Markov Chains and Probabilistic Computation—A General Framework for Multiplexed Nanoelectronic Systems

Yan Qi, Jianbo Gao, and José A. B. Fortes, *Fellow, IEEE*

***Abstract*—In emerging nanotechnologies, reliable computation will have to be carried out with unreliable components being integral parts of computing systems. One promising scheme for designing these systems is von Neumann's multiplexing technique. Using bifurcation theory and its associated geometrical representation, we have studied a NAND-multiplexing system recently proposed. The behavior of the system is characterized by the stationary distribution of a Markov chain, which is uni- or bi-modal, when the error probability of NAND gates is larger or smaller than the threshold value, respectively. The two modes and the median of the stationary distribution are the keys to the characterization of the system reliability. Examples of potential future nanochips are used to illustrate how the NAND-multiplexing technique can lead to high system reliability in spite of large gate error probability while keeping the cost of redundancy moderate. In nanoelectronic systems, while permanent defects can be taken care of by reconfiguration, probabilistic computation schemes can incorporate another level of redundancy so that high tolerance of transient errors may be achieved. The Markov chain model is shown to be a powerful tool for the analysis of multiplexed nanoelectronic systems.**

***Index Terms*—Fault tolerance, Markov chain, NAND multiplexing, probabilistic computation.**

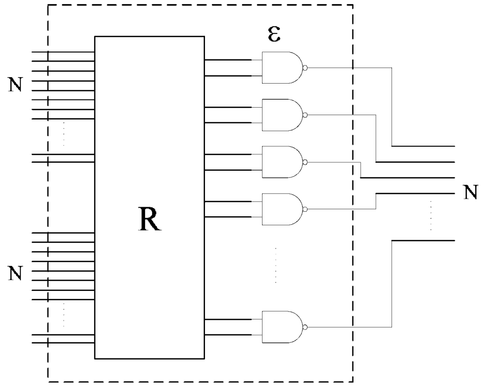


Fig. 1. NAND multiplexing unit (redrawn from [5]).

I. Introduction

IT HAS been widely recognized that novel nanoelectronic devices such as carbon nanotubes and molecular switches are bound to have high manufacturing defect rates due to the stochastic nature of the bottom-up physical and chemical self-assembly processes. Dimension scaling of CMOS devices into the nanoscale regime imposes similar problems. For example, transistor and interconnect parametric variations, leakage currents and power dissipation will cause a large number of functional faults, permanent and transient, in device and circuit operation [1]. Thus, we are inevitably faced with the question of how to build reliable systems out of unreliable components. To tackle this problem, several fault-tolerant techniques based on redundancy have been investigated for nanocomputers recently [2]–[4]. The three strategies that have received most attention are reconfigurable computer architectures, R-modular redundancy (R being an odd number not smaller than three) and its cascaded version, and schemes based on von Neumann's multiplexing technique. The former two techniques have traditionally been used for microsystems, where device failure rate is typically 10^{-7} to 10^{-6}. However, with these techniques alone, high fault-tolerance is hard to achieve for nanocomputers that are anticipated to have a device density of 10^{12} per chip and faulty components pervasive in space and time. Thus, the classic von Neumann multiplexing technique, which essentially treats faulty components as an integral part of the system, has received a revival.

Mathematically, the problem of reliable computation with unreliable components can be termed "probabilistic computation," which was initiated by von Neumann almost 50 years ago [5]. The prototypical scheme proposed by von Neumann is called multiplexing, in which automata are built upon one type of faulty universal gates such as the NAND gate and the three-input majority gate. Take NAND multiplexing as an example, multiplexing refers to the construction shown in Fig. 1. A NAND multiplexing unit is comprised of a randomizing unit R and N copies of NAND gates each failing (flipping the output bit value) with probability ϵ (ϵ will be referred to as gate error probability hereon). The NAND multiplexing unit takes two bundles of N wires as inputs and generates a bundle of N wires as the output. Through a random permutation by the "randomizing unit," the inputs in one bundle are randomly paired with those from the other bundle to form input pairs to the duplicated NAND gates. In systems based on this construction, each signal is carried on a bundle of

Manuscript received January 26, 2004; revised June 15, 2004. This work was supported by the National Science Foundation under ITR Grant 0 135 946 and by the National Aeronautics and Space Administration under Award NCC 2-1363.

The authors are with the Advanced Computing and Information Systems Laboratory, Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: yqi@ecel.ufl.edu; gao@ece.ufl.edu; fortes@acis.ufl.edu).

Digital Object Identifier 10.1109/TNANO.2004.834192

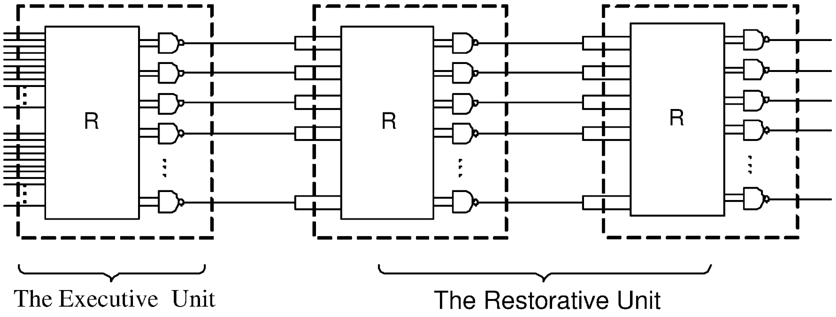


Fig. 2. Multistage NAND multiplexing system. Each "computation node" in the system consists of three NAND multiplexing units shown in Fig. 1. The executive unit does logic computation and two NAND multiplexing units act as a "restorative unit" (redrawn from [5]).

N wires instead of a single wire and every logic computation is done by N duplicated gates simultaneously. The logic state "1" or "0" is decided for a bundle when its "excitation level," i.e., the fraction of excited wires, is above or below a preset threshold. In a multiplexed system, each "computation node" is comprised of three such multiplexing units connected in series, as illustrated by Fig. 2. The "executive" unit carries out logic computation and the "restorative" unit (comprised of two NAND multiplexing units) restores the excitation level of the output bundle of the "executive" unit to its nominal level. von Neumann has stated that if ϵ is sufficiently small, computation by such constructions can always be done with arbitrarily high reliability by increasing the bundle size N. However, modeling at the system level was lacking and two fundamental questions remain, which are: 1) How does the system behavior depend on the individual faulty components? and 2) What are the mathematical frameworks to study such multiplexing schemes in general based on von Neumann's prototype?

Consideration of the first question leads one to study the maximum tolerable gate error probability ϵ_* (we shall refer to ϵ_* as the gate error threshold hereon) for universal gates. Much work has been done along this line [6]–[9]. The error threshold values are significant in that they define the valid working region for "probabilistic logics," beyond which no reliable computation is ever possible. When modeling multiplexing schemes for practical system design, these threshold values are important factors to consider.

To tackle the second question, recently, Han and Jonker have made a significant step forward [10]–[12]. Inspired by von Neumann's multiplexing scheme, they have proposed a Markov-chain-based model for a system architecture consisting of chains of parallel NAND multiplexing units. This modeling approach takes into account the stochasticity of the excitation levels of the input bundles while von Neumann assumed deterministic values for these quantities. Han and Jonker's approach has been further extended by Sadek *et al.* [13], where the stochasticity of the gate error probability is considered. These works have shown that: 1) by retaining parallelism throughout the network, the redundancy needed to achieve high system reliability is significantly reduced and 2) the Markov-chain model may be a powerful mathematical framework for the analysis of such multiplexing schemes. Their works have also stimulated several questions. A few of those are as follows.

1) *What impact does the error threshold value for the* NAND *gate have on the behavior of the system, especially in terms of the reliability of the scheme?* Han and Jonker have observed that the NAND multiplexing scheme hardly works when ϵ exceeds a certain threshold value, which is quite close to that obtained by Evans and Pippenger [9]. Through a different network setting, Sadek *et al.* have obtained a threshold value originally stated by von Neumann [5], [13].
2) *Does the Markov chain have a single stationary distribution, or does its stationary distribution only exist for odd (or even) stage numbers?* Han and Jonker have suspected the latter. This scenario is particularly appealing since the logic outputs of a cascade of NAND gates flip from one stage to the next.
3) *How may one characterize and evaluate the reliability of the system?*

In order to answer the questions above, in this paper, we have carried out a thorough analysis of the transient and stationary behavior of the Markov chain model by Han and Jonker. The stationary distribution is essential in that it captures the behavior of very large systems within which computations typically proceed with a large number of steps. We have observed two distinct types of stationary distributions, uni-modal and bi-modal, depending on whether the gate error probability ϵ is larger or smaller than the threshold value. The uni-modal distribution does not allow one to distinguish between the two states "0" and "1," thus it indicates a state of irrelevance. The bi-modal distribution, on the contrary, naturally defines the two states "0" and "1" by the two modes. We shall show below that the bi-modal distribution is also the key to characterize the reliability of the system. We note that, although we have focused on the scheme by Han and Jonker, our basic findings can help understand the more general scheme by Sadek *et al.*

This paper is organized as follows. In Section II, we analyze the situation when individual two-input NAND gates fail to work. In particular, we shall introduce a geometrical interpretation inspired by bifurcation theory. In Section III, we study the transient and stationary behavior of the Markov chain. We will show that bifurcation analysis is especially useful in this regard. We discuss how we may characterize and evaluate system reliability in Section IV. In Section V, we discuss the potential applications of this framework to the design of fault tolerant nanoelectronic systems. Section VI concludes this paper and discusses other analysis approaches to multiplexing schemes and potential hybrid schemes based on multiplexing and reconfiguration.

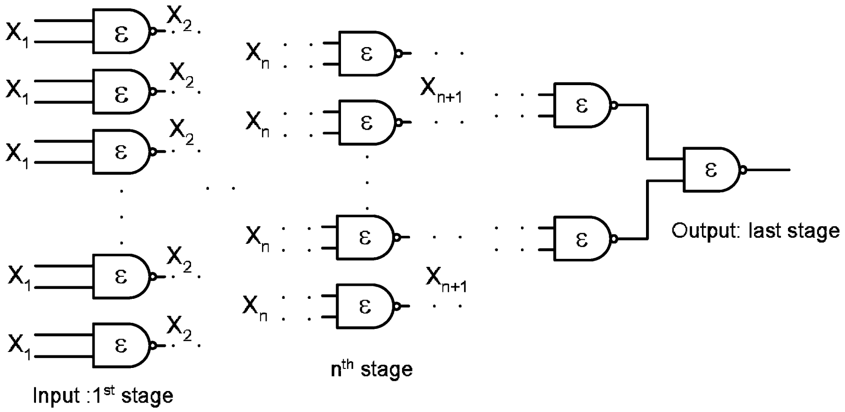


Fig. 3. Schematic of a full binary tree whose nodes are faulty two-input NAND gates with gate error probability ϵ.

II. Bifurcation Analysis of Individual Two-Input NAND Gates

In order to gain understanding of the system dynamics of probabilistic logics and the associated concept of reliable computation, here, we focus on the operation of individual two-input faulty NAND gates in a binary tree of cascaded NAND gates. In the analysis that follows, we shall assume that the NAND gates make a von Neumann error (bit-flip) with probability ϵ while the input and output lines function reliably. Using bifurcation analysis [14], [15], we have re-derived the error threshold value for two-input NAND gates $\epsilon_* = 0.088856\ldots$, which was previously obtained by Evans and Pippenger [9] through a more involved information theoretic approach. This threshold value reveals two parametric intervals of ϵ where reliable computation by noisy gates can or cannot proceed, which is in accordance with previous research along this line. In particular, the bifurcation diagram offers a geometrical view to interpret the distinct system dynamics in these two intervals.

Let us analyze an individual two-input NAND gate in the NAND multiplexing unit. Note that, for such probabilistic logics, we are only concerned with the probability of an input or output being in the conventional logic state of "1" or "0." Thus, all signals in a circuit built upon such noisy gates are interpreted as probabilities [5], [6], [9]. Let us denote the probabilities of the two inputs being "1" by X and Y and further assume that the two inputs are independent. The probability Z of the output being "1" is

$$\begin{aligned} Z &= (1-\epsilon)(1-XY) + \epsilon XY \\ &= (1-\epsilon) + (2\epsilon - 1)XY. \end{aligned} \tag{1}$$

As detailed in [15], the worst case scenario in terms of computation reliability occurs when $X = Y$ in (1). Let us study this case by constructing a full binary tree of faulty NAND gates, as shown in Fig. 3. In the analysis that follows, we assume that this circuit is a discrete time system. The leaves and root of the tree correspond to the start and end points of the computation, respectively. Assume that all inputs to the leaf NAND gates are independent and have equal probabilities of being "1," which we shall denote by X_1. The circuit structure then guarantees that the inputs to all gates at an arbitrary stage n are also independent and have equal probabilities of being "1," which we shall denote by X_n. For such a construction, (1) reduces to a simple iterative equation (or map [14])

$$X_{n+1} = f(X_n) = (1-\epsilon) + (2\epsilon - 1)X_n^2. \tag{2}$$

In order to find out how the signal propagation in this circuit depends on the gate error probability ϵ, we carry out a bifurcation analysis of (2). For each $\epsilon \in (0, 1/2)$, we choose an arbitrary initial condition X_1 and generate a sequence $X_i, i = 2, \ldots, n, \ldots$, by iterating (2) until convergence to attractors is observed. Those attractors for the X_i sequence are then plotted against each ϵ [14], [15]. This leads to the bifurcation diagram in Fig. 4 $(\Delta\epsilon = 0.001)$. This diagram reveals that a period-doubling bifurcation occurs at $\epsilon_* = (3 - \sqrt{7})/4 = 0.08856\ldots$. The bifurcation point ϵ_* divides the diagram into two regions, which are: 1) $\epsilon_* < \epsilon < 1/2$ and 2) $0 < \epsilon < \epsilon_*$. Analytically, we are able to explain the distinct system behavior through stability analysis of the fixed points for the map $X_{n+1} = f(X_n)$. By solving the equation $x_0 = (1-\epsilon) + (2\epsilon - 1)x_0^2$, one obtains

$$x_0 = \frac{-1 + \sqrt{4(1-\epsilon)(1-2\epsilon) + 1}}{2(1-2\epsilon)} \tag{3}$$

where x_0 is the fixed point solution for the interval $0 < \epsilon < 1/2$. The system behavior changes as a result of the change of stability of x_0 in regions 1) and 2) mentioned above. In region 1), $|f'(x_0)| = |(4\epsilon - 2)x_0| = |1 - \sqrt{1 - 4(1-\epsilon)(2\epsilon - 1)}| < 1$, implying that successive values of X_n differ from x_0 by an amount that tends to zeros as $n \to \infty$. Thus, x_0 is a stable fixed point. In region 2), since $|f'(x_0)| > 1, x_0$ turns unstable (see [15] for more details). From Fig. 4, we further observe that in region 2), the system exhibits a period-2 oscillation. These two points on the limit cycle have been labeled by x_+ and x_- in Fig. 4. The period-2 oscillation is described by $x_+ = f(x_-)$ and $x_- = f(x_+)$ [15], from which one obtains

$$x_+ = \frac{1 + \sqrt{4(1-\epsilon)(1-2\epsilon) - 3}}{2(1-2\epsilon)} \tag{4}$$

$$x_- = \frac{1 - \sqrt{4(1-\epsilon)(1-2\epsilon) - 3}}{2(1-2\epsilon)}. \tag{5}$$

To carry out computation, one has to map each probability to a logic state. A simple scheme is to set a threshold at $x_* = 0.5$

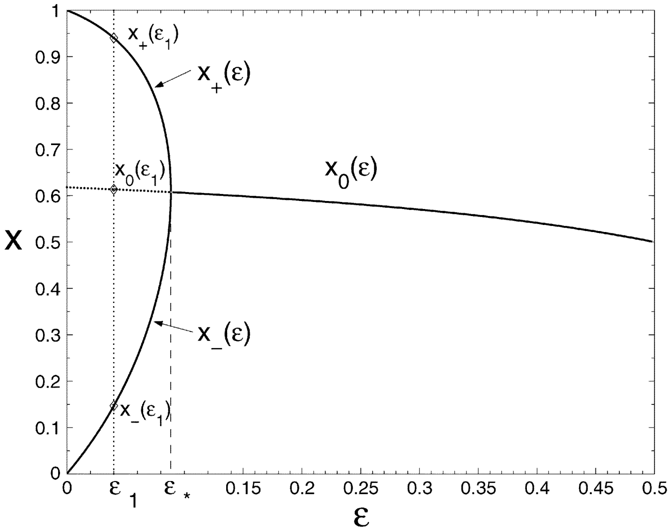


Fig. 4. Bifurcation diagram for the faulty two-input NAND gate.

so that any probability falling in the interval $[0, x_*)$ is assigned to logic "0" while that falling in $(x_*, 1]$ is assigned to logic "1." This scheme is attractive because it is symmetric for both states "0" and "1." For such a scheme to work, it is implicitly assumed that any input in the interval $(0.5, 1]$ leads to an output in the interval $[0, 0.5)$ according to (2), and vice versa. However, this is not true. In fact, by (2), one can readily prove that any input in the interval $[0, x_0)$ produces an output in the interval $(x_0, 1]$, and vice versa. This means that if we set $x_* = 0.5$, an input in the interval $(0.5, x_0)$, which is interpreted as state "1," produces an output in the interval $(x_0, f(0.5))$, which is also assigned to state "1." This indicates the NAND gate does not work at all in the interval $(0.5, f(0.5))$. In principle, this could be tolerated by requiring that the interval $(0.5, f(0.5))$ not be used for computation. However, when this is the case, this simple scheme is no longer symmetric for states "0" and "1." Thus, it is better to set $x_* = x_0$. In fact, such a choice was adopted by Evans and Pippenger [9].

Below, we shall interpret $[0, x_0)$ as state "0" and $(x_0, 1]$ as state "1." Now it is easy to see that $0 \leq \epsilon < \epsilon_*$ is the parameter interval where the NAND gate functions. When $\epsilon > \epsilon_*$, the outputs converge to the stable fixed point x_0 regardless of what the initial inputs are. In this case, "1" or "0" cannot be decided for the outputs, which corresponds to a state of irrelevance [5]. In this region, the gate no longer functions as a NAND gate.

It is interesting to compare Fig. 4 with [13, Fig. 10], from which an error threshold $\epsilon'_* = 0.0107$ is derived. Reference [13, Fig. 10] was determined by four equations, with the first three equivalent to our (2) and the fourth describing an inverter. Based on Fig. 4, it is clear that the consecutive use of (2) makes x, the output of the NAND tree, converge to either x_+ or x_-. The relation imposed by the inverter after every three iterations of (2) requires that $x_+ + x_- \simeq 1$. This is only possible when ϵ is very close to zero, which is the fundamental reason that the error threshold value obtained in [13] is much smaller than $\epsilon_* = 0.08856, \ldots$.

Before proceeding, we emphasize that as the gate error probability ϵ exceeds the threshold value ϵ_*, the individual NAND gates can no longer compute reliably under any circumstances, hence, the system built from these faulty gates is bound to break down. In fact, the error threshold value must show up at system level too, albeit in a form that depends on the particular system architecture. We further surmise that other important features of the basic bifurcation diagram such as the fixed point x_0, the two periodic points x_-, and x_+ should also manifest themselves in a system that falls into the framework of probabilistic computation. We shall make these arguments more concrete shortly.

III. Markov-Chain Analysis of the NAND Multiplexing System

Here, we shall first briefly revisit the Markov-chain model proposed by Han and Jonker [10]. We shall show that the input–output relation of each stage in the multistage multiplexing system can be described by a homogeneous transition matrix and the associated Markov chain characterizes signal propagation in the system. The outputs of such a multistage system (with a very large number of stages) always converge to one stationary distribution $\vec{\pi}_*$. The characteristics of the stationary distribution, transient distributions, and associated speed of convergence collectively determine the system dynamics, with the stationary distribution playing the essential role. Through numerical simulation, we have found a dependence of system dynamics on gate error probability ϵ analogous to that in the case of the binary tree of NAND gates considered in Section II. For the interval $0 < \epsilon < \epsilon_*, \vec{\pi}_*$ is bi-modal, while it is uni-modal for the interval $\epsilon_* < \epsilon < 1/2$. Only the bi-modal case is of practical interest since the two modes correspond to the logic states "0" and "1," while a uni-modal distribution only indicates a state of irrelevance. The relevance of the bifurcation analysis is apparent from the fact that x_0 is always the median of $\vec{\pi}_*$ and the two points on the bifurcation diagram x_+ and x_-, which defines a period-2 oscillation, are where the two modes of the bi-modal stationary distribution center around. For a fixed ϵ, the specific shape of $\vec{\pi}_*$ is further determined by the bundle size N. In the bi-modal case, $\vec{\pi}_*$ can be roughly approximated by two normal distributions with means x_+ and x_- and variances close to $(1/N)x_0x_-(1 - x_-)$ and $(1/N)(1 - x_0)x_+(1 - x_+)$. With increasing bundle size, the normal approximations become more accurate and system reliability (as formally characterized and analyzed in Section IV) is improved.

A. Markov-Chain Model for the NAND Multiplexing System

Consider the NAND multiplexing unit in Fig. 1. Let us denote the number of excited wires in a bundle of size N by a random variable K and call K/N the bundle's excitation level. Assume that each wire in the bundle has a probability of Z to be excited. Obviously K follows a binomial distribution with parameters N and Z as follows:

$$P(K = k) = \binom{N}{k} Z^k (1 - Z)^{N-k}. \tag{6}$$

The mean and variance of K are NZ and $NZ(1 - Z)$. The excitation level K/N is a normalized version of K with mean value Z. According to the central limit theorem, when Z is fixed and the bundle size N is very large, the distribution for K can be approximated by a normal distribution $f(k)$ [10] as follows:

$$f(k) = \frac{1}{\sqrt{2\pi}\sqrt{NZ(1-Z)}} e^{\frac{1}{2}\left(\frac{k-NZ}{\sqrt{NZ(1-Z)}}\right)^2}. \tag{7}$$

The model used by Han and Jonker [10] to study a multistage NAND multiplexing system is as follows. The number of stimulated wires in an output bundle of size N is described by a stochastic process $K = \{K_n; n \in M\}$, where n is the stage number, $K_n \in J = \{0, 1, \ldots, N\}$ is the state of the process K at stage n, and $M = \{0, 1, \ldots\}$. The relation between the input and output distributions of each multiplexing unit (also referred to as stage) is described by a first-order Markov chain

$$\begin{aligned} P\big(K_{n+1} = j_{n+1} \,|\, K_0 = j_0, \ldots, K_n = j_n\big) \\ = P\big(K_{n+1} = j_{n+1} \,|\, K_n = j_n\big) \end{aligned} \tag{8}$$

The probability transition matrix $\mathbf{P}$ is given by

$$\mathbf{P} = \begin{bmatrix} P(0\,|\,0) & P(1\,|\,0) & \cdots & P(N\,|\,0) \\ P(0\,|\,1) & P(1\,|\,1) & \cdots & P(N\,|\,1) \\ \vdots & & \vdots & \\ P(0\,|\,N) & P(1\,|\,N) & \cdots & P(N\,|\,N) \end{bmatrix}. \tag{9}$$

The matrix element $P(j\,|\,i)$ is the conditional probability $P(K_{n+1} = j\,|\,K_n = i)$. According to (6), $P(j\,|\,i) = (N/j)Z(i)^j(1 - Z(i))^{N-j}$ where

$$Z(i) = (1 - \epsilon) - (1 - 2\epsilon)(i/N)^2. \tag{10}$$

In (10), it has been assumed that the two inputs to each gate are independent and identically distributed. We find that $\mathbf{P}$ (one-step transition matrix) is essentially determined by ϵ and N. Let $\vec{\pi}_0 = [\pi_0(0), \pi_0(1), \ldots, \pi_0(N)]$ be the initial input distribution of the Markov chain, then the output distribution at the nth stage is

$$\vec{\pi}_n = \big[\pi_n(0), \pi_n(1), \ldots, \pi_n(N)\big] = \vec{\pi}_0\mathbf{P}^n. \tag{11}$$

The limiting probability distribution

$$\vec{\pi}_* = \lim_{n\to\infty} \vec{\pi}_n \tag{12}$$

is called the stationary distribution. It satisfies

$$\vec{\pi}_* = \vec{\pi}_*\mathbf{P}. \tag{13}$$

Apparently $\vec{\pi}_n$ and $\vec{\pi}_*$ are discrete with N points. Equations (12) and (13) suggest that there are two ways to compute a stationary distribution when it exists. The first approach is to compute the nth-step probability vector $\vec{\pi}_n$ and find the limit in (12). To do this, we need to assume an arbitrary initial distribution $\vec{\pi}_0$ and perform matrix multiplication. A more direct approach is to determine $\vec{\pi}_*$ by eigenanalysis as (13) suggests that $\vec{\pi}_*$ is actually the left eigenvector of $\mathbf{P}$ associated with the eigenvalue $\lambda_* = 1$. Thus, the stationary distribution is determined by the transition matrix $\mathbf{P}$ only and does not depend on the initial distribution $\vec{\pi}_0$.

Similarly, it is also convenient to analyze the transient behavior by performing eigendecomposition of the transition matrix $\mathbf{P}$. A usual way is to use spectral representation for $\vec{\pi}_n$ [16]. In [15], we introduce an equivalent, but simpler approach, which essentially represents $\vec{\pi}_n$ as a linear combination of the eigenvectors of the transition matrix $\mathbf{P}$ as follows:

$$\begin{aligned} \vec{\pi}_n &= \vec{\pi}_0\mathbf{P}^n \\ &= \Big(\vec{\pi}_* + \alpha_1\vec{B}_1 + \cdots + \alpha_N\vec{B}_N\Big)\mathbf{P}^n \\ &= \vec{\pi}_* + \alpha_1\lambda_1^n\vec{B}_1 + \alpha_2\lambda_2^n\vec{B}_2 + \cdots + \alpha_N\lambda_N^n\vec{B}_N \end{aligned}$$

where λ_i and $\vec{B}_i$ are distinct eigenvalues and the associated left eigenvectors of $\mathbf{P}$. The speed at which the transients die out (or at all) depends on the magnitudes of the eigenvalues. The smaller the magnitudes are, the faster $\vec{\pi}_n$ converges to $\vec{\pi}_*$.

B. Numerical Evaluation

1) Stationary Distribution: Noticing that the transition matrix $\mathbf{P}$ is determined by gate error probability ϵ and bundle size N, we proceed to analyze the dependence of the stationary distribution $\vec{\pi}_*$ on ϵ and N.

Through numerical evaluation of $\vec{\pi}_*$, we find that ϵ is the key parameter that determines the shape of $\vec{\pi}_*$. When ϵ is smaller than the threshold value ϵ_*, $\vec{\pi}_*$ is bi-modal, and when ϵ exceeds ϵ_*, $\vec{\pi}_*$ turns into a uni-modal curve. From Fig. 5, we clearly see the transformation of $\vec{\pi}_*$ from a bi- to uni-modal when we increase ϵ from 0.03 to 0.1, with $\epsilon_* = 0.08856\ldots$, as the "bifurcation" point.

We have shown in Fig. 6 a few stationary distributions for different values of ϵ and bundle sizes N, where the abscissa k/N is the bundle excitation level and the ordinate $\vec{\pi}_*(k)$ is the probability that a portion of k/N wires in the bundle are excited. It turns out that for both bi- and uni-modal curves, x_0 is the median of the distribution

$$\sum_{k=0}^{\lfloor NX_0 \rfloor} \vec{\pi}_*(k) = \sum_{k=\lceil Nx_0 \rceil}^{N} \vec{\pi}_*(k) = 1/2. \tag{14}$$

Further, we find that all the bi-modal curves peak around x_- and x_+ with x_0 as the point separating the two "features" of the stationary distribution. By segmenting the stationary distribution from $k/N = x_0$ asunder, we obtain two curves, $\vec{S}_-$ and $\vec{S}_+$, centering around x_- and x_+. In vector form, they are given by

$$\vec{S}_-(k) = \vec{\pi}_*(k), \qquad k = 0, 1, \ldots, \lfloor Nx_0 \rfloor$$

and

$$\vec{S}_+(k) = \vec{\pi}_*(k), \qquad k = \lceil N(1 - x_0) \rceil, \ldots, N.$$

When N is large, $\vec{S}_-$ and $\vec{S}_+$ are approximately normally distributed and we speculate that the "effective lengths" for $\vec{S}_-, \vec{S}_+, N_-$, and N_+ are close to Nx_0 and $N(1 - x_0)$, while

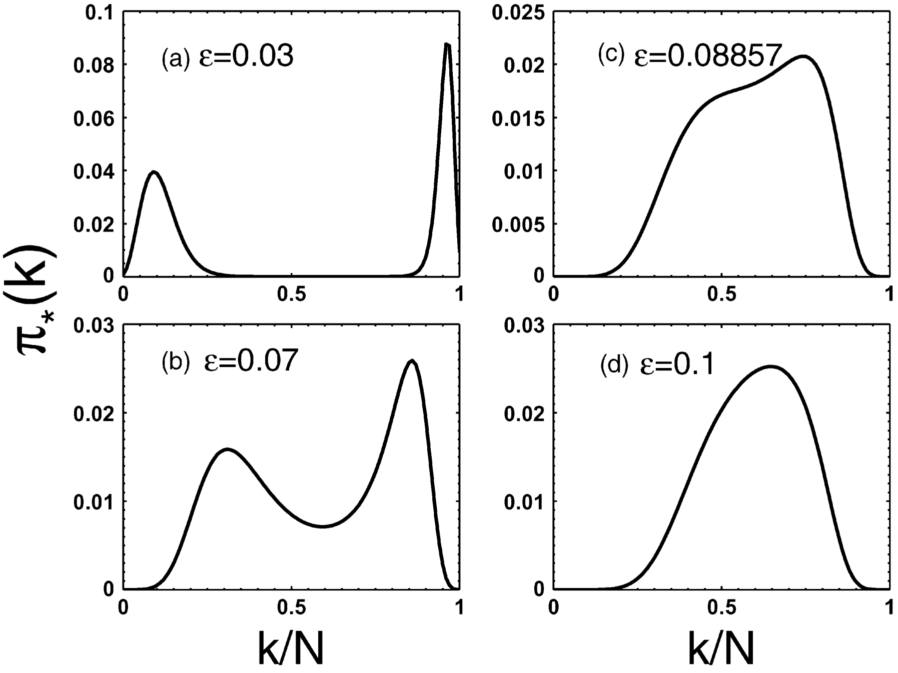


Fig. 5. Stationary distributions $\vec{\pi}_*(k)$ for the Markov chain with different gate error probability ϵ. For $0 < \epsilon < \epsilon_* = (3 - \sqrt{7}/4) = 0.08856\ldots$, $\vec{\pi}_*(k)$ is bi-modal. For $\epsilon > \epsilon_*$, $\vec{\pi}_*(k)$ is uni-modal. N is the bundle size and k is the number of "excited" wires. Note that each $\vec{\pi}_*$ "curve" represents a discrete probability mass function with N points. Thus, it should not be mistaken as continuous as it visually appears. This is also true for Figs. 6 and 7.

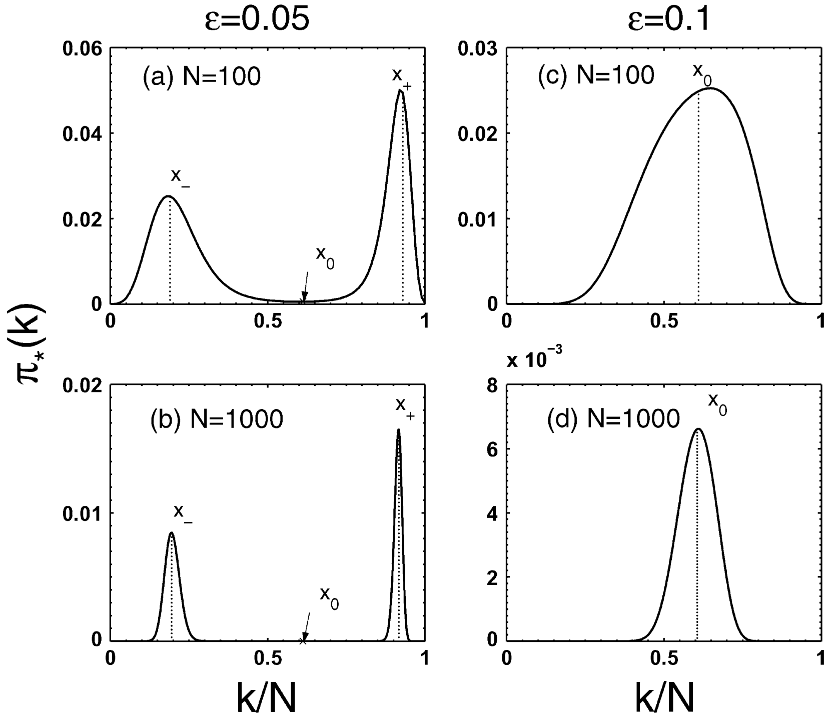


Fig. 6. Stationary distributions for: (a) $N = 100, \epsilon = 0.05$, (b) $N = 1000, \epsilon = 0.05$, (c) $N = 100, \epsilon = 0.1$, and (d) $N = 1000, \epsilon = 0.1$. x_0 is the median of $\vec{\pi}_*(k)$. The two modes of the bi-modal stationary distributions peak around x_+ and x_-.

the variances for $\vec{S}_-$ and $\vec{S}_+, V_-$, and V_+, are proportional to $(1/N)x_0x_-(1 - x_-)$ and $(1/N)(1 - x_0)x_+(1 - x_+)$. Through numerical simulation of the stationary distributions for a gate error probability $\epsilon = 0.05$ and bundle sizes $N = 100, N = 300, N = 500$, and $N = 1000$, we find that the Gaussian approximation of $\vec{S}_-$ and $\vec{S}_+$ gives good results when $N \geq 500$ (e.g., for $N = 1000$). It is reasonable to expect that when $N \to \infty$, the peaks will become δ impulses at x_- and x_+.

We conclude from the analysis above that beginning from an arbitrary initial distribution, the outputs of the system modeled as a Markov chain eventually approaches one stationary distribution. This is somehow contrary to our intuition that, for odd and even stages, the output might assume two different sta-

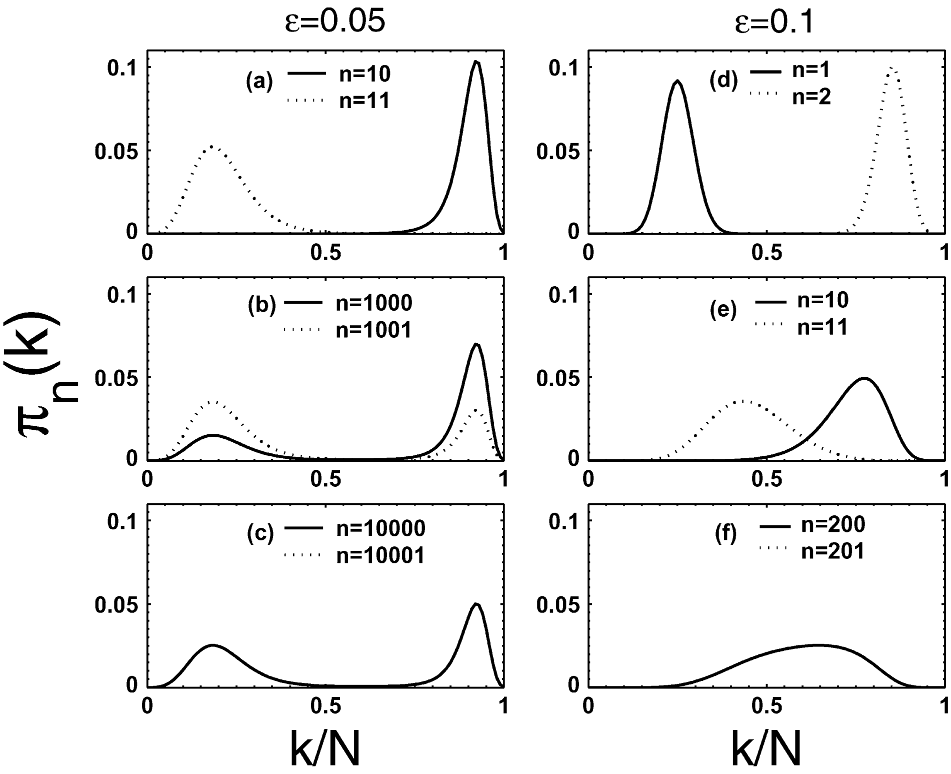


Fig. 7. Transient distributions $\vec{\pi}_n$. $N = 100$; stage number $= n$; $\vec{\pi}_0(i) = 0$ for $i = 0, 1, \ldots, 0.9N - 1, 0.9N + 1, \ldots, N$; $\vec{\pi}_0(i) = 1$ for $i = 0.9N$. (a)–(c) $\epsilon = 0.05$. (d)–(f) $\epsilon = 0.1$.

tionary distributions representing two logic states "1" and "0." Further more, it is interesting to note that when $\epsilon < \epsilon_*$, the logic states "1" and "0" are symmetric in the sense that the probabilities contained by the two modes of the bi-modal stationary distribution are equal according to (14). How every Markov chain with arbitrary ϵ finally converges to one stationary distribution and how the symmetry between "1" and "0" states is achieved when $\epsilon < \epsilon_*$ will become clearer through transient analysis in Section III-B.2.

2) Transient Analysis: The stationary distribution analysis reveals the nature of the system as a Markov chain and its long-term behavior. However, in digital computers, the logical depth of a functional block in a microprocessor is usually on the order of ten, thus, we are motivated to investigate the transient behavior of the NAND multiplexing system. In [17, Figs. 3 and 4], the output distributions after one computation node are shown for multiplexing schemes with one and four restorative units, respectively. A trend can be observed that the smaller the gate error probability is, the closer the one-step output distribution is to the nominal logic mode. To understand how the output distribution evolves after going through more than a few stages, let us analyze the transient behavior of the system for the following two cases:

Case 1) $\epsilon = 0.05 < \epsilon_*$.
Case 2) $\epsilon = 0.1 > \epsilon_*$.

To do this, we evaluate the output distributions at two consecutive stages (with the stage number denoted by n) using (11). The input bundle is chosen such that exactly a fraction of 90% wires are excited. Thus, the initial distribution is $\vec{\pi}_0 = [0, \ldots, 0, 1, 0, \ldots, 0]$, where the only nonzero element is $\vec{\pi}_0(0.9N) = 1$. The bundle size is fixed at $N = 100$. We obtain the transient output distribution curves for a number of different stage numbers n in Fig. 7. In Fig. 7(a)–(f), the solid curve represents $\vec{\pi}_{n=\text{even}}$, while the dotted curve represents $\vec{\pi}_{n=\text{odd}}$. Fig. 7(a)–(c) depicts the evolution of $\vec{\pi}_{n=\text{even}}$ and $\vec{\pi}_{n=\text{odd}}$ when $\epsilon = 0.05$, which is representative of $0 < \epsilon < \epsilon_*$. We observe that when the stage number n is approximately ten, $\vec{\pi}_{n=\text{odd}}$ and $\vec{\pi}_{n=\text{even}}$ are both uni-modal and their modes are very different [see Fig. 7(a)]. When n is increased to around 1000, both curves have turned from uni- to bi-modal, with two modes peaking around x_- and x_+. Take $\vec{\pi}_{n=\text{even}}$ as an example, the magnitude of its right mode decreases while that of its left mode grows larger. The opposite is true for $\vec{\pi}_{n=\text{odd}}$ [see Fig. 7(b)]. Thus, the difference between $\vec{\pi}_{n=\text{odd}}$ and $\vec{\pi}_{n=\text{even}}$ keeps decreasing during this process. When n is further increased to around 10 000, both $\vec{\pi}_{n=\text{odd}}$ and $\vec{\pi}_{n=\text{even}}$ are very close to the stationary distribution and the difference between them is negligible [see Fig. 7(c)]. Fig. 7(d)–(e) shows a similar evolution process of $\vec{\pi}_{n=\text{odd}}$ and $\vec{\pi}_{n=\text{even}}$ when $\epsilon = 0.1$, which is typical for $\epsilon > \epsilon_*$. However, it is interesting to note that, in this case, both $\vec{\pi}_{n=\text{odd}}$ and $\vec{\pi}_{n=\text{even}}$ remain uni-modal for arbitrary n and the two distributions have converged to $\vec{\pi}_*$ within 200 stages.

The speed of convergence to the stationary distribution depends essentially on the choice of $\vec{\pi}_0$ and the eigenvalues of the transition matrix $\mathbf{P}$ [see (11) and (12)]. By (13), we know that the transient output distribution $\vec{\pi}_n$ converges to $\vec{\pi}_*$ in one step when $\vec{\pi}_0 = \vec{\pi}_*$ and the smaller the difference between $\vec{\pi}_0$ and $\vec{\pi}_*$ is, the faster the convergence.

The general transient analysis above gives us a concrete idea of how the signals evolve as the number of stages gets larger. Now let us analyze a special case so as to fully understand how the symmetry of the stationary distribution mentioned at the end of Section III-B.1 is achieved. We segment the stationary distribution $\vec{\pi}_*$ into two portions and do normalization, from which we obtain two vectors $\vec{\pi}_-$ and $\vec{\pi}_+$ given by

$$\vec{\pi}_-(k) = \begin{cases} \vec{\pi}_*(k) \Big/ \left[\sum_{k=0}^{\lfloor Nx_0 \rfloor} \vec{\pi}_*(k)\right], & k = 0, 1, \ldots \lfloor Nx_0 \rfloor \\ 0, \quad k = \lceil Nx_0 \rceil, \ldots, N \end{cases}$$

$$\vec{\pi}_+(k) = \begin{cases} 0, \quad k = 0, 1, \ldots \lfloor Nx_0 \rfloor \\ \vec{\pi}_*(k) \Big/ \left[\sum_{k=\lceil Nx_0 \rceil}^{N} \vec{\pi}_*(k)\right], & k = \lceil Nx_0 \rceil, \ldots, N. \end{cases}$$

Let $\vec{\pi}_-$ be the input to a multiplexing stage, we find that the output $\vec{\pi}'_+ = \vec{\pi}_- \mathbf{P}$ almost exactly matches $\vec{\pi}_+$, except some small "leakage probability" for those elements whose indices are smaller than $\lfloor Nx_0 \rfloor$. Similarly, when we input $\vec{\pi}_+$, the output $\vec{\pi}'_-$ closely resembles $\vec{\pi}_-$ [15]. Generally speaking, the two modes of the stationary distribution represent two logic states, i.e., "0" and "1." After a sufficiently large number of stages, their interaction eventually brings the system to equilibrium, when the output bundle assumes a value of "1" or "0" with equal probability regardless of the initial input distribution $\vec{\pi}_0$. Using $\vec{\pi}_-$ as the input, we denote the measure of convergence by C_-, which is given by

$$C_- = \begin{cases} \sum_{k=0}^{\lfloor NX_0 \rfloor} \vec{\pi}_n(k), & \text{if } n \text{ is odd} \\ \sum_{k=\lceil Nx_0 \rceil}^{N} \vec{\pi}_n(k), & \text{if } n \text{ is even.} \end{cases} \tag{15}$$

Recall from Section III-A that the output distribution $\vec{\pi}_n$ can be represented as a linear combination of the eigenvectors. $\vec{\pi}_n$ approaches $\vec{\pi}_*$ exponentially as n grows and the speed of convergence depends largely on λ_1, the second largest eigenvalue of $\mathbf{P}$. When only odd stages are considered, we have $C_- \sim (\lambda_1^2)^n = e^{n \ln \lambda_1^2}$.

It is interesting to make a comparison between Figs. 5–7 and [13, Fig. 7], where the output distributions of a para-restituted network are shown for stages 1 and 33. Since $\epsilon = 0.1$ is much larger than the threshold value $\epsilon'_* = 0.0107$, we expect the convergence of their Markov chain to the uni-modal stationary distribution to be very rapid. This appears to be the case, as indicated by [13, Fig. 7].

IV. System Reliability

Equipped with a thorough understanding of the NAND multiplexing system through stationary and transient analysis of the Markov chain formulation in previous sections, we are now able to characterize and evaluate the reliability of a system. The characterizations of error and reliability are natural extensions of the notion of probabilistic computation established previously and are particularly important for the design of fault tolerant nanoelectronic systems with a large number of defective components. As discussed previously, when $\epsilon > \epsilon_*$, the stationary distribution is uni-modal and the logic states "1" and "0" become indistinguishable. Thus, only the case $\epsilon < \epsilon_*$ will be considered in the analysis that follows. We shall focus on the practical situation where the logic depth of small functional blocks in microprocessors is constrained to approximately ten. This corresponds to subsystems with small number of stages, for which the output distribution $\vec{\pi}_n$ is far from approaching the stationary distribution $\vec{\pi}_*$. For a fixed $\epsilon < \epsilon_*$ specified by a given technology, by properly controlling the initial distribution $\vec{\pi}_0$ (e.g., by using approximately error-free inputs) and the bundle size N, the output reliability can be greatly improved using our proposed rule to decide whether a "0" or a "1" is present in a bundle. We shall show that this is true even when ϵ is on the order of 10^{-2} and the number of stages is a few hundred while the bundle size is kept at a moderate value of $N = 100$.

First, let us give our definition of error and reliability. Computation by the NAND multiplexing units is essentially a binary communication problem. Errors occur at each stage as the signal propagates through the noisy channel—a cascade of NAND gates. Given the gate error probability ϵ, bundle size N, the system is specified by the transition matrix $\mathbf{P}$. For each input instance and a particular stage n, we need to make a decision of the output logic value based on the number of excited wires observed. Consider the output bundle at the nth stage, let $P_{e,0}$ and $P_{e,1}$ be the respective conditional error probabilities when "0" or "1" is sent. The distribution of K_n, the number of excited wires at the nth stage, is given by $\vec{\pi}_n$. Here, we employ a simplified decision rule: set a decision threshold $\gamma \in [0, 1]$; if K_n exceeds γN then "1" is decided; otherwise "0" is decided. For example, let us assume that the stage number n is odd. The output distribution $\vec{\pi}_n$ should emphasize a 0 mode for a logic "1" input and a 1 mode otherwise. Thus, when "0" is sent, we decide the wrong output if $K_n < N\gamma$, i.e.,

$$P_{e,0} = \Pr\left(K_n < \gamma N\right) = \sum_{i=0}^{i=\lfloor N\gamma \rfloor} \vec{\pi}_n(i). \tag{16}$$

Similarly, when "1" is sent, the error probability is

$$P_{e,1} = \Pr\left(K_n > \gamma N\right) = \sum_{i=\lceil N\gamma \rceil}^{i=N} \vec{\pi}_n(i). \tag{17}$$

Notice that, for the case we consider where the stage number n is small, $\vec{\pi}_n$ has not converged to the stationary distribution, thus, $P_{e,0} + P_{e,1} \neq 1$. The question is how to choose an appropriate decision threshold so that both $P_{e,0}$ and $P_{e,1}$ can be kept small. It is clear from the discussions in Section II that it is natural to choose $\gamma = x_0$, where x_0 is given by (3). However, in so far as system reliability is concerned, so long as $\epsilon \ll \epsilon_*, N \gg 1$, and γ is somewhere in the middle of x_- and x_+, $P_{e,0}$, and $P_{e,1}$, given by (16) and (17), will be fairly insensitive to the actual choice of γ. This is because the probability distributions are sharply peaked around x_- and x_+ (see Figs. 5–7).

As an example, we evaluate the performance of a NAND multiplexing system with input distribution $\vec{\pi}_0$ whose only nonzero element is $\vec{\pi}_0(0.9N) = 1$, and evaluate the output error probability P_e using (16) and (17). The output error probabilities versus the number of multiplexing stages are shown in Fig. 8 for gate error probabilities $\epsilon = 0.0001, \epsilon = 0.001$, and $\epsilon = 0.01$, while the bundle size is fixed at $N = 100$. It can be observed from Fig. 8 that for up to 100 stages, P_e is on the order of 10^{-7} when $\epsilon = 0.0001$ or 0.001. When $\epsilon = 0.01, P_e$ is on the order

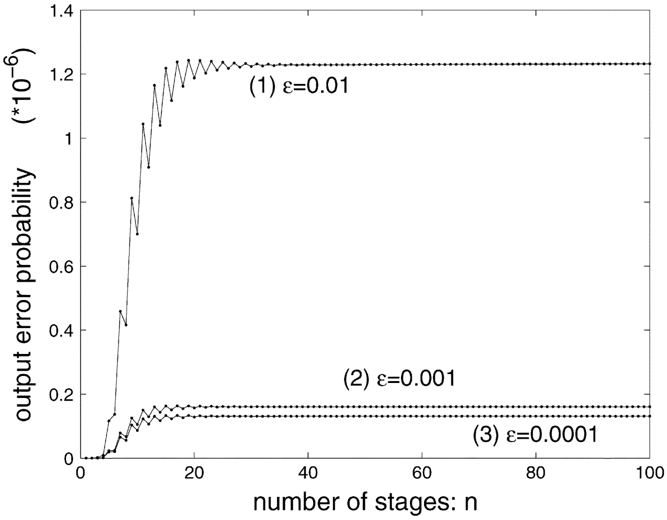


Fig. 8. Error probability of the output bundle versus the number of multiplexing stages. Initial distribution is $\vec{\pi}_0 = [0, 0, \ldots, 1, \ldots, 0, 0]$ where the only nonzero element is $\vec{\pi}_0(0.9N) = 1$. Bundle size is $N = 100$. Output error probability is defined by (16) and (17), where the decision threshold γ is chosen to be x_0 given by (3).

TABLE I
OUTPUT ERROR PROBABILITY FOR A SYSTEM WITH TEN MULTIPLEXING STAGES. ASSUME THAT THE GATE ERROR PROBABILITY IS $\epsilon = 0.01$

Bundle size N	100	200	300	500
P_e	7×10^{-7}	6×10^{-12}	6×10^{-17}	6×10^{-27}

of 10^{-6}. In addition, we have also found that for $\epsilon = 0.03$, P_e is on the order of 10^{-4}. These results show that when $\epsilon \le 0.01$, reasonably small output error probabilities may be obtained. In order to understand the influence of redundancy on the output error probability, we examine P_e for several different bundle sizes while the gate error probability is fixed at $\epsilon = 0.01$. The results are summarized in Table I. These results fit in exactly with the intuition that the reliability of a multiplexed system can be improved by increasing the bundle size or decreasing the gate error probability. The decision thresholds for the three gate error probabilities analyzed are $x_0 = 0.6180, x_0 = 0.6179$, and $x_0 = 0.6170$, respectively. As we have remarked, any thresholds in the vicinity of x_0 can be conveniently chosen to achieve similarly small output error probabilities.

We are now at a good standing to understand what has been observed by Han and Jonker [10]. Assuming that 90% of the inputs to the system are stimulated, they have evaluated the output reliability for odd stages, which is defined as the probability that no more than $\delta = 10\%$ of the wires in a bundle is stimulated, i.e.,

$$R_n = \sum_{i=0}^{\delta=0.1N} \vec{\pi}_n(i) \tag{18}$$

where n is an odd number. They have evaluated R_n for a system with $\epsilon = 10^{-5}$ for approximately 20 stages from which they suggest that the reliability improves as the number of multiplexing stages and the bundle size increase. This result can be readily explained by the transient and stationary behavior of the Markov chain. With the initial input distribution they have specified, the output distribution for the initial few stages is a uni-modal curve similar to the dotted curve in Fig. 7(a). As the number of stages grows, the center of this mode moves toward x_-, while another mode develops around x_+. However, when the number of stages is small, the probability "leaking" from the mode on the left-hand side to the one on the right-hand side is almost negligible. Evidently, R_n is close to zero when $\delta < x_-$, while it is close to one when $\delta > x_-$. For the case they have studied, $x_- = 0.003 < \delta = 0.1$, which has led to high reliability for the initial few stages. However, the same reasoning also suggests that, as far as only the initial few stages are concerned, R_n can be improved by using a δ larger than 0.1. This reasoning can also help us understand another experiment Han and Jonker have carried out about R_n. By analyzing R_n versus ϵ for a system with seven stages, they have observed that the output reliability decreases as ϵ increases. When ϵ reaches 0.1, the output reliability approaches zero and the multiplexing technique fails. This result is not surprising noticing that x_- grows with ϵ, while δ is a fixed value. When $\epsilon = 0.1$, it has exceeded the error threshold ϵ_* and the mode of the output distribution no longer peaks around x_-. Instead, the output distribution remains uni-modal for an arbitrary number of stages and the center of the mode moves toward x_0 as n grows. Thus, R_n drops sharply when ϵ reaches 0.1.

V. APPLICATION

The most direct implication of our analysis is that the multistage NAND multiplexing architecture can be used to design fault tolerant nanoelectronic systems. One important observation is that all computation and interpretation of the results must be carried out in a probabilistic sense. With the maturing of nanoscale devices, we have many promising candidates for the construction of the NAND gate and the multiplexing system, such as single electron transistors, quantum cellular automata, and carbon nanotube transistors [18]–[21]. Specific architectures for nanochips are still in their infancy, but conceptually, it is most likely that the nanocomputers will be comprised of massively distributed simple processing elements that communicate via local interconnect. Examples are the self-assembled crossbar-like systems described in [22]–[24]. These grid structures consist of intersecting nanowires that host programmable molecular switches at each node. Such grids of programmable switches allow for arbitrary connectivity between inputs and outputs of the crossbar. Thus, field-programmable gate-array (FPGA)-like structures have been proposed for digital implementation where computation is carried out by the programmable logic blocks built out of nanogrids [22], [23]. Within such programmable logic blocks, various gates such as NAND, NOR, AND, OR, and circuitry based on these can be implemented by properly configuring the switches. Grid structures comprised of such basic blocks can provide universal logic functionality with all logic and signal restoration operating at the molecular scale. In particular, grid structures are appropriate for the implementation of a NAND multiplexing system.

In the array-based architectures mentioned above, consideration of defect tolerance has been mainly targeted toward permanent defects, which are inevitable due to the stochastic nature of chemically assembled electronic nanotechnology (CAEN). The solutions proposed thus far mainly incorporate a reconfiguration

TABLE II

RELIABILITY ESTIMATES OF FUTURE NANOCHIPS. TYPE I: SYSTEMS WITHOUT SPECIAL RESTORATIVE UNITS. TYPE II: SYSTEMS WITH RESTORATIVE UNITS. d_p: PROCESSOR DENSITY; P_e: BIT ERROR PROBABILITY OF THE PROCESSOR; R_c: CHIP RELIABILITY

Type	devices/processor	N	stages/processor	d_p	P_e	R_c
I	8×10^4	200	10	$1.25 \times 10^6/cm^2$	6×10^{-12}	$1 - 7.5 \times 10^{-5}$
	$8 \times 10^4 \times \frac{300}{200}$	300	10	$8.33 \times 10^5/cm^2$	6×10^{-17}	$1 - 9.2 \times 10^{-10}$
II	$8 \times 10^4 \times 3$	200	3×10	$4.17 \times 10^5/cm^2$	1.49×10^{-11}	$1 - 6.2 \times 10^{-5}$
	$8 \times 10^4 \times 3 \times \frac{300}{200}$	300	3×10	$2.78 \times 10^5/cm^2$	2×10^{-16}	$1 - 6.2 \times 10^{-10}$

scheme. The method in [23] is to first configure the nanogrids for self-diagnosis and then reconfigure the system around the defects and implement the desired functionality. In [22], DeHon has proposed to use local wire sparing and array sparing so that manufacturing defects can be routed around through testing and reconfiguration. However, run-time/transient errors cannot be eliminated this way. The multiplexing technique with which all computations are carried out in terms of probabilities is appropriate for dealing with transient errors. We show by the below examples that the NAND-multiplexing technique can lead to high system reliability in spite of large gate error probability while the cost of redundancy is kept moderate. Note that a system based on von Neumann's original construction requires each faulty gate in a conventional circuit to be replaced by a "computation node," which is comprised of three NAND multiplexing units (see Figs. 1 and 2). Hence, such a system has a redundancy factor $M = 3 \times N$. However, in a construction considered by Han and Jonker, the restorative mechanism is achieved by the successive multiplexing units, thus special restorative units are removed from the system. In this case, the redundancy factor M is reduced to N. In the analysis that follows, we shall evaluate the system reliabilities of both constructions.

First, let us analyze a system without special restorative units. Revisiting the example considered in [10], we assume that device density of 10^{12}/cm^2 can be achieved for the nanochips built upon the grid structure. For each processor, the assumed word length is $l = 10$ bits and, for each bit, 40 logic devices are used. We shall use $N = 200$ for the multiplexing technique. In each processor, there are then $10 \times 40 \times 200 = 8 \times 10^4$ devices. The logic depth for the processor is assumed to be ten, hence, there are ten multiplexing stages, which both execute logic functions and restore the output excitation level. With the assumption that 10% of the devices are used for processors while the rest are for data storage and communication, the processor density of the anticipated nanochips is $d_p = (10^{12} \times 10\%)/(8 \times 10^4) = 1.25 \times 10^6$/cm^2. The probability that the processor with l-bit output works reliably is

$$R_p = (1 - P_e)^l \tag{19}$$

where P_e denotes the 1-bit output error probability of the processor. In the most stringent scenario, where all processors are required to work reliably, the reliability of the whole chip R_c is given by

$$R_c = R_p^{d_p}. \tag{20}$$

According to Table I, $P_e = 6 \times 10^{-12}$ when $\epsilon = 0.01, N = 200$ and the number of multiplexing stages is ten. Thus, $R_c = 1 - 7.5 \times 10^{-5}$. Note that with the same specifications for chip device density, word length, circuit logic depth, and number of devices per bit, Han and Jonker obtained a rather pessimistic estimate of $R_c \simeq 0.9$ when $N = 250$ and $\epsilon \sim 10^{-5}$. When we increase N to 300, R_c is further improved to 1-9.2 $\times$ 10^{-10}.

In a similar fashion, the chip reliability can be computed for a system based on von Neumann's original scheme (with two NAND multiplexing units acting as a restorative unit for each step of logic computation). With the same assumptions for device density, word length of processors and the number of devices per bit, there are now $10 \times 40 \times 200 \times 3 = 2.4 \times 10^5$ devices in each processor, and the number of NAND multiplexing stages is increased to 3×10 for a logic depth of ten. With the same assumption that 10% of the devices are allocated for processors, the processor density d_p is decreased to $(10^{12} \times 10\%)/(3 \times 8 \times 10^4) = 4.17 \times 10^5$/cm^2, due to the increase of redundancy inside each processor. With bundle sizes $N = 200$ and $N = 300$, the chip reliabilities according to (19) and (20) are $R_c = 1 - 6.2 \times 10^{-5}$ and $R_c = 1 - 6.2 \times 10^{-10}$, respectively. The parameter specifications and the reliabilities are summarized in Table II for the two systems analyzed above.

From the discussions above, it is clear that nanoelectronic systems based on NAND multiplexing may achieve reliabilities high enough for practical implementation. If self-diagnosis and reconfiguration can be performed before customizing such multiplexed systems, permanent errors may be eliminated at the initial phase and high overall fault tolerance may be achieved.

VI. CONCLUSION AND DISCUSSION

In this paper, we have studied two fundamental questions that lie at the heart of the design task for fault tolerant nanocomputer architectures based on NAND multiplexing, which are: 1) How does the behavior of a fault tolerant system depend on the error probability of potentially faulty individual components? and 2) What are the general mathematical frameworks to describe such computation schemes? We have obtained interesting insights into these two questions by studying the system architecture originally proposed by von Neumann and recently extended to a low degree of redundancy by Han and Jonker. The system uses multiple stages of NAND multiplexing units, which can be modeled as a Markov chain. Our analysis reveals that the system outputs always converge to one stationary distribution $\vec{\pi}_*$ when the number of stages n is sufficiently large. When the gate error ϵ is smaller than a threshold value $\epsilon_* = 0.08856\ldots, \vec{\pi}_*$ is bi-modal with its two modes representing logic "0" and "1," respectively. When ϵ exceeds ϵ_*, $\vec{\pi}_*$ is uni-modal and the two logic states are indistinguishable. From bifurcation analysis for individual NAND gates, we find that when $\epsilon > \epsilon_*$, the gates no longer perform the NAND function, thus, no reliable computation can proceed at the system level. Through analysis of the transient behavior of the system, we have observed that the speed at which the output distribution converges to the stationary distribution is essentially determined by the magnitudes of the eigenvalues of the transition matrix. The output error probability P_e (for a bundle or bit) can be characterized as the probability that "less/more than a portion of x_0 wires in the output bundle is excited," depending on the initial input and the parity of n. For microprocessors with a logic depth constrained to a few hundreds, we can achieve P_e on the order of 10^{-6} for $\epsilon = 0.01$ and $N = 100$. With the assumption that future nanochips will be comprised of massively parallel simple processing elements, we have studied the reliability of future nanochips based on two different NAND multiplexing constructions. Study of both cases shows that with a device failure rate on the order of 10^{-2}, nanoelectronic systems using the NAND multiplexing technique might still achieve reliabilities high enough for practical implementation while the redundancy factor is kept moderate.

As a general scheme, the multiplexing technique can be designed not only for the two-input NAND gate, but also for other gates such as the three-input majority gate. To analyze such multiplexing schemes, first one has to characterize the relation between the input and output bundles of one multiplexing stage. This was originally accomplished by combinatorial analysis. The combinatorial formulas for the two-input NAND gate were proposed by von Neumann [5]. These formulas provide precise results for deterministic input excitation levels and are computationally affordable for systems of small size. However, different approaches are needed for very large systems with stochastic input excitation levels. While extensions of combinatorial analysis can be considered [13], [25], particularly promising are stochastic modeling approaches where the excitation levels of all signal-carrying bundles are characterized by probability distributions and the multistage system is modeled as a Markov chain [10]–[13]. For those models to be useful, the conditions for good approximation must be carefully examined and the correctness of approximations under these conditions must be evaluated. For this purpose, Monte Carlo simulations have recently been carried out by Sadek *et al.*, where good agreement has been found between the results computed with the Markov chain model and those obtained by Monte Carlo simulations [13]. With the stochastic approximations well justified, the Markov-chain model can be very handy for the analysis of large systems. As shown in this study, Markov-chain analysis is computationally efficient since it is easy to compute the stationary and transient distributions, which are the keys to understanding system dynamics and characterizing system reliability. Though our analysis is based on the basic multistage multiplexing system, the results are generic and can help analyze and understand more general schemes, such as the one in [13].

Among many open issues in developing better multiplexing schemes, one possible direction is to explore other gates that have better properties than the two-input NAND gate, e.g., in terms of fault-tolerant capacity or ease of implementation. The three-input majority gate is appealing in this sense. It has a larger gate error threshold of (1/6) [7], [8] and can be conveniently implemented with threshold logic gate (TLG) circuits [25]. It has been shown that TLG circuits are more powerful/efficient than classical Boolean circuits [26] and may be built from novel nanoscale devices. System-level study of multiplexing schemes based on such versatile gates other than the two-input NAND gate has yet to be explored, where the same technique developed in this study may be applied.

Though multiplexing techniques have been shown to be efficient in controlling transient errors, no conclusion has been drawn for permanent errors. In order to build nanocomputers that are immune to both transient and permanent faults, a hybrid design approach may be needed. One possible solution is to implement NAND multiplexing for processing elements at the finest level and adopt a hierarchical reconfiguration scheme for modules at higher levels [12], [13]. It has been shown by Han and Jonker that by combining reconfiguration and the NAND multiplexing scheme, a device error rate of 10^{-2} can be tolerated while the degree of redundancy can be made significantly lower than that with NAND multiplexing alone. The reliability of such hybrid systems may be further improved based on smaller bit error probabilities obtained through our analysis approach. Formal comparisons of various fault tolerant techniques and their hybrid versions need to be carried out, e.g., in terms of maximum tolerable device error rate and degree of redundancy, as shown in [2]–[4]. In order to verify the models and automatically evaluate the efficiency of various novel fault tolerant solutions for nanoelectronics, computer-aided design (CAD) tools such as probabilistic model checking have been developed [17], [27]. It would be helpful to integrate new modeling approaches and theoretical results such as those developed in our study into existing CAD tools for the evaluation and comparison of the performance of various fault tolerant schemes.

References

[1] J. D. Meindl, Q. Chen, and J. A. Davis, "Limits on silicon nanoelectronics for terascale integration," *Science*, vol. 293, pp. 2044–2049, Sept. 2001.

[2] K. Nikoliæ, A. Sadek, and M. Forshaw, "Architectures for reliable computing with unreliable nanodevices," in *Proc. IEEE Nanotechnol.*, Oct. 2001, pp. 254–259.

[3] ——, "Fault-tolerant techniques for nanocomputers," *Nanotechnology*, vol. 13, pp. 357–362, June 2002.

[4] M. Forshaw, K. Nikoliæ, and A. Sadek, "Fault-tolerant techniques for nanocomputers: Third year report," Univ. College London, London, U.K., Tech. Rep. MEL-ARI28667, 2001.

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

[6] N. Pippenger, "Reliable computation by formulas in the presence of noise," *IEEE Trans. Inform. Theory*, vol. 34, pp. 194–197, Mar. 1988.

[7] B. Hajek and T. Weller, "On the maximum tolerable noise for reliable computation by formulas," *IEEE Trans. Inform. Theory*, vol. 37, pp. 388–391, Mar. 1991.

[8] W. S. Evans, "Information theory and noisy computation," Ph.D. dissertation, Univ. California at Berkeley, Berkeley, CA, 1994. [Online]. Available: http://www.cs.ubc.ca/will/papers/thesis.pdf.

[9] W. Evans and N. Pippenger, "On the maximum tolerable noise for reliable computation by formulas," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1299–1305, May 1998.

[10] J. Han and P. Jonker, "A system architecture solution for unreliable nanoelectronic devices," *IEEE Trans. Nanotechnol.*, vol. 1, pp. 201–208, Dec. 2002.

[11] ——, "A fault-tolerant technique for nanocomputers: NAND multiplexing," in *Proc. ASCI 8th Annu. Advanced School for Computing and Imaging Conf.*, Lochem, The Netherlands, June 2002, pp. 59–66.

[12] ——, "A defect- and fault-tolerant architecture for nanocomputers," *Nanotechnology*, vol. 14, pp. 224–230, Jan. 2003.

[13] A. S. Sadek, K. Nikoliæ, and M. Forshaw, "Parallel information and computation with restitution for noise-tolerant nanoscale logic networks," *Nanotechnology*, vol. 15, pp. 192–210, Jan. 2004.

[14] S. H. Strogatz, *Nonlinear Dynamics and Chaos—With Applications to Physics, Biology, Chemistry, and Engineering*. Cambridge, MA: Westview Press, 2000.

[15] Y. Qi, J. Gao, and J. A. B. Fortes, "Probabilistic computation: A general framework for fault tolerant nanoelectronic systems," Adv. Comput. Inform. Processing Lab., Univ. Florida, Gainesville, FL, Tech. Rep. TR-ACIS-03-002, Nov. 2003. [Online]. Available: http://www.acis.ufl.edu/techreports/acis03002.pdf.

[16] E. Cinlar, *Introduction to Stochastic Processes*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[17] G. Norman, D. Parker, M. Kwiatkowska, and S. K. Shukla, "Evaluating the reliability of defect-tolerant architectures for nanotechnology with probabilistic model checking," in *Proc. Int. VLSI Design Conf.*, Mumbai, Maharashtra, India, Jan. 2004, pp. 907–914.

[18] K. Likharev, "Single-electron devices and their applications," *Proc. IEEE*, vol. 87, pp. 606–632, Apr. 1999.

[19] A. Bachtold, P. Hadley, T. Nakanishi, and C. Dekker, "Logic circuits with carbon nanotube transistors," *Science*, vol. 294, pp. 1317–1320, Nov. 2001.

[20] I. Amlani, A. O. Orlov, G. Toth, G. H. Bernstein, C. S. Lent, and G. L. Snider, "Digital logic gate using quantum-dot cellular automata," *Science*, vol. 284, pp. 289–291, Apr. 1999.

[21] M. T. Niemier, A. F. Rodrigues, and P. M. Kogge. A potentially implementable FPGA for quantum dot cellular automata. presented at 1st Non-Silicon Computation Workshop. [Online]. Available: http://www.cse.nd.edu/cse_proj/qca-design/papers/nsc02-rpga.pdf

[22] A. DeHon. Array-based architecture for molecular electronics. presented at 1st Non-Silicon Computation Workshop. [Online]. Available: http://www-2.cs.cmu.edu/phoenix/nscl/paper/4-1.pdf

[23] M. B. S. C. Goldstein, "Nanofabrics: Spatial computing using molecular electronics," in *Proc. 28th Annu. Int. Computer Architecture Symp.*, June 2001, pp. 178–191.

[24] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, "A defect-tolerant computer architecture: Opportunities for nanotechnology," *Science*, vol. 280, pp. 1716–1721, June 1998.

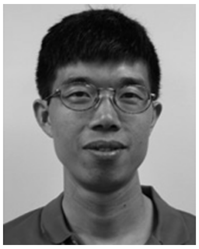
[25] S. Roy, V. Beiu, and M. Sulieman. Reliability analysis of some nano architectures. presented at NIPS Special Workshop. [Online]. Available: http://www.eecs.wsu.edu/vbeiu/workshop_nips03/Presentations/SJRoy.pdf

[26] R. Reischuk, "Can large fan-in circuits perform reliable computations in the presence of faults?," *Theor. Comput. Sci.*, vol. 240, pp. 319–335, June 2000.

[27] S. K. Shukla. Automated evaluation of reliability of defect tolerant nano architectures. presented at NIPS Special Workshop. [Online]. Available: http://www.eecs.wsu.edu/vbeiu/workshop_nips03/Presentations/SK_Shukla.pdf

Yan Qi received the B.Sc. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2002, the M.Sc. degree in electrical and computer engineering from the University of Florida, Gainesville in 2003, and is currently working toward the Ph.D. degree in biomedical engineering at The Johns Hopkins University, Baltimore, MD.

Her research interests include fault-tolerant computation, nanoelectronic system architectures, statistical modeling, and network dynamics analysis in computational biology and bioinformatics.



Jianbo Gao received the Ph.D. degree from the University of California at Los Angeles (UCLA), in 2000.

He is currently an Assistant Professor with the University of Florida, Gainesville. He has been involved in numerous fields including 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. Some of the tools he has developed have been applied to study a wide range of real-world signal-processing problems arising from diverse fields such as physics, geophysics, biology, economics, and engineering.

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 Professor and BellSouth Eminent Scholar. His research interests are in the areas of network computing, distributed information-processing systems, advanced computing architectures, and nanocomputing.

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