Testing the Configurable Interconnect/Logic Interface of SRAM-Based FPGA's

M. Renovell J.M. Portal
LIRMM-UM2 161 Rue Ada
34392 Montpellier Cedex France
renovell@lirmm.fr
Tel (33)467418523

J. Figueras
UPC Diagonal, 647
Barcelona Spain
figueras@eel.upc.es
Tel (34)934016603

Y. Zorian
Logic Vision Inc. 101 Metro Drive
San Jose CA 95110 USA
zorian@lvision.com
Tel (1)4084530146

Abstract

*The objective of this paper is to define a minimum number of configurations for testing the configurable modules that interface the global interconnect and the logic cells of SRAM-based FPGAs. In usual SRAM-based FPGAs, Configurable Interface Modules (**CIMs**) can be found between the global interconnect and inputs of the logic cells (Input CIMs) or between output of the logic cells and the global interconnect (Output CIMs). It is demonstrated that an input CIM that connects N^{in} segments to a logic cell input requires N^{in} test configurations and that an output CIM that connects a logic cell output to N^{out} segments requires 2 test configurations. Then, it is proven that a set of K^{in} input CIMs can be tested in parallel making the number of required test configurations equal to N^{in}. In the same way, a set of K^{out} output CIMs is shown to require only 2 test configurations if $N^{out} > K^{out}$. Finally, it is shown that the complete mXm array of logic cells with K^{in} input CIMs and K^{out} output CIMs can be tested with only N^{in} test configurations using the XOR tree and shift register structures.*

1. Introduction

Field Programmable Gate Arrays (FPGAs) combine the flexibility of mask programmable gate arrays with the convenience of field programmability [1-2]. In such a programmable circuit, a matrix of logic modules and interconnection elements can be configured in the field to implement a desired designed function. Because of their short turnaround time and programmability in the field, they have been widely used for rapid prototyping or reconfiguration of complex digital systems.

Testing of these chips has only recently been addressed [3-23]. In the published works, different aspects have been addressed. As an example, Inoue and al. address the problem of testing look-up table in [11], Huang and al. address the problem of testing the configurable logic in [18], Abramovici and al. focus on BIST for FPGA in [15,16,17] and Lombardi and al. focus on diagnosis in [19]. The authors have proposed a test procedure for the global interconnect of SRAM-based FPGAs in [4,5], for the configurable logic cells in [6,7] and for the LUT/RAM modules in [8,9].

The work presented in this paper concerns the test of the configurable modules interfacing the global interconnect and the logic cells. In usual SRAM-based FPGAs, Configurable Interface Modules (**CIM**s) can be found between N^{in} segments of the global interconnect and inputs of the logic cells (Input CIMs) or between output of the logic cells and N^{out} segments of the global interconnect (Output CIMs). In this paper, they are assumed to be implemented with FPGA multiplexers or simply with pass transistors but the results can be easily extended to any type of such FPGA interface module. The problem of testing CIMs has been investigated by Michinishi in [12] and Ashen in [20]. The complexity in terms of test configurations of the solution proposed in [12] is 2mN+18m+5N+6, m being the size of the FPGA configurable cell array (mXm) and N ($N=N^{in}=N^{out}$) being the number of segments in the global interconnect. The solution proposed in [20] is more oriented to PLDs like the Altera Flex family and targets the complete circuit resulting in an important number of test configurations. The authors have proposed a solution targeting only the input CIMs implemented with multiplexers [10]. The objective of this paper is to propose a general solution targeting the complete set of input and output CIMs.

Taking into account the FPGA configurability, a test procedure consists in successively configuring the FPGA using the configuration input then applying a test sequence using the operating inputs. A typical test procedure for FPGA includes a few tens of configurations with their associated test sequences. It is of prime importance to note that a FPGA configuration corresponds to a very long sequence of bits serially entered in the FPGA. Consequently, the FPGA configuration process is an excessively time-consuming process to such a point that only one configuration can be equivalent in test time to the application of a few hundred of thousand vectors. In such a condition, it is clear that the number of FPGA configurations must be minimized. Consequently, the objective of this paper is to define a minimum set of test configurations targeting the configurable modules interfacing the global configurable interconnect and the configurable logic cells of a SRAM-based FPGA. In section 2, the problem of testing isolated input or output CIM is considered. It is shown that the number of test configurations is N^{in} for an input CIM connecting N^{in} segments to a logic cell and that the number of test configurations is 2 for an output CIM connecting a logic cell to N^{out} segments . In section 3, the problem of testing a set of K^{in} input and K^{out} output CIMs around a configurable logic cell is considered. It is demonstrated that the CIMs can be tested in parallel implying that the required number of test configurations is equal to N^{in} for the input CIMs and to 2 for the output CIMs if $K^{out} < N^{out}$. In section 4, the test of the complete mXm array of CIMs is studied. Using the concept of one-dimensional array, a particular cascade of logic cell is proposed in order to form XOR trees and shift register. In such conditions, the test of the complete mXm array is shown to require only N^{in} test configurations. Finally, section 5 concludes the work with an example of application to the XILINX 4000 family.

2. An isolated CIM

The core of SRAM-based FPGA appears as a two dimensional array in which mXm logic cells can be configured to implement user defined combinatorial as well as sequential logic functions. These configurable logic cells are separated by a configurable interconnection network. Figure 1 illustrates this mXm two dimensional array with m=3. In figure 1, a set of K^{in}=2 input CIMs and K^{out}=2 output CIMs interface each configurable logic cell with N^{in}=N^{out}=3 segments in the global interconnect.

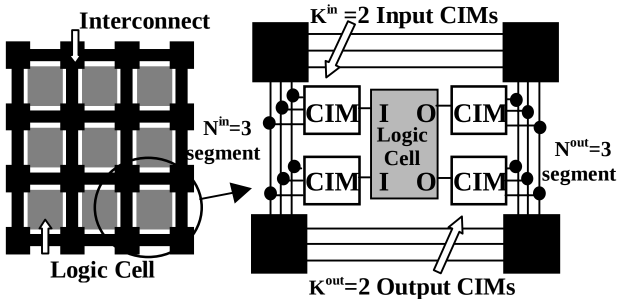


Figure 1 : The set of CIMs

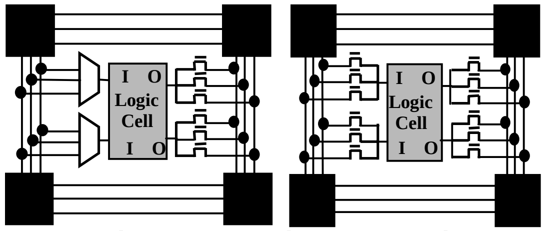


Figure 2 : CIM implementations

An input CIM can be implemented with a multiplexer or pass transistors as illustrated in Figure 2. An input CIM implemented with a multiplexer interconnecting N^{in} segments requires $n^{in}=\log_2 N^{in}$ configuration bits for the multiplexer address. An input CIM implemented with pass transistors interconnecting N^{in} segments requires N^{in} configurations bits for the transistor control. Figure 2 also shows that output CIMs are implemented with pass transistors.

This section is dedicated to the test of an isolated CIM. For a given CIM implementation, we assume that test configurations and test vectors are defined targeting a given fault model.

Multiplexer-based input CIM: We consider here an input CIM implemented with a multiplexer as illustrated in figure 3 with N^{in}=4. In a usual FPGA representation, the address bits of such multiplexer are not represented because they are connected to SRAM cells that store the FPGA configuration. Assuming now the stuck-at fault model on the multiplexer inputs and output, it has been demonstrated that N^{in} test configurations and 2 test vectors for each configuration are required [6,7]. As a matter of fact, for a n^{in} address bits multiplexer, the exhaustive number of configurations $N^{in}=2^{nin}$ is required while only the XOR and XNOR test vectors are required. As another example, we consider a functional fault model on the multiplexer of figure 3 [18]. It has been demonstrated that the same number of configurations $N^{in}=2^{nin}$ is required. But in this case, more than 2 test vectors for each configuration are required. Note that for SRAM-based FPGAs, the key point is the number of test configurations which is the same $N^{in}=2^{nin}$ whatever the assumed fault model. Consequently, the following demonstrations are given considering a number $N^{in}=2^{nin}$ of test configurations for a n address bit multiplexer whatever the considered fault model.

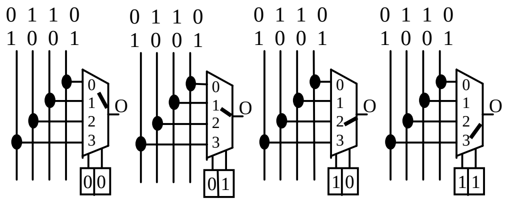


Figure 3: Test of a multiplexer-implemented input CIM

Pass-transistor-based input CIM: We consider here an input CIM implemented with pass transistors as illustrated in figure 4 with N^{in}=4.

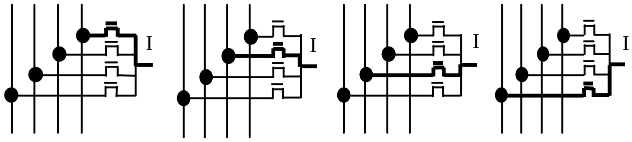


Figure 4: Test of a pass transistor-implemented input CIM

In case of transistor the most usual fault models are the stuck-ON and stuck-OFF. It is obvious that the number of required test configurations is equal to N^{in} for these fault models. One configuration is required per pass transistor. Note that for SRAM-based FPGAs, the key point is the number of test configurations which is the same N^{in} whatever the implementation (multiplexer or pass transistor) and whatever the assumed fault model (stuck-at or functional). Consequently, the following demonstrations are given considering a number N^{in} of test configurations for an input CIM connecting N^{in} segments.

Pass-transistor-based output CIM: We consider here an output CIM implemented with pass transistors as illustrated in figure 5 with $N^{out}=4$. As previously we consider the stuck-ON and stuck-OFF fault models. Using these fault models, only 2 test configurations are required. Each transistor must be ‘ON’ in one of the configurations and ‘OFF’ in the other configuration as illustrated in figure 5.

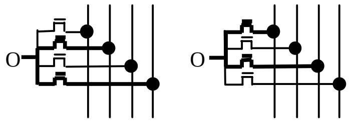


Figure 5: Test of a pass transistor-implemented output CIM

3. A set of CIMs

This section is dedicated to the test of a set of input and output CIMs around a configurable logic cell. Figure 6 gives an example of configurable logic cell with a set of $K^{in}=3$ input CIMs and $K^{out}=2$ output CIMs. In this example, the number of segments is the same for the inputs and ouputs $N^{in}=N^{out}=4$. In figure 6, the configurable logic cell is assumed to have the structure illustrated in figure 3 with a combinatorial output and a sequential output.

The set of input CIMs: According to the previous section, each input CIM requires $N^{in}=4$ test configurations. These input CIMs are assumed to be controlled by the input segments. Figure 6 clearly shows that they must be observed through the configurable logic cell. In the example of figure 6, the logic cell has 3 inputs (I_1, I_2 and I_3) and 2 outputs (O_1 and O_2). Two inputs and one output are connected to the combinatorial part of the logic cell and one input and one output are connected to the sequential part. Obviously, the logic cell is configurable and so in test mode, the combinatorial and the sequential functions can be chosen to guarantee the observability of the complete set of input CIMs.

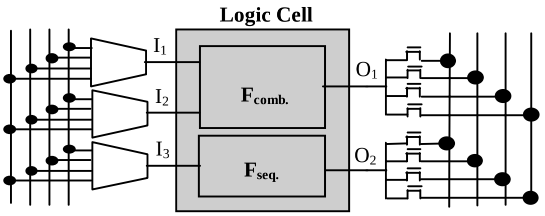


Figure 6: A set of input and output CIMs

Figure 7 gives an example where the combinatorial function of the cell is configured to implement the XOR function and the sequential function is configured to implement a simple flip-flop. It is absolutely obvious that the XOR function properties make observable any of its inputs i.e. any of the input CIMs. In the same way, the input CIM of the sequential part is fully observable after a top of the clock signal.

By configuring the logic cell with combinational or sequential functions that guarantee the observability of the input CIMs, these input CIMs can be tested in parallel. As an example, figure 7.a represents the 3 input CIMs in the first test configuration and figure 7.b in the second test configuration. Due to the parallel test, the number of required test configurations for the input CIMs of figure 7 is equal to $N^{in}=4$.

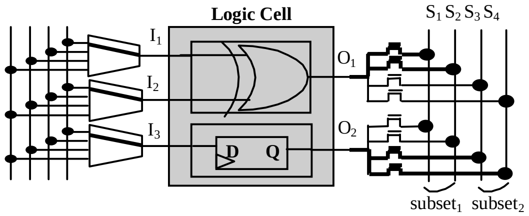


Figure 7.a: 1st test configuration

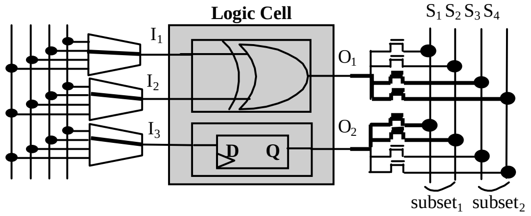


Figure 7.b: 2nd test configuration

Figure 7: Parallel test of input and output CIMs

The set of output CIMs According to the previous section, each output CIM requires only 2 test configurations. For a set of K^{out} output CIMs, the only constraint is because different output CIMs must be connected to different segments in order to avoid shorts. From a more theoretical point of view, we define K^{out} disjoint subsets of segments among the N^{out} segments assuming that $K^{out} < N^{out}$ (which is always the case in practice). Then each cell output is connected to a subset of segments. As an example in figure 7, we define $K^{out}=2$ subsets of segments among the $N^{out}=4$ segments. Then, the cell output O_1 is connected to $subset_1$ and output O_2 to $subset_2$ in the first test configuration of figure 7.a. In the second test configuration of figure 7.b, the cell output O_1 is connected to $subset_2$ and the output O_2 to $subset_1$. Finally, it appears that only 2 test configurations are require for the complete set of K^{out} output CIMs.

From the previous paragraph, it results that the total number of test configurations for a set of input and output CIMs is determined by the input CIMs because they require $N^{in}=4$ test configurations while the ouput CIMs require only 2 test configurations. This is true when $K^{out}<N^{ou\ t}$ which corresponds to most FPGAs in practice.

4. The mXm array of CIMs

This section is dedicated to the test of all the CIMs in the circuit i.e. the mXm array of set of CIMs. A very trivial approach consists in proposing to connect each logic cell with its surrounding input and output CIMs to primary inputs and outputs i.e. to IO pads. Looking for this optimal solution, it rapidly appears that actual FPGA do not have enough I/O pads to control and observe every set of CIMs in the array: actual FPGA have around 8m I/O pads and mXm modules [1,2,24]. Due to the limited number of I/O pads, a very usual and practical solution consists in forming m one-dimensional arrays of m interconnected modules as represented in figure 8 with m=3.

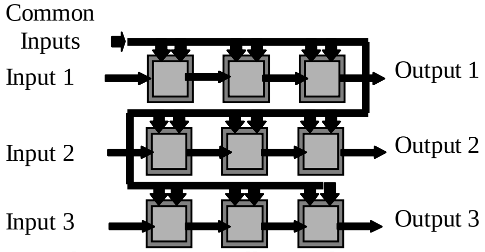


Figure 8: One-dimensional arrays

In the configuration of figure 8, the output of a given module is connected to an input of the following one. Each module is also connected to a set of primary inputs that are common to every module in the FPGA. Of course, the input of the first left most module is also a primary input, and the output of the last right most module is a primary output. This practical solution does not require many I/O pads, and has been used by many authors [4-12,18-21]. It must be clear that this solution requires less I/O pads but the fundamental problem is the controllability of the 'embedded' module through the preceding ones and the observability through the following ones. Figure 9 gives an example of a one dimensional array of 3 logic cells of figure 7. This example clearly illustrates the local connections between the combinatorial and sequential outputs of a given logic cell with the input CIMs of the following logic cell.

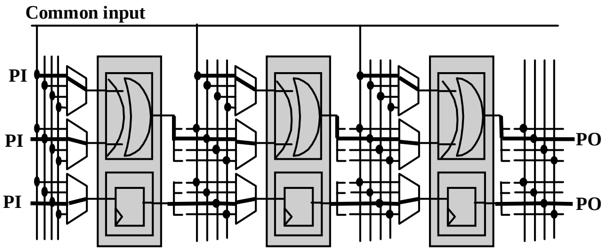
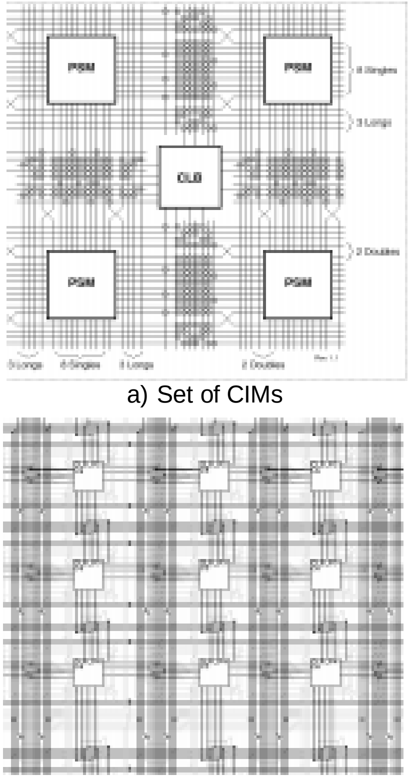


Figure 9: XOR cascade & pseudo-shift register

Using the principle of figure 9, the one-dimensional array includes a cascade of XOR functions and in parallel a cascade of flip-flops. Each XOR function receives the output of the previous XOR function and a set of common inputs. Due to the properties of the XOR function, it is clear that each CIM in the cascade of XOR is fully controllable and observable. Similarly, each flip-flop receives the output of the previous flip-flop forming a shift register. Due to the properties of the shift register, each CIM in the cascade of flip-flop is fully controllable and observable.

The one dimensional array of figure 9 covers one among the N^{in}=4 test configurations of each input CIM and covers one among the 2 test configurations of each output CIM. It is easy to imagine exactly the same structure with a permutation of the test configurations. This small example illustrates that the total number of test configuration is consequently equal to N^{in} for the complete mXm array of CIMs.

5. Application and Conclusion



a) Set of CIMs

b) Exemple of test configuration

Figure 10: Xilinx 4000 validation

This paper deals with the problem of minimizing the number of configurations to test the Configurable Interface Modules located between the global interconnect and the logic cells. It has been demonstrated that an input CIM connecting N^{in} segments to the logic cell requires N^{in} test configurations whatever the implementation and whatever the assumed fault model. An output CIM connecting the logic cell to N^{out} segments requires only 2 test configurations. It has also been demonstrated that a set of K^{in} input CIMs and k^{out} output CIMs around a configurable logic cell can be tested in parallel by adequately configuring the function of the logic cell. Using the XOR combinatorial function or the latch sequential function, the complete set of CIMs requires only N^{in} test configurations. Finally, using the concept of one-dimensional array, m logic cells are cascaded to form XOR trees or shift registers. Due to the properties of these cascaded structures, the test of the complete mXm array of set of CIMs requires only N^{in} test configurations.

This approach has been validated on an example of SRAM-based FPGA: the 4000 XILINX family [24]. The configurable logic cell (CLB) of the XILINX 4000 family illustrated in figure 10.a includes K^{in}=13 input CIMs and K^{out}=4 output CIMs. Each input CIM is connected to N^{in}=12 segments that belong to a set of 8 single-length lines, 4 double-length lines. For the sake of clarity, this 12 segments do not include the global long lines, that would make the example much more complex. Anyway, the basic principle with the global long lines would remain similar. Consequently, the set of CIMs can be tested in N^{in}=12 test configurations. Figure 10.b gives an example of test configuration implemented on the XILINX 4000. The configurable logic cell are cascaded to form one-dimensional horizontal arrays of XOR functions or latch. The segments are connected through the switch matrix (PSM) to form the common lines. On the XILINX 4000 example, the test of the complete array of CIMs requires only N^{in}=12 test configurations independently of the number of CIM and independently of the size mXm of the array.

6. References

[1] S.D. Brown, R.J. Francis, J. Rose, S.G. Vranesic, «Field-Programmable Gate Arrays», Kluwer Academic Publishers, 1992.

[2] S.M. Trimberger (ed), «Field-Programmable Gate Array Technology», Kluwer Academic Publishers, 1994.

[3] C. Jordan and W.P. Marnane, «Incoming Inspection of FPGAs», Proc. of IEEE European Test Conference, pp. 371-377, 1993.

[4] M. Renovell, J. Figueras and Y. Zorian, «Test of RAM-Based FPGA: Methodology and Application to the Interconnect», 15th IEEE VLSI Test Symposium, pp. 230-237, Monterey, CA, USA, May 1997.

[5] M. Renovell, J.M.Portal, J. Figueras and Y. Zorian, «Testing the Interconnect of RAM-Based FPGAs», IEEE Design & Test of Computer, special Issue on FPGAs, pp.45-50 , January-March 1998.

[6] M. Renovell, J.M. Portal, J. Figueras and Y. Zorian, «Test Pattern and Test Generation Methodology for the Logic of RAM-Based FPGA», IEEE Asian Test Symposium, pp. 254-259, Akita, Japan, November, 1997.

[7] M. Renovell, J.M. Portal, J. Figueras and Y. Zorian, «Testing the Configurable Logic of RAM-based FPGA», IEEE Int. Conf. on Design, Automation and Test in Europe, pp. 82-88, Paris, France, Feb 1998.

[8] M. Renovell, J.M. Portal, J. Figueras and Y. Zorian, «SRAM-Based FPGAs: Testing the LUT/RAM modules», IEEE International Test Conference, pp. to appear, Washington, USA, October, 1998.

[9] M. Renovell, J.M. Portal, J. Figueras and Y. Zorian, «SRAM-based FPGAs: Testing the RAM Mode of the LUT/RAM Modules», JETTA the Journal of Electronics Testing: Theory and Application, pp. to appear, 1998.

[10] M. Renovell, J.M. Portal, J. Figueras and Y. Zorian, «SRAM-based FPGAs: Testing the Interconnect/Logic Interface», IEEE Asian Test Symposium, pp. to appear, Singapore, December, 1998.

[11] T. Inoue, H. Fujiwara, H. Michinishi, T. Yokohira and T. Okamoto, «Universal Test Complexity of Field-Programmable Gate Arrays», 4th Asian Test Symposium, pp. 259-265, Bangalora, November 1995, India.

[12] H. Michinishi, T. Yokohira, T. Okamoto, T. Inoue, H. Fujiwara «A Test Methodology for Interconnect Structures of LUT-based FPGAs», IEEE 5th Asian Test Symposium, pp. 68-74, November 1996.

[13] H. Michinishi, T. Yokohira, T. Okamoto, T. Inoue, H. Fujiwara «Testing for the Programming Circuits of LUT-based FPGAs», IEEE 6th Asian Test Symposium, pp. 242-247, November 1997.

[14] T. Inoue, S.Miyazaki and H. Fujiwara «Universal Fault Diagnosis for Lookup Table FPGAs», IEEE Design & Test of Computer, special Issue on FPGAs, pp.39-44, January-March 1998.

[15] M. Abramovici and C. Stroud, «No-Overhead BIST for FPGAs», 1st IEEE International On-line Testing Workshop, pp. 90-92, Nice, FRANCE,1995.

[16] C. Stroud, P. Chen, S. Konala, M. Abramovici, «Evaluation of FPGA Ressources for Built-In Self Test of Programmable Logic Blocks», Proc. of 4th ACM/SIGDA Int. Symposium on FPGAs, pp. 107-113, 1996.

[17] M. Abramovici, C. Stroud, «ILA BIST for FPGAs: A Free Lunch with Gourmet Food», 2nd IEEE International On-line Testing Workshop, pp. 91-95, Biarritz, FRANCE,1996.

[18] W.K. Huang and F. Lombardi, «An Approach for Testing Programmable/Configurable Field Programmable Gate Arrays», 14th IEEE VLSI Test Symposium, pp. 450-455, Princeton, NJ, USA, May 1996.

[19] F. Lombardi, D. Ashen, X.T. Chen, W.K. Huang «Diagnosing Programmable Interconnect Systems for FPGAs», FPGA'96, pp. 100-106, Monterey CA, USA, 1996.

[20] D.G.Ashen, F.J.Meyer, N.Park and F.Lombardi, «Testing of Programmable Logic Devices (PLD) with Faulty Resources», IEEE International Workshop on Defect & Tolerance in VLSI Systems, pp.76-84, Paris, October 1997.

[21] W.K. Huang, F.J. Meyer, N. Park and F. Lombardi, «Testing Memory Modules in SRAM-based Configurable FPGAs», IEEE International Workshop on Memory Technology, Design and Test, August, 1997.

[22] M. Hermann and W. Hoffmann, «Fault modeling and test generation for FPGAs», in R.W. Hartenstein and M.Z. Servit (eds), Lecture Notes in Computer Science, Field Programmable Logic, Springer-Verlag, pp. 1-10, 1994.

[23] R.O. Durate and M. Nicolaidis, «A test methodology applied to cellular logic programmable gate arrays», in R.W. Hartenstein and M.Z. Servit (eds), Lecture Notes in Computer Science, Field Programmable Logic, Springer-Verlag, pp. 11-22, 1994.

[24] Xilinx, «The Programmable Logic Data Book », San Jose, USA, 1994