

# Cascaded Propagation and Reduction Techniques for Fault Binary Decision Diagram in Single-event Transient Analysis

Jong Kang Park, Myoungha Kim, and Jong Tae Kim \*

**Abstract**—Single Event Transient has a critical impact on highly integrated logic circuits which are currently common in various commercial and consumer electronic devices. Reliability against the soft and intermittent faults will become a key metric to evaluate such complex system on chip designs. Our previous work analyzing soft errors was focused on parallelizing and optimizing error propagation procedures for individual transient faults on logic and sequential cells. In this paper, we present a new propagation technique where a fault binary decision diagram (BDD) continues to merge every new fault generated from the subsequent logic gate traversal. BDD-based transient fault analysis has been known to provide the most accurate results that consider both electrical and logical properties for the given design. However, it suffers from a limitation in storing and handling BDDs that can be increased in size and operations by the exponential order. On the other hand, the proposed method requires only a visit to each logic gate traversal and unnecessary BDDs can be removed or reduced. This results in an approximately 20-200 fold speed increase while the existing parallelized procedure is only 3-4 times faster than the baseline algorithm.

**Index Terms**—Single Event Transient, soft error, Binary Decision Diagram, logic circuit, reliability

## I. INTRODUCTION

Feature shrinking of transistors and ever-increasing low power requirements result in device reliability issues including soft errors. It is well known that the root cause of these temporal faults can be high energy particles and signal or power integrity problems. Especially regarding radiation, the soft error rates (SERs) of logic circuits continue to increase in the current and the future technology nodes for terrestrial applications [1]. Their numbers are now comparable to those of memory SERs. Although new technologies such as fin field-effect transistor (FinFET) have better soft error immunity than bulk complementary metal-oxide-semiconductor (CMOS) processes [2], highly integrated devices or systems must cope with the total soft error rate representing one of the crucial reliability metrics for the target system. As more complex logic gates and memory elements are integrated, device or system-level errors should be considered the same as the individual errors within the component. Failure-in-time (FIT) is used to evaluate long-term failure rate which is defined by the number of errors observed in  $10^9$  hours. As an extreme case,  $10^8$  FIT of the total errors observed in a data center [3] should be continuously monitored and suppressed as a reliability metric.

Single event transients (SETs) from the collision of high energy particles create single or multiple temporal faults in logic circuits. Transient waveforms propagated along the circuit paths might be stored in the sequential element in the circuit and appear as soft errors. There have been numerous works [4-12] on the estimation of

SERs caused by SETs in a static procedure. Symbolic framework using a binary decision diagram (BDD) provides a natural view that simultaneously considers electrical, logical, and timing propagation properties [11, 12, 18]. Missing the correlation between these properties to reduce the complexity of manipulating BDDs degrades the evaluation results for soft errors. As reported in [18], such errors can be increased by up to 100% even in a small logic circuit in comparison to errors resulting from fully correlated BDDs.

The main contribution of this paper is to develop a BDD-based SER analysis technique that speeds up the run-time and reduces the number and the sizes of the target BDDs in comparison to a conventional algorithm. This enables a common digital logic design analysis where a large-scale integration of logic cells is distributed through the design hierarchy. To achieve this without much degrading the overall accuracy of the estimation, first, we employ the cascaded fault propagation method based on the topological order of the circuit graph. This effectively eliminates the iterative construction of the faulty BDD on the circuit path. Additionally, the traversal of the faulty gates, which have small portions of SERs contributing to the sensitized ports, can be stopped and eventually, we can safely skip successive visits to the posterior logic gates along the reverse propagation path. Establishing a virtual primary input (PI) with less correlation to the other circuit nodes on the sensitized paths reduces the corresponding BDDs. Consequently, these modifications result in 20-200 times faster analysis while the estimation errors are constrained to 1% in the benchmark circuits. In this paper, we employed a single fault model for technology mapped gate-level designs to validate our work, even if it can be further extensively applied to concurrent multi-fault models.

This paper is organized as follows. Section 2 provides a summary of existing work that has evaluated the SERs of gate-level designs. In Section 3, the fundamentals of fault BDD propagation and the relevant SER calculation of the gate-level circuits are briefly introduced. Section 4 presents the key procedures of the proposed analysis technique. Section 5 shows the comparative results to existing works. Finally, we conclude this work in Section 6.

## II. RELATED WORKS

Dynamic analysis on a circuit model [9, 19] is an intuitive and accurate way to evaluate the soft error tolerance of a given design. The Monte-Carlo simulation method is commonly used to implement a random sampling of SETs. However, it requires a large number of simulation steps, and it must consider the convergence of the result and the run-time of the simulation.

Path-based analysis techniques [4-8, 10] employ static probabilities for circuit nodes, which are events that are independent of or less correlated with the other nodes. These are efficient and fast methods to estimate the SERs of typical combinational circuits. However, an estimation error might exist when the correlation between propagated faults is more severe due to re-convergent fan-outs and simultaneous multiple transient faults. Without considering a re-convergent fan-out, the corresponding logical probability for signal propagation might have a significant error [20]. Weighted averaging of probabilities for each stem can improve this error. However, as previously mentioned, separating the logical and electrical properties of the propagated SET results in further estimation errors. Inaccurate identification for the critical region misguides the gate-level reduction procedures including cell-sizing and logical redundant techniques [11, 13], and potentially leading to over-sized and over-timed designs.

Alternatively, symbolic frameworks using BDD [11, 12, 18] provide every possible decision path for the input conditions in a binary tree. Attaching fault waveforms to the terminal nodes in BDD can concurrently take into consideration the major masking effects of the given logic design. However, the exponential growth of the BDD size on a large number of PIs and the outputs of flip-flops (F/Fs) increases run-time and memory overhead during analysis, makes it infeasible in practical block designs [14].

To overcome the run-time of a BDD-based analysis, our previous work employed multiple threads to run the individual BDD propagation procedure [16]. The result is execution time that is 3-4 times faster than the baseline algorithm. It is difficult to further parallelize the procedures as the shared memory system of the simulation host machine limits the bandwidth of the memory access. Moreover, the memory requirement is

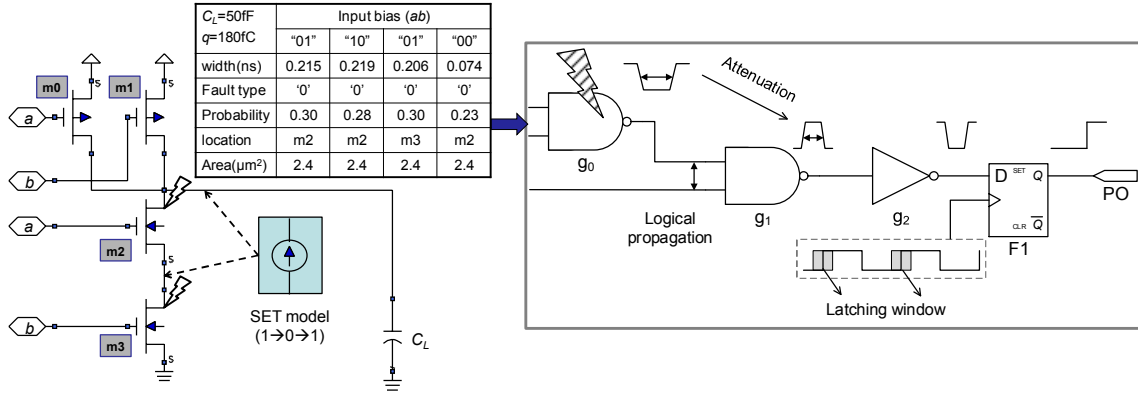


Fig. 1. Generation and propagation of SET instance.

still un-changed due to the size of the BDDs.

### III. SER ESTIMATION BASED ON BDD PROPAGATION

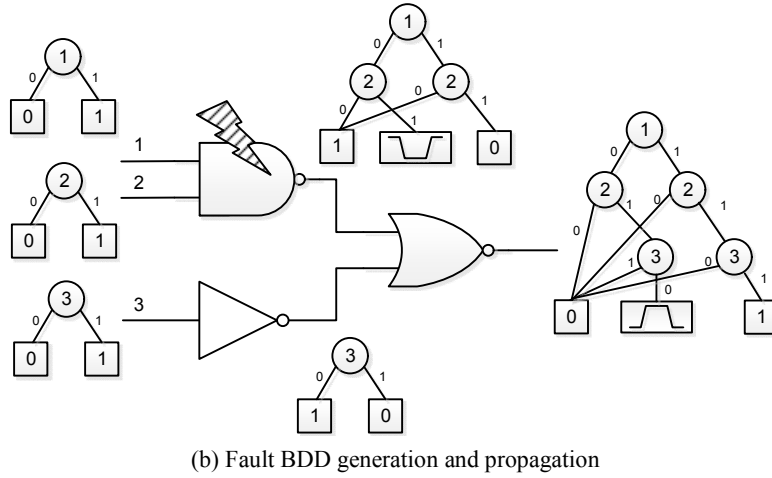
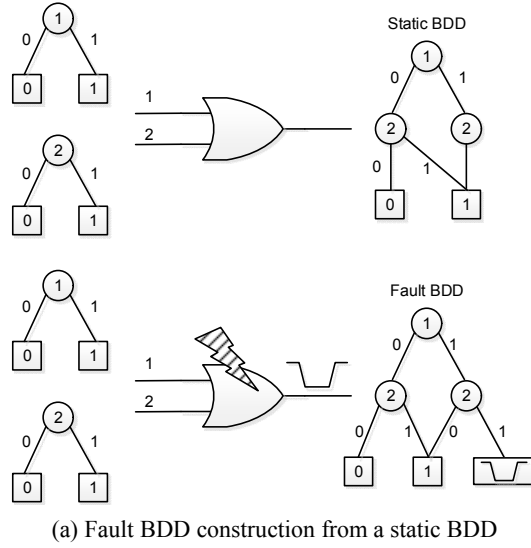
SER estimation from a SET indicates three masking effects of the logic circuit, electrical, logical and timing factors. Fig. 1 shows an example of SET generation and propagation from the two-input NAND gate. A faulty site, which typically resides in one of the drain nodes in a logic gate, generates fault waveforms based on the bias conditions, the load capacitance  $C_L$ , and the collected charge  $q$ . It can be characterized using SPICE-level simulation. As shown in Fig. 1, based on the input bias, m2 and m3 NMOS transistors generate four types of  $1 \rightarrow 0 \rightarrow 1$  SET instances where the widths, fault types, generation probabilities, and the site areas are defined. For the given load capacitance and selected  $q$ , a SET instance is iteratively selected and generates fault waveforms at the output of the target logic gate. When the fault is passing through the intermediate logic gate, we must consider the logical masking and the electrical attenuation of the output waveform. At the input of F/F or the primary output (PO), the SER can be calculated by checking whether the fault will be stored in the memory element. Let  $PI$ ,  $PO$  and  $FF$  be the sets of primary inputs, primary outputs and the flip-flops in the given design. We define  $ISER$  originating from the faulty set of site  $i$  and the total block SER as follows [8, 11] :

$$ISER_i(j) = F_n \cdot \alpha \cdot \sum_{SET_i} \sum_q (f_Q(q) \cdot A_i(SET_i) \cdot GP(SET_i) \cdot LP_{ij} \cdot EP_{ij} \cdot LW_{ij}) \Delta q \quad (1)$$

$$SER = \sum_i ISER_i = \sum_i \sum_j^{[PO] \cup [FF]} ISER_i(j) \quad (2)$$

where  $ISER_i(j)$  denotes the SER observed at port  $j$ , which is either a PO or an F/F, and its SETs are confined to those generated at  $i$ . This metric is used to evaluate the error impact that the individual logic gate has on the overall SER.  $F_n \cdot \alpha$  means the effective neutron flux at the given device.  $f_Q(q)$ ,  $A_i(SET_i)$  and  $GP(SET_i)$  denote a probability density function for the collected charge  $q$ , a region of the faulty site of the logic gate  $g$  and the logical generation probability for  $SET_i$  from  $i$ , respectively.  $LP_{ij}$ ,  $EP_{ij}$  and  $LW_{ij}$  are the probabilities of logical propagation, electrical attenuation and latching window from  $i$  to port  $j$ , respectively.  $SET_i$  is an event for a single event transient at logic gate  $i$ . The amount of charge collection due to  $SET_i$  is defined by  $q \in Q$ . Since  $f_Q(q)$  is the probability density function that decays exponentially, a discrete value  $q$  can be effectively constrained in the given technology as shown in [9].

In a static BDD, non-terminal nodes represent PI values and the terminal nodes contain output values. Each edge has a label, 0 or 1, which is the value of its parent non-terminal node. Thus, every non-terminal node value can be determined by the combination of PI values in this structure. As seen in Fig. 2(a), three static BDDs represent the pure logical values according to their sensitized PI values. If we assume that node 1 and node 2 are PIs of the given circuit, each input BDD contains two terminal nodes as output values, 0 or 1, depending on the value of its PI. Then, the output of the OR gate will be constructed by merging two input BDDs with the Boolean OR operation. This procedure uses Shannon's



**Fig. 2.** Examples of static and fault BDDs.

expansion [15]. In this paper, F/F outputs can also be the non-terminal vertices of BDD.

In Eq. (1), probabilities,  $GP$ ,  $LP$ ,  $EP$  and  $LW$  with simple products are of independent events representing generation, logical-electrical propagation and latching for SETs, respectively. A BDD structure can handle those events with full correlation. To employ it, we should rewrite Eq. (1) by conditional probabilities. If  $P(fBDD_j|SET_i)$  is defined as the latching probability of the fault waveforms in the propagated fault BDD at port  $j$  dependent on  $SET_i$ , we can simply rewrite Eq. (1) as follows:

$$ISER_{i,j} = F_n \cdot \alpha \cdot \sum_{SET_i} \sum_q (f_q(q) \cdot A_i(SET_i) \cdot P(fBDD_j | SET_i)) \Delta q \quad (3)$$

As shown in Fig. 2(a), a fault BDD for a certain  $SET_i$  can be constructed by its static BDD. Accordingly, there must be normal terminal nodes that have logic values of '0' or '1', and fault terminal nodes that include the possible SET waveforms that are attenuated by the generation and propagation procedures. Each edge from a vertex will contain a logical probability for its parent node which is one of the input indices. Since cell-based SET characterization can be conducted as shown in Fig. 1, a  $1 \rightarrow 0 \rightarrow 1$  transient event in Fig. 2(a) is added to the terminal node under input bias = "11". During the propagation, a fault BDD will be successively generated by merging one or more BDDs with the specified logic operation. In Fig. 2(b), a fault BDD at the NAND gate will be passed by considering another static BDD from the inverter. After logically and electrically synthesizing

two fault BDDs, the resultant BDD consists of three PIs and the fault waveform estimated by the logical and electrical characteristics of the NOR gate. If a fault BDD reaches any POs or F/Fs, the latching probability for a fault can be calculated by traversing the vertices and edges of the current BDD. In this way, every fault in the BDD is defined by all possible logic values for the sensitized PIs or F/F values. It is not regarded as an independent event in this structure.

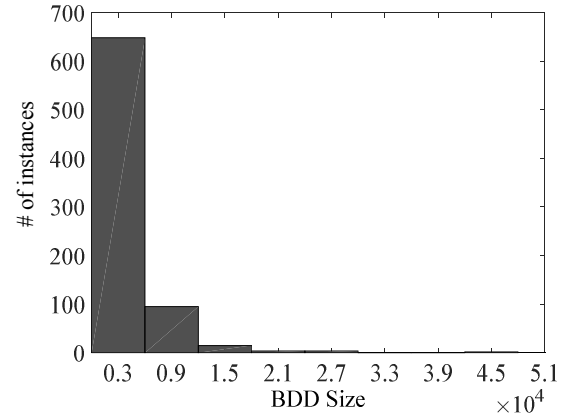
If we assume that  $|Q|$  is constant, the run-time of the calculation for Eq. (2) based on Eq. (3) obeys the following time complexity that can be derived by the time complexity required to manipulate the BDD operations [15].

$$O(|SET_i| \cdot |G| \cdot (|V_{BDD}| + |E_{BDD}|)^2) \quad (4)$$

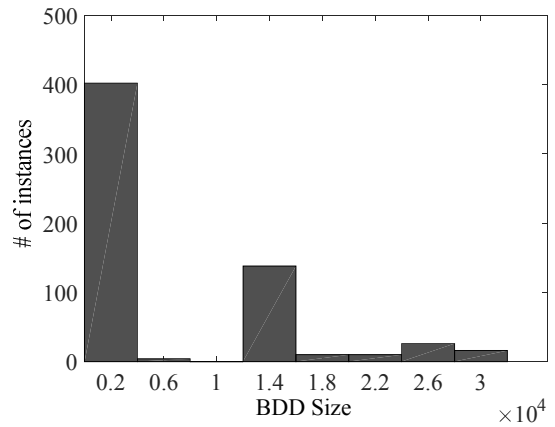
where  $G$  is a set of logic gates and F/Fs in the given design while  $V_{BDD}$  and  $E_{BDD}$  are the sets of vertices and edges in a BDD. Merging two different BDDs iteratively compares two vertices from BDDs. The number of  $|SET_i|$  can be limited and commonly proportional to  $|G|$  in a single SET propagation. For a given  $SET_i$ ,  $|G|$  is iteratively required for logic gate traversal during a fault propagation. Unnecessary operations will be removed with the cascaded propagation technique. However, the size of a BDD,  $|V_{BDD}| + |E_{BDD}|$ , is inherently dependent on  $2^{|F|+|PI|}$  and can be increased significantly, even when the relevant reduction algorithms [15] are applied. It has been reported that the best variable ordering to minimize the BDD is an NP-hard problem [17]. Fig. 3 shows the size distribution of the static BDDs observed in two designs. Although the target design has less than 1,000 gate count, the size will be varied up to  $4.5 \times 10^4$ . This is time- and memory-critical to manipulate many BDDs iteratively. The next section presents how the proposed techniques ease the time and memory requirement for the fault BDD propagation analysis.

#### IV. PROPOSED BDD PROPAGATION TECHNIQUES

To tackle the run-time of BDD propagation, three key techniques will be applied to the original algorithm [11, 12, 16]. We explain the details of the procedures and



(a) s1423



(b) c499

**Fig. 3.** Distribution of sizes of static BDDs during the SET propagation.

their advantages in this section.

#### 1. Cascaded Fault Insertion

To reduce the iterative propagation processes for  $SET_i$ , we employ successive fault waveform insertion for fault BDD propagation. In Fig. 4, we can see that a fault BDD is constructed at  $g_1$  with all possible SETs. The fault waveforms in terminal nodes should be tagged by the unique fault ID indicating which re-convergent faults will be manipulated by the logic and electrical operations in later gate traversal. The faults with different IDs are regarded as independent events. By this definition, multiple faults due to a single particle injection can have the same IDs even when they originate from different logic gates. The fault BDD can also be established in  $g_2$ , because the faults at  $g_1$  and  $g_2$  are all transmitted to the same propagation path in topological order. Therefore,

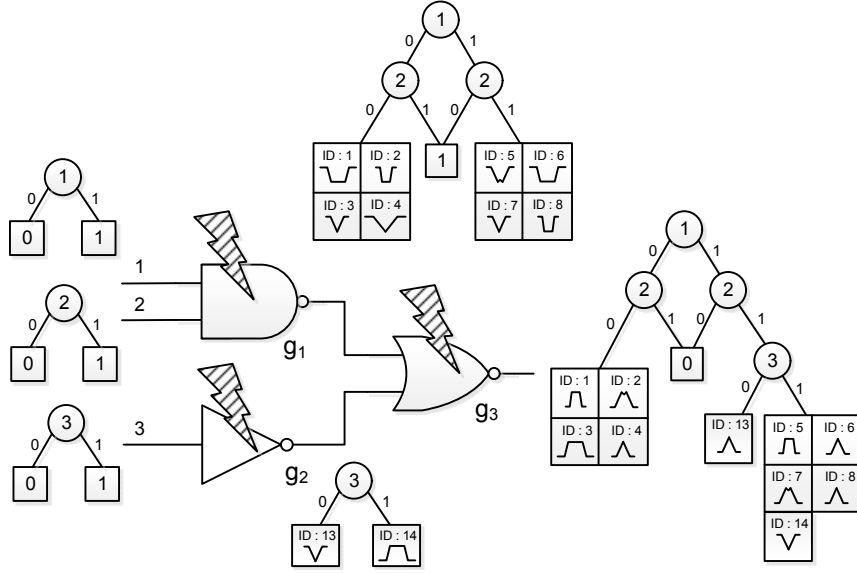


Fig. 4. Successive fault attachment to on-line BDD.

the propagated BDD at  $g_3$  will incorporate such faults by merging two fault BDDs for  $g_1$  and  $g_2$ . At this stage, other faults originating from  $g_3$  will be added to the fault BDD. This clearly eliminates the need to revisit the same propagation path for  $SET_i$  evaluation in different logic gates.

In the ideal case, only a visit to the logic gate to add  $SET_i$  will complete the entire analysis if we add all possible faults along the propagation paths. However, the faults in the terminal BDD cannot be eliminated by the reduction techniques for BDD. This will increase the size of the fault BDD exponentially. In Section 5, we practically limit the number of successive fault additions during BDD propagation.

## 2. Virtual PI Insertion

The logical probability for the fault in BDD is obtained by traversing either the 0 or 1 edges of each input index. Edges contain the corresponding probability that have their binary values. Let  $PI_j$  and  $FF_j$  be sets of PIs and F/Fs sensitized to port  $j$ . If we define  $IN_{jk}$  by an input event with index  $k$  in  $fBDD_j$ ,  $P(fBDD_j|SET_i)$  in Eq. (3) can be derived by its sensitized input event  $IN_{jk} \in \{PI_j, FF_j\}$  containing the proper edge value to reach the fault terminal as follows :

$$P(fBDD_j | SET_i) = \sum_{f_j} P \left( \bigcap_{PI_j, FF_j} IN_{jk} | f_j \right) LW_{f_j} \quad (5)$$

where  $f_j$  subordinate to the fault terminal in  $fBDD_j$ , denotes one of the possible faults originating from  $SET_i$  and  $LW_{f_j}$  is the probability for storing  $f_j$  in the setup and hold time periods of the F/F [9, 10, 12]. For every  $f_j$  at the terminal, the logical probability can be obtained by traversing the sensitized path to  $f_j$  from the root vertex of  $fBDD_j$ . If  $\cap IN_{jk} = \emptyset$ , meaning the input events are independent, the probability is simply defined by the product of individual probability for each  $IN_{jk}$ .

As shown in Fig. 5, suppose that two fault BDDs,  $fBDD_0$  and  $fBDD_1$  merge at the NOR gate to yield  $fBDD_2$  and they are not on the re-convergent fan-out stem.  $fBDD_2$  can also be propagated to the succeeding gates and further synthesized by other fault or static BDDs. However,  $fBDD_2$  can be reconstructed by two virtual PIs that contain two vertices, with new indices originating from the two inputs of the previous NOR gate. If the PI events are not correlated and the intersection of sensitized PIs for  $fBDD_2$  and  $fBDD_3$  is null, we can redefine  $fBDD_2$  upon acquiring  $fBDD_4$  in Fig. 5. The terminal vertices in the modified  $fBDD_2$  should contain the updated logical probability derived from the original  $fBDD_2$ . Consequently, the logical probabilities for the fault in the original and reduced  $fBDD_4$  are identical. Note that  $fBDD_2$  is not on the re-convergent fan-out of the circuit since virtual PIs alone cannot exactly define the BDD in later converging logic gates without using the original PIs of  $fBDD_2$ , which are already eliminated by the virtual PIs. This also agrees with the previous

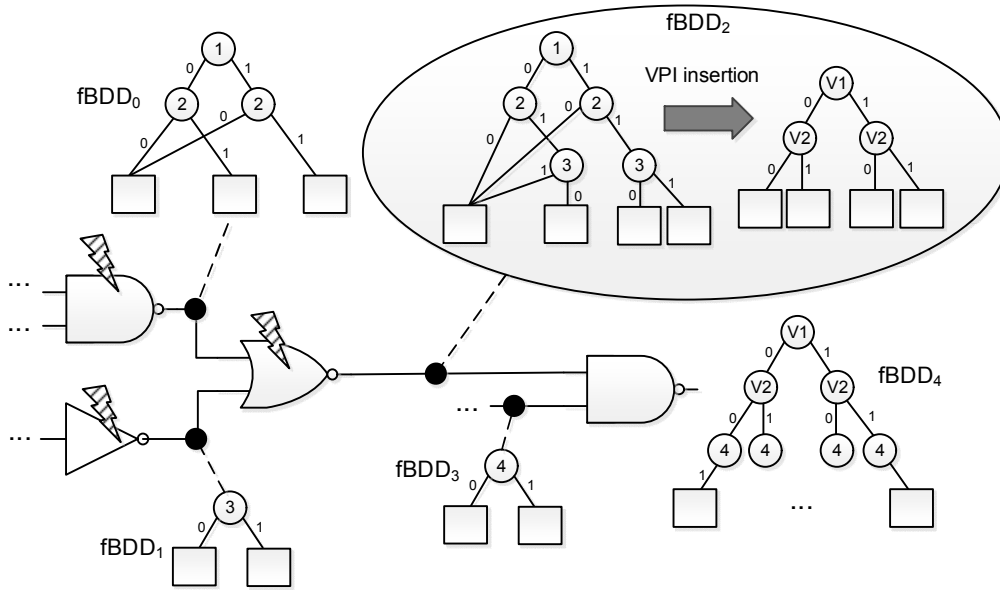


Fig. 5. Virtual PI insertion for limiting the size of BDD.

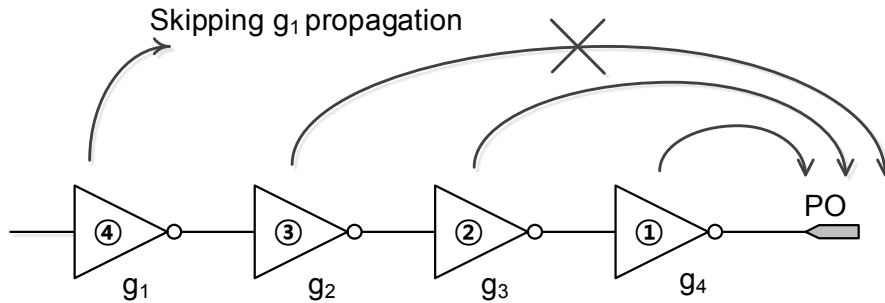


Fig. 6. Skipping the logic gate traversal in reverse topological order.

result where a combination of two input values in the logic gate of fan-in number two is sufficient for SER calculation when located on the non-re-convergent fan-out [7].

Without loss of generality,  $fBDD_j$  not on the re-convergent path where non-terminal vertices belong to  $\{PI_j, FF_j\}$  can be replaced by  $fBDD_j$  with a virtual input  $VI_j$  if two  $fBDD_j$  and  $fBDD_m$  are synthesized by the Boolean operation where  $\{PI_j, FF_j\} \cap \{PI_m, FF_m\} = \emptyset$ .

### 3. Skipping the Logic Gates with low ISERs

A SET is attenuated by the electrical characteristics and logical masking effects of the logic gates. As shown in [7, 13], the length of  $SET_i$  propagation largely affects  $ISER_i$  in Eq. (3). The longer the SET propagates, the lesser the SET width and logical probability expected at

POs and F/Fs. The main idea of the approximation in this sub-section involves skipping the logic gate traversal with low SER expected in a reverse topological order. In an inverter chain as illustrated in Fig. 6, we first evaluate the SETs generated at the last stage inverter  $g_4$  which is nearest to PO. The second and third visit will be conducted at  $g_3$  and  $g_2$  respectively. However, if  $ISER_2$  is a relatively small value with few contributions to the accumulated SER at PO, we can skip the evaluation of  $g_1$ . This will accelerate the logic gate traversal if we compare every  $ISER$  to POs or F/Fs and mark any skipped logic gates in a reverse topological order.

### 4. Procedures

Applying all techniques explained in Section 4.1-4.3, a new BDD-based SER estimation procedure is presented

---

Procedure *cascaded\_propagation*

---

$G'$  = topological sorting for gate-level design  $G$  from  $g$   
 $G''$  = reverse topological order for  $G$   
construct static BDDs for PIs and F/F outputs  
mark re-convergent fan-outs from PIs and F/F outputs using DFS  
**for** each logic gate or flip-flop  $g \in G''$   
  create a fault-BDD for  $g$  w/all SET instances  
   $m = 1$   
  **for** each logic gate or flip-flop  $p \in G'$   
    retrieving fault-BDD and static BDDs for inputs of  $p$   
    **if**  $p$  is not on the re-convergence path  
    **or** input BDDs exceed the critical size,  
      adding virtual PIs to BDDs  
    **if** any fault-BDDs exist at the inputs of  $p$   
    **and** all fan-out gates has no skip flag,  
      propagate a fault-BDD for  $p$   
    **if**  $m < MAX\_MERGE$ ,  
      merge all SET instances to the fault-BDD at  $p$   
       $m = m + 1$   
    **else if** there are no static BDDs for  $p$ , construct static BDD for  $p$   
    **if** all preceding gates for  $p$  are visited,  
      remove all preceding fault-BDDs  
    mark  $p$  as visited  
    **if**  $p$  is directly connected to PO or an input of flip-flop,  
      calculate SERs based on fault-BDDs  
      **if** SERs are relatively low to the accumulated values,  
      set the skip flag for  $p$

---

**Fig. 7.** Proposed algorithm based on all techniques in Section 4.1-4.3.

as shown in Fig. 7. Starting with topological sorting for the circuit graph, SETs confined to the logic gate will construct a fault BDD and then will be propagated through their sensitized circuit path. However, since we have no static BDDs sensitized to the fault gate initially, the inner loop must traverse the logic gate in topological order from all PIs and F/F outputs of the given design. Then it constructs a fault BDD after reaching the fault logic gates defined by  $SET_i$ . The outmost loop selects the faulty logic gates in reverse topological order so that the skipping check in Section 4.3 will be conducted to determine whether the succeeding gate has relatively low SERs compared to the POs and F/Fs. Here, we define a skipping ratio which is designated to be compared with  $ISER$  over SER ( $=ISER / SER$ ) at the port. If this ratio is less than the pre-defined threshold, the corresponding source gate is marked so that the preceding logic gates have chances to avoid the fault generation and propagation procedures. Since the total SER is the aggregated value for all  $ISERs$  exist in the circuit, small

$ISERs$  can be skipped to be added. In the on-line algorithm, however, the total SER cannot be estimated during the gate traversal unless we finalize the analysis. Instead, by the reverse topological order from POs and inputs of F/Fs, we can lead that early logic gate traversal having bigger  $ISERs$  determines “on-line SER”.

To construct the propagated fault BDD at the output of the logic gate  $p$ , the fault BDDs or static BDDs that exist in the other inputs are merged by the logical operation. Before synthesizing  $fBDD_p$ , a re-convergence check to determine whether  $p$  is on the re-convergent stem should be performed. If the corresponding path is not re-convergent to the succeeding logic gate,  $fBDD_p$  will be reduced by the virtual PIs presented in Section 4.2. As a practical implementation, a static BDD for each  $p$  is not necessarily constructed by iterative SET propagation [12]. Only one construction of each static BDD at the first visit to  $p$  can be reused by all propagation operations. The stored static BDD, which is either original or reduced by virtual PI, is later retrieved on another  $SET_i$  propagation.



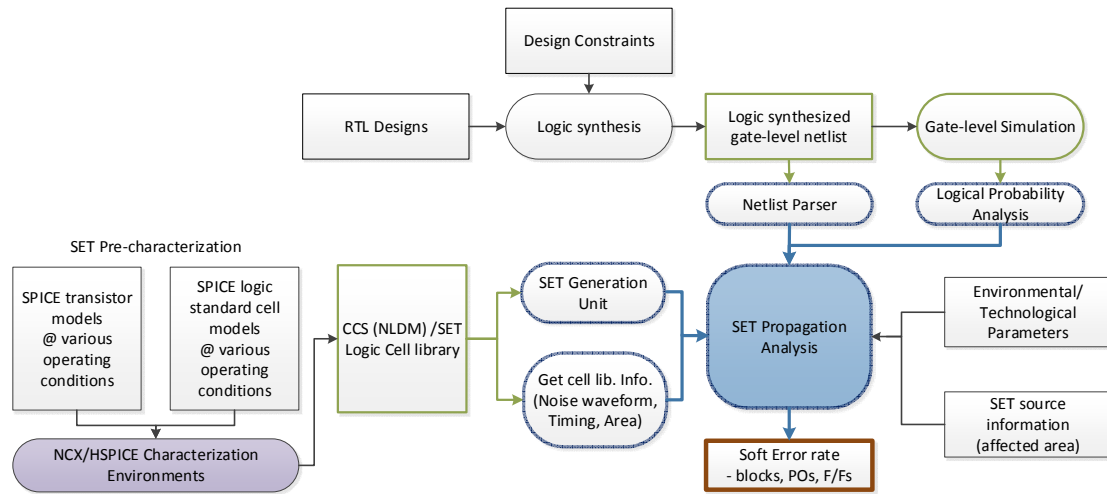


Fig. 8. BDD-based SER estimation framework for gate-level design.

Based on the cascaded propagation rule in Section 4.1, the faults originating from  $p$  can be added to  $fBDD_p$  if the maximum number of cascaded fault gates is not exceeded. If the preceding gates of  $p$  are all visited for  $SET_i$ , the prior fault BDDs should be de-allocated in order to reduce the unnecessary memory area. After reaching any F/F or PO, the algorithm calculates and accumulates  $ISER$  to the target port using Eq. (3). The severity of  $ISER$  at the given port should be evaluated at this stage for the skipping rule. In this paper, we do not cover concurrent multiple faults [4, 6] and relevant SER results, but  $SET_i$  in Fig. 7 can derive multiple fault BDDs with the same fault ID defined in Section 4.2. If those BDDs across the different fault gates will propagate in topological order, SERs due to multiple faults can be calculated in the same manner.

The re-convergence check procedure in Fig. 7 is executed once starting from PIs and F/Fs prior to enter the main loop. If the traversal from  $p$  by Depth First Search (DFS) finds the re-convergent point  $g$  in the circuit, the nodes on the backward path from  $g$  are marked as the re-convergent path. DFS is inherently a recursive structure so that the return path from the re-convergent point can be easily identified and marked. By the basic rule in Section 4.2, those nodes on the re-convergent path will not be replaced by virtual PIs during BDD propagation, except that input BDDs are greater than the critical size. Similar to the static BDD for  $p$  in Fig. 7, evaluation is performed once at the beginning and re-used in later propagation analysis.

As a more aggressive option to reduce the size of fault

BDDs, virtual PIs can be added if the fault BDD has more vertices and edges than the pre-defined maximum count (e.g.,  $2 \times 10^4$ ). This modification can further reduce the size of the propagated BDDs and the time complexity for the analysis in Eq. (4) would be limited to the polynomial time. As mentioned in Section 4.2, it implies, however, that the correlation between the values of PIs and F/Fs might be eliminated unintentionally. The experiment shows the amount of errors that can be generated with this approach.

## V. EXPERIMENTS

This section describes the framework for SER estimation used to conduct the experiments and comparative studies mainly focusing on the execution time of the analysis procedure. The results for practical logic designs will also be covered in this section.

### 1. Logic SER Estimation Framework

The procedures in Section 4.4 were written by C++. The entire framework as illustrated in Fig. 8, consists of a SET characterized cell library, a fault generation function, a gate-level netlist parser and the propagation engine running on an Intel Xeon E5-2697. We utilized only one thread of the host processor in this paper. The input netlist for the target design can be obtained with a commercial logic synthesis tool. The logic probabilities of PIs and F/Fs should be extracted by the gate-level simulators and the utility program using the tool

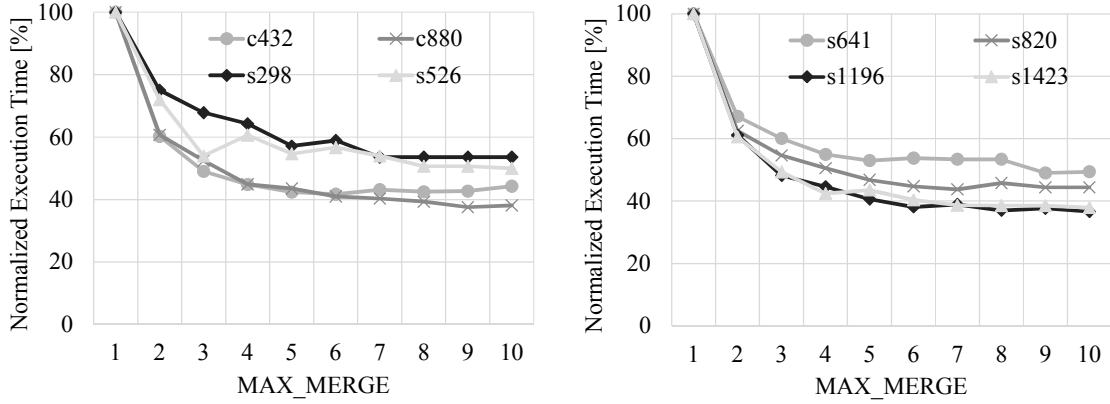


Fig. 9. Run-time improvement in the proposed technique with respect to  $MAX\_MERGE$ .

command language (TCL). A 45nm open cell library was chosen for the SET analysis. To characterize the library, we simulated each logic cell at SPICE-level. Possible SET sources as a behavioral current function were added to the faulty site of the SPICE circuit, varying its load capacitance  $C_L$  and collected charge  $q$ . SPICE-level simulation should be iterated until the entire target cells are characterized. Consequently, the SET cell library contains SET widths as well as falling and rising times from the simulation results. To extract the specific SET instance for  $q$  and  $C_L$ , two-dimensional interpolation will be conducted. During the SER analysis, electrical attenuation by a logic cell delay was estimated using existing techniques [7, 10]. This paper focused on evaluating sea-level SER for the logic circuit. Thus, the neutron flux ( $F_n$ ) is defined as  $56.15 \text{ n/m}^2/\text{s}$  for 10-1000 MeV [20] and the effective injection rate ( $\alpha$ ) of a neutron that is technology independent, is set to  $2.2 \cdot 10^{-5}$ , as mentioned in [21].

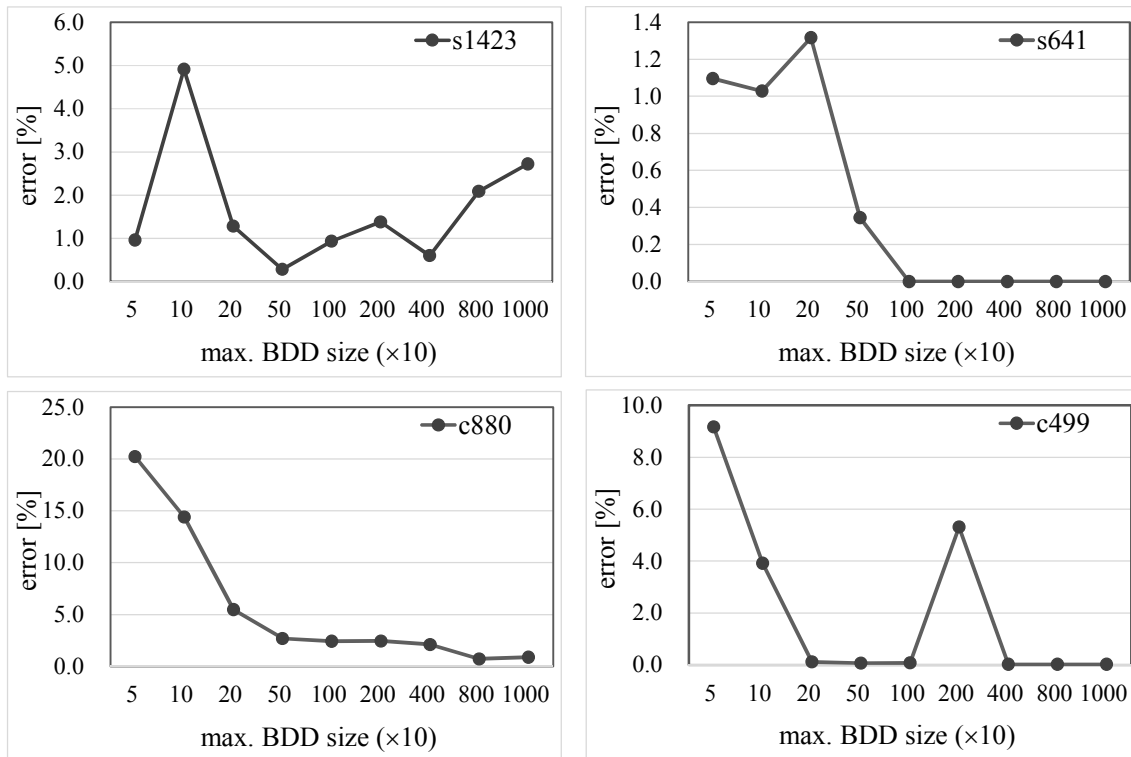
## 2. Results and Discussion

Cascaded fault propagation accompanies successive faults in topological order. The number of logic gates containing such faults should be practically limited by  $MAX\_MERGE$  due to the size of the fault BDDs mentioned in Section 4.1 and 4.4. Fig. 9 shows the reduction in the execution time of SER estimation when  $MAX\_MERGE$  changes from 1 to 10. The target designs were ISCAS-85/89 benchmark circuits that were logic-synthesized with a 45nm cell library. Note that all SET instances in a single logic gate are concurrently attached to BDD even with  $MAX\_MERGE=1$ . With  $MAX\_MERGE=5$  in Fig. 9, run-time was improved by 2-2.5 fold between  $MAX\_MERGE=1$  and 5. It was also slightly improved above  $MAX\_MERGE=5$ , but there were no noticeable differences.

Table 1. BDD size with and without virtual PI insertion on non-re-convergent fan-out

Circuits	Without VPI		With VPI on non-re-convergent path	
	Avg. BDD size	Max. BDD size	Avg. BDD size	Max. BDD size
c432	287	1568	287	1568
c499	5,936	28,445	4,765	28,349
c880	4,214	372,080	6,716	147,512
s641	104	932	87	662
s1196	66	1175	60	1049
s1423	2,370	45,611	317	6,821

The virtual PI insertion on non-re-convergent fan-out explained in Section 4.2 reduces the sizes of static and fault BDDs. Table 1 shows the maximum and average sizes of BDDs with and without virtual PI insertion. This technique reduces the vertices and edges of BDDs by 30% on average and up to 87% reduction can be accomplished in the case of  $s1423$ . Inserting the virtual PIs to over-sized static and fault BDDs effectively prevents exponential growth of propagated BDDs and improves the speed of analysis. However, this might lead to unbounded errors in the estimation when the target path has highly correlated input and fault events. Fig. 10 shows the errors of SER values for the original case with respect to the maximum size of the BDDs. In this setup, if a fault or a static BDD is larger than the pre-defined maximum size, a virtual PI would replace the existing input vertices. The results confirm that the differences can be extended by up to 20% of the original values so



**Fig. 10.** Errors due to compulsory virtual PI insertion on the over-sized BDDs.

we incorrectly estimate the SERs of the designs when over-limiting the size of the BDD. This comes from the fact that the logic masking effects in conjunction with the electrical property for the fault in Eq. (5) are eventually ignored when calculating SERs at the output ports as their sensitized vertices are removed by the virtual PI. Without considering correlation for PIs, reconversion fan-outs which are highly correlated paths can be chosen for such reduction when their BDDs are over-sized. In that case, errors on analysis will be increased regardless of size limit. It shows a small fluctuation for estimation errors in Fig. 10, as the size limit changes. This also indicates that logical redundant techniques such as triple modular redundancy (TMR) and redundant addition and removal (RAR) [11] might not be accurately estimated by such independent event processing. However, as shown in Fig. 10, the errors are less than 3% when the maximum size is set to over 1000. The virtual PI can also be selectively chosen when the target input of the logic gate has a small *ISER* value or less correlated with other input values, similar to the technique used in Section 4.3.

As shown in Fig. 11, skipping the logic gate traversal which generates small *ISER* value helps the speed up for the entire analysis within a limited estimation error.

When varying the skipping ratio, errors were extracted by identifying the difference in the SER in comparison to the SER without a skipping check. Errors due to the skipping policy varied among different logic designs and fan-out structures. With skip ratio=0.01, we expect less than 2% errors on total SER but over 80% faster in runtime on average.

Table 2 summarizes the results of run-time comparison with the existing works. The baseline algorithm was developed using the key procedures of the original BDD techniques [11, 12]. Exceptionally, static BDD constructions were involved in the main propagation analysis as in Fig. 7. A two-input standard cell in this experiment contains more than 40 SET instances whereas the existing works [11, 12] employed only a few candidate faults with different widths. The parallelized method based on BDD [16] in our previous work involved individual SET analysis separated into multiple threads. The static propagation method is a non-BDD analysis but the individual propagation paths are mostly regarded as independent events [7, 10]. This method was implemented by tool command language (TCL) running on the commercial static time analysis tool. As shown in Table 2, the proposed algorithm including three

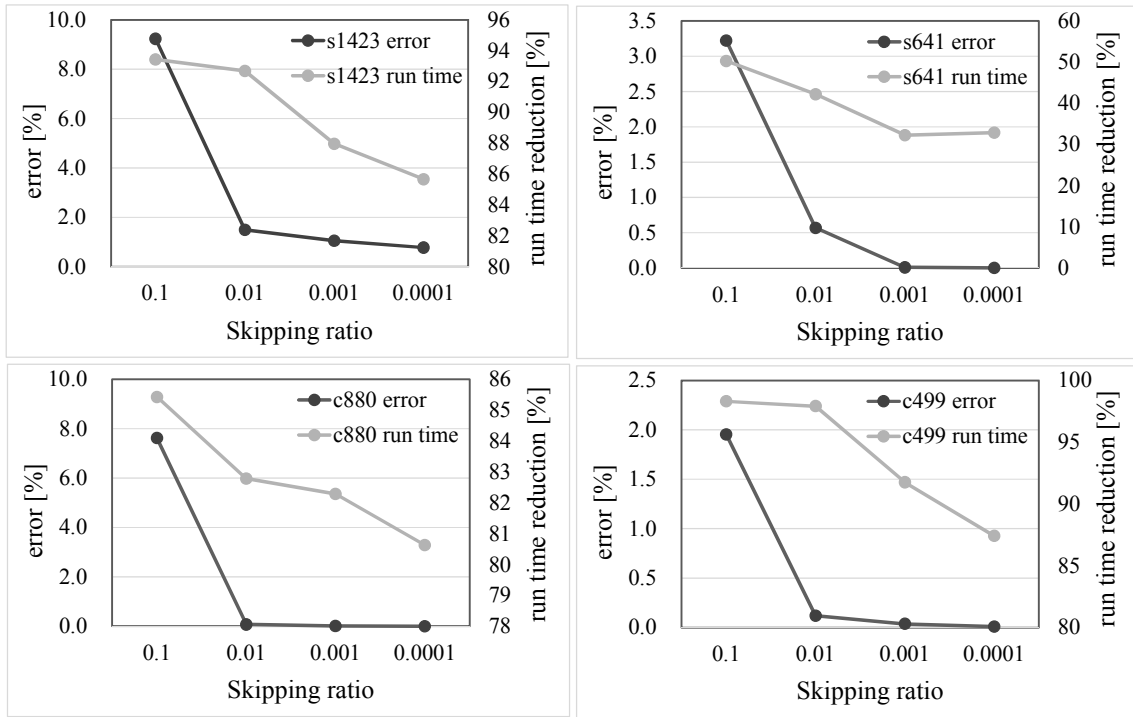


Fig. 11. Estimation errors and run-time reduction with respect to the skipping ratio.

Table 2. Comparative results on run-time for complete SER analysis in [sec.]

Circuits	Baseline [11, 12, 16]	Parallelized [16]	Static path [7, 10]	[6]	[22]	This work
c432	22.1	7.5	23	12.09	5.6	1.3
c499	1734.0	725.2	1606	35.68	30.1	4.8
c880	2663.2	1230.3	2311	21.43	4.3	2.2
c1355	-	-	1765	39.82	15.6	6.2
c2670	-	-	2604	48.54	8.4	5.1
c5315	-	-	6163	109.05	30.6	20.9
s298	18.9	6.7	12	-	-	0.3
s344	14.9	3.3	25	-	-	0.3
s444	35.1	14.5	40	-	-	0.5
s526	54.1	19.3	65	-	-	0.7
s641	148.7	32.0	59	-	-	0.9
s820	150.3	41.9	62	-	-	0.7
s1196	713.7	221.0	368	-	-	3.1

techniques is 20-200 times faster than its baseline algorithm, and is even 5-100 times faster than the execution time with parallelized work [16] where 12 concurrent threads are provided for the propagation analysis. In comparison of the results of the baseline algorithm, on average, errors on SERs were observed less than 1%. Compared to the other works employing non-BDD structures [6, 22], the proposed technique

Table 3. SER estimation for practical logic designs

Circuits	# of PIs	# of POs	Gate count	Block SER [FIT]	Run time [min.]
add16 TMR	32	16	730	2.54E-5	0.0082
mul16 TMR	32	32	10,965	5.33E-5	0.44
mul32 TMR	64	64	45,118	1.08E-4	3.32
DES-64	2	11	35,019	1.19E-4	1.93
cortex-m0	54	82	20,660	3.32E-3	2.95
leon3-minimal (processor only)	6,893	2,249	60,135	5.94E-3	30.60

shows competitive run-time performance.

Table 3 lists the SER evaluation results for several practical designs which have up to 60,000 flattened logic cells. The target designs were modified by a single logic block removing internal hierarchical boundaries. In more complex designs containing many logic blocks, a logic circuit containing more than an internal block can sum up the SER results of the lower hierarchical blocks and Eq. (2) will be used to evaluate the total SER of the complex design. A few TMR designs for the arithmetic units in Table 3 with the identical SER per PO were used to confirm that all soft errors due to the SET instances were mostly originated from their voting circuitry. Other SET sources in the remaining sites were masked by the TMR

structure. Besides, ones of the largest designs, *cortex-m0* and *leon3-minimal*, could also be estimated by the proposed technique within 30 min. of execution. By limiting several critical BDDs, the execution time is mostly proportional to the number of logic cells contained in the target design. This agrees with the time complexity of the BDD-based propagation procedure presented in Section 3, except *leon3-minimal* including many floating nets from un-used cache blocks. We believe that the run-time of our current framework can be further improved by applying several parallelized methods such as in [16]. The results show that a temporal fault analysis based on BDD structures is applicable to more complex logic designs.

## VI. CONCLUSION

In this paper, a cascaded fault propagation and reduction techniques for SER analysis are presented and validated. Applying BDD structures is necessary for SET propagation if we consider the exact logic masking effects within the internal logic circuit. The approximation method that involves inserting virtual PIs can limit the growth of the BDD size during the propagation analysis. Successive faults are added to the propagated BDD in topological order and eliminate unnecessary revisits of the logic gate traversal. These techniques make the estimation feasible when even a single logic gate has more than tens of SET sources inside. Consequently, the run-time can be improved by 20-200 times compared to the baseline algorithm. Our future works include a parallelization of the algorithm, a radiation hardened logic circuit design and tape-out validation by using the proposed framework prior to manufacture. The results will be compared to radiation test results obtained in an accelerator facility.

## ACKNOWLEDGEMENTS

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2060954).

## REFERENCES

- [1] M. Ebrahimi, A. Evans, M. B. Tahoori, D. Alexandrescu and V. Chandra, "Comprehensive Analysis of Sequential and Combinational Soft Errors in an Embedded Processor," *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol. 34, No. 10, pp. 1586-1599, Apr., 2015.
- [2] R. D. Schrimpf, M. A. Alles, F. E. Mamouni, D. M. Fleetwood, R. A. Weller and R. A. Reed, "Soft Errors in Advanced CMOS Technologies," *Proc. of 11th IEEE Solid-State and Integrated Circuit Tech.*, pp.1-4, Oct., 2012.
- [3] H. Liu and S. Datta, "Soft-Error Performance Evaluation on Emerging Low Power Devices," *IEEE Trans on. Device and Materials Reliability*, Vol. 14, No. 2, pp.732-741, Apr., 2014.
- [4] M. Ebrahimi, H. Asadi, R. Bishnoi and M. B. Tahoori, "Layout-Based Modeling and Mitigation of Multiple Event Transients," *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol. 35, No. 3, pp. 367-379, July, 2016.
- [5] M. Ebrahimi, R. Seyyedi, L. Chen and M. B. Tahoori, "Event-driven Transient Error Propagation: A Scalable and Accurate Soft Error Rate Estimation Approach," *Proc. of 20th ASP-DAC*, pp.743-748, Jan., 2015.
- [6] H-M. Huang and C. H.-P. Wen, "Layout-Based Soft Error Rate Estimation Framework Considering Multiple Transient Faults—From Device to Circuit Level," *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol. 35, No. 4, pp. 586-597, Aug., 2016.
- [7] S. Kwon, J. K. Park, J. T. Kim, "An Approximated Soft Error Analysis Technique for Gate-level Designs," *IEICE Electronic Express*, Vol.11, No.10, pp.1-7, May, 2014.
- [8] C.-C. Austin, H.-M. Ryan and W. H.-P. Chen, "CASSER: A Closed-Form Analysis Framework for Statistical Soft Error Rate," *IEEE Trans. on VLSI Systems*, Vol. 21, No. 10, pp.1837-1848, Oct., 2013.
- [9] M. Zhang and N. R. Shanbhag, "Soft-Error-Rate-Analysis (SERA) Methodology," *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol. 25, No. 10, pp. 2140-2155, Aug., 2006.
- [10] R. Rajaraman, J. S. Kim, N. Vijaykarishnan, Y. Xie and M. J. Irwin, "SEAT-LA: A Soft Error Analysis Tool for Combinational Logic," *Proc. of 19th Int.*

- Conf. on VLSI Design*, pp.499-502, Jan., 2006
- [11] K.-C. Wu and D. Marculescu, "A Low-Cost, Systematic Methodology for Soft Error Robustness of Logic Circuits," *IEEE Trans. on VLSI Systems*, Vol. 21, No. 2, pp. 367-379, Feb., 2013.
- [12] B. Zhang, W. Wang, and M. Orshansky, "FASER: Fast Analysis of Soft Error Susceptibility for Cell-Based Designs," *Proc. of 7th Int'l. Symp. on Quality Electronic Design*, pp. 755-760, Mar., 2006.
- [13] J. K. Park and J. T. Kim, "An Evolutionary Approach to the Soft Error Mitigation Technique for Cell-Based Design," *Advances in Electrical and Computer Eng.*, Vol. 15, No. 1, pp.33-40, Feb., 2015.
- [14] H. Asadi, M. B. Tahoori, M. Fazeli and S. G. Miremadi, "Efficient algorithms to accurately compute derating factors of digital circuits," *Microelectronics Reliability*, Vol. 52, pp.1215-1226, Jun., 2012.
- [15] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. on Computers*, Vol. C-35, No. 8, pp.677-691, Aug., 1986.
- [16] M. Kim, J. K. Park and J. T. Kim, "Implementation and Analysis of parallelized Binary Decision Diagram manipulation on multicore processors," *2015 Int'l Conf. on Parallel and Distributed Processing Techniques and Applications*, pp.394-397, Jul., 2015.
- [17] B. Bollig and I. Wegener, "Improving the Variable Ordering of OBDDs Is NP-Complete," *IEEE Trans. on Computers*, Vol. 45, No. 9, pp.993-1002, Sep., 1996.
- [18] N. M. Zivanov and D. Marculescu, "MARS-C: Modeling and Reduction of Soft Errors in Combinational Circuits" *proc. of DAC*, pp.767-772, Jul., 2006.
- [19] Y. Kuo, H. Peng, and C. Wen, "Accurate statistical soft error rate (SSER) analysis using a quasi-Monte Carlo framework with quality cell models," *proc. of 11th Int'l. Symp. on Quality Electronic Design*, pp.831-838, Mar., 2010.
- [20] J. F. Ziegler, "Terrestrial cosmic rays," *IBM J.*, Vol. 40, No. 1, pp.19-39, Jan., 1996.
- [21] T. Karnik and P. Hazucha, "Characterization of Soft Errors Caused by Single Event Upsets in CMOS Processes," *IEEE Trans. on Dependable and Secure Computing*, Vol. 1, No. 2, pp. 128-143. Nov., 2004.
- [22] H.-M. Huang and C. H.-P. Wen, "Fast-Yet-Accurate Statistical Soft-Error-Rate Analysis Considering Full-Spectrum Charge Collection," *IEEE Design and Test*, Vol. 30, No. 2, pp.77-86, Mar. 2013.



**Jong Kang Park** received BS and MS degrees in Electric, Electronics and Computer Engineering in 2001, 2003 and Ph.D. degree in Electric and Electronics Engineering from Sungkyunkwan University, Korea in 2008. From 2008 to 2013, he was

with Samsung Electronics where he designed touch sensor ICs. He is now a research professor, School of Electronic and Electrical Engineering, Sungkyunkwan University. His current research interests include the sensor data acquisition, embedded system design, soft error analysis and tolerance techniques.



**Myoungha Kim** received the BS degree and MS degree in Electronic and Electrical engineering at the Sungkyunkwan University in 2014, 2016. In 2016, He joined VISOL Corporation as a firmware engineer.

His current area of interest is LEDs for high power, high speed lighting application, and RTOS related issues.



**Jong Tae Kim** is a Professor at the School of Electronic and Electrical Engineering, Sungkyunkwan University, where he has been since 1995. He received the BS degree in electronics engineering from Sungkyunkwan University in Korea in

1982 and the MS and PhD degrees in electrical and computer engineering at the University of California, Irvine, in 1987 and 1992, respectively. From 1991 to 1993 he was with the Aerospace Corporation in Elsegundo, California. He was a full-time lecturer at Chunbuk National University in Korea from 1993 to 1995. His research interests include SoC design and design methodology, embedded systems, and multi-core processor architecture.