International Journal of Electronics and
Communication Engineering (IJECE)
ISSN(P): 2278-9901; ISSN(E): 2278-991X
Vol. 4, Issue 2, Mar 2015, 45-54
© IASET

International Academy of Science, Engineering and Technology
IASET
Connecting Researchers; Nurturing Innovations

POWER EFFICIENT TEST PATTERN GENERATION SCHEME FOR BUILT-IN-SELF-TEST USING LFSR

RAMPRASAD EDUBILLI

B.E, VLSI Engineer, Young Minds Technology Pvt Ltd, Andhra Pradesh, India

ABSTRACT

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. These consist of the input vectors and the respective responses. They check for perfect operation of a verified design by testing the internal chip nodes. the Functional tests cover a very high percentage of modeled faults in logic circuits and their generation is the main topic of this paper. These functional vectors are generated from the Test pattern generation scheme especially using LFSR due to internal architecture of LFSR Scheme repeated patterns are generated using the proposed scheme these test patterns are avoided and circuits are efficiently verified without any power losses. Any vectors applied are understood to be functional fault coverage vectors applied during manufacturing test. If the patterns of the input test vector results a fault simulation, then circuit test is going to fail. This paper shows the on chip test Generation for a bench mark circuit using simple fixed hardware design with small number of parameters altered in the design for the generation of no of patterns.

KEYWORDS: Power Efficient Test Pattern Generation Scheme

INTRODUCTION

Over testing due to the application of two-patterns can-based tests was described in [1]–[3]. Over testing is related to the detection of delay faults under non-functional operation conditions. One of the reasons for these non-functional operation conditions is the following. When we use arbitrary state as a scan-in state, and a two-pattern test can take the circuit through state-transitions that cannot occur during functional process. As a result, slow paths that cannot be sensitized during functional operation may cause the circuit to fail [1]. In addition, current demands that are higher than those possible during functional operation may cause voltage drops that will slow the circuit and cause it to fail [2], [3]. In both cases, the circuit will operate correctly during functional operation.

Functional broadside tests [4] ensure that the scan-in state is a state that the circuit can enter during functional operation. As broadside tests [5], they operate the circuit in functional mode for two clock cycles after an initial state is scanned in. This results in the relevance of a two-pattern test.

Since the scan-in state is a reachable state, now the two-pattern test takes the circuit throughout state-transitions that are guaranteed to be possible during functional operation. The Delay faults that are detected by the test can also affect functional operation, and the current difficulty do not exceed those possible during functional operation. This alleviates the type of over testing described in [1]–[3]. In addition, the power dissipation during fast functional clock cycles of functional broadside tests does not exceed that possible during functional operation.

Test generation procedures for functional and pseudo-functional scan-based tests were described in [4] and [6]–[13]. The procedures generate test sets offline for application from an peripheral tester. Functional scan-based tests use only reachable states as scan-in states. These Pseudo-functional scan-based tests use functional constraints to avoid unreachable states that are captured by the constraints.

This work considers the on-chip (or built-in) generation of functional broadside tests. On-chip test generation diminish the test data volume and facilitates at-speed test application. We know On-chip test generation methods for delay faults, such as the ones described in [14] and [15], do not impose any constraints on the states used as scan-in states. Experimental results indicate that an arbitrary state used as a scan-in state is unlikely to be a reachable state [4]. The on-chip test generation method from [16] applies pseudo-functional scan-based tests. Such tests are not adequate for avoiding unreachable states as scan-in states. The on-chip test generation process explained in this work guarantees that only reachable states will be used.

It should be noted that the delay fault coverage achievable using functional broadside tests is, in general, lower than that achievable using arbitrary broadside tests as in [14], [15] or pseudo-functional broadside tests as in [16]. This is due to the fact that functional broadside tests avoid unreachable scan-instates, these are tolerable by the methods described in [14]–[16]. However, the tests that are needed for achieving this higher fault coverage are also ones that can cause over testing. More power will be dissipate than possible during functional operation.

Only functional broadside tests are considered in this work. Under the anticipated on-chip test generation method, the same circuit is used for generating reachable states during test application.

This alleviates the need to compute reachable states or functional constraints by an offline process as in [4], [6]–[13] and [16]. The underlying observation is related to one of the methods used in [4] for offline test generation, and is the following.

If a primary input sequence A is applied in functional mode starting from a reachable state, all the states were traversed under a are reachable states. Some of these states can be used as the initial state for the application of a functional broadside test. By generating a on-chip and let it takes the circuit through a varied set of reachable states, the on-chip test generation progression is able to achieve high transition fault coverage using functional broadside tests. It should be noted that, for the detection of a set of faults F, at most |F| different reachable states are required. This number is typically only a small fraction of the number of all the reachable states of the circuit. so, the prime input sequence A does not need to take the circuit through all its reachable states, but only through a adequately large number relative to |F|, in order to be effective for the detection of target faults.

The hardware used in this paper for generating the primary input sequence A consists of a linear-feedback shift-register (LFSR) as a random source [17], and of a small number of gates (atmost six gates are needed for every one of the benchmark circuits considered). The gates used for altering the random sequence in order to avoid cases where the sequence takes the circuit into the same or similar reachable st [18]. 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 end result is a simple and fixed hardware structure, which is customized to a given circuit only through the following parameters.

- The number of LFSR bits.
- The span of the primary input sequence.

- The explicit gates used for modifying the LFSR sequence into the sequence.
- The gate used for selecting the functional broadside tests that will be applied to the circuit based on.
- Seeds for the LFSR in array to generate several primary input sequences and several subsets of tests.

The on-chip test generation hardware is based on the one described in [19]. It differs from it in the following ways.

BLOCK DIAGRAM OF LFSR RESEEDING SCHEME

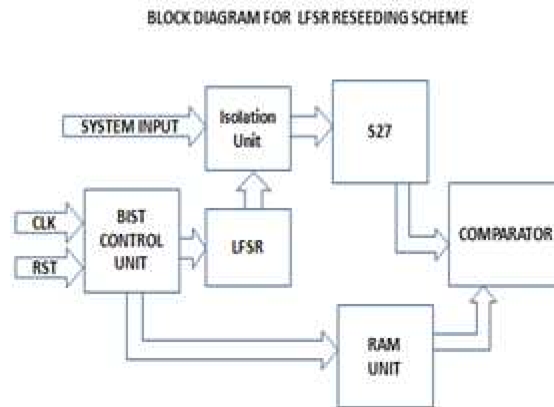


Figure 1: SIMPLE BIST Circuitry

Linear feedback shift register (LSFR) is a shift register whose input bit is a linear function of its previous state and The only linear function of single bits is XOR, thus it is a shiftregister whose input bit is driven by the exclusive-or (XOR)of some bits of the overall shift register value flops. The primary value of the LFSR is called the seed, and the operation of the register is known, the stream of values produced by the register is completely determined byts current (or previous) state.

The seed is used to generate a test pattern and their corresponding test cube. The LFSR length, r, is at leastsmax+20 where smax is the maximum number of specified bitsin any test cube. The r-bit LFSR is initialized with a starting-bit seed. This initial seed is used to generate the first testcube by running the LFSR for m clock cycles (where m is the scan length) to fill the scan chains.

BROADSIDE TESTS IN PARTIAL-SCAN CIRCUITS

In a full-scan circuit, a broadside test starts by scanning in a state denoted by two primary input vectors, denoted by and are then applied in functional mode. The final state observed at the end of the test is scanned out. The test can be partitioned into two patterns, given to one functional clock cycle, and applied in a consecutive functional clock cycle. The application of is done under a slow clock to allow signal transitions in the circuit to stay. The application of is finished under a fast clock in order to capture delayed signal-transitions. Faults are found by observing the primary output vector obtained in response to and when the final state is scanned out. In a full scan circuit, the scan-in state is a fully-precise state. After scanning in all the state variables of the circuit are assigned known values. However, the values of all the state variables are observed during the scan-out operation at the end of the test. For representing we consider a circuit with two primary inputs and five state variables, which are denoted by assume that and are scanned, and that are un scanned. A possible scan-in state is 000xx, where x stands for an unspecified value. Now In a broadside test for this circuit, may be

a partially-specified state as well. For an example, suppose that we obtain 1x01x. Let obtain two-pattern test000xx 00, 1x01x 11. And with partially- specified patterns, it is doubted to be possible to activate certain faults. In addition, faults whose effects are propagated by the second pattern to or will not be detected by the scan-out operation at the end of the test. Hence, it is essential to consider broadside tests with more than two primary input vectors. The primary input vectors are then driven to functional mode. The state of the test, where is the next-state obtained when the present-state is and the primary input vector is There is one time unit where such that is applied under a fast clock in order to capture delayed signal-transitions. Application of for such that is done under a slow clock to allow signal-transitions in the circuit to settle. Under the slow clock the circuit operates as a fault free circuit.

PESUDORANDOM TEST GENERATION

The three primary goals were:

- to develop a battery of statistical tests to detect non randomness in binary sequences constructed using random number generators and pseudorandom number generators utilized in cryptographic applications,
- to produce documentation and implementation of these tests using a software, and
- to provide guidance in the use and application of these tests. Pseudorandom- generate patterns that seems to be random but are in fact deterministic (repeatable). Linear Feedback Shift Register (LFSR) Weighted pseudo-random test generation Adaptive pseudo-random test generation

Algorithmic Test Generation

List initial inputs controlling position where a fault should be detected.

Finding initial input conditions to activate a fault and to sensitize the primary outputs such that the fault can be observed.

Linear Feedback Shift Registers (LFSRs)

Proficient design for Test Pattern Generators & Output Response Analyzers (also used in CRC) FFs plus a few XOR gates better than counter

- Fewer gates
- Higher clock frequency
- Two types of LFSRs External, Internal Feedback
- Higher clock frequency

The periodic sequence generated by an LFSR must start in a non-zero state, The maximum length of an LFSR sequence is 2n -1 does not generate all 0s pattern (gets stuck in that state)The characteristic polynomial of an LFSR generating maximum-length sequence is a primitive polynomial A maximum-length sequence is pseudo-random: number of 1s =number of 0s + 1 same number of runs of consecutive 0s and 1s 1/2 of the runs have length1 1/4 of the runs have length 2 (as long as fractions result in integral numbers of runs).

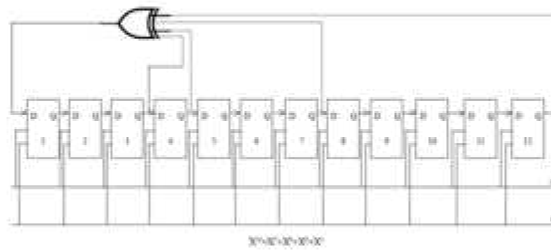


Figure 1.5: LFSR 12bit Circuit

OUTPUT OF LFSR

u	LFSR(u)
0	101 011 100 100
1	010 101 110 010
2	001 010 111 001
3	100 101 011 100
4	010 010 101 110
5	001 001 010 111
6	100 100 101 011
7	110 010 010 101
8	111 001 001 010
9	011 100 100 101
10	101 110 010 010
11	010 111 001 010
12	001 011 100 101
13	100 101 110 010
14	010 010 111 001
15	101 001 011 100

SEQUENTIAL BENCHMARK CIRCUIT s27

- Logic Gates are taken at initial input grouping in (2^4=16).
- S27 circuit have three scan circuits (f/f „s) and then its scan inputs are 2^3=8.
- S_a_0 fault generates at logic gate (a3).
- Scan-in-state input are s0, s1, s2 and these are scanned out denoted as s.
- LFSR register values initially at "1010". Circuit considered above, let 000xx 00, 1x01x01,0x1xx 01, 11x00 00, 10111 10. We note that is unspecified in the two un scanned state variables. This interprets that can be used as a scan-in state instead of and the first two patterns of the test can be neglected. The resulting test would be 0x1x 01. In the scan-in state of this test, we can specify the value of arbitrarily. Let it is specified to 1, and suppose that this causes the value of in to be specified to 0. The result we obtain 011xx 01, 11000 00, 10111 10. The final three states observed are 0x1xx, 11x00 and 10111. With more specified values under, more faults may be detected. Usually, to obtain a shorter test from a given test of length, we consider we search for the highest time unit such that has unspecified values on all the un scanned since is selected such that it is unspecified on the un scanned

state variables. All the tests will be of length two considering For a full-scan circuit,. This is due to the fact that all the states are fully-specified.

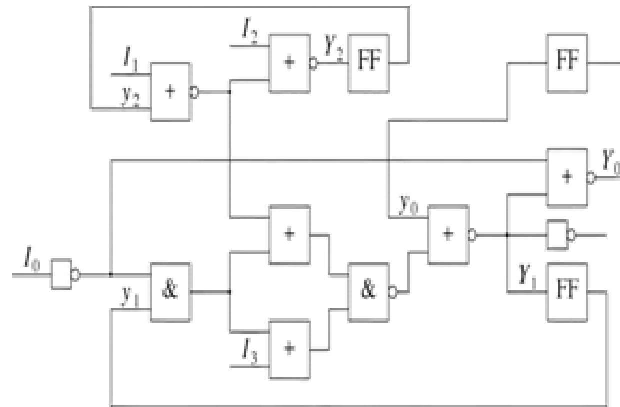


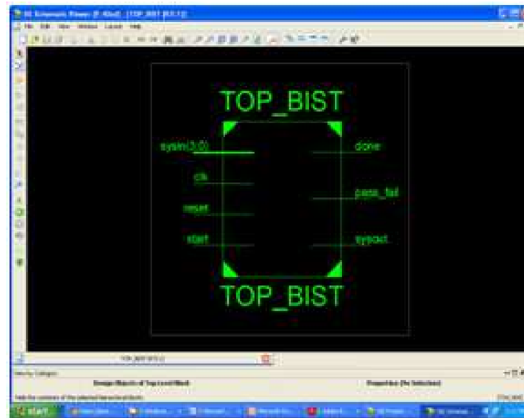
Figure 2.1: s27Sequential Circuit

- S27 bench mark circuit is a standard sequential circuit.
- Here we are used s27 bench mark circuit for as a testing circuit.
- Driving test vectors as input to the s27 bench mark sequential circuit.
- I0, I1, I2, I3 are the input of this circuit.

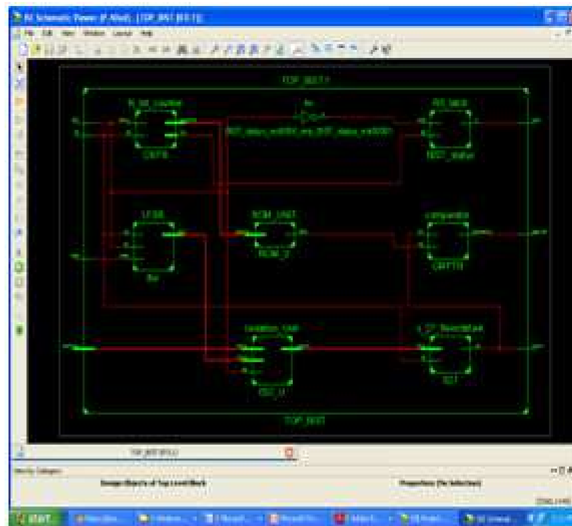
OUTPUT OF S27 BENCHMARK

U	Lfsr(U)	I Value	O/P of S27
0	101 011 100 100	1 0 0 1	X
1	010 101 110 010	1 1 1 0	X
2	001 010 111 001	0 0 1 0	X
3	100 101 011 100	1 1 0 1	1
4	010 010 101 110	1 0 0 1	1
5	001 001 010 111	0 0 0 1	1
6	100 100 101 011	1 1 0 0	0
7	110 010 010 101	1 0 0 1	0
8	111 001 001 010	1 0 0 0	0
9	011 100 100 101	1 1 0 1	0
10	101 110 010 010	1 1 0 0	0
11	010 111 001 001	1 1 0 0	0
12	101 011 100 100	1 0 0 1	0
13	010 101 110 010	1 1 1 0	0
14	001 010 111 001	0 0 1 0	0
15	100 101 011 100	1 1 0 1	1

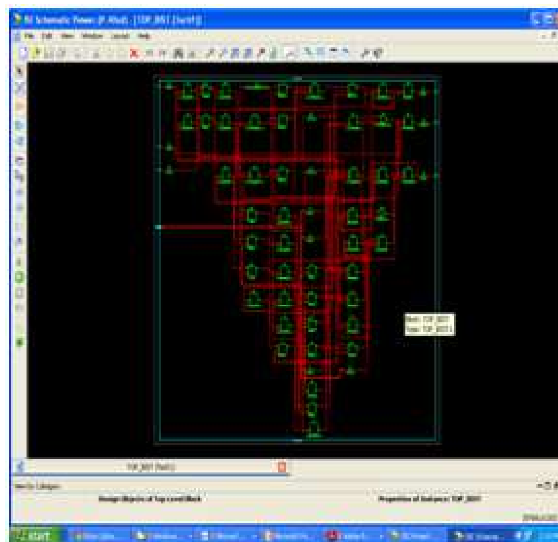
RESULT ANALYIS



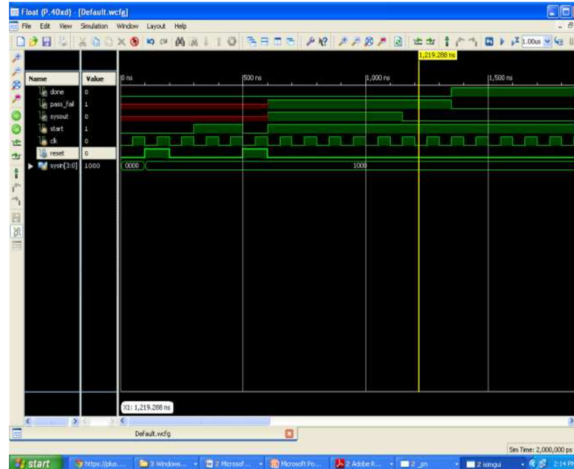
Block Diagram



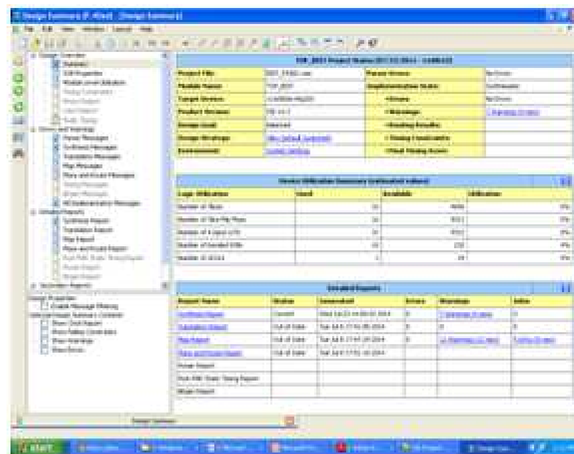
RTL Schematic



Technology Schematic Diagram



Design summary



Output Waveform

CONCLUSIONS

Delay problems are more concern in present Semiconductor industries. To test for such delay-causing defects, scan-based transition fault testing techniques are being implemented, reduced test pattern scheme are implemented and hence test data volume is reduced and circuits are effectively tested and facilitates at-speed test application.

REFERENCES

1. J. Rearick, “Too much delay fault coverage is a bad thing,” in Proc. Int. Test Conf., Oct. 2001, pp. 624–633.
2. J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreeprakash, and M. Hachinger, “A case study of IR-drop in structuredat-speed testing,” in Proc. Int. TestConf., 2003, pp. 1098–1104.
3. Pomeranz, “On the generation of scanbased test sets with reachablestates for testing under functional operation conditions,” in Proc. Des. Autom. Conf., Jun. 2004, pp. 928–933.
4. Y.-C Lin, F. Lu, K. Yang, and K.-TCheng, “Constraint extraction forpseudofunctional scan-based delay testing,” in Proc. Asia South Pacific Des. Autom. Conf., Jan.2005, pp. 166–171.

5. Z. Zhang, S. M. Reddy, and I. Pomeranz, "On generating pseudo-functionaldelay fault tests for scan designs," in Proc. Int. Symp. Defect Fault Toler. VLSI Syst., Oct.2005, pp. 398–405.
6. Pomeranz and S. M. Reddy,"Generation of functional broadside tests fortransition faults," IEEE Trans. Comput.-AidedDes. Integr. Circuits Syst., vol. 25, no. 10, pp.2207–2218, Oct.
7. Y.-C. Lin, F. Lu, K. Yang, and K.-T. Cheng, "Constraint extraction for pseudo-functional scan-based delay testing," in Proc. Asia South Pacific Design Autom. Conf., 2005, pp. 166–171.
8. Z. Zhang, S. M. Reddy, and I. Pomeranz, "On generating pseudo-functional delay fault tests for scan designs," in Proc. Int. Symp. Defect Fault Toler. VLSI Syst., 2005, pp. 398–405.
9. I. 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.
10. M. Syal et al., "Astudy of implication based pseudo functional testing," in Proc. Int. Test Conf., 2006, pp. 1–10.
11. A. Jas, Y.-S. Chan, andY.-S. Chang, "An approach tominimizing functional constraints," in Proc. Defect Fault Toler. VLSI Syst., 2006, pp. 215–226.
12. H. Lee, I. Pomeranz, and S. M. Reddy, "On complete functional broadside tests for transition faults," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., pp. 583–587, 2008.
13. I. Pomeranz and S. M. Reddy, "On reset based functional broadside tests," in Proc. Design Autom. Test Euro. Conf., 2010, pp. 1438–1443.
14. H. Lee, I. Pomeranz, and S. M. Reddy, "Scan BIST targeting transition faults using a Markov source," in Proc. Int. Symp. Quality Electron. Design, 2004, pp. 497–502.
15. 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.

AUTHOR'S DETAILS

Ramprasad Edubilli received the B.E. degreein electrical and electronics engineering from AndhraUniversity, Visakhapatnam, India, in 2012.

He is currently working as a VLSI engineer in Youngminds Technology Solutions pvt ltd. Tirupati

His current research interests include Testing and Front end HDL programming in VLSI designs.