DESIGN OF LOW ENERGY LFSR BASED GENERATION OF MULTICYCLE TESTS

[1]MAROJU SATYA ANUSHA, [2] K VIJAY KUMAR
[1]M.Tech Student, Dept of ECE, G.V.R & S College of Engineering & Technology, A.P, India
[2]Associate professor, Dept of ECE, G.V.R & S College of Engineering & Technology, A.P, India

ABSTRACT: In a random testing environment, a significant amount of energy is wasted in the LFSR and in the CUT by useless patterns that do not contribute to fault dropping. Another major source of energy drainage is the loss due to random switching activity in the CUT and in the scan path between applications of two successive vectors. In this work, a new built-in self-test (BIST) scheme for scan-based circuits is proposed for reducing such energy consumption. A mapping logic is designed which modifies the state transitions of the LFSR such that only the useful vectors are generated according to a desired sequence. Further, it reduces test application time without affecting fault coverage. Experimental results on benchmark circuits reveal a significant amount of energy savings in the LFSR during random testing.

KEY WORDS: Linear-feedback shift register (LFSR)-based test generation, multi-cycle tests, test compaction, test data compression.

I.INTRODUCTION

With the emergence of mobile computing and communication devices, design of low-energy VLSI systems has become a major concern in circuit synthesis. A significant component of the energy consumed in CMOS circuits is caused by the total amount of switching activity (SA) at various circuit nodes during operation. The energy dissipated at a circuit node is proportional to the total number of $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions the logic signals undergo at that node multiplied by its capacitance (which depends on its fan-out and its transistor implementation). Energy consumption in an IC may be significantly higher during testing due to increased SA than that needed during normal (system) mode, which can cause excessive heating and degrade circuit reliability. The average-power optimization help extend the battery life in mobile applications.

The generation of multi cycle tests for test compaction becomes more complex when test data compression is used. In one of the commonly used test data compression methods, a test is compressed into a seed for a linear-feedback shift register (LFSR). The on-chip decompression logic uses the LFSR to apply the test to the circuit. A seed is typically computed based on an incompletely specified test cube by solving a set of linear equations that relate the bits of the seed with the specified values of the test cube. With this process, optimizing a multicycle test <pi, vi, li> to increase the number of faults it detects requires a seed to be recomputed after every step that modifies the test, and some modifications of the test cannot be accepted because a seed does not exist for the modified test.

Motivated by these observations, the goal of this paper is to develop a procedure for computing seeds for LFSR-based generation of multicycle tests that are effective for test compaction. To avoid sequential test generation, the procedure uses a single-cycle test set similar, and optimizes the multicycle tests to increase the numbers of faults they detect. In contrast, the procedure optimizes the compressed multicycle tests in order to avoid producing tests for which seeds do not exist. The linear feedback shift register (LFSR) is most frequently used as a test pattern generator (TPG) in low area overhead built in self-test. BIST procedure can adequately reduce the adversity and intricacy of VLSI testing, by considering this actuality that an LFSR can be made with slight area overhead and decreases switching activity. In typical BIST architecture, the LFSR is frequently used

in test pattern generation and output analyser. The leading fault of this architecture is that pseudorandom Patterns generated by LFSR lead to extremely high switching activities in circuit-under-test (CUT) which can motive redundant power dissipation. They can also spoilage the circuit and reduce product yield and period. BIST is a design-for-testability approach that places the testing behaviour physically with the circuit under test (CUT). LFSR are the sequential logic circuits used to create pseudorandom binary sequences (PRBS) s.

To achieve the goal of producing compressed multicycle tests that are effective for test compaction, the procedure described in this paper optimizes the seed si, the primary input vector vi, and the number of functional clock cycles li together to increase the number of faults that the test detects. By considering the seed si directly, the procedure optimizes the scan-in state pi, and avoids modifications of pi for which a seed does not exist. Moreover, the single-cycle test set that the procedure uses as guidance does not need to be compressed. To accommodate this case, the procedure initializes the seed si randomly, and not based on the scan-in state qi of a single-cycle test. It is thus possible to use a compact single-cycle test set that is not constrained by the LFSR.

The possibility of optimizing a seed si was used to modify seeds that produce fault detection tests into seeds that produce diagnostic tests. The modification of a seed si is implemented by complementing bits of si one by one, and recomposing the test ti that the LFSR produces. A bit complementation is accepted when ti satisfies certain objectives (in these objectives are related to the generation of diagnostic tests). In the procedure described in this paper, bits of si and vi, as well as the value of li, are modified together in order to produce an effective multicycle test. The target faults in this paper are single stuck-at faults. The procedure is developed assuming that an LFSR is given. This paper also describes a modified binary search process for selecting an LFSR out of a given set of available LFSRs.

II. RELATED WORK

As the complexity of modern chips increases, external testing with ATE becomes extremely expensive. Instead, built-in self-test (BIST) is becoming more common in the testing of digital VLSI circuits since it overcomes the problems of external testing using ATE. BIST test patterns are not generated externally as in case of ATE; instead they are generated internally using some parts of the circuit, also the responses are analysed using other parts of the circuit. When the circuit is in test mode, test patterns generators (TPGs) generate patterns that are applied to the CUT, while the signature analyser (SA) evaluates the CUT responses. One of the most common TPGs for exhaustive, pseudo-exhaustive, and pseudorandom TPG is the linear feedback shift register (LFSR). LFSRs are used as TPGs for BIST circuits because, with little overhead in hardware area, a normal register can be configured to work as a test generator, and with an appropriate choice of the location of the XOR gates, the LFSR can generate all possible output test vectors (with the exception of the 0s-vector, since this will lock the LFSR). The pseudorandom properties of LFSRs lead to high fault coverage when a set of test vectors is applied to the CUT compared with the fault coverage obtained using normal counters as TPGs. Also LFSRs can be configured to act as signature analyzers for the responses obtained from the CUT. Despite their simple appearance, LFSRs are based on complex mathematical theory that helps explain their behavior as TPGs or SAs.

The characteristic polynomial of an LFSR determines which flip-flop locations of the

LFSR feed the inputs of the XOR gates in the feedback path. If the characteristic polynomial of an LFSR is primitive, then the LFSR will generate the maximum length non-repeating sequence, which is called an m-sequence. LFSRs can be divided into two main categories: external-XOR LFSR (simply external LFSR) and internal-XOR LFSR (simply internal LFSR). These are distinguished by the way in which XOR gates are inserted into the system. In an external LFSR the XORs appear only in the feedback, while in the internal LFSR the XORs appear between flip- flops

When the LFSR is used to generate test patterns for full scan-chain sequential circuits, one of its flip-flop outputs is connected with the scan-chain input. In this case the LFSR will be considered as a one-dimensional TPG. The main problem of this configuration is the long time needed to scan-in a test vector which is equivalent to the number of flip-flops in the scan-chain. In order to speed-up the scanning of test vectors (i.e. reducing test application time), the flip flops in the circuit can be divided into groups, and each group forms a separate scan-chain. This approach is called multiple scan-chains. In this case a two- dimensional TPG should be used to scan-in test vectors in the multiple scan-chains in parallel. The LFSR can be used for this purpose, where different flip-flops outputs can be connected with the different scan-chain inputs and the outputs of the scan-chains are connected with a multiple input signature register (MISR)

III. EXISTED SYSTEM

Between the scan-in and scan-out operations of a test, a single cycle test has a single functional clock cycle, while a multi cycle test has one or more functional clock cycles. Multicycle tests were considered. Their effectiveness for test compaction was demonstrated in and results from the following observations. During a functional clock cycle of a test, the combinational logic of the circuit receives an input pattern that can be used for detecting faults. A larger number of functional clock cycles allow more faults to be detected. As a result, a multicycle test may detect more faults than a single-cycle test. With more detected faults for every test, the number of tests is reduced. This reduces the number of scan operations that a test set requires. With fewer scan operations, the test data volume and test application time are reduced. The fact that each test consists of more functional clock cycles has a negligible effect on the test application time when the number of functional clock cycles is bounded. The test data volume is independent of the number of functional clock cycles if the primary input vector is kept constant during a test. This is a common requirement to address tester limitations that prevent the primary input vector from being changed during a test.

For the discussion in this paper a multicycle test is denoted by ti = <pi, vi, li>, where pi is the scan-in state, vi is the primary input vector, and li is the number of functional clock cycles. After pi is scanned-in, the primary input vector vi is applied for li functional clock cycles. The test ends with a scan-out operation. The generation of multicycle tests for test compaction requires the number of functional clock cycles li in a test to be determined in addition to its scan-in state pi and its primary input vector vi. To simplify the test generation procedure, and avoid the need for sequential test generation, it is possible to use a single-cycle test set to guide the generation of a multicycle test set. Thus, if <qi, ui, 1> is a single-cycle test, it is possible to define a multicycle test <pi, vi, li> by using pi = qi and vi = ui.

This section describes the computation of a compressed multicycle test set based on a single-cycle test set W1. The test set W1 is not producible by an LFSR with a limited

number of bits. With a bound LMAX on the number of functional clock cycles in a test, the multicycle test set is denoted by TLMAX. The procedure initially assigns TLMAX = ∅, and includes in a set F all the target faults that are detected by W1. The procedure constructs TLMAX by performing LMAX iterations over the tests of W1. The iterations differ in the initial target L for the number of functional clock cycles in a test. The procedure considers L = LMAX, LMAX − 1, 1 in order to achieve the following goals. By considering higher values of L earlier, the procedure gives a precedence to the computation of multicycle tests with larger numbers of clock cycles. Such tests allow more target faults to be detected, thus contributing to test compaction. By considering all the values of L down to 1, the procedure ensures that single-cycle tests will be included in TLMAX if this is necessary for detecting some of the faults.

For every value of L, the procedure attempts to compute a test ti based on every test wi = <qi, ui, 1> ∈ W1. If a test ti is computed, and dbest _= 0, the procedure adds ti to TLMAX, and simulates F under ti with fault dropping. After considering all the values of L, the procedure performs forward-looking reverse order fault simulation in order to remove unnecessary tests from TLMAX. Several features of the procedure are illustrated by the following example. The example uses a 24-bit primitive LFSR for ITC-99 benchmark b07. The test set W1 consists of 52 tests, and it achieves a single stuck-at fault coverage (the remaining faults are undetectable). The procedure is applied with LMAX = 8. The procedure terminates after considering L = 5 since all the target faults are detected. Column i shows the index of the test wi ∈ W1 that the procedure uses. Column spec shows the number of specified values in the scan-in state qi of wi. Column f.c. shows the single stuck-at fault coverage after the procedure adds a test based on wi to T8. A multicycle test ti that the procedure derives with a given value of L may have li _= L. The initial value of li is L, but the procedure may select a different value.

IV. PROPOSED SSYTEM

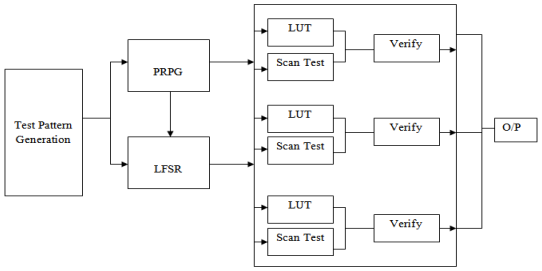


FIG. 1: PROPOSED SYSTEM

The above figure (1) shows the architecure of proposed system. The devices used in this system are test pattern generator, PRPG, LFSR, LUT and Scan-Test. We assume a test-per-scan BIST scheme as in the STUMPS architecture. A modulo-m bit counter keeps track of the number of scan shifts, where *m* is the length of the longest scan path. Since the number of useful patterns is known to be a very small fraction of all generated patterns, a significant amount of energy is still wasted in the LFSR while cycling through these useless patterns even though they are blocked at the inputs to the CUT. The LSB of the LFSR is shifted serially into the scan path generating a test sequence S. Some component of PRPG is intrinsic (invariant over a full test session), and the rest is variable. Hence, PRPG can be represented as a directed complete graph called activity graph, where each node represents a test vector, and the directed edge (eij) represents application of the ordered test pair (ti, tj). The weight w (eij) on the edge eij denotes the variable component of PRPG corresponding to the ordered pair of tests (ti, tj). The intrinsic component being independent of test ordering is represented as a node weight and may be ignored as far as the optimal ordering is concerned. The edge weights are represented as an asymmetric cost matrix, as the variable component of PRPG strongly depends on ordering of test

pairs. Thus, for the test sequence *S* (t1→ t2 → t3 → t4 → t5), the variable component of switching activity is 37. Now, if t3 is found to be a useless test pattern, it along with all incident edges, can be deleted. An optimal ordering of test vectors that minimizes the energy consumption is a min-cost Hamiltonian path: *S* (t1→t2 → t5 → t4), the path cost being equal to 23.

Thus, in the new sequence *S,* for the ordered pair (t1 → t2), no action is required, as t2 is generated by the LFSR as a natural successor of t1. So, for s9. we set the Y-outputs of the mapping logic to don't cares (*d*), and the control line C to 0. However, we need an additional transition from s14 (end-state of t2) to s8 (start-state of t5), and similarly from s11 (end-state of t5) to s6 (start-state of t4). For these combinations, the Y outputs are determined by the corresponding start states, and C is set to 1. For all other remaining combinations, all outputs are don't cares. These transitions generate the useful test patterns in a desired sequence, and prevent the LFSR to cycle through the states that generate useless patterns (in this example, test t3). Further, the output M of the modulo-m bit counter assumes 1 only when scan path (whose length is m) is filled, i.e., at the end-states of the test vectors. Thus, in order to generate the sequence *S*, we need to skip the natural next state of the LFSR and jump to the start state of the desired next test pattern. These state skipping transitions are shown with dotted lines.

In general, the mapping logic can be described as follows: given a seed, let *S* denote the original test sequence generated by the LFSR, and *S* = {t'1, t'2, … t'i, t'i +1, …} denote the optimally ordered reduced test sequence consisting of useful vectors only. Let yi denote the output of the i-th flip-flop of the LFSR, and Yi denote the output of the mapping logic feeding the i-th flip-flop through a MUX. *Case(i)* is applicable if the consecutive test pair (t'i, t'i +1) of *S* appears in consecutive order in the original sequence *S* as well; otherwise, *case (ii)* is applicable. Thus, the next-state of the LFSR follows the transition diagram of the original LFSR when either C = 0, or M = 0, and is determined by the outputs of the mapping logic if and only if CM = 1. Since these additional transitions emanate only from the end order to prevent the SA from occurring in ML for every scan shift cycle, an enable signal E controlled by M is used. Thus, the y-inputs become visible to ML if and only if M = 1. The test session terminates when the end-state of the last useful pattern in *S* is reached. Determination of optimal reordering of test patterns is equivalent to solving a traveling salesman problem (TSP), which being NP-hard, needs heuristic techniques for quick solution.

V. RESULTS

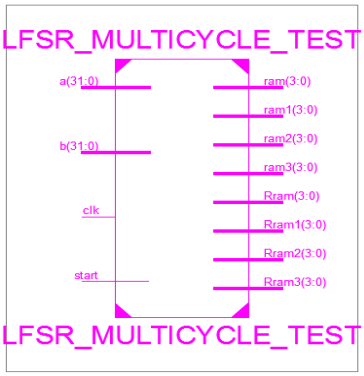


FIG. 2: RTL SCHEMATIC

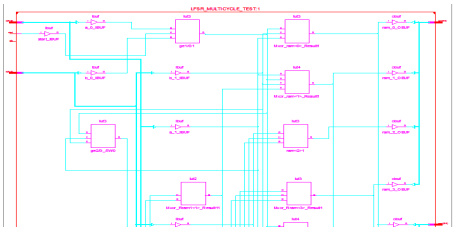


FIG. 3: TECHNOLOGY SCHEMATIC

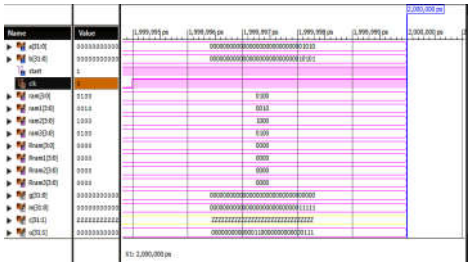
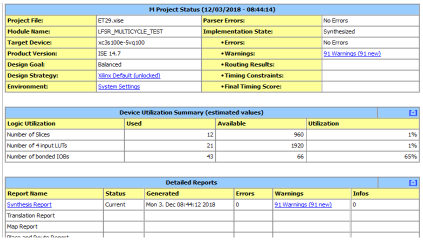


FIG. 4: OUTPUT WAVEFORM



H Project Status (12/03/2018 - 08:44:14)			
Project File:	ET29.xise	Parser Errors:	No Errors
Module Name:	LFSR_MULTICYCLE_TEST	Implementation State:	Synthesized
Target Device:	xc3s100e-5vq100	• Errors:	No Errors
Product Version:	ISE 14.7	• Warnings:	91 Warnings (91 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	12	960	1%
Number of 4 input LUTs	21	1920	1%
Number of bonded IOBs	43	66	65%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Mon 3. Dec 08:44:12 2018	0	91 Warnings (91 new)	0
Translation Report					
Map Report					

FIG. 5: REPORT

VI. CONCLUSION

This paper described a procedure for computing a multicycle test set with the following properties: 1) the scan-in states are compressed into seeds for an LFSR and 2) the primary input vectors are held constant during the application of a multicycle test. The procedure is guided by a single-cycle test set. This test set does not have to be applicable using an LFSR with a limited number of bits. The procedure adjusts an initially random seed, the primary input vector, and the number of functional clock cycles of each multicycle test to detect the largest possible number of faults. This process is guided by a single-cycle test. Experimental results for benchmark circuits demonstrated the effectiveness of multicycle tests in achieving test compaction when the tests are required to be producible by an LFSR in order to achieve test data compression

VII. REFERENCES

[1] S. Y. Lee and K. K. Saluja, "Test application time reduction for sequential circuits with scan," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 9, pp. 1128–1140, Sep. 1995.

[2] I. Pomeranz and S. M. Reddy, "Static test compaction for scan-based designs to reduce test application time," in *Proc. Asian Test Symp.*, Singapore, 1998, pp. 198–203.

[3] P. C. Maxwell, R. C. Aitken, K. R. Kollitz, and A. C. Brown, "IDDQ and AC scan: The war against unmodelled defects," in *Proc. Int. Test Conf.*, Washington, DC, USA, 1996, pp. 250–258.

[4] J. Rearick, "Too much delay fault coverage is a bad thing," in *Proc. Int. Test Conf.*, Baltimore, MD, USA, 2001, pp. 624–633.

[5] X. Lin and R. Thompson, "Test generation for designs with multiple clocks," in *Proc. Design Autom. Conf.*, Anaheim, CA, USA, 2003, pp. 662–667.

[6] G. Bhargava, D. Meehl, and J. Sage, "Achieving serendipitous N-detect mark-offs in multi-capture-clock scan patterns," in *Proc. Int. Test Conf.*, Santa Clara, CA, USA, 2007, pp. 1–7.

[7] I. Park and E. J. McCluskey, "Launch-on-shift-capture transition tests," in *Proc. Int. Test Conf.*, Santa Clara, CA, USA, 2008, pp. 1–9.

[8] E. K. Moghaddam, J. Rajski, S. M. Reddy, and M. Kassab, "At-speed scan test with low switching activity," in *Proc. VLSI Test Symp.*, Santa Cruz, CA, USA, 2010, pp. 177–182.

[9] I. Pomeranz, "Generation of multi-cycle broadside tests," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 8, pp. 1253–1257, Aug. 2011.

[10] I. Pomeranz, "Multicycle tests with constant primary input vectors for increased fault coverage," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 9, pp. 1428–1438, Sep. 2012.

[11] I. Pomeranz, "Multi-cycle broadside tests with runs of constant primary input vectors," *IET Comput. Digit. Tech.*, vol. 8, no. 2, pp. 90–96, Mar. 2014.

[12] I. Pomeranz, "A multicycle test set based on a two-cycle test set with constant primary input vectors," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 7, pp. 1124–1132, Jul. 2015.

[13] B. Koenemann, "LFSR-coded test patterns for scan designs," in *Proc. Eur. Test Conf.*, Munich, Germany, 1991, pp. 237–242.

[14] S. Hellebrand, S. Tarnick, J. Rajski, and B. Courtois, "Generation of vector patterns through reseeding of multiple-polynomial linear feedback shift register,"

in *Proc. Int. Test Conf.*, Baltimore, MD, USA, 1992, pp. 120–129.
[15] C. Barnhart *et al.*, "OPMISR: The foundation for compressed ATPG ectors," in *Proc. Int. Test Conf.*, Baltimore, MD, USA, 2001, pp. 748–757.

MAROJU SATYA ANUSHA completed her B.Tech in Vignans Nirula Institute of Technology and Science for Women and M.Tech in GVR&S College of Engineering and Technology. Her specialization is Digital Electronics and Communication Systems (DECS).

K VIJAY KUMAR completed his B.Tech in PBR VITS and M.Tech in NIT TRICHY. Present working as Associate professor in GVR&S College of Engineering and Technology, A.P, India.