In-Place FPGA Retiming for Mitigation of Variational Single-Event Transient Faults

Wenyao Xu, *Student Member, IEEE*, Jia Wang, *Member, IEEE*, Yu Hu, *Member, IEEE*, Ju-Yueh Lee, *Student Member, IEEE*, Fang Gong, *Student Member, IEEE*, Lei He, *Senior Member, IEEE*, and Majid Sarrafzadeh, *Fellow, IEEE*

***Abstract*—For anti-fuse or flash-memory-based field-programmable gate arrays (FPGAs), single-event transient (SET)-induced faults are significantly more pronounced than single-event upsets (SEUs). While most existing work studies SEU, this paper proposes a retiming algorithm for mitigating variational SETs (i.e., SETs with different durations and strengths). Considering the reshaping effect of an SET pulse caused by broadening and attenuation during its propagation, SET-aware retiming (*SaR*) redistributes combinational paths via postlayout retiming and minimizes the possibility that an SET pulse is latched. The *SaR* problem is formulated as an integer linear programming (ILP) problem and solved efficiently by a progressive ILP approach. In contrast to existing SET-mitigation techniques, the proposed *SaR* does not change the FPGA architecture or the layout of an FPGA application. Instead, it reconfigures the connection between a flip-flop and an LUT within a programmable logic block. Experimental results show that *SaR* increases mean-time-to-failure (MTTF) by 78% for variational SETs with a 10-min runtime limit while preserving the clock frequency on ISCAS89 benchmark circuits. To the best of our knowledge, this paper is the first in-depth study on FPGA retiming for SET mitigation.**

***Index Terms*—Field-programmable gate arrays (FPGAs), retiming, single-event transients.**

I. Introduction

AGGRESSIVE scaling of CMOS technology makes field-programmable gate arrays (FPGAs) increasingly susceptible to single-event effects. Specifically, the heavy ions in cosmic rays may cause single-event upsets (SEUs) in data latches and configuration bits. In addition, these ion strikes result in single-event transients (SETs) in both combinational logic and global clock lines, which can alter the functionality of a circuit. The effect of SET-induced faults is frequency-dependent [1]–[20], i.e., a higher frequency leads to a higher probability of an SET-induced fault. Therefore, the impact of SETs becomes increasingly pronounced for high-performance FPGA applications [21]. In this paper, we focus on SET mitigation on anti-fuse or flash-based FPGAs [22] where SEUs are less significant because the anti-fuse or flash-memory in them have virtually no SEUs for configuration bits, which is the primary soft error source in SRAM-based FPGAs [23].

In the past decade, various SEU mitigation techniques for FPGAs have been studied [23]–[34]. [35] uses retiming technology to harden the circuit for SEU as well as improve testability. However, because of the inherent difference between SEU and SET-induced faults, specific techniques for SET mitigation are needed. For example, one SET may cause multiple consequent faults, which may invalidate triple modular redundancy (TMR), one of the most popular fault-tolerant techniques for FPGAs. Most of the existing SET mitigation techniques—e.g., dual interlocked cells (DICE) [29], temporally redundant latches [36], and register hardening or selections [37], [38]—use modified latching structures to prevent the propagated SET from being latched. In addition, device and architecture cooptimization [3] has been studied. Unfortunately, the above techniques require modification of FPGA architectures and may introduce area, performance and power overhead. Moreover, strikes by ions with different energies may result in different signal pulse shapes (called "variational SETs" in this paper) and the tolerance of a wide spectrum of ion strikes therefore requires significantly more overhead. For example, to deal with variational SETs using the transistor sizing technique, nearly the 50% of gates need to be sized up, resulting in an overhead of roughly 90%, 77%, and 7% for area, power, and delay, respectively. for ASIC [39]. No in-depth techniques for dealing with variational SETs in FPGAs has been presented.

This paper proposes an SET-mitigation technique, which we refer to as "SET-aware retiming" or *SaR*, that is complementary to existing methods. Performed in the postlayout CAD stage for FPGA-based applications, our *SaR* redistributes combinational paths and minimizes the possibility of an SET-induced pulse being latched. The key enabler of *SaR* is to build a link between the following two factors: a) circuit topology and b) broadening and attenuation effects during the propagation of an SET-induced signal pulse along combinational paths. A necessary and sufficient condition is presented to characterize the SET-immune circuit topology. Based on this condition, *SaR* is formulated as an integer linear programming (ILP) problem solved efficiently by a progressive ILP approach. We also consider variational SET pulses with different initial shapes. Note that most of the existing retiming algorithms only minimize clock period

Manuscript received February 03, 2010; revised July 24, 2010 and October 19, 2010; accepted October 23, 2010. Date of publication May 19, 2011; date of current version May 27, 2011. This paper was partially funded by a UC MICRO grant sponsored by Actel. This paper was recommended by Associate Editor D. Chen.

W. Xu, J. Lee, F. Gong, L. He and M. Sarrafzadeh are with Electrical Engineering Department, University of California, Los Angeles, CA 90095 USA (e-mail: lhe@ee.ucla.edu).

J. Wang is with Electrical and Computer Engineering, Illinois Institute of Technology, IL 60298 USA.

Y. Hu is with Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6C 2V4, Canada.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2010.2094370

and [40] considers the constraints *inside* the range of the shortest and longest paths to satisfy the setup and hold time. These existing retiming algorithms cannot be applied for variational SET mitigation, however, since this problem essentially maximizes the number of paths *outside* the range defined by the shortest and longest paths.

Compared with existing SET mitigation techniques, *SaR* has a number of unique features. First, it does not require modification of the FPGA architecture since it is performed at the compilation time from an FPGA-based design. Also, *SaR* does not change the layout except reconfiguring the connection between LUTs and FFs in a programmable logic block (PLB), which leads to faster design closure. Moreover, *SaR* can handle variational SET-induced pulses and obtain the solution with the highest yield rate under these pulses. In experiments using ISCAS89 benchmarks [41] under 90 nm process technology, our proposed *SaR* increases mean-time-to-failure (MTTF) by 78% for variational SETs with a 10-min runtime limit while preserving the minimal clock period.

The remainder of this paper is organized as follows. Section II presents background and preliminaries. Section III formulates the *SaR* problem. Section IV describes the *SaR* algorithm, and Section V extends the algorithm to handle variational pulses resulting from ion strikes with different energies. Experimental results are shown in Section VI, and the paper concludes in Section VII.

II. Background and Preliminaries

A. FPGA Architecture

Our proposed *SaR* is performed after placement and routing. For simplicity of presentation, we assume that our algorithm targets an FPGA architecture similar to emerging FPGAs [22], [42], [43]. This FPGA is an array of PLBs. A PLB has a combinational cell that contains a look-up table (LUT) and a register cell that contains two multiplexes (MUXes) and one flip-flop (FF). The configuration bits in the register cell decide if a signal goes through the FF. Specifically, if the bit is 0, the signal bypasses FF; if the bit is 1, FF is included in the data path. Note that this kind of dual-put PLBs structure enables the case which requires register-output and non register-output simultaneously. Therefore, it is not necessary to change the design layout during *SaR*, and it can be applied to other similar FPGA architectures.

B. In-Place Retiming Graph for FPGAs

A sequential circuit is represented by a directed graph $\mathcal{G}(N, E)$, in which a node $n \in N$ denotes a combinational cell such as LUT, MUX or transmission gate (TG), and an edge $e(u, v) \in E$ denotes an interconnect from node u to node v. Each edge is associated with a nonnegative integer weight $w : E \to \mathcal{Z}^*$, which represents the number of FFs on the edge.

Using notations similar to Leiserson and Saxe [44], a retiming is a relocation of FFs in circuits, given by a labeling of vertices $r : V \to \mathcal{Z}$, which represents the number of FFs moved from the fanout edges of the node to its fanin edges. The number of FFs after retiming r is w_r as follows:

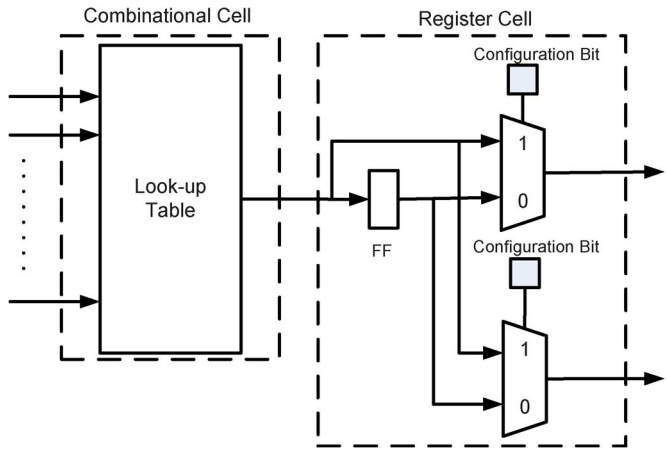

Fig. 1. Programmable logic block in FPGAs.

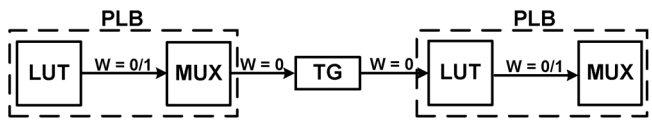

Fig. 2. Physical constraints in the FPGA-based circuit.

$$w_r(u, v) = w(u, v) + r(v) - r(u), \quad \forall (u, v) \in E. \tag{1}$$

$w_r(u, v)$ should be nonnegative, i.e.,

$$w(u, v) \geq 0, \quad \forall (u, v) \in E. \tag{2}$$

With the FPGA architecture in Fig. 1, we can model the postlayout circuit as in Fig. 2, where a transmission gate (TG) models the interconnect between PLBs and has the delay and reshaping abilities of the interconnect. There are two kinds of edges in such a retiming graph. One is the interedge between PLBs and TGs; weight w on interedges is always zero. The edge between an LUT and an MUX is called an "intra-edge", the weight w of which can be zero or one; w is determined by the value of MUX's configuration bit. Therefore, retiming a sequential circuit under such a retiming graph reassigns the configuration values of MUXes to change the weight on the intraedges without changing the placement and global routing (outside an PLB) after retiming.

Let $p(u, v)$ denote a path from node u to node v. For the simplicity of presentation, we refer to the path directly as p if u and v are irrelevant or obvious from the context. Let $w(p)$ be the number of FFs along the path p. For any pair of nodes u and v, let $W(u, v)$ be the minimum number of FFs along any path from u to v. We have

$$W_r(u, v) = W(u, v) + r(v) - r(u), \quad \forall u, v \in V. \tag{3}$$

A path $p(u, v)$ is called a *critical path* iff $w(p) = W(u, v)$. A path $p(u, v)$ is critical in the retimed circuit iff $w_r(p)$, the number of FFs along p after retiming, is equal to $W_r(u, v)$, the

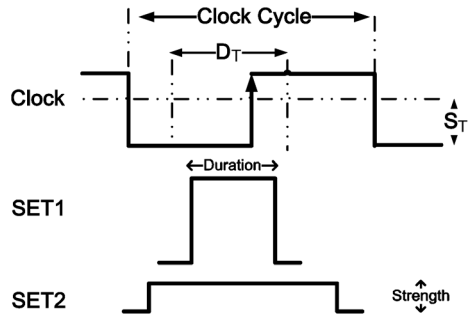

Fig. 3. Electrical and timing masking.

minimum number of FFs along any path from u to v after retiming. A critical path remains critical after retiming.

Since there is no more than one FF on any edge and there is always a nonnegative number of FFs along any path after retiming, a retiming r is *valid* iff the following constraint is satisfied:

$$0 \leq w_r(u,v) \leq 1, \quad \forall (u,v) \in E. \tag{4}$$

Also the following lemma holds:

Lemma 1: For any pair of nodes u and v, and any fanout edge (v,x) of v, the following condition holds for any valid retiming r:

$$w_r(v,x) \leq W_r(u,v) + 1.$$

C. SET Modeling

To model the effect of an SET pulse propagating through a circuit cell, each node u is associated with a pair of shaping parameters $(\delta_d(u), \delta_s(u))$. Specifically, due to the propagation-induced pulse broadening (PIPB) effect [45] and the electrical attenuation effect [46] in CMOS combinational circuits, an SET pulse λ with duration d_λ and strength s_λ is reshaped to the following duration and strength after passing gate u:

$$\begin{aligned} d'_\lambda &= d_\lambda + \delta_d(u), \\ s'_\lambda &= s_\lambda - \delta_s(u). \end{aligned} \tag{5}$$

If the duration or strength of SET is smaller than the threshold values determined by the electrical characteristics of FFs, then this pulse is not latched by any FF. For example, as shown in Fig. 3, neither SET1 nor SET2 can be latched since the duration of SET1 is too short (i.e., timing masking [47]) and the strength of SET2 is too weak (i.e., electrical masking [47]).

Suppose an SET pulse occurs at node u with initial duration $d_0(u)$ and strength $s_0(u)$. Let D_T and S_T be the minimum duration and strength for a pulse to be captured by an FF. A necessary condition for a pulse to be captured by an FF located at the fanout of node v is the existence of a path $p(u,v)$ from u to v, which has no FFs and satisfies

$$\begin{aligned} d_0(u) + \sum_{n \in p(u,v)} \delta d(n) &\geq D_T, \\ s_0(u) - \sum_{n \in p(u,v)} \delta s(n) &\geq S_T. \end{aligned} \tag{6}$$

Note that we apply the worst case analysis with respect to logic as we neglect logic masking effect [47] for SETs in combinational circuits.

III. Problem Formulation

In this section, we introduce two problem formulations based on the above FPGA architecture and SET modeling.

Definition 1: Given an SET pulse and a circuit, path $p(u,x)$, which consists of a subpath $p(u,v)$ and an edge $e(v,x)$, is called a **faulty path** iff the following conditions are satisfied.

1) $W(u,v) = 0$, i.e., there are no FFs in path $p(u,v)$.
2) Condition (6) is satisfied for path $p(u,v)$.
3) $w(v,x) > 0$, i.e., there is an FF in edge $e(v,x)$.

It is easy to verify that an SET pulse initiated at node u can be captured at an FF in the fanout of node v after its propagation along path $p(u,v)$, if the three conditions in Definition 1 are satisfied.

Definition 2: Node u is a **faulty node** iff there is a faulty path starting from node u.

Based on Definition 2, an SET pulse initiated at a faulty node can be captured by at least one FF.

Assuming that all SET pulses have the same initial dimension, the SET-aware retiming problem is formulated as follows:

Formulation 1: (Retiming for Deterministic SET): Given a retiming graph $\mathcal{G}(N,E)$, SET pulses all with the same initial duration d_0 and same strength s_0, minimum duration and strength (D_T, S_T) for a pulse to be captured by an FF, and a clock period constraint Φ, an SET-aware retiming finds a retiming which minimizes the number of faulty paths (or the number of faulty nodes) for a clock period no longer than Φ.

In reality, different SET pulses may have various dimensions due to the energy difference of the particle strikes. Therefore, the SET-aware retiming considering variational SET dimensions can be formulated as follows:

Formulation 2: (Retiming for Variational SETs): Given a retiming graph $\mathcal{G}(N,E)$, a set of SET pulses with k different initial durations, strengths and probabilities of occurrence $[(d_0^1, s_0^1, P^1), (d_0^2, s_0^2, P^2), \cdots, (d_0^k, s_0^k, P^k)]$, minimum duration and strength (D_T, S_T) for a pulse to be captured by an FF, and a clock period constraint Φ, an SET-aware retiming finds a retiming which minimizes the number of faulty paths (or the number of faulty nodes) with a clock period within Φ.

Fig. 4 shows an example to illustrate how retiming can reduce the number of faulty paths. For the simplicity of presentation, suppose each combinational block has the same $(\delta d(n), \delta s(n))$ and the same delay; also, each FF has the same (D_T, S_T). In addition, suppose SET pulses are identical on each combinational block (i.e., the same characteristics and probability of occurrence) and can be captured iff it is initiated at a node within 3 or 4 logic levels away from a reception FF, i.e., a combinational path with 3 or 4 logic levels satisfies condition (6). Initially, in Fig. 4(a), there are two faulty paths (P_1 and P_2) and two faulty nodes (N_1 and N_2). After retiming in Fig. 4(b), only one faulty path (P_3) and one faulty node (N_3) remain. Meanwhile, the clock rate remains the same after the retiming.

Considering SET-induced faults, the full-chip reliability can be measured by either the number of faulty paths or the number of faulty nodes. Therefore, we present a uniform algorithm that

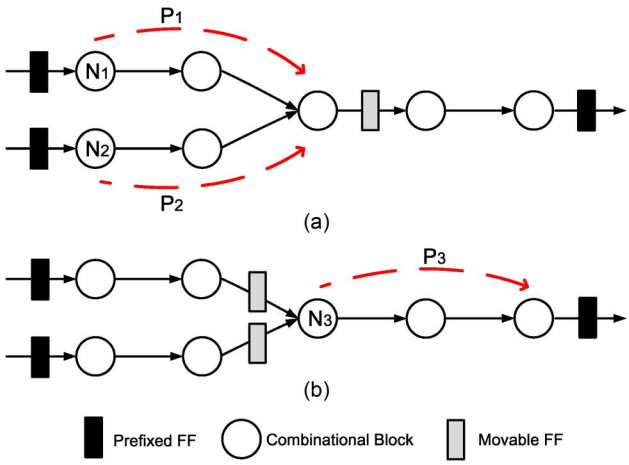

Fig. 4. An illustration for *SaR*. (a) Before retiming. (b) After retiming.

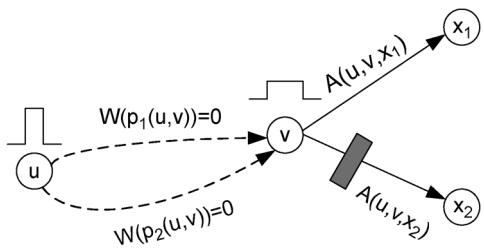

Fig. 5. Potential faulty path topology.

can be used to minimize either of these two metrics in the next section.

IV. Retiming for Deterministic SET

A. Characteristics of Faulty Paths

For a quantitative characterization, we define the *potential faulty path* as follows:

Definition 3: Path $p(u,x)$, which consists of a subpath $p(u,v)$ and an edge $e(v,x)$, is a **potential faulty path** iff subpath $p(u,v)$ is a critical path from node u to node v and it satisfies condition (6).

Fig. 5 shows an example of the topologies of potential faulty paths. Particularly, there are two critical paths from node u to node v. Suppose that they both satisfy condition (6) and there are two fanouts of node v, i.e., node x_1 and node x_2. If any of these two fanout edges, $e(v,x_1)$ or $e(v,x_2)$, contains an FF (the shadowed rectangle in Fig. 5), any SET pulses initiated at node u can be captured by the FF. In this example, path $p(u,v,x_2)$ is a faulty path but path $p(u,v,x_1)$ is not.

To ensure that a potential faulty path does not form a faulty path after retiming, we must guarantee that there are no FFs located at the last edge of this path. The following theorem states the sufficient and necessary condition that a path is not faulty:

Theorem 1: A potential faulty path $p(u,x) = p(u,v) + e(v,x)$ is not a faulty path after a valid retiming r iff

$$w_r(v,x) \le W_r(u,v). \tag{7}$$

Proof: According to (4), for a valid retiming r, $w_r(v,x)$ is either 0 or 1.

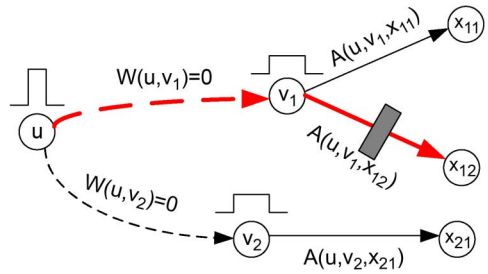

Fig. 6. Potential faulty node u.

If $p(u,x)$ is not a faulty path, then either $W_r(u,v) \ge 1$ or $w_r(v,x) = 0$. If $W_r(u,v) \ge 1$, then $W_r(u,v) \ge w_r(v,x)$ according to (4). If $w_r(v,x) = 0$, then $W_r(u,v) \ge w_r(v,x)$ since $W_r(u,v)$ is always nonnegative.

If $w_r(v,x) \le W_r(u,v)$, then $w_r(v,x) = 0$ or $W_r(u,v) \ge 1$. In either case, p is not a faulty path. ■

To model a potential faulty path $p(u,x) = p(u,v) + e(v,x)$, we introduce a binary variable, $A(u,v,x)$, where

$$w_r(v,x) - W_r(u,v) \le A(u,v,x)$$
$$A(u,v,x) \in \{0,1\}. \tag{8}$$

The physical meaning of the variable $A(u,v,x)$ can be interpreted as follows. If $A(u,v,x) = 0$, condition (7) is satisfied and therefore no FFs are inserted in edge $e(v,x)$, which indicates that path $p(u,x)$ is not a faulty path after retiming r. If $A(u,v,x) = 1$, there may be an FF inserted in edge $e(v,x)$ of a potential faulty path $p(u,x)$ because $w_r(v,x)$ could be 1. If we minimize the sum of all $A(u,v,x)$ over all potential critical paths, $A(u,v,x) = 1$ indicates that there must be an FF in edge $e(v,x)$, which can be proved by contradiction.

B. Characteristics of Faulty Nodes

Similarly, we define a faulty node below.

Definition 4: Node u is a **potential faulty node** iff there is at least one potential faulty path starting from node u.

Consider the example shown in Fig. 6. If any one of paths $p(u,x_{11})$, $p(u,x_{12})$ or $p(u,x_{21})$ is a potential faulty path, node u is a potential faulty node. Using the auxiliary variable $A(u,v,x)$, we can easily model the faulty potential of node u as follows:

$$B(u) = \bigvee_{\forall v, p(u,v) \in \mathrm{PFP}(u,v,x)} \; \bigvee_{\forall x, e(v,x) \in E} A(u,v,x), \tag{9}$$

where $p(u,v) \in \mathrm{PFP}(u,v,x)$ means path $p(u,x) = p(u,v) + e(v,x)$ is a potential faulty path. Intuitively, $B(u) = 1$ (i.e., node u is a faulty node after retiming) iff there is at least one edge $e(v,x)$ in a potential faulty path $p(u,x)$ which contains an FF.

C. Retiming for Faulty Path Minimization

To minimize the number of faulty paths by retiming, all potential faulty paths need to be investigated. In Fig. 5, there are four potential faulty paths, $p1(u,v) + e(v,x_1)$, $p2(u,v) + e(v,x_1)$, $p1(u,v) + e(v,x_2)$, and $p2(u,v) + e(v,x_2)$. In general, suppose there are $|p(u,v)|$ critical paths from node u to node v and there are $|e(v,x)|$ edges in the fanout of node

v, then there are $O(|N|^2 \cdot |p(u,v)| \cdot |e(v,x)|)$ potential faulty paths, where $|N|$ is the number of nodes in the graph.

With the help of the auxiliary variable $A(u,v,x)$, the number of variables to characterize the potential faulty paths can be reduced to $O(|N|^2 \cdot |e(v,x)|)$ by using a weight $f(u,v,x)$ in front of each $A(u,v,x)$ to describe the number of critical paths from node u to node v. For example, in Fig. 5, we have $f(u,v,x_1) = 2$ and $f(u,v,x_2) = 2$ for $A(u,v,x_1)$ and $A(u,v,x_2)$, respectively.

The minimization of the number of faulty paths is equivalent to the minimization of the sum of $A(u,x)$ over all potential faulty paths. Therefore, the overall SET-aware retiming problem can be formulated as the following ILP problem:

$$\begin{aligned} &P_{\text{path}}(S): \\ &\text{Minimize} \quad \sum_{(u,v,x)\in S} f(u,v,x)\cdot A(u,v,x) \\ &\quad s.t. \quad 0 \le w_r(u,v) \le 1, \forall (u,v) \in E, \\ &\qquad w_r(v,x) - W_r(u,v) \le A(u,v,x), \forall (u,v,x) \in S, \\ &\qquad A(u,v,x) \ge 0, \forall (u,v,x) \in S, \\ &\qquad r(u) \in \mathcal{Z}, \forall u \in V, \end{aligned} \tag{10}$$

where S is the set of all potential faulty paths represented by the tuples (u,v,x) and r is the retiming decision variable. Note that we relax the 0-1 integer constraint of variable $A(u,v,x)$ in ILP (10) because the binarity of $A(u,v,x)$ is automatically preserved as $r(u), w_r(v,x)$. Therefore, $W_r(u,v)$ are all integers, and the objective function is to minimize a positive weighted sum of $A(u,v,x)$.

D. Retiming for Faulty Node Minimization

Using techniques similar to those presented in the previous subsection, we can minimize the number of faulty nodes by the following ILP formulation:

$$\begin{aligned} &P_{\text{node}}(S): \\ &\text{Minimize} \quad \sum_{u\in V} B(u) \\ &\quad s.t. \quad A(u,v,x) \le B(u), \\ &\qquad \forall v, p(u,v) \text{ is a critical path, } \forall x, e(v,x) \in E, \\ &\qquad \text{All other constraints in } P_{\text{path}}(S). \end{aligned} \tag{11}$$

Note that the first constraint in Formulation P_{node} is a linear form of (9). Again, the binarity of the 0-1 integer variable $B(u)$ is preserved by the objective function and the binarity of $A(u,v,x)$.

E. Speedup by Progressive ILP

In Formulation $P_{\text{path}}(S)$ in Section IV-C, the number of variables is dominated by the number of $A(u,v,x)$ variables, with complexity $O(|N|^2 \cdot |p(u,v)| \cdot |e(v,x)|)$ where $|p(u,v)|$ is the number of critical paths between two nodes. In the worst case scenario, this number will be exponential to the number of edges in the graph, and $|e(v,x)|$ is the number of fanouts of a node. The number of constraints in $P_{\text{path}}(S)$ is in the same order as the number of variables because the second constraint is dominant. Formulation $P_{\text{node}}(S)$ in Section IV-D has additional constraints for increased complexity. Therefore, a straightforward solution to the ILP-based formulation presented in $P_{\text{path}}(S)$ and $P_{\text{node}}(S)$ might be prohibitively expensive. Below, we present a progressive retiming with improved scalability while preserving optimality. The proposed progressive retiming is a uniform approach which deals with both Formulation $P_{\text{path}}(S)$ and Formulation $P_{\text{node}}(S)$. We use $P(S)$ to represent any of these two formulations.

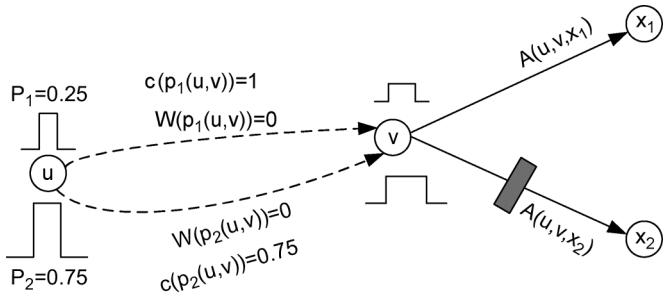

Fig. 7. Potential faulty paths under variational SETs.

The complexity in Formulation $P(S)$ is mainly caused by the enormous size of set S. We propose a progressive retiming using progressive ILP to solve the above problem. Specifically, we use a series of sets $S_0 \subset S_1 \subset \cdots \subset S$ to approximate S and generate a series of retiming $r_1, r_2, \cdots$. The set S_k is obtained by adding to S_{k-1} all tuples (u,v,x) induced by the faulty paths after solving retiming r_{k-1}. Then, retiming r_k is obtained by solving $P(S_k)$. Initially, one can choose $S_0 = \emptyset$ and r_1 to be any retiming. The algorithm converges based on the following theorem:

Theorem 2: There exists a finite m such that $S_{m+1} = S_m$. For such m, r_m is the optimal solution of $P(S)$.

Proof: We first prove that such m exists by contradiction. Assume such m does not exist. Then, since $S_k \subset S_{k+1}$ for any k, it is straightforward that $|S_{k+1}| \ge |S_k| + 1$, which implies $|S_k| \ge k$. Since $S_k \subset S$, we have $|S| \ge |S_k| \ge k$ for any k. However, that is not possible since S is a finite set.

Now, suppose $S_{m+1} = S_m$. Then there is no new faulty path induction after retiming r_m, i.e., r_m is a feasible solution of $P(S)$. Therefore, since r_m is the optimal solution of $P(S_m)$ and $S_m \subset S$, r_m must be the optimal solution of $P(S)$. ■

Note that each iteration in the progressive retiming only takes the constraints from faulty paths generated by all the previous retiming procedures, instead of those from all the potential faulty paths. For example, S_1 for Fig. 5 contains only one tuple, i.e., $S_1 = \{(u,v,x_2)\}$, since there is only one tuple forming the faulty path $p(u,v,x_2)$ under the current retiming. On the other hand, $S = \{(u,v,x_1),(u,v,x_2)\}$ since there are two tuples forming potential faulty paths.

In the first few iterations of the progressive retiming, the number of constraints in $P(S_k)$ can be much fewer than that in $P(S)$. During the course of the iterations, the number of constraints accumulates and $P(S_k)$ becomes slower. At the end of the progressive retiming, the number of constraints in $P(S_K)$ is no more than that in $P(S)$ because topology constraints enable that certain potential faulty paths will never become a faulty

TABLE I
CHARACTERISTICS OF SETS

Name	Duration	Strength	Probability
SET_1	700ps	1.1V	50%
SET_2	500ps	1.1V	50%

TABLE II
RESHAPING ABILITIES AND LATCHING WINDOW

Reshaping		
combinational cell	$\delta d(n)$	$\delta s(n)$
LUT	143ps	0.09V
MUX	28ps	0.010V
Transmission gate	12ps	0.002V
Latching window		
	D_T	S_T
FF	1.1ns	0.5V

path. Our experimental results show that progressive retiming can usually converge in very few iterations. Therefore, we can terminate the progressive retiming within a limited number of iterations.

V. RETIMING FOR VARIATIONAL SETS

In this section, we extend the proposed retiming to consider variational SETs as defined in Formulation 2. For any SET pulse λ_i with initial duration d_0^i, strength s_0^i and probability of the occurrence P^i, a path $p(u,x) = p(u,v) + e(v,x)$ is a potential faulty path iff $p(u,v)$ is a critical path and the below (12) (based on (6)) is satisfied:

$$\begin{aligned} D_T - \sum_{n \in p(u,v)} \delta d(n) &\le d_0^i(u) \\ S_T + \sum_{n \in p(u,v)} \delta s(n) &\le s_0^i(u). \end{aligned} \tag{12}$$

Considering the occurrence of variational pulses defined in Formulation 2, we define the *weighted potential faulty path* as follows:

Definition 5: Given path $p(u,x) = p(u,v) + e(v,x)$ and k different pulses λ_i, $i \in [1,k]$, $p(u,x)$ is a weighted potential faulty path iff $p(u,x)$ is a potential faulty path for at least one pulse λ_i and the weight of this path is

$$c\left(p(u,x)\right) = \sum_{\lambda_j \in J} P^j, \tag{13}$$

where J is the set of pulses under which path $p(u,x)$ is a potential faulty path.

Consider an example shown in Fig. 7. Suppose there are two pulses, λ_1 and λ_2, with occurrence probability $P_1 = 0.25$ and $P_2 = 0.75$, respectively. Additionally, path $p_1(u,v)$ satisfies (12) for both λ_1 and λ_2, and $p_2(u,v)$ satisfies (12) only for λ_2, i.e., λ_1 can pass both paths while λ_2 will be eliminated after path p_2. Therefore, we have $c(p_1) = P_1 + P_2 = 1.0$ and $c(p_2) = P_2$.

Using the concept of the weighted potential faulty path, we can minimize the number of faulty paths for variational SET pulses with the following ILP formulation for the same constraints in $P_{\text{path}}(S)$ by changing the objective function as follows:

$$\begin{aligned} \text{Minimize} \quad & \sum_{(u,v,x) \in S} \sum_{p(u,v)} c(p) \cdot A(u,v,x), \\ s.t. \quad & \text{All constraints from } P_{\text{path}}(S) \end{aligned}$$

where S is the set of weighted potential faulty paths.

To handle the faulty node minimization problem, we define the *weighted potential faulty node* as follows.

Definition 6: Node u is a weighted potential faulty node iff there is a weighted potential faulty path starting from node u. The weight of this node is

$$c(n) = \sum_{\lambda_j \in J} P^j \cdot B(u, \lambda_i), \tag{14}$$

where J is the set of pulses under which node u is a potential faulty node and $B(u,\lambda_i)$ is the faulty potential of node u under SET pulse λ_i which can be calculated as

$$B(u,\lambda_i) = \bigvee_{\forall v, p(u,v) \in \text{PFP}(u,v,x,\lambda_i)} \bigvee_{\forall x, e(v,x) \in E} A(u,v,x).$$

Using the concept of the weighted potential faulty node, we can minimize the number of faulty paths for variational SET pulses with the following ILP formulation for the the same constraints in $P_{\text{path}}(S)$ by changing the objective function as follows:

$$\begin{aligned} \text{Minimize} \quad & \sum_{u \in V} c(u), \\ s.t. \quad & \text{All constraints from } P_{\text{node}}(S) \end{aligned}$$

where S is the set of weighted potential faulty paths. The progressive retiming presented in Section IV-E can be extended in a straightforward manner to solve the two formulations for variational SET pulses.

VI. EXPERIMENTAL RESULTS

A. Experimental Settings

We have implemented the proposed retiming algorithm, *SaR*, for fault path minimization using C++ and solved the ILP using mosek [48]. All experimental results are collected in a Linux server with an Intel Xeon 3.2 GHz CPU and 2 GB memory. The algorithms are tested using ISCAS89 benchmarks.[1] Each benchmark is first mapped to 4-LUTs using the Berkeley ABC synthesis toolset [49]. After that, placement and routing are performed using VPR [50]. In our current experiment, we target the nonclustered FPGA architecture, which is similar to the FPGA mentioned in Section II. For the simplicity of presentation, we assume in this paper that all cells of the same type have the same reshaping abilities and delay.[2]

According to [51], we consider the two most possible SETs with duration of 500 ps and 700 ps, respectively, in 90 nm process technology, and we summarize their characteristics in

[1]We only list 15 largest ISCAS89 benchmarks in the paper, and more results can refer to our technology report.

[2]It is not difficult to see that our proposed algorithm can also handle the case without this assumption.

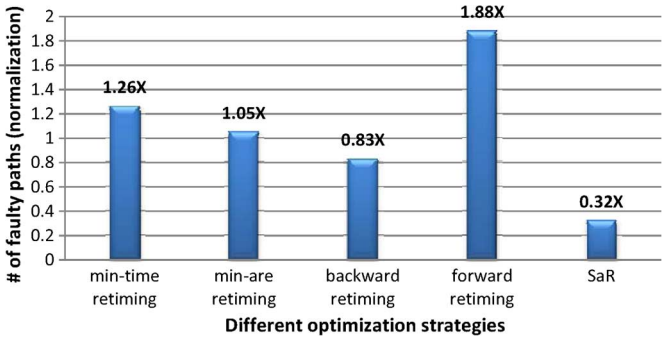

Fig. 8. Case study for *SaR* effectiveness: s444.

Table I. We have tested our proposed *SaR* under both deterministic and variational SETs. SET_1 is assumed in the experiments considering deterministic SET. Both SET_1 and SET_2 are assumed in the experiments considering variational SETs, and we assume that SET_1 and SET_2 have the same probabilities of occurrence. We use the PTM 90 nm model [52] to simulate the characteristics of the SET propagation in our FPGA architecture. We find that the reshaping ability $(\delta d(n), \delta s(n))$ of a given type of cells is practically independent of the duration and the strength of the pulses. Note that the reshaping effect is dominated by the LUT cell according to Table II, and the latching window information of FFs in 90 nm process technology[3] is also shown in Table II.

B. Solution Space Exploration

Retiming is a basic sequential circuit transformation. As stated in Section I, there are some existing retiming algorithms, such as retime for min-time, min-are, most forward, and most backward [21]. To explore the potential solution space for SET mitigation, we compare the performance among different retiming strategies. Fig. 8 illustrates faulty path reduction after min-time, min-area, forward, and backward for benchmark "s444." Similar observations also happen to other benchmarks. The results clearly show a large optimization room (around 68%) for SET mitigation, where backward retiming seems a good heuristic to reduce the faulty-path number. However, there is still a large gap (more than 2X) between backward retimed circuit and the optimal one (circuit after *SaR*) in terms of SET mitigation.

C. Runtime and Quality Trade-Off

To minimize the number of faulty paths, Section IV has presented two approaches, i.e., the global retiming presented in Section IV-C and the progressive retiming presented in Section IV-E. Both return optimal solutions but with different runtimes. This subsection experimentally compares these two approaches in terms of runtime. In addition, we show how to trade solution quality for runtime in order to deal with large benchmarks.

Table III compares the runtime of the global retiming and progressive retiming for six benchmarks, where "-" means the ILP solver mosek could not finish within 60 h. For the first two small benchmark circuits, the progressive retiming is significantly (up to 10 000 times) faster than the global one—the latter cannot finish most of the benchmark circuits. Furthermore, progressive retiming reduces the problem magnitude drastically (27 times on average). This observation confirms and approves the runtime efficiency brought by progressive ILP solving.

[3]Typically, D_T in the latching window of FFs is equivalent to $T_{\text{setup}}+T_{\text{hold}}$; S_T in the latching window of FFs is equivalent to the threshold voltage.

TABLE III
RUNTIME AND PROBLEM MAGNITUDE

	Global retiming		Progressive retiming	
ckt	runtime	# of ILP constraints	runtime	# of ILP constraints
s27	1min2s	354	1s	54
s208	5h15min12s	9038	2s	112
s298	-	13854	9min24s	459
s386	-	18495	8s	493
s444	-	29485	24min13s	1234
Geo Mean		7528		279
		1		0.037X

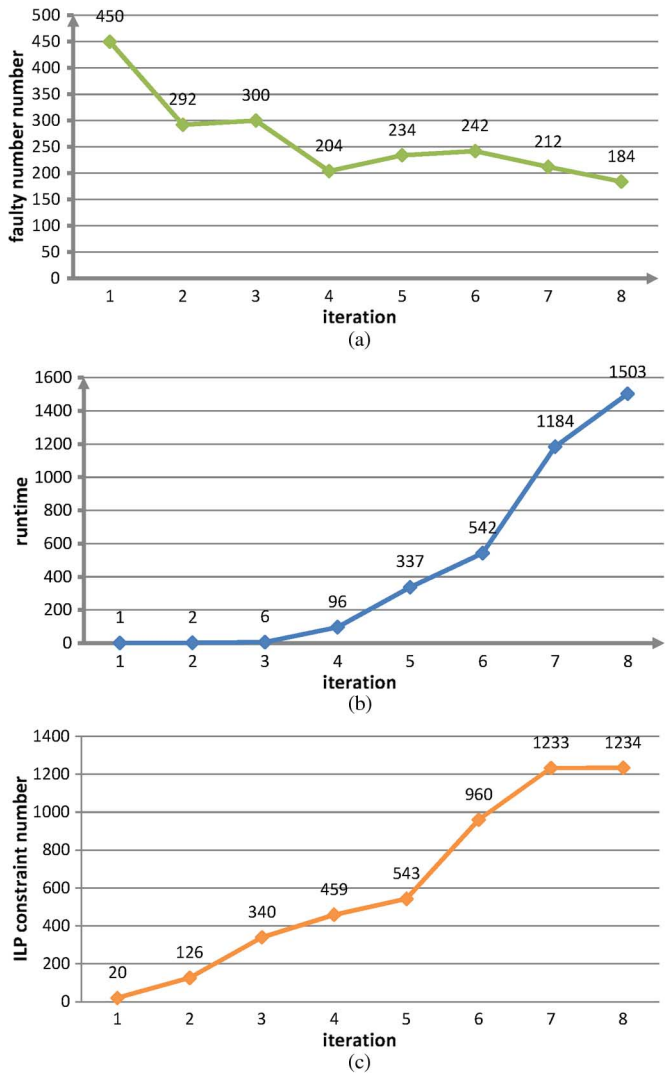

Fig. 9. Case study for optimization convergence: s444.bench. (a) FP number with iterations. (b) Runtime with iterations. (c) ILP constraint number with iterations.

We also study the convergence of the proposed progressive retiming. As an example, Fig. 9 plots the remaining faulty paths number, runtime and ILP constraint number during the course of the progressive retiming for benchmark "s444." Clearly, it takes significantly longer to achieve a reduction of a few faulty

TABLE IV
SaR RESULTS WITH THE MINIMUM CLOCK PERIOD CONSTRAINT

ckt	# of LUTs	# of FFs	Deterministic SET			Variational SETs		
			# of original faulty paths	# of remaining faulty paths		# of original faulty paths	# of remaining faulty paths	
				10m timeout	30m timeout		10m timeout	30m timeout
s526	106	21	1137	415	393	1364	732	571
s641	201	19	2996	2673	1992	3595	3312	2770
s713	199	19	3012	2721	2453	3614	3297	2691
s820	209	5	2456	921	921	2425	1018	1018
s832	234	5	3758	968	968	4032	1228	928
s838	201	32	2965	1711	1345	3547	1433	1298
s953	345	29	3459	1847	1456	3978	1657	1303
s1196	378	18	3867	1784	1566	4239	1866	1576
s1238	403	18	3758	1345	1195	4532	2049	1758
s1423	678	74	5743	3475	2890	6954	2345	2103
s1488	790	6	6758	3425	2456	7685	4322	3049
s1494	781	6	6452	2895	2869	7748	3566	2857
s5378	2301	179	12394	7687	5382	14456	8696	6821
s9234	1892	211	11385	5601	4866	13592	8325	6983
s35932	18587	1728	65768	38602	29463	86749	46294	40586
Faulty Path (Geomean)			3118	1604	1349	3673	2039	1708
			1	0.51X	0.43X	1	0.56X	0.47X
MTTF			1	1.96X	2.31X	1	1.78X	2.15X

TABLE V
FAULTY PATH REDUCTION WITH DIFFERENT TIMING CONSTRAINTS (W.R.T. CLOCK PERIOD Φ)

ckt	Deterministic SET				Variational SETs			
	# of original faulty paths	# of remaining faulty paths			# of original faulty paths	# of remaining faulty paths		
		Φ	1.2Φ	1.5Φ		Φ	1.2Φ	1.5Φ
s526	1137	415	382	305	1364	732	642	578
s641	2996	2673	2291	1983	3595	3312	3024	2817
s713	3012	2721	2475	2018	3614	3297	2859	2811
s820	2456	921	850	712	2425	1018	945	811
s832	3758	968	832	754	4032	1228	1023	828
s838	2965	1711	1458	1222	3547	1433	1277	1109
s953	3459	1847	1573	1323	3978	1657	1476	1298
s1196	3867	1784	1601	1601	4239	1866	1642	1542
s1238	3758	1345	1088	972	4532	2049	1828	1744
s1423	5743	3475	2410	2410	6954	2345	2011	2011
s1488	6758	3425	3005	2784	7685	4322	4017	3812
s1494	6452	2895	2334	2023	7748	3566	2958	2766
s5378	12394	7687	7302	6219	14456	8696	8077	7823
s9234	11385	5601	5123	4801	13592	8325	8022	7812
s35932	65768	38602	34321	32945	86749	46294	43422	41237
Faulty path (Geomean)	3118	1604	1354	1149	3673	2039	1806	1650
	1	0.51X	0.43X	0.37X	1	0.56X	0.49X	0.45X
MTTF	1	1.96X	1.93X	1.80X	1	1.78X	1.70X	1.48X

paths after the first four iterations due to the enlargement of problem magnitude. Similar observations are made across all benchmarks. To achieve the best tradeoff between the runtime and quality, we use four iterations for the progressive retiming in the rest of the experiments. To further control the runtime, we set the timeout (i.e., time limit) for mosek, the ILP solver. Two sets of timeouts (10 min and 30 min) are used. When the total runtime consumed by mosek exceeds the timeout, the current best feasible solution obtained by mosek[4] will be used as the final retiming solution.

D. Faulty Path Reduction for Deterministic SET

Table IV compares the number of faulty paths before ("original") and after ("remaining") optimization. With the minimum clock period constraint, our proposed algorithm *SaR* can reduce the number of faulty paths by 49% (57%) with a 10-min (30-min) timeout. Meanwhile, the estimated mean-time-to-failure (MTTF) is proportional to the area (A) and clock frequency (f) and inversely proportional to the fault rate, which in turn is proportional to the number of faulty paths (N_{FP}). We get as follows:

$$\mathrm{MTTF} \propto \frac{A * f}{N_{\mathrm{FP}}}. \tag{15}$$

Since our retiming preserves the area (in-place retiming) and clock frequency (min-time retiming), MTTF is approximately inversely proportional to the number of faulty paths. The proposed retiming is able to improve MTTF for the minimum clock period by 96% (131%) with a 10-min (30-min) timeout on average.

E. Faulty Path Reduction for Variational SETs

In this experiment, we used two kinds of SET pulses with the characteristics in Table I. As shown in Table IV, our proposed

[4]Mosek uses an iterative algorithm (the interior point method) for the ILP solving, and therefore can produce multiple intermediate (suboptimal) solutions during the iterations.

SaR algorithm effectively deals with the variational SET mitigation problem and improves MTTF by 78% (115%) for the minimum clock period constraint with a 10-min (30-min) timeout. Compared to the case with deterministic SET, faulty path reduction via retiming for variational SETs slightly decreases caused by the increase of the number of ILP formulation constraints.

F. Reliability Versus Performance

The above experiments have already proved the effectiveness and efficiency of *SaR* for SET mitigation. Moreover, we also investigated the tradeoff between reliability (MFFT) and performance (clock frequency). Table V shows faulty path reduction with various relaxation of the timing constraints, i.e., allowing 0.2X and 0.5X minimal clock period increase after *SaR* respectively. Intuitively, the retiming with a relaxed clock constraint searches larger solution space and reduces more faulty paths, which is confirmed by Table V. It illustrates that about 70% of more faulty path reduction is obtained when 50% delay increase is allowed. Nevertheless, delay will decease clock frequency, to which MTTF is proportional according to (15). Our experimental results show that the increasing of system delay overwhelms the faulty path reduction and makes MTTF worse ultimately.

VII. Conclusion and Future Work

We have presented *SaR*, an SET-aware retiming. The broadening and attenuation effects during the propagation of an SET-induced signal are modeled and linked to the topology (i.e., the distribution of combinational paths) of a circuit. Based on this model, the retiming problem considering variational SETs (i.e., SETs with different durations) is formulated as an integer linear programming (ILP) problem and solved by a progressive ILP algorithm. Tested on ISCAS89 benchmarks, our *SaR* improves MTTF by 78% with a 10-min time limit for variational SETs without performance and area penalty. In the future, we will take logic masking [47] into consideration for less conservative fault models.

References

[1] B. Narasimham *et al.*, "On-chip characterizaion of single-event transient pulsewidths," *IEEE Trans. Device Mater. Rel.*, vol. 6, no. 3, pp. 542–549, 2006.

[2] L. Cheng, X. Song, G. Yang, Z. Tang, and S. Gao, "A fast congestion estimator for routing with bounded detours," in *Proc. Asia South Pacific Design Automation Conf.*, Yokohama, Japan, 2004, pp. 666–700.

[3] Y. Lin and L. He, "Device and architecture concurrent optimization for FPGA transient soft error rate," in *Proc. IEEE/ACM Int. Conf. Comput.-Aid. Design*, San Jose, CA, 2007, pp. 194–198.

[4] K. Xu, W. Xu, J. Shen, and X. Xu, "Task scheduling algorithm based on dual-Vdd dynamic reconfigurable FPGA," *J. Zhejiang Univ.*, vol. 2, no. 1, pp. 300–304, 2010.

[5] L. Cheng, W. N. N. Hung, G. Yang, and X. Song, "Congestion estimation for 3-D circuit architectures," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 51, no. 2, pp. 655–659, 2004.

[6] S. Baeg, S. Wen, and W. Rong, "Minimizing soft errors in tcam devices: A probabilistic approach to determining scrubbing intervals," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 44, no. 7, pp. 814–822, 2010.

[7] W. Hung, X. Song, T. Kam, L. Cheng, and G. Yang, "Routability checking for three-dimensional architectures," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 1, pp. 1371–1374, 2004.

[8] N. Guan, Z. Gu, Q. Deng, W. Xu, and G. Yu, "Schedulability analysis of preemptive and nonpreemptive EDF on partial runtime-reconfigurable FPGAs," *ACM Trans. Design Autom. Electron. Syst.*, vol. 13, no. 6, pp. 745–759, 2008.

[9] L. Cheng and P. Gupta, "A levelized variation modeling scheme," in *Proc. IEEE Workshop Design Manufacturability Yield*, Los Angeles, CA, 2010, pp. 13–19.

[10] F. He, M. Gu, X. Song, Z. Tang, and L. Cheng, "Probabilistic estimation for routing space," *Comput. J.*, vol. 10, no. 4, pp. 667–676, 2005.

[11] D. Bhaduri, S. K. Shukla, P. S. Graham, and M. B. Gokhale, "Reliability analysis of large circuits using scalable techniques and tools," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 4, no. 7, pp. 2247–2460, 2007.

[12] Y. Jeng and L. Cheng, "Digital spectrum of a nonuniformly sampled two-dimensional signal and its reconstruction," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 3, pp. 1180–1187, 2005.

[13] F. Ren and D. Markovic, "True energy-performance analysis of the MTJ-based logic-in-memory architecture (1-bit full adder)," *IEEE Trans. Electron Devices*, vol. 57, no. 5, pp. 1023–1028, 2010.

[14] F. He, L. Cheng, G. Yang, X. Song, M. Gu, and J. Sun, "On theoretical upper bounds for routing estimation," *J. Universal Comput. Sci.*, vol. 11, no. 5, pp. 117–122, 2005.

[15] C. Windstead and S. Howard, "A probabilistic LDPC-coded fault compensation technique for reliable nanoscale computing," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 7, pp. 484–488, 2009.

[16] F. He, X. Song, M. Gu, L. Cheng, G. Yang, Z. Tang, and J. Sun, "A combinatorial congestion estimation approach with generalized detours," *Comput. Math. Appl.*, vol. 51, no. 6, pp. 232–241, 2006.

[17] F. He, X. Song, M. Gu, L. Cheng, G. Yang, Z. Tang, and J. Sun, "A combinatorial congestion estimation approach with generalized detours," *J. Circuits, Syst., Comput.*, vol. 51, no. 3, pp. 1113–1126, 2010.

[18] B. Vasic and S. K. Chilappagari, "An information theoretical framework for analysis and design of nanoscale fault-tolerant memories based on low-density parity-check codes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 3, no. 7, pp. 2438–2446, 2007.

[19] T. B. Chan, A. Pant, L. Cheng, and P. Gupta, "Design dependent process monitoring for back-end manufacturing cost reduction," in *Proc. Int. Conf. Comput. Aided Design*, San Jose, CA, 2010, pp. 116–122.

[20] W. Robinett *et al.*, "Defect tolerance based on coding and series replication in transistor-logic demultiplexer circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 3, pp. 2410–2421, 2007.

[21] L. Rockett *et al.*, "Radiation hardened FPGA technology for space applications," in *Proc. Aerospace Conf.*, Chicago, IL, 2007, pp. 1–7.

[22] [Online]. Available: http://www.actel.com/

[23] Y. Hu, Z. Feng, L. He, and R. Majumdar, "Robust FPGA resynthesis based on fault-tolerant Boolean matching," in *Proc. ICCAD*, San Jose, CA, Nov. 2008, pp. 706–713.

[24] Z. Cao, T. Jing, Y. Hu, Y. shi, X. Hong, X. Hu, and G. Yan, "DraXRouter: Global routing in X-architecture with dynamic resource assignment," in *Proc. Asia South Pacific Design Automation Conf.*, Yokohama, Japan, 2006, pp. 618–623.

[25] W. Xu, K. Xu, and X. Xu, "A novel placement algorithm for symmetrical FPGAs," in *Proc. Int. Conf. ASIC*, Guilin, China, 2007, pp. 1281–1284.

[26] L. Cheng, W. N. N. Hung, G. Yang, and X. Song, "Congestion estimation for 3D routing," in *Proc. Int. Symp. VLSI*, San Diego, CA, 2004, pp. 239–240.

[27] K. Morgan *et al.*, "SEU-induced persistent error propagation in FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 6, pp. 2438–2445, 2006.

[28] M. Rofouei, W. Xu, and M. Sarrafzadeh, "Computing with uncertainty in a smart textile surface for object recognition," in *Proc. Int. Conf. \Multisensor Fusion Integr. Intell. Syst.*, Salt Lake City, UT, 2010, pp. 174–179.

[29] C. Bolchini, D. Quarta, and M. D. Santambrogio, "SEU mitigation for SRAM-based FPGAs through dynamic partial reconfiguration," in *Proc. GLSVLSI*, New York, Nov. 2007, pp. 512–519.

[30] M. Gu, F. He, L. Cheng, X. Song, and G. Yang, "Congestion estimation for hexagonal routing," *Int. J. Comput. Math.*, vol. 16, no. 2, pp. 323–331, 2006.

[31] L. Cheng, X. Song, G. Yang, W. N. N. Hung, and Z. Tang, "A fast congestion estimator for routing with bounded detours," *Integr., VLSI J.*, vol. 60, no. 11, pp. 2562–2569, 2008.

[32] J. Xiong, Y. Shi, V. Zolotov, and C. Visweswariah, "Statistical multilayer process space coverage for at-speed test," in *Proc. Design Autom. Conf*, San Jose, CA, 2009, pp. 117–122.

[33] F. He, X. Song, L. Cheng, G. Yang, Z. Tang, M. Gu, and J. Sun, "A hierarchical method for wiring congestion prediction," in *Proc. Int. Symp. VLSI*, Salt Lake City, UT, 2005, pp. 16–25.
[34] Z. Feng, Y. Hu, L. He, and R. Majumdar, "IPR: In-place reconfiguration for FPGA fault tolerance," in *Proc. ICCAD*, San Jose, CA, 2009, pp. 242–247.
[35] S. Krishnaswamy, I. L. Markov, and J. P. Hayes, "Improving testability and soft-error resilience through retiming," in *Proc. DAC*, San Jose, CA, 2009, pp. 60–65.
[36] D. G. Mavis and P. H. Eaton, "Temporally redundant latch for preventing single event disruptions in sequential integrated circuits," in *Proc. Int. Microelectron. Conf.*, Albuquerque, NM, 2002, pp. 187–200.
[37] Using Synplify to design in Actel radiation-hardened FPGAs [Online]. Available: www.actel.com/documents/SynplifyRH_AN.pdf Technology Report
[38] R. R. Rao *et al.*, "Soft error reduction in combinational logic using gating resizing and flipflop selection," in *Proc. ICCAD*, San Jose, CA, 2006, pp. 18–25.
[39] Q. Zhou and M. Kartik, "Gate sizing to radiation harden combinational logic," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, pp. 155–166, Jan. 2006.
[40] C. Lin and H. Zhou, "An efficient retiming algorithm under setup and hold constraints," in *Proc. DAC*, San Francisco, CA, 2006, pp. 72–77.
[41] [Online]. Available: http://www.ece.vt.edu/mhsiao/iscas89.html
[42] [Online]. Available: http://www.xilinx.com/
[43] [Online]. Available: http://www.altera.com/
[44] C. E. Leiserson and J. B. Saxe, "Retiming synthronous circuitry," *Algorithmica*, vol. 10, no. 2, pp. 5–35, 1988.
[45] V. F. Carrois *et al.*, "Investigation of the propagation induced pulse broadening (PIPB) effect on single event transients in SOI and bulk inverter chains," *IEEE Trans. Nucl. Sci.*, vol. 55, pp. 2842–2853, Dec. 2008.
[46] P. E. Dodd and L. W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," *IEEE Trans. Nucl. Sci.*, vol. 50, pp. 583–602, Jun. 2003.
[47] K. J. Hass, J. W. Gambles, B. Walker, and M. Zampaglione, "Mitigating single event upsets from combinational logic," in *Proc. NASA Symp. VLSI Design*, Chicago, IL, 1998, pp. 10–22.
[48] Mosek Optimization Software [Online]. Available: http://www.mosek.com/
[49] B. L. Synthesis and V. Group, ABC—A system for sequential synthesis and verification [Online]. Available: http://www.eecs.berkeley.edu/alanmi/abc/
[50] V. Betz, J. Rose, and Alexander, Versatile placement and routing tools for FPGAs [Online]. Available: http://www.eecg.utoronto.ca/vpr/
[51] B. Narasimham, Single-event transient pulse-width measurements in advanced technologies Institute for Space and Defense Electronics, Vanderbilt Univ., 2008, Tech. Tep..
[52] Predictive Technology Model (PTM) [Online]. Available: http://www.eas.asu.edu/ptm/

Wenyao Xu (S'08) received the B.S. and M.S. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2006 and 2008, respectively. He is currently working toward the Ph.D. degree in the Electrical Engineering Department, University of California, Los Angeles.

His research interests include computer-aided design for programmable fabrics, wireless health, and brain–computer interface.

Jia Wang (M'08) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2002 and the Ph.D. degree in computer engineering from Northwestern University, Evanston, IL, in 2008.

He is an Assistant Professor of electrical and computer engineering at Illinois Institute of Technology, Chicago. His research interests are in computer-aided design of very large scale integrated circuits and algorithm design.

Yu Hu (M'10) received the B.Eng. and M.Eng. degrees from the Computer Science and Technology Department, Tsinghua University, Beijing, China, in 2002 and 2005, respectively, and the Ph.D. degree from the Department of Electrical Engineering, University of California, Los Angeles (UCLA) in 2009.

Since 2010, he has been an Assistant Professor in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. His research interests include general aspects of field-programmable gate arrays (FPGAs).

Dr. Hu is the recipient of the Outstanding Graduate Student Award from Tsinghua University in 2005, and he is the corecipient of the Best Contribution Award at International Workshop of Logic and Synthesis 2008. His work has been nominated for the Best Paper Award multiple times at the International Conference on Computer-Aided Design and Design Automation Conference.

Ju-Yueh Lee (S'08) received the B.S. degree in computer science from National Chiao-Tung University, Hsinchu, Taiwan, in 2004 and the M.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2006. He is currently working toward the Ph.D. degree in the Electrical Engineering Department, University of California, Los Angeles (UCLA).

His current research interests include computer-aided design for field-programmable gate array synthesis and application-specified integrated circuit physical synthesis.

Fang Gong (S'08) received the B.S. degree from the Computer Science Department, Beijing University of Aeronautics and Astronautics, China, in 2005 and the M.S. degree from the Computer Science Department, Tsinghua University, China, in 2008. He is currently working toward the Ph.D. degree in the Electrical Engineering Department, University of California, Los Angeles.

His research interests mainly focus on numerical computing and stochastic techniques for CAD, including fast circuit simulation, yield estimation and optimization. He also works on numerics parallel and distributed computing.

Lei He (M'99–SM'08) received the Ph.D. degree in computer science from the University of California, Los Angeles, in 1999.

He has published one book and over 200 technical papers with 12 best paper nominations. mainly from the Design Automation Conference and International Conference on Computer-Aided Design. and five best paper or best contribution awards, including the *ACM Transactions on Electronic System Design Automation* 2010 Best Paper Award. His research interests include modeling and simulation, VLSI circuits and systems, and cyber physical systems.

Majid Sarrafzadeh (F'96) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 1987.

He joined Northwestern University as an Assistant Professor in 1987. In 2000, he joined the Computer Science Department, University of California at Los Angeles (UCLA). He is currently a Codirector of the UCLA Wireless Health Institute, where he has a few dozen active projects with medical doctors and nurses around the world. He has published approximately 370 papers, coauthored 5 books, and is a named inventor on many U.S. patents. His recent research interests lie in the area of Embedded and Reconfigurable Computing with emphasis on healthcare.

Dr. Sarrafzadeh has served on the technical program committee of numerous conferences and been a general chair of many of them. He has collaborated with many industries in the past 25 years industries and was the architect of Monterey Design Systems—Synopsys acquired the company. He was a cofounder of Hier Design, Inc. Hier Design was acquired by Xilinx in 2004. He has recently cofounded Medisens and BioAssyst: both companies in the area of wireless health.