A Framework for Exploring Alternative Fault-Tolerant Schemes Targeting 3-D Reconfigurable Architectures

Kostas Siozios*, Ioannis Savidis§, and Dimitrios Soudris*
*School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece
E-mail: {ksiop, dsoudris}@microlab.ntua.gr
§Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA, USA
E-mail:isavidis@coe.drexel.edu

***Abstract*—For decades computer architects pursued one primary goal: performance. Transistor scaling has translated into remarkable gains in operating frequency and reduction in power consumption. However, increased complexity from the device to architecture levels impose several new challenges, including a decrease in dependability/reliability due to physical failures. Reconfigurable platforms are highly susceptible to scaling related complexity, typically resulting in higher power consumption as compared to application-specific integrated circuits. The concern becomes far more important in the 3-D integrated circuit (IC) domain as vertically stacked blocks exhibit increased thermal resistance to the heat sink. The degradation in dependability becomes an important design challenge, not only for safety critical systems, but for the majority of architectures. In this paper, a framework used to explore alternative fault-tolerant schemes is proposed that masks the degradation in reliability for 3-D FPGA platforms. Simulation results at the RTL level highlight the benefits of the introduced solution, as the maximum operating frequency and power consumption are improved by 33% and 26%, respectively, as compared to similar state-of-the-art solutions.**

I. Introduction

Field-Programmable Gate Arrays (FPGAs) have become the key implementation medium for a large portion of digital circuits [1]. As a result, the need for faster, smaller, economical, and lower energy FPGA devices has increased. Performance enhancements are not feasible, however, with incremental solutions that simply scale CMOS technologies. Three-dimensional (3-D) integration has emerged as a revolutionary technology that satisfies these requirements. The shift from horizontal scaling to volumetric stacking of circuits has the potential to mitigate many limitations of modern integrated circuits [2]. Three-dimensional ICs contain multiple physical layers and offer considerable improvement in circuit performance, such as lower power/energy consumption, less total wire-length, higher integration density, and greater speed, as compared to two dimensional (2-D) circuits [3]. The benefits of employing 3-D integration are particularly important for the FPGA paradigm, since the performance of these circuits is primarily limited by the interconnects [4].

Although the rapid evolution of process technology has addressed architectural concerns regarding transistor limited design, new concerns have emerged. Among others, reliability degradation becomes an important issue not only during the fabrication process but also for the lifetime of the product [5], with the more recent goal for architects to design reliable products from unreliable components. Defects in circuits are tightly coupled to the operating conditions of the IC. Specifically, the mean time to failure (MTTF) due to various intrinsic reliability factors, such as electromigration, time-dependent dielectric breakdown (TDDB), and negative bias temperature instability (NBTI), vary exponentially with on-chip temperatures [6] [7] [8] [9] [10]. The problem becomes far more severe when considering the non-uniformity in the thermal profile across the entire IC due to variations in the power density of the components.

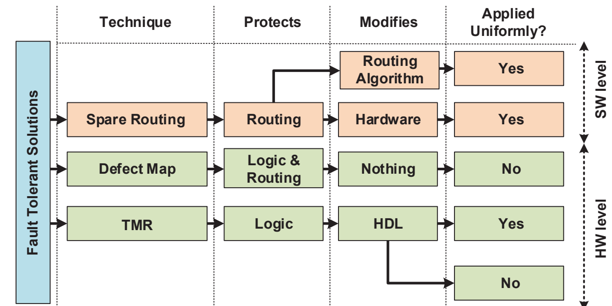

Fig. 1: Taxonomy of various pioneering works on fault tolerance.

Meeting the power, and consequently the thermal budget, is an essential criterion by which customers measure the success of an FPGA-based design. The importance of thermal budgets has become far more critical as the power density in FPGA devices has almost doubled every three years [11] [12], and is expected to continue to increase with technology scaling according to the "α-power" law [13]. For instance, with regard to commercial FPGAs, the maximum die temperature without performance degradation is reported as 80°C, whereas the absolute maximum temperature is 125°C [14]. Additionally, an average-sized design mapped onto a Virtex-E FPGA with 90% device utilization may lead to die temperatures of 50°C above the ambient temperature [14].

The importance of designing fault-tolerant systems is much more crucial for FPGAs as compared to other platforms, as upsets to an FPGA may alter not just user data but also the implemented logic. As a result, a number of architectures and design methodologies able to provide non-distributed device operation have been proposed at different levels of abstraction [15] [16], as they are summarized at Figure 1. Specifically, in literature, there are two primary approaches for designing fault tolerant systems. The first addresses the design of new hardware elements, which are fault tolerant enabled [17] [18] [19] [20], whereas the second approach implements the desired fault masking at the software-level with the use of specialized CAD tools [21] [22] [23] [24] [25].

978-1-5090-3076-7/16/$31.00 ©2016 IEEE

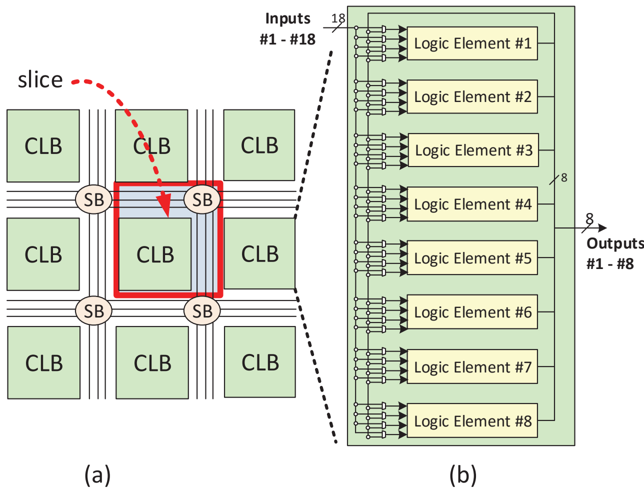

Fig. 2: Architectural template of target FPGA: (a) an instantiation of a 3×3 array and (b) the CLB's architecture.

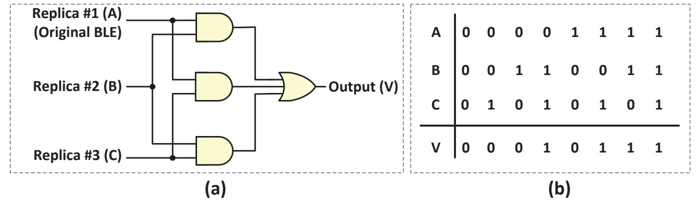

Fig. 3: Majority voter: (a) architecture and (b) truth table.

Both approaches exhibit advantages and disadvantages. The (re-)designed hardware blocks either replace existing components of conventional FPGAs, or new fault tolerant architectures may be designed to improve robustness. The drawback of applying such a strategy is increased design complexity, while the derived FPGA provides a static (defined at fabrication time) fault tolerant mechanism. Typical instantiations of block redesign involve the use of spare logic and routing resources [19] [20], as well as the use of defense-grade [26] and space-grade [27] FPGA devices. The software-based fault masking technique combines the required dependability level with the low-cost of commodity devices. For instance, by appropriately handling the inherent re-programmability feature of FPGAs, it is possible to overcome from hardware malfunctions. However, software-based fault tolerant systems assume that the designer is responsible for protecting the design. Since no hardware modifications are necessary with software-based fault tolerance, it is widely accepted for research and product development. Other algorithms that perform physical implementation (P&R) under fault tolerant [23] [24] and/or reliability [28] constraints have been proposed. In addition, a solution that embeds testing operations and alternative configurations into bitstream files is discussed in [29].

The only available commercial product for providing fault masking is based on triplication of the application functionality without imposing any architectural modifications [22]. The approach, also known as Triple Modular Redundancy (TMR), assumes three functional blocks that work in parallel, with the output derived by comparing the partial outputs from each block with a majority voter. Although TMR provides the maximum fault coverage for the implemented logic infrastructure, the additional (replica) functionality introduces significant performance, power, and area overheads, which often violate the system specifications [30]. A number of improvements have therefore been proposed. A partial TMR, where redundancy is applied selectively only to portions of the design is discussed in [31]. Similarly, an algorithm for inserting alternative voters in designs with TMR is presented in [32]. Finally, a methodology that supports application mapping with the maximum affordable (in terms of delay and power overheads) redundancy is found in [33].

In this paper, we introduce a framework for exploring different selections that mask faults in 3-D FPGAs both at architectural, as well as the algorithmic levels. More thoroughly, the contributions of this work are summarized as follows:

- A virtual platform for 3-D FPGAs was developed. The introduced platform is parametric and easily customizable in order to enable the exploration phase for alternative architectural customizations.
- An open-source software toolset that performs application implementation (i.e. partitioning, placement and routing) onto the 3-D virtual platform was also developed. Such a toolset enables the estimation of system characteristics in an early stage of developed (before the actual implementation phase onto the target device is performed).
- The efficiency of alternative fault-tolerant instantiations targeting to mask faults at logic and interconnection resources are also analyzed and evaluated.

The rest of the paper is summarize as follows: Sections II and III deal with the modeling of 3-D FPGA platform and the analysis of the employed fault-tolerant schemes both for logic and interconnection fabric, respectively. The proposed software-supported framework for exploring and quantifying alternative instantiations of the fault-tolerant schemes is presented in Section IV. The efficiency of the derived solution is quantifying in Section V, while Section VI concludes the paper.

II. Modeling 3-D FPGA

The employed reconfigurable architecture consists of an array of slices, each of which includes a configurable logic block (CLB) and the surrounding routing infrastructure. The next level of hierarchy assumes that the CLBs are formed by eight logic elements (LEs), while each LE in turn is composed of a 4-input look-up table (LUT), a flip/flop, and a number of multiplexers (at inputs and outputs). Communication among LEs is provided through low-latency local interconnects which enable LUT chain connections by transferring the output of one LE's LUT to the adjacent LE (in order to provide fast sequential LUT connections). Around each CLB, there are 27 uniformly distributed, logically equivalent I/O pins (18 input, 8 output and 1 clock). An overview of the architecture, as well as the organization of the CLBs is shown in Figure 2. Note that apart from the selected FPGA, our proposed methodology and the supporting CAD tools are also applicable to any other commercial device without a dedicated fault tolerant mechanism.

Apart from the logic infrastructure, the interconnection network is also of high importance especially at FPGA devices. Regarding the modeling of planar wires (inside each layer), their electrical equivalent characteristics (RLC) are based on public available models provided by ITRS [5]. On contrast, the impact on power, area, and delay of interlayer connections in a

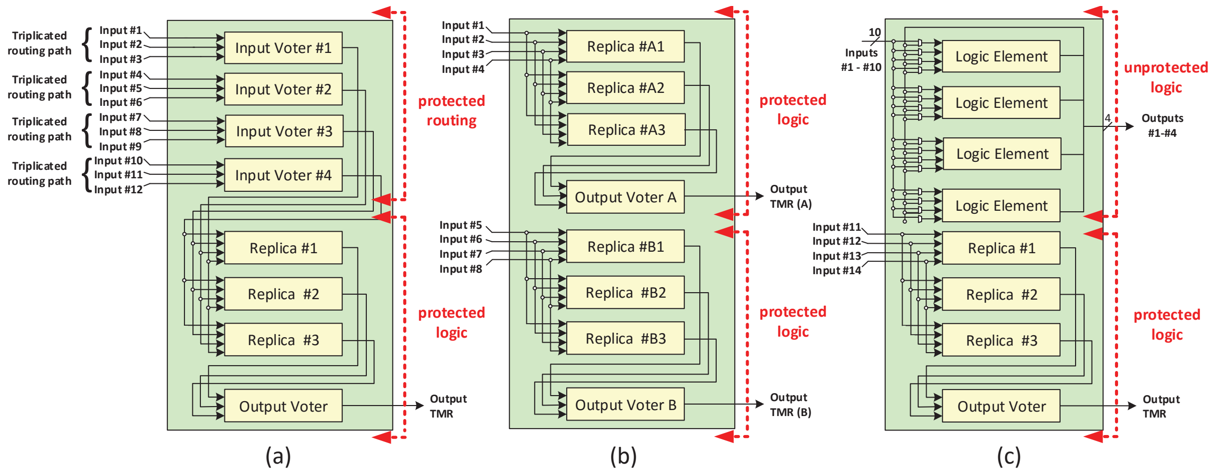

Fig. 4: Alternative instantiations of TMR: (a) protect both logic and routing fabric, (b) protect only logic fabric and (c) protect a subset of logic fabric.

3-D stacked IC dependents on (i) the number of device planes traversed, (ii) the material properties of the interconnect and TSVs, and (iii) the physical dimensions of the TSV.

The material properties and physical dimensions of the TSV are of particular interest as the impedance of the TSV is directly related to these two fabrication properties. The power and delay penalties due to the TSV impedance must therefore be considered for redundant paths in a 3-D system. In addition, as the TSV consumes active silicon area, the overhead due to the TSV diameter and corresponding keep-out-zone surrounding the TSV [34], [35] must be considered (particularly for larger diameter TSVs).

Accurate modeling of the TSV impedance is therefore of significant importance when considering path redundancy. Early closed-form expressions of the resistance, capacitance, and inductance that account for the length, width, dielectric thickness, and fill material of the TSV are provided in [36]–[38]. Models for high frequency signal propagation through a TSV have also been developed [39], [40].

Parameters, including the physical and electrical properties of the TSV, used for the simulation of the proposed platform are listed in Table I.

TABLE I: Architectural parameters for our experimentation.

Category	Parameter	Value
Technology	CMOS	45 nm
Architecture	shape	square
	layers	homogeneous
	style	island
	LUT size	$K = 4$
	number of layers	3
	layer's size	best fitted to circuit size
	routing channel width (X and Y directions)	$W_H = 50$
	vertical channel width (TSVs per 3-D SB)	$W_V = 3$
TSV	distribution	uniformly per layer
	shape	square
	die-bonding type	face-to-back
	length	4-9 μm
	diameter	1.2 μm
	min. pitch	4 μm
	resistance (R)	0.35 Ω
	capacitance (C)	2.5 fF
Routing wires per layer	length	$L1$, $L2$, $L6$ and *longlines*
	RLC modeling	PTM 45 nm

III. Alternative Fault-Tolerant Schemes

Previous studies have highlighted that an optimal fault coverage solution has to combine regions with different efficiency in fault masking [33]. Therefore, three candidate TMR-based techniques are introduced in this section, each of which trade-off the efficiency in fault masking with the resulting overhead in delay, power, and area. The architecture of the employed majority voter is shown in Figure 3, as well as the corresponding truth table. Note that the functionality of the voter could be mapped onto a single 4-input LUT.

The first variant of TMR, depicted schematically in Figure 4(a), protects both the logic and routing infrastructure of the architecture. For this purpose, all the active LEs and routing paths are protected with redundancy. This scheme assumes that only one of the eight (fabricated) LEs per CLB is actually utilized for the implementation of the application; the remaining seven LEs support the task of fault masking. More specifically, the first four LEs (denoted as "Input Voter #1 – #4") vote on the replicated routing paths, whereas the outputs are fed to the triple instantiation of the functionality of the application ("Replica #1, #2 and #3"). The three partial outputs from the replicated modules are then voted upon one more time at the "Output Voter" to produce the final result from the CLB. Note that in order to avoid data hazards, all the replicated routing paths that are fed to the same voter have to exhibit identical delay.

An alternative scheme that avoids data hazards altogether is depicted in Figure 4(b). More specifically, this instantiation of TMR masks upsets that occur only to the logic blocks, as these resources contain a higher number of programmable transistors and are therefore more susceptible to failures. Since each triplicated function requires four LEs, two functions per slice, denoted as "A" and "B", are mapped onto a single CLB. Although upsets are possible on the routing infrastructure, the significantly lower mitigation cost (due to redundancy) makes the approach suitable especially for non-mission critical (e.g. consumer products) applications. Finally, the third variant of the TMR analyzed in this paper is depicted in Figure 4(c), where the last four LEs per CLB implement a triplicated

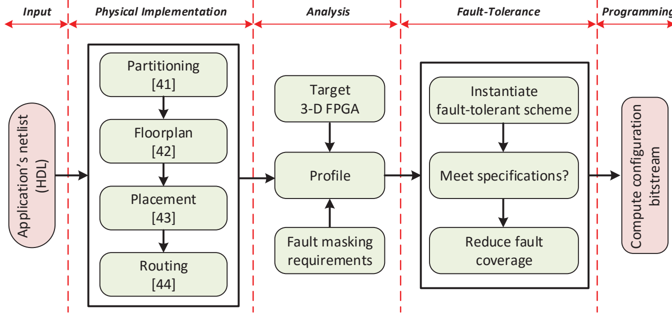

Fig. 5: The introduced framework for exploring alternative fault-tolerant schemes targeting 3-D FPGAs.

instantiation of the functionality of the application, while the remaining four LEs operate similar to a conventional application mapping. Regarding the outputs of the CLB, there are up to four signals for the unprotected LEs, as well as a single output from the majority voter. This scheme is suitable whenever only a subset of utilized logic resources have to be protected with redundancy. Compared to the other two TMR-based solutions, the third enables clustering of application functionalities with different requirements for redundancy into the same CLB.

IV. Proposed Framework

Starting from a high-level application description format such as VHDL or Verilog, our framework consists of three consecutive phases, as depicted in Figure 5. The first phase, described as "Physical Implementation", accounts for the partitioning [41], floorplanning [42], placement [43] and routing [44] of the application onto the targeted 3-D FPGA platform. Publicly available CAD algorithms from the 3-D MEANDER framework [45] are utilized during the first phase, with the objective being one of timing optimization (minimize critical path delay). Additional optimization objectives are also possible during phase 1, depending on the constrains of the given application.

System analysis is performed during the second phase, where the derived implementation of the application is analyzed in order to extract a number of reliability-aware performance metrics. Among others, the power consumption per logic block and the routing resources determined for the targeted 3-D FPGA are reported, as well as the thermal stress as derived from the Hotspot tool for the entire 3-D stack [46]. Additionally, the designer-defined constraints related to the desired level of fault masking are given as inputs for the analysis.

Depending on the profiling results, the fault-tolerance scheme that best matches the system requirements is instantiated. Starting from a uniform protection of the entire design, the level of fault masking is gradually reduced until an optimized point is selected that matches the required performance of the system. Since the alternative TMR schemes discussed in Figure 4 exhibit variations in fault coverage and resource efficiency, our framework selects the proper instantiation of TMR per CLB based on Algorithm 1. The design annotation does not impose any additional task execution of the physical implementation. The information on the swapping between alternative TMR instantiations is performed automatically with our framework by appropriately modifying the proper configuration bits.

Fig. 6: Software implementation of the proposed framework.

The framework is realized as part of a toolbox that enables exploration of alternative TMR instantiations onto targeted 3-D FPGAs under various operating conditions. The Computer-Aided Design (CAD) algorithms that are applied to the physical implementation phase of our framework are executed with the use of the 3-D MEANDER Framework [45], whereas the tasks of fault injection, selection of proper TMR scheme, as well as monitoring the effectiveness of such a selection are executed at OVP [47]. For this purpose, all the necessary interfaces that support vice-versa communication between the two software approaches were also developed and tested, as depicted in Figure 6.

Algorithm 1 Proposed algorithm for relaxation of the fault tolerance level.

Require: P $\leftarrow$ utilized slices
Require: M $\leftarrow$ desired level of fault coverage
Require: S $\leftarrow$ delay and power specifications
1: Function Relax_TMR (P, M, S)
2: **for** (each slice $\in$ P) **do**
3: G(slice) $\leftarrow$ failure_probability(M);
4: **end for**
5: PF $\leftarrow$ sort_descending(P, G(P));
6: **while** (S NOT satisfied) **do**
7: **if** (P NOT belong to control path) **then**
8: **for** (G(P) $\in$ (min{PF}, max{PF})) **do**
9: **for all** (TMR scenarios) **do**
10: evaluate(fault masking);
11: evaluate(new power, new delay);
12: **if** (S satisfied) **then**
13: select(optimal TMR scenario);
14: break;
15: **end if**
16: **end for**
17: **end for**
18: **end if**
19: **end while**
20: end Function

V. Experimental Results

A number of experimental results that prove the efficiency of the proposed methodology are provided in this section. For the purpose of evaluation, a number of industrial oriented kernels [48] are considered, as summarized in Table II. The target device is a general-purpose Altera Stratix-based FPGA without any dedicated fault tolerant mechanism [12]. The number of injected upsets for the analysis provided throughout this paper was assumed to be up to 10% of the configuration

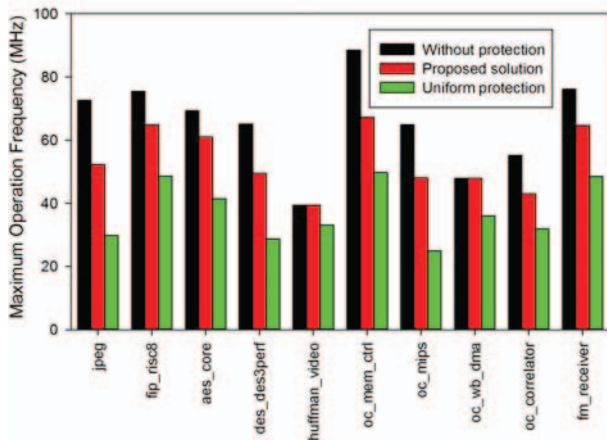

Fig. 7: Evaluating the proposed framework in term of maximum operation frequency.

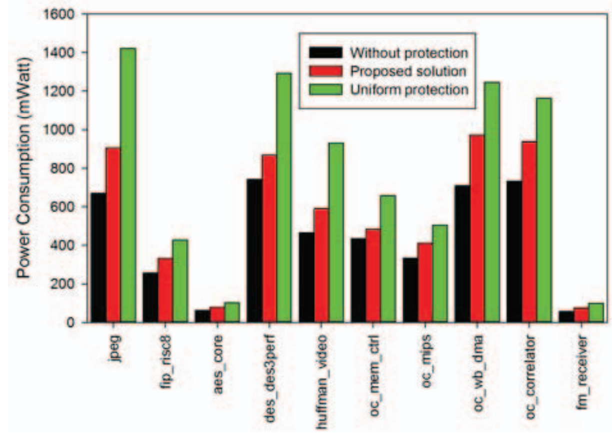

Fig. 8: Evaluating the proposed framework in term of total power consumption.

file size, while temporal distribution between consecutive upsets follows the NBTI model [49].

TABLE II: Complexity of employed kernels.

Benchmark	Functionality	Number of 4-LUTs
jpeg	Image processing	6,529
fip_risc8	RISC cpu	1,629
aes_core	Encryption algorithm	5,005
des_des3perf	Encryption algorithm	18,888
huffman_video	Video processing	689
oc_mem_ctrl	Memory controller	4,340
oc_mips	Processor controller	4,618
oc_wb_dma	Control logic for dma	5,073
oc_correlator	DSP	1,113
fm_receiver	Audio processing	1,328
Average:		4,921.2

In order to quantify the benefits of the introduced framework, the performance and power overheads are evaluated when the application is implemented onto the same FPGA with the flow discussed in Figure 5 (mentioned as "Proposed solution"). For demonstration purposes, two border cases are also evaluated: (*i*) no protection of the application's functionalities ("without protection") and (*ii*) when all the functionalities are protected against upsets with TMR triplication ("uniform protection"). The results of our analysis regarding the maximum operation frequency and power consumption are summarized in Figures 7 and 8, respectively. Throughout the study we assume that the desired level of fault masking was set to 90% of the injected upsets. Even though it is possible to derive an application implementation with even higher percentage of fault masking, the selected instantiation is acceptable for the majority part of consumer products.

Based on the results summarized in Figure 7, the uniform insertion of redundancy leads to the maximum performance degradation among the studied solutions. More specifically, on average the uniform TMR solution leads to a 1.2–2.6× lower operating frequency as compared to the "without protection" case, whereas the corresponding penalty for the proposed solution is a maximum of 1.38× slower. In other words, the proposed solution, where the level of TMR is customized according to the inherent requirements of the application, as determined during profiling, improves the operating frequency by about 33% on average as compared to a uniform TMR.

The selection of fewer triplicated resources (either logic and/or interconnect), apart from the performance enhancement is also expected to introduce considerable power savings. The total power dissipation for the implemented experimental setup is summarized in Figure 8. Based on the results, inserting redundancy uniformly over the architecture imposes considerable power overheads as compared to the initial application implementation (without TMR). More specifically, the analysis indicates that the penalty in power consumption ranges between 1.5–2.1×, whereas the corresponding power overhead with a tuned level of TMR per CLB (as is proposed by the introduced framework) is 1.11–1.37× greater. Therefore, the proposed solution achieves an average power savings of about 26% as compared to a uniform redundancy scheme.

VI. Conclusion

A software-supported framework for providing fault masking for 3-D reconfigurable architectures was introduced. Rather than similar approaches that protect the entire design, our solution provides a trade-off between the desired fault masking level and the resulting mitigation overheads due to the replica hardware. Simulation results with various benchmarks highlight the efficiency of the introduced solution, as an average improvement in maximum operating frequency and power consumption of 33% and 26%, respectively, is achieved as compared to the existing state-of-the-art approach (uniform redundancy). Additionally, the studied fault coverage (95%) ensures that the derived implementations of the application are applicable to a majority of digital applications mapped onto reconfigurable fabric (such as consumer products).

Acknowledgment

This work was partially supported by "TEAChER: TEach AdvanCEd Reconfigurable architectures and tools" project funded by DAAD (2014).

References

[1] N. Tredennick and B. Shimamoto, "The rise of reconfigurable systems," in *Engineering of Reconfigurable Systems and Algorithms*, 2003, pp. 3–12.

[2] C. Yu, "The 3rd dimension – more life for moores law," in *International Microsystems, Packaging, Assembly Conference*, Oct. 2006, pp. 1–6.

[3] V. F. Pavlidis and E. G. Friedman, *Three-Dimensional Integrated Circuit Design*. Morgan Kaufmann Publishers, 2009.

[4] K. Leijten-Nowak and J. Meerbergen, "An fpga architecture with enhanced datapath functionality," in *Proc. of the Int. Symposium on FPGA)*, 2003, pp. 195–204.

[5] ITRS, "International technology roadmap for semiconductors 2014 edition." 2014.

[6] G. Gielen and et.al., "Emerging yield and reliability challenges in nanometer cmos technologies," in *Proceedings of the conference on Design, automation and test in Europe*, ser. DATE '08. New York, NY, USA: ACM, 2008, pp. 1322–1327.

[7] J. R. Black, "Electromigration – a brief survey and some recent results," *IEEE Transactions on Electronic Devices*, pp. 338–347, Apr. 1969.

[8] P. Mangalagiri, S. Bae, R. Krishnan, Y. Xie, and V. Narayanan, "Thermal-aware reliability analysis for platform fpgas," in *IEEE/ACM Int. Conf. on Computer-Aided Design*, 2008, pp. 722–727.

[9] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The case for lifetime reliability-aware microprocessors," *SIGARCH Comput. Archit. News*, vol. 32, no. 2, pp. 276–, Mar. 2004.

[10] W. Wang, Z. Wei, S. Yang, and Y. Cao, "An efficient method to identify critical gates under circuit aging," in *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, ser. ICCAD '07. Piscataway, NJ, USA: IEEE Press, 2007, pp. 735–740.

[11] S. Borkar, "Design challenges of technology scaling," *Micro, IEEE*, vol. 19, no. 4, pp. 23–29, jul-aug 1999.

[12] Altera, "Introducing innovations at 28 nm to move beyond moores law," White paper, July 2010. [Online]. Available: http://www.altera.com/literature/wp/wp-01125-stxv-28nm-innovation.pdf

[13] T. Sakurai and A. Newton, "Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas," *Solid-State Circuits, IEEE Journal of*, vol. 25, no. 2, pp. 584–594, apr 1990.

[14] A. Lesea and M. Alexander, "Powering xilinx fpgas," Xilinx, Tech. Rep. XAPP158, aug 2002.

[15] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secur. Comput.*, vol. 1, pp. 11–33, January 2004.

[16] J. A. Cheatham, J. M. Emmert, and S. Baumgart, "A survey of fault tolerant methodologies for fpgas," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 11, no. 2, pp. 501–533, 2006.

[17] K. Nikolic, A. Sadek, and M. Forshaw, "Fault-tolerant techniques for nanocomputers," *Nanotechnology*, vol. 13, pp. 357–362, 2002.

[18] F. L. Kastensmidt, L. Carro, and R. Reis, *Fault-Tolerance Techniques for SRAM-Based FPGAs (Frontiers in Electronic Testing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[19] A. J. Yu and G. G. Lemieux, "Defect-tolerant fpga switch block and connection block with fine-grain redundancy for yield enhancement," in *FPL*, 2005, pp. 255–262.

[20] N. Campregher, P. Y. K. Cheung, G. A. Constantinides, and M. Vasilko, "Analysis of yield loss due to random photolithographic defects in the interconnect structure of fpgas."

[21] D. Bhaduri and S. K. Shukla, "Nanoprism: a tool for evaluating granularity vs. reliability trade-offs in nano architectures," in *ACM Great Lakes symposium on VLSI*. New York, NY, USA: ACM, 2004, pp. 109–112.

[22] Xilinx, "Xilinx tmr tool (url: http://www.xilinx.com/ise/optional_prod/tmrtool.htm)," 2011.

[23] R. Jain, A. Mukherjee, and K. Paul, "Defect-aware design paradigm for reconfigurable architectures," in *IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures*. Washington, DC, USA: IEEE Computer Society, 2006, p. 91.

[24] A. Doumar, S. Kaneko, and H. Ito, "Defect and fault tolerance fpgas by shifting the configuration data," in *DFT '99: Proceedings of the 14th International Symposium on Defect and Fault-Tolerance in VLSI Systems*. Washington, DC, USA: IEEE Computer Society, 1999, pp. 377–385.

[25] I. Koren and C. Krishna, Eds., *Fault-Tolerant Systems*. Elsevier Inc., 2007.

[26] Xilinx, "Defense-grade virtex-6q fpga family (url: http://www.xilinx.com/products/silicon-devices/fpga/virtex-6q/index.htm)," 2011.

[27] Xilnx, "Space-grade virtex-5qv fpga (url: http://www.xilinx.com/products/silicon-devices/fpga/virtex-5qv/index.htm)," 2011.

[28] S. Sivaswamy and K. Bazargan, "Statistical analysis and process variation-aware routing and skew assignment for fpgas," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 1, no. 1, pp. 1–35, 2008.

[29] R. Rubin and A. DeHon, "Choose-your-own-adventure routing: lightweight load-time defect avoidance," in *FPGA '09: Proceeding of the ACM/SIGDA international symposium on Field programmable gate arrays*. New York, NY, USA: ACM, 2009, pp. 23–32.

[30] C. Carmichael, "Triple module redundancy design techniques for virtex fpgas," Xilinx Application Notes 197, Tech. Rep., 2006.

[31] B. Pratt, M. Caffrey, P. Graham, K. Morgan, and M. Wirthlin, "Improving fpga design robustness with partial tmr," in *In 44th Annual IEEE International Reliability Physics Symposium Proceedings*, 2006, pp. 226–232.

[32] J. M. Johnson and M. J. Wirthlin, "Voter insertion algorithms for fpga designs using triple modular redundancy," in *FPGA '10: Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*. New York, NY, USA: ACM, 2010, pp. 249–258.

[33] K. Siozios and D. Soudris, "A methodology for alleviating the performance degradation of tmr solutions," *IEEE Embedded Systems Letters*, vol. 2, no. 4, Dec. 2010.

[34] W. Guo, V. Moroz, G. Van der Plas, M. Choi, A. Redolfi, L. Smith, G. Eneman, S. Van Huylenbroeck, P. D. Su, A. Ivankovic, B. De Wachter, I. Debusschere, K. Croes, I. De Wolf, A. Mercha, G. Beyer, B. Swinnen, and E. Beyne, "Copper through silicon via induced keep out zone for 10nm node bulk finfet cmos technology," in *Proceedings of the International IEEE Electron Devices Meeting*, Dec. 2013, pp. 340–343.

[35] S. K. Ryu, K. H. Lu, T. Jiang, J. H. Im, R. Huang, and P. S. Ho, "Effect of thermal stresses on carrier mobility and keep-out zone around through-silicon vias for 3-d integration," *IEEE Transactions on Device and Materials Reliability*, vol. 12, pp. 255–262, June 2012.

[36] I. Savidis and E. G. Friedman, "Closed-form expressions of 3-d via resistance, inductance, and capacitance," *IEEE Transactions on Electron Devices*, vol. 56, pp. 1873–1881, September 2009.

[37] I. Savidis, S. M. Alam, A. Jain, S. Pozder, R. E. Jones, and R. Chatterjee, "Electrical modeling and characterization of through-silicon vias (tsvs) for 3-d integrated circuits," *Microelectronics Journal*, vol. 41, pp. 9–16, January 2010.

[38] G. Katti, M. Stucchi, K. De Meyer, and W. Dehaene, "Electrical modeling and characterization of through silicon via for three-dimensional ics," *IEEE Transactions on Electron Devices*, vol. 57, pp. 256–262, January 2010.

[39] C. H. Wang, K. C. Fan, and H. H. Lee, "Wideband 40ghz tsv modeling analysis under high speed on double side probing methodology," in *Proceedings of the International IEEE Electron Devices Meeting*, Dec. 2013, pp. 340–343.

[40] W. Yao, S. Pan, B. Achkir, J. Fan, and L. He, "Modeling and application of multi-port tsv networks in 3-d ic," *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS*, vol. 32, pp. 487–496, April 2013.

[41] K. Siozios and D. Soudris, "A tabu-based partitioning and layer assignment algorithm for 3-d fpgas," *IEEE Embedded Systems Letters*, vol. 3, no. 3, pp. 97–100, Sept 2011.

[42] H. Sidiropoulos, K. Siozios, P. Figuli, D. Soudris, M. Hübner, and J. Becker, "Jitpr: A framework for supporting fast application's implementation onto fpgas," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 6, no. 2, pp. 7:1–7:12, Aug. 2013. [Online]. Available: http://doi.acm.org/10.1145/2492185

[43] P. Danassis, K. Siozios, and D. Soudris, "Ant3d: Simultaneous partitioning and placement for 3-d fpgas based on ant colony optimization," *IEEE Embedded Systems Letters*, vol. PP, no. 99, pp. 1–1, 2016.

[44] K. Siozios and D. Soudris, "A customizable framework for application implementation onto 3-d fpgas," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. PP, no. 99, pp. 1–1, 2016.

[45] MEANDER, "3-d meander exploration and design framework 2016 edition (url: http://proteas.microlab.ntua.gr)," 2016.

[46] K. Siozios, D. Rodopoulos, and D. Soudris, "On supporting rapid thermal analysis," *IEEE Computer Architecture Letters*, vol. 10, no. 2, pp. 53–56, July 2011.

[47] OVP, "Open virtual platforms 2016 edition (url: http://http://www.ovpworld.org/)," 2016.

[48] Altera, "Benchmark designs for the quartus university interface program version 1.0 (url: http://wl.altera.com/education/univ/research/quip/unv-quip.html#contact/)," 2005.

[49] S. Mahapatra, M. A. Alam, P. B. Kumar, T. R. Dalei, D. Varghese, and D. Saha, "Negative bias temperature instability in cmos devices," *Microelectron. Eng.*, vol. 80, pp. 114–121, June 2005.