Efficient Scan-Based BIST Architecture for Application-Dependent FPGA Test

Conference Paper · November 2013

CITATIONS
3

READS
1,374

5 authors, including:

Kazumi Hatayama
Gunma University
83 PUBLICATIONS **554** CITATIONS
SEE PROFILE

Michiko Inoue
Nara Institute of Science and Technology
174 PUBLICATIONS **1,065** CITATIONS
SEE PROFILE

Efficient Scan-Based BIST Architecture for Application-Dependent FPGA Test

Keita Ito†*, Tomokazu Yoneda†*, Yuta Yamato†*, Kazumi Hatayama†*, and Michiko Inoue†*
†Nara Institute of Science and Technology, 8916-5, Takayama, Ikoma, Nara, 630-0192, Japan
* Japan Science and Technology Agency, CREST, Chiyoda-ku, Tokyo, 102-0075, Japan
{keita-i, yoneda, yamato, k-hatayama, kounoe}@is.naist.jp

Abstract

FPGAs are attractive devices due to their low development cost and short time-to-market, and widely used not only for reconfigurable purpose but also as application-dependent embedded devices for low-volume products. This paper presents a scan-based BIST architecture for testing of application-dependent circuits configured on FPGA. In order to build up BIST components such as LFSR, MISR and scan chains for test points, the proposed architecture efficiently utilizes memory blocks, instead of logic elements, which are unused for application-dependent circuits. The proposed BIST architecture provides enhanced scan functionality for test points and performs a hybrid test application of LOC and enhanced scan to improve delay test quality. Experimental results show that the proposed BIST architecture achieves high delay test quality with efficient resource utilization.

keywords: *FPGA, BIST, delay test, DFT.*

1 Introduction

Continuous advances in silicon manufacturing technologies allow us to design large, high-speed and low power products. Among the many challenges imposed by the technologies, high in-field reliability is major concern and periodical online self-test is essential for overcoming reliability issues such as transistor aging [1, 2]. One of the major applications of online self-test is circuit failure prediction [3, 4, 5, 6]. Circuit failure prediction predicts the occurrence of a circuit failure before the appearance of any error and the basic principle is to capture the gradual delay shift caused by the transistor aging using delay test schemes.

Field Programmable Gate Arrays (FPGAs) are reconfigurable devices which are manufactured with most advanced technologies. And, FPGAs are widely used not only for its intrinsic reconfigurable purpose but also as application-dependent embedded devices for low-volume products due to their low development cost and short time-to-market compared with ASIC designs. Therefore, it is also important to ensure in-field reliability of application-dependent circuits implemented on FPGA devices.

Application-dependent FPGA test methods can be classified into three categories: 1) test re-configuration (TRC) methods [7, 8], 2) DFT methods [9, 10, 11] and 3) BIST methods [12, 13]. TRC methods prepare test-specific configurations so that the FPGA resources used the application-dependent circuits can be tested easily and efficiently. However, TRC methods generally require several test configurations to completely test the FPGA resources used in the application and it takes long time to load the test configurations to the target FPGA. DFT methods modify the structure of FPGA devices during design and manufacturing phases. The modifications include scan chain insertion [9] and improvement of test configuration loading scheme to shorten the loading time [10, 11]. However, these methods introduce high area overhead and decrease performance of FPGA devices. BIST methods utilize unused resources in the user application as pseudo-random pattern generators and response analyzers. In order to achieve high test quality, we should have several test-specific configurations [9], which require long time to load, and/or scan architecture [10] for the application-dependent circuit implemented on FPGA.

For ASIC designs, scan-based BIST with test point insertion is a well-established approach to perform delay test for circuit failure prediction in-field [3, 4, 6] . On the other hand, FPGAs are not scan-ready devices (no scan cell and no scan chain), and scan cells/chains must be implemented using general resources such as Look-Up-Tables (LUTs), registers in Logic Elements (LEs) and local/global interconnects. However, it was reported that scan insertion for an application-dependent circuit introduces 50% increase in LE usage [9], and it is not an efficient to adopt ASIC scan-based BIST architecture as it is to testing of application-dependent circuits on FPGAs.

This paper presents an efficient scan-based BIST architecture for application-dependent circuits for FPGAs. The contributions of the paper are summarized as follows.

- It presents FPGA-specific and area-efficient architecture for LFSR/MISR using shift register configurations of memory blocks.
- It presents a scan chain architecture for test point FFs. The proposed test point chain architecture is also implemented using memory blocks for area efficiency

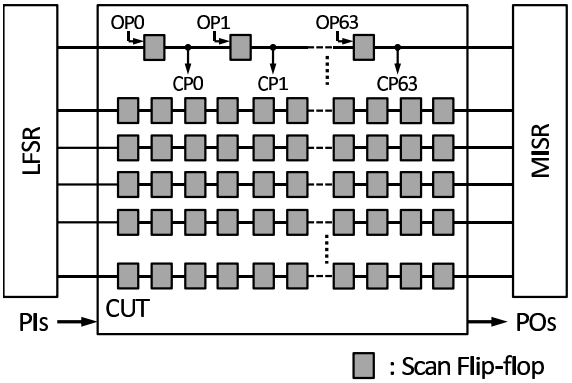

Figure 1. Conventional BIST architecture.

and provides *enhanced scan* functionality that can improve delay test quality.

- Experimental results show the effectiveness of the proposed architecture in terms of area and fault coverage of transition delay faults compared to a conventional scan-based BIST architecture used for ASIC designs.

The rest of the paper is organized as follows. Section 2 explains the target BIST architecture and motivation why conventional scan-based BIST architecture for ASIC designs is not efficient on FPGA devices in terms of area overhead. The proposed BIST architecture and its test application scheme are discussed in Section 3. Experimental results are shown in Section 4. Finally, Section 5 concludes this paper.

2 Motivation

Figure 1 shows a conventional scan-based BIST architecture with test point insertion which is widely used for ASIC products. LFSR and MISR are used as a test pattern generator and test response compactor. FFs in Circuit-Under-Test (CUT) are replaced with scan cells and several scan chains are constructed. Moreover, control points (CPs) and observation points (OPs) are added to improve random pattern testability. In the target BIST architecture, we assume that each CP and OP share one FF as a test point and these test points are stitched to form independent scan chains. We call the scan chain for test points as *test point chains (TPCs)* to distinguish from other scan chains in CUT. We also assume *Launch-On-Capture (LOC)* test application scheme for delay fault testing.

Unlike ASIC designs, FPGA devices do not equip scan cells and scan chains in many cases. Therefore, scan cells/chains should be implemented using general resources such as LUTs, registers in LEs and local/global interconnects as shown in Fig. 2. In this example, the LUT function is reconfigured from 2-input AND gate (application function) to a multiplexer for scan operation, and an additional LE is used to implement the original application function.

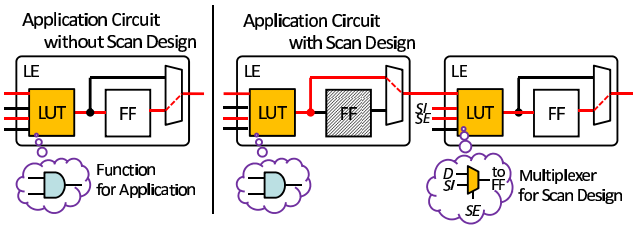

Figure 2. Scan chain implementation on FPGA.

It is reported in [9] that scan insertion for an application-dependent circuit on FPGA introduces 50% increase in LE usage [9]. Moreover, test point insertion is mandatory to improve random pattern testability and it further increases area overhead. Consequently, users have to keep LE utilization for application circuits at very low rate to implement scan-based BIST on FPGA devices and therefore it is not efficient to adopt ASIC scan-based BIST architecture to application-dependent FPGA testing without any change.

3 Memory Block Based BIST Architecture

3.1 Overall Architecture

Memory blocks are another resources embedded on FPGAs and can be configured to provide various memory functions such as RAM, ROM, FIFO buffers and shift registers without using LEs. In this paper, we assume that intended circuits for scan-based BIST insertion are logic-intensive circuits. In other words, the application-dependent circuits targeted in this paper require LE resources rather than memory resources to be implemented on FPGAs. Note that LE resources include LUTs and registers and general sequential circuits can be implemented on FPGAs only with LEs. The main idea is to efficiently utilize unused memory blocks, instead of LEs, to implement BIST components such as LFSR, MISR and TPCs.

The overview of the proposed BIST architecture is shown in Fig. 3. The detailed architectures of LFSR, MISR and TPCs and its test application scheme will be explained in the following subsections. The unique characteristics are summarized as follows.

- Shift register mode of memory blocks is efficiently configured to implement registers in LFSR, MISR and TPCs. And, only the remaining logics such as XOR gates and multiplexers in those BIST components are implemented using LEs. Therefore, the proposed BIST architecture reduces LE overhead drastically.

- TPCs and normal scan chains are controlled by independent scan enable signals, SE_{TPC} and SE_{CUT}, respectively. TPCs remain unchanged during half of the test application cycles and it avoids to consume unnecessary power.

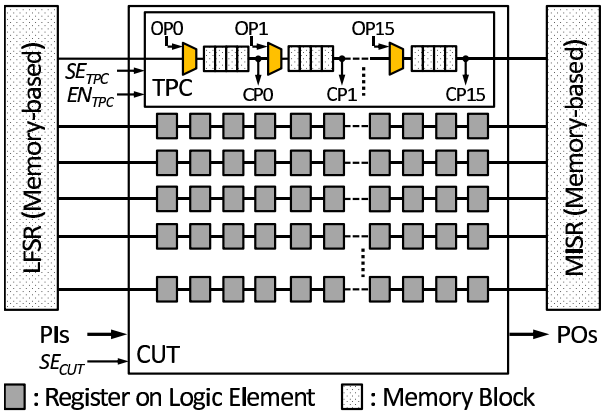

Figure 3. Proposed BIST architecture.

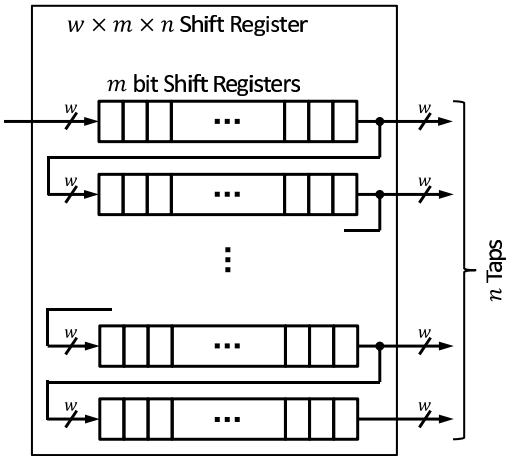

Figure 4. Shift register mode configuration of memory block embedded on Altera's FPGA device families.

- TPCs provide enhanced scan cell function that can improve delay test quality in LOC test application scheme.

3.2 LFSR and MISR

As we explained in Section 3.1, memory blocks can be configured to provide shift register functions without using LEs. However, there are several design constraints to be satisfied to implement a shift register on memory block. For example, memory block embedded on Altera's FPGA device families provides the shift register configuration shown in Fig. 4. The shift register configuration is determined by the input data width w, the length of the taps m and the number of taps n, and the size of a $w \times m \times n$ shift register must be less than or equal to the maximum number of memory bits. In addition, the length of the taps m must be more than or equal to 3.

In order to implement the shift register part of LFSR on a memory block, we select the following configuration as shown in Fig. 5.

1. $w = 1$
2. $m = 3$ (minimum length of the taps)
3. $n = N_{sc}$ where N_{sc} is the total number of scan chains including TPCs

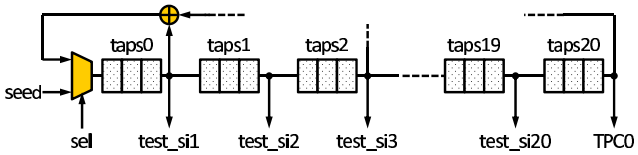

Figure 5. Proposed LFSR architecture and implementation.

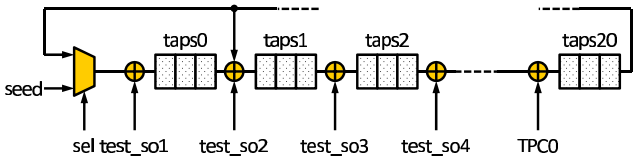

Figure 6. Proposed MISR architecture and implementation.

All the n bits output of the shift register on memory block are connected to scan-in ports to feed pseudo-random patterns to CUT. Some of the n bits output are connected to XOR gates, which are implemented by LEs, to construct a feedback loop of the LFSR. As you can observe from Fig. 5, the size of LFSR becomes m (3 in the FPGA device used in this paper) times larger than that used in the conventional BIST architecture to satisfy the design constraints. However, most part of the LFSR is implemented on memory block and there is not so much increase in LE usage.

Similarly, MISR is implemented using a shift register configuration of memory block as shown in Fig. 6. In MISR case, we select the following configuration: (1) $w = 1$, (2) $m = 3$ (minimum length of the taps) and (3) $n = 1$, and prepare N_{sc} 1-tap 3-bits shift registers. The N_{sc} 1-tap 3-bits shift registers are connected in serial by way of XOR gates, which are implemented by LEs, to form MISR.

3.3 Test Point Chain

A TPC is also implemented using a shift register configuration of memory block as shown in Fig. 7. We select the following configuration for each test point (i.e., a pair of CP and OP): (1) $w = 1$, (2) $m = 4$ and (3) $n = 1$. Each 1-tap 4-bits shift register is connected to a single CP and OP. They are also connected in serial by way of the multiplexers, which are implemented on LEs, to form a TPC. If

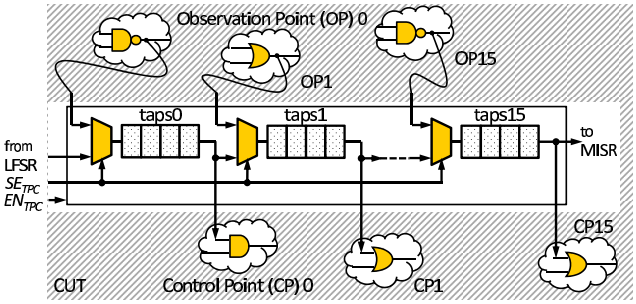

Figure 7. Proposed TPC architecture and implementation.

the length of a TPC is larger than normal scan chains, then the TPC must be divided into several TPCs which does not exceed the length of normal scan chains.

A 1-tap 4-bits shift register can store two 2-pattern delay tests (i.e., 4 bits) for each CP and works as an enhanced scan cell during test application (details will be explained in the next subsection). TPCs are controlled by a scan enable signal SE_{TPC} and clock enable signal EN_{TPC}. TPCs are active only when $EN_{TPC} = 1$. They work as a single shift register connected to LFSR/MISR when $SE_{TPC} = 1$ and work as independent 1-tap 4-bits shift registers that can provide 2-pattern delay test for CPs and observe responses from OPs (i.e., enhanced scan capability) when $SE_{TPC} = 0$.

3.4 Test Application Scheme

Figure 8 shows a timing diagram of the proposed BIST architecture during test application. Basically, it follows LOC-based at-speed delay test application scheme controlled by scan enable signal SE_{CUT}, and only TPCs have unique behavior controlled by SE_{TPC} and EN_{TPC} explained as follows.

Only when an even-numbered test pattern (i.e., pattern id is $i \times 2$) is scanned in, TPCs are active and work as a single shift register in between LFSR and MISR (i.e., $EN_{TPC} = 1$ and $SE_{TPC} = 1$). At the end of scan-in cycles for the even-numbered test pattern, each 1-tap 4-bits shift register in TPCs contains two 2-pattern delay tests (i.e., 4 bits) for a CP as shown in Fig. 9. Then, SE_{TPC} is de-activated before the launch cycle and TPCs are switched to work as independent 1-tap 4-bits shift registers to capture test responses from OPs. When the next test pattern (odd-numbered pattern) is scanned in, EN_{TPC} is de-activated and TPCs become in-active. This is because TPCs can contain two 2-pattern delay test per scan shift operation and they still contain another 2-pattern test after launch-and-capture cycles for the previous even-numbered pattern. It can avoid to consume unnecessary power. After that, EN_{TPC} is again activated and it performs launch and capture operation. This process is repeated until the BIST test application process is completed.

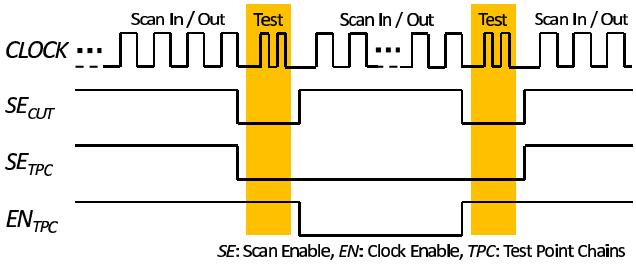

Figure 8. Timing diagram of control signals for proposed BIST.

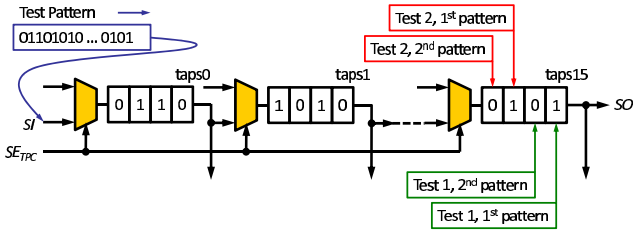

Figure 9. TPC function.

4 Experimental Results

In this section, we present experimental results for one of ITC'99 benchmark circuits b17. It was synthesized with 20 scan chains and its scan chain length is 65. We refer the design as "b17 w/ scan" and use it as a baseline for our comparison in Table 1.

Then, we compared two BIST architectures, "Conventional BIST" and "Proposed BIST" on an FPGA device. In the conventional BIST architecture, we prepare one FF per each CP/OP test point pair and the size of LFSR/MISR is equal to the total number of scan chains including TPCs. In contrast, we used the 3-bits shift register configuration for LFSR and MISR to satisfy the design constraints for shift register realization on memory blocks. Therefore the size of LFSR/MISR is 3 times larger than the conventional BIST architecture. Similarly, we chose the 1-tap 4-bits shift register configuration for TPCs, and then the number of FFs used for TPCs becomes 4 times larger than the conventional BIST architecture. In this experiments, we randomly added 128 CP/OP test point pairs to "b17 w/ scan" design but the location and type of the added test points are identical in "Conventional BIST" and "Proposed BIST". In this case, the conventional BIST architecture requires 128 FFs for the test points and 2 TPCs of length 64 were constructed. Then, the size of LFSR/MISR is 22 (20 scan chains + 2 TPCs). On the other hand, the proposed BIST architecture needs 512 FFs and 8 TPCs of length 64 were built. Consequently, the size of LFSR/MISR becomes 84 ((20 scan chains + 8 TPCs) $\times$ 3-bits shift register) as shown in Table 1.

The three architectures "b17 w/ scan", "Conventional BIST" and "Proposed BIST" were implemented on Altera Cyclone III EP3C16F484C6 using Quartus II 13.0 Web Edi-

Table 1. Results for b17.

Design	b17 w/ scan	conventional BIST	Proposed BIST	
		LE	Memory	LE
Test point	-	0	128 CP/OP test point pairs	
Number of flip-flops for TPCs	-	-	128	512
LFSR/MISR size	-	20	22 (20 + 2)	84 ((20 + 8) × 3)
Resource for shift register	-	LE	LE	Memory	LE
Total logic elements	9334	9406	9493	9550	9911
Total combinational functions	9126	9186	9278	9350	9268
Total registers	1317	1357	1489	1333	1997
Total memory bits	0	0	0	424	0
Delay test	-	LOC	LOC	LOC + enhanced scan	
Fault coverage [%]	-	43.18	49.43	54.47	

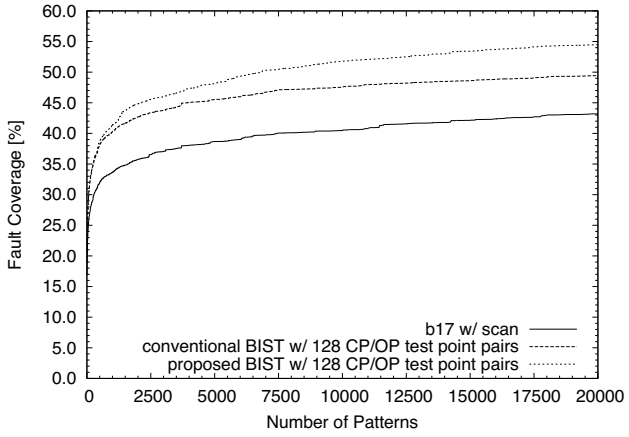

Figure 10. Fault coverage for 20,000 pseudo-random patterns.

tion, and area overhead is evaluated in terms of "Total logic elements", "Total Combinational Functions", "Total registers" and "Total memory bits". The experimental results for the three architectures are summarized in Table 1. Fault simulations were also performed to evaluate delay test quality for the two BIST architectures. The fault simulation results using 20,000 pseudo-random patterns from a seed are shown in Fig. 10.

First, we compare two results in columns "Memory" and "LE" of the proposed BIST architecture. These two methods have exactly the same architecture (as the proposed architecture). However, "Memory" implemented the architecture using memory blocks while "LE" implemented with LEs only. Therefore, the results for these two columns are identical except for area related items. From the results, we observe that the proposed architecture and its implementation using memory blocks can drastically reduce the LE utilization, and users do not worry about how much LE resources can be used for application and how much LE resources should be kept for BIST implementation.

Then, we compare two results in columns "Proposed BIST (Memory)" and "Conventional BIST" with 128 CP/OP test point pairs. Note that the conventional BIST architecture can not be implemented using memory blocks since it does not satisfy the design constraints for shift register realization on memory blocks. Despite the proposed BIST architecture has larger LFSR/MISR and includes more TPCs, the difference in LE usage is very small since the proposed BIST architecture efficiently replaces LE resources with memory bits. On the other hand, the difference in delay test quality is remarkable. The fault coverages of the two architectures at the end of 20,000 pseudo-random pattern application are 54.47% and 49.43% respectively, and the proposed method can achieve 5% higher coverage than the conventional method. When we compare the number of patterns to reach 45% fault coverage, the proposed and conventional BIST architectures require 2,130 and 3,930 patterns, respectively. This shows that the proposed method can obtain 45% reduction in pattern count for the same fault coverage. These gains come from the two reasons: (1) the proposed BIST architecture efficiently implemented larger size LFSR and MISR and (2) TPCs have enhanced scan capability for LOC-based delay test application scheme.

5 Conclusions

In this paper, we have presented a scan-based BIST architecture for application-dependent FPGA testing. The main idea presented in this paper is to utilize shift register configurations of memory blocks on FPGA devices to implement BIST components such as LFSR, MISR and TPCs. The proposed architectures for LFSR, MISR and TPCs satisfy the design constraints of shift register realization on memory blocks. Consequently, the proposed architecture can effectively reduce LE usage for scan-based BIST implementation. Moreover, the proposed TPC architecture pro-

vides an enhanced scan capability and improve test quality of LOC-based delay test application scheme.

One of the future works is to extensively evaluate the proposed architecture using larger and practical benchmark circuits and FPGA devices. In the experiments, we have randomly inserted CPs/OPs to CUT. We will analyze how the test points affect delay test quality and investigate a method to optimize delay test quality in the proposed BIST architecture.

Acknowledgments

This work was supported in part by Japan Society for the Promotion of Science (JSPS) under Grants-in-Aid for Scientific Research (B)(No.25280015).

References

[1] W. Wang, V. Reddy, A. Krishnan, R. Battikonda, S. Krishnan, and Y. Cao, "Compact modeling and simulation of circuit reliability for 65-nm CMOS technology," *IEEE Trans. Device and Materials Reliability*, vol. 7, pp. 509–517, Dec. 2007.

[2] T. W. Chen, K. Kim, Y. M. Kim, and S. Mitra, "Gate-oxide early life failure prediction," in *Proc. VLSI Test Symposium*, pp. 111–118, Apr. 2008.

[3] Y. Li, S. Makar, and S. Mitra, "CASP: Concurrent autonomous chip self-test using stored test patterns," in *Proc. Design, Automation, and Test in Europe*, pp. 885–890, Mar. 2008.

[4] H. Inoue, Y. Li, and S. Mitra, "VAST: Virtualization-assisted concurrent autonomous self-test," in *Proc. International Test Conference*, Paper 12.3, Oct. 2008.

[5] O. Khan and S. Kundu, "A self-adaptive system architecture to address transistor aging," in *Proc. Design, Automation, and Test in Europe*, pp. 81–86, Mar. 2009.

[6] Y. Sato, S. Kajihara, T. Yoneda, K. Hatayama, M. Inoue, Y. Miura, S. Ohtake, T. Hasegawa, M. Sato, and K. Shimamura, "DART: Dependable VLSI test architecture and its implementation," in *Proc. International Test Conference*, Paper 15.2, Nov. 2012.

[7] M. Tahoori, "Application-dependent diagnosis of FPGAs," in *Proc. International Test Conference*, pp. 645–654, Oct. 2004.

[8] M. Tahoori and S. Mitra, "Interconnect delay testing of design on programmable logic devices," in *Proc. International Test Conference*, pp. 635–644, Oct. 2004.

[9] M. Renovell, P. Faure, J. Portal, J. Figueras, and Y. Zorian, "IS-FPGA: A new symmetric FPGA architecture with implicit SCAN," in *Proc. International Test Conference*, pp. 924–931, Nov. 2001.

[10] A. Doumar and H. Ito, "Testing the logic cells and interconnect resources for FPGAs," in *Proc. Asian Test Symposium*, pp. 369–374, Nov. 1999.

[11] K. Inoue, M. Koga, M. Amagasaki, M. Iida, Y. Ichida, M. Saji, J. Iida, and T. Sueyoshi, "An easily testable routing architecture and prototype chip," *IEICE Transactions on Information and Systems*, vol. E95-D, no. 2, pp. 303–313, 2012.

[12] M. Abramovici and C. Stroud, "BIST-based test and diagnosis of FPGA logic blocks," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 9, no. 1, pp. 159–172, 2001.

[13] M. Rozkovec, J. Jenicek, and O. Novak, "Application dependent FPGA testing method," in *Proc. Euromicro Conference on Digital System Design: Architectures, Methods and Tools*, pp. 525–530, Sep. 2010.