# Single-Event-Upset Mitigation Placement and Routing Algorithms for Field-Programmable Gate Arrays

**Ren Xiaoxi*, Wu Chu, Ding Yu**
Department of Information Science and Engineering, Hunan University,
Changsha, China. telp/fax:+8615074816064
*Corresponding author, e-mail: chuchu9827@126.com

***Abstract***

*To reduce the effects of single-event upsets (SEUs) on field-programmable gate arrays (FPGAs), we propose anti-VPR, an anti-SEU algorithm. The Anti-VPR algorithm is based on VPR, a popular placement and routing tool. The proposed algorithm optimizes the FPGA place cost function and reduces the occurrence of errors, such as open circuit error and short circuit error, by computing the error propagation probability and node error rate of the Configurable Logic Blocks. The Anti-VPR algorithm is implemented and tested on several MCNC benchmark circuits. Experimental results show that the proposed Anti-VPR algorithm achieves a 36.2% greater reduction of sensitive bits compared with the original VPR algorithm without the need for extra hardware overhead, unlike the traditional TMR approach.*

***Keywords****: single-event upset, VPR, placement and routing, critical path delay*

## 1. Introduction

To shorten the development cycle and reduce the cost of modern satellites, research and development personnel use SRAM-based FPGA extensively for research and production. Although SRAM-based FPGAs use SRAM cells to configure logic and interconnects, these FPGAs are vulnerable to the effects of energetic particles, particularly single-event upsets (SEUs), which change their storage state and result in FPGA functional errors. These errors exhibit features such as transience, randomness, and recoverability, which are called “soft errors.” Results show that COTS-based FPGA is widely used because of its low cost and the size limitations of satellites [1, 2]. However, these advantages make it more vulnerable to soft errors. Despite using the Xilinx XQV300, an average of 2.05 single events [3] per day occur on radiation-hardened FPGA chips. SEUs have the most serious effect on electronic devices and have become a critical problem in SRAM-based FPGAs.

## 2. Background

Domestic and foreign scholars have put forward a variety of solutions by which to resist SEU. The most representative method includes the following: Triple Modular Redundancy (TMR) [4], scrubbing [5], and EDAC. The basic concept of TMR is that a circuit can be hardened against SEUs by designing three copies of the same module and building a majority voting system for the outputs of the replicated circuits. Although TMR enhances the ability of the system to resist SEUs, it also increases the system area and power consumption. Scrubbing involves periodically reloading the entire content of the configuration memory. The disadvantage of scrubbing is that the complexity of the system increases. EDAC can retrieve and detect errors, but the redundant information itself does not have a self-protection function: once the FPGA SEU itself suffers attacks, EDAC cannot guarantee the reliability and security of the system.

The above methods are proposed from a hardware perspective. Ref. [6] proposed a reliability-oriented placement and routing algorithm named RoRA. This algorithm imposes routing restrictions to reduce the occurrence of SEU, thus minimizing the effect of SEU-induced soft errors. However, this method was designed for TMR-based circuits only and is unsuitable for non-TMR systems. The algorithms put forward in [7, 8] add the reliability factor to the cost

*Received December 20, 2013; Revised July 17, 2014; Accepted August 4, 2014*

function of the simulated annealing-based placer and the Pathfinder-negotiated congestion algorithm router. In addition to timing and congestion, soft error is a target of placement and routing. However, the abovementioned method reduces SEU errors only from the view of detailed placement and routing information, and does not take error propagation probability (EPP), which describes the influence of a fault on the gate-level netlist of the mapped design, into account.

This paper proposes Anti-VPR, an anti-radiation algorithm that calculates the EPP of each Configurable Logic Block (CLB), and then optimizes FPGA CLB location and the routing of resource directions according to the error model, thereby reducing soft errors.

## 3. Introduction to Error Modeling and Error Propagation Probability

### 3.1. Error Modeling

The general structure of island SRAM-based FPGAs consists of a two-dimensional array of programmable blocks, called CLBs, with horizontal and vertical routing resources. CLBs and routing resources are interconnected by a connection box while the switch box is responsible for connecting wires. Connection boxes and switch boxes are connected by a series of programmable interconnect points (PIP). Figure 1 shows the architecture of FPGA.

SEUs can change the look-up table (LUT) or routing resources, thereby changing the connections of the net and finally resulting in a circuit fault, which is irreversible without re-downloading the configuration bit stream information to the FPGA configuration registers. Therefore, the anti-radiation performance of the FPGA is greatly enhanced by increasing the routing resources.

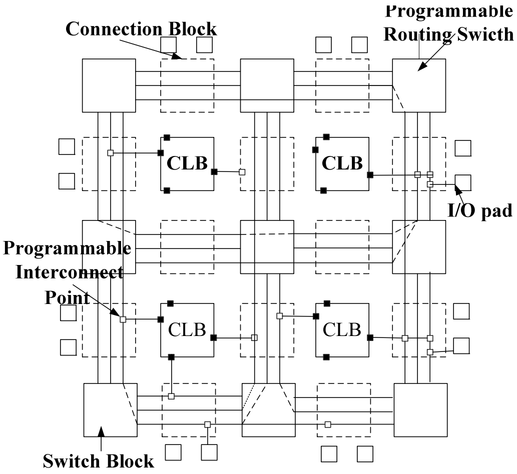


Figure 1. FPGA architecture (connection box, CB switch box, SB)

According to the anti-radiation immunity of FPGA, the configuration bits can be divided into sensitive and non-sensitive bits. SEUs attack the sensitive configuration bits of the FPGA and can cause functional failure. Non-sensitive bits will not affect the normal operation of the FPGA.

Sensitive configuration bits can be further divided into open and short sensitive bit errors.

**(1) Open Sensitive Error Bit**

Considering that a PIP connects two nodes, when an SEU occurs, a PIP flips up (from 1 to 0), so that the net can be shortened and an open sensitivity error occurs. Figure 2 shows a subset switch, where a, b, c, d, e, and f are flipped up from 1 to 0. These PIPs are open error bits.

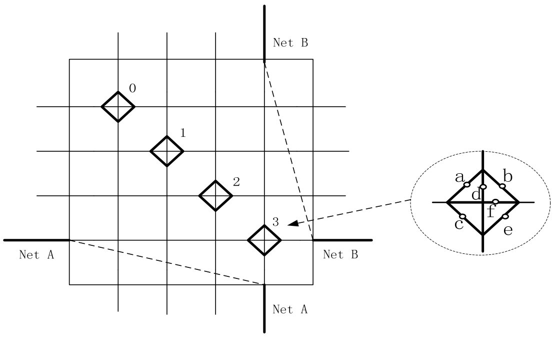


Figure 2. Schematic Diagram of Open Sensitive Error Bits

**(2) Short Sensitive Error Bit**

Unlike open error bits, the 0-to-1 flip caused by SEUs does not necessarily cause a short circuit sensitive error. In Figure 2, we consider two nets, A and B, connected by points c and b, respectively. At this time, PIPs b and c are configured to 1, whereas a, d, e, and f are configured to 0. If the SEU (0 -> 1) role in the PIP (such as a, d, e, and f) is unused, a short-sensitive error may occur. The corresponding PIP is called a short sensitive bit.

### 3.2. Error Propagation Probability

Soft error rate (SER) is an evaluation of system anti-radiation ability [9]. SER consists of two factors: EPP, which describes the effect of a failure on the gate-level netlist of the mapped design, and node error rate, which is the probability of failure based on the detailed placement and routing algorithm. EPP varies obviously across different designs. Therefore, reliability-oriented placement and routing algorithms alone cannot accurately denote the SER of the net. Furthermore, the ability to reduce soft errors by using these methods is limited.

Two methods can be used to calculate EPP: Monte Carlo simulation [10] and static analysis [9]. In the Monte Carlo simulation, an SEU is injected per iteration for a series of input signals. This method attempts to evaluate the percentage of inputs that do not generate the expected output. It is usually highly accurate, but is also time consuming.

Static analysis method uses probability theory to calculate the EPP of each error. It is not suitable for FPGA because the traditional CAD flow of FPGA includes Logic Optimization, FlowMap, T-VPack, VPR, Placement, and Routing Output Files. An LUT-based method that can calculate EPP is proposed to calculate the probability of error propagation in FPGAs.

X.blif is the netlist file that is generated after the logic optimization of FPGA. We extract the logic function from an LUT and then calculate the EPP from the LUT to the output according to the static analysis. Finally, Vpack tools are used to package the LUT and register into the logic of a given size cluster. Our study uses a 4-LUT FPGA; LUT EPP is as follows:

$$P(E)=\frac{\sum_{i\in O}\{P_a(E_i)+P_a(E_i)\}}{N} \qquad (1)$$

Where $P_a(E_i)+P_{\bar{a}}(E_i)$ represents the probability that the i-th input error value signal is passed to the output terminal E, N is the number of input signals, O = {1,2,3,……, N}, typically N = 4, and P (E) is the EPP when Vpack is mapped to each CLB.

## 4. Anti-vpr Placement and Routing Algorithm

The proposed Anti-VPR algorithm consists of two parts: Anti-VPR placement and the Anti-VPR routing algorithm. Both parts are based on classic VPR tools.

### 4.1. Anti-placement

In the placement stage of the VPR [11] tool, the tool randomly initializes a placement, calculates an initial temperature, and then enters the simulated annealing phase. The algorithm selects two CLB blocks, which are repeatedly exchanged. For each placement of each exchange, there exists a probability of whether or not to receive. The probability is based on both bounding box and timing cost. If a particular cost is less than the previous one, then this placement is more likely to be accepted.

**(1) Reduce Open Sensitive Bit**

Each PIP used in a net may form open sensitive error bit. We can reduce open sensitive error bit by reducing the length of bounding box. The wiring cost function is as:

$$Wiring\_Cost = \sum_{i=1}^{N_{nets}} q(i)[\frac{bb_x(i)}{C_{av,x}(i)^{\alpha}} + \frac{bb_y(i)}{C_{av,y}(i)^{\beta}}] \tag{2}$$

$bb_x(i)$ and $bb_y(i)$ are the length of x and y dimensions of the bounding box of net i. Where $C_{av,x}(i)^{\alpha}$ and $C_{av,y}(i)^{\beta}$ are the average channel capacities in the x and y direction of net i, respectively. $q(i)$ is used to scale the bounding boxes to estimate better wire length of nets which have more than 3 terminals.

**(2) Reduce Short Sensitive Bit**

We can reduce the overlap area of two nets to reduce the number of short sensitive bits. A short error bit is formed when two nets pass through the same switch box and a PIP between two adjacent nets is unused. If two nets do not share any switch box in their routing paths, no short error bit is associated with them. Our goal is to minimize the formation of these error short bits. Studies show that reducing the overlap area between the bounding boxes of the two nets in question can achieve this goal [14]. Carefully observing the overlap area in Figure 3, we find that the probability of a short error of circuit between netj and neti is near zero because no sink points exist in the overlap region. Based on this analysis, we propose the concept of node error rate, which refers to the circuit fault caused by short errors given a specific layout:

$$NERi = (W + L) * q(i, j) \tag{3}$$

Where W and L are the length and width, respectively, of the overlap area, $q(i, j)$ is the number of sinks in the overlap area of NET i and NET j. $NERi$ representing node error rate. The value of $NERi$ is both positively related with the size of the overlapping area and the number of sinks in it.

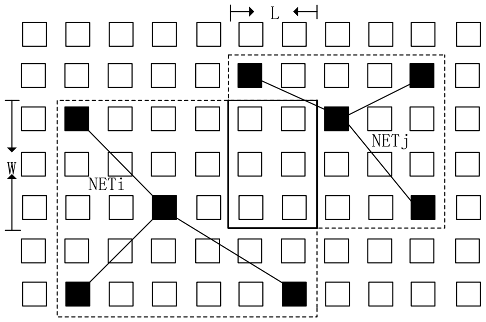


Figure 3. Reducing Short Sensitive Error Bits

Combined with the previous analysis of EPP, we put forward the new cost factor against radiation:

$$SEU\ \cos t = \sum_{\forall(i,j) in\_overlap\_area} NERi * (EPPi + EPPj) \tag{4}$$

Where $EPPi$ is the cost function equal to the arithmetic mean of the fault propagation probability of all the sinks in net i and $EPPj$ is the same principle as $EPPi$.

Through the above analysis, this paper considers not only the impact of timing and length, but also the effect caused by SEU soft errors. The cost function is modified as follows:

$$\Delta C=\lambda*\frac{\Delta Ti\min gCost}{PerviousTi\min gCost}+(1-\lambda)\frac{\Delta WiringCost+\Delta SEUCost}{PerviousWiringCost} \tag{5}$$

Where $\Delta ti\min gCost$ is the difference in the time delay cost between two CLB switches, $\Delta WiringCost$ is the difference in wire costs, and $PerviousTi\min gCost$ and $PerviousWiringCost$ are the delay cost and network cost, respectively of the last placement. FPGA placement is an NP-hard problem. Thus, $\lambda$ is a factor that balances the Wiring Cost and the Timing Cost.

### 4.2. SEU-aware Routing Algorithm

The routing algorithm of FPGAs is based on the Pathfinder-negotiated [12] congestion algorithm. An iteration of the router consists of consecutively ripping up and rerouting each net. The cost of using a route resource is a function of the current overuse of that resource and any overuse that occurs in the previous router iteration. During the first iteration, an overuse of routing resources is permitted, but in subsequent iterations, the cost for this overuse increases until every net uses only one net. To reduce the effect of SEUs on routing algorithms, we also take the anti-radiation disciplinary factor into consideration. The cost function of routing resource node N is defined as:

$$\begin{aligned}&Cost(n)=Critical(i,j)*delay(n)\\&+(1-Critical(i,j))*b(n)*h(n)*p(n)+\alpha*R\_SEUCost(n)\\&R\_SEUCost(n)=Num\_err/Sum\_node\end{aligned} \tag{6}$$

Where $Critical(i,j)$ is the criticality of terminal j for net i; $delay(n)$ is the delay cost; and $b(n)$, $h(n)$, and $p(n)$ are the base cost, historical congestion, and present congestion term of track n, respectively. The value of $R\_SERCost(n)$ is associated with the number of unused PIPs through which routing resource n and other nets can be connected. Moreover, $R\_SERCost(n)$ depends on the number of neighbor nodes of n. It reflects the whole circuit sensitivity as a short error. $Num\_err$ stands for the number of PIPs that may form a short-circuit fault with other nets. $Sum\_node$ refers to the neighbor node that can be extended to the priority queue. The effect of this routing causes the net to become more scattered. It also increases the minimum channel width, so the regulatory factor $\beta$ must be added.

As shown in Figure 4, the routing algorithm has two different paths. If we select Path1, $R\_SERCost(n)=2$. If we choose Path2, $R\_SERCost(n)=0$. In summary, our Anti-VPR algorithm selects Path2.

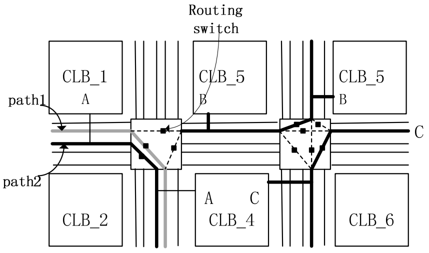


Figure 4. Result of Anti-VPR Against Radiation

### 4.3. Algorithm Flow Chart

Figure 5 shows the complete CAD process of the FPGA circuit. After logic synthesis, we obtain the .net format file with EPP using VPACK tools. Finally, the Anti-VPR placement and routing algorithms are run to reduce the system soft error rate.

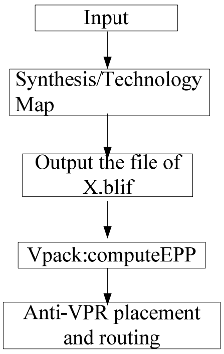
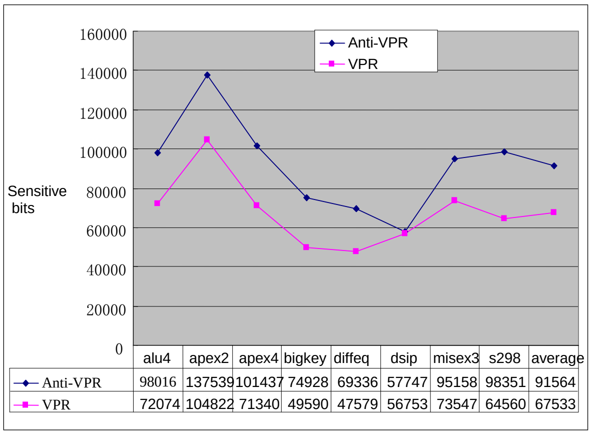


Figure 5. Algorithm Implementation Process

## 5. Implementation and Experimental Results

The experiment platform is based on a Centos5.5 Linux system. We modify the open source VPR to test our algorithm and use MCNC circuits as testing examples. In this experiment, we select some of the combinational and sequential circuits.



| | alu4 | apex2 | apex4 | bigkey | diffeq | dsip | misex3 | s298 | average |
|---|---|---|---|---|---|---|---|---|---|
| Anti-VPR | 98016 | 137539 | 101437 | 74928 | 69336 | 57747 | 95158 | 98351 | 91564 |
| VPR | 72074 | 104822 | 71340 | 49590 | 47579 | 56753 | 73547 | 64560 | 67533 |

Figure 6. The Reliability of Anti-VPR and VPR

Finally, the channel width, critical path delay, and reliability of the proposed algorithm is compared with those of the VPR algorithm. When testing the Anti-VPR algorithm, we set the placement coefficient R = 10-5 and routing coefficient $\alpha$ = 0.1; other parameters are set to default. The SEU-mitigation evaluation criterion of a circuit is the total number of $R\_SERCost\,(n)$. Figure 6 shows that the sensitive bits of each circuit have different degrees of reduction. Careful observation of the result shows that the reduction of the sensitive bits of the dsip is only 1.7%. As the probability of error propagation of the dsip itself is relatively low, the improvement is not obvious. The Anti-VPR algorithm can reduce sensitive bits by 36.2%, but the improved algorithm makes the critical path delay and channel width increase correspondingly. The experimental data are shown in Figure 8 and Table 1. The interconnection delay of FPGA accounts for 80% of the critical path delay, with the delay of the switch box accounting for the largest percentage of interconnect delay. The Anti-VPR algorithm adjusts the position of the nodes in a net, making the net more scattered. The channel width is thereby increases, but not to an obvious degree.

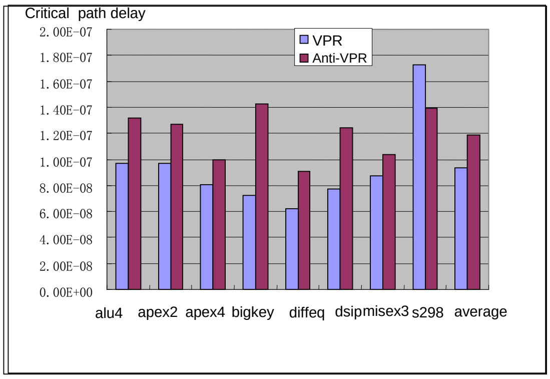


Figure 7. The Comparison of Critical Path Delay

The anti-SEU place and routing technique presented in [8] achieves only 12% to 18% reduction in the number of sensitive bits at the cost of a 5% increase in critical path delay. Furthermore, the SEU-aware routing algorithms presented in [9] result in 11% and 19% reduction, respectively, in SEU susceptibility at the cost of a 300% increase in channel width. The proposed algorithm achieves an even higher reduction in sensitive bits (36.2%) on average at a much lower channel width (2%) at the cost of a 21.6% increase in critical path delay. All the analyses show that the Anti-VPR algorithm has a considerable advantage in terms of sensitive bits reduction, critical path delay, and channel width. It also does not require hardware overhead.

Table 1. The Comparison of the Channel Width

| benchmark | VPR | Anti-VPR |
|---|---|---|
| | Channel | Channel |
| alu4 | 11 | 12 |
| apex2 | 13 | 13 |
| apex4 | 15 | 16 |
| bigkey | 8 | 8 |
| diffeq | 9 | 9 |
| dsip | 8 | 7 |
| misex3 | 13 | 13 |
| s298 | 9 | 10 |
| average | 10.75 | 11 |

## 6. Conclusion

VPR is a placement and routing tool widely used in academic fields. VPR does not feature an anti-radiation effect. This study analyzes the FPGA soft error rate and the type of error modeling, adjusts the placement and routing algorithm, and reduces the number of sensitive sites. Compared with other methods, Anti-VPR greatly reduces the number of sensitive bits and significantly improves channel width and critical path delay performance. As the size of the FPGA shrinks to nanometers, the proportion of FPGA radiation resistance in the circuit design becomes increasingly important. The study of using FPGA algorithms as a measure against radiation is also crucial.

## References

[1] DE Johnson, M Caffrey, P Graham, N Rollins, M Wirthlin.Accelerator Validation of an FPGA SEU Simulator. *IEEE Transactions on Nuclear Science*. 2003; 50(6): 2147 – 2157.

[2] E Normand. Single Event Upset at Ground Level. *IEEE Transactions on Nuclear Science.* 1996; 43(6): 2742-2750.

[3] Fuller E, Caffrey M, Blain P, et al. *Radiation test results of the Virtex FPGA and ZBT SRAM for space based reconfigurable computing*. Military and Aerospace Applications of Programmable Logic Devices International Conference. 1999: C2.

[4] Ruano O, Maestro JA, Reviriego P. A fast and efficient technique to apply selective TMR through optimization. *Microelectronics Reliability*. 2011; 51(12): 2388 - 2401.

[5] Shubhendu S Mukherjee, Joel Emer, Tryggve Fossum, et al. *Cache Scrubbing in Microprocessors: Myth or Necessity?*. Proceedings of the 10th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'04). Washington. 2004; 6: 37-42.

[6] L Sterpone, M Violante. A new Reliability-oriented Place and Route Algorithm for SRAM-based FPGAs. *IEEE Trans On Computers*. 2006; 55(6): 732-744.

[7] Hamid R Zarandi, Seyed G Miremadi, Pradhan, Dhiraj K Pradhan, et al. SEU-Mitigation Placement and Routing Algorithms and Their Impact in SRAM-based FPGAs. Quality Electronic Design, 2007. ISQED '07. 8th International Symposium on. 2007; 7: 380-385.

[8] S Golshan, E Bozorgzadeh. *Single-Event-Upset (SEU) Awareness in FPGA Routing*. Design Automation Conference, 44th ACM/IEEE. San Di-ego, CA. 2007; 07: 330-333.

[9] Hossein Asadi, MB Tahoori. Analytical Techniques for Soft Error Rate Modeling and Mitigation of FPGA-based Designs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems.* 2007; 15(12): 1320-1331.

[10] Jason Cong, Kirill Minkovich. *LUT-based FPGA Technology Mapping for Reliability.* DAC '10 Proceedings of the 47th Design Automation Conference. New York. 2010; 1: 517-522.

[11] A Marquart, V Betz, J Rose. Timing-Driven Placement for FPGAs. In the Proc. of Int'l Symposiom on FPGAs. Monterey. 2000; 2: 203-213.

[12] L Mcmurchie, C Ebeling. *Path Finder*: *A Negotiation-Based Performance-Driven Router for FPGAs*. *Field Programmable Gate Arrays*. Proceedings of the Third International ACM Symposium on. New York. 1995: 111-117.