# Single-Event-Upset (SEU) Awareness in FPGA Routing

S. Golshan
Computer Science Department
University of California, Irvine
Email: golshans@uci.edu

E. Bozorgzadeh
Computer Science Department
University of California, Irvine
Email: eli@ics.uci.edu

## ABSTRACT
The majority of configuration bits affecting a design are devoted to FPGA routing configuration. We present a SEU-aware routing algorithm that provides significant reduction in bridging faults caused by SEUs. Depending on the routing architecture switches, for MCNC benchmarks, the number of care bits can be reduced between 13% and 19% on average with comparable delay. In addition, in Asymmetric SRAM FPGA using our router average FIT (failure-in-time) rate is reduced by 36%.

## Categories and Subject Descriptors
B7.2 [**Hardware**]: Integrated Circuits – *Design Aids*

## General Terms
Algorithms, Design, Reliability, Theory

## Keywords
SRAM-based FPGA, soft error, routing, single-event-upset

## 1. INTRODUCTION
With smaller feature size, higher density, and lower operating voltage, tolerance to transient errors is manifesting itself as a key challenge in nanoscale SRAM-based FPGA devices. With smaller noise margins, transistors are more susceptible to bit flips [6]. In Xilinx Virtex FPGAs, about 91% of sensitive bits to soft errors are configuration bits [6]. The SEU (single-event-upset) caused by soft error on configuration memory can affect the functionality or connectivity of the implemented design on FPGA. If the SEU leads to a fault in configuration, it needs to be corrected by reconfiguration. However, not all the configuration bits belong to an implemented design on a FPGA. Configuration bits that affect the functionality of a design are referred to as *care bits* [2].
Configuration bits in FPGAs consist of both routing and logic block configuration bits, majority of which contribute to configuration bits for routing architectures (around 90% [5][6]). The majority of configuration bits in routing architectures are 0-bits. Most of 0-bits correspond to switches which do not belong to the routes of the nets in the design.

Reducing 0-bit care bits implies reducing the adjacency between the routed nets. A Bit flip in such 0-bit care bits caused by SEU leads to error on more than one net (bridging fault) which increases the complexity of testing, diagnosis, and error detection [3][6][7]. In this paper, our goal is to reduce 0-bit care bits during routing a given netlist. The challenge is to introduce care bit reduction while timing constraints and routability (or feasibility) constraints are satisfied. This paper focuses on impact of physical design tools to reduce the routing care bits for enhanced tolerance to soft error.
There is a large body of related work in the area of soft error mitigation for SRAM-based FPGAs [2-7]. Modular Redundancy techniques are presented as full TMR [4, 6] and partial TMR [6]. There also exists a large research work in testing (e.g. [7]), detection of errors and correction for LUTs and routing architecture. The close related works to this paper are the proposed place and route algorithms in [4][8]. The proposed place-and-route algorithm [4] is targeted for TMR-based designs. Our proposed algorithm is also applicable to non-TMR designs. In [8], place and route algorithm aims at switch boxes and does not consider connection boxes which are not negligible.
In this paper, we show that care bit reduction is an additive function along the route of a net and we prove that a maze-routing-based router is suitable for achieving enhanced immunity to soft errors. The proposed routing algorithm is generic and can be applied to several proposed methods and techniques for SEU mitigation in FPGAs. Our router reduces the bridging faults (between the two nets) by significantly reducing the net-adjacency (e.g. 64% in low-congested routing architecture). In [2], asymmetric SRAM (ASRAM) is introduced for configuration memory leakage and soft error reduction. In this architecture, in addition to total number of care bits, the distribution of 0's and 1's in the care bits affect the FIT rate. By using our router, the average FIT (failure-in-time) rate is reduced by 36% for MCNC benchmarks compared to SSRAM (Symmetric) FPGAs.

## 2. CARE BIT in FPGA ROUTING ARCHITECTURE
The routing resources of a SRAM-based FPGA consist of wire segments and two types of configurable modules, *connection boxes* and *switch boxes* [1]. Figure 1.a depicts a FPGA routing architecture. The connection boxes control the connectivity between logic module pins and wire segments within horizontal and vertical channels. In most of current FPGA devices mux-based connection boxes are used for input pins of logic blocks. Connections between horizontal and vertical wires are made by switch boxes. In order to realize a connection, it is required that SRAM cells controlling the routing switches (routing configuration bits) to be properly loaded. FPGA routing architecture is represented by a *routing graph* as shown in


*DAC 2007,* June 4–8, 2007, San Diego, California, USA.

Figure 1.b. Each node of the routing graph represents a routing resource in FPGA. These elements can be input pins, output pins or segments. Each edge represents a possible connectivity, i.e., a switch, between the two nodes.

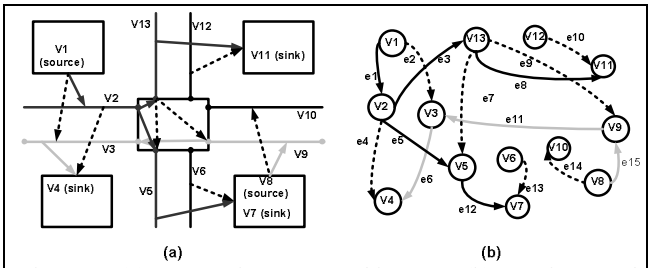


Figure 1: (a) A generic FPGA architecture. (b) Routing graph derived from (a).

The SRAM configuration bit can enable or disable a connection between two routing resources. Once two routing resources are connected to realize a route for a net, the switch is closed and the SRAM bit controlling the switch is a care bit. The switches in routing architecture are either buffered (active) or passive (like pass transistor) [1]. The care bits corresponding to open switches depend on switch types. Figure 2 demonstrates the care bit respect to routing resource A. Switches connecting resource A to resources B,C, and D are open. Resource A is used in the route for net A. The SRAM bit connecting resources A and D is not a care bit with respect to resource A. The SRAM bit connecting resources A and C is a care bit for resource A regardless of resource C being used or not. However, the SRAM bit connecting resources A and B is a care bit if B is used in some route. Depending on the routing architecture, the resources in Figure 2 can be input/output pins of the logic modules or routing segments. We can categorize the care bits to the following categories:

*1-per-net care bits:* any closed switch connecting two routing resources corresponding to the same net is considered as a 1-per-net care bit.
*0-per-net care bits:* any open switch that hurts the functionality of one net in case it is closed is referred to as 0-per-net-care bits. This mainly occurs in switch boxes. Either the both segments are used by the same nets (like T junction of a multi-terminal net) or it is between a used segment and unused segment in buffered switches. If a 0-per-net care bit gets closed, it either provides a redundant route between two nodes, or it would generate a combinational closed loop.
*Cross-net care bits:* Any interference on nets caused by closing an open switch between two different nets is considered as a cross-net care bit (Figure 2). An open switch between two routing resources corresponding to two different nets is considered as a cross-net care bit.

In Figure 1.b, Net 1 and Net 2 are depicted bold and gray respectively. The dashed lines represent open switches. $e_7$, $e_{10}$ and $e_{13}$ are the 0-per-net care bits. The cross-net care bits are: $e_4$, $e_2$ *and* $e_9$. All the closed switches are 1-per-net care bits. In buffered mux-based connection boxes, the care bits are a combination of 1's and 0's but they generate constant number of care bits. In switch boxes, any route crossing the box generates 1-per-net care bits. Depending on the switch connectivity, one or more cross-net care bits or 0-per-net care bits are generated. In the next section, we present our routing algorithm to reduce different types of care bits.

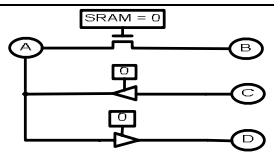


Figure 2: Care bit respect to routing resource A.

# 3. OUR APPROACH

The problem of soft-error-aware routing is as follows:

*Given a routing graph and a set of multi-terminal nets, route each net with the least care-cost, where care-cost is the number of routing care bits.*

In the routing graph, a pass transistor switch is represented by two directed edges. A buffered switch is represented by a directed edge. The care cost of the adjacent node depends on the switch type between the two nodes. There is no care bit with respect to the output pins of logic module. There are care bits with respects to each used input pin. Depending on mux-based or demux-based connection the care bit cost can vary. In our approach, we consider the cost of input pins. For simplicity, we do not include them in discussions.

## 3.1 Two-terminal Net Routing

Routing a two-terminal net in a routing graph is simply finding a path between the two nodes. Provided that the costs associated with each node remain constant during the execution, one can apply a maze-based routing algorithm to find the optimum path. In order to ensure the optimal result by maze routing algorithm on a two-terminal net, total cost of a path must be evaluated as the sum of the constant costs associated with the nodes along the optimum path. $X_v$, $Y_v$ and $Z_v$ are the number of cross-net care bits, 1-per-net care bits and 0-per-net care bits, respectively. The care-cost $C(V)$ of each node $V$ is

$$C(V) = X_V + Y_V + Z_V \qquad (1)$$

**Lemma 1-** *In routing graph G, the care bits with respect to an optimal route is the summation of care bit cost of each node along the route.*

**Lemma 2-** *Maze routing algorithm with care cost as the label of each node provides an optimal route between two- terminal nets.*

The tri-state buffer care bit cost for each node is a constant depending on the number of incoming tri-state buffer switches. We count the number of cross-net care bits, 0-per-net care bits and 1-per-net care bits. Therefore, care bits can be pre-calculated and used as a cost function. In order to calculate the number of pass transistor care bits associated with a node, $X_V$ and $Y_V$ can be calculated before the search for optimum path. $X_V$ can be calculated by counting the number of its occupied neighbors. Lemma 3 shows that the optimum path with minimum care cost does not have non-zero term $Z_V$.

**Lemma 3-** *No node in the optimum route of a two-terminal net is associated with a pass transistor 0-per-net care bit.*

Hence, during cost computation, the term $Z_V$ is not used in labeling the nodes for maze-based routing. The term $Z_V$ from Equation (1) is dropped. Based on the lemmas presented in this subsection, we have the following theorem.

**THEOREM I-** *Minimum care bit cost between two nodes in the routing graph can be computed in polynomial time by applying maze-routing algorithm.*

## 3.2 Multi-terminal Net Routing

Since optimum multi-terminal net routing is generally an NP-hard problem, we use a maze-based heuristic approach to find the routes. Given a source and *n* sinks, we route one sink at a time until all the sinks are connected to the route. We define a *branch* as a sequence of the nodes in the routed net where there is no Steiner node along the sequence. Routing each new terminal adds new branches to the partially-routed net. Lemma 4 implies that we do not need to update the costs of the routing nodes during finding a path to connect a terminal to the route. We need to update the cost of each node once a new sink is routed. The cost function for each node as expressed in Equation 1 still holds

**Lemma 4-** *All the nodes along the branches in an optimum route have no pass transistor 0-per-net care bits.*.

## 3.3 SEU-Aware FPGA Router

Conventional FPGA routers tackle two important issues: timing and congestion. Labeling of the nodes in the routing graph is based on the delay of the route as well as the routing congestion of the nodes. The challenge is the integration of care bit reduction cost in a timing-driven routing algorithm. VPR, state-of-the-art place and route tool for FPGAs, has a maze-based timing-driven algorithm [2]. We have modified the cost function for each routing resource node to:

$$\text{cost}(v) = crit(i, j) \cdot d(v) + (1 - crit(i, j)) \cdot r(v) \quad (2)$$

Where *d(v)* and *r(v)* are the delay cost and the general routability cost. *crit(i,j)* is the criticality of terminal *j* for net *i*. Timing-driven router chooses the routes with smaller load and capacitance. Routing switches between the wire segments add additional capacitance to the route and therefore routes with lesser number of routing switches are selected. As a result, the timing optimization factor inherently reduces the 1-per-net care bits but does not consider 0-per-net care bits or cross-net care bits. We add them to routability factor of cost function. The routability cost (*r(v)*) can be expressed as $C_i(v).cong(v)$ where *cong(v)* is the congestion cost of the node *v* as modeled in [1]. $C_i(v)$ is the care-cost function expressed in (1). Each net is ripped-up and re-routed sequentially with the cost update until all the routes are feasible. We update the care-cost function of the nodes neighboring with recently routed nodes in two cases: After each terminal is routed and after each net is rerouted.

# 4. EXPERIMNETS

The architecture used for all the experiments is Virtex-like FPGA architecture with four types of wire segments (single (8%), double (20%), hex (60%), and long (12%)). All the routing switches in connection modules are buffered (mux-based input connection and demux-based output connection). The switches in switch boxes are either buffered or passive. The switch type is set to subset switch. Each configurable logic block consists of two 4-input LUTs. We used TV-Pack ([1]) to generate the netlists from MCNC benchmarks followed by VPR timing-driven placement tool. We then applied two different routing algorithms: VPR timing-driven router, and SEU-aware router. We implemented our router on top of VPR tool. In evaluation, we exclude 3 I/O-dominated MCNC benchmarks (bigkey, dsip, and des). We evaluated the number of care bits in different channel widths (number of routing tracks per channel). We set the channel widths to 30%, 100% and 300% of minimum channel (shown as $\Delta W$ in Tables 1 and 2) for each benchmark to observe the impact of our router at different routing congestion rate. We report the care bits in two categories: the connection box care bits *(CB)* and the switch box care bits *(SB)*.

In the first experimental setup, all the switches in the switch boxes are fully buffered. The results for VPR routing tool and our routing tool are gathered in Table 1. Each entry is the average over all MCNC benchmarks. The overall care bit improvements are 11.11%, 12.67%, 12.21% for channel width increases of 30%, 100% and 300% ($\Delta W$), respectively. In all experiments, we observe significant decrease in the number of cross-net care bits (43.65%, 64.42, and 91.31%) as well as average reduction of 14.87%, 16.07%, and 15.49% in connection boxes (CB) for different channel widths. The rest of cross-net care bits in buffered switches are migrated to 0-per-net care bits and that is why we observe increase in such care bits. The average critical path delay is degraded by 3.68%, 1.45% and 0.42% for channel width increases of 30%, 100% and 300%. Significant reduction in cross-net care-bits in switch boxes and output pins reduces the bridging faults. Hence, the bit flips caused by SEU mostly affect one net. This is an important issue in testing, diagnosis, and error detection in FPGAs [3].

**Table 1: Care bit reduction (buffered switches)**

| ΔW% | Tool | Total care bits (av.) | Switch box care bits (SB) | | | CB |
|---|---|---|---|---|---|---|
| | | | Cross-net care bits (av.) | 0-per-net care bits (av.) | 1-per-net care bits (av.) | |
| 30% | VPR | 233057.4 | 41489.2 | 58662.7 | 12650.7 | 120254.7 |
| | NEW | 207186.3 | 23379.0 | 68624.9 | 12811.6 | 102370.8 |
| | % | **11.11** | **43.65** | **-16.98** | **-1.27** | **14.87** |
| 100% | VPR | 232905.4 | 25902.2 | 67812.0 | 11766.4 | 127424.7 |
| | NEW | 203386.9 | 9214.9 | 75169.5 | 12054.2 | 106948.3 |
| | % | **12.67** | **64.42** | **-10.84** | **-2.44** | **16.07** |
| 300% | VPR | 239254.7 | 15198.7 | 76217.5 | 11710.7 | 136127.8 |
| | NEW | 210025.4 | 1320.1 | 81620.1 | 12040.4 | 115044.7 |
| | % | **12.21** | **91.31** | **-7.09** | **-2.74** | **15.49** |

In the next setup, we set half of the switches to pass transistors and half to buffered switches in switch boxes [9]. Such architectures introduce less routing area as well as fewer routing configuration bits. Using our router, we observe improvement in total routing care bits (13.83%, 16.94%, and 19.63% on average for different channel widths). Reducing the cross net care bits would not generally lead to more per-net care bits in pass transistors. In this architecture, the total care bit reduction not only comes from connection boxes and cross-net care bits but also significant improvement in per-net-care bits. The delay degradation is 4.6%, 2.35% and 4.05% for channel width increases of $\Delta W$. we need to mention that compared to the fully-buffered switch box architecture, the delay degradation is only 8.6%, 1.8% and 1.01% in different channel width increase of $\Delta W$. In such architecture, our proposed router has significant impact with negligible degradation in delay.

Since multiplexed input connections are used in their connection boxes, we cannot improve the number of the care bits associated

with the input pins. In asymmetric SRAM-based FPGAs, the care bits in input connection boxes can be optimized for FIT rate reduction. In the rest of this section, we present the impact of SEU-aware router on ASRAM-based FPGA.

**Table 2: Care bit reduction (buffered + pass transistor switches)**

| ΔW% | Tool | Total care bits (av.) | Switch box care bits (SB) | | | CB |
|---|---|---|---|---|---|---|
| | | | Cross-net care bits (av.) | 0-per-net care bits (av.) | 1-per-net care bits (av.) | |
| 30% | VPR | 197683.8 | 32957.21 | 32502.6 | 13359.2 | 118864.7 |
| | NEW | 170339.2 | 22530.21 | 31031.4 | 13566.6 | 103211.0 |
| | % | **13.83** | **31.64** | **4.53** | **-1.55** | **13.17** |
| 100% | VPR | 195624.3 | 21195.8 | 36076.1 | 12508.3 | 125844.0 |
| | NEW | 162470.2 | 12255.3 | 30045.6 | 13236.5 | 106932.8 |
| | % | **16.94** | **42.18** | **16.72** | **-5.82** | **15.03** |
| 300% | VPR | 201257.3 | 12187.16 | 40945.8 | 12368.3 | 135756.0 |
| | NEW | 161745.7 | 4938.5 | 28990.3 | 13551.5 | 114265.4 |
| | % | **19.63** | **59.48** | **29.20** | **-9.57** | **15.83** |

As mentioned in [2], one method to enhance the immunity to soft errors in configuration bits is to use asymmetrical SRAMs. These SRAMs (ASRAM-0) make 0→1 upsets less probable than 1→0 upsets. Assuming that the configuration bits of a connection block multiplexer is the binary representation of the track number to which the input pin is connected, the new router tries to connect the input pins to those tracks whose track numbers have lesser number of 1's in their binary representation. As indicated in [2], the failure in time (FIT) rate for SRAMs storing 1 is about 2.4 times of those SRAMs storing 0. The conventional SRAM failure rate is assumed to be 1000 FIT/Mbit, where 1 FIT is 1 failure in $10^9$ hours of operation. In addition to the protection ASRAM-0 provides for zeros in the care bits, the care bits reduction can further reduce the FIT rate by significant reduction of 1's in configuration bits in connection boxes both for input and output pins.

**Table 3: FIT rate reduction for VPR and our tool using ARSAM**

| Tool | Input Con. Box | | Output Con. Box | | Switch Box | |
|---|---|---|---|---|---|---|
| | # 1's | # 0's | # 1's | # 0's | # 1's | # 0's |
| VPR | 29523 | 30759 | 3910 | 75425 | 11693 | 91002 |
| FIT % | **1.01%** | | **37.3%** | | **32.0%** | |
| NEW | 20193 | 40089 | 3466 | 60494 | 11428 | 79501 |
| FIT % | **13.81%** | | **49.13%** | | **38.9%** | |

Table 3 demonstrates the FIT rate reduction in ASRAM FPGAs compared to SSRAM (symmetric SRAM) FPGAs. We implement the circuits on such FPGA devices two times: once using VPR router and once using SEU-aware router. SEU router reduces the number of 1's in the input connection boxes care bits by almost 1/3 on average. VPR and SEU router reduce the average total FIT rate by 26% and 35%.

# 5. CONCLUSIONS

In this paper, we presented our method to integrate care bit reduction during FPGA routing with timing and congestion optimization for enhanced immunity to single-even-upsets caused by soft errors. The results show almost no degradation in delay while reducing the number of care bits by 19% on combined buffered and passive switches in routing architecture. The router significantly reduces the net adjacency which in turn reduces the bridging faults cased by SEUs.

# 6. REFERENCES


[1] V. Betz, J. Rose, and A. Marquardt, Eds. 1999 Architecture and CAD for Deep-Submicron FPGAs. Kluwer Academic Publishers.

[2] Srinivasan, S., Gayasen, A., Vijaykrishnan, N., Kandemir, M., Xie, Y., and Irwin, M. J. 2004. "Improving soft-error tolerance of FPGA configuration bits". In ICCAD 2004.

[3] Sundar Reddy, S., Chandrasekhar, V., Sashikanth, M., Kamakoti, V. and Vijaykrishnan, M. "Detecting SEU-caused Routing Errors in SRAM-based FPGAs", VLSI Design Conference, 2005.

[4] Sterpone, L. and Violante, M. , "A new reliability-oriented place and route algorithm for SRAM-based FPGAs", IEEE Transactions on Computers, Vol. 55, No. 6, June, 2006, pp. 732-744.

[5] Asadi, G. and Tahoori, M. B. "Soft error rate estimation and mitigation for SRAM-based FPGAs", ACM/SIGDA international Symposium on Field-Programmable Gate Arrays, 2005.

[6] P. Graham, M. Caffrey, J. Zimmerman, D. E. Johnson, P. Sundararajan, and C. Patterson, "Consequences and Categories of SRAM FPGA Configuration SEUs", Military and Aerospace Applications of Programmable Logic Devices (MAPLD) , 2003.

[7] Metra, C., Pagano A., and Ricco, B., "Online Testing of Transient and Crosstalk Faults Affecting Interconnections of FPGA-Implemented Systems", ITC 2001.

[8] Zarandi, H., Miremadi S., Pradhan, D. and Mathew J., "SEU-Mitigation Placement and Routing Algorithms and Their Impact in SRAM-based FPGAs", as a poster in ISQED 2007.

[9] Sheng, M. and Rose, J., "Mixing buffers and pass transistors in FPGA routing architectures", In Proc. of ACM International Symposium on FPGA, 2001.