ moves toward 1/2, it approaches 1/2.

To summarize, the convergence rate is highly dependent on the gate error rate ε when ε is not trivial, while it is dominated by the input error rate when ε is small or negligible.

D. Correlated Inputs

Theorem II applies when the inputs to a tree are independently distributed. However, correlations in inputs can be accounted for when the inputs are minimally biased. When correlated, the inputs share joint distributions. In an infinitely large ternary tree, it is shown in the following corollary that the output probability of the tree reaches 1/2 when either the gate error rate or the initial input error rate is not zero.

[1]This is due to an anonymous reviewer.

Corollary I: In a ternary tree with the minimally biased inputs, which may be correlated, the signal probability converges to 1/2, under one of the two conditions specified in Theorem II.

Proof: In a tree with correlated inputs, its output probability p_c is given by the marginalization of the conditional signal probabilities over all input distributions, i.e., $p_c = \sum_{k\in W} p_k p_{\text{in},k}$, where $p_{\text{in},k} = \text{prob}(\text{inputs} = k)$, the probability that the input vector has the kth value, p_k is the conditional probability of the output signal for the input vector k, and W is the set of all input vectors in the minimally biased case. As per Theorem II, we have $\lim_{i\to\infty} p_k = 1/2$, where i denotes the number of layers, under one of the two specified conditions. Then, we obtain $\lim_{i\to\infty} p_c = 1/2 \cdot (\sum_{k\in W} p_{\text{in},k}) = 1/2$, which proves the corollary. □

The implication of Corollary I extends that of Theorem II to majority circuits where inputs are correlated, as will be shown by the QCA-adder example in Section VI.

IV. Tree Structures of N-input Majority Gates With Minimally Biased Inputs

In general, minimally biased inputs occur when the majority gate has a minimal number of input values in consensus to produce the expected output. For an N-input majority gate ($N = 2K + 1$), this minimum is $K + 1$; therefore, the minimal bias occurs when $K + 1$ inputs are equal to the expected output and K inputs equal the complement of the expected output. An N-array tree of gates similar to that shown in Fig. 4 can be constructed for this case. By using combinatorial arguments, iterative expressions are derived for an N-input majority gate, as

$$p_{1(i+1)} = \varepsilon + (1-2\varepsilon) f_N(p_{1(i)}, p_{0(i)}) \tag{27}$$

$$p_{0(i+1)} = \varepsilon + (1-2\varepsilon) f_N(p_{0(i)}, p_{1(i)}) \tag{28}$$

where

$$f_N(a,b) = \sum_{i=0}^{K} \binom{K}{i} b^i (1-b)^{K-i} \cdot \left(\sum_{j=K+1-i}^{K+1} \binom{K+1}{j} a^j (1-a)^{K+1-j} \right). \tag{29}$$

Applications of (27) and (28) on five- and seven-input majority gates produce similar diagrams to that in Fig. 5, as shown in Fig. 6. Fig. 6 shows the outputs of three-, five-, and seven-input majority gates after two and six iterations of (27) and (28) for perfectly reliable initial inputs. It can be seen that after two iterations, the three-input majority gate has the best reliability, while the seven-input majority gate has the worst. After six iterations, the seven-input majority gate has nearly reached 1/2, while the three- and five-input majority gates have not. Clearly, the three-input majority gate converges to the indistinguishable 1/2 at a slower rate than either the five- or seven-input majority gate, and the five-input gate converges more slowly than the seven-input gate. In fact, as the number of inputs to a majority gate increases, its convergence rate also increases. This is stated in the following theorem.

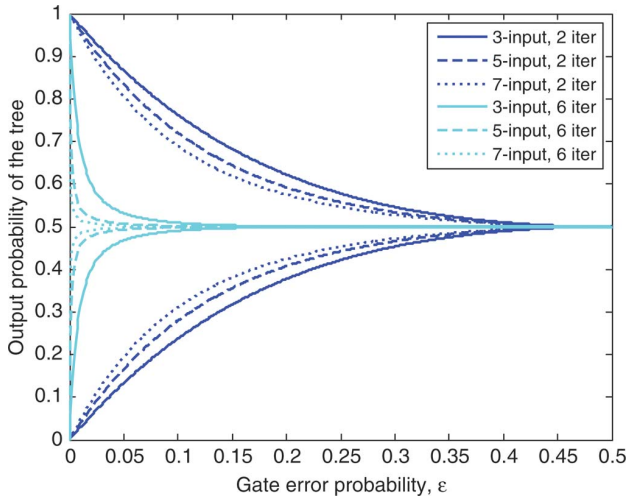


Fig. 6. Outputs of three-, five-, and seven-input majority trees after two and six iterations of (27) and (28).

Theorem III: In the tree structure as defined earlier, given independent and identical inputs that are minimally biased, the signal probability converges faster to an indistinguishable state for majority gates with a larger number of inputs, under one of the two following conditions:

1) $0 < \varepsilon < 1/2$;
2) $\varepsilon = 0$ and $0 < |p_{(0)} - 1/2| < 1/2$, where $p_{(0)}$ is the initial input probability at the first layer.

To prove Theorem III, we introduce the following two Lemmas.

Lemma I states that the output of a majority gate is bounded by 1/2 and whether it is above or below 1/2 depends on the initial probability of the inputs that are in the minimal majority.

Lemma I: Assume that a fault-free majority gate has $2K + 1$ independent and identical inputs $(K \geq 1)$, of which a minimum majority, $K + 1$, are the same as the expected output with probability p and the remaining K inputs are the same as the complement of the expected output with the same probability p. Let $p_{m,1}^{(2K+1)}$ be the expected output probability of the majority gate, it holds that for any $K \geq 1$, $1/2 < p_{m,1}^{(2K+1)} \leq 1$ when $1/2 < p \leq 1$ and $0 \leq p_{m,1}^{(2K+1)} < 1/2$, when $0 \leq p < 1/2$.

Proof of Lemma I is given in Appendix A.

Lemma II states that given independent, identical, and imperfect inputs, which have a minimal majority to produce an expected output, the output of a majority gate with $2K + 1$ inputs is more reliable than that of a majority gate with $2K + 3$ inputs.

Lemma II: For two majority gates with $2K + 1$ and $2K + 3$ inputs $(K \geq 1)$, respectively, given independent and identical inputs that are minimally biased, the output probabilities of the majority gates, $p_m^{(2K+1)}$ and $p_m^{(2K+3)}$, satisfy $|p_m^{(2K+1)} - 1/2| > |p_m^{(2K+3)} - 1/2|$ for any $0 \leq \varepsilon < 1/2$ and $0 < |p - 1/2| < 1/2$, where p is the input probability.

Proof of Lemma II is given in Appendix B.

Note that Lemma II does not hold when $p = 0$ or $p = 1$, where an equality sign would be needed to validate the Lemma.

Now the proof of Theorem III is given.

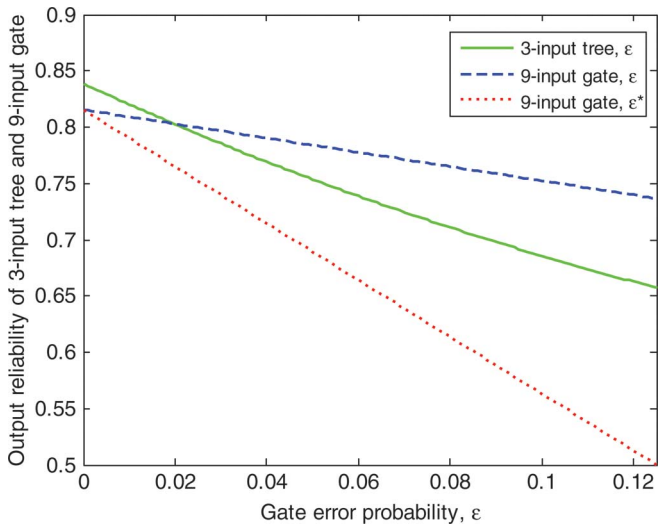


Fig. 7. Output reliability of a two-layer three-input majority tree and a nine-input majority gate with an error rate of ε and $\varepsilon^* = 4\varepsilon$.

Proof of Theorem III: As shown in Theorem II and its proof, for a ternary tree of three-input majority gates, its output monotonically converges to the indistinguishable 1/2 in a long computation, under one of the two given conditions. Assuming that this holds for majority gates with $2K+1$ inputs ($K \geq 1$), we show that it also holds for majority gates with $2K+3$ inputs. Since the signal probability in a tree structure is given by the output probability of the majority gates in the tree, the trees of gates with $2K+1$ and $2K+3$ inputs can be modeled by $p_m^{(2k+1)}$ and $p_m^{(2K+3)}$, respectively. As per Lemma II, we have $|p_m^{(2K+3)} - 1/2| < |p_m^{(2K+1)} - 1/2|$, for $0 \leq \varepsilon < 1/2$ and $0 < |p_{(0)} - 1/2| < 1/2$. For the special case when $0 < \varepsilon < 1/2$ and $p_{(0)} = 0$ or 1, we have $0 < |p_{(1)} - 1/2| < 1/2$ at the next layer, where $p_{(1)}$ is the output probability at the first layer and thus the input probability for the next layer. Applying Lemma II on the gates of the second layer gives us $|p_m^{(2K+3)} - 1/2| < |p_m^{(2K+1)} - 1/2|$. Thus, as per Theorem II, the signal probability in the $2K+3$ majority tree converges faster to 1/2 than that in the $2K+1$ majority tree. By induction, the signal probability converges faster to an indistinguishable state for majority gates with a larger number of inputs, under any one of the given conditions. □

Albeit with larger convergence rates, majority gates with more inputs can be useful because of their increased functionality. For example, a single nine-input majority gate can be used for a two-layer ternary tree of four three-input gates. This raises the question of whether a nine-input gate or a three-input tree is more reliable than the other. In an actual implementation, a nine-input gate can be (almost) as simple as a single three-input gate or as complex as four three-input gates. We consider both cases, i.e., when the nine-input gate is subject to the same error rate as a single three-input gate ε or to the same error rate as four three-input gates, approximately 4ε. The comparison results are shown in Fig. 7 for an input reliability of 0.95.

When the gates are highly reliable (with $\varepsilon < 0.02$ for an input of 0.95 in this case), as revealed in Fig. 7, the nine-input majority gate provides a less reliable output, while it offers a better result than the three-input tree as ε goes up. This is consistent with our earlier discussions that the convergence rate is more dependent on ε when ε increases. The most unreliable output results from a complex nine-input majority gate for any ε, as also shown in Fig. 7. This suggests that the use of majority gates with the minimum number of inputs placed serially may be a better solution than using their larger counterparts when they have similar complexities.

Note that the inputs here are considered minimally biased. While dominant inputs are more often seen in a fault-tolerant design (such as in TMR/NMR), a minimally biased or similar input scenario could arise in a real (hierarchical) voting or election system.

V. Tree Structures of Minority Gates With Minimally Biased Inputs

The analysis of the majority gates can be readily extended to minority gates. Since the minority gate performs the inverse majority function, the output probability of a fault-free minority gate is simply the complement of the output probability of a fault-free majority gate. This gives us the following iterative equations for minority gates:

$$p'_{1(i+1)} = (1-\varepsilon) - (1-2\varepsilon) \cdot f_N(p'_{0(i)}, p'_{1(i)}) \quad (30)$$

$$p'_{0(i+1)} = (1-\varepsilon) - (1-2\varepsilon) \cdot f_N(p'_{1(i)}, p'_{0(i)}). \quad (31)$$

A similar tree model ensures that Theorem I also applies to the case of minority gates, but using the following modified equation:

$$p'_{(i+1)} = (1-\varepsilon) - (1-2\varepsilon) \cdot f_N(1-p'_{(i)}, p'_{(i)}). \quad (32)$$

Iteration of (32) yields a diagram similar to that in Fig. 5. When the inputs to a minority gate are not in consensus, therefore, any nonzero error rate will not be tolerated and any disturbance to the initial inputs will result in an unreliable output in a long computation, even when the constituent minority gates are perfectly reliable.

The conclusions on converging rate of majority gates equally apply to minority gates: as the number of inputs to a minority gate increases, the rate of convergence to an indistinguishable value of 1/2 also increases. This is stated in the following corollary.

Corollary II: In the tree structure as defined earlier, given independent and identical inputs that are minimally biased, the signal probability converges faster to an indistinguishable state for minority gates with a larger number of inputs, under one of the two conditions specified in Theorem III.

The proof follows that of Theorem III, except that Lemma I does not hold any more because the signal probability oscillates between layers as per the function of minority gates (inverted majority).

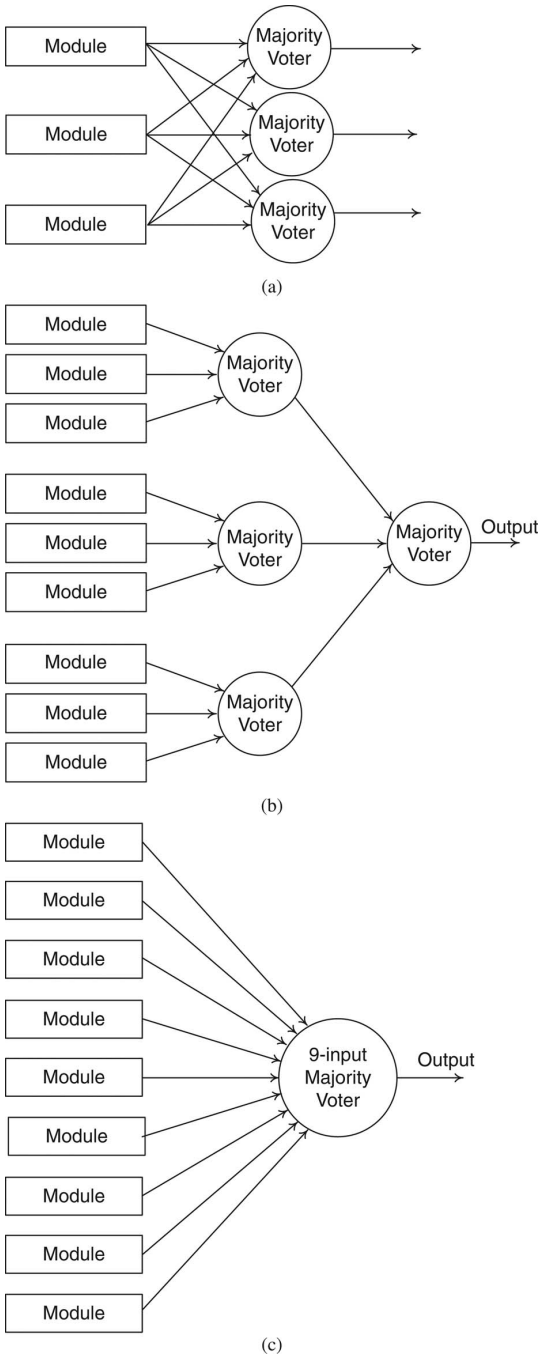


Fig. 8. (a) TMR with triplicated voters (TMR-TV). (b) Two-layer CTMR (2-CTMR). (c) Ninefold NMR (9MR).

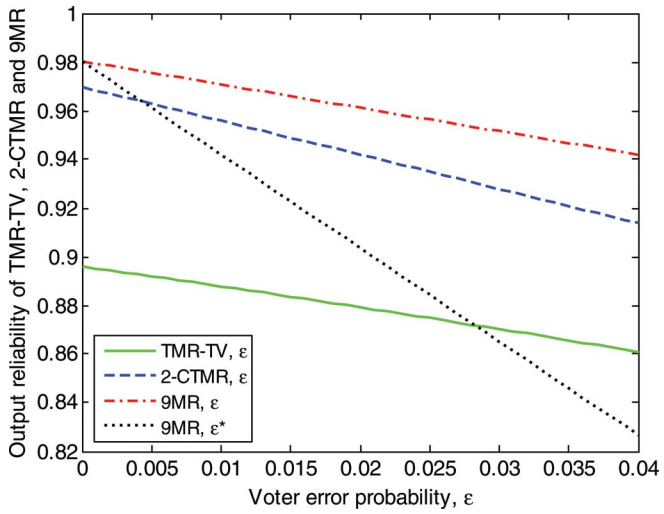


Fig. 9. Output reliability of the TMR with triplicated voters (TMR-TV), two-layer CTMR (2-CTMR), and ninefold NMR (9MR) with an error rate of ε and $\varepsilon^* = 4\varepsilon$ for the majority voters.

VI. Implications in Reliability of Computational Architectures

A. Fault-Tolerant Architectures

The majority logic provides an essential function in many fault-tolerant techniques, including TMR/NMR, cascaded TMR (CTMR), and majority multiplexing [8], [13], [20], [27], [28]. With the emergence of nanoelectronic devices, however, such fault-tolerant techniques have taken on an even greater level of significance due to the increased probability of error in these devices. In basic TMR, a single computational module is replicated three times and the output from each of the three modules is voted on by a majority gate. In practice, the majority voter is often triplicated for providing triple outputs, as shown in Fig. 8(a) (an additional voter is needed at the last layer). CTMR, as shown in Fig. 8(b), is created by combining three of the TMR units with another majority gate to form a second-order TMR unit with even higher reliability. TMR can be extended to general NMR for odd N—a ninefold NMR (9MR) is shown in Fig. 8(c).

The TMR with triplicated voters (TMR-TVs) is also the simplest implementation of the so-called distributed R-fold modular redundancy [29]; it has similar signal routings as the CTMR, except that the signals are from three modules and are thus correlated, whereas the signals in CTMR are from nine independent modules. Also, the two-layer CTMR (2-CTMR) uses similar resources as the 9MR. Given the similarities among the three different architectures, an interesting question arises as to which one of the three has the best reliability in the presence of noisy voters. This question is answered by evaluating the reliability of each architecture. The accurate PGM approach [26] is used to account for the signal correlations in the TMR-TV. For the nine-input majority voter, two cases are considered, i.e., when it is subject to the same error rate as 1) that of a three-input voter and 2) that of four three-input voters. The results are shown in Fig. 9 for an input error rate of 0.2.

In general, the TMR-TV results in the least reliability improvement as it uses the least resources, and the 9MR using a simple voter provides the best reliability. For the complex voter design, as shown, the 9MR offers better reliability than the two-layer CTMR when the voters are highly reliable (with $\varepsilon < 0.004$ and thus $\varepsilon^* < 0.016$ for an input of 0.8 in this case), while it produces a worse result as ε^* goes up. It could become even worse than the TMR-TV for a large ε^*. (This is in contrast to the case with minimally biased inputs, where the complex nine-input majority gate always provides an inferior result, while a simple implementation could also result in a worse output than the two-layer three-input majority tree.) This suggests that it

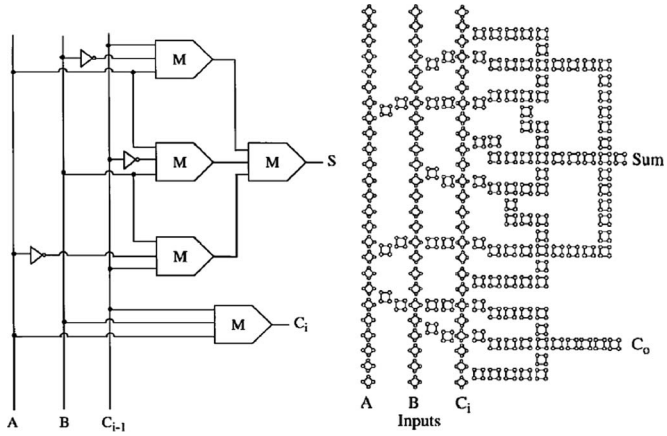


Fig. 10. Schematic circuit diagram and physical layout of a QCA full-adder [2].

TABLE I
TRUTH TABLE OF THE ADDER, INCLUDING THE MAPPINGS OF THE INPUTS TO THE OUTPUTS OF THE THREE INTERMEDIATE MAJORITY GATES

Inputs (ABC)	Majority Outputs	Sum (S)	Carry (C)
000	000	0	0
001	101	1	0
010	011	1	0
011	001	0	1
100	110	1	0
101	100	0	1
110	010	0	1
111	111	1	1

may be advantageous to use NMR than its equivalent CTMR when the voters have similar complexities, but not when the NMR voter is significantly more complex and thus subject to a much higher error rate.

Note that the reliability is improved by all the three architectures for a relatively small voter error rate $\varepsilon < 0.04$, as illustrated in Fig. 9. In the following, however, it is shown that the reliability can actually decline in a TMR structure.

B. Nanoscale Logic and Its TMR

In addition to utilizing fault-tolerant techniques based on majority gates to improve reliability, many emerging nanoelectronic devices, such as QCA, also depend on the majority or minority logic function to perform computation. A common full adder, shown in Fig. 10, requires two layers of majority gates to compute the Sum [2], [30]. This design closely resembles the two-layer ternary tree of majority gates used in the CTMR structure.

While in CTMR, the modules are expected to produce identical outputs; the input signals to the majority gates in the adder are provided by three independent inputs (A, B, and C). As a result, a minimally biased input scenario is much more likely to appear in the majority adder than in the CTMR. For the adder of Fig. 10, for example, the initial inputs of (0,0,1) would yield (1,0,1) at the output of the three intermediate majority gates. As

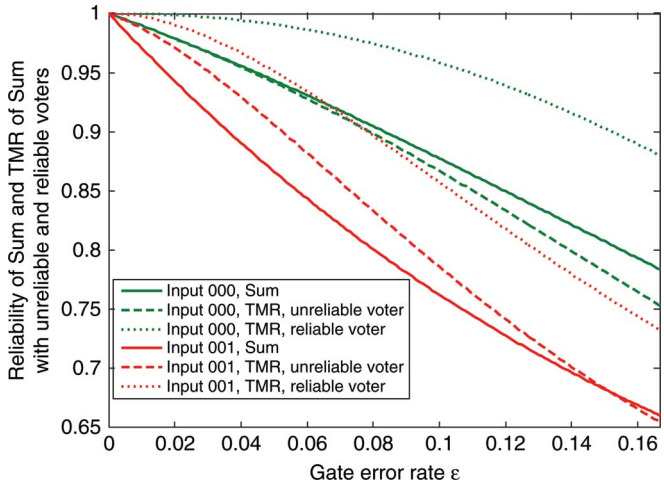


Fig. 11. Reliability of Sum versus gate error rate ε, for the adder of Fig. 10 and its TMR implementation. While the reliability of the TMR of the adder with an unreliable voter is better than that of the single adder for most inputs, it is worse than that of the single adder for the input (0,0,0) (and thus (1,1,1), by symmetry). The error rate ε is shown up to the error threshold 1/6.

shown in Table I, six out of the eight input combinations (75%) result in similar scenarios as the minimally biased inputs do in a tree of majority gates.

Although the signals are correlated in the adder, they are affected by the defects and faults in QCA devices [31]; therefore, their reliability is expected to decrease, as implied by Corollary I. The reliability of Sum is evaluated for two representative inputs: (0,0,0) and (0,0,1). The PGM approach is used to capture the signal correlations caused by reconvergent fanouts in the circuits [26]. As shown in Fig. 11, the input (0,0,1) results in a much lower reliability than that resulting from (0,0,0), in accordance with the earlier discussion. Although not shown in the figure, an imperfect input causes a drop in the reliability of Sum, even when $\varepsilon = 0$. This is an important characteristic to note as implied by Theorem II under condition 2). In contrast to CTMR, a large majority of the inputs here (75%, for uniformly distributed inputs) result in inferior reliability of the full adder. When the underlying devices and/or their inputs are not very reliable, therefore, the full adder design of Fig. 10 renders itself as an unreliable structure. A similar result was also obtained in a comparison study of several adder designs in [16].

Note that errors in wires, crossovers, and inverters are not considered in this analysis. The probabilistic modeling in [24] reveals that the reliability of QCA circuits is also influenced by the physical implementation of the majority function and by different input vectors. The simulation results obtained by the logic analysis in this section can further be enhanced by incorporating various factors, such as the noise in wires and the physical implementations of gates into analysis.

To improve the reliability of the full adder, it is possible to apply fault-tolerant techniques such as the TMR. Indeed, TMR has been considered for applications in QCA architectures [21]. Due to the use of the same unreliable majority gates as voters, however, TMR may not be effective for improving the reliability of the full adder of Fig. 10. The simulation results, shown in Fig. 11, indicate that the reliability of TMR using an unreliable

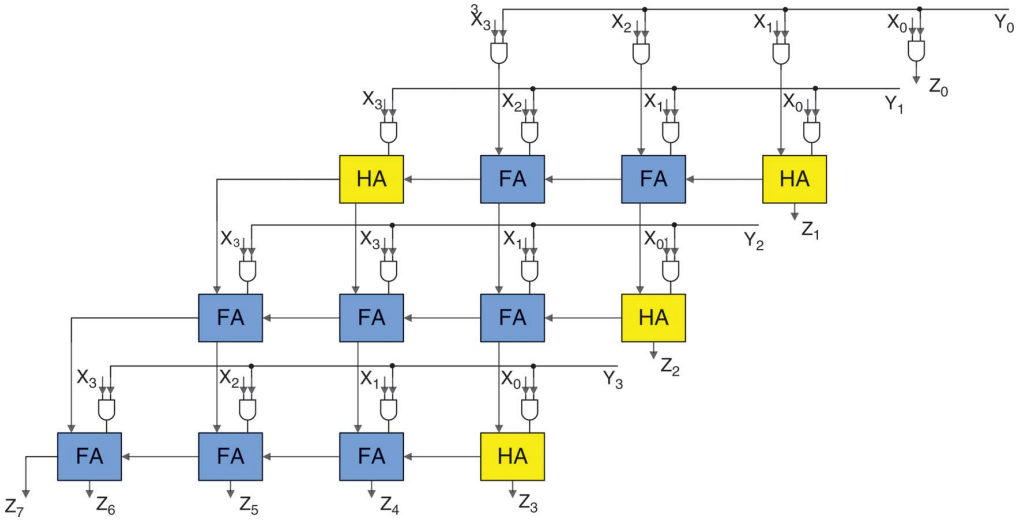


Fig. 12. Array multiplier of 4×4 consisting of full adders (FA), half adders (HA), and AND gates.

voter is even worse than that of the single adder for the input (0,0,0) (and thus (1,1,1), by symmetry). This is due to the fact that the error rate of the majority voter is too large to improve the relatively high reliability of the Sum of the single adder (as implied in Fig. 2). In contrast, a perfectly reliable voter would improve the reliability. Although TMR works for the other inputs and, therefore, improves the overall reliability of the adder, it is important to carefully evaluate the efficacy of TMR for certain nanoscale applications, since high defect and error rates are a prevalent feature of most nanodevices.

C. Complex Circuits

The full adder of Fig. 10 is a fundamental unit used in many high-level designs of functional blocks such as multipliers. Fig. 12 shows a 4×4 multiplier composed of an array of adders and AND gates [32]. In QCA technology, the adders, as well as the AND gates, will be implemented with majority logic. The array multiplier is scalable to higher orders of 8×8 and 16×16 multipliers, each of which, in fact, is a complex tree of majority gates. Let us take the most significant bit Z_7 in Fig. 12 as an example. The inputs to the adder resulting in the carry bit Z_7 are the Carry outputs of the neighboring adders and the output of an AND gate. The inputs to the neighboring adders can further be traced back to the Carry and/or Sum of their own neighbors. Traversing from Z_7 through the multiplier array results in nonideal treelike signal paths.

Since the inputs of the multiplier can have arbitrary values, certain inputs result in similar scenarios as those in a ternary tree by the minimally biased inputs. To find out the worst case reliability, we evaluate the reliability of the multiplier, in the form of its most significant bit, for different inputs. The modular PGM approach is used as it is able to provide highly accurate results while maintaining a moderate complexity for large circuits [26]. The results are shown in Fig. 13 for the three different multipliers of sizes 4×4, 8×8, and 16×16 bits.

In the simulation, all inputs are considered for 4×4 and 8×8 multipliers, while one million input vectors are used for the 16×16 multiplier. It is interesting to note that the three multipliers have very similar reliabilities in the best cases, as shown by the overlapping curves in Fig. 13. Due to the error correction capability of the majority gate, the reliability of the most significant bit stays high for the best case inputs—it is even better than 0.85 for a gate error rate of 0.1. In the worst cases, however, there are significant differences and the reliability starts to decrease drastically after a threshold. For the 4×4, 8×8, and 16×16 bit multipliers, this threshold is approximately 10^{-4}, 10^{-5}, and 10^{-6}, respectively. To have a circuit reliability better than 0.95, for instance, the gate error rate will have to be smaller than approximately 10^{-3}, 10^{-4}, and 10^{-5}, respectively, for the three multipliers. In the best cases, however, the gate error rate only needs to be on the order of 10^{-2}, which indicates a difference of up to three orders of magnitudes in the requirement of gate reliability. This effect of input vectors on the reliability of circuits has also been discussed in [33], [34] and large differences

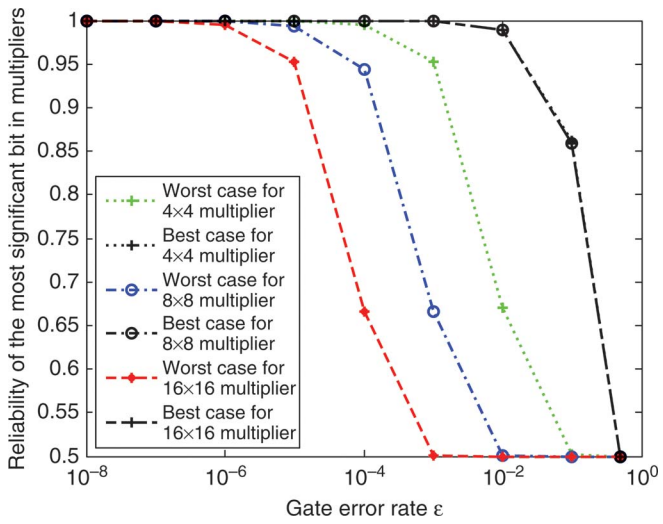
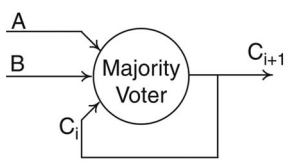


Fig. 13. Worst case and best case reliability of the most significant bit versus gate error rate ε, for the array multiplier of three different sizes. The best case reliabilities are very similar as shown by their overlapping curves, while the worst case reliabilities are very different for the three multipliers.



(a)

A	B	C_{i+1}
0	0	0
0	1	C_i
1	0	C_i
1	1	1

(b)

Fig. 14. C-element. (a) Schematic diagram. (b) Truth table.

in the resulting reliability have been observed. The result also indicates that highly reliable nanodevices are required to accommodate the worst case reliability scenario in a nanocircuit design, or significant reliability constraints are imposed on the correct functioning of the circuit.

D. Circuits With Feedbacks

Finally, we consider a majority-based circuit with feedbacks, namely, the Muller C-element [35]. Fig. 14 shows a schematic and the truth table of the Muller C-element. The output of the C-element switches to its input value whenever the two inputs agree; it remains at its previous value otherwise. As a common logic gate used in asynchronous circuits, the Muller C-element has been proposed for applications in fault-tolerant [36], [37], as well as in biological synthetic [38] circuit design.

When the two inputs are not in consensus, the feedback loop of the C-element provides a third input that is critical in deciding the output. Since the circuit is asynchronous (with typically a small propagation delay), the temporal operation of the C-element can be unfolded into a cascaded chain of majority gates with unequal nominal inputs. As per our previous discussion, the output reliability of the majority gate degrades in such a chain due to the accumulation of errors in the third input. Although the other two inputs can be reliable, it can be shown that the output of the C-element will reach a steady and unreliable state due to the Markov nature of the chain. This implies that the majority-based C-element may not be a reliable design for certain circuit applications. Note, however, that the error in the feedback loop can be masked by the two primary inputs when they are equal. Thus, this should be the primary application domain of the C-element in fault-tolerant designs.

VII. Conclusion

Majority and minority logics have become increasingly important in circuit design for two reasons. First, many nanoelectronic computing systems rely heavily on the majority or minority function to perform computation, and second, their low device reliability will require fault-tolerant techniques, most of which employ some form of majority voting. This paper addresses the reliability issues of majority-/minority-based computational structures. In a tree structure with the so-called minimally biased inputs, any nonzero error rate of gates and/or of its initial inputs will not be tolerated in a long computation. As the number of inputs to a majority/minority gate increases, the rate at which the gate output converges to an indistinguishable value also increases.

Although these results are obtained for tree structures, their implications to circuit design are explored by several case studies of fault-tolerant and nanoelectronic architectures, including TMR/CTMR/NMR, QCA, and feedback circuits. It is shown that inferior circuit reliability arises due to the existence of certain inputs, which are, to many circuits, the equivalent of what the minimally biased inputs are to a tree structure. When building fault-tolerant and nanoscale architectures, therefore, it is important to consider the possible reliability constraints imposed by the worst case reliability of a computational structure using majority/minority logic. The implications of these results to novel circuit design, such as those using threshold logic [39], multiplexed logic [13], [28], [40], and averaging cells [41], await further investigation.

Appendix A

Proof of Lemma I: Let $p_{m,0}^{(2K+1)}$ be the probability that the output is the complement of the expected value; we have

$$p_{m,1}^{(2K+1)} + p_{m,0}^{(2K+1)} = 1. \tag{A1}$$

Since there are $K+1$ inputs with probability p and K inputs with probability $1-p$ being the same as the expected output, by taking out one input with probability p, we have exactly K inputs with both probabilities p and $1-p$ in the remaining $2K$ inputs, as well as the following:

$$p_{m,1}^{(2K+1)} = p \cdot p_{m,1}^{(2K)}(\geq K) + (1-p) \cdot p_{m,1}^{(2K)}(\geq K+1) \tag{A2}$$

$$p_{m,0}^{(2K+1)} = (1-p) \cdot p_{m,0}^{(2K)}(\geq K) + p \cdot p_{m,0}^{(2K)}(\geq K+1) \tag{A3}$$

where $p_{m,1}^{(2K)}(\geq K)$ and $p_{m,1}^{(2K)}(\geq K+1)$ are, respectively, the probabilities that no less than K and no less than $K+1$ inputs have the expected output value of a $2K$ input majority gate; $p_{m,0}^{(2K)}(\geq K)$ and $p_{m,0}^{(2K)}(\geq K+1)$ are, respectively, the probabilities that no less than K and no less than $K+1$ inputs have the complement of the expected output value of a $2K$ input majority gate. Due to the symmetry in the probabilistic distribution of the remaining $2K$ inputs, it is easy to see that $p_{m,1}^{(2K)}(\geq K) = p_{m,0}^{(2K)}(\geq K)$, and $p_{m,1}^{(2K)}(\geq K+1) = p_{m,0}^{(2K)}(\geq K+1)$. Further from (A2) and (A3), we obtain

$$p_{m,1}^{(2K+1)} - p_{m,0}^{(2K+1)} = (2p-1)\cdot \left(p_{m,1}^{(2K)}(\geq K) - p_{m,1}^{(2K)}(\geq K+1)\right). \tag{A4}$$

As $p_{m,1}^{(2K)}(\geq K) > p_{m,1}^{(2K)}(\geq K+1)$, we have (A4) > 0 for $1/2 < p \leq 1$ and (A4) < 0 for $0 \leq p < 1/2$. Combining the above with (A1) proves the lemma. □

Appendix B

Proof of Lemma II: According to (1), the output probabilities of the two majority gates are given by

$$p_m^{(2K+1)} = \varepsilon + (1-2\varepsilon)p_{m,1}^{(2K+1)} \tag{B1}$$

$$p_m^{(2K+3)} = \varepsilon + (1-2\varepsilon)p_{m,1}^{(2K+3)} \tag{B2}$$

where $p_{m,1}^{(2K+1)}$ and $p_{m,1}^{(2K+3)}$ are the fault-free output probabilities of the majority gates. For the second majority gate, taking two inputs, one with probability p and the other $1-p$, out of the $2K+3$ inputs, gives us

$$\begin{aligned} & p_{m,1}^{(2K+3)} \\ &= p^2 \cdot p_{m,1}^{(2K+1)}(\geq K+1) + (1-p)^2 \cdot p_{m,1}^{(2K+1)}(\geq K+1) \\ &\quad + p(1-p) \cdot p_{m,1}^{(2K+1)}(\geq K) \\ &\quad + p(1-p) \cdot p_{m,1}^{(2K+1)}(\geq K+2) \\ &= (1-2p+2p^2) \cdot p_{m,1}^{(2K+1)}(\geq K+1) \\ &+ p(1-p) \cdot \left(p_{m,1}^{(2K+1)}(\geq K) + p_{m,1}^{(2K+1)}(\geq K+2) \right). \end{aligned} \quad \text{(B3)}$$

From (B3), we have

$$\begin{aligned} & p_{m,1}^{(2K+1)} - p_{m,1}^{(2K+3)} \\ &= p(1-p) \cdot (p_{m,1}^{(2K+1)}(\geq K+1) - p_{m,1}^{(2K+1)}(\geq K) \\ &\quad + p_{m,1}^{(2K+1)}(\geq K+1) - p_{m,1}^{(2K+1)}(\geq K+2)) \\ &= p(1-p) \cdot (p_{m,1}^{(2K+1)}(K+1) - p_{m,1}^{(2K+1)}(K)) \end{aligned} \quad \text{(B4)}$$

where $p_{m,1}^{(2K+1)}(K+1)$ and $p_{m,1}^{(2K+1)}(K)$ are the probabilities that exactly $K+1$ and K inputs have the same value as the expected output for a majority gate with $2K+1$ inputs, of which a minimum majority $K+1$, are the same as the expected output with probability p. It is easy to see that $p_{m,1}^{(2K+1)}(K+1) > p_{m,1}^{(2K+1)}(K)$ when $1/2 < p \leq 1$ and $p_{m,1}^{(2K+1)}(K+1) < p_{m,1}^{(2K+1)}(K)$ when $0 \leq p < 1/2$. Further from (B4), (B1), and (B2), we obtain that $p_m^{(2K+1)} \geq p_m^{(2K+3)}$ when $1/2 < p \leq 1$ and $p_m^{(2K+1)} \leq p_m^{(2K+3)}$ when $0 \leq p < 1/2$.

By Lemma I and (B1), we obtain that $1/2 < p_m^{(2K+1)} \leq 1$ when $1/2 < p \leq 1$, and $0 \leq p_m^{(2K+1)} < 1/2$ when $0 \leq p < 1/2$. The same holds for $p_m^{(2K+3)}$ given by (B2). Combining the above proves the lemma. □

ACKNOWLEDGMENT

The authors acknowledge the insightful comments provided by the reviewers, which improved the quality of the paper. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or BellSouth Foundation.

REFERENCES

[1] (2009). International Technology Roadmap for Semiconductors. [online]. Available: http://public.itrs.net/.

[2] C. S. Lent and P. D. Tougaw, "A device architecture for computing with quantum dots," *Proc. IEEE*, vol. 85, no. 4, pp. 541–557, Apr. 1997.

[3] A. Imre, G. Csaba, L. Ji, A. Orlov, G. H. Bernstein, and W. Porod, "Majority logic gate for magnetic quantum-dot cellular automata," *Science*, vol. 311, pp. 205–208, Jan. 2006.

[4] K. K. Likharev, "Single-electron devices and their applications," *Proc. IEEE*, vol. 87, no. 4, pp. 606–632, Apr. 1999.

[5] T. Oya, T. Asai, T. Fukui, and Y. Amemiya, "A majority-logic device using an irreversible single-electron box," *IEEE Trans. Nanotechnol.*, vol. 2, no. 1, pp. 15–22, Mar. 2003.

[6] G. I. Bourianoff, D. E. Nikonov, and J. F. Zheng, "Spin-orbital quantum cellular automata logic devices and systems," U.S. Patent 7 212 026, 2007.

[7] A. Khitun, M. Bao, and K. L. Wang, "Spin wave magnetic NanoFabric: A new approach to spin-based logic circuitry," *IEEE Trans. Magn.*, vol. 44, no. 9, pp. 2141–2152, Sep. 2008.

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

[9] B. Hajek and T. Weller, "On the maximum tolerable noise for reliable computation by formulas," in *IEEE Trans. Inf. Theory*, vol. 37, no. 2, Mar, 1991, pp. 388–391.

[10] W. Evans and L. Schulman, "On the maximum tolerable noise of k-input gates for reliable computation by formulas," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 3094–3098, Nov. 2003.

[11] W. Evans and N. Pippenger, "On the maximum tolerable noise for reliable computation by formulas," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 1299–1305, May 1998.

[12] F. Unger, "Noise threshold for universality of two-input gates," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3693–3698, Aug. 2008.

[13] S. Roy and V. Beiu, "Majority multiplexing—Economical redundant fault-tolerant design for nano architectures," *IEEE Trans. Nanotchnol.*, vol. 4, no. 4, pp. 441–451, Jul. 2005.

[14] J. Gao, Y. Qi, and J. Fortes, "Bifurcations and fundamental error bounds for fault-tolerant computations," *IEEE Trans. Nanotechnol.*, vol. 4, no. 4, pp. 395–402, Jul. 2005.

[15] M. H. Sulieman and V. Beiu, "On single-electron technology full adders," *IEEE Trans. Nanotechnol.*, vol. 4, no. 6, pp. 669–680, Nov. 2005.

[16] W. Ibrahim, V. Beiu, and M. H. Sulieman, "On the reliability of majority gates full adders," *IEEE Trans. Nanotechnol.*, vol. 7, no. 1, pp. 56–67, Jan. 2008.

[17] V. Beiu, S. Aunet, J. Nyathi, R. R. Rydberg, III, and W. Ibrahim, "Serial addition: Locally connected architectures," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 54, no. 11, pp. 2564–2579, Nov. 2007.

[18] V. Beiu, W. Ibrahim, and S. Lazarova-Molnar, "What von neumann did not say about multiplexing: Beyond gates failures—The gory details," in *Proc. Intl. Work-Conf. Artif. Neural Nets (IWANN'07)*, San Sebastián, Spain, Jun. 19–22, 2007, Springer LNCS 4507, pp. 487–496.

[19] D. P. Siewiorek and R. S. Swarz, *Reliable computer systems: Design and evaluation*. Nartick, MA: A K Peters, 1998.

[20] M Stanisavljevic, A Schmid, and Y Leblebici, "Optimization of nanoelectronic systems' reliability under massive defect density using cascaded R-fold modular redundancy," *Nanotechnology*, vol. 19, no. 46, pp. 340–348, Nov. 2008.

[21] T. J. Dysart and P. M. Kogge, "System reliabilities when using triple modular redundancy in quantum-dot cellular automata," in *Proc. 23rd IEEE Int. Symp. Defect Fault Tolerance Very Large Scale Integr. (VLSI) Syst. (DFT 2008)*. Boston, MA, Oct. 2008, pp. 72–80.

[22] T. J. Dysart and P. M. Kogge, "Analyzing the inherent reliability of moderately sized magnetic and electrostatic QCA circuits via probabilistic transfer matrices," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 4, pp. 507–516, Apr. 2009.

[23] T. J. Dysart. (2009, Jul.). It's all about the signal routing: Understanding the reliability of QCA circuits and systems. Ph.D. thesis, CS & E Dept., Notre Dame Univ., IN. [Online]. Available: http://www.cse.nd.edu/∼tdysart/Papers/DysartDissertation.pdf

[24] S. Bhanja and S. Sarkar, "Probabilistic modeling of QCA circuits using bayesian networks," *IEEE Trans. Nanotechnol.*, vol. 5, no. 6, pp. 657–670, Nov. 2006.

[25] E. R. Taylor, J. Han, and J. A. B. Fortes, "An investigation into the maximum tolerable error rate of majority gates for reliable computation," presented at the IEEE Int. Workshop Defect Fault Tolerant Nanoscale Architectures (NANOARCH), Boston, MA, Jun. 2006.

[26] J. Han, H. Chen, E. Boykin, and J. Fortes, "Reliability evaluation of logic circuits using probabilistic gate models," *Microelectron. Reliab.*, vol. 51, no. 2, pp. 468–476, Feb. 2011.

[27] D. Bhaduri and S. Shukla, "NANOLAB-a tool for evaluating reliability of defect-tolerant nanoarchitectures," *IEEE Trans. Nanotechnol.*, vol. 4, no. 4, pp. 381–394, Jul. 2005.

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

[29] M. Stanisavljevic, A. Schmid, and Y. Leblebici, "Optimization of nanoelectronic systems reliability under massive defect density using distributed r-fold modular redundancy (DRMR)," in *Proc. 24th IEEE Int. Symp. Defect and Fault Tolerance VLSI Syst.*, 2009, pp. 340–348.
[30] W. Wang, K. Walus, and G. A. Jullien, "Quantum-dot cellular automata adders," *3rd IEEE Conf. Nanotechnol.* 2003, vol. 2, pp. 461–464.
[31] M. B. Tahoori, J. Huang, M. Momenzadeh, and F. Lombardi, "Defects and faults in quantum cellular automata at nanoscale," in *Proc. IEEE VLSI Test Symp.*, 2004, pp. 291–296.
[32] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 2003.
[33] W. Ibrahim and V. Beiu, "How much input vectors affect nano-circuit's reliability estimates," in *Proc. IEEE Intl. Conf. Nanotechnol.*, Genoa, Italy, Jul. 26–30,2009, pp. 699–702.
[34] V. Beiu and W. Ibrahim, "On CMOS circuit reliability from the MOSFETs and the input vectors," presented at the IEEE/IFIP Intl. Conf. Dependable Sys. Nets., Estoril/Lisbon, Portugal, Jun. 29–Jul. 2, 2009.
[35] D. E. Muller and W. S. Bartky, "A theory of asynchronous circuits," in *Proc. Int. Symp. Theory Switching*, 1959, pp. 204–243.
[36] C. Winstead, "C-element multiplexing for fault-tolerant logic circuits," *IET Electron. Lett.*, vol. 45, no. 19, pp. 969–970, Sep. 2009.
[37] W. Jang and A. J. Martin, "Soft-error robustness in QDI circuits," in *Proc. 1st Workshop Syst. Effects Logic Soft Errors*, Univ. Illinois, Champaign-Urbana, IL, Apr. 2005.
[38] N. Nguyen, H. Kuwahara, C. Myers, and J. Keener, "Design of a genetic Muller c-element," in *Proc. 13th IEEE Int. Sym. Async. Circuits Syst.*, 2007, pp. 95–104.
[39] C. Lageweg, S. Cotofana, and S. Vassiliadis, "A linear threshold gate implementation in single electron technology," in *Proc. IEEE Comput. Soc. Workshop VLSI*, Orlando, FL, Apr.2001, pp. 93–98.
[40] J. Han and P. Jonker, "A system architecture solution for unreliable nanoelectronic devices," *IEEE Trans. Nanotechnol.*, vol. 1, no. 4, pp. 201–208, Dec. 2002.
[41] F. Martorell, S. Cotofana, and A. Rubio, "Fault tolerant structures for nanoscale gates," in *Proc. 7th IEEE Int. Conf. Nanotechnol.*, 2007, pp. 605–610.

Jie Han (S'02–M'05) received the B.Sc. degree in electronic engineering from Tsinghua University, Beijing, China, in 1999 and the Ph.D. degree from Delft University of Technology, The Netherlands, in 2004.

He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He was an Institute for Nanoelectronics and Computing (NASA INAC) Postdoctoral Fellow in the Department of Electrical and Computer Engineering, University of Florida from 2004 to 2007. From 2007 to 2009, he worked as a Research Scientist at the Advanced Medical Diagnostics S.A./B.V. in Belgium. His research interests include reliability and fault tolerance, nanoelectronic circuits and systems, novel computational models for nanoscale and biological applications.

Dr. Han was nominated for the 2006 Christiaan Huygens Price of Science by the Royal Dutch Academy of Science. His work was recognized by the 125th anniversary issue of Science, for developing theory of fault-tolerant nanocircuits. He served as Technical Committee Members in the IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT 2011) and the International Workshop on Unique Chips and Systems (UCAS-6) 2010. He is a member of the ACM.

Erin R. Boykin (maiden name Taylor) was born in Orlando, Florida in 1983. She received the B.S. degree in computer engineering from the University of Florida, Gainesville, in 2005, and the M.S. degree in electrical engineering from the University of Florida in 2007. She is currently a Ph.D. student in the Department of Electrical and Computer Engineering, University of Florida and plans to graduate with her doctorate in May 2011.

Hao Chen received the B.Sc. degree in electronic engineering from Zhejiang University, Hangzhou, China, in 2009. He is currently working toward the M.Sc. degree at the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada.

His research interests include statistical modeling of nanoscale devices and circuits, energy-efficient, reliability-aware design of nanoscale circuits and systems.

Jinghang Liang was born in Jilin, China, in 1986. He received the B.Eng. degree in the Institute of Microelectronics from Tsinghua University, Beijing, China, in 2009. Since 2010, he has been working toward the M.Sc. degree at the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada.

His research interests include nanoelectronic circuit and system design, fault-tolerant system design, and genetic network modeling. He is also interested in RF circuit designs and especially high-speed PLLs.

José A. B. Fortes (S'80–M'83–SM'92–F'99) received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1984.

From 1984 to 2001, he was on the Faculty of the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. In 2001, he joined both the Department of Electrical and Computer Engineering and the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, as a Professor and BellSouth Eminent Scholar. His research interests include distributed computing, fault-tolerant systems and nanocomputing.

Dr. Fortes is a Fellow of the IEEE and was a Distinguished Visitor of the IEEE Computer Society from 1991 to 1995.