

A Cell Sizing Technique for Mitigating Logic Soft Errors in Gate-level Designs

Jong Kang PARK, Jong Tae KIM*

School of Electronic and Electrical Eng., Sungkyunkwan Univ., 300 Cheoncheon-dong Jangan-gu, Suwon, Gyeonggi-do 440-746, South Korea

*jtkim@skku.edu

Abstract—The effect of logic soft errors on the degradation of the reliability becomes more crucial in the case of nanometer semiconductor designs. Several hardening techniques have been reported from the transistor- to system-level. In order to suppress the single event transients originating from logic gates, this paper presents an improved heuristic search utilizing the gate-sizing technique. The algorithm re-orders the gate-traversal to maintain the reduced soft error rates of the preceding logic gates. The preferential candidates for the two successive algorithms are the logic gates near the primary outputs and flip-flops, rather than those of the higher portions of block soft error rate. The proposed technique reduces the logic soft error rate by more than 60% compared to the existing method in 45nm CMOS cell designs.

Index Terms—single event transient, soft error, soft error mitigation, gate-level, gate sizing, cell sizing.

I. INTRODUCTION

Soft errors in silicon devices are one of the most important issues affecting the system reliability for the current and future technology nodes [1-4]. Although the individual sensitive region decreases as the feature size shrinks, the corresponding degree of transistor integration and the critical charge result in an increase in the soft error rate (SER) per device. Especially, the logic SER including single event transient (SET) grows rapidly compared to the SRAM SER of below 90nm technology [1,12]. Well-known low power controls such as dynamic voltage and frequency scaling (DVFS) can suffer from an increasing logic SER due to the exponential growth of the SER with increasing supply voltage [1]. Mitigation techniques in combinational and sequential logic circuits become more important for the reduction in system SER for upcoming technologies.

Gate sizing techniques [5-11] for the reduction of the logic SER are based on a heuristic search and are easily applicable to today's cell-based design flow. Because design metrics are existing, such as the path delays, power consumption and circuit size (area), it is difficult to obtain the optimal solution which also includes the minimization of the SER, without degrading the other design metrics. The SER can be minimized by priority-based traversing using a priority queue [9,10]. The priority of the logic gate is determined by its own contribution to the block SER; a logic gate with a larger individual SER is preferable to traverse during the incremental optimization. Since soft error minimization with a constrained gate-level design is a non-deterministic polynomial-time hard (NP-hard) problem [10], such heuristics must be effective in terms of the execution

time and quality of solution. This greedy approach could suffer from the inter-dependency of incremental optimization [7]. Other than gate-resizing techniques, the existing logic minimization techniques such as redundancy addition and removal (RAR) can be another good candidate to reduce the SER in gate-level designs utilizing the logical masking property [12].

In this paper, we present a lightweight soft error mitigation algorithm that searches for the proper gate sizes within constrained cell-based designs. The target designs have constraints in terms of the marginal circuit area and path delay. Generally, the total block SER is the cumulated probability for the SETs generated from the internal logic gates. Logic gates, which give a higher SER to the flip-flops (F/Fs) and primary outputs (POs), can be a preferred candidate for optimization. The individual gate sizing has an impact on the former optimization results and degrades the quality of the solution. To address this inefficiency, we use a modified topological sort that preserves the preceding local optima during gate sizing traversal. In the previous study [7], there were degraded optimization results especially for low margins of the design constraints. In this paper, two heuristics are proposed, one is for the total area and SER reduction and the other is to mitigate SER only within the area and delay constraints. The subsequent runs of the two algorithms have better reduction in SER than [7]. We have verified the proposed algorithms with the characterized 45nm open cell library.

II. A HEURISTIC SEARCH FOR SOFT ERROR MITIGATION

In this section, we explain a heuristic search that is primarily used for the soft error mitigation flow described in section III. We expand on our previous work [7] by providing a mathematical proof. This is an incremental optimization which performs gate sizing to reduce the individual SER for each logic gate in topological order. We assume that the target design is a logic-synthesized netlist and allows certain degrees of overhead for the design constraints. That is, the algorithm increases the size of the circuit elements compared to the original design, in order to have tolerance against radiation effects. In contrast to the existing greedy approaches, this heuristic algorithm preserves the former local optima [7].

A. Objective function of gate-level soft error mitigation

The technology-mapped netlist B consists of logic gates, $g_k \in G$, $0 \leq k \leq |G|-1$ and the interconnections. We re-define B with regard to the specific gate sizing as follows:

$$X = \{x_k \mid g_k \text{'s driving strength}, 0 \leq k \leq |G| - 1\} \quad (1)$$

where $x_k = \{1, 2, 4, \dots\}$, as defined by the target cell libraries and the types of logic cells. For example, if a NAND gate can be one of the three driving strengths, then it has different transistor sizes and input-to-output characteristics. Fig. 1 shows the gate sizing result which replaces driving strength x_1 with x_2 at g_1 , without changing its logical function. Here, the objective function for minimizing the number of soft errors can be written as follows:

$$\text{Minimize} \quad C(X) = \sum_i^{|G|} ISER_i \quad (2)$$

$$\text{s.t.} \quad d_{crit} \leq d_{orig} \cdot M_d, A_X \leq A_{orig} \cdot M_A$$

where $ISER_i$ denotes the individual SER from gate i to the POs and F/Fs. This is a similar metric to the MEI(mean error impact) [8] and EPP(error propagation probability) [10]. d_{crit} , A_X , M_d and M_A are the critical path delay, circuit area(=gate size) and constraint margins for the delay and circuit area, respectively. These design parameters are marginally constrained to the original critical path delay, d_{orig} and circuit area, A_{orig} . $C(X)$ in (2) is equivalent to the block SER, but the definition of $ISER$ indicates that the optimization procedure is based on gate-to-gate traversal. For a given total neutron flux F_n and technology-independent rate parameter α , we define $ISER_i$ from the re-definition of the 1st state SER(j) in [13] as follows:

$$ISER_i(j) = F_n \cdot \alpha \sum_{SETi} \sum_q f_Q(q) A_i(SETi) GP(SETi) LP_{ij} EP_{ij} LW_{ij} \quad (3)$$

$$ISER_i = \sum_j^{[PO]+[FF]} ISER_i(j) \quad (4)$$

where $ISER_i(j)$ denotes the SER at port j , which is either a PO or an F/F and the corresponding SETs are generated at gate i . $f_Q(q)$, $A_i(SETi)$ and $GP(SETi)$ denote a region of the faulty site (typically, the drain area of the MOSFET) of gate g and the logical generation probability for a $SETi$ from gate i , respectively. Also, LP_{ij} , EP_{ij} and LW_{ij} are the probabilities of logical propagation, electrical attenuation and latching window from gate i to port j , respectively. $GP(SETi)$, which is directly correlated with $A_g(SETi)$, is determined by the statistics obtained from the gate-level simulation results. For example, in a NAND gate, as shown in Fig. 2, SETs having different widths and peak voltages would be generated under different input bias conditions and faulty sites, even for the same collected charge, q [14,15]. Therefore, we can define a SET instance ($SETi$) as an object, which has a

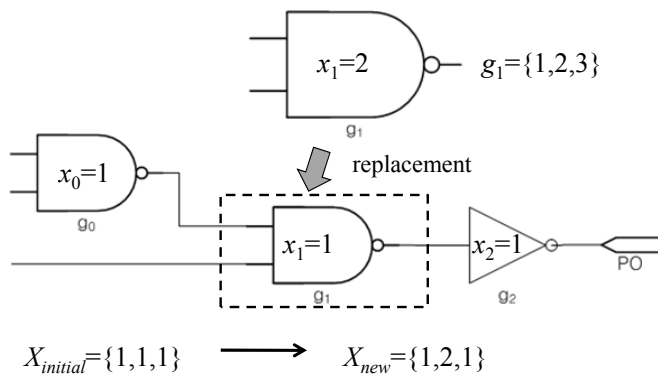


Figure 1. Cell sizing and its representation

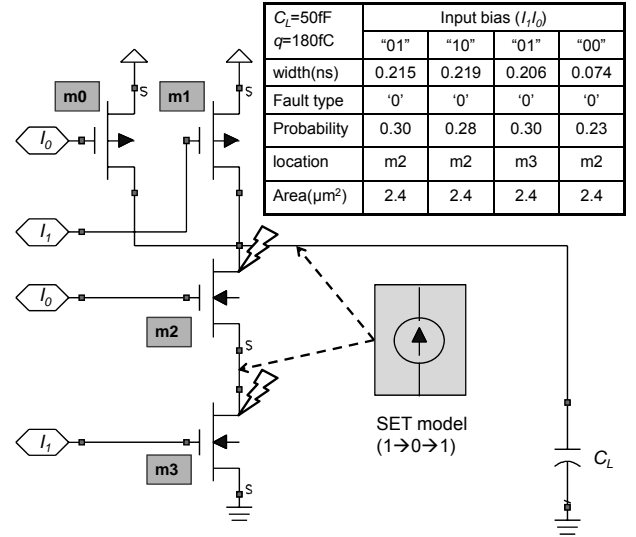


Figure 2. SET instances for 2-input NAND gate

specific pulse width, fault type ('0' or '1'), fault area and generation probability for the input combination. $SETi$ can be extracted using the pre-characterization results obtained from the SPICE and gate-level simulations for standard cell-based designs. For a given node capacitance, the width of $SETi$, which means the time difference to change 50% to 50% of the noise voltage, can be obtained from the linear interpolation between the discrete pulse widths and corresponding load capacitances.

B. Topological search

The gate sizing technique does not change the circuit topology. This implies that the logical propagation property of the circuit is maintained; only the delay characteristic is changed. Since the degree of electrical attenuation for the SETs is strongly related to the gate propagation delay [11,18,20], the block SER might vary with the specific driving power for the individual logic gate. The propagation delay can be modeled as a function of the transition time and the total load capacitance, which is determined by the gate driving capacity, as shown by the typical CMOS non-linear delay model (NLDM). Consequently, if the logic gate is re-sized during optimization, then the propagation delays of its precedent gates must be changed and the former local optimum for (2) is not preserved.

Fig. 3 shows an example of gate sizing for the SER reduction. Initial $ISER$ for each logic gate is defined as $ISER(g_1) > ISER(g_2) > ISER(g_3) > ISER(g_4) > ISER(g_5) > ISER(g_6)$. Assuming that the logic gates g_1 and g_2 have already been re-sized; g_1 and g_2 are preferentially visited due to their high $ISER$ values in the greedy approaches. If we allot the capacity of g_3 to another candidate, the transition time and load capacitance values at g_1 and g_2 will vary and, consequently, the $ISER$ values of g_1 and g_2 will be changed. This affects the optimal gate sizes for g_1 and g_2 . It is essential for preserving the previous optimization results to keep the partial order, $g_3 \rightarrow g_1 \rightarrow g_2$ and $g_3 \rightarrow g_2 \rightarrow g_1$ in visiting the logic gates. When we visit g_1 or g_2 after re-sizing g_3 , none of the choices for g_1 and g_2 alter the $ISER$ for g_3 . None of the SET instances generated from the sensitive region of g_3 propagate to g_1 and g_2 . Thus, the topological sort starting from the POs and F/Fs forms the basis for constructing the priority queue. It is also effective, because the logic gate

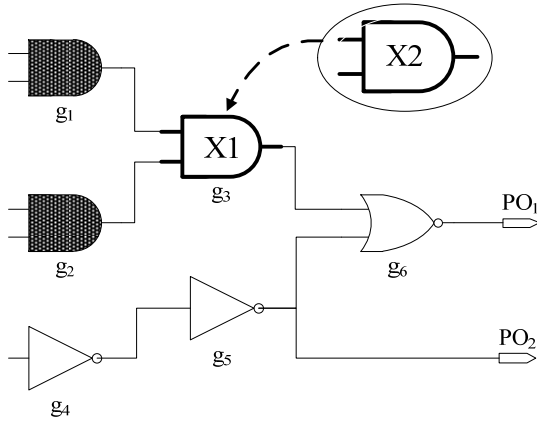


Figure 3. A gate-sizing example for SER reduction

near to the POs or F/Fs tends to have a high *ISER*. The SET instances propagated through a longer circuit path would be electrically and/or logically attenuated.

Since the greedy choice is still effective for the gates with no inter-dependency, a better solution can be obtained when the logic gates do not need to keep the partial order, but are ordered according to the *ISER*. In Fig. 3, it would be $g_6 \rightarrow g_3 \rightarrow g_5 \rightarrow g_1 \rightarrow g_2 \rightarrow g_4$.

For the proof of the local preservation in the proposed heuristic, we define an operator which represents the precedence relation between two nodes. Let $g_i \prec g_j$ in B mean that the output of g_j is logically dependent on the outputs of g_i (e.g., $g_1 \prec g_3$ and $g_2 \prec g_6$ in Fig. 3). Similarly, let $g_i \ll g_j$ in B mean that g_i immediately precedes g_j . There are no other logic gates between g_i and g_j . Further, we define the propagation delay and output transition time of the logic gate g_k based on the NLDM as follows:

$$d(g_k) = f_d^{(k)}(tr(g_j), C(g_k)), \quad g_j \ll g_k \quad (5)$$

$$tr(g_k) = f_{tr}^{(k)}(tr(g_j), C(g_k)), \quad g_j \ll g_k \quad (6)$$

where $tr(g_k)$ and $C(g_k)$ denote the output transition time and load capacitance for g_k , respectively. Here, g_j would be unique if no bus or three-state buffer is allowed. The input transition time of g_k is identical to the output transition time of g_j in this paper. Commonly, $f_d^{(k)}$ and $f_{tr}^{(k)}$ for (5) and (6) have been implemented by two dimensional lookup table and interpolation using EDA (electronic design automation) tools. Let g'_k be the replaced logic gate after gate re-sizing.

Lemma 1. For all g_k , g_l and g_m , satisfying $x_l \neq x'_l$, $x_m = x'_m$, $g_l \prec g_k$ and $g_k \prec g_m$, if $x_l \neq x'_l$, it is not true that $d(g_l) = d(g'_l)$, $tr(g_l) = tr(g'_l)$, $d(g_m) = d(g'_m)$, $tr(g_m) = tr(g'_m)$.

(direct proof) For $g_a \prec g_l$, $d(g_l) = f_d^{(l)}(tr(g_a), C(g_l))$, $tr(g_l) = f_{tr}^{(l)}(tr(g_a), C(g_l))$. Because $g_l \prec g_k$ and $x_l \neq x'_l$, it is clear that $C(g_l) \neq C(g'_l)$ and this also implies that $d(g_l) \neq d(g'_l)$ and $tr(g_l) \neq tr(g'_l)$. Similarly, $d(g_m) \neq d(g'_m)$ and $tr(g_m) \neq tr(g'_m)$ where $d(g_m) = f_d^{(m)}(tr(g_k), C(g_m))$, $tr(g_m) = f_{tr}^{(m)}(tr(g_k), C(g_m))$ and $tr(g_k) \neq tr(g'_k)$.

Lemma 2. For all g_n which satisfy $g_k \prec g_n$, if $x_k \neq x'_k$, it is not true that $d(g_n) = d(g'_n)$ and $tr(g_n) = tr(g'_n)$.

(proof by induction) By Lemma 1, it is not true that

$d(g_m) = d(g'_m)$, $tr(g_m) = tr(g'_m)$ where $g_k \prec g_m$ and $x_k \neq x'_k$ (initial condition). Again, for all g_a , $g_m \prec g_a$ where $d(g_a) = f_d^{(a)}(tr(g_m), C(g_a))$, $tr(g_a) = f_{tr}^{(a)}(tr(g_m), C(g_a))$, it is clear that $d(g_a) \neq d(g'_a)$, $tr(g_a) \neq tr(g'_a)$ because $tr(g_m) \neq tr(g'_m)$. These processes are repeated at the successive logic gate $g_b \in G_b$, where $G_b = \{g_b | g_b \prec g_a\}$. When $g_m \in G_m$ and $g_a \in G_b$, $G_n = G_m \cup G_a \cup G_b$.

When we replace the driving capacity x_k with x'_k , the *ISERs* for the precedent gates of g_k might vary. From Lemma 1 and 2, we can derive the characteristics for the variance and invariance of the *ISER* with gate sizing.

Lemma 3. If $x_k \neq x'_k$ then, it is not true that $ISER_p = ISER'_p$ for all g_p which agree with $g_p \prec g_k$.

(direct proof) Let $D(g_p) = \{d(g_i) | g_p \prec g_i\}$ and $TR(g_p) = \{tr(g_i) | g_p \prec g_i\}$. Since the electrical attenuation of the SET is dependent on the propagation delay for each logic gate, we derive $ISER_p = f(tr(g_p), D(g_p), TR(g_p))$ from (3)-(6). Note that $tr(g_p)$ is defined as the transition time of *SETi* which is the starting location for calculating $ISER_p$. Let $D(g_n) = \{d(g_n) | g_k \prec g_n\}$ and $TR(g_n) = \{tr(g_n) | g_k \prec g_n\}$. If $x_k \neq x'_k$ then, $d(g_l) \neq d(g'_l)$ and $tr(g_l) \neq tr(g'_l)$ for $g_l \prec g_k$, $D(g_n) \neq D(g'_n)$ and $TR(g_n) \neq TR(g'_n)$ by Lemma 1 and 2. Because $\{d(g_l)\} \cup D(g_n) \subset D(g_{pn})$ and $\{tr(g_l)\} \cup TR(g_n) \subset TR(g_{pn})$, it is clear that $ISER_p \neq ISER'_p$.

Lemma 4. If $x_k \neq x'_k$ then, it is true that $ISER_n = ISER'_n$ for all g_n which agree with $g_k \prec g_n$.

(direct proof) $ISER_n = f(tr(g_n), D(g_n), TR(g_n))$ for $D(g_n) = \{d(g_l) | g_n \prec g_l\}$ and $TR(g_n) = \{tr(g_l) | g_n \prec g_l\}$. From $x_n = x'_n$ and Eq.(5,6), we can see that $tr(g_n) = tr(g'_n)$, $D(g_n) = D(g'_n)$, $TR(g_n) = TR(g'_n)$. Therefore, it is clear that $ISER_n = ISER'_n$.

By the properties of Lemmas 1-4, we show the *preservation of the local optima* in incremental SER mitigation.

Theorem 1. For reverse topological ordered gates, $\{g_i | 0 \leq i \leq n-1\}$, if $x_k \neq x'_k$ then $ISER_i = ISER'_i$, $i < k$.

(direct proof) It is clear that the logic gate $g_i (0 \leq i \leq n-1)$ satisfies one of following conditions by the topological order $i < k$: i) $g_k \prec g_i$, it is clear that $ISER_i = ISER'_i$ by Lemma 4. ii) g_i and g_k have no connections, then $D(g_i) \cap D(g_k) = \Phi$, $TR(g_i) \cap TR(g_k) = \Phi$ and $ISER_i = ISER'_i$ where $ISER_i = f(tr(g_i), D(g_i), TR(g_i))$, $D(g_i) = \{d(g_j) | g_i \prec g_j\}$, $TR(g_i) = \{tr(g_j) | g_i \prec g_j\}$, $D(g_k) = \{d(g_j) | g_k \prec g_j\}$, $TR(g_k) = \{tr(g_j) | g_k \prec g_j\}$.

Theorem 2. For descending ordered gates with *ISER*, $\{g_i | 0 \leq i \leq n-1\}$, if $x_k \neq x'_k$ then it is not true that $ISER_i = ISER'_i$, $i < k$.

(direct proof) The logic gate $g_i (0 \leq i \leq n-1)$ satisfies one of the following three conditions: i) $g_k \prec g_i$ or ii) having no connection between g_k and g_i , we can see that $ISER_i = ISER'_i$ from Theorem 1. However, in iii) $g_i \prec g_k$, it is not true that $ISER_i = ISER'_i$ by Lemma 3.

III. PROCEDURES FOR SOFT ERROR MITIGATION

The heuristic search in the topological order described in Section II, provides a simple way to refine the design metrics while including the soft error susceptibility. Fig. 4 shows the proposed procedures for soft error reduction based on this topological order. The first algorithm, the local optimizer for size (SIZLOPT), searches for a smaller gate size, while keeping each *ISER* not increased. This makes additional room to reduce the SER; typically, the circuit size is the bottleneck in SER optimization rather than the path delays. Then, the second local optimizer for SER (SERLOPT) performs *ISER* minimization whereas the marginal area constraint and the result of SIZLOPT co-exist. These two algorithms traverse each logic cell by means of reversed-topological *ISER* sort (*ISER_RTOP_SORT*), which generates the sorted list in an *ISER*-based topological manner. The method employed for the SER analysis is based on our previous two-pass evaluation framework [13]. Since the 2nd stage of the SER evaluation is only dependent on the effects of logical masking, this paper mostly uses the pass-1 results for the given benchmark circuit optimization. The details of the proposed algorithms are presented in the following sub-chapters.

A. The Procedure for Topological Sort

In Fig. 5, we depicted the algorithm used for the *ISER*-based topological sort. There are two temporary queues and a sorted queue. *Sorting_q* is a list of logic gates whose succeeding gates have all been visited and sorted with *ISER* per gate size. Note that only the *ISER* value was used as the sorting key. Conversely, *Waiting_q* is a standby list of those gates whose succeeding gates have all been visited. First, *Sorting_q* is initialized as those nodes that are F/Fs or directly connected to POs. This list is then sorted by the function *sort_for_key* with *ISER* per gate size. On each iteration in the main loop, the first element of *Sorting_q* is put into *Sorted_q*. Whenever the gates immediately preceding the first element satisfy the partial order, they are inserted into *Waiting_q*. When *Sorting_q* is empty or the popped element is a sequential gate, it is replaced by *Waiting_q* and then sorted. This process is iterated until the final *Sorting_q* is empty.

Procedure *ISER_RTOP_SORT*

```

input : netlist with ISERs
output : Sorted_q
Mark all the gates as not visited
Sorting_q = {F/Fs and adjacent gates to POs}
while (Sorting_q is not empty)
    node = MAX_ISER_gate_pop_from(Sorting_q)
    Sorted_q ← node
    Mark node as visited
    if (node is a F/F)
        if (Sorting_q is empty)
            Sorting_q = Waiting_q
            Waiting_q = ∅
        foreach precedent_gate_of(node)
            if (next_gates for precedent_gate are all visited)
                Waiting_q ← next_gates
    if (Sorting_q is empty)
        Sorting_q = Waiting_q
        Waiting_q = ∅

```

Figure 5. Procedure for *ISER*-based topological sort

Procedure *SERLOPT*

```

input : netlist,  $X_{init}$ 
output : updated netlist,  $X'$ 
ISERs = calculate_ISERs( $X_{init}$ )
{ $X'$ , ISERs'} = ISER_based_topological_sort(netlist with ISERs)
for  $x_i \in X'$ 
    cur_ISER = ISERi
    LOPT = cur_ISER / area_of( $x_i$ )
    foreach available alternative cell,  $x_k$ 
         $X'$  = sizing_alternative_cell( $x_i, x_k$ )
        if (area constraint or delay constraint is violated)
            sizing_alternative_cell( $x_k, x_i$ )
            continue
        ISERk = update_ISER( $X'$ )
        LOPTk = ISERk / area_of( $x_k$ )
        if (LOPT > LOPTk)
            LOPT = LOPTk
    else
        sizing_alternative_cell( $x_k, x_i$ )

```

Figure 6. A procedure for SER mitigation

B. The Procedures for Soft Error Optimization

Fig. 6 shows the procedure used for the heuristic search, SERLOPT, using the *ISER*-based topological sort. The initial netlist X_{init} is pre-evaluated by a modified version of the soft error analysis technique [13] that calculates the individual SERs for the cell-based design. After performing the proposed topological sort, the logic gates are re-sized one-by-one. The function *sizing_cell* varies the driving strength from low to high so that *LOPT* is minimized. *LOPT* is equivalent to *ISER* per gate size. When the resultant design violates the design constraints defined in equation (2), the procedure restores the updated design and continues to select other candidate cells. The execution time of the *ISER* evaluation function, *update_ISER*, is highly dependent on the number of reconvergent fan-outs. Therefore, the time complexity of this heuristic algorithm is $O(|P||G|)$, where P denotes a set of signal paths appearing in B . In this procedure, the total summation of *ISER* is identical to the block SER, whereas re-evaluation is necessary to obtain the block SER in the greedy approach.

The procedure used for SIZLOPT in Fig. 7 is very similar to that of SERLOPT, except that *LOPT* is defined as the *ISER* itself. If we have more room for the marginal circuit size and delays in this step, the total SER can be reduced in

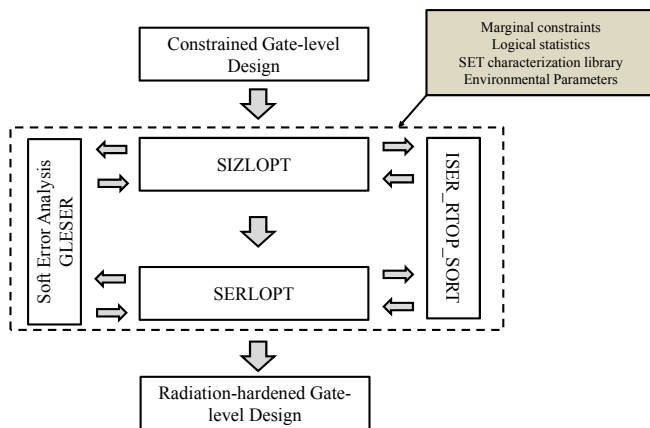


Figure 4. Proposed SER mitigation framework

the subsequent function, SERLOPT. The candidates to be replaced are those logic gates whose driving strengths are less than that of the input logic gate ($\in X_{init}$).

Procedure SIZLOPT

```

input : netlist,  $X_{init}$ 
output : updated netlist,  $X'$ 
 $ISERs$  = calculate  $ISERs(X_{init})$ 
 $\{X', ISERs'\}$  =  $ISER\_based\_topological\_sort$ (netlist with  $ISERs$ )
for  $x_i \in X'$ 
   $cur\_ISER = ISER_i$ 
   $LOPT = cur\_ISER$ 
  foreach available alternative cell,  $x_k \in X_i$ 
     $X' = sizing\_alternative\_cell(x_i, x_k)$ 
    if (prev_delay < current_delay and
      delay_constraint is violated)
      sizing_alternative_cell( $x_k, x_i$ )
      continue
     $ISER_k = update\_ISER(X')$ 
     $LOPT_k = ISER_k$ 
    if ( $LOPT \geq LOPT_k$ )
       $LOPT = LOPT_k$ 
  else
    sizing_alternative_cell( $x_k, x_i$ )
  
```

Figure 7. A procedure for cell size minimization

In conjunction with the circuit size, the path delays between the sequential elements are also protected in this algorithm. If the delay after re-sizing is increased and is greater than the design constraint, d_{origMd} , the result cannot be accepted as a solution. The re-sized design would be recovered to the previous one in this case.

IV. EXPERIMENTS

In this section, we show the experimental results for mitigating the block SERs of the *ISCAS-85* benchmark circuits. The target CMOS library is a 45nm open cell library [16] characterized by the PTM (predictive technology model). For the SET characterization process, spice simulations were performed iteratively and the SET widths of each logic gate were obtained according to the discrete load capacitances and collected charges. The logic cells consist of buffers, inverters, two-input NAND and NOR gates having different driving strengths. The input conditions and the valid sources of the SETs were chosen for each logic cell. The charge collection slope and time constant for the SET used in the SPICE simulation were extrapolated from the results of the existing research [17]. The gate-level netlists for the benchmark circuits were logic-synthesized from the structural-level Verilog descriptions. The corresponding pin-to-pin logical value statistics for the gate-level designs were also generated using the script-based gate-level simulation environment and random test vectors.

Table I lists the comparative results for the sea-level soft error reduction rates between the existing greedy approach and the proposed algorithm for the *ISCAS-85* benchmark circuits. The constraints were varied from 10% to 30% of the original gate-level design which had been already optimized by the logic-synthesis tool in terms of the area and path delays; there is little room to reduce the SERs without degrading the other design metrics. The sea-level SER in units of FITs(failure-in-time), which denotes the soft error rate after 1 billion hours, was used to evaluate the performance of the algorithms (the neuron flux is defined as

TABLE I. SER MITIGATION RESULT WITH DIFFERENT MARGINAL CONSTRAINTS

| Circuit | A_{orig} | d_{orig} [ns] | SER [FIT] | Heuristic | SER for $M_A, M_d=1.1$ [FIT] | SER for $M_A, M_d=1.2$ [FIT] | SER for $M_A, M_d=1.3$ [FIT] |
|---------|------------|-----------------|-----------|-----------|------------------------------|------------------------------|------------------------------|
| C17 | 8 | 0.1 | 5.14 E-6 | Greedy | 3.32E-6 | 3.41E-6 | 3.15E-6 |
| | | | | Proposed | 3.32E-6 | 3.32E-6 | 2.58E-6 |
| C432 | 205 | 1.4 | 3.52 E-5 | Greedy | 3.27E-5 | 2.44E-5 | 2.44E-5 |
| | | | | Proposed | 2.16E-5 | 1.99E-5 | 1.93E-5 |
| C499 | 547 | 1.3 | 1.97 E-4 | Greedy | 1.65E-4 | 9.34E-5 | 1.18E-4 |
| | | | | Proposed | 2.09E-4 | 3.64E-5 | 2.94E-5 |
| C880 | 421 | 1.5 | 1.16 E-4 | Greedy | 1.26E-4 | 1.47E-4 | 1.58E-4 |
| | | | | Proposed | 8.55E-5 | 8.15E-5 | 5.73E-5 |
| C1355 | 569 | 1.4 | 2.50 E-4 | Greedy | 1.30E-4 | 1.10E-4 | 1.53E-4 |
| | | | | Proposed | 2.72E-4 | 1.91E-5 | 2.43E-5 |
| C1908 | 511 | 1.8 | 1.64 E-4 | Greedy | 1.17E-4 | 2.32E-4 | 9.03E-5 |
| | | | | Proposed | 9.52E-5 | 6.83E-5 | 5.13E-5 |
| C2670 | 741 | 1.5 | 4.33 E-4 | Greedy | 4.38E-4 | 5.36E-4 | 5.20E-4 |
| | | | | Proposed | 3.30E-4 | 1.82E-4 | 7.51E-5 |

56.15n/m²/s for 10-1000MeV [19] and the effective neutron injection rate is $2.2 \cdot 10^{-5}$ [17]). The greedy algorithm employs the sorting method that re-orders the priority queue in the descending $ISERs$. Compared to greedy approach used in [7], the topological traversal reduces the SER to a lesser extent when allowing small design overhead to the algorithm. However, by the use of SIZLOPT prior to SERLOPT, it affords superior performance for most of the benchmark cases compared to the greedy search. This results in a 68% reduction for the SER at 130% released constraint on average as shown in Table II. The greedy algorithm suffers from its inefficient search structure. Larger margins of constraints cannot guarantee a higher SER reduction; the SERs are increased from 1.1x and 1.2x of the design constraints. Conversely, the proposed technique obtains monotonically decreased SERs by increasing the margins from 10% to 30%. At an increase in the design margin of 30%, it provides a reduction in the SER of more than 60% compared to the greedy technique. In addition, it shows a clearer improvement than the greedy from 10% to 30% of the increased design constraints, i.e., a reduction of 22% to 68% compared to the initial SER, while only a slight reduction is obtained in the greedy search. The time complexity for SIZLOPT and SERLOPT is nearly the same as that of the greedy approach; they traverse cell-by-cell including the evaluation of the $ISERs$.

V. CONCLUSION

This paper presents heuristic algorithms which reduce the block SER of a gate-level design which has a target delay

TABLE II. PERFORMANCE SUMMARY FOR SER REDUCTION

| Heuristics | Normalized to "Greedy Search" [%] | | | Normalized to Initial SER [%] | | |
|---------------------|-----------------------------------|-----|-----|-------------------------------|-----|-----|
| | 1.1 | 1.2 | 1.3 | 1.1 | 1.2 | 1.3 |
| Greedy-based Search | 100 | 100 | 100 | 82 | 88 | 80 |
| Proposed algorithm | 95 | 49 | 40 | 78 | 43 | 32 |

"Normalized to Greedy Search" summarizes the relative SER[%] obtained by each algorithm compared to the greedy approach.

"Normalized to Initial SER" denotes the reduction of the SER that can be achieved by each algorithm compared to the un-optimized design. The initial design has 100% of its own SER in this case.

and area constraint. The greedy search is the most common and cost-effective technique to suppress the SERs in a cell-based design, but its performance might be degraded in the increased marginal constraints. The proposed technique overcomes this limitation of the heuristic search and obtains improved SER mitigation results on average. Due to the nature of the cell-by-cell traversal, its computation complexity is the same as that of the existing greedy search.

REFERENCES

- [1] A. Dixit and A. Wood, "The Impact of New Technology on Soft Error Rates," IEEE Int. Reliability Physics Symposium, pp.5B4.1-5B4.7, 2011. [Online]. Available: <http://dx.doi.org/10.1109/IRPS.2011.5784522>
- [2] E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo and T. Toba, "Impact of Scaling on Neutron-Induced Soft Error in SRAMs From a 250 nm to a 22 nm Design Rule," IEEE Trans. On Electron Devices, Vol.57, No.7, pp.1527-1537, 2010. [Online]. Available: <http://dx.doi.org/10.1109/TED.2010.2047907>
- [3] C. Slayman, "Soft error trends and mitigation techniques in memory devices," proc. of Reliability and Maintainability Symposium, pp.1-5, 2011. [Online]. Available: <http://dx.doi.org/10.1109/RAMS.2011.5754515>
- [4] J. L. Leray, "Effects of atmospheric neutrons on devices, at sea level and in avionics embedded systems," Microelectronics Reliability, Vol.47 pp.1827-1835, 2007.
- [5] W. Sootkaneung, K. K. Saluja, "Soft error reduction through gate input dependent weighted sizing in combinational circuits," proc. of Int. Symposium on Quality Electronic Design, pp.1-8, 2011. [Online]. Available: <http://dx.doi.org/10.1109/ISQED.2011.5770790>
- [6] L. Xiao, W. Sheng and Z. Mao, "Soft error optimization of standard cell circuits based on gate sizing and multi-objective genetic algorithm," proc. of Design Automation Conference, pp.502-507, 2009.
- [7] J. K. Park and J. T. Kim, "A soft error mitigation technique for constrained gate-level designs," IEICE Electronics Express, Vol.5, No.18, pp.698-704, 2008.
- [8] N. M. Zivanov and D. Marculescu, "MARS-C: Modeling and Reduction of Soft Errors in Combinational Circuits," proc. of Design Automation Conference, pp.767-772, 2006.
- [9] Q. Zhou and K. Mohanram, "Cost-Effective Radiation Hardening Technique for Combinational Logic," Int. conf. on Computer Aided Design, 2004. [Online]. Available: <http://dx.doi.org/10.1109/ICCAD.2004.1382551>
- [10] H. Asadi and M. Tahoori, "Soft error hardening for logic-level designs," proc. of ISCAS, 2006. [Online]. Available: <http://dx.doi.org/10.1109/ISCAS.2006.1693540>
- [11] Y. S. Dhillon, A. U. Diril, A. Chatterjee, A. D. Singh, "Analysis and Optimization of Nanometer CMOS Circuits for Soft-Error Tolerance," IEEE Trans. on Very Large Scale Integration Systems Vol.14, No.5, pp.514-524, 2006. [Online]. Available: <http://dx.doi.org/10.1109/TVLSI.2006.876104>
- [12] W. Kai-Chiang and D. Marculescu, "soft error rate reduction using redundancy addition and removal," proc. of ASPDAC, pp.559-564, 2008. [Online]. Available: <http://dx.doi.org/10.1109/ASPAC.2008.4484014>
- [13] J. K. Park, H. S. Choi and J. T. Kim, "A Soft Error Analysis Tool for High-Speed Digital Designs," proc. of 2nd Int. conf. on Ubiquitous Information Management and Communication, pp.280-282, 2008.
- [14] R. R. Rao, K. Chopra and D. T. Blaauw, "Computing the Soft Error Rate of a Combinational Logic Circuit Using Parameterized Descriptors," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems Vol.26, No.3, 2007. [Online]. Available: <http://dx.doi.org/10.1109/TCAD.2007.891036>
- [15] B. Zhang, W. Wang and M. Orshansky, "FASER: Fast Analysis of Soft Error Susceptibility for Cell-Based Designs," 7th Int. Symposium on Quality Electronic Design, pp.755-760, 2006. [Online]. Available: <http://dx.doi.org/10.1109/ISQED.2006.64>
- [16] NANGATE, 45nm Open Cell Library 4th Release, 2009. [Online]. Available: <http://www.nangate.com>.
- [17] P. Hazucha and C. Svensson, "Impact of CMOS technology scaling on the atmospheric neutron soft error rate," IEEE Trans. on Nuclear Science, Vol.47, No.6, pp. 2586-2594, 2000. [Online]. Available: <http://dx.doi.org/10.1109/23.903813>
- [18] M. Zhang and N. R. Shanbhag, "Soft-Error-Rate-Analysis (SERA) Methodology," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol.25, No.10, pp.2140-2155, 2006. [Online]. Available: <http://dx.doi.org/10.1109/TCAD.2005.862738>
- [19] J. F. Ziegler, "Terrestrial cosmic rays," IBM J., Vol.40, No.1, pp.19-39, 1996.
- [20] 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, 2006. [Online]. Available: <http://dx.doi.org/10.1109/ISQED.2006.64>