



## LFSR-Based Deterministic TPG for Two-Pattern Testing\*

XIAOWEI LI

*Department of Computer Science, Peking University, Beijing 100871, P.R. China*

PAUL Y.S. CHEUNG

*Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong*

HIDEO FUJIWARA

*Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 630-0101, Japan*

*fujiiwara@is.aist-nara.ac.jp*

*Received April 15, 1999; Revised November 1, 1999*

Editor: S. Demidenko

**Abstract.** This paper proposes an approach to designing a cost-effective deterministic test pattern generator (TPG) for two-pattern testing. Given a set of pre-generated test-pair set (obtained by an ATPG tool) with a pre-determined (path delay) fault coverage, a simple TPG is synthesized to apply the given test-pair set in a minimal test time. To achieve this objective, a configurable linear feedback shift register (CLFSR) structure is used. Techniques are developed to synthesize such a TPG, which is used to generate an unordered deterministic test-pair set. The resulting TPG is efficient in terms of hardware size and speed performance. Experiments on benchmark circuits indicate that TPG designed using the proposed procedure obtain high path delay fault coverage in short test length.

**Keywords:** built-in self-test, two-pattern test, configurable LFSR, path delay faults

### 1. Introduction

Pseudo-random testing is popularly used in BIST applications. To achieve a desired fault coverage, pseudo-random patterns are often supplemented with few deterministic patterns. When deterministic patterns are generated a priori, pseudo-random sub-sequences can be chosen such that they also cover these deterministic patterns. We call this method *pseudo-deterministic testing* (PT).

The majority of BIST techniques for PT aim at combinational faults, i.e., faults that can be always detected using a single test pattern. Few work has been done on BIST for the detection of sequential faults, e.g., delay faults and stuck-open faults. To detect any of these faults, a specific sequence of input patterns has to be applied in two consecutive clock cycles. Such a pair of input patterns is referred to as a *test-pair*.

Several well-known BIST *test pattern generator* (TPG) schemes have been proposed for two-pattern testing. The two-pattern generation capability of TPGs can be evaluated using transition coverage [1] or the AC strength [2], which is defined as the ratio of the maximum number of test-pair that can be generated to the exhaustive test-pair count  $2^n(2^n - 1)$ :

\*This project is supported in part by Natural Science Foundation of China (NSFC).

- Exhaustive testing [3] and pseudo-exhaustive testing [1, 2, 4, 5]. Exhaustive test-pair pattern count for a  $n$ -input *circuit under test* (CUT) is  $2^n(2^n - 1)$ . For a  $n$ -input CUT, a  $2n$ -stage LFSR can apply exhaustive test-pair patterns which is to give all transitions between every distinct test-pair pattern. It has been shown that only  $n2^n$  test pairs that differ on a single bit are sufficient to detect all robustly testable path delay faults. These schemes are limited to CUTs with small number of inputs.
- Pseudo-random testing [2, 6] and weighted-random testing [3]. The test pattern can be generated using LFSR, cellular automata [6] or circular self-test path [2]. The linear nature of these structures reduces the area overhead. However, the achievable robust path delay fault coverage can only be determined via fault simulation in a post-processing step. If the fault coverage is poor, further selection steps with increased TPG size may be required.
- Pseudo-deterministic testing [7–9]. In [7], a  $n$ -stage LFSR is synthesized such that a set of  $n$  pre-computed test-pairs is embedded in the maximal length pseudo-random test sequence of the LFSR. The restriction that only  $n$  test-pairs being covered limits its practical application. In [9], a *multiple input shift register* (MISR) is re-seeded several times with a constant parallel input vector. Using an ATPG-based selection algorithm for the determination of the optimal input vectors, the MISR runs through its maximal length sequence and provides maximal robust path delay fault coverage. All possible test-pairs can not be generated unless all  $2^n$  different input vectors were tried.

The above approaches differ in fault coverage, test length and area overhead.

In this paper, we propose a new pseudo-deterministic BIST TPG scheme for generating a set of pre-generated test-pair set using a configurable LFSR structure. A set of deterministic test-pair is generated to detect all robust path delay faults in the CUT. A LFSR-based TPG synthesis method is proposed, such that the deterministic test-pair sequences can be included in the TPG *maximum length sequences* (MLSs). For a  $n$ -stage LFSR, our technique guarantee that these deterministically generated test-pairs embedded in a set of classical pseudo-random test sequences. This is solved in two steps. First, LFSR primitive polynomial is generated for each test-pair. Then, a minimum number of LFSR configurations are found such that corresponding MLSs cover all test-pairs. This problem is formulated

as a cluster covering, which could be efficiently solved using binary decision diagrams. Experimental results reveal that the test length can be drastically reduced.

This paper is organized as follows. Section 2 gives some definitions of configurable LFSR-based TPG. Our approach is described in Section 3, where the configurable LFSR-based TPG design algorithm is given. The experimental results based on benchmark circuits are presented in Section 4. The final section gives the conclusions and proposes future work.

## 2. Configurable LFSR

The objective of this paper is to explore the capability of LFSRs and to demonstrate its capability of generating deterministic test-pair sequences, for the detection of sequential faults, such as stuck-open and delay faults. For simplicity, we restrict our discussion to path delay testing.

LFSRs are a class of linear sequential machines constructed from clocked D flip-flop and modulo-2 adders (XOR gates). There are two configurations for the LFSR: a Type I LFSR which has the XOR gates between cells, and a Type II LFSR which has the XOR gates on the feedback chain. An  $n$  stage LFSR is characterized by its feedback (characteristic) polynomial given by

$$P(x) = \sum_{i=0}^n c_i x^{n-i} \quad (1)$$

where coefficients  $c_i \in \{0, 1\}$ ,  $i = 0, 1, \dots, n-1$ ,  $c_0 = c_n = 1$ ,  $x^k$  represents the state of  $k$ -th shift register bit.

In this paper, we use Type II LFSR and assume that shifting and numbering of the cells are from left to right. The next state of the LFSR is determined by the shift operation and the corresponding polynomial coefficients. The current state of each cell except the *most significant bit* (MSB) is equal to the content of the previous cell at the previous clock. The state of the MSB is determined by the corresponding XOR operations of the cells with feedback links (according to Eq. (1)).

If the feedback polynomial is primitive, then the LFSR (initialized to any non-zero state) generates a sequence of length  $2^n - 1$  before returning to the initial state. Such a LFSR is called a maximal length LFSR. The sequence of  $2^n - 1$  states generated by the LFSR with polynomial  $P(x)$  is called MLS.

The next state function of a LFSR can be represented by using a transition matrix, or briefly a  $T$ -matrix,  $T$ ,

$$X' = TX, \quad (2)$$

where  $X$  and  $X'$  denote the current and the next states.

**Theorem 1.** *Let  $P_j(x)$  and  $P_k(x)$  be primitive polynomial ( $T_j$  and  $T_k$  are corresponding  $T$ -matrix), let  $S_j$  and  $S_k$  denote two corresponding distinct MLSs. Then, two consecutive states  $\langle v, w \rangle$  (a test-pair) that are contained in  $S_j$  will not be repeated in  $S_k$ .*

**Proof:** Let  $\langle v_1, w_1 \rangle$  be from  $S_j$  and  $\langle v_2, w_2 \rangle$  be from  $S_k$ ,  $\langle v_1, w_1 \rangle = \langle v_2, w_2 \rangle$  implies  $v_1 = v_2$  and  $w_1 = w_2$ . Because addition (+) is modulo 2, this is equivalent to  $v_1 + v_2 = 0$  and  $w_1 + w_2 = 0$ . Since  $w_1(w_2)$  is the next state of  $v_1(v_2)$ ,  $w_1 + w_2 = 0$  can also be expressed as follows

$$\begin{aligned} w_1 + w_2 = 0 &\Leftrightarrow T_j v_1 + T_k v_2 = 0 \\ &\Leftrightarrow (T_j + T_k)v_2 = 0 \\ &\Leftrightarrow T_j + T_k = 0 \\ &\Leftrightarrow T_j = T_k \end{aligned}$$

This contradicts  $T_j \neq T_k$ , therefore,  $\langle v_1, w_1 \rangle \neq \langle v_2, w_2 \rangle$ . □

In this paper, we use additional logic to encode LFSRs with different polynomials into a single *configurable LFSR* (CLFSR). Since LFSRs are used as the basis of the TPG for two-pattern testing, then a large number of different polynomials can be generated which depends on the configuration of these LFSRs. A  $n$ -stage CLFSR is shown in Fig. 1. It is easy to see that the values of the configuration inputs are simply the coefficients of the LFSR's feedback polynomials. The values of configuration inputs could be stored in a ROM on chip or be set using an additional scan path.

It is easy to configure the feedback of a LFSR, this allow the choice of any characteristic polynomial for

the LFSR. Note that, we are using configuration inputs as means of encoding various LFSR-based TPG into one CLFSR, the final implementation of the synthesized BIST TPG will not necessarily have all configuration inputs. Suppose that a few configurations are selected, if any configuration input has a constant 0/1 value over all the configurations, that input and associated logic can be removed by applying the constant 0/1 value. As an extreme case, if exactly one configuration is selected, then the values of the configuration inputs are set to particular values, and the logic can be simplified accordingly.

According to Theorem 1, if a  $n$ -stage CLFSR encodes  $k$  primitive polynomials, a total of  $k(2^n - 2)$  distinct test-pairs can be generated.

The idea of configurable LFSR could be applied to MISR and/or *built-in logic-block observation* (BILBO). A configurable  $n$ -stage MISR is shown in Fig. 2, values of the configuration inputs are simply the coefficients of the MISR's characteristic polynomial,  $d_i$  ( $i = 1, 2, \dots, n$ ) are external inputs.

It is easy to prove that  $2^n(2^n - 1)$  distinct test-pairs can also be generated using a  $n$ -stage MISR (by using  $2^n$  different input vectors). It has been shown in [9] that MISRs have the capability of generating pre-deterministic test-pairs. This result can be applied together with ours.

In this paper, we use two terms feedback polynomial and characteristic polynomial interchangeably.

### 3. CLFSR-Based TPG Design Algorithm for Delay Testing

We consider a set of deterministically generated test-pair patterns  $V$  which detect all testable path delay faults in the CUT,  $V_i \in V, i = 1, 2, \dots, m$ . These test-pairs can be denoted as  $V_1 = \langle v_{1.1}, v_{1.2} \rangle, \dots, V_i = \langle v_{i.1}, v_{i.2} \rangle, \dots, V_m = \langle v_{m.1}, v_{m.2} \rangle, i = 1, 2, \dots, m$ .

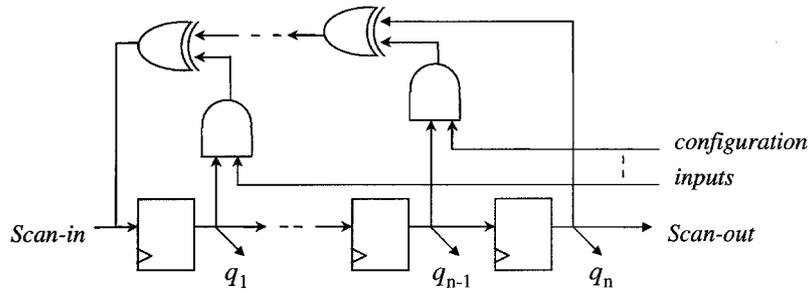


Fig. 1. Configurable LFSR.

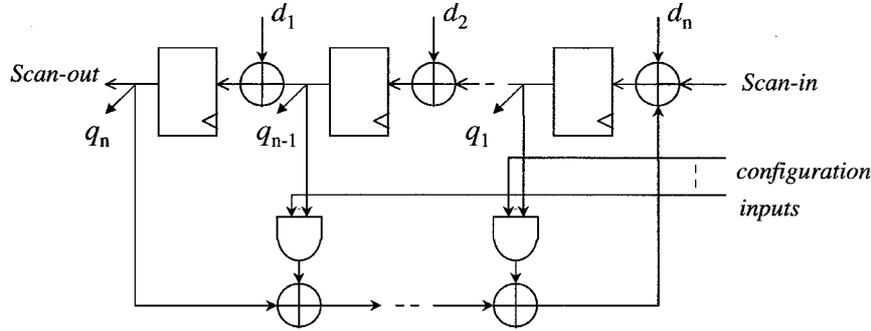


Fig. 2. Configurable MISR.

$v_{i,1}$  represents the first pattern of the  $i$ -th test pair.  $v_{i,2}$  represents the second pattern of the test pair.

In general,  $v_{i,1}$  and  $v_{i,2}$  are incompletely specified, i.e.,  $v_{i,1}$  and  $v_{i,2}$  contain don't care values,  $v_{i,1}, v_{i,2} \in \{0, 1, x\}$ . Hence, a number of MLSs may contain  $V_i$  after proper assignments of don't care values. According to Theorem 1, a fully defined test-pair is contained in exact one MLS. If  $v_{i,1}$  and  $v_{i,2}$  are fully defined vectors (min-terms), according to Eq. (2), the following relation

$$v_{i,2} = T v_{i,1}$$

can be used to check whether  $\langle v_{i,1}, v_{i,2} \rangle$  is contained in the maximal length sequence.

### 3.1. Problem Formulation

The problem of configurable LFSR-based TPG design for two-pattern testing can be viewed as: *determine a minimum cardinality set of the maximal length sequence  $S$  such that each test pair  $V_i \in V$  is contained in at least one MLS  $S_k \in S$ .*

This problem can be solved in two steps. First, a deterministic path delay test generation tool is employed to generate a set of test-pairs, which detect all robust path delay faults. Second, TPG configurations are calculated, such that the state sequences generated by the synthesized CLFSR include these pre-determined test-pairs. This is solved in another two steps:

- (1) LFSR primitive polynomial is generated for each test-pair;
- (2) A minimum number of LFSR configurations are found such that corresponding MLSs can cover all these test-pairs.

### 3.2. TPG Design for Delay Testing

Assume that we have a set of pre-generated test-pairs. We now wish to determine which configuration of the LFSR-based TPG will produce a particular test-pair  $V_i \in V$ . It is guaranteed that at least one configuration exists. Assuming the TPG is clocked enough times, this TPG will eventually produce  $V_i$ .

For each  $V_i \in V$ ,  $V_i = \langle v_{i,1}, v_{i,2} \rangle$ , suppose  $v_{i,1} = (x_{n-1}, \dots, x_2, x_1, x_0)$ ,  $v_{i,2} = (x_n, x_{n-1}, \dots, x_2, x_1)$ . We try to calculate one primitive polynomial in the form of Eq. (1). First, we combine the two consecutive patterns together, with one bit shift right, to form a sequence of  $n + 1$  bit:  $(x_n, x_{n-1}, \dots, x_2, x_1, x_0, )$ . There exist several efficient methods to calculate  $P_i(x)$ , generating this combined sequence [10].

Now, for each test-pair  $V_i \in V$ , we are able to calculate one primitive polynomial, i.e. CLFSR configuration, such that the corresponding MLS contains  $V_i$ . A partially determined test-pair may occur in different MLSs, but a minterm test-pair occurs only in one MLS. Our goal is to choose a minimum number of configurations, such that corresponding MLSs collectively generate all  $V_i \in V$ .

We transform this problem into a cluster-covering problem on an undirected graph  $G = (\text{MLS}, E)$ , where  $E$  is the set of edges. Each node corresponds to a test-pair. There is an edge between  $V_i = \langle v_{i,1}, v_{i,2} \rangle$  and  $V_j = \langle v_{j,1}, v_{j,2} \rangle$  iff

$$\exists \langle v_{i,1}, v_{i,2} \rangle \in \text{MLS}_k \wedge \langle v_{j,1}, v_{j,2} \rangle \in \text{MLS}_k$$

i.e.,  $V_i$  and  $V_j$  are both in the same  $\text{MLS}_k$ . This is checked by Theorem 1. A cluster is a fully connected subgraph of nodes where each node in the subgraph is connected to every other node via an edge. Each cluster  $C_r$  represents those test-pairs can be generated with the same MLS.

- DS* : = the deterministic test-pair set (with path delay fault coverage)  
*PS* : = primitive polynomials set
- Step 1: Obtain *DS* by a deterministic ATPG tool.
  - Step 2: Chose a uncovered test-pair from *DS*, generating primitive polynomial for it.
  - Step 3: Eliminating test-pairs that are covered by the *MLS* generate in step 2.
  - Step 4: Repeat Step 2 and 3 until *DS* is covered by these *MLS*s.
  - Step 5: Create a graph, each node represents a test-pair in *DS*, an edge between two nodes if they are covered by the same *MLS*.
  - Step 6: Find a minimum number of clusters to cover the graph, get *PS*.
  - Step 7: Using *PS* to form the CLFSR.

Fig. 3. CLFSR design procedure.

Given the test-pair set  $V$  and calculated configurations for each  $V_i \in V$ . Now, we create a graph  $G$  accordingly. Since the cluster covering problem, in general, is NP-complete, a heuristic approach must be adopted. It can be solved efficiently by existed algorithm [11]. The entire procedure for CLFSR design is described in Fig. 3.

The above procedure can be easily applied to the design of configurable MISR, with only little modification, i.e., it needs additional seed synthesis.

### 3.3. An Example

Consider the CUT shown in Fig. 4. The circuit has 10 structural paths from the PIs to the POs. Corresponding to the two possible transitions (rising, falling) at the primary inputs (PIs), there are 20 path delay faults. This circuit can be completely tested by a set of 12 deterministically generated test-pairs:

- $v_1 = \langle x0101, 0x010 \rangle, v_2 = \langle 1x00x, 0100x \rangle,$
- $v_3 = \langle 0x111, 10111 \rangle, v_4 = \langle xx01x, x0x01 \rangle,$
- $v_5 = \langle x1011, x1101 \rangle, v_6 = \langle 11110, x1111 \rangle,$
- $v_7 = \langle xx001, xx000 \rangle, v_8 = \langle 00101, x0010 \rangle,$
- $v_9 = \langle 1011x, x1011 \rangle, v_{10} = \langle 0x11x, 10xx1 \rangle,$
- $v_{11} = \langle 1100x, x110x \rangle, v_{12} = \langle xx011, xx001 \rangle.$

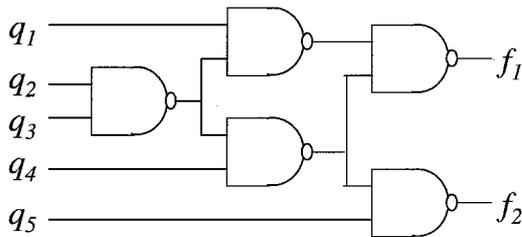


Fig. 4. Example of CUT.

Fig. 5 shows schematic of the CLFSR-based TPG that corresponds to the CUT in Fig. 4. Note that, the  $q_i$  ( $i = 1, 2, 3, 4, 5$ ) in the Fig. 5 connect to the same  $q_i$  in the Fig. 4.

Fig. 6 shows graph  $G$  for these 12 deterministically generated test-pairs. In  $G$ , there is an edge between  $v_1$  and  $v_5$  since both test-pairs are included in a *MLS* with properly specified don't care. A set of 3 clusters can cover the whole 12 test-pairs, which are listed as follows:

- $C_1 = (v_1, v_5, v_7, v_9, v_{10}),$  its primitive polynomial is  $X^5 + X^3 + 1;$
- $C_2 = (v_2, v_4, v_6, v_7, v_{12}),$  its primitive polynomial is  $X^5 + X^2 + 1;$
- $C_3 = (v_3, v_6, v_8, v_9, v_{11}),$  its primitive polynomial is  $X^5 + X^3 + X^2 + X + 1.$

Therefore, there are 3 *MLS*s that include all these test-pairs.

For a  $n$ -stage LFSR, the above approach allows the generation of a set of deterministic test-pairs embedded in a set of *MLS*s. Since all the basic properties of LFSR are preserved in our approach, therefore, any other TPG design technique using LFSR (or BILBO) can be beneficially applied in conjunction with this approach.

## 4. Experimental Results

The proposed algorithm was implemented and tested on selected circuits in the ISCAS'89 benchmark set [12] (full scan version). We choose a few small-size benchmark circuits for experiment. The configurable LFSR topology in Fig. 1 was chosen. We consider as CUT the combinational part of these circuits.

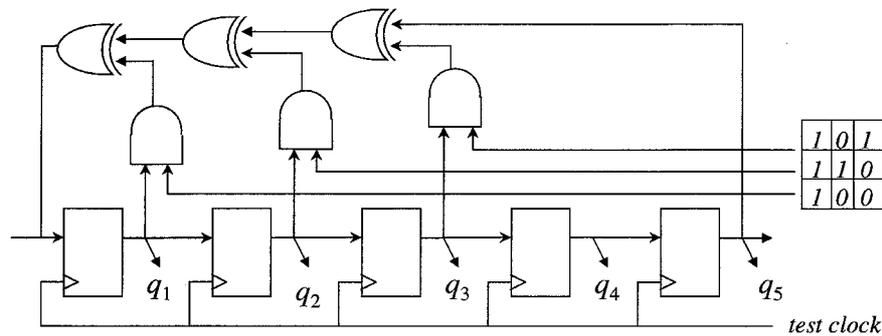


Fig. 5. Corresponding CLFSR-based TPG.

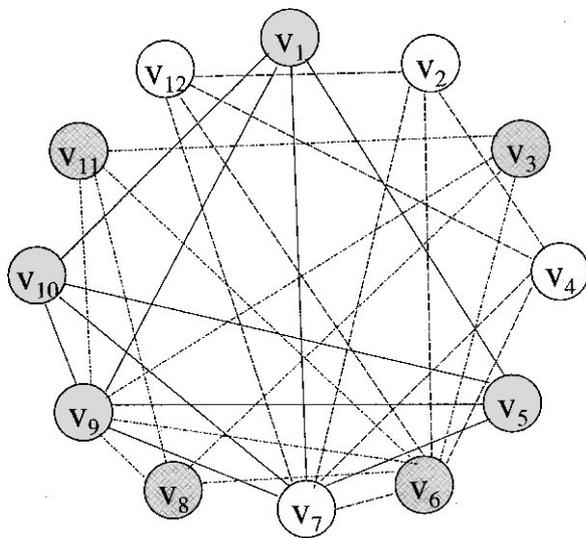


Fig. 6. Minimum cluster covering.

In Table 1, we report the results obtained for configurable LFSR-based TPG synthesis. For each circuit, get a set of test-pair patterns that are generated by a deterministic ATPG tool. We report the number of inputs (primary inputs + D-output of FFs) to the CUT, the number of test-pair patterns, the number of path delay faults, the number of TPG configurations chosen, fault coverage achieved and hardware overhead. TPG configuration is uniquely determined by LFSR primitive polynomial. The fault coverages are reported as a fraction of the number of robustly testable path delay faults, which evaluated by a path delay fault simulator. It can be seen that high fault coverages are achieved with reasonable hardware (a  $n$ -stage CLFSR).

In the experiments, increase the number of configuration may increase the fault coverage. It is possible that if the configurable LFSRs were run for more cycles, higher fault coverages could be achieved. Of course, the change of the configurations of CLFSR, using an

Table 1. Experiments on ISCAS'89 benchmarks.

Circuit name	# Inputs	# Test-pairs	# Path faults	# Poly	% Fault coverage	% Area overhead
S208	19	289	290	10	83.1	7.94
S298	17	352	352	9	68.4	8.11
S344	24	643	710	16	75.6	9.09
S349	24	643	730	16	73.0	9.09
S386	13	704	800	8	78.4	8.51
S510	25	736	738	19	85.8	10.11
S526	24	708	820	20	79.9	12.5
S820	23	982	984	18	89.7	10.34
S832	23	996	1012	23	86.7	12.5
S1488	14	1916	1924	12	91.7	11.11
S1494	14	1926	1952	11	90.8	12.79

Table 2. Comparison of TPGs for two-pattern testing.

TPG schemes	Test length	Fault coverage	Hardware requirements
LFSR/CA [2]	$<2^n(2^n - 1)$	High	$2n$ stage
LFSR [5]	$<(2^{2n} - 1)$	High	$2n$ stage
SIC-TPG [6]	$n(2^n - 1)$	Max	$n$ stage (LFSR + Shift Register)
CSTP [9]	$<2^n(2^n - 1)$	High	$2n$ stage
ACCU [12]	$2^n(2^n - 1)$	High	$n$ -bit (Counter + Accumulator)
MISR [13]	$m(2^n - 1)$	Max	$n$ stage
XLFSR [14]	$<(2^n - 1)$	High	$n$ stage
CLFSR [8]	$m(2^n - 1)$	Max	$n$ stage LFSR + $m$ AND

additional scan path, will require increased hardware cost.

In Table 2, a qualitative comparison of our BIST scheme (firstly proposed in [5]) with selected TPG schemes proposed for two-pattern testing. We selected a few outstanding contributions in this field for comparison. For each scheme, we listed its maximal test length, estimated fault coverage and hardware requirement.

In Table 2,  $m$  is the number of seeds. Fault coverage is estimated depending on the fact that it can be maximal (Max) or lower than maximal (High).

## 5. Conclusions and Future Work

In this paper, we have exploited BIST approach for two-pattern testing. The generation of deterministic test-pairs with configurable LFSR-based TPG was exploited. A two-step approach was proposed. First, a set of deterministic test-pairs is generated which is capable of detecting all robust path delay faults in the CUT. Second, configurable LFSR-based TPG configuration is calculated to have pre-generated test-pairs embedded in a set of maximal length pseudo-random test sequences. This is formulated as a cluster covering problem. Experimental results are presented to demonstrate the effectiveness of the proposed BIST (for two-pattern testing) methodology.

There are several ways the presented methodology can be improved upon. As maximum length sequence consists of  $k(2^n - 2)$  test-pair patterns, our approach is confined to circuits with less than 30 inputs. The general challenge is to develop a constructive CLFSR-based TPGs synthesis approach that determines the best configurations for a given CUT.

As part of our future work, the proposed LFSR-based TPG design approach will be tested on more

academic benchmarks circuits as well as real industrial circuits.

## References

1. K. Furuya and E.J. McClusky, "Two Pattern Test Capability of Autonomous TPG Circuits," *Proc. of IEEE 1991 Int'l Test Conf. (ITC'91)*, pp. 704–711.
2. S. Pilarski and A. Pierzyrska, "BIST and Delay Fault Detection," *Proc. of IEEE 1993 Int'l Test Conf. (ITC'93)*, pp. 236–242.
3. I. Voyiatzis, A. Paschalis, D. Nikolos, and C. Halatsis, "Accumulator-Based BIST Approach for Stuck-Open and Delay Fault Testing," *Proc. of IEEE 1995 European Design and Test Conf. (EURO DTC'95)*, pp. 431–435.
4. P. Girard, C. Landrault, V. Moreda, and S. Pravossoudovitch, "An Optimized BIST Test Pattern Generator for Delay Testing," *Proc. of IEEE 1997 VLSI Test Symposium (VTS'97)*, pp. 94–100.
5. X. Li and P.Y.S. Cheung, "Exploiting BIST Approach for Two-Pattern Testing," *Proc. of IEEE 1998 Asian Test Symp. (ATS'98)*, pp. 424–429.
6. C.A. Chen and S.K. Gupta, "BIST Test Pattern Generators for Two-Pattern Testing: Theory and Design Algorithms," *IEEE Trans. on Computers*, Vol. 45, No. 3, pp. 257–269, 1996.
7. C. Dufaza and Y. Zorian, "On the Generation of Pseudo-Deterministic Two-Patterns Test Sequence with LFSRs," *Proc. of IEEE 1997 European Design and Test Conf. (EURO DTC'97)*, pp. 69–76.
8. C.W. Starke, "Built-In Test for CMOS Circuits," *Proc. of IEEE 1984 Int'l Test Conf. (ITC'84)*, pp. 309–314.
9. B. Wurth and K. Fuchs, "A BIST Approach to Delay Fault Testing with Reduced Test Length," *Proc. of IEEE 1995 European Design and Test Conf. (ET&DC'95)*, pp. 418–423.
10. S.W. Golomb, L.R. Welch, R.M. Goldstein, and A.W. Hales, *Shift Register Sequences*, Aegean Park Press, 1982.
11. C.J. Shi and J.A. Brzozowski, "Cluster-Cover: A Theoretical Framework for a Class of VLSI-CAD Optimization Problem," *ACM Trans. on Design Automation of Electronic Systems*, Vol. 3, No. 1, pp. 76–107, 1998.
12. F. Brglez, D. Bryan, and K. Kozminski, "Combinational Profile of Sequential Benchmark Circuits," *Proc. of IEEE 1989 Int'l Symposium on Circuits and Systems (ISCAS'89)*, pp. 1929–1934.

13. S. Devadas and K. Keutzer, "An Algorithmic Approach to Optimizing Fault Coverage for BIST Logic Synthesis," *Proc. of IEEE 1998 Int'l Test Conf. (ITC'98)*, pp. 164–173.
14. S. Zhang, R. Byrne, and D.M. Miller, "BIST Generators for Sequential Faults," *Proc. of IEEE 1992 Int'l Conf. on Computer Design (ICCD'92)*, pp. 260–263.

**Xiaowei Li** received his B.Eng. and M.Eng. in computer science from Hefei University of Technology, China in 1985 and 1988, respectively. He received Ph.D. in computer science from the Institute of Computing Technology, the Chinese Academy of Sciences in 1991. Dr. Li joined Peking University, China as a Postdoctoral Research Fellow and an Assistant Professor in 1991, and as an Associate Professor in 1993, all with the Department of Computer Science and Technology. In 1997 and 1998, he was a Visiting Research Fellow in the Department of Electrical and Electronic Engineering at the University of Hong Kong. Currently, he is a Visiting Professor in the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. His research interests include VLSI testing, design for testability, built-in self-test, high-level synthesis for testability, design automation of embedded system, software testing and hardware/software co-design. Dr. Li received the Natural Science Prize from the Chinese Academy of Sciences in 1992. He is a member of IEEE, a senior member of Chinese Institute of Electronics, and a member of Fault Tolerant Computing Technical Committee of Chinese Computer Federation.

**Paul Y.S. Cheung** received his B.Sc. (Eng) degree with first class honours in 1973 and his Ph.D. degree in 1978, both in EE from the Imperial College of Science and Technology, University of London. After working for Queen's University of Belfast for two years as an engineer-in-charge of a laboratory, he returned to Hong Kong in 1978 to take up an academic position at the Hong Kong Polytechnic.

He joined the University of Hong Kong as lecturer in 1980 and was promoted to Senior Lecturer/Associate Professor in 1987. He served as the Associated Dean of Engineering from 1991–1994 and has been the Dean of Faculty of Engineering at the University of Hong Kong since 1994. He was the IEEE Asia Pacific Director in 1995–96 and served as the IEEE Secretary in 1997. His research interests include parallel computer architecture, Internet computing, VLSI design and testing, signal processing and pattern recognition.

**Hideo Fujiwara** received the B.E., M.E. and Ph.D. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1969, 1971 and 1974 respectively. He was with Osaka University from 1974 to 1985 and Meiji University from 1985 to 1993, and joined Nara Institute of Science and Technology in 1993. In 1981 he was a Visiting Research Assistant Professor at the University of Waterloo, and in 1984 he was a Visiting Associate Professor at McGill University, Canada. Presently he is a Professor at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. His research interests are logic design, digital systems design and test, VLSI CAD and fault tolerant computing, including high-level/logic synthesis for testability, test synthesis, design for testability, built-in self-test, test pattern generation, parallel processing, and computational complexity. Dr. Fujiwara is the author of *Logic Testing and Design for Testability* (MIT Press, 1985). He received the IEICE Young Engineer Award in 1977, IEEE Computer Society Certificate of Appreciation Award in 1991, Okawa Prize for Publication in 1994, and IEEE Computer Society Meritorious Service Award in 1996. He is an advisory member of IEICE Trans. on Information and Systems and an editor of IEEE Trans. on Computers, J. Electronic Testing, J. Circuits, Systems and Computers, J. VLSI Design and others. He is a fellow of the IEEE, a Golden Core Member of IEEE Computer Society, a member of the Institute of Electronics, Information and Communication Engineers of Japan and a member of the Information Processing Society of Japan.