VLSI DESIGN
2001, Vol. 12, No. 4, pp. 551–562
Reprints available directly from the publisher
Photocopying permitted by license only

An Efficient Test Pattern Generation Scheme for an On Chip BIST

B. K. S. V. L. VARAPRASAD[a,*], L. M. PATNAIK[c,†],
H. S. JAMADAGNI[d,‡] and V. K. AGRAWAL[b,¶]

[a]*Micro Electronics Design Facility,* [b]*Control Systems Group, ISRO Satellite Centre, Bangalore, India 560 017;*
[c]*Microprocessor Applications Laboratory, Department of Computer Science and Automation,*
[d]*Centre for Electronics Design and Technology, Indian Institute of Science, Bangalore, India 560 012*

(Received 15 August 1999; In final form 11 September 2000)

Testing and power consumption are becoming two critical issues in VLSI design due to the growing complexity of VLSI circuits and remarkable success and growth of low power applications (*viz.* portable consumer electronics and space applications). On chip Built In Self Test (BIST) is a cost-effective test methodology for highly complex VLSI devices like Systems On Chip. This paper deals with cost-effective Test Pattern Generation (TPG) schemes in BIST. We present a novel methodology based on the use of a suitable Linear Feedback Shift Register (LFSR) which cycles through the required sequences (test vectors) aiming at a desired fault coverage causing minimum circuit toggling and hence low power consumption while testing. The proposed technique uses circuit simulation data for modeling. We show how to identify the LFSR using graph theory techniques and compute its feedback coefficients (*i.e.*, its characteristic polynomial) for realization of a Test Pattern Generator.

Keywords: Design For Testability (DFT); Built In Self Test (BIST); Test Pattern Generator (TPG); Linear Feedback Shift Register (LFSR)

1. INTRODUCTION

Performance, area, power and testing are some of the most important attributes of complex VLSI systems. With the current reduction in device sizes, it is becoming possible to fit increasingly larger devices onto a single chip or wafer. As chip density increases, the probability of defects occurring in a chip increases as well. The quality, reliability and cost of the product are directly proportional to the degree of testing of the product. Over the time, the focus on testing of Integrated Circuits (ICs) has

*Fax: 91 -80-5266099, e-mail: bksvlvp@isac.ernet.in
†Corresponding author. Fax: 91 -80-3341683, e-mail: lalit@micro.iisc.ernet.in
‡Fax: 91 -80-3341808, e-mail: hsjam@cedt.iisc.ernet.in
¶Fax: 91 -80-5082321, e-mail: agrawal@isac.ernet.in

shifted from the final fabricated ICs to the design stage. In this context, many Design For Testability (DFT) techniques have been developed to ease testing of circuits [1]. The testability of a circuit can be increased with DFT. Testing of the circuit involves the generation of test patterns for the circuit, application of these test patterns to the circuit and analysis of the output response of the circuit. To get the maximum fault coverage of the circuit, one of the simplest approaches is to apply the exhaustive test patterns to the circuit. Many test pattern generation methods like, Random Pattern Testing, Pseudorandom Testing and Pseudo-Exhaustive Testing have been developed [2, 3] aiming to reduce the pattern length without compromising the fault coverage. Built In Self Test (BIST) is a suitable test methodology for highly complex VLSI devices like systems on a chip. Minimization of circuit area and delay has been the main objective in VLSI design for many years. However, low power consumption has recently emerged as another important design objective. Often circuit designers try to reduce the power consumption of a circuit at the cost of increasing the area and delay of the circuit. Power dissipation in CMOS circuits is dominated by the dynamic component [4] which is incurred whenever signals in the circuit undergo a logic transition. One of the several important factors affect-ing power consumption is the choice of circuit technique for logic drivers, latches and flip-flops. A good understanding of how the power consumption occurs in the circuits is essential to design more efficient circuits which consume less power. One method reported [5] suggests to switch off the power to the test area to reduce the power during normal operation of the circuit. However this does not look into the power consumption of the test area during testing. In this paper, a method is proposed by which the power consumption during testing is minimized.

The paper is organized as follows. Section 2 describes the power consumption in CMOS circuits. Section 3 describes problems faced in testing and methods to reduce them. Section 4 gives a description of the methods used to generate test patterns. Section 5 gives the simulation results on three circuits to study the behavior of the fault coverage, transitions in the logic of the internal nodes and capacitance switching in the circuit for various LFSR sequences. Based on the simulation results, a method to compute the characteristic polynomial which gives maximum fault coverage with minimum transitions and capacitance switching is explained in Section 6. Section 7 summarizes the salient features suggested in this paper to reduce the power consumption in the BIST implementation while testing.

2. POWER CONSUMPTION IN CMOS CIRCUITS

The power consumption of digital CMOS circuits is normally divided into three parts, short circuit power P_{Sc}, static power P_s and dynamic power P_d [4].

2.1. Short-circuit Power Consumption

When a static CMOS gate is switched by an input signal with a non-zero rise or fall time, both n-channel and p-channel transistors conduct simultaneously for a short time. During this time, there will be a "short-circuit current", flowing directly between V_{dd} and ground.

2.2. Static Power Consumption

Static power consumption in CMOS circuits is caused by leakage currents of transistors and pn-junctions.

2.3. Dynamic Power Consumption

Dynamic power consumption is related to a node capacitor which is charged and discharged. The average dynamic power dissipation in the circuit is given by,

$$P_{avg} = \frac{1}{2} f_{clk} (V_{dd})^2 \sum_{g=1}^{n} C_{load(g)} \cdot T_{(g)} \qquad (1)$$

where f_{clk} is the clock frequency, V_{dd} is the supply voltage, $C_{load(g)}$ is the load capacitance of gate g, and T(g) is the average number of transitions at gate g per clock cycle.

The total power consumption P of the circuit is given by,

$$P = P_{Sc} + P_s + P_d \quad (2)$$

Dynamic power consumption normally dominates static power consumption and short circuit power consumption. It may range from 80% to 85% of the total circuit power consumption. Once the operating frequency of the circuit f_{clk} is fixed, the parameter which can be optimized to reduce the average dynamic power consumption is $\Sigma C_{load(g)} \cdot T_{(g)}$. In this paper a method to reduce this switching factor for reduction of the dynamic power dissipation in the circuit during testing is proposed.

3. TESTING

Testing falls into a number of categories depending upon the intended goal. **The diagnostic test** is used during the debugging of a chip or board and tries to identify and locate the fault in a failing chip or board. **The functional test** (also called *go/no go* test) determines whether or not a manufactured component is functional. This problem is simpler than the diagnostics test since the analysis and debugging is not involved in this test. Since this test is to be executed on every manufactured die and has a direct impact on the cost (*viz.*, test equipment cost, testing time), it should be as simple and swift as possible. **The parametric test** checks on a number of nondiscrete parameters, such as noise margin, propagation delays and maximum clock frequencies, under a variety of working conditions, such as temperature and supply voltage.

3.1. Some of the Intricacies of the Test Problem

The functionality of a combinational circuit can be verified by exhaustively applying all possible input patterns to its inputs. For an N-input circuit, this requires the application of 2^N patterns. In a sequential circuit, the output of the circuit depends not only on the inputs applied, but also on the value of the state of the internal registers of the circuit. An exhaustive test of this finite state machine requires the application of 2^{N+M} test patterns, where M and N are the number of registers/flip-flops and inputs of the circuit respectively. As N and M increase, the test length expands exponentially. To reduce the test pattern length an alternate approach is required.

By eliminating redundancy and providing a reduced fault coverage, it is possible to test most combinational logic blocks with a limited set of input vectors. To test a given fault in a sequential circuit, the circuit must first be brought to the desired state/initial state before applying the input excitation. One approach adopted in scan test, is to convert the sequential circuit into a combinational one by breaking the feedback loop during the test.

Testability of the circuit, depends on the controllability and observability of the internal nodes of the circuit [6]. Combinational circuits fall under the class of easily observable and controllable circuits, since any node can be controlled and observed in a single cycle. Design For Testability (DFT) [1] is the design effort through which a design can be made an easily testable design by enhancing the controllability and observability. DFT is used to reduce test generation costs, enhance the quality of the tests and hence reduce defect levels. The DFT strategy contains two components:

(1) Provide the necessary *circuitry* so that the test procedure can be swift and comprehensive.
(2) Provide the necessary *test patterns* (excitation vectors) to be employed during the test procedure. For reasons of cost, it is desirable that the test sequence be as short as possible while covering the majority of possible faults.

3.2. Built-in-self-test(BIST)

An attractive approach to testability is having the circuit itself generate the test patterns instead of

using external patterns to test. Even more appealing is a technique where the circuit itself decides if the obtained results are correct. Depending upon the nature of a circuit, this might require the addition of extra circuitry for the generation and analysis of the patterns.

BIST [7] schemes provide advantages such as at-speed testing, easy reuse of the built-in test structure for diagnosis and repair. One salient feature of BIST that makes it stand out in design and test of VLSI is using a divide-and-conquer approach in which every block has its own embedded BIST structure and is controlled by a chip-wide BIST controller. An ideal BIST scheme achieves not only very high fault coverage with minimum area overhead, low or zero performance degradation and minimum test time but also has low power consumption while testing.

4. TEST PATTERN GENERATION (TPG) SCHEMES

There are many ways to generate and apply stimuli for the Circuit Under Test. Most widely used stimuli are the *exhaustive*, *pseudo-exhaustive* and *random* [2, 3] approaches. In the exhaustive approach, the test length is 2^N, where N is the number of inputs to the circuit. The exhaustive nature of the test means that all detectable faults will be detected, given the space of the available input signals. An N-bit counter is a good example of an exhaustive pattern generator. However as N increases the testing time increases exponentially. Normally in a system with N input variables, every function may depend only on a subset of M inputs. Then the test pattern set which contains 2^M vectors gives the same degree of confidence in a system as if it is tested with all 2^N vectors since each function in the circuit is tested exhaustively [3]. As mentioned earlier, the testing time can be reduced by shrinking the test pattern sequence. Many test generation algorithms and methods like PODEM, FAN and TOPS have been developed [1]. An alternate approach is to use random patterns to get maximum fault coverage with minimum test time.

Various types of test generation procedures for synchronous sequential circuits described at the gate level have been developed. Some of these are deterministic branch and bound techniques [1] and are successful in achieving high fault coverage with high computational complexity and others are based on genetic optimization procedures [8]. A test generation scheme MIX reported in [9] combines several test generation approaches to derive test sequences exhibiting very high fault coverage at relatively low CPU times. Using any of the above methods, an effective set of 'k' test vectors (patterns) can be identified. These 'k' patterns can be applied to the CUT in the following ways.

- The 'k' patterns can be stored in a lookup table (ROM) and addressed by a counter.
- By designing a logic circuit for the 'k' patterns using counter design techniques.
- By designing a suitable Cellular Automata.
- By choosing a suitable LFSR which can cycle through the minimum patterns in which all 'k' patterns are included.

The efficiency of the method is evaluated in terms of fault coverage, test application time, power consumption, area overhead, performance penalty/degradation and computational time to arrive at a particular solution.

5. SIMULATION STUDIES OF AN LFSR AS A TPG

Autonomous circuits such as Linear Feedback Shift Registers (LFSR) [1] are used as low-cost-test pattern generators for circuits testable by pseudo-random patterns. An LFSR with different feedback coefficients (characteristic polynomials) started with different initial seeds may yield significantly different pattern lengths. When these patterns are used as test sequence, it yields different fault coverage. To study the relationship among the features such as the fault coverage,

transitions and capacitance switching in a circuit using all possible sequences of an LFSR, fault simulation and circuit simulation have been carried out on three different circuits (*viz.*, ripple adder, Booth multiplier and comparator). The circuits were synthesized using ASIC Synthesizer of Compass tools. The gate level net list has been simulated using QSIM of Compass tools to calculate the number of transitions of each net. Fault simulation has been done using Verifault-XL Cadence tool for the same gate level net list. We considered all the possible characteristic polynomials of an 8-stage LFSR and all possible initial seeds for each characteristic polynomial in the simulation studies. The summary of simulation results is shown in Tables I, II and III. The results shown in these Tables are chosen on the basis of the following criteria:

- Maximum fault coverage achieved in a circuit with minimum number of patterns.
- The results corresponding to the number of patterns where the percentage difference between fault coverage is maximum.
- The results corresponding to the number of patterns where the percentage difference between number of transitions is maximum.

5.1. Simulation Results of a Ripple Adder

An 8-input ripple adder circuit was simulated and a summary of the simulation results is shown in Table I. The simulation results indicate that a fault coverage of 97.3% can be achieved by using patterns of length 5 to 255. It may be observed that, as the pattern length reduces, the number of transitions also reduces. Moreover, for the same

TABLE I Simulation results for an adder

No. of patterns	Minimum fault coverage (%)	Maximum fault coverage (%)	Minimum transitions	Maximum transitions	Minimum capacitance (pF)	Maximum capacitance (pF)
2	49.79	79.30	52	68	4.53	6.32
5	67	**97.30**	61	132	5.63	12.70
9	69	97.30	70	240	6.78	23.06
10	72.40	97.30	74	262	7.13	25.21
12	79.30	97.30	102	284	9.94	27.63
14	91.59	97.30	118	312	11.56	30.05
15	79.30	97.30	138	341	13.71	33.08
85	97.30	97.30	1082	1471	108.73	145.74
127	97.30	97.30	1810	1864	180.77	185.41
255	97.30	97.30	3623	3660	361.83	364.65

TABLE II Simulation results for a multiplier

No. of patterns	Minimum fault coverage (%)	Maximum fault coverage (%)	Minimum transitions	Maximum transitions	Minimum capacitance (pF)	Maximum capacitance (pF)
2	45.79	76.09	99	135	10.07	16.19
5	49.79	94.59	102	272	10.77	33.24
9	77	98.69	157	432	19.75	53.93
10	83	99.09	231	469	28.43	59.39
12	83.5	99.09	265	543	33.75	68.38
14	86.40	99.09	258	606	30.80	76.95
15	89.69	**99.59**	354	658	45.71	82.91
85	99.09	99.59	2469	3111	315.64	394.09
127	**99.30**	99.59	3878	4095	496.28	519.49
255	99.59	99.59	7875	8051	1006.94	1025.50

TABLE III Simulation results for a comparator

No. of patterns	Minimum fault coverage (%)	Maximum fault coverage (%)	Minimum transitions	Maximum transitions	Minimum capacitance (pF)	Maximum capacitance (pF)
2	58.2	73.5	34	35	4.11	4.20
5	60	88.2	33	101	3.90	11.63
9	58.2	97.1	33	170	3.90	20.04
10	58.2	97.6	38	185	4.53	22.29
12	58.8	97.6	43	211	5.05	25.30
14	60.6	**100**	56	234	6.65	28.28
15	61.2	100	75	240	8.81	28.82
85	88.8	100	810	1052	96.79	125.12
127	97.6	100	1314	1459	156.05	175.11
255	100	100	2739	2777	327.37	331.66

pattern length, depending on the pattern sequence produced by the LFSR, the fault coverage and the number of transitions vary. For example, with a pattern length of 5, depending on the pattern sequence, the number of transitions varies between 61 to 132 while the fault coverage varies between 67% to 97.3%. It is observed that a pattern length of 255 also yields the same fault coverage. However, the number of transitions varies between 3623 to 3660 for the pattern length of 255. This behavior of fault coverage and number of transitions with respect to the pattern length is reflected in Figure 1.

5.2. Simulation Results of a Booth Multiplier

Simulation were carried out on an 8-input Booth multiplier circuit. The results shown in Table II, indicate that a fault coverage of 99.59% we can achieve using patterns of various lengths between 15 and 255. It may be observed that, the number of transitions also reduces with reduction of the pattern length. The fault coverage and the number of transitions vary depending on the pattern sequence produced by the LFSR. For instance, consider a pattern length of 15. Depending on the pattern sequence, the number of transitions varies between 354 to 658 while the fault coverage varies between 86.69% to 99.59%. It may further be noted that a pattern length of 255 also yields the same fault coverage with the number of transitions varying between 7875 to 8051. Figure 2 illustrates, this behavior of fault coverage and number of transitions with respect to the pattern length.

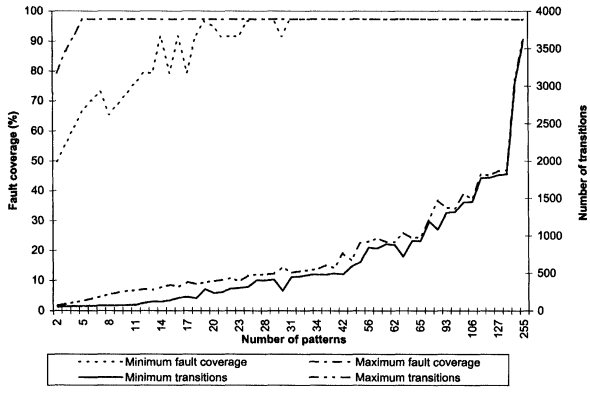


FIGURE 1 Fault coverage and number of transitions for an adder.

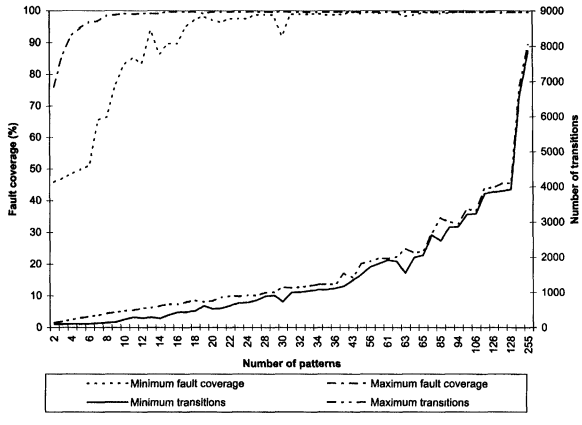


FIGURE 2 Fault coverage and number of transitions for a multiplier.

5.3. Simulation Results of a Comparator

Table III presents the simulation results obtained on an 8-input comparator circuit. The simulation results indicate that a fault coverage of 100% can be obtained by using patterns of length 14 to 255. In this example also, the number of transitions reduces with the pattern length. Moreover, for the same pattern length, depending on the pattern sequence produced by the LFSR, the fault coverage and the number of transitions vary. For example, with a pattern length of 14, depending on the sequence, the number of transitions varies between 56 to 234 and the fault coverage varies between 60.6% to 100%. Note that a pattern length of 255 also yields the same fault coverage, but the number of transitions varies between 2739 to 2777. This behavior of fault coverage and number of transitions with respect to the pattern length is reflected in Figure 3.

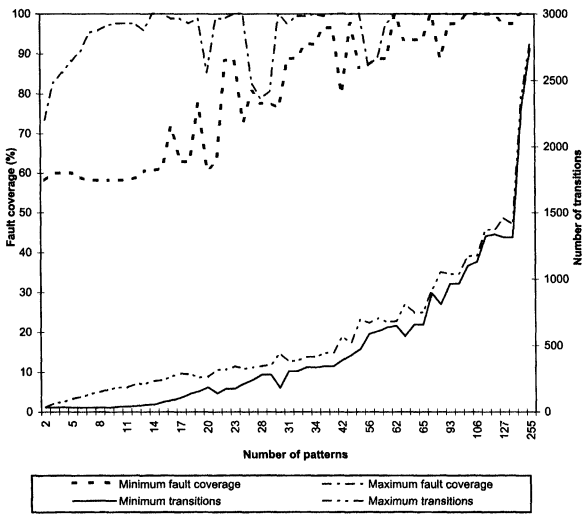


FIGURE 3 Fault coverage and number of transitions for a comparator.

5.4. Summary of Simulation Studies

A summary of these three simulation studies is presented in Table IV. These studies indicate that a method must be adopted to select the test pattern sequence such that the application of this test pattern sequence to the CUT gives the maximum fault coverage, while minimizing the number of patterns and number of transitions. Minimum number of transitions has the obvious advantage of reduced power consumption during testing of the circuit. For example, the number of transitions in the adder circuit for a fault coverage of 97.3% can be reduced from 3623 to 132, with a reduced test length from 255 to 5, by suitably selecting the characteristic polynomial of an LFSR.

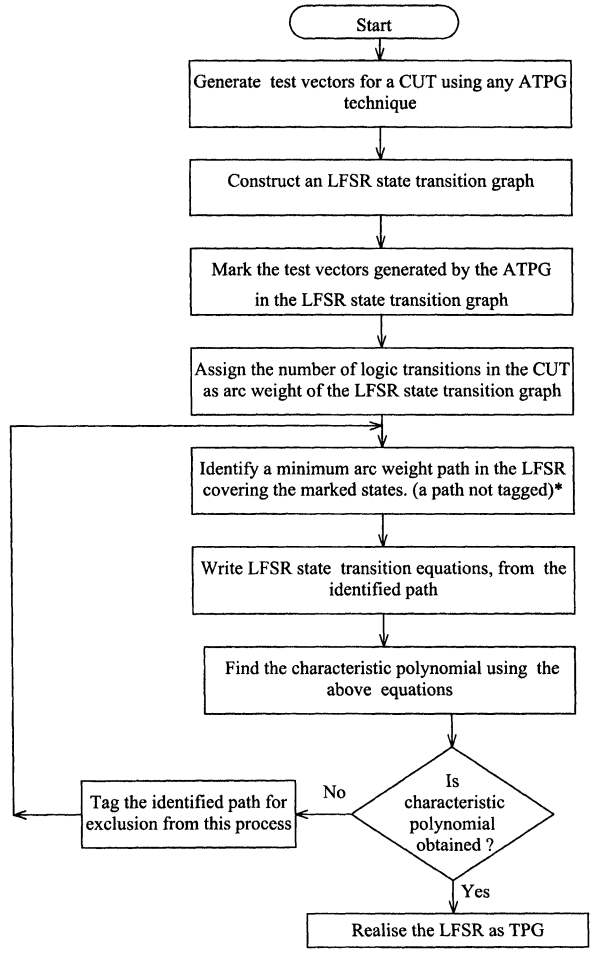
6. PROPOSED METHODOLOGY

Configuration of an LFSR for use as an efficient TPG in an On Chip BIST is a difficult problem. We must obtain a solution, such that the LFSR cycles through a minimum number of states in which all the required test patterns are included, ensuring high fault coverage consuming minimum power during testing. A method based on learning and genetic optimization process has been explained in [10]. In this approach, the authors have considered test length and fault coverage as the parameters of interest but have not aimed at power minimisation during testing. Considering minimization of power consumption during testing as the primary objective we propose a methodology shown in Figure 4.

Testing of an n-input combinational circuit requires an n-stage LFSR, which is used as a

TABLE IV A summary of the simulation results

Circuit	Minimum pattern length	Maximum pattern length	Minimum number of transitions	Maximum number of transitions	Fault coverage (%)	Power saving (%)
Adder	5	255	61	3660	97.3	98.3
Comparator	14	255	56	2777	100.0	97.9
Multiplier	15	255	354	8051	99.5	95.6



* Initially no path is tagged for exclusion.

FIGURE 4 Method to configure an LSFR as TPG.

TPG. Normally, the LFSR behavior can be represented in a directed graph. The vertices of the graph are the different output patterns generated by the LFSR. An edge connects two probable sequences of the output patterns.

Consider a 3-stage LFSR whose behavior is shown in the following graph (See Fig. 5).

For a characteristic polynomial $1+x+x^2+x^3$ and an initial seed of "010", the 3-stage LFSR cycles through the paths {a11, a12}. Similarly depending on its characteristic polynomial and its initial seeds, the LFSR can cycle through the

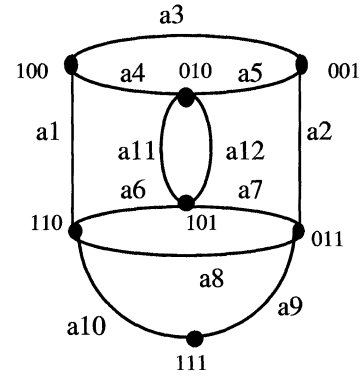


FIGURE 5 LFSR graph.

paths {a3, a4, a5}, {a6, a8, a7}, {a1, a8, a2, a3}, {a1, a8, a7, a12, a5, a3}, {a1, a10, a9, a7, a12, a5, a3} and {a2, a3, a4, a11, a6, a10, a9}.

6.1. Steps to Identify a Path in an LFSR Graph

(1) A set of 'k' test vectors are derived for the CUT using any Automatic Test Pattern Generation (ATPG) technique.
(2) Construct a state transition graph (arc weighted directed graph) for an n-stage LFSR as explained above, wherein the vertices (states of an LFSR) are the test vectors.
(3) Mark the 'k' test vectors, obtained in Step 1, in the above graph.
(4) Compute the total number of signal transitions from logic "0" to "1" in CUT, while the input test vector of the CUT is changed from $\mathbf{S_i}$ to $\mathbf{S_j}$, by simulation of the CUT.
(5) Represent the number of transitions, obtained in Step 4, as arc weights in the graph.
(6) In the above graph, find the path which has minimum arc weight which includes all the marked 'k' test vectors. The minimum arc weight path can be obtained using the algebraic method proposed in [11].

In this method, the n-stage LFSR graph, '**G**', is represented by its adjacency matrix **A**, $\mathbf{A} = [a_{ij}]$ and $a_{ij} = 1$ if arc $(\mathbf{S_i}, \mathbf{S_j})$ exists in **G**

$a_{ij} = 0$ if arc $(\mathbf{S_i}, \mathbf{S_j})$ does not exists in **G**.

A modified variable adjacency matrix, **B**, is formed from matrix **A** in which the element b_{ij} is set to $\mathbf{S_j}$ if there is an arc from vertex $\mathbf{S_i}$ to vertex $\mathbf{S_j}$ and zero otherwise. Let $\mathbf{P_i}$ be the path matrix, representing all possible paths with '**i**' arcs. All elementary paths in the graph **G**, can be obtained by successive multiplication of **B** by $\mathbf{P_i}$.

i.e.,

$$\mathbf{P_{i+1}} = \mathbf{B} * \mathbf{P_i} \tag{3}$$

The adjacency matrix **A** represents the paths with single arc. Hence $\mathbf{P_1} = \mathbf{A}$.

It may be noted, that in the iterative multiplication process, the leading diagonal elements of path matrix $\mathbf{P_i}$ are set to zero, before multiplication, to avoid self loops. The number of matrix multiplications in Eq. (3) to obtain a path which includes all the selected 'k' vectors, varies from $k-1$ to $2^n - 1$.

(7) Find the characteristic polynomial of the LFSR, which cycles through the states represented by the path obtained in Step 6.

6.2. Method to Compute the Coefficients of the Characteristic Polynomial

A set of equations is formed using all the required states sequence of the LFSR. Let $\mathbf{S_1}$, $\mathbf{S_2}$, $\mathbf{S_3}, \ldots, \mathbf{S_k}$ be the various states of the selected sequence produced by LFSR. Let s_{ij} be the jth element of the $\mathbf{S_i}$th state. The relation between two successive states produced by an n-stage LFSR can be expressed in terms of the matrix operation [12]

$$\mathbf{S_i} * \mathbf{C} = \mathbf{S_{i+1}} \tag{4}$$

where

$$\mathbf{C} = \begin{bmatrix} c_1 & 1 & 0 & \ldots & 0 & 0 \\ c_2 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & & & & & \\ c_{n-1} & 0 & 0 & \ldots & 0 & 1 \\ c_n & 0 & 0 & \ldots & 0 & 0 \end{bmatrix}$$

the LFSR transition matrix and $\mathbf{S_i} = (s_{i1}, s_{i2}, \ldots, s_{in})$, $1 < i < 2^n$.

The values of the first column vector in the matrix **C**, *viz.*, $(c_1, c_2, \ldots, c_{n-1}, c_n)$ are the feedback coefficients of the LFSR and are also the coefficients of the characteristic polynomial. In Eq. (4), $\mathbf{S_i}$ represents the current state and $\mathbf{S_{i+1}}$ represents the next state of an n-stage LFSR.

From the matrix operation (4), we have a set of equations

$$\begin{aligned} &s_{11}c_1 + s_{12}c_2 + \cdots + s_{1n}c_n = s_{21} \\ &s_{21}c_1 + s_{22}c_2 + \cdots + s_{2n}c_n = s_{31} \\ &\vdots \\ &s_{k1}c_1 + s_{k_2}c_2 + \cdots + s_{kn}c_n = s_{11} \end{aligned} \tag{5}$$

Now the problem reduces to solving these equations to find out the values of $c_1, c_2, \ldots, c_n$. Although the above set of equations appear to be a set of linear equations, the addition is a modulo 2 operation. Hence, the conventional techniques which are used to evaluate linear system of equations can not be used directly. We can solve these equations by modulo 2 addition of pairs of equations in the set. However, using this approach need not always lead to a solution of the system of equations. Hence, we propose a method of substitution to evaluate these coefficients. In this method we assign "0" as the value for one of the coefficients. The equations are rewritten with the given values and should be checked for consistency as defined below.

DEFINITION A system of equations is said to be consistent if there exist a vector $C = (c_1\ c_2 \cdots c_n)$ to satisfy

$$\sum_{i,j=1}^{n} s_{ij} * c_i = k; \quad \forall s_{ij}, c_i, \quad k \in [0, 1] \tag{6}$$

under modulo 2 addition.

If the resulting set of equations are consistent then we take the coefficient to the assigned value as "0", otherwise make the value of the coefficient as "1". Proceed further in the similar manner for the next coefficient evaluation. In this way all the coefficients can be evaluated. To explain the above process, we take an example of an 8-stage LFSR whose pattern sequences and steps to achieve the solution for coefficients of a characteristic polynomial is explained below.

6.3. Example to Compute the Coefficients of the Characteristic Polynomial

Consider an LFSR of 8 stages that has to cycle through the following twelve sequences.

0	0	1	0	1	0	0	1
1	0	0	1	0	1	0	0
1	1	0	0	1	0	1	0
0	1	1	0	0	1	0	1
1	0	1	1	0	0	1	0
1	1	0	1	1	0	0	1
0	1	1	0	1	1	0	0
0	0	1	1	0	1	1	0
1	0	0	1	1	0	1	1
0	1	0	0	1	1	0	1
1	0	1	0	0	1	1	0
0	1	0	1	0	0	1	1

From Eq. (4), these sequences result in the following system of equations.

		c_3		$+c_5$			$+c_8$	$= 1$
c_1			$+c_4$		$+c_6$			$= 1$
c_1	$+c_2$			$+c_5$		$+c_7$		$= 0$
	c_2	$+c_3$			$+c_6$		$+c_8$	$= 1$
c_1		$+c_3$	$+c_4$			$+c_7$		$= 1$
c_1	$+c_2$		$+c_4$	$+c_5$			$+c_8$	$= 0$
	c_2	$+c_3$		$+c_5$	$+c_6$			$= 0$
		c_3	$+c_4$		$+c_6$	$+c_7$		$= 1$
c_1			$+c_4$	$+c_5$		$+c_7$	$+c_8$	$= 0$
	c_2			$+c_5$	$+c_6$		$+c_8$	$= 1$
c_1		$+c_3$			$+c_6$	$+c_7$		$= 0$
	c_2		$+c_4$			$+c_7$	$+c_8$	$= 0$

This system of equations can be solved for the coefficients $c_1, c_2, \ldots, c_n$, by assigning values either "0" or "1" to the unknown coefficients and checking the resultant system for consistency. While choosing the values, "0" is given priority since the feedback connection can be avoided while realizing the LFSR. Simplification of the above equations is done by modulo 2 operations.

For the above example, the unknown coefficient vector $= (c_1\, c_2\, c_3\, c_4\, c_5\, c_6\, c_7\, c_8)$. By fixing $c_1 =$ "0", we have the following system of equations after simplification of the above set of equations.

$$
\begin{array}{cccccccc}
 & c_3 & & +c_5 & & & +c_8 & = 1 \\
 & & c_4 & & +c_6 & & & = 1 \\
c_2 & & & +c_5 & & +c_7 & & = 0 \\
c_2 & +c_3 & & & +c_6 & & +c_8 & = 1 \\
 & c_3 & +c_4 & & & +c_7 & & = 1 \\
c_2 & & +c_4 & +c_5 & & & +c_8 & = 0 \\
c_2 & +c_3 & & +c_5 & +c_6 & & & = 0 \\
 & c_3 & +c_4 & & +c_6 & +c_7 & & = 1 \\
 & & +c_4 & +c_5 & & +c_7 & +c_8 & = 0 \\
c_2 & & & +c_5 & +c_6 & & +c_8 & = 1 \\
 & c_3 & & & +c_6 & +c_7 & & = 0 \\
c_2 & & +c_4 & & & +c_7 & +c_8 & = 0
\end{array}
$$

The resultant set of equations satisfies $c_1 = 0$ since the above equations are consistent. By successive assignment of the values "0" or "1", to the rest of the coefficients we can get the solution. It may be noted that assigning $c_4 = 0$ after fixing $c_1 = c_2 = c_3 = 0$, we get the following set of inconsistent equations.

$$
\begin{array}{cccccc}
c_5 & & & +c_8 & = 1 & (7.1) \\
 & c_6 & & & = 1 & (7.2) \\
c_5 & & +c_7 & & = 0 & (7.3) \\
 & c_6 & & +c_8 & = 1 & (7.4) \\
 & & c_7 & & = 1 & (7.5) \\
c_5 & & & +c_8 & = 0 & (7.6) \\
c_5 & +c_6 & & & = 0 & (7.7) \\
 & c_6 & +c_7 & & = 1 & (7.8) \\
c_5 & & +c_7 & +c_8 & = 0 & (7.9) \\
c_5 & +c_6 & & +c_8 & = 1 & (7.10) \\
 & c_6 & +c_7 & & = 0 & (7.11) \\
 & & c_7 & +c_8 & = 0 & (7.12)
\end{array}
$$

Equation (7.1) and Eq. (7.6) are contradicting each other. Hence, we assign $c_4 = 1$. By applying the above method, the coefficients of the characteristic polynomial $(c_1\,c_2\,c_3\,c_4\,c_5\,c_6\,c_7\,c_8) = (0\ 0\ 0\ 1\ 0\ 0\ 0\ 1)$ is obtained. The corresponding LFSR is shown in Figure 6.

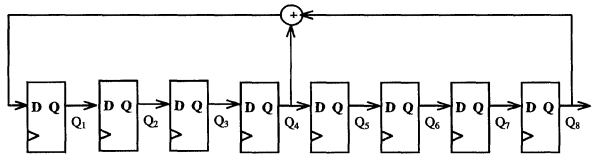


FIGURE 6 LFSR with characteristic polynomial $1 + c_4x^4 + c_8x^8$.

This way one can find LFSR coefficients and realize the circuit for required Test Pattern Generator. It may also be noted that some of the sequences generated by an LFSR can be obtained by more than one characteristic polynomial. Hence, our method of substitutions leads to one of these characteristic polynomials. For example, a 3-stage LFSR cycles through the same sequence 110, 011, 101 for two different characteristic polynomials, namely, $1 + c_1x + c_2x^2$, $1 + c_3x^3$. Though the theory is not mathematically proven here, we have carried out several experiments using the proposed methodology. We have achieved the result with few iterations.

7. CONCLUSIONS

This paper has discussed a suitable method to reduce the power consumption in the BIST implementation during testing. Since LFSRs are being used as TPG in the BIST implementation scheme as a cost effective TPG, we have conducted simulation study of LFSRs with different configurations and initial values for different test sequences. Simulation study on three different circuits has revealed that maximum fault coverage can be achieved with an LFSR sequence of minimum test length while keeping the power consumption of the circuit at a low value. We have shown how to choose a test sequence produced by LFSR to reduce the power consumption using graph theory techniques. Also we have shown how to find the characteristic polynomial for a given test sequence. Even though the approach shown here is convenient for small combinational circuits, large combinational circuits can be partitioned into smaller sub blocks and can be tested in a similar way. In case of sequential circuits, similar approach can be adopted by introducing scan methods. We are currently extending our approach for testing sequential circuits.

References

[1] Abramovici, M., Breuer, M. A. and Friedman, A. D., *Digital Systems Testing and Testable Design*, Rockville, Computer Science, 1990.

[2] Wagner, K. D., Chin, C. K. and McCluskey, E. J., "Pseudorandom Testing", *IEEE Trans. on Computers*, **C-36**(3), 332–343, March, 1987.

[3] Thyagaraju Damarla and Avinash Sathaye, "Application of One-dimensional Cellular Automata and Linear Feedback Shift Registers for Pseudo – Exhaustive Testing", *IEEE Trans. on CAD of ICs and Systems*, **12**(10), 1580–1591, October, 1993.

[4] Jan M. Rabaey and Massoud Pedram, *Low Power Design Methodologies*, Kulwer Academic Publishers, 1996.

[5] Paul S. Levy, "Designing in Power-down Test Circuits", *IEEE Design and Test of Computers*, pp. 31–35, September, 1991.

[6] Fujiwara, H., "Computational Complexity of Controllability/Observability Problems for Combinational Circuits", *IEEE Trans. on Computers*, **39**(6), 762–767, June, 1990.

[7] Agarwal, V. D., "A Tutorial on BIST", *IEEE Design and Test of Computers*, **10**(1), 73–82, March, 1993.

[8] Rajesh, V. and Ajai Jain, "Automatic Test Pattern Generation for Sequential Circuits Using Genetic Algorithms", *Proc. 11th International Conference on VLSI Design'98*, pp. 270–273, January, 1998.

[9] Xijianglin, Irith Pomeranz and Sudhakar M. Reddy, "MIX: A Test Generation for Synchronous Sequential Circuits", *Proc. 11th International Conference on VLSI Design'98*, pp. 456–463, January, 1998.

[10] Irith Pomeranz and Sudhakar M. Reddy, "On Methods to Match a Test Pattern Generator to a Circuit-Under-Test", *IEEE Transactions on VLSI Systems*, **6**(3), 432–444, September, 1998.

[11] Nicos Christofides, *Graph Theory an Algorithmic Approach*, Academic Press Inc., London, 1975.

[12] Golomb, S. W., *Shift Register Sequences*, Aegean Park Press, Laguna Hills, Calif., 1982.

Authors' Biographies

Prof. L. M. Patnaik is a Professor with the Indian Institute of Science, Bangalore, in the Department of Computer Science and Automation. His research interests have been in the areas of Parallel and Distributed Computing, Mobile Computing, Soft Computing, CAD of VLSI Circuits, and Real-Time Systems. In these areas, he has published over 330 papers in refereed International Journals and Conference Proceedings. He is Fellow of the IEEE, and The Third World Academy of Sciences. He has been awarded the IEEE Computer Society's 1999 Technical Achievement Award. He serves on the Editorial Boards of almost a dozen International Journals. He has served in various capacities for over 60 IEEE sponsored Conference Committees. He can be reached at lalit@micro.iisc.ernet.in.

Prof. H. S. Jamadagni is the Chairman of Centre for Electronic Design and Technology (CEDT) at the Indian Institute of Science, Bangalor. His area of work is Computer Networks and Embedded Systems. He is working for a number of industry sponsored Research and Development projects. He is involved in a national level education project. His email address is hsjam@cedt.iisc.ernet.in

Dr. V. K. Agrawal is working in ISRO since 1976. Currently, he is a Group Director, Control Systems Group, ISRO Satellite Centre. He has worked on the design and development of Onboard computer systems for satellite with state-of-art technology. His research interests include Parallel processing, Fault tolerant computing, VLSI design and microprocessor based design.

B. K. S. V. L. Varaprasad received the B.E. degree in Computer Engineering from Andhra University, India in 1987. In 1988, he joined ISRO Satellite Centre and working for Testing of VLSI circuits, design of ASIC's for logic circuits and software development. In 1997, he took admission in Indian Institute of Science to do his research in the area of VLSI testing. His research interests are Design and Development of VLSI circuits with DFT, Computer Architecture and software development for EDA tools.