International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 1 Issue 1, September 2013

Implementation of Restartable BIST Controller for Fault Detection in CLB of FPGA

M Swetha[1], V Roopa Reddy[2]

[1]Teegala Krishna Reddy Engineering College, Student, M. Tech, VLSI-SD, E.C.E Dept., Hyderabad, India

[2]Teegala Krishna Reddy Engineering College, Assistant Professor, Hyderabad, India

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 [1,11]. The Configurable Logic Blocks (CLBs) are the main logic resources for implementing sequential as well as combinatorial circuits in FPGA [8]. Built-In Self-Test (BIST) is a design technique that allows a circuit to test itself [2]. Here, We are implementing a restart able logic BIST controller for the configurable logic blocks by using the resources of FPGA itself [7, 10]. The design exploits the reprogramability of an 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 [3, 4]. 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 [9]. Integrated circuits are presently tested using a number of structured designs for testability (DFT) techniques. These techniques rest on the general concept of making all or some state variables directly controllable and observable.***

Keywords: FPGA, MXE, Modelsim, Xilinx, DFT

1. Introduction

1.1. 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.

Figure 1: 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-Loadable 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 [1]. 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 [6]. 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_@_0 or s_@_l faults for any wire. Signatures are generated for both the circuits and are compared to detect the fault. Restartable Bist controller comprises of:

1. LFSR(Linear Feedback shift Register)
2. MISR(Multiple Input Signature Register)
3. Bist controller.
4. CUT(circuit under test) with and without Faults.
5. CUT selector.
6. Multiplexor.

1.2 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 [5]. 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 value.

The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle.

Applications of LFSRs include generating pseudo-random numbers, pseudo-noise sequences, fast digital counters, and whitening sequences. Both hardware and software implementations of LFSRs are common.

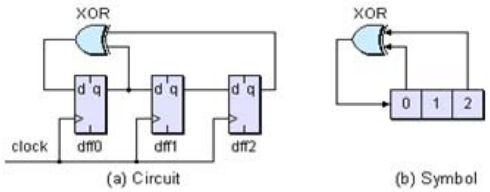

Figure 2

1.3 Multiple Input Signature Register (MISR)

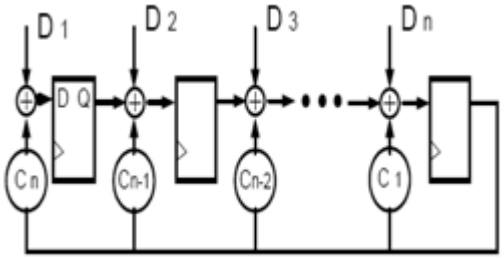

Figure 3: MISR Diagram

MISR (Multiple input signature register) is used to Reduce test logic by using multiple bit streams to create a signature. The MISR (multiplied input signature register) is formed by adding the xor gates at the each flip-flop. Depending on the input provided to the MISR we get an output and depending on the no of registers we capture the output and compare it with the signature register contents. In our project we are implementing a 3 bit MISR. For a 3bit MISR after 7th clock pulse we will be capturing the MISR output. For this we take a counter which counts from 1 to 7 and at 8th clock pulse the output is captured. In this way we compressing the multiple bit stream to perform signature analysis. Signature analysis uses an MISR output to compress the input stream to a single value. Depending on this output we conclude whether there is a fault in the CUT that we are testing.

1.4 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.

1.5 Cut Selector

CUT selector is used to select a cut out of three cut's upon which the bist operation is performed. This is a Multiplexer which has a two bit selection line. This selection line (enable) is the input given by the user. Depending on this select line the three cuts (one cut without fault, one cut with struct at fault, one cut with bit flip fault) are selected at a time. The fault detection logic is implemented on this block to determine the fault in the circuit under test.

2. Results and Discussion

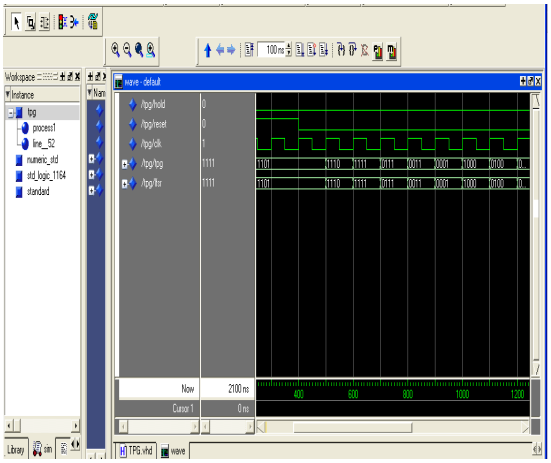

Figure 4: Simulation Result for Test Pattern Generator (LFSR)

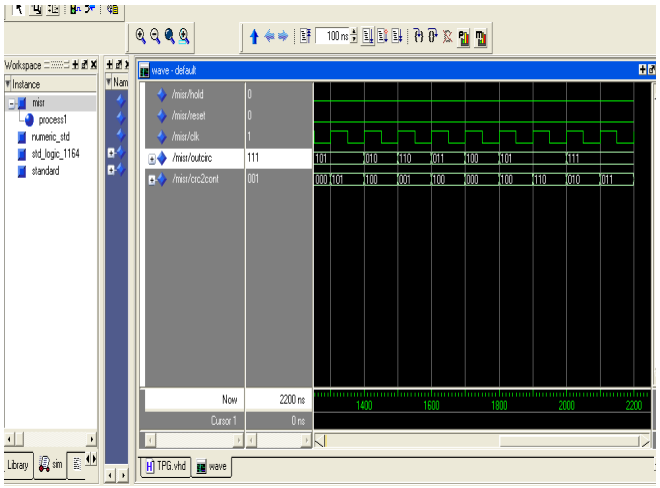

Figure 5: Simulation Result for MISR (Multiple input signature register):

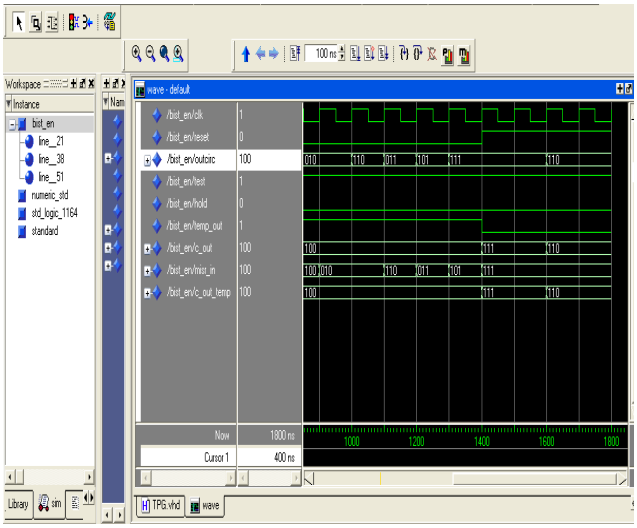

Figure 6: Simulation Result for Selection of Bist/Normal Mode

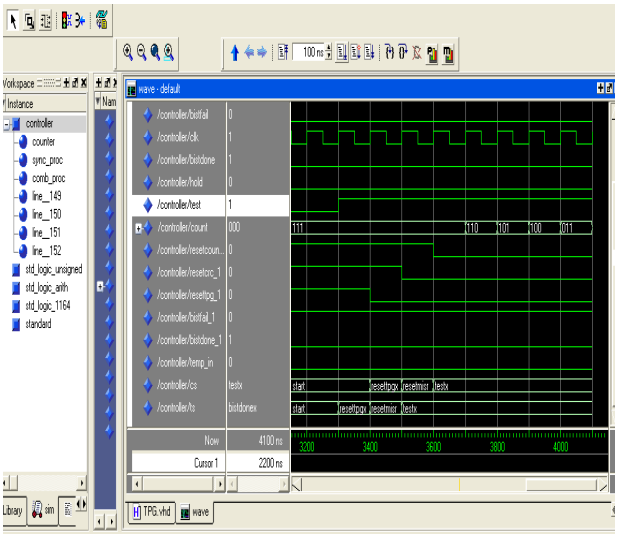

Figure 7: Simulation Results for Bist Controller

3. Chipscope Results

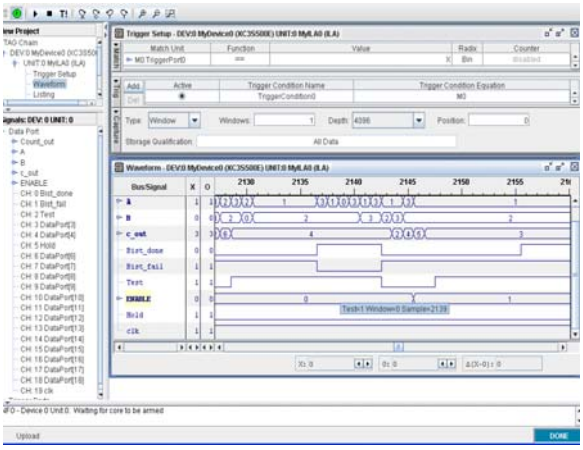

Figure 8: Chipscope Result 1

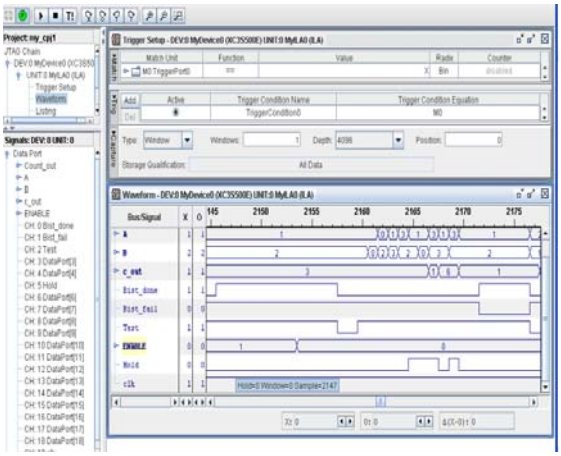

Figure 9: Chipscope Result 2

4. Conclusion

In this paper we have shown the simulation results and Chipscope results for a BIST controller using VHDL by adding Restartable Logic. Restartable BIST controller is designed to monitor fault detection activity with hold logic

and a signature generation element [14]. The hold logic is operable to suspend signature generation in the signature generation element at any desired point in the test sequence. Signature mismatch with the reference signature means that the circuit is faulty. Fault detection using Restartable logic BIST is implemented for different cuts. Three 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. Similarly bit flip faults are for other cut for any of the wire. Signatures are generated for both the circuits and are compared to detect the fault.

References

[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-based FPGAs," 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.
[11] 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.