10. COMPRESSION TECHNIQUES

About This Chapter

In previous chapters we have considered a conventional testing approach, which involves a bit-by-bit comparison of observed output values with the corrected values as previously computed and saved. This approach requires a significant amount of memory storage for saving the correct outputs associated with all test vectors. In this chapter we consider an alternative approach, which is simpler and requires less memory storage. In this approach the information saved is a compressed form of the observed test outcome, called a *signature*. A circuit is tested by comparing the observed signature with the correct computed signature. The process of reducing the complete output response to a signature is referred to as *response compacting* or *compressing*.

This concept is illustrated in Figure 10.1. A fault is detected if the signature $S(R')$ obtained from the circuit under test (CUT) differs from the precomputed signature $S(R_0)$ of a fault-free circuit.

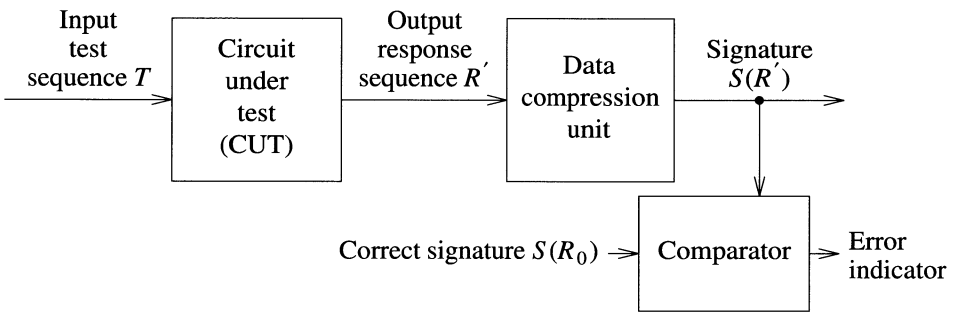


Figure 10.1 Testing using test-response compression

By making the compression circuitry simple, it is possible to embed this function within the circuit being tested, thus realizing one important aspect of *built-in self-test* (BIST), namely detection of response errors.

This chapter describes five compression techniques, namely *ones counting*, *transition counting*, *parity checking*, *syndrome checking*, and *signature analysis*. Since signature-analysis techniques have become extremely popular, they will be discussed in detail. Chapter 11 deals with the use of data-compression techniques in BIST circuits.

10.1 General Aspects of Compression Techniques

A reliable and practical compression technique should be easily implemented by a simple circuit, which, if desired, can be included in the CUT as part of its BIST logic. Furthermore the compression procedure should not introduce signal delays that affect

either the normal behavior or the test execution time of the CUT. In addition the length of the test signature should be a logarithmic factor of the length of the output-response data so that the amount of storage is significantly reduced. And for any fault and associated output response containing one or more errors, the signatures of the faulty and fault-free cases should not be the same. This is required to ensure that the compression method does not lose information.

There is no known compression procedure that satisfies all these requirements. Of particular difficulty is insuring that the faulty and fault-free signatures are different, since a fault may produce offsetting errors and hence go undetected. This situation is referred to as *error masking*, and the erroneous output response is said to be an *alias* of the correct output response. There are three common ways to measure the masking characteristics associated with a compression technique. Some also provide information on the associated fault coverage.

The first way is to simulate the circuit and compression technique and determine which faults are detected. This method requires fault simulation and hence is computationally expensive, especially if the test sequence is long.

A second method classifies the output-response sequences from faulty circuits into categories, such as single-bit error or burst errors, which are errors that lie within a fixed number of patterns of one another. This set of error patterns is then analyzed to determine the degree of masking associated with various compression procedures.

A third method of assessing masking is to compute the fraction of all possible erroneous response sequences that will cause masking. To obtain a realistic result one needs to know the distribution of erroneous sequences. This often is impractical to determine. Usually all possible output sequences are assumed to be equally likely. The problem with this technique is that it is not possible to correlate the probability of obtaining an erroneous signature with fault coverage.

Because the degree of masking is so hard to measure in arbitrary circuits with arbitrary test sets, some compression techniques restrict the configuration of the CUT or require special test sets or both. Most common BIST compression methods fall into one of the following categories:

1. Those that do not require special test sets or circuit designs.
2. Those that require special test sets.
3. Those that require special test sets and circuit designs. Examples of these methods can be found in Chapter 14 dealing with PLAs.

To get the same signature for multiple copies of a fault-free circuit, all sequential circuits are assumed to be initialized to some fixed state before applying a test sequence.

One further problem associated with response-compression techniques is that of calculating the good signature. One way to obtain this signature is to identify a good part, apply the actual test to the part, and have the compression hardware generate the signature. Another approach is to simulate the CUT and compression procedure using the actual test patterns. If the number of test vectors is large, this process can be computationally expensive. Still another technique is to produce many copies of a CUT

and attempt to deduce the correct signature by finding a subset of circuits that produce the same signature. These circuits are assumed to be fault-free.

10.2 Ones-Count Compression

Assuming a single-output circuit C, let the output response of C be $R = r_1, r_2, ..., r_m$. In *ones counting*, the signature $1C(R)$ is the number of 1s appearing in R, i.e.

$$1C(R) = \sum_i r_i$$

where

$$0 \le 1C(R) \le m.$$

The compressor is simply a counter and the degree of compression is $\lceil \log_2(m+1) \rceil$. Figure 10.2 shows a simple circuit, an exhaustive test set, and the response data corresponding to the fault-free and two faulty circuits. Figure 10.3 shows how this circuit can be tested via ones-count compression and the corresponding signatures. Here R_0 refers to the fault-free response and R_i, $i > 0$, to the response corresponding to fault f_i.

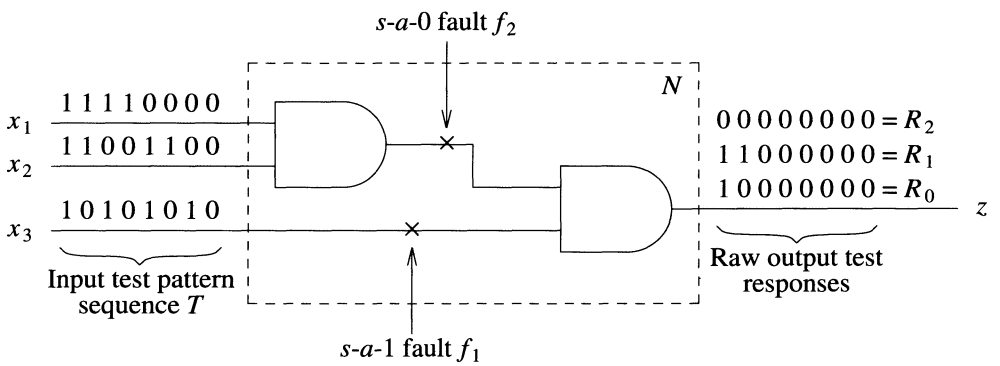


Figure 10.2 Input/output response data

Consider a circuit tested with m random input vectors and let $1C(R_0) = r$, $0 \le r \le m$. The number of m-bit sequences having r 1s is $\binom{m}{r}$. Thus $\binom{m}{r} - 1$ such sequences are aliases. The ratio of masking sequences to all possible erroneous sequences, given that $1C(R_0) = r$, is

$$P_{1C}(M \mid m,r) = \frac{\binom{m}{r} - 1}{2^m - 1}.$$

If all $2^m - 1$ error sequences are equally probable, which is usually not a realistic assumption, then $P_{1C}(M \mid m,r)$ can be considered to be the probability of masking. Ones-count testing has the following additional attributes.

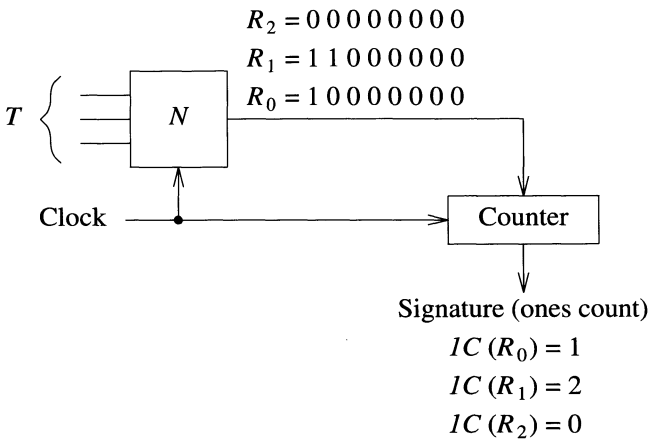


Figure 10.3 Testing via ones counting

1. Because of the bell-shaped form of the binomial coefficient $\binom{m}{r}$, the probability of masking is low when the signature lies near the extreme values of its range, but increases rapidly as it approaches the midpoint, i.e., $\lfloor m/2 \rfloor$.
2. When $1C(R_0) = 0$ or m, then no masking can occur (see Problem 10.1).
3. A fault that creates an odd number of errors in the output response is always detected; if the number of errors is even, it may be detected.
4. Inverting the output does not change the statistical properties of this test technique.
5. If the circuit under test is combinational, then the test vectors in T can be permuted without affecting the fault coverage. This permutation will not affect the signature or the masking characteristics.

Multiple-output circuits can be tested using a separate counter on each of the k outputs. To reduce area overhead, a parallel-to-serial conversion of the outputs can be used along with a single counter. Here each output is loaded into a register, and the test process stops while the k-bit response is shifted out of the register into the ones counter, which must have $\lceil \log_2(km+1) \rceil$ flip-flops.

Combinational Circuits

For the following result we assume that the CUT is tested by random test vectors and that all error patterns are equally likely to occur.

Theorem 10.1: The masking probability for ones-count compression for a combinational circuit asymptotically approaches $(\pi m)^{-1/2}$.

Proof: As shown previously, when $1C(R_0) = r$ the masking probability is

$$P_{1C}(M \mid m,r) = \frac{\binom{m}{r} - 1}{2^m - 1} \tag{10.1}$$

Consider the set S of all possible n-input functions. Assuming random test vectors, the probability that the random test vectors will produce r ones at the output F of these functions over this set S is

$$P(r) = \frac{\binom{m}{r}}{2^m} \tag{10.2}$$

The masking probability is therefore

$$P(M) = \sum_{r=0}^{m} P_{1C}(M \mid m,r)\, P(r) \tag{10.3}$$

Substituting (10.1) and (10.2) into (10.3) yields

$$P(M) = \frac{\binom{2m}{m} - 2^m}{2^m(2^m - 1)} \tag{10.4}$$

Using Stirling's formula $n! \simeq (2\pi n)^{1/2} e^{-n} n^n$ in (10.4) produces the result $P(M) \simeq (\pi m)^{-1/2}$ □

Next we show that for any combinational circuit C there exists a test $T'(1C)$ which detects all faults of interest, even when the results are compressed using ones counting. Let $T = \{T^0, T^1\}$ be a set of m test vectors that detect a set of faults F in C, and where for all $t \in T^0$, the fault-free response is 0, and for all $t \in T^1$, the fault-free response is 1. Let the test set $T'(1C)$ consist of one copy of every test pattern in T^0, and $\mid T^0 \mid + 1$ copies of every test pattern in T^1. In the worst case, $T'(1C)$ consists of the order of m^2 test vectors.

Theorem 10.2: When applying ones counting and the test set $T'(1C)$ to C, no error masking occurs for any fault in F.

Proof: See Problem 10.3.

Further details on ones counting can be found in [Barzilai *et al.* 1981], [Hayes 1976], [Losq 1978], and [Parker 1976].

10.3 Transition-Count Compression

In *transition-count* (TC) *testing*, the signature is the number of 0-to-1 and 1-to-0 transitions in the output data stream. Thus the transition count associated with a sequence $R = r_1, r_2, ..., r_m$ is

$$TC(R) = \sum_{i=1}^{m-1} (r_i \oplus r_{i+1})$$

where $\sum$ denotes ordinary arithmetic addition and $\oplus$ is modulo-2 addition. Since $0 \le TC(R) \le (m-1)$, the response-compression circuitry consists of a transition detector and a counter with $\lceil \log_2 m \rceil$ stages. Figure 10.4 illustrates this concept. Note that the

output of the transition detector in Figure 10.4(b) is a function of the initial state of the flip-flop.

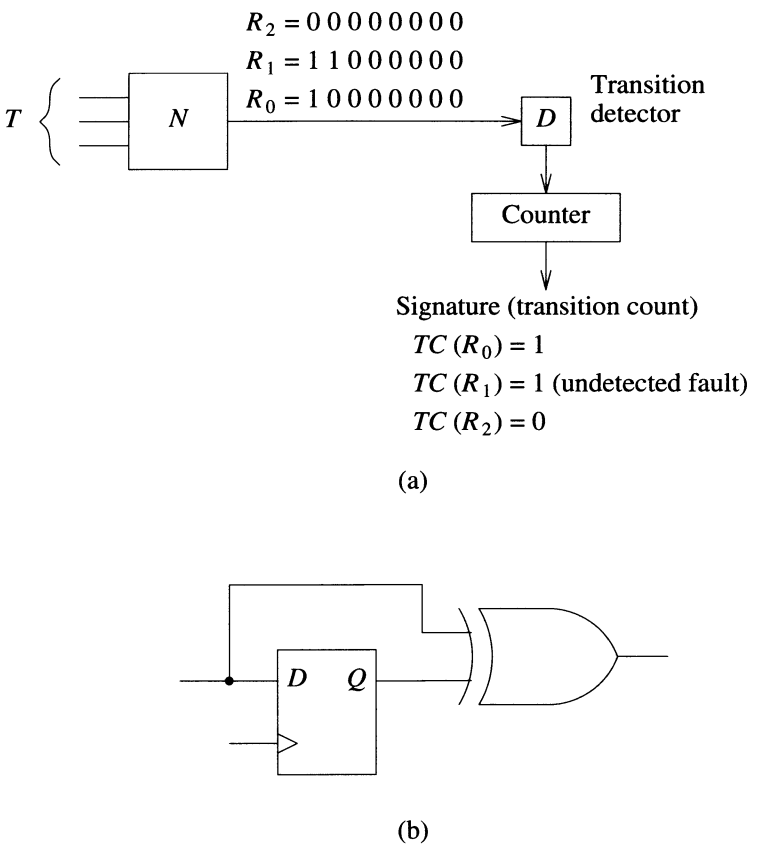


Figure 10.4 (a) Transition-count testing (b) A transition detector

Let T be a test sequence of length m for a circuit and R_0 be the fault-free response, where $TC(R_0) = r$. Let R' be an arbitrary binary sequence of length m. R' has $(m-1)$ boundaries between bits where a transition can occur. There are $\binom{m-1}{r}$ ways of assigning r transitions to $m-1$ boundaries so that R' will also have a transition count of r. Since the sequence $\overline{R'}$ obtained by complementing every bit of R' has the same transition count as R', there are $2\binom{m-1}{r}$ possible sequences that have transition count r, only one of which is the response of the fault-free circuit. Thus there are $2\binom{m-1}{r}-1$ possible error sequences that lead to aliasing. If all faulty sequences are equally likely to occur as the response of a faulty circuit, then the probability of masking is given by

$$P_{TC}(M \mid m,r) = \frac{2\binom{m-1}{r} - 1}{2^m - 1}.$$

This function has similar properties to that derived for the case of ones counting.

Unlike ones counting, transition counting is sensitive to the order of the bits in the response vector. Also transition counting does not guarantee detecting all single-bit errors. (See Figure 10.4). The single-bit error masking capability of a TC test is summarized in the following result.

Theorem 10.3: In an arbitrary m-bit sequence the probability of a single-bit error being masked is $(m-2)/2m$.

Proof: Note that for the sequence of bits r_{j-1}, r_j, r_{j+1}, where $r_{j-1} \neq r_{j+1}$, if r_j is in error, then this error will not be detected. Consider bit r_j. Selecting a value for r_{j-1} also determines the value of r_{j+1}. Thus there are 2^{m-1} sequences in which an error on bit r_j is TC-undetectable. Also there are $(m-2)$ choices for j, since r_j cannot be the first or last bit in R. Thus there are $(m-2)\,2^{m-1}$ TC-undetectable single-bit errors in an arbitrary sequence of length m. Since there are 2^m possible m-bit sequences and each can be associated with m single-bit errors, the probability of a TC-undetectable single-bit error occurring is

$$P_r = \frac{(m-2)\,2^{m-1}}{m\,2^m} = \frac{(m-2)}{2m}.$$

This value approaches 1/2 for large values of m. □

For multiple-output circuits, a transition detector is needed for each output pin. Each detector can drive a counter, or a single counter can be used to compress a weighted sum of each of the detectors.

Combinational Circuits

Again assume that the CUT is tested by random test vectors and that all error patterns are equally likely to occur.

Theorem 10.4: The masking probability for transition-count compression for a combinational circuit asymptotically approaches $(\pi m)^{-1/2}$. □

The proof is similar to that of Theorem 10.1, but now

$$P(M) = \frac{4\binom{2m-2}{m-1} - 2^m}{2^m\,(2^m-1)}.$$

We next focus on the problem of testing a combinational circuit using a deterministic test set where faults are detected by means of transition-count testing.

Let X be an input test sequence and $F = \{ f_1,\ldots,f_n \}$ be a set of faults in a combinational circuit C. Let R_0 be the response to X from C, and R_i be the response to X from C when fault f_i is present. For simplicity we assume that C is a single-output circuit. Then X is a transition-count test for C with respect to F if $TC(R_0) \neq TC(R_i)$ for $1 \leq i \leq n$. Here, every fault is said to be TC-detectable. Note that in general not all logic faults are TC-detectable. For example, consider a fault that causes the output function realized by C to be changed from $f(x_1,\ldots,x_n)$ to $\bar{f}(x_1,\ldots,x_n)$. For this case the response to X will be

$\overline{R_0}$; that is, each bit in the output sequence will be complemented. But $TC(R_0) = TC(\overline{R_0})$; hence this fault is TC-undetectable even though every response bit is in error (see Problem 10.11).

Faults f_i and f_j are said to be TC-*distinguishable* if there exists an input sequence X such that $TC(R_i) \neq TC(R_j)$. Transition-count testing is weaker than the conventional testing procedure in the following sense.

1. A fault in C is TC-detectable only if it is detectable; the converse is not true.
2. Two faults in C are TC-distinguishable only if they are distinguishable; the converse is also false.

We will restrict out attention to TC testing of single and multiple stuck-type faults. Note that in an irredundant single-output combinational circuit C no single fault can change the output function $f(x_1,...,x_n)$ to $\overline{f}(x_1,...,x_n)$. It is conjectured that this result also holds true for multiple faults. However, in a single-output redundant circuit it is possible for a multiple fault to complement the output.

The following theorem specifies a constructive procedure for generating a complete TC test for a combinational circuit C.

Theorem 10.5: Let T be a single-fault test set for an irredundant single-output combinational circuit C. Let $T^0(T^1)$ be all tests in T producing output 0(1). Construct a test sequence $X^* = t(1)t(2)...t(p)$ as follows:

1. X^* contains every element in T.
2. X^* is an alternating sequence of tests from T^0 and T^1. If $|T^0| \geq |T^1|$, let $t(1) \in T^0$, otherwise $t(1) \in T^1$. If $t(i) \in T^d$, then select $t(i+1) \in T^d$ for $1 \leq i \leq p-1$. The resulting sequence X^* is a single-fault TC test for C.

Proof: The response R^* to X^* is an alternating sequence of 0s and 1s. The only other sequence with this same transition count is $\overline{R}_0$. But X^* detects all stuck-at faults, and none of these faults can produce $\overline{R}_0$. Note that the length p of X^* is bounded by the inequalities $p \leq 2 \max\{|T^0|, |T^1|\} < 2(|T|-1)$. In addition, if T is a minimal-length test for C, and if the difference D between $|T^0|$ and $|T^1|$ is at most 1, then X^* is a minimal-length single-fault TC test for C. □

Thus, every single stuck fault in a single-output irredundant combinational circuit is TC-detectable using less than twice the number of tests required by conventional testing. For $D > 1$, the problem of finding a minimal-length TC test is unsolved. An extension to this analysis dealing with both single and multiple faults in two-level sum of product circuits is given in Problem 10.12.

Further material on transition count testing can be found in [Hayes 1976a, 1976b], [Reddy 1977], and [Savir and McAnney 1985].

10.4 Parity-Check Compression

Parity-check compression is carried out using a simple circuit such as that shown in Figure 10.5. As will be shown later, the compression circuit consisting of the D flip-flop and the XOR gate implements a linear feedback shift register whose primitive polynomial is $G(x) = x + 1$. If the initial state of the flip-flop is 0, the signature S is the parity of the

circuit response, namely it is 0 if the parity is even and 1 if the parity is odd. This scheme detects all single-bit errors and all multiple-bit errors consisting of an odd number of error bits in the response sequence. Faults that create an even number of errors are not detected. Assuming all faulty bit streams are equally likely, the probability of masking approaches 1/2 as m increases.

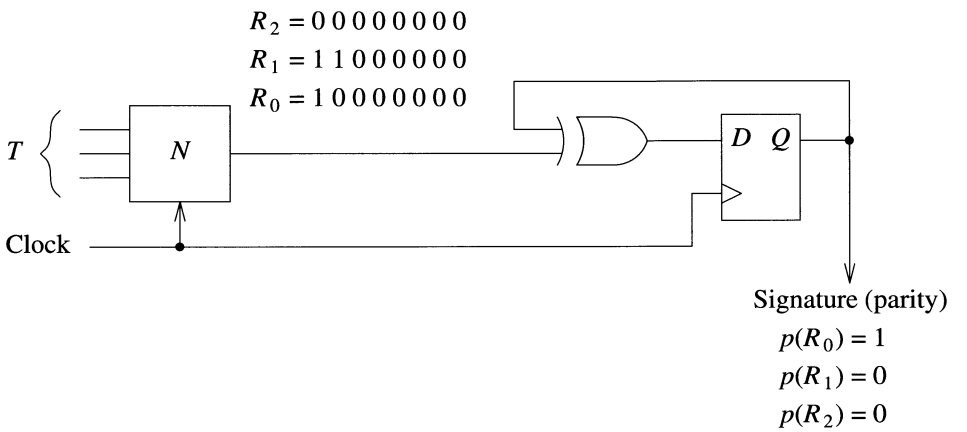


Figure 10.5 Parity-check compression

This technique can be extended to multiple-output circuits in several ways. One way is to replace the input exclusive-or gate by a multiple-input gate or exclusive-or network and have all outputs of the CUT drive this network. Unfortunately, an error internal to a circuit may affect more than one output line. If such an error propagates simultaneously to an even number of output lines, then it will have no effect on the signature.

A second method is to use a separate parity-check compressor in each output, though this requires additional hardware.

More information on parity-check compression can be found in [Tzidon *et al.* 1978] and [Carter 1982a, 1982b].

10.5 Syndrome Testing

Syndrome testing relies on exhaustive testing, i.e., on applying all 2^n test vectors to an n-input combinational circuit. First consider a single-output circuit implementing a function f. The *syndrome* S (or signature) is the normalized number of 1s in the resulting bit stream; i.e., $S = K/2^n$, where K is the number of minterms in the function f. Thus syndrome testing is a special case of ones counting. Clearly $0 \leq S \leq 1$. The syndrome of a 3-input AND gate is 1/8 and that of a 3-input OR gate is 7/8. Also the syndrome of f is a functional property of f independent of its implementation.

Syndrome testing is of interest because of the concept of *syndrome testability*; i.e., any function f can be realized in such a way that all single stuck-at faults are syndrome detectable.

Consider the circuit configuration shown in Figure 10.6. Assume circuits C_1 and C_2 have no inputs in common, and let $S(F) = S_1$, and $S(G) = S_2$. The input-output syndrome relation for this circuit as a function of the type of gate used to realize the circuit C is given in Figure 10.7.

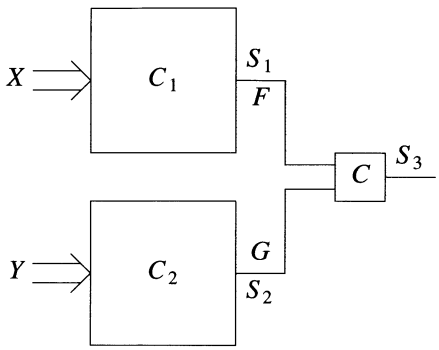


Figure 10.6 Circuit for calculating syndromes

Gate type for C	Syndrome S_3
OR	$S_1 + S_2 - S_1S_2$
AND	S_1S_2
NAND	$1 - S_1S_2$
NOR	$1 - (S_1+S_2-S_1S_2)$
XOR	$S_1 + S_2 - 2S_1S_2$

Figure 10.7 Syndromes for circuits having nonreconvergent fanout

Let C_i have n_i inputs and K_i minterms. For the case where C is an OR gate, the result in Figure 10.7 can be derived as follows. Note that $K = K_1 2^{n_2} + K_2 2^{n_1} - K_1K_2$. Clearly each minterm of C_1 can be associated with 2^{n_2} inputs to C_2, each of which produces an output of 1. The last term is required so that no minterms are counted twice. Then

$$S = \frac{K}{2^{n_1+n_2}} = \frac{K_1}{2^{n_1}} + \frac{K_2}{2^{n_2}} - \frac{K_1}{2^{n_1}}\frac{K_2}{2^{n_2}} = S_1 + S_2 - S_1S_2 .$$

The reader can verify the other entries in Figure 10.7 in a similar way. (See Problem 10.17.)

Assuming C_1 and C_2 share inputs, i.e., there is reconvergent fanout, the output syndrome can be expressed as shown in Figure 10.8.

Gate type for C	Syndrome S_3
OR	$S_1 + S_2 - S(FG)$
AND	$S_1 + S_2 + S(\bar{F}\,\bar{G}) - 1$
XOR	$S(F\bar{G}) + S(\bar{F}G)$

Figure 10.8 Syndromes for circuits having reconvergent fanout

Some basic results dealing with the testability of combinational circuits using syndrome testing are given below. Further details can be found in [Savir 1980], [Barzilai *et al.* 1981], and [Markowsky 1981].

A realization C of a function f is said to be *syndrome-testable* if no single stuck-at fault causes the circuit to have the same syndrome as the fault-free circuit.

A function $f\ (x_1,x_2,...,x_n)$ is said to be *unate* in x_i if there exists a sum of product expression for f where variable x_i appears only in uncomplemented (x_i) or complemented ($\bar{x}_i$) form. For example, $f = x_1\bar{x}_2x_3 + x_1\bar{x}_2x_4 + \bar{x}_4\bar{x}_5$ is unate in x_1, x_2, x_3, and x_5, but not in x_4.

Lemma 10.1: A two-level irredundant circuit that realizes a unate function in all its variables is syndrome-testable. □

There exist two-level irredundant circuits that are not syndrome-testable. For example, for the function $f = x\bar{z} + yz$, the faults z *s-a*-0 and z *s-a*-1 are not syndrome-testable. Note that z is not unate in f. By adding an extra input, this function can be converted into a function g, which is syndrome-testable. For example, $g = wx\bar{z} + yz$ is syndrome-testable, and by setting $w = 1$ during normal operation, g reduces to f. Using this technique leads to the following result:

Lemma 10.2: Every two-level irredundant combinational circuit can be made syndrome-testable by adding control inputs to the AND gates. □

Lemma 10.3: Every fanout-free irredundant combinational circuit composed of AND, OR, NAND, NOR, and NOT gates is syndrome-testable. □

Lemma 10.4: Let g be an internal line in an arbitrary combinational circuit. The output f can be expressed as $f = Ag + B\bar{g} + C$ where A, B, C, and g are functions of some or all of the variables of f. Then g *s-a*-0 is syndrome-untestable iff $S(A\bar{C}g) = S(B\bar{C}g)$, and g *s-a*-1 is syndrome-untestable iff $S(A\bar{C}\,\bar{g}) = S(B\bar{C}\,\bar{g})$. □

Note that if only one path exists from g to the output f, then f is syndrome-testable with respect to the faults g *s-a*-0 and g *s-a*-1.

10.6 Signature Analysis

Signature analysis is a compression technique based on the concept of cyclic redundancy checking (CRC) and realized in hardware using linear feedback shift registers (LFSRs). Before we discuss the details of signature analysis, some background on the theory and

operation of LFSRs is necessary. General information on this subject can be found in [Golomb 1982] and [Peterson and Weldon 1972].

10.6.1 Theory and Operation of Linear Feedback Shift Registers

In this section we present some of the formal properties associated with linear feedback shift registers. These devices, as well as modified versions of LFSRs, are used extensively in two capacities in DFT and BIST designs. One application is as a source of pseudorandom binary test sequences; the other is as a means to carry out response compression — known as *signature analysis.*

Consider the feedback shift registers shown in Figure 10.9. These circuits are all autonomous; i.e., they have no inputs except for clocks. Each cell is assumed to be a clocked D flip-flop. It is well known that such circuits are cyclic in the sense that when clocked repeatedly, they go through a fixed sequence of states. For example, a binary counter consisting of n flip-flops would go through the states 0, 1, ..., 2^{n-1}, 0, 1, ... The maximum number of states for such a device is 2^n. The shift register shown in Figure 10.9(a) cycles through only two states. If the initial state were 00 or 11, it would never change state. An n-bit shift register cycles through at most n states. Notice that the output sequence generated by such a device is also cyclic. The circuit of Figure 10.9(b) starting in the initial state 111 (or 000) produces a cyclic sequence of states of length 1. In Figure 10.9(c) we show the sequence generated for the circuit of Figure 10.9(b) if the initial state is 011. (The reader should analyze the case if the initial state is 101.)

In Figure 10.9(d) we illustrate the case where the state sequence generated by the feedback shift register is of length $2^3 - 1$. Note that for the class of circuits being illustrated, the all-0 state leads to a state sequence of length 1, namely the all-0 state itself. The circuit of Figure 10.9(d) is said to be a maximal-length shift register, since it generated a cyclic state sequence of length $2^n - 1$, as long as its initial state is not all-0s. Note also that if one of these circuits generates a cyclic state sequence of length k, then the output sequence also repeats itself every k clock cycles.

A *linear circuit* is a logic network constructed from the following basic components:

- unit delays or D flip-flops;
- modulo-2 adders;
- modulo-2 scalar multipliers.

In the analysis of such circuits, all operations are done modulo 2. The truth table for modulo-2 addition and subtraction is shown below.

$\pm$	0	1
0	0	1
1	1	0

Thus $x + x = -x - x = x - x = 0$.

Such a circuit is considered to be linear since it preserves the principle of superposition; i.e., its response to a linear combination of stimuli is the linear combination of the responses of the circuit to the individual stimuli.

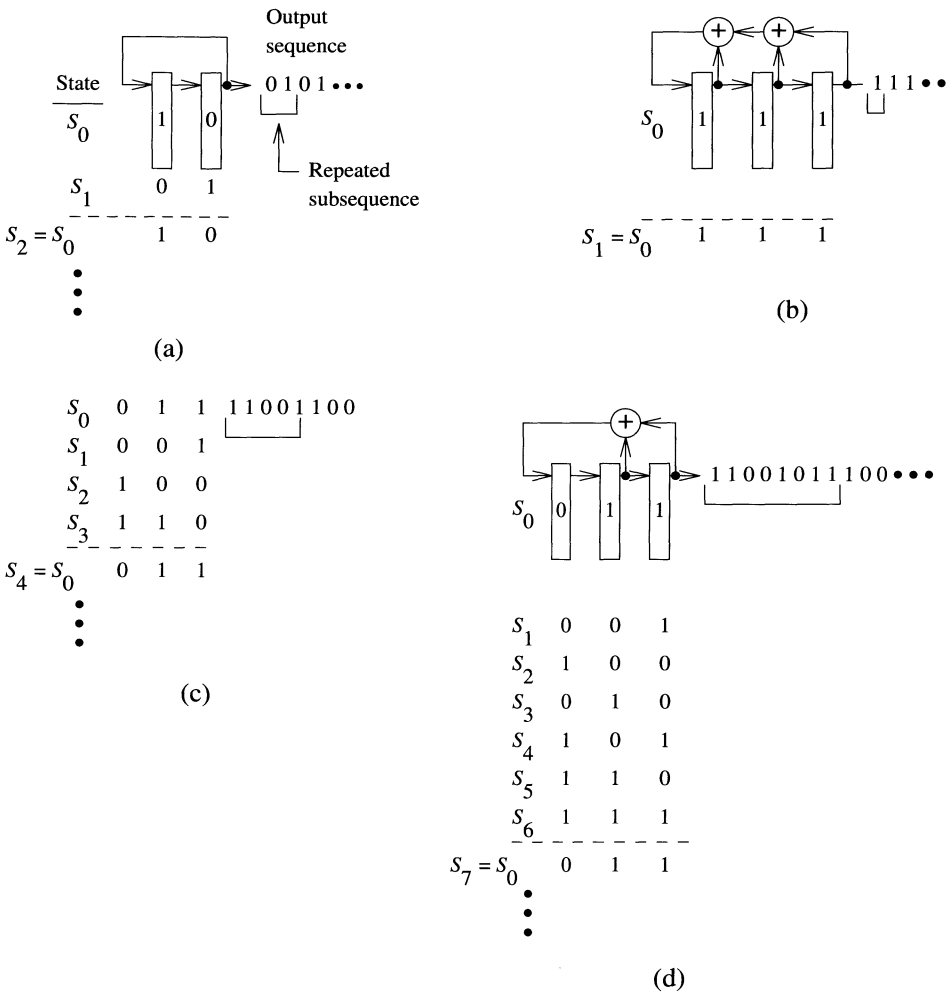


Figure 10.9 Feedback shift registers

In this section we will deal primarily with a class of linear circuits, known as *autonomous linear feedback shift registers*, that have the canonical form shown in Figures 10.10 and 10.11. Here c_i is a binary constant, and $c_i = 1$ implies that a connection exists, while $c_i = 0$ implies that no connection exists. When $c_i = 0$ the corresponding XOR gate can be replaced by a direct connection from its input to its output.

Characteristic Polynomials

A sequence of numbers $a_0, a_1, a_2, \ldots, a_m, \ldots$ can be associated with a polynomial, called a *generating function G(x)*, by the rule

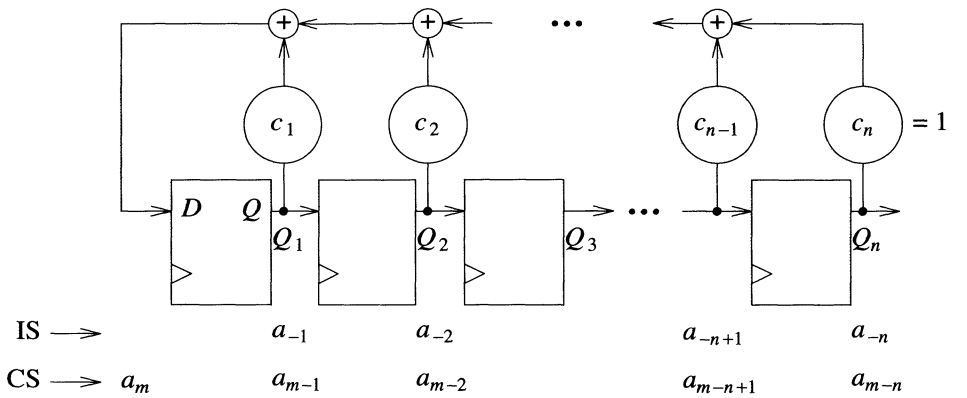


Figure 10.10 Type 1 (external-XOR) LFSR

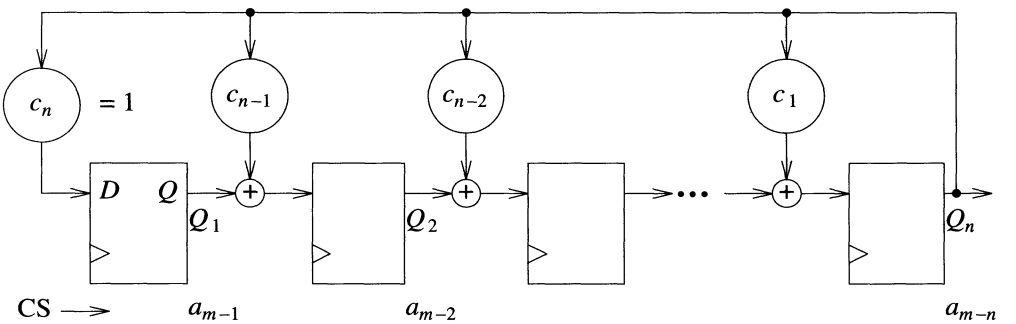


Figure 10.11 Type 2 (internal-XOR) LFSR

$$G(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m \ldots .$$

Let $\{ a_m \} = a_0, a_1, a_2, \ldots$ represent the output sequence generated by an LFSR, where $a_i = 0$ or 1. Then this sequence can be expressed as

$$G(x) = \sum_{m=0}^{\infty} a_m x^m \tag{10.5}$$

Recall that polynomials can be multiplied and divided (modulo 2). An example is shown below.

Example 10.1:

(a)

$$\begin{array}{rr} & x^2 + x + 1 \\ \times & x^2 + 1 \\ \hline & x^2 + x + 1 \\ (+) \quad x^4 + x^3 + x^2 & \\ \hline x^4 + x^3 \quad + x + 1 & \end{array}$$

since $x^2 + x^2 = 0$.

(b)

$$\begin{array}{r|l} & x^2 + x + 1 \\ \hline x^2 + 1 & x^4 + x^3 \quad + x + 1 \\ (-) & x^4 \quad + x^2 \\ \hline & x^3 + x^2 + x + 1 \quad \text{(note } x^2 = -x^2\text{)} \\ (-) & x^3 \quad + x \\ \hline & x^2 \quad + 1 \\ (-) & x^2 \quad + 1 \\ \hline & 0 \end{array}$$

There is no remainder; hence $x^2 + 1$ is said to *divide* $x^4 + x^3 + x + 1$. □

From the structure of the type 1 LFSR it is seen that if the current state (CS) of Q_i is a_{m-i}, for $i = 1, 2, \ldots, n$, then

$$a_m = \sum_{i=1}^{n} c_i \, a_{m-i} \tag{10.6}$$

Thus the operation of the circuit can be defined by a recurrence relation. Let the initial state (IS) of the LFSR be $a_{-1}, a_{-2}, \ldots, a_{-n+1}, a_{-n}$. The operation of the circuit starts n clock periods before generating the output a_0. Since

$$G(x) = \sum_{m=0}^{\infty} a_m \, x^m$$

substituting for a_m we get

$$G(x) = \sum_{m=0}^{\infty} \sum_{i=1}^{n} c_i a_{m-i} x^m = \sum_{i=1}^{n} c_i x^i \sum_{m=0}^{\infty} a_{m-i} x^{m-i}$$

$$= \sum_{i=1}^{n} c_i x^i [a_{-i} x^{-i} + \ldots + a_{-1} x^{-1} + \sum_{m=0}^{\infty} a_m x^m]$$

$$= \sum_{i=1}^{n} c_i x^i [a_{-i} x^{-i} + \ldots + a_{-1} x^{-1} + G(x)] .$$

Hence

$$G(x) = \sum_{i=1}^{n} c_i x^i G(x) + \sum_{i=1}^{n} c_i x^i (a_{-i} x^{-i} + \ldots + a_{-1} x^{-1})$$

or

$$G(x) = \frac{\sum_{i=1}^{n} c_i x^i (a_{-i} x^{-i} + \ldots + a_{-1} x^{-1})}{1 + \sum_{i=1}^{n} c_i x^i} \tag{10.7}$$

Thus $G(x)$ is a function of the initial state $a_{-1}, a_{-2}, \ldots, a_{-n}$ of the LFSR and the feedback coefficients $c_1, c_2, \ldots, c_n$. The denominator in (10.7), denoted by

$$P(x) = 1 + c_1 x + c_2 x^2 + \ldots + c_n x^n$$

is referred to as the *characteristic polynomial* of the sequence $\{a_m\}$ and of the LFSR. For an n-stage LFSR, $c_n = 1$. Note that $P(x)$ is only a function of the feedback coefficients. If we set $a_{-1} = a_{-2} = \ldots = a_{1-n} = 0$, and $a_{-n} = 1$, then (10.7) reduces to $G(x) = 1/P(x)$.

Thus the characteristic polynomial along with the initial state characterizes the cyclic nature of an LFSR and hence characterizes the output sequence. For $a_{-1} = a_{-2} = \ldots = a_{1-n} = 0$, and $a_{-n} = 1$, then

$$G(x) = \frac{1}{P(x)} = \sum_{m=0}^{\infty} a_m x^m \tag{10.8}$$

and since the sequence $\{ a_m \}$ is cyclic with period p, (10.8) can be rewritten as

$$\begin{aligned} 1/P(x) &= (a_0 + a_1 x + \ldots + a_{p-1} x^{p-1}) \\ &+ x^p (a_0 + a_1 x + \ldots + a_{p-1} x^{p-1}) \\ &+ x^{2p} (a_0 + a_1 x + \ldots + a_{p-1} x^{p-1}) + \ldots \\ &= (a_0 + a_1 x + \ldots + a_{p-1} x^{p-1}) (1 + x^p + x^{2p} + \ldots) \\ &= \frac{a_0 + a_1 x + \ldots + a_{p-1} x^{p-1}}{1 - x^p}. \end{aligned}$$

Thus it is seen that $P(x)$ evenly divides into $1 - x^p$.

The analysis of the type 1 LFSR can also proceed as follows.

Recall that

$$a_m(t) = \sum_{i=1}^{n} c_i a_{m-i}(t)$$

and note that $a_i(t) = a_{i+1}(t-1)$ Let x be a "shift" operator such that

$$x^k a_i(t) = a_i(t-k) \tag{10.9}$$

Then

$$a_m(t) = \sum_{i=1}^{n} c_i a_{m-i}(t) = \sum_{i=1}^{n} c_i x^i a_m(t) .$$

Note, for example, that $x a_m(t) = a_m(t-1) = a_{m-1}(t)$.

Thus

$$a_m + c_1 x a_m + c_2 x^2 a_m + \ldots + c_n x^n a_m = 0$$

or equivalently

$$[1 + c_1 x + c_2 x^2 + \ldots + c_n x^n] a_m = 0.$$

The term in brackets is again the characteristic polynomial associated with the LFSR.

Given a characteristic polynomial, it is easy to implement a type 1 LFSR to realize it. Conversely, given a type 1 LFSR, it is a simple process to determine its corresponding characteristic polynomial. For the circuit of Figure 10.9(b), $P(x) = 1 + x + x^2 + x^3$, and for Figure 10.9(d), $P(x) = 1 + x^2 + x^3$.

Referring to (10.9), let $y^{-k} a_i(t) = a_i(t-k)$. Then, carrying out the same algebraic manipulation as before, we obtain

$$[1 + c_1 y^{-1} + c_2 y^{-2} + \ldots + c_n y^{-n}] a_m = 0$$

or equivalently

$$[y^n + c_1 y^{n-1} + c_2 y^{n-2} + \ldots + c_n] a_m y^{-n} = 0 .$$

Again the term in the brackets can be considered to be a characteristic polynomial of the LFSR. Replacing y by x we obtain

$$P^*(x) = c_n + c_{n-1} x + c_{n-2} x^2 + \ldots + c_1 x^{n-1} + x^n .$$

$P^*(x)$ is said to be the *reciprocal polynomial* of $P(x)$, since $P^*(x) = x^n P(1/x)$.

Thus every LFSR can be associated with two characteristic polynomials. Referring to Figures 10.9(c) and 10.9(d), if $P(x) = 1 + x + x^2 + x^3$, then $P^*(x) = P(x)$, and if $P(x) = 1 + x^2 + x^3$, then $P^*(x) = 1 + x + x^3$.

If in Figure 10.10 one associates x^i with Q_i, then $P(x)$ can be read off directly from the figure. If, however, one associates x^i with Q_{n-i} and labels the input to the first flip-flop Q_0, then $P^*(x)$ can be read off directly from the figure. Figure 10.12 illustrates these two labelings.

Finally, note that a given characteristic polynomial, say $Q(x)$, can be realized by two different LFSRs depending on the labeling used.

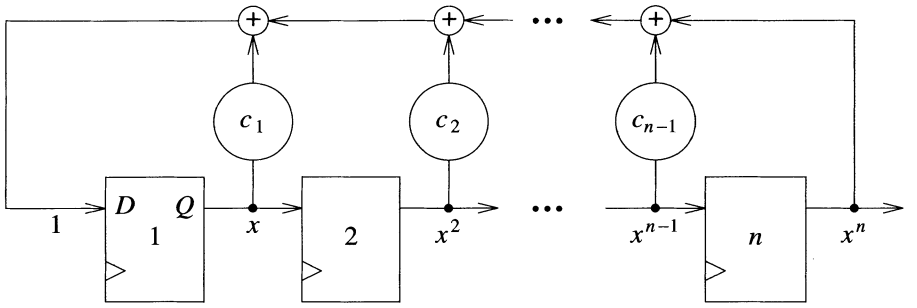
Referring to Figure 10.11, we see that for $i = 2, 3, \ldots, n$

$$a_{m-i}(t+1) = a_{m-i+1}(t) + c_{n-i+1} a_{m-n}(t) \tag{10.10}$$

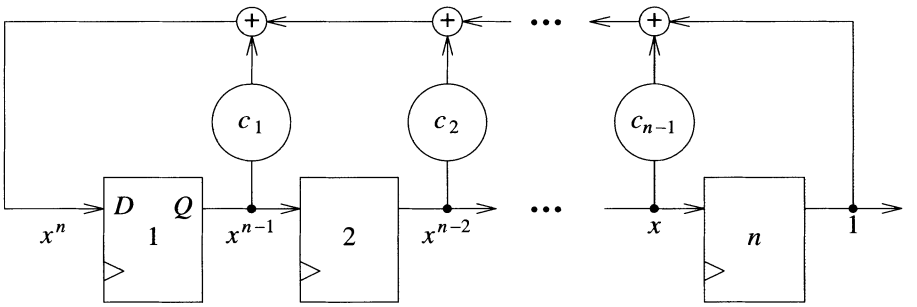
If we define $a_m(t) = 0$, then (10.10) is also true for $i = 1$. Let x be a "shift" operator such that $x^k a_i(t) = a_i(t-k)$. Then the equations (10.10) can be written as

$$x^{-1} a_{m-i}(t) = a_{m-i+1}(t) + c_{n-i+1} a_{m-n}(t) \tag{10.11}$$

for $i = 1, 2, \ldots, n$. Multiplying the i-th equation by x^{-i+1} we get



(a)



(b)

Figure 10.12 Reciprocal characteristic polynomials
(a) $P(x) = 1 + c_1\, x + c_2\, x_2 + ... + \ c_{n-1}x^{n-1} + x^n$
(b) $P^*(x) = 1 + c_{n-1}\, x + c_{n-2}\, x^2 + ... + c_1\, x^{n-1} + x^n$

$$x^{-1}a_{m-1} = c_n a_{m-n}$$

$$x^{-2}a_{m-2} = x^{-1}a_{m-1} + c_{n-1}x^{-1}a_{m-n}$$

$$x^{-3}a_{m-3} = x^{-2}a_{m-2} + c_{n-2}x^{-2}a_{m-n}$$

$$\vdots$$

$$x^{-n}a_{m-n} = x^{-n+1}a_{m-n+1} + c_1 x^{-n+1}a_{m-n}.$$

Summing these n equations and canceling terms that appear on both sides of the equation yield

$$x^{-n}a_{m-n} = [c_1 x^{-n+1} + ... + c_{n-2}x^{-2} + c_{n-1}x^{-1} + c_n]a_{m-n}$$

or

$$[x^{-n} + c_1 x^{-n+1} + ... + c_{n-1}x^{-1} + c_n]a_{m-n} = 0 .$$

Multiplying by x^n we get

$$[1 + c_1 x + ... + c_{n-1}x^{n-1} + c_n x^n]a_{m-n} = 0 .$$

The term in the brackets is the characteristic polynomial for the type 2 LFSR.

Again it can be shown that a characteristic polynomial *P(x)* has two type 2 realizations, or conversely, a type 2 LFSR can be associated with two characteristic polynomials that are reciprocals of each other.

Periodicity of LFSRs

We have seen that an LFSR goes through a cyclic or periodic sequence of states and that the output produced is also periodic. The maximum length of this period is $2^n - 1$, where n is the number of stages. In this section we consider properties related to the period of an LFSR. Most results will be presented without proof. Details can be found in [Bardell *et al.* 1987], [Golomb 1982], and [Peterson and Weldon 1972].

Theorem 10.6: If the initial state of an LFSR is $a_{-1} = a_{-2} = ... = a_{1-n} = 0$, $a_{-n} = 1$, then the LFSR sequence $\{a_m\}$ is periodic with a period that is the smallest integer k for which $P(x)$ divides $(1-x^k)$. □

Definition: If the sequence generated by an n-stage LFSR has period $2^n - 1$, then it is called a *maximum-length sequence.*

Definition: The characteristic polynomial associated with a maximum-length sequence is called a *primitive polynomial.*

Definition: An *irreducible polynomial* is one that cannot be factored; i.e., it is not divisible by any other polynomial other than 1 and itself.

Theorem 10.7: An *irreducible polynomial* $P(x)$ of degree n satisfies the following two conditions:

1. For $n \geq 2$, $P(x)$ has an odd number of terms including the 1 term.
2. For $n \geq 4$, $P(x)$ must divide (evenly) into $1 + x^k$, where $k = 2^n - 1$. □

The next result follows from Theorems 10.6 and 10.7.

Theorem 10.8: An irreducible polynomial is primitive if the smallest positive integer k that allows the polynomial to divide evenly into $1 + x^k$ occurs for $k = 2^n - 1$, where n is the degree of the polynomial. □

The number of primitive polynomials for an n-stage LFSR is given by the formula

$$\lambda_2(n) = \Phi(2^n - 1)/n$$

where

$$\Phi(n) = n \prod_{p \mid n} (1 - 1/p)$$

and p is taken over all primes that divide n. Figure 10.13 shows some values of $\lambda_2(n)$.

n	$\lambda_2(n)$
1	1
2	1
4	2
8	16
16	2048
32	67108864

Figure 10.13 Number of primitive polynomials of degree n

Figure 10.14 gives one primitive polynomial for every value of n between 1 and 36. A shorthand notation is employed. For example, the polynomial $x^{12} + x^7 + x^4 + x^3 + 1$ is represented by the entry 12:7 4 3 0, listing the exponents of those x^i associated with a "1" coefficient.

1:	0				13:	4	3	1	0	25:	3	0		
2:	1	0			14:	12	11	1	0	26:	8	7	1	0
3:	1	0			15:	1	0			27:	8	7	1	0
4:	1	0			16:	5	3	2	0	28:	3	0		
5:	2	0			17:	3	0			29:	2	0		
6:	1	0			18:	7	0			30:	16	15	1	0
7:	1	0			19:	6	5	1	0	31:	3	0		
8:	6	5	1	0	20:	3	0			32:	28	27	1	0
9:	4	0			21:	2	0			33:	13	0		
10:	3	0			22:	1	0			34:	15	14	1	0
11:	2	0			23:	5	0			35:	2	0		
12:	7	4	3	0	24:	4	3	1	0	36:	11	0		

Figure 10.14 Exponents of terms of primitive polynomials

Characteristics of Maximum-Length Sequences

Sequences generated by LFSRs that are associated with a primitive polynomial are called *pseudorandom sequences*, since they have many properties like those of random sequences. However, since they are periodic and deterministic, they are pseudorandom, not random. Some of these properties are listed next.

In the following, any string of $2^n - 1$ consecutive outputs is referred to as an *m-sequence*.

Property 1. The number of 1s in an m-sequence differs from the number of 0s by one.

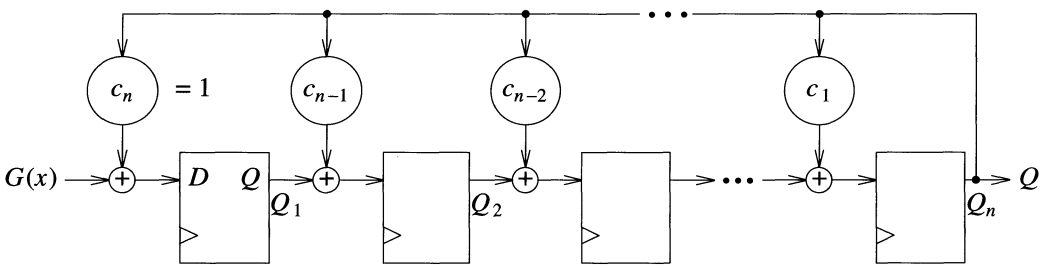
Property 2. An m-sequence produces an equal number of runs of 1s and 0s.

Property 3. In every m-sequence, one half the runs have length 1, one fourth have length 2, one eighth have length 3, and so forth, as long as the fractions result in integral numbers of runs.

These properties of randomness make feasible the use of LFSRs as test sequence generators in BIST circuitry.

10.6.2 LFSRs Used as Signature Analyzers

Signature analysis is a compression technique based on the concept of *cyclic redundancy checking* (CRC) [Peterson and Weldon 1972]. In the simplest form of this scheme, the signature generator consists of a single-input LFSR. The signature is the contents of this register after the last input bit has been sampled. Figure 10.15 illustrates this concept.



$$P(x) = 1 + \sum_{i=1}^{n} c_i x^i \qquad \text{Initial state: } I(x) = 0; \text{ Final state: } R(x)$$

$$P^*(x) = 1 + \sum_{i=1}^{n} c_i x^{n-i}$$

$$\frac{G(x)}{P^*(x)} = Q(x) + \frac{R(x)}{P^*(x)}$$

or

$$G(x) = Q(x)\,P^*(x) + R(x)$$

Figure 10.15 A type 2 LFSR used as a signature analyzer

The proportion of error streams that mask to the correct signature $S(R_0)$ is independent of the actual signature. For a test bit stream of length m, there are 2^m possible response streams, one of which is correct. It will be shown later that the structure of an LFSR distributes all possible input bit streams evenly over all possible signatures; i.e., the number of bit streams that produce a specific signature is

$$\frac{2^m}{2^n} = 2^{m-n} \tag{10.12}$$

where the LFSR consists of n stages, and the all-0 state is now possible because of the existence of an external input. For a particular fault-free response, there are $2^{m-n} - 1$ erroneous bit streams that will produce the same signature. Since there are a total of $2^m - 1$ possible erroneous response streams, the proportion of masking error streams is

$$P_{SA}(M \mid m,n) = \frac{2^{m-n} - 1}{2^m - 1} \simeq 2^{-n} \tag{10.13}$$

where the approximation holds for $m \gg n$.

If all possible error streams are equally likely, which is rarely the case, then $P_{SA}(M \mid m,n)$ is the probability that an incorrect response will go undetected; i.e., the probability of no masking is $1 - 2^{-n}$. This is a somewhat strange result since it is only a function of the length of the LFSR and not of the feedback network. Increasing the register length by one stage reduces the masking probability by a factor of 2. Note that because of the feedback network, all single-bit errors are detectable. However, there is no direct correlation between faults and error masking. Thus a 16-bit signature analyzer may detect $100(1-2^{-16}) = 99.9984$ percent of the erroneous responses but not necessarily this same percentage of faults.

Signature analysis is the most popular method employed for test data compression because it usually produces the smallest degree of masking. This results from the signature being sensitive to the number of 1s in the data stream as well as to their positions. Because of the widespread use of signature analyzers in both built-in test circuitry as well as in ATE equipment for PCB, the rest of this chapter will deal primarily with both practical and theoretical aspects of their design.

Shift Register Polynomial Division

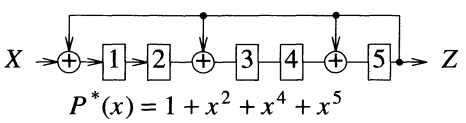
The theory behind the use of an LFSR for signature analysis is based on the concept of polynomial division, where the "remainder" left in the register after completion of the test process corresponds to the final signature.

Consider the type 2 (internal-XOR) LFSR shown in Figure 10.15. The input sequence $\{a_m\}$ can be represented by the polynomial $G(x)$ and the output sequence by $Q(x)$. The highest degree of the polynomials $G(x)$ and $Q(x)$ correspond, respectively, to the first input bit to enter the LFSR and the first output bit produced n clock periods later, where n is the degree of the LFSR. If the initial state of the LFSR is all 0s, let the final state of the LFSR be represented by the polynomial $R(x)$. Then it can be shown that these polynomials are related by the equation

$$\frac{G(x)}{P^*(x)} = Q(x) + \frac{R(x)}{P^*(x)}$$

where $P^*(x)$ is the reciprocal characteristic polynomial of the LFSR. The reciprocal characteristic polynomial is used because a_m corresponds to the first bit of the input stream rather than the last bit. Hence an LFSR carries out (polynomial) division on the input stream by the characteristic polynomial, producing an output stream corresponding to the quotient $Q(x)$ and a remainder $R(x)$. We illustrate this in the next example.

Example 10.2: Figure 10.16(a) shows a single-input signature analyzer where $P^*(x) = 1 + x^2 + x^4 + x^5$. Let the input sequence be 1 1 1 1 0 1 0 1 where the data are entered in the order shown. The first bit corresponds to a_m in the sequence $a_m a_{m-1} \ldots a_2 a_1$. Figure 10.16(c) shows a simulation of the processing of this input by the LFSR. The first five bits of the output response are ignored since they are independent of the input sequence. It is seen that $R(x) = x^4 + x^2$ and $Q(x) = x^2 + 1$.



(a)

Input sequence: 1 1 1 1 0 1 0 1 (8 bits)

$G(x) = x^7 + x^6 + x^5 + x^4 + x^2 + 1$

(b)

Time	Input stream	Register contents	Output stream
		1 2 3 4 5	
0	1 0 1 0 1 1 1 1	0 0 0 0 0 ←	Initial state
1	1 0 1 0 1 1 1 1	1 0 0 0 0	
•	•	•	
•	•	•	
•	•	•	
5	1 0 1	0 1 1 1 1	
6	1 0	0 0 0 1 0	1
7	1	0 0 0 0 1	0 1
8	Remainder →	0 0 1 0 1	1 0 1
		Remainder	Quotient
		$R(x) = x^2 + x^4$	$1 + x^2$

(c)

Figure 10.16 Polynomial division

To check this result we have

$$\begin{array}{rl} P^*(x): & x^5 + x^4 + x^2 + 1 \\ \times Q(x): & x^2 + 1 \\ \hline & x^7 + x^6 + x^4 + x^2 + x^5 + x^4 + x^2 + 1 \\ = & x^7 + x^6 + x^5 + 1 \end{array}$$

Thus

$$P^*(x)Q(x) + R(x) = x^7 + x^6 + x^5 + x^4 + x^2 + 1 = G(x)\,.$$

□

Type 1 (external-XOR) LFSRs also carry out polynomial division and produce the correct quotient. However, the contents of the LFSR is not the remainder as is the case for type 2 LFSRs. But it can be shown that all input sequences which are equal to each other modulo $P(x)$ produce the same remainder.

Error Polynomials and Masking

Let $E(x)$ be an *error polynomial*; i.e., each non-0 coefficient represents an error occurring in the corresponding bit position. As an example, let the correct response be $R_0 = 10111$ and the erroneous response be $R' = 11101$. Then the difference, or error polynomial, is 01010. Thus $G_0(x) = x^4 + x^2 + x + 1$, $G'(x) = x^4 + x^3 + x^2 + 1$, and $E(x) = x^3 + x$. Clearly $G'(x) = G(x) + E(x)$ (modulo 2). Since $G(x) = Q(x)P^*(x) + R(x)$, an undetectable response sequence is one that satisfies the equation $G'(x) = G(x) + E(x) = Q'(x)P^*(x) + R(x)$; i.e., $G'(x)$ and $G(x)$ produce the same remainder. From this observation we obtain the following well-known result from algebraic coding theory.

Theorem 10.9: Let $R(x)$ be the signature generated for an input $G(x)$ using the characteristic polynomial $P(x)$ as a divisor in an LFSR. For an error polynomial $E(x)$, $G(x)$ and $G'(x) = G(x) + E(x)$ have the same signature $R(x)$ if and only if $E(x)$ is a multiple of $P(x)$. □

Thus both type 1 and type 2 LFSRs can be used to generate a signature $R(x)$. Henceforth, the final contents of the LFSR will be referred to as the *signature*, because sometimes, depending on the initial state and polynomial $P(x)$, the final state does not correspond to the remainder of $G(x)/P(x)$.

Theorem 10.10: For an input data stream of length m, if all possible error patterns are equally likely, then the probability that an n-bit signature generator will not detect an error is

$$P(M) = \frac{2^{m-n} - 1}{2^m - 1}$$

which, for $m \gg n$, approaches 2^{-n}. □

This result follows directly from the previous theorem because $P(x)$ has $2^{m-n} - 1$ non-0 multiples of degree less than m. It also corresponds to the same result given earlier but based on a different argument. Note that this result is independent of the polynomial $P(x)$. This includes $P(x) = x^n$, which has no feedback, i.e., is just a shift register. For this case, the signature is just the last n bits of the data stream. In fact one can use the first n bits and truncate the rest of the test sequence and obtain the same results. These strange conclusions follow from the assumption that all error patterns are equally likely. If this were the case, long test sequences would certainly not be necessary.

To see why this assumption is flawed, consider a minimal-length test sequence of length m for a combinational circuit. Clearly the i-th test vector t_i detects some fault f_i not detected by t_j, $j = 1, 2, ..., i - 1$. Thus if f_i is present in the circuit, the error pattern is of the form 00...01xx...; i.e., the first i–1 bits must be 0. Several other arguments can be made to show that all error patterns are not equally likely.

Theorem 10.11: An LFSR signature analyzer based on any polynomial with two or more non-0 coefficients detects all single-bit errors.

Proof: Assume $P(x)$ has two or more non-0 coefficients. Then all non-0 multiples of $P(x)$ must have at least two non-0 coefficients. Hence an error pattern with only one non-0 coefficient cannot be a multiple of $P(x)$ and must be detectable. □

As an example, $P(x) = x + 1$ has two non-0 coefficients and thus detects all single errors. (See Figure 10.5.)

Definition: A (k,k) *burst error* is one where all erroneous bits are within k consecutive bit positions, and at most k bits are in error.

Theorem 10.12: If $P(x)$ is of degree n and the coefficient of x^0 is 1, then all (k,k) burst errors are detected as long as $n \geq k$. □

Rather than assuming that all error patterns are equally likely, one can assume that the probability that a response bit is in error is p. Then for $p = 0.5$, the probability of masking is again 2^{-n}. For very small or very large values of p, the probability of masking approaches the value

$$2^{-n} + (2^n - 1)(1-2p)^{m(1-1/(2^n-1))}$$

where m is the length of the test sequence [Williams *et al.* 1987].

Experimental results also show that using primitive polynomials helps in reducing masking.

In conclusion, the bound of 2^{-n} on error masking is not too useful since it is based on unrealistic assumptions. However, in general, signature analysis gives excellent results. Results are sensitive to $P(x)$ and improve as n increases. Open problems deal with selecting the best $P(x)$ to use, characterizing error patterns, correlating fault coverage to $P(x)$, and determining the probability of masking.

Further results on signature analysis can be found in [Frohwerk 1977], [Chan 1977], [Smith 1980], [Bhaskar 1982], [Sridhar *et al.* 1982], and [Carter 1982]. McAnney and Savir [1988] describe a procedure for determining the initial state of a signature register so that the final signature in a fault-free circuit is a given constant, say all zeros. Sometimes this can simplify the checking of the final signature, such as when more than one test sequence is used.

10.6.3 Multiple-Input Signature Registers

Signature analysis can be extended to testing multiple-output circuits. Normally a single-input signature analyzer is not attached to every output because of the resulting high overhead. A single signature analyzer could be time-multiplexed, but that would require repeating the test sequence for each output, resulting in a potentially long test time. The most common technique is to use a multiple-input signature register (MISR), such as the one shown in Figure 10.17 [Hassan *et al.* 1983, David 1984]. Here we assume that the CUT has n (or less) outputs. It is seen that this circuit operates as n single-input signature analyzers. For example, by setting $D_i = 0$ for all $i \neq j$, the circuit computes the signature of the data entering on line D_j. The mathematical theory associated with MISRs will not be presented here, but follows as a direct extension of the results presented previously [Bardell *et al.* 1987]. One can again associate an error pattern with each input D_i. These error patterns are merged within the LFSR. Again, assuming all error patterns are equally likely, the probability that a MISR will not detect an error is approximately 2^{-n}.

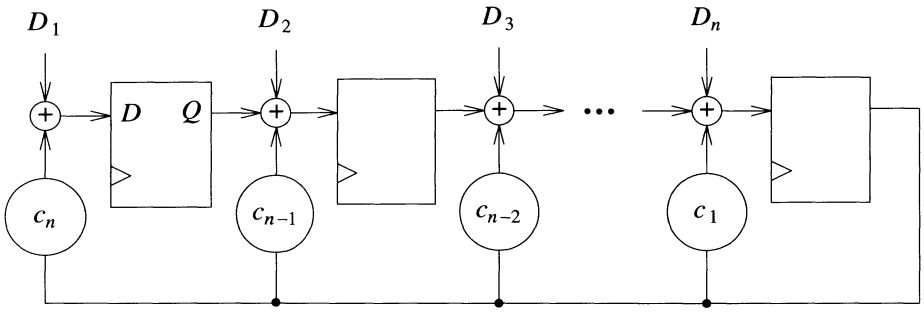


Figure 10.17 Multiple-input signature register

Selection of the Polynomial P(x)

As stated previously, a MISR having n stages has a masking probability approximately equal to 2^{-n} for equally likely error patterns and long data streams. Also, this result is independent of $P(x)$. Let the error bit associated with D_i at time j be denoted by e_{ij}, where $i = 1,2,...,n$, and $j = 1,2,...,m$. Then the error polynomial associated with D_i is $E_i = \sum_{j=1}^{m} e_{ij}x^{j-1}$. Then the *effective error polynomial* is $E(x) = \sum_{i=1}^{n} E_i x^{i-1}$ assuming that the initial state of the register is all zeros. The error polynomial $E(x)$ is masked if it is a multiple of $P(x)$. So a complex-feedback LFSR structure is typically used on the assumption that it will reduce the chances of masking an error. For example, the Hewlett-Packard 5004A signature analyzer employs the characteristic polynomial $P(x) = x^{16} + x^9 + x^7 + x^4 + 1$ [Frohwerk 1977]. $P(x)$ is often selected to be a primitive polynomial.

When the characteristic polynomial is the product of the parity generator polynomial $g(x) = x + 1$ and a primitive polynomial of degree $(n-1)$, an n-stage MISR has the property that the parity over all the bits in the input streams equals the parity of the final signature. Hence masking will not occur for an odd number of errors.

Increasing the Effectiveness of Signature Analysis

There are several ways to decrease the probability of masking. Based on the theory presented, the probability of masking can be reduced by increasing the length of the LFSR. Also a test can be repeated using a different feedback polynomial. When testing combinational circuits, a test can be repeated after first changing the order of the test vectors, thus producing a different error polynomial. This technique can also be used for sequential circuits, but now the fault-free signature also changes.

Masking occurs because once an error exists within an LFSR, it can be canceled by new errors occurring on the inputs. Inspecting the contents of the signature analyzer several times during the testing process decreases the chance that a faulty circuit will go undetected. This technique is equivalent to periodically sampling the output of the

signature analyzer. The degree of storage compression is a function of how often the output is sampled.

Implementation Issues

It is often desirable to modify a functional register in a circuit so that it can also operate as a signature analyzer. Figure 10.18(a) shows the original circuit where the output of network N feeds a register R. In the modified circuit, shown in Figure 10.18(b), R is modified to be both a signature analyzer (MISR) as well as a scan register. The MISR is used to generate a signature when N is tested. The scan aspect of R^* is used first to initialize R^*, referred to as seeding R^*, and for scanning out the final signature. Figure 10.19 shows one design for an appropriate cell for R^*, based on the LSSD double latch SRL, where R^* is a type 1 LFSR. Referring to Figure 10.19, when clock pulses are applied to CK and then B, the register functions in a normal parallel-load mode, accepting data from D_i. When $S/T = 0$ and clocks A and B are alternately activated, the register acts as a scan register, each cell accepting the data bit from S_i and transferring it to Q^+. When $S/T = 1$, the scan input is $I = D_i \oplus S_i$, which is the condition required for the MISR. Again A and B are alternately activated. Usually these design changes have a small impact on the normal performance of the circuit. However, during the test mode, the extra gates in the path D_i to I may require that the clock rate be reduced.

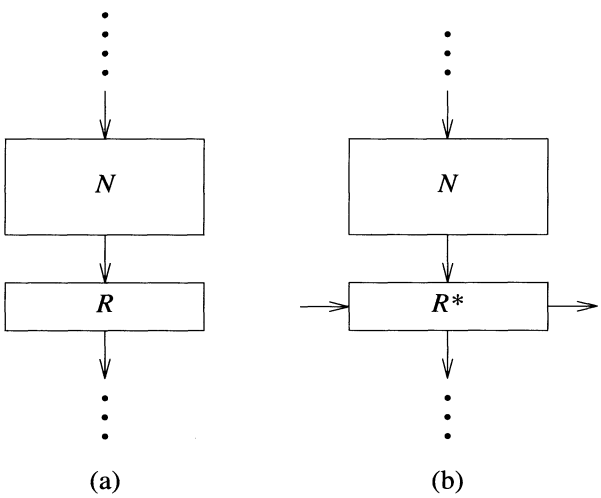


Figure 10.18 (a) Original circuit (b) Modified circuit

10.7 Concluding Remarks

Compression techniques are widely used since they are easy to implement, can be used for field test and self-testing, and can provide high fault coverage, though the correlation between error coverage and fault coverage is hard to predict.

Transition-count testing can provide good fault coverage with short, deterministic test sequences. This technique can be used to monitor asynchronous line activity if the

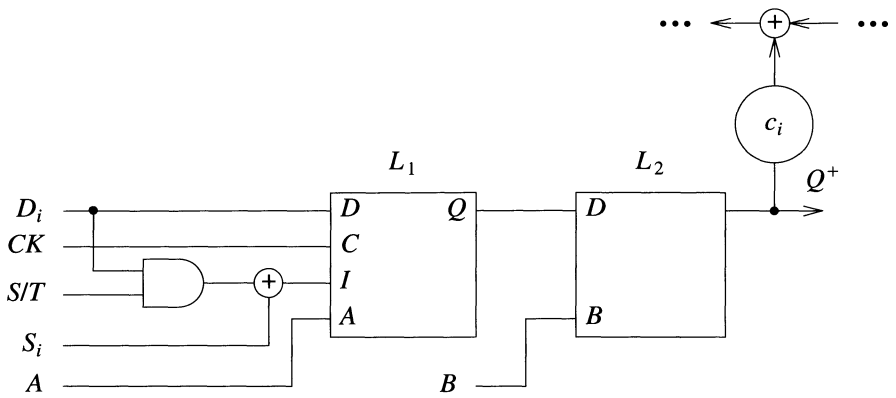


Figure 10.19 Storage cell for a signature analyzer

circuitry driving the line is race-free and hazard-free. The results depend on the order of the test patterns.

Ones counting can provide adequate fault coverage, though longer test sequences may be required. For combinational circuits the results are independent of the order of the test patterns.

All Boolean switching functions can be implemented by a circuit that is syndrome-testable. Since syndrome testing requires an exhaustive test set, it is not a practical technique for circuits having many inputs.

Signature analysis is widely used because it provides excellent fault and error coverage, though fault coverage must be determined using a fault simulator or a statistical fault simulator. Unlike the other techniques, several means exist for improving the coverage without changing the test, such as by changing the characteristic polynomial or increasing the length of the register.

REFERENCES

[Abadir 1987] M. S. Abadir, "Efficient Scan Path Testing Using Sliding Parity Response Compaction," *Proc. Intn'l. Conf. on Computer-Aided Design*, pp. 332-335, November, 1987.

[Agarwal 1983] V. K. Agarwal, "Increasing Effectiveness of Built-In Testing by Output Data Modification," *Digest of Papers 13th Annual Intn'l. Symp. Fault-Tolerant Computing*, pp. 227-233, June, 1983.

[Bardell and McAnney 1982] P. H. Bardell and W. H. McAnney, "Self-Testing of Multichip Logic Modules," *Digest of Papers 1982 Intn'l. Test Conf.*, pp. 200-204, November, 1982.

[Bardell *et al.* 1987] P. H. Bardell, W. H. McAnney, and J. Savir, *Built-in Test for VLSI: Pseudorandom Techniques,* John Wiley and Sons, New York, 1987.

[Barzilai *et al.* 1981] Z. Barzilai, J. Savir, G. Markowsky, and M. G. Smith, "The Weighted Syndrome Sums Approach to VLSI Testing," *IEEE Trans. on Computers*, Vol. C-30, No. 12, pp. 996-1000, December, 1981.

[Beauchamp 1975] K. G. Beauchamp, *Walsh Functions and Their Applications*, Academic Press, New York, 1975.

[Bhaskar 1982] K. S. Bhaskar, "Signature Analysis: Yet Another Perspective," *Digest of Papers 1982 Intn'l. Test Conf.*, pp. 132-134, November, 1982.

[Bhavsar and Heckelman 1981] D. K. Bhavsar and R. W. Heckelman, "Self-Testing by Polynomial Division," *Digest of Papers 1981 Intn'l. Test Conf.*, pp. 208-216, October, 1981.

[Bhavsar and Krishnamurthy 1984] D. K. Bhavsar and B. Krishnamurthy, "Can We Eliminate Fault Escape in Self Testing by Polynomial Division (Signature Analysis)?," *Proc. Intn'l. Test Conf.*, pp. 134-139, October, 1984.

[Brillhart *et al.* 1983] J. Brillhart, D. H. Lehmer, J. L. Selfridge, B. Tuckerman, and S. S. Wagstaff, Jr., *Factorization of* $b^n \pm 1$, $b = 2,3,5,6,7,10,11,12$, *Up to High Powers*, American Mathematical Society, Providence, 1983.

[Carter 1982] J. L. Carter, "The Theory of Signature Testing for VLSI," *14th ACM Symp. on the Theory of Computing*, pp. 66-76, May, 1982.

[Carter 1982a] W. C. Carter, "The Ubiquitous Parity Bit," *Digest of Papers 12th Annual Intn'l. Symp. Fault-Tolerant Computing*, pp. 289-296, June, 1982.

[Carter 1982b] W. C. Carter, "Signature Testing with Guaranteed Bounds for Fault Coverage," *Digest of Papers 1982 Intn'l. Test Conf.*, pp. 75-82, November, 1982.

[Chan 1977] A. Y. Chan, "Easy-to-use Signature Analysis Accurately Troubleshoots Complex Logic Circuits," *Hewlett-Packard Journal*, Vol. 28, pp. 9-14, May, 1977.

[Daniels and Bruce 1975] R. G. Daniels and W. C. Bruce, "Built-In Self Test Trends in Motorola Microprocessors," *IEEE Design & Test of Computers*, Vol. 2, No. 2, pp. 64-71, April, 1985.

[David 1984] R. David, "Signature Analysis of Multi-Output Circuits," *Digest of Papers 14th Annual Intn'l. Symp. on Fault-Tolerant Computing*, pp. 366-371, June, 1984.

[Frohwerk 1977] R. A. Frohwerk, "Signature Analysis: A New Digital Field Service Method," *Hewlett-Packard Journal*, Vol. 28, pp. 2-8, September, 1977.

[Golomb 1982] S. W. Golomb, *Shift Register Sequences*, rev. ed., Aegean Park Press, Laguna Hills, California, 1982.

[Gupta and Pradhan 1988] S. K. Gupta and D. K. Pradhan, "A New Framework for Designing and Analyzing BIST Techniques: Computation of Exact Aliasing Probability," *Proc. Intn'l. Test Conf.*, pp. 329-342, September, 1988.

[Hassan *et al.* 1983] S. Z. Hassan, D. J. Lu, and E. J. McCluskey, "Parallel Signature Analyzers — Detection Capability and Extensions," *26th IEEE Computer Society Intn'l. Conf.*, COMPCON, Spring 1983, pp. 440-445, February-March, 1983.

[Hassan and McCluskey 1983] S. Z. Hassan and E. J. McCluskey, "Increased Fault Coverage Through Multiple Signatures," *Digest of Papers 14th Annual Intn'l. Fault-Tolerant Computing Symp.*, pp. 354-359, June, 1984.

[Hayes 1976a] J. P. Hayes, "Transition Count Testing of Combinational Logic Circuits," *IEEE Trans. on Computers*, Vol. C-25, pp. 613-620, June, 1976.

[Hayes 1976b] J. P. Hayes, "Check Sum Methods for Test Data Compression," *Journal of Design Automation & Fault-Tolerant Computing*, Vol. 1, No. 1, pp. 3-17, 1976.

[Hsiao and Seth 1984] T-C. Hsiao and S. C. Seth, "An Analysis of the Use of Rademacher-Walsh Spectrum in Compact Testing," *IEEE Trans. on Computers*, Vol. C-33, No. 10, pp. 934-937, October, 1984.

[Hurst *et al.* 1985] S. L. Hurst, D. M. Miller, and J. C. Muzio, *Spectral Techniques in Digital Logic*, Academic Press, New York, 1985.

[Karpovsky and Nagvajara] M. Karpovsky and P. Nagvajara, "Optional Time and Space Compression of Test Responses for VLSI Devices," *Proc. Intn'l. Test Conf.*, pp. 523-529, September, 1987.

[Losq 1978] J. Losq, "Efficiency of Random Compact Testing," *IEEE Trans. on Computers*, Vol. C-27, No. 6, pp. 516-525, June, 1978.

[Markowsky 1981] G. Markowsky, "Syndrome-Testability Can Be Achieved by Circuit Modification," *IEEE Trans. on Computers*, Vol. C-30, No. 8, pp. 604-606, August, 1981.

[McAnney and Savir 1986] W. H. McAnney and J. Savir, "Built-In Checking of the Correct Self-Test Signature," *Proc. Intn'l. Test Conf.*, pp. 54-58, September, 1986.

[McAnney and Savir 1988] W. H. McAnney and J. Savir, "Built-in Checking of the Correct Self-Test Signature," *IEEE Trans. on Computers*, Vol. C-37, No. 9, pp. 1142-1145, September, 1988.

[Parker 1976] K. P. Parker, "Compact Testing: Testing with Compressed Data," *Proc. 1976 Intn'l. Symp. on Fault-Tolerant Computing*, pp. 93-98, June, 1976.

[Peterson and Weldon 1972] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed., MIT Press, Cambridge, Massachussetts, 1972.

[Reddy 1977] S. M. Reddy, "A Note on Testing Logic Circuits by Transition Counting," *IEEE Trans. on Computers*, Vol. C-26, No. 3, pp. 313-314, March, 1977.

[Robinson 1985] J. P. Robinson, "Segmented Testing," *IEEE Trans. on Computers*, Vol. C-34, No. 5, pp. 461-471, May, 1985.

[Robinson and Saxena 1987] J. P. Robinson and N. R. Saxena, "A Unified View of Test Compression Methods," *IEEE Trans. on Computers*, Vol. C-36, No. 1, pp. 94-99, January, 1987.

[Saluja and Karpovsky 1983] K. K. Saluja and M. Karpovsky, "Test Compression Hardware Through Data Compression in Space and Time," *Proc. Intn'l. Test Conf.*, pp. 83-88, 1983.

[Savir 1980] J. Savir, "Syndrome-Testable Design of Combinational Circuits," *IEEE Trans. on Computers*, Vol. C-29, No. 6, pp. 442-451, June 1980; and No. 11, pp. 1012-1013, November, 1980.

[Savir 1981] J. Savir, "Syndrome-Testing of Syndrome-Untestable Combinational Circuits," *IEEE Trans. on Computers*, Vol. C-30, No. 8, pp. 606-608, August, 1981.

[Savir and Bardell 1984] J. Savir and P. H. Bardell, "On Random Pattern Test Length," *IEEE Trans. on Computers*, Vol. C-33, No. 6, pp. 467-474, June, 1984.

[Savir *et al.* 1984] J. Savir, G. S. Ditlow, and P. H. Bardell, "Random Pattern Testability," *IEEE Trans. on Computers*, Vol. C-33, No. 1, pp. 79-90, January, 1984.

[Savir and McAnney 1985] J. Savir and W. H. McAnney, "On the Masking Probability with Ones Count and Transition Count," *Proc. Intn'l. Conf. on Computer-Aided Design*, pp. 111-113, November, 1985.

[Segers 1981] M. T. M. Segers, "A Self-Test Method for Digital Circuits," *Digest of Papers 1981 Intn'l. Test Conf.*, pp. 79-85, October, 1981.

[Shanks 1969] J. L. Shanks, "Computation of the Fast Walsh-Fourier Transform," *IEEE Trans. on Computers*, Vol. C-18, No. 5, pp. 457-459, May, 1969.

[Smith 1980] J. E. Smith, "Measures of the Effectiveness of Fault Signature Analysis," *IEEE Trans. on Computers*, Vol. C-29, No. 6, pp. 510-514, June, 1980.

[Sridhar *et al.* 1982] T. Sridhar, D. S. Ho, T. J. Powell, and S. M. Thatte, "Analysis and Simulation of Parallel Signature Analyzers," *Digest of Papers 1982 Intn'l. Test Conf.*, pp. 661-665, November, 1982.

[Susskind 1983] A. K. Susskind, "Testing by Verifying Walsh Coefficients," *IEEE Trans. on Computers*, Vol. C-32, No. 2, pp. 198-201, February, 1983.

[Tzidon *et al.* 1978] A. Tzidon, I. Berger, and M. Yoeli, "A Practical Approach to Fault Detection in Combinational Networks," *IEEE Trans. on Computers*, Vol. C-27, No. 10, pp. 968-971, October, 1978.

[Williams *et al.* 1986] T. W. Williams, W. Daehn, M. Gruetzner, and C. W. Starke, "Comparison of Aliasing Errors for Primitive and Non-Primitive Polynomials," *Proc. Intn'l. Test Conf.*, pp. 282-288, September, 1986.

[Williams *et al.* 1987] T. W. Williams, W. Daehn, M. Gruetzner, and C. W. Starke, "Aliasing Errors in Signature Analysis Registers," *IEEE Design & Test of Computers*, Vol. 4, pp. 39-45, April, 1987.

[Zorian and Agarwal 1984] Y. Zorian and V. K. Agarwal, "Higher Certainty of Error Coverage by Output Data Modification," *Proc. Intn'l. Test Conf.*, pp. 140-147, October, 1984.

PROBLEMS

10.1 For the ones-count compression technique, show that if $1C(R) = 0$ or m, then no error masking can occur.

10.2 The number of binary sequences of length m having a transition count of r is $2\binom{m-1}{r}$. Based on the concept of the transition count of a sequence, show that $\binom{m}{r} = \binom{m-1}{r} + \binom{m-1}{r-1}$. Don't use the fact that $\binom{m}{r} = \frac{m!}{r!(m-r)!}$.

10.3 Prove Theorem 10.2.

10.4 Assume that a single-output combinational circuit C is tested using the parity-check compression technique along with an exhaustive test set, and that the output stream has odd parity. Show that this scheme will detect all primary input and output stuck-at faults associated with C.

10.5 Prove that any two-level (AND-OR) irredundant circuit that realizes a unate function in all its variables is syndrome-testable.

10.6 The function $F = x_1x_2 + \bar{x}_2\bar{x}_3$ is not syndrome-testable. Determine which stuck-at fault associated with x_2 is not syndrome-detectable, and show that the modified function $F' = cx_1x_2 + x_2x_3$ is syndrome-testable, where c is a new primary input. Note that for $c = 1$, $F' = F$.

10.7 For an autonomous LFSR, show that if its initial state is not the all-0 state, then it will never enter the all-0 state.

10.8 Consider an autonomous LFSR whose characteristic polynomial is primitive.

a. Show that if at time t its state is in error, then at any time after t its state will always be in error.

b. Does the result for part (a) hold true when $P(x)$ is not a primitive polynomial? If so, prove it; otherwise show an example.

10.9 Consider an n-input NAND gate, denoted by G_n. All single and multiple stuck faults associated with G_n can be detected by the unique minimal test set T consisting of $n + 1$ test vectors shown below:

$$\begin{aligned} u &= (1,1,.....,1) \\ e_1 &= (0,1,1,...,1,1) \\ e_2 &= (1,0,1,...,1,1) \\ &\vdots \\ e_n &= (1,1,1,...1,0). \end{aligned}$$

The correct response to u and e_i is 0 and 1 respectively. Every complete TC test sequence X_n for G_n must include every test pattern in T, hence $|X_n| \geq n + 1$.

a. Show that $X_1 = ue_1$ and $X_2 = e_1ue_2$ are minimal TC tests for $n = 1$ and 2 respectively.

b. For $n > 2$, the minimal value of $|X_n|$ is $n + 2$, as stated in the following theorem.

Theorem [Hayes 1976a]: $X_n = ue_1e_2...e_{n-1}e_ne_1$ is a minimal TC test of length $n+2$ for both single and multiple stuck-type faults for an n-input NAND gate G_n. Prove this theorem.

10.10 For an n-input NAND gate G_n, how many single and multiple stuck faults are distinguishable and how many have a unique transition count?

10.11 Prove the following theorem [Hayes 1976a]:

Let T be any test set for a two-level circuit C. Let R_0 denote the fault-free response of C to any sequence X containing every member of T. No single or multiple fault can change the response of C to X from R_0 to $\bar{R}_0$.

10.12 Let $T = [T^0, T^1]$ be an arbitrary single-fault test set for a two-level sum of product irredundant circuit C containing $r = p + q$ vectors, where $T^0 = \{t_1^0, t_2^0, ..., t_p^0\}$ and $T^1 = \{t_1^1, t_2^1, ..., t_q^1\}$. It is well known that T also detects all multiple stuck-type faults.

a. Show that the sequence $X^2 = t_q^1 t_1^1 t_2^1 ... t_q^1 t_p^0 t_1^0 t_2^0 ... t_p^0$ of length $r+2$ is a TC test for C with respect to both single and multiple faults.

b. Assume T is a minimal test set. Show that (1) if $D = 0$ or 1, the sequence X^* defined in Theorem 10.5 is a minimal length TC test; (2) for $D = 2$, both X^* and X^2 are of the same length; and (3) for $D > 2$, X^2 is shorter than X^*.

10.13 Consider the circuit shown in Figure 10.20. Construct a minimal-length test set T for all stuck-at faults. Construct X^* and X^2 (defined in Problem 10.12). Is X^* a TC test for all multiple faults?

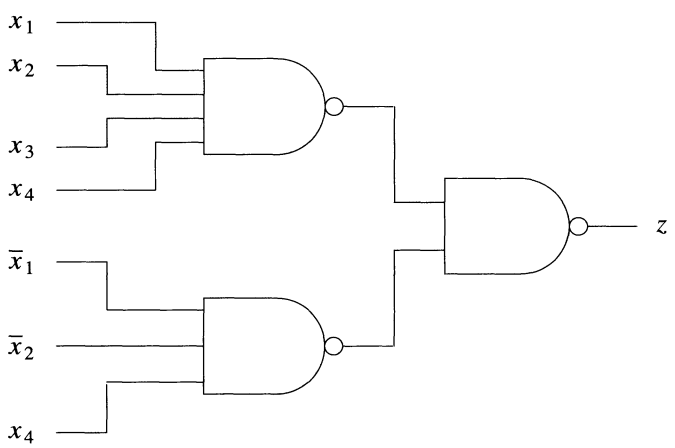


Figure 10.20

10.14 Construct a minimal-length test sequence T for all single stuck-at faults in a 3-input NAND gate and identify one fault not TC-detectable by T.

10.15 A circuit implementing the function $f = xy + \bar{y}z$ is to be tested using the syndrome-test method. Show that the single faults z s-a-0 and z s-a-1 are not detected,

while all other single stuck-at faults are detected.

10.16 Consider a circuit implementing the function $f = x_1x_3 + x_2\bar{x}_3$ and an input sequence consisting of all eight input patterns generated by a 3-bit binary counter that starts in the state $(x_3,x_2,x_1) = (0,0,0)$ and produces the sequence (0,0,0), (0,0,1), (0,1,0), ..., (1,1,1). Assume the response is compressed using a type 2 signature analyzer that has a characteristic polynomial $x^4 + x^3 + 1$ and whose initial state is all 0s. Verify that the following faults are detected.

a. x_3 *s-a-*0

b. x_3 *s-a-*1

c. x_1 *s-a-*0

d. x_1 *s-a-*1

10.17 Verify the entries in Figures 10.7 and 10.8.