International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 03 Issue 14
October 2016

A Novel Fault Detection Method in Fpgas

B.Harikrishna[1]
[1]Professor,Department of Electronics and communication Engineering,CMR Engineering College,
harikrishna07@gmail.com

C.Ashok kumar[2]
[2]Professor, Department of Electronics & communication Engineering, CMR Engineering College,
cheelikumar@gmail.com

***Abstract*— Field-programmable gate-array (FPGA) devices are becoming the most suitable platform for implementing modern electronic systems due to their high level of reconfigurability, low cost and wide availability. FPGA devices are programmable logic circuits that can be programmed or reprogrammed with almost any circuit or system. They are now- a- days being used for coprocessors in high-performance systems to speed-up difficult tasks, for various embedded systems, systems on a chip, networks on a chip, or as a platform to design-circuit prototypes, etc. The FPGA devices are also increasingly being used in critical systems like space avionics, exploration missions, security systems, banking systems, secure servers, etc. Faults are very common in FPGA devices. A fault defined as a substantial occurrence within an FPGA that causes to malfunction, for example, a broken wire caused during manufacture by a dust particle. Faults can occur at the beginning and end of a chip's life cycle. This paper presents a novel BIST based fault detection method for detecting faults in FPGA devices.**

***Keywords*— BIST, Fault, FPGA, Granularity, Redundancy.**

I. INTRODUCTION

A Field Programmable Gate Array (FPGA) is intended for customer or designer configurable after manufacturing, so called as "field-programmable". The hardware description language is used to design the system under consideration and compiled using the software provided by the FPGA vendor. This software converts the design file written using HDL is converted to FPGA compatible bit streams and downloaded from the personal computer using a cable. FPGAs can be used to implement any logical function that an Application Specific Integrated Circuit could perform [1]. The ability to change the functionality after shipping, partial re-configuration of the design and the low non-recurring engineering costs compared to an ASIC design are the advantages of a FPGA- based system for many applications.

FPGAs, which contain an array of logic blocks whose functionality, can be determined with the help of multiple programmable configuration bits. Logic blocks configured to realize a specific function, are connected using the set of programmable interconnections. FPGA- based systems has several advantages compared to conventional systems. The major advantage with the FPGA- based is the ease of prototyping. A designer can check the functionality of a system under consideration by simply downloading the bit stream into the FPGA. FPGAs are widely used in various domains of human life like medical electronics, avionics, communication, signal processing etc. Based on the programming technique used, FPGAs can be classified as- SRAM based, anti fuse based, EPROM based and EEPROM based. Once programmed, contents of anti fuse based FPGAs are made permanent and so called as "One Time Programmable (OTP)", while others are reprogrammable.

The general architecture of a FPGA is shown in Figure 1.

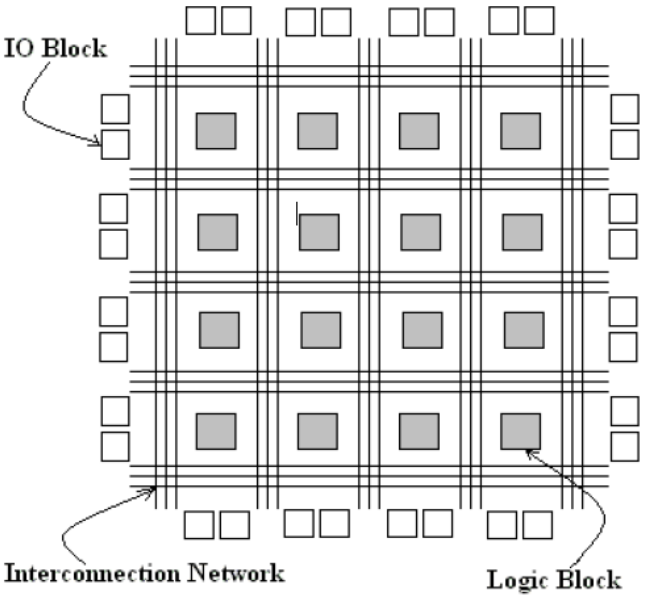

Figure 1: General FPGA Architecture

The architecture of FPGA can be explained as arrays of logic block, which can be interconnected using a programmable interconnect network along with the input output block (IO Blocks). The logic block in an FPGA can be as simple as a transistor or as complex as a micro processor, which is capable of implementing various combinational and the sequential logic functions [2][3]. The logic block in a

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 03 Issue 14
October 2016

commercial FPGA is basically multiplexer, Look-up-table (LUT) or AND-OR array. The periphery of the FPGA consists of I/O blocks, which process the signal to and from the FPGA. The routing network in FPGA consists of wire segments of different lengths, which are interconnected using the programmable switches. Wires for interconnection are laid in the wiring channels or routing channels that run horizontally and vertically through the chip.

II. FAULTS IN FPGAS

A fault in FPGA is a physical occurrence that causes the malfunctioning of operation. For example, a broken wire caused during manufacture by a dust particle. Faults may occur at any stage of chips life cycle. Fabrication faults are detected during manufacture testing and are usually caused by contaminants or other flaws. Failure of device resources generally occurs at late life of circuits. On the other hand a fault occurs at any stage during the life cycle of FPGA. Faults may also be transient or impermanent. Transient faults are very hard to identify because they do not constantly noticeable themselves. As seen in the Figure 2, the failure rate is typically constant after the chip has left the factory and before the end of its life, and is due to environmental stresses. If FPGA is used in harsh environment the time axis may be compressed.

.

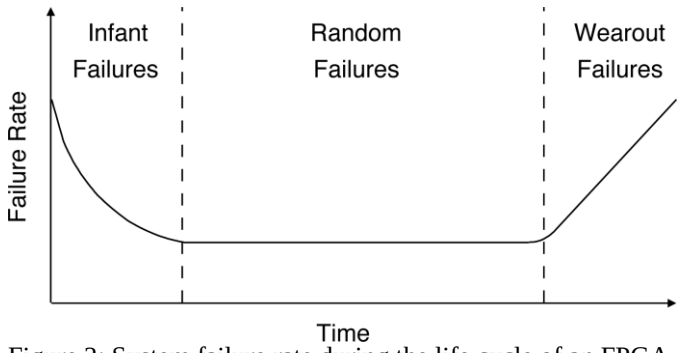

Figure 2: System failure rate during the life cycle of an FPGA

III. EXISTING METHODS

Fault detection methods can be divided into three broad types:

- Redundant/concurrent error detection
- Off-line test methods
- Roving test methods

A. Redundant/ Concurrent error detection

When a function is not generating the correct output Redundant or concurrent error detection methods uses further logic as a means of detecting errors. Redundancy is commonly used method for fault detection in FPGAs, predominantly in the form of Triple Modular Redundancy (TMR) [4] [5] [6]. Redundancy provides fast means of detecting faults, as a fault is uncovered as soon as a discrepancy occurs. In addition, this type of error detection has a little impact on timing performance, just the latency of voting or parity logic, or similar. The main drawback of fault detection using redundancy is the area overhead needed to replicate functionality, which can be over three times in the case of TMR[7].

.

B. Offline fault detection

Any testing which is carried when the FPGA is not performing its operation function is Off-fault detection method. In this method fault schemes have one or more test configurations which are loaded individually in to the operating configuration. Within the test section a test pattern generator is configured, an output response analyzer and, logic and interconnect to be tested are arranged in paths-under-test. The main advantage of these methods is that it has no impact on the FPGA during normal operation. The only overhead is the requirement to store the test configurations which are typically small.

C. Roving test methods

Roving test methods performs a scan on to the FPGA structure by swapping the blocks of functionality with a block carrying a test function. Roving detection exploit run-time reconfiguration to take out the BIST techniques on-line, in the field, with a least of area overhead. In roving detection method, the FPGA is divided into equal-sized regions. One of these is configured to perform the self-test, while the remaining areas carry out the design function of the FPGA. Over time, the test region is swapped with the functional regions one at a time so that the entire array can be tested while the FPGA remains functional. Roving test has a lower area overhead than the redundancy methods; the overhead comprising of one self-test region and a controller to manage the reconfiguration process. The method gives excellent fault coverage and granularity, comparable to the BIST methods.

IV. PROPOSED METHOD

In this proposed method FPGA BIST structure is to configure groups of ten CLBs into a test block. In every test block, four CLBs are configured as a test pattern generator (TPG) to generate the addresses for check patterns. In addition,

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 03 Issue 14
October 2016

two CLBs are configured as output response analyser (ORA) for evaluating with each output of the block under test (BUT) to observe the check results. The test methods of the proposed FPGA BIST structure are concurrently performed by the BIST controller, which constantly reconfigures the test blocks for testing. Figure 3 shows the proposed BIST method.

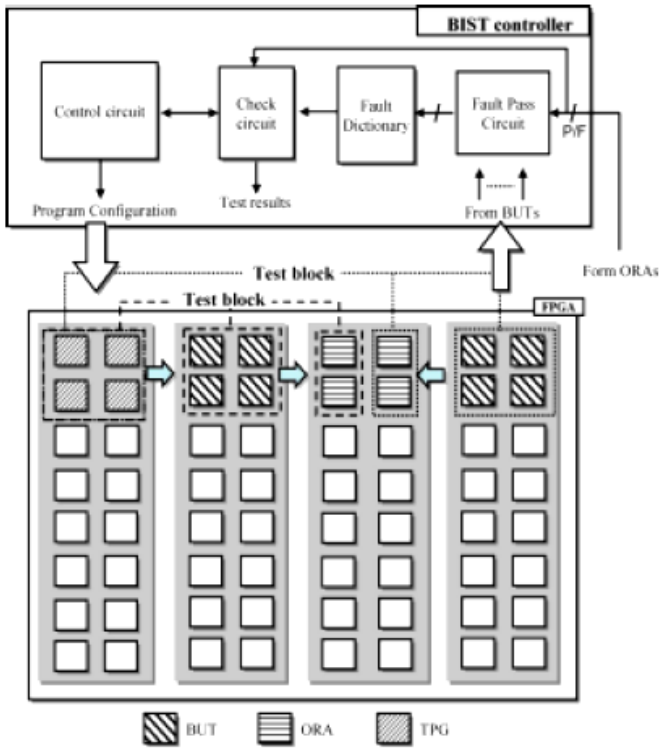

Figure 3: Proposed BIST method

The proposed method is to design a BIST structure for both CLB and interconnect resource testing in SRAM-based FPGAs. The faults in this method can be categorized into four major groups:

- Open/ short
- Stuck on/ off
- Stuck at 0/ 1
- Interconnect delay faults

The stuck-on/off faults appear in the pass transistor of PIP-PSs or MUX-PSs in the local interconnects, while open/short faults occur on PCP-PSs or wire segments in global interconnects. Significantly, the delay fault is also presented with a path under test (PUT). On the other hand, the stuck-at-0/1 faults can be found in the LUTs of the CLBs. PIP-PSs and MUX-PSs with the stuck on/ off faults in the local interconnects. Note that the stuck-on/off fault causes the pass transistor in PIP-PSs or MUX-PSs to be permanently on/off, regardless of the value of the SRAM cell controlling the pass transistor in PIP-PSs or MUX-PSs. An open fault in the global interconnect is a disconnection of any wires, while the short fault indicates a bridging between two wires. PCP-PS open/short faults that occur when there is a stuck-off fault and a stuck-on fault in the PIP-PS of the connectable and non-connectable directions of wire segments, respectively. Moreover, since the TCs for interconnect resource and CLB testing are the same and the TPG and ORA of the proposed BIST method are built by using the LUTs, the faults in a CLB only.

To effectively achieve the fault detection, the BIST controller is required to control the test and diagnosis processes in FPGAs. Additionally, the roving self-testing area (STAR) approach is needed to test the CLBs in the FPGA. A STAR is a temporarily offline section of the FPGA where the self- testing occurs without disturbing the normal system activity in the remaining portions of the FPGA. Figure 4 illustrates a self-tester structure for testing the CLBs in the STAR.

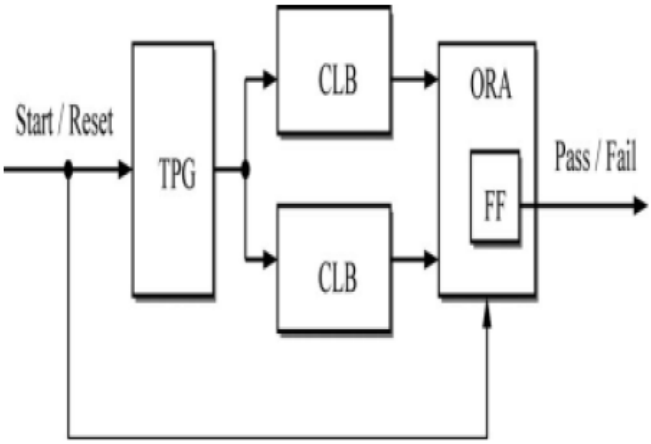

Figure 4: Self Tester Structure

A STAR consists of several self testers working concurrently. Specifically, the self tester contains a TPG that applies pseudo exhaustive test patterns to two identical CLBs. The outputs of the test CLBs are compared by an ORA, which the latches and reports mismatches as test failures. The signal start/reset is provided to initiate the self tester sequence and to reset the TPG and ORA functions. Based on the self test processes, the CLBs in FPGA can be checked offline to confirm whether the elements in the CLBs are fault free or not.

After the self tester processes according to the proposed FPGA BIST structure. A BIST scheme for a test block to detect and diagnose the faults on both interconnect resources and LUTs in the CLBs online. The testing process is performed by configuring the TPG, ORA, and BUT in each of the test block. Additionally, the TPG and ORA of the the proposed FPGA BIST structure are designed using the existing CLBs to reduce the extra area and test cost needed. The TPG is an address generator, which consists of the eight MUXs, eight LUTs, and eight DFFs to continuously generate the addresses (0–15) in the LUTs to produce the corresponding test patterns. Significantly, to ensure the accuracy of the LUTs in the TPG, an algorithm is needed to write different test patterns into LUTs, read the contents of LUTs, and check whether the read data are correct or not.

V. Conclusion

The first step in fault tolerant is detection of faults. Fault detection has two purposes- initially, it alerts the supervising process that action needs to be taken for the system to remain operational and, secondly, it identifies which components of the device are defective so that a solution can be determined. Fault detection methods can be divided into three broad types- Redundant or concurrent error detection, Off- fault detection method, Roving test method. Redundant or concurrent error detection uses additional logic as a means of detecting when a logic function is not generating the correct output. Off- fault detection methods cover any testing which is carried out when the FPGA is not performing its operational function. Roving test methods perform a progressive scan of the FPGA structure by swapping the blocks of functionality with a block carrying out a test function. In our proposed method, the Output response analyzer(ORA) based fault detection architecture of the Interconnect and Logic block for designing high performance fault detection structure of the FPGA is presented .The proposed ORA- based fault detection approach detects faults in both interconnect and logic block . This ORA-based fault detection scheme reaches the maximum fault coverage when compared to the other methods of fault detection.

References

[1] S. Brown, and J. Rose, Architecture of FPGAs and CPLDs-A tutorial, IEEE design and test journal, vol. 13, issue 2, pages 42-57, 1996.

[2] Jonathan Rose, "Architecture of Field Programmable Gate Arrays", Proceedings of the IEEE , Page(s): 1013 – 1029, 1993.

[3] I. Kuon, R. Tessier and J. Rose, "FPGA Architecture: Survey and Challenges", Foundations and Trends in Electronics Design Automation, Vol. 2, pages 135-253, 2007

[4] S. D'Angelo et al, "Fault-tolerant voting mechanism and recovery scheme for TMR FPGA-based systems", Int. Symposium on Defect and Fault Tolerance in VLSI Systems, p 233-40, 1998.

[5] S. D'Angelo et al, "Transient and permanent fault diagnosis for FPGA-based TMR systems", IEEE Int. Symposium on Defect and Fault Tolerance in VLSI Systems, p 330-8, 1999.

[6] G.A. Mojoli et al, "KITE: A behavioural approach to fault-tolerance in FPGA-based systems", International Workshop on Defect and Fault Tolerance in VLSI Systems, p 327-334, 1996.

[7] M. Berg, "Fault tolerance implementation within SRAM based FPGA designs based upon the increased level of single event upset susceptibility", Int. On-Line Testing Symposium, 2006.