IMPROVING THE PERFORMANCE OF FPGA USING BIST CONTROLLER

P. Nagaraju,
Assistant Professor, Department of ECE, Narayana Engineering College, Gudur, AP, 524101
Email:[1] pnaga5151@gmail.com

K.Anusha, A.Divya, M.Sankeerthana G. Janardhana D.Santhan Kumar
UG Student, Department of ECE, Narayana Engineering College, Gudur, AP, 524101
,konduruanusha10@gmail.com

Abstract: Today field programmable Gate Arrays (FPGA) are extensively used in majority applications. These FPGAs are prone to various Types of faults to other complex integrated circuits chips. Faults may occur due to lot of reasons like environmental conditions or aging of the device. The probability of occurrence of permanent faults can be clearly high in emerging technologies and hence there is necessity for periodic testing of such FPGA. Built-in self-test (BIST) is a design technique that allows a circuit to test itself It is a set of structured-test techniques for combinational and sequential logic, memories, multipliers and other embedded logic blocks. The principle is to generate test vectors, apply them to the circuit under test or device under test, and then verify the response. Being an automated testing, BIST enables testing at high speed and high fault coverage. BIST controller coordinates the operations of different blocks of the BIST. Based on the test mode (TM) input to the controller, the system either operates in the normal mode or in the test mode. In this paper we explain an implementation of a restartable logic BIST controller for a combinational logic circuit using VHDL. It allows us to suspend the signature generation at any desired point in the test sequence. In this case, the BIST circuit is considered to comprise hold logic and a signature generation element. The hold logic will be implemented such that an external signal (HOLD) can temporarily suspend signature generation in the signature generation element at specified times during the BIST session.

I.INTRODUCTION

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 reprogram ability 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.

II.LITERATURE SURVEY

This chapter reviews some references from previous projects, journals, articles and books. All this information were collection from different sources such as internet, products, manuals etc. The information gathered in this chapter is related to background study of this project.

The main drivers for the widespread development of BIST techniques are the fast rising costs of ATE testing and the growing complexity of integrated circuits. It is now common to see complex devices that have functionally diverse blocks built on different technologies inside them. Such complex devices require high-end mixed-signal testers that possess special digital and analog testing capabilities. BIST can be used to perform these special tests with additional on-chip test circuits, eliminating the need to acquire such high-end testers.

The paper “Hardware Implementation of Functional Verification Using Signature Analysis” by Sethulakshmi.R and Greeshma.R states that The main consequences facing in the semiconductor technology is the testing time, and the one of the solution for it is the BIST, which allows the system to check itself. In these paper the functional test is done sing BIST, that is we checking whether the circuit is behaved as intended. BIST implementation techniques are of different types, in all of the techniques the main disadvantage is that as the size of the CUT changes the LFSR and the MISR also need to change, So to overcome that we designing a BIST processor that can test the circuit unto sixteen bits.

The paper “Build-In-Generation of functional broadside tests using fixed hardware structure” by Irith Pomeranz states that Functional broadside tests are two-pattern scan-based tests that avoid over testing by ensuring that a circuit traverses only reachable states during the functional clock cycles of a test. In addition, the power dissipation during the fast functional clock cycles of functional broadside tests does not exceed that possible during functional operation. On-chip test generation has the added advantage that it reduces test data volume and facilitates at-speed test application. This paper shows that on-chip generation of functional broadside tests can be done using a simple and fixed hardware structure, with a small number of parameters that need to be tailored to a given circuit, and can achieve high transition fault coverage for testable circuits. With the proposed on-chip test generation method, the circuit is used for generating reachable states during test application. This alleviates the need to compute reachable states offline.

The paper “On generating pseudo-functional delay fault tests for scan designs” by] Z. Zhang, S. M. Reddy, and I. Pomeranz states that In designs using DFT, such as scan, some of the faults that are untestable in the circuit without DFT become testable after DFT insertion. Additionally, scan tests may scan in illegal or unreachable states that cause non functional operation of the circuit during test. This may cause higher than normal power dissipation and demands on supply current. We propose new techniques to determine illegal states of circuits that can be used during ATPG to prohibit tests using such states. The resulting tests are essentially functional or pseudo functional.

The paper “An approach for testing programmable/configurable field programmable gate arrays” by W. K. Huang and F. Lombardi states that This paper presents a new general technique for testing field programmable gate arrays (FPGAs) by fully exploiting their programmable and configurable characteristics. A hybrid fault model is introduced based on a physical and behavioral characterization; this permits the detection of a single fault, as either a stuck-at or a functional fault. A general approach which regards testing as can application for the reconfigurable FPGA, is then proposed. It is shown that different arrangements of disjoint one-dimensional arrays with unilateral horizontal connections and common vertical input lines provide a very good solution. A further feature that is considered for array testing, is the relation between the configuration of the logic blocks and the number of I/O pins in the chip. As an example, the proposed approach is applied for testing the Xilinz 4000 family of FPGAs.

The paper “Multiple fault detection in logic resources of FPGAs” by W. K. Huang, F. J. Meyer, and F. Lombardi proposed an approach to detect multiple faults in FPGAs. The approach exploits the testability of the AND tree and OR tree with the configurability and programmability of SRAM-based FPGAs. The proposed AND tree- and OR tree-based testing structure is simple and the conditions for constant testability can easily be satisfied. Test generation for only a single CLB is required as the AND/OR approach scales to larger FPGAs. CLB test generation can assume any desired fault model. Any number of faulty CLBs in the chip can be detected.

The paper "Testing of uncustomized segmented channel FPGAs," by T. Liu, W. K. Huang, and F. Lombardi states that methodology for testing the configurable logic of RAM based FPGAs taking into account the configurability of such flexible devices. The methodology is illustrated using the XILINX 4000 family. On this example of FPGA, we obtain only 8 basic test configurations to fully test the whole matrix of CLBs. In the proposed test configurations, all the CLBs have exactly the same configuration forming a set of one dimensional iterative arrays. The iterative arrays present a C-testability property in such a way that the number of test configurations 8 is fixed and independent of the FPGA size.

III.IMPLEMENTATION OF FPGA WITH BIST TECHNIQUE

(i).Field Programmable Gate Array:

FPGAs, alternative to the custom ICs, can be used to implement an entire System On one Chip (SOC). The main advantage of FPGA is ability to reprogram. User can reprogram an FPGA to implement a design and this is done after the FPGA is manufactured. This brings the name “Field Programmable. “Custom ICs are expensive and takes long time to design so they are useful when produced in bulk amounts. But FPGAs are easy to implement within a short time with the help of Computer Aided Designing (CAD) tools (because there is no physical layout process, no mask making, and no IC manufacturing)

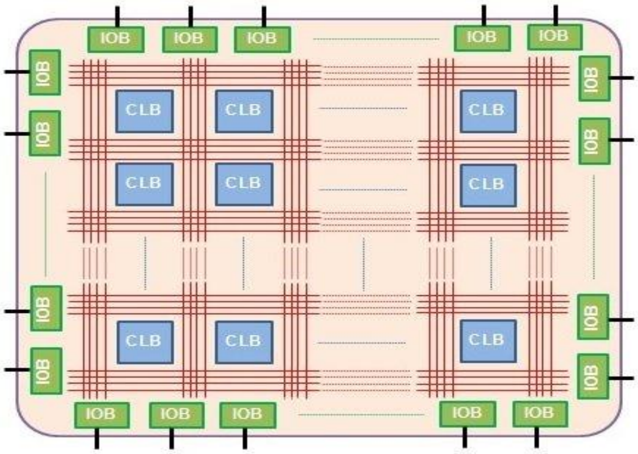


Fig 1: FPGA Architecture

(ii).Fault and Fault Models

A fault is the representation of a defect reflecting a physical condition that causes a circuit to fail to perform in a required manner. Faults can be divided into two categories:

1. Permanent Faults

2. Transient Faults

Fabrication faults and design faults are among the permanent faults. Transient faults, commonly called single event upsets (SEUs), are brief incorrect values resulting from external forces (terrestrial radiation, particles from solar flares, cosmic rays, and radiation from other space phenomena) altering the balance or locations of electrons, usually in a small area of the system.

Fault models can be divided into three types

1. Interconnect fault model

2. Logic Block fault model

3. Delay Fault

Logic Block fault model: Logical faults represent the effect of physical faults on the behavior of the system. A fault model is an engineering model of something that could go wrong in the construction or operation of a piece of equipment. From the model, the designer or user can then predict the consequences of this particular fault. Fault models can be used in almost all branches of engineering.

(iii). BIST Technique:

Built-in Self-Test, or BIST, is the technique of designing additional hardware and software features into integrated circuits to allow them to perform self-testing, i.e., testing of their own operation(functionally, parametrically, or both) using their own circuits, thereby reducing dependence on an external automated test equipment (ATE).

BIST is a Design-for-Testability (DFT) technique, because it makes the electrical testing of a chip easier, faster, more efficient, and less costly. The concept of BIST is applicable to just about any kind of circuit, so its implementation can vary as widely as the product diversity that it caters to. As an example, a common BIST approach for DRAM's includes the incorporation onto the chip of additional circuits for pattern generation, timing, mode selection, and go-/no-go diagnostic tests.

BIST is also the solution to the testing of critical circuits that have no direct connections to external pins, such as embedded memories used internally by the devices. In the near future, even the most advanced tester may no longer be adequate for the fastest chip, a situation wherein self-testing may be the best solution for.

BIST techniques are classified in a number of ways, but two common classification of BIST are the Logic BIST (LBIST) and the Memory BIST (MBIST). LBIST, which is designed for testing random logic, typically employs a pseudo-random pattern generator (PRPG) to generate input patterns that are applied to the device's internal scan chain, and a multiple input signature register (MISR) for obtaining the response of the device to these test input patterns. An incorrect MISR output indicates a defect in the device.

MBIST, as its name implies, is used specifically for testing memories. It typically consists of test circuits that apply, read, and compare test patterns designed to expose defects in the memory device. There now exists a variety of industry-standard MBIST algorithms, such as the "March" algorithm, the checkerboard algorithm, and the varied pattern background algorithm.

In this project we are implementing Restartable BIST controller which is designed to monitor fault detection activity 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. Signature mismatch with the reference signature means that the circuit is faulty. The block diagram for Restartable BIST controller is shown as below.

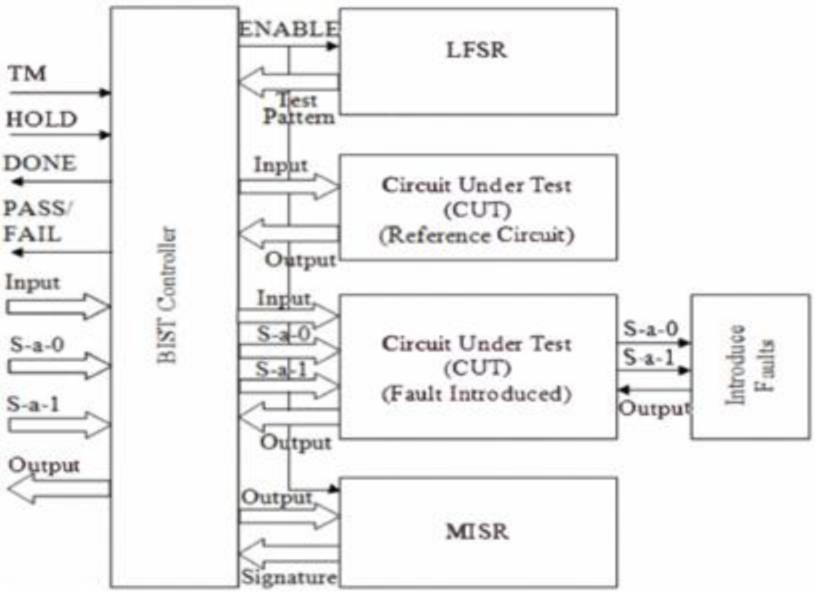


Fig 2: Block Diagram of BIST Controller

The Restartable BIST Controller is a 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). 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.

Initially, the registers in LFSR and MISR are reset. Then checking for Test Mode or Normal Mode is done by seeing the Test mode (Test) input pin. If Test is low, external inputs are applied to the circuit under test and the circuit works in normal mode. When the Test 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 [CUT] and the output is fed to the MISR (Multiple input Signature register). 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 BISTFAIL as Low and BISTDONE to high. Then the registers are reset and the Controller waits for the next Test 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.

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 two cut's are faulty circuits. One cut is added introducing s_@_O or s_@_l faults for any wire and other cut with bit flip faults. The particular bit of the actual output of cut is flipped from '1' to '0' vice versa. Thus we will be using an Enable signal for selecting the particular cut for detecting the fault. Signatures are generated for all the circuits and are compared to detect the fault.

IV.SIMULATION RESULTS

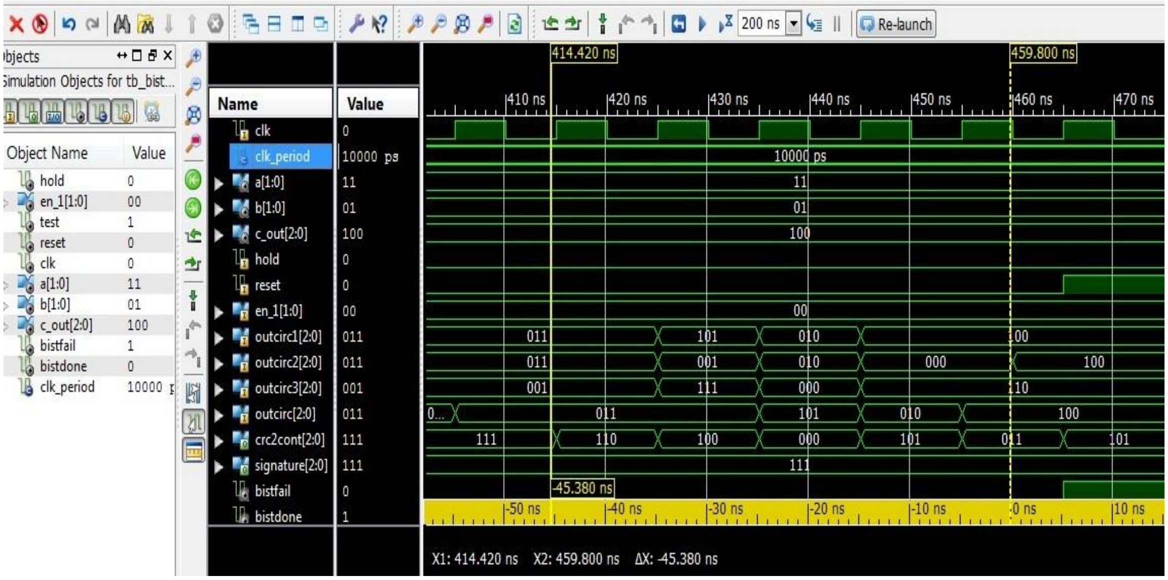
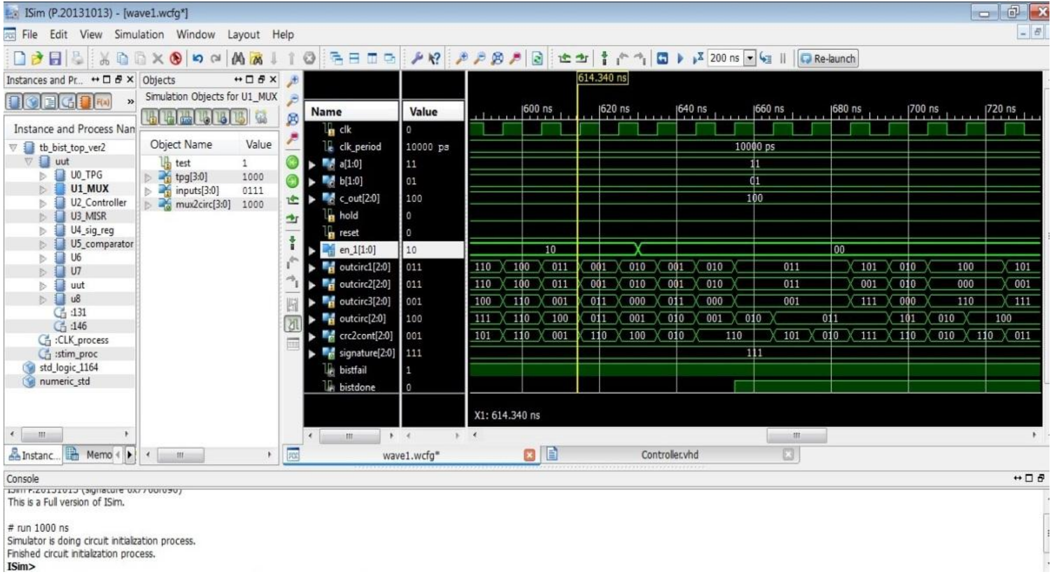


Fig 3: Simulation result for fault free circuit



Object Name	Value
test	1
tpg[3:0]	1000
inputs[3:0]	0111
mux2circ[3:0]	1000

Name	Value
clk	0
clk_period	10000 ps
a[1:0]	11
b[1:0]	01
c_out[2:0]	100
hold	0
reset	0
en_1[1:0]	10
outcirc1[2:0]	011
outcirc2[2:0]	011
outcirc3[2:0]	001
outcirc[2:0]	100
crc2cont[2:0]	001
signature[2:0]	111
bistfail	1
bistdone	0

```
This is a Full version of ISim.

# run 1000 ns
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Fig 4: Simulation result for faulty circuit

V.CONCLUSION

In this paper, we have shown the simulation 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. 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. Jamuna.S and V.K. Agrawal, “Implementation Of BISTControllerfor Fault Detectionin CLB Of FPGA”
2. M.Bushnell and V.D. Agarwal, “Essentials of Electronic Testing for Digital, Memory and Mixed-signal VLSI Circuits" Kluwer Academic Publishers, 2000.
3. L.T. Wang, Cheng-Wen Wu and Xiaoqing Wen, "VLSI Test Principles & Architectures Design for testability".
4. 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.
5. 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.
6. 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.
7. 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.
8. "Testing configurable LUT-based FPGAs," IEEE Trans. VLSI Syst., vol. 6, pp. 276- 283,June 1998.
9. 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.

10. 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.