

A Dual Configuration BIST-Based Modular Diagnostic Methodology for Embedded Cores in FPGAs

Alireza Sarvi
Al.Sarvi@xilinx.com
Xilinx, Inc.
2100 Logic Dr.
San Jose, CA 95124

Jenny Fan
Jenny.Fan@xilinx.com
Xilinx, Inc.
2100 Logic Dr.
San Jose, CA 95124

Reto Stamm
Reto.Stamm@xilinx.com
Xilinx, Inc.
2100 Logic Dr.
San Jose, CA 95124

***Abstract** – The number of embedded cores in an FPGA has been increasing and different devices use different numbers of different types of hard IP cores. To facilitate failure analysis and reduce its turnaround time, we present an automated BIST-based methodology that exploits the existing redundant resources of an FPGA and its reconfigurability to efficiently locate the faulty IP block(s) in addition to pass/fail test. It doesn't impose any area overhead, cost, or performance degradation and it is applicable to different types of cores and scalable for different devices.*

Keywords: BIST, FPGA Diagnostic, Embedded IP Cores

1. Introduction

FPGAs were initially developed to replace glue logic. Process technology of 90-nm and smaller geometries has allowed us to place more transistors on a chip and integrate greater numbers of different types of hard IP blocks into different devices of FPGA, known as Application Specific Modular Block (ASMBL) architecture. For example, XC4VSX55 device in Xilinx Virtex4 family, SX platform, includes 320 embedded Block RAMs and 512 DSP blocks [1]. With the help of the dedicated embedded cores, designers can now create an entire system on a single programmable device, which is called System On Programmable Chip (SOPC). Unlike soft cores that are basically source level libraries of high level HDL, these hard cores have already been designed, placed and routed, characterized and verified to optimize performance and save Configurable Logic Blocks (CLB) as general resources, while reducing silicon real state and power consumption. In order to reduce failure analysis turnaround time, meet time-to-market goals, achieve higher yields, and eventually ensure product profitability, it is important to be able to locate defected blocks quickly and efficiently. The following section presents an automated methodology to realize this goal.

2. Methodology

BIST (Built-In Self-Test) for FPGAs was first introduced for testing CLBs and then extended to testing the

programmable interconnect matrix [2]. We implemented BIST to debug embedded hard cores and locate the defective ones in addition to pass/fail test. By configuring the device twice, we can obtain total numbers and locations of faulty blocks, and determine their failure modes. The BIST architecture (Figure 1) employs three components: a pattern generator, an error detector unit, and a test controller. The controller activates the self-test by asserting the START signal, terminating the test and transmitting the result by asserting DONE signal.

We assume that CLBs and routing resources have been already tested and are functional. Today's FPGAs are very resourceful in terms of configurable logic cells and depends on the target cores under test, we can implement the most suitable test pattern generator circuitry to produce stimulus [3]. In ASIC conventional BIST techniques, there are typically between 10 to 30 percent area overhead and delay penalties. The advantage of utilizing the programmability of FPGA to test itself is that BIST logic "disappears" after testing and so area or overhead isn't the main criteria in selecting the best method. For example, Block RAM can be initialized to store predefined vectors for the test. In another scenario, combination of Linear Feedback Shift Register (LFSR) and Finite State Machine (FSM) can be used to derive pseudo-random and deterministic vectors for input and control signals respectively. It is also possible to use processors such as soft MicroBlaze processor or embedded IBM Power PC to accelerate unit testing. Automated C-to-RTL compilation generates a hardware representation of the software vector producer module. These functions interact with the unit under test using FIFO or other interfaces to implement and supply the stimulus [4].

Existing redundant embedded cores enables us to implement the comparison technique to detect an error. In our scheme, we group the blocks in sets of pairs. All pairs receive identical inputs and the outputs of two members of each pair are compared with each other. In the case of a mismatch, the write enable (WE) signal goes to High and the corresponding flip flop will be set to High. When an error is identified, the status of the control signals can also

be registered to indicate the function or operation that was active when the failure occurred. It is unlikely that both members of a pair have an identical defect. However, to minimize the probability of error masking, in addition to comparison, other technique such as signature analysis with Multiple Input Signature Register (MISR) can be used [3]. The error detector circuitry is systematically replicated to generate an appropriate number of module instantiations to cover all blocks in the target device (Figure 1). At the end of the test, the DONE signal is asserted by the controller to activate the scan chain and serialize the results. It remains High for certain period of time depends on the number of blocks until all the test results are shifted out to an output IO. The reconfigurability of the FPGA allows us to map the blocks in different arrangements. Post processing the result of each configuration generates a list of pairs where, in each pair at least one block is faulty. In the second round, we map the blocks in different locations than in the first step. This configuration is tested and the result of post processing shows different pairs. The common blocks that appear in both results indicate the location of the actual faulty blocks.

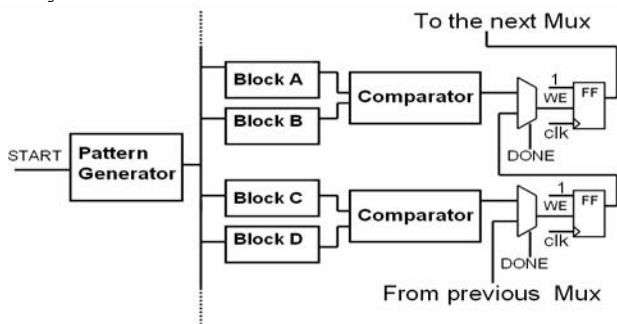


Figure 1. BIST Structure

Different algorithms for mapping are possible. Following is a general approach:

1. List all blocks in a target device in a set called L.
2. Count the number of blocks and label them from 0 through n-1.
3. Create two partitions (P_0 and P_1) of the set of blocks that have the following properties:
 - A. All members of the partitions are pairs of blocks. Note: This condition can only be satisfied if the number of blocks is even.
 - B. P_0 and P_1 are disjointed. In other words, there is no pair of blocks common in P_0 and P_1 .
4. Construct the partitions. Define partition P_0 as the set of pairs (i, j) from L, where i is even and $j = (i + 1) \bmod n$. Define partition P_1 as the set of pairs (q, r) from L, where q is odd and $r = (q + 1) \bmod n$.

One realistic algorithm forms a meandering flow. Map instantiations of the blocks in vertical order from column to column the first time, and map them in horizontal order from row to row the second time. This concept is illustrated in Figure 2 in a small scale where, continuous

line shows vertical order and dashed line indicates horizontal arrangement. For example, if the core located in column 3 (C3) and row 3 (R3) is faulty, the register corresponding to the pair of C3R3 and C3R4 will be High indicating that one of them is defective. In the second run, the corresponding register of pairs C3R3 and C4R3 will be High. Using this approach, the faulty block is isolated to C3R3. For any given device, the process of proper RTL code generation, constraint assignment, synthesis, map, place and route, as well as post processing of the results, is performed in an automated flow. After defective blocks have been located, comprehensive defect screening for failure analysis is applied. The detected blocks are examined extensively cycle by cycle against expected values to have a better understanding of the failure mechanism and its root cause.

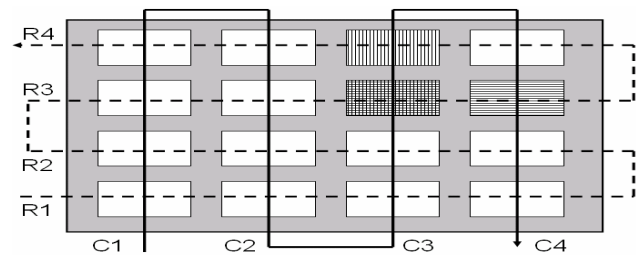


Figure 2. Two Configurations Generated to Isolate the Faulty Block

3. Conclusion

We presented a two phase diagnostic approach for embedded IP blocks in an FPGA. It achieves rapid localization of a faulty block or multiple blocks by generating and applying two different configurations to the device under test. It has shown its efficiency and applicability to expedite and facilitate failure analysis turnaround time by localizing the faulty blocks in an automated process. It is applicable to different types of cores (DSP, Multiplier, Block RAM, etc.) and scalable for different devices. A variation of this method can also be used to perform the test remotely over the network wherever it is possible and applying an alternative configuration that excludes the detected defective block(s).

4. References

[1] Xilinx, Inc., *Virtex4 FPGA Handbook, One Family, Multiple Platforms, Endless Possibilities*, August 2004.
[2] A. Doumar and H. Ito, "Detecting, Diagnosing, and Tolerating Faults in SRAM-Based Field Programmable Gate Arrays: A Survey", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol.11, No.3, pp. 386-405, June 2003.
[3] C. Stroud, *A Designer's Guide to Built-In Self-Test*, Kluwer Academic Publishers, 2002.
[4] D. Pellerin and M. Saini, "MicroBlaze and PowerPC Cores as Hardware Test Generators", *Xilinx Xcell journal*, Issue 51, 2004.