J Electron Test (2008) 24:105–116
DOI 10.1007/s10836-007-5040-4

# Analysis and Evaluations of Reliability of Reconfigurable FPGAs

**Salvatore Pontarelli · Marco Ottavi · Vamsi Vankamamidi · Gian Carlo Cardarilli · Fabrizio Lombardi · Adelio Salsano**

Received: 30 November 2006 / Accepted: 31 August 2007 / Published online: 17 January 2008

**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 where a subset of the available resources is used as spares for replacing the faulty ones. The choice of the best reconfiguration technique depends on both the required reliability and on the architecture of the chosen FPGA . This paper presents a survey of these techniques and explains how the architectural organization of the FPGA affects the choice of a reconfiguration strategy. 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. To provide a comparison between these techniques FPGAs of different size are taken into account. Also the relationship between the area overhead and the overall reliability has been investigated. Considerations about time to repair and feasibility of these techniques are provided. The ultimate goal of this paper is therefore to present a state-of-the-art repair techniques as applicable to FPGA and to establish their performance for reliability.

**Keywords** Fault model · Reliability · Defect tolerance · FPGA

Responsible Editor: N. A. Touba

S. Pontarelli (✉) · G. C. Cardarilli · A. Salsano
Dipartimento di Ingegneria Elettronica,
Università di Roma "Tor Vergata", Rome, Italy
e-mail: pontarelli@ing.uniroma2.it

A. Salsano
e-mail: salsano@ing.uniroma2.it

M. Ottavi · V. Vankamamidi · F. Lombardi
Department of Electrical and Computer Engineering,
Northeastern University, Boston, USA
e-mail: mottavi@ece.neu.edu

V. Vankamamidi
e-mail: vvankama@ece.neu.edu

F. Lombardi
e-mail: lombardi@ece.neu.edu

## 1 Introduction

FPGAs are widely used for rapid prototyping and realizing low cost, yet complex digital systems. The reprogrammability feature of these chips is extremely useful for circumventing 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 fault-free spares, once detection has occurred. This feature, if correctly used, 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 resources inside the same FPGA. 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. While transient faults can be repaired reprogramming the same resources, permanent faults require the definitive use of spare blocks allocated in the FPGA 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 utilizing a set of fault-free resources and 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 in the selection of an optimized 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 performance is presented. This paper is organized as follows: Section 2 presents a generic FPGA architecture and the considered fault model, Section 3 introduces and analyzes the different repair models. Section 4 presents the reliability evaluation of the repair models. A comparison of the models is provided in Section 5. Finally, conclusions are drawn in Section 6.

## 2 FPGA Architecture and Fault Model

A FPGA can be viewed as an array of complex logic blocks, CLBs, which can be generically described as functional boxes containing a look-up table and a flip-flop. The CLBs are connected by so-called routing resources, consisting of programmable switch matrices (PSMs), as shown in Fig. 1.

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 device, 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 also cause lattice modifications, such as displacement or doping, thus permanently modifying the electrical characteristics of the semiconductor material. This effect, commonly known as the total ionizing dose or TID, can cause permanent faults to appear in a circuit. 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 the 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. The fault tolerance and reliability of the FPGA is obtained using a subset of CLBs 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 [3, 8]; for this paper, some of the assumed conditions are outlined.

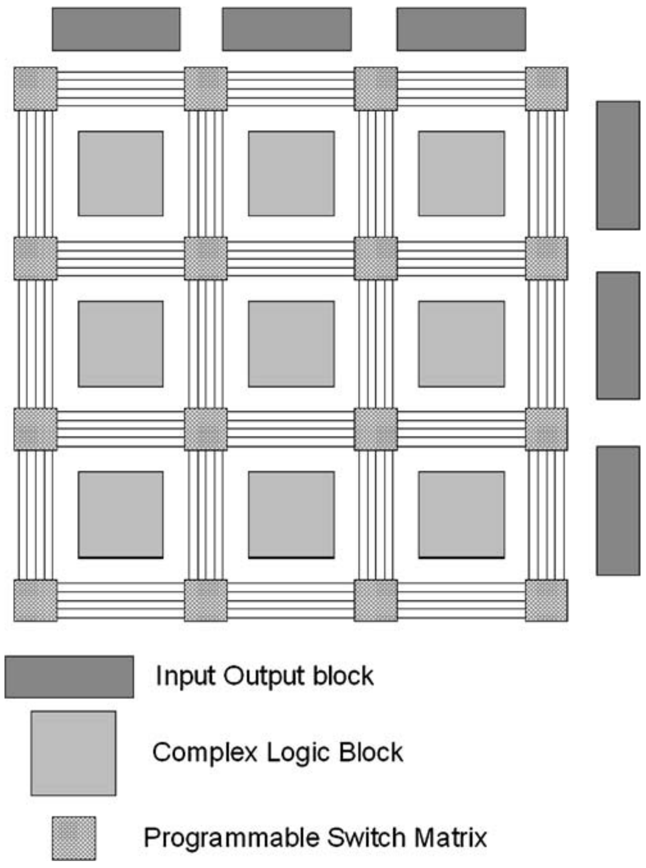


**Fig. 1** Generic structure of an FPGA

- The structure of Fig. 1 is referred to as fully segmented interconnected, because each wire is segmented into sub-wires connecting CLBs on

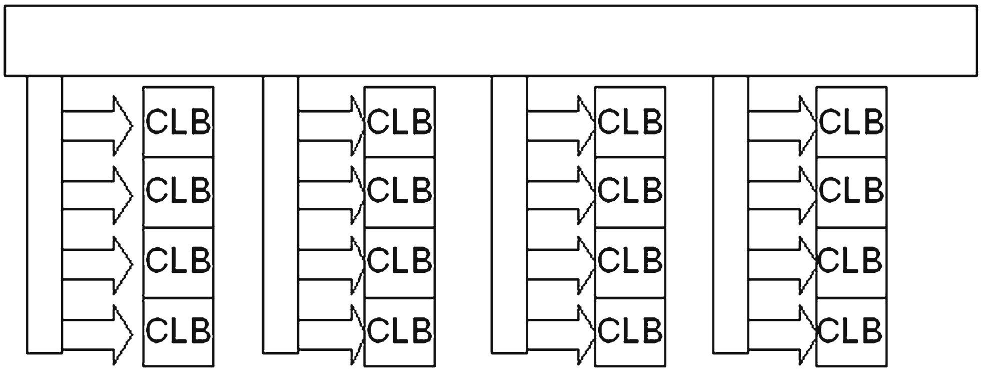


**Fig. 2** Structure of an FPGA with hierarchical routing

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 but, however, it has some 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 high, significantly degrading performance. 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 complicated strategy to find the path.

- Another type of interconnection structure is presented in Fig. 2. In this structure, the first level hierarchy 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 3. 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.
- 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 that the spare and the faulty CLBs are located on the same segment of the wire. The tile-based approach can be easily utilized in this case and the characteristics of the spare allocation process related to the number of CLBs and to the number of spares in a tile are determined by the interconnection structure.

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, [7] 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 a very fast reprogramming and provides a fine granularity in partial reconfiguration. In [7] 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

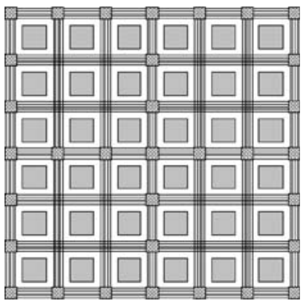
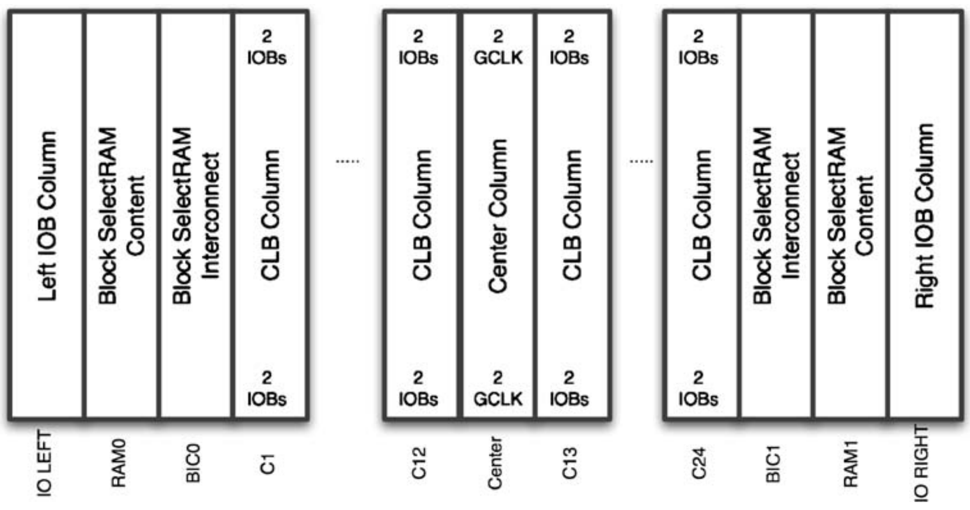


**Fig. 3** Structure of an FPGA with non segmented routing

**Fig. 4** Column-based partial reprogramming of an FPGA



allocation strategy of [4] and is referred to as coarse repair in the next section.

## 3 Repair Models

A four-step algorithm can be used to repair a permanent fault in an FPGA. These steps, shown in Fig. 5 can be summarized as follows.

- *Step 1* The first step of the algorithm deals with fault detection which is clearly preliminary to any reconfiguration. Detection of faults 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 of a self-checking circuit, 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 [1]. Detection of a permanent fault is achieved by continuously executing an off-line test on a subset of the FPGA, consisting in some CLBs 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 the detection has a very high degree of granularity (usually approximately six CLBs). The main drawback of this method is the time between fault occurrence and its 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.
- *Step 2* This step allows to discriminate between transient and permanent faults. When a 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 [5, 9, 10]).
- *Step 4* In the last step, the replacement of the faulty CLB is performed. The possible repair mechanisms strictly depend on the architecture of the 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

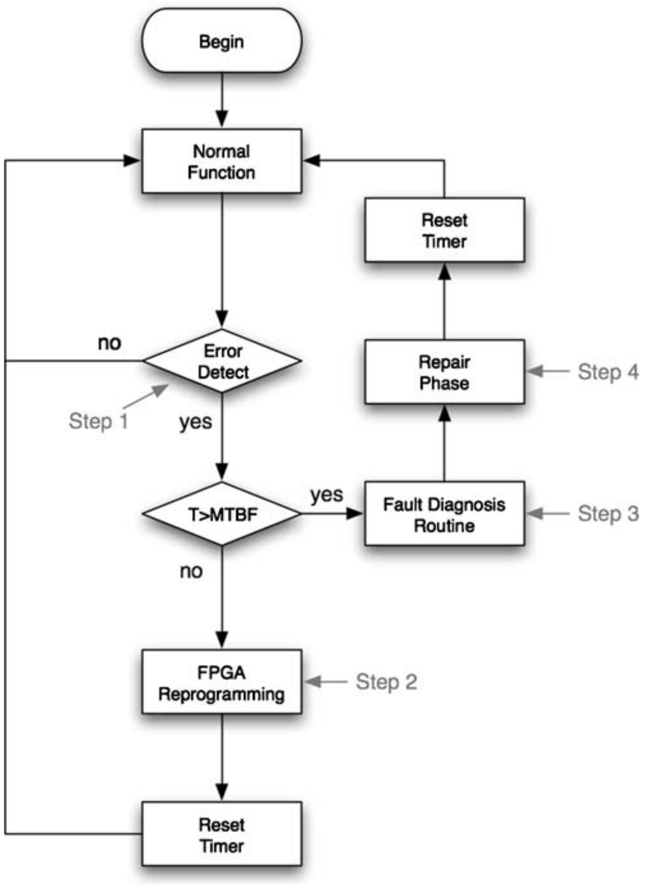


**Fig. 5** Flow chart of the four-step algorithm

structure of the interconnection resources. In this paper, they are differentiated as follows:

1. *Hierarchical Model:* two hierarchical levels of redundancy, at the lower level the FPGA is organized in tiles, each tile includes spare CLBs; at a higher level, the faulty tiles can be replaced with spare tiles [6].
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 [2] 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 the same tile [6].

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

### 3.1 Hierarchical Model

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 Fig. 8 can not be repaired. To solve this problem while maintaining the other characteristics, the tile-based approach must utilize additional spare tiles [6]. 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 both the characteristics of the coarse and the tile-based approaches. An example of a two level hierarchical spare allocation is shown in Fig. 6.

### 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 requirement 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 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.

### 3.3 Coarse Redundancy Model

The spare CLBs are lumped in tiles [2] 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

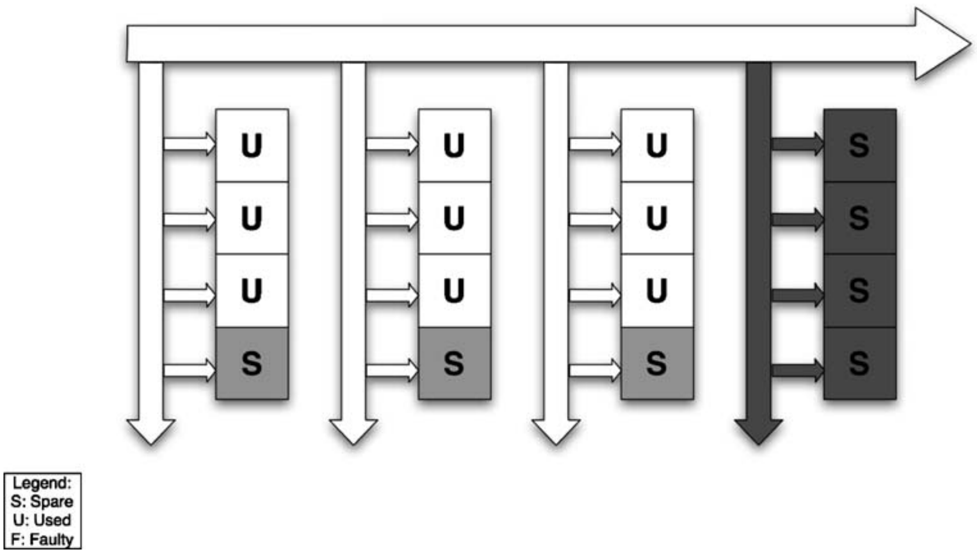


**Fig. 6** Hierarchical spare allocation

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. An example of the spare allocation process for this approach is shown in Fig. 7.

### 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 the new mapping; also, the interconnections between the perimeter of the tile and the remaining part of the FPGA must be unaffected by the reconfiguration 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, while faults in the global interconnect require a different approach, since this interconnect traverses tiles and their perimeter, thus making tiles dependent on each

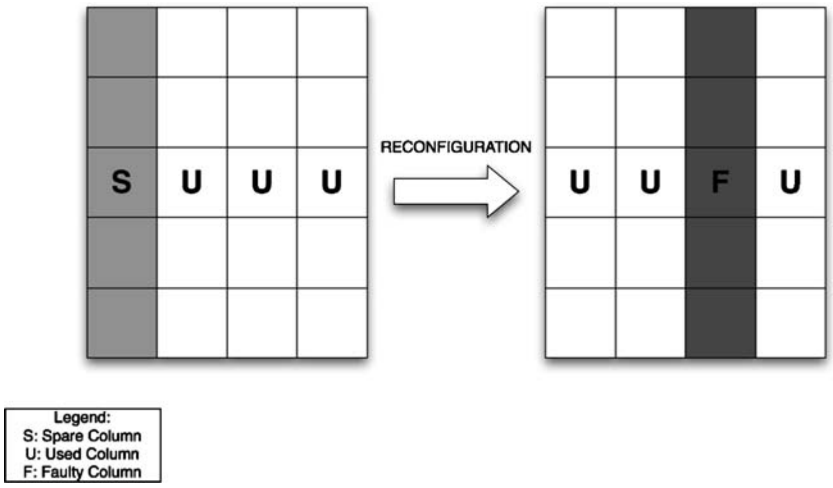


**Fig. 7** Spare allocation for coarse redundancy

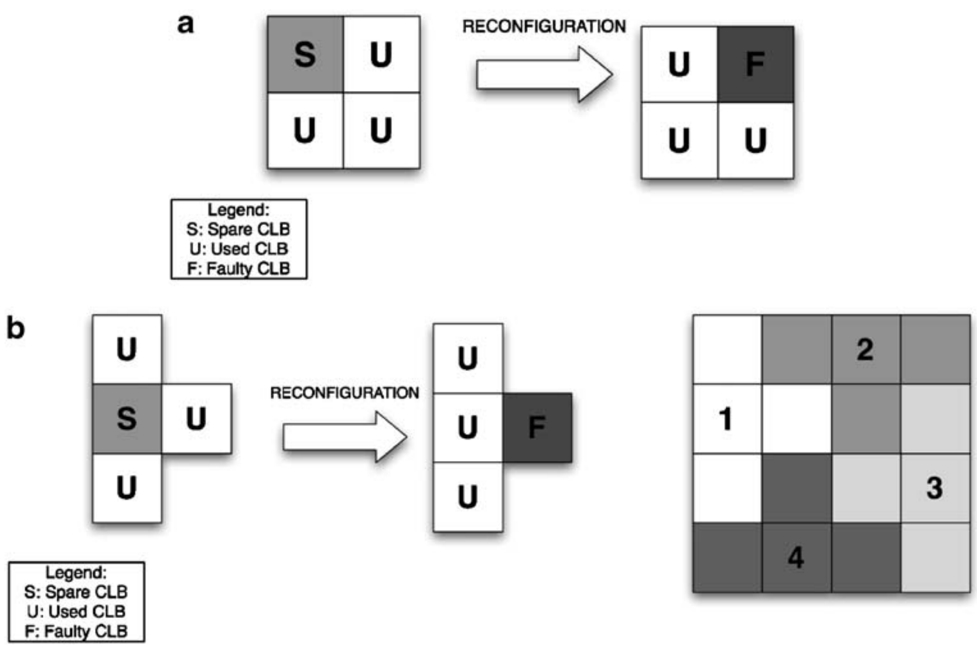


**Fig. 8** **a** A tile forming a block. **b** A tile that exploits diagonal interconnections

others. The structure of a tile is dependent on the interconnection structure of the FPGA. In [6], different tile structures have been presented for diverse FPGAs. Figure 8, 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 [7]) a structure similar to Fig. 8b, must be used. The Atmel FPGA uses diagonal interconnections and therefore, the structure proposed in Fig. 8b is better than the one of Fig. 8a for assembling a tile (see [7] for details).

## 4 Reliability Evaluation

Reliability models for the above introduced repair approaches are proposed in this section. The models are combinatorial and compute the probability of repairing a fault at time $t$ based on the assumption that the repair process requires a negligible execution time (this assumption permits to avoid the use of a Markov model that is needed when the repair time is not negligible and race conditions between the occurrence of a second fault and the repair are taken into account). The analysis of reliability is performed based on the assumption that all CLBs have the same reliability $R_{\text{CLB}}$.

### 4.1 Hierarchical Repair

The reliability of hierarchical repair can be considered at two levels. At the 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 the low level, the probability that a tile is operational 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

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

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

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

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 $R_{\text{CLB}}(t)$. As the probability of failure at time t is $p_f(t) = 1 - R_{\text{CLB}}(t)$, then the reliability is

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

### 4.3 Coarse Redundancy Repair

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 tiles in the FPGA. A faulty tile is a tile in which at least one CLB is faulty. Consider an FPGA with $M^2$ CLBs; as above define the reliability of a CLB as $R_{\mathrm{CLB}}(t)$ and consider a tile composed by $M$ CLBs, then the reliability of the tile can be expressed as

$$R_{\mathrm{tile}}(t) = R_{\mathrm{CLB}}(t)^M$$

and the probability of failure at time t is $p_f(t) = 1 - R_{\mathrm{tile}}(t)$. Therefore, the overall reliability is:

$$R_{ov}(t) = \sum_{i=0}^{g} \binom{M}{i} R_{\mathrm{tile}}(t)^{M-i}(p_f(t))^i$$

Note that the expression for the coarse redundancy repair can be derived from the reliability of the hierarchical based repair by having the number of spare CLBs in a tile as $n = 0$.

### 4.4 Tile-Based Repair

As for the hierarchical model, the reliability of tile-based repair consists of two levels. The probability that an FPGA with a tile-based repair approach is operational 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). The analytical expression of the reliability can be expressed as follows: define $R_{\mathrm{tile}}$ as the reliability of a tile at the high level; the overall reliability is

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

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

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

where $l$ is the number of CLBs per tile. Therefore, by combining these two equations we obtain

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

## 5 Comparison

In this paper the repair models are compared using the reliability analysis of the previous section as the function of $\lambda \times t$: therefore the results are not a function of a specific value of the failure rate and a broader analysis is possible. The reliability of a CLB can vary as related to its implementation, i.e. the logic functions and the set of used interconnects. In this manuscript however, a constant average failure rate $\lambda$ is assumed. This assumption ensures that the proposed analysis has general applicability thus avoiding functional and FPGA structural dependencies. Moreover, this assumption is applicable to very large circuits (made of a many thousand of CLBs), such as those used as examples in this manuscript.

The examples assume that the allocated redundancy are 25, 12.5 and 6.25% of the overall number of CLBs in the FPGA. These ratios correspond to 1 spare for each 4, 8, 16 CLBs respectively. In the simulation, the parameters are for a square FPGA with $M = 64$ (number of rows/columns) and thus, the total number of CLBs is $M^2 = 4,096$.

The previously presented repair methodologies are defined as follows:

1) Optimal repair: the number of used CLBs are 3,072, 3,584, 3,840, while the number of spare CLBs are 1,024, 512, 256.
2) Tile repair: each tile has 4, 8, 16 CLBs, therefore there are 1,024, 512, 256 tiles in total; one CLB is used as spare in each tile.
3) Coarse repair: $M = 64$ columns, each column consists of 64 CLBs, 48, 56, 60 of these columns are utilized and 16, 8, 4 columns are used as spares.
4) Hierarchical repair: each tile has 8, 16, 32 CLBs and therefore there are 512, 256, 128 tiles. At level 1 one CLB is used as spare. At level 2, there are 64, 16, 4 tiles that are used as spares. The total number of spare CLBs is the same as in the previous cases i.e. 1,024, 512, 256.

Simulations have been performed using Matlab and plotted to compare the performance of the different methods. Figure 9 shows all repair methods as a function of $\lambda \times t$ with a redundancy of 25%. As expected, the reliability obtained using optimal repair outperforms all other methods, hierarchical repair is still a viable alternative.

A high failure rate or a long mission time are only encountered in some applications, so for a better understanding of the behavior of the repair process for small values of $\lambda \times t$ the results are shown in Fig. 9b.

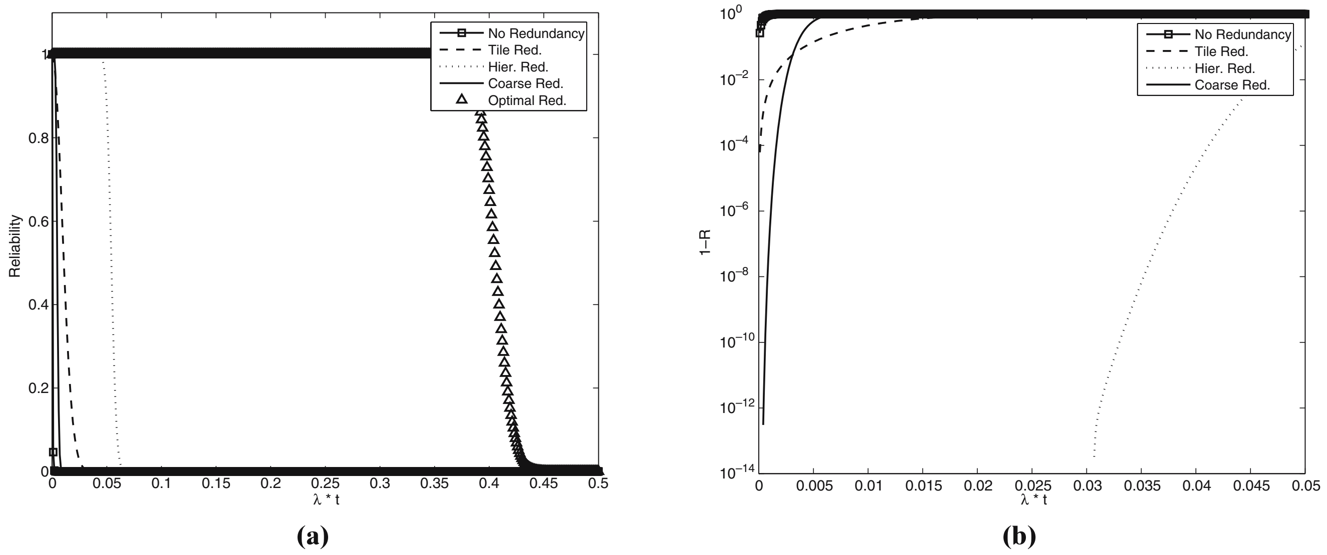


**Fig. 9** Comparison of repair methods with 25 % of spare **a** with high $\lambda \times t$, **b** with low $\lambda \times t$

The failure probability $1 - R$ is plotted on the Y-axis in a logarithmic scale.

Figure 9b shows that for a range of $\lambda \times t$ values the reliability with coarse repair outperforms tile-based repair; however, once $\lambda \times t$ increases, then the reliability of the former repair method drops (accounting for the lack of spares), while the latter repair method smoothly decreases (accounting for a better allocation of spares). With a high number of spare resources, the tile-based repair method shows a better behavior only in the range in which the overall reliability is less than 0.955. This value is unacceptable for many high reliable application. Hence, tile-based repair can not be used in practice, because it shows a better behavior only in applications with no demanding reliability features. The choice is between the coarse redundancy and the hierarchical schemes. The first scheme can be used if the reliability of a CLB is high (low $\lambda \times t$ values), while the second scheme must be used if the reliability of a CLB is low. For a lower percentage of redundancy the same plots have been drawn for 12.5% (Fig. 10) and for 6.25% (Fig. 11) of spares. The behavior of the plots for the different spare allocation schemes is the same; however, the cross-point between the coarse and tile allocation schemes changes. In particular, when the redundancy is 12.5% the cross-point is at $R = 0.992$. This level of redundancy can be useful in some applications and therefore the tile-based allocation scheme

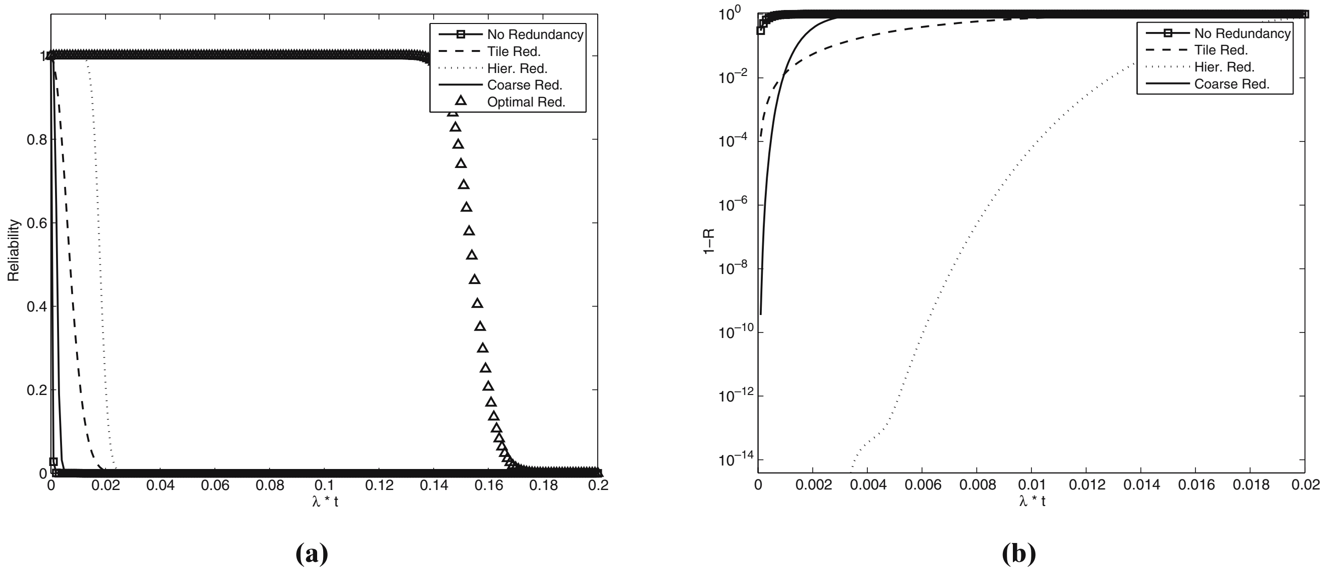


**Fig. 10** Comparison of repair methods with 12.5 % of spare **a** with high $\lambda \times t$ **b**, with low $\lambda \times t$

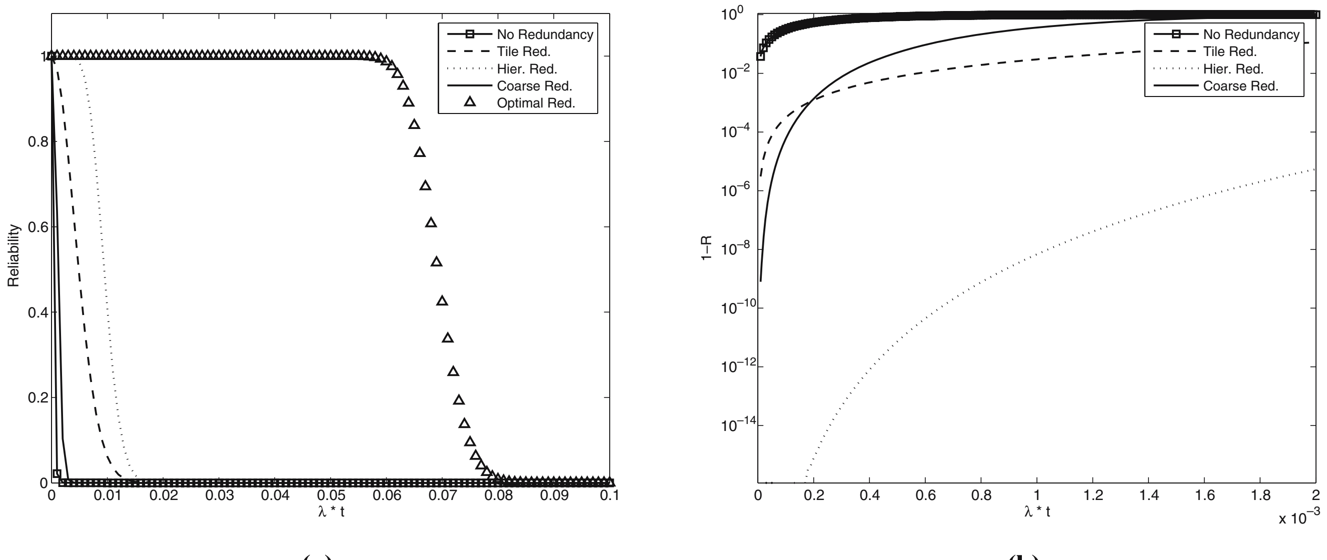


**Fig. 11** Comparison of repair methods with 6.25 % of spare **a** with high $\lambda \times t$, **b** with low $\lambda \times t$

can be the appropriate choice for applications in which the reliability of the single CLB is high, while the probability of failure must be less than 1%. For a redundancy level of 6.25% (see Fig. 11) the cross-point between the coarse and the tile allocation schemes is $R = 0.9991$, and therefore the tile based repair scheme can be used also if the probability of failure is less than $10^{-3}$. In Fig. 11a the cross-point between the tile and the hierarchical schemes can be observed. This cross-point occurs at a reliability value of 0.2 and therefore can not be utilized in practice. Finally, we repeat the reliability analysis with a redundancy level of 6.25% but changing the total number of CLBs in the FPGA. We focus our attention on this case because all the allocation schemes can be used in practical applications with this level of redundancy. Hierarchical and optimal schemes can be used for FPGAs with low reliable CLBs; coarse and tile schemes can be used for FPGAs with CLBs of better reliability. The choice between these schemes depends on the required level of reliability of the overall system. These results are plotted in Fig. 12a for $M = 128$ and Fig. 12b for $M = 256$.

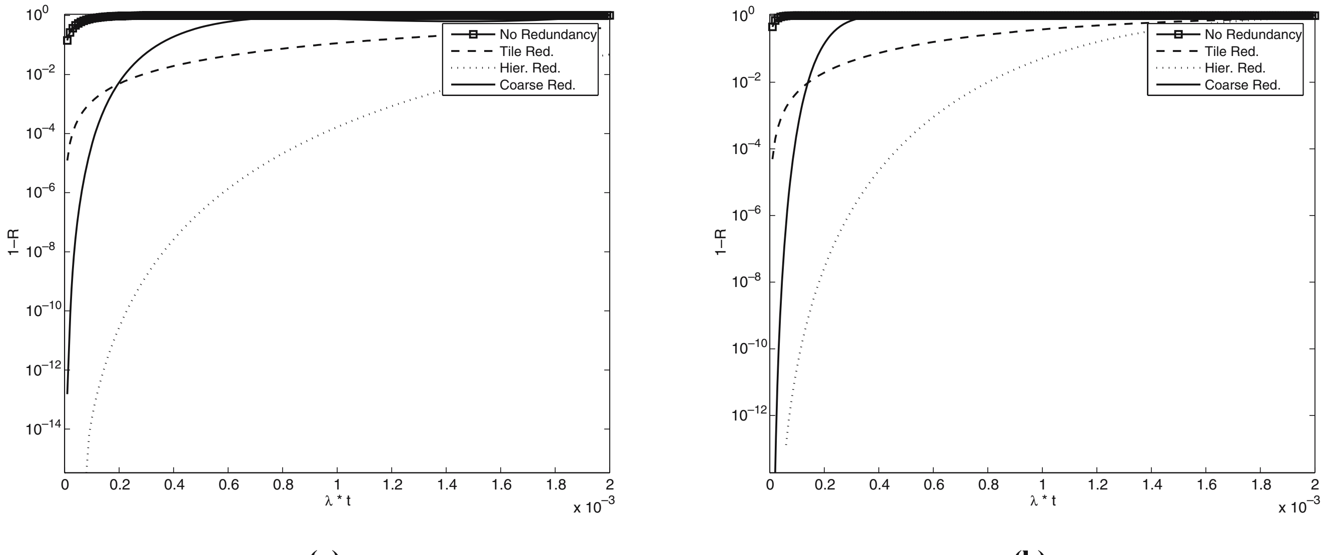


**Fig. 12** Comparison of repair methods with 6.25% of spares **a** with $M = 128$, **b** $M = 256$

These plots show that for bigger FPGAs the reliability cross-point of the coarse and the tile schemes becomes higher. Again, in these cases the use of the tile approach is limited to applications that do not require high reliability.

## 6 Conclusion

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 available for FPGAs. This paper reviews these repair techniques and provides a comparison of their characteristics by utilizing an uniform reliability model with an equal spare allocation. Reliability has been calculated under different overheads and size of FPGAs. This approach has permitted to evaluate performance; the results have shown the superior performance of an optimal repair model (albeit it has practical limitations due to the complex rerouting in terms of execution time and complexity). It has been also shown that the reliability for coarse and tile based redundancy techniques offers a balanced alternative that can be assessed with respect to the desired application. The choice between hierarchical, coarse and tile based approaches can be done as function of the reliability of the single CLB as well as of the overall required reliability level of the FPGA.

## References

1. Abramovici M, Stroud CE, Emmert JM (2004) Online BIST and BIST-based diagnosis of FPGA logic blocks. IEEE Trans Very Large Scale Integr (VLSI) Syst 12(12):1284–1294, December
2. Antola A, Piuri V, Sami M On-line Diagnosis and Reconfiguration of FPGA Systems. Proceedings of the First IEEE international workshop on electronic design, test and applications (DELTA.02)
3. Brown S, Rose J (1996) FPGA and CPLD architectures: a tutorial. IEEE Des Test Comput 13(2):42–57, Summer
4. Huang W-J, McCluskey EJ (2001) Column-based precompiled configuration techniques for FPGA. Field-programmable custom computing machines, 2001. FCCM '01. The 9th Annual IEEE Symposium, pp 137–146
5. Huang WK, Meyer FJ, Chen X-T, Lombardi F (1998) Testing configurable LUT-based FPGA's. IEEE Trans Very Large Scale Integr (VLSI) Syst 6(2):276–283, June
6. Lach J, Mangione-Smith WH, Potkonjak M (2000) Enhanced FPGA reliability through efficient run-time fault reconfiguration. IEEE Trans Reliab 49(3):296–304, September
7. Pontarelli S, Cardarilli GC, Malvoni A, Ottavi M, Re M, Salsano A (2001) System-on-chip oriented fault-tolerant sequential systems implementation methodology. Proceedings in IEEE international symposium on defect and fault tolerance in VLSI systems, pp 455–460, 24–26, October 2001
8. Rose J, El Gamal A, Sangiovanni-Vincentelli A (1993) Architecture of field-programmable gate arrays. Proceedings of the IEEE 81(7):1013–1029, July
9. Shnidman NR, Mangione-Smith WH, Potkonjak M (1998) On-line fault detection for bus-based field programmable gate arrays. IEEE Trans Very Large Scale Integr (VLSI) Syst 6(4):656–666, December
10. Wang S-J, Tsai T-M (1999) Test and diagnosis of faulty logic blocks in FPGAs. IEE Proc Comput Digit Tech 146(2): 100–106, March

**Salvatore Pontarelli** is currently postdoctoral research associate at the University of Rome, Tor Vergata. He received the Laurea degree in Electronic Engineering from the University of Bologna in 1999 and the Ph.D. in Microelectronics and Telecommunications Engineering from the University of Rome Tor Vergata in 2003. His research mainly focuses on fault tolerance, on-line testing and reconfigurable digital architectures.

**Marco Ottavi** is currently with AMD. He previously held post doc positions with Sandia National Laboratories and with the ECE Department of Northeastern University in Boston. He received the Laurea degree in Electronic Engineering from University of Rome "La Sapienza" in 1999 and the Ph.D. in Microelectronics and Telecommunications from University of Rome "Tor Vergata" in 2004. In 2000 he was with ULISSE Consortium, Rome as designer of digital systems for space applications. In 2003 he was visiting research assistant at ECE Department of Northeastern University. His research interests include yield and reliability modeling, fault-tolerant architectures, on-line testing and design of nano scale circuits and systems.

**Vamsi Vankamamidi** received the B.S. degree in computer engineering from the University of Mumbai, Mumbai, India, in 2000, and the M.S. degree in electrical engineering and computer science from the University of Toledo, Toledo, OH, in 2001. He is currently working toward the Ph.D. degree in computer engineering in the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA. As part of his dissertation, he is working on quantum-dot cellular automata, which is a nanoscale device architecture to supersede the conventional silicon-based technology. His research interests include design of nanoscale circuits and systems, electronic design automation, defect tolerance, and reliability.

**Gian Carlo Cardarilli** received the Laurea (summa cum laude) in 1981 from the University of Rome La Sapienza. He works for the University of Rome Tor Vergata since 1984. At present he is full professor of Digital Electronics and Electronics for Communication Systems at the University of Rome Tor Vergata. During the years 1992–1994 he worked for the University of L Aquila. During the years 1987–1988 he worked for the Circuits and Systems team at EPFL of Lausanne (Switzerland). Professor Cardarilli interests is in the area of VLSI architectures for Signal Processing and IC design. In this field he published over 140 papers in international journals and conferences. He also participated to the work group of JESSISMI for the support to the medium and small industries. For this structure he consulted different SMIs, designing a number ASICs, in order to introduce the microelectronics technology in the industry's products. He has also regular cooperation with companies like Alenia Aerospazio, Rome, Italy, STM, Agrate Brianza, Italy, Micron, Avezzano, Italy, Ericsson Lab, Rome, Italy and with a lot of SMEs. Scientific interests of Professor Cardarilli concern the design of special architectures for signal processing. In particular, he works in the field of computer arithmetic and its application to the design of fast signal digital processor. He also developed mixed-signal neural network architectures implementing them in silicon technology. Recently, he also proposed different new solutions for the implementation of fault-tolerant architectures.

**Fabrizio Lombardi** graduated in 1977 from the University of Essex (UK) with a B.Sc. (Hons.) in Electronic Engineering. In 1977 he joined the Microwave Research Unit at University College London, where he received the Master in Microwaves and Modern Optics (1978), the Diploma in Microwave Engineering (1978) and the Ph.D. from the University of London in 1982.

He is currently the holder of the International Test Conference (ITC) Endowed Professorship at Northeastern University, Boston. At the same Institution during the period 1998–2004 he served as Chair of the Department of Electrical and Computer Engineering. Prior to Northeastern University he was a faculty member at Texas Tech University, the University of Colorado-Boulder and Texas A&M University.

Dr. Lombardi has received many professional awards: the Visiting Fellowship at the British Columbia Advanced System Institute, University of Victoria, Canada (1988), twice the Texas Experimental Engineering Station Research Fellowship (1991–1992, 1997–1998) the Halliburton Professorship (1995), the Outstanding Engineering Research award at Northeastern University (2004) and an International Research award from the Ministry of Science and Education of Japan (1993–1999). Dr. Lombardi was the recipient of the 1985/86 Research Initiation award from the IEEE/Engineering Foundation and a Silver Quill award from Motorola-Austin (1996).

Dr. Lombardi was an Associate Editor (1996–2000) of IEEE Transactions on Computers and a Distinguished Visitor of the IEEECS (1990–1993 and 2001–2004). Since 2000, he has been the Associate Editor-In-Chief of IEEE Transactions on Computers and an Associate Editor of the IEEE Design and Test Magazine. Since 2004 he serves as the Chair of the Committee on "Nanotechnology Devices and Systems" of the Test Technology Technical Council of the IEEE.

Dr. Lombardi has been involved in organizing many international symposia, conferences and workshops sponsored by professional organizations as well as guest editor of Special Issues in archival journals and magazines such as the IEEE Transactions on Computers, IEEE Transactions on Instrumentation and Measurement, the IEEE Micro Magazine and the IEEE Design & Test Magazine. He is the Founding General Chair of the IEEE Symposium on Network Computing and Applications.

His research interests are testing and design of digital systems, quantum and nano computing, ATE systems, configurable/network computing, defect tolerance and CAD VLSI. He has extensively published in these areas and edited six books.

**Adelio Salsano** was born in Rome on December 26, 1941 and is currently full professor of Microelectronics at the University of Rome, Tor Vergata where he teaches the courses of Microelectronics and Electronic Programmable Systems. His present research work focuses on the techniques for the design of VLSI circuits, considering both the CAD problems and the architectures for ASIC design. In particular, of relevant interest are the research activities on fault tolerant/fail safe systems for critical environments as space, automotive etc.; on low power systems considering the circuit and architectural points of view; and on fuzzy and neural systems for pattern recognition. An international patent and more than 90 papers on international journals or presented in international meetings are the results of his research activity. At present he is the President of a national consortium named U.L.I.S.S.E., between ten universities, three polytechnics and several of the biggest national industries, as STMicroelectronics, ESAOTE, FINMECCANICA. He is responsible for contracts with the ASI, Italian Space Agency, for the evaluation and use in space environment of COTS circuits and for the definition of new suitable architectures for space applications. Professor Salsano is also involved in professional activities in the field of information technology and is also consultant of many public authorities for specific problems. In particular, he is consultant of the Departments of the Research and of the Industry, of IMI and of other authorities for the evaluation of industrial public and private research projects. Professor Salsano was a member of the consulting Committee for Engineering Sciences of the CNR (National Research Council) from 1981 to 1994 and participated in the design of public research programs in the fields of Telematics, Telemedicine, Office Automation, Telecommunication and, recently, Microelectronics and Bioelectronics.