FASER: Fast Analysis of Soft Error Susceptibility for Cell-Based Designs

Bin Zhang, Wei-Shen Wang and Michael Orshansky

Department of Electrical and Computer Engineering, University of Texas

Abstract

This paper is concerned with statically analyzing the susceptibility of arbitrary combinational circuits to single event upsets that are becoming a significant concern for reliability of commercial electronics. For the first time, a fast and accurate methodology FASER based on static, vector-less analysis of error rates due to single event upsets in general combinational circuits is proposed. Accurate models are based on STA-like pre-characterization methods, and logical masking is computed via binary decision diagrams with circuit partitioning. Experimental results indicate that FASER achieves good accuracy compared to the SPICE-based simulation method. The average error across the benchmark circuits is 12% at over 90,000X speed-up. The accuracy can be further improved by more accurate cell library characterization. The run-time for ISCAS'85 benchmark circuits ranges from 10 to 120 minutes. The estimated bit error rate (BER) for the ISCAS'85 benchmark circuits implemented in the 100nm CMOS technology is about 10^{-5} FIT.

1. Introduction

Reliability of commercial electronics with respect to the single event upsets (SEU) caused by extrinsic radiation is becoming a significant concern. Historically, the most significant impact of SEU was on memory units (latches, flip-flops, registers, and arrays). However, as the transistor feature size scales down, the error rate due to single event upsets in the combinational logic becomes substantial. It is predicted that by 2011, the soft error rate (SER) due to combinational logic may be comparable to that of the memory units [1]. Because of the increasing error rates in combinational circuitry, new tools and analysis methodologies are needed to ensure circuit reliability. System designers, micro-architects, and circuit designers need accurate prediction of error rates in the designed components. Having this capacity is a prerequisite for choosing the proper hardening strategy for the design.

A soft error may occur when a high-energy particle, typically, an alpha particle, or a neutron, hits the diffusion regions of an MOS transistor and produces charge that leads to a faulty transition. The pulse will cause an error only if it successfully propagates to the latching element and is latched at the clock arrival (sampling) time. There are several mechanisms that reduce the overall likelihood of the pulse producing an erroneous value at the memory units, making the actual SER substantially lower than the raw particle strike rate. In the literature [1-6], these mechanisms are referred to as electrical masking, logic masking, and latching-window masking.

In this paper, we propose an efficient and accurate approach for SER analysis of cell-based designs. The efficiency is achieved by resorting to symbolic representation of the error pulses using binary decision diagrams (BDD). The accuracy is guaranteed by relying on the precise description of the non-linear gate transfer characteristics using the SPICE-based pre-characterization of the cells in the library. In addition to the electrical properties of the cells, the logic structure of the circuit also has a significant impact on the SER. Failing to account for logic masking may over-estimate SER by order of magnitude (25X for a tree-structured circuit with logic depth 7), as illustrated in Figure 1. It is evident that as the logic depth increases, logic masking plays a more important role. *Accurately yet efficiently accounting for the reduction of error rate likelihood due to these masking mechanisms is the focus of this work.*

Prior work in this area has concentrated on modeling and describing the particle interactions at the very low nuclear level [7], performing device-level simulations to predict the electrical response of individual transistors to a particle strike [8], and performing circuit-simulation of a small set of gates to model the propagation of pulses [1]. Several authors

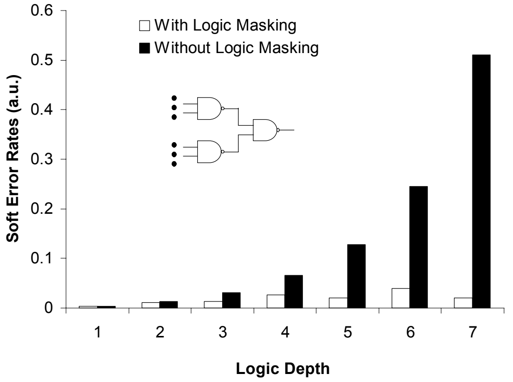


Figure 1. Soft error analysis ignoring logic masking can overestimate soft error rates by up to 25X.

have addressed the problem of SER analysis for general combinational logic [2-4]. Accurately estimating the SER due to particle strikes on combinational logic gates represents a significant computational challenge. The primary reason is that SER de-ratings due to electrical, logic, and latching-window masking are all input vector dependent. Existing techniques approach this problem by explicitly enumerating all input vectors, or a set of randomly picked input vectors [2-4]. However, the size of the input vector space is exponential in the number of primary inputs, and for circuits with a large number of primary inputs these techniques usually take hours, or even days, to achieve reasonable accuracy [4].

The rest of the paper is organized as follows. Section 2 describes the cell characterization procedure. In Sections 3, 4 and 5, we discuss the static analysis of SER. Section 6 presents the experimental results, and we draw conclusions in Section 7.

2. Cell Library Characterization

The proposed static SER analysis methodology FASER is targeted towards the use with the cell-based design methodology. Accurate library characterization is thus a key consideration. The two essential characterization steps are *pulse generation* and *pulse attenuation* (*propagation*).

A high-energy particle striking a node deposits charge which leads to a time-varying voltage pulse of a certain magnitude and shape. The characteristics of the pulse are dependent on the specific transistor network of each gate. Thus, the goal of the library characterization is to predict for every library gate the waveforms produced at the cell output for particle strikes at each vulnerable region. The current flow created by the charge deposited into the node is modeled as a single exponential for cosmic-ray related soft errors [1][4] (alternative models for alpha-particle related soft errors are also in existence [17-19]):

$$I(q,t) = \frac{2q}{\sqrt{\pi}T_s}\sqrt{\frac{t}{T_s}}e^{-\frac{t}{T_s}} \qquad (1)$$

where q is the collected charge and T_s is the technology-dependent charge-collection time constant. Collected charge q depends on the particle energy, and follows an exponential distribution [1][4].

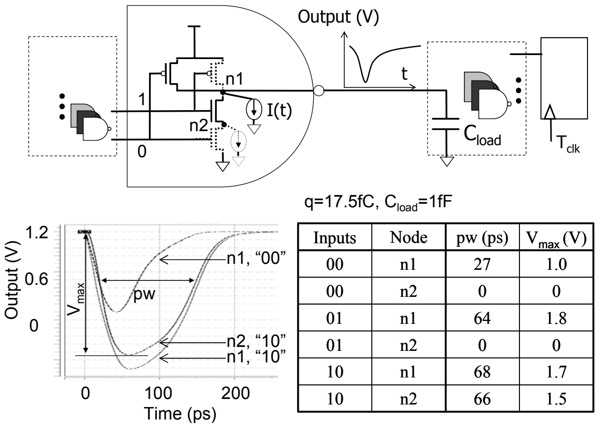
Figure 2 shows the SPICE simulation setup for characterization of pulse generation, where every vulnerable node is taken into account. The simulations are performed for a range of charge values (q) and load capacitances (C_{load}). The voltage pulses produced at the cell output by a specific charge deposited on an intra-cell node strongly depends on the biasing condition determined by the input vector. In the example circuit of Figure 2, both n1 and n2 are sensitive if the input vector is "10" (the pulse generated at n2 is slightly attenuated by the transistor above it), while only n1 is sensitive if the input vector is "01" or "00". When the input vector is "00", the pull-up network has the smallest resistance, resulting in the smallest falling pulse generated.

Existing tools [1-6] either ignore the effects of biasing conditions, or assume the worst case biasing conditions for every gate in the circuit. Experiments show a difference of 1.5X - 4X between the SPICE simulation result and the analysis performed under the worst-case assumptions. Clearly, accurate soft error analysis tool needs to consider different biasing conditions of the gates.

The voltage pulse produced at the cell output is approximated by a trapezoidal waveform, and are captured by two parameters, pulse width (*pw*) measured at $0.5V_{DD}$ and maximum voltage value (V_{max}). The rise and fall times of the trapezoidal waveform are chosen to be typical values.

After a transient faulty pulse is generated, it propagates toward the primary outputs of the circuit. In the course of its propagation, the pulse's electrical properties, such as width and magnitude, evolve as a result of the low-pass characteristics of the gates it propagates through. Short pulses tend to be attenuated, while long pulses tend to maintain their original width and magnitude after passing through a combinational logic gate. Figure 3 shows the SPICE simulation setup for characterization of this dynamic transfer function, where *pw* and V_{max} of the output pulse is found as a function of the input pulse. While different input-pin to output paths may be characterized by somewhat different transfer characteristics, this is a secondary effect, which we have for now ignored.

Pulses may re-converge and overlap at a gate in a circuit if multiple paths exist between the particle-striking point (fault-site) and the gate. The interaction of two pulses arriving simultaneously can be modeled. Currently, characterization captures only the first-order effect of pulse-overlapping with the output produced by simple superposition, followed by low-pass filtering by the gate's dynamic transfer function. The error of estimation for circuit SER due to this approximation appears to be minor compared with SPICE simulation for the benchmark circuits we have tested.



Inputs	Node	pw (ps)	V_{max} (V)
00	n1	27	1.0
00	n2	0	0
01	n1	64	1.8
01	n2	0	0
10	n1	68	1.7
10	n2	66	1.5

Figure 2. Pulse generation is characterized by circuit simulation with SPICE. The table shows the pulse produced at the output of NAND gate.

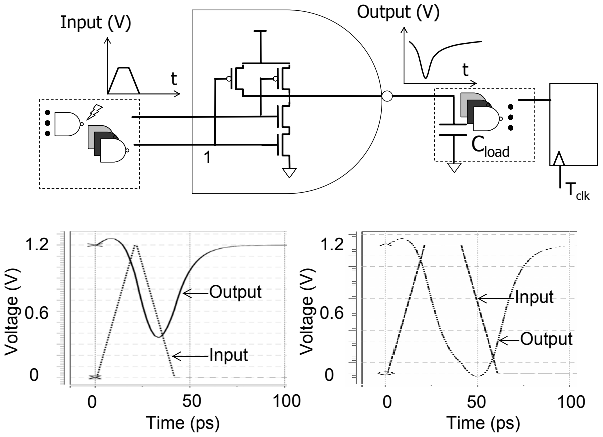


Figure 3. Pulse propagation is characterized by circuit simulation with SPICE. The curves show the low-pass characteristic of NAND gate.

3. Static Analysis of Fault Events Propagation

FASER is a static SER analysis methodology in that it relies on the *implicit enumeration* of the input vector space. The algorithm *formally encodes and propagates the error pulses using binary decision diagrams*. Binary decision diagrams are a powerful data structure proposed by Bryant [9] for efficient representation and manipulation of Boolean functions. By propagating the fault-encoding function to the primary outputs the algorithm can accurately predict output error probabilities. The error propagates only if the path from a fault-site to the output is sensitizable under the specific assignment of side inputs to the gates. The proposed BDD-based symbolic error manipulation algorithm succeeds in effectively capturing such logical masking [3][4]. However, the formation and propagation of the pulses symbolically is intrinsically linked in the algorithm with the accurate characterization of cell electrical properties, contained in the library. This guarantees accurate modeling of electrical masking [3].

The BDD describing the Boolean function at a given node in an error-free environment is termed *static* BDD. It is constructed using the classic rules of [9]. A particle strike at a node creates a transient pulse that can be represented by modifying the static BDD. Such a data structure is referred to as *event BDD*. In the event BDD, the terminal vertices encode both the error pulses and the original static logic values. The event BDD encoding will contain the arrival time (AT), the maximum voltage (V_{max}), and the width of the pulse (pw). Figure 4 shows fault-encoding with event BDD for biasing condition "10". If the pins of the gates are not primary inputs, the BDD describing the Boolean function of the biasing condition is found first, which is then modified to contain a pulse at one of its terminal vertex to become an event BDD.

Constructing the output event BDD for an operation on two input event BDDs is a recursive process similar to that of constructing the static BDD, which utilizes the standard BDD operations [9]. The operations are different only in how the terminal vertices are processed. Specifically, when the terminal vertex of one operand is reached, we check if the state of the output can be determined. If it can, a terminal vertex for the output event BDD is generated. Otherwise, a non-terminal vertex for the output is generated, and the event BDD of the other operand is searched one level deeper. Determining the state of the output is through logic operation and table look-up from the library. Logic operation is performed, for example, if one operand has a controlling value and has no pulse, in which case, the output value is determined regardless of the state of another operand (logic masking). Table lookup is performed when the analog characteristics of the output pulse is to be determined (electrical masking).

As noted in the previous section, different biasing conditions may result in very distinct output pulse, sensitive area and hence latching error probability due to particle strikes at a given gate. In order to achieve near SPICE-level accuracy, this dependence needs to be taken into account during the analysis, which requires enumerating all gate biasing conditions and all intra-gate nodes. This clearly increases the computational burden on the algorithm since for each biasing condition and each intra-gate node an event BDD is now generated at the fault site and propagated to the latches. We have found that this is crucial for accuracy improvement and that the penalty is affordable in most cases. Indeed, assuming that the average cell fan-in is k, the increase in complexity due to this enumeration is $O(k2^k)$. Since k is typically between 1 and 3, the cost is manageable.

Propagating the fault events *statically* is equivalent to constructing the event BDDs for the circuit nodes in the fan-out cone of the fault-site where the particle-strike occurs. The event BDD of a circuit node is simply its static BDD if it is outside the fan-out-cone of the fault site, since no error-pulses will occur at the node.

To illustrate fully the working of the algorithm, consider a small circuit example of Figure 5. To simplify the discussion, the pulse magnitude is ignored and only the pulse width is taken into account. Given the collected charge, an event BDD is generated for each biasing condition and intra-gate node of the fault site (node M) is constructed. Due to electrical masking, the pulse width changes along the propagation. The error-pulse is logically-masked when B=1. The event BDD at node X is the same as its static BDD because the pulse at node X is too small due to re-convergence to reach the gate threshold voltage.

4. Algorithm Flow and Latching Probability Computation

Ultimately, the static analysis of FASER is based on computing the probability of an error at the latch due to the totality of pulses propagating towards primary outputs. First, it is assumed that a particle can strike every node (diffusion region) in the circuit with the probability given by the ratio of the node area to total area. Second, the primary inputs remain stable. The validity of this assumption for SER analysis was demonstrated in [2]. Third, the equilibrium probabilities of the primary inputs are known

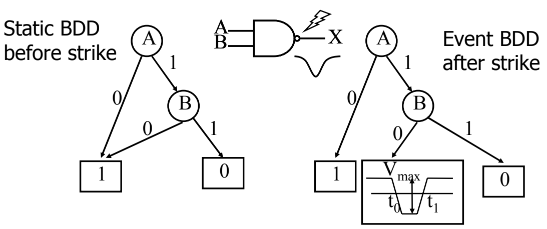


Figure 4. Fault-encoding with event BDD for the biasing condition "10". The strength of the pulse depends on the biasing condition and the strike location within the gate.

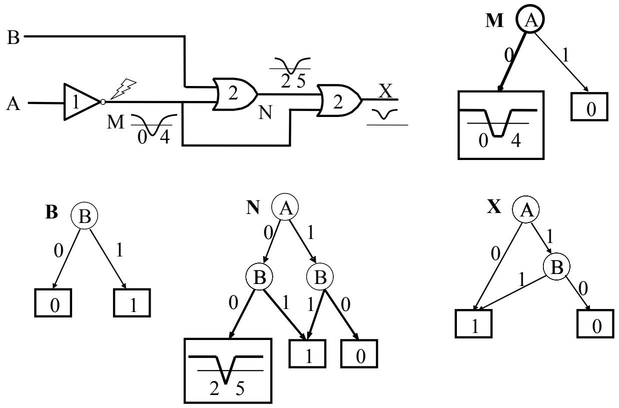


Figure 5. Pulse propagation in a simple circuit. Numbers inside the gates are their propagation delays. Terminals of event BDDs contain pulse propagation delays and durations. Pulses encoded with event BDDs of N and M cancel each other, resulting in no pulse at X.

and independent of each other. This last assumption has been successfully applied for power estimation and circuit reliability assessment [10]. Soft error estimation for circuits with strongly correlated primary inputs will be our future work.

The core of the algorithm is to find the conditional latching error probability *P*(*q, bc ,i, j, k*), given the collected charge *q*, biasing condition *bc* of the victim gate (fault site), intra-gate node *j* of gate *i*, and latch *k*. Calculating *P*(*q, bc ,i, j, k*) is discussed in the next paragraph. The contribution to the bit error rate (BER) of latch *k* by gate *i* is,

$$BER(i,k) = \sum_{j} \sum_{bc} \sum_{q} P(q,bc,i,j,k)(R\,(q,i,j)\Delta q) \qquad (2)$$

where $R\,(q,i,j)\Delta q$ is the strike rate for collected charge in the range of q and $q+\Delta q$, which is proportional to the area of node j of gate i. We use the average BER of all output latches by particle strikes on all gates as a merit of a circuit's soft error susceptibility. However, other criteria, such as the largest BER of the latches, can be used as well, depending on the application.

Propagating an event BDD to the primary output gives us a reliable measure of the occurrence probabilities and strengths of the pulses that will appear at the latch inputs. However, the latching error probability is linked to another masking mechanism, known as latching window masking. Latching window masking occurs due to the *temporal randomness* of the particle strike time [1-4], and the realization that the pulse arrival time at the latch has to be within the latching window for the error to occur. Assuming a uniform strike-time probability, the actual latching probability for a pulse is:

$$PL = \frac{pw - w}{T_{clk}} \qquad (3)$$

where PL is the latching probability, pw is the width of the faulty pulse present at the input of the latch, w is latching-window size of the latch, and T_{clk} is the clock period. Given an event described by the set of parameters (*q, bc ,i, j, k*), the event BDD at the primary output *k* is constructed first (Figure 6). With the assumption that the primary inputs are independent of each other, each edge of the event BDD is assigned a probability, based on the primary input the edge corresponds to. By recursively traversing the event BDD, the probability for an event contained in a terminal v to occur, $p(v)$ can be calculated [10]. The conditional latching error probability *P*(*q, bc ,i, j, k*), is then

$$P(q,bc,i,j,k) = \sum_{v} p(v)PL(v) \qquad (4)$$

where $PL(v)$ is the latching error probability of the pulse contained in terminal v of the event BDD, determined by (3).

5. Circuit Partitioning for Speed-Up

It is well known that the worst-case complexity of the BDD encodings of logic functions is exponential in the number of variables [9]. To make manipulation of BDDs efficient, a partitioning heuristic is adopted. This is a common practice in CAD techniques that use BDDs [14][15]. The use of partitioning allows a significant speed-up without a noticeable loss of prediction accuracy.

The circuit is partitioned into smaller domains as shown in Figure 7. Some nodes are designated to be pseudo primary inputs, and serve as the boundary between the partitions. Signal correlations are only considered within the domains. To estimate the latching error probability due to a particle-strike on a particular gate, an event BDD is generated at the fault site and propagated to the boundary nodes, where the pulse occurrence probabilities are estimated by traversing the event BDDs. Next, these pulses are treated as being generated at the boundary nodes and assumed to be independent of each other. In principle, these secondary pulses can be independently propagated further to the latches. The latching error probability due to particle strike at the fault site is approximated by the sum of the latching probabilities of the secondary pulses weighted by their respective occurrence probabilities. In practice, if we process the fault sites backward starting from the gates closest to the latches, the latching probabilities of the secondary pulses can be directly estimated from the latching error probabilities due to particle-strike at the boundary nodes, without further propagation of the secondary pulses.

```
Generate static BDD for every circuit node;
$BER \leftarrow 0$;
for all combinations of $(q, bc, i, j, k)$
    1. Retrieve the generated pulse shape for $(q, bc, i, j, k)$,
       and generate an event BDD at the fault site $i$.
    2. Propagate the event BDD in the fan-out cone of fault
       site $i$.
    3. The event BDD at the primary output $k$ (input of
       latch $k$) is traversed to find $P(q, bc, i, j, k)$.
    4. $BER \mathrel{+}= P(q,bc,i,j,k)(R\,(q,i,j)\Delta q)$
end
$BER \leftarrow BER / num\_of\_latches$;
```

Figure 6. Pseudo-code of FASER flow.

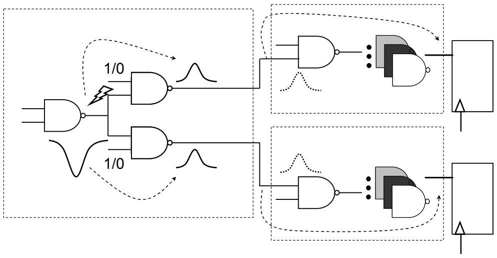


Figure 7. Partitioning circuits for speed-up leads to the loss of correlations between pulses of different domains. The loss of accuracy is minimal for partition size beyond a certain point.

We define the partition size as the maximum number of primary/pseudo primary inputs of each domain. A larger partition size allows a more global account of signal correlations but at the cost of a rapidly growing run time and memory usage. The tradeoff between speed and accuracy is performed by adjusting the partition size. We believe that the most significant impact of signal correlation on pulse propagation occurs in the neighborhood of the fault site. Therefore, the improvement of accuracy beyond a certain partition size (typically, 15-20) is expected to be minimal, and this is confirmed by the experimental results.

6. Experimental Results

The static SER analysis tool FASER was implemented in C++. The experiments investigated the accuracy of the static technique, the runtime of the algorithm, and the speed-accuracy trade-off using the partition heuristic. FASER takes a technology-mapped netlist, equilibrium probability of the primary inputs, clock period, and flux rate of the high-energy neutrons, and gives BER of the latches at the primary outputs. The widely utilized flux rates for New York City were used for analysis [16]. The SPICE technology files were based on the Berkeley Predictive Technology Model (BPTM) for the 100nm technology [11].

In order to verify the validity of the FASER, an experiment based on SPICE simulation was utilized. Since SPICE simulation is very time-consuming, we were only able to perform the tests on small artificially constructed benchmark circuits, with the largest circuit containing 35 gates.

The SPICE simulation is designed to measure the latching error probability given a particle strike with a random data set of (collected charge q, strike time t, gate i, node j, input vector V). Collected charge q follows an exponential distribution [1][4]. Strike time t is uniformly distributed between 0 and T_{clk}. The probability for a strike to occur in node j of gate i is proportional to node j's area. We assume that all input vectors have equal probability.

The experiments were conducted as follows. For each data set of (q, t, i, j, V), a current pulse with magnitude corresponding to the collected charge q and polarity corresponding to node j's diffusion type is injected to node j of gate i at time t, with input vector V. Voltage samples are taken at the latch output at T_{clk} and $2T_{clk}$. If either value did not match the correct one, an error is declared. Under this set-up, the conditional latching error probability is equal to the number of errors divided by the total number of simulations. The run time of the SPICE simulations ranges from 15 minutes to 45 minutes for the artificial benchmark circuits. FASER takes into account 5 different pulse strengths and its run-time for every test circuit is less than 0.01 seconds, giving a speed-up of over 90,000X. All experiments are conducted on a Dell GX260 workstation running Redhat Linux. The latching error probabilities of the benchmark circuits are compared in Figure 8. The average error between the two sets of data is 12% and can be well attributed to the simplified 2-parameter modeling of the error pulse in cell library characterization. Figure 9 shows the latching error probability due to each gate in circuit C1.

FASER with circuit partitioning heuristic is validated on the ISCAS'85 benchmark circuits. Partition size is expressed in terms of the maximum number of primary/pseudo primary inputs of the pulse propagation domains. Experimental results in Table I show that good accuracy can be achieved with relatively small partition size. The run-time varies between 22 seconds and 638 seconds for partition size 15 (Table II). The runtime increases rapidly with partition size as shown in Table II. However, improvement in accuracy is very small for partition sizes beyond 15. The estimated BER for 100nm CMOS technology is on the order of 10^{-5} FIT, where 1 FIT is defined as 1 failure in 10^{9} hours. It is to be noted that the partition size is not the sole factor that affects the run time. The circuit structure and the choice of pseudo primary inputs in partitioning also can greatly affect the BDD size, which is a well known property of BDD.

As a proxy of the memory usage, the maximum BDD size in terms of the number of vertices of the BDD is measured and shown in Table II. The general trend is that the BDD size increases drastically with partition size. The increase of BDD size with respect to the circuit size under the same partition size is due to the increased complexity of the nodal functions.

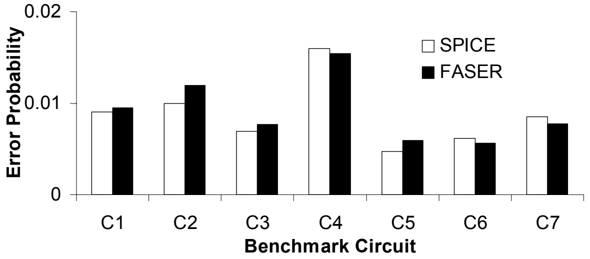


Figure 8. Error probabilities by FASER and SPICE simulation. The average error is 12%.

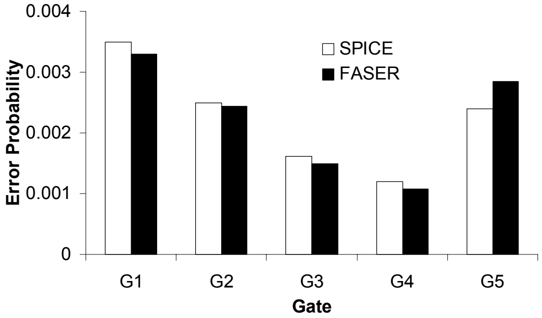


Figure 9. Latching error probability due to each gate in circuit C1.

Table I. Bit Error Rates for ISCAS' 85 benchmark circuits with different partition sizes (Np)

Circuits	Bit Error Rates (10^{-5} FIT)		
	Np=15	Np=20	Np=30
C432	3.0	3.2	3.1
C499	2.0	2.0	2.0
C1908	2.2	2.1	1.9
C1355	2.0	2.0	2.0
C3540	2.6	2.6	2.4
C5315	1.1	1.1	1.1
C7552	1.9	1.8	1.8

7. Conclusions

In this paper, we proposed a fast static soft error analysis tool FASER. Accurate models are based on STA-like pre-characterization methods, and logical masking is computed via binary decision diagrams with circuit partitioning. Experimental results indicate that the FASER achieves good accuracy compared to the SPICE-based simulation method. The average error across the benchmark circuits is 12% at over 90,000X speed-up.

Future work will focus on estimating the soft error rates with strong correlations among the primary inputs. The correlations of soft errors at the primary outputs will also be investigated.

8. Acknowledgment

This research was supported in part by GSRC, SRC, NSF, SUN, and Intel.

9. References

[1] P. Shivakumar, *et al*, “Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic,” *Proc. DSTN*, pp. 389-398, 2002.

[2] H. Cha, *et al*, “A Gate-Level Simulation Environment for Alpha-Particle-Induced,” *IEEE Trans. Computers*, Vol. 45, pp. 1248-1256, 1996.

[3] K. Mohanram, *et al*, “Cost-Effective Approach for Reducing Soft Error Failure Rate in Logic Circuits,” *Proc. ITC*, pp. 893-901, 2003.

[4] M. Zhang, *et al*, “A Soft Error Rate Analysis (SERA) Methodology,” *Proc. ICCAD*, pp.111-118, 2004.

[5] Q. Zhou, *et a*l, "Transistor Sizing for Radiation Hardening," *Proc. IRPS,* pp. 310-315, 2004.

[6] Q. Zhou, *et al*, "Cost-Effective Radiation Hardening Technique for Combinational Logic," *Proc. ICCAD,* pp. 100-106, 2004.

[7] P. C. Murley, *et al*, “Soft-error Monte Carlo modeling program, SEMM,” *IBM J. Res. Develop*., Vol. 40, pp. 109-118, 1996.

[8] P. Hazucha, *et al*, “Impact of CMOS Technology Scaling on the Atmospheric Neutron Soft Error Rate,” *IEEE Trans. Nucl. Sc*i., Vol. 47, pp. 2586–2594, 2000.

[9] R. Bryant, “Graph-based algorithms for Boolean function manipulation,” *IEEE Trans. Computers*., Vol. 35, pp.677-691, 1986.

[10] F. Najm, “Transition density, a stochastic measure of activity in digital circuits,” *Proc. DAC*, pp.644-649, 1991.

[11] BPTM, http://www-device.eecs.berkeley.edu/~ptm/.

[12] K. Mohanram, “Closed-form simulation and robustness models for SEU tolerant design,” *Proc. VLSI Test Symposium*, pp. 327–333, 2005.

[13] K. Mohanram, "Simulation of transients caused by single-event upsets in combinational logic," *Proc. ITC,* 2005.

[14] J. Jain, *et al*. “Functional partitioning for verification and related problems,” *Brown/MIT VLSI Conference*, 1992.

[15] D. Sahoo*, et al,* “A Partitioning Methodology for BDD-based Verification,” *Proc. FMCAD,* 2004.

[16] P. E. Dodd, *et al*, “Basic mechanisms and modeling of single-event upset in digital microelectronics,” *IEEE Trans. Nucl. Sci.* Vol. 50, pp. 583–602, 2003.

[17] P. Dahlgren, *et al*, “A switch-level algorithm for simulation of transients in combinational logic,” *Proc. Int. Fault-Tolerant Computing Symp.*, pp. 207-216, 1995

[18] G. C. Messenger, “Collection of charge on junction nodes from ion tracks,” *IEEE Trans. Nucl. Sci.*, Vol. 29, pp.2024-2031, 1982

[19] G. R. Srinivasan, *et al*, “Accurate predictive modeling of soft error rate due to cosmic rays and chip alpha radiation,” *Proc. Intl. Reliability Phys. Symp*., pp. 12-16, 1994.

Table II. Run-time and the maximum BDD size for the ISCAS' 85 benchmark circuits.

Circuit	Run-time (s)			Max BDD Size		
	Np=15	Np=20	Np=30	Np=15	Np=20	Np=30
C432	22	76	465	99	1223	86083
C499	39	63	129	101	145	404
C1908	66	86	1050	169	187	26393
C1355	40	62	119	101	145	406
C3540	149	195	5400	1028	1353	17861
C5315	278	546	1515	1372	6115	12100
C7552	638	780	7200	1813	6702	11602