Accelerating Soft-Error-Rate (SER) Estimation in the Presence of Single Event Transients

Ji Li
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, USA
jli724@usc.edu

Jeffrey Draper
Information Sciences Institute
Ming Hsieh Department of Electrical Engineering
University of Southern California
Marina Del Rey, USA
draper@isi.edu

ABSTRACT

Radiation-induced soft errors have posed an ever increasing reliability challenge as device dimensions keep shrinking in advanced CMOS technology. Therefore, it is imperative to devise fast and accurate soft error rate (SER) estimation methods. Previous works mainly focus on improving the accuracy of the SER results, whereas the speed improvement is limited to partitioning and parallel processing. This paper presents an efficient SER estimation framework for combinational logic circuits in the presence of single-event transients (SETs). A novel top-down memoization algorithm is proposed to accelerate the propagation of SETs. Experimental results of a variety of benchmark circuits demonstrate that the proposed approach achieves up to 560.2X times speedup with less than 3% difference in terms of SER results compared with the baseline algorithm.

1. INTRODUCTION

With the advent of nanoscale computing, soft errors have become one of the most challenging issues that impact the reliability of modern electronic systems at ground level for the semiconductor industry [1, 2]. A radiation-induced soft error occurs in a semiconductor device when the free mobile carriers generated by the passage of an energetic radiation particle are collected by the depletion region of a reverse-biased p-n junction [3, 4]. Consequently, a transient noise pulse is generated due to the momentary current flowing through the device [4]. This single-event transient (SET), if propagated through subsequent circuitry and captured by a storage element, becomes a single-event upset (SEU), i.e., a bit error [1, 5]. Such SEUs caused by the SETs are referred to as "soft" errors since there are no permanent damage to the hardware, and the rate at which they occur is called the soft error rate (SER). Failure to address radiation-induced soft errors can lead to silent data corruptions and system failures, with potentially disastrous results in mission-critical systems such as mainstream servers, automobiles and spacecrafts [6]. Therefore, logic circuits, especially those in critical applications, should have a certain degree of SET tolerance.

In recent years, drastic device shrinking, reduction of parasitic nodal capacitances, low operating voltages, and high operating frequency have added to the increase in the sensitivity of both memory and combinational circuits to the radiation-induced soft errors [5, 6, 7], posing a severe problem for system resilience. It is shown that the proportion of combinational logic soft errors at chip level increases with technology scaling and the combinational logic SER is comparable with the memory SER under high operating frequencies [8]. In addition, unlike errors in memory that can be corrected efficiently with the error code correction (ECC) techniques [5, 9], errors in combinational logic cannot be rectified without incurring significant area overhead and performance penalties [9]. In order to quantify the degree of SET tolerance in combinational logic, circuit-level SER estimation method is required. Many soft error estimation methods, such as FASTER [10] and SEAT-LA [11], have come into existence. However, the prior works have mainly concentrated on the accuracy of the SER estimation results, and the efforts to speed up the estimation process is limited to circuit partitioning, where the circuit of interest is divided into sub-blocks for parallel processing. Nevertheless, SER estimation process has become computationally expensive and the reason is twofold: (i) the SER estimation process is required to process a large set of parameters, in order to describe various complex effects, and (ii) runtime of the estimation process increases near-exponentially as the size and logic complexity of the circuit increases.

In this paper, we propose an efficient SER estimation framework, which significantly reduces the runtime with improved scalability and preserves the solution quality in terms of accuracy at the same time. The proposed SER estimation framework is comprised of two phases: (i) *characterization* and (ii) *propagation*. In the characterization phase, we adopt the double exponential current source model for SET pulse generation and extract SER estimation related parameters using HSPICE simulations. Various two-dimensional *lookup tables* (LUTs) are established to store the characterization results. All these characterizations need to be performed only once for each technology node, however, the propagation phase needs to be performed for each combinational circuit. In the propagation phase, the SETs are propagated from the particle striking sites towards the outputs (i.e., inputs of next stage flip-flops), and the final SER of the circuit is calculated based on the propagation results. In this paper, we propose a top-down memoization algorithm for accelerating the SET propagation. In the proposed al-

DAC '16, June 05-09, 2016, Austin, TX, USA
 ISBN 978-1-4503-4236-0/16/06…$15.00
DOI: http://dx.doi.org/10.1145/2897937.2897976

gorithm, overlapping SET propagations are only processed once and the results are cached into maps, such that the following recursive call never re-processes a SET propagation if it has been processed before and cached in the maps. The memory overhead is negligible since most temporary data are released immediately after they have been consumed.

The contributions of this work are twofold. First, we carry out a detailed analysis on the soft error vulnerabilities in CMOS combinational circuits and determine the key parameters that need to be extracted during the characterization process. Second, a top-down memoization algorithm is proposed to effectively accelerate the computation expensive propagation process. The proposed SER estimation framework is also compatible for the FinFET technology since LUT data structure is highly flexible. Experimental results on various benchmarks demonstrate that the proposed framework achieves up to 560.2X speedup with less than 3% SER difference compared to the baseline algorithm.

2. RELATED WORK

Considerable research efforts have been conducted in the context of estimating SER in combinational circuits. The previous studies can be categorized into two groups:

1) *Soft error characterization studies* [2, 12, 13, 14]: The first step of characterization is the generation of soft errors, which models the physical effects of particle strike as current pulses at the striking nodes. A number of current models, such as Weibull function [14], exponential current pulse [15], and double exponential current model [4, 5, 16, 11], have been adopted by different circuit level SER estimation works. Double exponential current model is one of the most widely accepted models in the circuit level works, and several works [2, 14] have concentrated on determining technology dependent parameters in the double exponential formula. In addition to the soft error generation related works, the authors in [16] proposed a soft error characterization method that captured the pulse widths of SETs, whereas the authors in [12] considered both pulse widths and pulse heights of SETs during soft error characterization. The authors in [13] proposed a sensitive area calculation method in order to model the actual sensitive area. The aforementioned works achieve high accuracy, however, the added parameters will lead to increased workload and processing time in the following propagation phase.

2) *SER estimation studies* [4, 11, 15, 17]: An RTL-based combinatorial SER estimation method was proposed in [17] in order to achieve fast RTL level SER analysis. Compared with RTL level SER analysis, circuit-level SER estimation is more accurate. FASTER [10] used binary decision diagrams with circuit partitioning for SER estimation. SEAT-LA [11] presented an SER estimation framework which characterized the SET parametric waveforms using analytical equations. A hierarchical approach called HSEET was developed in [4], which improved the speed of the SER estimation process in structural combinational logic. Multi-cycle effects were considered in [15], which accounted for those SETs lasting more than one clock cycle. These prior works have established circuit-level SER estimation frameworks that target high accuracy, however, their efforts on speed improvement are limited to circuit partitioning and parallel processing.

3. OVERALL FLOW

In this section, the overall flow of the proposed SER estimation framework is described. When a particle strikes the

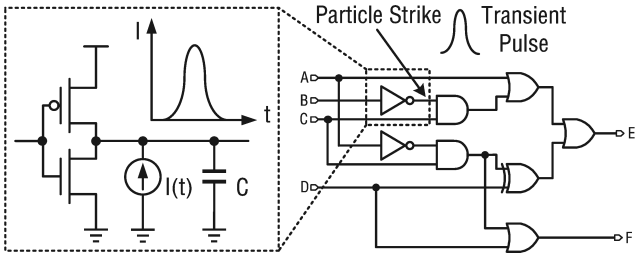

Figure 1: Current source model of a particle strike at a circuit node

circuit, it generates electron-hole pairs inside some transistors on the chip, leading to parasitic transient current pulses at the striking nodes. Each generated transient current pulse is modeled by a current source, as shown in Figure 1.

In the characterization phase, three steps are performed:

1) Based on HSPICE simulations with different SET current pulses, driver states and output capacitances, a *generation LUT* (G-LUT) is built for each standard combinational cell in the library, in order to transform the radiation-induced current pulses into voltage pulses for propagation.

2) The next step is to propagate these voltage pulses towards the outputs. Based on HSPICE simulations, a *propagation LUT* (P-LUT) is established for each standard cell in the library, which records the mapping from voltage pulses at the input of the cell to the propagated voltage pulses at the output for a certain cell state and output capacitance.

3) The propagated SET voltage pulses at outputs need to be captured by the next stage flip-flops to become an error. Therefore, apart from the G-LUTs and P-LUTs, the characteristics of the flip-flops in the library are examined.

In the propagation phase after characterization, a process is conducted, where the final SER is calculated by accumulating the probabilities of SETs that are generated at vulnerable nodes, propagated to outputs and captured by flip-flops. Figure 2 provides the proposed SER estimation flow. The details of characterization and the proposed propagation algorithm are explained in Section 4 and Section 5, respectively.

4. CHARACTERIZATION

In this section, SET current pulse generation, the steps to establish LUTs, and flip-flop characterization are presented.

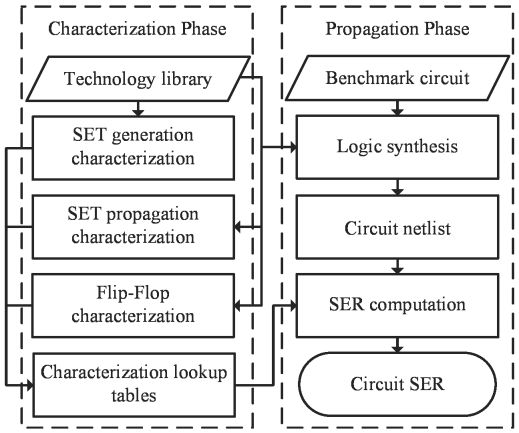

Figure 2: Overall flow of our SER estimation framework

4.1 Parasitic Transient Current Pulse Model

In this paper, the soft error impact is described as a current pulse generated at each particle strike node. Note that there are some debates about the validity of different current models. We choose the double exponential current pulse model because this model has been adopted in [5] for the technology that is used in our experiments, and our SER estimation framework can accommodate other current models as well. The double exponential current is calculated as

$$I(t) = I_{peak} \cdot (e^{-\frac{t}{\tau_\alpha}} - e^{-\frac{t}{\tau_\beta}}) \quad (1)$$

where peak current $I_{peak} = \frac{Q}{\tau_\alpha - \tau_\beta}$, in which Q is the maximum collected charge as the result of the particle strike; τ_α is the charge collection time constant and τ_β is the Ion-track establishment time constant. Figure 3 (a) shows a typical double exponential current pulse waveform with the time parameters extracted from 3D technology computer aided design (TCAD) simulations for a bulk CMOS technology. The rising part of the current pulse corresponds to the establishment of the Ion track at the particle strike node, whereas the falling part represents the process where the excess carrier concentration is moved by drift until the carrier concentration is restored to the background doping. The technology dependent time constants τ_β and τ_α correspond to the rising and falling part, respectively.

According to reference work [5], the charge collection time for bulk CMOS technology is calculated as

$$\tau_\alpha = \frac{k\epsilon_0}{q\mu DN} \quad (2)$$

where $k\epsilon_0$ is the substrate dielectric constant, q is the electron charge, μ is the carrier mobility, D is the doping concentration, and N is a scaling factor, which scales doping concentration D to the generation rate of electron-hole pairs. The Ion-track establishment time constant can be calculated using the method developed in [2].

The probability of deposited charge Q is required during the characterization of G-LUTs. In this paper, we adopt an exponential distribution function of charge probability developed in [4], which constructs the charge probability based on data points obtained for neutron energy and the corresponding differential flux at sea level from the JEDEC Solid State Technology Association Standard JESD89 [18]. In this model, the probability of charge is calculated as

$$P(Q) = a_0 \cdot e^{-a_1 \cdot Q} + a_2 \quad (3)$$

where a_0, a_1 and a_2 are constants. Figure 3 (b) shows an example normalized probability of charge deposition.

4.2 Generation-LUTs and Propagation-LUTs

The characterization results are stored in 2D G-LUTs and P-LUTs. As mentioned in Section 1, G-LUTs are established to record the mapping that transforms the radiation-induced SET current pulses into voltage pulses at the striking node, whereas P-LUTs are constructed to save the mapping from SET voltage pulses at the input of a standard cell to the propagated SET voltage pulse at the output of this cell.

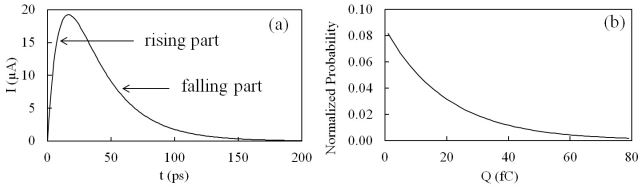

Figure 3: Parasitic transient current pulse model. (a) double exponential current shape (b) probability of charge deposition

In order to make the characterization process general, we conduct a comprehensive investigation of all the factors that could potentially affect the generation and propagation of the SETs, including deposited charge, load capacitance, cell state (i.e., input combination), gate type and gate size. Accordingly, the G-LUTs and P-LUTs are obtained for all possible input vector states, gate sizes, and a wide range of output capacitance as well as deposited charges for the given technology. Figure 4 provides the simulation setup for establishing G-LUTs and P-LUTs. The pulse width measured at 50% supply voltage is captured as the main parameter that describes the SET voltage pulse, and the rise and fall times of the input pulses are chosen to be typical values for P-LUT simulations. Without any loss of generality, the proposed characterization method can be extended to consider more parameters such as SET pulse height and establish more types of LUTs for capturing more radiation-induced physical effects. The flexibility of LUT data structures also enables characterization for FinFET technology. For example, based on reference work [3], the number of generated electron-hole pairs in a FinFET device for different particles energies can be characterized and stored into LUTs.

In general, detailed characterization results lead to high accuracy of SER results but also increased workload for the propagation phase. Therefore, the size of LUTs should be carefully decided in order to avoid long LUT accessing time. For a given technology file that contains M combinational standard cells with various sizes and logic functions, the maximum input pin number of a combinational cell is denoted by C, and the total number of types of k input combinational cells is denoted by $M_k (1 \leq k \leq C)$. For each input combination, each gate size of each combinational standard cell type, we establish (i) one G-LUT that uses deposited charge and output capacitance as index keys, and (ii) one P-LUT that uses input SET pulse width and output capacitance as index keys. Therefore, we construct $M_{total} = 2 \cdot \Sigma_{k=1}^{C} M_k \cdot 2^k$ two-dimensional LUTs in the characterization phase.

4.3 Flip-flop characterization

The characteristics of flip-flops play an important role in determining the SER because the SET pulse has to be latched into one of the next stage flip-flops to become an

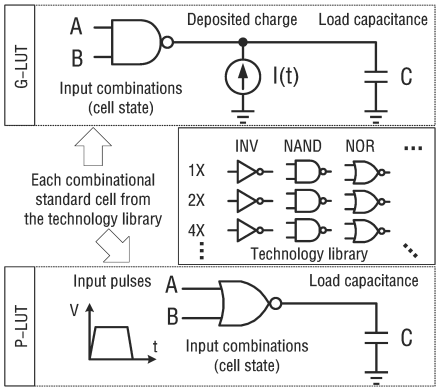

Figure 4: Simulation setup for establishing G-LUTs and P-LUTs considering various factors

error. A flip-flop is insensitive to arrival SET voltage pulses that fall outside the latching window (i.e., setup time + hold time). Therefore, the setup time and hold time of flip-flops are required.

5. PROPAGATION METHODOLOGY

In this section, we present the SER estimation method and the proposed top-down memoization algorithm for accelerating the SER propagation process.

5.1 SER Estimation Method

In the propagation phase, the three important masking factors affecting the propagation of a SET voltage pulse through the combinational circuit are *logical masking*, *electrical masking*, and *latch window masking* [4, 15]. Logical masking occurs if the SET pulse arrives to the input of a gate when at least one of its other inputs has a controlling value. Electrical masking occurs when the SET voltage pulse is attenuated or even completely disappears due to the electrical properties of propagated gates. Latch window masking eliminates the error when the propagated pulse cannot get latched into the flip-flop (as mentioned in Section 4.3).

The total SER is the summation of SER contributed by each node n, which is calculated as

$$SER_{total} = \sum_{n=1}^{N} SER_n \tag{4}$$

where N is the total number of nodes in the circuit. Each node n is susceptible for particle strikes with the deposited charge over the range of Q_{min} and Q_{max}. Accordingly, SER_n is calculated as

$$SER_n = \int_{Q=Q_{min}}^{Q_{max}} P(Q) \cdot SER(n,Q)\, dQ \tag{5}$$

where $SER(n,Q)$ is the SER induced by the SETs that are generated at node n with deposited charge Q, propagated through the circuit under the electrical and logical masking effects, and latched into the flip-flops under the latching window masking effect. $P(Q)$ is the probability of charge calculated in (3). $SER(n,Q)$ can be further formulated as

$$SER(n,Q) = \sum_{d=1}^{D} P_{log}(n,d) \cdot P_{ele}(n,Q,d) \cdot P_{lat}(n,Q,d) \tag{6}$$

where d and D represent the d-th output and the total number of outputs, respectively. The terms $P_{log}(n,d)$, $P_{ele}(n,Q,d)$ and $P_{lat}(n,Q,d)$ in (6) represent the logical masking, electrical masking and latch window masking effects, respectively.

More specifically, the first term $P_{log}(n,d)$ in (6) indicates the total sensitized probability of SETs propagating through the circuit from node n to output d along all paths, which can be computed by accumulating logic probability (P_{side}) for non-controlling values on all side-inputs along the paths,

$$P_{log}(n,d) = \prod_{i \in n \to d} P_{side}(i) \tag{7}$$

where i represents one node along the path from n to d. The logic probability of each signal can be calculated using the correlation coefficient method [19], which considers reconvergent paths. Alternatively, it can be estimated by simulations over a large set of typical vectors (possibly obtained by running a set of benchmark programs), which implicitly considers the reconvergent paths. The second approach is adopted in this paper. The second term $P_{ele}(n,Q,d)$ in (6) denotes the overall possibility of SETs that are generated from node n with Q charge deposited and propagated to output d with recognizable electrical strength along all paths assuming all side-inputs always have non-controlling values. The third term $P_{lat}(n,Q,d)$ in (6) indicates the probability of the SETs, that are induced by charge Q at node n, getting latched into the flip-flop at output d considering no logical masking or electrical masking effects.

In order to calculate (6), LUT access functions $\psi(\cdot)$ are required. We denote the generated SET voltage pulse width at node n with charge Q as $\psi_{gen}^{h_n,r}(n,Q)$, where h_n indicates the gate type of node n and r represents the cell state. With SET pulse width pw at the input of gate j, the propagated pulse width from node i to j is denoted as $\psi_{prop}^{h_i,r}(i,Q,j)$. These two LUT access functions are associated with G-LUTs and P-LUTs characterized in Section 4, respectively.

Therefore, SER_n is computed through generating all possible current pulses at node n, transforming the SET current pulses to voltage pulses by accessing G-LUTs, propagating the SET voltage pulses to the outputs along all possible paths, and getting latched into one flip-flop. The three masking effects are inherently considered during this process. Propagating through one gate requires one P-LUT access. Note that in a large scale circuit, generated SET pulses at each node need to be propagated through all possible paths towards the outputs, making the propagation process computational very expensive.

5.2 Top-Down Memoization Algorithm to Accelerate Propagation

In this subsection, an efficient propagation algorithm using the top-down memoization technique is proposed in order to accelerate the computational expensive SER estimation process.

The first step of the proposed algorithm is to levelize the circuit logic and find the desired node evaluation order such that a node will not be evaluated until all its driving nodes have been evaluated. Several levelization algorithms can be found in reference [20]. The levels of inputs and outputs are set to 0 and L_{max}, respectively, and we denote the level of node n by $l(n), (0 \le l(n) \le L_{max})$.

From (4), the total SER is calculated by accumulating SER_n in each node n. Figure 5 shows the detailed steps for analyzing the SER of one node. First, all the possible SET high-voltage pulses generated at node n and their associated probabilities form a *high-voltage vector* (HVV), whereas a *low-voltage vector* (LVV) stores all the possible low-voltage SET pulses with their associated probabilities. The high-voltage pulses and low-voltage pulses are separated into two different vectors due to the fact that propagating a high-voltage SET and a low-voltage SET through the same gate with the same pulse width can result in different SET pulse widths at the output. This step requires a number of G-LUT accesses and considers all possible charge and input combinations for generating the SET voltage pulses. Next, the HVV and LVV at node n need to be propagated to its output node(s) in the next level. The example in Figure 5 propagates the vectors from a node in level i into two output branch nodes in level $i+1$. The propagation continues until the vectors reach the outputs, and the SER_n is calculated by accumulating the probabilities of the SET pulses getting latched into the next stage flip-flops (or other storage components from the technology library).

In the propagation process, all the possible paths from the

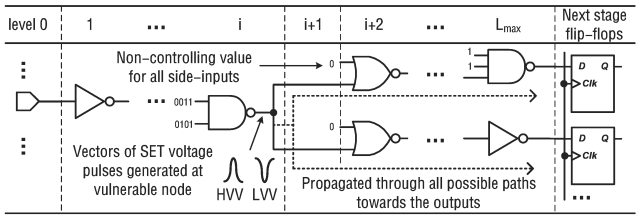

Figure 5: SER analysis for one node

particle striking node to the outputs are automatically covered, and logical masking is considered by the accumulated logic probability for non-controlling values for all side-inputs along the paths. Electrical masking effects are considered in the P-LUTs, i.e., if a SET voltage is attenuated or disappears, the propagated SET pulse width becomes smaller or 0, respectively. Of note, the propagation process requires a large number of P-LUT accesses. The SER estimation framework is provided in Algorithm 1. We have the following important observations:

Observation 1: The nodes that are closer to inputs (i.e., in lower levels) tend to require more P-LUT accesses because the paths from these nodes to outputs involve more gates.

Observation 2: Repeated P-LUT accesses occur frequently at each node because all upper logic nodes that have at least one path to the outputs involving this node need to propagate vectors through this node, and the vectors from upper logic nodes may have many overlapping SET pulses.

Based on the above observations, we propose an effective top-down memoization algorithm in order to accelerate the SER propagation. In the proposed algorithm, a *high-voltage map* (HVM) and a *low-voltage map* (LVM) are dynamically established at each node, which cache the mappings from the high-voltage SET pulses and low-voltage SET pulses at this node with their overall contributed SER at all outputs, respectively. For each SET pulse in the HVV and LVV to be processed, it first checks the maps and removes this SET entry from the vectors if it is a hit. SER is updated accordingly. The remaining "missing" SETs are propagated to form the HVV and LVV at the next node(s), and the previous steps are repeated. In this way, the "missing" SETs will be propagated recursively until they reach one of the output nodes, where their SER can be finally calculated, and the missing entries in the maps will be filled. Then the proposed algorithm goes backwards from this output node to the upper nodes along the path (i.e., return the previous recursive function calls), and fills the missing entries in the maps of those nodes. During this process, the SET entries in the HVV and LVV are released after their corresponding entries in the HVM and LVM are found or filled, in order to reduce the memory overhead. Algorithm 2 summarizes the proposed algorithm.

Algorithm 1: Overall SER estimation framework

```
   /* characterization phase                                    */
1  Read in the technology library;
2  Generate characterization results such as P-LUTs, G-LUTs and
   flip-flop timing;
   /* propagation phase                                         */
3  Initialize the circuit logic, signal probabilities and clock period;
4  Levelization ();
5  for level i ← 0 to L_max do
6  |  foreach node x in this level do
7  |  |  Generate (x);                  /* generate HVV and LVV */
8  |  |  Analyze (x);                      /* calculate SER_x */
9  |  |  UpdateSER ();                     /* update SER_total */
10 |  end
11 end
12 return SER_total;
```

Algorithm 2: Top-down memoization algorithm for accelerating SER propagation

```
1  Function Analyze (node x)
   /* HVV and LVV have been prepared at node x                  */
2  if HVV and LVV empty then return;
   /* check the cached results in HVM and LVM                   */
3  if HVM not empty then
4  |  foreach SET pw_a in HVV do
5  |  |  if pw_a exists in HVM then
6  |  |  |  Update SER and update this entry in HVM;
7  |  |  |  Remove this entry from HVV;
8  |  |  end
9  |  end
10 end
11 if LVM not empty then
12 |  foreach SET pw_a in LVV do
13 |  |  if pw_a exists in LVM then
14 |  |  |  Update SER and update this entry in SER;
15 |  |  |  Remove this entry from LVV;
16 |  |  end
17 |  end
18 end
19 if HVV and LVV empty then return;
   /* process the SETs that are not found                       */
20 foreach output node y of node x do
21 |  Propagate (node y);                  /* propagate vectors */
22 end
   /* update the missing SETs to the maps                       */
23 foreach SET in LVV or HVV do
24 |  Add new entry or update the SER to HVM or LVM;
25 end
26 Delete HVV and LVV at node x;          /* release space */
27 return;
```

6. EXPERIMENTAL RESULTS

In this section, we demonstrate the effectiveness of the proposed framework on a set of ISCAS85 benchmarks. The technology library is 45nm Nangate Open Cell Library [21], which is characterized using Predictive Technology Model (PTM) [22]. In the characterization phase, a number of HSPICE simulations are conducted to generate G-LUTs and P-LUTs, and several Perl scripts are used to assist the characterization process. The SER propagation phase is implemented in C++ on a laptop with an Intel Core i5 processor and 8GB RAM, and the proposed algorithm is compared with two baselines. Baseline 1 is the main baseline algorithm, which is implemented in the same C++ environment except that no memoization technique is applied. In order to make the comparison fair, LUT access functions and SER update functions are shared between baseline 1 and the proposed algorithm. Besides, no parallel processing techniques are allowed, in order to make sure that the speedup is achieved by the proposed algorithm (instead of parallel computing). The proposed algorithm is also compared with baseline 2, which is one of the state-of-the-art SER estimation algorithms [15]. Baseline 1 is more important than baseline 2, since the platform and detailed implementation of baseline 2 are not available whereas baseline 1 is implemented in the same platform as the proposed algorithm.

Table 1 concludes the experiments on a variety of ISCAS85 benchmarks. The information of each benchmark is provided, including the number of nodes (#nodes), the number of primary inputs (#PI), the number of primary outputs (#PO), and the number of levels (L_{max}). The middle columns in Table 1 conclude the runtime comparison among the proposed algorithm and the two baselines. The last column provides the final SER of each circuit in terms of failure-in-time (FIT) generated by our SER estimation

Table 1: Experimental results of various ISCAS85 benchmark circuits

Circuit Information					Runtime Comparison					SER
Circuit	#nodes	#PI	#PO	L_{max}	Proposed(s)	Baseline 1(s)	Speedup	Baseline 2(s)	Speedup	Proposed(FIT)
c432	233	36	7	32	0.0245	0.5862	23.9	1.77	72.1	4.14E-04
c499	638	41	32	34	0.2407	28.6909	119.2	9.38	39.0	8.13E-03
c880a	433	60	26	28	0.1601	1.4876	9.3	1.79	11.2	2.83E-03
c1355	629	41	33	35	0.2645	26.6811	100.9	9.63	36.4	6.88E-03
c1908	425	33	25	45	0.3408	80.0182	234.8	4.29	12.6	4.47E-03
c2670	872	157	64	27	0.1965	1.6831	8.6	2.47	12.6	5.53E-03
c3540	901	50	22	53	1.3183	738.5540	560.2	9.29	7.0	2.48E-03
c5315	1833	178	123	41	1.7709	105.5140	59.6	8.59	4.9	1.09E-02
c6288	2788	32	32	129	26.5855	6271.8153	235.9	120.50	4.5	1.09E-02
c7552	2171	207	108	39	4.7501	182.4250	38.4	13.2	2.8	7.73E-03
average					1.0074	129.5156	128.3	6.71	22.1	

framework.

Results in Table 1 demonstrate that the proposed algorithm consistently outperforms both baselines in all the ISCAS85 benchmark circuits. The proposed algorithm achieves up to 560.2X and 72.1X speedup compared to baseline 1 and baseline 2, respectively. Of note, the implementation of the proposed algorithm does not take the advantage of parallel computing (in order to make comparison fair between the proposed algorithm and baseline 1) and baseline 2 is implemented with parallel computing, resulting in the trend that the proposed algorithm has lower speedups over baseline 2 for larger circuit. The proposed algorithm can be further accelerated using circuit partitioning and parallel computing techniques that have already been exploited in prior works. Difference in SER results between the proposed algorithm and baseline 1 have been observed ($< 3\%$), which is caused by the round up/down of the SET pulse widths (key for the maps) during map checking. Considering the significant speedup achieved by the proposed algorithm, this amount of SER difference can be accepted. Besides, the peak memory usage reported by the largest benchmark c6288 is 50.2MB, indicating that the memory overhead from the additional maps is negligible.

7. CONCLUSION

In this paper, a novel top-down memoization algorithm was proposed to accelerate the SER estimation process for combination circuits. The proposed algorithm cached solutions to avoid overlapping SET propagations, enabling fast SER estimation. The run-time of the proposed algorithm was of the order of few seconds for various ISCAS85 benchmark circuits. Compared with the baseline algorithm, the proposed algorithm achieved up to 560.2X speedup with less than 3% difference of SER results.

8. REFERENCES

[1] S. Krishnaswamy, I. L. Markov, and J. P. Hayes, *Design, analysis and test of logic circuits under uncertainty.* Springer Science & Business Media, 2012, vol. 115.

[2] F. Wrobel, L. Dilillo, A. Touboul, V. Pouget, and F. Saigné, "Determining realistic parameters for the double exponential law that models transient current pulses," *Nuclear Science, IEEE Transactions on*, vol. 61, no. 4, pp. 1813–1818, 2014.

[3] S. Kiamehr, T. Osiecki, M. Tahoori, and S. Nassif, "Radiation-induced soft error analysis of srams in soi finfet technology: A device to circuit approach," in *Proceedings of the 51st Annual Design Automation Conference.* ACM, 2014, pp. 1–6.

[4] K. Ramakrishnan, R. Rajaraman, N. Vijaykrishnan, Y. Xie, M. J. Irwin, and K. Unlu, "Hierarchical soft error estimation tool (hseet)," in *Quality Electronic Design, 2008. ISQED 2008. 9th International Symposium on.* IEEE, 2008, pp. 680–683.

[5] J. Velamala, R. LiVolsi, M. Torres, and Y. Cao, "Design sensitivity of single event transients in scaled logic circuits," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE.* IEEE, 2011, pp. 694–699.

[6] M. Nicolaidis, *Soft errors in modern electronic systems.* Springer Science & Business Media, 2010, vol. 41.

[7] N. Mahatme, N. Gaspard, T. Assis, I. Chatterjee, T. Loveless, B. Bhuva, W. H. Robinson, L. W. Massengill, S.-J. Wen, and R. Wong, "Kernel-based circuit partition approach to mitigate combinational logic soft errors," *Nuclear Science, IEEE Transactions on*, vol. 61, no. 6, pp. 3274–3281, 2014.

[8] N. Mahatme, N. Gaspard, T. Assis, S. Jagannathan, I. Chatterjee, T. Loveless, B. Bhuva, L. W. Massengill, S. Wen, and R. Wong, "Impact of technology scaling on the combinational logic soft error rate," in *Reliability Physics Symposium, 2014 IEEE International.* IEEE, 2014, pp. 5F–2.

[9] V. Ferlet-Cavrois, L. W. Massengill, and P. Gouker, "Single event transients in digital cmos-a review," *Nuclear Science, IEEE Transactions on*, vol. 60, no. 3, pp. 1767–1790, 2013.

[10] B. Zhang, W.-S. Wang, and M. Orshansky, "Faser: Fast analysis of soft error susceptibility for cell-based designs," in *Proceedings of the 7th International Symposium on Quality Electronic Design.* IEEE Computer Society, 2006, pp. 755–760.

[11] R. Rajaraman, J. Kim, N. Vijaykrishnan, Y. Xie, and M. J. Irwin, "Seat-la: a soft error analysis tool for combinational logic," in *VLSI Design, 2006. Held jointly with 5th International Conference on Embedded Systems and Design., 19th International Conference on.* IEEE, 2006, pp. 4–pp.

[12] X. Gili, S. Barceló, S. Bota, J. Segura *et al.*, "Analytical modeling of single event transients propagation in combinational logic gates," *Nuclear Science, IEEE Transactions on*, vol. 59, no. 4, pp. 971–979, 2012.

[13] S. Chen, Y. Du, B. Liu, and J. Qin, "Calculating the soft error vulnerabilities of combinational circuits by re-considering the sensitive area," *IEEE Trans Nucl Sci*, vol. 61, pp. 646–653, 2014.

[14] N. Kehl and W. Rosenstiel, "An efficient ser estimation method for combinational circuits," *Reliability, IEEE Transactions on*, vol. 60, no. 4, pp. 742–747, 2011.

[15] R. H.-M. Huang and C. H.-P. Wen, "Advanced soft-error-rate (ser) estimation with striking-time and multi-cycle effects," in *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE.* IEEE, 2014, pp. 1–6.

[16] D. B. Limbrick and W. H. Robinson, "Characterizing single event transient pulse widths in an open-source cell library using spice," in *IEEE Workshop on Silicon Errors in Logic-System Effects (SELSE)*, 2012.

[17] A. Evans, D. Alexandrescu, E. Costenaro, and L. Chen, "Hierarchical rtl-based combinatorial ser estimation," in *On-Line Testing Symposium (IOLTS), 2013 IEEE 19th International.* IEEE, 2013, pp. 139–144.

[18] J. Standard, "Jesd89a," *Measurement and Reporting of Alpha Particles and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices*, 2006.

[19] B. Ricc, "Estimate of signal probability in combinational logic networks," 1989.

[20] N. K. Jha and S. Gupta, *Testing of digital systems.* Cambridge University Press, 2003.

[21] Nangate 45nm Open Library, Nangate Inc., 2009. [Online]. Available: http://www.nangate.com/

[22] Predictive Technology Model. [Online]. Available: http://ptm.asu.edu/