Mitigating FPGA Interconnect Soft Errors by In-Place LUT Inversion

Naifeng Jing[1], Ju-Yueh Lee[2], Weifeng He[1], Zhigang Mao[1], Lei He[2]
[1] School of Microelectronics, Shanghai Jiao Tong University, Shanghai, China*
[2] Electrical Engineering Department, University of California, Los Angeles, USA

Abstract — **Modern SRAM-based FPGAs (Field Programmable Gate Arrays) use multiplexer-based unidirectional routing, and SRAM configuration cells in these multiplexers contribute to the majority of soft errors in FPGAs. In this paper, we formulate an In-Placed inVersion (IPV) on LUT (Look-Up Table) logic polarities to reduce the Soft Error Rate (SER) at chip level, and reveal a locality and NP-Hardness of the IPV problem. We then develop an exact algorithm based on the binary integer linear programming (ILP) and also a heuristic based on the simulated annealing (SA), both enabled by the locality. We report results for the 10 largest MCNC combinational benchmarks synthesized by ABC and then placed and routed by VPR. The results show that IPV obtains close to 4x chip level SER reduction on average and SA is highly effective by obtaining the same SER reduction as ILP does. A recent work IPD has the largest LUT level SER reduction of 2.7x in literature, but its chip level SER reduction is merely 7% due to the dominance of interconnects. In contrast, SA-based IPV obtains nearly 4x chip level SER reduction and runs 30x faster. Furthermore, combining IPV and IPD leads to a chip level SER reduction of 5.3x. This does not change placement and routing, and does not affect design closure. To the best of our knowledge, our work is the first in-depth study on SER reduction for modern multiplexer-based FPGA routing by in-placed logic re-synthesis.**

Keywords - SRAM-based FPGA; Soft Errors; SER; Routing; Interconnect; Fault Mitigation; Logic Re-synthesis; Logic Polarity

I. INTRODUCTION

Modern SRAM-based FPGAs use SRAM cells to configure logic and interconnects, numerically totaling up to 160 million in XilinxVirtex-6 [1]. These FPGAs suffer from single event upset (SEU) induced soft errors, and their resilience against SEU decreases with technology scaling. Therefore, reducing the soft error rate (SER) for SRAM-based FPGAs has gained growing significance. Classic Triple Module Redundancy (TMR) employs circuit redundancy both in LUT and interconnect but with high overhead in area, power and performance. Recent logic re-synthesis techniques, such as ROSE [2], IPR [3], IPD [4] and R2 [5] reduce SER in LUTs by leveraging different logic masking techniques. Targeting on low or no cost in area, power, performance and design closure, they are preferred by non-mission-critical applications, e.g. networking and communications, which in fact are the primary applications of FPGAs. However, these techniques do not explicitly consider the interconnect SER and thus chip level SER reduction could be limited due to the interconnect dominance in FPGAs.

Modern FPGAs have shifted to multiplexer-based (MUX-based) unidirectional routing architecture [6][7], where the fault mechanism is different from conventional bidirectional routing in the previous studies [8][9]. In this paper, considering MUX-based unidirectional routing, we formulate an In-Place inVersion (IPV) of LUT logic polarities to reduce the interconnect SER, and reveal a locality and NP-Hardness of the IPV problem. We then develop an exact algorithm based on the binary Integer Linear Programming (ILP) and a heuristic algorithm based on the Simulated Annealing (SA), both enabled by the locality. We report results for the 10 largest MCNC benchmark circuits mapped by ABC [10] and then placed and routed by VPR [11]. The results show that IPV can obtain nearly 4x reduction on chip level SER. In addition, the SA approach is highly effective, obtaining the same quality of results when compared to exact ILP solutions but is much faster in runtime. In contrast to the IPD algorithm with highest 7% chip-level SER reduction among [2-5], SA-based IPV obtains nearly 4x reduction and runs 30x faster. Furthermore, combining IPV and IPD leads to 5.3x SER reductions at the chip level. This does not change placement and routing, and thus has no impact on design closure. To the best of our knowledge, this work is the first in-depth study on SER reduction for modern MUX-based FPGA routing by the in-placed logic re-synthesis.

The paper is organized as follows. Section II introduces the behavior and evaluation of the soft error on the unidirectional routing architecture. Next, we formulate the IPV problem in section III and present IPV properties and algorithms in section IV. Section V shows the experimental results and section VI concludes this paper.

II. PRELIMINARIES

A. FPGA Architecture and Interconnect Fault

An FPGA architecture is mainly defined by Configurable Logic Blocks (CLBs) and routing architectures as illustrated in Fig. 1. Interconnects are critical because they contribute a large portion of total FPGA area and total configuration bits. In our concerned unidirectional routing architecture, both the inter-CLB routing (including connection boxes and switch boxes) and intra-CLB routing employ directional MUXes to route signals. Each MUX is typically configured by several encoded configuration bits (CRAM bits), which contribute to the majority of the CRAM bits in an FPGA. For example, we observe that interconnects contribute to nearly 80% of the CRAM bits for the 10 largest MCNC benchmarks when they are synthesized to the minimum FPGA dimensions by 6-input LUT [12].

As illustrated in Fig. 1, when one of the bits flips its value due to a soft error on a MUX, an erroneous input signal (dotted) is then selected. If this signal from the faulty MUX reaches the

* Supported by National Hi-Tech Research and Development Program (863) of China under Grant No.2009AA01170.

This work was partially sponsored by Cisco and JPL.

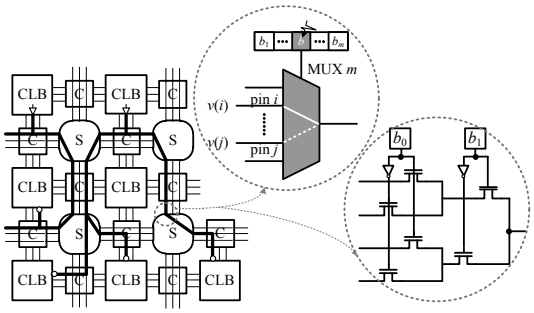

Figure 1. FPGA unidirectional routing and multiplxer structure.

primary outputs of the chip, a functional failure at the chip level occurs. Note that this fault mechanism is not a bridging fault as studied for bidirectional routing in [8][9].

B. Fault Rate Evaluation

We evaluate the failure rate for each CRAM bit under the single fault assumption[1] by SER_b, which is defined as follows.

Definition 1: *For a circuit* **C** *with n primary inputs, the soft error rate of a* CRAM *bit b, denoted as* SER_b *is the probability of functional failures on the circuit due to the* SEU *on bit b.*

$$SER_b = \Pr(\mathbf{C}_b(x) \neq \mathbf{C}_{\bar{b}}(x) \mid b \xrightarrow{\text{SEU}} \bar{b}) \quad (1)$$

where $x \in (0,1)^n$ is the exhaustive set of input vectors and $\mathbf{C}_b(x)$ is the circuit outputs under x, and $\mathbf{C}_{\bar{b}}(x)$ is the circuit outputs when b is flipped due to SEU. The SER_b can be obtained by exhausting 2^n input vectors, which is a very time-consuming process. In practice, it can be approximated by a Monte Carlo based fault simulation of as many as K times, which can provide a good accuracy as studied in [14].

In general, the metric of SER_b applies to any circuit element as long as it is configured by CRAM bits. For example, we introduce $SER_{\mathbf{R}}$ in (2), as the total routing SER to evaluate the sensitivity of functional failure to all the CRAM bits in routing elements denoted by **R**. We also quantify the circuit fault rate by all the CRAM bits in various elements in circuit **C** as in (3), which is referred as chip level SER in this paper.

$$SER_{\mathbf{R}} = \sum_{b \in \mathbf{R}} SER_b \quad (2)$$

$$SER = SER_{\mathbf{C}} = \sum_{b \in \mathbf{C}} SER_b \quad (3)$$

III. Problem Formulation

A. Motivating Example of Fault Masking

As illustrated in Fig. 1, when MUX m has one of its CRAM bits b flipped due to SEU, the output is driven by net j instead of the desired net i. If j carries a different logic value $v(j)$ from $v(i)$, a fault is injected onto the inputs of the immediate fan-outs of m. However, if $v(j)$ equals to $v(i)$, no fault is injected even if SEU happens. That is, the fault can be instantly masked at m. In addition, SER_b also depends on the observability don't care of MUX m, $obv(m)$, which indicates if the fault can be masked by logic during its propagation to circuit outputs as in [3]. As a result, SER_b can be given by $(v(i) \oplus v(j)) \cdot obv(m)$.

Note that logic polarity can be independently determined on each input and the output of an LUT in FPGA (see one example in Fig. 2). This technique is called logic polarity inversion and has been used to optimize timing [15] and power [16]. Here, we use an example in Fig. 3 to show how logic polarity inversion helps to reduce SER on a single LUT. Given the observability for the MUX and the two values on pin i and pin j, one can see that the SER of b_k can be reduced from 0.5 to 0 by inverting the logic polarity on net j.

[1] Simultaneous multiple SEUs seldom happen in commercial FPGAs as reported in [13].

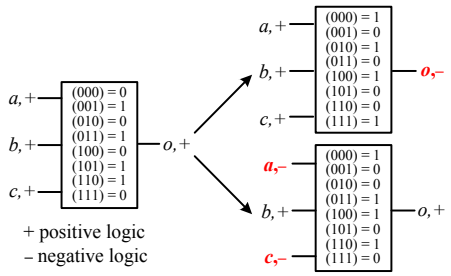

Figure 2. Atomic operations for LUT logic polarity inversion.

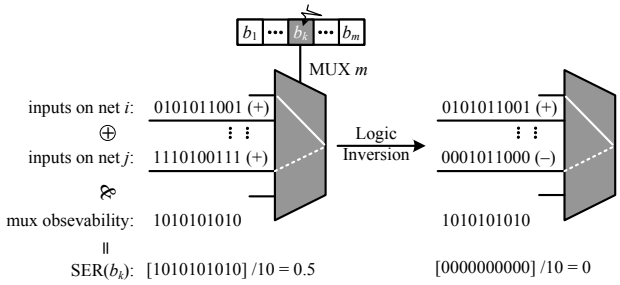

Figure 3. LUT polarity inversion improves fault tolerance on interconnects.

B. Problem Formulation

Logic polarity inversion may lead to conflict among multiple LUTs. An example is illustrated in Fig. 4, where m_1 may require LUT 2 as negative to locally mitigate the fault, while m_2 may require LUT 2 to stay positive. To find an optimal logic polarity assignment for all the LUTs and minimize $SER_{\mathbf{R}}$, we formulate the In-Place inVersion (IPV) problem as follows.

Formulation 1 (In-Place inVersion Problem): *Given a circuit, assign the logic polarity for each* LUT, *such that the* SER *for all multiplexer-based interconnects is minimized.*

We provide a bipartite graph representation in Fig.5 for a better understanding of our IPV problem, where each bottom node L_i represents one of the n LUTs, and each top node b_k represents one of the m CRAM bits used in the routing MUXes in the placed and routed circuit. There is an edge between L_i and b_k if L_i is either correctly or mistakenly by b_k at a MUX.

Given the single fault assumption, each CRAM bit connects two LUT nodes, known as a pseudo fan-in LUT pair, and are denoted $L(b_k)$ and $l(b_k)$. For the example in Fig. 4, $L(b_k)$ = LUT 3 is the desired driving LUT and $l(b_k)$ = LUT 1 is the selected driving LUT due to SEU. Therefore in Fig. 5, each b_k has exact two incoming edges, but the degree of each L_i depends on the number of MUXes it connects to. Each b_k is annotated with a bit SER value that is associated with the polarities of its pseudo

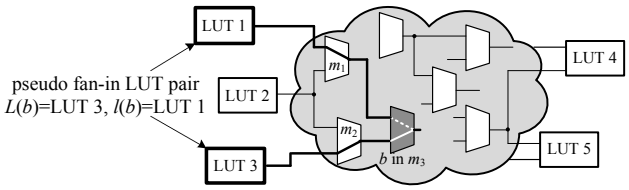

Figure 4. The pseudo fan-in pair of a SEU affected bit.

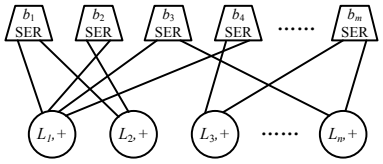

Figure 5. IPV problem representation by a bipartite graph

fan-in LUT pair (as stated by Property 2 below). Thus, our IPV problem tries to reassign the polarity for each L_i, so that $SER_{\mathbf{R}}$ can be minimized for all b_k (k=1, …, m).

IV. Properties and Algorithms

A. Properties of IPV Problem

Property 1: *Our* IPV *problem is* NP-Hard.

Sketch of Proof: IPV problem can be reduced from the binary Max-Sum (labeling) problem which is known to be NP-Hard [17]. We skip the details of the proof due to the limited space.

In IPV problem, when one or multiple LUTs are selected to be inverted for fault masking, $SER_{\mathbf{R}}$ should be updated accordingly after each inversion. Intuitively, each update needs a new pass of circuit fault simulation that is highly time-consuming. In this paper, we reveal that $SER_{\mathbf{R}}$ can be analytically updated by a pre-calculation of the bit fault rates based on property 2.

Property 2 (Locality of Bit SER upon Polarity Inversion): *Under the single fault assumption and for a given logic network,* SER_b *for a routing* CRAM *bit solely depends on the logic polarities of its pseudo fan-in* LUT *pair, independent of the polarities of the other* LUT*s in the network.*

We also skip the proof due to the limited space here.

B. Locality based SER Calculation

Based on the locality property, the $SER_{\mathbf{R}}$ with possible inversions can be calculated for at most 4x complexity to $SER_{\mathbf{R}}$ as in (4).

$$SER_b^4 = \begin{cases} SER_b^{00} & +L(b)\,\&\,+l(b) \\ SER_b^{01} & +L(b)\,\&\,-l(b) \\ SER_b^{10} & -L(b)\,\&\,+l(b) \\ SER_b^{11} & -L(b)\,\&\,-l(b) \end{cases} \quad (4)$$

where the quadruplicate values of $\{SER_b^{00}, SER_b^{01}, SER_b^{10}, SER_b^{11}\}$ are called a SER *quadruplet* of bit b. Each quadruplet provides four error rates indicated by the two superscripted numbers, indicating if one of its pseudo fan-in LUTs is inverted or not. For abbreviation, we denote it as $SER_b^4[P_{L(b)}, P_{l(b)}]$, where P is a function indicating the polarities of LUTs $L(b)$ and $l(b)$, i.e. + or –. Thus, the total routing $SER_{\mathbf{R}}^4$ can be written as

$$SER_{\mathbf{R}}^4 = \sum_{b\in\mathbf{R}} SER_b^4[P_{L(b)}, P_{l(b)}] \quad (5)$$

Eq. (5) reveals that $SER_{\mathbf{R}}$ for a given circuit can be updated as an algebraic sum upon each CRAM bit by its SER quadruplet. To get their values, we currently use the fault simulation method that is within 4x complexity of the $SER_{\mathbf{R}}$ in (2). In this way, the iterative fault simulation after each reassignment of LUT polarity can be avoided.

C. Binary ILP Based Algorithm

In this section, we use a binary ILP formulation to provide an insight on the capability of IPV improvement. We take a set of binary variables x_i to denote whether a LUT i is inverted or not, i.e., a positive LUT has its x_i as 0. A binary *inverting quadruplet* $\{f_b^{00}, f_b^{01}, f_b^{10}, f_b^{11}\}$ is used to denote the polarities of the pseudo fan-in LUTs for each routing bit b. As a result, the binary ILP formulation for our IPV problem is given by

$$\min \; SER_{\mathbf{R}}^4 = \sum_{b\in\mathbf{R}} \sum_{\substack{ij=00,\\01,10,11}} SER_b^{ij} \cdot f_b^{ij} \quad (6)$$

$$\text{s.t.} \;\; \begin{aligned} f_b^{00} &\le 1 - x_s \\ f_b^{00} &\le 1 - x_t \\ f_b^{00} + 1 &\ge 1 - x_s + 1 - x_t \end{aligned} \quad (7)$$

where x_s=$L(b)$, x_t=$l(b)$. This set of constraints models the fact that exactly one value in the quadruplet $\{SER_b^{ij}\}$ should be selected for each bit b using f_b^{ij} as a mask. Other constraints on f_b^{01}, f_b^{10} and f_b^{11} can be similarly written as in (7).

By forcing the corresponding x_i values to be 0 in the constraints, our ILP formulation also applies to the situation where some LUT input or output polarities are not invertible

D. Simulated Annealing Based Algorithm

We also propose a Simulated Annealing (SA) based algorithm to solve IPV efficiently while providing good quality of routing SER reduction compared to ILP. The SA based algorithm starts from the initial circuit with positive logic polarities for all the LUTs. Then, it switches to another LUT polarity assignment by inverting a random LUT polarity at each move. The objective function of the new assignment is evaluated by (5). New assignment with a better cost is always accepted while the worse assignment is accepted conditionally based on the acceptance probability. The annealing starts from a temperature of 0.008, and is updated by a decreasing factor of 1.003. It continues till the minimum temperature of 2.0e-6 is reached.

E. Overall Algorithm Flow

As illustrated in Fig. 6, our IPV algorithm mitigating SEU fault on FPGA interconnects by LUT logic polarity inversion consists of three phases. Starting from the given netlist of a circuit, it first applies logic optimization and technology mapping. The mapped circuit is packed into logic blocks, and then placed and routed by physical design tools. Secondly, in order to obtain the bit SER values, we develop an SEU fault analysis framework which starts right after P&R. It performs logic simulation based on post-layout circuit information to calculate

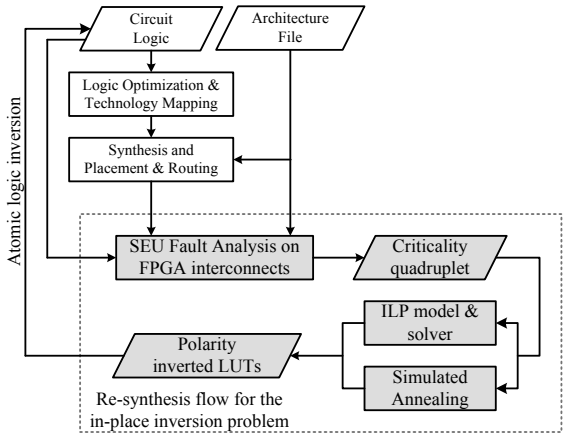

Figure 6. The overall flow of our IPV algorithm

the fault rate for each CRAM bit in routing MUXes in the form of SER quadruplet. This is the basis for both the binary ILP and the SA-based approaches. Finally, we start the ILP solver and the SA-based approach to seek the optimal reassignment of the polarities for all the LUTs. The result with maximum interconnect fault rate reduction is then selected and back-annotated to the initial circuit by the atomic logic inversion operations to finish our proposed re-synthesis flow.

V. Experimental Results

A. Experimental Settings

In this section, we report the experiments for the 10 largest MCNC combinational circuits as in [4]. We used parameterized architecture in VPR [11] to characterize different FPGA architectures. Firstly, we perform logic optimization and technology mapping onto 4 and 6-input LUTs by Berkeley ABC [10]. Each mapped circuit is packed by two different CLB architectural settings respectively, i.e. 4-input LUT with cluster size 4 and 6-input LUT with cluster of 8. Then, VPR was used to implement a minimum dimension FPGA array without incurring extra unused CRAM bits that exceed the actual need of the circuit. We applied a Monte Carlo based fault simulation to generate the bit SER quadruplets. Note that our IPV algorithm applies to either combinational or sequential circuits to mitigate interconnect SEU faults as long as the bit SER quadruplets are available. We used Mosek as our ILP solver to seek the globally optimal assignment of the LUT polarities.

B. Comparision between the ILP and SA appraoches

Table I first presents size statistics of the benchmark circuits. Next, it compares fault rate reductions in terms of SER ratios before and after applying IPV by ILP and SA approaches for all routing MUXes. From the table, one can see that both ILP and SA approaches significantly reduce SER. For example, for a 4-input LUT with cluster size 4, the interconnect SER is reduced by 1.2x to 17.2x with an average around 6x. For a 6-input LUT with cluster size 8, the SER is reduced about 5.4x on average. In addition, by considering the CRAM bit percentage, the SER can be reduced by 3.97x and 3.67x on average at the chip level for the 4-input and 6-input LUTs, respectively.

The SER reduction for 6-input LUT and 8 LUTs per cluster is slightly smaller, because larger LUT and cluster sizes have fewer interconnects and fewer MUXes that limit the room for improvement. While it is natural for different circuits to obtain different improvements, "des" has the lowest and much smaller SER reduction. This is because the SER quadruplicate values are high and close to each other in "des", which limits the design freedom that can be leveraged by IPV.

Table I also reports runtimes. The runtime excludes the fault simulation time for SER quadruplets, which is relatively small compared to the runtime consumed by ILP. From the table, one can see that ILP is able to solve most of the circuits optimally while SA can obtain the same SER reductions as ILP but runs almost 100x faster. For the other circuits as marked, where a timeout of 10 hours is applied to the ILP solver like in "des", SA obtains slightly higher SER reductions. These results show that the SA approach is highly effective and efficient for IPV problem. It is worthwhile to point out that circuits like "des" are not the largest circuits in experiments, so the efficiency of ILP depends on both circuit size and structure. When ILP and SA obtain the same SER reductions, LUT inversions in their solutions are often not the same and ILP inverts fewer LUTs in general. This implies that there are multiple "optimal" solutions from the point of view of SER reduction.

TABLE I. Interconnect and chip level SER reduction for 10 benchmark circuits

Circuit	LUT size *k*=4, Cluster size *N*=4							LUT size *k*=6, Cluster size *N*=8						
	#LUT	Int. SER Reduce		Int. CRAM bit%	Chip SER Reduce	Runtime (s)		#LUT	Int. SER Reduce		Int. CRAM bit%	Chip SER Reduce	Runtime (s)	
		ILP	SA			ILP	SA		ILP	SA			ILP	SA
ex5p	622	2.51	2.51	93.02	2.27	4131.4	35.5	458	1.81	1.90	77.95	1.78	36000*	25.6
alu4	744	2.04	2.05	91.61	1.87	36000*	41.3	524	1.96	1.99	72.37	1.82	36000*	27.2
misex3	773	3.05	3.05	91.61	2.57	4830.0	44.9	530	2.98	2.98	73.39	2.50	3845.6	29.2
apex4	821	4.79	4.79	93.57	3.83	2990.7	58.1	618	4.91	4.91	79.70	3.86	5876.8	43.0
apex2	1014	3.91	3.91	92.76	2.94	584.8	64.8	729	3.75	3.75	76.93	2.82	295.2	51.3
seq	1084	3.32	3.32	92.76	2.75	2115.4	78.5	782	3.40	3.40	77.61	2.78	7284.4	57.9
ex1010	1120	7.26	7.26	93.18	4.72	4132.3	70.4	682	7.46	7.46	78.87	4.94	5899.2	55.8
des	1750	1.16	1.17	88.22	1.16	36000*	71.6	1056	1.07	1.09	60.82	1.09	36000*	34.5
spla	2229	17.22	17.22	94.74	8.95	4602.6	183.8	1524	14.05	14.05	82.32	7.77	811.5	141.5
pdc	2304	14.60	14.60	94.54	8.59	3159.5	206.3	1609	12.51	12.51	82.71	7.31	5474.1	153.8
Avg.	–	5.99	5.99	92.60	3.97	–	–	–	5.39	5.40	76.27	3.67	–	–

TABLE II. CHIP LEVEL SER REDUCTION COMPARED TO PREVIOUS WORK

Circuit	IPV SER Reduce		IPD SER Reduce		IPD+IPV Chip SER Reduce	Runtime (s)	
	Int.	Chip	LUT	Chip		IPV	IPD
ex5p	1.90	1.78	3.60	1.04	1.97	26	795
alu4	1.99	1.82	3.88	1.09	2.09	27	1466
misex3	2.98	2.50	5.58	1.08	3.12	29	1235
apex4	4.91	3.86	4.59	1.09	4.89	43	1430
apex2	3.75	2.82	8.39	1.12	4.02	51	1137
seq	3.40	2.78	5.72	1.09	3.53	58	1659
ex1010	7.46	4.94	4.27	1.07	7.05	56	1635
des	1.09	1.09	1.48	1.02	1.10	34	2022
spla	14.05	7.77	7.04	1.07	13.23	142	3270
pdc	12.51	7.31	8.63	1.07	12.17	154	3429
Avg.	5.40	3.67	5.32	1.07	5.32	62	1808

C. Chip Level SER Reduction Compared to Previous Work

Because SA is effective and efficient, SA is used for IPV in the rest of this paper. Table II reports chip level SER reductions, by comparing IPV with a previous IPD algorithm [4] on mapped 6-input LUTs. Although IPD significantly reduces the fault rate on LUTs (around 5x), its chip level SER reduction is merely 7%, because it does not consider interconnects explicitly. Therefore, future SER mitigation algorithms should be developed with explicit consideration of interconnects.

In Table II, we also evaluated the combined algorithm of IPD+IPV on 6-input LUTs, where the IPV is applied after IPD. Because IPD is performed within each LUT, it is orthogonal to IPV. From the table, we see that the combination of these two algorithms reduces chip level SER by 5.3x on average. With much reduced interconnect SER by IPV, the SER reduction by IPD for LUTs (or in general, by algorithms from [2-5]) gains more significance. Finally, Table II compares runtimes for IPV and IPD and shows that IPV runs 30x faster on average.

VI. CONCLUSION AND FUTURE WORK

By targeting on routing multiplexers that are the dominant routing elements in the modern unidirectional FPGA routing architecture, we have identified a new fault masking mechanism and formulated an IPV problem for In-Place inVersion of LUT logic polarities to maximize fault masking for all interconnect multiplexers. We have shown that the problem is NP-Hard and also revealed a locality for IPV. Due to the locality, we have developed two algorithms based on ILP and SA approaches. Experiments have shown that IPV obtains nearly 4x reduction on average for the chip level SER. In addition, SA is effective and efficient because it obtains the same fault reductions as ILP, if not better, but runs almost 100x faster. Note that IPV does not change placement and routing, therefore it is an in-place optimization. It should have little or no impact on design closure. This will be verified in the future.

IPV has obtained close to 4x reduction on average for the chip level SER. The best existing in-place algorithm is IPD [4], which has merely about 7% chip-level SER reduction but runs 30x slower. However, combining IPV and IPD leads to a chip-level SER reduction of 5.3x on average. This is because IPV and IPD target at interconnects and LUTs respectively and they are complementary to each other.

While the proposed IPV algorithms and implementations apply to both combinational and sequential circuits, we have not yet developed SER calculation for sequential feedbacks. This will be the focus of our future work. Furthermore, the interaction between IPV and those techniques from literatures such as [2-5] will be studied for further SER mitigation.

REFERENCES

[1] Xilinx, "Virtex-6 Family Overview". Jan-2010.

[2] Yu Hu, Zhe Feng, Lei He, and Rupak Majumdar, "Robust FPGA resynthesis based on fault-tolerant Boolean matching", *Proceedings of International Conference on Computer-Aided Design*, 2008, pp. 706-713.

[3] Zhe Feng, Yu Hu, Lei He, and Rupak Majumdar, "IPR: in-place reconfiguration for FPGA fault tolerance", *Proceedings of the International Conference on Computer-Aided Design*, 2009, pp. 105-108.

[4] Ju-Yueh Lee, Zhe Feng, and Lei He, "In-Place Decomposition for Robustness in FPGA", *Proceedings of the International Conference on Computer-Aided Design*, 2010, pp. 143-148.

[5] Manu Jose, Yu Hu, Rupak Majumdar, and Lei He, "Rewiring for robustness", *Proceedings of Design Automation Conference*, 2010, pp. 469-474.

[6] G. Lemieux, E. Lee, M. Tom, and A. Yu, "Directional and single-driver wires in FPGA interconnect", *Proceedings of the International Conference on Field Programmable Technology*, 2004, pp. 41-48.

[7] Smith A.M., Constantinides G.A., Wilton S., and Cheung P., "Concurrently optimizing FPGA architecture parameters and transistor sizing: Implications for FPGA design", *Proceedings of the International Conference on Field Programmable Technology*, 2009, pp. 54-61.

[8] S. Golshan and E. Bozorgzadeh, "Single-event-upset (SEU) awareness in FPGA routing", *Proceedings of the Design Automation Conference*, 2007, pp. 330-333.

[9] E. S. S. Reddy, V. Chandrasekhar, and M. Sashikanth et al, "Detecting SEU-caused routing errors in SRAM-based FPGAs", *Proceedings of the International Conference on VLSI Design*, 2005, pp. 736-741.

[10] *ABC: A system for sequential synthesis and verification*. Berkeley Logic Synthesis and Verification Group.

[11] J. Luu, I. Kuon and P. Jamieson et al., "VPR 5.0: FPGA cad and architecture exploration tools with single-driver routing, heterogeneity and process scaling", *Proceeding of the International Symposium on Field Programmable Gate Arrays*, 2009, pp.133-142.

[12] Naifeng Jing, Ju-Yuen Lee and Zhe Feng et al., "Quantitative SEU fault evaluation for SRAM-based FPGA architectures and synthesis algorithms", *Proceedings of the International Conference on Field Programmable Logic and Applications*, 2011.

[13] Ken Chapman, "SEU Strategies for Virtex-5 Devices", Xilinx, 2009.

[14] Samuel Luckenbill, Ju-Yueh Lee, Yu Hu, Rupak Majumdar, and Lei He, "RALF: reliability analysis for logic faults: an exact algorithm and its applications", *Proceedings of the Conference on Design, Automation and Test in Europe*, 2010, pp. 783–788.

[15] Kai Zhu, "Post-route LUT output polarity selection for timing optimization", *Proceedings of the International symposium on Field Programmable Gate Arrays*, 2007, pp. 89–96.

[16] J. H. Anderson, F. N. Najm, and T. Tuan, "Active leakage power optimization for FPGAs", *Proceedings of the International Symposium on Field Programmable Gate Arrays*, 2004, pp. 33–41.

[17] Tomas Werner, "A Linear Programming Approach to Max-Sum Problem: A Review", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1165–1179, Jul. 2007.