Hardware Implementation of Functional Verification Using Signature Analysis

Sethulakshmi. R
PG Student
Electronics and Communication
Mangalam College of Engineering,Kottayam

Greeshma. R
Asst. Professor
Electronics and Communication
Mangalam College of Engineering,Kottayam

Abstract

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.

Key words: BIST, CUT, LFSR, MISR.

I. Introduction

Testing of integrated circuits (ICs) is of crucial importance to ensure a high level of quality in product functionality in both commercially and privately produced products. The impact of testing affects areas of manufacturing as well as those involved in design. Given this range of design involvement, how to go about best achieving a high level of confidence in IC operation is a major concern. This desire to attain a high quality level must be tempered with the cost and time involved in this process. These two design considerations are at constant odds. It is with both goals in mind (effectiveness vs. cost/time) that Built-In-Self Test (BIST) has become a major design consideration in Design-For-Testability (DFT) methods

Automatic test equipments (ATE) that are used for testing the circuits are relatively expensive. This hinders the use of automatic test equipments in smaller applications. Also with the increased level of integration ATEs became inefficient. Since the Automatic test equipments are testing the entire circuit there is an increase of test time. Due to the development of VLSI the amount of volume confined in a small region has also increased considerably leading to an increase of test data volume. Also many of the modern embedded SoC (system on chip) systems cannot be accessed directly from outside the chip. With increasing integration density,the amount of manufacture faults is increasing due to these issues it became necessary to develop a system that is capable to generate self testing mechanism. Thus the idea of a new DFT technique, BIST came up, with which it became possible to generate patterns for self test. With BIST it became possible to generate a uniform test mechanism for production, systems and maintenance. Also dynamic test of system properties became possible. It can be tested during operation and maintenance.

Functional Tests consist of the input vectors and the corresponding responses. They check for proper operation of a verified design by testing the internal chip nodes. Functional tests cover a very high percentage of modeled (e.g., stuck type) faults in logic circuits and their generation is the main topic of this tutorial. Often, functional vectors are understood as verification vectors, which are used to verify whether the hardware actually matches its specification. However, in the ATE world, any vectors applied are understood to be functional fault coverage vectors applied during manufacturing test. These two types of functional tests may or may not be the same.

II. BIST implementation

BIST techniques have long been recognized as a means to reduce life cycle test and maintenance costs by embedding external tester features, such as test pattern generation (TPG) and output response analysis (ORA), into the part that contains the circuit under test (CUT) (Fig. 1).

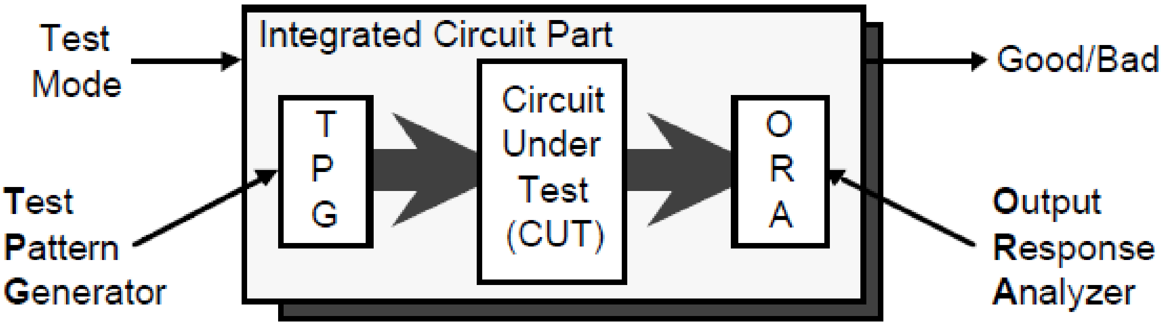

Fig. 1 Built-in self-test configuration

In general, the basic BIST consist of a Test Pattern Generator (TPG), and a circuit to be tested (CUT) and a output response analysis to compress results into a signature for simplicity. The circuit to be tested is driven by external primary inputs during the normal mode, and during the test mode patterns are applied to the circuit that is generated by built pattern generator and. response of the circuit is observed. Responses of the actual and tested circuit are noted, and if an mismatch between the two responses indicate a fault in the circuit (CUT).

Test pattern generation is the process of defining an effective test set which will drive the circuit under test so that the faults in the circuit will cause a different response at the primary outputs from the non-faulty outputs. In this paper we using Pseudorandom pattern generator which is accomplished by utilization of LFSRs (Linear Feedback Shift Register) because they require a small amount of hardware to generate a large set of patterns. For an n stage LFSR, there are 2_n-1 states, and the M-sequence is 2n-1 bits long. Hence, the M-sequence is periodic, and after the 2n-1 distinct values, it repeats itself in the next samples.

The response of a BIST-driven circuit can be compared to its expected response to determine whether the circuit is operating correctly. Instead of storing the patterns of the expected responses, a multiple-input signature register (MISR) compresses the patterns generated by the circuit to form a signature. The signature of a correctly operating circuit is stored for comparison to the actual response. Thus, the MISR circuit and the signature eliminate the need to monitor and compare the response of the individual test patterns. The MISR shown in fig. 1 is driven by the response vectors of the circuit.

However, BIST designs can differ in many ways. Each method has its own set of trade-offs and design considerations. If the BIST design is not appropriate for the IC it is testing, then it can actually be a detriment to the design.

The main drawback in the basic BIST mention above is the number of bits of the circuit to be tested. That is as the bit size of the circuit is changed the LFSR and the MISR also need to change. So in order to overcome this problem a BIST processor is used which control all the operations include initialization and length of the BIST sequence. In this paper we presented a BIST processor that can execute and manages the circuit up to sixteen bit.

The main blocks in the BIST processor are Control block, Arithmetic and logic block (ALU), data memory, program memory .Before functional operation starts each block in the processor need to be initialized, and this initialization may be achieved by applying by asserting a reset input to each block. In the ALU all the arithmetic and logical operations are done, that is sequence generation and signature analysis is performed in the ALU block. The main components in the ALU are LFSR which generate the input patterns for the circuit, MISR which used to find the signature values and a comparator which compares the signature values of the actual circuit and the circuit to be tested. The input signals to be the ALU are signature value of the actual circuit, input value for the MISR, reset for LFSR, reset for comparator, and reset for MISR.Initially when the LFSR reset is asserted the corresponding LFSR according the size of the CUT is selected, and the sequence is applied to the actual circuit and, then the output of circuit is given to the MISR ,when reset for MISR is asserted the signature value of the circuit is produced. After getting the signature value of the actual circuit the reset button of both the MISR and LFSR need to made a high to low transition for initializing it again for finding the sequence and the signature value of the circuit to be tested. Then the comparator reset need to be asserted to compare the signature value of both the circuits. If any mismatch between the two values it indicates error in the CUT. Since comparator output value is initialized to zero, so we set a flag to indicate the comparator starts its operation.

Each block in the ALU need to be synchronized with each other for the proper operation, so for that we introducing a control block which generates the entire reset button for the ALU block. The input to these control block only contain the clock and the main reset for the BIST processor, that is these control block control all the operations of the BIST processor. For the ease of operation the control block is designed using a state machine.

Each state in the state machine is defined in the program memory. In the real time the actual circuit is tested only once and its signature value is stored in the data memory. Then the stored signature value is compared with the signature value of the CUT.

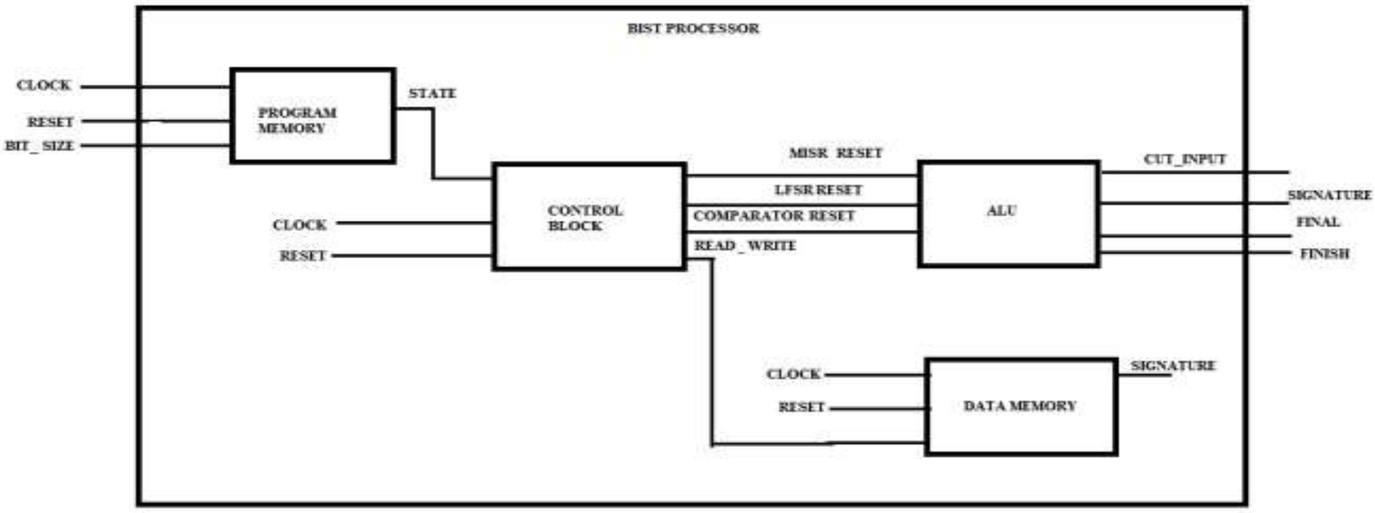

Fig.2.BIST Processor

III. SIMULATION AND SYNTHESIS RESULTS

First the basic BIST is designed and each block such as TPG,ORA and the CUT is described using verilog hardware description language and its functionality is checked. Then individual blocks are combined together to form a top level module .The functional verification is checked using the Xilinx 13.2. Circuit is synthesized using Synthesis tool. The output of synthesis process generates a netlist. The netlist is list of interconnection of components used to implement the BIST. The synthesis report also tells us about the device utilization i.e how much of FPGA's resources (components) are used to realize the BIST on FPGA and about the propagation delays of the circuit implemented on FPGA in timing report.

Then additional blocks like data memory, program memory, control block, and the ALU are added to implement built-in-self-test capability. The Spartan-3 Field Programmable Gate Arrays (FPGA) from Xilinx is used for realization of the BIST capability.

IV. CONCLUSION

Testing cost of the chips cost is very-very high using (Automatic Test Equipment). Testing cost of chips is reaching about 40% of the total chip cost. So to overcome from this problem, BIST Technique is a solution. This reduces the testing cost, but hardware requirements are increased Also the speed of circuit is decreased due to delay in extra hardware. But there is trade-off between testing cost, extra hardware and speed of the circuit. In this project we implemented a BIST processor that is more reliable than the basic BIST. And these can check circuit up to sixteen bits.

REFERENCES

[1] Irith Pomeranz,"Build-In-Generation of functional broadside tests using fixed hardware structure" Ieee Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 21, No. 1, January 2013.

[2] Mrs.Jamuna.S, Dr. V.K. Agrawal,**"** Implementation of BIST Structure using VHDL for VLSI Circuits" IJEST Vol. 3 No. 6 June 2011

[3] M.Bushnell and V.D. Agarwal, " Essentials of Electronic Testing for Digital, Memory and Mixed-signal VLSI Circuits" Kluwer Academic Publishers, 2000

[4] M.Bushnell and V.D. Agarwal, " Essentials of Electronic Testing for Digital, Memory and Mixed-signal VLSI Circuits" Kluwer Academic Publishers, 2000.

[5] L.T. Wang, Cheng- Wen Wu and Xiaoqing Wen, " VLSI Test Principles & Architectures Design for testability".

[6] Z. Navabi, "VHDL Analysis and Modeling of Digital" Systems*",* McGraw-Hill Inc., 1991.