



FPGA Implementation of Functional Broadside Test Using Fixed Hardware Structure

Ancy Joy¹, Sani John², Sruthi James³

PG Student [VLSI & Embedded Systems], Dept. of ECE, VJCET, Vazhakulam, Kerala, India¹

Assistant professor, Dept. of ECE, VJCET, Vazhakulam, Kerala, India²

PG Student [VLSI & Embedded Systems], Dept. of ECE, VJCET, Vazhakulam, Kerala, India³

ABSTRACT: Testing of VLSI circuits are normally a tedious process due to the test data volume. It leads to extra power dissipation along with minimum fault coverage. Initialization of states to known reachable states is very important to reduce unwanted operations. The fault coverage can be improved by using efficient lfsr structure to generate input pattern. 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 broad side 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. Also this paper explains the variations in output fault coverage due to different methods of input generation.

KEYWORDS: Functional broadside tests, Lfsr, Reachable states.

I. INTRODUCTION

Very Large Scale Integration (VLSI) has made a dramatic impact on the growth of integrated circuit technology. It has not only reduced the size and the cost but also increased the complexity of the circuits. The positive improvements have resulted in significant performance/cost advantages in VLSI systems [1]. There are, however, potential problems which may retard the effective use and growth of future VLSI technology. Among these is the problem of circuit testing, which becomes increasingly difficult as the scale of integration grows. Because of the high device counts and limited input/output access that characterize VLSI circuits, conventional testing approaches are often ineffective and insufficient for VLSI circuits. Built-in self-test (BIST) is a commonly used design technique that allows a circuit to test itself. BIST has gained popularity as an effective solution over circuit test cost; test quality and test reuse problems. Test time is a significant component of IC cost. It needs to be minimized and yet has to have maximum coverage to ensure zero-defect.

The goal of testing is to apply a minimum set of input vectors to the device to determine if it contains a defect. Costs increase dramatically as faulty components find their way into higher levels of integration. Thus, there is a need for design for testability techniques. For any testing methodology, the following factors should be considered- high and easily verifiable fault coverage, minimum test pattern generation, minimum performance degradation, at-speed testing, short testing time, and reasonable hardware overhead. With increasing integration density, the amount of manufacture faults is increasing. Thus we have to test the chip. With increasing complexity of the design, it becomes impossible to test the chip externally. Thus, we have to use Built-In Self-Test (BIST) provides a feasible solution to the above demands. Another advantage of this methodology is that the test patterns are not applied by external Automatic Test Equipments (ATEs) but generated by in built testing circuit. It saves the memory requirement during test. Built-In Self-Test is a design technique in which parts of a circuit are used to test the circuit itself.

A scan-based delay test consists of two patterns, V_1 and V_2 , applied on two successive clock pulses. The first pattern initializes the node or path being tested, the second pattern causes a transition on that node or along that path. But scan based design tests limited to functional justification have lower fault coverage. Also it causes to unnecessary yield loss.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

There is a wide range of deterministic logic BIST (DLBIST) methods[2] that apply deterministic test patterns and hence improve the low fault coverage often obtained by pseudorandom patterns. But due to the presence of extra hardware the power dissipation gets increases.

In functional broad side test[3] also inefficient design of finite state machine will guide the circuit to unreachable states and it will not return back. Therefore initialize the circuit to a known state then the operations will lead through the known states of finite state machine. Here pseudo random generators like linear feedback shift registers(LFSR)[4] are used to generate random sequences. This will help to improve the fault coverage. All the transition delay faults can be tested with minimum power and delay time. The units used is an LFSR, Circuit under test(CUT) ,Compactor and a Comparator to compare true value with faulty circuit output.This is the main logic of all the broadside tests[5].By using these logic we can derive a number of method to test circuits.

This paper is organized as follows. Section II gives an overview of generation of input patterns and application of functional broadside tests. Section III describes the details. Section IV presents experimental results demonstrating the achievable fault coverage.

II OVERVIEW

There are many built-in self test(BIST) schemes in existence. Among them most widely used BIST approach is the signature analysis. In signature analysis,the test responses of a system are compacted into a signature using a linear feedback shift register (LFSR) or a multiple input signature register (MISR).

Figure 3.1 shows the BIST block diagram. The signature of the circuit under test (CUT) is compared with the expected (reference) signature. If they both match, the CUT is declared fault free, else it is declared faulty. Since several thousands of test responses are compacted into a few bits of signature by an LFSR/MISR, there is an information loss. As a result some faulty devices may have the same correct signature. The probability of a faulty device having the same signature is called the probability of aliasing. This probability may be higher if the faults in the CUT are correlated. As a result there is no guarantee that the device declared to be fault free is really fault free.

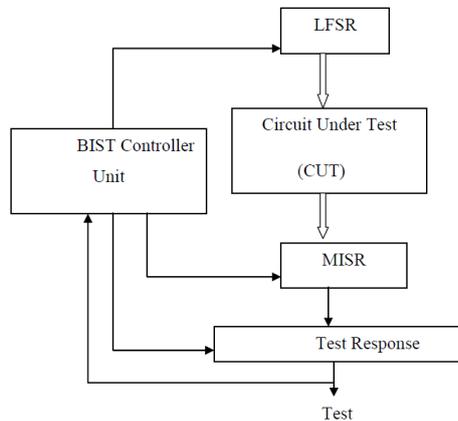


Fig. 1-BIST Block Diagram

Actual test selection depends upon the manufacturing level (processing, wafer, or package) being tested[6]. Although some testing is done during device fabrication to assess the integrity of the process itself, most device testing is performed after the wafers have been fabricated. The first test, known as wafer sort or probe, differentiates potentially good devices from defective ones. After this, the wafer is scribed and cut, and the potentially good devices are packaged. Also, during wafer sort, a test site characterization is performed. Specially designed tests are applied to certain test sites containing specific test patterns. These are designed to characterize the processing technology through measurement of parameters such as gate threshold, polysilicon field threshold, bypass, metal field threshold, poly and metal sheet resistances, contact resistance[6], etc.

The hardware used in this paper for generating the primary input sequence consists of a linear-feedback shift-register (LFSR)[7] as a random source, and of a small number of gates (at most six gates are needed for every one of the benchmark circuits considered). The gates are used for modifying the random sequence in order to avoid cases where



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

the sequence takes the circuit into the same or similar reachable states repeatedly. This is referred to as repeated synchronization. In addition, the on-chip test generation hardware consists of a single gate that is used for determining which tests based on will be applied to the circuit. The result is a simple and fixed hardware structure, which is tailored to a given circuit only through the following parameters.

- 1) The number of *LFSR* bits.
- 2) The length of the primary input sequence .
- 3) The specific gates used for modifying the *LFSR* sequence into the sequence .
- 4) The specific gate used for selecting the functional broadside tests that will be applied to the circuit based on .
- 5) Seeds for the *LFSR* in order to generate several primary input sequences and several subsets of tests.

The test generation hardware used in this paper has a simple and fixed structure, and it is independent of the number of sequences used. The sequences differ only in the seed used for the *LFSR*. The seeds can be stored on-chip, or a seed can be scanned in together with the initial state of the circuit before the application of every primary input sequence. When the circuit-under-test is embedded in a larger design, its primary inputs may be driven by other logic blocks that are part of the same design. In addition, the primary inputs of the circuit-under-test include any external inputs of the design that drive the circuit-under-test. The primary outputs of the circuit-under-test may drive other logic blocks, or they may be primary outputs of the complete design. For simplicity this paper assumes that primary inputs can be assigned any combination of values.

III FUNCTIONAL BROADSIDE TEST

This section describes the on-chip generation of input patterns by using new efficient LFSR design based on the concepts discussed in Section II. Section III-A describes the generation of the input sequence using LFSR Section III-B describes the selection of tests that will be applied based on the number of input pins and fault level of the circuit. Section III-C describes the selection of parameter values in order to match the on-chip test generation hardware to a given circuit. The parameters of the on-chip test generation hardware are summarized in Table I for ease of reference.

A. GENERATION OF INPUT PATTERN

The easiest way to generate input pattern is using binary counter. But this is very old technique and it takes a long time to detect faults. LFSR is an advanced version ,it gives maximum randomness in the generation of patterns. Therefore within minimum clock cycles maximum number of faults detected. Here an advanced LFSR is designed with some additional parameters. By using these parameters for a fixed hardware structure an LFSR can be designed. The basic parameters used in the design of LFSR is shown in TABLE II.

Parameters	Meaning
L	Length of primary input sequence
D	Number of LFSR bits per primary input
mod	Number of LFSR bits for modifying the value of primary input
sel	Test starting at the time units that are divisible by sel

TABLE II. TESTING PARAMETERS

Consider an LFSR with length L , from these LFSR we are deriving the input patterns. L can be any finite value .As the length of LFSR goes on increasing the efficiency of test patterns also increases. D is the number of LFSR[8] bits per primary input. As D increases the overlapping between the successive input bits gets decreases. The overlapping of bits may leads to reduce the fault coverage because the overlapped input patterns are not capable to activate all the nets. Therefore as the value of D increases the randomness in successive input bits gets increases ,output fault coverage will be maximum with minimum number of clock cycles, mod is the number of LFSR bits modified to get the input pattern. The normal combinational logic or simple two input gates such as *AND* ,*OR* are used to modify the LFSR bits.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

Also the input patterns will initialize the nets to a known state. Once it enters into unreachable state, then the finite state machine will start operation with some unwanted states. For illustration, Fig.2 shows the hardware used for with parameters.

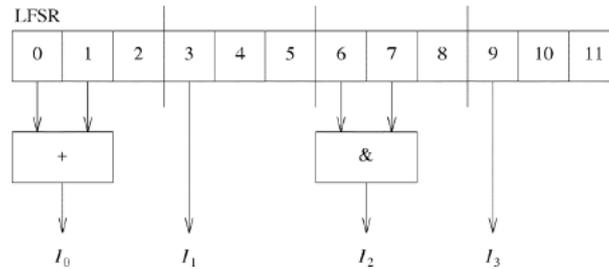


Fig. 2-Input Pattern Generation

The 12-bit LFSR is shown at the top of Fig. 2. Here $d=3$, $\text{mod}=2$ and $n=4$, it is used for tests for input circuits such as S27 standard bench mark circuits. Bits 0, 1 and 2 of the LFSR are used for producing the values of I_0 . Since, an OR gate is used for increasing the probability that will be assigned the value 1. The OR gate is driven by bits 0 and 1 of the LFSR. Bit 2 of the LFSR reduces the dependencies between the values of and the values of I_0 and I_1 . Bits 3, 4, and 5 of the LFSR are used for producing the values of I_1 . Therefore, I_1 is driven directly by bit 3 of the LFSR. Bits 4 and 5 reduce the dependencies between the values of and the values of I_1 and I_2 . Bits 6, 7, and 8 are used for producing the values of I_2 . Since, an AND gate is used for increasing the probability that will be assigned the value 0. The AND gate is driven by bits 6 and 7. Bit 8 of the LFSR reduces the dependencies between the values of I_2 and those of I_3 . Finally, bits 9, 10, and 11 are used for producing the values of I_3 . Therefore, is driven directly by bit 9 of the LFSR. In general, if there are primary inputs with, the implementation illustrated by Fig. 2 requires a 12-bit LFSR, and AND or OR gates with mod inputs.

B.FAULT TESTING

The on-chip test generation hardware described so far has parameters L , mod , sel and d . These parameters determine the primary input sequence, and the tests that will be applied based on it. Keeping L , mod , d and sel constant in order to keep the hardware fixed, there is flexibility only in determining the seed of the LFSR. Different seeds yield different primary input sequences and different tests. Therefore, it is possible to increase the fault coverage by using several different seeds.

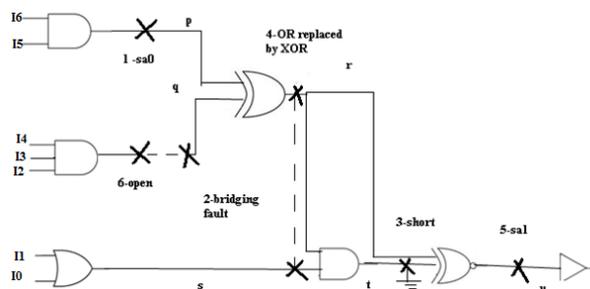


Fig. 3- 7 Input Combinational Circuit

Consider a 7 input combinational circuit in Fig.3. It is associated with 6 faults. Let F be the target fault list and here F contains;

- 1) Stuck at zero fault



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

- 2) Bridging fault
- 3) Short
- 4) OR gate is replaced by XOR gate
- 5) Stuck at one fault
- 6) Open

By applying suitable input patterns, it is possible to detect the faults in each net one by one. As the pseudo random patterns having maximum randomness, then with minimum clock cycles itself the maximum fault coverage can be obtained. Also it leads to reduce the power dissipation. Therefore by every clock cycle the detected fault from the target list will eliminate and at the end, it is possible to make the target fault list empty. The efficiency of detection and elimination is completely based on the LFSR designed and the input patterns developed from it. If the input patterns are not capable to activate all the nets, then it is not possible to detect errors in all nets.

C. SELECTION OF PARAMETERS

Increasing L and d can potentially increase the fault coverage. Increasing L increases the number of available tests, and increasing d reduces the dependencies between the values of the primary inputs. Increasing sel can potentially decrease the fault coverage since it decreases the number of tests that will be applied to the circuit.

Increasing mod is not beneficial always. Based on the number of input pins the selection of mod is determined. The values of the input parameters will determine the effectiveness of test patterns. If the test patterns are designed in a highly efficient manner then it is very easy to determine the target faults in minimum clock cycles. The target fault detection in earlier cycles will reduce the amount of power dissipation in the chip. This is also helpful to protect other circuits present in the IC. Otherwise by increasing power dissipation may cause unintended damage of components.

Based on the values of parameters we can obtain the results as;

- 1) If the fault coverage does not exceed that of ordinary methods for any solution, we report on the following solutions.
 - a) The solution with the highest fault coverage and the lowest number of seeds.
 - b) The solution with the highest fault coverage and the lowest number of applied tests.
- 2) If the fault coverage of at least one solution exceeds that of broadside test, we report on the following solutions.
 - a) The solution with the highest fault coverage.
 - b) The solution with a fault coverage higher than that of broadside test and the lowest number of seeds.
 - c) The solution with a fault coverage higher than that of broadside test and the lowest number of applied tests.

One of these solutions is expected to be the most appropriate for the circuit. In addition to these solutions there are others with intermediate numbers of seeds and applied tests, which are not reported.

IV. RESULT AND DISCUSSION

Based on this the application of different test patterns gives the f_list (target fault list) as shown in the Fig 4. As the number of input pins increases the complexity of testing increases. Output contains the f_list based on binary counter, input patterns due to LFSR also with respect to efficient LFSR by selecting parameters.

Parameters	Efficient LFSR	Ordinary LFSR	Binary Counter
Clock cycles	33	27	16
Fault coverage	90-95%	80%	60%

TABLE III. COMPARISON

Here the experiments carried out on the s27 standard benchmark circuit gives considerable fault detection of 95% in the earlier clock cycles itself.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

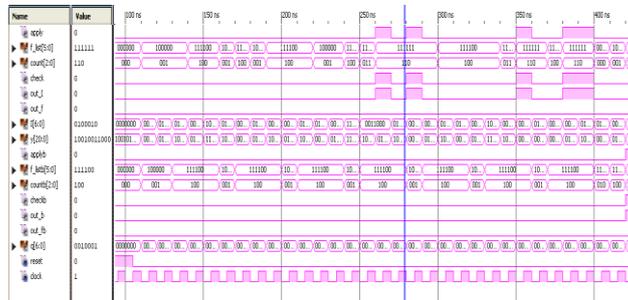


Fig. 4- Simulation Of 7 input Combinational Circuit

Parameters	Efficient LFSR	Ordinary LFSR	Binary Counter
Clock cycles	33	27	16
Fault coverage	90-95%	80%	60%

TABLE III. COMPARISON

The Table III represents the comparison of different input generation methods based on clock cycles and fault coverage. From the table it is clear that the Efficient LFSR design will give maximum fault coverage with minimum input clock cycles therefore it reduces the power consumption as compared to other methods of fault detection.

V. CONCLUDING REMARKS

This paper described a test generation method for functional broadside tests using efficient LFSR with selecting parameters. The hardware was based on the application of primary input sequences starting from a known reachable state, thus using the circuit to produce additional reachable states. Random primary input sequences were modified to avoid repeated synchronization and thus yield varied sets of reachable states. Two-pattern tests were obtained by using pairs of consecutive time units of the primary input sequences. The hardware structure was simple and fixed. Input patterns designed with selecting parameters will give maximum output fault coverage with minimum clock cycles as compared with ordinary binary counter input sequence and LFSR random pattern generation. Also it reduces the power dissipation and increases the reliability of testing combinational circuits.

REFERENCES

- [1] J.Rearick, "Too much delay fault coverage is a bad thing," in Proc. Int. Test Conf., 2001, pp. 624–633.
- [2] V. Gherman, H.-J. Wunderlich, J. Schloeffel, and M. Garbers, "Deterministic logic BIST for transition fault testing," in Proc. Euro. Test Symp., 2006, pp. 123–130.
- [3] I. Pomeranz and S. M. Reddy, "On reset based functional broadside tests," in Proc. Design Autom. Test Euro. Conf., 2010, pp. 1438–1443.
- [4] I. Pomeranz, "Built-in generation of functional broadside tests," presented at the Design Autom. Test Euro. Conf., Grenoble, France, 2011.
- [5] P. H. Bardell, W. H. McAnney, and J. Savir, *Built-In Test for VLSI*.
- [6] Polian and F. Fujiwara, "Functional constraints vs. test compression in scan-based delay testing," in Proc. Design, Autom. Test Euro. Conf., 2006, pp. 1–6.
- [7] B. Konemann, "LFSR-coded test patterns for scan designs," in Proc. Euro. Test Conf., 1991, pp. 237–242.
- [8] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. Piscataway, NJ: IEEE Press, 1995.