Reliability Evaluation of Repairable/Reconfigurable FPGAs

S. Pontarelli†, M. Ottavi‡, V. Vankamamidi‡, A. Salsano†, F. Lombardi‡
†Dipartimento di Ingegneria Elettronica,
Università di Roma "Tor Vergata", Roma, ITALY
{pontarelli, salsano@ing.uniroma2.it}
‡Department of Electrical and Computer Engineering,
Northeastern University, Boston, USA
{mottavi, vvankama, lombardi@ece.neu.edu}

Abstract

Many techniques have been proposed in the technical literature for repairing FPGAs when affected by permanent faults. Almost all of these works exploit the dynamic reconfiguration capabilities of an FPGA; a subset of the available resources is used as spares for replacing the faulty ones. Initially in this paper, a survey of these techniques is presented; subsequently, a framework is proposed for these techniques by which a fair comparison among them can be assessed and evaluated with respect to reliability. A reliability evaluation is provided for different repair strategies under the assumption that the area overhead is constant. Moreover, considerations about time to repair and feasibility of these techniques are provided. The ultimate goal of the paper is therefore to present the state-of-the-art repair techniques as applicable to FPGA and to establish their performance for reliability.

1 Introduction

FPGAs are widely used for rapid prototyping and realizing low cost, yet complex digital systems. The reprogrammability feature of these chips can be extremely useful to circumvent defects as well as faults. The modular structure of an FPGA allows to reprogram it by replacing defective/faulty logic resources (usually referred to as a block) with a fault-free spares, once detection has occurred. This feature assures a high degree of fault tolerance, even in extremely hostile applications, such as space or radioactive environments. Commercial FPGAs can be fully tested prior to programming. Implementation of off-line and on-line testing is made possible using dedicated capabilities readily applicable in the FPGAs. However, due to the pervasive use of these chips in critical applications, there is also a substantial interest for digital systems with on-line testing capabilities. Permanent and transient faults can be detected and localized using different testing techniques. The allocation of some resources in the FPGA as spares can be used to replace faulty/defective resources once a permanent fault is located. The replacement of faulty resources can be accomplished by reprogramming the FPGA with an alternative configuration that preserves the logical functionality but utilizing a set of fault-free resources (and by excluding the faulty ones). The scheme by which spare resources are allocated inside the FPGA (and consequently the reconfiguration algorithm), is closely dependent on the type of FPGA that is utilized in a specific application. The use of a partial configuration process can drastically reduce both the mean time to repair and the size of the precompiled bistream that is usually stored for the alternative configuration of the FPGA.

The interconnection structure of the FPGA is an important parameter that must be considered for selecting an efficient spare allocation strategy that fully utilizes the FPGA architecture. Different techniques on this topic have been presented in the literature; however, a fair comparison between them and a metric for evaluating their performance has not been fully investigated. The objective of this paper is to present a review of the state-of-the-art repair models available for FPGAs. A reliability assessment of these models is then pursued and finally a comparison of their performances is presented. This paper is organized as follows: Section II presents a generic FPGA architecture

Proceedings of the 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06)
0-7695-2706-X/06 $20.00 © 2006 IEEE
IEEE COMPUTER SOCIETY

and the considered fault model, Section III introduces and analyzes the different repair models. Section IV presents the reliability evaluation of the repair models. A comparison of the models is provided in Section V. Finally, conclusions are drawn in Section VI.

2 FPGA Architecture and Fault model

A FPGA can be viewed as an array of Complex Logic Blocks, CLBs (usually made of by a Look-up table and a Flip-Flop) that are connected by so-called routing resources. Routing resources consist of programmable switch matrices (PSMs), as shown in fig. 1.

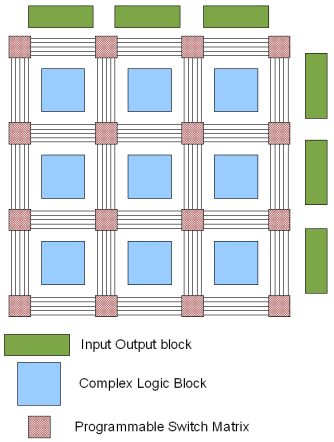


Figure 1. Generic structure of an FPGA

The use of an FPGA in radioactive environments (such as space) may result in the occurrence of faults: the release of charge from high energy particles can induce couples of electron-holes in the substrate, such that current spikes can be generated and transient faults (Single Event Upsets or SEU) may appear; the accumulation and impact of heavy particles can cause lattice modifications, such as displacement or doping, thus permanently modifying the electrical characteristics of the semiconductor material (this effect is commonly known as the Total Ionizing Dose or TID and can cause permanent faults to appear in a circuit, such as stuck-at faults). In this paper, the latter effect is considered and it is assumed that the effect of TID causes the complete failure of the CLB in which this fault appears. Moreover, no TID effects are considered in the routing resources of the FPGA. The objective of this paper is therefore to evaluate the effects of the accumulation of TID in an FPGA as modeled by a failure rate λ. For fault tolerance and reliability of an FPGA, a subset of CLBs is reserved as spares. They are used to replace faulty CLBs, thus preserving the functionality of the implemented system in the presence of faults. The choice of the CLB subset is closely related to the interconnection structure of the FPGA. A survey of interconnection strategies can be found in the technical literature (e.g. [1],[2]); for this paper, some of the assumed conditions are outlined.

- The interconnection structure of fig. 1 is referred to as fully segmented interconnected, because each wire is segmented into sub-wires connecting CLBs on a nearest neighbor basis. To connect two non-neighboring (distant) CLBs, all segments of the wire in the path between the CLBs must be programmed to route the signal across the programmable switch matrices (PSMs). This structure ideally allows for an high level of flexibility in the selection of CLBs for spare allocation; however it has some serious drawbacks. For example, fully segmented interconnections are not commonly used in an FPGA with an high number of CLBs, because performance (in terms of propagation delay) is dependent on the number of traversed PSMs. In a large FPGA, this number can be too high, significantly degrading the performances. Moreover, the reprogramming process for a signal path between PSMs is not a simple task, and routing can be very difficult when the logic replacement of a faulty CLB requires a

Proceedings of the 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06)
0-7695-2706-X/06 $20.00 © 2006 IEEE
IEEE COMPUTER SOCIETY

complicated strategy to find the path.

- Another type of interconnection structure is presented in fig. 2. In this structure, the first hierarchy level of interconnection resources allows to connect the nearest CLBs. For connecting distant CLBs, interconnections of the second level of the hierarchy must be used to reduce the number of PSMs that the signal must traverse. This structure can efficiently utilize the tile and hierarchical spare allocation strategies described in section III. The number of CLBs in a block (tile) and the distribution of spares in the hierarchy are decided as a consequence of the interconnection hierarchy of the FPGA.

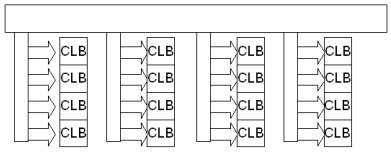


Figure 2. Structure of an FPGA with hierarchical routing

- The last interconnection structure that is reviewed in this paper, is based on a partial (not-fully) segmented approach, in which a wire is divided into sub-wires that span between various CLBs, as shown in fig. 3. This interconnection structure can avoid the problems associated with rerouting using long paths, provided the spare and the faulty CLBs are located on the same segment of the wire. Also in this case, the tile-based approach can be easily utilized and the characteristics of the spare allocation process (as related to the number of CLB and the number of spares in a tile) are determined by the interconnection structure.

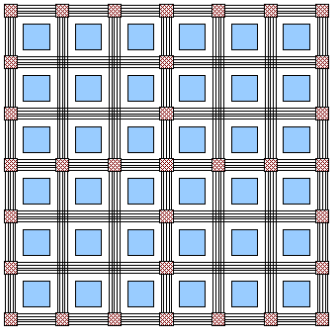


Figure 3. Structure of an FPGA with non segmented routing

Another feature to consider when selecting a repair strategy, is the reprogramming protocol of the FPGA. Two different methods for partial reconfiguration are reviewed in this paper.

- The first method is used by the Atmel AT40K FPGA, [10] and allows to identify the resources of the FPGA to be reprogrammed using a set of control registers. The control registers select the row and the column of the resource to be reconfigured. After programming the control registers, the new configuration of the specified resources can be uploaded using yet another control register. This reprogramming strategy allows for a very fast reprogramming and fine granularity in partial reconfiguration. In [10] this feature is used to realize a tile-based spare allocation for the AT40K FPGA.
- The second reprogramming strategy considered in this paper is the one used by Xilinx; this strategy divides the resources to be reprogrammed in columns, thus providing a granularity higher than the one used by the Atmel FPGAs. The column-based scheme is shown in fig. 4.

This reprogramming strategy allows to implement the spare allocation strategy of [4] and is referred to as coarse repair in the next section.

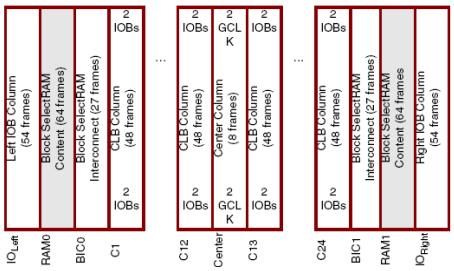


Figure 4. Column-based partial reprogramming of an FPGA

3 Repair models

To repair a permanent fault in an FPGA a four step algorithm can be used. These steps (shown in Fig. 5) can be summarized as follows.

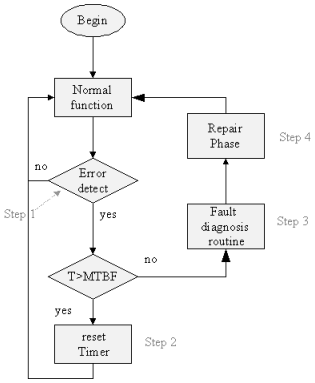


Figure 5. Flow chart of the four-step algorithm

- *Step 1.* The first step of the algorithm deals with detection. Prior to starting reconfiguration, detection of the fault in the FPGA must take place. Detection of a fault is usually achieved by using self-checking circuits. The application to be implemented in the FPGA is divided and embedded into various self-checking circuits to allow the detection of a fault inside a single self-checking unit. The granularity by which the fault is detected, is measured by the number of CLBs (usually in the order of few hundreds). A detailed step for fault location is needed and is performed in the third step of this procedure. Another method to achieve fault detection has been presented in [5]. Detection of a permanent fault is achieved by continuously executing an off-line test on a subset of the FPGA; these CLBs are grouped in blocks referred to as the " roving self-testing areas"' (STARs). The remaining part of the FPGA continues operating as per its normal functionality. After completing the test, the FPGA is reconfigured to perform the off-line test for another subset of STAR, while the application is remapped. This method allows to automatically correct a transient fault on the configuration memory of the FPGA, because the chip is continuously reconfigured, and detection has a very detailed granularity (usually approximately six CLBs). The main drawback of this method is the time between fault occurrence and detection (latency); this depends on the product of the time needed to perform a test on a STAR and the number of subsets in which the FPGA is divided. Due to latency, there is no guarantee of correctness of the implemented functionality.

Proceedings of the 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06)
0-7695-2706-X/06 $20.00 © 2006 IEEE

- *Step 2.* This step allows to discriminate between transient and permanent faults. When the checker detects the occurrence of a fault, a refresh of the configuration memory of the FPGA takes place. This procedure corrects any occurrence of a transient fault in the FPGA. Therefore, as soon as the configuration has been restored, the timer controlling the MTBF is initialized to discriminate permanent from transient faults. Moreover, if two errors are revealed at the same position in a time interval smaller than the MTBF, it is assumed that they are related to the presence of a permanent fault. In this case, Steps 3 and 4 are executed.
- *Step 3.* If a permanent fault is detected, a fault diagnosis routine is executed to locate the fault with a granularity better than the one provided by the partition of the circuit into self-checking units. Various methods for locating a faulty CLB have been proposed in the literature (the interested reader should refer to [6],[7],[8]).
- *Step 4.* In the last step, the replacement of the faulty CLB is performed. The possible repair mechanisms depend strictly on the architecture of an FPGA. Different methods and associated models can be used depending on the partial and dynamic reconfiguration capabilities of the FPGA, the structure of the bitstream for reprogramming the chip and the structure of the interconnection resources. In this paper, they can be differentiated as follows:

  1. *Hierarchical Model:* two hierarchical levels of redundancy: at the lower level the FPGA is organized in tiles, each tile is made of spare CLBs; at a higher level, the faulty tiles can be replaced with spare tiles [9].
  2. *Optimal Model:* the spare CLBs of the FPGA can be used to repair any faulty resource in the device; this represents the best possible case and it does not take into account any of the problems associated with rerouting.
  3. *Coarse redundancy Model:* The used and spare CLBs are lumped in tiles [3] or columns, and they are all allocated for repair [4].
  4. *Tile-based Model:* the FPGA is divided into tiles, each tile contains a spare CLB that can repair only a faulty CLB in a tile [9].

  The hierarchical model is the most general whereas the other three models are effectively subcases. In fact the optimal model can be considered as the lower hierarchical level while the coarse redundancy model has spares only on the higher hierarchy level. Finally, the tile based model has redundancy only at the lower hierarchy level.

3.1 Hierarchical redundancy

This approach refers to the more general case of the repair models as described below. For the reliability analysis of the tile-based approach, some faults are unrepairable. For example, two faults on the same tile of the types shown in Figure 6 can not be repaired. To solve this problem while maintaining the other characteristics, the tile-based approach must utilize additional spare tiles [9]. The spares can be used to facilitate repair in cases such as multiple faults in a tile and faults in the interconnection resources that could not be repaired in the original tile-based approach. This second level of redundancy is effectively a coarse spare allocation and therefore this solution has the characteristics of the coarse and the tile-based approaches.

3.2 Optimal model

In this model, a spare CLB can be used to repair any faulty CLB in the FPGA. The assumption of this model is that rerouting is always possible. This approach is quite independent of the structure of the FPGA to be repaired, but it has some drawbacks in terms of both time to repair and size of the precompiled bitstreams. In this case, the use of precompiled bitstreams is mandatory, because rerouting of the resources can involve the whole FPGA and therefore the complete place-and-route algorithm must be executed. This is very time consuming and in most application it cannot be performed on-line. It must be executed at compile time, and suitable methods must be be developed to reduce the size of the precompiled bitstream. Finally, if this method is applied on an FPGA that does not allow partial reconfiguration, the mean time to repair is the same as other techniques as discussed below.

Proceedings of the 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06)
0-7695-2706-X/06 $20.00 © 2006 IEEE
IEEE COMPUTER SOCIETY

3.3 Coarse redundancy

The spare CLBs are lumped in tiles [3] or columns and are all allocated for repair [4]. When a fault is detected in a column, the whole column is marked as faulty and it is replaced by a spare. This approach exploits the reconfiguration partition of a bitstream as used by Xilinx FPGAs; therefore, the reconfiguration procedure can be performed fast and an algorithm can be developed easily. Moreover, due to the coarse granularity of this approach, the step of the reconfiguration algorithm outlined below can be implemented easily because the level of granularity in the fault location procedure can be lower than the one required in a different approach. The drawback of this solution is that when a fault occurs in a CLB, also other fault-free CLBs in a tile must be marked as unusable.

3.4 Tile-based model

With this technique the FPGA is divided in small partitioned blocks that have fixed interfaces to the others tiles. Diagnosis must locate the faulty resource with a granularity better than the dimension of a tile, such that faulty resources can be replaced with the spares in the tile. Reconfiguration of a tile must preserve the original functionality in new mapping; also, the interconnections between the perimeter of the tile and the remaining part of the FPGA must be unaffected by the configuration process. This technique reduces post-fault-detection downtime, while requiring a small area overhead. Only, the finely located faulty parts of the FPGA are logically removed. The new configurations can be generated at design-time and must be in memory. Each tile is made of a set of FPGA resources (CLBs and interconnections) through an interface specification that defines and binds the interconnections with other tiles in the same FPGA. The use of a tile interface allows not to propagate to other tiles the reconfiguration process for repair, thus reducing the storage overhead. This procedure allows to repair either CLB and local interconnect faults; faults in the global interconnect require a different approach, because this interconnect traverses tiles and their perimeter, thus making tiles dependent on each others. The structure of a tile is dependent on the interconnection structure of the FPGA. In [9], different tile structures have been presented for diverse FPGAs. Figure 6, shows a tile made of four CLBs. Three of the four CLBs are used for processing while the forth CLB is reserved as a spare. When a fault is detected on a CLB, the tile is reconfigured to exclude the faulty CLB. For an FPGA (such as the one used in [10]) a structure similar to Figure 6.b), must be used. The Atmel FPGA uses diagonal interconnections and therefore, the structure proposed in Figure 6.b is better than the one of Figure 6 for assembling a tile (see [10] for details).

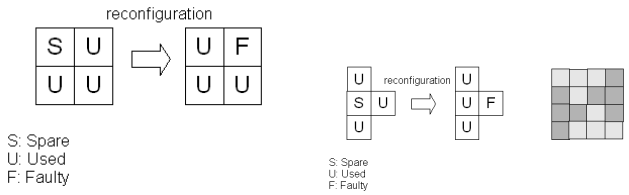


Figure 6. a) A tile forming a block (b) A tile that exploits diagonal interconnections

4 Reliability evaluation

In this section, reliability models for the above introduced repair approaches are proposed. The models are combinatorial and calculate the probability of repairing a fault at time t based on the assumption that the repair process requires a negligible time to execute (this assumption permits to avoid the use of a Markov model that is needed when the repair time is not negligible).

4.1 Hierarchical redundancy

The reliability of hierarchical repair can be considered at two levels. At a high level, the probability that an FPGA with a tile-based approach is operational in the presence of no more than g faulty

tiles (g is also defined as the number of spare tiles in the FPGA). At a low level, the probability of an operational tile is the probability that no more than n CLB are faulty (where n is defined as the number of spare CLBs in a tile).

Therefore at the high level of hierarchy, the probability that the FPGA operates correctly is given by

$$R_{ov}(t) = \sum_{i=0}^{g} \binom{m}{i} R_{tile}(t)^{m-i}(1 - R_{tile}(t))^i$$

where m is the total number of tiles. At low level (similar to the previous case), the reliability is given by

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

where l is the number of CLBs per tile.

4.2 Optimal repair

In this case, the reliability is also a bound. As every CLB can be substituted by a spare, the reliability of an FPGA with N spare CLBs can be expressed as the probability of having up to N faulty CLBs. Consider an FPGA made of M^2 CLBs and define the reliability of a CLB as $R_CLB(t)$. As the probability of failure at time t is $p_f(t) = 1 - R_CLB(t)$, then the reliability is

$$R_{ov}(t) = \sum_{i=0}^{N} \binom{M^2}{i} R_{CLB}(t)^{M^2-i}(1 - R_{CLB}(t))^i$$

4.3 Coarse redundancy

In this model, the allocation of the spare resources is not optimal. A faulty CLB can be substituted by a whole tile, so the reliability of a FPGA with g spare tiles can be expressed as the probability of having up to g faulty CLBs in the FPGA. Consider an FPGA with M^2 CLBs, as above define the reliability of a CLB as $R_CLB(t)$ and consider that the probability of failure at time t is $p_f(t) = 1 - R_CLB(t)$, therefore the overall reliability is:

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

4.4 Tile-based repair

Similar to the hierarchical model, the reliability of tile-based repair consists of two levels. The probability that an FPGA with a tile-based repair approach can be computed as the probability that all tiles are operational (high level) while the probability that a tile is operational is the probability that at most n CLB are operational (n is the number of spare CLBs in a tile) (low level). This can be expressed analytically as follows: define R_{tile} as the reliability of a tile at the high level; the overall reliability is a series expression, i.e.

$$R_{ov}(t) = \prod_{i}^{k} R_{tile}(t) = R_{tile}^{k}(t)$$

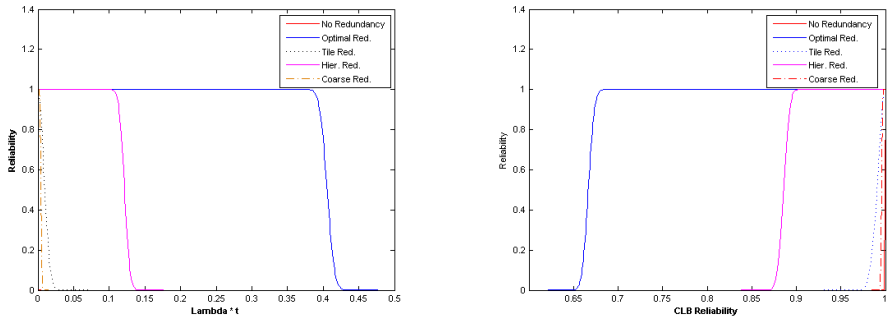
where k is the total number of tiles in the FPGA and the second equality is caused by the reliability of all the tiles being the same. At low level, the reliability of a tile is

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

where l is the number of CLBs per tile. Therefore, by combining these two expressions

Proceedings of the 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06)
0-7695-2706-X/06 $20.00 © 2006 IEEE

IEEE COMPUTER SOCIETY



(a) (b)

Figure 7. Comparison of Repair methods: (a) as function of $\lambda \times t$ (b) as function of $R(t)$

$$R_{ov}(t) = [\sum_{i=0}^{n} \binom{l}{i} R_{CLB}(t)^{l-i}(1 - R_{CLB}(t))^{i}]^{k}$$

5 Comparison

In this sections the repair models are compared using the reliability analysis of the previous section and as function of both $lambda \times t$ and the reliability of a single CLB $R(t)$. Using $lambda \times t$, the results are not a function of a specific value of the failure rate and it allows for a broader analysis. Using $R(t)$, the results are also made independent of the reliability, i.e. this analysis would also account for a non exponential behavior of $R(t)$.

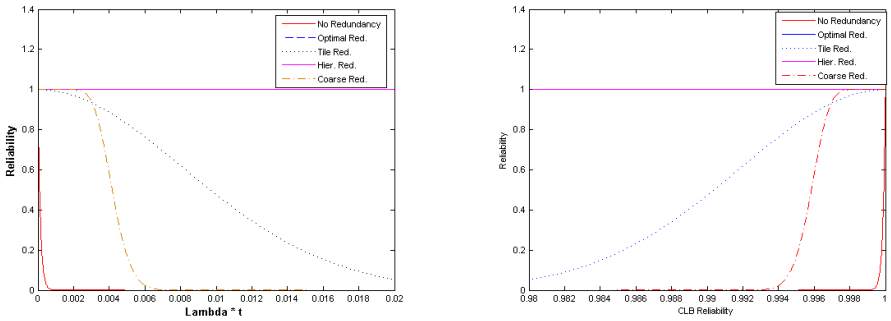
The examples assume that the allocated redundancy is 25% of the overall number of CLBs in the FPGA. In the simulation, the parameters are for a square FPGA with M=100 (number of rows/columns) and thus, the total number of CLBs is $M^2 = 10.000$.

The previously presented repair methodologies are defined as follows:

1. optimal repair: $(3/4) \times M^2 = 7500$ CLBs used and $(1/4) \times M^2 = 2500$ spare CLBs
2. tile repair: each tile has 4 CLBs, therefore there are $1/4 \times M^2 = 2500$ tiles; in a tile three CLBs are used while one CLB is a spare.
3. coarse repair: M=100 columns, each column consists of 100 CLBs, 75 of these columns are utilized and therefore 25 columns are spares.
4. hierarchical repair: each tile has 5 CLBs and therefore there are $1/5 \times M^2 = 2000$ tiles. In a tile, 4 CLBs are used and one CLB is spare (level 1). At level 2, there are 1875 used tiles and 125 spare tiles. The total number of spare CLBs is 125×5+1875=2500

Simulations have been performed using Matlab and plotted to compare the performance of the different methods. Figure 7 shows all repair methods as a function of $lambda \times t$ and $R(t)$; as expected, the reliability obtained using optimal repair outperforms all other methods (hierarchical repair is also a viable alternative).

As a high failure rate or a long mission time are only encountered in some applications, for a better understanding of the behavior of the repair process for small values of $\lambda \times t$ and high values of $R(t)$ the results are shown in Figure 8. Figure 8 shows that for a range of $\lambda \times t$ values (or in general for high values of $R(t)$) the reliability with coarse repair outperforms tile-based repair; however, once $\lambda \times t$ increases ($R(t)$ decreases), then the reliability of the former repair drops (accounting for lack of spares), while the latter repair method smoothly decreases (accounting for a better allocation of spares).



(a) (b)

Figure 8. Comparison of Repair methods: (a) low values of $\lambda \times t$ (b) for high values of $R(t)$

6 Conclusions

The repair of permanent faults in FPGAs has been extensively proposed in previous works, however little comparison has been reported on the different repair techniques made available for FPGAs. This paper has reviewed these repair techniques and provided a comparison of their characteristics by utilizing an uniform reliability model with an equal spare allocation. Different reliability figures have been obtained both in function of the $\lambda \times t$ product and the reliability of a single CLB $R(t)$; this has permitted to evaluate performance; the results have shown the superior performance of an optimal repair model (albeit it has practical limitations due to complex rerouting in terms of execution time and complexity). It has been also shown that the reliabilities for coarse and tile based redundancy techniques offer a balanced trade off that can be assessed with respect to the desired application.

References

[1] Rose, J.; El Gamal, A.; Sangiovanni-Vincentelli, A., "'Architecture of field-programmable gate arrays"' Proceedings of the IEEE, Volume 81, Issue 7, July 1993 Page(s):1013 - 1029

[2] Brown, S.; Rose, J.; "'FPGA and CPLD architectures: a tutorial"' Design & Test of Computers, IEEE Volume 13, Issue 2, Summer 1996 Page(s):42 - 57

[3] A. Antola, V. Piuri, M.Sami "' On-line Diagnosis and Reconfiguration of FPGA Systems"', Proceedings of the First IEEE International Workshop on Electronic Design, Test and Applications (DELTA.02)

[4] Wei-Je Huang; McCluskey, E.J., "Column-Based Precompiled Configuration Techniques for FPGA," Field-Programmable Custom Computing Machines, 2001. FCCM '01. The 9th Annual IEEE Symposium on , vol., no.pp. 137- 146, 2001

[5] Abramovici, M.; Stroud, C.E.; Emmert, J.M.; "Online BIST and BIST-based diagnosis of FPGA logic blocks" IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume 12, Issue 12, Dec 2004 Page(s):1284 - 1294

[6] Wei Kang Huang; Meyer, F.J.; Xiao-Tao Chen; Lombardi, F.;"'Testing configurable LUT-based FPGA's"' IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume 6, Issue 2, June 1998 Page(s):276 - 283

[7] Shnidman, N.R.; Mangione-Smith, W.H.; Potkonjak, M.; "'On-line fault detection for bus-based field programmable gate arrays"' IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume 6, Issue 4, Dec. 1998 Page(s):656 - 666

[8] Wang, S.-J.; Tsai, T.-M.; "'Test and diagnosis of faulty logic blocks in FPGAs"' IEE Proceedings- Computers and Digital Techniques, Volume 146, Issue 2, March 1999 Page(s):100 - 106

[9] Lach, J.; Mangione-Smith, W.H.; Potkonjak, M.; "'Enhanced FPGA reliability through efficient run-time fault reconfiguration"' IEEE Transactions on Reliability, Volume 49, Issue 3, Sept 2000 Page(s):296 - 304

[10] Pontarelli, S.; Cardarilli, G.C.; Malvoni, A.; Ottavi, M.; Re, M.; Salsano, A "'System-on-chip oriented fault-tolerant sequential systems implementation methodology"' IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2001. Proceedings. 2001 24-26 Oct. 2001 Page(s):455 - 460