A System Architecture Solution for Unreliable Nanoelectronic Devices

Jie Han and Pieter Jonker

***Abstract*—Due to the manufacturing process, the shrinking of electronic devices will inevitably introduce a growing number of defects and even make these devices more sensitive to external influences. It is, therefore, likely that the emerging nanometer-scale devices will eventually suffer from more errors than classical silicon devices in large scale integrated circuits. In order to make systems based on nanometer-scale devices reliable, the design of fault-tolerant architectures will be necessary. Initiated by von Neumann, the NAND multiplexing technique, based on a massive duplication of imperfect devices and randomized imperfect interconnects, had been studied in the past using an extreme high degree of redundancy. In this paper, this NAND multiplexing is extended to a rather low degree of redundancy, and the stochastic Markov nature in the heart of the system is discovered and studied, leading to a comprehensive fault-tolerant theory. A system architecture based on NAND multiplexing is investigated by studying the problem of the random background charges in single electron tunneling (SET) circuits. Our evaluation shows that it might be a system solution for an ultra large integration of highly unreliable nanometer-scale devices.**

***Index Terms*—Computer architecture, fault tolerance, Markov processes, multiplexing, nanotechnology, stochastic system.**

I. Introduction

THIS PAPER presents an evaluation of the NAND multiplexing technique as originally introduced by von Neumann [1]. Our evaluation leads to the possibility to calculate optimal redundancies for nanoelectronic system designs, using statistical analysis of chains of stages, each of which contains many NAND circuits in parallel. Basically, a single NAND (or NOR) gate design is sufficient for the implementation of a complex digital computer. Currently, logic gates are made of reasonably reliable field effect transistor (FET) circuits, however, future logic circuits may be built up from less reliable devices, among which the single electron tunneling (SET) technology is one of the most visible circuits dawning. The shrinking of nanometer-scale devices will introduce more defects in the manufacturing process and make them more sensitive to external influences such as cosmic radiation, electromagnetic interference, thermal fluctuations, etc. Permanent faults may emerge during the manufacturing process, while transient ones may spontaneously occur during the computer's lifetime. It is, therefore, likely that the emerging nanometer-scale devices will eventually suffer from more errors than classical silicon devices in large scale integrated circuits. In order to make future systems based on nanometer-scale devices reliable, the design of fault-tolerant architectures will be necessary.

In 1952, von Neumann initiated the study of using redundant components to obtain reliable synthesis from unreliable components, namely, the multiplexing technique [1]. It was then theoretically demonstrated that with an extreme high degree of redundancy, the integration of unreliable logic units could be made reliable. In his construction, von Neumann considered two sets of basic logic, the Majority Voting and NAND logic, and assumed that they are not completely reliable, i.e., each of them fails with constant probability. By using a bundle of unreliable gates functioning as an ideally reliable one, von Neumann proved that if the failure probability of the gates are sufficiently small and the failures are statistically independent, computations may be done reliably with a high probability. However, the construction requires a large number of redundant components, which was seen as a major shortcoming of this method.

In 1977, Dobrushin and Ortyukov provided a rigorous proof to improve von Neumann's result [2], showing that logarithmic redundancy is actually sufficient for any Boolean function [2] and, at least for certain Boolean functions, necessary [3]. This argument was later strengthened by Pippenger, Stamoulis and Tsitsiklis [4]. In 1980s, Pippenger proved that a variety of Boolean functions may be computed reliably by noisy networks requiring only constant multiplicative redundancy [5], [6]. Furthermore, it was shown that von Neumann's construction works only when the failure probability per gate has a limit strictly smaller than 1/2, and that computations with failures due to noise proceed more slowly than in the absence of failures, since a fraction of the layers has to be devoted to correction [7], [8].

Current fault-tolerant techniques are basically built on redundancy technologies:

- N-tuple modular redundancy (NMR) [1] (e.g., triple modular redundancy or TMR [9]);
- reconfiguration [10], [11].

A reconfigurable architecture is a computer architecture which can be configured or programmed after fabrication to implement desired computations. Faulty components are detected during testing and excluded during reconfiguration. Reconfigurable computers have been successfully implemented for the protection against permanent failures, mainly generated during manufacturing, however, they are much less efficient in the protection against transient ones [12].

NMR systems use redundant components to mask the effect of faulty components. In TMR, which is the most general technique of NMR, three identical modules perform the same computation, and a voter accepts outputs from all three modules,

Manuscript received June 6, 2002; revised August 16, 2002. This work was supported by Delft University of Technology, The Netherlands, under its DIRC project "Novel Computation Structures Based on Quantum Devices."

The authors are with the Pattern Recognition Group, Faculty of Applied Sciences, Delft University of Technology, The Netherlands (e-mail: jie@ph.tn.tudelft.nl; pieter@ph.tn.tudelft.nl).

Digital Object Identifier 10.1109/TNANO.2002.807393

1536-125X/02$17.00 © 2002 IEEE

producing a majority vote at its output. With NMR the effect of modest transient errors are effectively eliminated, however some critical components (e.g., the Majority Voting logic in TMR) have to be highly reliable.

Since nanometer-scale devices will be much smaller than current CMOS devices, the device failure rate increases due to the limit of manufacturing and less amiable operating environments. The unreliability of devices is crucial in that in some cases it prevents promising nanometer-scale devices from being used in large-scale applications, such as the SET technology influenced by random background charges [13]. In this paper, we seek fault-tolerant architectures for unreliable nanoelectronic devices, by extending the study of von Neumann's NAND multiplexing to a rather low degree of redundancy. The problem of the random background charges in SET circuits is addressed to study a system architecture based on NAND multiplexing as a solution for the integration of unreliable nanometer-scale devices.

Within a digital computer, the bulk of the logic gates is spent on memory and caches. The processor itself is made from a number of functional units, each of which can be separated into function blocks. Let us assume that the function block on the most refined level evaluates its inputs and produces a stable output within one clock cycle. Within this function block, many logic circuits may be cascaded, however to avoid timing problems (hazard) usually the number of circuits cascaded and hence the possible paths from inputs to outputs through the various logic circuits is kept within bounds, and hence their path lengths are similar. Such function blocks are found everywhere in the processor and in memory. In this paper, we make an abstraction of such a function block and assume at first, to be able to make a statistical analysis, that it is made entirely out of n stages of N parallel NAND gates. In a design with unreliable logic, the upper bound is that we must replace each logic gate with $n \cdot N$ unreliable – hopefully much smaller – gates. However, we hope to prove in future work, that due to the logic design of the function block, we may end up with less redundancy.

The paper is organized as follows. Von Neumann's NAND multiplexing theory is briefly reviewed in Section II, and is extended to a rather low degree of redundancy in Section III. In Section IV we study the stochastic Markov nature of a multistage multiplexing system, and in Section V we give discussions. In Section VI the application of NAND multiplexing to be used in a nanoelectronic computer architecture is addressed. Section VII concludes the paper.

II. VON NEUMANN'S THEORY ON NAND MULTIPLEXING

A. A NAND Multiplexing Unit

Consider a NAND gate. Replace each input of the NAND gate as well as its output by a bundle of N lines, and duplicate the NAND N times, as shown in Fig. 1. The rectangle U is supposed to perform a "random permutation" of the input signals in the sense that each signal from the first input bundle is randomly paired with a signal from the second input bundle to form the input pair of one of the duplicated NAND's.

Let X be the set of lines in the first input bundle being stimulated (a logic TRUE or " 1 "). Consequently, $(N - X)$ lines are not stimulated (they have the value FALSE or " 0 "). Let Y be the corresponding set for the second input bundle; and let Z be the corresponding set for the output bundle.

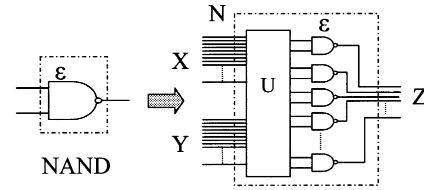

Fig. 1. A NAND multiplexing unit.

Assume that the failure probability of a NAND gate is a constant ε and assume that the type of fault the NAND makes is that it inverts its output; i.e., acts as an AND gate (a von Neumann fault). Let (X, Y, Z) have $(\bar{x} \cdot N, \bar{y} \cdot N, \bar{z} \cdot N)$ elements. Clearly $(\bar{x}, \bar{y}, \bar{z})$ are relative levels of excitation of the two input bundles and of the output bundle, respectively. The question is then: what is the distribution of the stochastic variable $\bar{z}$ in terms of the given $\bar{x}$ and $\bar{y}$?

With an extremely large N, von Neumann had concluded that $\bar{z}$ is a stochastic variable, approximately normally distributed [1]. He also gave an upper bound for the failure probability per gate that can be tolerated, $\varepsilon_0 = 0.0107$. Recently, it was shown that if each NAND gate fails independently, the tolerable threshold probability of each gate will be $\varepsilon_0 = (3 - \sqrt{7})/4 = 0.088\,56 \cdots$[14] (although this result is obtained by formulas constructed from noisy NAND gates rather than circuits). In other words, according to von Neumann, if $\varepsilon \geq \varepsilon_0$, the failure probability of the NAND multiplexing network will be larger than a fixed, positive lower bound, no matter how large a bundle size N is used.

B. The Restorative Unit

If we assume that the two input bundles have almost the same stimulated or nonstimulated levels (which is likely in circuits), it is then intuitively known that:

- if almost all lines of one input bundle are stimulated and almost all lines of the other bundle are nonstimulated, then the error probability of the output bundle (NAND; hence, the probability of the number of lines that are nonstimulated) will approximately be the same as the error probability in either one of the input bundles;
- if almost all lines of both input bundles are nonstimulated, then the error probability of the output bundle (NAND; hence the probability of the number of lines that are nonstimulated) will be smaller than the error probability in either one of the input bundles;
- if almost all lines of both input bundles are stimulated, then the error probability of the output bundle (NAND; hence the probability of the number of lines that are stimulated) will be larger than the error probability in either one of the input bundles.

For this last case, we need a unit that restores the original stimulation level without destroying the NAND function.

Von Neumann had built a multiplexing system with two types of units, the first being the executive unit, which performs the NAND function and the second a restorative unit which annuls the degradation caused by the first one [1]. The restorative unit

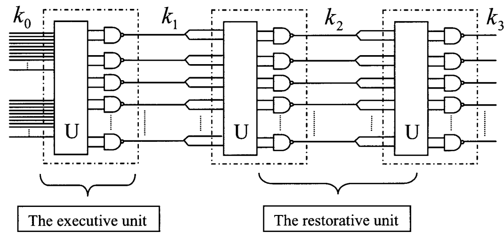

Fig. 2. NAND multiplexing system.

was made by using the same NAND multiplexing technique while duplicating the outputs of the executive unit as the inputs. To keep the NAND function, the multiplexing unit was iterated to give the effective restoring mechanism [1], see Fig. 2.

III. Error Distributions in a Multiplexing Unit

A. An Alternative Method

The NAND multiplexing unit was constructed as Fig. 1. In this section an alternative method is given to extend the study of the NAND multiplexing technique from an extreme high degree to a rather low degree of redundancy.

Let us consider a single NAND gate in the NAND multiplexing scheme. We still assume that there are $\bar{x}N$ and $\bar{y}N$ input lines stimulated. If the error probabilities in the two input lines are independent, the probability of the output of the NAND gate that is found stimulated (by at least one nonstimulated input) is $\bar{z}' = 1 - \bar{x}\bar{y}$ (assuming that the NAND gate is fault-free). If each NAND gate has a probability ε of making a (von Neumann) error, the probability of its output being stimulated is:

$$\bar{z} = (1 - \bar{x}\bar{y}) + \varepsilon(2\bar{x}\bar{y} - 1). \tag{1}$$

For other types of faults (such as fault models Stuck-at-0 and Stuck-at-1) $\bar{z}$ has slightly different appearance, however, at first it is reasonable to take the von Neumann model as representative.

For each NAND gate, thus, the probability of the output to be stimulated (event 1) is $\bar{z}$ and the probability to be nonstimulated (event 0) is $1 - \bar{z}$. If the N NAND gates function independently, the entire NAND multiplexing unit constitutes a Bernoulli sequence. The distribution of the probability of stimulated outputs is, therefore, the binomial distribution. The probability of exactly k outputs being stimulated is then

$$P(k) = \binom{N}{k} \bar{z}^k (1 - \bar{z})^{N-k}. \tag{2}$$

When N is extremely large and $\bar{z}$ is extremely small, the Poisson Theorem gives us

$$P(k) \doteq \lim_{N\to\infty} \binom{N}{k} \bar{z}^k (1 - \bar{z})^{N-k} = \frac{\bar{\lambda}^k e^{-\bar{\lambda}}}{k!} \tag{3}$$

where

$$\bar{\lambda} = N\bar{z}. \tag{4}$$

Given N extremely large and $\bar{z}$ extremely small, therefore, the distribution of probability of exactly k outputs from the N output lines of the NAND multiplexing unit being stimulated could be approximately a Poisson distribution.

If both inputs of the NAND gates are expected to be in stimulated states, the stimulated outputs are then considered as faulty ones. To evaluate the effect of faults, the probability of possible errors below an acceptable threshold level, i.e., $P(k \le x)$, needs to be computed. Since the number of the stimulated outputs is a stochastic variable, which comply with the binomial distribution, the De Moivre-Laplace Theorem [15], when N is extremely large and $0 < \bar{z} < 1$, applies

$$\lim_{N\to\infty} P\left\{ \frac{k - N\bar{z}}{\sqrt{N\bar{z}(1 - \bar{z})}} \le x' \right\} = \int_{-\infty}^{x'} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt \tag{5}$$

replacing

$$x' = \frac{x - N\bar{z}}{\sqrt{N\bar{z}(1 - \bar{z})}} \tag{6}$$

then

$$P(k \le x) \doteq \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sqrt{N\bar{z}(1 - \bar{z})}} e^{-1/2\left(t - N\bar{z}/\sqrt{N\bar{z}(1-\bar{z})}\right)^2} dt. \tag{7}$$

The probability density of k can be obtained now as:

$$f(k) = \frac{1}{\sqrt{2\pi}\sqrt{N\bar{z}(1 - \bar{z})}} e^{-1/2\left((k - N\bar{z})/\sqrt{N\bar{z}(1-\bar{z})}\right)^2}. \tag{8}$$

This shows that the probability of the number of stimulated outputs (event 1) of the NAND multiplexing unit could be approximated by a normal distribution with mean $N\bar{z}$ and standard deviation $\sqrt{N\bar{z}(1 - \bar{z})}$, when N is extremely large and $0 < \bar{z} < 1$.

B. Numerical Evaluation

Consider next the fault distribution of the NAND multiplexing unit for different N and ε within certain ranges. We assume that the largest possible error rate ε for a future nanoelectronic system is 0.1 , meaning that one of ten devices is faulty on average. Consequently, the ε under investigation will be in the range of [0,0.1]. We further assume that the input excitation rates are identical to each other, i.e., $\bar{x} == \bar{y}$. This is often true for circuits using similar devices. Hence, the fault probability of one output of the NAND multiplexing unit, i.e., the probability of an output line being stimulated, becomes

$$\bar{z} = (1 - \bar{x}^2) + \varepsilon(2\bar{x}^2 - 1). \tag{9}$$

For simplicity, we assume $\bar{x}' = 1 - \bar{x}$. Replacing $\bar{x}$ with $\bar{x}'$ in (9)

$$\bar{z} = (2\varepsilon - 1)\bar{x}'^2 + 2(1 - 2\varepsilon)\bar{x}' + \varepsilon. \tag{10}$$

For $\varepsilon \in [0, 0.1]$, the formula (10) is monotonically-increasing as $\bar{x}'$ varies from 0 to 0.5. For a typical $\bar{x}'$, say, 0.1, $\bar{z} \in [0.19, 0.25]$. This condition does not favor a conclusion in the direction of a Poisson distribution.

We proceed with a study on the approximation of a Poisson and a Normal distribution to the binomial distribution for different sizes of the NAND multiplexing unit, i.e., for different N. We first take $N = 1000$. Specifying $\bar{x} = 0.8$ and $\varepsilon = 10^{-4}$,

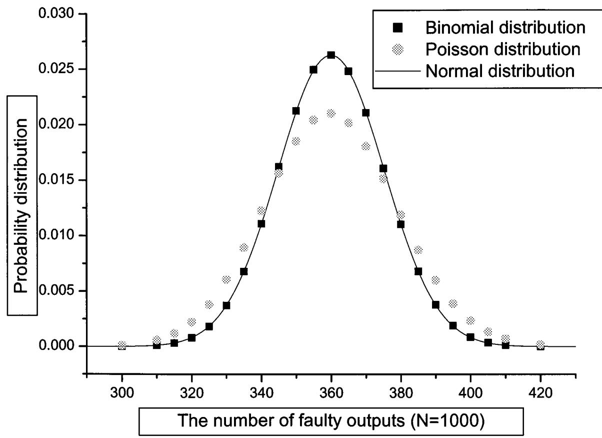

Fig. 3. The fault distributions ($N = 1000, \varepsilon = 10^{-4}$).

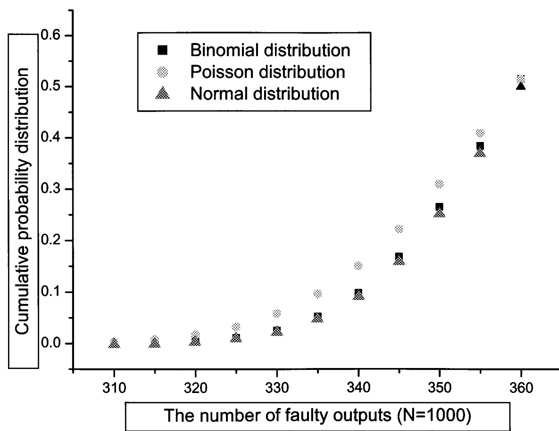

Fig. 4. The cumulative fault distributions ($N = 1000, \varepsilon = 10^{-4}$).

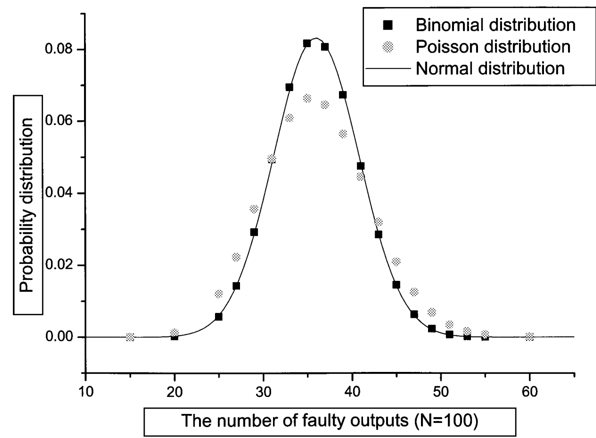

Fig. 5. The fault distributions ($N = 100, \varepsilon = 10^{-4}$).

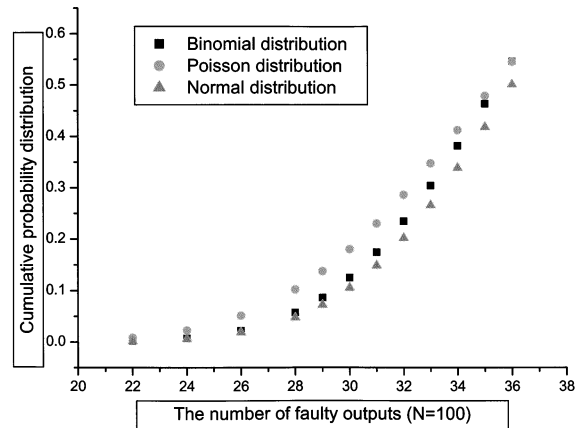

Fig. 6. The cumulative fault distributions ($N = 100, \varepsilon = 10^{-4}$).

the probability (density) of the binomial, Poisson, and Normal distribution against the number of faulty outputs are plotted in Fig. 3. As the probability of possible errors below an acceptable threshold level $P(k \leq x)$ is an important feature to evaluate the approximations, the cumulative probability distribution $P(k \leq x)$ for the binomial, Poisson and Normal distribution are plotted as well, in Fig. 4. We can see that in both figures the Normal distribution is in good accordance with the binomial distribution, while the Poisson distribution is not. The approximation for the Normal distribution is very well kept when $\bar{x}$ varies in the range [0.7,0.9] and ε varies in the range [0,0.1]. Obviously, the larger N is, the better the approximation. If N is large enough, the error probability of the NAND multiplexing unit is approximately normally distributed.

Now consider the case that $N = 100$. We still set $\bar{x} = 0.8$ and $\varepsilon = 10^{-4}$. The fault probability and cumulative distributions are shown in Figs. 5 and 6, respectively. In Fig. 5, the probability density of the Normal distribution fits in quite well with the samples of the binomial distribution. As the total samples N is not so large here, on the other hand, the discrete binomial distribution is no longer appropriately described by the Normal distribution. Therefore, as shown in Fig. 6, neither Normal nor Poisson gives good approximation to the binomial for the cumulative distribution.

Therefore, in terms of the probability (density) and the cumulative probability distribution, the error probability of the NAND multiplexing unit can be approximated by the Normal distribution when $N > 1000$.

IV. Error Distributions in a Multistage System

A. For Modest $N(N < 1000)$

We have discussed the set-up of a NAND multiplexing system as depicted in Fig. 2, which executes multiple NAND operations in parallel. If there are k_0 of the N incoming lines stimulated for both inputs of the executive unit in the NAND multiplexing system, and each NAND gate has a definite probability ε of making a (von Neumann) error, according to (1) and (2), the probabilities of the stimulated outputs k_1, k_2, and k_3 of the three multiplexing units in cases of the corresponding stimulated inputs k_0, k_1, and k_2 are given by

$$P_1\left(k_1|k_0\right) = \binom{N}{k_1} \bar{z}_1^{k_1}(k_0)(1 - \bar{z}_1(k_0))^{N-k_1} \tag{11}$$

$$P_2\left(k_2|k_1\right) = \binom{N}{k_2} \bar{z}_2^{k_2}(k_1)(1 - \bar{z}_2(k_1))^{N-k_2} \tag{12}$$

$$P_3\left(k_3|k_2\right) = \binom{N}{k_3} \bar{z}_3^{k_3}(k_2)(1 - \bar{z}_3(k_2))^{N-k_3} \tag{13}$$

where

$$\bar{z}_1(k_0) = (1-\varepsilon) - (1-2\varepsilon)\left(\frac{k_0}{N}\right)^2 \quad (14)$$

$$\bar{z}_2(k_1) = (1-\varepsilon) - (1-2\varepsilon)\left(\frac{k_1}{N}\right)^2 \quad (15)$$

$$\bar{z}_3(k_2) = (1-\varepsilon) - (1-2\varepsilon)\left(\frac{k_2}{N}\right)^2. \quad (16)$$

Noting the stochastic nature of k_1, k_2, and k_3, the probabilities of them being stimulated in all cases are then obtained by

$$P_1(k_1) = \sum_{k_0=0}^{N} P_1(k_1|k_0)\, P_1(k_0) \quad (17)$$

$$P_2(k_2) = \sum_{k_1=0}^{N} P_2(k_2|k_1)\, P_1(k_1) \quad (18)$$

$$P_3(k_3) = \sum_{k_2=0}^{N} P_3(k_3|k_2)\, P_2(k_2). \quad (19)$$

In (17)-(19) the most significant parts are the conditional probabilities, $P_3(k_3|k_2)$, $P_2(k_2|k_1)$ and $P_1(k_1|k_0)$ (which is $P_1(k_1)$ with fixed k_0). For any identical set of inputs and outputs, all the three conditional probabilities are the binomial distribution with identical parameters, i.e.

$$P(k_l|k_{l-1}) = \binom{N}{k_l} \bar{z}^{k_l}(k_{l-1})(1-\bar{z}(k_{l-1}))^{N-k_l} \quad (20)$$

where

$$\bar{z}(k_{l-1}) = (1-\varepsilon) - (1-2\varepsilon)\left(\frac{k_{l-1}}{N}\right)^2. \quad (21)$$

Therefore, a $(N+1)\times(N+1)$ matrix $\mathbf{\Psi}$, whose elements are $P(k_l|k_{l-1})$, $k_l, k_{l-1} \in [0,1,2,\dots N]$, can be made as shown in (22), so that all conditional probabilities for any set of (k_l, k_{l-1}) are included

$$\mathbf{\Psi} = \begin{matrix} P(0|0) & P(1|0) & P(2|0) & \dots & P(N|0) \\ P(0|1) & P(1|1) & P(2|1) & \dots & P(N|1) \\ P(0|2) & P(1|2) & P(2|2) & \dots & P(N|2) \\ \dots & \dots & \dots & \dots & \dots \\ P(0|N) & P(1|N) & P(2|N) & \dots & P(N|N) \end{matrix}. \quad (22)$$

Accordingly, given a fixed input distribution

$$\mathbf{P}_0 = [p_0, p_1, p_2 \dots p_N] \quad (23)$$

where p_i is the probability of i inputs being stimulated, the stimulated output distributions of (17), (18) and (19) are given by

$$\mathbf{P}_1 = [P_1(0), P_1(1), \dots P_1(N)] = \mathbf{P}_0\mathbf{\Psi} \quad (24)$$

$$\mathbf{P}_2 = [P_2(0), P_2(1), \dots P_2(N)] = \mathbf{P}_0\mathbf{\Psi}^2 \quad (25)$$

$$\mathbf{P}_3 = [P_3(0), P_3(1), \dots P_3(N)] = \mathbf{P}_0\mathbf{\Psi}^3. \quad (26)$$

B. A Stochastic Markov Chain

The number of stimulated outputs of each NAND multiplexing stage is actually a stochastic variable and its state space is $A = [0,1,2,\dots N-1, N]$. If we name this variable $\bar{\xi}_n$, where n is the index of the multiplexing unit, the evolution of $\bar{\xi}_n$ in the NAND multiplexing system is a stochastic process. With fixed N and ε, the distribution of $\bar{\xi}_n$ for every n is totally determined by the number of stimulated inputs of the nth multiplexing unit. This can be mathematically described by

$$P\left(\bar{\xi}_n \in A | \bar{\xi}_1 = k_1, \bar{\xi}_2 = k_2, \dots \bar{\xi}_{n-1} = k_{n-1}\right) = P\left(\bar{\xi}_n \in A | \bar{\xi}_{n-1} = k_{n-1}\right). \quad (27)$$

Equation (27) is the condition for a stochastic process to be a Markov process. The evolution of $\bar{\xi}_n$ in the NAND multiplexing system, therefore, is a Markov process, or a Markov chain for discrete states and parameters.

In a stochastic Markov chain, the transition probability, which indicates the conditional probability from one specified state to another, is the most significant factor. Since the transition probability matrix $\mathbf{\Psi}$ for each $\bar{\xi}_n$ is identical and irrelevant with regard to n, $\bar{\xi}_n$ evolves as a homogeneous Markov chain. Therefore, an initial probability distribution and a transition probability matrix as (22) are sufficient to get all output distributions.

If a NAND multiplexing system has n individual stages in series and its transition probability matrix is given by (22), the output distribution of it is then

$$\mathbf{P}_n = \mathbf{P}_0\mathbf{\Psi}^n. \quad (28)$$

The NAND multiplexing system with one executive and two restorative stages can be described as three stochastic variables $\bar{\xi}_1$, $\bar{\xi}_2$ and $\bar{\xi}_3$. In principle a system with arbitrary number of NAND multiplexing stages, say, $n = 5, 7, 9, \dots$, can be built (note that the odd number is necessary to keep the NAND function). When n gets large, $\mathbf{\Psi}^n$ approaches a constant matrix $\boldsymbol{\pi}$, i.e.

$$\lim_{n\to\infty} \mathbf{\Psi}^n = \boldsymbol{\pi}. \quad (29)$$

Each row of $\boldsymbol{\pi}$ is identical. This indicates that, as n becomes extremely large, not only the transition probabilities in a NAND multiplexing system will get stable, but also the output distribution will become stable and independent of the number of multiplexing stages.

C. N is Rather Large ($N > 1000$)

If N is rather large (> 1000), the output error of each NAND multiplexing stage is approximately normally distributed. If for the lth multiplexing stage there are k_{l-1} stimulated inputs and accordingly k_l stimulated outputs, according to (8) the probability density of k_l is given by

$$f(k_l|k_{l-1}) = \frac{1}{\sqrt{2\pi}s(k_{l-1})} e^{-1/2(k_l - Nz(k_{l-1})/s(k_{l-1}))^2} \quad (30)$$

where

$$s(k_{l-1}) = \sqrt{Nz(k_{l-1})(1-z(k_{l-1}))} \quad (31)$$

$$z(k_{l-1}) = (1-\varepsilon) - (1-2\varepsilon)\left(\frac{k_{l-1}}{N}\right)^2. \quad (32)$$

Then the probability of the multiplexing stage having k_l stimulated outputs under the condition of k_{l-1} inputs is approximately

$$P(k_l|k_{l-1}) = f(k_l|k_{l-1})\triangle k, \ldots \triangle k \sim 1. \quad (33)$$

The probability of k_l outputs being stimulated in all cases for $0 \leq k_{l-1} \leq N$ is then

$$P(k_l) = \sum_{k_{l-1}=0}^{N} P(k_l|k_{l-1}) P(k_{l-1}). \quad (34)$$

Replacing

$$P(k_l) = f(k_l)\triangle k \quad (35)$$

and

$$P(k_{l-1}) = f(k_{l-1})\triangle k \quad (36)$$

we have in all cases that the probability density of k_l outputs being stimulated is

$$f(k_l) = \sum_{k_{l-1}=0}^{N} f(k_l|k_{l-1}) f(k_{l-1})\triangle k. \quad (37)$$

In the limit we obtain

$$f(k_l) = \int_0^N f(k_l|k_{l-1}) f(k_{l-1}) dk. \quad (38)$$

Equation (38) is an inductive expression, from which conclusions on the outputs of any NAND multiplexing system can be derived from its initial inputs. As the number of NAND multiplexing stages increase, however, it becomes extremely hard to be computed. A practical way is to use the mean of the previous outputs as the fixed inputs of the successive stage.

If, for example, there is a NAND multiplexing system with $N = 1000$ and $\varepsilon = 10^{-5}$, given that 90% of the initial inputs are stimulated, the stimulated outputs are approximately normally distributed, with a mean of 71 and a standard deviation of 8.

V. Discussion

We now study the fault tolerance of a NAND multiplexing system while we vary the I/O bundle sizes. It might be interesting to evaluate the performance of a NAND multiplexing system with $\varepsilon = 10^{-5}$ and 90% of its inputs stimulated, and the probability that no more than 10% of its outputs is stimulated. A system with more restorative stages is investigated as well. The probability distributions versus the number of multiplexing stages are shown in Fig. 7 for different bundle sizes $N = 10$, $N = 100$, and $N = 1000$. Let us take an example with $N = 100$. The probability that less than 10% of the outputs is faulty (stimulated) is approximately 0.70 in a 3-stage system while this is 0.99 in a 7-stage system. As the number of multiplexing stages increases, it shows that the reliability of the signals greatly improves, but, on the other hand, the rate of the improvement is getting smaller.

If we pick the number of multiplexing stages to be $n = 7$, then the system has a good performance while the required redundancy ($7N$) is not too high. The fault tolerance of the system for a varying number of error rates ε of the NAND circuits can be studied in this specific case. In Fig. 8 the probability distribution of errors less than 10% are drawn against the error rate of an individual NAND gate, with $n = 7$. It is obvious that the NAND multiplexing system has a better fault tolerance when the bundle size N grows. The tradeoff, however, has to be made between performance and redundancy. Another conclusion is that the NAND multiplexing technique hardly works when the error rate of basic logic devices approaches 0.1 (this value is $0.088\,56\cdots$ in [14]).

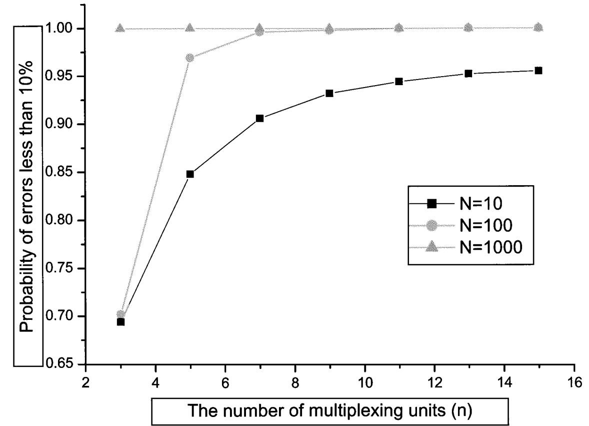

Fig. 7. Error distribution versus number of stages.

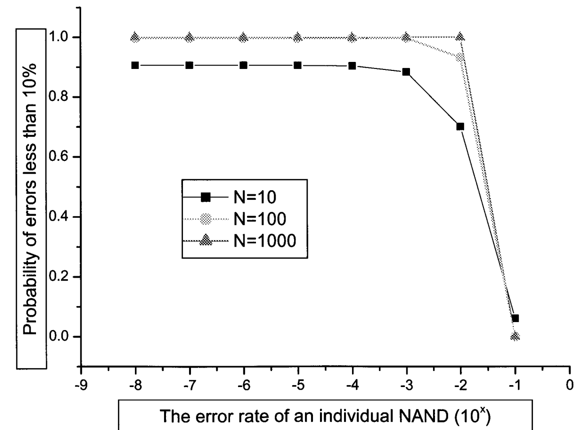

Fig. 8. Error distribution versus error rate of a NAND.

VI. Application

To give an example of how the suggested fault-tolerant architecture is applicable to nanoelectronic systems, we address the problem of random background charges in SET circuits. SET devices and circuits have been widely studied as one of those prospective substitutions to CMOS digital logic and memory [13]. With appropriate configuration a simple SET circuit can function as NAND logic, as shown in Fig. 9 [16]. The SET NAND gate consists of a single tunnel junction C_j and one capacitor C_0 as well as two input capacitors. When properly functioning, the output voltage is either low when both the inputs are high, or high in other cases. A so called island is created, so that the single electron can tunnel from and to it through the junction. The island can be made as small as a few nanometers,

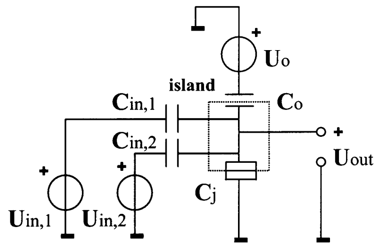

Fig. 9. An unreliable NAND implemented into SET circuits.

thus, an ultra dense system could be integrated. However, unfortunately, the SET circuit suffers from random background charges. Impurities and trapped electrons in the substrate induce image charges Q_0 on the surface of the island. If Q_0 is comparable with e (a single electron charge), the correct device function ise destroyed. Optimistically with a minimum device density of $10^{10}/\text{cm}^2$, about one in 1000 devices will have a considerable background charge fluctuation ($|Q_0| > 0.1e$) [13], i.e., $\varepsilon = 10^{-3}$. This is generally unacceptable for any VLSI system.

However, if in future SET chips with 10^{12} devices are eventually realizable, we could use the NAND multiplexing to achieve fault tolerance. Although it is difficult to speculate on the architecture of future nanochips, it seems plausible to make it a massively parallel computer consisting of a large number of rather simple processors with associated memories [17]. To evaluate the reliability, we assume that each processor has a 10-bit output and for each bit 40 logic devices are required. If we implement the multiplexing with $N = 250$ in such processors, then in each processor there are 10^5 devices. We further assume that a processor has a logical depth of 10, which is sufficient for general computation tasks, thus, accordingly, the NAND multiplexing will be repeated ten times. In this practical implementation, which has ten stages of multiplexing units, the restorative mechanism is achieved by the successive multiplexing units, therefore, the special restorative units would not be necessarily present and, hence, the redundancy level reduces to N from $n \cdot N$ in a n-stage system. For circuits with a few stages of logic, additional restorative stages could be needed to reach the required error bounds.

In such a processor, if no more than 10% of the outputs being faulty is seen as reliable, it is not hard to see that, given perfect inputs, the unreliability of the 1-bit NAND multiplexing output after ten stages is 10^{-8}. Since each processor only works reliably if none of the output bits are faulty, the reliability of the processor is then given by

$$R_p = (1 - U_r)^l \tag{39}$$

where U_r is the unreliability of 1-bit NAND multiplexing output and the processor has a l-bit output. If on the chip there are m processors, the reliability of the whole chip is then given by:

$$R_c = R_p^m. \tag{40}$$

We assume that 10% of the total 10^{12} devices are allocated to processors (others for memories, communications, etc.), therefore, the number of processors on the nanochip is about 10^6, i.e., $m = 10^6$. Thus, the ultimate reliability of the conceived nanochip can be calculated to be 0.9, at the expense of hundreds of redundant components. This indicates that future nanochips with 10^{12} devices, implemented using the NAND multiplexing technique, might be working at an acceptable reliability level, virtually having $10^9 \sim 10^{10}$ effective devices. This could be competitive in future nanoelectronics.

VII. Conclusion

A fault-tolerant technique, based on a massive duplication of imperfect devices and randomized imperfect interconnects, was comprehensively studied. Within a NAND multiplexing unit with a given number N of identical NAND logic gates, input error rate $\bar{x}$, and the error rate of the NAND logic being ε, the probability of the number of faulty outputs is theoretically a binomial distribution. It can be approximated by the Normal distribution when N is large (> 1000). The NAND multiplexing system can have more stages to improve the fault tolerance. The error distributions evolve as a stochastic homogeneous Markov process (chain).

A system architecture based on NAND multiplexing is investigated by studying the problem of random background charges in SET circuits. Although the conceived fault-tolerant architecture requires a rather large amount of redundant components, which makes it inefficient for the protection against permanent faults, normally compensated by reconfiguration techniques, it might be a system solution for ultra large integration of highly unreliable nanometer-scale devices affected by dominant transient errors. In addition, this multiplexing technique can be implemented in combination with a reconfigurable architecture, so that the obtained system will be efficiently robust against both permanent and transient faults.

Acknowledgment

The authors would like to thank M. Forshaw of University College London, U.K., for his fruitful contributions to the discussions. The editor and reviewers' valuable comments were highly appreciated.

References

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

[2] R. L. Dobrushin and S. I. Ortyukov, "Upper bound on the redundancy of self-correcting arrangements of unreliable functional elements," *Prob. Inform. Trans.*, vol. 13, pp. 203–218, 1977.

[3] ——, "Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements," *Prob. Inform. Trans.*, vol. 13, pp. 59–65, 1977.

[4] N. Pippenger, G. D. Stamoulis, and J. N. Tsitsiklis, "On a lower bound for the redundancy of reliable networks with noisy gates," *IEEE Trans. Inform. Theory*, vol. 37, pp. 639–643, 1991.

[5] N. Pippenger, "On networks of noisy gates," in *Proc. 26th Annu. Symp. Foundations Computer Science*, 1985, pp. 30–38.

[6] ——, "Invariance of complexity measures for networks with unreliable gates," *J. ACM*, vol. 36, pp. 531–539, 1989.

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

[8] T. Feder, "Reliable computation by networks in the presence of noise," *IEEE Trans. Inform. Theory*, vol. 35, pp. 569–571, 1989.

[9] S. Spagocci and T. Fountain, "Fault rates in nanochip devices," in *Proc. Electrochem. Soc.*, vol. 98–19, 1999, pp. 582–593.

[10] 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, 1998.
[11] D. Mange, M. Sipper, A. Stauffer, and G. Tempesti, "Toward robust integrated circuits: The embryonics approach," *Proc. IEEE*, vol. 88, pp. 516–541, 2000.
[12] K. Nikolic, A. Sadek, and M. Forshaw, "Architectures for reliable computing with unreliable nanodevices," *Proc. IEEE-NANO*, pp. 254–259, 2001.
[13] K. K. Likharev, "Single-electron devices and their applications," *Proc. IEEE*, vol. 87, pp. 606–632, 1999.
[14] W. Evans and N. Pippenger, "On the maximum tolerable noise for reliable computation by formulas," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1299–1305, 1998.
[15] D. Lu, *Stochastic Process and Applications*. Beijing, China: Tsinghua Univ. Press, 1986.
[16] R. H. Klunder and J. Hoekstra, "Programmable logic using a SET electron box," in *Proc. ICECS*, 2001, pp. 185–188.
[17] T. J. Fountain, M. J. B. Duff, D. G. Crawley, C. D. Tomlinson, and C. D. Moffat, "The use of nanoelectronic devices in highly parallel computing systems," *IEEE Trans. VLSI Syst.*, vol. 6, pp. 31–38, 1998.

Jie Han received the B.Sc. degree in electronic engineering from Tsinghua University, Beijing, China, in 1999. He is currently working toward the Ph.D. degree at the Pattern Recognition Group, Faculty of Applied Sciences, Delft University of Technology, Delft, The Netherlands.

His research interests include nanoelectronic circuit and system design, fault-tolerant system design, reconfigurable architectures, massively parallel computing structures and quantum computation.

Pieter Jonker received the B.Sc. and M.Sc. degrees in electrical engineering from the Twente University of Technology, Twente, The Netherlands, in 1977 and 1979, respectively, and the Ph.D. degree in applied physics from the Delft University of Technology, Delft, The Netherlands, in 1992.

In 1980, he worked at the Netherlands Organization for Applied Scientific Research (TNO) Laboratory of Applied Physics, The Hague, The Netherlands. In 1985, he became an Assistant Professor and, in 1992, an Associate professor with the Pattern Recognition Group Department of Applied Physics, Delft University of Technology. He was a Visiting Scientist and Lecturer at the ITB Bandung Indonesia in 1991. He was Coordinator of several large multidisciplinary projects – including EU projects- in the field of computer architecture and robotics. From his experience in the hard and software design of massively parallel machines he entered the field of fault tolerant solutions for nanoscale devices.

Dr. Jonker is a Member of the IEEE Computer Society. He was Chairman of the International Association for Pattern Recognition (IAPR) PR TC3 on special architectures for Machine Vision and he has been a Fellow of the IAPR since 1994.