DS-LFSR: A BIST TPG for Low Switching Activity

Seongmoon Wang, *Member, IEEE* and Sandeep K. Gupta

***Abstract*—A test pattern generator (TPG) for built-in self-test (BIST), which can reduce switching activity during test application, is proposed. The proposed TPG, called *dual-speed LFSR* (DS-LFSR), consists of two linear feedback shift registers (LFSRs), a *slow LFSR* and a *normal-speed LFSR*. The slow LFSR is driven by a *slow clock* whose speed is $1/d$th that of the *normal clock*, which drives the normal-speed LFSR. The use of DS-LFSR reduces the frequency of transitions at the circuit inputs driven by the slow LFSR, leading to a reduction in switching activity during test application. A procedure is presented to design a DS-LFSR so as to achieve high fault coverage by ensuring that patterns generated by it are unique and uniformly distributed. A new gain function and a method to compute its value for each circuit input are proposed to select inputs to be driven by the slow LFSR. Also, a procedure to increase the number of inputs driven by the slow LFSR by combining compatible inputs is presented to further decrease the switching activity. Finally, DS-LFSRs are designed for the ISCAS85 and ISCAS89 benchmark circuits and shown to provide a 13% to 70% reduction in the numbers of load-capacitance weighted transitions with no loss of fault coverage (for stuck-at as well as transition delay faults) and at very slight area overheads.**

***Index Terms*—Built-in self-test (BIST), heat dissipation during test, logic BIST, low-power testing, LFSR, low-power testing, pseudorandom pattern testing, switching activity during test.**

I. Introduction

THE LINEAR feedback shift register (LFSR) is commonly used as a test pattern generator (TPG) in low overhead built-in self-test (BIST). This is due to the fact that an LFSR can be built with little area overhead and used not only as a TPG, which provides high fault coverage for a large class of circuits, but also as an output response analyzer.

A significant correlation exists between consecutive vectors applied to a circuit during its normal operation. This fact is what has motivated several architectural concepts, such as cache memories, and is central to their effectiveness. This is also true for the high-speed circuits that process digital audio and video signals—the inputs to most of whose modules change relatively slowly over time. In contrast, the consecutive vectors of a sequence generated by an LFSR are *proven* to have low correlation. Since the correlation between consecutive vectors applied to a circuit during BIST is significantly lower, the switching activity in the circuit can be significantly higher during BIST than during its normal operation.

Excessive switching activity during test can cause several problems. Foremost, since power dissipation in a CMOS circuit is proportional to weighted switching activity, a circuit under test (CUT) can be permanently damaged due to high temperature that is caused by excessive power dissipation if the switching activity in the circuit during test application is much higher than that during its normal operation. (In this paper, the number of transitions and hazards at a line are *always* weighted by the line's load capacitance. For simplicity, we often drop the modifier "weighted".)

Power dissipated during test application is already influencing the design of test methodologies for practical circuits. For example, it is reported in [1] and [2] that one of their major considerations in test scheduling is the fact that power dissipated during test application is typically significantly higher than that during normal circuit operation (sometimes 100%–200% higher).

The seriousness of excessive power dissipation during test application is exacerbated by trends such as circuit miniaturization for portability and high performance (smaller chips can be placed closer, decreasing interconnect delays). These objectives are typically achieved by using circuit designs that decrease power dissipation and reducing the package size to aggressively match the average power dissipation during the circuit's normal operation. In order to ensure nondestructive testing of such a circuit, it is necessary to either apply test vectors that cause switching activity that is comparable to that during normal circuit operation or remove any excessive heat generated during test using special cooling equipment. The use of special cooling equipment to remove excessive heat dissipated during test application becomes increasingly difficult and costly as tests are applied at higher levels of circuit integration, such as BIST at board and system levels. More importantly, it cannot solve other problems caused by excessive switching activity described next.

It has been observed that metal migration (electromigration) causes the erosion of conductors and subsequent failure of circuits [3], [4]. Since temperature and current density are major factors that determine electromigration rate, elevated temperature and current density caused by excessive switching activity during test application can severely decrease the reliability of circuits under test. This is even more severe in circuits equipped with BIST, since such circuits may be tested frequently.

To test a bare dice, power must be supplied during the period of test through probes that typically have higher inductance than power and ground pins of a circuit package. Hence, the bare dice under test will experience higher power/ground noise that is given by Ldi/dt, where L is the inductance of power and ground line and di/dt is the rate of change of current flowing in power and ground lines. Excessive power/ground noise can

Manuscript received May 15, 2001; revised November 26, 2001. This work was supported in part by the National Science Foundation under CAREER Award MIP-9502300. This paper was recommended by Associate Editor R. Aitken.

S. Wang is with CCRL, NEC USA, Princeton, NJ 08540 USA (email: swang@nec-lab.com).

S. K. Gupta is with Electrical Engineering Systems, University of Southern California, Los Angeles, CA 90089-2562 USA (e-mail: sandeep@boole.usc.edu).

Publisher Item Identifier S 0278-0070(02)05633-6.

erroneously change the logic state of circuit lines causing some good die to fail the test, leading to unnecessary loss of yield.

All the above-mentioned problems are being brought to light by increasing acceptance of at-speed testing. In the past, the tests were typically applied at rates much lower than a circuit's normal clock rate (since only the coverage of stuck-at faults was deemed to be important and slow testers provided an inexpensive way of testing). However, in recent years, aggressive timing has made it essential for the tests to identify slow chips via delay testing. Delay testing is almost imperative for the growing number of circuits manufactured for use in *MCMs*, a fact reflected in the extensive demand for performance-certified die [5], [6]. Circuits are now tested at higher clock rates, if possible, at the circuit's normal clock rate (*at-speed testing*), to achieve coverage of delay faults. Consequently, power dissipation during test application is on the rise and is fast becoming a problem that requires close attention. (It should be noted, however, that at-speed testing is not necessary to achieve high delay fault coverage. It is sufficient to apply a rich set of two-pattern tests and capture the circuit response one normal clock delay after the application of the second pattern in each two-pattern test.)

Recently, several papers that address the problem of reducing power dissipation during built-in self-test have been published [7]–[11]. Methods to reduce power dissipation in deterministic tests are proposed in [12]–[14]. Techniques proposed in [7], [8], [11] require extra gates (AND gates, latches, or pass transistors) to be inserted to block transitions at the TPG outputs. These extra gates are inserted between TPG stages and CUT inputs and hence degrade circuit performance beyond typical BIST methodologies. Furthermore, if many extra gates are required to achieve desired reduction in power dissipation, additional hardware overhead becomes significant.

This paper, which is a significant extension of [10], presents a BIST technique that can reduce switching activity thus mitigating the above-mentioned problems by decreasing, during test application, the magnitudes of power dissipation, average power supply current ($i_{\rm av}$), and current spikes in power and ground lines (di/dt). We assume that the given circuit is sequential but during self-test all flip–flops in the circuit are configured as pattern generators and/or response analyzers. Hence, the CUT is combinational. Unlike [7], [8], and [11], the technique proposed in this paper can achieve significant reduction in switching activity without affecting circuit performance. Simulation results presented in Section V show that the proposed technique can be implemented at low hardware overhead.

The random pattern test length required to achieve high fault coverage is sometimes determined by only a few *hard-to-detect faults* [15]. These faults are also called *random pattern resistant faults* because they escape most randomly generated test patterns. Hence, uniformly distributed test patterns may not achieve high fault coverage for circuits that have many random pattern resistant faults. We have also developed a test pattern generator that can be used to cover random pattern resistant faults without causing excessive switching activity during test application [16]. Due to space limitations, here we focus on the construction of pattern generators that are counterparts of LFSRs that help reduce weighted switching activity.

This paper is organized as follows. In Section II, the architecture of the proposed DS-LFSR TPG is described. In addition, the sequences generated by the proposed TPGs are analyzed and compared with the sequences generated by LFSRs with primitive feedback polynomials. In Section III, a procedure to select inputs to be driven at slow speed is described. In Section IV, a compatibility analysis-based method to increase the number of inputs driven at slow speed without loss of fault coverage is proposed. In Section V, simulation results are reported for ISCAS85 and ISCAS89 circuits. Concluding remarks are finally given in Section VI.

II. Dual-Speed LFSR

Let $n_l(T)$ be the number of transitions at a circuit line l in the time interval $(-T/2, T/2]$. The *transition density* at l, i.e., the number of transitions per second at l, is defined as [17], [18]

$$D(l) = \lim_{T \to \infty} \frac{n_l(T)}{T}. \tag{1}$$

Consider a CUT with m inputs, $p_1, p_2, \ldots, p_m$, which are driven by an m-bit LFSR. Assume that outputs of the LFSR (inputs of the CUT) are not *correlated*, so that the value applied to any input p_i of the CUT is independent of the value applied to any other input p_j, where $i \neq j$. The *Boolean difference* of the Boolean function implemented by line l, v_l, with respect to input p_i is defined as

$$\frac{\partial v_l}{\partial p_i} = v_l|_{p_i=1} \oplus v_l|_{p_i=0} \tag{2}$$

where $\oplus$ denotes an exclusive -OR operation. The transition density of a line l can be redefined in terms of the Boolean difference with respect to each input $\partial v_l/\partial p_i$ and the transition density of each input $D(p_i)$, as

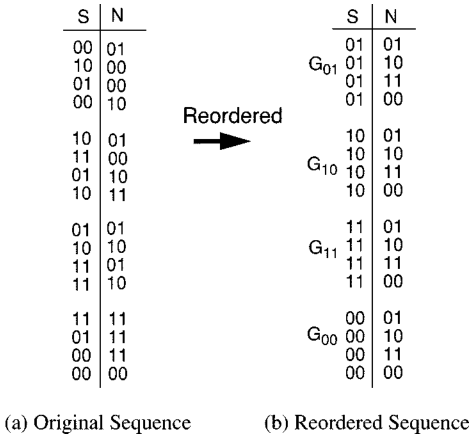
$$D(l) = \sum_{i=1}^{m} P\left(\frac{\partial v_l}{\partial p_i}\right) D(p_i) \tag{3}$$

where $P(\partial v_l/\partial p_i)$ is the probability that the Boolean difference, $\partial v_l/\partial p_i$, evaluates to a 1. Finally, the average power dissipated in the CUT during BIST is given by

$$\begin{aligned} H =& \frac{1}{2} V_{dd}^2 \sum_l C_l D(l) \\ =& \frac{1}{2} V_{dd}^2 \sum_l C_l \sum_{i=1}^{m} P\left(\frac{\partial v_l}{\partial p_i}\right) D(p_i) \end{aligned} \tag{4}$$

where V_{dd} is the power supply voltage and C_l is the load capacitance at line l. Equation (4) shows that the average power dissipated in a CUT during BIST is proportional to the transition density at the inputs of the CUT. In this paper, the average power dissipation during BIST is reduced by lowering the transition density at a subset of the inputs of the CUT; at the same time, the same, or sometimes even higher, fault coverage is obtained.

Let $Y = \{y^1, y^2, y^3, \ldots, y^L\}$ be a sequence of random patterns generated by an m-bit LFSR, where $L = 2^j \leq 2^m$ is the length of test sequence and y^i is an m-bit pattern $\{y_1^i, y_2^i, y_3^i, \ldots, y_m^i\}$, for $i = 1, 2, \ldots, L$. This L pattern



S	N
00	01
10	00
01	00
00	10
10	01
11	00
01	10
10	11
01	01
10	10
11	01
11	10
11	11
01	11
00	11
00	00

	S	N
	01	01
G01	01	10
	01	11
	01	00
	10	01
G10	10	10
	10	11
	10	00
	11	01
G11	11	10
	11	11
	11	00
	00	01
G00	00	10
	00	11
	00	00

(a) Original Sequence (b) Reordered Sequence

Fig. 1. The original and reordered sequence of 16 four-bit patterns.

sequence, Y, can also be written as an $L \times m$ matrix each of whose rows corresponds to a pattern and column i corresponds to the bit sequence appearing at the output of the ith stage of the LFSR. Let $S = \{s^1, s^2, s^3, \ldots, s^L\}$ be a sequence of k-bit patterns consisting of k arbitrary columns of Y and let $N = \{n^1, n^2, n^3, \ldots, n^L\}$ be the sequence consisting of the remaining $(m - k)$ columns of Y not included in S. In the following, we will refer to the sequences S and N as the S *portion* and N *portion*, respectively, of Y. In the following, we will assume that all LFSRs have been modified to generate the all-zero pattern in addition to generating maximal length sequences.

First, consider the case where $L = 2^m$. In this case, S contains exactly $2^{(m-k)}$ repetitions of each of the 2^k distinct k-bit patterns. Hence, Y can be partitioned into 2^k groups such that the S portions of all patterns within each group are identical. Let us first reorder Y such that patterns that belong to the same group are placed next to each other. Next, let us reorder $2^{(m-k)}$ elements in each of the 2^k groups in such a manner that the N-portions of the patterns in the group appear in the same order as they would be generated by an $(m-k)$-bit LFSR. Let the reordered sequence be Y'. Fig. 1 shows an example for $m = 4$, $L = 16$, and $k = 2$. The random pattern sequence shown in Fig. 1(a) is called the *original sequence* Y and the one in Fig. 1(b) is called the *reordered sequence* Y'. Because Y' is obtained by merely reordering the patterns in Y, all patterns in Y are also in Y'. Therefore, the sequence Y' is *equivalent* to Y in the sense that the stuck-at fault coverage obtained by the application of Y' to a combinational CUT is the same as that obtained by applying Y (since, for a combinational CUT, the stuck-at fault coverage is independent of the order in which test patterns are applied). However, Y' will cause less switching activity in the CUT than Y, since the patterns in the S portion of Y' change at a much slower speed, namely, once every $2^{(m-k)}$ patterns. For example, the leftmost bit of the reordered sequence Y' in Fig. 1 changes only two times, while in the corresponding original sequence, Y, the same bit changes eight times.

The sequence of the type Y' can be generated by two independent LFSRs — a k-bit LFSR to generate the S portion and an $(m-k)$-bit LFSR to generate N portion. Both these LFSRs will be designed to generate maximum length sequences including the all-zero pattern. Since the patterns in S portion of Y' change every $2^{(m-k)}$ clocks but the patterns in N portion change every clock, the k-bit LFSR must be driven by a *slow clock* whose period is $2^{(m-k)}$ times as that of the *normal clock*, which drives a $(m-k)$-bit LFSR. We will call the k-bit LFSR the *slow LFSR* and the $(m-k)$-bit LFSR the *normal-speed LFSR* to distinguish from the *original single m-bit LFSR* that generates the sequence Y. Recall that both the slow and normal-speed LFSRs generate k and $(m-k)$-bit all-zero patterns.

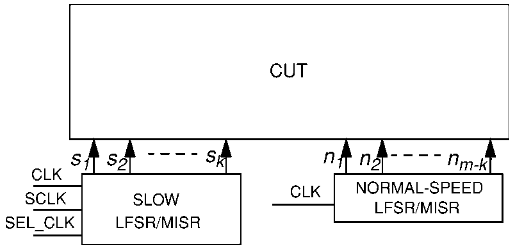


Fig. 2. The proposed DS-LFSR with slow and normal LFSR.

Fig. 2 shows a BIST architecture that is equipped with a slow and a normal-speed LFSR. SCLK denotes the slow speed clock and CLK the normal-speed clock. The slow speed LFSR will be clocked by a clock whose frequency is $1/d$th of that of the normal clock i.e., slow clock speed $=$ normal clock speed$/d$. (To simplify the following discussion as well as the hardware, d will be assumed to be a power of 2.) Note that the slow LFSR has both SCLK and CLK as clock inputs and has a control signal SEL_CLK that selects either SCLK or CLK. SCLK is selected when the slow LFSR is used as a test pattern generator; CLK is selected when the CUT is in the normal mode or when the slow LFSR functions as a multiple input signature register (MISR). The TPG that is composed of the slow and normal-speed LFSR is called *dual speed LFSR* or *DS-LFSR*.

When $L < 2^m$, the sequence of random patterns generated by an m-bit LFSR is not complete in that it does not generate all possible m-bit patterns. Hence, S may have fewer than $2^{(m-k)}$ repetitions of some of the k-bit patterns. The patterns generated are determined by the polynomial and seed of the LFSR. Hence, if $L < 2^m$ and $d \neq 2^{(m-k)}$, the patterns in the *modified sequence* generated by the slow and normal-speed LFSRs may not be equivalent to those in the original sequence. Our objective is not only to reduce switching activity of the CUT during BIST but also to achieve fault coverage that is comparable to, or higher than, that obtained by the sequence generated by the corresponding original LFSR. Hence, we need a quantitative analysis to compare the modified sequence with the original sequence.

A. Uniqueness of Patterns

The first condition to be satisfied by a DS-LFSR to achieve the same fault coverage is that the sequence it generates should not contain any repeated m-bit patterns. (Recall that, for simplicity of notation, L and d are both assumed to be powers of 2.) If $L \leq 2^{(m-k)}$, L consecutive m-bit patterns in the sequence generated by a DS-LFSR with the k stage slow LFSR and $(m-k)$ stage normal-speed LFSR (both of which have primitive feedback polynomials and generate the all-zero pattern) are distinct. This is due to the fact that when $L \leq 2^{(m-k)}$, patterns in N cannot repeat, since an $(m-k)$-bit LFSR that has a primitive feedback polynomial and generates the all-zero pattern has a period $2^{(m-k)}$.

Even though the L pattern sequence generated by a DS-LFSR of above type contains L distinct patterns, this sequence is **not** suitable for pseudorandom testing, since the S portion of each vector contains only one fixed k bit pattern. Next we show that in a more practical case, where $L > 2^{(m-k)}$, hits slow and normal-speed LFSR can collectively generate L distinct patterns, provided that the values of d and k satisfy certain conditions.

Lemma 1: If $L \leq 2^k \times d$ and $d \leq 2^{(m-k)}$, L consecutive m-bit patterns in the sequence generated by a DS-LFSR with the slow and normal-speed LFSRs (both of which have primitive feedback polynomials and also generate the all-zero pattern) are distinct.

Proof: Since a k-stage slow LFSR that has a primitive feedback polynomial and generates the all-zero pattern has a period $2^k \times d$ cycles, when $l \leq 2^k \times d$, the slow LFSR does not exhaust its period. Hence, each distinct k-bit pattern in the S portion appears over exactly one time frame of d consecutive cycles in the entire period with L normal clock cycles.

In order for any two m-bit patterns, V_i and V_j, to be identical, both the k-bit patterns in S portion and the $m-k$-bit patterns in N portion of V_i and V_j must be identical. Hence, in order for the DS-LFSR to generate any repeated patterns, the $m-k$ stage normal-speed LFSR, which has a period of $2^{(m-k)}$ cycles (the normal-speed LFSR also has a primitive feedback polynomial and generates the all-zero pattern), must repeat its period within a time frame of d consecutive cycles during which the k stage slow LFSR generate repeated patterns. However, since $d \leq 2^{(m-k)}$, the normal-speed LFSR does not repeat within d consecutive cycles. Hence, the DS-LFSR does not generate any repeated patterns. Q.E.D.

The sequence generated by a DS-LFSR satisfying conditions specified in Lemma 1 always contains L distinct patterns. However, if a large number is selected for d, only few distinct k-bit patterns in the space of 2^k possible k-bit patterns will be contained in the S-portion of the L patterns. In an extreme case where $d = L$, the S portion contains only one k-bit pattern for the entire L m-bit pattern. In contrast, when $L = 2^k \times d$, the slow LFSR generates all possible $2^k k$-bit patterns and hence the DS-LFSR will generate patterns that are shown ahead to be uniformly distributed in the space of all possible $2^m m$-bit patterns. Due to this property, which is demonstrated next, DS-LFSRs that satisfy the conditions, $L = 2^k \times d$ and $d \leq 2^{(m-k)}$, are the ones that must be used.

The number of DS-LFSR stages, m, is determined by the number of circuit inputs and the test sequence length L is determined by desired fault coverage. Hence, the conditions of Lemma 1 are used to determine the number of stages in the slow LFSR, k, when the clock ratio, d, is given, or to determine d when k is given.

Theorem 1: If $L \leq 2^m$, there always exists d that satisfies the conditions, $L = 2^k \times d$ and $d \leq 2^{(m-k)}$, for any given L, m, and k. Similarly, there always exists k that satisfies the conditions, $L = 2^k \times d$ and $d \leq 2^{(m-k)}$, for any given L, m, and d.

Proof: We will prove only the first statement, since the proof for the second statement is similar. Let us claim that for given values of L, m, and k, the condition, $L = 2^k \times d$, is satisfied only if $d > 2^{(m-k)}$. Substituting the latter condition, i.e., $d > 2^{(m-k)}$ into the former, we get $L > 2^k \times 2^{(m-k)}$. This implies that $L > 2^k \times 2^{(m-k)} = 2^m$. This contradicts the condition that $L \leq 2^m$. Q.E.D.

Theorem 1 shows that for any combination of L, m, and k, or L, m, and d we can design a DS-LFSR that satisfies the condition: $L = 2^k \times d$ and $d \leq 2^{(m-k)}$, i.e., a DS-LFSR that can generate a sequence with uniform distribution. *In the remainder of this paper, we only study DS-LFSRs that satisfy all requirements identified in this section.*

B. Equidistribution of Patterns

The second condition to be satisfied by sequences generated by a DS-LFSR is randomness. Several methods to test randomness of a sequence are described in [19]. These tests include equidistribution test (frequency test), serial test, gap test, poker test, coupon collector's test, permutation test, run test, maximum-of-t test, collision test, serial correlation test, and tests on subsequences. The sequences generated by the proposed DS-LFSR are already shown, by Lemma 1, to pass collision test, which tests for the repetition of any pattern. Since the stuck-at fault coverage for a combinational CUT is independent of the order in which the test patterns are applied, the tests that test randomness of the sequence, such as serial, poker, and run tests, are not useful to quantify the quality of random test patterns for stuck-at faults. Therefore, for our application, the most important test is the equidistribution test. χ^2 test [19] is a method used to test the degree of agreement between distribution of a sample of generated random values and a targeted distribution. Assume that we generate L m-bit patterns where $L \leq 2^m$ and 2^m is divisible by L. Let $I = 2^m/L$ be a natural number. For the generated patterns to be uniformly distributed in the space of 2^m patterns, when expressed as natural numbers, there is only one generated pattern in any interval $[(i-1) \times I, i \times I)$, for $i = 1, 2, \ldots, L$. Therefore, the formula for χ^2 test is given by

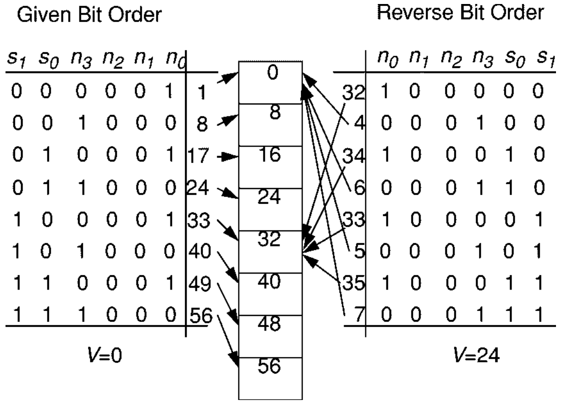
$$V = \sum_{i=1}^{L} \frac{(f_e^i - f_o^i)^2}{f_e^i} \tag{5}$$

where f_e^i is the expected number of patterns that belong to the interval $[(i-1) \times I, i \times I)$, $i = 1, 2, \ldots, L$, and f_o^i is the number of generated patterns belonging to the same interval. For uniform distribution, $f_e^i = 1$, for $i = 1, 2, \ldots, L$.

If test patterns are not uniformly distributed, then there might be some inputs that are assigned the same values in most test patterns. Hence, faults that can be detected only by patterns that are not applied may escape and cause low fault coverage. For example, if inputs p_1 and p_2 of a circuit under test are mostly assigned 00 when a sequence of patterns is applied to C, then all faults that can be detected only when inputs p_1 and p_2 are assigned values other than 00, i.e., 01, 10, and 11, will escape detection. Hence, uniformly distributed test patterns generally achieve higher fault coverage for most circuits than test patterns that are not uniformly distributed.

TABLE I
χ^2 TEST RESULTS

# FFs	# Vect.	V	V (DS-LFSR)/V (LFSR) for various d					
m	L	LFSR	2	4	8	16	32	64
17	1024	1258	.79	.76	.71	.80	.74	.80
17	2048	2074	1.08	1.70	.92	.84	.95	.95
17	4096	3870	1.04	1.14	1.53	1.10	1.02	.99
24	2048	2608	.72	.75	.77	.74	.75	.75
24	4096	4472	.90	.89	.93	.89	.91	.89
24	8192	8498	.83	.93	.95	.92	.95	.96
25	2048	3352	.62	.62	.62	.59	.66	.65
25	16384	17554	.90	.92	.92	.90	.93	.96
25	32768	33978	.95	.96	.94	.98	.98	.95
31	2048	14268	.13	.14	.13	.14	.13	.14
31	8192	18556	.44	.44	.45	.44	.45	.44
31	32768	43920	.74	.73	.73	.76	.75	.74
63	8192	11156	.07	.07	.06	.06	.06	.06
63	32768	127206	.26	.26	.25	.25	.25	.26
63	1048576	1344904	.78	.78	.78	.78	.78	.78



s_1	s_0	n_3	n_2	n_1	n_0	
0	0	0	0	0	1	1
0	0	1	0	0	0	8
0	1	0	0	0	1	17
0	1	1	0	0	0	24
1	0	0	0	0	1	33
1	0	1	0	0	0	40
1	1	0	0	0	1	49
1	1	1	0	0	0	56

	n_0	n_1	n_2	n_3	s_0	s_1
32	1	0	0	0	0	0
4	0	0	0	1	0	0
34	1	0	0	0	1	0
6	0	0	0	1	1	0
33	1	0	0	0	0	1
5	0	0	0	1	0	1
35	1	0	0	0	1	1
7	0	0	0	1	1	1

Fig. 3. Mapping binary bit patterns to ranges of natural numbers.

Table I compares the value of V for the sequences generated by single LFSRs (for short, *LFSR sequences*) with those for sequences generated by DS-LFSRs (for short, *DS-LFSR sequences*).

As described in the preceding paragraph, the binary bit pattern of each pattern generated by DS-LFSRs and single LFSRs is translated into the corresponding natural number to calculate V. Since the use of a different bit order can cause the same pattern to be translated into a different natural number, the value of V for the sequences generated by DS-LFSRs are highly dependent on the positions of bits generated by the slow LFSR in the patterns.

Fig. 3 shows two identical sets of six-bit patterns with different bit orders. Let us assume that the patterns are generated by a DS-LFSR that is composed of a two-bit slow LFSR and four-bit normal-speed LFSR. For convenience, let us call the set of patterns on the left-hand side of the figure the patterns with the *given bit order* and the set of patterns on the right-hand side of the figure the patterns with the *reverse bit order*. The patterns with the given bit order are obtained by considering the bits generated by the slow LFSR as the most significant bits (MSBs) and those generated by the normal-speed LFSR as the least significant bits (LSBs). The patterns with the reverse bit order are obtained by simply reversing the bit positions in the given bit order; hence the MSB (LSB) in the given bit order becomes the LSB (MSB) in the reverse bit order. Since the DS-LFSR has six stages and only eight patterns are generated in this example, the space of $2^6 = 64$ patterns is partitioned into eight groups, each of which has an interval of 8. The integers in the columns next to each set of patterns in the figure (the column to the right of the patterns with the given bit order and that to the left of the reverse bit order) are natural numbers (radix 10) that correspond to each binary bit pattern in the two pattern sets. In the set of patterns with the given bit order, exactly one pattern is mapped onto each of eight intervals, i.e., $V = 0$. However, in the set of patterns with reverse bit order, all eight patterns are mapped to only two intervals, namely [0, 8) and [32, 40), resulting in $V = 24$. This illustrates that χ^2 results for the patterns generated by a DS-LFSR may be highly dependent on the bit order.

A sequence where the bits generated by the slow LFSR are considered MSBs of each vector of the sequence will produce better V (i.e., a smaller V) than a sequence where the bits generated by the slow LFSR are considered LSBs. This is due to the fact that, by our construction, the slow LFSR always generates an exhaustive sequence of k-bit patterns, which are uniformly distributed. Hence, in the experiments shown in Table I, each vector in a sequence generated by DS-LFSRs is shuffled according to a randomly generated order to reduce the effect due to biased bit ordering.

Results are shown in Table I for several sequences of patterns of different lengths (L) and pattern widths (m). The column "*V (LFSR)*" shows values of V for m-stage LFSR sequences with primitive feedback polynomials. The numbers in the columns under the heading "*V (DS-LFSR)/V (LFSR) for various d*" are the ratios of V for DS-LFSR sequences to V for the corresponding LFSR sequences for various values of d. The headings, $2, 4, \ldots, 64$, below this heading denote the d values. Thus, for example, the column labeled "4" shows normalized V values for the sequence of the patterns generated by a DS-LFSR containing a slow LFSR which is driven by a slow clock whose speed is 1/4 of the normal clock.

Normalized Vs for most DS-LFSR sequences are smaller than 1 except when $m = 17$ and $L = 2048$ or 4096. Note that V for the DS-LFSR sequence is smaller than 0.1 when $m = 63$ and $L = 8192$. These results indicate that DS-LFSR sequences are *more uniformly distributed* than LFSR sequences if $L \ll 2^m$. Typical IC chips have hundreds of thousands of flip–flops where many flip–flops are configured as TPG stages in order to achieve high fault coverage. However, in order to finish testing in a reasonable time, test vectors that can be applied to CUTs should be limited to, say, 2^{25}. Hence, 2^m is many orders of magnitude greater than L in typical IC chips. This implies that the DS-LFSR can generate more uniformly distributed sequences and achieve higher fault coverage for most practical IC chips than the traditional LFSR. Variations in V for different clock speeds, i.e., for different values of d, are not significant except when $m = 17$.

A circuit consists of multiple cones. A *cone* is a set of circuit lines and gates that are logically connected to an output. Fig. 4 shows a circuit that has two cones, cone A and cone B. Inputs of cone A and cone B are driven by both a four-stage slow LFSR and an $m - k$ stage normal-speed LFSR. Only two stages out of four stages of the slow LFSR drive cone A while

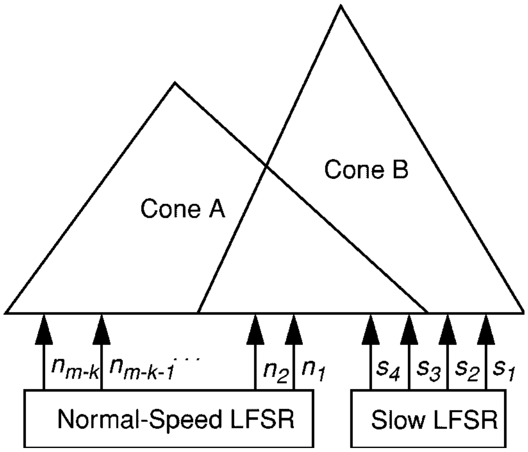


Fig. 4. A circuit with multiple cones.

TABLE II
RATIO OF χ^2 TEST RESULTS FOR PARTS OF VECTORS

clk ratio (d)	# bits from S (g)	ratio of χ^2 test results									
		m=17, L =		m=24, L =		m=25, L =		m=31, L =		m=63, L =	
		1k	2k	4k	8k	2k	32k	2k	32k	32k	1M
2	1	.19	.54	.69	.47	.58	.06	.12	.65	.06	.77
	2	.57	.69	.69	.82	.54	.17	.12	.63	.06	.77
	3	.56	.85	.69	.47	.58	.42	.12	.64	.06	.81
	4	.67	1.50	.75	.89	.65	.91	.12	.70	.06	.77
4	1	.40	4.09	.80	.49	.65	.12	.13	.69	.06	.77
	2	.59	4.06	.79	.48	.62	.35	.11	.67	.07	.78
	3	.37	4.70	.79	.49	.61	.84	.13	.68	.06	.82
	4	.66	3.23	.82	.72	.65	.89	.12	.73	.07	.78

all four stages of the slow LFSR drive cone B. Hence, even though the modified sequence of m-bit patterns is uniformly distributed, a sequence obtained by selecting $j < m$ columns of the sequence, where one or more columns are selected from the sequences generated by the normal-speed as well as the slow LFSR, may not be as uniformly distributed as the sequence obtained by selecting j columns of the sequence generated by an m-stage original LFSR. The sequence of patterns generated by an LFSR is said to hold a good randomness property. Hence, the N portion of the modified sequence that is generated by the normal-speed LFSR can be assumed to hold a good randomness property [20]. Since the slow LFSR always generates $2^k k$-bit patterns, the k-bit patterns in the S portion are also uniformly distributed even though the slow LFSR is driven by the slow clock. Thus, our concern is restricted to the j-bit patterns that consist of parts of both N and S portions.

Table II shows the ratios of χ^2 test results for $(h+g)$-bit portions of the m-bit DS-LFSR sequences to those for $(h+g)$-bit portions of the single LFSR sequences, where for DS-LFSRs, all $h = m - k$ stages of normal-speed LFSR, and $g = 1, 2, 3,$ and four stages of slow LFSRs are selected. Recall that when a number shown in the table is smaller than 1, the χ^2 result for the corresponding DS-LFSR sequence is lower (i.e., better) than that for a sequence generated by a single LFSR. The heading "*clk ratio (d)*" denotes the ratios of the slow clock speed to the normal clock speed. The column labeled "*# bits from S (g)*" denotes the number of bits in the S portion that are concatenated with the N portion, i.e., the g in $(h+g)$ bits. The headings below the values of m $(1k, 2k, 4k,$ or $32k)$ denote the values of L as multiples of 1024. In most cases, $(h+g)$-bit patterns obtained from sequences generated by DS-LFSRs are more uniformly distributed than those obtained from sequences generated by LFSRs. As in Table I, when $2^m \gg L$, sequences generated by DS-LFSRs produce much smaller χ^2 results than those generated by LFSRs.

III. SELECTING INPUTS DRIVEN BY THE SLOW LFSR

Depending on the circuit structure, the transitions at some inputs of a CUT cause more transitions at internal lines than those at other inputs. Therefore, driving at slower speed those inputs that may cause more transitions in the internal circuit will yield greater reductions in switching activity.

We have developed a procedure to select the inputs to be driven at slower speed. A *gain function*, which is based on the transition density formulation, is calculated for each input of the circuit. The transition density of a circuit line l is the sum of the transitions at each input that propagate to line l as shown in (3). Hence, the portion of the transition density of line l due to the transition at a specific input p_i is given by

$$Dp_i(l) = P\left(\frac{\partial v_l}{\partial p_i}\right) D(p_i) \tag{6}$$

where v_l is the Boolean function of line l. The sum of transition densities of all lines in the circuit, weighted by each line's capacitance C_l that can be attributed to the transitions at p_i, is given by

$$Dp_i = \sum_{\forall l} C_l Dp_i(l). \tag{7}$$

The *gain function* Dp_i is computed for all inputs of the circuit and used as the criterion to select the inputs that are to be driven by the slow LFSR of the DS-LFSR. In other words, k inputs that have the greatest Dp_i values are chosen to be driven by the slow LFSR.

The probability that the Boolean difference of v_l with respect to p_i evaluates to a 1, $P(\partial v_l / \partial p_i)$, is derived from the signal probability of each line using a procedure similar to what is used to calculate detection probability in [15]. An auxiliary AND gate is introduced for each path from p_i to l, which is in the transitive fanout of p_i. The *controlling value* of a gate is the value which, when applied to an input of a gate, determines the value of the output of the gate independent of the values applied to its other inputs. The controlling value of an AND gate is a 0 and the controlling value of an OR gate is a 1. If the controlling value of a gate that is on a path from input p_i to l is 1, all inputs of the gate that are not on the path are directly connected to the inputs of the auxiliary AND gate. Otherwise, all such inputs are connected to the inputs of the auxiliary AND gate through inverters. Hence, the probability that a path from p_i to l is sensitive to p_i is the probability that the output of the auxiliary AND gate evaluates to a 1. The outputs of the auxiliary AND gates, which are introduced for each path from p_i to l, are connected to the inputs of an auxiliary OR gate. Finally, the probability that $P(\partial v_l / \partial p_i)$ evaluates to a 1 is the probability that the output of the auxiliary OR gate evaluates to a 1.

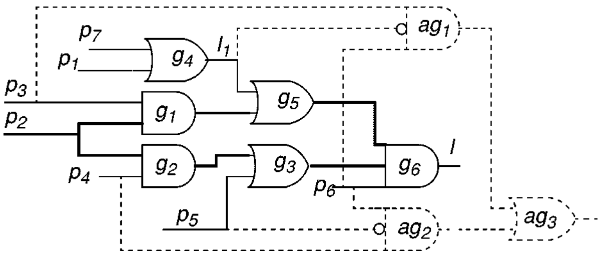


Fig. 5. Inserting auxiliary gates.

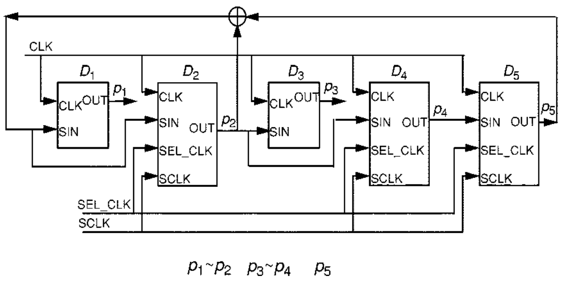


Fig. 6. An example slow LFSR.

Fig. 5 shows an example of deriving $P(\partial v_l/\partial p_2)$ by using auxiliary gates. Since there exist two paths from p_2 to l, (p_2, g_1, g_5, g_6, l) and (p_2, g_2, g_3, g_6, l), two auxiliary AND gates, ag_1 and ag_2, are inserted, one for each path. Signal lines that are on the paths from p_2 to l are denoted by thick lines. Hence, signal lines l_1 and p_5 that are inputs of OR gates along the paths (p_2, g_1, g_5, g_6, l) and (p_2, g_2, g_3, g_6, l) are connected to inputs of auxiliary AND gates through inverters and signal lines, p_3, p_4, and p_6, that drive AND gates along the two paths, are connected to inputs of auxiliary AND gates directly. A transition at p_2 can propagate to l when the output of auxiliary AND ag_1 evaluates to a 1, i.e., $p_3 = 1$, $l_1 = 0$, and $p_6 = 1$ or the output of auxiliary AND gate ag_2 evaluates to a 1, i.e., $p_4 = 1, p_5 = 0$, $p_6 = 1$. Hence, the probability that $P(\partial v_l/\partial p_w)$ is given by the probability that the output of the auxiliary OR gate ag_3 evaluates to a 1.

IV. Merging Compatible Inputs

Assume that L random patterns generated by a combination of slow and normal-speed LFSR are applied to the CUT. Recall that the slow clock speed is $1/d$th of that of the normal clock. Then, the number flip–flops in the slow LFSR required to avoid repetition of any pattern and to generate uniformly distributed patterns is given by $k = \log_2(L/d)$ that can be derived from the condition $L \leq 2^k \times d$. Note that this value of k is determined by L and d, without taking the structure of the CUT into consideration. Hence, if the CUT has many inputs and L is small, the number of inputs that are driven by the outputs of the slow LFSR will be low compared with the number of inputs that are driven by the normal-speed LFSR. This means that most lines of the CUT will be driven by the normal-speed LFSR. Therefore, for such circuits, the reduction in the number of transitions obtained by driving k inputs at the slow speed may not be significant.

If no output of a circuit is driven by input p_i as well as $p_j (i \neq j)$, then p_i and p_j are said to be *compatible* [21] and denoted by $p_i \sim p_j$. The testability of the cones driven by p_i is independent of the value of p_j and *vice versa*. Hence, a set of compatible inputs can be merged into a *test signal* and driven, during test application, by the same output of an LFSR without any loss of test coverage. This implies that the actual number of inputs driven by the slow LFSR can be greater than k when it drives test signals comprised of multiple inputs.

In order to take merged inputs into consideration, the gain function Dp_i given by (7), which was calculated for each input, is now calculated for each set of inputs in a test signal, or a clique, where a clique [22] consists of inputs that are compatible to each other. The modified gain function is given by

$$D_{\Psi_a} = \sum_{\forall p_i \in \Psi_a} \sum_{\forall l} C_l P\left(\frac{\partial v_l}{\partial p_i}\right) \qquad (8)$$

where l denotes a line in the circuit and Ψ_a a set of inputs in test signal a. The outputs of the k-bit slow LFSR are connected to k test signals instead of k compatible inputs that have the greatest gain function values. Let k' denote the total number of inputs driven by the slow LFSR. Thus $k' > k$, if any multiple input test signals are driven by the slow LFSR. In this case, only $m - k'$, instead of $m - k$, inputs are driven by the normal-speed LFSR after compatible inputs are merged. The problem of identifying test signals by combining compatible inputs and selecting test signals to be connected to the outputs of the slow LFSR can be formulated as a maximum clique problem [22]. Since the maximum clique problem is a well-known NP-complete problem, a heuristic that is based on the *Kernighan–Lin bipartitioning algorithm* [23] is developed to select the clique that has the greatest gain function for the circuits with many inputs and outputs. Since the inputs in a clique must be disjoint to the inputs in other cliques, after a clique with the greatest gain function value is selected, all inputs in the selected clique are removed from all unselected cliques. This is repeated until k cliques are selected.

Fig. 6 shows a slow LFSR with $k = 3$. (Even though this register has five stages, as will become clear later, $k = 3$.) Two pairs of inputs, (p_1, p_2) and (p_3, p_4), are compatible (denoted by $p_1 \sim p_2$, $p_3 \sim p_4$). Note that the inputs of the flip–flops whose outputs are connected to inputs belonging to a clique are driven by the output of the same flip–flop. In Fig. 6, the inputs of D_1 and D_2 are both connected to the LFSR feedback signal $(p_2 \oplus p_5)$ and the inputs of D_3 and D_4 are both connected to p_2. Hence, the output value of D_1 is always identical to that of D_2 and the output of D_3 is always identical to that of D_4. Hence, the LFSR generates only three independent bit patterns even though it has five flip–flops. The flip–flops, D_2, D_3, and D_5, have two clock inputs, *SLCK* and *CLK*, and a control input *SEL_CLK*. Hence, these *dual clocked flips–flops*, such as D_2, are more expensive than the *normal flip–flops*, such as D_1 and D_3, which have only a single clock, *CLK*. If we do not merge inputs, five dual clocked flip–flops are required to drive five inputs at slow speed. However, merging (p_1, p_2) and (p_3, p_4) into test signals, only three dual clocked flip–flops and two normal flip–flops are used in the slow LFSR shown in Fig. 6.

In summary, by merging compatible inputs, we can increase the number of inputs driven by the slow LFSR, which is determined by the length L and clock ratio d, without any loss of testability. The second advantage of merging inputs is a reduction in the number of the dual clocked flip–flops hence a reduction in area overhead.

Some circuits have few compatible inputs. In those circuits, the number of inputs that can be driven by the slow LFSR will be low, resulting in small reduction in the number of transitions. However, even though two inputs p_i and p_j are not compatible, if merging p_i and p_j does not decrease the testability of the circuit, we can merge p_i and p_j. This idea has not been pursued in this paper but is the subject of ongoing research.

The procedure to design a DS-LFSR can now be described.

1) For an m-input CUT, perform fault simulation using patterns generated by an m-stage LFSR to determine the length of the test sequence L, to be applied. L is the test length which either achieves desired fault coverage or after which the sequence does not detect any fault for a predefined number of consecutive patterns.
2) Select a clock ratio d. (In all the following experiments, we selected $d = 4$.)
3) Calculate the number of stages in the slow LFSR $k = \log_2(L/d)$.
4) Find compatible inputs, combine them into sets, and select k sets of inputs to be driven by the slow LFSR using the gain function. The number of inputs to be driven by the normal-speed LFSR is then $m - k'$, where k' is the total number of inputs in the k sets of inputs connected to the slow LFSR.

V. Simulation Results

Fault simulations and true value simulations (to count the number of transitions and potential hazards) were performed on ISCAS85 and ISCAS89 benchmark circuits. Table III compares the numbers of transitions caused and transition delay as well as stuck-at fault coverages achieved by the DS-LFSR sequence with those for the original LFSR sequence. (Recall that the term “number of transitions” is being used as a short-form for “number of weighted transitions,” since our methodology weights the number of transitions at each line with the line’s load capacitance.) The slow clock speed used for all the circuits is 1/4 of the normal clock speed. The column labeled L shows the number of patterns in both sequences, m shows the number of inputs (sum of primary and state inputs for ISCAS89 and primary inputs for ISCAS85) of the circuit, and *SFC* and *TFC* stand for stuck-at and transition delay fault coverage, respectively. The columns under the headings *LFSR* and *DS-LFSR* show data for patterns generated by the original LFSR and DS-LFSR, respectively. For both TPGs, the columns entitled *# haz.* and *# trans.* show the average number of circuit lines that can potentially have hazards and the average number of transitions in the circuits, per test pattern, during the application of these sequences. The number of transitions is counted under the zero-delay model. For the DS-LFSR, the numbers in parenthesis under this column denote the ratios of the numbers of transitions for the DS-LFSR sequences to those for the original LFSR sequences. The column labeled k under the heading DS-LFSR displays the number of test signals that are driven by the slow LFSR, in other words, the number of stages in the slow LFSR. Note that these numbers are smaller than the numbers of inputs driven by the slow LFSR, k', for most circuits, because multiple compatible inputs are merged into test signals. The column labeled $m - k'$ shows the number of inputs driven by the normal-speed LFSR.

Since k is determined solely by values of L and d, the number of dual clocked flip–flops required to construct a DS-LFSR is low even in circuits that have many inputs. For example, even though 1011 inputs of s38584 are driven at slow speed, only 18 (k) dual clocked flip–flops are required. This shows that hardware overhead required to implement a DS-LFSR is very low.

The reductions in the average numbers of transitions range from 13% to 70%. As expected, large reductions in the average numbers of transitions occur in the circuits many of whose inputs are driven by the slow LFSR. For example, 11 out of 14 inputs of s1488 are driven by the slow LFSR, where the largest reduction in the average number of transitions occurs. The reductions for ISCAS85 benchmark circuits are not as significant as those for ISCAS89 benchmark circuits. This is primarily due to the fact that most ISCAS85 benchmark circuits do not have many compatible inputs. Furthermore, the lengths of the sequences applied to these circuits are short compared with the number of inputs of these circuits; since these circuits are easily tested with random patterns, long test sequences are not necessary. Since k, the number of stages in the slow LFSR, is determined by the test length and clock ratio, the number of inputs that are driven by the slow LFSR is much smaller than that of inputs that are driven by the normal-speed LFSR of the DS-LFSR.

Since the average numbers of transitions are counted under the zero-delay model, we report the average numbers of circuit lines that can possibly have hazards to predict the number of transitions under general-delay model. The reductions in these numbers are greater than those in the average numbers of transitions, for most circuits. This indicates that the reductions in the average numbers of transitions under general-delay model will be at least as great as those under zero-delay model.

For almost all circuits, the stuck-at fault coverages obtained by the DS-LFSR and LFSR are very similar. (s420, for which DS-LFSR achieves higher fault coverage, and s838, for which LFSR coverage is higher, are some of the notable exceptions.)

Table III also compares the transition delay fault coverages (TFC) obtained by applying the original sequences with those obtained by applying the DS-LFSR sequences. DS-LFSR sequences achieve higher (by 0.5% or more) fault coverage for 12 circuits and LFSR sequences achieve higher fault coverage for 9 circuits and for remaining 11 circuits, DS-LFSR and LFSR sequences achieve almost the same fault coverage. These results show that the sequences generated by the proposed DS-LFSRs can achieve transition delay fault coverages comparable to those provided by sequences generated by the original LFSRs.

Since the slow LFSR is driven by a slow clock, gates that are driven by only the slow LFSR are not exercised at-speed. However, the transition delay fault coverage reported is accurate, since the response at the outputs of the CUT is captured one normal clock period after the application of each distinct pattern at its inputs.

Consider a scenario where one is interested beyond delay testing and intends to cover unmodeled faults via at-speed

TABLE III
SIMULATION RESULTS

			LFSR				*DS-LFSR*						
CKT	*L*	m	# *tran.*	# *haz.*	*SFC*	*TFC*	k	k'	$m - k'$	# *tran.*	# *haz.*	*SFC*	*TFC*
c880	16384	60	275	205	99.94	95.11	12	27	33	200(.73)	120	100	97.10
c1355	16384	41	392	631	99.71	96.97	12	12	29	337(.86)	579	99.71	96.97
c1908	16384	33	617	551	99.48	97.56	12	12	11	462(.82)	367	99.69	97.43
c2670	65536	157	866	781	89.24	83.62	14	31	126	662(.76)	330	88.53	83.17
c3540	32768	50	1031	1199	96.30	90.41	13	13	37	679(.70)	616	96.38	89.52
c5315	32768	178	1888	1490	99.42	96.47	13	87	91	1540(.81)	949	99.42	96.94
c6288	32768	32	1941	3404	99.46	99.18	13	13	19	1695(.87)	3215	99.46	99.18
c7552	32768	206	2623	2815	96.27	94.84	13	22	184	2250(.86)	1589	95.40	94.53
s208	2048	19	52	18	100.0	88.70	9	10	9	19(.37)	4	99.76	86.06
s344	2048	24	116	51	100.0	94.04	9	20	4	48(.41)	14	100.0	92.60
s349	4096	24	119	54	99.43	93.41	10	21	3	44(.37)	14	99.43	93.27
s386	2048	13	134	38	100.0	78.76	9	10	3	46(.34)	10	100.0	78.11
s420	65536	35	92	33	92.26	80.36	14	15	20	42(.46)	7	94.65	85.12
s444	16384	24	153	57	97.52	86.60	12	22	2	50(.33)	13	97.52	88.62
s510	2048	25	153	57	100.0	90.39	9	11	14	50(.33)	15	100.0	90.49
s526	8192	24	205	27	99.14	86.79	11	21	3	70(.34)	8	99.62	87.07
s641	16384	54	216	108	98.51	96.31	12	39	15	106(.49)	38	98.59	95.21
s713	65536	54	229	146	93.83	90.74	14	41	13	104(.45)	55	93.76	89.48
s820	65536	23	328	61	100.0	77.99	14	16	7	101(.31)	16	100.0	81.95
s832	32768	23	334	63	98.98	76.74	13	14	9	106(.32)	16	98.98	80.17
s838	131072	67	167	55	87.43	73.27	15	16	51	114(.68)	17	83.35	69.87
s953	8192	45	211	79	99.59	94.16	11	35	10	96(.45)	22	98.36	89.91
s1196	32768	32	357	160	99.88	85.59	13	15	17	121(.34)	43	99.87	90.56
s1238	65536	32	371	184	93.82	79.01	15	21	11	194(.52)	94	94.08	83.10
s1423	16384	91	514	235	98.98	95.61	12	41	50	290(.56)	114	99.02	95.36
s1488	4096	14	551	223	99.70	79.13	10	11	3	168(.30)	52	100.0	83.80
s1494	4096	14	554	224	99.16	78.55	10	11	3	170(.31)	52	99.47	83.23
s5378	32768	214	1671	753	98.71	93.09	13	113	101	1016(.61)	361	98.60	95.08
s9234	65536	247	2952	1512	89.32	72.75	15	131	116	1573(.53)	632	89.51	81.40
s13207	262144	700	4766	1690	97.98	87.98	17	497	203	2968(.62)	909	98.09	88.73
s15850	524288	611	5610	3446	93.75	86.85	18	342	269	3729(.66)	2143	93.74	83.66
s38584	524288	1464	13752	5674	94.56	88.31	18	1011	453	10069(.73)	3527	94.51	88.50

testing. First note that a large portion of a fanout cone that is typically driven by the slow LFSR is also driven by the normal LFSR (as illustrated in the circuit shown in Fig. 4) and many gates in the portion may still be exercised at-speed. We also can exercise at-speed the gates that are driven by only the slow LFSR (e.g., gates that are in the nonoverlapping portion of cone A of the circuit shown in Fig. 4) by clocking the slow LFSR by the normal speed clock for a short period of time between two normal sessions when the slow LFSR is clocked by the slow clock $1/d$ speed of the normal clock speed. However, the period during which the slow LFSR is clocked by the normal clock should be short enough not to risk damaging the circuit under test.

VI. CONCLUSION

A BIST TPG, which can reduce switching activity during test application, is proposed. The reduction in switching activity is achieved by lowering the transition densities at selected inputs. The proposed TPG, called DS-LFSR, consists of two LFSRs, a slow LFSR and a normal-speed LFSR. The slow LFSR is driven by a slow clock whose speed is $1/d$th that of the normal clock that drives the normal-speed LFSR, thereby, lowering transition densities at inputs driven by the slow LFSR. The DS-LFSR is designed in such a way that the generated patterns are all unique and uniformly distributed to achieve high fault coverage. The empirical analysis using χ^2 tests demonstrates that the DS-LFSR generated sequences are more uniformly distributed than the sequences generated by single LFSRs with primitive feedback polynomials.

The inputs to be driven by the slow LFSR are selected using a gain function whose value is computed for all inputs. The gain function of an input denotes the sum of load-capacitance weighted transition density values of the circuit lines in its transitive fanout. k inputs that have the greatest gain functions are selected to be driven by the slow LFSR. The number of inputs driven by the slow LFSR is further increased by merging multiple compatible inputs into test signals. Merging compatible inputs also reduces the area overhead by reducing the number of dual clocked flip–flops required to implement the slow LFSR.

The 13% to 70% reductions in the numbers of weighted transitions are attained for the ISCAS85 and ISCAS89 benchmark circuits. High reductions in the numbers of weighted transitions are achieved for the circuits that have many compatible inputs. When we extend the definition of compatible inputs to *inputs that can be merged without loss of random pattern testability*, even higher reductions in the numbers of weighted transitions will be possible.

For most circuits, the stuck-at fault coverages obtained by applying the sequences generated by the proposed DS-LFSRs are equal to or higher than those obtained by applying the sequences generated by LFSRs with primitive feedback polynomials. The additional area overhead due to DS-LFSR is low, since only few (no more than 20) dual-clock flip–flops are required even in circuits with hundreds and even thousands of flip–flops.

The simulation results demonstrate that the DS-LFSR generated sequences typically achieve high transition delay fault coverages as well. If at-speed testing is desired to serendipitously cover unmodeled faults beyond the modeled stuck-at and delay faults, the test application scheme can be somewhat modified to accomplish that goal as well.

REFERENCES

[1] Y. Zorian, “A distributed BIST control scheme for complex VLSI devices,” in *Proc. VLSI Testing Symp.*, 1993, pp. 4–9.

[2] T.-L. Chou, K. Roy, and S. Prasad, “Estimation of circuit activity considering signal correlation and simultaneous switching,” in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1994, pp. 300–303.

[3] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd ed. Boston, MA: Addison-Wesley, 1992.

[4] J. J. Clement, “Electromigration modeling for integrated circuit interconnect reliability analysis,” *IEEE Trans. Device Materials Reliability*, vol. 1, pp. 33–42, Mar. 2001.

[5] D. A. Doane and P. D. Franzon, *Multichip Module Technologies and Alternatives: The Basics*. New York: Van Nostrand Reinhold, 1992.

[6] Y. Zorian, “A universal testability strategy for multi-chip modules based on BIST and boundary scan,” in *Proc. IEEE Int. Conf. Computer Design*, 1992, pp. 59–66.

[7] S. Gerstendorfer and H.-J Wunderlich, “Minimized power consumption for scan-based BIST,” in *Proc. IEEE Int. Test Conf.*, 1999, pp. 77–84.

[8] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, “A test vector inhibiting technique for low energy BIST design,” in *Proc. VLSI Testing Symp.*, 1999, pp. 407–412.

[9] S. Wang and S. K. Gupta, “LT-RTPG: A new test-per-scan BIST TPG for low heat dissipation,” in *Proc. IEEE Int. Test Conf.*, 1999, pp. 85–94.

[10] ——, “DS-LFSR: A new BIST TPG for low heat dissipation,” in *Proc. IEEE Int. Test Conf.*, 1997, pp. 848–857.

[11] F. Corno, M. Rebaudengo, M. Sonza Reorda, and M. Vilante, “A new BIST architecture for low power circuits,” in *Proc. Eur. Test Workshop*, 1999, pp. 160–164.

[12] S. Wang and S. K. Gupta, “ATPG for heat dissipation minimization for scan testing,” in *Proc. IEEE-ACM Design Automation Conf.*, 1997, pp. 255–259.

[13] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, “Techniques for minimizing power dissipation in scan and combinational circuits during test application,” *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 1325–1333, Dec. 1998.

[14] S. Wang and S. K. Gupta, “ATPG for heat dissipation minimization during test application,” *IEEE Trans. Comput.*, vol. 47, pp. 256–262, 1998.

[15] P. H. Bardell, W. H. McAnney, and J. Savir, *Built-In Test for VLSI: Pseudorandom Techniques*, New York: Wiley, 1987.

[16] S. Wang, “Minimizing heat dissipation during test application,” Ph.D., Univ. Southern California Electrical Engineering Systems, 1998.

[17] F. N. Najm, “Transition density, A stochastic measure of activity in digital circuits,” in *Proc. IEEE-ACM Design Automation Conf.*, June 1991, pp. 644–649.

[18] ——, “Transition density: A new measure of activity in digital circuits,” *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 310–323, Feb. 1993.

[19] D. E. Knuth, *The Art of Computer Programming*. Boston, MA: Addison-Wesley, 1981, vol. 2.

[20] S. W. Golomb, *Shift Register Sequences*. Laguna Hills, CA: Aegean Park, 1982.

[21] E. J. McCluskey, “Verification testing — A pseudoexhaustive test techniques,” *IEEE Trans. Computers*, vol. C-33, pp. 541–546, June 1984.

[22] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.

[23] B. W. Kernighan and S. Lin, “An efficient heuristic procedure for partitioning graphs,” *Bell Syst. Tech. J.*, vol. 49, pp. 291–307, Feb. 1970.

Seongmoon Wang (M’02) received the B.S. degree in electrical engineering from ChungBuk National University, Korea, in 1988, the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology, in 1991, and the Ph. D. degree in electrical engineering from University of Southern California, in 1998.

He is currently a Research Staff Member at CCRL, NEC USA, Princeton, NJ. He was previously a Design Engineer at GoldStar Electron, Korea, and a DFT Engineer at Syntest technologies and 3Dfx interactive. His research interests include design for testability and CAD.

Sandeep K. Gupta received the B.S. degree in electrical engineering from the Indian Institute of Technology, Kharagpur, India, in 1985 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts, Amherst, in 1989 and 1991, respectively.

He is an Associate Professor at the Department of Electrical Engineering—Systems at the University of Southern California, Los Angeles. His research interests include the area of VLSI testing and design. He is currently involved in projects on test and validation of deep submicrometer circuits, testing multicore systems-on-silicon, and delay testing and diagnosis of digital circuits. He is also involved in a project on testing and verification of network protocols.

Dr. Gupta is also an Associate Editor of IEEE TRANSACTIONS ON COMPUTERS. He is a recipient of the National Science Foundation’s Research Initiation Award (1992) and CAREER Award (1995). He is also a recipient of the Northrop Grumman Assistant Professorship (1995) and Zumberge Fellowship (1996) from the University of Southern California. He has received the Honorable Mention Award from the International Test Conference in 1997 and the Best Paper Award from the Asian Test Symposium in 2000.