Master-Slave TMR Inspired Technique for Fault Tolerance of SRAM-based FPGA

Farid Lahrach, Abderrahim Doumar, Eric Châtelet and Abderrazek Abdaoui

Institue Charles Delaunay (ICD), STMR, UMR CNRS 6279.
Laboratoire de modélisation et Sûreté des Systèmes (LM2S)
Université de Technologie de Troyes (UTT)
12, rue Marie-Curie, 10010, Troyes Cedex, France
E-mail: {farid.lahrach, abderrahim.doumar, chatelet, abdaoui }@utt.fr
Telephone: (+33) 325759732

***Abstract*—In order to increase reliability and availability of Static-RAM based field programmable gate arrays (SRAM-based FPGAs), several methods of tolerating defects and permanent faults have been developed and applied. These methods are not well adapted for handling high fault rates for SRAM-based FPGAs. In this paper, both single and double faults affecting configurable logic blocks (CLBs) are addressed. We have developed a new fault-tolerance technique that capitalizes on the partial reconfiguration capabilities of SRAM-based FPGA. The proposed fault-tolerance method is based on triple modular redundancy (TMR) combined with master-slave technique, and exploiting partial reconfiguration to tolerate permanent faults. Simulation results on reliability improvement corroborate the efficiency of the proposed method and prove that it compares favorably to previous methods.**

***Index Terms*—Fault-tolerance, SRAM-based FPGA, triple modular redundancy (TMR), configurable logic block(CLB), master-slave technique (MST), partial reconfiguration.**

I. Introduction

Static-RAM based field programmable gate arrays (SRAM-based FPGAs) provide programmable platforms for many applications such as networking, signal processing, and rapid prototyping logic emulation systems [1]. The programmability of SRAM-based FPGAs has helped to achieve a short design cycle and low development costs. In SRAM-based FPGAs all logic elements and programmable switches can be reprogrammed by loading a configuration bit stream, thus providing the flexibility to implement many digital circuits on the same chip. These devices are often employed in harsh and hostile environments such as critical space missions and mobile devices [2].

Many methods used to achieve fault-tolerance in SRAM-based FPGA configurable logic blocks and routing resources have been developed recently [3]. Triple modular redundancy is widely used as a method of fault tolerance to be hardened against transient faults occurred when charged particles hit the silicon, this phenomenon is concern in space but also in other harsh operating environments [4]-[7]. The basic concept of TMR is that a sensitive circuit can be hardened against transient faults (single event upsets) by implementing three copies of the same circuit and performing a bit-wise (majority vote) on the output of the triplicate circuit [8]. Although TMR combined with configuration scrubbing is used as mitigation technique in high radiation environments for Xilinx-specific block RAMs [9], TMR comes at very large resource overhead penalty. Furthermore, it provides a limited resolution for identification of the faulty component.

In order to achieve fault-tolerance of FPGA systems with low overhead Lach et al. [10] have proposed fault tolerance achievement method based on the partitioning of the physical design of the FPGA into a set of tiles. In each tile, some CLBs are employed as a spare to repair any defective CLB of the FPGA. When a fault is detected in one tile, only the concerned tile is reconfigured using partial reconfiguration. However this approach requires that reconfiguration is always possible and this is limited to low connectivity designs.

In [11] Doumar et al. have proposed a reconfiguration in the FPGA's SRAM cells to avoid faulty CLB. Specifically, in their method, the data are shifted vertically or/and horizontally in order to switch from the faulty CLB to another fault-free one. In [12] when a fault is detected in a column, the whole column is marked as faulty and it is then replaced by a spare CLB. In [13], Johon M. Emmert et al. have proposed an online fault-tolerant technique for FPGA logic blocks. In their contribution, the authors have reused defective configurable logic blocks to increase the number of effective spares. This fault-tolerant approach is based on self-test areas technique (STARs).

To the best of the author's knowledge, there is no published work that derives tolerating both single and double faults with low overhead. In this paper, we present an innovative technique for tolerating defects (maker side) and faults (user side) in SRAM-based FPGA logic blocks. The proposed MSU architecture consists of two types of CLBs; CLB-M and CLB-S. The relationship between these two architectures is managed by the master-slave technique (MST). When a defect occurs in a CLB-S, an application of partial reconfiguration on the MSU in order to replace the defective CLB-S by another among the three CLBs of the CLB-M then the defective CLB-S is avoided.

978-0-7695-4076-4/10 $26.00 © 2010 IEEE
DOI 10.1109/ISVLSI.2010.38

The remainder of this paper is organized as follows. The new fault tolerance inspired TMR method and the corresponding system architecture are detailed in Section II. The reliability evaluation of the proposed technique is given in Section III. Simulation results and discussions are addressed in Section IV. Finally, Section V concludes the paper.

II. The Proposed Fault Tolerance Method

A. Approach

The SRAM-based FPGA can be viewed as an array of configurable logic blocks (CLBs). These CLBs are usually made by look-up table (LUT), D Flip-Flops (DFFs), input/output blocks (IOBs) and multiplexors connected by the so-called routing resources (RRs) which are composed by switch blocks (SBs) and connection blocks (CBs) [14]. Furthermore, SRAM-based FPGA consists of special function models such as block RAMs, digital clock managers (DCMs), digital delay locked loop (DLLs), multipliers etc.

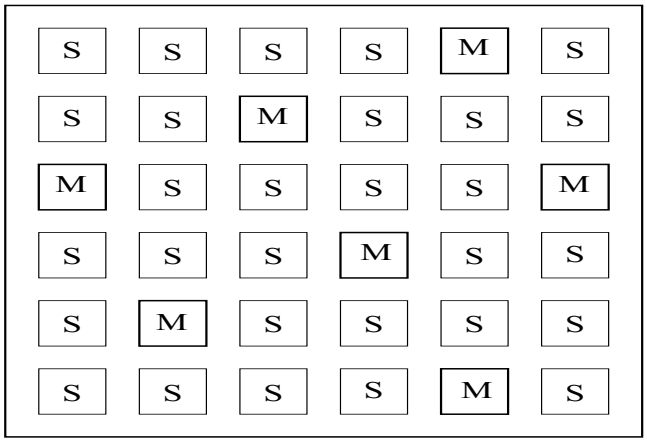

Fig. 1. Proposed SRAM-based FPGA architecture based on MSUs, where each CLB-M (M) is surrounded by four or three CLB-S (S).

The proposed approach requires fault detection and diagnosis method as a preprocessing step. In its current form, our approach suppose that routing resources (RR) are faulty-free and does not address interconnect faults.

The key element of our fault-tolerant approach is partially reconfiguring the SRAM-based FPGA to alternate configuration in response to a fault. If the new configuration implements the same function as the original, while avoiding the faulty hardware block, the system can be restarted. The challenging step is to propose the architecture level of SRAM-based FPGA enabling the efficiency reconfiguration. In this section, we elaborate on the key elements of our approach.

The key concepts of the new approach are master-slaves units (MSU) and fault-tolerance based master-slave technique (FT-based MST).

Definition 1: A MSU is a set composed by three elements: one CLB-M surrounded by two, three or four CLB-Ss, and a specification of how to interface the MSU to adjacent MSUs.

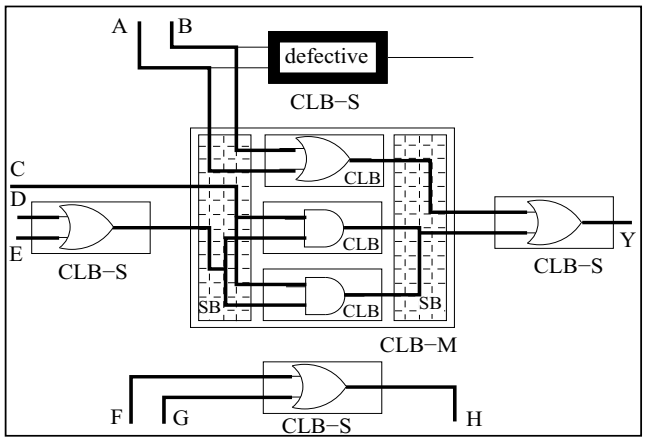

Fig. 3. Overcoming single defect.

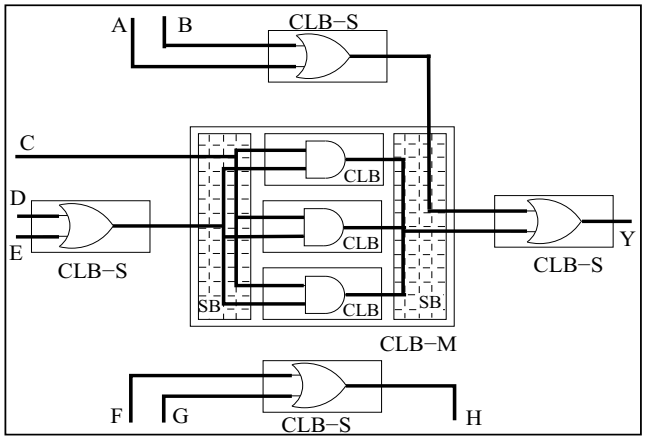

Fig. 2. First implementation in a MSU.

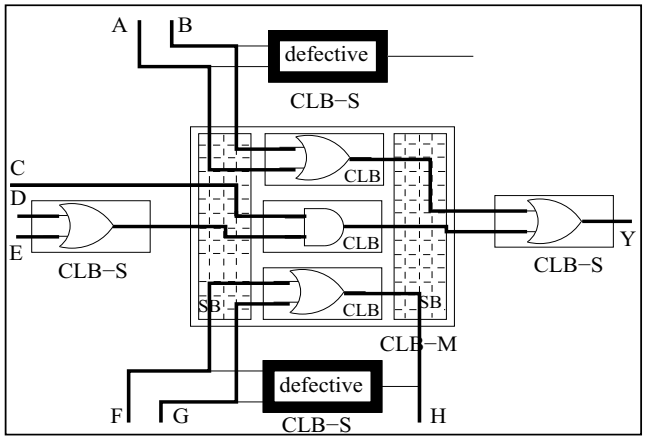

Fig. 4. Overcoming double defects.

Definition 2: A CLB-M encompass three elements: three CLBs with system voting and two switch blocks, a netlist which must be triplicated and placed on those CLB's and routed across the interconnect. The voter let the erroneous outputs of one CLB to be detected then ignored.

. .

Definition 3: A CLB-S is a CLB which in the case of defect or fault, can be tolerated by one instance of the CLB-M using partial reconfiguration.

. .

The Figure.1 illustrates the proposed architecture of SRAM-based FPGA based on MSUs, within each MSU a CLB-M surrounded by two, three or four CLB-Ss. In our approach the interface between the MSU and the surrounding areas is fixed and the MSU's function remains unchanged. In section III we explain how to distribute the CLB-Ms in the proposed architecture.

Consider a boolean function $Y = (A \wedge B) \wedge [C \vee (D \wedge E)]$, which might be implemented in a MSU as shown in Fig.2. This configuration contains a part of the netlist is replicated following the TMR process and placed on the CLB-M, which two CLBs among the three forming the CLB-M are available if a fault should be detected in one or two occupied CLB-Ss. Upon detecting such a fault or a defect, an alternate configuration of the tile is activated which does not rely on the faulty CLB-Ss, this enabled by using partial reconfiguration which presents the cornerstone of our approach. Thus, the concerned MSU avoids the faulty CLB-S and replaced by the adjacent CLB of CLB-M. Fig.3 shows some cases of overcoming single fault, on the other side Fig.4 illustrates a case of tolerating double faults affecting two CLB-Ss within the same MSU. As already mentioned, we assume that the fault-detection and diagnosing system is able to identify the faulty CLB whether a CLB-S or those of CLB-M. The timing of the circuit may vary, however, due to the change in routing.

This approach has three main benefits compared to redundancy-based fault-tolerance: low overhead, the option of runtime management, and complete availability. indeed, the hardware overhead used in the proposed architecture is low: in the FT-based MST technique only one CLB among a set of five CLBs is subject to the TMR method which is comparable to existing methods that assume 25% of the overall number of CLBs [15]. Run-time management can be a very valuable feature of system, particularly for mission-critical applications. This fault-tolerant approach eliminates the need for outside intervention. The availability that this approach provides allows for application specific solutions.

B. FT-based MST repairing algorithm

To repair a permanent fault within the proposed architecture, a three steps algorithm can be used. These steps (shown in Fig.5) can be summarized as follows.

When a CLB-S within a MSU is detected and located as faulty, the repair process starts with the checking mode to verify if the CLBs encompassed in the CLB-M are fault-free.

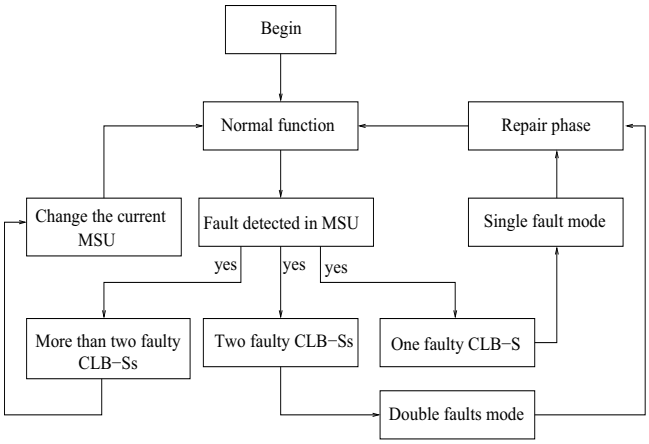

Fig. 5. Adaptive algorithm diagram explaining the FT-based MST repair process.

Then repair process intervenes while passing to the single fault mode.

The second step is activated when two CLB-Ss are detected faulty within the same MSU, then the double faults mode is triggered under the assumption that the CLB-M is fault-free.

If the CLB-M is faulty then the double faults can't take place and if the number of CLB-Ss within the same MSU is more than one, then the repair process forces the system to change the current MSU and replace it by an other fault-free one.

III. Reliability Analysis

A commonly adopted definition of a system reliability is given as follows [17].

Definition 4: The reliability, $R(t)$, of a system is defined as the probability that, when operating under stated environmental conditions, it will perform its intended function adequately in the specified time interval $[0, T]$.

. .

We assume that the reliability of of a CLB is $R_{CLB}(t)$,i.e, the probability that a CLB being faulty is $(1 - R_{CLB}(t))$. It is easy to see that the reliability $R_{init}(t)$ that the original design is faulty-free is.

$$R_{init}(t) = (R_{CLB}(t))^n \tag{1}$$

The reliability provided by the FT-based MST technique can be expressed analytically as follows: define $R_{MST}(t)$ as the overall reliability of a design and define $R_{MSU}(t)$ as the reliability of one MSU. At the high level, the overall reliability is a series expression, i.e.

$$R_{MST}(t) = \left(\prod_{i=1}^{\alpha} R_{MSU_4}(t)\right) \times \left(\prod_{j=1}^{\beta} R_{MSU_3}(t)\right) \times \left(\prod_{k=1}^{\gamma} R_{MSU_2}(t)\right) \tag{2}$$

TABLE I
CHARACTERISTICS AND PERFORMANCE OF THE FT-BASED MST TECHNIQUE.

Architecture ($Row \times Column$)	**# CLBs**	α	β	γ	**# MSU**	**Resource overhead(# CLB)**	**# CLBs not covered**	**Area coverage (%)**
32×32	1,024	180	24	0	204	408	28	97.3
40×40	1,600	289	30	1	319	638	33	98.0
48×48	2,304	423	37	1	460	920	38	98.4
56×56	3,136	583	44	0	627	1,254	43	98.6
104×80 [16]	8,320	1,591	73	0	1,664	3,328	71	99.15

where:

$R_{MSU_4}(t)$, $R_{MSU_3}(t)$, $R_{MSU_2}(t)$ express the reliability of a MSU with two, three and four CLB-S's respectively and α, β and γ are the total number of these MSUs respectively.

As each MSU unit has the same reliability, the overall reliability of the system is:

$$R_{MST}(t) = (R_{MSU_4}(t))^{\alpha} \times (R_{MSU_3}(t))^{\beta} \times (R_{MSU_2}(t))^{\gamma} \quad (3)$$

On the other hand the fact of the presence of two switch blocks (SBs) in the MSU structure gives to that flexibility in the fault tolerance process of such kind. With a reconfiguration of these two switch blocks the structure allows in the case of Fault the tolerance of any CLB whether a CLB-S or a CLB constituting the CLB-M by two possibilities. Thus the reliability of the MSU unit is:

$$R_{MSU}(t) = \sum_{i=0}^{n} \binom{l}{i} R_{CLB}(t)^{l-i}(1 - R_{CLB}(t))^{i} \quad (4)$$

where n is the number of the CLBs which can replace the defective CLBs. l is the total number of CLBs per MSU.

To determine the parameters α, β and γ we consider first that, the SRAM-based FPGA is a matrix with $N \times M$ CLB entries and a coordinate (r,c). For each CLB-M the coordinates (r,c) are given according to the equation (5), where $(r, c) \in \{2, ..., N-1\} \times \{2, ..., M-1\}$. The parameters β and γ are obtained by the same equation where $(r, c) \in \{(i, j)\ i \in \{1; N\}\ and\ j \in \{1, 2, 3, ..., M\}\} \cup \{(i, j)\ j \in \{1, 2, 3, ..., N\}\ and\ j \in \{1; M\}\}$.

$$c(\text{mod } 5) = (2r(\text{mod } 5) + 1)(\text{mod } 5) \quad (5)$$

Notice that the formula (5) has been used first in [13] to locate the positions of spare CLBs, and in [15] the reliability provided by coarse redundancy (CR) method $R_{CR}(t)$ is modeled by the equation:

$$R_{CR}(t) = \sum_{i=0}^{g} \binom{M^2}{i} R_{CLB}(t)^{M^2-i}(1 - R_{CLB}(t))^{i} \quad (6)$$

where M^2 is the total number of the FPGA CLBs and g is the number of columns or rows reserved as spares.

In our assessment of the reliability of the proposed method we have considered that SRAM-based FPGAs as an array of $N \times M$ CLBs, however recent FPGAs has other dedicated blocks as indicated in Section II.

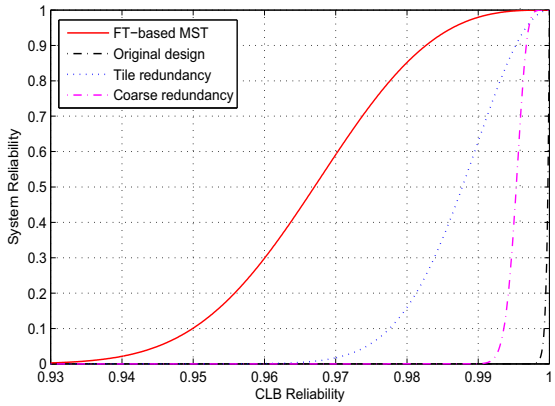

Fig. 6. Reliability of traditional methods versus FT-based MST method for 56×56 SRAM-based FPGA.

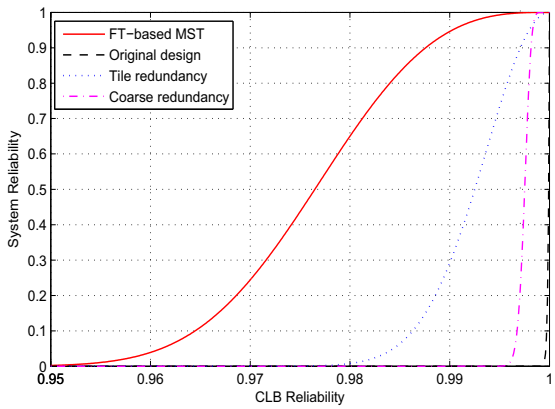

Fig. 7. Reliability of traditional methods versus FT-based MST method for 104×80 SRAM-based FPGA.

IV. NUMERICAL RESULTS

In order to illustrate the effectiveness of the FT-based MST for reliability, the method has been applied to several academic FPGAs (Table. I) and compared with other existing approaches [10] [15]. The reliability of the system is given as a function of CLB's reliability. Characteristics of the proposed method are

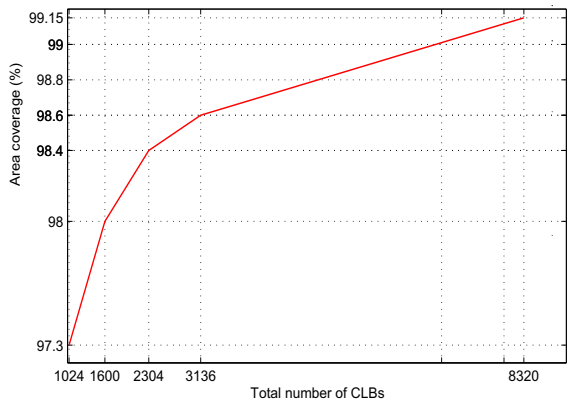

Fig. 8. The area coverage by the FT-based MST method.

given in this section, and we note that all simulations shown in this paper are implemented with Matlab software version 7.5.

Table. I gives the parameters α, β, γ characterizing the number of MSUs with four, three and two CLB-Ss. These parameters are obtained using the expression (5). It can be noted that the parameter α dominates in all studied architectures. Therefore, the dominance of the parameter α compared to β and γ settings allow a reduction of resource overhead. The Table.I gives also the overhead required by the FT-based MST and the area coverage for each studied architecture.

Figures 6-7 present simulation results of reliability of our method and other methods using comparable amount of overhead. In order to better analysis of the reliability we have applied the technique on several academic architectures. Simulation results show the performance payoff that can be expected. Furthermore, these results show that our method reaches its maximum of reliability before the other techniques. That is due to the flexibility of our approach as we deployed two switch blocks in each MSU to make the routing of a design easier.

As shown in Table.I and Fig.8 the FT-based MST technique did not cover all the CLBs located at the edges. This amount of CLBs not covered generally not exceed 2.7% does not affect our strategy because the most FPGA circuits use only 80% of their available configurable logic blocks in order to enhance system routing [13]. Higher utilization of CLBs make the routing of the design more difficult.

V. Conclusion

The repair of permanent faults in SRAM-based FPGAs has been extensively proposed in previous works, however little works has been reported on handling high fault rates for these devices. This paper has proposed a new fault-tolerance approach targeting both single and double faults by exploiting the master-slave technique. Simulation results show that our method compares well to previous work in term of reliability. However, our method with the complexity of the architecture of recent SRAM-based FPGAs stays at a preliminary stage. Detection and location of permanent faults in CLBs are among our perspectives.

Acknowledgment

This study was conducted in the MBSIE project, CPER with the support of the 'Conseil Général de l'Aube' France.

References

[1] K. Banovic, M. Khalid, E. Abdel-Raheem, and P. Banovic, "FPGA-based rapid prototyping of digital signal processing systems," in *Proc. IEEE Int. Midwest Symposium on Circuits & Systems, Cincinnati, OH*, 2005, pp. 647–650.

[2] G. Lum and G. Vandenboom, "Single event effects testing of xilinx FPGA," in *MAPLD 98: Military and Aerospace Applications of Programmable Devices and Technologies Conference*, 1998.

[3] E. Stott, P. Sedcole, and P. Cheung, "Fault tolerant methods for reliability in FPGAs," in *Proc. Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on*, 2008, pp. 415–420.

[4] S. D'Angelo, C. Metra, and G. Sechi, "Transient and permanent fault diagnosis for FPGA-based TMR systems," in *Proc. Defect and Fault Tolerance in VLSI Systems, 1999. DFT'99. International Symposium on*, 1999, pp. 330–338.

[5] S. D'Angelo, C. Metra, S. Pastore, A. Pogutz, and G. Sechi, "Fault-tolerant voting mechanism and recovery scheme for TMR FPGA-based systems," in *Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 1998, pp. 233–240.

[6] H. Asadi and M. Tahoori, "Analytical techniques for soft error rate modeling and mitigation of FPGA-based designs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 12, pp. 1320–1331, 2007.

[7] P. Ostler, M. Caffrey, D. Gibelyou, P. Graham, K. Morgan, B. Pratt, H. Quinn, and M. Wirthlin, "SRAM FPGA Reliability Analysis for Harsh Radiation Environments," *IEEE TRANSACTIONS ON NUCLEAR SCIENCE*, vol. 56, no. 6, p. 3519, 2009.

[8] C. Carmichael, "Triple module redundancy design techniques for Virtex FPGAs," *Xilinx Application Note XAPP197*, vol. 1, 2001.

[9] G. Miller, C. Carmichael, and G. Swift, "Single-Event Upset Mitigation for Xilinx FPGA Block Memories," *XILINX Application Note XAPP962 v1*, vol. 1.

[10] J. Lach, W. Mangione-Smith, and M. Potkonjak, "Low overhead fault-tolerant FPGA systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 2, pp. 212–221, 1998.

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

[12] A. Antola, V. Piuri, and M. Sami, "On-line diagnosis and reconfiguration of FPGA systems," in *Proc. of the The First IEEE International Workshop on Electronic Design, Test and Applications (DELTA'02)*. IEEE Computer Society Washington, DC, USA, 2002, p. 291.

[13] J. Emmert, C. Stroud, and M. Abramovici, "Online fault tolerance for fpga logic blocks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 2, pp. 216–226, 2007.

[14] A. Doumar and H. Ito, "Detecting, diagnosing, and tolerating faults in SRAM-based field programmable gate arrays: a survey," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 3, pp. 386–405, 2003.

[15] S. Pontarelli, M. Ottavi, V. Vankamamidi, A. Salsano, and F. Lombardi, "Reliability Evaluation of Repairable/Reconfigurable FPGAs," in *Proc. 21st IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2006. DFT'06*, 2006, pp. 227–235.

[16] X. Inc, "Spartan-3 1.2 V FPGA family: introduction and ordering information," 2003.

[17] N. Viswanadham, V. Sarma, and M. Singh, *Reliability of computer and control systems*. Elsevier Science Pub. Co. Inc., New York, NY, 1987.