Restartable BIST controller for fault detection in CLB of FPGA

M.S.PRASANTHI[#1], M.RANGASWAMY[#2] M.E.DINAKAR[#3]
#1 Intellectual Institute of Technology,Anantapur
Ph:9440709744.
#2 Sri Venkateswara Institute of Technology,Anantapur
Ph:9966223201.
#3 Intellectual Institute of Technology,Anantapur
Ph:9985357419.

ABSTRACT

Today Field Programmable Gate Arrays (FPGAs) are widely used in many applications. Complicated integrated circuit chips like FPGAs are prone to different types of Faults due to environmental conditions or aging of the device. The rate of occurrence of permanent faults increases with emerging technologies because of increased density and reduced feature size, and hence there is a need for periodic testing of such FPGAs. Efficient testing schemes that guarantee very high fault coverage while minimizing test costs and chip area overhead have become essential.

The Configurable Logic Blocks (CLBs) are the main logic resources for implementing sequential as well as combinatorial circuits in FPGA. Built-In Self-Test (BIST) is a design technique that allows a circuit to test itself. Here, We are implementing a restartable logic BIST controller for the configurable logic blocks by using the resources of FPGA itself . The design exploits the reprogramability ofan FPGA to create the BIST logic by configuring it only during off-line testing. The technique achieves the testability without any extra burden as the BIST logic disappears when the circuit is reconfigured for its normal operation. The proposed technique implemented through VHDL, after verifying the simulation results the code will be synthesized on Xilinx FPGA.

Modelsim Xilinx Edition (MXE) and Xilinx ISE will be used for simulation and synthesis respectively. Xilinx FPGA board will be used for testing and demonstration of the implemented system. The Xilinx Chipscope tool will be used to test the FPGA inside results while the logic running on FPGA

Key words: Configurable Logic Blocks, Built-In Self-Test, antigenicity, increased density and reduced feature size, logic.

Corresponding Author: M.S.Prasanthi

INTRODUCTION

As integrated circuits are produced with greater and greater levels of circuit density, efficient testing schemes that guarantee very high fault coverage while minimizing test costs and chip area overhead have become essential. As the complexity of circuits continues to increase, high fault coverage of several types of fault models becomes more difficult to achieve with traditional testing paradigms.Integrated circuits are presently tested using a number of structured design for testability (DFT) techniques. These techniques rest on the general concept of making all or some state variables directly controllable and observable .

BIST Technique :

Built-In Self Test is a technique of integrating the functionality of an automatic test system onto a chip. It is a Design for Test technique in which testing (test generation and test application) is accomplished through built in hardware features. The general BIST architecture has a BIST test
controller which controls the BIST circuit, test generator which generates the test address sequence, response verification as a comparator which compares the memory output response with the expected correct data and a circuit under test (CUT). We have used LFSR and signature analyzer for testing a CLB.

A Field-Programmable Gate Array (FPGA) is a logic device that can be programmed to implement a variety of digital circuits. FPGAs are widely used both in product prototyping and development because of their ability for configuration and re-configuration. Some of the advantages are reduced design time and implementation cycles, the low non-recurring engineering cost. FPGA consists of an array of configurable logic blocks inter connected by programmable routing resources, and programmable 110 cells. The set of all programming bits establishes a configuration which determines the function of the device. In contrast, conventional BIST approaches introduce both area overhead (typically between 10 and 30 percent) and delay penalties. Our approach is applicable to any in-circuit reprogrammable FPGA, such as SRAM-based FPGAs. However, with the increase in density, capability and speed, FPGAs have become more vulnerable to faults, as it is the case for all circuits. A percentage of manufactured FPGA chips are determined to be faulty after initial application-independent tests. Faulty FPGAs can also be found after delivery to users, during the system development or operation. They may be still usable for some particular application if only a portion of the circuitry is defective.

FPGA family architecture consists of five fundamental programmable functional elements:

• Configurable Logic Blocks (CLBs) contain flexible Look-Up Tables (LUTs) that implement logic plus storage elements used as flip-flops or latches. CLBs perform a wide variety of logical functions as well as store data.

• Input/Output Blocks (IOBs) control the flow of data between the I/O pins and the internal logic of the device. Each IOB supports bidirectional data flow plus 3-state operation. Supports a variety of signal standards, including four high-performance differential standards. Double Data-Rate (DDR) registers are included.

• Block RAM provides data storage in the form of 18-Kbit dual-port blocks.

• Multiplier Blocks accept two 18-bit binary numbers as inputs and calculate the product.

• Digital Clock Manager (DCM) Blocks provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase-shifting clock signals.

We concentrate on Configurable logic blocks since we are going to implement bist controller logic for CLB of fpga.

CLB Overview:

The Configurable Logic Blocks (CLBs) constitute the main logic resource for implementing synchronous as well as combinatorial circuits. Each CLB contains four slices, and each slice contains two Look-Up Tables (LUTs) to implement logic and two dedicated storage elements that can be used as flip-flops or latches. The LUTs can be used as a 16x1 memory (RAM16) or as a 16-bit shift register (SRL16), and additional multiplexers and carry logic simplify wide logic and arithmetic functions. Most general-purpose logic in a design is automatically mapped to the slice resources in the CLBs.

Slices:

Each CLB comprises four interconnected slices. These slices are grouped in pairs. Each pair is organized as a column with an independent carry chain. The left pair supports both logic and memory functions and its slices are called SLICEM. The right pair supports logic only and its slices are called SLICEL. Therefore half the LUTs support both logic and memory (including both RAM16 and SRL16 shift registers) while half support logic only, and the two types alternate throughout the array columns. The SLICEL reduces the size of the CLB and lowers the cost of the device, and can also provide a performance advantage over the SLICEM.

Slice Overview:

A slice includes two LUT function generators and two storage elements, along with additional logic.Both SLICEM and SLICEL have the following elements in common to provide logic, arithmetic, and ROM functions:

• Two 4-input LUT function generators, F and G
• Two storage elements
• Two wide-function multiplexers, F5MUX and FiMUX
• Carry and arithmetic logic.

The SLICEM pair supports two additional functions:

• Two 16x1 distributed RAM blocks, RAM16
• Two 16-bit shift registers, SRL16.

Look-Up Tables:

The Look-Up Table or LUT is a RAM-based function generator and is the main resource for implementing logic functions. Furthermore, the LUTs in each SLICEM pair can be configured as Distributed RAM or a 16-bit shift register .Each of the two LUTs (F and G) in a slice have four logic inputs (A1-A4) and a single output (D)(for Spartan 3E XC3S500E). Any four-variable Boolean logic operation can be implemented in one LUT. Functions with more inputs can be implemented by cascading LUTs or by using the wide function multiplexers.

RESTARTABLE BIST LOGIC :

BIST Controller is a finite state machine, whose state transition is controlled by the Test Mode (TM) input. It provides the clock signal to the test pattern generator (LFSR), Circuit Under Test (CUT) and the signature generation circuit (MISR). The BIST controller also decides the input to the circuit under test based on whether the module is in normal mode or test mode on seeing the Test Mode (TM) input.

BLOCK DIAGRAM OF RESTARTABLE BIST LOGIC

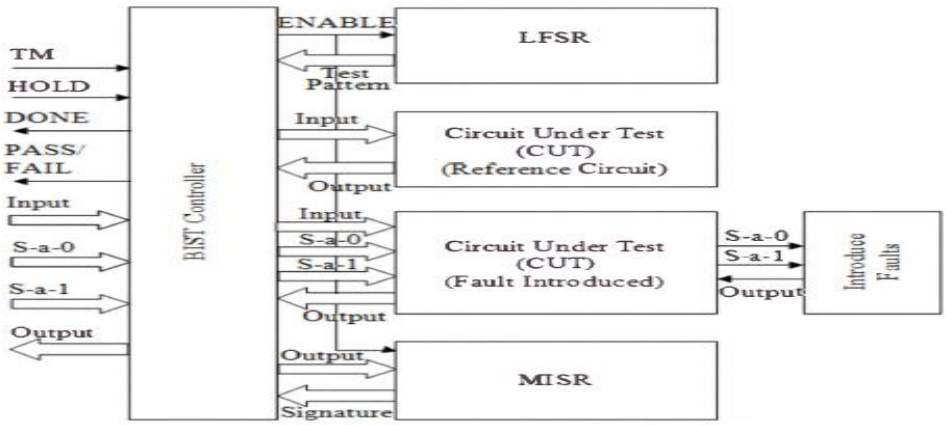


Fig3. Block diagram of BIST controller

A BIST circuit comprises a scan monitor with hold logic and a signature generation element. The hold logic is operable to suspend signature generation in the signature generation element at any desired point in the test sequence. In some embodiments, the hold logic comprises a scan-Ioadable signature hold flip-flop which allows the logic BIST controller to be restarted from any selected pattern within a pattern range and to run to any subsequent pattern. The BIST session can be run incrementally, testing and reporting intermediate MISR signatures. External automatic testing equipment can suspend signature generation at selected times during BIST session to prevent tainting of the signature generation element.

Initially, the registers in LFSR and MISR are reset. Then checking for Test Mode or Normal Mode is done by seeing the TM input pin. If TM is low, external inputs are applied to the circuit under test and the circuit works in normal mode. When the TM signal is changed to high, the BIST Controller enters the test mode. Now BIST Controller sets or resets the ENABLE signal depending on whether the HOLD signal is high or low respectively. When the ENABLE is high, LFSR generates the test vectors. These test vectors are applied to the circuit under test and the output is fed to the MISR. MISR computes the signature. When all the test vectors are applied to the circuit, the signature computed by the MISR is compared with a reference value learned from a fault free replica of the circuit under test. If the signatures match, the circuit is considered as fault free. The BIST Controller sets the outputs PASS/FAIL and DONE to high. Then the registers are reset and the Controller waits for the next TM signal.

If while the BIST is in Test Mode, when HOLD signal is enabled, the ENABLE signal is reset by the BIST Controller. In this case, the circuit goes back to the normal mode and the external signals are applied to the circuit under test. Here the registers are not reset.

Instead, they will hold their current values, so that the LFSR can continue generating test vectors from the point where it got the HOLD signal and the MISR also will continue computation from the paused value. BIST Controller will check for the HOLD signal low to resume testing the circuit under test. Fault detection using restartable logic BIST is implemented as shown in the fig3. Two replicas of same circuit are used for implementing fault detection. One circuit is taken as a reference fault free circuit and on the other; logic is added for introducing s_@_O or s_@_l faults for any wire. Signatures are generated for both the circuits and are compared to detectthe fault.

MATERIALS AND METHODS

Linear Feedback shift Register (LFSR):

A linear feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is XOR. Thus, an LFSR is most often a shift register whose input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register

SIMULATION RESULT FOR TEST PATTERN GENERATOR(LFSR) :

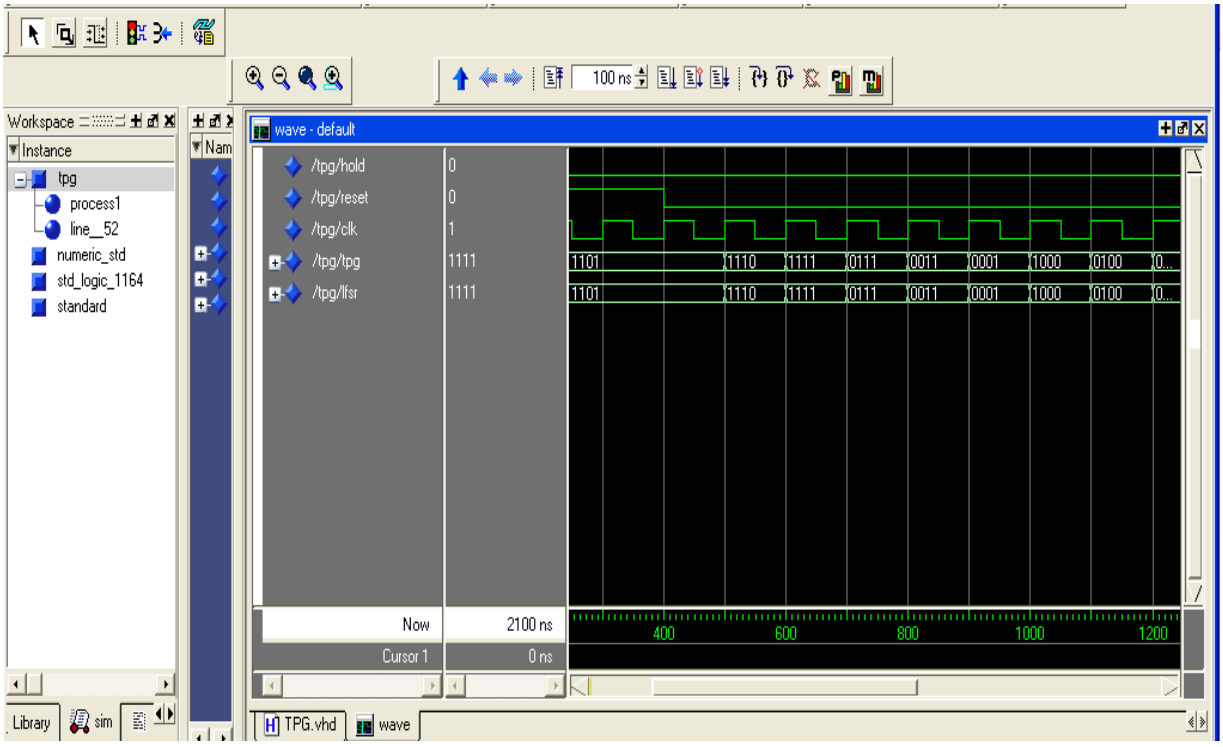


Fig. LFSR

BIST CONTROLLER :

Bist controller is the main module in the project which controls the bist operation. It is a state machine used for state transition from one state to other state. The BIST controller also decides the input to the circuit under test based on whether the module is in normal mode or test mode on seeing the Test Mode (Test) input.

The State machine comprises of six states,(start,resettpg,resetmisr,test,hold,Bistdone). The finite state machine, whose state transition is controlled by the Test Mode (Test) input. It provides the clock signal to the test pattern generator (LFSR), Circuit Under Test (CUT) and the signature generation circuit (MISR).In test state MISR calculates different signatures.

After the seven clock pulses the MISR output is captured and is compared with the Signature register. The output of the comparator is given to the BIST fail signal and the BistDone signal will be raised after completion of Test state.

SIMULATION RESULTS FOR BIST CONTROLLER :

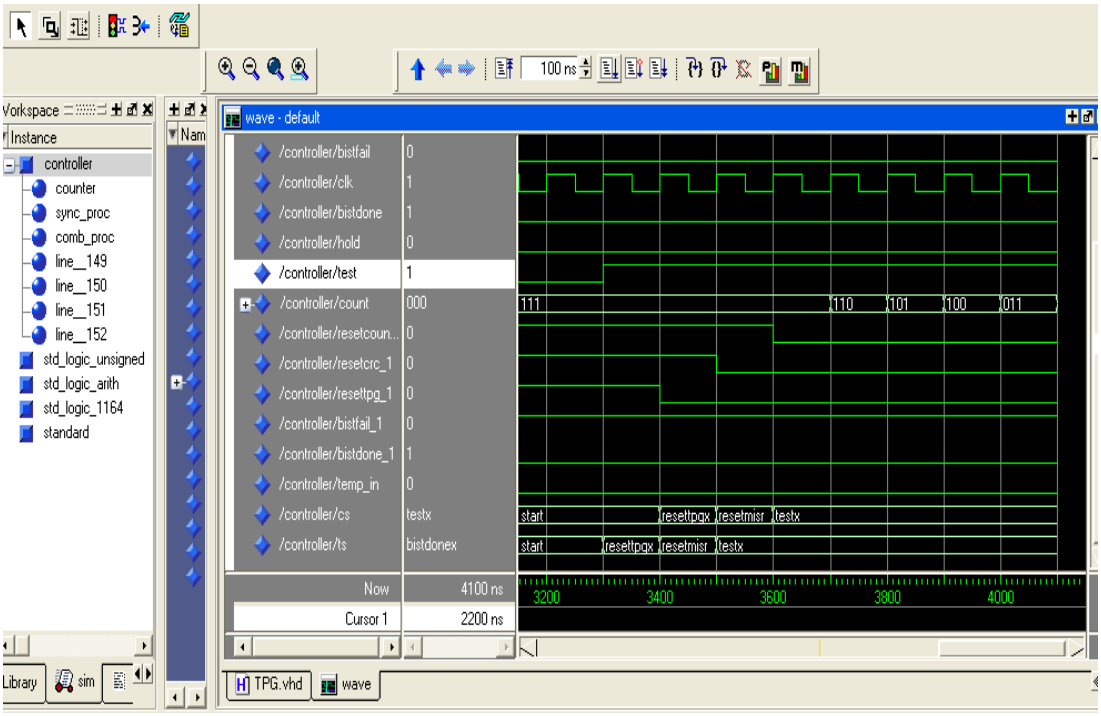
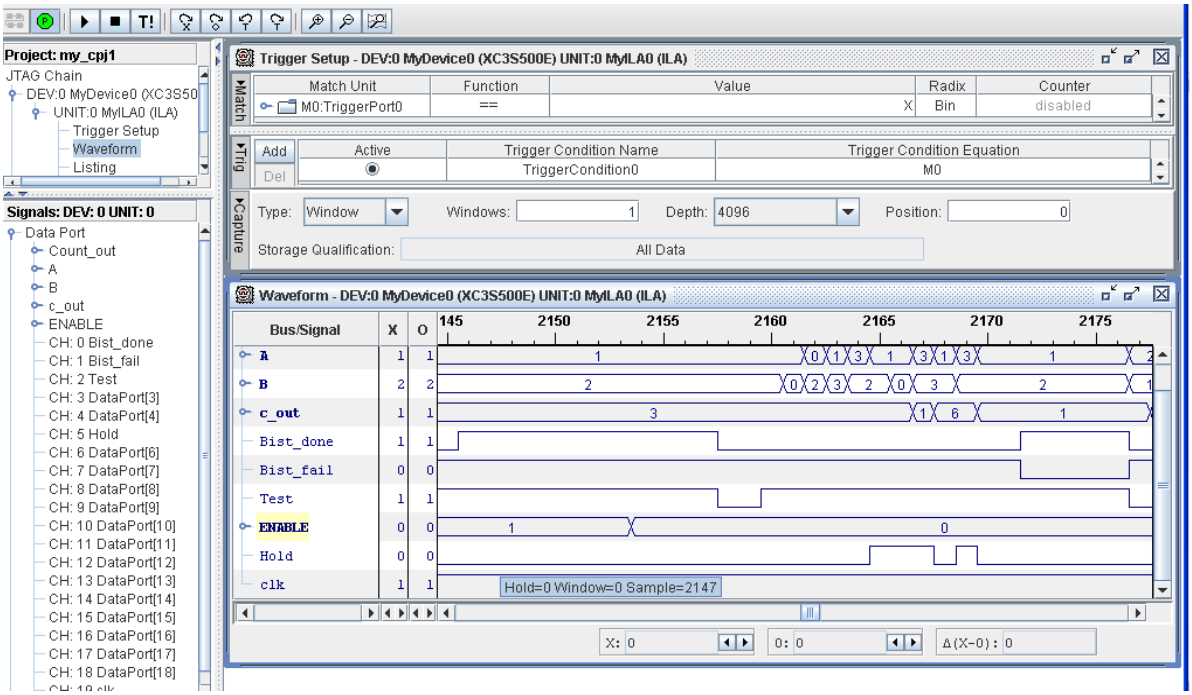


Fig . Bist Controller

CHIPSCOPE RESULTS :



CONCLUSION

The Restartable BIST Controller for Fault detection in CLBs of FPGA was designed successfully and the coding was done in Verilog HDL. The RTL simulations were performed using ModelSim III SE 6.2c from Mentor Graphics. The synthesis was done using Xilinx ISE 9.2. This Design is verified for all test cases. The Restartable BIST works properly for all the test values.

REFERENCE

[1].M.Bushnell and V.D. Agarwal, " Essentials of Electronic Testing for Digital, Memory and Mixed-signal VLSI Circuits" Kluwer Academic Publishers, 2000.
[2]. L.T. Wang, Cheng-Wen Wu and Xiaoqing Wen, "VLSI Test Principles & Architectures Design for testability".
[3] C. Stroud, S. Konala, P. Chen, and M.Abramovici, "Built-in self-test of logic in FPGAs," in Proc. 14th Very Large Scale Integration (VLSI) Test Symp., 1996, pp. 387-392.
[4] C. Metra, G. Mojoli, S. Pastore, D. Salvi, and G.Sechi, "Novel technique for testing FPGAs," Proc. IEEE Design Automation and Test in Europe, pp. 89-94, 1998.
[5] W. K. Huang, M. Y. Zhang, F. J. Meyer, and F. Lombardi, "A XOR-tree based technique for constant testability of configurable FPGAs," in Proc. Asian Test Symp., 1997, pp. 248-253.
[6] W. K. Huang, F. J. Meyer, and F. Lombardi, "Multiple fault detection in logic resources of FPGAs," in Proc. Defect and Fault Tolerance in Very Large Scale Integration (VLSI) Systems, 1997, pp. 186-194.
[7] , "Testing configurable LUT-basedFPGAs," IEEE Trans. VLSI Syst., vol. 6, pp. 276-283,June 1998.
[8] W. K. Huang and F. Lombardi, "An approach for testing programmable/ configurable field programmable gate arrays," in Proc. IEEE Very Large Scale Integration (VLS!) Test Symp., Princeton, NJ, 1996, pp. 450-455.
[9] T. Inoue, H. Fujiwara, H. Michinishi, T.Yokohira, and T. Okamoto, "Universal test complexity offield-programmable gate arrays," in Proc. 4th IEEE Asian Test Symp., Nov. 1995, pp. 259-265.
[10] M. Renovell, J. Figueras, and Y. Zorian, "Test of RAM-based FPGA: Methodology and application to the interconnect structure," in Proc. 15th IEEE Very Large Scale Integraton (VLS1) Test Symp., 1997, pp.
204-209.
[I I] F. Ferrandi, F. Fummi, L. Pozzi, and M. G. Sami, "Configuration-specific test pattern extraction for field programmable gate arrays," in Proc. Defect and Fault Tolerance in Very Large Scale Integration (VLSI) Systems, 1997, pp. 85-93.
[12] T. Liu, W. K. Huang, and F. Lombardi, "Testing of uncustomized segmented channel FPGAs," in Proc. ACMInt. Symp. on FPGAs, Feb. 1995, pp. 125-13I.
[13] A. Doumar, T. Ohmameuda, and H. Ito, "Design of an automatic testing for FPGAs," in Proc. IEEE Euro. Test Workshop, May 1999.
[14] Fast testable design for SRAM-based FPGAs," 1E1CE Trans. Inform. Sys., vol. E83-D, no. 5, pp. 116-1127, May 2000.
[15] "Virtex-5 FPGA User Guide," user manual: UGl90 (v5.3)., May 17, 2010.