ISSN 23499842(Online), Volume 2, issue 1 March 2017
International Journal of Innovative Trends and Emerging Technologies

DETECTING FAULTS AN INTEGRATED CIRCUITS BY USING LFSR BASED TEST GENERATION FROM NONTEST CUBES

J. SUGANTHI[1], P. SARAVANA KUMAR[2]
[1]*PG Scholar, VLSI Design, Department of ECE, Mahabarathi Engineering College, India*
[2]*Assistant professor, Department of ECE, Mahabarathi Engineering College, India*

Abstract
The objective of this project is to carry out hybrid non-test cubes driven seed computation for hard to detect faults. In the existing method, the LFSR pattern out is applied independently to all the target faults and optimal seeds selection is carry out. So it is optimally considered as a hard-to-detect method. Therefore the aim is to remove unnecessary seeds which covered same faults more than once in the detection of faults in integrated circuits by using LFSR based test generation from non-test cubes. In order to support to reduce latency single cycle access method is used with priority bits to isolate hard to detect faults. Finally optimal seeds for 100% fault coverage with optimal number of test patterns have been found.

Index terms: LFSR.

1. Introduction
In the production of integrated circuits, testing is done to identify defective chips. This is very important for shipping high quality products. Testing is also done to diagnose the reason for a chip failure in order to improve the manufacturing process. In system maintenance, testing is done to identify parts that need to be replaced in order to repair a system. Testing a digital circuit involves applying an appropriate set of input patterns to the circuit and checking for the correct outputs. The conventional approach is to use an external tester to perform the test. However, built-in self-test (BIST) techniques have been developed in which some of the tester functions are incorporated on the chip enabling the chip to test itself. BIST provides a number of well-known advantages. It eliminates the need for expensive testers. It provides fast location of failed units in a system because the chips can test themselves concurrently.

And, it allows at-speed testing in which the chip is tested at its normal operating clock rate which is very important for detecting timing faults. Despite all of these advantages, BIST has seen limited use in industry because of its area and performance overhead, increased design time, and lack of BIST design tools. These are problems that this dissertation addresses. The research described in this dissertation is timely because the interest in BIST is growing rapidly. The increasing pin count, operating speed, and complexity of IC's is outstripping the capabilities of external testers. BIST provides solutions to these problems.

The circuit that is being tested is called the circuit-under-test (CUT). There is a test pattern generator which applies test patterns to the CUT and an output response analyzer which checks the outputs. The test pattern generator must generate a set of test patterns that provides a high fault coverage in order to thoroughly test the CUT. Pseudo-random testing is an attractive approach for BIST. A linear feedback shift register (LFSR) can be used to apply pseudo-random patterns to the CUT. An LFSR has a simple structure requiring small area overhead. Moreover, an LFSR can also be used as an output response analyzer thereby serving a dual purpose.

There are limits on the test length, which is the number of pseudo-random patterns that can be applied during BIST. One limit is simply the amount of time that is required to apply the patterns. Another limit is the fault simulation time required to determine the fault coverage. A third limit is heat dissipation for an unpackaged die. Thus, in order for pseudo-random pattern testing to be effective, a high fault coverage must be obtained for an "acceptable" test length. What is considered acceptable depends on the particular test environment. The probability of detecting a fault with a single random pattern is defined as the detection probability for the fault and is given by the number of patterns that detect the fault divided by the total number of inputs patterns, 2n, where n is the number of inputs in the circuit. Unfortunately, many circuits contain faults with very low detection probabilities. Such faults are said to be random-pattern-resistant (r.p.r.) because they are hard to detect with random patterns and therefore limit the fault coverage for pseudo-random testing. A circuit is said to be random pattern testable if it does not contain any r.p.r. faults.

If the fault coverage for pseudo-random BIST is insufficient, then there are two solutions. One is to modify the circuit-under-test to make it random pattern testable, and the other is to modify the test pattern generator so that it generates patterns that detect the r.p.r. faults. Innovative techniques for both of these approaches are described in this dissertation. These techniques enable automated design of pseudo-random BIST implementations that satisfy fault coverage requirements while minimizing area and performance overhead. These techniques have been incorporated in the TOPS (Totally Optimized Synthesis-for-test) tool being developed at the Center for Reliable Computing.

2. Related Work
In BIST LFSR reseeding is a powerful approach for reducing test storage. The presented seed computation scheme by Dhrumeel Bakshi and Michael S. Hsiao [1] provides a way to reduce the test power for LFSR reseeding while still preserving or even improving the compression that is achieved. The block size can be easily adjusted to tradeoff test-power reduction versus hardware overhead. This paper presents a new low-power test-data-compression scheme based on linear feedback shift register (LFSR) reseeding. A drawback of compression schemes based on LFSR reseeding is that the unspecified bits are filled with random values,

which results in a large number of transitions during scan-in, thereby causing high-power dissipation.

A new encoding scheme that can be used in conjunction with any LFSR-reseeding scheme to significantly reduce test power and even further reduce test storage is presented. It accomplishes many goals to reduce the number of specified bits that need to be generated via LFSR reseeding. Experimental results indicate that the proposed method significantly reduces test power and in most cases provides greater test-data compression than LFSR reseeding alone. The main limitation of this method is that this support only for predetermined faults flow not applicable for BIST. The complexity of the system will be very high.

Janusz Rajski and Nagesh Tamarapalli [2] presented an approach called automated synthesis of phase shifters for built-in self-test applications In this paper they presents novel systematic design techniques for the automated register transfer level synthesis of phase shifters and it is used to remove effects of structural dependencies featured by pseudorandom test pattern generators driving parallel scan chains. They proposed a new systematic design technique that can be employed in BIST automation tools to synthesize circuit independent phase shifters at the register transfer level and to explore in a time-efficient manner a large domain of possible realization of pseudorandom pattern generators. The presented technique takes advantage of simple logic simulation of LFSRs by exploiting the principle of the LFSR duality. Using a concept of linear feedback shift register (LFSR) duality this paper relates the logical states of LFSRs and circuits spacing their inputs to each of the output channels. Consequently, the method generates a phase-shifter network satisfying criteria of channel separation and circuit complexity by taking advantage of simple logic simulation of the LFSRs. It is shown that it is possible to synthesize in a time-efficient manner very large and fast phase shifters for built-in self-test applications with guaranteed minimum phase shifts between scan chains, and very low delay and area of virtually one two-way XOR gate/channel. Predetermined number of states & its omission is essential leads more testing time and the addition of phase shifter leads complexity.

In most cases the test storage for LFSR reseeding depends only on the number of specified bits. For each block that is not a don't care block, the hold flag for that block is specified. If the number of specified hold flags becomes larger than the number of the specified test data bits that are reduced by using the proposed encoding scheme, then the encoding scheme would be reducing test power dissipation at the cost of test storage. Here Jinkyu Lee and Nur A. Touba [3] proposed novel encoding scheme provides a way to reduce test power for LFSR reseeding. It acts as a second stage of compression after LFSR reseeding. By employing hold flags, not only is test power reduced, but also test storage can be reduced. The actual drawback of these schemes is that the unspecified bits are filled with random values resulting in a large number of transitions during scan-in thereby causing high power dissipation. This paper presents a new encoding scheme that can be used in conjunction with any LFSR reseeding scheme to significantly reduce test power and even further reduce test storage. The proposed encoding scheme acts as a second stage of compression after LFSR reseeding. It accomplishes two goals. First, it reduces the number of transitions in the scan chains (by filling the unspecified bits in a different manner), and second it reduces the number of specified bits that need to be generated via LFSR reseeding. Experimental results indicate that the proposed method significantly reduces test power and in most cases provides greater test data compression than LFSR reseeding alone. The main disadvantage of this implementation is the patterns that contain lesser transitions will reduce fault coverage.

3. Proposed Method

The basic idea of BIST, in its most simple form, is to design a circuit so that the circuit can test itself and determine whether it is "good" or "bad" (fault-free or faulty, respectively). This typically requires additional circuitry whose functionality must be capable of generating test patterns as well as providing a mechanism to determine if the output responses of the circuit under test (CUT) to the test patterns correspond to that of a fault-free circuit. In all the cases test-per-clock and the test-per-scan schemes are required.

Pseudorandom Built-In Self Test (BIST) generators have long been successfully utilized for the testing of integrated circuits and systems. The arsenal of pseudorandom generators includes Linear Feedback Shift Registers (LFSRs) [4], Cellular Automata and Accumulators accumulating a constant value. However, some circuits contain faults for which large number of random patterns has to be generated before high fault coverage is achieved. Therefore, weighted pseudorandom techniques have been proposed; in weighted pseudorandom techniques inputs are biased by changing the probability of a 0 or a 1 on a given input from 0.5 (for pure pseudorandom tests) to some other value

Weighted random pattern methods that rely on a single weight assignment, usually fail to achieve complete fault coverage using a reasonable number of test patterns, since, although the weights are computed to be suitable for most faults, some faults may require long test sequences to be detected with weight assignments that do not match their activation and propagation requirements. Therefore, multiple weight assignments have been suggested in cases where different faults require biases of the input combinations applied to the circuit, to ensure that a relatively small number of patterns detect all faults.

In this paper, we propose a PRPG for LP BIST [5] applications. The generator primarily aims at reducing the switching activity during scan loading due to its preselected toggling (PRESTO) levels.

Minimum transitions: In the proposed pattern, each generated vector applied to each PRNG output, which can minimize the input transition and reduce test power.

2) Uniqueness of patterns: The proposed sequence does not contain any repeated patterns, and the number of distinct patterns in a sequence can meet the requirement of the target fault coverage for the CUT.

3) Uniform distribution of patterns: The conventional algorithms of modifying the test vectors generated by the LFSR use extra hardware to get more correlated test vectors with a low number of transitions. However, they may reduce

the randomness in the patterns, which may result in lower fault coverage and higher test time.

4) Low hardware overhead consumed by extra TPGs: The linear relations are selected with consecutive vectors or within a pattern, which has the benefit of generating a sequence with a sequential de compressor. Hence, the proposed TPG can be easily implemented by hardware.

A. Generation of Test Set

It is always desirable to obtain other polynomials, which have the same degree Linit and which can generate the segment, and then examine whether any one of them can generate every test cube in the test set. If none can generate the whole test set as shown in figure 1, we want to be able to look for polynomials of degree Linit + 1, capable of generating the segment, to determine if any can generate the test set. This iterative process could go on to compute polynomials of degree Linit + 2, Linit + 3, until we find a polynomial of degree Lm capable of generating the given test set. The difficulty is that the BM algorithm and its extensions for handling don't cares compute only one polynomial degree and only one polynomial of that degree to generate the target sequence. The mechanics of testing, as illustrated in Figure 3 are similar at all levels of testing, including design verification. A set of input stimuli is applied to a circuit and the output response of that circuit is compared to the known good output response, or

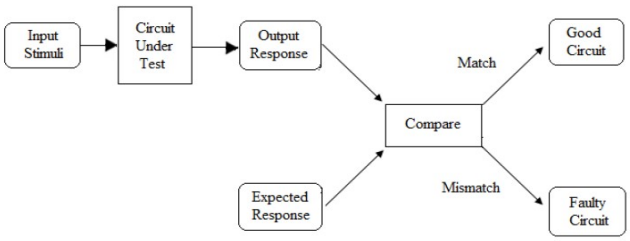

Figure 1: Generation of Test

We need to be able to specify any desired number as the degree of the polynomial that we want for the generation of a target sequence and to be able to obtain multiple polynomials of that degree, if they are more than one, in order to have more candidate polynomials in our pool.

B. **Functionality of Polynomials Degree**

Much higher flexibility in forming low-toggling test patterns can be achieved by deploying a scheme presented. The Berlekamp-Massey algorithm solves the problem of finding an LFSR characteristic polynomial of minimum degree Lmin capable of generating a given fully-specified sequence. A generalization of this problem is the computation of all LFSR characteristic polynomials of a userspecified degree L ≥ Lmin, capable of generating the given sequence. The following theorem, adapted here to Galois Field GF, provides a solution to this problem. This is accomplished by placing AND gates on the control register outputs to allow freezing of all phase shifter inputs. This property can be crucial in SoC designs where only a single scan chain crosses a given core, and its abnormal toggling may cause locally unacceptable heat dissipation that can only be reduced due to temporary hold periods.

$R_0(x) = X_5 + X_4 + X_2 + 1,$
$R_1(x) = X_6 + X_4 + X_3 + X_2 + X_1 + 1.$

The application of Procedure 1 with target degree L = 7, yields four polynomials, which are:

$R_0(x) = X_5 + X_4 + X_2 + 1,$
$R_1(x) = X_6 + X_4 + X_3 + X_2 + X_1 + 1,$
$R_2(x) = X_7 + X_6 + X_5 + 1,$
$R_3(x) = X_7 + X_3 + X_1 + 1.$

As noticed, when computing the polynomials for a given degree L returns a list R which includes all polynomials from degree L_{min} up to the required degree L. However, only the second half of these polynomials have the desired degree L that concerns us here. Therefore, the computation can be limited to the second half of list R.

C. LP Decompressor

The core principle of the decompressor is to disable both weighted logic blocks (V and H) and to deploy deterministic control data instead. In particular, the content of the toggle control register can now be selected in a deterministic manner due to a multiplexer placed in front of the shift register. Furthermore, the Toggle and Hold registers are employed to alternately preset a 4-bit binary down counter, and thus to determine durations of the hold and toggle phases. When this circuit reaches the value of zero, it causes a dedicated signal to go high in order to toggle the T flip-flop. The same signal allows the counter to have the input data kept in the Toggle or Hold register entered as the next state. Both the down counter and the T flip-flop need to be initialized every test pattern. The initial value of the T flip-flop decides whether the decompressor will begin to operate either in the toggle or in the hold mode, while the initial value of the counter, further referred to as an offset, and determines that mode's duration. As can be seen, functionality of the T flip-flops remains the same as that of the LP PRPG but two cases.

First of all, the encoding procedure may completely disable the hold phase (when all hold latches are blocked) by loading the Hold register with an appropriate code, for example, 0000. If detected (No Hold signal), it overrides the output of the T flip-flop by using an additional OR gate. As a result, the entire test pattern is going to be encoded within the toggle mode exclusively. In addition, all hold latches have to be properly initialized.

4. Weighted Pattern Generation

All weighted random and related methods suffer from the following limitation. To produce weights which are different from 0.5, several cells of an LFSR or a shift register are connected to a gate whose output is used to derive the corresponding primary input of the circuit under test; e.g., to produce a weight of 0.25, two cells of the LFSR are connected to an AND gate, whose output drives a primary input of the circuit. When weights are allowed to assume arbitrary values, arbitrary numbers of shift-register cells have to be used to

produce the required input values. Register cells are generally not allowed to be shared between circuits that generate weights for different primary inputs, to avoid correlation between the values different primary inputs assume.

The hardware consists of a counter and XOR gates. In some cases, the hardware cost to achieve complete fault coverage may become high, or the number of test patterns may become extremely large. Work on efficient generation of tests thus varies from pure random tests, which have low hardware requirements but may require large test sequences (or, alternatively, may leave some faults undetected), to application of a deterministic test set, which is short but may have a high hardware cost and may not provide adequate coverage of un modeled faults.

Other approaches that were investigated, and which mix deterministic tests and random tests, include the use of several seeds, or initial LFSR states, to generate random patterns, instead of a single seed conventionally used, and the use of transformations, other than the one performed by the LFSR**,** on the states of the LFSR. A minimal number of weight assignments is searched for, to keep hardware requirements low, and at the same time, the number of tests generated for every weight assignment is limited, to limit the total test length. In contrast to other weighted random methods (e.g., [lo]), a fixed number of random tests (sufficient to detect all target faults) is generated for every weight assignment to avoid the hardware overhead related to the use of different numbers of tests.

Under the approach proposed here, tests are generated using several assignments of weights. In this section, the selection of the weights is described. To derive weight assignments to be used in weighted pseudo-random test pattern generation, we use a known deterministic test set that provides the desired (in our case 100%) coverage of detectable faults. As in earlier work, a weight of **a,** 0 **I a I** 1, assigned to an input x of the circuit under test implies that the probability of input **x** being assigned the value 1 is **a.** The following example illustrates the basic idea behind our method for deriving the weight assignments to be used for weighted pseudorandom test pattern generation.

5. Simulation Results

Model Sim output provides the address of the faults in the form of don't care (z). Decompressor compress the address don't care generate automatically either '1'or'0' values. Quartus II is a software tool produced by Altera for analysis and synthesis of HDL designs, which enables the developer to compile their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

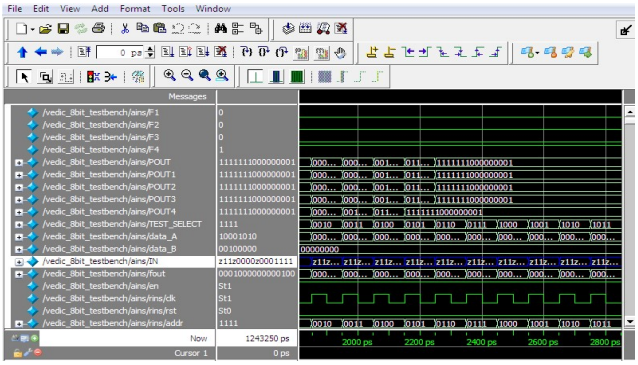

Figure 2: Simulated output

Basically, a design with embedded BIST architecture consists of a test controller, hardware pattern generator, input multiplexer, circuit under test (CUT) which in this project is the IOP and output response compactor. Optionally, a design with BIST capability may includes also the comparator and Read-Only-Memory (ROM). As shown in Figure 3, the test controller is used to control the test pattern and test generation during BIST mode for the synthesis report. Hardware pattern generator functions to generate the input pattern to the CUT.

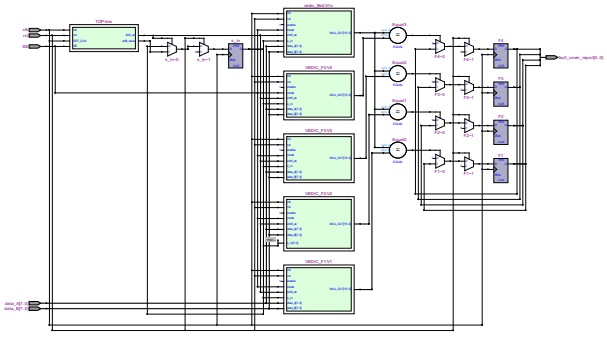

Figure 3: RTL Schematic report

Performance report provides total power used in the circuit .power used in existing and proposed method is varied because power usage is varying in different circuit. From the Table 1, the experimental evaluation shows that the total logic elements in the existing system are 514 and in the proposed system the total logic elements is reduced to 413. On comparing both the existing and proposed system the overall area is reduced.

Type	Area	Total no .of transition	Power
Conventional test compressor	514	8638	64.39mW
Proposed test compressor	413	973	63.38mW

TABLE 1 Trade off analyzes of TEST COMPRESSER with QUARTUS II hardware synthesis using CYCLONE III family (EP3C5F56C6)

6. Conclusion

In the project work carried out dynamic computation based LFSR and its decompresser design for its testing quality metrics and it is verified through modelsim based simulation which can produce pseudorandom test patterns with computation of seeds precisely. The switching activity can also be easily controlled by the pattern generator, so that the resultant test vectors can either yield a desired fault coverage faster than the conventional pseudorandom patterns while still reducing toggling rates down to desired levels, and offer visibly higher coverage numbers if run for comparable test times. To overcome the inefficiency arises to test deep registers and its counterpart of continuous switching from all register single cycle access methodology can be used. And this design is extended into fully functional test data decompressor with the ability to control scan shift-in switching activity through the process of encoding. The efficiency of proposed combine test compression with logic BIST is verified and proved to deliver high quality test.

7. References

[1] Dhrumeel Bakshi and Michael S. Hsiao, "LFSR Seed Computation and Reduction Using SMT-Based Fault-Chaining" in IEEE Design, Automation and test. Dig. Papers, Mar. 2014.

[2] Janusz Rajski, Nagesh Tamarapalli, "Automated Synthesis of Phase Shifters for Built-In Self-Test Applications" IEEE CAD of Integrated circuits and systems., vol. 19, issue10, pp 1175-1188, Aug. 2002.

[3] Jinkyu Lee and Nur A. Touba, "Low Power Test Data Compression Based on LFSR Reseeding" IEEE Trans. VLSI in computers and processors., pp. 1063–6404, Nov 2004.

[4] S. Wang and S. Gupta, "DS-LFSR: A BIST TPG for low switching activity," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 21, no. 7, pp. 842–851, Jul. 2002.

[5] D. Gizopoulos, N. Krantitis, A. Paschalis, M. Psarakis, and Y. Zorian, "Low power/energy BIST scheme for datapaths," in Proc. 18th IEEE VLSI Test Symp., Apr.–May 2000, pp. 23–28.

.