AN OPTIMIZED PLACEMENT AND ROUTING ALGORITHM FOR CONFIGURABLE ELEMENT IN FPGA USING ERROR PROPAGATION ANALYSIS

J.TAMIZHILAKKIYA[1], SHANA VARGHESE[2]
[1]PG Student LSI Design, [2] Asst Professor, [12]Department of Information and communication Engineering,
[12]Surya group of institutions, Vikravandi

Abstract
As the feature size shrinks to the nanometer scale, SRAM-based FPGAs will become increasingly vulnerable to soft errors. So I proposed a cube-based analysis algorithm to efficiently and accurately estimate the error propagation probability. Based on such a model, propose a novel reliability-oriented placement and routing algorithm that combines both the fault occurrence probability (node error rate) and the error propagation probability together to enhance system-level robustness against soft error. Experimental results show that compared with the baseline versatile place and route technique, the proposed scheme can reduce the failure rate and increase the mean time between failures (MTBF).

I. INTRODUCTION

A novel reliability-oriented placement and routing algorithm that combines both the fault occurrence probability and the error propagation probability together to enhance system-level robustness against soft errors. To evaluate the performance by implement this technique in benchmark circuits. Field-programmable gate arrays (FPGAs) provide an attractive design platform because of their short design cycle and low development cost with exponential growth in performance and capacity. SRAM-based FPGAs are widely used in many application domains such as telecommunication, industrial control, and Embedded applications. Even in the aerospace domain, aircraft designers try to apply SRAM based FPGAs in electronic systems [1], [2], because they offer a significant advantage in high density and on-orbit re programmability. Although SRAM-based FPGAs have many advantages, they are more vulnerable to single-event upsets (SEUs) induced by high- energy particles than application-specific integrated circuits [3], [4], which limits their widespread usage in mission critical applications. When high-energy particles hit sensitive sections of the FPGA silicon, they may cause an SRAM cell to flip its state. Since the behavior of FPGA is determined by configuration bits (CBs) stored in SRAM cells, an erroneous CB may alter the implemented design, and consequently, provoke a soft error that manifests as a permanent fault until the affected bit is rewritten. The most common fault recovery approach is to combine triple modular redundancy (TMR) [6] with configuration scrubbing.TMR involves the triplication of circuit modules and the use of majority voters to mask the fault effect in a single module. To prevent the buildup of configuration upsets in more than one module, configuration scrubbing continuously configures the FPGA to clean the occurred upsets. Together, these two techniques allow an FPGA to be used reliably in a variety of space environments. While configuration scrubbing is an important mitigation technique to harden a FPGA system, frequently scrubbing incurs high performance and power overhead. The frequency of configuration scrubbing mainly depends on the mean time between failures (MTBF) of the module, which is inversely proportional to the failure rate of each TMR module. To decrease the performance and power overhead of configuration scrubbing, many computer-aided design techniques for soft error mitigation [1] have been proposed to increase MTBF (or decrease failure rate) of each TMR module. In this paper, i focus on soft error mitigation at the design stage of Placement and routing. The following are our contributions in this paper. To observe that a gap exists between the placement and routing guidance metric and the reliability evaluation metric. I also provide quantitative evaluation of this gap for the benchmark circuits. Besides fault occurrence probability, I additionally consider error propagation probability to characterize the criticality of each node, and propose a reliability oriented placement and routing algorithm, maintaining a consistent framework for the placement and routing guidance metric and the reliability evaluation metric. Experimental results show that, compared with the VPR baseline and previous placement and routing techniques, the proposed approach is highly effective at reducing failure rate and increasing MTBF. To accurately and efficiently evaluate the error propagation probability, I propose a cube-based EPP analysis technique. Experimental results show that the accuracy of the cube-based EPP analysis technique is more than 99%, while its run time can be 1600× shorter than that of a Monte Carlo simulation

a) Failure rate estimation

To mitigate soft errors in SRAM-based FPGAs, first I need to characterize the criticality of each node. As stated in the reliability of a design against soft errors can be characterized by failure rate, which is the probability that a transient fault occurs and then affects the predefined function of the design. Furthermore, failure rate is inversely proportional to MTBF, which strongly affects the frequency of configuration scrubbing. The failure rate and MTBF can be computed as follows

b*) Soft error mitigation*

For work focusing on soft error mitigation during placement and routing , their basic idea is to minimize the fault occurrence probability. The algorithms proposed in use NER to estimate the reliability of the design, and then add the term as a reliability factor into the cost function of a simulated annealing-based placer and a maze-based router. Therefore, besides timing and congestion, soft error mitigation is also an objective of placement and routing. A key problem in these placement and routing works is how to quickly estimate NER However, EPP is left out in these mitigation methods, hence there is a gap between the used placement and routing guidance metric NER and the reliability evaluation metric failure rate. Fig. 2(a) as an example, assume there are three configurable logic blocks (CLBs) to be placed along from PIs to POs, denoted as CLB1, CLB2, and CLB3. Because of logic masking effect in the design, the EPPs of wires near POs are greater than that of wires near PIs .

II. RELATED WORK

Field Programmable Gate Arrays (FPGAs) can improve dependability by detecting and correcting errors in on chip configuration data. Such an error recovery process can be executed online with minimal interference [1] of user applications. However, because Look-up Tables (LUTs) in Configurable Logic Blocks (CLBs) of FPGAs can also implement memory modules for user applications, a memory coherence issue arises such that memory contents in user applications may be altered by the online configuration data recovery process. In this paper, we investigate this memory coherence problem and propose a memory coherence technique that does not impose extra constraints on the placement of memory-configured LUTs. Two primary contributions are made. First, an experimental framework in which the viability of predictive models of routing congestion[2] for optimization during detailed placement can be evaluated is developed. The main criteria of consideration in these experiments is how (un)reliably various models from the literature detect routing hot-spots. We conclude that such models appear to be too unreliable for detailed placement optimization. Second, motivated by the result, we present a unified combinatorial framework in which cell placement and exact routing structures are captured and optimized; the framework relies on the trunk-decomposition of global routing structures and optimization is performed by a generalized optimal interleaving algorithm. FPGA-based designs are more susceptible to single-event upsets (SEUs) compared to ASIC designs. Soft error rate (SER) estimation [3] is a crucial step in the design of soft error tolerant schemes to balance reliability, performance, and cost of the system. Previous techniques on FPGA SER estimation are based on time-consuming fault injection and simulation methods. To address the inconsistency between the placement and routing objectives by fully integrating global routing into placement. As a first attempt to this novel approach, we focus on rout ability issue. We call the proposed algorithm for routing congestion minimization IPR (Integrated Placement and Routing). To ensure the algorithm [4] to be computationally efficient, efficient placement and routing algorithms Fast Place, Fast DP and Fast Route are integrated, and well-designed methods are proposed to integrate them efficiently and effectively. Previous techniques on FPGA SER estimation are based on time-consuming fault injection and simulation methods. In this paper, we present an analytical approach to estimate the failure rate of designs mapped into FPGAs. The proposed approach does not require physical implementation. We develop an algorithm for fault tolerant Boolean matching (FTBM) [5], which exploits the flexibility of the LUT configuration to maximize the stochastic yield rate for a logic function. Using FTBM, we propose a robust re synthesize algorithm (ROSE) [6] which maximizes stochastic yield rate for an entire circuit. Finally, we show that existing PLB (programmable logic block) templates for area-aware Boolean matching and logic resynthesize.

III.PROPOSED RELIABILITY-ORIENTED ALGORITHM

A novel reliability-oriented placement and routing algorithm is proposed, as shown in the bottom block labeled as proposed flow in Fig. 3.1. In contrast to the original flow, the proposed flow estimates the reliability of internal nodes by considering both NER and EPP. Based on such an enhanced placement and routing guidance metric, the proposed scheme can mitigate soft errors much more effectively

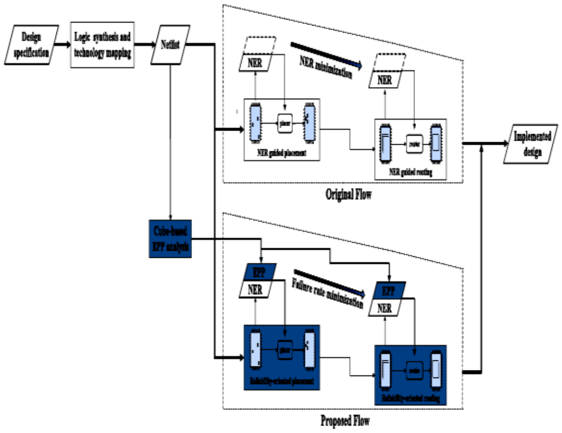


Fig.3.1.Flowchart of the proposed reliability oriented placement and routing algorithm

The EPP estimated at the stage of cube-based EPP analysis and the NER estimated at the stage of placement and routing jointly guide the placer and router towards the optimization goal of reducing failure rate, thus increasing MTBF to reduce the frequency of configuration scrubbing.

a) Generic architecture of SRAM-based FPGA

SRAM-based FPGAs have a fixed number of wire segments, switch boxes, and CLBs. The CLB is a multi-input, multi output digital circuitry that is composed of many LUTs, multiplexers, and flip-flops (FFs). A k-input LUT has 2k LUT entries, which can implement any k-input logic functions by configuring its internal 2k CBs. Wire segments are connected into wires by configuring CBs of switch boxes. LUT has *K* inputs and one output where *K* is specified as an architectural parameter of the architecture. It can be programmed to implement any *K*-input logic function. The LUT is implemented as a multiplexer whose select lines are the LUT inputs. These inputs select a signal from the outputs of *2K* SRAM cells to generate the LUT output.

b) Cube and cover

To accurately and efficiently describe the Boolean function of a design, the concepts of cube and cover have been introduced in logic synthesis. In the context of combinational logic circuits, a cube defines a relationship between inputs and outputs. It is represented as a1 a2 .ap|b1b2. bq, where a1, a2, . . . ,a_z are the values of inputs, and b1, b2,, b_q are the values of outputs. A cover is a set of cubes that have the same output value given the function of the design, in which the output term of cubes can be omitted. A cover 0 means all of the cubes contained in the cover have the same output value 0, and, vice versa, cover 1 means all of the cubes have an output value 1. An example of using cubes and covers to describe an AND logic and an OR logic are shown in Fig. 3.2 (a) and (b), respectively.

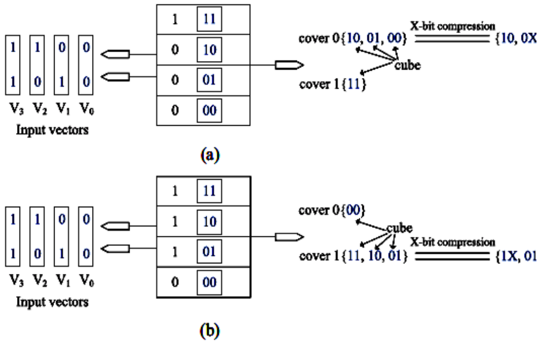


Fig 3.2 Example of cube and cover (a) 2-input AND logic. (b) 2-input OR logic

During cubical operations, X-bit compression can be performed to reduce the number of cubes in a cover, therefore reducing the computation complexity. Each “X” in a cube can be replaced by a “1” or a “0.”

c) Cube based EPP analysis

To obtain the EPP of each node, I propose a cube-based EPP analysis technique. Assuming a fault occurs at a fault site w, the fault effect propagation procedure can be partitioned into three phases, First, the fault effect is propagated from the fault site w to its immediate successor FFs (e.g., FF_i) at the first clock cycle. Correspondingly, in this phase, the error propagation probability of the fault is represented as EPP (w, i). Then, the fault effect is propagated from the immediate successor FFs (e.g., FF_i) to the last level FFs (e.g., FF_j) in t−1 clock cycles, and in this phase the error propagation probability of the fault is EPPt−1(i, j). Finally, at thet^{th} clock cycle, the fault effect is propagated from the lastlevelFFs (e.g., FF_j) to some POs (e.g., PO1), and in this phase the error propagation probability of the fault is EPP (j,∀POs).

Note that, as long as the fault effect is propagated to any POs, i consider that the fault affects the normal function of the design. Therefore, the EPP at the t^{th}clock cycle (EPP^t) is the cumulative product of the three propagation phases' EPPs.

$$\begin{aligned} EPP &= \sum_{t=1}^{t\le c} EPP^t \\ &= \sum_{t=1}^{t\le c}\sum_{i=1}^{i\le n}\sum_{j=1}^{j\le n} EPP(w,i) \\ &\quad \times EPP^{t-1}(i,j) \times EPP(j,\forall POs) \end{aligned}$$

Where n is the total number of FFs in the circuit.

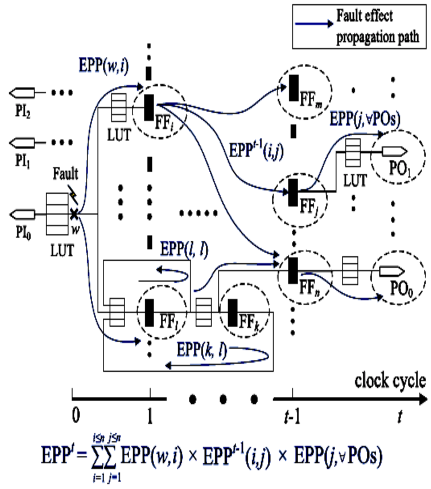


Fig 3.3 Three phases of fault propagation in a sequential circuit

The two-pass EPP analysis can be conducted simultaneously on all combinational logic parts of the design. Here we use one combinational logic part as an example to illustrate the two-pass procedures follows.

d) Forward Traverse

The first pass is forward traverse, which performs logic simulation to obtain the control covers of each wire by cubical operations. Each input wire (e.g., In0 and In1) of the combinational logic (e.g., the LUT), all input vectors that set the logic value of the wire to be 0 are packed into the control-cover 0 of the wire. In this example, the entire input vectors of the LUT are {11, 10, 01, 00}, so the control-cover 0 of In0 is {X0} and the control-cover 0 of In1 is {0X}, respectively, where the rightmost bit representsIn0 and the leftmost bit represents In1. Afterwards, according to the logic function of the LUT, the control-cover 0 of the LUT's output wire u is the adjoin of In0's control-cover 0 and In1's control-cover 0, i.e., {X0} **V** {0X} = {10, 00} **V** {01, 00} = {10, 0X}. In the same way, the control-cover 1 of u is the interface of In0's control-cover 1 and In1's controlcover 1, i.e., {X1} **I** {1X} = {11}. Finally, as fanout branches will inherit the stem's control cover, the control-cover 0s of fanout0 and fanout1 are also {10,
0X}

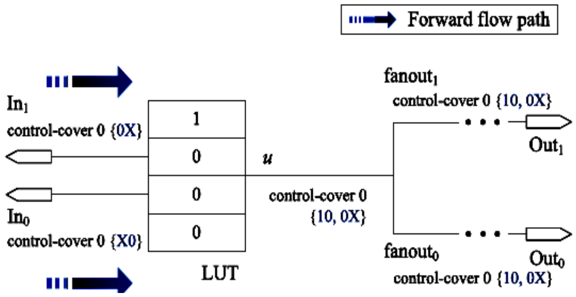


Fig 3.4 Example of forward pass

e) Backward Pass

The second pass is backward traverse, which computes the care-covers of each wire based on the cubes stored in the previously obtained control-covers. Fan out and logic masking effect are considered during the second pass. Note that, for the sake of clarity, indexes of the outputs that the fault effect Is propagated to are recorded in each cube. Fig.5.2 as an example, when targeting a stuck-at 1 fault, suppose the care-cover 1 of fanout1 is {10(Out1), 01(Out1)} and the care cover 1 of fanout0 is {0X(Out0)}, where Out1 and Out0 are the indexes of outputs. Then the care-cover 1 of *u* is the adjoin of the care-cover 1 of fanout0 and the care-cover 1 of fanout1, that is, {10(Out1), 01(Out0, Out1), 00(Out0)}. After back tracing from outputs to inputs, the EPP is the ratio of vectors represented by cubes stored in the care cover to the total number of input vectors.

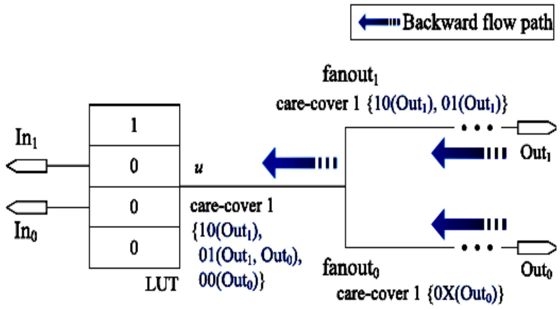


Fig 3.5 Example of backward pass

f) Algorithm discussion

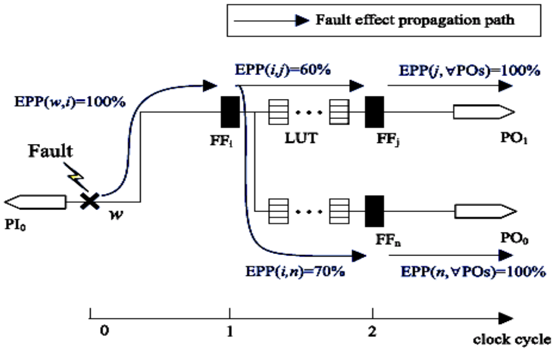


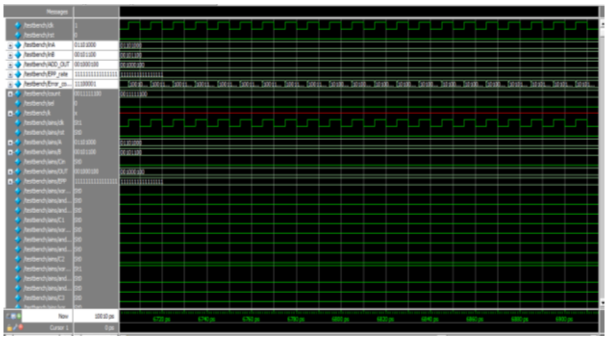
Fig3.6.Example of EPP analysis for sequential logic in case of dependent propagation

g) Analysis of Computational Complexity

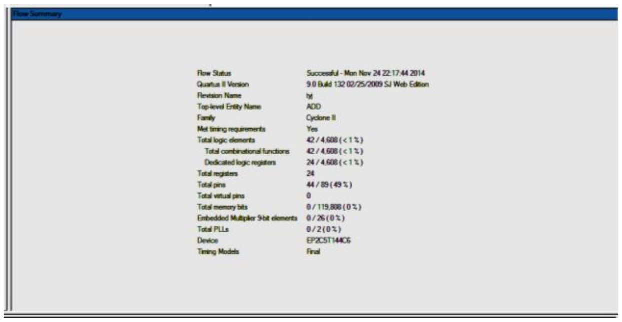
The computational complexity of Monte Carlo simulation is $O(N^C \times V \times g \times (V + E))$. For static analysis, the complexity of estimating signal probability is $O(V^2)$ Afterwards, the computational complexity of EPP in combinational logic part is linear to the scale of the circuit [6], i.e., $(V + E)$. Finally, the EPP in sequential circuit is obtained by self-multiplying$n \times n$matrix MEPP c times. So the computational complexity of static analysis is $O(V^2 \times (V + E) \times c \times n^3)$. For the proposed cube-based analysis, due to the introduction of "X" bit, the average number of cubes in each cover is N/C_{avg}, where C_{avg} is the average compression ratio of all covers. As a result, the computational complexity of cube-based analysis is $O((N/Cavg)^2 \times (V + E) \times c \times n^2)$.

IV.SIMULATION AND OUTPUT

a) Output for error circuit

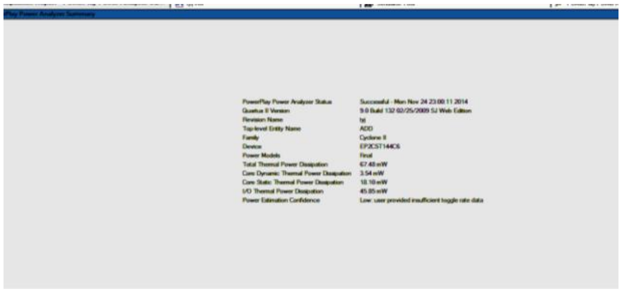


b) Area utilization for error circuit



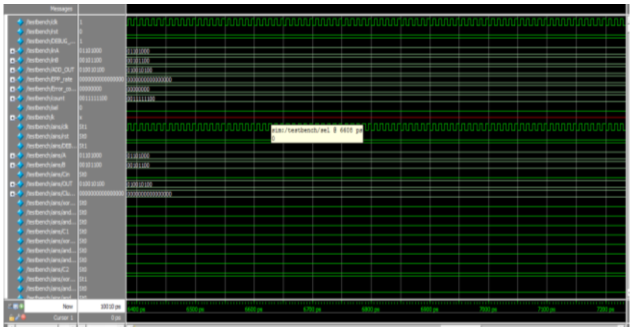
```
Flow Status                          Successful - Mon Nov 24 22:17:44 2014
Quartus II Version                   9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name                        hj
Top-level Entity Name                ADD
Family                               Cyclone II
Met timing requirements              Yes
Total logic elements                 42 / 4,608 ( < 1 % )
   Total combinational functions     42 / 4,608 ( < 1 % )
   Dedicated logic registers         24 / 4,608 ( < 1 % )
Total registers                      24
Total pins                           44 / 89 ( 49 % )
Total virtual pins                   0
Total memory bits                    0 / 119,808 ( 0 % )
Embedded Multiplier 9-bit elements   0 / 26 ( 0 % )
Total PLLs                           0 / 2 ( 0 % )
Device                               EP2C5T144C6
Timing Models                        Final
```

c) Power Utilization for error circuit

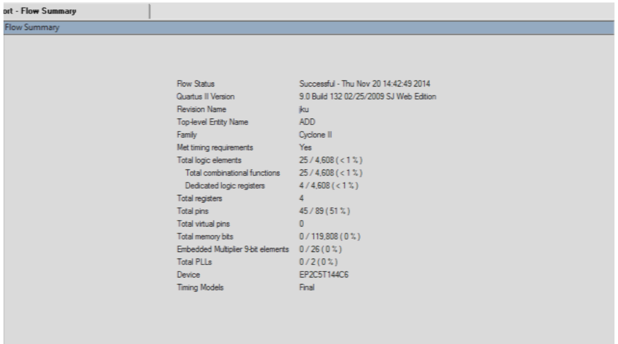


```
PowerPlay Power Analyzer Status            Successful - Mon Nov 24 23:00:11 2014
Quartus II Version                         9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name                              hj
Top-level Entity Name                      ADD
Family                                     Cyclone II
Device                                     EP2C5T144C6
Power Models                               Final
Total Thermal Power Dissipation            67.48 mW
Core Dynamic Thermal Power Dissipation     3.54 mW
Core Static Thermal Power Dissipation      18.10 mW
I/O Thermal Power Dissipation              45.85 mW
Power Estimation Confidence                Low: user provided insufficient toggle rate data
```

d) Error corrected output



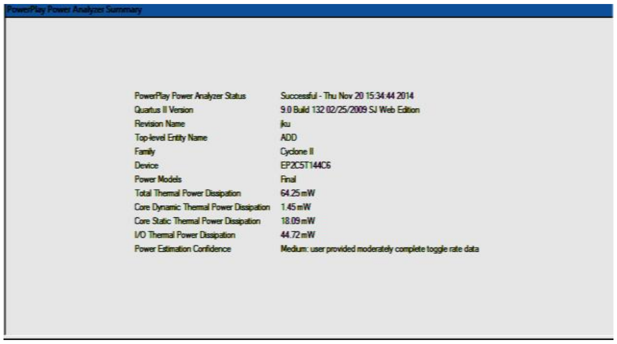
e) Area Utilization Error Corrected Circuit



```
ort - Flow Summary
Flow Summary

Flow Status                          Successful - Thu Nov 20 14:42:49 2014
Quartus II Version                   9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name                        jku
Top-level Entity Name                ADD
Family                               Cyclone II
Met timing requirements              Yes
Total logic elements                 25 / 4,608 ( < 1 % )
   Total combinational functions     25 / 4,608 ( < 1 % )
   Dedicated logic registers         4 / 4,608 ( < 1 % )
Total registers                      4
Total pins                           45 / 89 ( 51 % )
Total virtual pins                   0
Total memory bits                    0 / 119,808 ( 0 % )
Embedded Multiplier 9-bit elements   0 / 26 ( 0 % )
Total PLLs                           0 / 2 ( 0 % )
Device                               EP2C5T144C6
Timing Models                        Final
```

f) Power Analyzes Error Corrected Circuit



```
PowerPlay Power Analyzer Summary

PowerPlay Power Analyzer Status          Successful - Thu Nov 20 15:34:44 2014
Quartus II Version                       9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name                            jku
Top-level Entity Name                    ADD
Family                                   Cyclone II
Device                                   EP2C5T144C6
Power Models                             Final
Total Thermal Power Dissipation          64.25 mW
Core Dynamic Thermal Power Dissipation   1.45 mW
Core Static Thermal Power Dissipation    18.09 mW
I/O Thermal Power Dissipation            44.72 mW
Power Estimation Confidence              Medium: user provided moderately complete toggle rate data
```

V. Conclusion

The novel reliability-oriented placement and routing algorithm for SRAM based FPGA will become increasing vulnerable to soft error. Compared to the existing soft error mitigation algorithm during placement and routing, the proposed algorithm successfully bridges the gap between the placement and routing guidance metric and the reliability evaluation metric. The results showed that, compared to the VPR baseline and previous placement and routing techniques, algorithm is highly effective at soft error mitigation. To accurately and efficiently evaluate the reliability of the design, I proposed a cube-based EPP analysis technique. Based on this analysis the failure rate will be reduced and mean time between failures(MTBF) also increases.

References

[1] W. J. Huang and E. J. McCluskey, "A memory coherence technique for online transient error recovery of FPGA configurations," in *Proc. ACM/SIGDA Int. Symp.Field-Program. Gate Arrays*, 2001, pp. 183–192.

[2] D. Jariwala and, J. Lillis, "On interactions between routing and detailed placement," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Mar. 2004, pp. 387–393.

[3] G. Asadi and M. B. Tahoori, "Soft error rate estimation and mitigation for SRAM-based FPGAs," in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, 2005, pp. 149–160.

[4] M. Pan and C. Chu, "IPR: An integrated placement and routing algorithm," in *Proc. IEEE/ACM Design Autom. Conf.*, Jul. 2007, pp. 59–62.

[5] H. Asadi and M. B. Tahoori, "Analytical techniques for soft error rate modeling and mitigation of FPGA-based designs," *IEEE Trans.Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 12, pp. 1320–1331, Dec. 2007.

[6] Y. Hu, Z. Feng, L. He, and R. Majumdar, "Robust FPGA resynthesis based on fault-tolerant Boolean matching," in *Proc. IEEE/ACM Int.Conf.Comput.-Aided Design*, Nov. 2008, pp. 706–713.

[7] J. Cong and K. Minkovich, "LUT-based FPGA technology mapping for reliability," in *Proc. IEEE/ACM Design Autom. Conf.*, Jun. 2010, pp. 517–522.

[8] K. Huang, Y. Hu, and X. Li, "Cross-layer optimized placement and routing for FPGA soft error mitigation," in *Proc. IEEE/ACM DesignAutom. Test Eur. Conf.*, Mar. 2011, pp. 58–63

[9] M. Violante, L. Sterpone, M. Ceschia, D. Bortolato, P. Bernardi, M. S. Reorda, and A. Paccagnella, "Simulation-based analysis of SEU effects in SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 6, pp. 3354–3359, Dec. 2004.

[10] J. P. Roth, *Computer Logic, Testing and Verification*. New York, USA: W. H. Freeman, 1980.