Scalable and Accurate Estimation of Probabilistic Behavior in Sequential Circuits

Chien-Chih Yu and John P. Hayes
Advanced Computer Architecture Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109, USA
{ccyu, jhayes}@umich.edu

***Abstract* – We present a new methodology for fast and accurate simulation of signal probabilities in sequential logic. It can be used for analyzing soft error effects at the logic level, estimating circuit reliability, and the like. Experimental results for large benchmarks show that signal error probabilities can be estimated over many cycles with high accuracy.**

***Keywords* – signal probability estimation; sequential circuits; soft error models; simulation methods.**

I. Introduction

Due to increasing gate density and operating frequencies, as well as decreasing supply voltages, ICs are becoming very sensitive to probabilistic effects, such as soft (transient) errors and random process variations. The effects in question are generally described in terms of a circuit's signal probabilities. However, the complexity of exactly calculating signal probabilities grows exponentially with circuit size, and is impractical for large circuits [14]. Inexact or approximate methods have been developed to improve computational efficiency [6][7][16], but their accuracy may be unacceptable.

Considerable attention has been devoted recently to analyzing the impact of radiation-induced soft errors [2-3] [5][9-13][18]. Several techniques have been proposed for estimating the soft-error rate (SER) in combinational logic, such as binary decision diagrams (BDDs) [18], probabilistic transfer matrices (PTMs) [9-10], and Markov random fields [5]. Sequential circuits have received far less attention, since simulating probabilistic behavior over multiple cycles further increases computational complexity. Prior work has used various simplifying assumptions such as limiting soft-error vulnerability to flip-flops [2-3] or only injecting soft errors to circuits in the first simulated cycle [11-12]. In addition, some design tasks [17] require simulating signal probabilities over hundreds or thousands of cycles. Therefore, a need exists for methods that are able to analyze probabilistic behavior over many cycles without sacrificing accuracy or running out of processor time or memory.

We propose a new logic-level technique for probability-based SER measurement in sequential circuits. Since the probabilities assumed in the literature are not always precisely defined, we begin with some definitions. Consider a sequential circuit *SC* that is being analyzed or simulated over *n* clock cycles. Let $v(k)$ denote the input vectors applied to *SC* in cycle *k*, where $1 \leq k \leq n$; the input vector is characterized by some input probabilities, which determine the probabilities of all signals appearing in *SC* at any time. The probability of a particular line or signal *s* in *SC* being 1 in clock cycle *k* is called its *signal probability*, and is denoted by $p(s(k))$. The occurrence of one or more errors *e* during the first *k* clock cycles can change $p(s(k))$ to $p(s^e(k))$. Thus the difference in signal probabilities of these two cases $|p(s(k)) - p(s^e(k))|$ indicates the overall impact of *e* on signal *s* in cycle *k*. We define *circuit error probability* (CEP) in the k^{th} cycle, denoted $CEP(k)$, as the average difference in signal probabilities between the error-free and erroneous cases over all *q* primary outputs *Z* of *SC*.

$$CEP(k) = \sum_{z\in Z} |\, p(z(k)) - p(z^e(k))| \,/\, q$$

CEP represents how likely an incorrect signal is to be observed at any primary output *z* in a particular cycle. The SER of a sequential circuit in each cycle will be measured in this paper by the corresponding CEP.

In [11-12], SER is defined in terms of the mean error susceptibility (MES), the density of high energy particles, or the density of transistors per unit area. The MES of a primary output *z* is the probability that an erroneous value is observed on *z* in a particular cycle, which is similar to CEP. However, unlike CEP, which is directly obtained from the signal probabilities, MES is calculated at the electrical (transistor) level in terms of the duty cycle, signal amplitude (voltage), and the gate error probability of individual gates associated with different input probabilities. In [3], the SER is evaluated at the logic level by the mean time to manifest error (MTTM), which indicates how long, on average, the first incorrect outcome will be observed, while the CEP and MES represent the probability that the circuit produce erroneous results in a particular cycle.

Most SER computations for sequential circuits assume that an error occurs only in the first clock cycle of an *n*-cycle simulation, and no new errors arrive in subsequent cycles. Hence, although the aftereffects of the initial error may linger, the simulated environment is implicitly assumed to be error-free in cycles 2 through *n*. To address this oversimplification, we introduce a general soft error model denoted $SE(k)$, which allows errors to occur in the first *k* clock cycles, where $1 \leq k \leq n$ and *n* is the number of simulation cycles. The $SE(n)$ error model thus allows soft errors to occur in any cycle during the entire simulation period. Most existing error models, on the other hand, are of the $SE(1)$ type.

978-1-4244-6650-4/10/$26.00 ©2010 IEEE

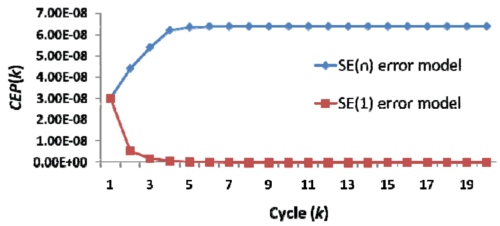

Figure 1. Circuit error probabilities for the S298 benchmark assuming a gate output error probability of 10^{-7}.

Figure 1 demonstrates that the chosen value of k in $SE(k)$ can have a dramatic impact on SER calculations, even for small sequential circuits. It shows CEP for the ISCAS-89 benchmark S298 under typical and identical simulation conditions with $k = 1$ (the standard model) [12] and $k = n = 20$. Observe that the two predicted behaviors are qualitatively as well as quantitatively different.

Another major source of inaccuracy is the assumption [2] that (non-erroneous) signal probabilities are uncorrelated. An example is shown in Fig. 2. Suppose that the signal probabilities of x_1 and x_3 are 0.3 and 0.7, respectively, and an erroneous 0/1 value is generated at input x_2. If the two fanout branches of x_1 are treated as uncorrelated, the estimated probability that the error signal propagates from x_2 to the output z is 0.063, whereas the exact value is 0.21. Thus, a computation error of 70% occurs in this case.

The common assumption that SEUs in sequential circuits only directly affect flip-flops while the combinational logic is error-free [2-3] can also lead to serious underestimation of SER. Yet another source of inaccuracy is inability to simulate circuits over many cycles. For example to evaluate SER for the n^{th} simulated clock cycle in [12], the original circuit needs to be unrolled n times, making the runtime exponential in the number of simulated cycles.

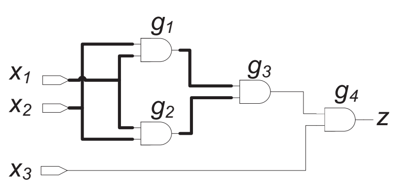

Figure 2 . Four-gate circuit with reconvergent fanout.

Thus can be seen that prior work on soft error analysis for sequential logic has several shortcomings that limit its applicability or accuracy:

- Soft errors are assumed to occur only in the first cycle
- Errors are restricted to flip-flops
- Structural and temporal correlations among error probabilities are not properly accounted for

We present a general probabilistic evaluation methodology that addresses all the above issues. Combinational logic is simplified by partitioning it into supergates [15], and a PTM is constructed for each supergate using a fast fault-simulation technique [1]. Supergates and PTMs allow precise representations of signal probabilities and their correlations. Signal probabilities are obtained by PTM multiplication with time-frame expansion. The approach is implemented in a tool called SAMPLE (Scalable and Accurate Matrix-based Probabilistic Algorithm for Logic Signal Estimation). As we demonstrate later, the SAMPLE technique can generate accurate results over many cycles, while fully accounting for structural and temporal correlations. Moreover, SAMPLE scales well to large sequential circuits, and its memory usage does not increase significantly with time.

Although electronic and latching-window masking effects are also important for accurate SER estimation, these two masking phenomena can only be calculated at the transistor level with considerable computation effort. Since a major goal of SAMPLE is to provide fast and high level (logic-level) soft error estimation, electronic and latching-window effects are not considered here. However, there are ways to integrate these two masking effects into the PTM framework. For instance, we can use the methods proposed in [4] and [8] to simulate latching-window and electrical effects in PTMs.

To validate the proposed probabilistic analysis methodology, we applied it to SER estimation in sequential circuits represented by the ISCAS-89 benchmarks. Our experimental results demonstrate that our algorithm is capable of processing even the largest benchmarks using relatively modest computation resources, and maintaining an average computation error of less than 3% over 100 simulation cycles.

In summary, we propose a powerful probabilistic methodology, which can simulate the situation where soft errors occur at flip-flops or logic gates in one or more cycles. In addition, it accounts for structural and temporal correlations, and is capable of processing very large circuits over many cycles while maintaining relatively high accuracy.

II. PTM BASICS

Since the proposed method makes use of PTMs, we briefly review the basic technique for signal probability calculation with PTMs.

A PTM for an r-input q-output component is a $2^r \times 2^q$ matrix M whose $(i,j)^{th}$ element is the probability of output j occurring in response to input i. The PTM of a fault-free component is called its ideal transfer matrix (ITM) and the probability of every correct output value is 1. Figure 3 shows a two-input AND gate, its ITM, and a PTM. In this gate PTM, p_1, p_2, p_3, and p_4 are *gate output error probabilities* associated with corresponding input vectors (rows). Each gate output error probability represents the probability that the gate produces an incorrect value when the corresponding input vector is applied. The PTM of a gate g is denoted by G, and its gate output error probability associated with an input vector v is denoted by $G^e[v]$. For instance, in this case, $G^e[(x_1, x_2) = (0, 1)] = p_2$.

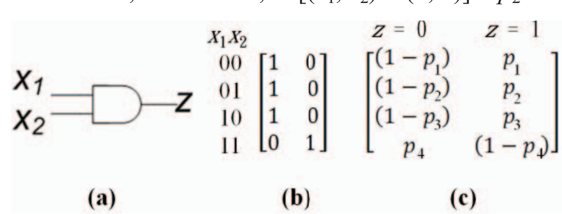

Figure 3. (a) Two-input AND gate, (b) its ITM, and (c) a PTM with various error probabilities for each input vector.

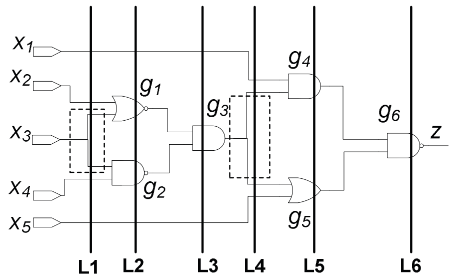

Figure 4. Circuits CC_1 demonstrating PTM construction; dashed lines enclose fanout gates.

PTM algorithms involve several types of matrix operations, one of which is the tensor product. Given an $a \times b$ matrix A and a $c \times d$ matrix B, their tensor product $TP = A \otimes B$ is an $ac \times bd$ matrix whose elements are:

$$TP(i_0 \ldots i_{n+p-1}, j_0 \ldots j_{m+q-1}) = A(i_0 \ldots i_{n-1}, i_0 \ldots j_{m-1}) \times B(i_n \ldots i_{n+p-1} \ldots j_{m+q-1}) \qquad (1)$$

For example,

$$\begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 & 0 & 2 & 4 \\ 3 & 4 & 0 & 0 & 6 & 8 \\ 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 3 & 4 & 0 & 0 \end{bmatrix} \qquad (2)$$

Next we describe how the PTM of an l-level combinational circuit can be constructed from the PTMs of its component gates and wires. First, derive ITMs or PTMs for all components. Then for each topological level i containing h component PTMs $\{M_{ij}\}$, form the level PTM $M_i = M_{i1} \otimes M_{i2} \otimes \ldots M_{ih}$ by repeated application of the tensor product. Finally, using ordinary matrix multiplication multiply all l level PTMs together to form the circuit PTM $M = M_1 \cdot M_2 \cdot \ldots \cdot M_l$.

The six-level circuit CC_1 in Fig. 4 shows how a circuit PTM is constructed. First, insert explicit wiring and fanout "gates" into CC_1 as needed. Then construct level PTMs M_1:M_6 for each level of logic. PTMs of a single wire and a j-branch fanout gate are denoted by the identity matrix I_2 and F_j, respectively, in the following symbolic representations. The level PTM of the first level L1 is given by

$$M_1 = I_2 \otimes I_2 \otimes F_2 \otimes I_2 \otimes I_2$$

The other five level PTMs M_2:M_6 can be calculated in the same way. The final circuit PTM is

$$M = M_1 \cdot M_2 \cdot M_3 \cdot M_4 \cdot M_5 \cdot M_6$$

which is a 32×2 matrix.

Once the overall PTM is known, the joint output signal probabilities J can be calculated very easily by multiplying the input-probability vector V (the set of given input probabilities) by the circuit PTM M thus:

$$J = V \cdot M \qquad (3)$$

For example, if all gates in CC_1 have the same gate output error probability $G^e[v] = 0.1$ associated with any input vector v, and all input signal probabilities are 0.5, CC_1's joint output signal probabilities are

$$J = [0.03125\ 0.03125\ \ldots\ 0.03125] \cdot M = [0.81, 0.19].$$

Since CC_1 is a single-output circuit, $J[0, 1]$ is the corresponding signal probability $p(z)$ of z.

The circuit PTM M of a single-output circuit with r primary inputs is a $2^r \times 2^1$ matrix. For such two-column PTMs, one entry associated with the input vector v (in the v^{th} row) is the probability that the circuit produces the correct result, while the other entry is the error probability with respect to v. We define the *circuit output error probability* associated with v as the entry $M[v, u]$ whose column u represents the incorrect output with respect to v, and denote it by $M^e[v]$. Like the gate output error probability, the circuit output error probability is also the conditional probability of one particular input vector. For a single-output combinational circuit CC whose ITM and PTM are I and M, respectively, if the input-probability vector is V, CC's joint output signal probabilities J_{ITM} and J_{PTM} can be obtained using (3). The CEP of CC is given by $|J_{ITM}[0, 1] - J_{PTM}[0, 1]|$.

Directly constructing a circuit PTM from its level PTMs may consume a very large amount of runtime and memory. To improve the efficiency, Krishnaswamy *et al*. [9-10] proposed using algebraic decision diagrams (ADDs) and developed several heuristics, which can reduce the complexity by several orders of magnitude. However, the modified PTM calculations are restricted to relatively small circuits because they still construct the circuit PTM by multiplying large level PTMs.

The foregoing PTM approach has the advantage of correctly combining all a circuit's signal probabilities, even when the signals are highly correlated due to fanout and subsequent reconvergence. Note that all logical masking effects are implicitly accounted for. This accuracy comes at considerable computational cost, however [9-10]. To reduce this cost, we develop a new PTM construction algorithm that combines a new partitioning method and fault simulation technique, and can accurately generate the circuit PTM, even with very small probability values.

III. Probabilistic Estimation Algorithm

To estimate signal probability efficiently in sequential circuits, we use two key concepts: circuit partitioning and multi-cycle probability calculation. We first partition the combinational part of the sequential circuit into its supergates and derive the circuit PTM. The gate signal probabilities for each cycle can be obtained by directly multiplying the circuit PTMs by the corresponding input probabilities.

A. Supergate Partitioning and Cone Clustering

A major challenge in signal probability computation is to estimate the output probability of a reconvergent structure while maintaining high accuracy. Several methods have been developed for this purpose [13-15]. Seth and Agrawal [15] first proposed the use of supergates for testability computation. For a single-output combinational circuit CC, the supergate of gate w, denoted $SG(w)$, is the smallest sub-circuit of CC whose output is w and whose inputs are independently controllable from the inputs of CC. A supergate typically encloses one or more reconvergent structures associated with its output. For instance, the supergates of g_6 and g_3, $SG(g_6) = \{g_4, g_5, g_6\}$ and $SG(g_3) = \{g_1, g_2, g_3\}$, are shown by dashed lines in Fig. 5.

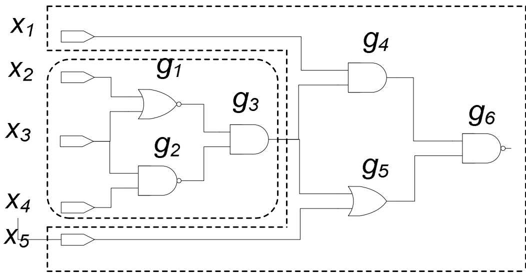

Figure 5. The circuit of Fig. 4 with its two supergates marked by dashed lines.

Because supergate inputs are independent, complete accuracy can be maintained in signal probability calculations by processing supergates instead of gates. Of course, it is possible that the entire original circuit is a maximal supergate, in which case no simplification results from supergate partitioning.

Some supergates may contain many inputs, *e.g.*, $r = 32$, and generating circuit PTMs for such cases may be infeasible. To guarantee that the size of all circuit PTMs is manageable, we apply a heuristic called "cone clustering" to each supergate for which $r > r_{max}$. The cone-clustering heuristic traverses a supergate *SG* from output to inputs. The initial cone *C* consists of the output gate *g* of *SG.* Gates connected to the inputs of *g* are added level by level to *C* until the number of primary inputs reaches $r_{\max}$. The final inputs of *C* become the outputs of new cones, which are similarly formed. The heuristic is executed recursively until all gates of *SG* belong to at least one cone. An example is shown in Fig. 6.

Some internal clustered cones may contain incomplete reconvergent structures, which can lead to inaccurate probability calculations. In practice, however, very few supergates need to be reduced by cone-clustering. For example, in the case of the ISCAS-89 benchmarks, only 0.9% of supergates have $r > 10$. To determine the value of r_{max} that makes the best trade-off between accuracy and efficiency, we carried out simulation experiments with various r_{max} values, and measured their accuracy by comparing the results with signal probabilities obtained by Monte Carlo sampling. According to these experiments, 10 inputs are accurate enough for all ISCAS-89 benchmark circuits. Our simulation data also show that, in most cases, we can maintain 98% accuracy over 100 simulated cycles.

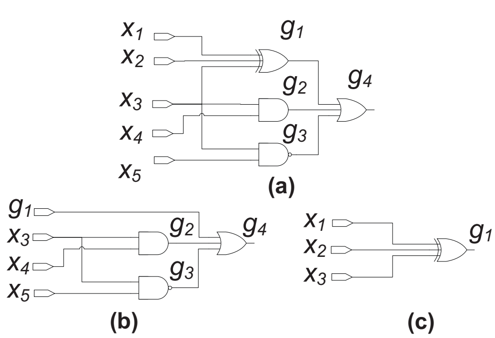

Figure 6. (a) Original circuit consisting of a single supergate; (b-c) the two sub-circuits resulting from cone clustering with $r_{\max} = 4$.

B. PTM Construction Method

After supergate partitioning and (if needed) cone clustering, we generate the PTM of each circuit partition. The basic PTM algorithm provides a way to generate the circuit PTM by constructing and merging level PTMs, which can preserve complete accuracy. Generally, the supergates are much smaller than the original circuit, thus making the basic construction procedure more efficient due to the fact that fewer level PTMs need to be calculated and merged in a supergate. However, many matrix operations may be required for the construction process and the computational complexity in terms of runtime and memory usage is exponential in the circuit size. As a result, the basic PTM construction algorithm is still limited to relatively small circuits, even when the circuits are partitioned into supergates, so a speedup heuristic is needed to enhance the efficiency of matrix construction.

Since supergates and cones are single-output sub-circuits, the corresponding PTMs are two-column matrices. As mentioned in Section II, constructing a two-column PTM is equivalent to calculating the circuit output error probabilities associated with all input vectors. In addition, recall that soft errors have very low occurrence probability, so we can assume that any observable circuit error is caused by a single erroneous gate, *e.g.,* a gate that receives a particle strike. In other words, the circuit error is mainly caused by individual erroneous gates at any time, and ignoring multiple-error probability has no significant impact on accuracy. Take ISCAS-89 benchmark S27 as an example. Suppose that all gate output error probabilities are 10^{-7}, and all multiple-error effects are ignored, the average calculation error for PTM entries compared with the exact results is 0.62%.

Finding all gates whose incorrect output value can flip the primary output is equivalent to identifying gates whose stuck-at-faults can propagate to the primary output. These may be called sensitized gates, where a gate is sensitized if there exists at least one path from the gate along which an error can propagate to a primary output. For a given input vector, only some stuck-at-faults can propagate errors to the primary output, while others are blocked by logic masking effects. One method for identifying sensitized gates is critical-path tracing (CPT) proposed by Abramovici *et al*. [1], which can quickly identify many gates sensitized by a given input vector.

Here, we introduce a CPT-based matrix construction heuristic called CPT-MC, which can efficiently generate circuit PTMs for cases with low gate output error probabilities. Let *M* be the PTM of a circuit partition. CPT-MC first identifies sensitized gates associated with each input vector by using the CPT method. Then for each input vector *v,* the circuit output error probability $M^e[v]$ of *M* is computed as

$$M^e[v] = \sum_{i\in\Omega} G_i^e[v_i]\cdot\left(\prod_{j\notin\Omega}(1 - G_j^e[v_j])\right) \qquad (4)$$

where Ω is the current set of sensitized gates, while $G_i^e[v_i]$ and $G_j^e[v_j]$ are the gate output error probabilities of g_i and g_j associated with the current input vectors v_i and v_j, respectively. For instance, the sensitized gates associated with the input vector $v = (0, 1, 0)$ of the circuit shown in Fig. 7 are g_4 an g_6, and the corresponding circuit output error probability is

$$M^e[v] = G^e_4[v_4](1-G^e_6[v_6]) + G^e_6[v_6](1-G^e_4[v_4]) = 1.52\times 10^{-6}$$

This procedure is recursively applied until the circuit output error probabilities associated with all input vectors are obtained.

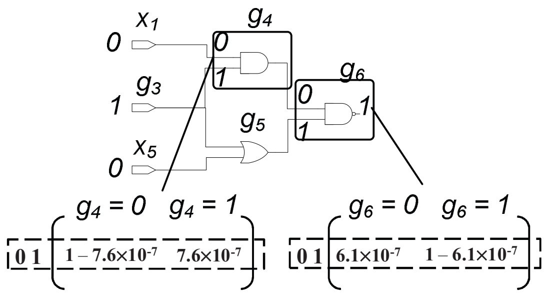

Figure 7. Supergate of g_6 with input vector (0, 1, 0) and sensitized gates g_4 and g_6

C. Probability Evaluation for Sequential Circuits

After calculating the circuit PTM, its output signal probabilities are obtained by SAMPLE from the initial input probabilities and time-frame expansion. SAMPLE first applies a given input probability to each primary input. After the first cycle, the signal probabilities stored in flip-flops are updated to the corresponding secondary inputs at the beginning of the next cycle. Within each cycle, SAMPLE estimates the primary and secondary output signal probabilities of each sub-circuit in topological order by multiplying its input-probability vector by the circuit PTM according to (3). Finally, the signal probabilities of the secondary outputs are stored in the corresponding flip-flops at the end of the cycle. Pseudo-code for the resulting probability estimation algorithm is in Fig. 8.

Since a circuit PTM contains the output signal probability associated with each input combination, and SAMPLE preserves the probability of each flip-flop in every cycle, the circuit probability behavior can be fully expressed in terms of its circuit PTMs and the corresponding input probabilities. In other words, the functionality of the target circuit's transition function can be completely represented by the circuit PTMs, and all temporal correlations are implicitly examined by this estimation procedure.

```
SAMPLE(sequential circuit SC, no. of simulation cycles n,
gate PTM configuration GP_conf, max. no. of inputs r_max of
a supergate)
SG_List = Supergate_Partitioning(combinational part of
SC)
for each sg in SG_List
   if (no. of inputs > r_max)
      Rcone_List = Rcone_Clustering(sg)
      SG_List = SG_List – sg + Rcone_List
PTM_List = CPT-MC(SG_List, GP_conf, r_max)

Initialize all input probabilities
for i = 1 to n
   for each PTM M in PTM_List in topological order
      Construct input-probability vector V of M
      Output signal probability J = V·M
   Update flip-flop values at the corresponding secondary
inputs
END SAMPLE
```

Figure 8. Probability estimation algorithm.

IV. SER ESTIMATION

To evaluate the efficiency and accuracy of the proposed algorithm, we applied SAMPLE to the calculation of SER for the ISCAS-89 benchmark circuits.

A. SER Measurement

Accurately calculating the impact of a soft error for all feedback loops of a circuit over several cycles is difficult. Prior work on this problem uses the *SE*(1) model and various other heuristic simplifications to reduce the computational complexity. As discussed in Section I, we measure the SER of a sequential circuit by its CEP, which can be directly obtained from its output signal probabilities. SAMPLE can efficiently calculate the signal probability even for very large sequential circuits. Moreover, SAMPLE is able to simulate any *SE*(k) error model $1 \le k \le n$, and all logical masking effects are automatically and implicitly included in the calculations.

B. Experimental Results

To estimate the performance of the proposed probabilistic estimation method, we used SAMPLE to evaluate the SER of all ISCAS-89 circuits, assuming two representative gate output error probabilities associated with any input vector v: $G^e[\underline{v}] = 10^{-3}$ and 10^{-7}, and simulating $n = 100$ cycles. The first case shows that SAMPLE can maintain high accuracy even with the relatively high gate output error probability of 10^{-3}. The second case simulates the situation where a gate has the relatively low error probability of 10^{-7} of producing an erroneous value in every cycle. The experiments were performed on an Intel Quad-Core, 2.35 GHz, 64-bit PC, with 4GB RAM. They produced the accuracy, runtime and memory data appearing in Table 1. In order to validate the accuracy of the SER evaluation results, we independently estimated the CEPs of all circuits by the Monte Carlo method, and compared the results with those of SAMPLE. For each circuit, we repeatedly apply input vectors that are randomly generated with equal probabilities of being 1 and 0 to all primary inputs, and simulate the circuit until the signal probability converges to fixed values.

	PTM runtime (s)		Memory	Error
Circuit	Construction	Eval'n.	usage (MB)	(%)
S298	0.21	<0.01	0.02	1.42
S510	3.79	0.04	0.06	1.09
S713	2.79	0.06	1.28	1.09
S838	0.62	0.02	0.44	1.83
S953	2.07	0.05	0.37	2.14
S953	2.06	0.02	0.65	6.71
S5378	6.59	0.16	5.24	0.43
S1423	18.22	0.20	7.82	0.53
S1238	37.57	0.05	2.03	0.93
S9234	1,075.16	0.74	24.70	4.71
S13207	138.24	2.82	115.26	2.31
S15850	614.92	10.70	510.86	4.31
S35932	987.80	4.42	7.43	2.63
S38417	1,796.79	83.57	55.40	2.97
S38584	2,155.85	352.87	171.04	3.72

Table 1. Performance results for the ISCAS-89 benchmark circuits, including the largest circuits.

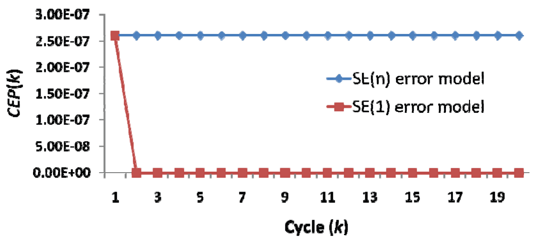

Figure 9. Circuit error probabilities for the S1238 benchmark assuming a gate output error probability of 10^{-7}.

Column 5 of Table 1 indicates that the maximum and average errors of the SER analysis with the *SE*(*n*) model over *n* = 100 cycles are 6.71% (S953) and 2.16% respectively. The maximum runtime and memory usage are 0.69 hours and 510 MB (S15850) for 100-cycle simulation.

Figure 1 (Section I) and Fig. 9 show four typical SER behaviors for the *SE*(1) and *SE*(*n*) error models in representative sequential circuits with primary input signal probability = 0.5 in every cycle. These two sets of results serve to illustrate two probabilistic behaviors of a sequential circuit with different boundary conditions. As Fig. 1 shows, the SER with the *SE*(1) model decreases dramatically in the first few cycles, which implies that most soft errors have a high chance of quickly escaping from the circuit. The SER with the *SE*(*n*) error model in Fig. 1 rises in the first few cycles and grows much more slowly thereafter. Furthermore, the stable value (steady-state) with *SE*(*n*) in Fig. 1 is almost an order of magnitude higher than that with *SE*(1). This suggests that, in certain cases, the accumulated effect of soft errors has great impact, and should not be ignored. In other words, the *SE*(1) model may underestimate the SER after just a few cycles, and lead to unrealistic prediction of overall soft error behavior.

Some pipeline-structured circuits such as S1238 contain no global feedback loops and so have different SER behavior from most other sequential circuits. SER results for S1238 with the *SE*(1) and *SE*(*n*) models are shown in Fig. 9. The SER with the *SE*(1) drops sharply to zero at the second cycle because no captured soft error can propagate to any secondary input in the next cycle. Therefore, any soft error that appears in a particular cycle will disappear in the next cycle. Similarly, since no global feedback is present in such circuits, the SER with the *SE*(*n*) model remains constant in each cycle. In other words, the SER behavior of such pipeline-structured circuits is similar to that of a combinational circuit.

V. CONCLUSIONS

We have presented a new probabilistic calculation tool for sequential circuits, SAMPLE, which can be used for accurate modeling of non-deterministic effects, and scales well to very large circuits. Prior work on this issue tends to be inaccurate and does not scale well. Unlike some prior methods, SAMPLE is capable of simulating fairly realistic error situations, and also provides a fast and accurate estimation of SER by combining supergate partitioning, critical-path tracing, and the PTM technique. Moreover, it can analyze a sequential circuit over hundreds or thousands cycles without increasing memory usage dramatically, which is useful for applications require long-term circuit simulation. To determine SAMPLE's scalability and accuracy, we applied it here to the evaluation of SER. Our experimental results show that SAMPLE can efficiently analyze all the ISCAS-89 benchmarks with modest resources, while maintaining an average accuracy over 97% for 100-cycle simulation. They also show that different error modeling assumptions and initial input probabilities can dramatically affect SER simulation results for sequential circuits.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation under grant CCF-0702276.

REFERENCES

[1] M. Abramovici, P. R. Menon, and D. T. Miller: "Critical path tracing: an alternative to fault simulation," *IEEE. Design & Test*, pp. 83-93, 1984.

[2] H. Asadi and M. B. Tahoori: "An analytical approach for soft error rate estimation in digital circuits," *Proc. ISCAS*, pp. 2991-2994, 2005.

[3] H. Asadi and M. B. Tahoori: "Soft error modeling and protection for sequential elements," *Proc. DFT*, pp. 463-471, 2005.

[4] H. Asadi and M. B. Tahoori: "Soft error derating computation in sequential circuits," *Proc. ICCAD*, pp. 497-501, 2006.

[5] R. I. Bahar, J. Mundy and J. Chan: "A probabilistic based design methodology for nanoscale computation," *Proc. ICCAD*, pp. 480-486, 2003.

[6] A. Dutta and N. A. Touba: "Iterative OPDD based signal probability calculation," *Proc. VTS*, pp. 72-77, 2006.

[7] R. Kodavarti and M. R. Mercer: "Signal probability calculations using partial functional manipulations," *Proc. VTS*, pp. 194-200, 1993.

[8] S. Krishnaswamy, I. L. Markov and J. P. Hayes: "Tracking uncertainty with probabilistic logic circuit testing," *IEEE Design & Test*, pp. 312-321, 2007.

[9] S. Krishnaswamy, G. F. Viamontes, I. L. Markov and J. P. Hayes: "Accurate reliability evaluation and enhancement via probabilistic transfer matrices," *Proc. DATE*, pp. 282-287, 2005.

[10] S. Krishnaswamy, G. F. Viamontes, I. L. Markov and J. P. Hayes: "Probabilistic transfer matrices in symbolic reliability analysis of logic circuits," *ACM TODAES,* vol. 13*,* Article No. 8, 2008.

[11] N. Miskov-Zivanov and D. Marculescu: "MARS-S: modeling and reduction of soft errors in sequential circuits," *Proc. ISQED*, pp. 893-898, 2007.

[12] N. Miskov-Zivanov and D. Marculescu: "Modeling and optimization for soft-error reliability of sequential circuits," *IEEE Trans. CAD*, pp. 803-816, 2008.

[13] T. Rejimon and S. Bhanja: "An accurate probabilistic model for error detection," *Proc. VLSI Design Conf.*, pp. 717-722, 2005.

[14] S. J. Russell and P. Norvig: *Artificial Intelligence−A Modern Approach*, Prentice-Hall, 2001.

[15] S. C. Seth and V. D. Agrawal: "PREDICT− probabilistic estimation of digital circuit reliability," *Proc. FTCS*, pp. 220-225, 1985.

[16] T. Uchino, F. Minami, T. Mitsuhashi and N. Goto: "Switching activity analysis using Boolean approximation method," *Proc. ICCAD*, pp.20-25, 1995.

[17] K. M. Zick and J. P. Hayes: "High-level vulnerability over space and time to insidious soft errors," *Proc. HLDVT*, pp. 161-168, 2008.

[18] B. Zhang, W. S. Wang and M. Orshansky: "FASER: fast analysis of soft error susceptibility for cell-based designs," *Proc. ISQED,* pp. 27-29, 2006.