VLSI Design
1994, Vol. 1, No. 4, pp. 313–326
Reprints available directly from the publisher
Photocopying permitted by license only

© 1994 Gordon and Breach Science Publishers S.A.
Printed in the United States of America

Empirical Bounds on Fault Coverage Loss Due to LFSR Aliasing

WARREN H. DEBANY JR.
MARK J. GORNIAK
ANTHONY R. MACERA
DANIEL E. DASKIEWICH
KEVIN A. KWIAT
HEATHER B. DUSSAULT
Rome Laboratory (RL/ERDA), Griffiss AFB, New York

Built-in-self-test (BIST) response data can be compacted using a linear-feedback shift register (LFSR). Prior work has indicated that the probability of aliasing tends to converge to 2^{-k} for a polynomial of degree k and large test length, and that primitive polynomials perform better than non-primitive polynomials. Nearly all analytical models and simulations have been based on the assumption that error occurrences are statistically-independent. This paper presents the first statistical results, based on fault simulation, that show that this convergence property holds for actual digital logic circuits and randomly-generated test vector sequences. However, it is shown that the *average* probability of aliasing is unsuitable as a design metric, and that a 95% *upper confidence limit (UCL)* is more useful. This paper introduces a UCL for the loss of fault coverage due to test response compaction. The theoretical or "ideal" UCL is shown to match closely the empirically-derived UCL obtained by fault simulation. The result is that a tight lower bound on fault coverage for LFSR-based BIST configurations can be obtained easily. Fault coverage for a BIST configuration can be obtained without the LFSR, eliminating costly fault simulation of the full structure with the LFSR. These results have been incorporated in the standard procedure for fault coverage measurement.

Key Words: *Signature analysis; Fault detection; Digital circuits; Linear-feedback shift registers; Multiple-input shift registers; Aliasing probability*

Nearly all built-in-self-test (BIST) techniques for digital logic circuits use some method of compacting the test response data. The most common approach involves the use of a linear-feedback shift register (LFSR) in a process called *signature analysis (SA)*. The SA configuration considered in this paper is where the primary outputs of a multiple-output circuit-under-test (CUT) are connected to a multiple-input LFSR, as shown in Figure 1. This type of LFSR is often called a *multiple-input signature register (MISR)*. LFSR and MISR design has been discussed by numerous authors (e.g., [1–3]).

Unfortunately, the use of SA, rather than direct monitoring of the primary outputs of the CUT, results in a loss of fault coverage due to the phenomenon called *aliasing*. Many investigators have addressed the problem of determining the probability that a fault that would be detected by direct monitoring of the primary outputs would not be detected by SA. Mathematical analyses have invariably been based on the assumption that error occurrences (due to faults present in the CUT) are statistically-independent. The limited fault simulation data published to date have generally considered only a single, given fault in the CUT, rather than the overall effects of all possible single faults in the CUT.

This paper reports the results of fault simulations involving a variety of CUTs based on actual circuits and the ISCAS '85 combinational benchmark set. An upper confidence limit on the loss of fault coverage is derived, and it is shown that this value closely matches the empirically-derived values that result from the fault simulation experiments. An early ver-

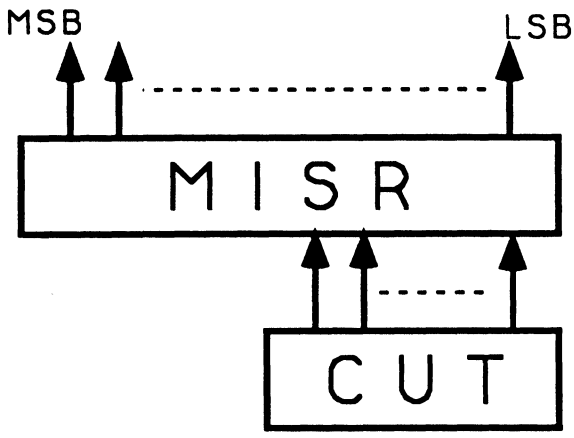


FIGURE 1 Signature analysis configuration using a MISR.

sion of the analyses presented here appeared in a paper by Debany, Manno, and Kanopoulos [4]. Some of the empirically-derived results of these experiments have been incorporated in a standard procedure for the reporting of fault coverage for digital circuits [5]. A shorter version of this paper was published at the 1992 IEEE VLSI Test Symposium [6].

Signature Analysis

A *test* consists of initializing the MISR to a known value and applying a sequence of test vectors to the CUT. The test vector sequence is generated by algorithmic or pseudorandom means, and is repeatable. At the end of the test the contents of the MISR (the *signature*) is compared to the known-good signature. It is assumed that the MISR is fault-free.

In this paper fault coverage is expressed in terms of *detectable* fault equivalence classes of the stuck-at-zero and stuck-at-one faults on the logic lines of the circuits, and is performed in accordance with the standard procedure for reporting fault coverage [5]. Only single faults are considered. Each "fault" is the representative of a fault equivalence class.

Let N be the number of CUT faults considered. Denote by $D(n)$ the cumulative number of faults detected by test vectors $1, 2, 3, \ldots n$, based on direct observation of the primary outputs of the CUT. The fault coverage with respect to the primary outputs of the CUT after the application of test vector n (step n) is given by $F(n) = D(n)/N$. Denote by $S(n)$ the number of faults that would be detected based *only on observation of the signature on step n*. The fault coverage with respect to the signature only on step n is given by $F'(n) = S(n)/N$.

Probability of Aliasing

Errors (incorrect output values caused by faults in the CUT) may be captured by the MISR yet be canceled by subsequent errors. This loss of error information is called *aliasing,* and the number of aliased faults on step n is given by $A(n) = D(n) - S(n)$. With respect to a given fault f, the *probability of aliasing* on step n, $P_{al}(n)$, is the conditional probability

$$\Pr\{\text{correct signature on step } n | f \text{ has been detected}\}$$

With respect to the entire universe of N faults, $P_{al}(n)$ is

$$\frac{A(n)}{D(n)} = 1 - \frac{S(n)}{D(n)} = 1 - \frac{F'(n)}{F(n)} \qquad (1)$$

for $D(n) > 0$, and is 0 otherwise.

A great deal of research has been devoted to the problem of determining and/or minimizing P_{al}. Most of the analytical models are based on assumptions about the occurrences of errors at the output of the CUT. For a given fault, its *probability of fault detection (PFD),* which we denote by q, is defined to be the probability that the fault causes an error at any primary output of a CUT in response to any randomly-generated test vector.

Smith [1] showed that if all error patterns are equiprobable, than for a k-bit LFSR and large n, $P_{al}(n)$ approaches 2^{-k}. This result is independent of the polynomial implemented by the LFSR; in fact, it holds even for a simple shift register that captures only the last k bits of output. In order for all error patterns to be equiprobable, it is necessary that every fault in the CUT have a statistically-independent PFD of 1/2. Smith pointed out that the assumption of equiprobable error patterns is a very strong one and is not generally valid for the errors expected to be produced by actual faulty circuits.

Carter [7] showed that $P_{al}(n) \leq 4/n$, for $n \leq 2^k$, for any k-bit MISR with internal feedback, regardless of q. This bound is considered to be extremely conservative. His approach is in a sense complementary to other methods in that his bound holds for *small n,* rather than *large n.*

Convergence to a Steady-State Value

Williams, Daehn, Gruetzner, and Starke [8–10] used a Markov model to determine both exact values and upper bounds for $P_{al}(n)$ for single-input LFSRs based

on the assumption of statistically-independent PFDs. They considered the behavior of primitive and non-primitive polynomials. Their major finding was that $P_{al}(n)$ appears to *converge* to a "steady-state" value of 2^{-k} for large n, for both primitive and non-primitive polynomials, regardless of q. They found that convergence appears to be faster for primitive polynomials. (Reducible, irreducible, and primitive polynomials and their properties are discussed in the section on Experiments.)

The convergence property observed by Williams, *et al.*, has since been validated by many researchers, under numerous error models and for many implementations of single-input and multiple-input LFSRs [11–18]. All of the models assume that error occurrences are statistically-independent.

Test Length for Convergence

For a single-input LFSR that implements a primitive polynomial of degree k, Williams *et al.* [10], presented an upper bound on aliasing probability:

$$P_{al}(n) \leq 2^{-k} + (2^k - 1)|1 - 2q|^{n(1-1/(2^k-1))} \quad (2)$$

Example 1 Let us consider the single-output circuit obtained from the 54LS181 ALU by eliminating all gates and lines in the gate-level logic model that do not contribute to the primary output "A = B"; this is the largest single-output subcircuit of this ALU, and was one of the circuits considered by Xavier, *et al.* [12–13]. Of the 193 detectable fault equivalence classes in this circuit, only one has a PFD greater than 1/2: $q = 0.85938$. The next largest PFD is 0.16699. More than 95% of the PFDs are greater than or equal to 0.00952148. Using (2), the test lengths required to guarantee that $P_{al}(n)$ is within 1% of its asymptotic value of 2^{-k} for a PFD of 0.00952148 are $n = 820$ for $k = 8$, $n = 1{,}394$ for $k = 16$, and $n = 1{,}970$ for $k = 24$.

For a MISR that implements a primitive polynomial of degree k, Damiani, *et al.* [16], showed that the aliasing probability for a fault with a given PFD is "very close" to 2^{-k} after a test length of k/q.

Example 2 Of the 257 detectable fault equivalence classes in the full logic model of the 54LS181 ALU, 11.7% have PFDs equal to 1/2 and 6.2% have PFDs greater than 1/2. More than 95% of the PFDs are greater than or equal to 0.0234375. Using the results given by Damiani, *et al.*, the test lengths required to effectively reach the asymptotic value of 2^{-k} for a PFD of 0.0234375 are $n = 342$ for $k = 8$, $n = 683$ for $k = 16$, and $n = 1{,}024$ for $k = 24$.

Based on these results, it can be seen that the predicted rate of convergence is relatively rapid for the majority of the possible single stuck-at faults in a typical digital circuit.

Henceforth, the postulated convergence of $P_{al}(n)$ to a "steady-state" value of 2^{-k} for large n, regardless of the PFD, will be referred to as the *convergence property (CP)*.

The remainder of this paper is devoted to two topics. First, it is established that CP, the proofs for which are based on the statistically-independent error assumption, holds for actual digital logic circuits. Second, information about the steady state value of $P_{al}(n)$ is used to determine bounds on the loss of fault coverage due to LFSR aliasing.

USING A MISR

$F(n)$, the fault coverage of a test vector sequence with respect to the *primary outputs* of a CUT, can be obtained easily by means of fault simulation. This process is relatively efficient because "fault dropping" (i.e., eliminating a fault from further consideration after its first detection) can be used.

Now, let the response data be compacted using a MISR, as shown in Figure 1. The test vector sequence is applied to the CUT and the signature is read on a *single* test vector n (resulting in a fault coverage $F'(n)$); the test is terminated at that point. In a direct assault on the problem, it is possible to obtain the actual value of $F'(n)$ by fault simulation. However, fault dropping provides no computational savings because all fault detection information is obtained on the basis of the final signature alone; no faults are dropped until the final signature is read. (In the case where multiple signatures can be read, reading an early signature causes a large fraction of the faults to be dropped so that subsequent fault simulation executes faster.) For circuits consisting of only a few thousand gates and sequences consisting of only a few thousand test vectors, obtaining $F'(n)$ can be expected to require several orders of magnitude more computational effort than does obtaining $F(n)$.

Another problem with the direct approach for determining $F'(n)$ is that observation of the fault signature is essentially a "destructive read" of the fault coverage. That is, if $F'(n)$ is obtained, then obtaining $F'(n + 1)$ requires that *all* of the computations on steps 1, 2, 3, . . . , n be repeated, unless the fault simulator is capable of checkpointing and restarting from intermediate points.

In only a handful of cases have aliasing results been reported that are based on fault simulation of an actual digital logic circuit, in contrast to modeling the occurrence of independent errors [19–23]. Xavier, *et al.* [12, 13], used the z-statistic as a measure of the goodness-of-fit between their analytical experiments for $P_{al}(n)$ and values observed during a set of fault simulation experiments. They simulated only *one* fault per experiment, however, which ensured that errors were statistically-independent. It appears that the only statistically-based study involving all single stuck-at faults in a circuit, was an early version of the analysis performed in this paper [4].

The large CPU time requirements prohibit the actual measurement of $F'(n)$ in any practical BIST design. Therefore, it has been necessary to develop a procedure for estimating $F'(n)$ on the basis of the measured $F(n)$ and knowledge of the probability of aliasing due to the LFSR.

LOSS OF FAULT COVERAGE

If CP does in fact hold for actual circuits, a statistical relationship between the pre- and post-compaction fault coverages can be established. However, CP and any proposed relationships between $F(n)$ and $F'(n)$ must be validated empirically to ensure they are valid.

Given that the purpose of performing the test is to provide good fault coverage, the effects of aliasing are irrelevant until $F(n)$ is close to 1. It is assumed here that a signature is never taken before a *crossover* point has been reached in the test vector sequence such that $F(n)$ is at or near 1. Arbitrarily, the smallest value of n for which $F(n) \geq 0.95$ was selected as the crossover point; this value of n is denoted by n_c. All statistics were obtained after crossover in the experiments summarized in the next section. In the course of these experiments the test response compaction process begins on step 1; it is only the beginning of data collection for the sample statistics that is deferred until step n_c.

Randomly-generated test sequences generally are much longer than those generated by deterministic algorithms (for a survey of deterministic test generation techniques see Kirkland and Mercer [24]). A random test is seldom terminated at exactly the point where the stuck-at fault coverage is deemed to be sufficient (such as on step n_c). Instead, it is customary to exploit the length of the random test sequence in order to detect fault types other than single stuck-at-zero and stuck-at-one. Coverages of other classes of failures, such as shorts, opens, and transition and delay faults, are difficult to quantify, but it is well-known that long, randomly-generated sequences provide serendipitous detection of these "non-classical" faults (e.g., Waicukauski and Lindbloom [25]). Thus, random test length is usually much greater than that required only to achieve high single stuck-at fault coverage.

Crossover for the 54LS181 occurred on step $n_c = 81$. Based on the examples of the predicted rates of convergence given in the Introduction (using bounds given by Williams, *et al.*, and Damiani, *et al.*), "convergence" to the asymptotic value of $P_{al}(n)$ cannot be expected to occur by step n_c for most faults. Therefore, the early samples obtained starting at n_c represent a population with a larger variance than those from later in the test vector sequence, and make the sample statistics more conservative. A test length of 5,000 was used in each case, however, so the majority of the samples represented the process in the "steady state."

We now obtain two *upper confidence limits (UCLs)* on the loss of fault coverage due to aliasing. The first is an "ideal" or theoretical upper bound based on CP. The second is an empirical estimate based on sample statistics obtained from fault simulation of actual CUTs and MISRs.

An Ideal UCL

Consider any step n ($n \geq n_c$) in a test vector sequence. Assuming that each of the $D(n)$ detectable faults has a statistically-independent probability 2^{-k} of being aliased, the number of aliased faults, $A(n)$, is distributed as a binomial random variable (r.v.) [26]. Because $2^{-k} \ll 1$, the distribution of $A(n)$ can be approximated as a Poisson distribution with parameter $\lambda = 2^{-k}D(n)$. Therefore, $\mathrm{E}\{A(n)\} = \mathrm{Var}\{A(n)\} = 2^{-k}D(n)$. A customary conservative 95% UCL for a Poisson r.v. is given by the mean plus three standard deviations. Since $F(n) \approx 1$ and $D(n) \approx N$ for $n \geq n_c$, (1) reduces to $P_{al}(n) \approx F(n) - F'(n)$. This results in an upper bound on the fraction of the N faults given by

$$2^{-k} + 3\sqrt{\frac{2^{-k}}{N}} \tag{3}$$

Expression (3) is an *ideal UCL* for the loss of fault coverage due to aliasing. Note that it *decreases* with the number of faults (hence, the size of the CUT) considered.

Example 3 Expression (3) was applied to two SA configurations involving standard logic circuits [27]. (These two circuits are considerably smaller than the size that would justify using SA, because the MISR would be larger than the CUT. However, the sizes of these circuits are commensurate with those of the test cases considered in the literature.) The 54LS85 four-bit comparator has $N = 137$ fault equivalence classes. Compacting this circuit's three primary outputs using a MISR of degree $k = 4$, $2^{-4} = 0.0625$ is the predicted average value of $P_{al}(n)$. Expression (3) yields a 95% UCL of 0.1266 for $P_{al}(n)$, which is twice the average value. The 54LS181 four-bit ALU has $N = 261$ fault equivalence classes. Compacting this circuit's eight primary outputs using a MISR of degree $k = 16$, $2^{-16} = 1.53 \times 10^{-5}$ is the predicted average value of $P_{al}(n)$. Expression (3) yields a 95% UCL of 7.41×10^{-4} for $P_{al}(n)$, which is more than 48 times the average value.

An Empirical UCL

Consider r.v. X distributed with mean μ and variance σ^2; let x be a numerical sample value of X. If a distribution is roughly mound-shaped and unimodal, then Camp and Meidel's Extension of Tchebychev's Inequality [26] states that the probability of a sample being outside K standard deviations from the mean is less than or equal to $1/(2.25K^2)$; thus, $K = 2.9814$ for a 95% confidence interval. Given a set of M samples $x_1, x_2, x_3, \ldots x_M$, the true mean is approximated by the sample mean:

$$m = \frac{1}{M} \sum_{i=1}^{M} x_i \tag{4}$$

The true standard deviation is approximated by the square root of the sample variance given by:

$$s^2 = \frac{1}{M-1} \left[\sum_{i=1}^{M} x_i^2 - Mm^2 \right]$$

In this case, it is the distribution of $P_{al}(n)$ that is of interest. Thus, a conservative 95% UCL for $P_{al}(n)$ is given by

$$m + 2.9814s \tag{5}$$

where the samples $x_1, x_2, x_3, \ldots x_M$ used to calculate m and s are the values of $P_{al}(n)$ *taken after crossover*. That is, $x_i = P_{al}(n_c + i - 1)$. The values of $P_{al}(n)$ are calculated from the observed values of $D(n)$ and $S(n)$ obtained by *full fault simulation* of the CUTs with SA. Expression (5) is an *empirical UCL* for the loss of fault coverage for a specific CUT, MISR, and test vector sequence.

Because we wish to determine how nearly the average probability of aliasing converges to 2^{-k}, it is convenient to take the logarithm (base 2) of the sample mean. We define "Ave P_{al}" to denote the value given by the expression

$$-\log_2 m \tag{6}$$

where m is the sample mean of the values of $P_{al}(n)$ after crossover given by (4). This value is defined only if $m > 0$.

EXPERIMENTS

Experiments were conducted to obtain statistics on aliasing. This section summarizes details of the MISR configuration, circuits, and test vector sequences for which the statistics were obtained.

MISR Implementations

The LFSRs used here are of the *internal XOR* type, where the most significant bit is fed back to taps between the flip-flop stages. An example of this configuration is shown in Figure 2. The outputs of the CUT are inputs to XOR gates between the flip-flop stages, and so the degree of the MISR must be greater than or equal to the number of primary outputs of the CUT. (It is common practice to XOR several outputs of a circuit in order to use a MISR with a degree less than the number of primary outputs of the circuit. If this is done, however, the CUT must be defined to be the circuit *with* the addition of the XOR gates.)

A polynomial in GF(2) of degree k is called *irreducible* if and only if it is not divisible by any polynomial of degree less than k but is greater than 0; otherwise it is called *reducible*. A polynomial of degree k is called *primitive* if and only if it is irreducible and it does not divide any polynomial of the form $x^j + 1$ for any j less than $2^k - 1$ [2]. Any primitive polynomial is also irreducible, but in this paper a polynomial identified as "irreducible" should be understood to mean "irreducible but not primitive."

Primitive polynomials are also called "maximal length" polynomials, because a LFSR of degree k that is initialized to any non-zero value cycles through all of the $2^k - 1$ non-zero values if and only if the LFSR implements a primitive polynomial.

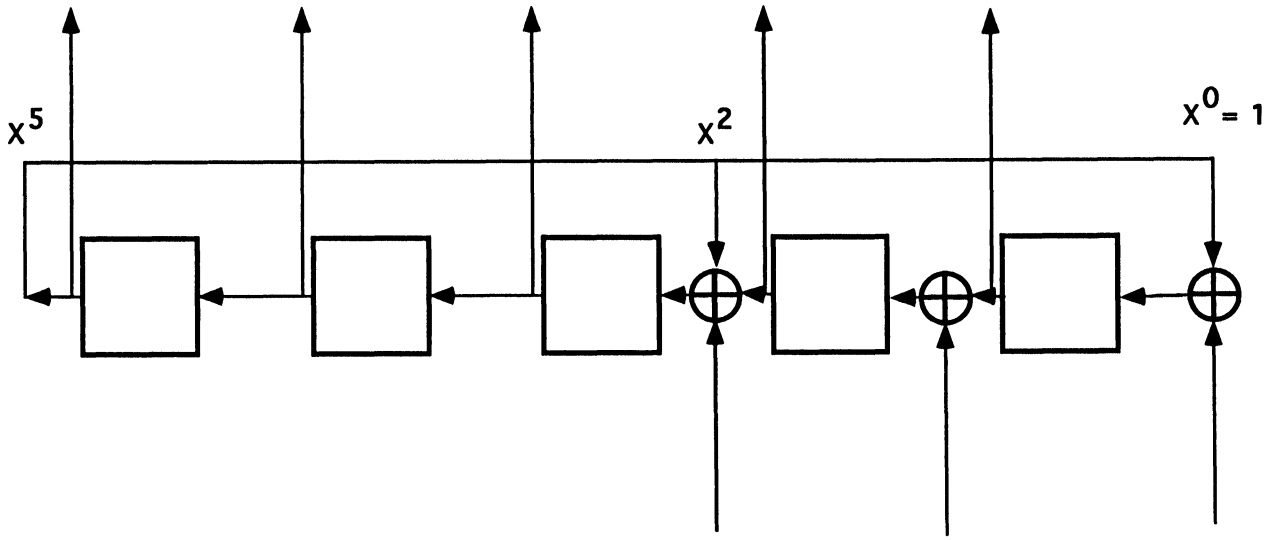


FIGURE 2 Three-input MISR implementing primitive polynomial of degree 5: $x^5 + x^2 + 1$.

Primitive polynomials have traditionally been important in BIST design not only for test response compaction, but also because of their usefulness as the generators of the test stimuli. Lists of primitive polynomials and discussions of their properties are given by Peterson and Weldon [2] and Bardell, McAnney, and Savir [28].

Twelve "families" of feedback polynomials were used in this study: four *primitive* polynomials P1-4, four *irreducible* polynomials IR1-4, and four *reducible* polynomials R1-4. Results only for polynomials of degrees 8, 16, 24, and 32 are reported here.

P1 is the primitive polynomial of appropriate degree given in Bardell, *et al.* [28]. P2 is the first primitive polynomial of appropriate degree listed in Peterson and Weldon [2]. P3 is the reciprocal polynomial of P2 (i.e., it has the same coefficients as P2, but listed in reverse order). P4 is the second primitive polynomial of appropriate degree listed in Peterson and Weldon [2].

IR1 is the first irreducible but non-primitive polynomial of appropriate degree listed in Peterson and Weldon [2]. IR2 is the reciprocal polynomial of IR1. IR3 is the second primitive polynomial of appropriate degree listed in Peterson and Weldon [2], and IR4 is the third.

R1 is the simplest nondegenerate polynomial of degree k: $x^k + 1$. R2 is constructed by starting with R1, and including all terms with even powers. R3 is constructed by starting with R1, and including all terms with powers divisible by 2 or 3. R4 is the reciprocal polynomial of R3.

The reducible polynomials are related structurally within a "family," but the primitive and irreducible polynomials are related within a "family" only by virtue of the fact that they happen to be tabulated together.

Circuits-Under-Test

Twelve combinational digital logic circuits were used in this study. Five of the circuits are standard MSI/LSI functions [27, 30] representative of those used to implement more complex VLSI devices and systems. Seven are from the standard "ISCAS '85" combinational benchmark set [31]. Table I lists the name, number of detectable fault equivalence classes, number of primary outputs, and function for each circuit. The number of detectable fault equivalence classes for the ISCAS circuits is given by Schulz and Auth [29].

Because polynomials of degrees up to only 32 were considered, only CUTs with 32 or fewer primary outputs were used. Circuits with larger gate count could not be fault-simulated (without the advantage of fault dropping) in a reasonable amount of computer time. In any case, based on the ideal UCL given by (3), it appears that small circuits provide the worst-case test cases.

Stimuli and Test Length

The test vector sequences were generated by applying statistically-independent, equiprobable 0*s* and 1*s* to the primary inputs of the CUTs. A modified multiplicative congruential random number generator

TABLE 1
Circuits Summary

Circuit Name	Number of Detectable Fault Equivalence Classes (N)	Number of Primary Outputs	Function
54LS182	83	5	Carry-lookahead generator
54LS85	137	3	Comparator
Am25S05	258	7	Multiplier
54LS181	257	8	ALU
Am25LS2517	385	6	ALU
C17	22	2	ISCAS circuit; trivial function
C432	520	7	ISCAS circuit; priority decoder
C499	750	32	ISCAS circuit; ECAT
C880	942	26	ISCAS circuit; ALU and control
C1355	1566	32	ISCAS circuit; ECAT
C1908	1870	25	ISCAS circuit; ECAT
C3540	3291	22	ISCAS circuit; ALU and control

was used to create the stimuli. For a given CUT, the *same* randomly-generated test vector sequence was used for every degree and polynomial.

The test length for each trial was 5,000, and all statistics were obtained after crossover, i.e., the point where fault coverage first equaled or exceeded 95% of the detectable faults.

Demonstration

Figure 3 exemplifies the SA process and the collection of the statistics. The CUT in this example is the 54LS85 4-bit comparator. The three primary outputs of the CUT are input to a MISR that implements $x^4 + x + 1$ (one of the primitive polynomials listed in

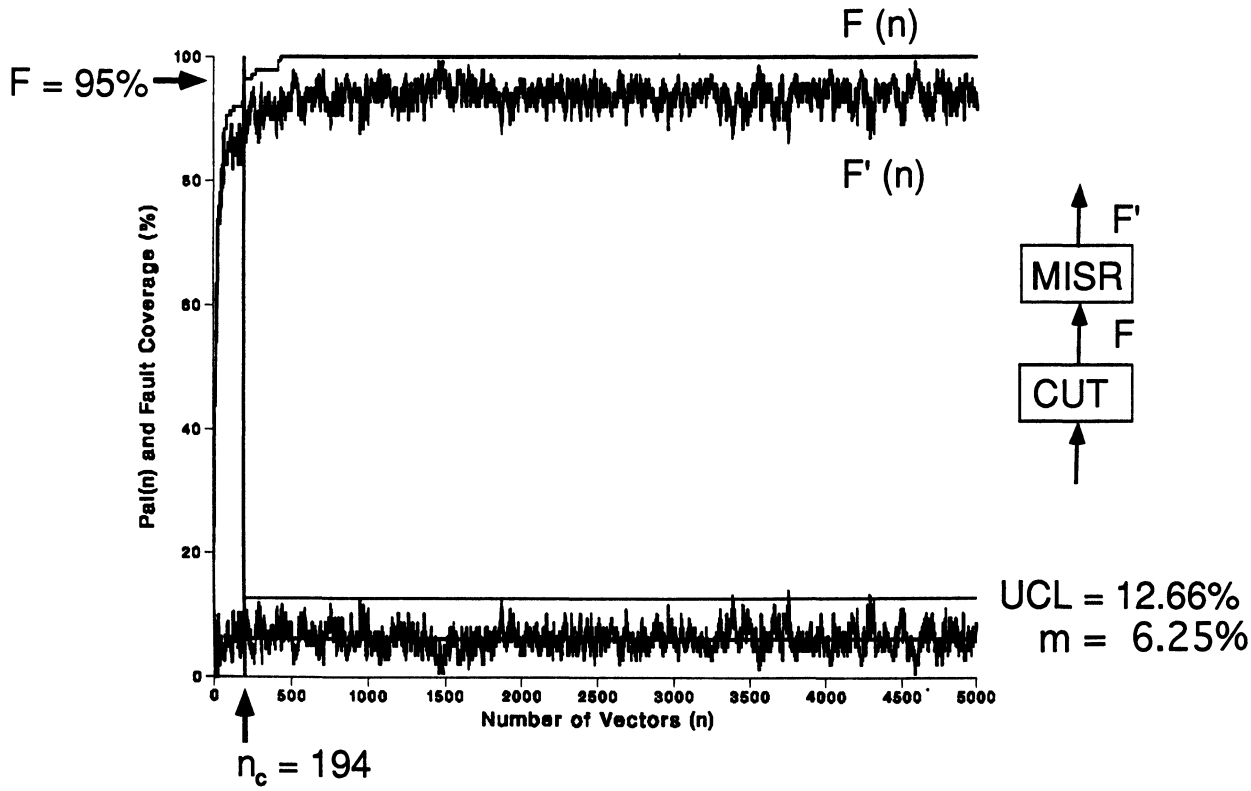


FIGURE 3 Example of a signature analysis experiment. Circuit is 54LS85 4-bit comparator (three primary outputs). MISR implements primitive polynomial $x^4 + x + 1$.

Bardell, *et al.* [28]). The small degree was chosen for this example so that the aliasing behavior would be easily visible in the figure.

On any vector n, the cumulative fault coverage obtained by direct observation of the CUT's outputs is denoted by $F(n)$. The fault coverage obtained by SA is denoted by $F'(n)$; that is, $F'(n)$ is the fault coverage that would be obtained if the signature on step n were the only signature observed during the test.

At the top of the figure, it can be seen that $F(n)$ rises monotonically to 100%. $F'(n)$, on the other hand, almost immediately begins to display aliasing in the form of rises and dips in its signature-based coverage.

$F(n)$ crosses 95% at $n = n_c = 194$ (the "crossover" point). Beyond the crossover point, $F'(n)$ can be taken to be in its "steady state," but $F'(n)$ tends to remain well below $F(n)$.

At the bottom of the figure is shown the value of $P_{al}(n)$ calculated using (1). The value of m shown in Figure 3 is the sample mean of the values of $P_{al}(n)$ taken after crossover. A horizontal line is drawn at the value of the UCL given by (3).

The value of 6.25% for m coincides with the value $2^{-4} = 0.0625$ predicted by CP. Using (6), "Ave P_{al}" is 4.0, which corresponds to the degree of the polynomial used. Clearly, using this value as the predicted loss of fault coverage would be optimistic on approximately half the test vectors. On the other hand, the value of the UCL (12.66%) is a reliably conservative upper bound on the loss of fault coverage; it can be seen that the actual value of $P_{al}(n)$ is almost always less than this UCL. It can also be seen that this is a good 95% UCL in that $P_{al}(n)$ is greater than or equal to the UCL occasionally, but less than 5% of the time. Thus, this UCL is conservative, but only to the degree it is designed to be.

SIGNATURE ANALYSIS RESULTS

Results of experiments for each circuit for polynomials of degree 8, 16, 24, and 32 are summarized in Tables II-X. C3540 has 22 primary outputs, so its statistics are given only for polynomials of degrees 24 and 32. Results are given only for degree 32 for circuits C499, C880, C1355, and C1908, and aliasing was observed only for polynomials R1 and R2; these results are summarized together in Table IX. Each table lists the ideal UCL given by (3), the empirical UCL given by (5), and "Ave P_{al}" (obtained after crossover) given by (6). If no aliasing events occurred, then m was zero, and "***" is listed for "Ave P_{al}" because the value is undefined. If the average value of $P_{al}(n)$ were to converge to exactly 2^{-k} after crossover, then "Ave P_{al}" would be exactly k.

The line labeled "Ideal" in each table lists the ideal UCL given by (3) for the UCL, and k for "Ave P_{al}." If the average probability of aliasing converged to exactly 2^{-k}, and the Poisson assumptions perfectly described the occurrence of aliasing events, then these values are what would have been obtained from the fault simulation experiments.

TABLE II
Circuit 54LS182

	Degree 8		Degree 16		Degree 24		Degree 32	
Polynomial	UCL	Ave P_{al}	UCL	Ave P_{al}	UCL	Ave P_{al}	UCL	Ave P_{al}
Ideal	0.024487	8.00	0.001302	16.00	0.000080	24.00	0.000005	32.00
P1	0.022902	8.24	0.001375	15.83	0.000000	***	0.000000	***
P2	0.025152	7.84	0.000000	***	0.000000	***	0.000000	***
P3	0.032107	7.44	0.000896	17.05	0.000000	***	0.000000	***
P4	0.022263	8.26	0.000896	17.05	0.000000	***	0.000000	***
IR1	0.029958	7.76	0.000516	18.63	0.000000	***	0.000000	***
IR2	0.029276	7.70	0.000731	17.63	0.000000	***	0.000000	***
IR3	0.023780	8.09	0.001036	16.63	0.000000	***	0.000000	***
IR4	0.020337	8.61	0.000516	18.63	0.000000	***	0.000000	***
R1	0.054416	6.57	0.009510	10.41	0.025647	8.74	0.009463	10.42
R2	0.025611	8.02	0.016579	8.74	0.009113	10.48	0.000000	***
R3	0.021589	8.39	0.001471	15.63	0.000000	***	0.000000	***
R4	0.030603	7.51	0.000731	17.63	0.000000	***	0.000000	***

TABLE III
Circuit 54LS85

Polynomial	Degree 8 UCL	Ave P_{nl}	Degree 16 UCL	Ave P_{nl}	Degree 24 UCL	Ave P_{nl}	Degree 32 UCL	Ave P_{nl}
Ideal	0.019925	8.00	0.001016	16.00	0.000063	24.00	0.000004	32.00
P1	0.021009	7.82	0.000000	***	0.000000	***	0.000000	***
P2	0.021111	7.93	0.000548	17.74	0.000000	***	0.000000	***
P3	0.022160	7.93	0.001534	14.94	0.000000	***	0.000000	***
P4	0.018399	8.15	0.000447	18.33	0.000000	***	0.000000	***
IR1	0.017564	8.22	0.000000	***	0.000000	***	0.000000	***
IR2	0.020044	7.89	0.000000	***	0.000000	***	0.000000	***
IR3	0.025804	7.61	0.001057	15.87	0.000000	***	0.000000	***
IR4	0.023013	7.65	0.000000	***	0.000000	***	0.000000	***
R1	0.023881	7.96	0.003749	12.33	0.000000	***	0.000000	***
R2	0.023202	7.93	0.000779	17.32	0.000000	***	0.000000	***
R3	0.021215	7.97	0.000000	***	0.000000	***	0.000000	***
R4	0.023135	7.80	0.000315	19.33	0.000000	***	0.000000	***

TABLE IV
Circuit Am25S05

Polynomial	Degree 8 UCL	Ave P_{nl}	Degree 16 UCL	Ave P_{nl}	Degree 24 UCL	Ave P_{nl}	Degree 32 UCL	Ave P_{nl}
Ideal	0.015579	8.00	0.000745	16.00	0.000046	24.00	0.000003	32.00
P1	0.016256	7.95	0.000784	15.83	0.000000	***	0.000000	***
P2	0.015747	8.01	0.000551	16.83	0.000000	***	0.000000	***
P3	0.016765	7.93	0.000498	17.12	0.000000	***	0.000000	***
P4	0.015154	8.13	0.000331	18.29	0.000000	***	0.000000	***
IR1	0.016028	7.93	0.000576	16.71	0.000000	***	0.000000	***
IR2	0.016210	7.98	0.000623	16.49	0.000000	***	0.000000	***
IR3	0.015473	8.00	0.000469	17.29	0.000000	***	0.000000	***
IR4	0.015093	8.09	0.000406	17.71	0.000000	***	0.000000	***
R1	0.016359	7.92	0.001616	13.80	0.000678	16.27	0.000000	***
R2	0.019040	7.87	0.002033	14.44	0.000000	***	0.000529	16.96
R3	0.015776	8.04	0.000784	15.83	0.000165	20.29	0.000000	***
R4	0.016044	7.94	0.000576	16.71	0.000000	***	0.000000	***

TABLE V
Circuit 54LS181

Polynomial	Degree 8 UCL	Ave P_{nl}	Degree 16 UCL	Ave P_{nl}	Degree 24 UCL	Ave P_{nl}	Degree 32 UCL	Ave P_{nl}
Ideal	0.015602	8.00	0.000746	16.00	0.000046	24.00	0.000003	32.00
P1	0.016076	8.00	0.001252	15.14	0.000000	***	0.000000	***
P2	0.015866	8.10	0.000474	17.27	0.000000	***	0.000000	***
P3	0.016594	7.99	0.000443	17.46	0.000000	***	0.000000	***
P4	0.018464	7.79	0.000289	18.68	0.000000	***	0.000000	***
IR1	0.015662	8.03	0.000557	16.81	0.000000	***	0.000000	***
IR2	0.015473	8.18	0.000474	17.27	0.000000	***	0.000000	***
IR3	0.016573	7.99	0.000582	16.68	0.000000	***	0.000000	***
IR4	0.017055	7.90	0.000528	17.68	0.000000	***	0.000000	***
R1	0.018081	7.78	0.003276	12.29	0.000000	***	0.000929	15.36
R2	0.018427	7.83	0.001020	15.18	0.000000	***	0.000548	16.90
R3	0.016271	7.99	0.000557	16.81	0.000000	***	0.000000	***
R4	0.017086	7.87	0.000441	17.95	0.000000	***	0.000000	***

TABLE VI
Circuit Am25LS2517

Polynomial	Degree 8 UCL	Ave P_{al}	Degree 16 UCL	Ave P_{al}	Degree 24 UCL	Ave P_{al}	Degree 32 UCL	Ave P_{al}
Ideal	0.013462	8.00	0.000613	16.00	0.000037	24.00	0.000002	32.00
P1	0.014466	8.07	0.000503	17.52	0.000000	***	0.000000	***
P2	0.015234	7.87	0.000553	16.68	0.000000	***	0.000000	***
P3	0.015927	8.10	0.000387	18.26	0.000000	***	0.000000	***
P4	0.015268	7.94	0.000598	16.84	0.000000	***	0.000000	***
IR1	0.013471	8.29	0.000725	15.80	0.000000	***	0.000000	***
IR2	0.016669	7.83	0.000448	17.53	0.000000	***	0.000000	***
IR3	0.015512	7.87	0.000448	17.84	0.000000	***	0.000000	***
IR4	0.016314	7.88	0.000907	15.39	0.000000	***	0.000000	***
R1	0.017541	7.73	0.001373	13.83	0.000486	16.70	0.000000	***
R2	0.022275	6.96	0.006834	11.02	0.000112	20.84	0.000000	***
R3	0.018092	7.55	0.000389	17.26	0.000000	***	0.000000	***
R4	0.017044	7.63	0.001841	13.56	0.000000	***	0.000000	***

TABLE VII
Circuit C17

Polynomial	Degree 8 UCL	Ave P_{al}	Degree 16 UCL	Ave P_{al}	Degree 24 UCL	Ave P_{al}	Degree 32 UCL	Ave P_{al}
Ideal	0.043881	8.00	0.002514	16.00	0.000156	24.00	0.000010	32.00
P1	0.040298	8.19	0.002730	15.75	0.000000	***	0.000000	***
P2	0.045379	7.95	0.000000	***	0.000000	***	0.000000	***
P3	0.043971	7.96	0.002730	15.75	0.000000	***	0.000000	***
P4	0.041866	8.06	0.002730	15.75	0.000000	***	0.000000	***
IR1	0.044342	7.95	0.000000	***	0.000000	***	0.000000	***
IR2	0.044127	7.89	0.000000	***	0.000000	***	0.000000	***
IR3	0.048758	7.69	0.000000	***	0.000000	***	0.000000	***
IR4	0.039747	8.19	0.000000	***	0.000000	***	0.000000	***
R1	0.042931	8.03	0.001927	16.75	0.000000	***	0.000000	***
R2	0.043247	8.05	0.005830	13.58	0.000000	***	0.000000	***
R3	0.041235	8.19	0.005830	13.58	0.000000	***	0.000000	***
R4	0.041608	8.11	0.000000	***	0.000000	***	0.000000	***

TABLE VIII
Circuit C432

Polynomial	Degree 8 UCL	Ave P_{al}	Degree 16 UCL	Ave P_{al}	Degree 24 UCL	Ave P_{al}	Degree 32 UCL	Ave P_{al}
Ideal	0.012129	8.00	0.000529	16.00	0.000032	24.00	0.000002	32.00
P1	0.012191	8.05	0.000498	16.14	0.000000	***	0.000000	***
P2	0.011870	8.06	0.000402	16.75	0.000000	***	0.000000	***
P3	0.012404	7.98	0.000313	17.46	0.000000	***	0.000000	***
P4	0.012072	7.93	0.000375	16.95	0.000000	***	0.000000	***
IR1	0.011592	8.09	0.000335	17.27	0.000000	***	0.000000	***
IR2	0.012979	7.93	0.000780	14.88	0.000000	***	0.000000	***
IR3	0.012898	7.90	0.000725	15.57	0.000083	21.27	0.000000	***
IR4	0.011333	8.18	0.000547	15.88	0.000000	***	0.000000	***
R1	0.014238	7.74	0.001306	14.21	0.001405	13.25	0.000000	***
R2	0.014812	7.79	0.000186	18.95	0.000000	***	0.000669	15.33
R3	0.011423	8.11	0.000741	15.02	0.000000	***	0.000000	***
R4	0.012050	7.95	0.000510	16.34	0.000000	***	0.000000	***

TABLE IX
Circuits C499, C880, C1355, C1908: Degree 32 only. Aliasing was observed only for polynomials R1 and R2.

	C499	C880	C1355	C1908
Polynomial	UCL Ave P_{al}	UCL Ave P_{al}	UCL Ave P_{al}	UCL Ave P_{al}
Ideal	0.000002 32.00	0.000001 32.00	0.000001 32.00	0.000001 32.00
R1	0.002239 13.10	0.000746 14.24	0.004564 10.20	0.002420 11.80
R2	0.003540 12.38	0.001274 12.70	0.003473 10.41	0.002668 10.30

TABLE X
Circuit C3540

	Degree 8	Degree 16	Degree 24	Degree 32
Polynomial	UCL Ave P_{al}	UCL Ave P_{al}	UCL Ave P_{al}	UCL Ave P_{al}
Ideal	N/A	N/A	0.000013 24.00	0.000001 32.00
P1	N/A	N/A	0.000000 ***	0.000000 ***
P2	N/A	N/A	0.000000 ***	0.000000 ***
P3	N/A	N/A	0.000000 ***	0.000000 ***
P4	N/A	N/A	0.000000 ***	0.000000 ***
IR1	N/A	N/A	0.000000 ***	0.000000 ***
IR2	N/A	N/A	0.000000 ***	0.000000 ***
IR3	N/A	N/A	0.000000 ***	0.000000 ***
IR4	N/A	N/A	0.000036 21.02	0.000000 ***
R1	N/A	N/A	0.001317 11.67	0.000781 13.38
R2	N/A	N/A	0.001507 11.02	0.001326 11.43
R3	N/A	N/A	0.000000 ***	0.000000 ***
R4	N/A	N/A	0.000057 19.77	0.000000 ***

Ideal vs. Empirical Results

The four families of primitive polynomials P1-4 performed well. Their empirical UCLs and values for "Ave P_{al}" hover around the theoretical values. For degrees 24 and 32, no aliasing events were observed between crossover and step 5,000. The performance of these four families of polynomials cannot be distinguished between on the basis of these experiments.

The four families of reducible polynomials IR1-4 for the most part performed as well as the primitive polynomials. For circuit C432, IR2 showed aliasing (one fault on one vector) for $k = 24$. For circuit C3540, IR4 showed aliasing for $k = 24$. For circuit C432, IR2 had a rather poor "Ave P_{al}" (14.88) for $k = 16$.

The reducible polynomials R1-4 generally performed well for $k = 8$, except for the 54LS182, but performed poorly for other degrees. Their performances at $k = 24$ and $k = 32$ were usually worse than those of the other polynomials at $k = 16$. The worst performance was exhibited by R1 and R2, which is particularly evident in the cases of the ISCAS circuits.

For the primitive and irreducible polynomials, the empirical UCLs are remarkably close to the ideal UCLs. This was surprising because both UCLs are conservative, and thus one of them would have been expected to have been consistently much greater than the other.

The fact that "Ave P_{al}" tends toward k indicates that CP indeed holds for circuits with stuck-at faults (albeit weakly for the reducible polynomials). The fact that the ideal UCL obtained by modeling aliasing events as a Poisson process closely matches the empirical UCL indicates that other prior results obtained using the independent-error assumption (such as predicted rates of convergence) may hold as well.

Behavior in the Long Run

The mathematical models of SA aliasing that appear in the literature seem to imply that a large test length results in $P_{al}(n)$ reaching a constant value, or at least a more consistent time-averaged value. However, this is not necessarily the case.

Example 4 A running estimate of m based on the last 500 values (i.e., samples $P_{al}(n - 499)$, $P_{al}(n -$

498), $P_{al}(n - 497), \ldots, P_{al}(n)$) was calculated for each value of $n \geq n_c + 499$. In the case of the 54LS181 for $k = 8$, for P1 the *smallest* running estimate of m was 0.001084 (at $n = 4{,}230$), and the *largest* was 0.006578 (at $n = 2{,}062$); thus, the *best* value of the running estimate occurred later in the sequence. However, for P2 the *smallest* running estimate of m was 0.001494 (at $n = 1{,}593$), and the *largest* was 0.007952 (at $n = 3{,}173$); thus the *worst* value of the running estimate occurred later in the sequence.

APPLICATIONS

The results of the experiments reported in this paper have been incorporated in the military- and industry-standard procedure for the consistent reporting of fault coverage for digital microcircuits. MIL-STD-883, Procedure 5012 [5], provides guidelines for the development of the logic model, the assumed fault model and fault universe, fault classing, fault sampling, fault simulation, and reporting of results. A unique feature of 5012 is that it addresses built-in-self-test based on the use of MISRs for test response compaction. A BIST structure based on LFSRs may be fault-graded without the LFSR and the resulting fault coverage can be reported, scaled by a penalty value p based on the degree k of the LFSR. The penalty values are listed in Table XI.

The penalty values are used as follows. Let the measured fault coverage of the CUT, including any stimulus generation logic but *without* the LFSR, be denoted as F. Suppose that the LFSR implements a primitive polynomial of degree k, and there is at least one flip-flop stage between inputs to the MISR. Obtaining from Table XI the value of p that corresponds to k, the fault coverage of the CUT would be reported as $F' = (1 - p)F$. That is, a penalty of $p \times 100\%$ is incurred in assessing the effectiveness of SA if the actual effectiveness is not determined.

The penalty values listed in Table XI were determined empirically from a large number of experiments involving many different CUTs and MISR polynomials. These tabulated values are based on the worst-case observed probabilities of aliasing and do not take into account the size of the CUT. For any practical CUT, the penalty values listed in Table XI are well within the ideal UCL given by (3) and they grow more conservative as the size of the CUT increases.

TABLE XI
Penalty values specified in Procedure 5012 for MISRs implementing a primitive polynomial of degree k

Degree (k)	Penalty value
$k < 8$	1.0
$8 \leq k \leq 15$	0.05
$16 \leq k \leq 23$	0.01
$23 < k$	0.0

CONCLUSIONS

Our experiments have validated the prediction that the average probability of aliasing for a MISR that implements a primitive polynomial of degree k converges to 2^{-k} for the types of errors produced by stuck-at faults in digital logic circuits. An ideal 95% UCL for the loss of fault coverage, based on the assumption of statistically-independent errors, has been shown to match the empirically-obtained results.

The four primitive polynomials considered in this study performed well for all circuits. The four irreducible polynomials performed well except in a few cases. The four reducible polynomials performed poorly under almost all conditions. The tentative indication is that primitive polynomials provide the most consistent performance with respect to aliasing probability. This, together with the weight of the theoretical arguments in favor of primitive polynomials, clearly dictates that only primitive polynomials should be used in LFSR/MISR design.

The 95% UCL on the fraction of faults aliased is a better metric for SA design than the average probability of aliasing. Using this confidence limit, a lower bound on the fault coverage for a given circuit, MISR, and test vector sequence can be determined from the value of the fault coverage obtained without consideration of the MISR. This provides an orders-of-magnitude improvement in CPU time for fault grading. This technique has already been incorporated in MIL-STD-883, Procedure 5012: "Fault Coverage Measurement for Digital Microcircuits."

This study of SA takes a more realistic approach compared to most earlier studies in that it considers the effects of stuck-at faults internal to the circuits-under-test, rather than assuming that statistically-independent errors occur at the circuits' primary outputs. This is an important difference because error-based models do not take into account information that is available about the specific behavior of faulty devices.

References

[1] J.E. Smith, "Measures of the effectiveness of fault signature analysis," *IEEE Transactions on Computers*, 511–514, June 1980.

[2] W.W. Peterson and E.J. Weldon, *Error-Correcting Codes,* 2nd Ed., Cambridge MA: MIT Press, 1972.

[3] P.C. Maxwell, "Comparative analysis of different implementations of multiple-input signature analyzers," *IEEE Transactions on Computers,* 1411–1414, November 1988.

[4] W.H. Debany, P.F. Manno, and N. Kanopoulos, "Built-in-test library," *Proceedings, ATE & Instrumentation Conference West,* 417–435, 1988.

[5] W.H. Debany, K.A. Kwiat, H.B. Dussault, M.J. Gorniak, A.R. Macera, and D.E. Daskiewich, "Fault coverage measurement for digital microcircuits," MIL-STD-883 Procedure 5012, Rome Laboratory (RL/ERDA), Griffiss AFB NY 13441, 18 December 1989 (Notice 11) and 27 July 1990 (Notice 12).

[6] W.H. Debany, M.J. Gorniak, D.E. Daskiewich, A.R. Macera, K.A. Kwiat, and H.B. Dussault, "Empirical bounds on fault coverage loss due to LFSR aliasing," *Proceedings, IEEE VLSI Test Symposium,* 143–148, 1992.

[7] J.L. Carter, "The theory of signature testing for VLSI," *Proceedings, 14th ACM Symposium on Theory of Computing,* 66–76, 1982.

[8] T.W. Williams, W. Daehn, M. Gruetzner, and C.W. Starke, "Comparison of aliasing errors for primitive and non-primitive polynomials," *Proceedings, IEEE International Test Conference (ITC),* 282–288, 1986.

[9] T.W. Williams, W. Daehn, M. Gruetzner, and C.W. Starke, "Aliasing errors in signature analysis registers," *IEEE Design & Test,* 39–45, April 1987.

[10] T.W. Williams, W. Daehn, M. Gruetzner, and C.W. Starke, "Aliasing errors with primitive and non-primitive polynomials," *Proceedings, IEEE International Test Conference (ITC),* 637–644, 1987.

[11] A. Ivanov and V.K. Agarwal, "On a fast method to monitor the behavior of signature analysis registers," *Proceedings, IEEE International Test Conference (ITC),* 645–655, 1987.

[12] D. Xavier, R.C. Aitken, A. Ivanov, and V.K. Agarwal, "Experiments on aliasing in signature analysis registers," *Proceedings, IEEE International Test Conference (ITC),* 344–354, 1989.

[13] D. Xavier, R.C. Aitken, A. Ivanov, and V.K. Agarwal, "Using an asymmetric error model to study aliasing in signature analysis registers," *IEEE Transactions on Computers,* 16–25, January 1992.

[14] M. Damiani, P. Olivo, M. Favalli, and B. Riccò, "An analytical model for the aliasing probability in signature analysis testing," *IEEE Transactions on Computer-Aided Design,* 1133–1144, November 1989.

[15] M. Damiani, P. Olivo, M. Favalli, S. Ercolani, and B. Riccò, "Aliasing in signature analysis testing with multiple input shift registers," *IEEE Transactions on Computer-Aided Design,* 1344–1353, December 1990.

[16] M. Damiani, P. Olivo, and B. Riccò, "Analysis and design of linear finite state machines for signature analysis testing," *IEEE Transactions on Computers,* 1034–1045, September 1991.

[17] S.K. Gupta and D.K. Pradhan, "A new framework for designing & analyzing BIST techniques: computation of exact aliasing probability," *Proceedings, IEEE International Test Conference (ITC),* 329–342, 1988.

[18] J.P. Robinson, "Aliasing probabilities for feedback signature compression of test data," *IEEE Transactions on Computers,* 867–873, July 1991.

[19] T. Sridhar, D.S. Ho, T.J. Powell, and S.M. Thatte, "Analysis and simulation of parallel signature analyzers," *Proceedings, IEEE International Test Conference (ITC),* 656–661, 1982.

[20] J.C. Muzio, F. Ruskey, R.C. Aitken, and M. Serra, "Aliasing probabilities of some data compression techniques," in *Developments in Integrated Circuit Testing,* D.M. Miller, ed. San Diego, CA: Academic Press, 169–217, 1987.

[21] J.P. Robinson, "A unified view of test compression methods," *IEEE Transactions on Computers,* January 94–99, 1987.

[22] A. Ahmad, N.K. Nanda, and K. Garg, "Are primitive polynomials always best in signature analysis?," *IEEE Design & Test of Computers,* August 36–38, 1990.

[23] D.M. Miller, S. Zhang, W. Pries, and R.D. McLeod, "Estimating aliasing in CA and LFSR based signature registers," *Proceedings, IEEE International Conference on Computer Design (ICCD),* 157–160, 1990.

[24] T. Kirkland and M.R. Mercer, "Algorithms for automatic test pattern generation," *IEEE Design & Test of 43-55, Computers.* June 1988.

[25] J.A. Waicukauski and E. Lindbloom, "Fault detection effectiveness of weighted random patterns," *Proceedings, International Test Conference,* 245–255, 1988.

[26] A.J. Duncan, *Quality Control and Industrial Statistics,* 5th ed., Homewood, IL: R.D. Irwin, 1986.

[27] Texas Instruments (TI), *The TTL Data Book for Design Engineers,* 2nd Ed., 1976.

[28] P.H. Bardell, W.H. McAnney, and J. Savir, *Built-In Test for VLSI: Pseudorandom Techniques,* NY: John Wiley & Sons, 216–225, 1987.

[29] M.H. Schulz and E. Auth, "Advanced automatic test pattern generation and redundancy identification techniques," *Proceedings, Fault Tolerant Computing Symposium,* 30–35, 1988.

[30] Advanced Micro Devices (AMD), *Bipolar Microprocessor Logic and Interface Data Book,* 1985.

[31] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in FORTRAN," *Proceedings, International Symposium on Circuits and Systems (ISCAS),* 1985.

Biographies

WARREN H. DEBANY JR. is with the U.S. Air Force Rome Laboratory. He heads the Microcircuit Simulation and Testability Group in the Microelectronics Reliability Division. He received his B.S. in Electrical Engineering from the State University of New York at Buffalo, and M.S. in Computer Engineering and Ph.D. in Computer and Information Science from Syracuse University. His current research interests include digital logic modeling, test generation, the development and assessment of design-for-testability and built-in-test techniques, and testability measurement. He is a Senior Member of the IEEE, and is a licensed Professional Engineer in the State of New York.

MARK J. GORNIAK is with the U.S. Air Force Rome Laboratory. He is an electronics engineer in the Microcircuit Simulation and Testability Group. He received his B.S. in Electrical Engineering from the Rochester Institute of Technology, and his M.S. in Electrical Engineering from Syracuse University. He is managing the DOD qualification programs for monolithic and hybrid microcircuits, MIL-I-38535 and MIL-H-38534, respectively. His current research interests are in assuring the reliability of bare dice, multichip modules, and tape automated bonding techniques. He is a member of the IEEE.

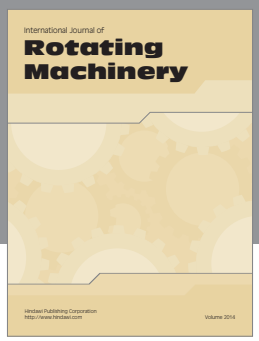
ANTHONY R. MACERA is with the U.S. Air Force Rome Laboratory. He is an electronics engineer in the Interoperability Standards and Testing Group. He received his B.S. in Electrical Engineering from Clarkson University, and is currently pursuing his M.S. in Computer and Information Science at the State University of New York at Utica/Rome. His current research interests include techniques for assuring "known good die" as well as design-for-testability techniques for the testing of multichip modules. He is a member of the IEEE Computer Society.

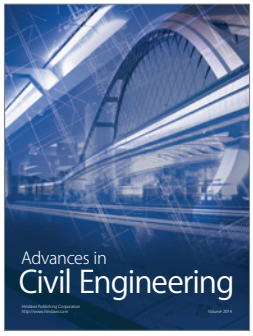
DANIEL E. DASKIEWICH is with the U.S. Air Force Rome Laboratory. He is an electronics engineer in the Microcircuit Sim-

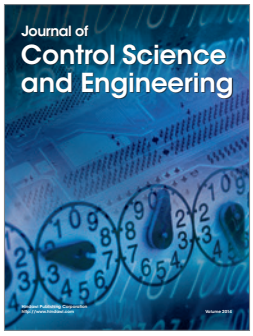
ulation and Testability Group. He received his B.S. in Electrical Engineering from Clarkson University, and is currently pursuing his M.S. in Electrical Engineering at Syracuse University. His current research interests include dependency modeling and analyzing design-for-testability techniques such as boundary scan and BIST for assuring "known good die" for multichip modules.

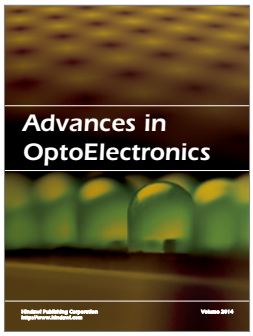
KEVIN A. KWIAT is with the U.S. Air Force Rome Laboratory. He is an electronics engineer in the Microcircuit Simulation and Testability Group. He received his B.S. in Computer Science, B.A. in Mathematics, and M.S. in Computer Engineering from Syracuse University, and is currently pursuing his Ph.D. in Computer Engineering at Syracuse University. He is currently working in the areas of fault simulation, testability, and fault-tolerance. He is a member of the Adjunct Faculty at the Utica College of Syracuse University, and is a member of the IEEE Computer Society.

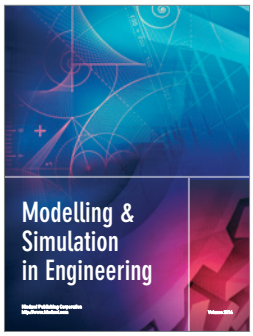
HEATHER B. DUSSAULT is with the U.S. Air Force Rome Laboratory. She is an electronics engineer in the Microcircuit Simulation and Testability Group. She received her B.S and M.S. in Nuclear Engineering from Rensselaer Polytechnic Institute, and is currently pursuing her Ph.D. in Nuclear Engineering at Rensselaer. Her current research interests include single-event phenomena, radiation testing procedures, and testability measurement. She is a member of the IEEE and a student member of the American Nuclear Society.

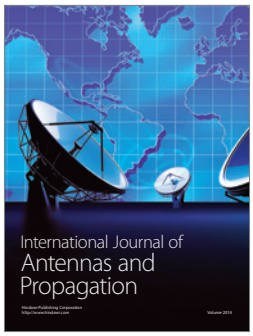


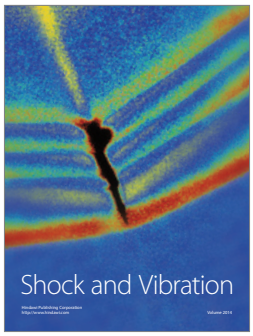


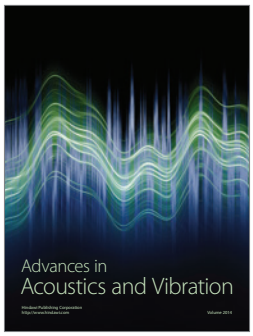












Hindawi

Submit your manuscripts at
http://www.hindawi.com