Novel Technique for Testing FPGAs

C. Metra G. Mojoli S. Pastore D. Salvi G. Sechi

DEIS, U. Bologna
V. Risorgimento 2
40136 Bologna (Italy)

Ist. Fisica, U. Milano
V. Celoria 16
20133 Milano (Italy)

IFCTR, CNR
V. Bassini 15
20133 Milano (Italy)

Abstract

This paper presents a novel technique for testing Field Programmable Gate Arrays (FPGAs), suitable to be used in case of frequent FPGA reuse and rapid dynamic modifiability of the implemented function.

1 Introduction

Field Programmable Gate Arrays (FPGAs) are currently widely used for rapid prototyping and hardware emulation of VLSI chips, as well as manufacturing of complex digital systems [1, 2, 3]. Testing FPGAs is a quite complex problem, which has been widely addressed in the literature [4, 2, 3, 5, 6, 7, 8, 9, 10].

In particular, FPGAs can be used for designing prototypes of systems (e.g. that in [11]) whose correct operation is a critical factor for the evaluation and experimentation of new architectures. In these cases, it may be necessary to evolve the system functionalities according to experiment needs, or to reuse the same FPGAs in alternative solutions to be experimented on the field. This scenario is typical of several physic and space applications [12, 11, 13].

However the frequent reuse of the same FPGA (in the case of the adoption of *pin grid* packages) or its dynamic functional modification (in case of *surface mounting*) makes it very prone to faults. For instance, some faults may occur that are due to the experiments where the FPGA is used (e.g. environmental-induced faults) or to the exploited FPGA's extreme operating conditions (e.g. in terms of frequency, current density, etc.).

Thus, a testing procedure ensuring the reliable FPGAs' reuse and functional modifiability is needed. This problem can be considered as an intermediate problem between those constituting the FPGA testing problem [9]: 1) finding a "manufacturing test procedure (MTP)"; 2) finding a "user test procedure (UTP)". In fact, on the one hand, MTP is generally too much time consuming and possibly also uselessly exhaustive to be continuously applied between following FPGA reuses, or functional modifications. Reversely, the application of UTPs, specified for each FPGA functional modification, would be too much expensive and, likely, uselessly costly, because the functional modifications introduced during FPGA reuse may be not very significant. Moreover, it is important that the number of required FPGA I/O pins is as lower as possible, thus enabling the on-board FPGA testing.

On the basis of these considerations, this paper proposes a novel testing procedure that requires low time and no FPGA I/O pin occupation for testing FPGA CLBs with respect to a wide set of faults representative of realistic failures possibly occurring during reuse. While testing the various CLBs, our procedure enables to test also a significant fraction of all available routing resources. However, in order to exhaustively test all routing resources, one of the techniques in [14, 7, 15, 9] should be used. Our testing procedure enables also the faulty FPGA diagnosis (by simply memorizing the location of the faulty CLB), thus allowing to reprogram the FPGA in order to tolerate faults. Based on these characteristics, the proposed procedure will be hereafter referred to as Reuse-oriented Testing and Diagnosis Procedure (RTDP).

Compared to the MTPs most recently presented in the literature for CLBs' testing [3, 16], RTDP requires a shorter "global testing time" (i.e. the time necessary to configure the FPGA for testing and to perform testing). In particular, the obtained relative reduction has been estimated equal 66% and 75% with respect to [3] and [16], respectively. Moreover, similarly to the technique in [16], RTDP does not require any I/O pin (thus allowing the FPGA on-board testing). If testing and diagnosis is desired, RTDP enables a time reduction equals $56 \div 85\%$ with respect to the technique in [6], recently shown [16] to be possibly used for both testing and diagnosis. Compared to a possible UTP (e.g. that in [10]), RTDP tests the unconfigured FPGA, thus allowing reliable dynamic modifications of the implemented function.

RTDP has been developed and implemented considering the XILINX XC4000 FPGA family. All test facilities necessary for the application of the proposed procedure (i.e. test vector generator, response analyser, etc.) have been implemented to test the FPGAs currently under (dynamic) use for the development of prototypes for a space-oriented project (*PHOCA* project [12]).

This paper is organized as follows. In section 2, a general overview, recalling FPGA structure and state of the art in FPGA testing is introduced. In section 3, RTDP is described. In section 4, the testing and/or diagnosis time of RTDP is evaluated and compared to existing approaches. In section 5, the test facilities used for the practical application of the proposed technique are described, while some conclusive remarks are drawn in section 6.

2 FPGA overview

FPGAs are programmable logic devices consisting of several: i) Configurable Logic Blocks (CLBs), interconnected by a programmable hierarchy of routing resources; ii) programmable Input/Output Blocks (IOBs). The CLBs provide the functional elements for constructing the user's logic, while the IOBs constitute the interface between the package pins and internal signal lines. The programmable interconnect resources provide routing paths to connect the inputs and outputs of the CLBs and IOBs onto the appropriate networks.

In particular, we consider RAM-based FPGAs, which exhibit two types of input: the *operation* inputs (used during normal operation to apply the *input data*), and the *configuration* ones (used before normal operation to load the *configuration data* into the internal memory cells).

A simplified representation of a XILINX XC4000 CLB is shown in Fig. 1.

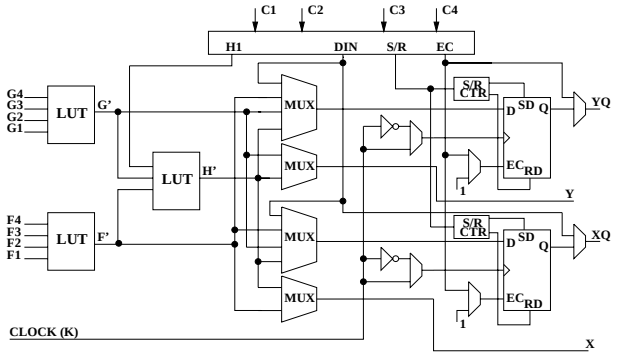

Figure 1. Schematic representation of a XILINX XC4000 FPGA CLB.

Each CLB is composed of three function generators implemented as memory Look-Up Tables (LUTs), hereafter referred to as LUT G', LUT F', LUT H', depending on the name of the output signal. LUT G' and LUT F' can implement any arbitrarily defined Boolean function of their four inputs, while LUT H' can implement any Boolean function of the three inputs F', G' and H1. More in detail each LUT presents the internal block structure shown in Fig. 2.

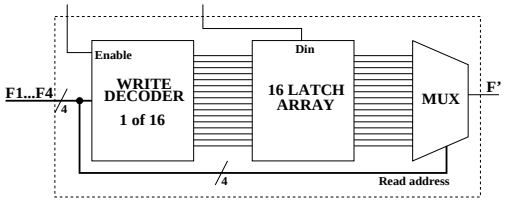

Figure 2. Internal block structure of each LUT.

Moreover, each CLB consists of several MUXs and two flip-flops. In particular, the flip-flops are edge-triggered D-type flip-flops with common clock (K) and clock enable (EC) inputs. A third common input (S/R) can be programmed as either an asynchronous set or reset signal independently for each flip-flop.

As previously introduced, several MTPs have been presented in the literature. They are targeted to testing either the interconnections of the FPGA [14, 7, 15, 9], or the CLBs [3, 5, 6]. This common practice derives from the classical "divide and conquer" approach [9].

As for the CLBs' testing, in [3] a hybrid fault model (based on a physic and functional FPGA characterization) has been introduced for CLBs. The use of an external storage device (such as a PROM) is considered to provide the configuration bit stream and the input test vectors, and a test strategy is proposed which is based on the CLB partitioning into two modules (one *combinational* and one *sequential*) and on the construction of one-dimensional arrays. Instead, in [5, 6] a BIST approach to FPGA testing has been proposed.

3 Proposed Reuse-oriented Test and Diagnosis Procedure (RTDP)

3.1 Considered fault models

The frequent reuse of the same FPGA in adverse environment makes it very prone to physical failures that, absent at the end of the manufacturing process, originate during reuse. Such faults could be realistically described as stuck-ats (SAs), transistors stuck-on (SONs), transistors stuck-open (SOPs), bridgings (BFs).

Unfortunately, the unavailability of transistor level details regarding the implementation of the FPGA allows only to analyse the effects of such physical failures on the logic behavior of the FPGA component blocks (Figs. 1 and 2),

thus, for instance, making it impossible to consider the electrical effects produced by BFs [17, 18].

On the basis of these considerations, we have considered the following set of faults ($\mathcal{F}$) possibly affecting each CLB (excluded the carry logic): a) for multiplexers and flip-flops, all single and multiple functional faults [3]; b) for LUTs: b1) all decoder faults resulting in one or more bit address SAs, b2) all single and multiple SAs affecting the various cells of the memory matrix, b3) all single BFs (AND/OR equivalent) between the various cells (affecting the logic values stored(read) in(from) the faulty cells), b4) all single and multiple SAs affecting the input and output lines.

3.2 Global test procedure

As known, the FPGA is basically an N×N array of CLBs, which can be seen as N (1-D) Iterative Logic Arrays (ILAs) [2, 3, 6]. We suppose that the FPGA can be correctly configured (this can be for instance verified by EPROM parity checking and FPGA *readback* [1]). Moreover, we use functional testing (prior to RTDP application) to check the used *Boundary Scan* instructions.

Based on these assumptions, in RTDP the FPGA is configured in order to form: 1) an horizontal array (consisting of N 1-D horizontal ILAs); 2) a vertical array (consisting of N 1-D vertical ILAs).

For each horizontal/vertical array, 12 inputs (out of which 4 independent) of each CLB are used to: a) apply the test patterns in case of the first CLB of each row/column; b) connect it to the CLB preceding it in the row/column. The independent inputs are: one of the 4 inputs of LUT F', one of those of LUT G', one of those of LUT H', and DIN.

LUT F', G', and H' are configured to implement the transparent function, or the NOT function, or the EXOR function.

Our testing procedure simply consists of 3 phases: 1) FPGA configuration; 2) test vectors' application; 3) comparison of the given response with the applied test vector.

Both test vectors' application and response analysis are performed by using the FPGA *Boundary Scan*.

Differing from [2, 3], here the CLBs in each row (column) of the horizontal (vertical) array are connected to the following and previous CLB in order to ensure that the clocked outputs of a CLB are connected to the combinatorial outputs of the following CLB, and viceversa, as schematically shown in Fig. 3 for the horizontal array. In this way, the propagation input/output delays of the four different paths (for each row/column) connecting an input and an output line are equal to each other. The I/O blocks at the end of each line/row carry out the test results by the *Boundary Scan* shift-data-registers. These characteristics enable to simplify the design of the circuitry analysing the FPGA responses (described in section 5).

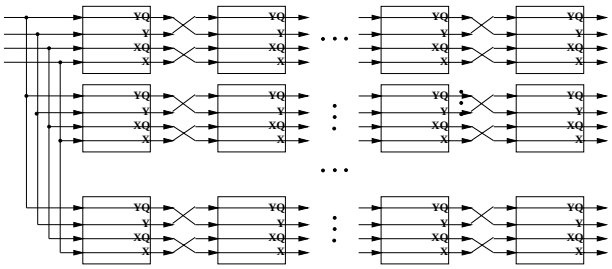

Figure 3. Considered CLB's connections in the proposed horizontal array structure.

For both the vertical and the horizontal arrays, 7 CLB's internal different configurations are considered (the same for all CLBs in the array), described in details in the next subsection. For each one of the 7 resulting vertical and horizontal array structures, a set of 16 test vectors is simultaneously applied to each row/column. Such test vectors have been chosen in order to ensure that their application to the considered different CLBs' configurations is such as to activate all faults $\in \mathcal{F}$, and to propagate the produced responses to the FPGA outputs, thus allowing 100% fault coverage with respect to $\mathcal{F}$.

Simultaneously, the interconnections involved in the considered configurations (estimated $\simeq 80\%$ of global routing resources) are tested with respect to single SAs, break faults, and BFs (AND/OR equivalent). However, as previously introduced, in order to exhaustively test all routing resources, one of the techniques in [14, 7, 15, 9] must be employed.

The use of both vertical and horizontal structures enables to easily locate the faulty CLB, thus allowing to reprogram the FPGA in order to tolerate faults.

3.3 CLB configurations

As previously introduced, for the horizontal (vertical) array structure, 7 configurations are considered for the CLBs in each row (column).

Each configuration is chosen so that the 4 independent inputs are driven to the output through independent paths.

These configurations are reported in Fig. 4.

In order to describe more in details these 7 configurations and how they enable to achieve 100% fault coverage with respect to $\mathcal{F}$, let us consider $\mathcal{F}$ as composed of two subsets $\mathcal{F}^R$ and $\mathcal{F}^{RR}$, such that $\mathcal{F} = \mathcal{F}^R \bigcup \mathcal{F}^{RR}$. In particular, $\mathcal{F}^R$ includes all the faults possibly affecting the CLB internal test vector propagation path (note that, as previously introduced, each configuration is chosen so that the 4 independent input bits are driven to the output through paths independent from each other), that is the three LUTs, the four MUXs which connect them to the FFs and the outputs,

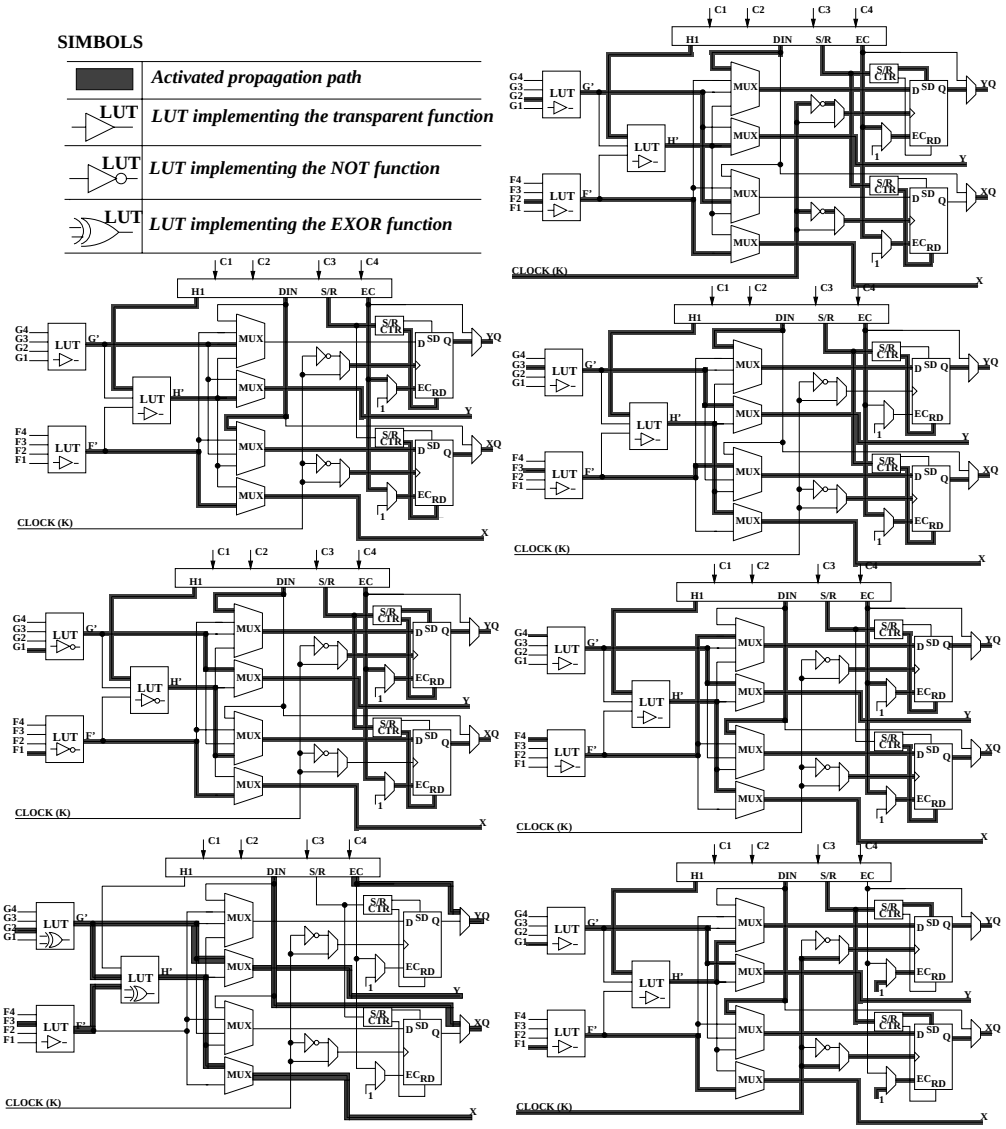

Figure 4. Used CLB configurations.

the H1 and DIN inputs with the relative MUXs, and the inputs and outputs of the FFs. Instead, $\mathcal{F}^{RR}$ consists of all the control signals of the FFs and related MUXs.

Let us analyse in details the case of faults $\in \mathcal{F}^{R}$.

It can be easily verified that, for at least one of the configurations in Fig. 4, a fault $\in \mathcal{F}^{R}$ results in a CLB output single error for at least one test vector, thus it can be detected at the end of the test vector shift through the row/column.

In fact, since for any configuration all the 16 test vectors are generated and forwarded through the 4 data propagation paths, FF's D and Q SAs can be detected. Moreover, as the configurations in Fig. 4 are devised so as to select all possible (at most 4) inputs of every MUX, faults affecting the inputs, output and SEL configuration bits of the MUXs can be detected.

In addition, as mentioned before, LUTs are configured to implement either the transparent function, or the NOT function, or the EXOR function. By configuring the LUTs as transparent and NOT functions of the same input, every SA on the 16 latches can be detected. In fact, in this way, the inputs are all connected and exhaustively stimulated.

As for decoder faults, those resulting in one or more address bit and input SAs are detected, for LUT F' and G', by considering the outputs as a transparent function of every input in different configurations. For LUT H', since not all its inputs are directly observable from the CLB primary inputs, an *ad hoc* configuration is necessary. In such a configuration, LUT G' and H' implement the EXOR function,

while LUT F' implements the transparent function (Fig. 4). This latter configuration enables also to test the MUXs connected to the outputs XQ and YQ.

Regarding the BFs (AND/OR equivalent) between LUTs' cells, they are also covered by the LUT functions so far considered.

This accounts for the faults of the set $\mathcal{F}^R$. Instead, in the set $\mathcal{F}^{RR}$, three kinds of faults can be individuated, that is those related to the signals S/R, CK and EC, respectively. Let us analyze them in details.

1. S/R faults: a global S/R signal is distributed to all the CLBs and, at the beginning of every test session, the signal is applied. Then the result is shifted out of the CLB arrays in the standard manner and checked at every shift cycle. This procedure is adopted in at least 4 configurations, in order to cover SAs of the S/R MUX 4 inputs and MUX selection bits. Also set and reset options are alternatively chosen in order to test the S/R controllers.

2. CK SAs are intrinsically tested by the data shifting through the array. A selection fault on the MUXs, which select the transparent or negated clock, results in one cycle delay of the vector first shifted, which can be detected by the checking circuitry.

3. EC SAs 1 are tested by applying to EC a clock-like signal with double period with respect to the shifting clock signal, which results in the first vector shift of one cycle in advance in case of faults. SA 0 is easily detected since it prevents one or more vector bits from shifting. This procedure must be implemented in at least 4 configurations to cover the EC MUX.

4 Required testing and diagnosis time

As described in the previous section, the total number of vectors of the considered test set is 16. The number of configurations required for testing only is 7, while that required for testing and diagnosis is 14.

For the XILINX XC4000 family, about 350 bits of configuration data are used for each CLB and its associated interconnects, and 0.1 μs configuration time is required for each bit. The XC4013 has 24×24 = 576 CLBs, thus the global configuration time is $\simeq$ 25 ms.

For each test vector: i) 12 clock periods are necessary to apply the test vectors; ii) 3 to update the *Boundary Scan* TAP controller; iii) (N/2) to obtain the FPGA response; iv) 3 to memorize the responses on the *Boundary Scan*; v) (6× N) to analyse the responses of the horizontal array configuration, or (2× 6× N) in case of the vertical array configuration. Moreover, the S/R faults require N data extraction and must be performed in 4 configurations, similarly to the EC faults, which require one more extraction and double the propagation time of the test vectors.

Thus, on the whole, (2784× 7 + (4 × N × 144) + (4 × (N/2 +144))) clock periods are required for testing only, while ((2784 + 5088) × 7 + (4 × N × 144 + 4 × N × 288) + (4 × (N/2 + 144) + 4 × (N/2+288))) for testing and diagnosis.

Consequently, with a clock frequency of 40 MHz: 1) if only testing is required, the testing time is 33936 × 0,025 = 848μs; 2) if testing and diagnosis is desired, 98400 × 0,025 = 2460 μs are necessary.

Instead, the configuration time equals 25 ms.

Consequently, the global testing time (i.e. the time necessary to configure the FPGA for testing and to perform testing) is 25ms × 7 + 0,848 = 175 ms. Moreover, the global testing and diagnosis time (i.e. the time necessary to configure the FPGA for testing and diagnosis and to perform testing and diagnosis) is 25 × 14 + 2,46ms = 352 ms.

It appears clear that, as usual, the configuration time is significantly higher than the testing and/or diagnosis time.

Based on these considerations, compared to [3], our procedure enables a global testing time relative reduction $\simeq$ 66%. With respect to [16], the global testing time reduction is $\simeq$ 56 or 84%, depending on the considered case [16].

Thus, our testing procedure is suitable to applications (e.g. space- and physic-oriented experiments) where it is particularly important that testing time is as shorter as possible, in order to allow the frequent and reliable reuse and dynamic functional modification of the FPGA. Moreover, similarly to the procedure in [16], RTDP does not require any I/O pin (because of the *Boundary Scan* use), thus allowing to directly test FPGAs mounted on *Boundary Scan* supporting boards.

5 Employed test facilities

The proposed RTDP has been developed and implemented to test the FPGAs under (dynamic) use for the development of prototypes for a space-oriented project (*PHOCA* project [12]).

In particular, the test facilities implemented for the RTDP application are shown in Fig. 5, where we have denoted by: a) "TPG", the Test Pattern Generator consisting of a 4-bit exhaustive generator ("Ex. Generator"), two shift registers (one used to insert the test patterns in the *Boundary Scan* by the TDI (Test Data Input) pin, the other to extract the FPGA responses from the *Boundary Scan* by the TDO (Test Data Output) pin); b) "Comparator", a simple bit-by-bit comparator; c) "Diagnostic Block", the logic used to perform diagnosis, shown more in details in Fig. 6. Note that the used clock (CK) is the system clock.

Diagnosis is accomplished in the following way. In case the result of the comparison between a test vector and the produced response indicates the presence of a fault, a constant high logic value is memorized in a memory location,

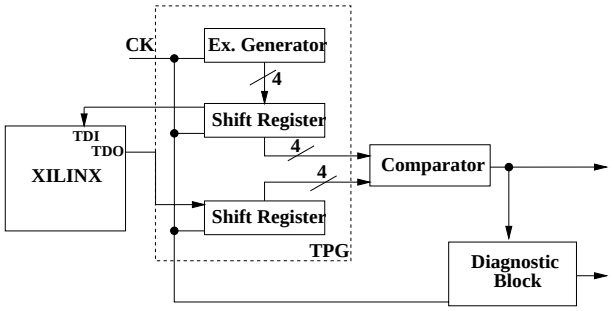

Figure 5. Block structure of the test facilities implemented for RTDP application.

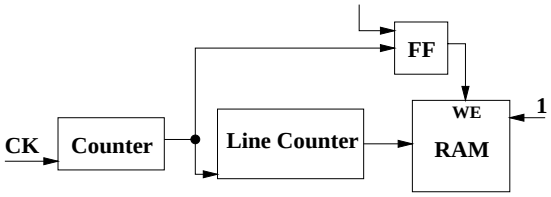

Figure 6. Internal schematic structure of the Diagnostic Block of Fig. 5.

whose address is representative of the vector row/column. This is obtained by using two counters, the former ("Count" in Fig. 6) producing the clock signal for the connected one ("Line Counter" in Fig.6), which updates the line (row or column) index.

The described testing facilities can be implemented by using another FPGA or an unused part of the *under test* FPGA which, in turn, can be suitably tested by changing its role with the *under test* FPGA, or the unused part of FPGA. In particular, for the development of prototypes for the space-oriented *PHOCA* project [12], another FPGA has been used to test several FPGAs.

6 Conclusions

This paper has presented a novel technique for testing Field Programmable Gate Arrays (FPGAs) that: i) enables a short time to accomplish test and diagnosis, thus being suitable to testing FPGA during their frequent reuse and/or dynamic functional modification; ii) does not present any practical problem due to the FPGA I/O limitations; iii) considers a wide set of realistic faults.

References

[1] X. Inc., *The Programmable Logic Data Book*. 1996.

[2] X. T. Chen, W. K. Huang, F. Lombardi, and X. Sun, "A Row-Based FPGA for Single and Multiple Stuck-At Fault Detection," in *Proc. of IEEE Int. Work. on Defect and Fault Tolerance in VLSI Systems*, pp. 225 – 233, 1995.

[3] W. K. Huang and F. Lombardi, "An Approach for Testing Programmable/Configurable Field Programmable Gate Arrays," in *Proc. of IEEE VLSI Test Symp.*, pp. 450 – 455, 1996.

[4] T. Inoue, H. Fujiwara, H. Michinishi, T. Yokohira, and T. Okamoto, "Universal Test Complexity of Field-Programmable Gate Arrays," in *Proc. of Asian Test Symp.*, pp. 259 – 265, 1995.

[5] C. Stroud, P. Chen, and M. Abramovici, "Built-In Self-Test of Logic Blocks in FPGAs," in *Proc. of IEEE VLSI Test Symp.*, pp. 387 – 392, 1996.

[6] M. Abramovici and C. Stroud, "ILA BIST for FPGAs: A Free Lunch with Gourmet Food," in *Proc. of 2nd IEEE Int. On-Line Testing Work.*, pp. 91 – 95, 1996.

[7] W. K. Huang, X. T. Chen, and F. Lombardi, "On the Diagnosis of Programmable Interconnect Systems: Theory and Application," in *Proc. of IEEE VLSI Test Symp.*, pp. 204 – 209, 1996.

[8] N. Itazaki, Y. Matsumoto, and K. Kinoshita, "BIST for CLBs of a Look-Up Table Type FPGA - A Comparator Based BIST Technique under Definite Fault Models," in *Proc. of 3rd IEEE Int. On-Line Testing Work.*, pp. 202 – 206, 1997.

[9] M. Renovell, J. Figueras, and Y. Zorian, "Test of RAM-Based FPGA: Methodology and Application to the Interconnect," in *Proc. of IEEE VLSI Test Symp.*, pp. 230 – 237, 1997.

[10] A. L. Burress and P. K.Lala, "Logic Design for Self-Checking FPGA Implementation," in *Proc. of 3rd IEEE Int. On-Line Testing Work.*, pp. 207 – 211, 1997.

[11] M. Alderighi, E. Gummati, V. Piuri, and G. Sechi, "A FPGA-based Implementation of a Fault Tolerant Neural Architecture for Photon Identification," in *fpga*, pp. 166 – 172, 1997.

[12] E. G. Tanzi, "Photon Counting and Analog Intensified Imagers for UV and X-Ray Radiation," in *IFCTR-CNR Tech. Rep.*, 1995.

[13] S. D'Angelo, L. Mantoani, R. Mazzei, S. Russo, and G. Sechi, "Modular design of communication node prototype," pp. 170 – 175, 1997.

[14] M. Renovell, J. Figueras, and Y. Zorian, "Testing the Interconnect Strucutre of Uncofigured IFPGA," in *Proc. of IEEE European Test Work.*, pp. 125 – 129, 1996.

[15] H. Michinishi, T. Yokohira, T. Okamoto, T. Inoue, and H. Fujiwara, "A test methodology for interconnect structures of LUT-based FPGAs," in *Proc. of Asian Test Symp.*, pp. 68 – 74, 1996.

[16] M. Abramovici, E. Lee, and C. Stroud, "BIST-Based Diagnostic of FPGA Logic Blocks," in *Proc. of 3rd IEEE Int. On-Line Testing Work.*, pp. 196 – 201, 1997.

[17] C. Metra, M. Favalli, P. Olivo, and B. Riccò, "CMOS Checkers with Testable Bridging and Transistor Stuck-on Faults," in *Proc. of IEEE Int. Test Conf.*, pp. 948 – 957, 1992.

[18] C. Metra, M. Favalli, P. Olivo, and B. Riccò, "On-Line Detection of Bridging and Delay Faults in Functional Blocks of CMOS Self-Checking Circuits," *to appear in IEEE Trans. on CAD*.