ESD-TR-84-019

Technical Report
685

VLSI Self-Testing Using Exhaustive Bit Patterns

B.I. Dervisoglu

31 May 1984

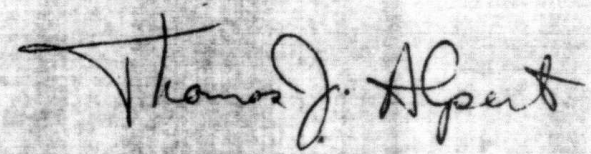
Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Defense Advanced Research Projects Agency
under Electronic Systems Division Contract F19628-80-C-0002.

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Defense Advanced Research Projects Agency under Air Force Contract F19628-80-C-0002 (ARPA Order 3797).

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER



Thomas J. Alpert, Major, USAF
Chief, ESD Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

VLSI SELF-TESTING USING EXHAUSTIVE BIT PATTERNS

B.I. DERVISOGLU

Group 24

TECHNICAL REPORT 685

31 MAY 1984

LEXINGTON MASSACHUSETTS

ABSTRACT

The use of Linear Feedback Shift Register functions in generating exhaustive test case coverage for Very Large Scale Integrated circuits with SCAN/SET capability is presented. Both deterministic and probabilistic approaches to test pattern generation are discussed. A technique for signature generation is presented with analysis of its effectiveness. Also, a technique is described for consolidating the test pattern generation and signature capture functions into a single test/detect capability that requires less built-in hardware for implementation.

CONTENTS

1. INTRODUCTION

The ever increasing densities of digital Integrated Circuit (IC) components have passed the point beyond which traditional testing methods are no longer usable. In the first place, as IC densities increase with little or no increase in the number of external Input/Output (I/O) pins, so does the logic-to-pin ratios of the chips. This makes it more difficult to figure out the appropriate input test patterns that will simultaneously control the internal nodes and allow their logic states to be observed at some I/O pin of the IC. Indeed, as the logic-to-pin ratios increase, it becomes necessary to use sequences of test patterns in place of a single test pattern to perform each controlling/observing task. That is, for complex IC chips the testing problem becomes more sequential in nature. It is well known that Automatic Test Pattern Generation (ATPG) algorithms perform with less efficiency and success rates when the degree of sequentiality of the Device Under Test (DUT) is increased. Hence, even if some major breakthrough were to occur in the area of ATPG algorithms, any resultant gain in performance is likely to be lost due to the increased complexity of the problems being considered. Furthermore, all existing ATPG algorithms require a precise (NOT equivalent) gate level description for the DUT, since they all use fault models which are implementation specific. Whereas such accurate circuit descriptions are available to the component manufacturers, in general, end users do not have access to this kind of proprietary data.

An even more important but many times overlooked factor in choosing test strategies for use in the VLSI domain, is the ratio of the testing

costs to the overall system cost. It should be noted that VLSI implies far more than a mere reduction in geometric dimensions of a digital circuit. As complete systems/subsystems become integrated as a single component, performing testing in the traditional ways would imply that the current system testing costs will translate into future component testing costs. This, of course, is an entirely unacceptable situation. For example, it may be appropriate to test an entire Printed Circuit Board (PCB) using expensive test equipment that can store and search through large volumes of test related data. However, using the same approach to test a single VLSI chip version of the same system would bring testing costs totally out of proportion with the overall cost of the DUT itself. In addition, the widespread use of digital systems in all aspects of our lives have increased the importance of field testing. In many cases, digital systems are being used by people who are not knowledgeable enough to suspect that a faulty condition exists and make the proper judgments to invoke appropriate diagnostic procedures to detect and isolate the problem to the nearest Field Replaceable Unit (FRU).

It is clear from the above discussion that a radically different approach is needed for testing digital systems in the VLSI domain. Indeed, the "difference" must be more than being just a clever technique that enables existing (or improved) ATPG algorithms to perform better. That is, our focus should not only be on improving algorithm efficiency, but we should also improve the testability of VLSI designs through changes in their implementation. A very good start has been made in this direction by the introduction of Level Sensitive Scan Design (LSSD) [1] rules.

Currently, several major digital systems manufacturers are using (variations of) these rules in their designs. Using the LSSD rules enables the designers to eliminate many potential timing problems and makes it possible to implement a "Scan Path" whereby each and every individual bi-stable element in the circuit becomes separately controllable and observable. Generically, this is achieved by configuring the DUT such that, for testing purposes, all of its latches become part of a single shift register, called the scan register. The serial data input and output terminals of the scan register are made accessible from two of the external I/O pins of the device. Then, any combination of bit values can be loaded into the scan latches by serially shifting the desired combination into the scan register. The values stored in the scan latches act as input patterns to the combinational part of the DUT. During testing, first the desired bit pattern is shifted into the scan register. Next, the combinational circuit outputs are latched (in parallel) into the scan register. Finally, as the next bit pattern is being shifted in, the results of the previous test become available at the output of the scan register, one bit at a time. This technique, which is commonly referred to as the SCAN/SET technique, reduces the problem of testing a complex digital system to that of testing only the combinational part of its circuitry. However, apart from increasing the controllability/observability of the internal nodes of a system, SCAN/SET does not offer a new approach to the ATPG problem.

Given that modern VLSI systems are capable of operating at very high clock rates, a natural extension of the SCAN/SET approach is to drop the ATPG altogether and exercise the combinational part of a digital system

exhaustively. Despite the potentially very large size of such combinational circuits, exhaustive testing appears to be feasible. This can be seen by observing that a multi-input/multi-output combinational circuit consists of multiple single-output circuits, each of which may receive inputs from only a subset of the bits of the scan register. For example, a scan register may have several thousand bits but any single-output logic cone may use, say, only 30 of these as its inputs. Then, if a new input test pattern can be generated at every clock period, applying all 2^{30} bit permutations to such a logic cone would take less than two minutes if the clock rate is 10 MHz. However, to achieve this it is necessary that test results (i.e., combinational circuit output values) should not be latched back into the scan register, as this prevents us from generating a new test pattern with each clock. Instead, we can utilize a separate "signature" register where test results can be accumulated. This technique will be discussed later in detail.

The rest of this report is arranged as follows: Section 2 presents the underlying theory and proposes a deterministic approach whereby exhaustive testing of the combinational portion of a digital system is assured. This section also presents a technique for generating/storing a signature for interpreting the test results. Section 3 presents a probabilistic approach where exhaustive testing may become possible with some probability of success. The advantage of the probabilistic approach is that it avoids the design specific computations that are necessary to implement the deterministic approach. Then, Section 4 presents an extension to the probabilistic approach whereby using an explicit signature register becomes no longer necessary. The report ends with a conclusion section.

2. EXHAUSTIVE TEST PATTERN GENERATION

Figure 1 depicts the general model for a digital system which has SCAN/SET capability. To enable the generation of exhaustive input test patterns for the individual logic cones, the serial data input to the scan register will be tied to the serial output of a prime Linear Feedback Shift Register (LFSR). A prime LFSR of degree n (i.e., has n bits) is capable of generating all of the 2^n-1, non-zero n-bit permutations if it is started in any non-zero state and 2^n-1 shifts are performed. Such LFSR functions are known to exist for any degree n (e.g., see [2]).

Definition 1: The Characteristic Polynomial, G(X), for an LFSR is given by

$$G(X) = 1 + a_1X + a_2X^2 + a_3X^3 + \ldots + a_nX^n \qquad (1)$$

where a_i belong to $\{0,1\}$ for all $1 \leq i \leq n$, and "+" means EXCLUSIVE-OR.

Note:

Through the remainder of this report, the "+" sign appearing in symbolic expressions will be used to mean EXCLUSIVE-OR (modulo 2 sum), unless otherwise stated. Coefficients a_i of the characteristic polynomial for an LFSR indicate the bit positions (bit 1 is the leftmost one when the shift direction is to the right) whose EXCLUSIVE-OR determines the next value of the first bit position, when the next shift operation takes place.

Without any loss of generality, the combinational part of the circuit shown in Fig. 1 can be considered as an s-input/m-output circuit, which is made from m-many single output logic cones. Let "t" be the maximum number of taps (inputs) that any single output logic cone may receive from the scan register. Since G(X) generates all non-zero n-tuples which are

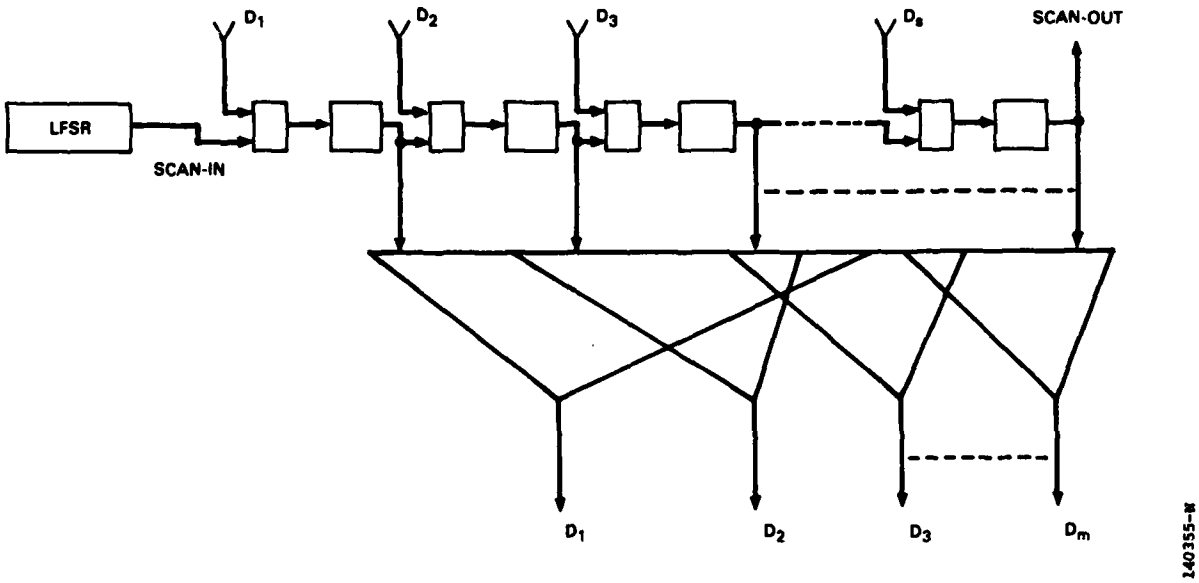


Fig. 1. Logical equivalent of circuit with scan/set capability.

shifted along the scan register, it follows that when the t-input taps of any logic cone all fall within some consecutive n bits of the scan register, that cone will be exercised under all possible non-zero t-tuples. At this point we can assume that the missing all-zero t-tuple is a special case that can be handled separately. Note that with each new shift the entire scan register will change so that every consecutive set of n bit positions will receive a new n-bit permutation not previously encountered across those bits. Then, taking into account the length of the scan register, the length of the input test sequence necessary for exhaustive testing is

$$l = s + 2^n - 1 \qquad (2)$$

Whereas this technique provides complete test case coverage for all logic cones whose span (distance in number of bits of the scan register between the first and last input tap positions) does not exceed n, we must also consider the case when the span of a logic cone exceeds the degree of the input LFSR. In this case, the first restriction that must be imposed is that

$$s \leq 2^n - 1 \qquad (3)$$

This is necessary since the input LFSR will start repeating its output sequence after 2^n-1 shifts so that if $s > 2^n-1$, then bits i and $i + 2^n-1$ will always have the same value for any i. Thus, let $s \leq 2^n-1$. To see how such a system may behave, consider the following example:

Example: Let $G(X) = 1 + X + X^3$ be the characteristic polynomial for an LFSR whose serial outputs are shifted along a $2^3-1 = 7$ bit scan

register. Assume that initially the LFSR and the scan register bits are all set to zero, except for the output bit of the LFSR, which is set to 1. Table I shows the values of the LFSR and the scan register after each successive shift operation. It is seen that following proper initialization of the scan register which occurs after s ($s = 2^3-1$) shifts, the scan register attains 2^3-1 distinct states, after which it starts to repeat itself.

Let M be the $(2^n-1) \times (2^n-1)$ matrix whose rows corresponds to the rows of Table I, starting with row 2^n-1 (for the above example, n=3). Then, the following properties can be stated:

- The rows of M represent all of the distinct states of the scan register. The same is also true for the columns of M.

- Each row or column of M has exactly 2^n, 1-bits and 2^n-1_n, 0-bits.

- The n-bit columns of any consecutive n rows of M correspond to all non-zero n-tuples.

From Table I we can observe that tap positions 1, 2 and 7 of the scan register will receive all non-zero 3-tuples even though their span is 7 (7-1+1). On the otherhand, the same is not true for positions 1, 2 and 4 even though their span is smaller (i.e., 4). Furthermore, we observe that positions 1, 2 and 4 see only those input 3-tuples that have even parity (i.e., 000, 110, 101, 011). These observations can be formalized by the following theorems which are stated without their proofs:

Theorem 1: Let $R = \{r_1, r_2, \ldots, r_t\}$, $1 \leq r_1 \leq 2^n-1$, and $n \geq t$, be a set of tap positions along a scan register which is driven by a degree n

TABLE I

SHIFT SEQUENCES FOR $G(X) = 1 + X + X^3$

SHIFT CYCLE	LFSR	SCAN REGISTER
0	001	0000000
1	100	1000000
2	110	0100000
3	111	0010000
4	011	1001000
5	101	1100100
6	010	1110010
7	001	0111001
8	100	1011100
9	110	0101110
10	111	0010111
11	011	1001011
12	101	1100101
13	010	1110010
14	001	0111001

prime LFSR function, G(X). The set R of tap positions receive all possible 2^t-1, non-zero t-tuples if, and only if, no proper subset of columns $r_1, r_2, \ldots r_t$ of M are linearly dependent. Furthermore, only even parity bit permutations will appear across any subset of tap positions whose corresponding columns in the M matrix are linearly dependent. (A set of columns of M form a linearly dependent set if, and only if, their bitwise EXCLUSIVE-OR is equal to all zeros.)

Lemma 1: If the set R of tap positions are linearly independent and $t < n$, then the all-zero t-tuple will also appear across these tap positions.

Theorem 2: Let A be an n X (2^n-1) matrix whose n rows correspond to any consequtive n rows of M. Then, a set of columns of A are linearly dependent if, and only if, the corresponding set of columns of M are also linearly dependent. Hence, linear dependence can be tested using any n successive rows of M.

The above theorems establish the necessary conditions for a logic cone with t-taps to receive exhaustive input patterns. Furthermore, Theorem 2 provides a straightforward method for testing a given set of tap positions for compliance with Theorem 1. Thus, for any given set of tap positions one can search through the known prime LFSR functions until one that satisfies the linear independence condition is found. Using that LFSR will enable the exhaustive testing of the combinational part of the DUT.

2.1 Interpreting the Test Results

As each test pattern appears across the scan register, the individual logic cones produce their corresponding outputs. Let there be m ($m \leq s$) logic cones. These output values can be captured and "merged" with the

previous values using an m-bit parallel input signature register. A parallel input signature register is obtained by adding EXCLUSIVE-OR gates between each stage of a prime LFSR. This is illustrated in Fig. 2. In this way, the next state of the signature register is determined by the result of taking the EXCLUSIVE-OR of the next state of a prime LFSR and the incoming parallel data. Thus, if the parallel inputs to the signature are constantly kept at the logic 0 levels, the signature register behaves like an ordinary prime LFSR. On the other hand when the parallel inputs are obtained from the outputs of a combinational circuit, the signature register acts as a hashing circuit that combines previous results with the new one in a pseudo-random manner. Since the hashing is dependent on the parallel input values, a different next state (and hence, a different final state) will result for every different sequence of combinational circuit output values. This enables the detection of circuit failures by comparing the final state (signature) of the signature register to its final state when there are no faults. The fault-free (good network) signature can be determined via simulation or experimenting with several copies of the same network.

Unfortunately, the signature mechanism is not foolproof since it is possible that a fault condition may affect the values of the intermediate states of that register but the final signature may be the same as that for the good network. However, the probability for this to happen can be computed and controlled. Consider the state table for a signature register which has m bits of storage and m parallel inputs. This table has 2^m rows and 2^m columns, corresponding to all possible state and parallel

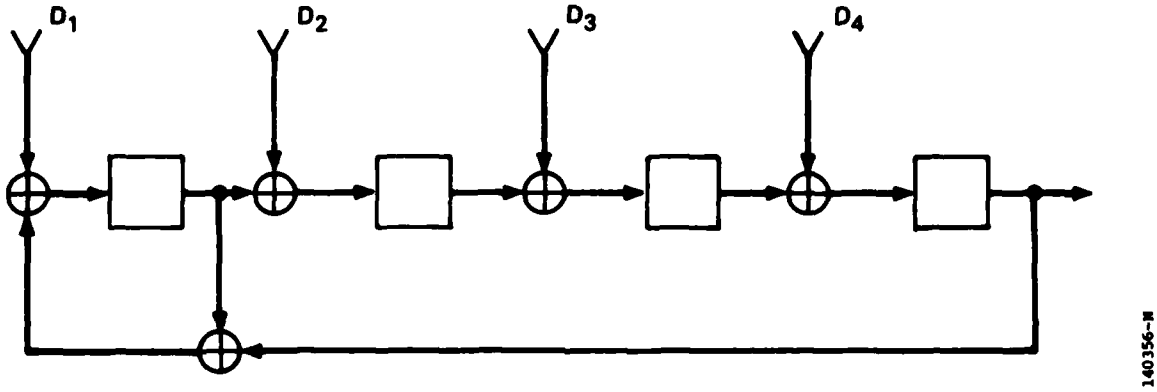


Fig. 2. A 4-bit parallel-input signature register.

input encodings, respectively. Let $S_1,S_2,S_3,\ldots,S_i$ and $I_1,I_2,I_3,\ldots,I_i$ be labels representing the encodings for the rows and columns, respectively, such that subscript "i" represents the particular encoding for that row or column. Let $g(S_i,I_j)$ be the next state of the signature register when the present state and input are S_i and I_j, respectively. It is clear that entries in column I_0 correspond to the next states of a prime LFSR. Thus, $g(S_i,I_0)$, $1 \leq i \leq 2^m$, comprise all non-zero encodings of m bits and $g(S_0,I_0) = S_0$. That is, all possible state encodings are present in column I_0. Entries in any other column I_j are determined by

$$g(S_i,I_j) = g(S_i,I_0) + I_j \qquad (4)$$

It is then obvious that all possible state encodings are present in every row S_i or column I_j of the state table. This implies that the signature register corresponds to a fully-connected sequential machine which also has self-loops on every state.

Now, assume that for the fault-free case, the signature register is started in state S_a and after a test sequence of length k, its final state becomes S_b. We can formulate the number of possible parallel input sequences of length "k" that will result in the same final state if the initial state is S_a. There is only one m-bit input sequence of length k (say, I_t) such that $g(S_a,I_t) = S_b$. For a desired input sequence of length 2, we can use the first one (say, I_f) to transfer to any one of the 2 states and then use the appropriate sequence of length k to reach state S_b. Thus, there are 2^m input sequences of length 2 to reach state S_b from state S_a. Continuing in this fashion it can be shown that there are

$$N(m,k) = 2^{m(k-1)} \tag{5}$$

input sequences of length "k" between states S_a and S_b. On the other hand, the total number of length "k" input sequences of m-tuples is 2^{mk}. Therefore, the probability that a randomly selected input sequences will also bring the signature register into the same final state S_b, is given by

$$P(m,k) = \frac{2^{m(k-1)}}{2^{mk}} = 2^{-m} \tag{6}$$

It can be shown that Eq. (6) is still valid when the number of bits of the register is increased to become greater than the number of parallel inputs (m). Note that even when a failure occurs, it means that only a few faults will not be detected.

The signature technique can also be employed using a serial input signature register. Thus, if the combinational circuit outputs are captured in parallel by the scan register and shifted out into a single input signature register while the next test pattern is being loaded into the scan register, fault detection probability remains the same. However, this technique would result in longer test times as well as requiring a different mechanism for generating the desired test patterns.

3. PROBABILISTIC APPROACH TO EXHAUSTIVE PATTERN GENERATION

Being motivated by a desire to find a general solution to the VLSI self-testing problem such that applicability of the solution to specific designs does not require individual analysis, the following question is investigated:

- Given G(X), a prime (LFSR) of degree n, whose output feeds a scan register with s ($s \leq 2^n-1$) bits and given an arbitrarily selected set of tap positions $R = \{r_1, r_2, \ldots, r_t\}$, where $t \leq n$, what is the probability that some proper subset of the t-taps will be linearly independent?

Solution: Assume that k-1 of the t-taps form a linearly independent set. We can first formulate the probability that a randomly selected kth tap will form a linearly dependent set with some subset of the k-1 tap positions. There are $2^{k-1}-1$ non-empty subsets of the k-1 taps. Since $s \leq 2^n-1$, no pair of taps can be linearly dependent. Therefore, the number of subsets of the k-1 taps that tap r_k may be linearly dependent with is $2^{k-1}-(k-1)$, or, $2^{k-1}-k$. Also, since k-1 many taps have already been selected, there are $2^n-1-(k-1) = 2^n-k$ tap positions where r_k can be selected from. Therefore, the probability that r_k will be linearly dependent with some subset of the first k-1 taps is given by

$$P(n,k)' = \frac{2^n - k - 2^{(k-1)} + k}{2^n - k} = \frac{2^n - 2^{(k-1)}}{2^n - k} \tag{7}$$

Since P(n,k)' has been formulated assuming that the first k-1 taps are linearly independent, the probability that the t-taps are also linearly independent is given by

$$\text{Prob}(n,t) = P(n,1)' \times P(n,2)' \times \ldots \times P(n,t)' \tag{8}$$

and the probability that some subset of the t-tap positions are linearly dependent is

$$F(n,t) = 1 - \text{Prob}(n,t) \tag{9}$$

Table II lists F(n,t) for various values of n and t. From Table II it is observed that the probability of picking a linearly dependent set of tap positions depends on n-t, so that given the maximum number of tap positions (t) for any logic cone, we can pick the degree (n) of the LFSR that will produce a probability value less than some arbitrary threshold. Furthermore, we can improve upon the values shown in Table II by employing more than one independent LFSR to generate the scan patterns. This way if the output sequences from q independent prime LFSR generators of degree n are cascaded and fed to the scan register, a set of t-tap positions which form a linearly dependent set under $LFSR_i$, may not be linearly dependent under some other generator, say, $LFSR_j$. In this case, the probability that all of the generators will fail to exercise those tap positions is given by

$$F(n,t,q) = F(n,t)^q \qquad (10)$$

Table III lists the values of F(n,t,q) in terms of n-t and q. The advantage of using more than one LFSR is that this may lead to a shorter test period. For example, if t = 20, then we can use 4 LFSRs of degree 21, which results in the probability of failure = 0.03. In this case the test length will be $4 \times 2^{21} = 2^{23}$. However, to achieve the same failure probability using a single LFSR would require $n-t \geq 5$, or n = 25. This would result in a test sequence whose length is 2^{25}.

The above presentation shows that it is possible to adopt a probabilistic approach to the selection of a single (or multiple LFSR) function to exercise the combinational logic under all possible input combinations. Whereas this approach carries the natural risk of finding

TABLE II

VALUES OF F(n,t)

LFSR DEGREE	n - t 0	1	2	3	4	5
n = 10	0.69	0.40	0.20	0.09	0.04	0.02
11	0.70	0.41	0.21	0.10	0.05	0.02
12	0.71	0.41	0.22	0.11	0.05	0.02
13	0.71	0.42	0.22	0.11	0.06	0.03
14	0.71	0.42	0.23	0.12	0.06	0.03
..	....	....	....	....	....	....
20	0.71	0.42	0.23	0.12	0.06	0.03

TABLE III

VALUES OF F(n,t,q)

NO. OF LFSRs	n - t 0	1	2	3	4	5
q = 1	0.71	0.42	0.23	0.12	0.06	0.03
2	0.51	0.18	0.05	0.01	0.00	0.00
3	0.36	0.08	0.01	0.00	0.00	0.00
4	0.26	0.03	0.00	0.00	0.00	0.00
5	0.18	0.01	0.00	0.00	0.00	0.00
6	0.13	0.01	0.00	0.00	0.00	0.00
7	0.09	0.00	0.00	0.00	0.00	0.00
8	0.07	0.00	0.00	0.00	0.00	0.00

oneself on the undesirable side of the probability distribution, it has the advantage of being straightforward and does not require special analysis of the given circuit. Furthermore, as far as permissible, the test length can be made longer by choosing a higher degree n for the LFSR to increase the chances of success. It should also be remembered that even when a logic cone may not be fully exercised, it may still be fully tested by the portion of the exhaustive patterns that it may have received.

4. EXTENSION/CONSOLIDATION OF THE PREVIOUS TECHNIQUES

Let us examine more closely the state transitions in a parallel input signature register. Assume that the parallel input combinations received by the signature register have a uniform probability distribution. From Eq. (4) we have the next state of the signature register given by

$$S_b = g(S_a, I_j) = g(S_a, I_0) + I_j$$

where S_a is the current state and I_j is the current m-bit parallel input. For any possible value of $g(S_a, I_0)$ there exists exactly one value of I_j such that $I_j = g(S_a, I_0)$, that is, $S_b = S_a$. Thus, during a test sequence of length "k", the signature register may go through some set of states more than once. Let P(k,r) represent the probability that exactly r distinct states have been visited after "k" shifts. Then, one of the following two conditions must be true:

1 - Exactly r distinct states have been visited during the first k-1 shifts and the state after the rth shift is a repeat of some state already visited, or

2 - Exactly r-1 distinct states have been visited during the first k-1 shifts and a previously unvisited state is reached with the kth shift. Then, we can write

$$P(k,r) = P(k-1,r) \cdot P(R) + P(k-1,r-1) \cdot P(\approx R) \quad (11)$$

where, P(R) and P(≈R) represent the probabilities that the next state after the kth shift does or does not belong to the previously visited r or r-1 states, respectively. It is obvious that

$$P(R) = \frac{r}{2^m} \quad \text{and} \quad P(\approx R) = \frac{2^m - r + 1}{2^m} \quad (12)$$

Furthermore, since the signature register has already visited one distinct state (i.e., its initial state) before any shifting is done, we have

$$P(0,1) = 1 \quad \text{and} \quad P(i,0) = 0 \text{ for any } i > 0. \quad (13)$$

Equation (11) represents a system of equations which can be solved using a computer program to compute P(1,1), P(1,2),P(1,3),.... for various different values of m. These can then be used to compute the expected value of the number of distinct states visited in "k" shifts of an m-bit signature register, using

$$e(k) = \sum_{r=1}^{k} r \times P(k,r) \quad (14)$$

An alternate formulation for e(k) is also given by

$$e(i) = e(i-1) + \frac{2^m - e(i-1)}{2^m} \quad (15)$$

which should be computed for i = 1,2,3,...,k, with the boundary condition that e(0) = 1. Evaluating either Eq. (14) or (15) reveals that if the test length "k" is chosen to be a multiple of 2^m, i.e., $k = hx2^m$, then the percentage of total states visited is independent of m. This is shown in Table IV.

TABLE IV

PERCENTAGE OF TOTAL (2^m) STATES VISITED IN "k" SHIFTS

TEST LENGTH (k)	value of h for test length = h x 2^m					
	1/4	1/2	1	2	3	4
m = 32	21.88	40.62	62.50	84.38	93.75	96.88
128	22.66	39.84	63.28	85.94	94.53	97.66
512	22.27	39.45	63.28	86.33	94.92	98.05
1024	22.17	39.36	63.18	86.43	95.02	98.14
2048	22.12	39.36	63.23	86.47	95.02	98.14
4096	22.12	39.36	63.21	86.45	95.02	98.17

As can be seen from Table IV, a signature register will go through 98% of all of its states if it receives $4x2^m$, m-bit parallel inputs which are selected with uniform distribution. Furthermore, if the input patterns are uniformly distributed, then any subset of bits of the entire signature register are expected to go through the same percentage of combinations applicable across that subset. This can be seen from Eq. (4) and noting that if the probability distribution of m-tuples I_j is independent from the probability distribution of m-tuples $g(S_i I_0)$, when both distributions are uniform, then the probability distribution of $g(S_i, I_j)$ is also uniform. This makes the signature register itself become a good candidate for use in generating the input test patterns applied to the combinational logic cones. To achieve this, one would eliminate the signature register all together and load the combinational circuit outputs, in parallel, into the scan register. Then, if the maximum number of taps for any logic cone is t, the required test length would be selected as $4x2^t$, for an expected 98% exhaustive coverage of all t-tuples for any logic cone. However, there is an obvious flaw in this argument since, in practice, the output values produced by the combinational logic are not uniformly distributed. Indeed, in many cases an m-input/m-output combinational circuit may produce much less than 2^m distinct output m-tuples in response to all possible input m-tuples. This is because several different input m-tuples may generate the same output values. Let the number of distinct combinational circuit output m-tuples be q. This means that the state transitions of the signature register will be confined to q columns of its state table and, in effect, from each internal state

only q of the total states will be accessible. Then, Eq. (15) which gives the expected number of distinct states that will be visited, must be modified as

$$e(i) = e(i-1) + \frac{2^m - e(i-1)}{2^m} \times \frac{q}{2^m} \qquad (16)$$

which indicates that fewer distinct states of the signature register will be visited. The solution to this problem is to pass the sequence of m-tuples through what might be called a "pattern amplifier" circuit which maps a sequence of q-many m-tuples into another sequence of length "1", while preserving the information content of the input m-tuples. This of course implies that the pattern amplifier circuit will be a sequential circuit, since repeated occurrences of the same input m-tuple must be mapped into different output m-tuples. One simple way to implement such a circuit is to use a hashing circuit whose outputs will be controlled by the input m-tuples. The hashing circuit can be implemented using a prime LFSR whose degree should be comparable to $2^m/q$, so that sufficiently large numbers of output m-tuples can be generated. Thus, if we let $Q_1, Q_2, Q_3, \ldots, Q_p$ be the individual bits of a prime LFSR and $D_1, D_2, D_3, \ldots D_m$ be the m-outputs from the combinational circuit, the pattern amplifier circuit can be implemented using

$$D_1' = D_1 + Q_1$$

$$D_2' = D_2 + Q_2$$

$$D_{p+1}' = D_{p+1} + Q_1$$

$$D_{p+2}' = D_{p+2} + Q_2$$

etc.

This is illustrated in Fig. 3. Finally, Fig. 4 illustrates the system architecture to implement the test technique described in this section. Note that an input LFSR is no longer needed to feed the scan register since the scan register already has its own feedback signal defined as part of its role as a signature register. The hashing LFSR used in the pattern amplifier circuit must be chosen to be independent of any other circuit in the system in order to prevent adverse effects that may result otherwise.

To implement the technique described in this section we can use a modified scan-latch with multiple clocks, as shown in Fig. 5. The advantage of using this latch is that (1) all test related circuitry is placed along the L2 latch, away from the inputs of the L1 latch, which is used to implement the original functions of the DUT; (2) an independent scan-path is retained so that specific test patterns can be fed to the circuit, if desired.

5. CONCLUSIONS

It is no longer justifiable to use the area of a chip exclusively for implementing the primary functions of a circuit. As we move into the VLSI domain, the testability requirement becomes at least as important as any other requirement. This report has presented a basic technique and some variations that would help solve testing related problems for VLSI components and/or digital systems that use such components. It is inevitable that this or any similar technique will be met with skepticism and resistance by some designers. However, it is the author's firm belief that, in the long run, techniques having the same flavor as the ones described here will be the predominant ones used in building reliable/testable digital systems.

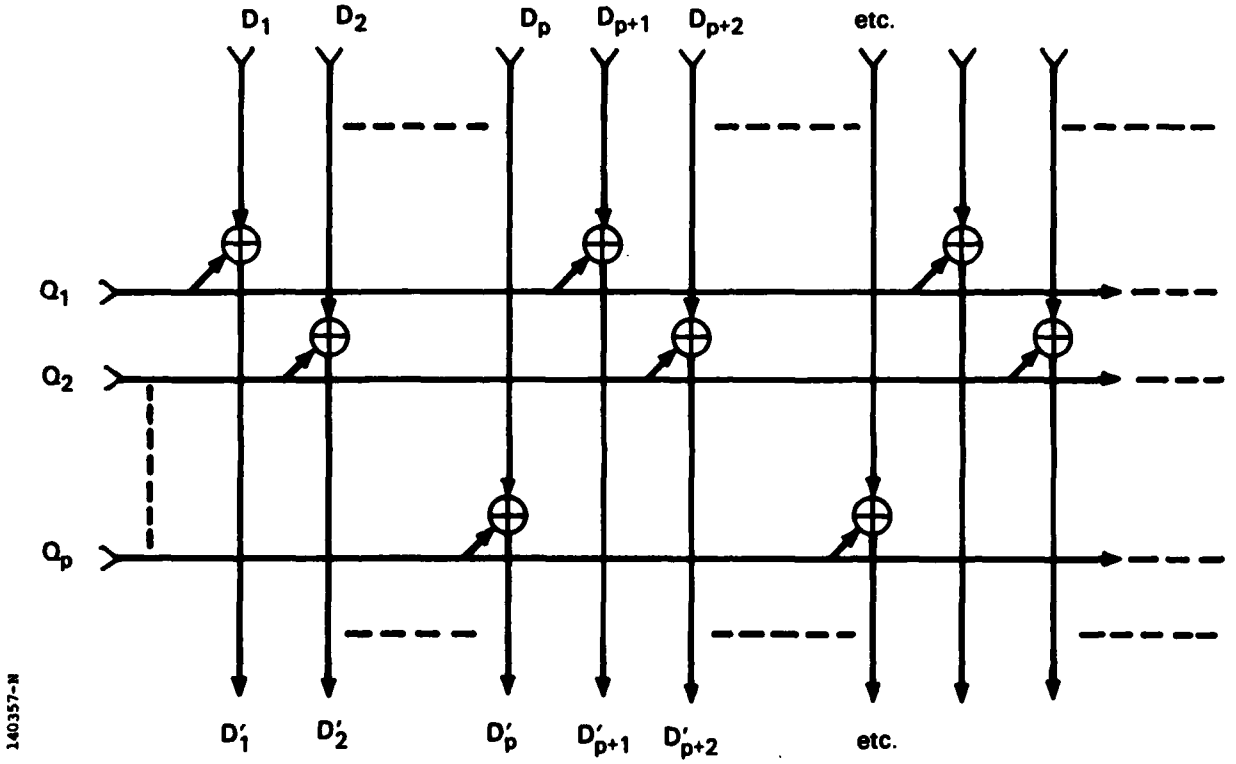


Fig. 3. Hashing the combinational circuit outputs.

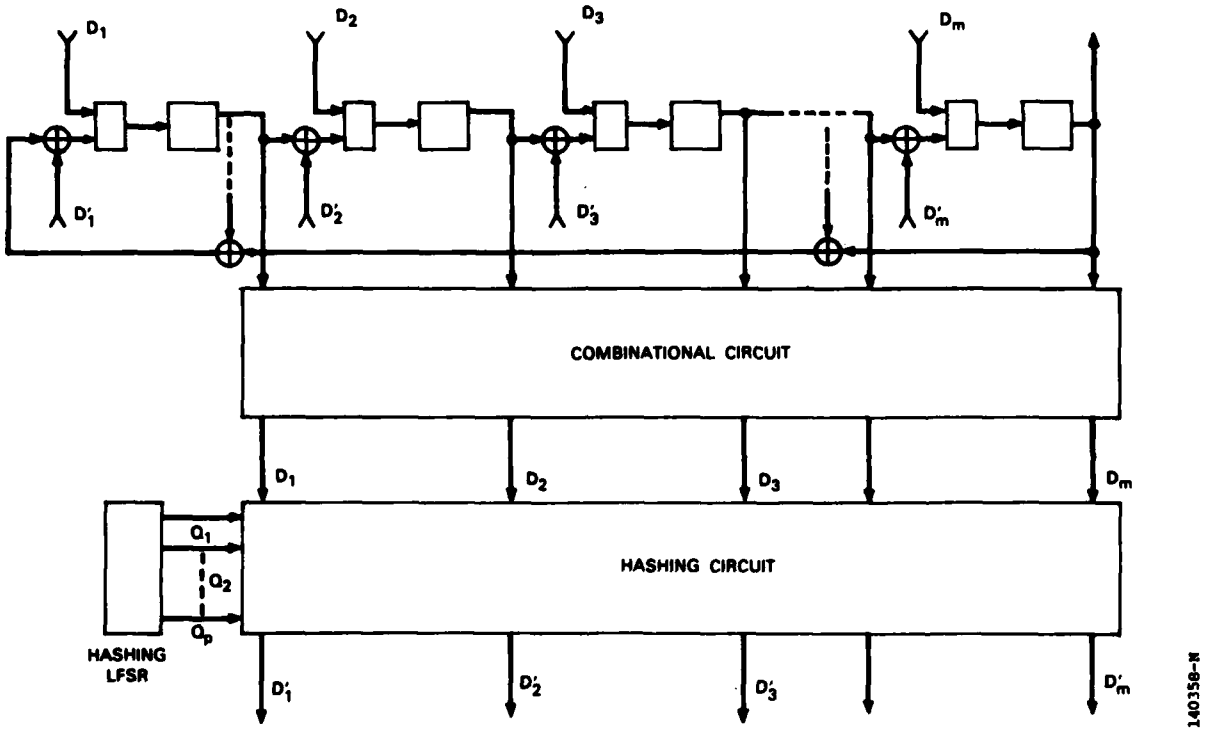


Fig. 4. System architecture for randomized self-test technique.

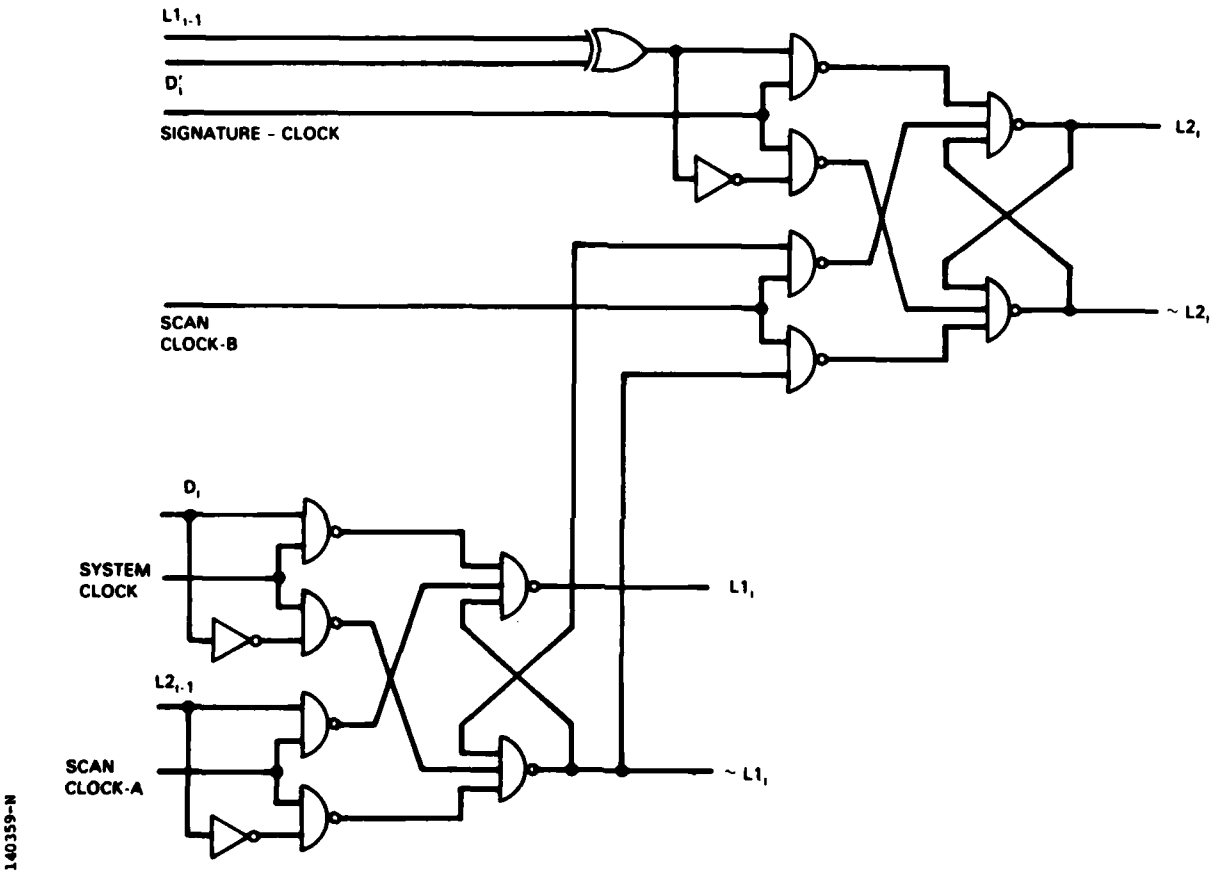


Fig. 5. Modified scan-latch with signature capability.

On the other hand, it should not be thought that what is presented here is a complete and universal solution to all of our testing needs. There are many areas (such as memory testing) where other test methods must be used to do the job.

ACKNOWLEDGEMENTS

The work presented here had been started at the Sperry Research Center during the author's employment there. The original problem formulation is due to Dr. T. A. Welch, who also formulated the probability equation given in Eqs. (7)-(9). Theorems (1) and (2) were originally stated by Dr. M. Cohn and Dr. A. Lempel, who also provided their proofs (not included here). The author would also like to acknowledge the contributions made by A. Domic of M.I.T. Lincoln Laboratory, for his formulation of Eq. (11).

REFERENCES

[1] E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testability," Proc. 14th Design Automation Conf., New Orleans, LA, June 1977.

[2] S. Golomb, Shift Register Sequences, Revised Ed. (Aegean Park Press, Laguna Hills, CA, 1982).

[3] Z. Barzilai, D. Coppersmith, and A. Rosenberg, "Exhaustive Bit Pattern Generation in Discontiguous Positions with Applications to VLSI Self-Testing," IBM Research Report No. RC-8750 (#38299), IBM T. J. Watson Research Center (19 March 1981).

[4] A Lempel and M. Cohn, "Design of Universal Test Sequences for VLSI," Sperry Research Center Research Report, SRC-RR-82-1 (January 1982).

[5] T. A. Welch, "On-Chip Test Generation for VLSI," Sperry Research Center Research Report, SRC-RR-81-48 (August 1981).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-84-019	2. GOVT ACCESSION NO. AD-A144299	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* VLSI Self-Testing Using Exhaustive Bit Patterns		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER Technical Report 685
7. AUTHOR(s) Bulent I. Dervisoglu		8. CONTRACT OR GRANT NUMBER(s) F19628-80-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173-0073		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order 3797 Program Element No. 61101E Project No. 4D30
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209		12. REPORT DATE 31 May 1984 13. NUMBER OF PAGES 38
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* Electronic Systems Division Hanscom AFB, MA 01731		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

VLSI testing	shift register
self-test	signature generation
linear feedback	SCAN/SET

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The use of Linear Feedback Shift Register functions in generating exhaustive test case coverage for Very Large Scale Integrated circuits with SCAN/SET capability is presented. Both deterministic and probabilistic approaches to test pattern generation are discussed. A technique for signature generation is presented with analysis of its effectiveness. Also, a technique is described for consolidating the test pattern generation and signature capture functions into a single test/detect capability that requires less built-in hardware for implementation.

DD FORM 1 Jan 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*