Analysis of Error Masking and Restoring Properties of Sequential Circuits

Jinghang Liang, Jie Han, *Member, IEEE*, and Fabrizio Lombardi, *Fellow, IEEE*

Abstract—Scaling of CMOS technology into nanometric feature sizes has raised concerns for the reliable operation of logic circuits, such as in the presence of soft errors. This paper deals with the analysis of the operation of sequential circuits. As the feedback signals in a sequential circuit can be logically masked by specific combinations of primary inputs, the cumulative effects of soft errors can be eliminated. This phenomenon, referred to as error masking, is related to the presence of so-called restoring inputs and/or the consecutive presence of specific inputs in multiple clock cycles (equivalent to a synchronizing sequence in switching theory). In this paper, error masking is extensively analyzed using the operations of state transition matrices (STMs) and binary decision diagrams (BDDs) of a finite state machine (FSM) model. The characteristics of state transitions with respect to correlations between the restoring inputs and time sequence are mathematically established using STMs; although the applicability of the STM analysis is restricted due to its complexity, the BDD approach is more efficient and scalable for use in the analysis of large circuits. These results are supported by simulations of benchmark circuits and may provide a basis for further devising efficient and robust implementations when designing FSMs.

Index Terms—Finite state machines (FSMs), Error masking, Sequential circuits, State transition matrices (STMs), Transition probability matrices, Binary decision diagrams (BDDs), Soft errors

1 Introduction

The aggressive scaling of CMOS technology has resulted in small device dimensions and low tolerance to design and process variations, thus having a negative impact on the reliability of digital circuits [1]. New failure modes have been observed due to high integration and device fabrication effects, such as time-dependent dielectric breakdown of materials, hot carrier injection and negative/positive bias temperature instability in transistors. In addition to permanent defects, soft errors have also become a concern as the temporary interference by noisy environments affects the reliable operation of nanometric digital circuits. High integration densities and low voltage/current thresholds have increased the soft error rates (SERs) of circuits and systems.

To address the above issues, various techniques for reliability evaluation have been proposed. These include techniques using probabilistic transfer matrices (PTMs) [2, 3], probabilistic gate models (PGMs) [4, 5], Bayesian networks [6, 7], probabilistic decision diagrams (PDDs) [8] and stochastic computational models [9]. Methodologies have also been developed for the analysis of SERs, including those based on error propagation [10], symbolic models using binary and algebraic decision diagrams (BDDs and ADDs) [11, 12], Markov chains [13] and random vector based signatures [14]. The effect of soft errors on sequential circuits has been analyzed using a finite state machine (FSM) model [15]. If a transient fault is limited to occur in a single clock cycle, the resulting error is classified as critical (or non-critical) when it appears at the next state (or output) of a sequential circuit [16]. It has been observed that a critical state transition error may not necessarily cause a system error because it can be eliminated by utilizing some inputs of an FSM [17]. As a framework, a statistical analysis has been developed for analyzing the probability of such a self-recovery due to the effect of transient errors in a sequential circuit [18].

Self-recovery is an interesting process, because the reliability of a circuit is typically expected to decrease with time due to the presence and likelihood of errors. Reliability refers to the probability of sys-

- *J. Liang and J. Han are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada T6G 2V4. E-mail: jinghang@ualberta.ca, jhan8@ualberta.ca.*
- *F. Lombardi is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, USA. E-mail: lombardi@ece.neu.edu.*

Manuscript received Jan. 15, 2011. Please note that all acknowledgments should be placed at the end of the paper, before the bibliography.

tem survival, i.e., the probability that a system (in this paper, a sequential circuit) functions correctly and produces correct outputs. This is stringent for computing at nanometric scales. For sequential circuits this is more pronounced because accumulation of errors through feedback signals may occur and therefore, the reliability may decrease.

A main contribution of this paper is to show, characterize and assess the reliable operation of a sequential circuit through a detailed analysis of its state transition matrices (STMs). The dependency of circuit reliability on its input distribution and sequences is revealed due to the masking of errors. Error masking in a sequential circuit refers to the logic masking effect imposed on the feedback signals by specific combinations of primary inputs (referred to as restoring inputs). As a result, the presumably monotonically decreasing reliability of a sequential circuit can actually be interrupted and restored by the primary inputs. The restoring inputs are equivalent to the synchronizing sequences in switching theory [19, 20], that have been extensively used to facilitate testing of sequential circuits [21]. Both experimental and theoretical approaches have been used to compute synchronizing sequences for testing an FSM [22, 23].

In this paper, error masking is theoretically analyzed using the STMs in an FSM model as a mathematical framework. To alleviate the complexity issues in the STM computation, an efficient approach using binary decision diagrams (BDDs) is further employed for analyzing error masking in large circuits. Using benchmarks as evaluated in [14] and [18] for SER analysis, simulation results are presented to show that single- and multiple-step restoring inputs can be found by the proposed approach. The contribution of this paper is, therefore, to present an analytical framework that utilizes STMs and BDDs to characterize, analyze and assess the fundamental mechanism of error masking. Albeit beyond the scope of this paper, this methodology and related observations can be complemented by incorporating timing and electrical information into the analysis for formulating error mitigation schemes for nanoscale systems.

This paper is organized as follows. Section 2 defines the terminologies used in this paper. Section 3 presents the STM framework for characterizing error masking. Section 4 presents an efficient evaluation using BDDs; simulation results are also provided. Section 5 concludes the paper.

2 DEFINITIONS

Consider a Mealy model of a sequential circuit, as shown in Fig. 1. In this circuit, there are $m+n$ inputs: m of them are Primary Inputs while the remaining n inputs are Present States (i.e., the feedback signals from the flip-flops). There are also $l+n$ outputs: l of them are Primary Outputs, while the remaining n outputs are Next States (they will be stored in the flip-flops and then fed back into the inputs during the next clock period).

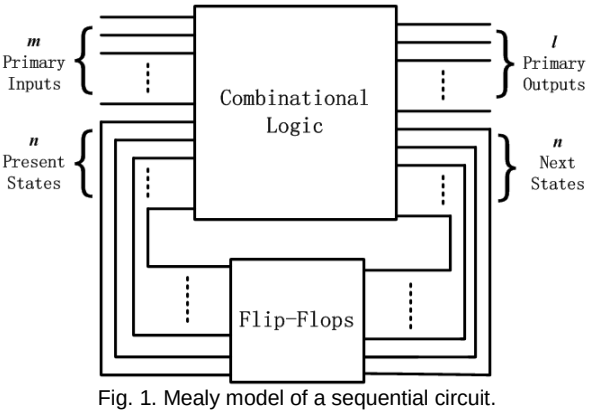


Fig. 1. Mealy model of a sequential circuit.

A *finite state machine* (FSM) is a classical abstract model for the functions of a sequential circuit. An FSM of the general Mealy model is defined as a six tuple $<I, S, \delta, S_0, O, \lambda>$, where I is the set of inputs, S is the set of states, $\delta: S \times I \rightarrow S$ is the next-state function of an input and the present state, $S_0 \subseteq S$ is the set of initial states, O is the set of outputs, and $\lambda: S \times I \rightarrow O$ is the output function of an input and the present state [24]. For the sequential circuit of Fig. 1, I is a set of vectors of m bits, O is a set of vectors of l bits, and S can be represented by a set of vectors of n bits. An FSM can efficiently be described by a state transition graph, in which every node (or vertex) represents a state of the machine and every arc (or directed edge) indicates a state transition.

The state transitions in an FSM can be described by a *state transition matrix* (STM); in the traditional representation, the STM has Boolean entries (0 or 1) to denote the deterministic functions of a sequential circuit. For a probabilistic operation (due to the occurrence of soft errors for instance), the state transitions are described by a *transition probability matrix* [15, 16] due to the underlying Markov nature of the FSM. For the sequential circuit of Fig. 1, let $I =$

$\{x_0, x_1, \ldots, x_{2^m-1}\}$, $S = \{s_0, s_1, \ldots, s_{2^n-1}\}$ and $O = \{y_0, y_1, \ldots, y_{2^l-1}\}$; the transition probability matrix $\mathbf{\Phi_i}$ is a $2^n \times 2^n$ matrix for a given input vector x_i:

$$\mathbf{\Phi_i} = \begin{bmatrix} p(s_0|s_0) & p(s_1|s_0) & \ldots & p(s_{2^n-1}|s_0) \\ p(s_0|s_1) & p(s_1|s_1) & \ldots & p(s_{2^n-1}|s_1) \\ \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ p(s_0|s_{2^n-1}) & p(s_1|s_{2^n-1}) & \ldots & p(s_{2^n-1}|s_{2^n-1}) \end{bmatrix}, \quad (1)$$

where the *(k, j)* entry $p(s_k|s_j)$ denotes the transition probability from the present state s_j to the next state s_k for the input x_i. For deterministic operations, $p(s_k|s_j) = 0$ or 1 for any *k* and *j*, thus yielding an ideal STM $\mathbf{T_i}$ for the input x_i. Since an STM is unique for every input vector, a total of 2^m STMs are required to describe the operations of the sequential circuit of Fig. 1.

Next, a cumulative STM (CSTM) is defined; a CSTM, $\mathbf{T_{t_1,t_2}}$, describes the state transitions from time t_1 to t_2 for a set of inputs (between t_1 and t_2). It is given by:

$$\mathbf{T_{t_1,t_2}} = \prod_{r=t_1}^{t_2} \mathbf{T}(r), \quad (2)$$

where $\mathbf{T}(r)$ (with $t_1 \le r \le t_2$) is an STM at time *r* (for a corresponding input). For a set of inputs between time *0* and *t-1*, for example, the corresponding CSTM is $\mathbf{T_{0,t-1}}$. Given an initial state, $\boldsymbol{s(0)}$, the state at a subsequent time t can be computed as:

$$\boldsymbol{s(t)} = \boldsymbol{s(0)} * \mathbf{T_{0,t-1}}. \quad (3)$$

Similar matrices can be defined for the transition probabilities between the present states and the outputs. These matrices are referred to as *output transition matrices*. The STMs and the transition probability matrices are essentially equivalents of the ideal transfer matrices (ITMs) and probabilistic transfer matrices (PTMs) [2, 3], so they can be constructed by extending and combining the gate ITMs and PTMs (as applicable to combinational circuits) to the topology and operation of a sequential circuit. Transition probability matrices have also been used for the Markovian analysis of FSMs [25] and fault-tolerant systems [26].

In a sequential circuit, the *restoring inputs* are the primary inputs or a sequence of primary inputs that logically mask the feedback signals. *Error masking* in a sequential circuit refers to the phenomenon that the feedback signals are logically masked by specific combinations of primary inputs (i.e., the restoring inputs). This error masking can occur in one or multiple steps. An *N*-step error masking is illustrated in the state transition diagram of Fig. 2. Assume that the state of an FSM at *t=0* is not deterministic, but probabilistic (possibly due to the effects of soft errors in a sequential circuit); so in principle, it can take any of the 2^n states, as shown in the first row in Fig. 2. However, this state space can be reduced at later steps as result of the state transition properties of the FSM. This is determined by the STMs and thus the primary input at each step. If this state space is reduced to one that has only one single state after *N* steps, then the initial state at time t_0 becomes *irrelevant* for determining the final state; hence, any initial error would be masked by this *N*-step transition process. The inputs that result in the occurrence of this error masking are a set of restoring inputs. In a state transition graph, this is indicated by various state transition paths (represented by directed edges) that eventually lead to the same destination state (represented by a vertex). Therefore, a sequential circuit is said to be *reliable* if error masking frequently occurs; it is *unreliable* otherwise.

3 Error Masking

3.1 Restoring inputs

In switching theory it is well known that among all output combinations of a sequential circuit, some are determined only by the primary inputs, and not by the feedback signals (or present states). This property is very useful as these inputs can be utilized for determining the values of the outputs and therefore voids the cumulative effects of errors. As defined previously, these inputs are called *restoring inputs*. A circuit is considered *reliable* if these restoring inputs frequently appear. In a reliable circuit, when the feedback signals are logically masked (i.e. not relevant for determining the next state or output values), the cumulative effects of errors are effectively mitigated and/or possibly eliminated.

The next-state function in an FSM is particularly important as it determines whether an accumulation of errors could occur. If the next state is always fully or partially determined by the present state, then the design is considered as unreliable, i.e., errors and their effects through the feedback signals will accumulate. An unreliable design will fail with a high probability after a sufficiently long time. Hence, a Moore machine in which the outputs are only determined by the present states (or feedback signals), tends to be unreliable in the presence of random soft errors. This is consistent with the homogeneity in the

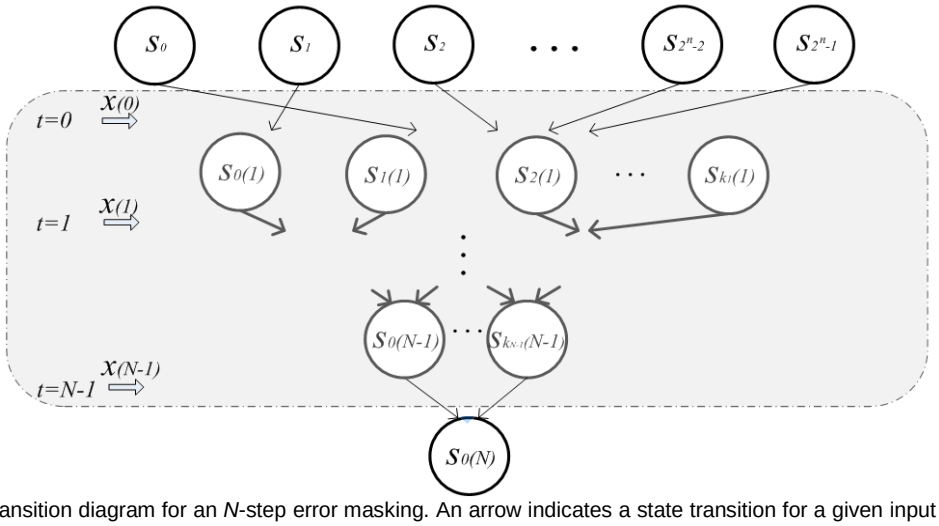


Fig. 2. State transition diagram for an *N*-step error masking. An arrow indicates a state transition for a given input (shown on the left).

Markov characterization of the FSMs.

For nanoscale computing, the rate of an error is projected to be finite but small, so the next state is expected to be ideal with a very high probability. In a sequential circuit, therefore, a transition probability matrix is expected to have entries that are approximately 0's and 1's. This leads to the *convergence* of the transition probability matrix and its ideal STM. In fact, the STMs contain original information on the distinctive features of a circuit, so they are of fundamental importance when determining the restoring inputs.

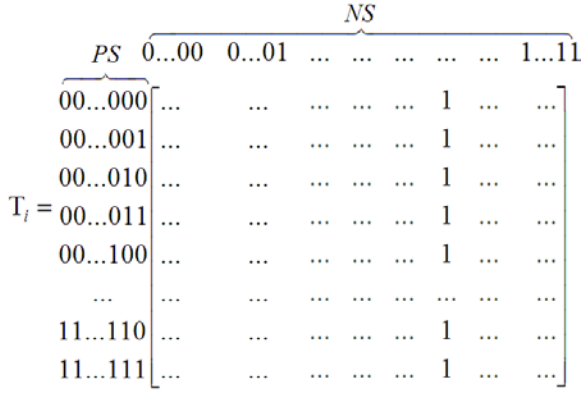


Fig. 3. An STM indicating the existence of a restoring input. '$\mathbf{T_i}$' is the STM for the *i*th input. 'PS' denotes the present state and 'NS' denotes the next state.

Consider as an example the STM shown in Fig. 3 for an input in the set I_0. In this STM, if the entries in a column, which indicate the transitions from the present states to a specific next state, are all 1's, then the primary inputs for this STM (i.e., those in the set I_0) are a set of restoring inputs. A similar procedure can be applied to the analysis of output transition matrices, which characterizes whether restoring inputs exist to at least partially eliminate the accumulated effects of errors to the primary outputs.

3.2 Error masking in multiple steps

Restoring inputs can appear in single and multiple steps (or clock cycles). For a single step, the STM for a restoring input is expected to have all 1's in a column. For multiple steps, restoring inputs can be found by analyzing the CSTM obtained by (2), i.e., the product of STMs at these steps.

3.2.1 Two step process

As an example, the two-step case will be first presented for establishing the conditions such that the restoring inputs exist. Let $\mathbf{T_u}$ and $\mathbf{T_v}$ be the two $2^n \times 2^n$ STMs involved in a two-step operation (for inputs x_u and x_v respectively). If x_u and x_v are the restoring inputs for the *j*th next state, the two-step CSTM is given by

$$\mathbf{T_u} * \mathbf{T_v} = \begin{bmatrix} 0 & \dots & j-1 & j & j+1 & \dots & 2^n-1 \\ 0 & \vdots & 0 & 1 & 0 & \vdots & 0 \\ 0 & \vdots & 0 & 1 & 0 & \vdots & 0 \\ 0 & \vdots & 0 & 1 & 0 & \vdots & 0 \\ \dots & \vdots & \dots & \dots & \dots & \vdots & \dots \\ 0 & \vdots & 0 & 1 & 0 & \vdots & 0 \end{bmatrix}. \quad (4)$$

Let the elements in the *i*th row and the *j*th column in $\mathbf{T_u}$ and $\mathbf{T_v}$ be given by a_{ij} and b_{ij} respectively ($0 \le i \le 2^n - 1$ and $0 \le j \le 2^n - 1$), then

$$\mathbf{T_u} * \begin{bmatrix} b_{0j} \\ b_{1j} \\ b_{2j} \\ \dots \\ b_{(2^n-1)j} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}. \quad (5)$$

Also, the following is always applicable

$$\mathbf{T_u} * \begin{bmatrix} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}. \quad (6)$$

Subtracting (5) from (6) gives

$$\mathbf{T_u} * \begin{bmatrix} 1 - b_{0j} \\ 1 - b_{1j} \\ 1 - b_{2j} \\ \dots \\ 1 - b_{(2^n-1)j} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}. \quad (7)$$

Since a_{ij} and b_{ij} can be either 0 or 1 for any i and j, there is only one element in each row of $\mathbf{T_u}$ being 1, due to the deterministic operation in the state transitions.

Assume for any i,

$$a_{ic_i} = 1. \quad (8)$$

Then by (7),

$$1 - b_{c_ij} = 0 \text{ or } b_{c_ij} = 1. \quad (9)$$

Both (8) and (9) determine the positions of 1's in the two STMs that establish the conditions for the restoring inputs in two steps.

Moreover, (8) and (9) can be used to analyze two extreme conditions. If $c_0 \neq c_1 \neq c_2 \neq \cdots \neq c_{2^n-1}$, the following condition must be applicable for a two-step masking:

$$b_{0j} = b_{1j} = b_{2j} = \dots b_{(2^n-1)j} = 1. \quad (10)$$

This implies that if the first step does not contribute to error masking, then the second step must solely contribute to masking. Consider the second extreme condition; if $c_0 = c_1 = c_2 = \cdots = c_{2^n-1} = k$, then for a two-step masking it is only required that,

$$b_{kj} = 1. \quad (11)$$

Of course, this corresponds to the opposite case of the first condition.

To better understand the relationship between $\mathbf{T_u}$ and $\mathbf{T_v}$ in a two-step restoring process, an example is presented next.

Example I: Let $\mathbf{T_{u_0}}$ and $\mathbf{T_{v_0}}$ be two 4x4 STMs, as shown in Fig. 4. The Boolean digit under each column of a matrix indicates whether the next state corresponding to this column is possible (value of 1) or not (value of 0) after each step. For example, a 1 for the first and third columns of $\mathbf{T_{u_0}}$ indicates that the first and third next states (i.e., "00" and "10") are the two possible next states after the first step of operation. Since the output from the first step serves as the present state for the second step, only the possible next states from the first step are still relevant and should be considered in the second step. In this example, these next states are the first and third, i.e., "00" and "10". They determine that only the first and third rows of $\mathbf{T_{v_0}}$ are relevant in the second step of operation (all other rows are masked in the first step). This is illustrated in the second row of Fig. 4, where the relevant columns are marked in red. Subsequently, if the 1's in the remaining rows of $\mathbf{T_{v_0}}$ belong to a single column, this two-step process is then expected to yield a single next state corresponding to this column, regardless of the present states. Therefore, a two-step error masking occurs.

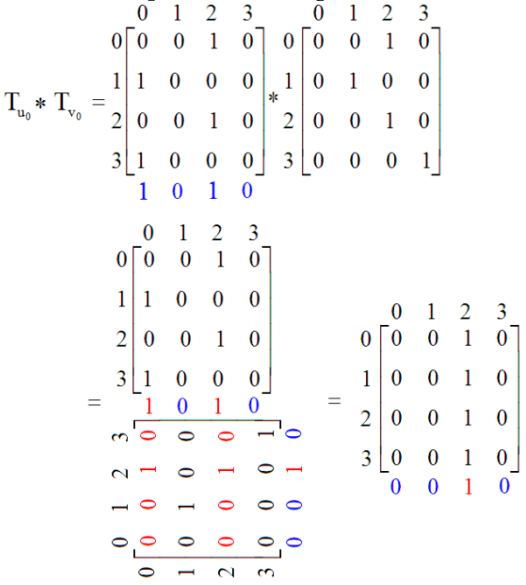


Fig. 4. Example of a two-step error masking.

3.2.2 *N step process*

In the general case that restoring occurs in N steps, the CSTM is given by

$$\mathbf{T_{u_1}} * \mathbf{T_{u_2}} * \dots * \mathbf{T_{u_N}} = \begin{matrix} 0 & \dots & j-1 & j & j+1 & \dots & 2^n-1 \end{matrix}$$
$$\begin{bmatrix} 0 & \vdots & 0 & 1 & 0 & \vdots & 0 \\ 0 & \vdots & 0 & 1 & 0 & \vdots & 0 \\ 0 & \vdots & 0 & 1 & 0 & \vdots & 0 \\ \dots & \vdots & \dots & \dots & \dots & \vdots & \dots \\ 0 & \vdots & 0 & 1 & 0 & \vdots & 0 \end{bmatrix} \quad (12)$$

as applicable to a restoring to the jth next state.

Due to the state-transition property, each row in $\mathbf{T_{u_i}}$ $(1 \le i \le N)$ has only one 1 and all other elements are 0. Assume that $\mathbf{r_{ij}} = (0,0 \dots 0,1,0 \dots 0,0)$ is the jth row in $\mathbf{T_{u_i}}$, then

$$\mathbf{T_{u_i}} = \begin{bmatrix} \mathbf{r_{i0}} \\ \mathbf{r_{i1}} \\ \mathbf{r_{i2}} \\ \dots \\ \mathbf{r_{i(2^n-1)}} \end{bmatrix}. \quad (13)$$

Further, let c_{ij} be the column index of the 1 in $\mathbf{r_{ij}}$;

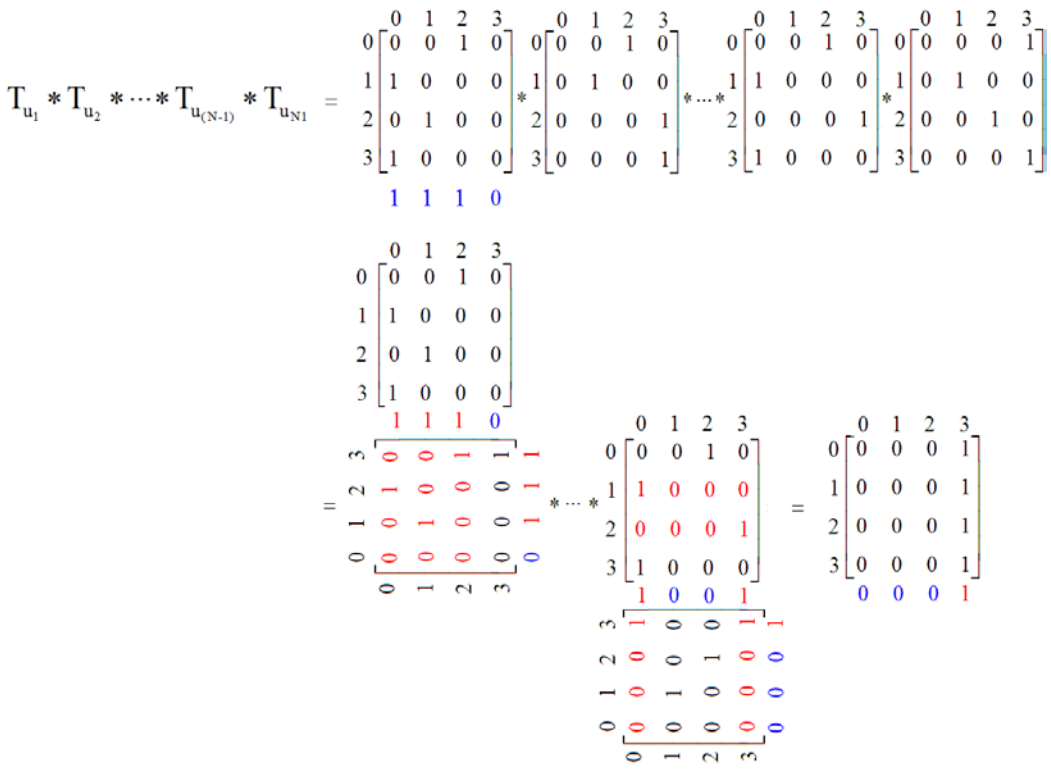


Fig. 5. Example of *N*-step error masking.

then for the first two steps,

$$\mathbf{T_{u_1}} * \mathbf{T_{u_2}} = \begin{bmatrix} \mathbf{r_{10}} \\ \mathbf{r_{11}} \\ \mathbf{r_{12}} \\ \dots \\ \mathbf{r_{1(2^n-1)}} \end{bmatrix} * \begin{bmatrix} \mathbf{r_{20}} \\ \mathbf{r_{21}} \\ \mathbf{r_{22}} \\ \dots \\ \mathbf{r_{2(2^n-1)}} \end{bmatrix} = \begin{bmatrix} \mathbf{r_{2c_{10}}} \\ \mathbf{r_{2c_{11}}} \\ \mathbf{r_{2c_{12}}} \\ \dots \\ \mathbf{r_{2c_{1(2^n-1)}}} \end{bmatrix}. \quad (14)$$

For the *N* steps, the CSTM is

$$\mathbf{T_{u_1}} * \mathbf{T_{u_2}} * \dots * \mathbf{T_{u_N}} = \begin{bmatrix} \mathbf{r_{10}} \\ \mathbf{r_{11}} \\ \mathbf{r_{12}} \\ \dots \\ \mathbf{r_{1(2^n-1)}} \end{bmatrix} * \begin{bmatrix} \mathbf{r_{20}} \\ \mathbf{r_{21}} \\ \mathbf{r_{22}} \\ \dots \\ \mathbf{r_{2(2^n-1)}} \end{bmatrix} * \dots * \begin{bmatrix} \mathbf{r_{N0}} \\ \mathbf{r_{N1}} \\ \mathbf{r_{N2}} \\ \dots \\ \mathbf{r_{N(2^n-1)}} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{r}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{10}\right)}\right)}\right)} \\ \mathbf{r}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{11}\right)}\right)}\right)} \\ \mathbf{r}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{12}\right)}\right)}\right)} \\ \dots \\ \mathbf{r}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{1(2^n-1)}\right)}\right)}\right)} \end{bmatrix}, \quad (15)$$

where $c_{(N-1)\dots\dots(c_{2(c_{1i})})}$ with $0 \le i \le 2^n - 1$, determined by the positions of 1's in the STMs of previous steps, gives the index of the row vector in $\mathbf{T_{u_N}}$.

If error masking occurs, then all the 1's are in the same column in the matrix obtained by (15), so all the rows of the matrix are the same, i.e.,

$$\mathbf{r}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{10}\right)}\right)}\right)} = \mathbf{r}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{11}\right)}\right)}\right)} = \mathbf{r}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{12}\right)}\right)}\right)} = \dots = \mathbf{r}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{1(2^n-1)}\right)}\right)}\right)}, \quad (16)$$

and equivalently, the column indices of 1's are the same, i.e.,

$$\mathbf{c}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{10}\right)}\right)}\right)} = \mathbf{c}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{11}\right)}\right)}\right)} = \mathbf{c}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{12}\right)}\right)}\right)} = \dots = \mathbf{c}_{\mathbf{N}\left(\mathbf{c}_{(N-1)\dots\dots\left(\mathbf{c}_{2\left(\mathbf{c}_{1(2^n-1)}\right)}\right)}\right)} \quad (17)$$

For a two-step restoring process (as given by (4)), it can be obtained from (17) that

$$c_{2c_{10}} = c_{2c_{11}} = c_{2c_{12}} = \dots = c_{2c_{1(2^n-1)}} = j, \quad (18)$$

i.e., as $a_{ic_{1i}} = 1$ and $c_{2c_{1i}} = j$, we have $b_{c_{1i}j} = 1$, for any $0 \le i \le 2^n - 1$.

Hence, (16) and (17) reveal the underlying relationships among multiple STMs, as required for obtaining the restoring inputs. STMs are sparse

matrices, so these relationships can be used to efficiently analyze *N*-step error masking.

Similar to the case of two-step masking presented previously, an example is given in Fig. 5. It reveals the mappings between the STM elements (as characterized by (16) and (17)) and the accumulating effect of error masking in an *N*-step process.

For an unreliable design, fault-tolerant and error-mitigation techniques can be used to reduce the error effects [27, 28]. A simple method to accomplish this objective is to reset a sequential system periodically, as this may help to recover from the accumulated errors. This however may not be possible in all applications due to the disruption of normal circuit operation caused by the reset.

A possible solution to this problem is to use approximate logic to change the truth table, thus introducing error masking into the circuit. This will also introduce a tradeoff in the precision of the computed function, hence as applicable to inexact computing (also often referred to as soft computing), attaining more masking and less error accumulation.

3.3 Case study: S27, ISCAS'89 benchmark circuit

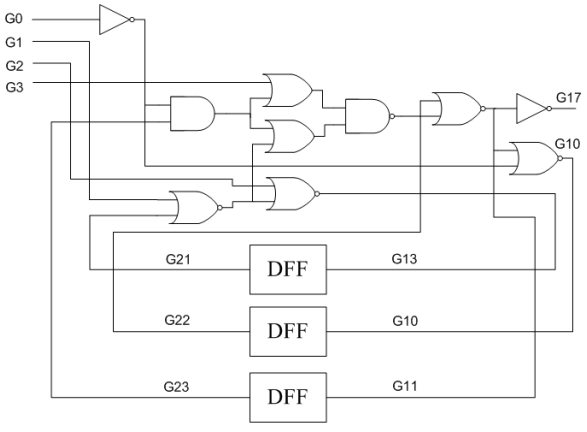


Fig. 6. Schematic diagram of S27.

The ISCAS'89 benchmark circuit S27 is considered in this section using the STM analysis to substantiate the characterization of error masking. S27 is a circuit with four primary inputs, three D flip-flops and one primary output (Fig. 6). When error masking occurs, the next state is totally dependent on the primary inputs, but not on the feedback (present state); this is applicable to S27 and shown as follows.

For S27, its STM is an 8x8 matrix. Consider the STM for the input "1110," as shown in Fig. 7; the next state (NS) is expected to be "110" regardless of the present state (*PS*). So when "1110" appears at the primary input, errors may have accumulated but are logically masked; therefore the reliability increases. An error masking STM can also be a result of several steps of STM operations. For example, neither the input "1001" nor "0110" is a restoring input; however, the synergetic effect of the consecutive presence of these two inputs leads to a two-step error masking.

$$T_{14} = \begin{array}{c} \\ PS \\ 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \begin{array}{c} \overbrace{}^{NS} \\ \begin{array}{cccccccc} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Fig. 7. $\mathbf{T_{14}}$ of S27 is the STM for the 14th primary input (i.e., "G0G1G2G3"="1110"). "*PS*" denotes the present state (G21, G22 and G23) and "NS" denotes the next state (G13, G10 and G11).

Table 1 shows the probability of occurrence of multiple-step error masking in S27; this probability is given by the ratio of the number of input sequences causing error masking over the total number of input sequences. For example, 34 out of the total 256 input sequences result in error masking for the two-step operation. Note that an *N*-step error masking process means that at least *N* steps are required to ensure error masking, i.e. if error masking occurs in *M* steps (*M*<*N*), then this is classified as an *M*-step masking and is not considered as a part of the *N*-step masking process.

Table 1. Simulation results for S27. The probability that multiple-step error masking occurs is given as the ratio between the number of restoring inputs and the total number of inputs.

Number of steps	1	2	3	4	5
Error masking probability	5/16	34/256	60/4096	92/65536	136/1048576
Run time (s)	0.0055	0.0159	0.2019	3.7354	72.3031
Memory (MByte)	0.1	0.3	1.1	2.1	2.8

The run time and memory usage of the STM analysis are also shown in Table 1. Although the memory usage steadily increases with the number of error masking steps, the run time changes more

drastically, because a significantly increased number of inputs must be considered in a multiple step masking. This makes the analysis of large circuits difficult, if not impossible. The complexity issues are further discussed in Section 3.5.

3.4 Partial error masking

When errors only occur in part of the state bits, their effects can be affected by the so-called *partial error masking*. Partial error masking refers to the phenomenon in which some of the feedback signals are logically masked by specific combinations of primary inputs and other feedback signals. Consider the sequential circuit model of Fig. 1; if some of the present state signals are unreliable, then their error effects can be masked by a combination of the other present state signals and a primary input. In the STM for such a primary input, this means that the rows can be re-ordered such that the unreliable state bits are next to each other and the corresponding adjacent rows lead to the same next state. This process is shown in Fig. 8.

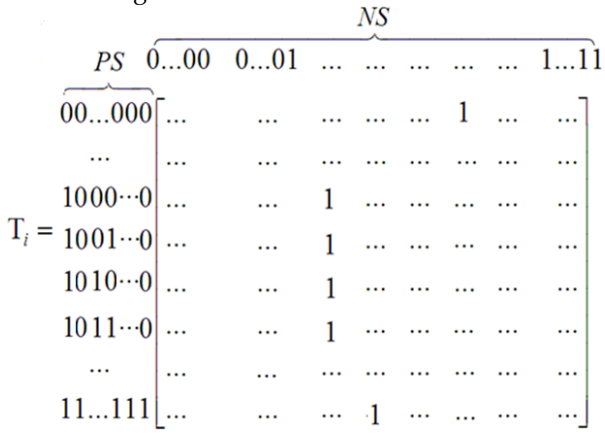


Fig. 8. An STM with the occurrence of partial error masking. In $\mathbf{T_i}$ (i.e., the STM for the *i*th input), the third and fourth bits in the present state (PS) are masked by the combinations of the *i*th input and the remaining present-state signals, shown in the four rows in the middle of the STM.

The analytical procedure outlined in Section 3.2 is applicable also to multiple-step partial error masking; however, since only a subset of the state signals are of interest, a sub-matrix of each STM is needed in the analysis. Similarly, as partial error masking also applies to output signals, therefore it can be analyzed using the output transition matrices of the sequential circuit.

3.5 Complexity

The analysis using STMs reveals the fundamental mechanism of error masking. The mapping relationships given in (16) and (17) can be used for an optimized analysis by leveraging the fact that STMs are sparse matrices. Nevertheless, this analysis incurs a large computational complexity for finding the restoring inputs. Consider the circuit model of Fig. 1 as an example; there are a total of 2^m STMs. To find the restoring inputs in an N-step process, a total of $(2^m)^N$ CSTMs need to be examined, thus resulting in a computational complexity of at least $O(2^{mN})$. This computation is of course not scalable for analysing large circuits. In the next section, an approach using BDDs is proposed for a more efficient analysis.

4. Analysis using BDDs

A binary decision diagram (BDD) is a canonical (or unique) representation of a Boolean function [29]. It is also efficient in representing a large combinatorial set. BDDs have been shown to be effective in many applications involving FSMs [24]. In this section, a computationally efficient technique employing BDDs is used for analyzing error masking in sequential circuits. The CUDD package has been used throughout this study [31].

4.1 Finding the restoring inputs

A BDD is a directed acyclic graph, in which each node represents a variable and each edge is labeled "True" or "False" (or, "1" or "0"). The edges lead to leafs labelled "1" or "0" at the bottom of the graph. For the sequential circuit model of Fig. 1, a BDD can be generated for every variable (or bit) of the next state (and the primary output) as a function of the primary inputs and the present state. Therefore, a total number of n diagrams must be generated for the n variables in the next state. The variables in a BDD are usually ordered to find an optimal diagram; this ordering is typically done heuristically by relying on specific features of the system being analyzed [30]. To find the restoring inputs, a special ordering is imposed such that the variables in the primary inputs are first analyzed, followed by those in the present state. The BDDs generated for the next state of the benchmark circuit S27 are shown in Fig. 9.

In a BDD, if there exists a path that starts from the root (or a primary input) and reaches a leaf "1" or "0," without traversing through any present state variable, then the primary input dictated by this path is a restoring input for this variable in the next state. Given the BDDs for the other variables in the next state, the restoring inputs for those variables can be found in the same way. The restoring inputs for a circuit are then obtained as the intersection of the set of restoring inputs for each variable in the next state. This establishes the conditions for a single-step error

masking. For partial error masking, a similar procedure applies when only a subset of the present state variables are considered as required by nature of the partial masking process.

Example II: Consider S27 again. Fig. 9(b) shows the BDD for the variable G13 in the next state. In this BDD, G21 is the only existing present state variable and two paths starting from the primary input reach the end leafs without traversing G21. These paths correspond to the primary inputs G2="1" and G2G1="01," or equivalently, G0G1G2G3 ="XX1X" and "X10X," where "X" denotes the don't-care condition. Similar analysis can be performed for G10 and G11, and the results are shown in the second row of Table 2. The restoring inputs for s27 can then be obtained as the intersection of the three sets of inputs as shown in the third row of Table 2. This confirms the results obtained by the STM analysis in Table 1.

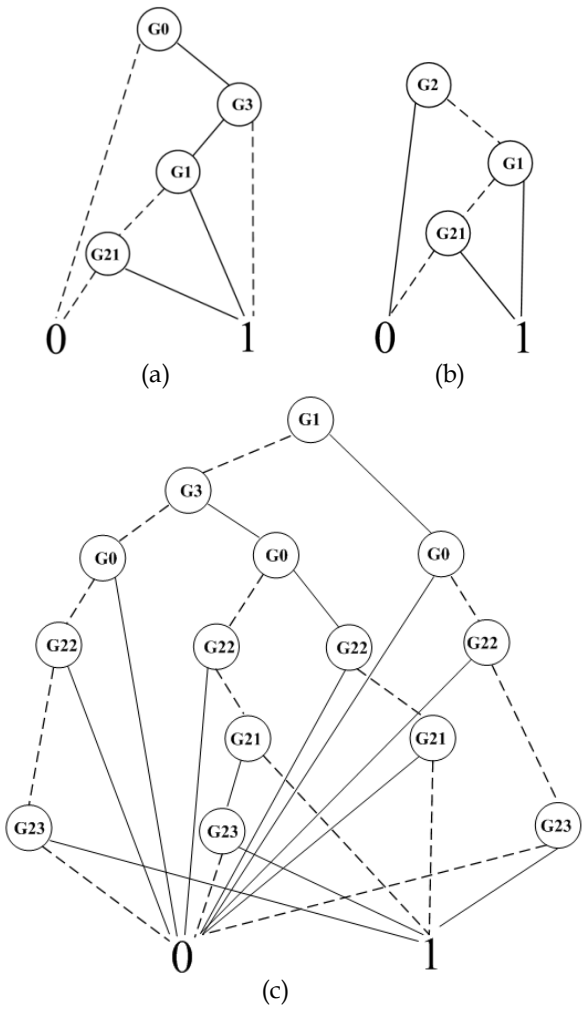


Fig. 9. BDDs for the next state of s27: (a) G10, (b) G13, (c) G11.

Table 2. Restoring inputs obtained by the BDD analysis for s27; "X" denotes a don't-care value.

The next state variable	G10	G11	G13
Restoring inputs for each variable (G0G1G2G3)	0XXX 1XX0 11X 1	11XX 10X0	XX1X X10X
Restoring inputs for s27	1010, 1100, 1101,1110,1111		

4.2 Multiple-step error masking

For a multiple-step operation in the temporal domain, the so-called time-frame expansion technique can be used to convert the operation into a single step process, as proposed for soft error analysis of sequential circuits in [12]. This is illustrated in Fig. 10. For an *N*-step operation, the present states from the second to the *N*th steps are treated as internal signals; only the initial present state serves as the present state of the expanded iterative circuit, while all the primary inputs in the N steps become the primary inputs of the new circuit. The BDD analysis proposed in the previous subsection can then be used for a multiple-step error masking analysis.

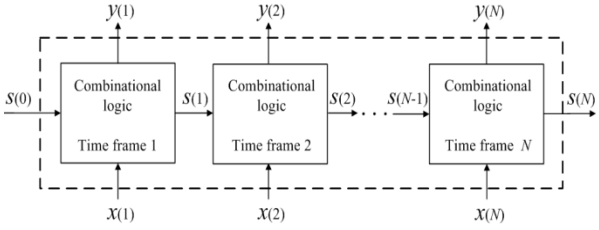


Fig. 10. Time frame expansion of a sequential circuit. The *N* frame expansion of a sequential circuit can be treated as a single sequential circuit with x(1), x(2),…,x(*N*) as inputs, y(1), y(2),…,y(*N*) as outputs, s(0) as the present state and s(N) as the next state.

4.3 Simulation results

The CUDD package [31] was used to generate the BDDs from the netlist of a circuit. A customized program was then written for extracting the restoring inputs from the BDDs. Table 3 shows the simulation results for the ISCAS'89 sequential benchmark circuits for finding (if any) single-step restoring inputs using the proposed BDD method. These benchmarks have also been used in [14] and [18] for SER analysis. A single-step error masking mostly occurs due to the presence of a "reset" signal, as observed for s382, s400, s444, s526, s820, s832 and s1488. Although other single-step restoring inputs are present in some circuits (such as s27, as discussed previously, and s1196), they do not always exist in a sequential circuit; this is generally due to the feature of a sequential circuit by which the next state is determined by both the primary inputs and the present state.

Even though beyond the scope of this paper, don't-care values in the logic definition of a sequential circuit could be assigned through synthesis to implement restoring inputs, thus avoiding the additional implementation of the "reset" signal as well as improving error masking.

The runtime and memory usage required by this approach are also reported in Table 3. The runtime includes the time for generating BDDs and extracting the restoring inputs, while the memory usage is only for using the CUDD package to produce the BDDs. While the memory usage is relatively stable for different circuits, the runtime is largely affected by the number of restoring inputs that must be extracted from the BDDs. Typically, it takes no more than a few seconds to generate BDDs for circuits of this size.

Table 3. The number of single-step restoring inputs (No.) for ISCAS'89 benchmark circuits found using BDDs with runtime (T) and memory usage (M). The runtime includes the time for generating the BDDs and extracting the restoring inputs; the memory usage is for the use of the CUDD package in producing BDDs.

Circuits	Gates	Inputs	Outputs	FFs	No.	T (s)	M (MByte)
s27	10	4	1	3	5	0.8147	1.068
s382	158	3	6	21	4	1.6324	1.089
s400	164	3	6	21	4	1.9575	1.089
s444	181	3	6	21	4	2.1576	1.089
s526	193	3	6	21	4	1.8003	1.089
s820	289	18	19	3	2^{17}	335.79	1.172
s832	287	18	19	5	2^{17}	389.91	1.172
s953	395	16	23	29	0	23.098	1.160
s1196	529	14	14	18	4890	79.965	1.166
s1488	653	8	19	6	128	15.971	1.145

To find a multiple-step restoring input, the netlist of a time frame extended circuit was first produced. Table 4 shows the results of multiple-step error masking; the reported runtime and memory usage further confirm the efficiency of the proposed BDD method. In these cases, the next state of the circuit is determined by a multiple clock-cycle state dependency. This process is more complicated than a single-step masking, as it implies that a time domain overhead will be incurred in the masking process due to the inherent latency. Hence, a designer is confronted with a tradeoff assessment of achieving error masking with a smaller number of cycles (or simply in one step) versus additional design complexity in the implementation of a sequential circuit.

The error masking phenomena observed for some circuits, such as the semaphore circuits s382 and s400, result from the "reset" signal. This occurs because the next state of a semaphore circuit is always determined by its present state unless the circuit is reset. Binary counters, whose next state is totally determined by the present state, also exhibit this property. Therefore, these features should be considered by designers when assessing the reliable operation of these types of circuits.

5. Discussion and Conclusion

This paper analyzes the reliable operation of sequential circuits in the presence of errors as likely to occur at nanometric feature sizes. The major contribution of this paper is the analysis of the phenomenon (referred to as "error masking") that affects the reliability of a sequential circuit, by utilizing the state transition matrices (STMs) and the binary decision diagrams (BDDs) in an FSM model. In a sequential circuit, restoring inputs allow for the masking of feedback signals and thus eliminating the cumulative effect of errors. A partial error masking occurs when part of the feedback signals are logically masked by a specific combination of the primary input and the other feedback signals.

In spite of its large computational complexity and limited applicability, the STM-based analysis reveals the fundamental mechanism of error masking. This framework is enhanced by using BDDs to extend the proposed analysis to large circuits. Computational efficiency can further be improved by using appropriate ordering of variables in the construction of BDDs, as well as an optimized process for extracting the restoring inputs.

Simulation results have shown the effectiveness of the proposed approach. They also point out a few attractive features that albeit beyond the scope of this paper, can be exploited to improve the reliable operation of sequential circuits. In an implementation of FSMs, for example, the don't-care values at the inputs can be configured into restoring inputs in logic synthesis such that errors in the state variables can be corrected during normal operation. Although an external reset can be utilized to clear the state variables, the use of restoring inputs has the following advantages:

1) Error masking due to restoring inputs occurs as an inherent part of the operation of an FSM without incurring an interruption. Therefore, the restored state is readily available for the next-step operation in the FSM. The time overhead incurred in this process is hence significantly reduced.

Table 4. The number of restoring inputs of benchmark circuits found using BDDs (No.) and the required memory usage (M) and runtime (T). Both the numbers of restoring inputs for up to *N*-step masking and for only *N*-step masking are reported (separated by '/'). The memory is from the use of the CUDD package in producing the BDDs and the runtime includes the time for generating the BDDs and extracting the restoring inputs.

Circuits	N=1			N=2			N=3			N=4			N=5		
	No.	M (MByte)	T (s)	No.	M (MByte)	T (s)	No.	M (MByte)	T (s)	No.	M (MByte)	T (s)	No.	M (MByte)	T (s)
lion	0/0	1.068	0.0743	1/1	1.068	0.3922	13/5	1.068	1.1712	88/0	1.079	4.0344	476/0	1.079	8.6948
train4	1/1	1.068	0.0609	7/0	1.068	0.7547	37/0	1.068	1.6463	175/0	1.079	7.4456	781/0	1.079	12.490
train11	0/0	1.068	0.0186	0/0	1.068	0.4387	2/2	1.068	1.3816	21/5	1.079	4.7655	154/20	1.079	7.7952
dk27	0/0	1.068	0.0224	0/0	1.068	0.5497	0/0	1.068	1.7537	2/2	1.068	2.0119	9/1	1.068	3.5688
s8	0/0	1.068	0.0585	9/9	1.079	1.5853	552/282	1.079	8.3371	16978 /1042	1.079	21.531	399238 /4454	1.089	48.779
tav	0/0	1.068	0.0498	0/0	1.068	0.8308	0/0	1.079	1.0540	0/0	1.079	3.1299	0/0	1.089	4.0540
bbtas	0/0	1.068	0.1190	0/0	1.068	0.3517	1/1	1.068	1.9340	13/6	1.079	2.6892	79/0	1.079	5.7482
mc	0/0	1.068	0.0959	0/0	1.068	0.9172	24/24	1.079	5.6541	464/80	1.079	12.263	7584/1152	1.079	32.469
beecount	4/4	1.068	0.1340	52/4	1.068	0.3804	351/0	1.079	2.6020	2464/0	1.079	15.162	24303/0	1.079	26.794
dk17	0/0	1.068	0.0163	0/0	1.068	0.8407	4/4	1.068	1.3112	43/15	1.079	4.5285	228/32	1.079	7.4427
dk512	0/0	1.068	0.0655	0/0	1.068	0.2543	0/0	1.068	1.0782	2/2	1.068	3.1656	8/0	1.068	5.2290
donfile	0/0	1.068	0.0680	0/0	1.084	0.2435	6/6	1.111	2.1524	88/40	1.111	6.9961	609/0	1.128	13.913
ex2	0/0	1.068	0.0276	0/0	1.084	0.2511	0/0	1.111	0.8258	2/2	1.128	2.1067	33/18	1.160	5.0838
ex3	0/0	1.068	0.1057	0/0	1.068	0.8143	0/0	1.068	1.5383	13/13	1.079	3.2599	72/17	1.095	5.8687
ex5	0/0	1.068	0.0849	1/1	1.068	0.1966	19/11	1.068	2.7749	126/12	1.079	3.4505	502/0	1.079	6.0046
ex7	0/0	1.068	0.1386	0/0	1.068	0.3500	1/1	1.068	1.9106	20/12	1.079	3.4314	154/26	1.095	7.9619
s27	5/5	1.068	0.8147	169 /34	1.079	2.6787	3471/60	1.079	7.7577	61173 /92	1.079	18.849	1018443 /136	1.089	38.934
s208.1	0/0	1.095	1.5472	0/0	1.122	1.9293	0/0	1.149	4.1450	0/0	1.163	7.8173	0/0	1.224	21.182
s382	4/4	1.089	1.9593	48/0	1.106	2.6160	448/0	1.155	5.3998	3840/0	1.303	9.2638	31744/0	1.424	20.622
s400	4/4	1.089	1.8909	48/0	1.106	2.4733	448/0	1.155	5.1233	3840/0	1.303	9.5132	31744/0	1.424	18.580
s444	4/4	1.089	1.6991	48/0	1.139	2.7513	448/0	1.204	6.0497	3840/0	1.405	14.945	31744/0	1.608	26.369
s526	4/4	1.089	1.5060	48/0	1.139	3.2551	448/0	1.239	5.1112	3840/0	1.528	11.184	31744/0	1.685	21.344

2) The use of restoring inputs eliminates the need for an external reset signal, therefore it simplifies the related logic design and reduces the required numbers of pins and pads in chip packaging. Subsequently, this has an impact on the performance, area and cost of a chip [32].

3) Multiple-step restoring and partial error masking allow for more flexibility as well as an extended functionality in the operation of an FSM, compared to the basic function of a reset.

Hence, the proposed error masking is a potentially useful property of FSMs that can be exploited for an efficient and robust implementation of sequential circuits.

REFERENCES

[1] S. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation," in IEEE Micro, vol. 25, no. 6, pp. 10-16, Nov. 2005.

[2] S. Krishnaswamy, G. F. Viamontes, I.L. Markov and J. P. Hayes: "Probabilistic transfer matrices in symbolic reliability analysis of logic circuits." ACM Trans. on Design Automation of Electronic Systems, vol. 13, article 8, Jan. 2008.

[3] C-C Yu and J. P. Hayes, "Scalable and accurate estimation of probabilistic behavior in sequential circuits" IEEE VLSI Test Symposium (VTS), 2010 28th. Santa Cruz, CA. pp. 165 – 170.

[4] E.R. Taylor, J. Han, J.A.B. Fortes, "Towards Accurate and Efficient Reliability Modeling of Nanoelectronic Circuits," in Proc. IEEE-NANO 2006, Vol. 1, pp. 395-398.

[5] J. Han, H. Chen, E. Boykin and J. Fortes, "Reliability evaluation of logic circuits using probabilistic gate models," Microelectronics Reliability, vol. 51, no. 2, 2011, pp. 468-476.

[6] T. Rejimon and S. Bhanja, "Scalable probabilistic computing models using Bayesian networks," in Proc. IEEE Int. Midwest Symp. Circuits Syst., 2005, pp. 712–715.

[7] K. Lingasubramanian and S. Bhanja, "Probabilistic Error Modeling for Sequential Logic," in Proc. IEEE Conf. Nanotechnology, 2007.

[8] A. Abdollahi, "Probabilistic Decision Diagrams for Exact Probabilistic Analysis," Proc. IEEE Int'l Conference on Computer Aided Design, 2007.

[9] H. Chen and J. Han, "Stochastic computational models for accurate reliability evaluation of logic circuits," ACM/IEEE GLSVLSI. May, 2010.

[10] H. Asadi, M.B. Tahoori, "Soft Error Modeling and Protection for Sequential Elements," in Proc. of IEEE Symposium on Defect and Fault Tolerance (DFT) in VLSI Systems, pp. 463-471, October 2005.

[11] N. Miskov-Zivanov, D. Marculescu, "Circuit Reliability Analysis Using Symbolic Techniques," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Vol. 25, Issue. 12, pp. 2638-2649, 2006.

[12] N. Miskov-Zivanov, D. Marculescu, "Modeling and Optimization for Soft-Error Reliability of Sequential Circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Vol. 27, Issue. 5, pp. 803-816, 2008.

[13] F.C. Sabou, D. Kazazis, R.I. Bahar, J. Mundy, W.R. Patterson, A. Zaslavsky, "Markov Chain Analysis of Thermally Induced Soft Errors in Subthreshold Nanoscale CMOS Circuits," IEEE Tran. on Device and Materials Reliability, vol. 9, no. 3, pp. 494-504, Sep. 2009.

[14] Krishnaswamy, S.; Plaza, S.M.; Markov, I.L.; Hayes, J.P.; , "Signature-Based SER Analysis and Design of Logic Circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 28, no. 1, pp. 74-86, Jan. 2009.

[15] R. Das and J. P. Hayes, "Monitoring Transient Errors in Sequential Circuits," in 16th IEEE Asian Test Symposium (ATS 2007), pp. 319-322, 2007.

[16] R. Das and J. P. Hayes, "Recovery from Transition Errors in Sequential Circuits" 2nd International Design and Test Workshop, IDT 2007, Cario, Egypt, pp. 105-110.

[17] J, P. Hayes, I. Polian, and B. Becker, "An analysis framework for transient-error tolerance," In IEEE VLSI Test Symp., pages 249–255, 2007.

[18] I. Polian, J.P. Hayes, S.M. Reddy and B. Becker, "Modeling and mitigating transient errors in logic circuits," IEEE Trans. on Dependable and Secure Computing, Vol. 8, No. 4, 2011, pp. 537 - 547.

[19] E.F. Moore, "Gedanken-experiments on sequential machines," in *Automata Studies*. C.E. Shannon and J. McCarthy, Eds., Princeton University Press, 1956, pp. 129-153.

[20] Z. Kohavi and N.K. Jha, Switching and Finite Automata Theory, Cambridge University Press, 2010.

[21] I. Pomeranz, S.M. Reddy, "On removing redundancies from synchronous sequential circuits with synchronizing sequences," Computers, IEEE Transactions on. Jan. 1996, pp. 20 – 32.

[22] I. Pomeranz, S.M. Reddy, "On Synchronizable Circuits and Their Synchronizing Sequences," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Vol. 19, No. 9, pp. 1086-1092, September 2000.

[23] C. Pixley, S.-W. Jeong, and G. D. Hachtel, "Exact calculation of synchronizing sequences based on binary decision diagrams," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Vol. 13, No. 8, pp. 1024-1034, August 1994.

[24] G. D. Hachtel and F. Somenzi, Logic Synthesis and Verification Algorithms, Springer; 1 edition. Jun 30, 1996

[25] G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi, "Markovian Analysis of Large Finite State Machines," In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Vol. 15, No. 12, pp. 1479-1493, December 1996.

[26] J. Han and P. Jonker, "A System Architecture Solution for Unreliable Nanoelectronic Devices," IEEE Trans. Nanotechnology, vol. 1, no. 4, pp. 201-208, December 2002.

[27] J. Han and P. Jonker, "A defect- and fault-tolerant architecture for nanocomputers," Nanotechnology, vol. 14, no. 2, pp. 224–230, 2003.

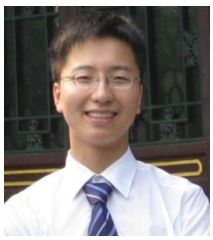
[28] J. Han, J. Gao, Y. Qi, P. Jonker, J.A.B. Fortes. "Toward Hardware-Redundant, Fault-Tolerant Logic for Nanoelectronics," IEEE Design and Test of Computers, July/August 2005, vol. 22, no. 4, 328-339.

[29] Bryant, R.E. Graph-Based Algorithms for Boolean Function Manipulation. Computers, IEEE Transactions on. Aug. 1986 pp.677 – 691.

[30] R. E. Bryant, ``Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams," ACM Computing Surveys, Vol. 24, No. 3 (September, 1992), pp. 293-318.

[31] F. Somenzi. CUDD: CU decision diagram package release 2.4.2. http://vlsi.colorado.edu/~fabio/CUDD/. (2009).

[32] J.M. Rabaey, A. Chandrakasan and B. Nikolic, "Digital Integrated Circuits; a design perspective," Second Edition, Pearson Education, 2003.



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

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

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

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

Dr. Han was nominated for the 2006 Christiaan Huygens Prize of Science by the Royal Dutch Academy of Science (Koninklijke Nederlandse Akademie van Wetenschappen (KNAW) Christiaan Huygens Wetenschapsprijs). His work was recognized by the 125th anniversary issue of *Science*, for developing theory of fault-tolerant nanocircuits. He has published a book, two book chapters and about 25 refereed articles in journals and conferences. He served as a Technical Program Chair in IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT 2012) and as a Technical Committee Member in DFT 2011, the International Workshop on Unique Chips and Systems, 2010 (UCAS-6) and 2012 (UCAS-7), and the First International Workshop on Secure and Resilient Architecture and Systems (WSRAS 2012).

Fabrizio Lombardi (M'81–SM'02-F'09) graduated in 1977 from the University of Essex (UK) with a B.Sc. (Hons.) in Electronic Engineering. In 1977 he joined the Microwave Research Unit at University College London, where he received the Master in Microwaves and Modern Optics (1978), the Diploma in Microwave Engineering (1978) and the Ph.D. from the University of London (1982).

He is currently the holder of the International Test Conference (ITC) Endowed Chair Professorship at Northeastern University, Boston. At the same Institution during the period 1998-2004 he served as Chair of the Department of Electrical and Computer Engineering. Prior to Northeastern University, he was a faculty member at Texas Tech University, the University of Colorado-Boulder and Texas A&M University.

During 2007-2010 Dr. Lombardi was the Editor-In-Chief of the IEEE Transactions on Computers. He is also an Associate Editor of the IEEE Transactions on Nanotechnology and IEEE Transactions on CAD of ICAS. Dr. Lombardi serves as the Chair of the Committee on "Nanotechnology Devices and Systems" of the Test Technology Technical Council of the IEEE (2003 -). In the past, Dr. Lombardi was an Associate Editor (1996-2000) and Associate Editor-in-Chief (2000-2006) of the IEEE Transactions on Computers and twice a Distinguished Visitor of the IEEE-CS (1990-1993 and 2001-2004). Dr. Lombardi is a Golden Core Member of the IEEE Computer Society and Fellow of the IEEE for "contributions to testing and fault tolerance of digital systems". He is the recipient of the 2011 Meritorious Service Award by the IEEE Computer Society. Starting January 1, 2012 he is serving as an elected Member of the Board of Governors of the IEEE Computer Society.

Dr. Lombardi has received many professional awards: the Visiting Fellowship at the British Columbia Advanced System Institute, University of Victoria, Canada (1988), twice the Texas Experimental Engineering Station Research Fellowship (1991-1992, 1997-1998) the Halliburton Professorship(1995), the Outstanding Engineering Research Award at Northeastern University (2004) and an International Research Award from the Ministry of Science and Education of Japan (1993-1999). Dr. Lombardi was the recipient of the 1985/86 Research Initiation Award from the IEEE/Engineering Foundation and a Silver Quill Award from Motorola-Austin (1996). At the IEEE DFT07 Symposium, one of his papers won the best paper award.

Dr. Lombardi has been involved in organizing many international symposia, conferences and workshops sponsored by professional organizations as well as guest editor of Special Issues in archival journals and magazines such as IEEE Transactions on Computers, IEEE Transactions on Instrumentation and Measurement, IEEE Transactions on VLSI, the IEEE Micro Magazine and the IEEE Design & Test Magazine. He is the Founding General Chair of the IEEE Symposium on Network Computing and Applications and the IEEE International Workshop on Design and Test of Nano Devices, Circuits and Systems.

His research interests are bio-inspired and nano manufacturing/computing, VLSI design, testing, and fault/defect tolerance of digital systems. He has extensively published in these areas and coauthored/edited seven books.