Microelectronics Reliability 54 (2014) 1627–1635

Contents lists available at ScienceDirect

Microelectronics Reliability

journal homepage: www.elsevier.com/locate/microrel

A bitstream readback-based automatic functional test and diagnosis method for Xilinx FPGAs

Aiwu Ruan *, Bairui Jie, Li Wan, Junhao Yang, Chuanyin Xiang, Zujian Zhu, Yu Wang

State Key Laboratory of Electronic Films and Integrated Devices, University of Electronic Science and Technology of China, Chengdu 610054, China

ARTICLE INFO

Article history:
Received 15 January 2013
Received in revised form 12 March 2014
Accepted 25 March 2014
Available online 24 April 2014

Keywords:
FPGA
Readback
Bitstream
Test
Diagnosis

ABSTRACT

In this paper, a novel bitstream readback-based test and diagnosis method including a bitstream parsing algorithm as well as a corresponding bitstream readback-based fault and diagnosis algorithm for Xilinx FPGAs is presented. The proposed method can be applied to both configurable logic block (CLB) and interconnect resource (IR) test. Further, the algorithm is suitable for all Virtex and Spartan series FPGAs. The issues such as fault coverage, diagnostic resolution, I/O numbers, as well as configuration numbers not addressed well by some previous works can be solved or partly relieved. The proposed method is evaluated by testing several Xilinx series FPGAs, and experimental results are provided.

1. Introduction

An FPGA is composed of a large amount of repeated and regular configurable logic blocks (CLBs), interconnect resources (IRs), and Input/Output blocks (IOBs), each of which incorporates logic gates, D Flip-Flops (DFFs) and control units. FPGAs have increasingly played an important role in modern electronic industry due to their reconfigurability, flexibility, low development cost, and reduced time-to-market, since they were introduced by Philips in the early 1970s. Applications for FPGAs are diversified, e.g. communications, storage systems, adaptive computing, etc.

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 and delay faults. 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.

Two engineers from Xilinx summarize special challenges of FPGA test methodology in their paper [2]. The first three challenges can be translated to generic, scalable and reusable test and diagnosis algorithms for CLBs and IRs. The last challenge means that metrics of fault coverage, diagnosis resolution, and test configuration numbers should be used. In other words, the main issues should be addressed in FPGA test and diagnosis are generic test strategies, fault coverage, diagnostic resolution, I/O numbers, as well as configuration numbers. As we all know, test for an FPGA is supposed to consist of two steps. The first step is to configure the FPGA-under-test. Then, the configured FPGA-under-test can be scanned after applied by test vectors (TVs) in the second step. Usually, time spent on configuration is much greater than time spent on scan steps. In other words, configuration time is determined by the configuration numbers and size of configuration bitstream. On the other hand, when an FPGA array size grows, the configuration numbers will increase exponentially. As a result, test time will reach to astronomical numbers if every resource in the FPGA-under-test is covered [2].

On the other hand, the amount of input and output bandwidth available to the FPGA-under-test, or IOB numbers available to test, affects the metrics of fault coverage, diagnosis resolution, and test configuration numbers. This is due to the fact that the capability to move test vectors and results on and off the FPGA-under-test is decided by the available IOB numbers.

All previous research targets at the aforementioned main issues. Many works extensively investigated CLB testing methods or IR testing methods. Only very little research has been performed on both IR and CLB test. An automated approach for locating multiple faulty Look-Up tables (LUTs) in an FPGA is presented in [3]. This work does not study how to identify faulty types for LUTs. If stuck-at-1/0 faults are required to be tested, the actual configuration number is two

* Corresponding author. Tel.: +86 2883201942.
E-mail address: ruanaiwu@uestc.edu.cn (A. Ruan).

http://dx.doi.org/10.1016/j.microrel.2014.03.017
0026-2714/

times the number required in the paper. An offline and an online Built-In-self-Test (BIST) approaches for CLB test and diagnosis are described in [4–6], respectively. In paper [7], the proposed BIST approach was utilized to test CLBs of an N × M FPGA. N and M denote row and column numbers of the FPGA-under-test. The configuration numbers are 2 × N × M for BISTer-1case and 4 × N × M for BISTer-2 case, respectively. The limitation of I/O pin counts can be solved by BIST approach. However, the parts occupied by the test pattern generators (TPGs) and output response analyzers (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. Further, the configuration numbers are strongly dependent on FPGA array size as well as complexity of TPGs and ORAs.

In Ref. [8], IRs in XC4000 FPGAs are modeled by adjacency graphs. Thus, a unified local and global IR test scheme is proposed. An automatic test-configuration technique for IRs in Virtex FPGA is presented in [9]. Interleaving approach is employed to minimize configuration numbers. However, fault masking may occur due to the configuration based on interleaving approach actually a cascaded chain. Work presented in [10] discusses test for IRs in Virtex-4 FPGAs. Ref. [11] proposed a fine grain fault diagnosis method for IRs, which cascades the LUTs to propagate the value of a specific net under test to the primary output for verification against the expected value of the net. IR faults detected at the primary output are back-tracked until the faulty nets are precisely located. Fault ambiguity may occur in the presence of multiple faults. IR full coverage carried out by these schemes fails to be obtained because programmable-interconnect-points (PIPs) between different types of wire segments are not considered. Further, these techniques are not generic and cannot be extended to other Virtex series FPGAs.

Authors of paper [12] classified CLBs in an XC4000 FPGA into TPGs, ORAs and CUTs. The CUTs include not only the CLBs under test, but the corresponding IRs as well. Thus, the test scheme proposed in the paper can be applied to both CLBs and IRs. However, the configuration numbers are strongly dependent on FPGA array size as well as complexity of TPGs and ORAs. Further, the scheme cannot be extended to Virtex series FPGAs because IRs in Virtex series FPGAs are much complicated than those in XC4000 FPGAs.

Since most FPGAs support reading configuration memory through readback instruction, authors of Refs. [13,14] 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 Static random access memory (SRAM) testing rather than IR and CLB testing and diagnosis. The reason for this is that the method 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.

In this paper, a novel bitstream readback-based test and diagnosis method including a bitstream parsing algorithm as well as a corresponding bitstream readback-based fault and diagnosis algorithm for Xilinx FPGAs is presented. Leveraging the bitstream parsing algorithm, current states of DFFs and configuration memory cells 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 algorithm. The proposed method can be applied to both CLB and IR test. Further, the algorithm is suitable for all Virtex and Spartan series FPGAs. Good diagnostic resolution for an FPGA-under-test can be achieved. No matter what array size of an FPGA-under-test is, the FPGA-under-test can be tested automatically and repeatedly with higher fault coverage. The issue of IOB number limitations not addressed well by some previous work can be partly relieved. Configuration numbers are reduced compared with BIST techniques.

The rest of the paper is organized as follows. In Section 2, our proposed scheme is presented. In this section, the proposed testing system with accompanying algorithms for FPGA test and diagnosis is described in detail. In Section 3, implementation of the proposed testing system is accounted for, and experiment is conducted on several Xilinx series FPGAs. We conclude in Section 4.

2. Proposed FPGA test and diagnosis method

In this section, two proposed algorithms, bitstream parsing algorithm as well as fault test and diagnosis algorithm are discussed, respectively. 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 can be derived from the algorithm. On the other hand, since the values of DFFs can be readback from readback bitstreams, the DFFs in FPGA can be used as output IOBs to collect responses. Therefore, the fault test and diagnosis algorithm for CLBs and IRs is based on DFFs which are distributed widely in FPGA to monitor the outputs of CLBs and IRs.

2.1. Bitstream parsing algorithm

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 readback from the configuration memory includes state information of internal registers or storage units. This state information is vital for FPGA test and diagnosis [16].

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.

So far, very few references about bitstream parsing algorithm can be found except Ref. [17] which presents an attempt to reverse bitstream by employing a correlation approach. However, the algorithm lacks an exhaustive list of available resources and has no means of determining if it has decoded all resources successfully. Further, the project has since been stopped. In contrast, our algorithm can exhaustively reverse bitstream of all available resources as can be seen in the following part.

Before the bitstream parsing algorithm is described, we have to assume that locations of PIPs and associated CLBs are regular. This hypothesis holds true because FPGAs have highly regular architectures. The second hypothesis we make is that an FPGA consists of many repeatable building blocks (RBBs). The repeatable building block comprises a switch matrix (SM) and an accompanying CLB. This hypothesis is necessary due to the fact that any response data will be retrieved from DFFs and SRAMs in the CLBs. The third hypothesis is that the function f relates configuration frames to actual physical sites on the FPGA. Put another way, there is a relationship between configuration memory bit mapping and actual physical information on the FPGA.

The function f can be achieved by generating a Xilinx Design Language (XDL) design file X_{B_i} with a preferably minimal resource change compared to a reference XDL design file X_{A_i}. File X_{A_i}, X_{B_i} should meet conditions ①, ②. E is a set of resources in a RBB and I is the number of elements in the set E. Then the two XDL files X_{A_i}, X_{B_i} are converted into bitstream files which are downloaded into an FPGA-under-test. Consequently, two bitstream readback files R_{A_i}, R_{B_i}, can be achieved. Comparing the R_{B_i} to R_{A_i}, we can get the equation of function f_i Eq. (1), d_i is the DFF and configuration cell and, D_i is the location of the logic value of the DFF and configuration cell in the readback bitstream. The flowchart of the algorithm is shown in Fig. 1.

① $X_{A_i}, X_{B_i} \neq \Phi$

② $X_{A_i} \subset X_{B_i},\ X_{B_i} - X_{A_i} = \{d_i\} \in E$

$$D_i = f_i(d_i) \tag{1}$$

$$P(f) = \cup_i^I P(f_i) = \cup_i^I \{D_i = f(d_i)\} \tag{2}$$

As a result, function f_i can be established between d_i and D_i. Thus, the bitstream parsing library of a RBB can be achieved by repeating the bitstream parsing procedure until each function f_i is obtained according to Eq. (2). On the other hand, in terms of the second hypothesis, an FPGA consists of many RBBs. Consequently, other bitstream parsing library $P_{u,v}(f)$ of RBBs can be derived from Eqs. (3) and (4). $P_{m,n}(f)$ is the bitstream parsing library for a RBB located at row m and column n which has been already gotten according to Eqs. (1) and (2).

$$P_{u,v}(f) = \cup_i^I P_{m,n}(f_i) + A_{u,v}C, \quad u \neq m,\ v = n \tag{3}$$

If these RBBs are in the identical column with the $RBB_{m,n}$, $A_{u,v} = u - m$, C is a constant.

$$P_{u,v}(f) = \cup_i^I P_{m,n}(f_i) + A_{u,v}C + B_{u,v}D$$

$$u \neq m,\ v \neq n \tag{4}$$

If these RBBs are not in the identical column with the $RBB_{m,n}$, $A_{u,v} = u - m$, $B_{u,v} = v - n$. C and D are two constants. Please note that $0 \leq u,\ m \leq M,\ 0 \leq v,\ n \leq N$, M and N are row and column numbers of an FPGA.

Therefore, the bitstream parsing library of the FPGA can be derived by Eq. (5).

$$P = \cup_{u,v}^{M,N} P_{u,v}(f) \tag{5}$$

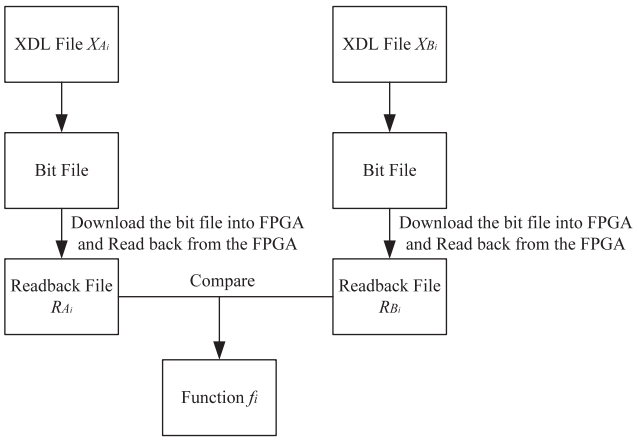

Fig. 1. Bitstream parsing flow.

In this way, the bitstreams of all available resources of an FPGA-under-test can be reversed exhaustively. The bitstream parsing library for Virtex FPGA derived from the algorithm is displayed in Table 1. For example, Block = CLB_R1C25. S1 denotes the location of a DFF in Slice1of a $CLB_{1,25}$ at row 1 and column 25. After looking up Table 1, we can find the output of the DFF is XQ with its current state in the Bit 7443 of the readback bitstream. The second example is about a LUT configured as a RAM. Block = CLB_R1C25. S1 refers to the location of the LUT_G in Slice1of the $CLB_{1,25}$. According to Table 1, the current state of the first address bit in the 16-bit RAM memory can be determined in the Bit 6100 of the readback bitstream. In the third instance, if a PIP in a SM corresponding to the $CLB_{16,24}$ located at row 16 and column 24, it connects single line E0 to single line S6. The current state of the SRAM controlling the PIP is in the Bit 64,842 of the readback bitstream in terms of Table 1. E0 and S6 denote the 1st single line in the east of the SM and the 7th single line in the south of the SM, respectively. Similarly, the bitstream parsing libraries for XC4000 and Virtex-5 FPGAs have also been developed based on the algorithm.

2.2. Fault test and diagnosis algorithm

The bitstream readback-based algorithm for FPGA fault test and diagnosis, covering internal FPGA resources such as CLBs and IRs are studied.

2.2.1. Fault types

In light of Ref. [2], The CLB and IR faults in this paper are classified into the following categories:

- Stuck-at 0/1 fault on LUTs.
- Functional faults on DFFs in a CLB, except the clock and reset function.
- The IR faults in this paper can be categorized into two groups, namely, stuck-at 0/1 and bridging faults. A stuck-at fault appears in a pass transistor of a SM, while a bridging fault occurs on two neighboring wire segments connecting to input terminals of LUTs.

These faults are injected at configuration ports [2]. For example, function faults on DFFs are presented when the input ports of the DFFs are connected to GND or VDD. In the second example, a stuck-at 0/1 fault for a PIP is observed when the controlled terminal of the pass transistor is connected to GND or VDD.

2.2.2. Algorithm for CLB test and diagnosis

A CLB in an FPGA is composed of LUTs, DFFs as well as multiplexers. We will concentrate on the test of LUTs and DFFs in the paper to facilitate implementation of the bitstream readback based algorithm. In addition, the LUTs and DFFs have to be tested separately to eliminate fault masking. This is owing to the fact that the DFFs are not only subject to testing, but utilized to collect output responses as well.

2.2.2.1. DFF test and diagnosis. It is not necessary for the DFFs in CLBs to be cascaded as a chain because the output of each DFF can be gained through readback. As a result, no fault masking is incurred. Furthermore, the algorithm for DFF test and diagnosis is straightforward and configuration number is reduced. A configuration for the DFFs in CLBs in the case of Virtex FPGAs is shown in Fig. 2. A CLB of Virtex FPGA constitute two slices, each of which has two 4-input LUTs and two DFFs. All DFF1 and DFF2 in the slices are connected to the input bus DIN1 and DIN2, respectively. TVs are applied to DIN1 and DIN2 through IOBs and the output bitstream ROi ($1 \leqslant i \leqslant 4 \times m \times n$) are collected through readback, m and n denote row and column number in an FPGA. After parsing

Table 1
Bistream parsing library for Virtex FPGA.

Absolute location of bit			Physical location in Virtex FPGA	
DFFs in CLBs	Bit	7443	Block = CLB_R1C25.S1	Latch = XQ
	Bit	11,475	Block = CLB_R1C25.S1	Latch = YQ
	Bit	36,339	Block = CLB_R1C25.S2	Latch = XQ
	Bit	32,307	Block = CLB_R1C25.S2	Latch = YQ
...	...	...	...	...
LUT configured as RAM mode	Bit	6100	Block = CLB_R1C25.S1	RAM = G:1
	Bit	6101	Block = CLB_R1C25.S1	RAM = F:1
	Bit	27,604	Block = CLB_R1C25.S2	RAM = G:16
	Bit	27,605	Block = CLB_R1C25.S2	RAM = F:16
...	...	...	...	...
PIPs of SM	Bit	64,842	Block = CLB_R16C24	pip E0 == S6
	Bit	70,217	Block = CLB_R16C24	pip E0 == W0
	Bit	68,202	Block = CLB_R16C24	pip E1 == N1
	Bit	68,874	Block = CLB_R16C24	pip E1 == S3
...	...	...	...	...

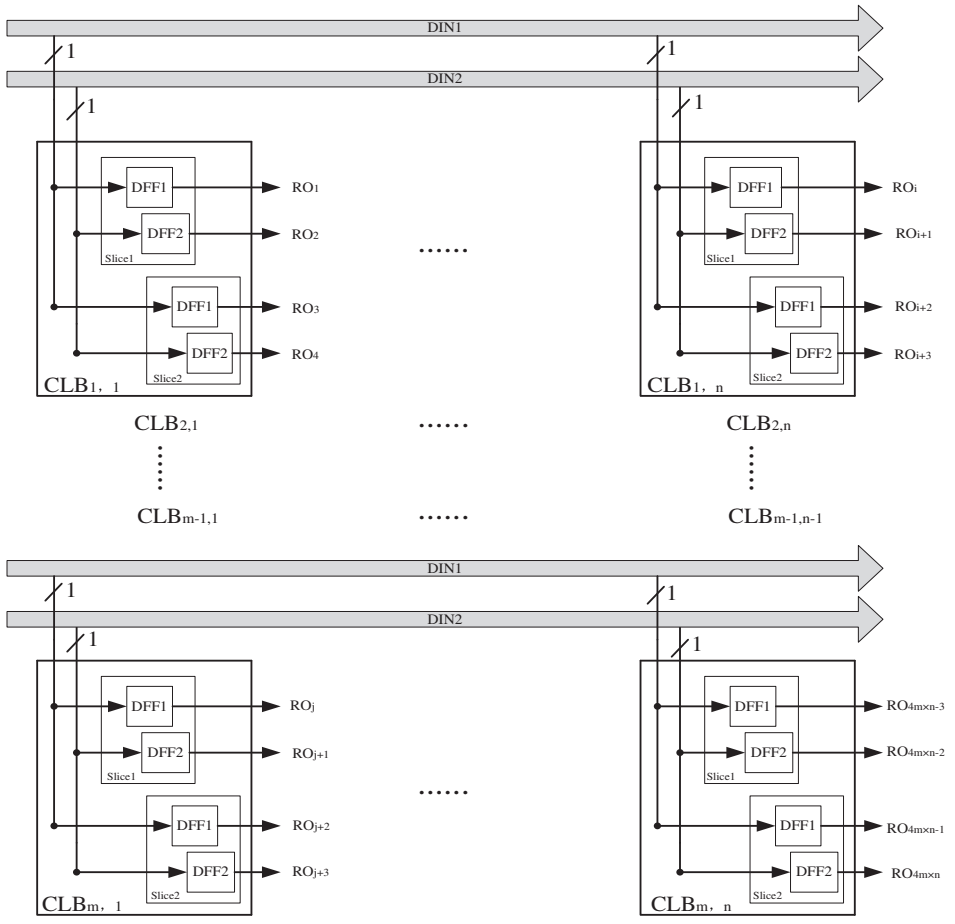

Fig. 2. Configuration of DFFs in CLBs.

the bitstream, actual response data can be gained. Comparing the actual response data against expected values, we can judge whether the DFFs are faulty or not. Pseudo code for DFF test and diagnosis flow is illustrated in Fig. 3. Only one configuration is needed for the DFFs test and diagnosis and diagnostic ambiguity cannot occur in the presence of multiple faulty DFFs.

Other series FPGAs differ in DFF and LUT numbers in a CLB, i.e. two 4-input LUTs and two DFFs in the case of XC4000 FPGAs, The aforementioned configuration and associated algorithm for DFF test and diagnosis hold true for these series FPGAs.

2.2.2.2. LUT test and diagnosis. It is also not necessary for the LUTs in CLBs to be cascaded as a chain. Instead, all 4-input LUT1 and LUT2 in the slices are connected to the input bus DIN1[3:0] and DIN2[3:0] with the outputs of these LUTs linked to DFFs in the slices, respectively. The outputs of the LUTs are inspected by DFFs. TVs are applied to DIN1[3:0] and DIN2[3:0] through IOBs and the response bitstream ROi ($1 \leqslant i \leqslant 4 \times m \times n$) are collected through readback. A configuration for the LUTs in CLBs in the case of Virtex FPGAs is shown in Fig. 4. Note that an LUT has 4-b inputs, thus, 16 addresses can be used to store data in an LUT. When LUTs are

```
1   Apply configuration
2   DIN1 and DIN2 are the input buses
3   Set all DFFs in CLBs
4   After one clock cycle do
5         Readback Capture ouput ROi,  1≤i≤4m×N
6   End after
7   Reset all DFFs in CLBs
8   After one clock cycle do
9         Readback Capture output ROi
10  End after
11  If (The values of the DFFs are not consistent with the expected values) then
12        The DFFs are fault and the sites of the DFFs are determined
13  Else all the DFFs are fault free
14  End if
```

Fig. 3. Pseudo code for DFFs of CLB test and diagnosis.

tested, two configurations, Exclusive OR (XOR) and Exclusive NOR (XNOR) are required for LUT test and diagnosis. Fault masking cannot happen in the presence of multiple faulty LUTs. Comparing the parsed ROi against expected values, we can judge whether the DFFs are faulty or not. Pseudo code for LUT test and diagnosis flow is illustrated in Fig. 5.

2.2.3. Algorithm for IR test and diagnosis

IR architecture is the most complicated part in an FPGA. Thus, IR test and diagnosis confront with a big challenge. Generally, IR architecture is comprised of wire segments and SMs made up by PIPs. For instance, the IRs in a Virtex FPGA are composed of Global lines, Single lines, Hex lines, pin wires as well as PIPs. The difference between Single and Hex lines is that Single lines connect two neighboring SMs while Hex lines span over five SMs.

The algorithm for IR test and diagnosis is based on our previous fault mapping method [15] combined with bitstream readback approach. Take the Hex lines in a Virtex FPGA as example to account for the algorithm. As displayed in Fig. 6, groups of four Hex lines are connected through PIPs in SMs to input terminals of LUTs as addresses. We can get access to the values of the LUTs in terms of the addresses. Then, the output terminals of the LUTs are linked to the DFFs in the same CLBs. TVs are delivered through input buses DIN1[3:0], DIN2[3:0], DIN3[3:0] and DIN4[3:0] and the output responses of the DFFs are collected through readback. Assume the LUTs and DFFs are fault-free, the output bitstream is actually the test response bitstream of IRs. Comparing the parsed bitstream of IRs against expected values, we can determine whether the Hex lines and PIPs are fault or not as long as the faulty Hex lines and PIPs do not occur in the same route. The route in the paper refers to a way from TVs application to collection of response bitstream. Five configurations are needed to cover all Hex–Hex lines and associated PIPs. Hex–Single lines and corresponding PIPs which are often overlooked by some previous researches [9–10] are also covered in the five configurations. Pseudo code for IR test and diagnosis flow is illustrated in Fig. 7. Single-Single lines and corresponding PIPs can be tested by the similar six configurations. Full coverage for the IRs in Virtex FPGAs can be achieved by 11 configurations. Fault masking cannot happen in the presence of multiple faulty wire segments and PIPs provided that the faulty wire segments and PIPs will not present in the same route.

Fault types and their locations can be determined in terms of a fault diagnosis chart. Details of the fault diagnosis chart will be described in Section 3.2.

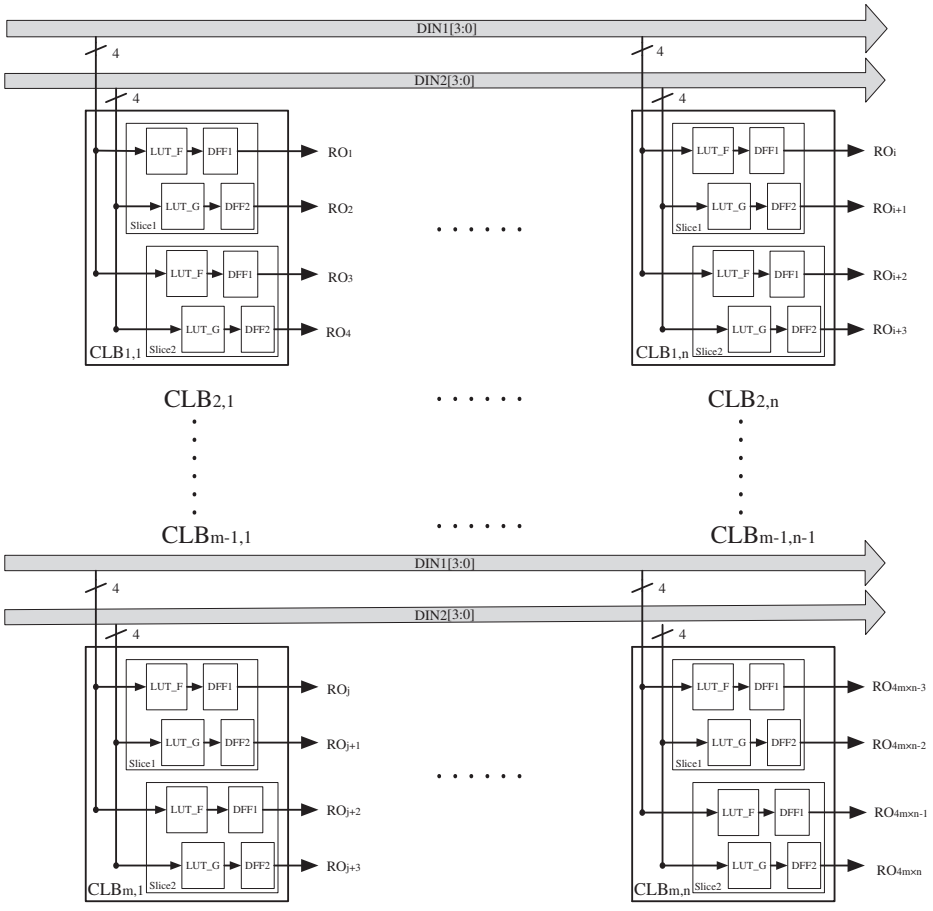

Fig. 4. Configuration of LUTs in CLBs.

1. Apply first configuration
2. m×n CLBs
3. **For** each CLB **do**
4. The output terminals of LUTs are connected to the input terminals of DFFs.
5. **End for**
6. DIN1[3:0] and DIN2[3:0] are the input buses
7. Test vectors ranging from 0000 to 1111 are applied to the inputs of the LUTs
8. **For** each test vector **do**
9. Readback Capture output ROi, $1 \le i \le 4 \times m \times n$
10. **End for**
11. **If** (The values of the DFFs are not consistent with the expected values) **then**
12. The corresponding LUTs are falut
13. The sites of the DFFs are determined leading to the sites of the LUTs determined
14. **Else** all the LUTs are fault free
15. Apply second configuration and repeat step 2 to 14

Fig. 5. Pseudo code for LUT of CLB test and diagnosis flow.

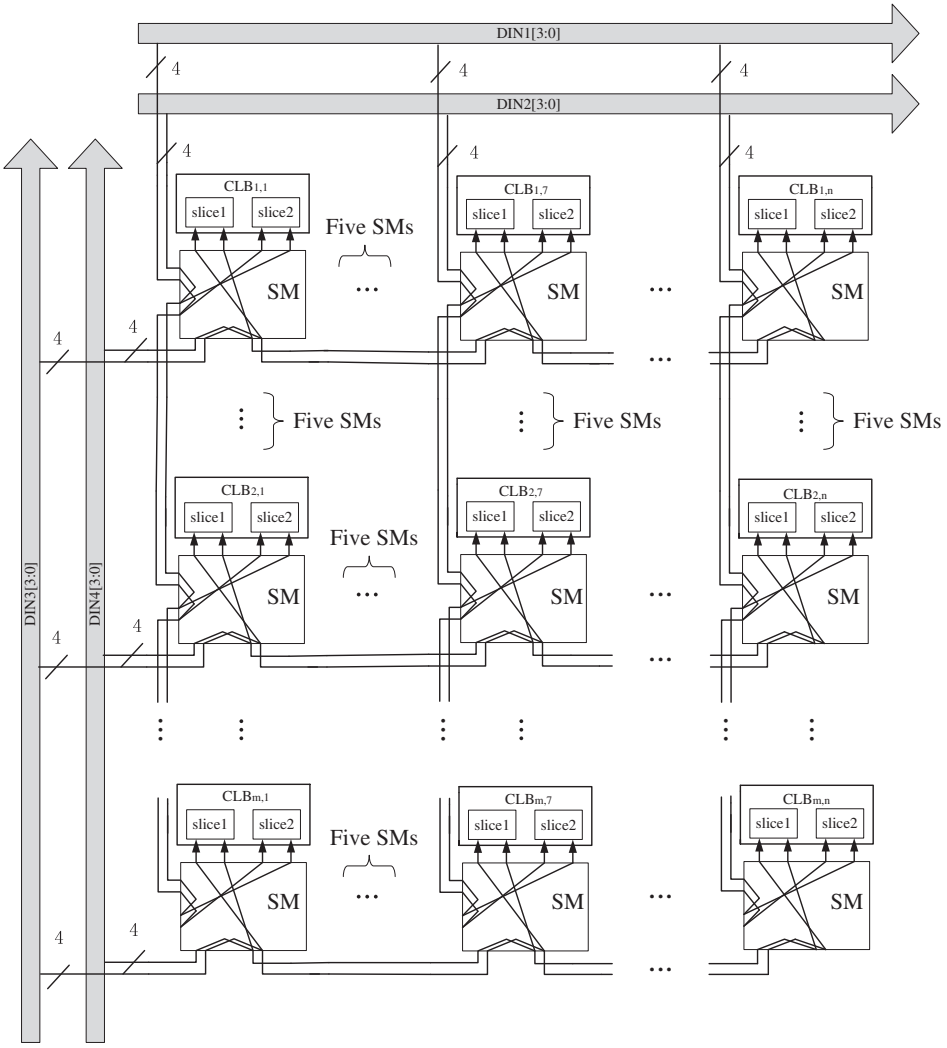

Fig. 6. IR configuration.

3. Experiment and discussion

3.1. Implementation of a test system

A bitstream readback based automatic functional testing system for FPGA test and diagnosis with a Joint Test Action Group (JTAG) as the communication channel has been developed. The proposed FPGA testing system physically consists of software tools in a PC and an FPGA-under-test with the JTAG cable in between. The software part includes a boundary scan controlling module, a bitstream parsing module, a fault test and diagnosis module, a commercial Electronic design automation (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 TVs

1. Apply configuration
2. Initialization LUTs
3. DIN1[3:0], DIN2[3:0], DIN3[3:0] and DIN4[3:0] are the input buses
4. Test vectors ranging from 0000 to 1111 are applied to the inputs of the LUTs
5. **For** each test vector **do**
6. Readback Capture responses
7. **End for**
8. Compare the responses against fault diagnosis chart
9. **If** (the responses are not equal to expected values) **then**
10. Fault type and location are determined
11. **Else** all the IRs are fault-free
12. **End if**
13. Apply next configuration and repeat step 2 to 12

Fig. 7. Pseudo code for IR test and diagnosis flow.

to the FPGA-under-test as well as bitstream readback. A testbench can run in the simulation software tool and if it is necessary, waveforms of response values can be observed. 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. Automation flow of the test steps is shown in Fig. 8.

3.2. Experiment

Xilinx FPGAs including an XC4010EHQ208 in XC4000 FPGAs, an XQVR300CB288 in Virtex FPGAs and an XC5VLX110t in Virtex-5 series FPGAs were tested by the proposed bitstream readback based FPGA test system in the experiment and the experimental results are listed in Tables 2–4. 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 algorithm for fault test and diagnosis as well as algorithm for bitstream parsing in Section 3 and corresponding input TVs. Since IR test and diagnosis

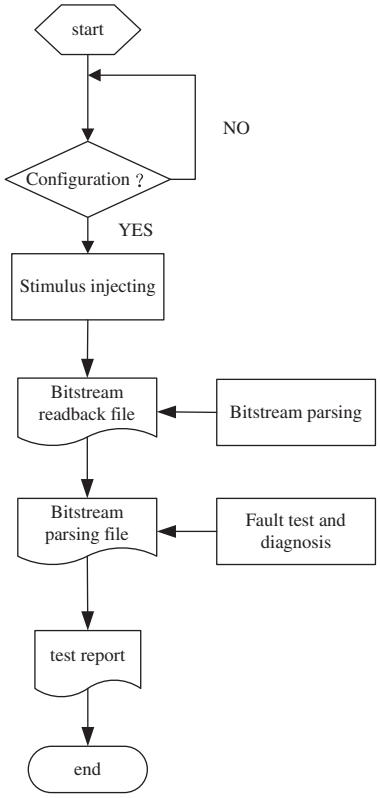

Fig. 8. Automated test flow.

is much more complicated, we will demonstrate how IR diagnosis results can be obtained for Virtex FPGAs as listed in Table 5. The experiments were performed on a 2.8 GHz Intel Pentium 4 microprocessor with 2 GB RAM.

Assume that the stuck-at and bridging faults are presented in the Virtex FPGA, 33 cases are listed below.

- Stuck-at-1 fault is defined as any one or several inputs of the four-input LUT being stuck-at-1. The cases of stuck-at-1 are $2^4 - 1 = 15$.
- Stuck-at-0 fault is defined as any one or several inputs of the four-input LUT being stuck-at-0. The cases of stuck-at-0 are $2^4 - 1 = 15$.
- Bridging fault is defined as two neighboring interconnect metal lines are crossed. There are totally 3 cases.

When initial values "0110110111111110" were applied to all LUTs followed by applications of 16 group TVs from "0000" to "1111", the response bitstream of the LUTs can be gotten by readback. The parsed data for each LUT are 16-b "0110110111111110" if the LUT is fault-free. Note that each parsed data correspond to a physical location of an LUT, which relates to a PIP in an SM linked to the LUT and a line connected to the LUT. Therefore, faulty PIPs or faulty lines can be determined in terms of the fault diagnosis chart listed in Table 5. For example, if the parsed data of an LUT is "0011110011111111" in Table 5, a stuck-at-1 fault at line wire5 in the LUT_G1 of slice2 of a $CLB_{1,1}$ located at row 1 and column 1 can be detected. Furthermore, Table 5 clearly indicates that the fault signals "1100111111111100" and "0000111111111110" correspond to stuck-at-0 fault and bridging fault, each appearing on wire1-LUT_F1, wire1-LUT_F1, and wire2-LUT_F2, respectively. F1, F2 denote wire segments connect to the input terminals of an LUT_F in slice1 of a $CLB_{1,1}$.

IRs in an XC4010EHQ208 FPGA and an XC5VLX110t FPGA can be tested and diagnosed in the same way as the XQVR300CB288 FPGA does. Since the XC4010EHQ208 FPGA and the XC5VLX110t FPGA have 4-input and 6-input LUTs, respectively, their faulty cases are 33 and 131, respectively. Similar fault diagnosis charts for the two FPGAs can be derived. Additionally, configuration numbers of an FPGA-under-test have nothing to do with the array size of the FPGA, but rather depending on the maximum input terminal numbers of PIPs as well as types of wire segments. The maximum input terminal numbers for a PIP in the case of XC4000 and Virtex-5 FPGAs are 3 and 37, respectively. Thus, the minimum configuration numbers for the two types of FPGAs are 3 and 37, respectively. In fact, practical configuration numbers will be larger than the numbers considering some reasons. Thus, configuration numbers

Table 2
Experimental results of Xilinx XC4010 FPGA.

Resources of FPGA	CLB		IR
	LUT	DFF	
Configuration numbers	2	1	6
Configuration time	1s × 2	1s × 1	1s × 6
Test application time	10s × 2	10s × 1	10s × 6

Table 3
Experimental results of Xilinx XQVR300 FPGA.

Resources of FPGA	CLB		IR
	LUT	DFF	
Configuration numbers	2	1	12
Configuration time	10s × 2	10s × 1	10s × 12
Test application time	28s × 2	28s × 1	28s × 12

Table 4
Experimental results of Xilinx XC5VLX110t FPGA.

Resources of FPGA	CLB		IR
	LUT	DFF	
Configuration numbers	2	1	56
Configuration time	190s × 2	190s × 1	190s × 56
Test application time	270s × 2	270s × 1	270s × 56

Table 5
Fault diagnosis chart for Virtex FPGAs.

Wire1-LUT_F1 stuck-at-0	CLB_R1C1.S1	1100111111111100
Wire2-LUT_F2 stuck-at-0	CLB_R1C1.S1	1010010111111010
...	...	...
Wire5-LUT_G1 stuck-at-1	CLB_R1C1.S2	0011110011111111
Wire6-LUT_G2 stuck-at-1	CLB_R1C1.S2	0101111111111111
...	...	...
Wire1-LUT_F1, wire2-LUT_F2 bridging	CLB_R1C1.S1	0000111111111110
Wire5-LUT_G1, wire6-LUT_G2 bridging	CLB_R1C1.S2	0110011001101110
...	...	...

for the XC4010EHQ208 FPGA and the XC5VLX110t FPGA are 6 and 56, respectively.

3.3. Discussion

(1) *Fault coverage evaluation:* The fault coverage evaluation for CLBs and IRs has been performed on some Xilinx series FPGAs. 100% fault coverage can be achieved if test configurations are applied which can be obtained from Tables 2–4.

(2) *Generic method:* The proposed CLB test and diagnosis algorithm holds true for the other series FPGAs, for these series FPGAs have equal numbers of DFFs and LUTs. For example, eight 4-input LUTs and eight DFFs in the case of Virtex-2 and Virtex-4 FPGAs, eight 6-input LUTs and eight DFFs in the case of Virtex-5, as well as eight 6-input LUTs and sixteen DFFs in the case of Virtex-6 and Virtex-7 FPGAs. Thus, two configurations are sufficient to test and diagnose some faulty LUTs in these series FPGAs without diagnostic ambiguity. Likewise, the IR test and diagnosis algorithm is also applicable to the other series FPGAs as explained in Section 3.2.

(3) *Configuration and test application time:* The test time in the paper is defined as the time required to carry out the test to completion, which includes the time required to set up the test configurations plus the time required to apply the TVs and the time for bitstream readback. The test times for three FPGAs are listed in Tables 2–4, respectively. The configuration time is relevant to size of configuration bitstream, while the test application time is subject to the length of TVs, size of bitstream readback, readback velocity, as well as readback numbers for one configuration. In the case of our test system, readback velocity and multiple readback for one configuration dominate the test application time. This is actually the drawback of the proposed test system. There is a considerable room for improvement in reduction of test application time including adopting partial bitstream readback and replacing the JTAG cable with Peripheral component interface express (PCIE) communication channel.

(4) *Limitation of IOB numbers:* IOB number has impact on algorithms for FPGA test and diagnosis for IOBs play a role of monitoring internal resources. Due to restriction of IOB number, the algorithms for FPGA test and diagnosis have to face the challenge of covering as more internal resources as possible with as small configuration numbers as possible. Fault masking sometimes can occur. Luckily, bitstream readback technique can eliminate the requirement of output IOBs resulted in reduction of IOB numbers used in configurations as shown in Table 6. Put another way, readback bitstream can be achieved by readback through DFFs and configuration memory cells. Furthermore, since DFF numbers are much larger than IOB numbers, restriction of IOB number is partially relieved leading to relax requirement for the FPGA test and diagnosis algorithms.

(5) *Diagnostic resolution:* Diagnostic resolution in this paper is defined as the smallest circuit that can be diagnosed without fault masking and much occupied IOB numbers. The smallest circuits detected in the previous research [15] are identical to our research in the paper, i.e. DFFs and LUTs in CLBs as well as wire segments and PIPs in IRs, respectively. In the research [15], fault masking may happen due to DFFs and LUTs in CLBs also configured as a chain. In contrast, bitstream readback based IR and CLB test and diagnosis presented in the paper do not introduce fault masking. Albeit diagnostic ambiguity may not occur to IR test and diagnosis in the research [15], much more IOBs are used when compared with the bitstream readback based IR test and diagnosis method presented in the paper. Some BIST approaches may introduce diagnostic ambiguity [9] in spite of the fact that BIST methods can address the issue of IOB limitation well.

(6) *Applications:* The proposed bitstream readback technical can be used to test and diagnose Single Event Upset (SEU) errors in SRAM-based FPGAs for space applications [13]. It can also improve the performance of some previous FPGA test methods. For example, the IR test method presented in [11] can be improved by connecting internal output nodes to DFFs

Table 6
Comparison between used IOB numbers by two methods.

Device name	Type of internal resources	Percentage of used IOBs vs. IOB number in one configuration in paper [12]	Percentage of used IOBs vs. IOB number in one configuration by bitstream readback technique (%)
XC4010EHQ208	CLB	30%	4.38
	IR	30%	6.88
XQVR300CB228	CLB	57.28%	10.76
	IR	70.89%	16.14
XC5VLX110t	CLB	–*	9.41
	IR	–*	9.41

* Data are unavailable.

Table 7
Performance comparisons.

	Diagnostic ambiguity	Full fault coverage	Relevance to FPGA array size	Configuration numbers	Relieved I/O number requirement
[9]	Yes	No	Yes	8 (IRs, Virtex)	Yes
[12]	No	Yes	Yes	120 (CLBs & IRs, XC4000)	Yes
[10]	No	No	Yes	51 (IRs, Virtex-4 LX series)	Yes
[15]	Yes	Yes	No	21 or 23 (IOBs & CLBs & IRs, XCV300)	No
This work	No	Yes	No	Tables 1–3	Yes

to get the output responses by the proposed bitstream readback technique, rather than cascade the LUTs to propagate the value and locate the faulty by back-tracking. Thus, fault ambiguity may not be induced in the presence of multiple faults.

(7) *Comparison:* Comparisons between the proposed bitstream readback based FPGA test system and some previous works are listed in Table 7. None of the previous works can simultaneously provide good diagnostic resolution, higher fault coverage, relieved I/O number requirement, as well as derive configuration numbers independent on array size of an FPGA-under-test. Our proposed system along with corresponding algorithms can offer these capabilities.

4. Conclusion

This paper has proposed a bitstream readback-based test and diagnosis method for Xilinx FPGAs. The presented method tries to address the main issues in the field of FPGA test and diagnosis. Bitstream readback-based algorithm is generic, reusable and suitable for test of both IRs and CLBs with higher fault coverage, reduced IOB as well as configuration numbers. The bitstream parsing algorithm results in good diagnostic resolution and makes bitstream readback-based algorithm possible. The proposed method is cost effective and have engineering applications such as SEU detections in space. Improving readback velocity can reduce test time and will be our future work.

Appendix A

BIST: Built-In-self-Test
CLB: configurable logic block
CUT: circuit under test
DFF: D Flip-Flop
EDA: Electronic design automation
IOB: Input/Output block
IR: Interconnect resource
JTAG: Joint Test Action Group
LUT: Look-Up table
ORA: output response analyzer
PIP: programmable-interconnect-point
PCIE: Peripheral component interface express
RAM: Random access memory
RBB: repeatable building block
SM: switch matrix
SRAM: Static random access memory
SEU: Single Event Upset
TPG: test pattern generator
TV: test vector
XDL: Xilinx Design Language

References

[1] Campregher N, Cheung YK, 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] Toutoundi S, Lai A. FPGA test and coverage. In: Proceedings of international test conference. 2002. p. 599–607.
[3] Kumar TN, Chong CW. An automated approach for locating multiple faulty LUTs in an FPGA. Microelectron Reliab 2008;48(11–12):1900–6.
[4] Abramovici M, Stroud C. BIST-based test and diagnosis of FPGA logic blocks. In: IEEE transactions on very large scale integration (VLSI) systems vol. 9(1). 2001. p. 159–172.
[5] 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.
[6] Abramovici M, Stroud C, Emmert J. Online BIST and BIST-based diagnosis of FPGA logic blocks. In: IEEE transactions on very large scale integration (VLSI) systems. vol. 12(12). 2004. p. 1284–1294.
[7] 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. IEEE Trans Comput Aided Des Integr Circuits Syst 2008;27(2):309–26.
[8] Sun X, Trouborst P. A unified global and local interconnect test scheme for Xilinx XC4000 FPGAs, In. IEEE transactions on instrumentation and measurement. vol. 53(2). 2004. p.368–377.
[9] Tahoori MB, Mitra S. Application-independent testing of FPGA interconnects. IEEE Trans Comput Aided Des Integr Circuits Syst 2005;24(11):1774–83.
[10] Yao J, Dixon B, Stroud C, Nelson V. System-level built-in-self-test of global routing resources in Virtex-4 FPGAs. In: Proceedings of 41st southeastern symposium on system and theory. 2009. p. 29–33.
[11] Kumar TN, Haider AF, Almurib NC. Fine grain faults diagnosis of FPGA interconnect. Microprocess Microsyst 2013;37(1):33–40.
[12] Hsu CL, Chen TH. Built-in-self-test design for fault detection and fault diagnosis in SRAM-based FPGA. IEEE Trans Instrum Meas 2009;58(7):2300–15.
[13] 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.
[14] 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.
[15] Ruan AW, Kang S, Wang Y, Han X, Zhu ZJ, Liao YB, et al. A Built-In-self-Test (BIST) system with non-intrusive TPG and ORA for FPGA test and diagnosis. Microelectron Reliab 2013;53(3):488–98.
[16] Virtex FPGA series configuration and readback. Xilinx Inc., XAPP138 (v2.8). 2005.
[17] Note JB, Rannaud E. From the bitstream to the netlist. In: 16th International ACM/SIGDA symposium on field programmable gate arrays. 2008. p. 264–271.