Military Technical College
Kobry El-Kobba
Cairo, Egypt

12-th International Conference on Aerospace Sciences & Aviation Technology

TWO-PATTERN TEST CAPABILITIES OF AUTONOMOUS LFSR/SR GENERATOR IN PSEUDO-EXHAUSTIVE TESTING

M. H. El-Mahlawy[1], Emad H. Khalil[1], Fawzy Ibrahim[1] and M. H. Abdel-Azeem[1]

ABSTRACT

Testing for delay and CMOS stuck-open faults requires two-pattern tests. Built-in self-test (BIST) schemes are required to comprehensive testing of such faults. BIST test pattern generators for two-pattern testing should be designed to ensure high transition coverage. The test pattern generator (TPG) circuits treated here are not limited to linear feedback shift registers (LFSRs) but include autonomous linear feedback shift register / shift register (LFSR/SR) circuits. It is required to increase the number of each subset of the state variables for complete transition coverage with the optimal test lengths.

In this paper, the two-pattern test capabilities of LFSR/SRs are explored using transition coverage as the metric. The necessary and sufficient conditions to ensure complete transition coverage for LFSR/SRs are derived. The theory developed here identifies all LFSR/SR TPGs that determine the complete transition coverage under any given TPG size constraint. It is shown that LFSRs with primitive feedback polynomials with large number of terms are better for two-pattern testing. Based on the necessary and sufficient conditions, two-pattern testing have been developed. Experiments indicate that TPGs designed using the procedures outlined in this paper obtain high robust path delay fault coverage with the optimal shortest test lengths.

Keywords: Built-in self-test, test pattern generator, pseudo-exhaustive testing, two-pattern testing, linear feedback shift register.

1. INTRODUCTION

The pseudo-exhaustive test retains almost all benefits of an exhaustive test [1-2]. The choice of pseudo-exhaustive test technique depends on whether or not any combinational circuit outputs depend on all of the circuit inputs. If any circuit output depends on all of its inputs, a partitioning (or segmentation) test technique must be used to test these circuits [3]. For circuits with restricted output dependency, the pseudo-exhaustive test techniques provide an alternative test method. The combinational circuit with *n* inputs and *m* outputs is modelled as a direct acyclic graph.

[1] **Egyptian Armed Forces**

The nodes represent gates and the interconnection signals are represented by edges. Each output cone of the circuit forms a sub graph need not be disjoint. The *dependency set*, D_i, of the output cone *i* is considered the set of the primary inputs and the pseudo-primary inputs that feed it directly or affect it through another node. The *dependency,* $|D_i|$, of the output cone *i* is the cardinality of its dependency set. Let *k* be the maximum value among the dependencies of the *m* output cones. The circuit can be characterized as an (*n, m, k*) circuit. The circuit is segmented into *m* output cones, and each cone is tested exhaustively. The test ensures detection of all irredundant combinational faults with a single pattern within individual cones of the circuit without fault simulation. The time required for pseudo-exhaustive testing depends on the sizes of the output cones. Therefore, pseudo-exhaustive testing reduces the testing time to a feasible workable value while retaining many of the advantages of exhaustive testing. Many test pattern generators have been proposed for pseudo-exhaustive testing. Examples are *modified convolved LFSR/SRs* [4], and permuted *convolved LFSR/SRs* [5].

A transistor *stuck-open fault* in a CMOS circuit can convert a combinational circuit under test (CUT) into a sequential one [2, 6]. Detection of these failures requires *two-pattern tests* [7-8]. Proper operation of a digital circuit requires that less propagation delays along paths in the circuit than a specified limit. Some defects often cause propagation delays to fall outside the desired limits. In this case, a delay fault is said to have occurred. A *delay fault* does not affect a circuit's operation at slow speed, but may cause circuit malfunction at clock speed [9-10]. The application of consecutive input patterns is also effective for delay testing of CUTs. The analysis and synthesis of TPG circuits oriented for two-pattern testing are current research subjects.

Testing for delay faults requires two-pattern tests (V_I, V_2). V_I, the initialization pattern, is first applied to initialize the circuit to a certain state at time t_o. At time t_1 and after the signals in the CUT have stabilized, the second input pattern V_2 is applied to sensitize the fault and propagate the effect of the fault to one of the primary outputs along the tested path. The rising or falling transition is propagated from the input of the path under test, along the tested path, to the output of the path. The output state is sampled at time t_2, where t_2 - t_1, is the operating clock of the CUT to determine the existence of these faults [9-10]. In other words, to detect a path delay fault, a two-pattern test is applied that creates and propagates appropriate signal transitions along the path to be tested.

Due to the nature of two-pattern tests, long test sequences are usually required, leading to high cost of testing. Built-in self-test (BIST) provides a simple, low-cost test solution by building test circuitry inside the very large scale integration (VLSI) chip. BIST has come to relieve the difficulties of the testing problems of VLSI circuits [2, 6]. Most BIST schemes employ linear feedback shift registers (LFSRs) as the test pattern generators (TPGs) [11-14]. One important issue in BIST for delay faults is to ensure that sufficient two-pattern tests are applied to the combinational CUT. The capability of a TPG to generate two-pattern tests is measured by the metric transition coverage for each segment. Transition coverage is the number of distinct two-pattern tests applied to a CUT and is less than or equal to 2^{2n} - 1.

A method to explore the two-pattern testing capabilities of LFSR circuits was presented in [13]. The method is very simple since it only needs to calculate ranks of binary matrices according to the main theorem in [13]. Based on the transition matrix of a TPG circuit, a method was shown to derive to what extent distinct transitions occur on a subset of state variables of the TPG circuits. It quantifies the two-pattern test capability of a TPG. For fixed value of *y,* the number *of y*-dimensional *v*-spaces is a polynomial order of *z.* Therefore, for large *z,* the computation time, required to have the transition coverage of all the v-spaces, increases exponentially. The main theorems of [14] are the derivation of the necessary and sufficient conditions for a LFSR tap selection to ensure complete/maximal transition coverage. The number of possible choices of each type is also derived. The possibility of achieving complete/maximal transition coverage depends on the relative size of *w* (the number of TPG stages) and *n* (the number of CUT inputs). If $w \geq 2n$, it is possible to obtain complete transition coverage. If $w < 2n$, only maximal transition coverage is achievable. Experimental results for the TPGs, designed by [14], do not generate test patterns for complete transition coverage with the optimal test lengths.

This paper focuses on designing the new two-pattern test generator for pseudo-exhaustive testing. The necessary and sufficient conditions to increase the complete transition coverage for the new TPG are presented in the optimal test lengths. It is shown that the proposed TPG can achieve high *robust path delay fault coverage* in the optimal shortest test lengths. In addition, the results described in this paper provide basic theory in BIST TPG design for two-pattern pseudo-exhaustive testing. Most practical circuits have multiple outputs and, in many cases, none of the outputs depends on all the circuit inputs. In such cases, the concept of two-pattern pseudo-exhaustive testing can help reduce the test lengths and TPG hardware complexity, without reducing fault coverage.

This paper is organized into five main sections. Section *2* presents necessary and sufficient conditions that an LFSR satisfies complete transition coverage. Section 3 presents the derivation of the contiguous stages of LFSR/SRs for two-pattern testing. The derivation of the non-contiguous stages of LFSR/SRs for two-pattern testing is presented in section 4. Finally, concluding remarks are presented in section 5.

2. CONDITIONS FOR TWO-PATTERN COVERAGE

The state transition of a *w*-stage autonomous LFSR type 1 and type 2 shown in Fig. 1 and Fig. 2 respectively can be defined by a transition matrix. (Fig. 1 and Fig. 2 have c_i 's as binary constants, $c_i = 1$ implies that a connection exists, while $c_i = 0$ implies that there is no connection.)

Let the next state Y and current state *X* of the autonomous LFSR are related by

$$\boldsymbol{Y = T\,X} \qquad \textbf{(1)}$$

where, matrix *T* is a *transition matrix*. There are two forms of matrix *T* according to the type of ALFSR; T_1 for the first type and T_2 for the second type.

$$T_1 = \begin{bmatrix} c_1 & c_2 & \cdots & c_{w-2} & c_{w-1} & 1 \\ 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}, \qquad T_2 = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 & c_1 \\ 0 & 1 & \cdots & 0 & 0 & c_2 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & c_{w-2} \\ 0 & 0 & \cdots & 0 & 1 & c_{w-1} \end{bmatrix}$$

where, (c_i) is either 1 or 0, depending on the existence or absence of a feedback path. *The characteristic polynomial p(x)* of the *transition matrix* is rewritten as

$$p(x) = 1 + c_1 x^1 + c_2 x^2 + \ldots + c_{w-2} x^{w-2} + c_{w-1} x^{w-1} + x^w \quad (2)$$

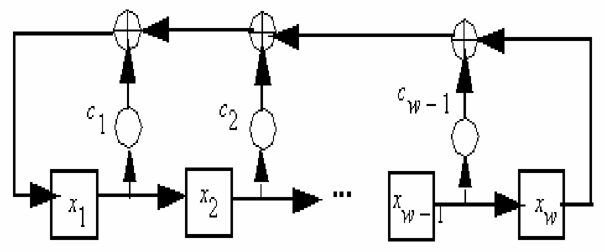

Fig. 1. Type 1 ALFSR.

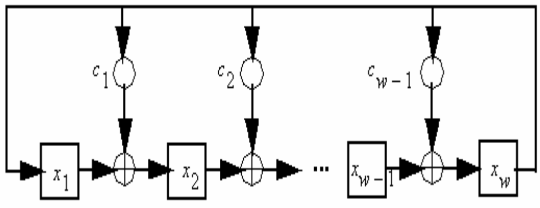

Fig. 2. Type 2 ALFSR.

An ALFSR is a finite state machine. Each state is uniquely determined from the previous state by feedback connection. Thus if a state ever repeats, all the following states will repeat, therefore the sequence of the states is *periodic*. Consider a *w*-stage ALFSR initialized with any nonzero state, then the ALFSR sequence is periodic with a period at most 2^w - 1 possible states. If the sequence, generated by an *w*-stage ALFSR, has period 2^w - 1, it is called a *maximum length sequence*. The characteristic polynomial of a maximum length sequence is called a *primitive polynomial*.

For thorough two-pattern testing, the number of TPG stages *w* required is normally larger than the number of CUT inputs *n,* i.e., $w \geq n$. Hence, only a subset of the TPG outputs are connected to the CUT inputs.

Tapped variables (or simply *taps)* $\mathbf{v} = \{v_1, v_2, \ldots, v_n\}$ are defined as the stages of the TPG whose outputs are connected to the CUT inputs. The remaining TPG stages $\mathbf{u} = \{u_1, u_2, \ldots, u_{w-n}\}$ are called *untapped variables.* Let $\mathbf{X_v}$ and $\mathbf{X_u}$ be the states of the TPG corresponding to the tapped and untapped variables. Then, the next state of the tapped variables, Y_n, can be represented by

$$Y_n = T_n X \quad (3)$$

$$Y_n = X_v T_v + X_u T_u \quad (4)$$

where T_n is the submatrix of *n* rows $\{v_1, v_2, \ldots, v_n\}$ of *T* of the size $n \times w$ and $\mathbf{T_v}$ and $\mathbf{T_u}$ are the submatrices of T_n of the sizes $n \times n$ and $n \times (w - n)$. The submatrix $\mathbf{T_u}$ is constructed from the *n* rows $\{v_1, v_2, \ldots, v_n\}$ of *T,* with the corresponding *n* columns removed. If *r* is the rank of $\mathbf{T_u}$, then there are 2^r distinct transitions from each $\mathbf{X_v}$ state. The rank of $\mathbf{T_u}$ thus determines the transition coverage at the *n* tapped variables, which are connected to the *n* CUT inputs [13].

To obtain maximal transition coverage, the submatrix $\mathbf{T_u}$ must have full rank **r** = min {n, w - n}. There are 2^n possible input combinations for an n-input CUT, each with 2^r possible next states. Maximal transition coverage is thus given by 2^{n+r}. A TPG with w ≥ 2n is needed to provide complete transition coverage, but the n-taps must be carefully selected to avoid dependency between consecutive stages of a TPG. Maximal transition coverage for a w-stage TPG is simply 2^w - 1**.** However, the taps must still be carefully selected to obtain maximal transition coverage.

One main subject of [14] is the identification of necessary and sufficient conditions that an *n*-tap selection must satisfy to obtain complete/maximal transition coverage for an *w*-stage LFSR type 2, assuming that the CUT has a single output.

In pseudo-exhaustive testing, each cone is tested exhaustively. Assume that the largest cone depends on k inputs. Such a circuit can be tested two-pattern pseudo-exhaustively with N tests, where $2^{2k} \le N \le 2^{2n}$. If *k* is small compared to the number of inputs *n*, then the circuit can be tested using a short sequence without decreasing fault coverage. The theoretical results developed in [13-14] can be applied to each cone of the CUT.

2.1 Complete Conditions for Two-Pattern Coverage of a Type 2 LFSR

In this section, necessary and sufficient conditions for a tap selection to ensure complete transition coverage for a type 2 LFSR were derived in [14]. For all the following Lemmas, it is assumed that *n* taps are selected and w ≥ 2n. Complete transition coverage for an *n*-input CUT is achievable only if a *w*-stage TPG with *w ≥ 2n* is used. In this case, the n x *(w - n)* submatrix $\mathbf{T_u}$ has full rank *r* = *n* if all its rows are linearly independent.

LEMMA 1 [14]: *For a w-stage type 2 LFSR, if no two consecutive stages are tapped, then the matrix* $\boldsymbol{T_u}$ *has full row rank*.

LEMMA 2 [14]: *For a w-stage type 2 LFSR, if stage 1 and w are both untapped, and there exists exactly one incidence of consecutive tapped stages* $\varepsilon - 1$ *and* ε *with* $c_{\varepsilon-1} = 1$*, then the matrix* $\boldsymbol{T_u}$ *has full row rank.*

THEOREM 1 [14]: *For an n-input CUT and a w-stage type 2 LFSR with w ≥ 2n, a tap selection provides complete transition coverage if and only if*

1) *no two consecutive stages are tapped, or*
2) *stage 1 and m are untapped and there exists exactly one incidence of consecutive tapped stages* $\varepsilon - 1$ *and* ε *with* $c_{\varepsilon-1} = 1$.

If permutations of CUT inputs are not considered, there are totally $\binom{w}{n}$ possible tap selections. They can be divided into three categories. The tap selections satisfying condition 1 of theorem 1 obtain complete transition coverage *independent* of the feedback polynomial. Such tap selections are *feedback independent.* Other tap selections satisfying condition 2 of theorem 1 are *feedback dependant* because specific coefficients of the feedback polynomial are required to be nonzero in order to achieve complete transition coverage. The remaining tap selections always have less than optimal transition coverage. Determining the number of choices in each category is of practical interest.

LEMMA 3 [14]: *Let* C_{fi} *(*C_{fd}*) be the number of feedback independent (dependent) tap selections for an n-input CUT and a w-stage LFSR with w ≥ 2n.*

Then

$$C_{fi} = \binom{w-n+1}{n} - \binom{w-n-1}{n-2} \quad (5)$$

$$C_{fd} = (w-n-1) \times \binom{w-n-2}{n-2} \quad (6)$$

COROLLARY 1 [14]: *The number of tap selections with complete transition coverage, C, is tightly bounded by* $C_{fi} \le C \le C_{fi} + C_{fd}$.

***Special Case w = 2n*:** For this case, a 2*n*-stage type 2 LFSR with all odd/even stages connected the CUT inputs can generate a two-pattern exhaustive test set for any n-input circuit. However, these are only the two feedback independent tap selections to achieve complete transition coverage in Lemma 3. The following result gives all possible tap selections for the special case *w* = 2*n*.

COROLLARY 2 [14]: *For an n-input CUT and a 2n-stage type 2 LFSR with the feedback polynomial* $p(x) = c_0 + c_1 x + c_2 x^2 + ... + C_{2n} x^{2n}$, *the tap selections to achieve complete transition coverage are*

1) *Select all odd (or all even) stage outputs, or*
2) *Select stages 2, ..., 2i, 2i + 1, ..., 2n- 1 for any i such that* $c_{2i} = 1$.

COROLLARY 3 [14]: *The number of possible ways to connect the n inputs of a CUT to a 2n-stage type 2 LFSR for two-pattern exhaustive testing is given by*

$$\sum_{i=0}^{n} c_{2i} .$$

The summation in Corollary 2 is maximum if $c_{2i} = 1$, for all *i*. Therefore, a type 2 LFSR with many nonzero coefficients of the form c_{2i} offers more choices of tap selections for two-pattern exhaustive testing. This summation can also be obtained by replacing *w* with *2n* in Lemma 3. The maximum numbers of the feedback independent and feedback dependent tap selections that achieve complete transition coverage are 2 and *n* - 1, respectively.

2.2 Complete Conditions for Two-Pattern Coverage of a Type 1 LFSR

In this section, necessary and sufficient conditions for a tap selection to ensure complete transition coverage for a type 1 LFSR are derived in this paper. It is assumed that *n* taps are selected and w ≥ 2n.

LEMMA 4: *For a w-stage type 1 LFSR, if no two consecutive stages are tapped, then the matrix* $\boldsymbol{T_u}$ *has full row rank*.

PROOF: The condition implies that any tapped stage *i* must be preceded by an untapped stage *i* - 1. The unique nonzero entry in column *i* - 1 of the transition matrix is included in $\boldsymbol{T_u}$. Hence, all rows of $\boldsymbol{T_u}$ are linearly independent.

DEFINITION 1: *If there exists an incidence tapped stages ε - 3 and ε with* $c_{\varepsilon-2} = 1$, *then it is called 3-distance tap with connection,* td_{3c}, *and if there exists an incidence tapped stages ε - 3 and ε with* $c_{\varepsilon-2} = 0$, *then it is called 3-distance tap with no connection,* td_{3nc}.

LEMMA 5: *For a w-stage type 1 LFSR and w = 2n, if stage 1 and w are tapped, and there exists only one incidence 3-distance tap with connection,* td_{3c}, *then the matrix* $\boldsymbol{T_u}$ *has full row rank. Also, for w > 2n, if stage 1 and w are tapped, and there* *exists more incidences 3-distance taps with at least exactly one 3-distance tap with connection, td_{3c}, then the matrix $\boldsymbol{T_u}$ has full row rank.*

PROOF: Since stage 1 and *w* are tapped, the row 1 and row *w* are included in $\boldsymbol{T_n}$, and the column 1 and w are removed from $\boldsymbol{T_u}$. Removing column w from $\boldsymbol{T_n}$, removes the nonzero entry in the first row and the existence of the nonzero entry due to the feedback connection of the type 1 LFSR is required. Assume there are one incidence tapped stages *ε - 3* and *ε*. Each of the columns in $\boldsymbol{T_u}$ has a nonzero entry in different rows as shown in Lemma 4 with the nonzero entries in the first row due to the existence of the feedback connections of the type 1 LFSR. The matrix $\boldsymbol{T_u}$ must have full row rank if $c_{\varepsilon-2} = 1$. *For w > 2n*, more incidences *3-distance tap* are possible, the matrix $\boldsymbol{T_u}$ must have full row rank if *at least one 3-distance tap with connection, td_{3c}* exists.

DEFINITION 2: *If there exists an incidence tapped stages ε - 4 and ε with* $c_{\varepsilon-2} = 1$ or $c_{\varepsilon-3} = 1$*, then it is called 4-distance tap with connection, td_{4c}, and if there exists an incidence tapped stages ε - 4 and ε with* $c_{\varepsilon-2} = 0$ and $c_{\varepsilon-3} = 0$*, then it is called 4-distance tap with no connection, td_{4nc}.*

LEMMA 6: *For a w-stage type 1 LFSR and w > 2n, if stage 1 and w are tapped, and there exists an incidence 4-distance tap with connection, td_{4c}, then the matrix $\boldsymbol{T_u}$ has full row rank.*

PROOF: Since stage 1 and *w* are tapped, the row 1 and row *w* are included in $\boldsymbol{T_n}$, and the column 1 and w are removed from $\boldsymbol{T_u}$. Removing column w from $\boldsymbol{T_n}$, removes the nonzero entry in the first row and the existence of the nonzero entry due to the feedback connection of the type 1 LFSR is required. Assume there are an incidence *4-distance tap*. Each of the columns in $\boldsymbol{T_u}$ has a nonzero entry in different rows as shown in Lemma 4 with the nonzero entries in the first row due to the existence of the feedback connections of the type 1 LFSR. The matrix $\boldsymbol{T_u}$ must have full row rank if *4-distance tap with connection, td_{4c} exists*.

THEOREM 2: *For an n-input CUT and a w-stage type 1 LFSR with w ≥ 2n, a tap selection provides complete transition coverage if and only if Lemma 4 or Lemma 5 or Lemma 6 is valid.*

PROOF: Since w ≥ 2n, the matrix $\boldsymbol{T_u}$ must have full row rank. The sufficiency of the theorem is a direct result of Lemma 4, 5, and 6. For necessity, assume that two stages ε - 3 and ε are tapped. The columns of ε - 3 and ε are removed from $\boldsymbol{T_u}$. These removed columns have either all zero entries or nonzero entry in the row 1 only. In order for $\boldsymbol{T_u}$ to still have full row rank, stage 1 or w must be tapped, and $c_{\varepsilon-2} = 1$. Otherwise, the rows of $\boldsymbol{T_u}$ are linearly dependent. Assume that two stages ε - 4 and ε are tapped. The columns of ε - 4 and ε are removed from $\boldsymbol{T_u}$. These removed columns have either all zero entries or nonzero entry in the row 1 only. In order for $\boldsymbol{T_u}$ to still have full row rank, stage 1 or w must be tapped, and either $c_{\varepsilon-2} = 1$ or $c_{\varepsilon-3} = 1$. Otherwise, the rows of $\boldsymbol{T_u}$ are linearly dependent. Next, if stage 1 and stage w are untapped, then tapping stages ε - 3 and ε introduce all zero columns. Also, tapping stages ε - 4 and ε introduce all zero columns. The tap selection does not obtain complete transition coverage.

Example 1: Consider a 4-input CUT and a 8-stage LFSR with primitive polynomial given by $p(x) = 1 + x^2 + x^3 + x^4 + x^8$. The number of possible tap selections, which achieve complete transition coverage for type 1 LFSR according to theorem 2, is 4 and for type 2 LFSR according to theorem 1 is also 4. These tap selections in the case of type 1 LFSR are (1, 3, 5, 7), (2, 4, 6, 8), (1, 3, 6, 8), and (1, 4, 6, 8). The tap selections in the case of type 2 LFSR are (1, 3, 5, 7), (2, 4, 6, 8), (2, 3, 5, 7), and (2, 4, 5, 7).

Let us take one tap selection that achieves complete transition coverage, for example, (1, 4, 6, 8). The following transition matrix *T* is presented with primitive polynomial given by $p(x) = 1 + x^2 + x^3 + x^4 + x^8$.

$$T = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \Rightarrow T = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The following matrix $\boldsymbol{T_n}$ is submatrix of *T* of size 4 × 8. The submatrix $\boldsymbol{T_n}$ is constructed from the 4 rows $\{\mathbf{v_1}, \mathbf{v_4}, \mathbf{v_6}, \mathbf{v_8}\}$ of *T*. The submatrix $\boldsymbol{T_u}$ is constructed from $\boldsymbol{T_n}$ with removing 4 columns {1, 4, 6, 8}. The rank of $\mathbf{T_u}$ is 4. It determines the transition coverage at the 4 tapped variables, which are connected to the 4 CUT inputs.

$$T_n = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \Rightarrow T_u = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A given primitive polynomial *p(x)* of degree *k* may not exercise every output cone, in which case other polynomials of degree *k* + 1 may exercise every output cone.

Example 2: Consider a 4-input CUT and a 9-stage LFSR with primitive polynomial given by $p(x) = 1 + x^3 + x^4 + x^5 + x^7 + x^8 + x^9$. The number of possible tap selections, which achieve complete transition coverage, is 15, for type 1 LFSR according to theorem 2. The tap selections based on

Feedback independent	According to Lemma 4	(2, 4, 6, 8), (1, 3, 5, 7), (3, 5, 7, 9), (2, 5, 7, 9), (1, 3, 5, 8), (1, 3, 6, 8), (1, 4, 6, 8), (2, 4, 7, 9), and (2, 4, 6, 9).
Feedback dependent	According to Lemma 5	(1, 3, 6, 9), (1, 4, 7, 9), (1, 4, 6, 9).
	According to Lemma 6	(1, 5, 7, 9), (1, 3, 5, 9), and (1, 3, 7, 9).

The number of possible tap selections to achieve complete transition coverage is 17 in the type 2 LFSR according to theorem 1. The tap selections based on

Feedback independent	According to Lemma 1	(2, 4, 6, 8), (1, 3, 5, 7), (3, 5, 7, 9), (2, 5, 7, 9), (1, 3, 5, 8), (1, 3, 6, 8), (1, 4, 6, 8), (2, 4, 7, 9), and (2, 4, 6, 9).
Feedback dependent	According to Lemma 2	(3, 4, 6, 8), (2, 4, 5, 7), (2, 4, 5, 8), (3, 5, 6, 8), (2, 5, 6, 8), (2, 4, 7, 8), (3, 5, 7, 8), and (2, 5, 7, 8).

Let us take one tap selection that achieves complete transition coverage, for example, (1, 4, 6, 9). The following transition matrix *T* is presented with primitive polynomial given by $p(x) = 1 + x^3 + x^4 + x^5 + x^7 + x^8 + x^9$.

$$T = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \Rightarrow T = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The following matrix T_n is submatrix of *T* of size 4 × 9. The submatrix $\mathbf{T_n}$ is constructed from the 4 rows $\{\mathbf{v_1}, \mathbf{v_4}, \mathbf{v_6}, \mathbf{v_9}\}$ of *T*. The submatrix $\mathbf{T_u}$ is constructed from T_n with removing 4 columns {1, 4, 6, 9}. The rank of $\mathbf{T_u}$ is 4. It determines the transition coverage at the 4 tapped variables, which are connected to the 4 CUT inputs.

$$T_n = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \Rightarrow T_u = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3. DRIVATION OF THE CONTIGUOUS STAGES OF LFSR/SRs FOR TWO-PATTERN TESTING

Our goal is to design the efficient pseudo-exhaustive TPG for two-pattern testing that generate complete transition coverage for each output cone. It is desirable for the TPG to have connections from the output stages *i* to the input of stages *i* + 1 which can significantly reduce routing overhead and each single shift gives a new test pattern to the CUT inputs. Simple LFSR/SR structures have the desired shift register configuration and therefore lead to low hardware overhead. The approach is compatible with scan path design [6]. Consider a (*w, 2k*) simple LFSR/SR composed of *w* register stages and consisting of an LFSR of degree 2*k*. It is divided into two portions, the first portion is called the LFSR portion with size 2*k*, and the second one is called shift register portion (SR) with size *w - 2k*.

Fig. 3 illustrates a (15, 4) simple LFSR/SR with type 1 LFSR and Fig. 4 illustrates a (15, 4) simple LFSR/SR with type 2 LFSR.

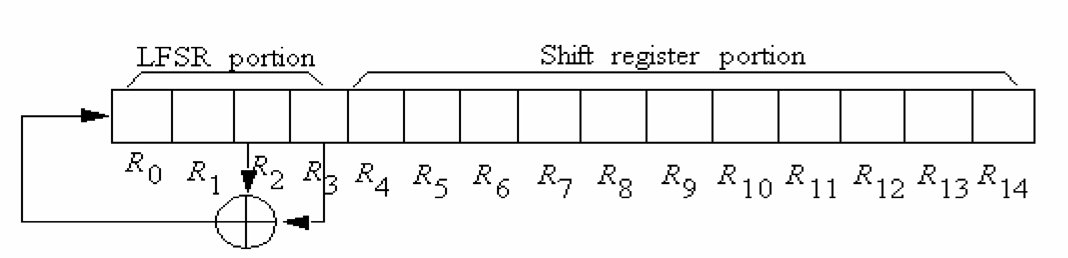

Fig. 3. The residue for (15, 4) LFSR/SR with type 1 LFSR.

A (*w, 2k*) simple LFSR/SR can generate the required patterns in minimal test time for some circuits. For a (*w, 2k*) simple LFSR/SR based on *p(x)*, stage *i* generates a residue x^i *mod p(x)* denoted as R_i which is a polynomial of degree less than 2*k*. A simple LFSR/SR has fixed residues for a given *p(x)*. The theorem presented in [15] indicates that these residues should be linearly independent for the test pattern generation stages to generate all possible combinations of test patterns.

For *p(x)* of degree 2*k*, a (*w, 2k*) simple LFSR/SR in Fig.3, consisting of (*w* – *2k* + 1) different contiguous LFSRs. Residues R_0, R_1, R_2, and R_3 are assigned to the output stages of the LFSR. The outputs assigned to residues R_1, R_2, R_3, *and* R_4, which are linearly independent, are considered the second LFSR with the same primitive polynomial *p(x)*. These outputs produce the same sequences of the LFSR portion (all possible combinations of test patterns) with different initial seed and theorem 2 is applicable. The contiguous positions or the difference between the maximum and minimum indices of the residues that are equal to 2*k,* have identical LFSRs of degree 2*k* with primitive polynomial, *p(x)* [16], and*,* produce the same sequences of test patterns with different initial seeds.

Example 3: For the (15, 4) simple LFSR/SR in Fig. 3, the selected primitive polynomial *p(x)* is $1 + x^3 + x^4$. An initial seed for the LFSR stages is 1000 is seen to be shifted from the left during the initialization phase (the second column in Table 1). The testing phase is in the third column of Table 1. The initial seed for all stages of the simple LFSR/SR is calculated as in Table 1. All patterns generated from all stages of the (15, 4) simple LFSR/SR in the initialization phase and the testing phase will be shown in Table 1.

In Table 1, stage 0 is considered the left-most bit of the pattern and stage 14 is considered the right most bit of the pattern. In the last row of the second column of Table 1, the initial seed of all stages of (15, 4) simple LFSR/SR is the initial pattern in the testing phase which is highlighted. From Table 1, the order of the test pattern sequence of the twelve different contiguous LFSRs is the same with different initial seed. Therefore, the transition matrix for every LFSRs is the same and theorem 2 is valid for each LFSR.

Table 2 illustrates this concept.

Table 1. Initial seed determination for type1 LFSR/SR

Pattern number	**Initialization phase**	***Testing phase***
0	**1000** 00000000000	**1000** 11110101100
1	0**100 0**0000000000	0**100 0**1111010110
2	00**10 00**000000000	00**10 00**111101011
3	100**1 000**00000000	100**1 000**11110101
4	1100 **1000**0000000	1100 **1000**1111010
5	0110 0**1000**000000	0110 0**1000**111101
6	1011 00**1000**00000	1011 00**1000**11110
7	0101 100**1000**0000	0101 100**1000**1111
8	1010 1100**1000**000	1010 1100**1000**111
9	1101 01100**1000**00	1101 01100**1000**11
10	1110 101100**1000**0	1110 101100**1000**1
11	1111 0101100**1000**	1111 0101100**1000**
12	0111 10101100100	0111 10101100100
13	0011 11010110010	0011 11010110010
14	0001 11101011001	0001 11101011001
The initial seed	**1000 11110101100**	**1000 11110101100**

Table 2. Twelve LFSRs generated from type1 LFSR/SR.

Pattern #	***Residue (0,1,2, 3)***	***Residue (1,2,3,4)***	***Residue (2,3,4 5)***	***Residue (3 ,4,5,6)***	***Residue (4, 5,6,7)***	***Residue (5,6,7,8)***	***Residue (6,7,8,9)***	***Residue (7,8,9,10)***	***Residue (8,9,10,11)***	***Residue (9,10,11,12)***	***Residue (10,11,12,13)***	***Residue (11,12,13,14)***
0	1000	0001	0011	0111	1111	1110	1101	1010	0101	1011	0110	1100
1	0100	1000	0001	0011	0111	1111	1110	1101	1010	0101	1011	0110
2	0010	0100	1000	0001	0011	0111	1111	1110	1101	1010	0101	1011
3	1001	0010	0100	1000	0001	0011	0111	1111	1110	1101	1010	0101
4	1100	1001	0010	0100	1000	0001	0011	0111	1111	1110	1101	1010
5	0110	1100	1001	0010	0100	1000	0001	0011	0111	1111	1110	1101
6	1011	0110	1100	1001	0010	0100	1000	0001	0011	0111	1111	1110
7	0101	1011	0110	1100	1001	0010	0100	1000	0001	0011	0111	1111
8	1010	0101	1011	0110	1100	1001	0010	0100	1000	0001	0011	0111
9	1101	1010	0101	1011	0110	1100	1001	0010	0100	1000	0001	0011
10	1110	1101	1010	0101	1011	0110	1100	1001	0010	0100	1000	0001
11	1111	1110	1101	1010	0101	1011	0110	1100	1001	0010	0100	1000
12	0111	1111	1110	1101	1010	0101	1011	0110	1100	1001	0010	0100
13	0011	0111	1111	1110	1101	1010	0101	1011	0110	1100	1001	0010
14	0001	0011	0111	1111	1110	1101	1010	0101	1011	0110	1100	1001

For the (15, 4) simple LFSR/SR in Fig. 4, the LFSR portion has type 2 LFSR with primitive polynomial *p(x)* is $1 + x^3 + x^4$. According, the invention presented in [17] suggested simulating the state of a type 1 LFSR by clocking a type 2 LFSR to produce an output sequence. This sequence is shifted through the shift register. Cascading a type 2 LFSR output sequence into a shift register is the exact equivalent of a type 1 LFSR. The shift register output will contain data corresponding to the state of the type 1 LFSR. Table 3 output sequence illustrates that the shift register output is the exact equivalent of a type 1 LFSR with the primitive polynomial *p*(x)*, where *p*(x)* is the reciprocal primitive polynomial of *p(x)*. The shift register portion has different contiguous type 1 LFSRs. Therefore, the transition matrix for every LFSRs is the same and theorem 2 is valid for each

LFSR. Theorem 1 is valid in the LFSR portion with type 2 LFSR. Table 4 illustrates this concept starting from stage R_0 that is the first stage of the shift register portion.

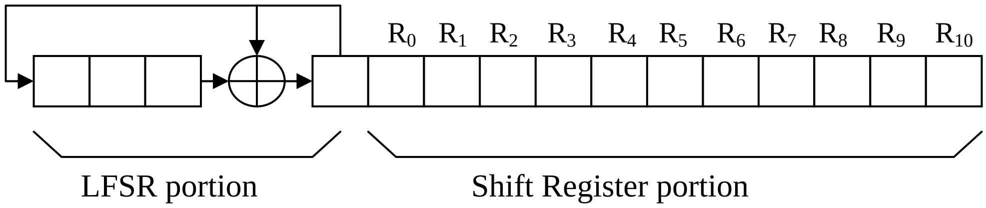

Fig. 4. The residue for (15, 4) LFSR/SR with type 2 LFSR.

Table 3. Initial seed determination for type2 LFSR/SR

Pattern number	Initialization phase	Testing phase
0	**1000** 00000000000	**1000** 10011010111
1	0**100** **0**0000000000	0**100** **0**1001101011
2	00**10** **00**000000000	00**10** **00**100110101
3	000**1** **000**00000000	000**1** **000**10011010
4	1001 **1000**0000000	1001 **1000**1001101
5	1101 1**1000**000000	1101 1**1000**100110
6	1111 11**1000**00000	1111 11**1000**10011
7	1110 111**1000**0000	1110 111**1000**1001
8	0111 0111**1000**000	0111 0111**1000**100
9	1010 10111**1000**00	1010 10111**1000**10
10	0101 010111**1000**0	0101 010111**1000**1
11	1011 1010111**1000**	1011 1010111**1000**
12	1100 11010111100	1100 11010111100
13	0110 01101011110	0110 01101011110
14	0011 00110101111	0011 00110101111
The initial seed	**1000 10011010111**	**1000 10011010111**

Table 4. Twelve LFSRs generated from type2 LFSR/SR.

Pattern #	*LFSR portion*	*Residue (0,1,2, 3)*	*Residue (1,2,3,4)*	*Residue (2,3,4 5)*	*Residue (3 ,4,5,6)*	*Residue (4, 5,6,7)*	*Residue (5,6,7,8)*	*Residue (6,7,8,9)*	*Residue (7,8,9,10)*
0	1000	1001	0011	0110	1101	1010	0101	1011	0111
1	0100	0100	1001	0011	0110	1101	1010	0101	1011
2	0010	0010	0100	1001	0011	0110	1101	1010	0101
3	0001	0001	0010	0100	1001	0011	0110	1101	1010
4	1001	1000	0001	0010	0100	1001	0011	0110	1101
5	1101	1100	1000	0001	0010	0100	1001	0011	0110
6	1111	1110	1100	1000	0001	0010	0100	1001	0011
7	1110	1111	1110	1100	1000	0001	0010	0100	1001
8	0111	0111	1111	1110	1100	1000	0001	0010	0100
9	1010	1011	0111	1111	1110	1100	1000	0001	0010
10	0101	0101	1011	0111	1111	1110	1100	1000	0001
11	1011	1010	0101	1011	0111	1111	1110	1100	1000
12	1100	1101	1010	0101	1011	0111	1111	1110	1100
13	0110	0110	1101	1010	0101	1011	0111	1111	1110
14	0011	0011	0110	1101	1010	0101	1011	0111	1111

Example 4: Consider a 4-input CUT and a 13-stage LFSR/SR with primitive polynomial given by *p(x)* = 1 + x^2 + x^3 + x^4 + x^8 illustrated in Fig. 5. Since the coefficient c_o, c_2, c_4, and c_8 are 1, the number of possible tap selections to achieve complete transition coverage is 4 in the type 1 LFSR according to theorem 2. The total tap selections to achieve the complete transition coverage are 19 in a 13-stage LFSR/SR. (There is five repeated tap selections.) The tap selections are

LFSR portion	*Residues {1, 2, 3, 4, 5, 6, 7, 8, 9}*	*Residues { 2, 3, 4, 5, 6, 7, 8, 9,10}*	*Residues { 3, 4, 5, 6, 7, 8, 9,10, 11}*	*Residues { 4, 5, 6, 7, 8, 9,10, 11, 12}*	*Residues { 5, 6, 7, 8, 9,10, 11, 12, 13}*
(1, 3, 5, 7)	(2, 4, 6, 8)	(3, 5, 7, 9)	(4, 6, 8, 10)	(5, 7, 9, 11)	(6, 8, 10, 12)
(2, 4, 6, 8)	(3, 5, 7, 9)	(4, 6, 8, 10)	(5, 7, 9, 11)	(6, 8, 10, 12)	(7, 9, 11, 13)
(1, 3, 6, 8)	(2, 4, 7, 9)	(3, 5, 8, 10)	(4, 6, 9, 11)	(5, 7, 10, 12)	(6, 8, 11, 13)
(1, 4, 6, 8)	(2, 5, 7, 9)	(3, 6, 8, 10)	(4, 7, 9, 11)	(5, 8, 10, 12)	(6, 9, 11, 13)

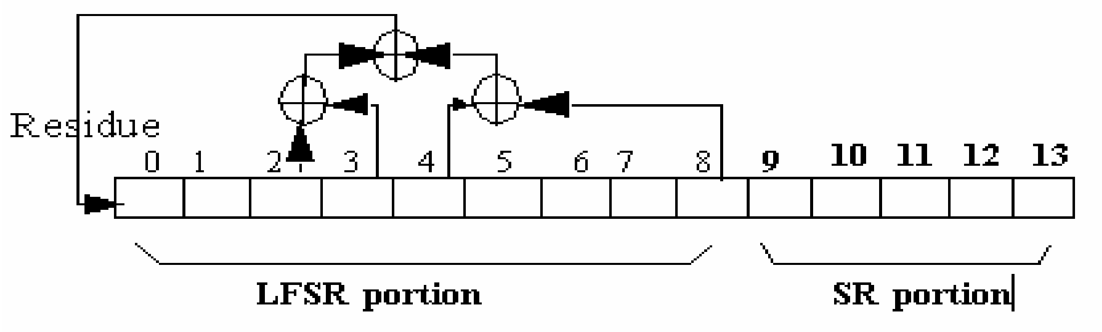

Fig. 5. (13, 8) LFSR/SR with *p(x)* = 1 + x^2 + x^3 + x^4 + x^8.

4. DRIVATION OF THE NON-CONTIGUOUS STAGES OF LFSR/SRS FOR TWO-PATTERN TESTING

The non-contiguous situation is not as simple as its contiguous case. A (*w, 2k*) simple LFSR/SR has $\binom{w}{2k}$ *2k-subsets* of w. For a (*w, 2k*) simple LFSR/SR based on *p(x)*, stage *i* generates a residue $x^i \bmod p(x)$ denoted as R_i which is a polynomial of degree less than 2*k*. In the previous section, the determination of the tap selections that achieve full transition coverage from contiguous *2k-subsets* have been presented. Let set Z_i be non-contiguous *2k-subset* of the LFSR/SR. The set Z_i will be exhaustively tested when the residues assigned it is linearly independent. In this section, the procedure to calculate the tap selections to achieve the complete transition coverage in Z_i.

Let the current state of the non-contiguous output stages of the LFSR/SR, Z, and current state of the LFSR portion, *X,* are related by

$$\boldsymbol{Z = A\,X} \qquad \textbf{(7)}$$

where, matrix *A* is a matrix of size 2*k* × 2*k*, Z is a matrix of size 2*k* × 1, and X is a matrix of size 2*k* × 1.

$$\begin{bmatrix} x_{z_0}(t) \\ x_{z_1}(t) \\ \vdots \\ x_{z_{2k-1}}(t) \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0(2k-1)} \\ a_{10} & a_{11} & \cdots & a_{1(2k-1)} \\ \vdots & \vdots & \vdots & \vdots \\ a_{(2k-1)0} & a_{(2k-1)1} & \cdots & a_{(2k-1)(2k-1)} \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_1(t) \\ \vdots \\ x_{(2k-1)}(t) \end{bmatrix}$$

Then,

$$\boldsymbol{X = A^{-1} Z} \qquad \textbf{(8)}$$

$$\begin{bmatrix} x_0(t) \\ x_1(t) \\ \vdots \\ x_{(2k-1)}(t) \end{bmatrix} = \begin{bmatrix} b_{00} & b_{01} & \cdots & b_{0(2k-1)} \\ b_{10} & b_{11} & \cdots & b_{1(2k-1)} \\ \vdots & \vdots & \vdots & \vdots \\ b_{(2k-1)0} & b_{(2k-1)1} & \cdots & b_{(2k-1)(2k-1)} \end{bmatrix} \begin{bmatrix} x_{z_0}(t) \\ x_{z_1}(t) \\ \vdots \\ x_{z_{2k-1}}(t) \end{bmatrix}$$

Then

$$\boldsymbol{X' = T X} \qquad \textbf{(9)}$$

where, matrix T is a transition matrix of size $2k \times 2k$, X' is a next state matrix of the output stages of LFSR of size $2k \times 1$.

$$\begin{bmatrix} x_0(t+1) \\ x_1(t+1) \\ \vdots \\ x_{(2k-1)}(t+1) \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & \cdots & c_{(2k-1)} & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_1(t) \\ \vdots \\ x_{(2k-1)}(t) \end{bmatrix}$$

Let the next state of the non-contiguous output stages of the LFSR/SR, Z', then

$$\boldsymbol{Z' = S X} \qquad \textbf{(10)}$$

where, matrix S is a matrix of size $2k \times 2k$, $\mathbf{Z}'$ is a matrix of size $2k \times 1$.

$$\begin{bmatrix} x_{z_0}(t+1) \\ x_{z_1}(t+1) \\ \vdots \\ x_{z_{2k-1}}(t+1) \end{bmatrix} = \begin{bmatrix} s_{00} & s_{01} & \cdots & s_{0(2k-1)} \\ s_{10} & s_{11} & \cdots & s_{1(2k-1)} \\ \vdots & \vdots & \vdots & \vdots \\ s_{(2k-1)0} & s_{(2k-1)1} & \cdots & s_{(2k-1)(2k-1)} \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_1(t) \\ \vdots \\ x_{(2k-1)}(t) \end{bmatrix}$$

From equation (8),

$$\boldsymbol{Z' = S A^{-1} Z} \qquad \textbf{(11)}$$

Then, $\boldsymbol{S A^{-1}}$ is the transition matrix of the non-contiguous output stages of the LFSR/SR of size $2k \times 2k$. The tap selections that achieve the complete transition coverage in the subset $\{z_0, z_1, \ldots, z_{(2k-1)}\}$ for k CUT inputs is calculated according the previous conditions. The matrix $\boldsymbol{S A^{-1}}$ will equal to the transition matrix $\boldsymbol{T}$ of the case of contiguous output stages. The case of contiguous output stages, derived in section 3, is considered special case of the non-contiguous output stages.

Example 5: For the (8, 4) simple LFSR/SR in Fig. 6, the selected primitive polynomial *p(x)* is $1 + x^3 + x^4$. Table 5 indicates the residues of each output stages of the (8, 4) simple LFSR/SR. Table 6 indicates the test patterns of each output stages of the (8, 4) simple LFSR/SR and the test patterns of the subset {0, 2, 5, 7}.

It is required to determine the tap selections that achieve full transition coverage of two CUT inputs from non-contiguous output stages of the (8, 4) LFSR/SR.

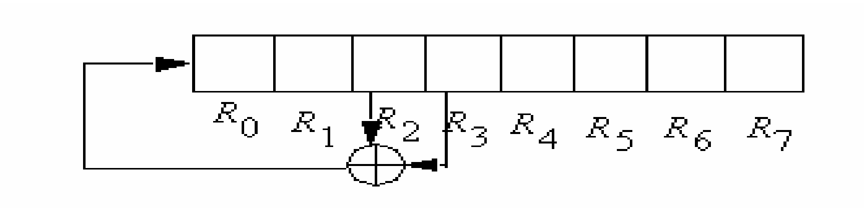

Fig. 6. The residue for (8, 4) LFSR/SR.

Table 5. The residue for (8, 4) LFSR/SR

Residue number R_i	**X^i mod p(x)**	**Polynomial form**	**Binary form**
0	x^0 mod $(1+x^3+x^4)$	1	1 0 0 0
1	x^1 mod $(1+x^3+x^4)$	x^1	0 1 0 0
2	x^2 mod $(1+x^3+x^4)$	x^2	0 0 1 0
3	x^3 mod $(1+x^3+x^4)$	x^3	0 0 0 1
4	x^4 mod $(1+x^3+x^4)$	$1 + x^3$	1 0 0 1
5	x^5 mod $(1+x^3+x^4)$	$1 + x + x^3$	1 1 0 1
6	x^6 mod $(1+x^3+x^4)$	$1 + x + x^2 + x^3$	1 1 1 1
7	x^7 mod $(1+x^3+x^4)$	$1 + x + x^2$	1 1 1 0

Table 6. Test patterns for (8, 4) LFSR/SR

Pattern number	**Initialization phase**	***Testing phase***	{0, 2, 5, 7}
0	1000 0000	1000 1111	1 0 0 0
1	0100 0000	0100 0111	0 1 1 0
2	0010 0000	0010 0011	1 1 0 0
3	1001 0000	1001 0001	0 0 0 1
4	1100 1000	1100 1000	1 1 1 0
5	0110 0100	0110 0100	1 0 1 0
6	1011 0010	1011 0010	1 1 0 1
7	0101 1001	0101 1001	1 1 1 1
8	1010 1100	1010 1100	0 1 0 0
9	1101 0110	1101 0110	0 1 1 1
10	1110 1011	1110 1011	0 0 1 0
11	1111 0101	1111 0101	1 0 1 1
12	0111 1010	0111 1010	0 0 1 1
13	0011 1101	0011 1101	0 1 0 1
14	0001 1110	0001 1110	1 0 0 1
The initial seed	**1000 1111**	**1000 1111**	**1 0 0 0**

The residues assigned to the subset {0, 2, 5, 7} are linearly independent and its output stages generate all possible combination as shown in Table 6. It is required to determine the tap selections that achieve full transition coverage of two CUT inputs from non-contiguous output stages of {0, 2, 5, 7}.

From Table 5, the relationship between the LFSR/SR output stages assigned for the subset {0, 2, 5, 7} and the main signals of LFSR output stages is follows.

$$\begin{aligned} x_0(t) &= x_0(t) \\ x_2(t) &= x_2(t) \\ x_5(t) &= x_0(t) + x_1(t) + x_3(t) \\ x_7(t) &= x_0(t) + x_1(t) + x_2(t) \end{aligned} \qquad \textbf{(12)}$$

Then,

$$\Rightarrow \begin{bmatrix} x_0(t) \\ x_2(t) \\ x_5(t) \\ x_7(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x_0(t) \\ x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} x_0(t) \\ x_2(t) \\ x_5(t) \\ x_7(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_2(t) \\ x_5(t) \\ x_7(t) \end{bmatrix} \qquad \textbf{(13)}$$

Transition matrix and transition equations of the LFSR output stages are follows.

$$\begin{bmatrix} x_0(t+1) \\ x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \Rightarrow \begin{aligned} x_0(t+1) &= x_2(t) + x_3(t) \\ x_1(t+1) &= x_0(t) \\ x_2(t+1) &= x_1(t) \\ x_3(t+1) &= x_2(t) \end{aligned}$$

From equations (7), the transition matrix and transition equations of the LFSR/SR output stages assigned for the subset {0, 2, 5, 7} are follows.

$$\begin{aligned} x_0(t+1) &= x_0(t+1) &&= x_2(t) + x_3(t) \\ x_2(t+1) &= x_2(t+1) &&= x_1(t) \\ x_5(t+1) &= x_0(t+1) + x_1(t+1) + x_3(t+1) &&= x_2(t) + x_3(t) + x_0(t) + x_2(t) = x_0(t) + x_3(t) \\ x_7(t+1) &= x_0(t+1) + x_1(t+1) + x_2(t+1) &&= x_0(t) + x_1(t) + x_2(t) + x_3(t) \end{aligned}$$

$$\Rightarrow \begin{bmatrix} x_0(t+1) \\ x_2(t+1) \\ x_5(t+1) \\ x_7(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_2(t) \\ x_5(t) \\ x_7(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_2(t) \\ x_5(t) \\ x_7(t) \end{bmatrix}$$

Then the transition matrix of the LFSR/SR output stages assigned for the subset {0, 2, 5, 7} is

$$T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad (14)$$

We have $\binom{4}{2}$ possible tap selections. The tap selections that achieve the complete transition coverage in the subset {0, 2, 5, 7} for two CUT inputs is five as follows.

{0, 2}	{0, 5}	{2, 7}	{0, 7}	{3, 4}
$T_n = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$ $\Rightarrow T_u = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$T_n = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ $\Rightarrow T_u = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$T_n = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ $\Rightarrow T_u = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$T_n = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ $\Rightarrow T_u = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$T_n = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ $\Rightarrow T_u = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

The tap selections that do not achieve the complete transition coverage in the subset {0, 2, 5, 7} is one that is {2, 5}. □

The next example will present the efficiency of our approach to get a two-testing pseudo-exhaustive test pattern generator with optimal test set lengths.

Example 6: Consider the (7, 5, 3) CUT with its dependency sets according to the following:

D_0 = {1, 2, 3} D_1 = {1, 2, 4} D_2 = {2, 4, 6} D_3 = {3, 5, 6} D_4 = {3, 5, 7} D_5 = {1, 5, 7}

The dependency, *k*, of this CUT is 3. It is required to select a TPG with $w \geq 2k$. Select a LFSR with degree 6 and primitive polynomial $p(x) = 1 + x + x^2 + x^5 + x^6$. The number of possible tap selections,which achieve complete transition coverage for type 1 LFSR according to theorem 2, is **3** and for type 2 LFSR according to theorem 1 is also **3**. These tap selections in the case of type 1 LFSR are (1, 3, 5), (2, 4, 6), and (1, 4, 6). The tap selections in the case of type 2 LFSR are (1, 3, 5), (2, 4, 6), and (2, 3, 5). These tap selections are not enough to test the CUT.

By using (7, 6) LFSR/SR and according to the design steps presented in section 3, and section 4, the number of possible tap selections, which achieve complete transition coverage is **7**. These seven tap selections are generating from the following table:

Residue assignment to the selected subset	Approach	Tap selections
$R_0, R_1, R_2, R_3, R_4, R_5$	According to section 3	(1, 3, 5), (2, 4, 6), (1, 4, 6)
$R_1, R_2, R_3, R_4, R_5, R_6$	According to section 3	(2, 4, 6), (3, 5, 7), (2, 5, 7)
$R_0, R_1, R_3, R_4, R_5, R_6$	According to section 4	(1, 4, 6), (2, 5, 7), (1, 5, 7), (2, 4, 6), (1, 4, 7)
$R_0, R_2, R_3, R_4, R_5, R_6$	According to section 4	(1, 3, 5), (3, 5, 7), (1, 5, 7), (1, 4, 7)
$R_0, R_1, R_2, R_3, R_4, R_6$	According to section 4	(1, 3, 5), (2, 5, 7), (1, 5, 7), and (1, 4, 7)

By reducing the repeated tap selections, these tap selections in this case are (1, 3, 5), (2, 4, 6), (1, 4, 6), (3, 5, 7), (2, 5, 7), (1, 5, 7), and (1, 4, 7).

Fig. 7 illustrates the assignment of the output stages of the presented TPG to the CUT inputs. The generated test patterns achieve complete transition coverage for each output cone. The test time required for two-pattern pseudo-exhaustive testing is $2^6 - 1$ which is the optimal test set lengths. The design in Fig. 5 requires three extra XOR gates at the location **1, 2,** and **5**.

To solve this problem using the type 2 LFSR presented in [14], it is required to increase the order of the primitive polynomial to 7. The test length according to the design approach in this paper is shorter *(*2^6 vs. 2^7 in this example). Even though the transition coverage in the exhaustive case for the two TPG designs are identical, we conjecture that the fault coverage for a TPG designed in this paper should rise faster with test length than a TPG designed by [14].

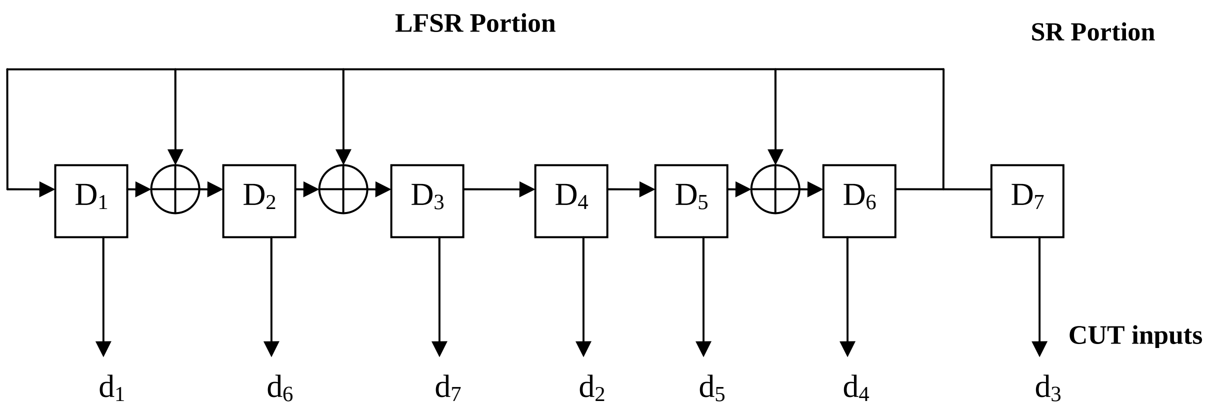

Fig. 7. (7, 6) LFSR/SR design for example 6.

5. CONCLUSIONS AND FUTURE WORK

Testing for delay and CMOS stuck-open faults requires two-pattern tests. This paper presents the designing of the new two-pattern test generator for pseudo-exhaustive testing. BIST test pattern generators for two-pattern testing are designed to ensure complete transition coverage. The TPG circuits treated here are LFSR/SR circuits. It is required to increase the number of possible tap selections to achieve complete transition coverage with the optimal test lengths. In this paper, the two-pattern test capabilities of LFSR/SRs were explored. The necessary and sufficient conditions to ensure complete transition coverage for LFSR/SRs were derived. The theory developed here determines the complete transition coverage under any given TPG size constraint. Primitive polynomials of the LFSRs with large number of terms are better for two-pattern testing.

Simple examples indicate that TPGs designed using the procedures outlined in this paper obtains complete transition coverage with the optimal shortest test lengths. In addition, the results described in this paper provide basic theory in BIST TPG design for two-pattern pseudo-exhaustive testing. Most practical circuits require the concept of two-pattern pseudo-exhaustive testing that reduces the test set lengths and TPG hardware complexity, without reducing fault coverage.

In this paper, the problem of selecting the proper output stages of the (*w, 2k*) LFSR/SR to the CUT inputs that maximize the transition coverage is not addressed. Application of the theory derived in this paper to utilize circuit specific information is the topic of ongoing research.

REFERENCES

[1] E.J. McCluskey, "Verification Testing-A Pseudo-Exhaustive Test Technique", IEEE Trans. Computers, vol. 33, no. 6, pp. 541-546, June (1984).

[2] Mohamed H. El-Mahlawy, Pseudo-Exhaustive Built-In Self-Test for Boundary Scan, Ph.D. thesis, Kent University, U.K., (2000).

[3] Mohamed H. El-Mahlawy, Winston Waller, "A New Segmentation Approach for Pseudoexhaustive Testing of Combinational Circuits", 4th International Conference of the Electrical Engineering, Military Technical College, Egypt, pp. 251-265, Nov. (2004).

[4] Mohamed H. El-Mahlawy, Winston Waller, "An efficient algorithm to design convolved LFSR/SR", IEEE 17th National Radio Science Conference, Minufiya, Egypt, pp. C23 (1-10), Feb. (2000).

[5] Mohamed H. El-Mahlawy, Winston Waller, "A New Single Test Pattern Generator for Pseudoexhaustive Testing", 11th International Conference on Aerospace Sciences & Aviation Technology, Military Technical College, Cairo, Egypt, pp. 989-1002, 17-19 May (2005).

[6] M. Abramovici, M.A. Breuer, and A.D. Friedman, Digital Systems Testing and Testable Design, New York Computer Science Press, (1990).

[7] Rochit Rajsuman, "Digital hardware testing: Transistor-level fault modelling and testing", Artech House, (1992).

[8] Parag K. Lala, "Digital circuit testing and testability," Academic Press, (1997).

[9] Angela Krstic, and Kwang-Ting (Tim) Cheng, Delay Fault Testing for VLSI Circuits, Kluwer Academic Publishers, (1998).

[10] Mukund Sivaraman, and Andrzej J. Strojwas, A unified Approach for Timing Verification and delay Fault Testing, Kluwer Academic Publishers, (1998).

[11] S.W. Golomb, Shift Register Sequences, Laguna Hills, Calif.: Aegean Park Press, (1982).

[12] P.H. Bardell, W.H. McAnney, and J. Savir, Built-In Testfor VLSI: Pseudorandom Techniques, John Wiley & Sons, (1987).

[13] K. Furuya and E.J. McCluskey, "Two-Pattern Test Capabilities of Autonomous TPG Circuits", Proc. IEEE Int'l Test Conf., pp. 704-711, Oct. (1991).

[14] Chih-Ang Chen and Sandeep K. Gupta, "BIST Test Pattern for Two-Pattern Testing- Theory and Design Algorithms", IEEE Transactions on Computers, Vol. 45, NO. 3, pp. 257-269, March (1996).

[15] Z. Barzilai, D. Coppersmith, and A. Rosenberg, "Exhaustive Bit Pattern Generation in Discontiguous Positions with Applications to VLSI Testing", IEEE Trans. Computers, vol. 32, no. 2, pp. 190-194, Feb. (1983).

[16] Rajagopalan Srinivasan, Sandeep K. Gupta, and Melvin A. Breuer, “Novel Test Pattern Generators for Pseudoexhaustive Testing”, IEEE Transactions on Computers, Vol. 49, NO. 11, pp. 1228-1240, November (2000).

[17] Kotzin, Michael D., Wilson, Alan L., “A method of simulating the state of a linear feedback shift register”, European Patent 0 467 640 B1, April (1999).