Fault diagnosis using LP-TPG for Random Logic circuits

J. Kaushik, M. Jayasanthi

Assistant Professor, Department of ECE, Karpagam College of Engineering, Coimbatore, India

ABSTRACT: The proposed Built-In Self-Diagnosis method (BISD) is based on the standard BIST architecture and can be integrated with recent, commercial DFT techniques, LP-TPG for in-field testing and in-field diagnostic data collection. To find maximum faults, structural diagnosis is used which does reveal the diagnostic information.

A new low power test pattern generator using a linear feedback shift register (LFSR), called LP-TPG, is used instead normal TPG to reduce the average and peak power of a circuit during test. LP-TPG inserts intermediate patterns between the random patterns. The goal of having intermediate patterns is to reduce the transitional activities of primary inputs which eventually reduces the switching activities inside the circuit under test, and hence, power consumption. The random natures of the test patterns are kept intact. The area overhead of the additional components to the LFSR is negligible compared to the large circuit sizes.

KEYWORDS: LP-TPG, Built in self diagnosis.

I. Introduction

Although, there are many testing methods, Built-In Self Test (BIST) has been recognized as a promising solution since it applies structural tests to isolated components and can provide structural test information to the system level. At the system level, only a limited view on the hardware structure of a chip is available and, therefore, it is often not possible to track down the exact root cause of a failure. System vendors are then forced to hand the failure information down the supplier chain until it eventually reaches the chip manufacturer.

Moreover power dissipation of a system in test mode is more than in normal mode. As a result, the failure diagnosis in the field requires a considerable amount of time and power. BIST can be executed in the field and allows at-speed test as the test data is generated within the component. System integrators and workshops can, therefore, benefit greatly from BIST if proper diagnosis methods are provided together with BIST-equipped components.

The standard Bist architecture can also be referred to as self-testing using MISR and parallel SRSG (STUMPS) architecture [1]. In order to reduce the length of PRPG and MISR, STUMPS based architecture that includes a linear phase shifter and a linear phase compactor. But it takes repeated test sessions for the proposed diagnosis methods, whereas the repeated test sessions themselves cause overhead in terms of test time [2], [3], [4]. The second diagnosis technique for random logic BIST relies on dedicated hardware architecture [5]. While only one test session is necessary for the collection of diagnostic information, the architecture has the disadvantage, that it requires a dedicated hardware structure and its corresponding synthesis and test pattern generation (ATPG) processes with power dissipation. Power dissipation occurs due to the correlation between the consecutive test vectors.

In this paper a diagnosis technique and a new LP-TPG are presented, which for the first time overcomes the disadvantages of the above architectures and methods. It is based on the standard STUMPS architecture and thus can be used with the standard tool chains, LP-TPG and synthesis flows. Yet, it overcomes the necessity of repeated tests and extra bandwidth by adding a small response memory and slightly modifying the BIST controller. Modifying the LFSR, by adding weights to tune the pseudorandom vectors for various probabilities, decrease energy consumption and increases fault coverage. The resulting failure information can be read out upon test completion and is input to the proposed diagnosis technique. The method is not confined to the stuck-at fault model, but is able to cover a large variety of defects. The diagnostic success of the method is even superior to that typically achieved in external testing.

II. DIAGNOSIS METHODS FOR RANDOM LOGIC BIST AND THEIR SHORTCOMINGS

Basically, the motivation for the development of the built-in self-test (BIST) technique arose particularly from the cost of test pattern generation and the volume of data that keeps increasing with circuit size. In addition, BIST enables testing at speed. This aspect of testing is especially relevant for present technology, BIST approaches employ the generic architecture shown in figure 1. In order to provide scan input to operate the circuit at scan mode, pseudo random patterns are used from the pattern generation in the normal BIST architecture.

Not only pseudo random patterns, but also deterministic patterns are required in most designs to achieve more fault coverage. Signature register is used to compress the resulting test sequences into a single signature, whereas it is compared with the fault free signatures by using an external tester. test vectors applied to a circuit under test at nominal operating frequency may have more average and peak power dissipation than those in normal mode. The reason is that the random nature of patterns reduces the correlation between the pseudorandom patterns generated by LFSR compared to normal functional vectors. It results in more switching and power dissipation in test mode. While this procedure is adequate to detect failures, it has been recognized that a single signature does not provide enough information to enable fault diagnosis. Most available diagnostic solutions for BIST require several test sessions to narrow down the number of fault candidates in the diagnostic procedure. The test sessions may target specific scan elements [2], [3] work on different pattern sets [4], or employ different response compactors [6]. Once a set of faulty signatures is identified, logic diagnosis can proceed following one of two approaches.

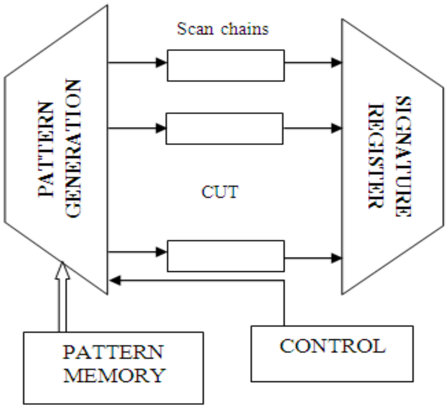

Fig 1. BIST infrastructure

In indirect diagnosis, the values captured by the scan elements are computed for each pattern from the failing signatures [6], [7], [8]. Diagnosis algorithms for combinational logic can then be used on the resulting failure information [9], [10].

In direct diagnosis, the fault location is identified directly from the faulty signature, without sorting out the values of each and every scan element. Such a direct approach has been proposed in [11], where the authors achieve high diagnostic resolution from the failure responses from a multiple input signature register (MISR). However, this method still requires two test sessions: one to gather the complete test response covering all patterns, and, only in the case of a faulty chip, a second test session where each test pattern response is compacted into a signature register.

More recently, a novel built-in self diagnosis (BISD) architecture was proposed in [5], requiring only a single test session and achieving high fault coverage and diagnosis resolution. The architecture, however, substantially differs from the STUMPS scheme and the high fault coverage and diagnostic resolution result from dedicated synthesis and ATPG methods. In contrast to this, the diagnosis method and according architectural modifications proposed in this paper, are built upon the standard STUMPS architecture and can be used with its well-established tool chains. Nonetheless, the advantage of a single test session and a high fault coverage and diagnostic resolution is still achieved.

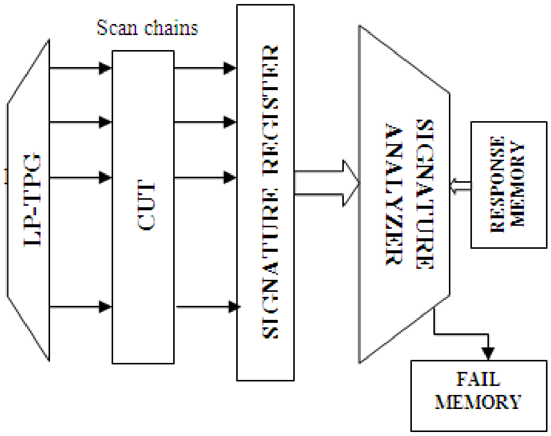

Fig 2. Proposed BISD architecture using LP-TPG

Figure 2 depicts a generalization of the STUMPS architecture assumed in the approach presented below. It is based on the observations from [4], where it was shown, that diagnosis from intermediate signatures is possible. The architecture in figure 2 can collect these intermediate signatures in the field. The LP-TPG plays a major role to reduce the average and peak power while testing. An n-bit linear feedback shift register (LFSR) is fed by a space compactor succeeding the scan chains. A response memory is added. It contains h intermediate test responses, each of which contains n bits. After an intermediate test signature is obtained, it is compared to the expected test signature in the response memory. If the two signatures differ, the obtained signature is stored in the fail memory along with its intermediate signature index, thus resulting in a fail memory width of n+log h bits. The state of the LFSR is reset after every intermediate test response is generated. The depth of the fail memory is limited to g. The content of the fail memory can be downloaded at system level. MISR or signature register here is to reduce length the length of the test intermediate signatures by compaction which it performs bit by bit comparison. If the number of scan chains exceeds, a space compactor may be used.

III. Performance of LP-TPG and Diagnosis algorithm

Generally, power dissipation of a system in test mode is more than in normal mode. This is because a significant correlation exists between consecutive vectors applied during the circuit's normal mode of operation, whereas this may not be necessarily true for applied test vectors in the test mode. Reduced correlation between the consecutive test vectors increases the switching activity and eventually the power dissipation in the circuit. The second reason of increasing the power dissipation during test is because the test engineers may test cores in parallel to reduce the test application time. This extra power (average or peak) can cause problems such as instantaneous power surge that causes circuit damage, difficulty in performance verification and decreased overall product yield and cost. Low power test application has become important in today's VLSI design and test.

A. LP-LFSR

We design the proposed technique into LFSR architecture to create LP-TPG. In this section, we propose a new test pattern generation technique which generates three intermediate test patterns between each two consecutive random patterns generated by a conventional LFSR [12]. The proposed test pattern generation method does not decrease the random nature of the test patterns. Consider two patterns T^1 and T^2, the intermediate patterns going to generate are T^{k1}, T^{k2} *and* T^{k3}. Figure 3 shows 8bit LP-TPG with added circuitry to generate intermediate test patterns. The LFSR used in LPTPG is an external-XOR LFSR. As shown, the rectangular boxes indicate the r-injection, R injection circuit taps the present state (T^i pattern) and the next state (T^{i+1} pattern) of the LFSR. As shown, *R*-injection circuit is included one AND, one OR and one 2×1 MUX. When t_j^i and t_j^{i+1} are equal, both AND and OR gates generate the same bit and regardless of *R*, that bit is transferred to the MUX output. When they are not equal, random bit *R* is sent to the output.

The LP-TPG is activated by two non-overlapping enable signals (*en*1 and *en*2). The first two lines indicate the enable signals *en1* and *en2*. Each enable signal activates one half of the LFSR. In other words, when *en*1*en*2=10, first half of the LFSR is active and the second half is in idle mode. The second half is active when *en*1*en*2=01.The flip flop between *n* /2th and *n/2+1*th flip flops is used to store the *n* /2th bit of the LFSR when *en*1*en*2=10 and that bit is used for the second half when *en*1*en*2=01. MUX selects either the injection bit or the exact bit in the LFSR. One small finite state machine (FSM) controls the pattern generation process as follows:

Step 1: *en*1*en*2=10, sel_1 sel_2 =11. The first half of the LFSR is active and the second half is in idle mode. Selecting sel_1 sel_2=11, both halves of the LFSR are sent to the outputs (O_1 to O_n). In this case, T^i is generated.

Step 2: *en*1*en*2=00, sel_1sel_2=10. Both halves of the LFSR are in the idle mode. The first half of the LFSR is sent to the outputs (O_1 to $O_{n/2}$), but the injector circuit outputs are sent to the outputs ($O_{n/2+1}$ to O_n). T^{k1} is generated.

Step 3: *en*1*en*2=01, sel_1sel_2=11. The second half of the LFSR is active and the first half of the LFSR is in idle mode. Both halves are transferred to the outputs (O_1 to O_n) and T^{k2} is generated.

Step 4: *en*1*en*2=00, sel_1sel_2=01. Both halves of the LFSR are in the idle mode. From the first half the injector outputs are sent to the outputs of LP-TPG (O_1 to $O_{n/2}$) and the second half sends the exact bits in the LFSR to the outputs ($O_{n/2+1}$ to O_n) to generate T^{k3}.

Step 5: The process continues by going through Step 1 to generate T^{i+1}. The LP-TPG with R-injection keeps the random nature of the test patterns intact. The FSM controls the test pattern generation through steps 1 to 4 and it is independent of the LFSR size and polynomial. *clk* and *test_en* are the inputs of the FSM. When *test_en*=1, FSM starts with step 1 by setting *en*1*en*2=10 and sel_1sel_2=11. It continues the process by going through step 1 to step 4. One pattern is generated in each clock cycle. The size of the FSM is very small and fixed as reported in Section IV.

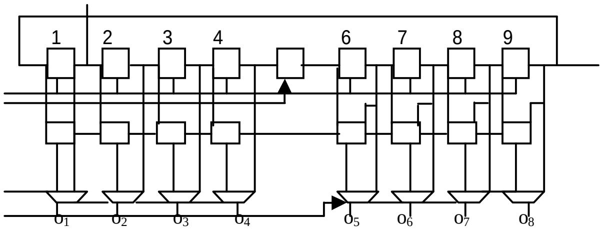

Fig 3. 8-bit LP-LFSR

B. Basis concepts in diagnosis method

In this paper, we assume that defects are arbitrary because if a defect acts like a stuck-at fault, the intermediate signatures stored in the fail memory are sufficient for diagnosis [4].

To analyze complex defect mechanisms, we make use of the conditional stuck-at fault model [13]. The conditional stuck-at model represents a stuck-at fault on a single line in certain situations. That is, depending on some internal or external condition, a line may behave as a stuck-at-1 or stuck-at-0 fault for some patterns, or even as a fault-free line for other patterns that would usually excite a fault behavior. For each line v, we consider the conditional stuck-at-faults cond_0_v and cond_1_v. The condition *cond* may describe arbitrary Boolean or timing properties. For instance, (v = 1)_0_v is a permanent stuck-at-0, and (v−1 = 0^v = 1)_0_v describes a slow-to-rise fault.

Diagnosis based on this fault model relies on pattern-wise information of outputs in order to reason about possible stuck-at candidates and their activation conditions.

The following two sections describe how the most likely response sequences can be derived for every stuck-at fault with the information stored in the fail memory.

C. Generating fault-free signatures

We consider m is the maximum length of all the scan chains in the STUMPS scheme, and the results of each test pattern are compacted into a single signature in m clock cycles. Let n be less than or equal to the length of the LP-LFSR, if the number of scan chains exceeds n, a space compactor may be used. Let T be the set of test patterns, which is partitioned into h = [|T|/n] of blocks B. Each block contains n patterns at most. The patterns of each block are compacted into a single signature. For that purpose, MISR (multiple input signature register) is used which performs compaction, so it is possible for information loss.

The state transition function of an LFSR can be represented as a feedback matrix, e.g. for a type-I LFSR

$$L = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ . & . & . & \ldots . & & . \\ . & . & . & \ldots . & & . \\ & & & & & \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ 1 & l_1 & l_2 & \ldots & l_{n-2} & l_{n-2} \end{bmatrix}$$

Where matrix elements l_i correspond to the coefficients of the LFSR generator polynomial.

The matrix H = Lm describes the autonomous function of the LFSR after m cycles. Each block

$$B = \begin{bmatrix} p_1 \\ . \\ . \\ \ \end{bmatrix} \qquad (1) \qquad p_n$$

provides a signature S_B. Let s_i be the signature from pattern p_i which is obtained by shifting the m response vectors of p_i into the LFSR starting in the all-zero state. Applying linear superposition, the final signature after applying all patterns in B is captured by the equation:

$$S_B = \sum_{i=1}^{n} H^{n-i} s_i \qquad (2)$$

After each block B, the LFSR is reset and only SB has to be stored in the response memory.

D. *Analysis of erroneous signatures*

The block B may contain some test patterns that activate the fault s@0-v, that is, a stuck-at-0 on location v (or s@1-v, a stuck-at-1 on v) and, according to the conditional stuck-at fault model, depending on a given condition *cond*, these patterns may also detect cond_0_v (or cond_1_v). Hence, we have to determine the signatures for the unconditional s@0-v (or s@1-v), and we have to select a pattern subsequence which fits cond_0_v (or cond_1_v).Let f be such an unconditional stuck-at fault, and let s_i^f be the signature of pattern pi in the presence of f if the LFSR starts in the all-zero state. The error vectors are defined as:

$$e_i^f \coloneqq s_i \oplus s_i^f \qquad (3)$$

We have $|e_i^f| \neq 0$ if and only if pattern p_i detects f in its signature. Now assume, the real fault $\tilde{f}$ is a conditional stuck-at fault. In this case, we have either $|e_i^{\tilde{f}}| = e_i^f$, if in pattern p_i the condition is true, or we have $|e_i^f| = 0$. This can be decided by solving a set of linear equations. Let

$$d_i^f \coloneqq H^{n-I} e_i^f \qquad (4)$$

Now, we have to look for constants $c_1, c_2, \cdots, c_{n-1}, c_{n-2}, c_n \in \{0, 1\}$ with:

$$\begin{bmatrix} d_1^f \, d_2^f \ldots\ldots d_{n-1}^f \, d_n^f \end{bmatrix} \begin{bmatrix} c_1 \\ \\ c_{n-1} \\ c_n \end{bmatrix} = S_B \oplus S_B^f \quad (5)$$

The matrix $[d_1^f \; d_2^f \; \ldots. \; d_n^f]$ can be pre-computed for any unconditional stuck-at fault f, the correct signature S_B after pattern block B can be pre-computed as well, and $S_B^{\tilde{f}}$ is the observed faulty signature. Hence, equation (5) contains at least n equations with n unknowns. If equation (5) is solvable for a stuck-at fault f, $[c_1, c_2, \cdots, c_{n-1}, c_n]$ describes the fault conditions for the patterns [p_1, p_2, · · · , p_n], otherwise the fault location of f cannot be the single culprit. This approach only requires the solution of a system of linear equations after the fault simulation step usually employed for logic diagnosis.

E. Fault diagnosis

For each pattern block B, let B be the set of faults that can fully explain the observed faulty behavior in $S_B^{f\tilde{}}$, which is:

B ≔ {f| equation (5) is solvable for f}.

The method described in section III-C provides for each block B such a set of suspect locations (faults). The number of blocks B with f ϵ B is a measure of the fault's *evidence*. That is, the higher this number is, the more likely *f* is in fact the real cause of the defect behavior. With these criteria, a ranked fault list can be created for logic diagnosis as follows. Let F be the set of conditional stuck-at faults.

Mapping *evidence*: F → N0 is defined as

$$Evidence\ (f) := |\ \{B|\ f \in B\}| \qquad (6)$$

The faults f_i are ordered due to decreasing values of *evidence* (*fi*) and each fault f_i is assigned a rank, which is its position in the resulting ordered list. The fail memory contains at most *g* fault signatures, and there may be blocks in between providing correct signatures. If there are two faults f1 and f2 with *evidence* (f_1) = *evidence* (f_2) we order these two faults by using the correct signatures. Let f'_1 and f'_2 be the unconditional counterparts of f_1 and f_2, let ˆB be the set of blocks which provided a correct signature until the fail memory was full. We assign a fault f_1 a higher priority, if the corresponding unconditional stuck-at fault f'_1 would also lead to a correct signature. To exemplify this, assume a test set was divided into four pattern blocks, B0 provided a correct signature, and the three remaining sets e B looked as follows:

$$B_1 = \{ f_1 \} \qquad (7)$$
$$B_2 = \{ f_2, f_1, f_3 \} \qquad (8)$$
$$B_3 = \{ f_2 \} \qquad (9)$$

Hence, *evidence* (f_1) = 2, *evidence* (f_2) = 2 and *evidence* (f_3) = 1. If the unconditional f'_2 led to a correct signature in B_0, but did not, we would rank f_2 before f_1.

IV. Experimental results

A. *Power analysis and simulation results*

In this section, the experimental results are presented on the ISCAS benchmark circuit c432. The power estimation results are shown based on switching activity (WSA). Table 1 shows the specifications of the circuit c432, i.e., the number of primary inputs and outputs, fault coverage. Moreover, fault coverage and power is estimated by comparing the average and peak power of the normal LFSR and our proposed LP-TPG.

circuit	PI	PO	WSA_{avg}	WSA_{peak}	FC%
C432	36	7	94	153	98
C499	41	32	128	203	97
C880	60	26	167	288	97

Table 1. Average and peak WSA for normal LFSR with zero delay.

circuit	PI	PO	WSA_{avg}	WSA_{peak}	FC%
C432	36	7	36	134	98
C499	41	32	97	174	97
C880	60	26	89	268	97

Table 2. For LP-TPG with zero delay

V. CONCLUSION

This paper presented a new low-power LFSR to reduce the average and peak power of a random logic circuit during the test mode and a new BISD scheme based on standard STUMPS architecture. Due to LP-TPG, additional intermediate test patterns are inserted between the original patterns which reduce the average and peak power but do not affect the fault coverage.

Several test patterns are compacted into intermediate response signatures and are compared to their corresponding reference signatures stored on-chip. The scheme can be implemented with only minimal modifications to the available design-for test infrastructure and with insignificant storage overhead.

REFERENCES

[1] P. Bardell and W. McAnney, "Self-testing of multichip logic modules." in *Proceedings of the IEEE International Test Conference,* 1982, pp. 200-204.

[2] I. Bayraktaroglu and A. Orailoglu, "Gate level fault diagnosis in scan- based BIST." in *Proceedings of the Design, Automation and test in Europe conference and Exhibition,* 2002, pp. 376-381.

[3] J. Ghosh-Dastidar and N.A. Touba, "A rapid and scalable diagnosis scheme for BIST environments with a large number of scan chains," in *Proceedings of the 18th IEEE VLSI Test Symposium*, 2000, pp. 79-85.

[4] P. Wohl, J. A. Waicukauski, S. Patel, and G. Maston, "Effective diagnostics through interval unloads in a BIST environment," in *Proceedings of the 39th Conference on Design Automation*, 2002, pp. 249-254.

[5] C. Liu, K. Chakrabarty, and M. Goessel, "An interval-based diagnosis scheme for identifying failing vectors in a scan-BIST environment," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition,* 2002, pp. 382-386.

[6] N. Touba, "X-cancelling MISR: An X-tolerant methodology for compacting output responses with unknowns using a MISR," in *Proceedings of the IEEE International Test Conference,* 2007, pp. 1-10.

[7] J. Patel, S. Lumetta, and S. Reddy, "Application of Saluja-Karpovsky compactors to test responses with many unknowns." in *Proceedings of the 21st VLSI Test Symposium,* 2003, pp. 107-112.

[8] R. Desineni, O. Poku, and R.D.S Blanton, "A logic diagnosis methodology for improved localization and extraction of accurate defect behavior," *in Proceedings of the IEEE International Test Conference,* 2006, pp. 1-10.

[9] M.E. Amyeen, D. Nayak, and S. Venkataraman, "Improving Precision using mixed-level fault diagnosis," in *Proceedings of the IEEE International Test Conference*, 2006, pp. 1-10.

[10] W.-T. Cheng, M. Sharma, T. Rinderknecht, L. Lai, and C. Hill, "Signature based diagnosis for random logic BIST," in *Proceedings of the IEEE International Test Conference*, 2006, pp. 1-9.

[11] Seongmoon Wang and Sandeep. k. Gupta, "DS-LFSR: A BIST TPG for low switching Activity, " in *Proceedings of the IEEE Transactions on computer-Aided Design of Integrated Circuits and systems,* vol. 21, no.7, 2002.

[12] S. Holst and H.-J. Wunderlich, "Adaptive debug and diagnosis without fault dictionaries." in *Proceedings of the 27th IEEE VLSI test symposium,* 2006, pp, 179-184.