A Bitstream Readback Based FPGA Test and Diagnosis System

T. Du A.W. Ruan* P. Li B.R. Jie
State Key Laboratory of Electronic Thin Films and Integrated Devices, University of Electronic Science and Technology of China
Chengdu, 610054, China
Phone: +86-28-83201942, Email: ruanaiwu@uestc.edu.cn

***Abstract*—This paper presents a bitstream readback based automatic functional testing system for field-programmable gate array (FPGA) test and diagnosis. The proposed testing system physically consists of software tools in a PC and an FPGA-under-test with a communication channel in between. Hardware overhead is minimized due to no additional hardware required. Some software tools for boundary scan controlling, bitstream parsing as well as fault test and diagnosis are in-house developed. Leveraging the bitstream parsing tool, current states of D flip-flops (DFFs) and configuration memory cell along with their absolute addresses can be obtained from readback bistreams and bitstream parsing library, respectively. Fault types and physical locations can be further determined by the fault test and diagnosis tool. The issue of IOB number limitations not addressed well by some previous work can be partly relieved along with reduced configuration numbers and improved diagnostic resolution. The BIST system is evaluated by testing several Xilinx series FPGAs, and experimental results are provided.**

Keywords—FPGA; Bitstream; Readback; Test; Diagnosis

I. INTRODUCTION

FPGAs have increasingly played an important role in modern electronic industry due to their reconfigurability, flexibility, low development cost, and reduced time-to-market. Thanks to shrinking size of transistors fabricated by nano-CMOS manufacturing process, more and more complicated structures of FPGAs can be implemented. As a result, more and more functions can be realized by FPGAs. However, several types of defects can be introduced by the manufacturing process, such as stuck-at faults, delay faults, etc. A study showed that FPGAs at and beyond the 45 nm technology node have low yield [1]. FPGAs have high density of transistors and interconnect wires. Hence, after FPGAs have been fabricated, they are tested extensively to find faulty ones from the batch.

Some functional testing methods such as intrusive and non-intrusive BIST, JBits and bitstream readback for FPGA test and diagnosis have been extensively explored [2-11, 14]. However, the main issues not addressed well are configuration numbers, diagnostic resolution, as well as I/O pin numbers.

In this paper, a bitstream readback based automatic functional testing system for FPGA test and diagnosis is presented. The proposed bitstream readback based FPGA testing system along with FPGA test algorithms can address the aforementioned disadvantages of some FPGA testing approaches based on intrusive BIST, non-intrusive BIST and readback techniques.

The paper is organized as follows. In Section 2, some techniques for FPGA test is introduced. Our testing system is presented in Section 3. In this section, the architecture and operational principle of the proposed testing system are described in detail. In Section 4, implementation of the proposed testing system with accompanying algorithms for FPGA test and diagnosis are accounted for, and experiment is conducted on several Xilinx series FPGAs. We conclude in Section 5.

II. RELATED WORK

In this section, we introduce some current FPGA testing approaches as well as their drawbacks.

Current FPGA test techniques include traditional test technique, Built-in Self Test (BIST) and JBits. The traditional test technique refers to utilize standard, commercially available test machines to test FPGAs in a production environment. The FPGA-under-test is configured before TVs are written to the device via external input pins. Test results are typically collected through other device output pins. Then, the test results are compared to expected data to decide whether there are faults in the FPGA-under-test. This technique leverages standard tools and methodologies and requires no specific changes to the FPGA hardware. However, this technique has the disadvantages such as limitation of embedded memory depth and I/O pin counts of test machines available for the FPGA-under-test [13]. As a result, the ability to move test vectors and results on and off the FPGA-under-test, as well as configuration numbers are restricted.

The limitation of I/O pin counts can be solved by BIST approach [2-8]. It typically requires additional circuitry and functionality are incorporated into the design of the circuit to facilitate the self-testing feature. This additional functionality is implemented by a test pattern generator (TPG) and an output response analyzer (ORA). A sequence of patterns for testing the FPGA-under-test is produced by the TPG, while the ORA determines whether the output responses are corresponding to those of a fault-free circuit. However, internal FPGA memory required for storing test vectors and response data can lead to additional overhead in the FPGA-under-test. Further, the parts occupied by the TPGs and ORAs have to be assumed fault-free. This assumption is not always true. Consequently, configuration numbers are increased because the sections occupied by TPGs and ORAs are frequently exchanged for those occupied by circuits-under-test (CUTs) to bring every section of the FPGA to be tested.

Xilinx provides JBits tool, a DARPA-funded research, to give low-level read/write access to the bitstream of Virtex and

978-1-4799-4833-8/14/$31.00

Virtex-II FPGAs [14].The ability to probe and stimulate internal states eliminates traditional testing concerns about I/O pad limitation, coverage requirements as well as the need for storing numerous bitstream files and TVs. However, JBits does not provide access to bitstreams of Virtex-4 and above series FPGAs, and is not unmaintained by Xilinx.

Since most FPGAs support reading configuration memory through readback instruction, authors of reference [10, 11] study whether errors occur or not during configuration by comparing the readback bitstream to the fault-free configuration bitstream. This method can only be used for SRAM testing rather than IR and CLB testing and diagnosis. The reason for this is that we cannot locate where the required information, i.e. the logic values in the IRs and CLBs with their corresponding addresses in the readback bitstream without detailed knowledge about configuration memory map, which is very reluctant to be provide by FPGA vendors.

III. Proposed FPGA Testing System

A. Architecture of Proposed FPGA Testing System

As shown in Fig.1, the architecture of the proposed FPGA testing system is composed of some software tools in a PC and an FPGA-under-test with a communication channel in between.

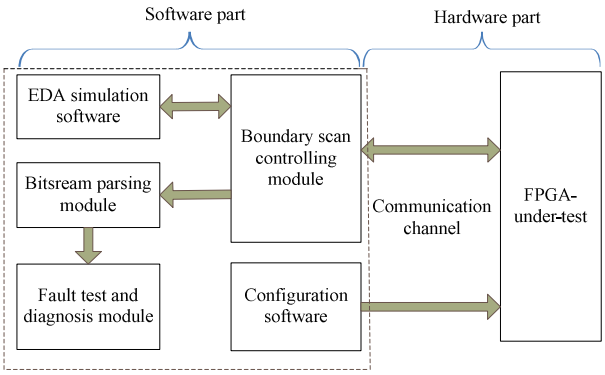

Fig. 1. Block diagram of bitstream readback based FPGA test system

The software part includes a boundary scan controlling module, a bitstream parsing module, a fault test and diagnosis module, a commercial EDA simulation software tool as well as an FPGA configuration software tool. The configurations for the FPGA-under-test are generated and downloaded by the commercial configuration software tool. The boundary scan controlling module implements application of test vectors (TVs) to the FPGA-under-test as well as bitstream readback. The bitstream parsing module carries out parsing the readback bitstreams and retrieving the response values from the readback bitstreams. The fault test and diagnosis module can find and position faults in the FPGA-under-test. The TVs and readback bitstream are in the text format to facilitate TVs revision as well as readback bitstream processing.

A JTAG cable acts as the communication channel between the software part and the FPGA-under-test to conduct downloading configurations for the FPGA-under-test, application of TVs as well as bitstream readback.

B. Operational Principle of the Proposed Testing System

1) Readback

Most FPGAs support reading configuration memory as well as the contents of DFFs. Readback is the process of reading back all the data in the form of bitstream from the configuration memory and DFFs of an FPGA. The readback process is actually the inverse process of configuring the FPGA. The bitstream read back from the configuration memory includes state information of internal registers or storage units. This state information is vital for FPGA test and diagnosis.

Readback Verify and Readback Capture are two available readback modes [12]. During Readback Verify, the user can get access to all configuration memory cells, including the current values in all user memory elements, which refer to Look-up Tables (LUTs) operating in RAM mode and block RAMs. Comparing readback bitstream against configuration bitstream, we can judge whether the configuration memory cells in the FPGA are good or not, or whether the FPGA configuration is successful or not.

Readback Capture is a superset of Readback Verify − in addition to reading all configuration memory cells, the current states of all internal DFFs in CLBs and IOBs are read, and are useful for test. In other words, combining Readback Capture with corresponding FPGA test and diagnosis algorithms, we can test and diagnose FPGAs by comparing data extracted from readback bitstream against expected data. We have studied bitstream readback based algorithms for IOBs, CLBs and IRs and will describe in detail in Section 3.B.3

Typically, TVs application and test responses collection for an FPGA-under-test are through IOBs. In other words, IOBs are also used as signal monitoring. However, due to limitation of programmable IOB number, it is impossible to cover all resources of the FPGA-under-test with a single test configuration. Accordingly, the FPGA test and diagnosis algorithms have to be designed carefully to cover as many FPGA resources as possible with a reasonable small number of test configurations. Further, diagnostic ambiguity can occur due to lack of sufficient IOBs. When the bitstream readback technique is introduced, the current state of all internal DFFs in CLBs and IOBs can be read by Readback Capture mode. Put another way, all DFFs in the FPGA-under-test are available for signal monitoring. As a result, most of the fault masking is eliminated. Restriction of IOB number is partially relieved leading to relax requirement for the FPGA test and diagnosis algorithms as well as test procedures.

2) Bitstream parsing module

After reading back the bitstreams from the configuration memory cells and DFFs, the required information must be extracted from the bitstreams. The required information refers to logic values of the DFFs and configuration memory cells which can be used for FPGA test and diagnosis. If we want to extract these logic values from readback bitstream, we must know the absolute addresses of these logic values in the readback bitstream. On the other hand, names and physical locations of the DFFs and configuration memory cells corresponding to the extracted information have to be identified in the architecture of the FPGA-under-test. If one

retrieved data is not what we expect, then we can detect the exact position of the fault.

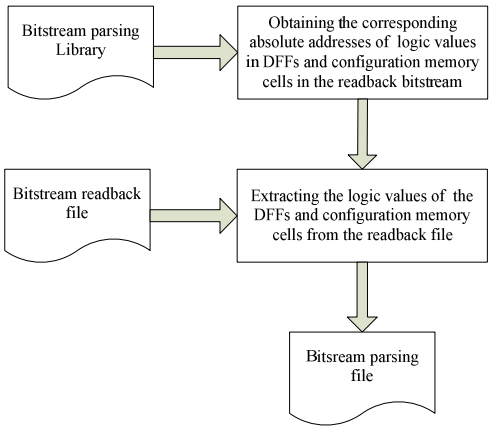

Fig. 2. Bitstream parsing flow

A bitstream parsing library containing names, physical locations of the DFFs and configuration memory cells accompanying with corresponding absolute addresses of logic values in the bitstreams have been designed. The bitstream parsing library for Virtex-II FPGA is displayed in TABLE I. For example, Block=SLICE_X1Y79 denotes the location of a DFF in Slice1of a CLB at row 79 and column 1. After looking up Table 1, we can find the output of the DFF is XQ with its current state in the Bit 108660 of the readback bitstream. The second example is about a LUT configured as a RAM. Block=SLICE_X0Y79 refers to the location of the LUT_G in Slice1 of a CLB at row 70 and column 0. According to TABLE I, the current state of the first address bit in the 16-bit RAM memory can be determined in the Bit 108679 of the readback bitstream. In the third instance, a PIP named W2END1 -> N2BEG3 in an SM refers to the first double wire in the west connecting to the third double wire in the north. Similarly, the bitstream parsing libraries for XCV300 and Virtex-5 FPGAs have also been developed.

The in-house developed bitstream parsing module manages to parse the readback bitstreams to get a bitstream parsing file based on the bitstream parsing library. The diagram of the bitstream parsing module is illustrated in Fig 2.

TABLE I. BITSTREAM PARSING LIBRARY FOR VIRTEX-II FPGA

Absolute location of Bit			**Physical location in Virtex FPGA**	
DFFs in CLBs	Bit	108660	Block=SLICE_X1Y79	Latch=XQ
	Bit	112052	Block=SLICE_X1Y79	Latch=YQ
	Bit	108700	Block=SLICE_X1Y78	Latch=XQ
	Bit	112092	Block=SLICE_X1Y78	Latch=YQ
	…	…	…	…
LUT configured as RAM mode	Bit	108679	Block=SLICE_X0Y79	RAM=M:0
	Bit	108678	Block=SLICE_X0Y79	RAM=M:1
	Bit	108654	Block=SLICE_X0Y79	RAM=M:30
	Bit	108655	Block=SLICE_X0Y79	RAM=M:31
	…	…	…	…
PIPs of SM	Bit	1502438	Block=SLICE_X0Y79	pip BX0 -> BY3
	Bit	1505193	Block=SLICE_X0Y79	pip BX0 -> F2_B1
	Bit	1505199	Block=SLICE_X0Y79	pip BX0 -> F2_B3
	Bit	1505201	Block=SLICE_X0Y79	pip BX0 -> F3_B0
	…	…	…	…

3) Fault test and diagnosis module

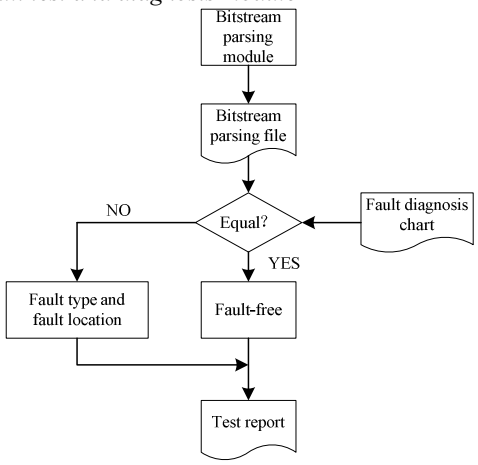

Fig. 3. Fault test and diagnosis flow

Fault types and their locations can be determined in terms of a fault diagnosis chart. The fault test and diagnosis module compares the retrieved states of DFFs and configuration memory cells from the bitstream against the fault diagnosis chart as shown in Fig.3.

It should be noted that fault diagnostic resolution is dependent on algorithms for FPGA test and diagnosis, whereas the algorithms are relevant to IOB numbers. The detail of the algorithms is introduced in our paper [15]. Instead of observing response values through I/O pins, our bitstream readback based FPGA test system employs DFFs in CLBs as monitoring points. It is well known that DFF number is much larger than IOB number in an FPGA. Consequently, our test system can provide good observability with fewer configuration numbers.

IV. EVALUATION OF THE PROPOSED TEST SYSTEM

A bitstream readback based FPGA test and diagnosis system has been developed. The proposed FPGA testing system physically consists of software tools in a PC and an FPGA-under-test with the JTAG-to-USB cable in between. Except some available commercial EDA software tools, the other modules or software tools are developed based on the algorithms mentioned in paper [15].

When the proposed test system is utilized for testing an FPGA, test steps are listed as follows:

- Configuration files are delivered to the FPGA-under-test through boundary scan module and JTAG-to-USB cable. We can use iMPACT to perform JTAG-based Readback Verify to determine whether there are any configuration errors.
- TV application for the FPGA-under-test is applied following by execution of bitstream readback function to read back bitstream from the FPGA-under-test. The readback bitstream is saved in a file format.
- The bitstream parsing module extracts the response values from the bitstream readback file and save them in a bitstream parsing file.

- The fault test and diagnosis module compares the retrieved states of DFFs and configuration memory cells from the bitstream against the fault diagnosis chart to determine whether there faults or not, fault types and locations.

Each module execution can be controlled by a batch file containing TCL script commands to implement FPGA test and diagnosis automatically.

TABLE II. EXPERIMENTAL RESULTS OF XILINX FPGAS

FPGA	XCV300		XC2V1000		XC5VLX110t	
Resources	CLB	IR	CLB	IR	CLB	IR
Configuration numbers	3	12	3	25	3	56
Configuration time	7s × 3	7s × 12	10s × 3	10s × 25	20s × 3	20s × 56
Test application time	7s × 3	7s × 12	25s × 3	25s × 25	110s ×3	110s × 56

TABLE III. FAULT DIAGNOSIS CHART FOR VIRTEX-II FPGAS

Wire1-LUT_F1 stuck-at-0	SLICE_X1Y79	1100111111111100
Wire2-LUT_F2 stuck-at-0	SLICE_X1Y79	1010010111111010
…	…	…
Wire3-LUT_G1 stuck-at-1	SLICE_X1Y79	0110011011111111
Wire4-LUT_G2 stuck-at-1	SLICE_X1Y79	0110110101101101
…	…	…
Wire1-LUT_F1, wire2-LUT_F2 bridging	SLICE_X1Y78	0000111111111110
Wire3-LUT_G1, wire4-LUT_G2 bridging	SLICE_X1Y78	0110011001101110
...	...	…

Xilinx FPGAs including an XQVR300CB288 in Virtex FPGAs, an XC2V1000BGA575 in Virtex-II FPGAs and an XC5VLX110t in Virtex-5 FPGAs were tested by the proposed bitstream readback based FPGA test system in the experiment and the experimental results are listed in TABLE II. Fault diagnosis charts for the three FPGAs including fault types, the locations of the faults as well as faulty outputs can be derived based on the algorithms in paper [15], bitstream parsing approach in Section 3.B.3 and corresponding input TVs. The fault diagnosis chart for Virtex FPGAs is listed in TABLE III. The experiments were performed on a 2.8GHz Intel Pentium 4 microprocessor with 2GB RAM.

V. CONCLUSION

This paper has proposed a bitstream readback based automatic functional testing system for FPGA test and diagnosis. The FPGA test system is generic and can be applied to Xilinx Virtex series FPGAs test and Xilinx Spartan FPGAs test with small modifications. The test system can provide good diagnostic resolution for an FPGA-under-test. No matter what array size of an FPGA-under-test is, the CUT can be tested automatically, and repeatedly. Restriction of IOB number is partially relieved leading to relax requirement for the FPGA test and diagnosis algorithms as well as test procedures.

REFERENCES

[1] Campregher N., Cheung Y. K. Vasilko M. Analysis of yield loss due to random photolithographic defects in the interconnect structure of FPGAs. In: ACM international workshop on FPGAs; 2005. p. 138-148.

[2] Stroud C, Konala S, Chen P, Abramovici M. Built-in self-test of logic of blocks in FPGAs. In: Proceedings of VLSI test symposium; 1996. p. 387-392.

[3] Stroud C, Wijesuriya S, Hamilton C. Built-in self-test of FPGA interconnect. In: Proceedings of test conference; 1998. p. 404-411.

[4] Bradley D, Stroud C. Built-in self-test of configurable logic blocks in Virtex-5 FPGAs. In: Proceedings of 41st southeastern symposium on system theory; 2009. p. 230-234.

[5] Dutt S. Mixed PLB and interconnect BIST for FPGAs without fault-free assumptions. In: Proceedings of the 24th IEEE VLSI test symposium; 2006. P. 36-43.

[6] Dutt S, Verma V, Suthar V. Built-in self-test of FPGAs with provable diagnosis abilities and high diagnostic coverage with application to online testing. In: IEEE transactions on computer-aided design of integrated circuits and systems; 2008. P. 309-326.

[7] Hsu CL, Chen TH. Built-in self-test design for fault detection and fault diagnosis in SRAM-based FPGA. In: IEEE transactions on instrumentation and measurement; 2009. P. 2300-2315.

[8] Kumar TN, Chong CW. An automated approach for locating multiple faulty LUTs in an FPGA. In: Microelectronics Reliability; 2008. Vol. 48, Issues 11-12, P. 1900-1906.

[9] Ruan AW, Kang S, Wang Y, Han X, Zhu ZJ, Liao YB, Li P. A Built-in self-test (BIST) system with non-intrusive TPG and ORA for FPGA test and diagnosis. In: Microelectronics Reliability; 2013.Vol. 53, Issues 3, P. 488-498.

[10] He W, Wang YK, Xing KF, Chen L. SEU readback interval strategy of SRAM-based FPGA for space application. In: Proceedings of IEEE international conference on computer science and automation engineering; 2011. P. 238-241.

[11] Paulsson K, Viereck U, Hübner M, Becker J. Exploitation of the external JTAG interface for internally controlled configuration readback and self-reconfiguration of Spartan 3 FPGAs. In: Proceedings of IEEE computer society annual symposium on VLSI; 2008. P. 304-309.

[12] Virtex FPGA series configuration and readback, Xilinx Inc., XAPP138 (v2.8); 2005.

[13] Sundararajan P, McMilan S, Guccione S. Testing FPGA devices using JBits, Xilinx Inc. 2100 Logic Drive. San Jose, CA 95124 (USA).

[14] Singh S, Jame-Roxby, P. Lava. JBits: from HDL to bitstreams in seconds. In: IEEE symposium on FPGAs for custom computing machines; 2001.P. 91-100.

[15] Aiwu Ruan, Bairui Jie, et al. A bitstream readback-based automatic functional test and diagnosis method for Xilinx FPGAs; In: Microelectronics Reliability ,Vol.54, Issue 8, P.1627-1635, 2014.