2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)

Test Configuration Generation for different FPGA Architectures for Application Independent Testing

Shukla Banik
Department of CSE
National Institute of Technology
Durgapur, India

Suchismita Roy
Department of CSE
National Institute of Technology
Durgapur, India

Bibhash Sen
Department of CSE
National Institute of Technology
Durgapur, India

***Abstract*—This paper presents a FPGA interconnect test configuration generation strategy for application-independent testing scenario using Satisfiability Modulo Theory (SMT). The technique generates all possible configurations for the interconnect to obtain full fault coverage. To generate test configurations automatically, constraints have been designed using SMT. The proposed technique targets open and short faults in the interconnect resources. Test configurations have been generated for different FPGA architectures. The objective of the proposed approach is to minimize number of configurations without reducing the fault coverage. Experimental results show that the proposed strategy can obtain the minimum number of test configurations with 100% fault coverage, with respect to the fault list.**

Keywords- Application-independent testing, Field-Programmable Gate Array (FPGA), interconnect testing, test configurations

I. Introduction

Field Programmable Gate Arrays (FPGAs) are prefabricated electronic devices that can be programmed to implement any digital logic in the field. FPGAs are very popular target devices for various range of applications as it is reprogrammable and also provide a faster time to market solutions as compared to other existing ICs (e.g. ASIC). The main components of FPGAs are configurable logic blocks (CLBs), programmable interconnection network and programmable input/output blocks (IOBs). The configurable logic block is used to implement user defined logic function. CLBs are generally placed in a two dimensional grid and are interconnected by programmable interconnection network. CLBs contain all the resources that are required for implementing an user-defined digital design to be implemented on a FPGA. Inside CLBs, there exist Look-up tables(LUTs), flip flops, multiplexers etc. A look-up table can be used to implement an n-input user defined combinational logic function. I/O blocks provide all the input/output connections to the FPGA. These are placed at the boundary of the grid and are connected to the programmable interconnect. The interconnection resources comprises of line segments and programmable switches. Connections among line segments can be altered by programming the corresponding switches. Among all the resources, the interconnections are most error-prone, as it occupies a huge die area and also it contains enormous number of transistors. Hence interconnection resources testing is very important in order to guarantee the correct functionality of the FPGA. Also FPGA interconnection testing should cover maximum possible faults found in the interconnection.

In the proposed approach, the entire FPGA interconnection is tested for opens and short faults found in the interconnection resources i.e. line segments and programmable interconnection points(PIPs). The key features of the proposed approach are listed below:

- the proposed technique uses SMT based modelling of the problem to generate test configurations for the entire interconnect.
- this approach does not address a specific architecture, rather it can be applied to various type of FPGAs, which shows the flexibility and scalability of the proposed test generation method.
- experimental results which are obtained by applying the proposed technique on different FPGA devices, are also encouraging.

FPGA interconnection resources consist of line segments and switch matrices. Inside switch matrices, the programmable interconnection points are used to connect various line segments. Different type of line segments are found in different FPGA architectures. For e.g. single length lines are used to connect adjacent CLBs, double length line segments connect CLBs that are two blocks apart, hex-length line segments connect CLBs that are three or six blocks apart. Similarly pent type of line segments connect CLBs that are five blocks apart. The FPGA switch matrix literally acts as a black box. Different type of horizontal and vertical line segments entering into the switch matrix are connected through programmable interconnection points(PIPs). As these interconnection points are programmable, connection can be established between desired line segments by programming the appropriate PIP. The FPGA configuration memory is used to store the PIP values. The PIP value will be changed once the connection break up. Similarly a new connection can be done between two line segments by changing the corresponding PIP value in a particular configuration. As these PIPs are used to connect different type of line segments inside the switch matrix, therefore PIPs can be categorized as single-single PIP, hex-hex PIP, hex-single PIP and so on.

Based on the available FPGA testing approaches, FPGA interconnect testing methods are broadly categorized as

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00086

application-dependent and application-independent [1]. Application dependent testing does not test the entire FPGA resources, rather it aims to test only the resources that have been utilized in a specific design i.e application. This type of testing is favourable for application specific FPGA where the design will be fixed for the lifetime of the device. On the other hand, application-independent testing targets the entire FPGA . It is also known as "manufacturing testing". In this approach, the FPGA interconnect is tested for all possible configurations to test all the resources. This type of testing can assure the reliability of an FPGA device. It does not concentrate on testing only design specific resources, rather it deals with all the resources that may be used in some design, therefore it broadens the acceptability of the FPGA, irrespective of the application to be built on it. FPGA configuration time is a crucial factor for FPGA testing. More number of test configurations will take longer time to test the FPGA. The test vector application time is less significant than the time to configure an FPGA before applying the test vector. So minimum number of test configurations must be achieved to reduce the FPGA test time.

In this work, we have applied SMT to find minimum number of test configurations. These configurations represents connection subsequently among specific resources. Each configuration will be tested for different faults using suitable test vectors.

The Satisfiability problem, deals with finding a satisfiable assignment of the variables of a problem represented as a first-order logic formula. Various type of problems from different areas (such as verification of hardware and software, routing problem) can be solved using satisfiability. Boolean satisfiability (SAT) solvers are used to solve problems that are described using Boolean formulae. Satisfiability Modulo Theories (SMT) is the problem of finding if a formula is satisfiable with respect to some background theory [2]. SMT has gained popularity recently as it supports non-Boolean expression of a problem and different background theories to solve the problem. Various theories are supported by SMT solvers for example the theory of linear arithmetic, arrays, bit-vectors, equality etc. [3]. SMT offers modelling of a problem using richer language constructs (for example bit vectors or arrays). A supporting theory (of bit vector or of arrays) is then needed to express the meaning of the formulas used in the problem. SMT-LIB includes the all logics that are used by these theories. SMT solvers also offer the use of optimization functions which was missing in Boolean SAT solvers. Problems using minimization or maximization functions can be easily encoded using SMT solvers. Various SMT solvers are available now such as Z3, Yices etc

II. Background

A. Interconnect structure

Generally the connections between line segments, inside the switch matrix is controlled by pass transistors. Each switch box contains six pass transistors. These pass transistors are programmable, which provides the routing flexibility inside the switch box.

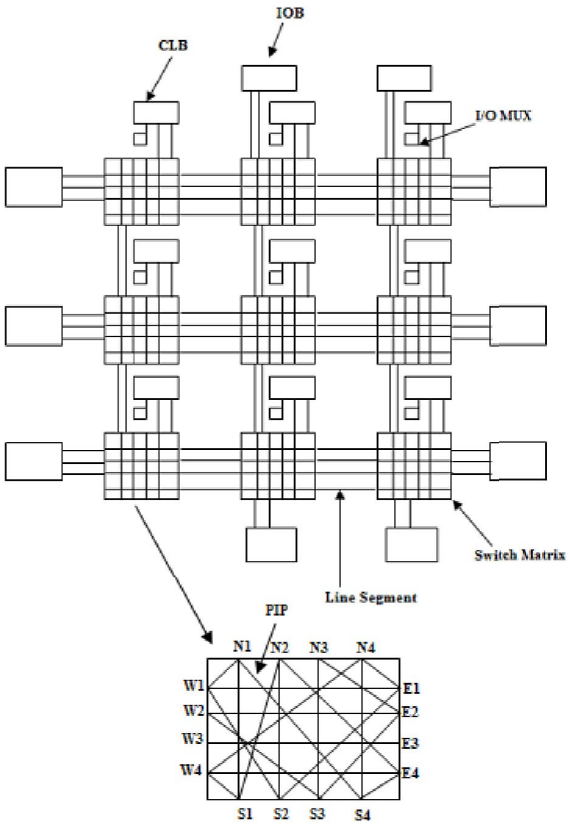

FIG. 1: Virtex FPGA architecture

In figure 1 an architecture of a Virtex FPGA is shown. The switch matrix points W1-W4 indicates the terminal points of west direction, similarly terminal points of east, north and south directions are denoted by E1-E4, N1-N4 and S1-S4 respectively. A programmable interconnect point (PIP) is actually consists of one or more transmission gates. A PIP can be controlled by a configuration memory bit. Three types of PIPs are found in the FPGA architecture, break-point PIPs, cross-point PIPs and multiplexer-PIPs [4]. Break-point PIPs are used to establish connection between either horizontal or vertical line segments. The cross-point PIPs are responsible for connecting horizontal and vertical line segments. A group of transmission gates and more than one level of configuration memory bits to the gates can be used to implement a multiplexer based PIP.

Each switch box offers interconnection among line segments in six directions, north-south, north-east, north-west, east-west, east-south and west-south. Line segments are generally arranged in horizontal as well as vertical manner across the four sides of a switch box. XCV300 contains 24 single lines and 12 hex lines. Some of these lines are bidirectional and rest of them are unidirectional. Based on the number of

CLBs connected by a line segment, these line segments are categorised as long lines, single-length, double-length, hex-length etc. Long lines are useful for connecting nets which are placed faraway. Also long lines are responsible for carrying time-critical signals. PIPs can also be classified as intra-PIP and inter-PIP [4]. PIPs which establish connection between same type of line segments are called intra-PIPs (e.g. single-single PIP). On the other hand PIPs that are used to connect different type of line segments are called inter-PIPs (e.g. hex-single PIP).

The interconnect resources of an FPGA can be categorized as inter-CLB and intra-CLB resources [5]. The first category i.e.inter-CLB interconnects consists of wire segments (or line segments) and programmable switch matrices. On the other hand the intra-CLB interconnects offers connectivity inside a CLB. Inter-CLB interconnect resources testing is discussed in this paper.

B. Fault Model

Two significant faults are found in the FPGA interconnect, opens and shorts [5]. PIPs and line segments both can be affected by open and short faults. Open fault in PIPs causes the PIP to be remain open which in turn will disconnect the line segmnets that were previously connectd by this PIP. Similarly, a PIP will remain closed when short fault occurs [6]. The configuration memory bit or SRAM cell stores '0' when PIP is open and it stores '1' to indicate that the PIP is closed in the fault free circuit [7].

C. Related Work

Numerous works have addressed application-independent testing of FPGA interconnect in the literature and various techniques have been proposed to decrease the number of interconnect configurations. Different strategies have been taken in the previous works to test the global interconnect resources. Most of these works have utililized built-in self-test (BIST) strategy to reduce the input-output pin requirements for testing. Generally in the global interconnect testing, the interconnect resources are configured as global buses and through input-output pins, test vectors are applied. In built-in self-test (BIST) strategy the test vectors are applied internally [8], [5], [9], [10], [11], [12], [13], [6]. Local interconnect testing, which involves local interconnection inside the CLBs, is discussed in [14].

Various type of techniques are found in the recent FPGA testing process. The traditional manufacture-oriented FPGA testing involves commercial test equipments and testing is performed in the production environment [15]. The FPGA is first configured and then test vectors are applied externally via input pins. After that the test responses are collected via output pins, which are compared with the expected output. BIST approaches are quite popular in application independent FPGA testing to solve the shortage of I/O pin problems [8], [5], [9], [10], [11], [12], [13], [6]. BIST technique requires test pattern generator (TPG) and output response analyzer (ORA) to implement the additional BIST logic and circuitry for test generation. TPG generates the required test pattern for FPGA and ORA is responsible for checking the output response with the desired response. In this approach FPGA internal memory is needed to store the test vectors and responses. Also it is assumed that the TPG and ORA circuits are fault-free. In [16] the global interconnect resources of Virtex-4 series FPGA is tested using BIST technique.

In [15], a bit stream read back based method is proposed for functional testing of FPGA interconnect. The authors in [17], proposed a routability-aware algorithm for test generation of SRAM based FPGA. Interconnect test configuration in [5] and [18], are generated on the basis of the PIPs directions. In this approach, four direction of PIPs were proposed, horizontal, vertical, left-diagonal and right diagonal. In addition to that, configuration for these PIPs were generated using 1-1 mapping and 1-N mapping. This type of technique is applicable only for simple interconnect structure such as Xilinx 4000, and for newer FPGA such as Virtex-4, Virtex-5, Virtex-6 etc, this approach is inapplicable since newer versions of FPGA do not follow the regular switch matrix structure like XC4000 [19].

III. Proposed Approach

In general, FPGA interconnect testing is performed by finding the test configurations. The programmable resources of a FPGA are configured by loading a configuration bit stream. To activate a particular resource in a design, the corresponding bit value needs to be changed. Since application independent FPGA does not have prior design information, so all resources must be tested to ensure reliability of the device. In order to test interconnect resources, in each configuration, several paths are formed, which can drive signals from input terminal to output terminal simultaneously. A path is formed by connecting line segments and corresponding PIPs. Only independent paths (having non overlapped PIPs) are tested in a configuration, as bridging fault activation requires to apply opposite values to the adjacent wires, which can generate a conflict if overlapped PIPs exist. If the resources of a path are fault free, then any input test vector applied to the path will be same as the output response. On the other hand, a mismatch between the input test sequence and the output sequence will confirm a fault.

In most of the test configuration generation methods found in the literature, heuristics are used to find independent paths (connecting non-overlapped PIPs) in an iterative manner where every iteration removes the previously used PIPs. But while finding a new path, some PIPs might need to be reused, since more than one PIPs can drive the same line segments, when full switch matrix connectivity is considered. So this type of approach may not get full fault coverage. In the proposed approach, instead of removing used PIPs, which may reduce the fault coverage, all PIPs are kept in the input file and appropriate constraints have been designed to solve test configuration generation problem.

The novelty of the work lies in its strategy of generating minimum number of test configurations using SMT based modelling of the problem. The proposed approach utilizes the

ability of SMT to express various attributes of test configuration generation method with the help of arithmetic constraints. SMT solver Z3 will solve this problem by extending the underlying SAT solver and taking help of theory of arithmetic. Also SMT supports optimization, with the help of which the objective function has been designed. For path generation, arithmetic constraint has been used to describe the attribute of a path.

The proposed strategy finds out a path by connecting the PIPs and line segments. Based on the different type of line segments entering into the switch matrix, various type of PIPs are found. For single length line segments, single-single PIPs are there and for hex length line segments hex-hex PIP, hex-single PIPs exist. Paths are formed by connecting either single-single PIPs along with corresponding line segments or hex-hex PIPs (or hex-single PIPs) along with corresponding line segments. Similarly for double and pent type of line segments, paths can be formed. In this approach, paths are formed by connecting the horizontal line segments entering into the switch matrix from one end and all possible PIPs (horizontal and/or vertical), which may have a possible subsequent connectivity inside the switch matrix. A configuration consists of one or more independent paths (having non-overlapped PIPs), which can be tested simultaneously. The SMT solver takes the connectivity among line segments and PIPs as input and generates path and configuration assignment for every PIP.

The presented work uses the following constraints to generate test configurations for application independent FPGA interconnect testing. SMT solver Z3 is used to solve the optimization problem by providing a satisfying assignment to all variables used in the problem by using the constraints. A brief description of all the constraints used in this work are given below:

.

1) *Coverage constraint*: Although the main objective is to generate the minimum number of test configurations but fault coverage is also equally important. In order to get full fault coverage, all PIPs should be included in the paths as well as configurations. To activate fault in all PIPs, every PIP should be a part of a path and configuration. To connect the same input line segment of a switch matrix to a specific output line segment, many options (PIPs) are there. Since a configuration will contain only specific PIPs (among many possible PIPs), between two line segments, so to activate the rest of the PIPs, separate paths need to be formed. Therefore every PIP will get assigned to a $path\_val$ as well as a $config\_val$, which must be assigned positive integer values.

For each PIP,

$$path\_val * config\_val \neq 0$$

to ensure that all PIPs in the circuit are covered by some path in some configuration.

2) *Path generation constraint*: Line segments can enter into the switch matrix horizontally as well as vertically. A path is generated by assigning a $path\_val$ to a horizontal line segment entering into the switch matrix and PIP which connect the line segment with other PIPs until the output end of the switch matrix is reached. Every path will start from one input end of the switch matrix and will connect subsequent PIPs to reach to the output end of the switch matrix. As this approach aims to find the path as well as configuration at the same time, so when one or more PIPs are used to connect two line segments, same ($path\_val * config\_val$) will be assigned to these PIPs. If pip_i and pip_j are consecutively associated to connect specific line segments, then

$$path\_val_i * config\_val_i = path\_val_j * config\_val_j$$

.

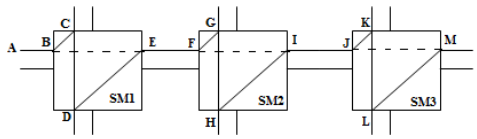

FIG. 2: Path generation (a) Path 1.

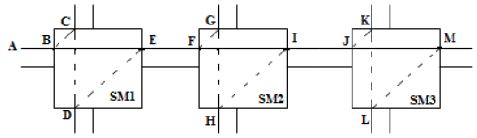

FIG. 3: Path generation (b) Path 2

In figure 2 and 3, two simple paths are illustrated, where solid lines indicate used PIPs and dashed lines indicate unused PIPs. In figure 2, PIP BC, CD, DE will be assigned same ($path\_val * config\_val$).

3) *PIP conflict avoidance constraint*: Every configuration should contain paths having non-overlapped PIPs only. If two paths contain same PIP, then the paths containing common PIP should be assigned to two different configurations. In other words, these paths (or PIPs) will be activated in two different configurations, so that they can carry different test signals at the time of test vector application.

If pip_1 is used in two paths (e.g. $path_i$ and $path_j$) then, $config\_val\_i \neq config\_val\_j$
where, $config\_val\_i$ and $config\_val\_j$ represents the $config\_val$ of pip_1 for $path_i$ and $path_j$.

FIG. 4: (a) Configuration 1 (b) Configuration 2

In figure 4, two paths are shown, where solid lines and dashed lines are used to denote the used and unused PIPs

respectively. PIP AB should be assigned to different configuration values to avoid conflict.

4) *Objective function*:

$$Min \sum_{\forall PIPs} \sum_{\forall config\_val} config\_val$$

The objective function is designed to obtain minimum number of test configurations. All the above mentioned conditions must be satisfied in order to solve the objective function. The objective function will generate test configurations for the entire interconnect. The following method is applicable for different type of FPGA having different type of line segments which shows the flexibility of the proposed approach.

Example:

In figure 5, an example of an interconnect consisting of three switch matrices is shown. Test configurations have been generated for the above mentioned figure using proposed approach. PIP A1B1 is included in two paths, therefore A1B1 is active in two different configurations.

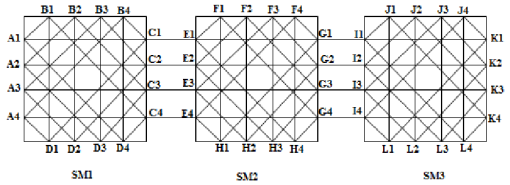

FIG. 5: FPGA interconnect

Figure 6 shows all three test configurations generated using proposed approach. In the first configuration, PIP A1B1, B1D1, D1C1 in switch matrix 1 are active. This configuration consists of 4 paths. Path 1 consists of PIP A1B1, B1D1, D1C1, E1F1, F1H1, H1G1, I1J1, J1L1 and L1K1. Path 2 consists of PIP A2B2, B2D2, D2C2, E2F2, F2H2, H2G2, I2J2, J2L2 and L2K2. Path 3 includes of PIP A3B3, B3D3, D3C3, E3F3, F3H3, H3G3, I3J3, J3L3 and L3K3. Path 4 contains PIP A4B4, B4D4, D4C4, E4F4, F4H4, H4G4, I4J4, J4L4 and L4K4.

The second configuration also contains 4 paths. Path 1 consists of PIP A1B1, B1C4, E4F4, F4G1, I1J1 and J1K4. Path 2 contains PIP A2B2, B2C3, E3F3, F3G2, I2J2 and J2K3. Path 3 consists of PIP A3B3 ,B3C2, E2F2, F2G3, I3J3 and J3K2. Path 4 consists of PIP A4B4, B4C1, E1F1, F1G4, I4J4 and J4K1.

In the third configuration, 4 paths have been assigned. Path 1 consists of PIP A1C1, E1G1, and I1K1. Path 2 contains PIP A2C2, E2G2, and I2K2. Path 3 consists of PIP A3C3, E3G3, and I3K3. Path 4 consists of PIP A4C4, E4G4, and I4K4.

IV. Experimental Results

We have generated test configurations for Virtex (XCV300), Viretx-II, Virtex 4 and Virtex-5 FPGAs. We have used Z3 SMT solver [20] for solving the SMT problem generated for each circuit.

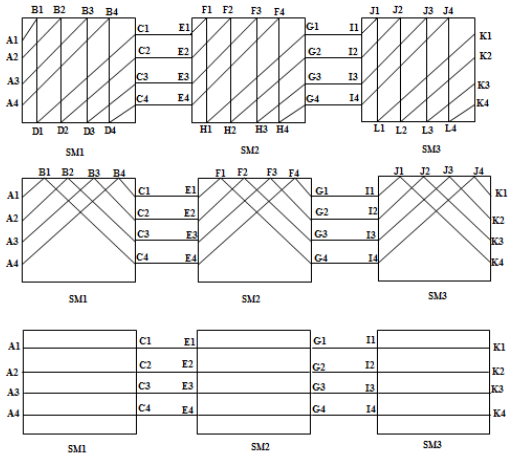

FIG. 6: Test configurations generated using proposed approach

Circuit	Number of Test Configurations required	
	Proposed Method	Existing method [4]
Virtex	10	11
Virtex-II	24	25
Virtex 4	26	28

TABLE I: Comparison of simulation results using proposed method and Existing method [4]

Table I and table II shows comparison of the test configuration number of the proposed method with [4] for Virtex, Virtex II, Virtex 4 and [21] for XC4010 and Virtex-5. XCV300 of Virtex FPGA contains 24 single lines and 12 hex lines. Virtex-II contains 120 hex lines, 40 double lines and 24 long lines. Virtex-5 contains total 42 double lines, 120 pent lines and 18 long lines, and Virtex 4 contains total 40 double lines, 120 hex lines and 24 long lines [22]. Our approach is able to reduce the number of test configurations for XCV300 as well as Virtex-II and Virtex 4. All the line segments and associated PIPs have been included in the test configuration generation process.

V. Conclusion

This paper has proposed a test configurtion strategy for different Xilinx Virtex FPGA families in application-independent

Circuit	Number of Test Configurations required	
	Proposed Method	Existing method [21]
XC4010	5	6
Virtex 5	58	56

TABLE II: Comparison of simulation results using proposed method and Existing method [21]

testing scenario. As FPGA test time is dominated by the number of test configurations, therefore it is important to generate minimum number of test configurations without decreasing the fault coverage. The target fault list consists of opens and shorts faults in the FPGA interconnect. To obtain the minimum number of interconnect test configurations, the presented approach uses satisfiability modulo theory. Test configuration numbers have been compared with [4] and [21] for different FPGA architectures. This work can be further extended by generating the bit stream file using commercial tool.

ACKNOWLEDGMENT

This work was supported by the Department of Electronics and Information Technology, Ministry of Communication and IT, Government of India under the Visvesvaraya PhD Scheme administered by Digital India Corporation (formerly Media Lab Asia).

REFERENCES

[1] M. Tahoori, "Application-dependent testing of fpgas," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 9, pp. 1024–1033, Sep. 2006.

[2] C. W. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli, "Satisfiability modulo theories." *Handbook of satisfiability*, vol. 185, pp. 825–885, 2009.

[3] L. De Moura and N. Bjørner, "Satisfiability modulo theories: Introduction and applications," *Communications of the ACM*, vol. 54, no. 9, pp. 69–77, Sep. 2011.

[4] A. Ruan, J. Yang, L. Wan, B. Jie, and Z. Tian, "Insight into a generic interconnect resource model for xilinx virtex and spartan series fpgas," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 11, pp. 801–805, Nov 2013.

[5] M. B. Tahoori and S. Mitra, "Application-independent testing of fpga interconnects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 11, pp. 1774–1783, Nov 2005.

[6] M. B. Tahoori, S. Mitra, S. Toutounchi, and E. J. McCluskey, "Fault grading fpga interconnect test configurations," in *Proceedings. International Test Conference*, 2002, pp. 608–617.

[7] J. Zhao, J. Feng, T. Lin, and Z. Tong, "A novel fpga manufacture-oriented interconnect fault test," in *2008 9th International Conference on Solid-State and Integrated-Circuit Technology*, Oct 2008, pp. 2091–2094.

[8] J. Yao, B. Dixon, C. Stroud, and V. Nelson, "System-level built-in self-test of global routing resources in virtex-4 fpgas," in *2009 41st Southeastern Symposium on System Theory*, March 2009, pp. 29–32.

[9] M. B. Tahoori, "High resolution application specific fault diagnosis of fpgas," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 10, pp. 1775–1786, Oct 2011.

[10] J. Smith, T. Xia, and C. Stroud, "An automated bist architecture for testing and diagnosing fpga interconnect faults," *Journal of Electronic Testing*, vol. 22, no. 3, pp. 239–253, Jun 2006.

[11] I. G. Pereira, L. A. Dias, and C. P. de Souza, "A shift-register based bist architecture for fpga global interconnect testing and diagnosis," *Journal of Electronic Testing*, vol. 31, no. 2, pp. 207–215, Apr 2015.

[12] T. N. Kumar, H. A. Almurib, and N. Chin-Ee, "Fine grain faults diagnosis of fpga interconnect," *Microprocessors and Microsystems*, vol. 37, no. 1, pp. 33 – 40, 2013.

[13] C. L. Hsu and T. H. Chen, "Built-in self-test design for fault detection and fault diagnosis in sram-based fpga," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 7, pp. 2300–2315, July 2009.

[14] X. Sun, K. Ogden, H. Chan, and P. Trouborst, "A novel fpga local interconnect test scheme and automatic tc derivation/generation," *Journal of Systems Architecture*, vol. 50, no. 5, pp. 267 – 280, 2004, design and Test of Systems on a Chip.

[15] T. D. A. W. Ruan and P. L. B. R. Jie, "A bitstream readback based fpga test and diagnosis system," in *2014 International Symposium on Integrated Circuits (ISIC)*, Dec 2014, pp. 592–595.

[16] V. Nelson, C. Stroud, J. Yao, and B. Dixon, "System-level built-in self-test of global routing resources in virtex-4 fpgas," in *Southeastern Symposium on System Theory(SSST)*, vol. 00, 03 2009, pp. 29–32.

[17] A. Ruan, H. Huang, J. Wang, and Y. Zhao, "A routability-aware algorithm for both global and local interconnect resource test and diagnosis of xilinx sram-fpgas," *Journal of Electronic Testing*, vol. 32, no. 6, pp. 749–762, Dec 2016.

[18] M. B. Tahoori and S. Mitra, "Automatic configuration generation for fpga interconnect testing," in *Proceedings. 21st VLSI Test Symposium, 2003.*, April 2003, pp. 134–139.

[19] Z. Yang, C. Liang, J. Wang, and J. Lai, "A new automatic method for testing interconnect resources in fpgas based on general routing matrix," *IEICE Electronics Express*, vol. 12, no. 20, pp. 20 150 747–20 150 747, 2015.

[20] L. de Moura and N. Bjrner, "Z3 - a tutorial," Microsoft, Tech. Rep., 2012.

[21] A. Ruan, B. Jie, L. Wan, J. Yang, C. Xiang, Z. Zhu, and Y. Wang, "A bitstream readback-based automatic functional test and diagnosis method for xilinx fpgas," *Microelectronics Reliability*, vol. 54, no. 8, pp. 1627 – 1635, 2014.

[22] P. Minev and V. Kukenska, "The virtex-5 routing and logic architecture," 2009.