

Integrated On-Line and Off-Line Error Detection Mechanisms in the Coding Theory Framework *

MARK G. KARPOVSKY, *Fellow, IEEE*[†]

Abstract—In this paper we present an approach for combining on-line concurrent checking (CC) with off-line built-in self-test (BIST). We will show that a reduction of an aliasing probability can be obtained for manufacturing testing by monitoring the output of a concurrent checker and a reduction of a probability of not detecting an error in the computing mode can be obtained by a short periodic BIST. We will present a technique for optimal selection of error-detecting codes for combined on-line CC and off-line space-time compression of test responses for BIST and estimate probabilities of not detecting an error for the approach based on integrating CC and BIST. We also present a technique for on-line error-detection in space-time compressors of test responses for BIST.


Key words—Built-in self-test, on-line detection, space-time compression of test responses, multiple-input linear feedback registers, error detecting codes.

*This work has been supported by the NSF under Grant MIP 920847 and by the NATO under Grant 910411.

[†]The author is with the Department of Electrical, Computer and Systems Engineering, Boston University, Boston, MA 02215.

I. Introduction

Off-line testing techniques such as Built-In Self-Test (BIST) and boundary scan have been the focus of VLSI test engineers concerned with product quality. BIST techniques have been widely used for manufacturing testing and repair. The block diagram for a BIST design is given by Fig. 1.

 For the block diagram of Fig. 1 space compression (SC) may be useful for overhead reduction only for the case of a large number of outputs of the device-under-test. It was shown in [1], [2] that optimal space compressors (SCs) are linear networks computing syndromes of linear error detecting codes. If a (n, s, d) code V_{SC} is used for space compression, then the corresponding SC can be constructed from XOR gates only, the SC has $m = n - s$ outputs ($m \ll n$, see Fig. 1), and any error which distorts signals at most $d - 1$ output lines for the device-under-test (D) at any given moment cannot be masked in the SC. Moreover, any error which is detected by V_{SC} cannot be masked in the corresponding SC.

The most popular technique for time compression (TC) of test responses is based on usage of multiple input linear feedback shift registers (MISRs). The structure of MISR feedback taps is defined by the corresponding binary generating polynomial $P(x)$ [3]. For a m -bit MISR the degree of $P(x)$ is equal to m , and to decrease a probability of masking an error in the TC (time aliasing) in many cases $P(x)$ should be primitive [3].

Any m -bit MISR can be described as a network computing syndromes for the corresponding linear nonbinary $(T, T - 1, 2)$ Reed-Solomon code [11], V_{TC} , over $GF(2^m)$

where the length, T , of the code is equal to the number of test patterns applied [4]. If $l \geq 1$ MISRs are used for self-testing or self-diagnosis (multisignature analysis [5], [6]), then the corresponding TC is the network computing syndromes for the $(T, T-l, l+1)$ Reed-Solomon code, V_{TC} , over $GF(2^m)$.

Analysis of space and time aliasing probabilities in terms of weight distributions of the corresponding codes V_{SC} and V_{TC} for different error models is presented in [4]. Generalization of these results for a board or system level testing was given in [6].

Since system availability is becoming a key feature of computer systems, on-line error detecting techniques based on concurrent checking (CC) are extremely important for design of fault-tolerant systems. The block diagram for on-line error detection is presented at Fig. 2.

Fig 2

For the block diagram of Fig. 2, the parity prediction network, $R(D)$, is built in such a way that for any input if there are no errors in the original device, D , and $R(D)$, then an extended output $(y_1, \dots, y_k, y_{k+1}, \dots, y_n)$ is a codeword of the binary (n, k, d) systematic code V_{CC} . All errors resulting in distortions of at most $d-1$ bits in $(y_1, \dots, y_k, y_{k+1}, \dots, y_n)$ will be detected on-line by the CC. Used as V_{CC} are, for example, parity prediction $(k+1, k)$ codes, duplication $(2k, k)$ codes, $(k + \lceil \log_2(k+1) \rceil, k)$ nonlinear Berger codes and $(n, n - \lceil \log_2(n+1) \rceil)$ Hamming codes for computer memories [7].

Concurrent checkers (CCs) are networks computing syndromes for the corresponding codes V_{CC} and verifying that these syndromes are not equal to zero. Techniques for design of self-checking CCs can be found, e.g., in [7]-[10].

Example 1. To illustrate the relations between off-line space-time compression

$$V_{LSE} \mid 1002 \mid 3$$

of test responses, on-line CC and the corresponding codes V_{SC} , V_{TC} and V_{CC} let us consider the case when the original device, D , is the control ROM for MC68020.

In this case the number of outputs for the original device $k = 116$ and for concurrent checking one can select a $(123, 116, 3)$ Hamming code as V_{CC} [11]. This will result in $r = 7$ redundant outputs for $R(D)$ and on-line detection of all single and double errors in $n = 123$ outputs of the expanded ROM. In addition to this, all errors distorting three or more output bits will not be detected with probability 2^{-7} (assuming all these errors being equiprobable).

For space compression of responses from the expanded ROM a $(123, 95, 9)$ BCH code [11] can be used. In this case the SC will have $m = 28$ outputs, all errors resulting in a distortion of at most 8 out of 123 output bits for the expanded ROM will not be masked in the SC and the probability of masking (space aliasing) for errors distorting more than 8 bits is 2^{-28} .

For time compression of output sequences from the SC in this case one can use a Reed-Solomon code with distance 2 over $GF(2^{28})$ [11]. Then the corresponding TC can be implemented by the MISR with primitive generating polynomial $P(x) = x^{28} \oplus x^3 \oplus 1$ and the probability of time aliasing is 2^{-28} . The total overhead in terms of equivalent two-input gates for on-line and off-line error detection based on the above codes is about 15%. \square

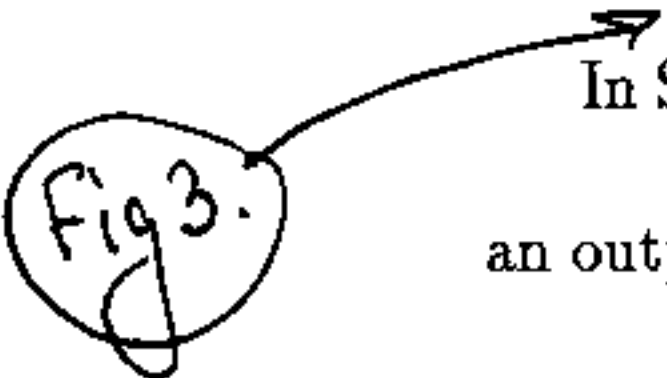
Thus, a strong relationship exists between concurrent checking and BIST since both are based on computing syndromes of corresponding error detecting codes V_{CC} , and V_{SC} , V_{TC} . This provides a natural framework to integrate both approaches.

We note that current design approaches have major drawbacks since mechanisms for on-line and off-line error detection are chosen separately, without consideration of any potential interactions. These approaches are not tailored to the most efficient combined utilization of the available silicon area. Also, in treating on-line and off-line techniques separately, performance gains are lost. For example, in the design of an off-line fault detection/diagnosis hardware, the fact that the circuit may also include support logic for on-line error-detection is not considered.

Although there is a significant overlap between on-line and off-line error detection techniques, only a few papers have been published exploring the potential for combining these approaches. The idea of merging on-line and off-line BIST was suggested in [22], [23], where partitioning of the logic, placement and design of test circuitry was discussed. In [12], [13], a concurrent BIST (CBIST) approach has been proposed. The off-line testing resources are modified for this approach such that during system operation, they can observe normal inputs and outputs. When a normal input matches a test pattern, the circuit output is compressed into a developing signature. Another approach (UBIST) was proposed in [14]. For this approach, test vector generators are used for off-line testing of concurrent checkers, ensuring that all test patterns are applied. It may, though, be difficult to implement UBIST for devices with many input lines. A combination of UBIST with boundary scan was suggested in [24]. Applications of linear cellular automata for concurrent checking, signature analysis and boundary scan have been studied in [25]. Modifications of conventional BIST designs to improve fault coverages for stuck-at faults by monitoring output of a parity predictor during manufacturing testing were proposed in [24]. Applications

of this approach for ISCAS benchmarks and ROMs are also presented in [26]. Implementations of built-in testable error detection and error correction circuits have been presented in [15]. The problem of parity calculation for parity checking with BIST and pseudorandom testing was considered in [16].

In this paper we will investigate the general framework presented in Fig. 3 where off-line built-in self-test mechanisms of Fig. 1 are coupled with on-line error detection mechanisms of Fig. 2. In the proposed scheme, the original device, D augmented with parity (code) predictor, $R(D)$, is not required to be fault-secure.

 In Section II we will discuss how much improvement one can achieve by monitoring an output of a concurrent checker during off-line manufacturing testing. We will also show that on-line concurrent checking only may not be sufficient for detection of stuck-at faults and a short periodic off-line BIST can drastically increase a probability of detection for these faults. Section III will be devoted to optimal selection of codes V_{CC} , V_{SC} , V_{TC} , for concurrent checking, space compression and time compression. In Section IV we present analysis of probabilities of error detection for the combined BIST and concurrent checking scheme. Design issues related to on-line concurrent checking of space and time compressors of test responses will be discussed in Section V.

II. Manufacturing and Field Testing by Combining BIST and Concurrent Checking (CC)

First, let us show that monitoring the output of the concurrent checker (CC) during off-line manufacturing BIST results in a drastic decrease of the aliasing probability. To estimate aliasing we have to choose a model describing a distribution of errors at outputs $y_1, \dots, y_k, y_{k+1}, \dots, y_n$ ($n = k + r$) of the device.

There are two important components of any error model — temporal and spatial. The first, temporal, models the correlation between the errors caused by different input vectors. For combinational circuits and combinational faults, errors can be assumed to be independent in time [17]–[21]. In this case if input vectors are random, then there is no time correlation between errors due to any two input vectors.

Let $e(t) = y(t) \oplus \tilde{y}(t)$, where $y(t)$ and $\tilde{y}(t)$ are fault-free and distorted outputs at the moment t . Then the space distribution of errors at the output of the device is given by probabilities $p_0, p_1, \dots, p_{2^n-1}$ ($p_i = \Pr\{e(t) = i \text{ for any } t\}$, $p_0 + p_1 + \dots + p_{2^n-1} = 1$). Methods of statistical determination of parameters for the above models and their applicability can be found in [17]. These models have been widely used in estimations of off-line aliasing (see e.g. [4], [17], [18], [19], [21]).

In the case when space compression (SC) is used before time compression for reduction of an overhead required for BIST (Fig. 3) to estimate a probability of aliasing in a time compressor, distributions $p_0, p_1, \dots, p_{2^n-1}$ of errors in the device should be replaced by distributions $q_0, q_1, \dots, q_{2^m-1}$ of distortions $\Delta z(t)$ at the output of the SC, where $q_i = \Pr\{\Delta z(t) = i\}$. If the space compression is based on the

$(n, n - m)$ code, V_{SC} , with check matrix H_{SC} , then

$$z(t) = H_{SC}y(t); z(t) \oplus \Delta z(t) = H_{SC}(y(t) \oplus e(t)); \text{ and } \Delta z(t) = H_{SC}e(t). \quad (1)$$

Thus,

$$q_i = \sum_{j: H_{SC}j = i} p_j. \quad (2)$$

For example, for 2^n -ary symmetrical errors at the output of the device [17], we have symmetrical 2^m -ary errors $\Delta z(t)$ at the output of the SC with

$$q_i = \begin{cases} 1 - p + (2^{n-m} - 1)p(2^n - 1)^{-1}, & \text{for } i = 0; \\ 2^{n-m}p(2^n - 1)^{-1}, & \text{for } i \neq 0; \end{cases} \quad (3)$$

where $p = \Pr\{e(t) \neq 0 \text{ for any } t\}$.

Example 2. Suppose that the original device have $k = 4$ output lines and the $(5, 4, 2)$ single error detecting code, V_{CC} , is used for concurrent checking (CC). Then there are 15 nonzero 5-bit error patterns with even weights (numbers of ones) which are not detectable by CC.

If the $(5, 2, 3)$ Hamming code with check matrix $H_{SC} = \begin{bmatrix} 00011 \\ 01100 \\ 10101 \end{bmatrix}$ is used for space compression ($n = 5, m = 3$, see Fig. 1), then by (3) we have 2^3 -ary symmetrical errors at the outputs of the SC with $q_i = \Pr\{\Delta z(t) \neq 0 \text{ for any } t\} = \frac{4}{31}p$, ($i = 1, 2, \dots, 7$), where $p = \Pr\{e(t) \neq 0 \text{ for any } t\}$ and only one error, (01111) , out of 31 nonzero errors is not detected by the CC and masked by the SC.

If the 3-bit MISR based on primitive polynomial $P(x) = x^3 \oplus x \oplus 1$ is used for time compression and $T = 7$ test patterns are applied, then $e = (e(0), \dots, e(6))$ is not detected by both concurrent checking and signature analysis iff a number of ones in $e(t)$ is even ($t = 0, 1, 2, \dots, 6$) and

$$\alpha^6(H_{SC}e(0)) \oplus \alpha^5(H_{SC}e(1)) \oplus \dots \oplus \alpha(H_{SC}e(5)) \oplus H_{SC}e(6) = 0,$$

where α is primitive in $GF(2^3)$ and $P(\alpha) = \alpha^3 \oplus \alpha \oplus 1 = 0$ ($\alpha^i \neq \alpha^j$, $i \neq j$, $i, j = 0, 1, \dots, 6$). For $p = \Pr\{e(t) \neq 0 \text{ for any } t\} = 0.1$ we have for a probability, P_{ON} , of not detecting $e \neq 0$ by concurrent checking $P_{ON} = 0.211$. Similarly for a probability, P_{OFF} , of not detecting $e = (e(0), \dots, e(6)) \neq 0$ by the off-line signature analysis using results from [4] for the aliasing probability for symmetrical errors we have $P_{OFF} = 0.076$ and, finally, for a probability, $P_{ON,OFF}$, of not detecting e by both CC and the MISR we have $P_{ON,OFF} = 0.037$. (Exact formulas for P_{ON} , P_{OFF} and $P_{ON,OFF}$ will be presented in Section IV.) \square

Tables \rightarrow

Off-line and on-line error detection techniques can be designed to complement each other. A periodic BIST can be aimed at precisely those stuck-at faults that escape detection by CC.

To justify this approach, we note first that CCs, in most cases, provide for a poor fault-coverage with respect to stuck-at faults at primary input lines; these faults, though, form an important class of permanent faults, since in many cases interconnections between components are less reliable than components themselves. For example, parity prediction CCs for adders or multipliers [7], based on comparing


VLSI / 002 / 9

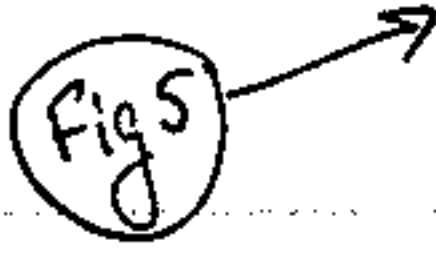
parities of input operands, internal carries and outputs, cannot detect many input stuck-at faults.

In addition to this, since an overhead for CCs grows rapidly with an increase in the error detecting capability of the corresponding (n, k) code, V_{CC} , only codes with a very limited error detecting capability (like $(k + 1, k)$ parity, Berger codes or Hamming codes for memories) have been used. This may result in a low fault coverage for internal stuck-at faults. The following three examples illustrate this situation.

Finally, there is a problem of detection of stuck-at faults in the CC itself. This problem can be solved by using self-checking checkers. However, self-checking checkers require considerable additional overheads [7]–[10].


Example 3. Consider a class of networks H_m ($m = 1, 2, \dots$) with 2^m inputs and $m + 1$ outputs defined recursively by Fig. 4. (H_m ($m = 1, 2, \dots$) have been used for decoding of extended Hamming codes.)

 The parity predictor, $R(H_4)$, and the CC for H_4 based on the $(6, 5, 2)$ parity code are presented in Fig. 5. From Fig. 5 one can see that out of 86 single stuck-at faults (SSFs) in H_4 , 30 cannot be detected by the CC.

 When m is growing, the fault coverage for SSFs detectable by the parity prediction in H_m is converging to 50%. (One can improve this fault coverage by using CCs based on codes with distances more than 2, but already for $(n, m + 1, 3)$ Hamming codes with $r = n - (m + 1) \geq \lceil \log_2(n + 1) \rceil$, the corresponding parity predictors require about $2^{m-1} \lceil \log_2(n + 1) \rceil$ two-input gates.) We note, however, that a fault coverage close to $1 - 2^{-(m+1)}$ can be obtained for SSFs in this case by off-line testing using a $(m + 1)$ -bit MISR. \square

The previous example demonstrates that for linear devices and CCs based on linear codes, fault coverages for SSFs may be very low. We will show now that this may be true also for nonlinear devices and CCs based on more powerful and not necessarily linear codes (e.g. Berger codes).

Example 4. Let us consider a class Q_m of devices with m inputs and 2^{m-1} outputs, defined recursively by Fig. 6 ($m = 2, 3, 4, \dots$). For Q_4 only 7 out of 25 SSFs cannot be detected by the CC, based on the parity prediction (9, 8) code, or by a nonlinear (12, 8) Berger code. As m grows, the fraction of these undetectable SSFs in Q_m converges to $1/3$. \square

 **Example 5.** Let us consider a $(N \times k)$ ROM with concurrent parity checking. In this case, stuck-at faults in the address decoder that result in a selection of a wrong cell, as well as multiple faults within a cell, will not be detected. These stuck-at faults can be detected with a probability $1 - 2^{-(k+1)}$ by off-line testing, using a $(k + 1)$ -bit MISR at the output of the ROM. \square

The previous examples indicate that CC only may result in a low fault coverage for SSFs and a short periodic BIST detects in most cases a very large percentages of SSFs. Techniques for optimal selection of error-detecting codes for CC and BIST which complement each other are presented in the next section.

III. Optimal Selection of Error-Detecting Codes for Concurrent Checking (CC) and Space-Time Compression (STC) of Test Responses

The space compressor (SC) is a combinational network with n inputs $y(t) = (y_1(t), \dots, y_n(t))$ and m outputs $z(t) = (z_1(t), \dots, z_m(t))$ such that $m \ll n$ and for a given class E of errors, the SC has the following error propagating property. Let $\tilde{y}(t) = y(t) \oplus e(t)$, $e(t) \in E$ and outputs of the SC for $y(t)$ and $\tilde{y}(t)$ are $z(t)$ and $\tilde{z}(t)$. Then the SC is error propagating iff for any $e(t) \in E$, $z(t) \neq \tilde{z}(t)$.

A SC is self-testing iff the fault-free response of the device is a test detecting all single stuck-at faults in the SC. Design methods for self-testing error propagating optimal SCs (STEP SCs), that minimize a number of observation points for a given class of errors E , have been developed [1], [2]. (The major difference between STEP SCs and totally self-checking (TSC) checkers is that the inputs and outputs of STEP SCs are not necessarily codewords of a concurrent code V_{CC} . This implies that a STEP SC could have a single output; a TSC checker, however, must have at least two outputs [7]).

It was shown in [1], [2] that STEP SCs can be designed as networks computing syndromes for linear codes. If a (n, s, d) code, V_{SC} , was chosen for space compression, then $m = n - s$ and E is a set of all errors with a multiplicity, at most, $d - 1$.

In the case when CC and space compression are integrated (Fig 3), there is a problem of optimal selection of the corresponding (n, k) codes, V_{CC} , for concurrent

checking and (n, s) codes V_{SC} for space compression. For example, if $V_{CC} = V_{SC}$, then any error missed by the CC will also be missed by the SC. This is a bad choice for (V_{CC}, V_{SC}) .

Below, we will describe our approach for solution of this problem minimizing a fraction, η , of errors which are undetectable by the CC and masked in the space compression process; i.e.,

$$\eta = |V_{CC} \cap V_{SC}|2^{-k}, \quad 2^{-m} \leq \eta \leq 1, \quad m = n - s. \quad (4)$$

If V_{CC} is the parity $(n, n-1, 2)$ code, then V_{SC} should have an odd distance to minimize η . In this case, $|V_{CC} \cap V_{SC}| = 2^{n-m-1}$ and $\eta = 2^{-m}$.

For the general case, to minimize η , one can use as V_{CC} and V_{SC} shortened BCH or shortened cyclic Hamming codes [11] such that sets of roots of their generating polynomials do not intersect.

This can be implemented in the following way. Let $2^{a-1} - 1 < n \leq 2^a - 1$. Then check matrices H_{CC} and H_{SC} for V_{CC} and V_{SC} can be selected as

$$H_{CC} = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_v & \alpha_v^2 & \cdots & \alpha_v^{n-1} \end{bmatrix}, \quad H_{SC} = \begin{bmatrix} 1 & \alpha_{v+1} & \alpha_{v+1}^2 & \cdots & \alpha_{v+1}^{n-1} \\ 1 & \alpha_{v+2} & \alpha_{v+2}^2 & \cdots & \alpha_{v+2}^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_{v+w} & \alpha_{v+w}^2 & \cdots & \alpha_{v+w}^{n-1} \end{bmatrix}, \quad (5)$$

where α_i ($i = 1, \dots, v+w$) are different primitive elements in $GF(2^a)$ ($\alpha_i^p \neq \alpha_i^q$; $p \neq q$; $p, q < 2^a - 1$). Then $k = n - av$, $m = aw$ and $\eta = 2^{-aw} = 2^{-m}$.

In the case when the CC is implemented by duplicating the original device, V_{CC} is the $(2k, k, 2)$ duplication code and any shortened BCH or Hamming code V_{SC} with check matrix

$$H_{SC} = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{2k-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{2k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_l & \alpha_l^2 & \cdots & \alpha_l^{2k-1} \end{bmatrix}, \quad (6)$$

(where α_i ($i = 1, \dots, l$) are different primitive elements in $GF(2^a)$ and $2^{a-1} - 1 < 2k \leq 2^{a-1}$) is optimal for space compression. In this case V_{SC} is the $(2k, 2k - la, 2l + 1)$ code, $m = la$, $\log_2 |V_{CC} \cap V_{SC}| = k - la$ and $\eta = 2^{-m} = 2^{-la}$.

If V_{CC} is a t out of n nonlinear nonsystematic code [7] (all codewords of V_{CC} have weight t , $t < n$), then any code V_{SC} with a distance d such that $\lceil \frac{d}{2} \rceil > t$ is optimal; if $\lceil \frac{d}{2} \rceil \leq t$, then d should be odd.

Example 6. Let $n = 7$ ($a = 3$). We will construct two $(7, 4, 3)$ Hamming codes V_{SC} and V_{CC} such that $|V_{CC} \cap V_{SC}| = 2$ and $\eta = 2^{-m} = 2^{-3}$. (Representation of $GF(2^3)$ based on the primitive polynomial $x^3 \oplus x \oplus 1$ is given in Table 1).

Selecting $\alpha_1 = \alpha$ and $\alpha_2 = \alpha^3$ we have from (5)

$$H_{CC} = [1\alpha\alpha^2\alpha^3\alpha^4\alpha^5\alpha^6] = \begin{bmatrix} 0010111 \\ 0101110 \\ 1001011 \end{bmatrix}, \quad H_{SC} = [1\alpha^3\alpha^6\alpha^2\alpha^5\alpha\alpha^4] = \begin{bmatrix} 0011101 \\ 0100111 \\ 1110100 \end{bmatrix},$$

and $V_{CC} \cap V_{SC} = \{0000000, 1111111\}$. \square

Example 7. Let us apply the above approach to the design of a CC and a SC

VLSI / 002 / 14

for the control ROM for MC68020 with $k = 116$ outputs from Example 1.

We assume that a shortened $(123, 116, 3)$ Hamming code V_{CC} detecting single and double errors is used for on-line error detection, and a shortened 8 error-detecting $(123, 95, 9)$ BCH code V_{SC} is used for space compression (see Example 1).

Then the best pair (V_{CC}, V_{SC}) minimizing η is defined by the following check matrices ($\alpha_1 = \alpha$, $\alpha_2 = \alpha^3$, $\alpha_3 = \alpha^5$, $\alpha_4 = \alpha^7$, $\alpha_5 = \alpha^9$, α is a primitive in $GF(2^7)$, $\alpha^{127} = \alpha^0 = 1$.)

$$H_{CC} = [1\alpha\alpha^2\alpha^3\cdots\alpha^{122}], H_{SC} = \begin{bmatrix} 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{366} \\ 1 & \alpha^5 & \alpha^{10} & \cdots & \alpha^{610} \\ 1 & \alpha^7 & \alpha^{14} & \cdots & \alpha^{854} \\ 1 & \alpha^9 & \alpha^{18} & \cdots & \alpha^{1098} \end{bmatrix}. \quad (7)$$

In this case, the SC has $r = 4 \times 7 = 28$ outputs and the fraction of errors not detectable by the CC and masked in the SC is $\eta = 2^{-28}$. \square

Let us consider now the problem of optimal selection of time compressors (TCs) for a redundant device with a given CC code V_{CC} .

We assume that V_{CC} is a binary linear (n, k) code ($n \leq 2k$), the TC is a m -bit MISR and there is no space compression ($m = n$). Then the TC is a network computing syndrome for the $(T, T - 1, 2)$ Reed Solomon code, V_{TC} , over $GF(2^n)$, where T is a number of test patterns applied.

Let us denote the check matrix for this code as

$$H_{TC} = [h_0, h_1, \dots, h_{T-1}], \quad (8)$$

where $h_i \in GF(2^n)$.

Then $e(t) = (e_0(t), \dots, e_{n-1}(t))$ ($e_i(t) \in \{0, 1\}, t = 0, 1, \dots, T-1$) is not detected iff $e(t) \in V_{CC}$ for any t and $e = (e(0), \dots, e(T-1)) \in V_{TC}$ or from (8)

$$h_0 e(0) \oplus h_1 e(1) \oplus \dots \oplus h_{T-1} e(T-1) = 0. \quad (9)$$

If $H_{TC} = (1, 1, \dots, 1)$ (which corresponds to case when the MISR is a collection of disconnected T-flipflops), one can select $e(1), \dots, e(T-1)$ arbitrarily such that $e(i) \in V_{CC}$ ($i = 1, \dots, T-1$) and find from (9) in a unique way $e(0) \in V_{CC}$. Thus, the fraction of errors which are not detected by off-line BIST and CC in this case is equal to $(2^{k(T-1)} - 1)(2^{kT} - 1)^{-1} \simeq 2^{-k}$.

If a primitive MISR is used as a TC, then $H_{TC} = (\alpha^{T-1}, \alpha^{T-2}, \dots, \alpha, 1)$ (α is a primitive in $GF(2^n)$; $\alpha^i \neq \alpha^j$; $i, j = 0, 1, \dots, 2^n - 2$), and from (9) e is not detected by CC and BIST for $T \leq 2^n - 1$ iff $e(t) \in V_{CC}$ for any t and

$$e(0)\alpha^{T-1} \oplus e(1)\alpha^{T-2} \oplus \dots \oplus e(T-2)\alpha \oplus e(T-1) = 0. \quad (10)$$

Now let us fix $e(0) \in V_{CC}, \dots, e(T-3) \in V_{CC}$. Then we have the following equations for $e(T-2)$ and $e(T-1)$

$$e(T-1) \in V_{CC}, e(T-2) \in V_{CC} \text{ and,}$$

$$e(T-1) \oplus \alpha e(T-2) = B, \quad (11)$$

where $B = e(0)\alpha^{T-1} \oplus \dots \oplus e(T-3)\alpha^2$.

For the (n, k) code, V_{CC} , we can always assume that its check matrix H_{CC} is represented in the following form

$$H_{CC} = [P|I_r], \quad (12)$$

where I_r is the $(r \times r)$ identity matrix ($r = n - k$) and P is the $(r \times k)$ matrix with nonzero rows and $k \geq r$ [11].

Denote $B = (B_0, \dots, B_{n-1})$, $Y = (e(T-1), e(T-2)) = (e_0(T-1), \dots, e_{n-1}(T-1), e_0(T-2), \dots, e_{n-1}(T-2))$ ($B_i, e_i(T-1), e_i(T-2) \in \{0, 1\}$).

If the TC is a primitive MISR with external XORs and generating polynomial $P(x) = x^n \oplus p_{n-1}x^{n-1} \oplus \dots \oplus p_1x \oplus 1$ [3], then (11) can be represented over $GF(2)$ as a system of $n + 2r$ linear equations with $2n$ unknowns $e_0(T-1), \dots, e_{n-1}(T-1), e_0(T-2), \dots, e_{n-1}(T-2) \in \{0, 1\}$ in the following matrix form

$$MY = (\overbrace{0, \dots, 0}^{2r}, B_0, \dots, B_{n-1})^{tr}, \quad (13)$$

where

$$M = \left[\begin{array}{c|c|c} P & I_r & 0 \\ \hline 0 & P & I_r \\ \hline \vdots & \vdots & \vdots \\ \hline I_n & \begin{array}{c} 0 \quad 0 \quad \dots \quad 0 \\ \hline I_{n-1} \end{array} & \begin{array}{c} 1 \\ p_1 \\ p_2 \\ \vdots \\ p_{n-1} \end{array} \end{array} \right] \quad \left. \vphantom{\begin{array}{c} P \\ 0 \\ \vdots \\ I_n \end{array}} \right\} n + 2r, \quad (14)$$

Since for any primitive $P(x) = x^n \oplus p_{n-1}x^{n-1} \oplus \dots \oplus p_1x \oplus 1$ we have $p_{n-2} \oplus \dots \oplus p_1 \oplus 1 \neq 0$, and there are no all zero rows in P , it is easy to show that rank of M over $GF(2)$ is $n + 2r$ and there are $2^{2n-(n+2r)} = 2^{k-r}$ solutions Y for (13) for any B .

Thus, in this case the fraction of errors which are not detectable by off-line BIST and CC is equal to $(2^{k(T-2)}2^{k-r} - 1)(2^{kT} - 1)^{-1} \simeq 2^{-k-r}$, and the transition from a nonprimitive MISR to a primitive one results in a reduction of probability of masking by the factor of $2^{-r} = 2^{-(n-k)}$.

Example. 8 For the original device, D , with $k = 2$ output lines, concurrent checking based on the $(3, 2, 2)$ code with $H_{CC} = [111]$ and time compression by the

primitive MISR with $P(x) = x^3 \oplus x \oplus 1$ we have

$$M = \begin{bmatrix} 111 & 000 \\ 000 & 111 \\ 100 & 001 \\ 010 & 101 \\ 001 & 010 \end{bmatrix},$$

rank $M = 5$. In this case (13) has $2^{k(T-2)}2^{2n-\text{rank } M} - 1 = 2^{2(T-2)}2 - 1 = 2^{2T-3} - 1$ nonzero solutions and the fraction of errors which are not detected by V_{CC} and the MISR is equal to $(2^{2T-3} - 1)(2^{2T-1})^{-1} \simeq 2^{-3}$. \square

In the case when both space and time compression are used for compression of test responses $e = (e(0), \dots, e(T-1))$ ($e(t) = (e_0(t), \dots, e_{n-1}(t))$, $e_i(t) \in \{0, 1\}$) is not detected by monitoring the output of the CC for every t ($t = 0, 1, \dots, T-1$) and by verifying a space-time signature iff

$$h_0(H_{SC}e(0)) \oplus h_1(H_{SC}e(1)) \oplus \dots \oplus h_{T-1}(H_{SC}e(T-1)) = 0, \quad (15)$$

and $e(t) \in V_{CC}$ for any $t = 0, 1, \dots, T-1$.

If for an (n, k) code V_{CC} and an $(n, n-m)$ code V_{SC} we have $k \geq m$ (see e.g. Example 1) and $\eta = 2^{-m}$ (see (4)), then $\{z | z = H_{SC}y, y \in V_{CC}\} = GF(2^m)$ and for any (not necessarily primitive) MISR the fraction of errors which satisfy (15) is equal to 2^{-m} .

Thus, for the case of a BIST without space compression the fraction of errors

which are not detected by CC and BIST is equal to 2^{-n} for primitive MISRs and can go up to 2^{-k} for nonprimitive ones. For the case of a BIST with space compression this fraction is equal to 2^{-m} ($k \leq m$) for any (not necessarily primitive) MISR.

IV. Analysis of Probabilities of Error Detection for Combined BIST and Concurrent Checking.

We will estimate now probabilities of not detecting $e = (e(0), \dots, e(T-1)) \neq 0$ by CC only, by the signature analysis based on space-time compression of test responses and by CC combined with the signature analysis. We assume that the output of the concurrent checker is monitored for every $t = 0, 1, \dots, T-1$. As in [4], [18], [19] and [22] we assume that errors are independent in time and we will use the 2^n -ary symmetrical error model for the space distribution of errors:

$$\Pr\{e(t) = i\} = \begin{cases} 1 - p, & \text{for } i = 0; \\ p(2^n - 1)^{-1}, & \text{for } i \neq 0; \end{cases} \quad (16)$$

Then we have for the joint probability, P_{ON} , that e is not detected by CC and $e \neq 0$

$$\begin{aligned} P_{ON} &= (\Pr\{e(t) \in V_{CC}\})^T - \Pr\{e = 0\} \\ &= (1 - p + p(2^k - 1)(2^n - 1)^{-1})^T - (1 - p)^T. \end{aligned} \quad (17)$$

For small p we have from (17)

$$P_{ON} \simeq \exp(-pT(1 - 2^{-r})) - \exp(-pT), \quad (18)$$

where $r = n - k$.

For the probability, P_{SC} , that $H_{SC}e(t) = 0$ we have from (16)

$$P_{SC} = 1 - p + p(2^{n-m} - 1)(2^n - 1)^{-1} \simeq 1 - p(1 - 2^{-m}). \quad (19)$$

Then we have for the joint probability, P_{OFF} , that e is not detected by space-time signature analysis and $e \neq 0$

$$P_{OFF} = \Pr\{(H_{SC}e(0), \dots, H_{SC}e(T-1)) \in V_{TC}\} - \Pr\{e = 0\}. \quad (20)$$

Denote

$$P_{TC}(l) = \Pr\{v \in V_{TC} \mid \|v\| = l\}, \quad (21)$$

where $\|v\|$ is the number of nonzero components in $v = (v(0), \dots, v(T-1))$, $v(i) \in GF(2^m)$.

Then we have from (20) and (21)

$$P_{OFF} = \sum_{l=0}^T \binom{T}{l} (1 - p_{SC})^l p_{SC}^{T-l} P_{TC}(l) - (1 - p)^T. \quad (22)$$

It was shown in [4] that for any m -bit MISR

$$P_{TC}(l) = 2^{-m} (1 + (-1)^l (2^m - 1)^{-l+1}). \quad (23)$$

From (19), (22) and (23) we have for P_{OFF}

$$\begin{aligned}
 P_{OFF} &= 2^{-m} + (1 - 2^{-m}) \sum_{i=0}^T \binom{T}{i} ((1 - p_{SC})(-1)(2^m - 1)^{-1})^i p_{SC}^{T-i} - (1 - p)^T \\
 &= 2^{-m} + (1 - 2^{-m}) (p_{SC} - (1 - p_{SC})(2^m - 1)^{-1})^T - (1 - p)^T \\
 &= 2^{-m} + (1 - 2^{-m}) (1 - p(1 - (2^n - 1)^{-1}))^T - (1 - p)^T.
 \end{aligned} \tag{24}$$

We note that for the case when there is no space compression we have $m = n$ and from (24) we have the following well known result (see e.g. [4]) for the aliasing probability, P_{AL} , of a n -bit MISR

$$P_{AL} = P_{OFF} = 2^{-n} (1 - 2^n (1 - p)^T + (2^n - 1) (1 - p(1 - (2^n - 1)^{-1}))^T). \tag{25}$$

For small p we have from (24)

$$P_{OFF} \simeq 2^{-m} + (1 - 2^{-m}) \exp(-pT(1 - (2^n - 1)^{-1})) - \exp(-pT). \tag{26}$$

Let us estimate now the joint probability, $P_{ON,OFF}$, that e is not detected by CC, not detected by the signature analysis and $e \neq 0$.

As in the previous section we assume that $k \geq m$ and

$$\eta = |V_{SC} \cap V_{CC}| 2^{-k} = 2^{-m}. \tag{27}$$

Then we have for $P_{ON,OFF}$

$$P_{ON,OFF} = \Pr\{e(t) \in V_{CC} \text{ for all } t = 0, 1, \dots, T-1; H_{sce} \in V_{TC}\} - \Pr\{e = 0\}. \quad (28)$$

where $H_{sce} = (H_{sce}(0), \dots, H_{sce}(T-1))$.

Denote

$$a = \Pr\{e(t) \in V_{CC}, H_{sce}(t) = 0\}, \quad (29)$$

$$b = \Pr\{e(t) \in V_{CC}, H_{sce}(t) \neq 0\}.$$

Then in a view of (27) we have

$$a = 1 - p + p(2^{k-m} - 1)(2^n - 1)^{-1}, \quad (30)$$

$$b = p(2^k - 2^{k-m})(2^n - 1)^{-1},$$

and

$$P_{ON,OFF} = \sum_{l=0}^T \binom{T}{l} b^l a^{T-l} p_{TC}(l) - (1-p)^T, \quad (31)$$

where $P_{TC}(l)$ is defined by (21) and (23).

Finally, from (31), (23) and (30) we have

$$\begin{aligned} P_{ON,OFF} &= 2^{-m}(a+b)^T + (1-2^{-m})(a-b(2^m-1)^{-1})^T - (1-p)^T \\ &= 2^{-m}(1-p(1-(2^k-1)(2^n-1)^{-1}))^T + (1-2^{-m})(1-p(1-(2^n-1)^{-1}))^T \\ &\quad - (1-p)^T. \end{aligned} \quad (32)$$

We note that (32) generalizes both (17) and (25) and we have from (32)

$$\begin{aligned} P_{ON,OFF} &= P_{ON} \quad \text{for } m = 0, \text{ and} \\ P_{ON,OFF} &= P_{OFF} \quad \text{for } k = m. \end{aligned} \tag{33}$$

For small p one can use the following approximation for $P_{ON,OFF}$

$$P_{ON,OFF} \simeq \exp(-pT(1 - 2^{-r})) + (1 - 2^{-m}) \exp(-pT(1 - (2^n - 1)^{-1})) - \exp(-pT). \tag{34}$$

Probabilities of not detecting an error as functions of test length T for concurrent checking, signature analysis and combined concurrent checking and signature analysis are presented in Fig. 7 for $n = m = 32$, $r = 1$ and $p = 10^{-4}$.

Example 9. Let us consider the control ROM from Example 1 with $n = 123$, $k = 116$, $r = 7$ and $m = 28$. For $p = 10^{-5}$ and $T = 2^{15}$ we have from (18), (26) and (34), $P_{ON} = 1.85 \times 10^{-3}$, $P_{OFF} = 1.04 \times 10^{-9}$ and $P_{ON,OFF} = 6.88 \times 10^{-12}$. Probabilities of not detecting an error as functions of test length T for concurrent checking, signature analysis and combined concurrent checking and signature analysis for this ROM are presented in Fig. 8 for $p = 10^{-4}$. \square

V. Concurrent Checking of Space and Time Compressors in the Off-line Testing Mode

Let us consider now the problem of concurrent checking of SCs and TCs to detect permanent and intermittent errors in the process of off-line testing.

A self-testing error propagating SC is a set of m XOR trees with input $y(t) = (y_1(t), \dots, y_n(t))$ and output $z(t) = (z_1(t), \dots, z_m(t))$ ($m \leq n$), where

$$z(t) = y(t)H_{SC}^{tr}, \quad (35)$$

and H_{SC}^{tr} is the transposed check matrix of a linear $(n, n - m)$ code V_{SC} selected for the space compression [1], [2]. A TC (MISR with internal XORs, MISR with external XORs, cellular automaton) with input $z(t)$ and internal state $D(t)$ can be described by the following characteristic equation

$$D(t+1) = D(t)A \oplus z(t+1), \quad (36)$$

where $D(t)$, $z(t)$ are m -bit binary vectors corresponding to internal states and inputs of TC and A is a $(m \times m)$ non-singular over $GF(2)$ binary transition matrix.

Suppose a linear (M, m) code, V_{STC} , has been chosen for concurrent checking of the SC and TC (Fig. 3). The generating matrix of this code can be represented as $G = [I_m | Q]$, where I is the $(m \times m)$ identity matrix, and Q is the $(m \times (M - m))$ parity matrix. Then $(v_1, \dots, v_m, v_{m+1}, \dots, v_M) \in V_{STC}$ iff

$$R(v) = (v_{m+1}, \dots, v_M) = (v_1, \dots, v_m)Q. \quad (37)$$

From (35) and (37), we have for the parity prediction part, $R(SC)$, of the SC

$$R(z(t)) = (z_{m+1}(t), \dots, z_M(t)) = y(t)H_{SC}^{tr}Q = y(t)H', \text{ where } H' = H_{SC}^{tr}Q, \quad (38)$$

H' is a $(m \times (M - m))$ binary matrix, and all the computations are mod 2.

For the parity prediction part $R(\text{TC})$ of the TC, we have from (15) and (16)

$$R(D(t+1)) = D(t)AQ \oplus z(t+1)Q = D(t)A' \oplus z(t+1)Q = D(t)A' \oplus y(t+1)H', \quad (39)$$

where A' is a $(m \times (M - m))$ binary matrix, $A' = AQ$.

For the case $M - m = 1$, V_{STC} is the $(m + 1, m, 2)$ parity code, Q is the column of all ones, H' is the column which is equal to the sum mod 2 of all columns of H_{SC}^{tr} , and A' is the column which is equal to the sum of all columns of A .

Let us consider the case of time compression by a m -bit MISR with external XORs. In this case [3],

$$A = \begin{bmatrix} p_1 & 1 & 0 & 0 & \cdots & 0 \\ p_2 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{m-1} & 1 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad p_i \in \{0, 1\} (i = 1, \dots, m-1), \quad (40)$$

and for CC by parity prediction $(m + 1, m, 2)$ code we have $Q = \overbrace{(11 \dots 1)}^m$, $A' =$

$AQ = (\bar{p}_1, \bar{p}_2, \dots, \bar{p}_{m-1}, 1)$ (\bar{p}_i is negation of p_i) and

$$R(D(t+1)) = D_{m+1}(t+1) = D_1(t)p_1 \oplus D_2(t)p_2 \oplus \cdots \oplus D_{m-1}(t)p_{m-1} \oplus z_{m+1}(t+1) \quad (41)$$

The general block diagram of a space-time compressor with the parity prediction is

VLSI / 002 / 26

represented in Fig. 9.

Fig 9

Example 10. Let us consider CC for space-time compression of test responses for a device with $n = 7$ outputs. Suppose that the (7,4) Hamming code with check matrix

$$H_{SC} = A = \begin{bmatrix} 0001111 \\ 0110011 \\ 1010101 \end{bmatrix},$$

is selected for space compression and the 3-bit MISR with external XORs and primitive polynomial $x^3 \oplus x \oplus 1$ is selected for time compression. In this case $H' = (1101001)^{tr}$ and the corresponding space-time compressor with parity checking, based on the (4,3,2), code V_{STC} is given in Fig. 10. \square

Fig 10

Example 11. Let us consider a concurrent checker for the SC and the TC for the case of off-line testing the control ROM for MC68020 from Example 1. In this case, $k = 116$, $n = 123$; space compression can be implemented by the (123,95,9) code ($m = 28$) and time compression by the MISR with primitive generating polynomial $x^{28} \oplus x^3 \oplus 1$. Then the CC for the SC and the TC can be implemented by the (29,28,2) parity code. Additional overhead required for this CC (in terms of equivalent two-input gates) is less than 5%. \square

VI. Conclusions

In this paper we proposed an approach for combining an on-line concurrent checking and off-line BIST based on space-time compression of test responses to maximize

probabilities of error detection for both manufacturing and field testing. An approach for optimal selection of error-detecting codes for concurrent checking and space-time compression of test data have been developed, and probabilities of error detection for combined on-line and off-line techniques have been estimated. An approach for concurrent checking of space and time compressors for test responses was proposed. The presented techniques can be used for design of fault-tolerant devices with BIST.

Acknowledgment

The author thanks Dr. L. B. Levitin, Mr. S. M. Chaudhry and Mr. M. J. Ryniejski from the Research Laboratory on Design and Testing of Computer Hardware, Boston University, for many useful discussions related to the subject of this paper.

References

- [1] Suluja, K. K. and M. G. Karpovsky, "Testing Computer Hardware Through Data Compression in Space and Time," Proc. International Test Conf., 1983, pp. 1194-1198.
- [2] Reddy S. M., K. K. Suluja and M. G. Karpovsky, "A Data Compression for Built-In Self Test," IEEE Trans. on Computers, Sept 1988, pp. 1151-1156.
- [3] Bardell, P. H., McAnney, W. H. and Savir, "Built-In Test for VLSI: Pseudorandom Techniques," John Wiley & Sons, 1987.

- [4] D.K. Pradhan, S.K. Gupta and M.G. Karpovsky "Aliasing Probability for Multiple Input Signature Analyzer and A New Compression Technique," IEEE Trans. on Computer, April, 1990.
- [5] M. G. Karpovsky, S. Chaudhry, "A Design of Self-Diagnostic Boards by Multiple Signature Analysis," IEEE Trans. on Computer, to appear, 1992.
- [6] M. G. Karpovsky, S. Chaudhry, L. B. Levitin "Multiple Signature Analysis: A Framework for Built-In Self-Diagnostic," Proc. Fault-Tolerant Computing Symp., 1992.
- [7] D. K. Pradhan (ed.), "Fault Tolerant Computing: Theory and Techniques," Prentice Hall, NJ, 1986.
- [8] N. K. Jha, "A Totally Self-Checking Checker For Borden's Code," IEEE Trans. on CAD, July 1989, pp. 731-736.
- [9] N. K. Jha and J. A. Abraham, "The Design of Totally Self-Checking Embedded Checkers," Proc. FTCS, June 1984, pp. 265-270.
- [10] S. M. Reddy, "A Note On Self-Checking Checkers," IEEE Trans. on Computers, Vol. C-19, No. 11, November 1970, pp. 1035-1038.
- [11] F. J. MacWilliams and N. J. A. Sloane, The Theory of Error-Correcting Codes, New York: North-Holland, 1977.
- [12] K. K. Saluja, R. Sharme, C. K. Kime, "A Concurrent Testing Technique for Digital Circuits," IEEE Trans. on CAD, Vol. 7, No. 12, December 1988.

- [13] R. Sharme, K. K. Saluja, "An Implementation and Analysis of a Concurrent BIST Technique," Int. Symp. on Fault Tolerant Computing, June 1988, pp. 164-169.
- [14] Nicolaidis, "Self-Exercising Checkers for Unified Built-In Self-Test (UBIST)," IEEE Trans. on CAD, March 1989, pp. 203-218.
- [15] M. Katoozi, A. Nordsieck, "Low Overhead Built-In Testable Error Detection and Correction With Excellent Fault Coverage," Proc. Int. Test Conf., 1981, pp. 916-924.
- [16] S. Park, S. Akers, "Parity Bit Calculation and Test Signal Compaction for BIST Application," Proc. Int. Test Conf., 1981, pp. 1016-1023.
- [17] Karpovsky, M. G., D. K. Pradhan, and S. K. Gupta, "Aliasing and Diagnosis Probability in MISRs and STUMPS Using a General Error Model," Proc. Int. Test Conf., 1991, pp. 828-840.
- [18] Gupta, S. K. and D. K. Pradhan, "A New Framework for Designing and Analyzing BIST Techniques: Computation of Aliasing Probability," Proc. Int. Test Conf., 1988.
- [19] Williams, T. W., W. Dahn, M. Gruetzner, C. W. Starke, "Bounds and Analysis of Aliasing Errors in LFSRs," IEEE Trans. CAD/ICAS Vol. 7, No. 1, January 1988, pp. 75-83.
- [20] Iwasaki, K and N. Yameguchi, "Design of Signature Circuits Based on Weight Distributions of Error Correcting Codes," Proc. ITC 1990, pp. 779-785.

- [21] Pradhan D. K., S. K. Gupta, "A Framework for Designing and Analyzing New BIST Techniques and Zero Aliasing Compression," *IEEE Trans. on Computers*, Vol. 40, No. 6, June 1991, pp. 743-763.
- [22] Sedmak, R. M., "Design for Self-Verification: An Approach for Dealing with Testability Problems in VLSI-Design," *Proc. Int. Test Conf.*, 1979, pp. 112-124.
- [23] Sedmak, R. M., "Implementation Techniques for Self-Verification," *Proc. Int. Test Conf.*, 1980, pp. 267-278.
- [24] Lubaszewski, M., B. Courtois, "On the Design of Self-Checking Boundary Scannable Boards," *Proc. Int. Test Conf.*, 1992, pp. 372-381.
- [25] Sun, X., M. Serra, "Merging Concurrent Checking and Off-Line BIST," *Proc. Int. Test Conf.*, 1992, pp. 958-968.
- [26] Gupta, S. and D. K. Pradhan, "Can Concurrent Checkers Help BIST?," *Proc. Int. Test Conf.*, 1992, pp. 140-150.

Figure 1: Block Diagram for Built-In Self-Test

Figure 2: Block Diagram for On-line Error Detection

Figure 3: Block Diagram for Combined On-Line and Off-Line Error Detection Mechanisms

Figure 4: Recursive Construction for H_m Networks ($m = 2, 3 \dots$)

Figure 5: Implementation for H_4 With Concurrent Checking

Figure 6: Recursive Construction for Q_m Networks ($m = 2, 3 \dots$)

Figure 7: Probabilities Of Not Detecting An Error As Functions Of Test Lengths T for Concurrent Checking P_{ON} , Signature Analysis P_{OFF} And Combined Concurrent Checking And Signature Analysis $P_{ON,OFF}$ For $n = m = 32, r = 1, p = 10^{-4}$

Figure 8: Probabilities Of Not Detecting An Error As Functions Of Test Lengths T for Concurrent Checking P_{ON} , Signature Analysis P_{OFF} And Combined Concurrent Checking And Signature Analysis $P_{ON,OFF}$ For The Control ROM From Example 1 ($n = 123, k = 116, m = 28$) For $p = 10^{-4}$

Figure 9: Space-Time Compressor with Concurrent Checking by Parity Prediction

Figure 10: Self-Error-Detecting Space-Time Compressor with the (7, 4, 3) Hamming Code for Space Compression $P(x) = 1 \oplus x \oplus x^3$ for Time Compression and the (4, 3, 2) Parity Code for Concurrent Checking of Space and Time Compressors

Table 1: Representation of $GF(2^3)$ with $P(x) = x^3 \oplus x \oplus 1, P(\alpha) = 0.$

	1	α	α^2	α^3	α^4	α^5	α^6
0	1	0	0	1	0	1	1
0	0	1	0	1	1	1	0
0	0	0	1	0	1	1	1