Embedded Test Resource for SoC to Reduce Required Tester Channels Based on Advanced Convolutional Codes

Yinhe Han, *Member, IEEE*, Xiaowei Li, *Senior Member, IEEE*, Huawei Li, *Member, IEEE*, and Anshuman Chandra, *Member, IEEE*

***Abstract*—Test resources can be embedded on the chip to reduce required external tester channels. In order to obtain the maximal reduction of tester channels, a single-output encoder based on the check matrix of the $(n, n-1, m, 3)$ convolutional code is presented. When the five proposed theorems are satisfied, the encoder can avoid two and any odd erroneous bit cancellations, handle one unknown bit (X-bit), and diagnose one erroneous bit. Two types of encoders are proposed to implement the check matrix of the convolutional code. A large number of X-bits can be tolerated by choosing a proper memory size and weight of the check matrix, which can also be obtained by an optimized input assignment algorithm. In order to get the full diagnostic capability, the proposed encoder can be reconfigured into a simple linear-code-based encoder by adding some additional gates. Experimental results show that the proposed encoder has an acceptable level of X-bits tolerance and a low aliasing probability.**

***Index Terms*—Automatic test equipment, convolutional code, diagnosis, error cancellation, masking, unknown bits (X-bits).**

I. Introduction

TESTING a digital circuit accounts for a large part of the cost to design, manufacture, and service an electric system—a trend that is projected to continue and accelerate [1]. With the increasing device complexity and performance requirements, the "big-iron" test approach requires even more expensive high-performance testers with expensive test resources, such as a large number of test channels, a large volume of vector memory, a high frequency, etc. These testers, which cost more than $3 500 000 apiece for a 512-pin 400-MHz version, represent a large and significant contribution to the overall cost of a chip. Embedded test resources can help mitigate the need for more complex testers. A scan-based circuit demands a large number of inputs and outputs to transfer test patterns. In the conventional way, a tester with a high bandwidth is required. However, this problem can be solved through an embedded test resource, an on-chip test response encoder, which is inserted between the scan chains and the external pins as shown in Fig. 1. In this circuit, the large-scale scan-out information (high bandwidth) is encoded into a small-scale code (low bandwidth) by the encoder. The encoded code words are stored in memory buffers of automatic test equipment (ATE) and compared during online testing. Thus, only a small number of tester channels are needed to monitor the core during testing.

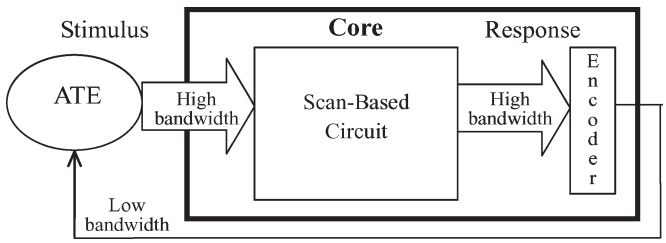

Fig. 1. Encoder design for single-core test response compaction.

In the multicore system, the test-access-mechanism (TAM) width is an important resource to optimize the test time. In the conventional design, the input TAM width is equal to the output TAM width. However, if an encoder is embedded to compact test responses as shown in Fig. 2(a) and (b), the output TAM width can be reduced. Thus, the fewer channels are required to test the system. It is obvious that an embedded resource enables the use of lower performance testers to test a complex system-on-a-chip (SoC).

The encoder designed as on-chip test resource is a very-well-researched topic, and a number of techniques have been presented in literature [2]–[16]. These techniques can be divided into two categories: circuit-function specific and circuit-function independent. The techniques based on the first category are discussed in [3]–[5]. The techniques based on the second category use encoders based on the coding theory. The technique presented in [6] first constructs a linear-code encoder and discusses the bounds of some parameters for the proposed encoder. An encoder (X-Compact) that can correct one error based on the Hamming code is presented in [7] and is then extended to detect any number of odd errors. The encoder presented in [8] and [9] is based on linear block codes to tolerate some unknown bits (X-bits) in the test response. The encoders in [7]–[9] are all implemented using XOR gates. A sequential encoder to implement a linear block code is presented in [10]. The schemes proposed in [11] and [12] present an encoder based on the parity check matrix of a convolutional code. Several issues on aliasing probability, X-masking, and

Manuscript received August 12, 2004; revised August 28, 2005. This paper was supported in part by the National Basic Research Program of China (973) under Grant No. 2005CB321605 and in part by the National Natural Science Foundation of China under Grants 90207002 and 60576031.

Y. Han, X. Li, and H. Li are with the Advanced Test Technology Laboratory, Institute of Computing Technology, Chinese Academy of Sciences and Graduate University of the Chinese Academy of Sciences, Beijing, China (e-mail: yinhes@ict.ac.cn; lxw@ict.ac.cn; lihuawei@ict.ac.cn).

A. Chandra is with the Synopsys Inc., Mountain View, CA 94043 USA (e-mail: anshuman@synopsys.com).

Digital Object Identifier 10.1109/TIM.2006.870332

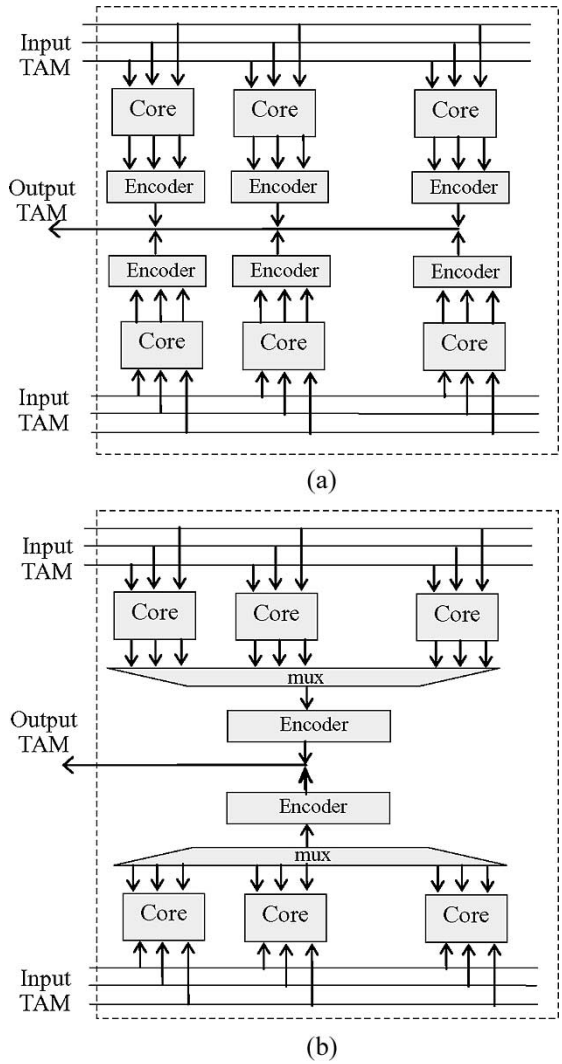

Fig. 2. Encoder designs for TAM width reduction in SoC. (a) Encoder per core. (b) Shared encoder.

diagnosis of the convolutional compactor [11] and the quotient (q)-compactor [12] are discussed.

In this paper, we will develop and extend the study presented in [12], [31], and [32]. This paper can be considered as an extension of the convolutional compactor in [12] by relating it with the convolutional-code theory and the linear-code matrix-based encoder [6]. We conducted a study on the behavior of defects in an actual chip, and the results are reported in Fig. 3. From Fig. 3, we observe that the numbers of erroneous bits (86.8%) in major patterns are less than 4. No pattern produced ten or more errors. This study shows that the proposed encoder, which provides zero-aliasing when the number of errors is small, is very efficient since the number of errors is small in practical test cases.

The rest of this paper is organized as follows. Section II introduces the background of convolutional codes. Section III presents some design theorems to construct the single-output encoder. The two styles of implementations of the proposed encoder are presented in Section IV. In Section V, the theoretical analysis and an input assignment algorithm are presented to handle a large number of X-bits. Section VI reconfigures

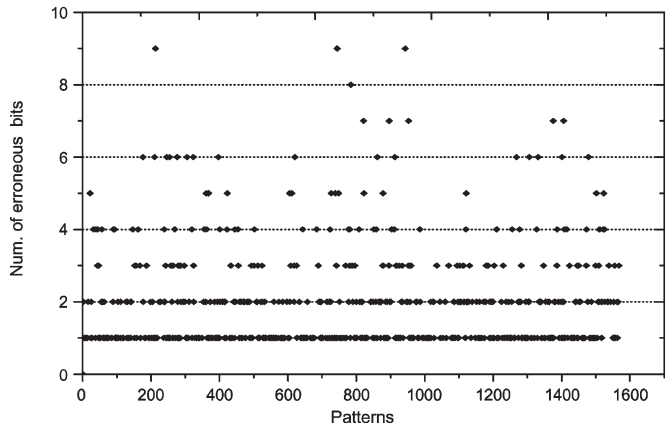

Fig. 3. Distribution of erroneous bits in the test response of defects in an industrial chip.

the convolutional code H_m-based encoder to a linear-code-based encoder to achieve the enhanced diagnosis capability. Experimental results on the aliasing probability are reported in Section VII.

II. Background of Convolutional Codes

The convolutional code was first introduced by Elias [17] in 1955 and is widely applied today in telecommunication systems, e.g., radio, satellite links, and mobile communication [21] and [24].

We review the convolutional code and its related definitions first. A convolutional code can be denoted as (n, k, m, d), where n is the number of parallel output information bits, k is the number of parallel input information bits at one time interval (cycle), m is the maximum number of memory elements in the encoder, and d is the minimum distance between code words. The distance between two code words is the Hamming distance. The minimum distance d is the minimum Hamming distance between all pairs of complete convolutional code words and can be calculated as

$$\begin{aligned} d &= \min\{d(y1, y2) | y1 \neq y2, \text{ and } y1, y2 \text{ are code words}\} \\ &= \min\{w(y) | y \neq 0, \ y \text{ is a code word}\} \end{aligned}$$

where $d(\bullet, \bullet)$ is the Hamming distance between a pair of convolutional code words, and $w(\bullet)$ is the weight of a convolutional code word, which is equal to the number of ones (1's) in it.

The convolutional code can be characterized by an arbitrarily large generator matrix G_∞. Each k-input information during ∞ cycles can be mapped into ∞ k-tuple polynomials: $I = \{I_0(x), I_1(x), \ldots, I_\infty(x)\}$, and $I_i(x)$ is a k-degree polynomial. Then, the "code word" $C = \{C_0(x), C_1(x), \ldots, C_\infty(x)\}$, where $C_i(x)$ is an n-tuple polynomial, is defined as

$$C = I \bullet G_\infty.$$

The dot in this formulation denotes vector–matrix multiplication.

An example of generator matrix G_∞ is shown in Fig. 4. In G_∞, each entry g_i is a (k, n) submatrix.

We mentioned that the degree of generator matrix G_∞ can be arbitrarily large in principle if the degree of the input

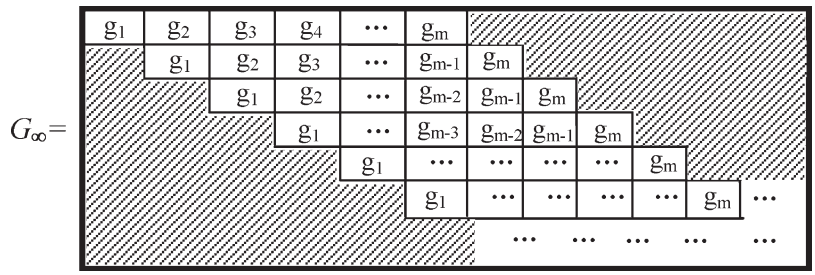

Fig. 4. Polynomial generator matrix of a convolutional code (shaded area = all 0's).

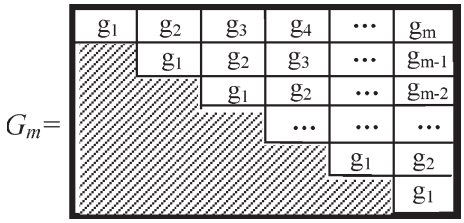

Fig. 5. Truncation matrix G_m of G_∞.

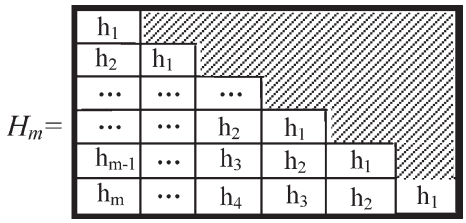

Fig. 6. Check matrix H_m of G_m.

polynomial is arbitrarily large. However, in any constrained-length m cycles, the relation of affected code words are the same. This leads us to define an m-truncation of a polynomial matrix. The inside information polynomial $I_m = \{I_0(x), I_1(x), \ldots, I_m(x)\}$ contains $m \times k$ binary bits, and the code word C_m contains $m \times n$ bits. The encoding mapping from I to C can be represented as

$$C_m = I_m \bullet G_m$$

where the submatrix G_m is a truncation of the generator matrix of G_∞. G_m is described in Fig. 5. It is an $(m \times k,\ m \times n)$ matrix, where $m \times k$ is the number of rows and $m \times n$ is the number of columns.

The check matrix H_∞ of a convolutional code is defined as $G_\infty \bullet H_\infty^{\mathrm{T}} = 0$. If the m-truncation generator matrix is used, the corresponding truncation check matrix is $G_m \bullet H_m^{\mathrm{T}} = 0$, where H_m^{T} is an $(m \times (n-k),\ m \times n)$ matrix. In H_m (Fig. 6), each entry h_i is an $(n-k, n)$ matrix. H_m is determined by the first n columns since the other columns can be looked at as a shift version of the first n columns. We define the matrix $H = \{h_1^{\mathrm{T}}, h_2^{\mathrm{T}}, h_3^{\mathrm{T}}, \ldots, h_m^{\mathrm{T}}\}^{\mathrm{T}}$ as the basic check matrix.

III. Construction of an Advanced $(n, n-1, m, d)$ Convolutional Code

As explained in the previous section, the convolutional code can be looked at as a linear code of which the generator matrix is G_m; therefore, some design theorems of other linear codes can be used to construct the convolutional code. The $(2m,\ 2m-1,\ m)$ WA code [25] is presented when referring to the construction procedure of a Hamming code. A WA code can correct one error. Massey *et al.* [18] and Justesen [20] construct some convolutional codes based on design theorems of cyclic codes and Reed–Solomon codes. Recently, relations between convolutional and MDS codes [23] were used to construct convolutional codes.

Saluja and Karpovsky [6], Mitra and Kim [7], and Patel *et al.* [8] give some theorems on the relations between the linear codes and H-based encoders. We review these theorems first.

Lemma 1 (Saluja and Karpovsky [6]): If H is a check matrix of $(n,\ n-k,\ d)$ linear codes, the implementation of H (also denoted as an H-based encoder) can be used as a response encoder. No test responses closer than distance d can produce the same check words by this encoder. This encoder has n inputs and k outputs.

Lemma 2 (Patel et al. [8] and Mitra and Kim [7]): To a linear code $(n,\ n-k,\ d)$, $H(k \times n)$ is the check matrix of this code. The following relations between errors detectability, X-masking, and diagnose capability of H-based encoders exist.

1) H-based encoder can detect up to $d-1$ erroneous bits in the input information.
2) H-based encoder can detect up to e errors in the presence of u unknown bits, where $e + u \leq d - 1$.
3) H-based encoder can diagnose up to e errors in the presence of u unknown bits, where $2e + u \leq d - 1$.
4) H-based encoder can detect any odd errors if the weight of each row in H is odd [7].

Lemma 3 [22]: The linear code $(n,\ n-k,\ d)$ has the minimum distance d iff any $d-1$ columns in the check matrix $H(k \times n)$ are linearly independent.

Lemma 4 [22]: The convolutional code $(n,\ k,\ m,\ d)$ can be looked at as a linear code of which the generator matrix is G_m, and the check matrix is H_m.

It is mentioned that 2) and 3) in Lemma 2 are true only when the off-line compare is used, which is presented in Patel *et al.* [8]. Since the convolutional code is a linear code from Lemma 4, Lemmas 1–3 can be directly applied to the construction of the encoder based on the convolutional code.

In this paper, we aim to present a single-output encoder, which means that we need to construct an $(n,\ n-1,\ m,\ d)$ convolutional code. Some theorems to construct such codes

have been presented in [18]–[20], and [25]. The ($2m$, $2m-1$, m) WA code [25] is presented, referring to the design of a Hamming code. Justesen [20] presents a code with the rate 4/5 in which the distance can reach 6. Other details are also discussed in [18] and [19]. However, the design and implementation of these convolutional codes are complicated. At the same time, a large d is not necessary since the erroneous bits in the response are often less than four. The distance of 3 is sufficient. In the following paragraphs, we will present a simple convolutional code of which the Hamming distance is 3.

Some definitions are introduced first.

Definition 1: To two m-tuple polynomials $R_1(x)$ and $R_2(x)$ with the relation

$$R_1(x) \cong R_2(x) \text{ iff}$$

1) $\exists\sigma | R_1(x) = x^\sigma \times R_2(x),\ (0 \le \sigma \le m-1-\deg(R_2(x))$ or
2) $\exists\sigma | R_2(x) = x^\sigma \times R_1(x),\ (0 \le \sigma \le m-1-\deg(R_1(x))$

is an equivalent relation in GF(2), and where $\deg(\bullet)$ is the maximum degree of nonzero coefficient in a polynomial.

Definition 2: "Equivalent class" and "characteristic polynomial." We define a set as an equivalent class in which each pair of polynomials is equivalent. If all equivalent polynomials are contained in an equivalent class, there is one and only one polynomial whose maximum degree of nonzero coefficient is $m-1$. This $(m-1)$-degree polynomial is defined as a characteristic polynomial of this equivalent class.

Based on the definitions of equivalent class and characteristic polynomial, we derive the following corollary.

Corollary 1: The complete set of m-tuple polynomials with the weight ω contains C_m^ω elements; the number of equivalent classes in this set is $C_{m-1}^{\omega-1}$.

Proof: The first conclusion can be directly deduced from the combination theory. To the second conclusion, the number of equivalent classes is equal to the number of characteristic polynomials. Since the weight of a characteristic polynomial is ω besides the fixed nonzero coefficient in the $(m-1)$th degree, there are other $\omega-1$ nonzero coefficients, and they are needed to be located. The degrees corresponding to these nonzero coefficients can be looked at as $(\omega-1)$ combinations from $\{0, 1, \ldots, m-2\}$, and their number is $C_{m-1}^{\omega-1}$. ■

Theorem 1: To the convolutional code (n, $n-1$, m, d), in the basic check matrix H, if no two column polynomials are equivalent, then the minimum distance of this code is $d \ge 3$.

Proof: In the (n, $n-1$, m, d) code, the basic check matrix H can be represented by a set of n column polynomials: $H = \{C_1(x), C_2(x), C_3(x), \ldots, C_n(x)\}$. $C_i(x)$ is an $(m-1)$-degree polynomial. Then, the check matrix of this code is $H_\mathrm{m} = \{C_1(x), C_2(x), C_3(x), \ldots, C_n(x), xC_1(x), xC_2(x), \ldots, xC_n(x), x^2C_1(x), x^2C_2(x), \ldots, x^2C_n(x), \ldots, x^{m-1}C_1(x), x^{m-1}C_2(x), \ldots, x^{m-1}C_n(x)\}$. The product $x^iC_j(x)$ should be a polynomial with a maximum degree $(m-1)$. The degree higher than $(m-1)$ must be truncated. From the definition of equivalent class, if there are no equivalent polynomials in $\{C_1(x), C_2(x), C_3(x), \ldots, C_n(x)\}$, then there are no two the same column polynomials in H_m. Thus, any two columns in H_m are linearly independent. From the Lemma 3, the distance of the code will be larger than 2. ■

Corollary 2: If the check matrix H_m is constructed based on Theorem 1, the memory size m must be bounded as $m \ge \log_2(n) + 1$.

Proof: The weights of nonequivalent column polynomials can be varied from 1 to m. Based on Corollary 1, the total number of nonequivalent polynomials is bounded as

$$\begin{aligned} n &\le 1 + C_{m-1}^1 + C_{m-1}^2 + C_{m-1}^3 + C_{m-1}^4 + \ldots + C_{m-1}^{m-1} \\ &= 2^{m-1} \\ \therefore\ m &\ge \log_2(n) + 1. \end{aligned}$$

■

Theorem 2: To the convolutional code (n, $n-1$, m, d), if the basic check matrix H is constructed based on Theorem 1 and the weight of each column polynomial is odd, then the H_m-based encoder can guarantee to detect two and any odd errors in the test response within m cycles.

Proof: Theorem 2 can be obtained from the 1) and 4) of Lemma 2. ■

Corollary 3: If the check matrix H_m is constructed based on Theorem 2, the memory size m must be bounded as

$$m \ge \log_2(n) + 2.$$

Proof: An odd number can be selected from 1 to m as the weight of a column polynomial. Thus, the total number of nonequivalent polynomials with odd weights is bounded as

$$n \le 1 + C_{m-1}^2 + C_{m-1}^4 + C_{m-1}^6 + \ldots + C_{m-1}^{m-1}\ (m \text{ is odd})$$

or

$$n \le 1 + C_{m-1}^2 + C_{m-1}^4 + C_{m-1}^6 + \ldots + C_{m-1}^{m-2}\ (m \text{ is even}).$$

Then

$$\begin{aligned} n &\le 2^{m-2} \\ \therefore\ m &\ge \log_2(n) + 2. \end{aligned}$$

■

Corollary 4: If the check matrix H_m is constructed based on Theorem 2 and each column polynomial has a same weight ω, then the bound on the maximum number of inputs of H_m-based encoder is

$$n = C_{m-1}^{\omega-1}.$$

Proof: It can be obtained from Corollary 1. ■

Theorem 3: One error with an X-bit produced in the test response is guaranteed to be detected by the H_m-based encoder if Theorem 1 is satisfied and the weight of each column polynomial is equal.

Proof: It can be obtained from 2) of Lemma 2. ■

Theorem 4: The H_m-based encoder can locate one error from the output slices of scan chains if Theorem 1 is satisfied.

Proof: It can be obtained from 3) of Lemma 2. ■

Theorems 2–4 can be looked as extensions of the three properties presented in [11]. They are also the extensions of our study presented in [12]. Although Theorems 3 and 4 provide the capacity of handling the X-bits and diagnosis, they are

$$H=\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Fig. 7. Example of the basic check matrix H.

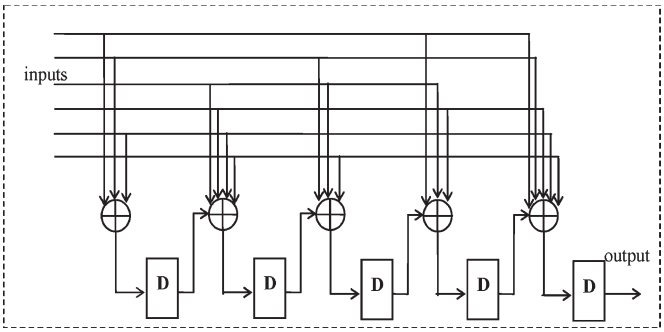

Fig. 8. Type I nonintrusive implementation of an encoder.

inefficient for practical cases and need to be extended. All of these will be discussed in the following sections of this paper.

IV. Implementations of H_m-Based Encoder

The implementation of check matrix H_m of an $(n,\ n-1,\ m,\ d)$ convolutional code can be constructed by using modulo-2 adders and shift registers. Two styles of encoders can implement this check matrix. The type-I encoder is nonintrusive in which the inputs of the encoder are only connected to the outputs of scan chains. The type-II encoder is intrusive, where the inputs of the encoder must be connected to several internal-scan-cell outputs of the scan chains. The terms "intrusive" and "nonintrusive" indicate whether the encoder is connected to the internal scan cells or not. An "intrusive" decoder is connected to the internal scan cells and a "nonintrusive" decoder is independent of the internal scan cells. The basic check matrix H of these two encoders is shown in Fig. 7. Figs. 8 and 9 show examples of intrusive and nonintrusive styles.

Fig. 8 shows the nonintrusive implementation of H_m. It consists of some XOR gates and registers. The XOR gates compose an XOR tree. An XOR tree is synthesized based on m H. Each row in H represents an output in the XOR tree, and each column represents an input. In type I, only the outputs of scan chains are connected to the encoder. It is suitable for the core-based design since it does not need to depend on the internal DFT structures of the third-part cores.

Fig. 9 shows the intrusive implementation of an encoder. The inputs of the encoder are connected to the last m stages of scan chains. The connection of m stages of scan cells in a scan chain corresponds to a column in H. Because this implementation reuses the memory elements of scan chains, it is a linear combinational compactor.

The advantages of type I are obvious to the integrators. It does not require a redesign of the circuit under test, which is the case with the IP hard cores. The encoder can be inserted

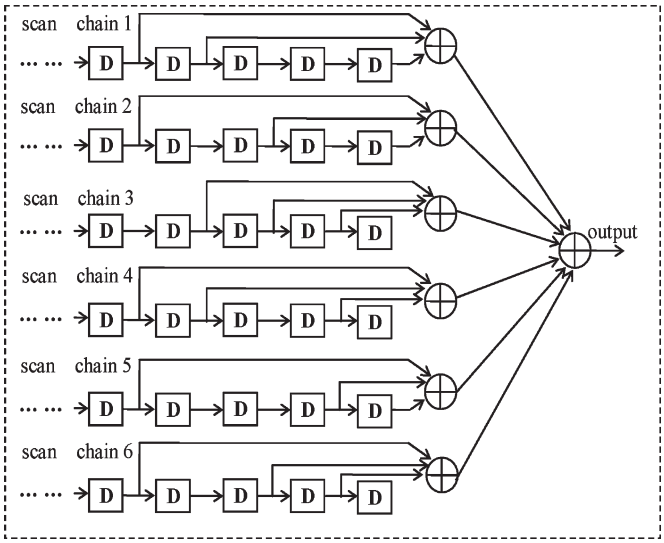

Fig. 9. Type II intrusive implementation of an encoder.

as a separate logic core. It can be easily integrated into the test environment as shown in Fig. 2. The type-II encoder can be embedded in the core as shown in Fig. 1. Since the type II encoder uses only a small number of XOR gates, the area overhead is small, and some other merits, such as smaller delay and better wire placement, can be obtained.

Area Overhead: We use a program written by C language to automatically generate the Verilog netlist of the proposed encoder. These Verilog netlists are used to evaluate the area overhead. To a 200-input Type-I encoder with $r = 5$, when k is 30, the area is 2050 equivalent NAND gates; when k is 100, the area is 3100 equivalent NAND gates. As the technology feature size plunged to the nanometer range, the concern about the area overhead has been eased. Currently, allocating 10 000 gates to the proposed encoder has become acceptable.

V. Design for X-Bits Tolerance

There are some undetermined states in the scan chains of which the outputs are not known during the simulation. These are also called unknown bits or X-bits (X's) [7], [14], [28]. Typical potential X generators include nonscan flip-flops, RAMs and central access memories, combinational loops, undriven primary inputs, bus contention, and violation on a wired gate. In a discrete Fourier transform (DFT), potential X generators can be identified by a design rule checker and may be fixed by some additional logic. However, the area and delay overhead of fixing a logic circuit is large. Thus, an encoder with high X-bits tolerance is very attractive.

Theorem 3 guarantees to handle one X-bit. However, this is not sufficient for real-life circuits. Finding an efficient convolutional code can help us improve this capability. However, as shown in [8], it is not always necessary. This paper proposes to "tolerate X-bits" rather than "eliminate X-bits." This means that we can improve the detection probability of errors with X-masking through other simple methods. The detection probability P is proposed in [8], and we redefine it as the detection probability of one error in the presence of X-bits.

An analysis of the detection probability of an H_m-based encoder using a stochastic theory is presented in the following paragraph. The check matrix H_m of $(n,\ n-1,\ m,\ d)$ is an $(m,\ m \times n)$ matrix. The weight of each column polynomial is the same and is equal to ω. We assume that each entry in H_m assigned as 1 is of an equal probability and is equal to $P_1 = \omega/m$ (though 1's are not randomly distributed when they are constrained by Theorems 1 and 2). The input information (response data of scan chains) is I, the check word of the output is C, and $C = H_m \times I^{\mathrm{T}}$, where I is an $m \times n$-bit word and C is a m-bit check word. One error is injected in the first n bits of I. Assuming the density of X-bits in the response data is P_{X}, if the X-bits are randomly distributed, then P_{X} is the probability of one bit assigned as X in the response data. We use $\overline{P}_j$ to denote the undetection probability of injected error only through observing the jth bit C_j in C. C_j can be calculated as

$$C_j = [h_1, \ldots, h_j, 0, \ldots, 0] \times I^{\mathrm{T}}$$

where $[h_1, \ldots, h_j, 0, \ldots, 0]$ is the jth row in H_m. Thus, only the X-bits that locate in the first $j^*n - 1$ bits can mask an injected error in the calculation of C_j. The $\overline{P}_j$ can be calculated as

$$\overline{P}_j = P_1 \times \left\{1 - [P_{\mathrm{X}} \times (1 - P_1) + (1 - P_{\mathrm{X}})]^{j \times n - 1}\right\} + (1 - P_1).$$

It can be reduced as

$$\overline{P}_j = 1 - P_1 \times (1 - P_1 \times P_{\mathrm{X}})^{j \times n - 1}.$$

Thus, the probability of an error when it is not detected through observing all m bits of the check word is

$$\overline{P} = \prod_{j=1}^{m} \left[1 - P_1 \times (1 - P_1 \times P_{\mathrm{X}})^{j \times n - 1}\right].$$

Then, the detection probability of the single error is

$$P = 1 - \left\{\prod_{j=1}^{m} \left[1 - P_1 \times (1 - P_1 \times P_{\mathrm{X}})^{j \times n - 1}\right]\right\}.$$

If P_u represents "the detection probability of one error with u X-bits," then combining $P_1 = \omega/m$, $P_{\mathrm{X}} = (u/mn)$, and the formulation of P, P_u can be calculated as

$$P_u = 1 - \left\{\prod_{j=1}^{m} \left[1 - \frac{\omega}{m} \times \left(1 - \frac{\omega}{m} \times \frac{u}{mn}\right)^{j \times n - 1}\right]\right\}.$$

If there are e errors, then the detection probability of these errors in the presence of u X-bits is

$$P_u^e = 1 - \left\{\prod_{j=1}^{m} \left[1 - \frac{\omega}{m} \times \left(1 - \frac{\omega}{m} \times \frac{u}{mn}\right)^{j \times n - 1}\right]\right\}^e.$$

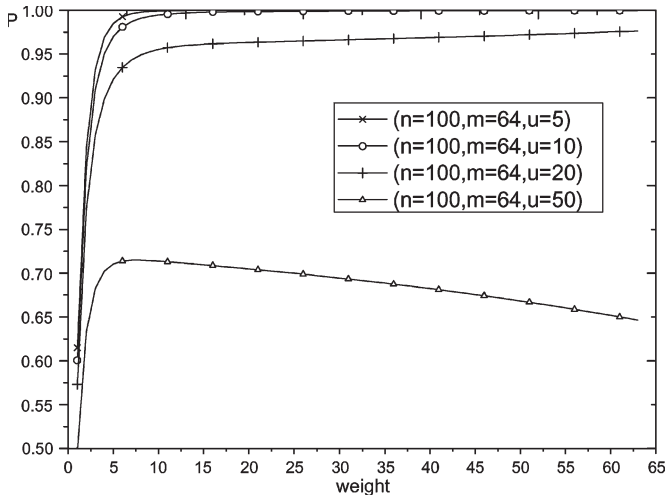

Fig. 10. Detection probability versus weight and X-density.

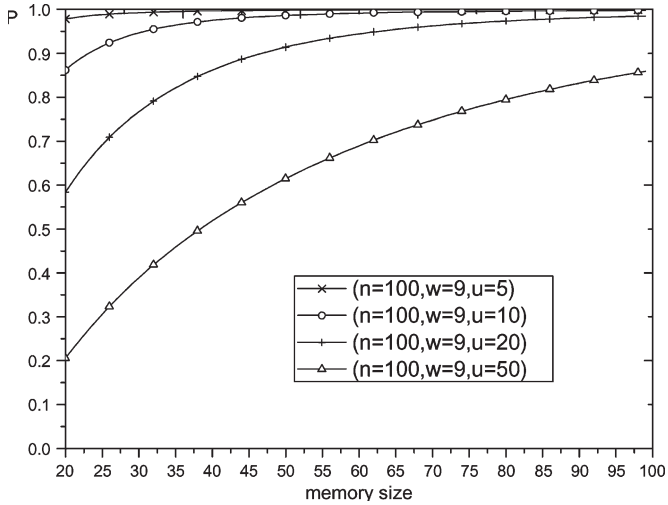

Fig. 11. Detection probability versus memory size and X-density.

Although we cannot get the further form of this equation, we can see that the detection probability is closely related with the memory size and the weight. Their relations are shown in Figs. 10 and 11. In Fig. 10, four cases are presented. In the first three case designs, the detection probabilities of encoders increase with an increase in weight. However, a large weight is not always a good choice. In the fourth case, the P of the encoder reaches the maximum point (71.5%) when weight is equal to 9. The analysis of Fig. 10 shows that the weight must be picked up carefully, which affects the detection probability significantly. To the four cases in Fig. 10, the weight in the range of 7–13 is appreciated.

The curves in Fig. 10 also indicate which times a DFT technique needs to be used to bound X generates. When the X-density is about 0.15% $(u = 10)$, if we select the proper weight, the X-masking probability can be reduced to 0.01%. This can be accepted in the practical scene. However, when the X-density increases to 0.78% $(u = 50)$, the minimal X-masking should reach 29%. This is too large to accept. At this time, some DFT logic must be augmented to reduce the X-density to an acceptable level.

Fig. 11 shows the relation between the detection probability and the memory size. This relation is simple and monotonic. The larger memory size is always good.

In the above analysis procedure, we assume that the distribution of X-bits is random. However, many observations of X-bits in industrial chips show that the X-bits in the test response are largely clustered. This means that the majority of X-bits are produced by only a small number of scan chains. The detailed experimental results are reported in [13].

The probability of one error masked by X-bits is also related with the time that the X-bits reside in the memory elements. This time is determined by the column polynomial of the basic check matrix. Let us examine the type-I encoder. If one X-bit is propagated into the registers that are near the output, the time of X-bits in memory elements are short. These X-bits have little force to the erroneous signature. Seeing an example, if one X-bit is produced in scan-out pin I_i and it is propagated into the zeroth, the first, and the second registers, the column polynomial of I_i in H is $C_i(x) = x^0 + x^1 + x^2$. Thus, these propagated X-bits will be shifted out during three cycles (the zeroth register is nearest to the output). If the column polynomial of I_i is $C_i'(x) = x^3 + x^4 + x^5$, then the propagated X-bits needs six cycles to shift out. It is obvious that C_i is superior to $C_i'(x)$. We define the "force" of an input I_i as τ_i to denote the maximum shift-out time when a single X-bit is injected in this input. It is calculated as

$$\tau_i = \sum_{l=0}^{l=m-1} (d) \left[C_i^l x^d | C_i^l = 1\right]$$

where C_I^l is the coefficient of degree l in the ith column polynomial C_i, and τ_i represents the sum of a nonzero coefficient degree in C_i. Thus, the input with a smaller τ_i can help us reduce the X-masking probability when its X-density is large.

Based the above discussions, the following column-polynomial-generating input assignment algorithm is presented to achieve a low X-masking probability.

H Synthesis and Input Assignment Algorithm

1) Order the outputs of scan chains based on the X-density produced in these outputs. Then, get the ordered-output set S.
2) Divide S into two subsets S_1 and S_2. The sum of the X-densities of outputs in S_1 exceed a threshold (such as 80%). The other outputs are in S_2.
3) Select a small ω (such as 3 or 5). Generate the column polynomials with a small τ. Order these polynomials and connect inputs corresponding to these polynomials with outputs in S_1.
4) Select a large ω (such as 9, 11, or larger). Generate the column polynomials at random. Order these polynomials and connect inputs corresponding to these polynomials with the outputs in S_2.

Some experimental results using this algorithm are listed in Table I. To S_1, we select 3 as ω. To S_2, we select 11 as ω. Cases 1 and 2 have fewer X-bits, cases 5 and 6 have massive X-bits, and cases 3 and 4 are between them. A total of 10^8 erroneous bits are randomly injected and the number of masked erroneous bits are reported in the table. In these cases, the X-bits are clustered in some inputs of the encoder. Two measurements are conducted for each case. For cases 1, 3, and 5, main X-bits are clustered in one input, and we find that the X-masking probability can be reduced drastically when using the proposed algorithm. In cases 2, 4, 6, the X-masking probability is also reduced. For the worst case (case 6), the X-masking probability is reduced from 3% to 0.05%. It is now acceptable.

VI. Design for Diagnosis

The diagnosis capability is needed in prototype or validation testing. If the yield is low in production testing, the designer and the foundry need to analyze the reasons. Thus, the diagnosis is required. Therefore, the diagnosis capacity is important and can help us reduce development costs and debug times. These are especially important when the chip is not mature.

To make the failure diagnosis of the failing gate possible, the exact data for every erroneous bit unloaded from the scan chains must be known to recreate the actual fail information. Theorem 4 brings the primary capability of diagnosis. One error from any scan chain can be identified by comparing the code word with the expected result. Since no two column polynomials in H are equivalent, we can match the column of the matrix with the erroneous bits in the code-word sequence to identify the syndrome. For example, if there are errors in first, third, and fifth bits of a five-bit fraction in failing data, we can try to match it in H and find the second column matched. Then, we confirm that the second scan chain has produced an error. However, this matching operation is not always true if we can not confirm that the only one error is produced.

To extend the diagnostic capability of a convolutional-code H_m-based encoder, some modifications must be added into the encoder. The encoder can be reconfigured into a mixture of a simple-linear-code and a convolutional-code H_m-based encoder. An example of an extended encoder is shown as in Fig. 12.

As compared to Fig. 8, the block diagram presented in Fig. 12 has additional AND gates and control signals. The control signal CM can gate the outputs of an XOR tree and freeze the internal scan chains when it is set to 0. In the normal operation, CM is set to 1. When diagnosing, the procedure can be divided into two phases. For the first phase, $\mathrm{CM} = 1$, and the information of scan outputs are captured into memory elements. In the second phase, CM is set to 0, and outputs of the XOR tree are gated. The content in the memory elements is shifted to the ATE cycle-by-cycle. An off-line software can compare the code word with the expected results and locate the error. From the above procedure, we can see that the proposed encoder has been configured into a simple-linear-code H-based encoder when diagnosing.

Lemma 5: The extended convolutional-code H_m-based encoder can be configured into a simple-linear-code H-based encoder when $\mathrm{CM} = 0$. The check matrix of the new encoder is the basic matrix of a convolutional code H.

TABLE I
X-MASKING PROBABILITY REDUCTION AFTER THE PROPER INPUT ASSIGNMENT

Cases	Encoder	X- Density(%X-bits / total response bits)	X-Bits Distribution	Random Inputs Assignment	Proper Inputs Assignment
Case 1	(n=100, m=64, d=3)	0.0025%	I_1(90%), other(10%)	369	26
				257	22
Case 2	(n=100, m=64, d=3)	0.0025%	I_1(30%), I_2(25%), I_3(10%), I_4(15%), I_5(5%), other(15%)	3267	649
				4657	256
Case 3	(n=100, m=64, d=3)	0.025%	I_1(90%), other(10%)	1846	127
				1432	308
Case 4	(n=100, m=64, d=3)	0.025%	I_1(30%), I_2(25%), I_3(10%), I_4(15%), I_5(5%), other(15%)	10654	1775
				9786	1042
Case 5	(n=100, m=64, d=3)	0.25%	I_1(90%), other(10%)	54123	915
				47899	843
Case 6	(n=100, m=64, d=3)	0.25%	I_1(30%), I_2(25%), I_3(10%), I_4(15%), I_5(5%), other(15%)	302796	6832
				434523	5091

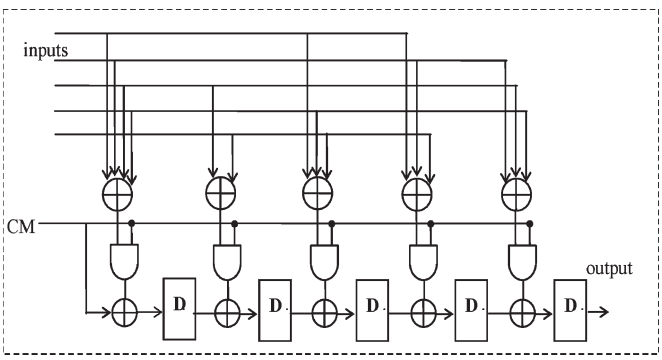

Fig. 12. Extended convolutional-code H_m-based encoder to diagnose.

Since the design of H is far easier than H_m, the extended encoder can enhance the diagnostic capability by increasing the distance in H. This can be implemented by selecting a large memory size of m. Extending this principle, we have a possibility to get the full information of scan chains. The following theorem will change the possibility to a reality.

Theorem 5: The extended convolutional-code H_m-based encoder can identify any error, which can be produced by any scan chain at any time, if the design of basic check matrix follows

$$\text{Rank}(H) = n$$

where $\text{Rank}(H) = n$ is the rank of the check matrix H.

Proof: From Lemma 5, if the outputs information from scan chains at the t cycle is I_t and the state of memory elements is S_t, we get

$$\left(H \times I_t^{\mathrm{T}}\right) \oplus S_{t-1} = S_t$$
$$\Rightarrow H \times I_t^{\mathrm{T}} = S_t \oplus S_{t-1_t}$$

where S_{t-1} can be derived through the preunloading of memory elements. The capability of getting the full input information is

equivalent to solving this equation and getting a unique solution of I_t. Thus, Theorem 6 can be concluded from Cramer's Law [26].

As an example, for the encoder in Fig. 12, if we found that the errors occur at the t cycle, unloading intervals are inserted, and we get the symptomatic states $S_t^E = [1\ 1\ 0\ 1\ 0]$; it is different with the expected state $S_t = [1\,0\,0\,0\,0]$. Assuming that $S_{t-1} = [0\ 0\ 0\ 0\ 0]$, we can calculate the output of scan chains at the t cycle.

A Gauss–Jordan elimination can be used to solve this equation and get the erroneous input vector $I_t^E = [1\ 1\ 0\ 0\ 1]$. Compared to the expected input vector $I_t = [1\ 0\ 1\ 0\ 1]$, we can confirm that the second and the third scan chains have produced the errors. Given the number of scan chains, the minimal number of memory elements satisfying Theorem 5 is

$$m \geq n.$$

This equation can be directly concluded from Theorem 5 and the definition of the rank of matrix.

The diagnostic operation of an extended encoder is simply described as follows.

1) Seed the memory cells in the encoder to 0: $\mathrm{CM} = 0$.
2) Capture the outputs of scan chains to memory cells in the encoder: $\mathrm{CM} = 1$.
3) Freeze the clock of scan chains and unload out the contents of memory cells and seed the memory cells to 0: $\mathrm{CM} = 0$.

This diagnostic operation can be rerun on the tester to collect the successive failing data. It may be written as a test protocol in an Standard Test Interface Language (STIL) format, which can be automatically executed in the tester [29] and [30]. After the interval data is collected, a linear equation is set up to determine the values of scan chains. Then, a diagnostic tool can be used to determine faults that best explain the failing scan cells. ■

VII. Experimental Results of Aliasing

The aliasing of a convolutional-code H_m-based encoder is derived from the error cancellation in the XOR network and memory elements. Because the convolutional code is also a linear code, the aliasing analysis method in [6] is also suited for it. The compaction procedure of the proposed encoder can be looked at as an $m \times n$ bit space mapped into m bit space. If the erroneous bits are distributed in inputs of the encoder with an equal probability, then the aliasing probability can be expressed as

$$P(\text{aliasing}) = \frac{\text{The number of elements in one } m \text{ bit space} - 1}{\text{The number of elements in an } m \times n \text{ bits space}}$$

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

$$\approx \left(\frac{2^{(n-1)}}{2^n}\right)^m = 2^{-m}.$$

TABLE II
Analysis of the Aliasing Probability

100 inputs 4 errors cancellation			
m	ω=3	ω=5	ω=7
30	3722	542	98
40	360	43	33
50	55	24	11
60	32	12	0
70	7	4	0
80	3	8	0
90	0	2	1
100	0	1	0

200 inputs 4 errors cancellation			
m	ω=3	ω=5	ω=7
30	1780	323	112
40	242	75	45
50	63	25	8
60	5	19	3
70	1	2	5
80	0	0	0
90	0	0	0
100	0	0	0

We can see that the aliasing probability of the proposed encoder is only related with the memory size and independent with the compaction ratio. This is a good property. It means that we can design an arbitrary compaction ratio, and we do not need to consider the aliasing probability.

When the memory size is not large enough, the aliasing probability is also related with the weight ω. Some extensive experiments are conducted to evaluate this relation. Table II lists the results. Four errors were randomly inserted in an m-depth pattern to evaluate the aliasing probability. Every measurement was done by 10^8 simulations. Observing the data in Table II, when the memory size is small, the large weight leads a small

TABLE III
ALIASING PROBABILITY OF ACTUAL STUCK-AT FAULTS

Circuits	Faults	Injected Faults	Num. of Simulations	Detected Faults / Total Faults
		1	9815	100%
s13207	9815	2	10^5	100%
		10	10^5	100%
		1	11727	100%
s15850	11727	2	10^5	100%
		10	10^5	100%
		1	39094	100%
s35932	39094	2	10^5	100%
		10	10^5	100%
		1	31180	100%
s38417	31180	2	10^5	100%
		10	10^5	100%
		1	36305	100%
s38584	36305	2	10^5	100%
		10	10^5	100%

aliasing probability. On the other hand, if the memory size is larger than 80, the aliasing probability will be less than 10^{-8}.

We also conducted some experiments to evaluate the effectiveness of the error detection in the benchmark circuits. The flow of experiment is as follows. Several random stuck-at faults are injected in the circuits. A fault simulator (FSIM) [27] is used to get the erroneous response. The response will be encoded into code words and compared with the faultless code words. If they are distinct, we mark this fault as a detected fault. We count all the detected faults. For all these designs, the encoders are designed based on a (62×62) basic check matrix that satisfies Theorems 2 and 5.

Table III shows the results of experiments. We inject one or two faults per simulation to simulate the mild defect and ten faults per simulation to simulate a serious defect. From the experimental data in Table III, we can see that no faults are cancelled completely, and no aliasing occurs in all cases. Although the proposed encoder cannot guarantee zero aliasing, our experiments show that the encoder is very efficient in practice.

VIII. CONCLUSION

In this paper, we presented a test response compaction technique based on a convolutional-code H_m-based encoder. The proposed encoder is a single-output encoder; therefore, the maximum compaction ratio can be obtained. If the design theorems presented are satisfied, the encoder can benefit from the aliasing resistance, diagnosis, and handling of massive X-bits. The proposed encoder is suited for SoC designs where a small area-overhead penalty can greatly enhance handling of X-bits and avoid a full diagnostic capability.

The experimental results show that the weight and the memory size of a code are two very important parameters to optimize the performance of the encoder. The large memory size is always better without considering the area overhead. If optimized weights and memory sizes are selected, the X-masking probability can be drastically reduced to an acceptable level. The experimental results also show that the aliasing probability of this encoder is low enough to be ignored for practical designs.

The proposed encoder enables low-cost testers with a small vector memory and a small number of test channels to have the capacity to test complex chips. This means low cost per hour in production testing. Moreover, the response compaction can be used to reduce the test application time under the physical pin count limitation. Since only one pin is used for outputting a test response in our proposed encoder, shorter scan chains can be achieved by more input pins compared to other response compaction, such as X-compact. The shorter scan chains mean that we can get the benefit of the test cost due to the shorter test application time in production testing.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments, which immensely helped in the preparation of the revised version of the manuscript; Prof. A. Orailoglu of the University of California, San Diego, for the valuable discussion on the related study of this paper during his visit in China; and Prof. K. Saluja for his helpful comments and encouragement.

REFERENCES

[1] Semiconductor Industry Association (SIA), *International Technology Roadmap for Semiconductors (ITRS)*, 1999.
[2] D. Xiang, S. Gu, J. Sun, and Y. Wu, "A cost-effective scan architecture for scan testing with non-scan test power and test application cost," in *Proc. Design Automation Conf.*, Anaheim, CA, 2003, pp. 744–747.
[3] A. Ivanov, B. Tsuji, and Y. Zorian, "Programmable space compactors for BIST," *IEEE Trans. Comput.*, vol. 45, no. 12, pp. 1393–1405, Dec. 1996.
[4] K. Chakrabarty, B. T. Murray, and J. P. Hays, "Optimal zero-aliasing space compaction of test responses," *IEEE Trans. Comput.*, vol. 47, no. 11, pp. 1171–1187, Nov. 1998.
[5] B. B. Bhattacharya, A. Dmitriev, and M. Goessel, "Zero-aliasing space compaction of test responses using a single periodic output," *IEEE Trans. Comput.*, vol. 52, no. 12, pp. 1646–1651, Dec. 2003.
[6] K. K. Saluja and M. Karpovsky, "Testing computer hardware through data compression in space and time," in *Proc. Int. Test Conf.*, Philadelphia, PA, 1983, pp. 83–88.
[7] S. Mitra and K. S. Kim, "X-compact: An efficient response compaction technique," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 23, no. 3, pp. 421–432, Mar. 2004.

[8] J. H. Patel, S. S. Lumetta, and S. M. Reddy, "Application of Saluja–Karpovsky compactors to test responses with many unknowns," in *Proc. VLSI Test Symp.*, Napa, CA, 2003, pp. 107–112.
[9] S. S. Lumetta and S. Mitra, "X-codes: Error control with unknowable inputs," in *Proc. Int. Symp. Information Theory*, Yokohama, Japan, 2003, p. 102.
[10] C. Wang, S. M. Reddy, I. Pomeranz, J. Rajski, and J. Tyszer, "On compacting test response data containing unknown values," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, 2003, pp. 855–862.
[11] J. Rajski, J. Tyszer, C. Wang, and S. M. Reddy, "Convolutional compaction of test responses," in *Proc. Int. Test Conf.*, Charlotte, NC, 2003, pp. 745–754.
[12] Y. Han, Y. Xu, H. Li, X. Li, and A. Chandra, "Test resource partitioning based on efficient responses compaction for test time and tester channels reduction," in *Proc. Asian Test Symp.*, Xi'an, China, 2003, pp. 440–445.
[13] P. Wohl and L. Huisman, "Analysis and design of optimal combinational compactors," in *Proc. VLSI Test Symp.*, Napa, CA, 2003, pp. 101–106.
[14] I. Pomeranz, I. S. Kundu, and S. M. Reddy, "On output response compression in the presence of unknown output values," in *Proc. Design Automation Conf.*, New Orleans, LA, 2002, pp. 255–258.
[15] P. Wohl, J. A. Waicukauski, and T. W. Williams, "Design of compactors for signature-analyzers in Built-IN SELF-TEST," in *Proc. Int. Test Conf.*, Baltimore, MD, 2001, pp. 54–63.
[16] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, A. Ferko, B. Keller, D. Scott, B. Koenemann, and T. Ondera, "Extending OPMISR beyond 10× scan test efficiency," *IEEE Des. Test Comput.*, vol. 19, no. 5, pp. 65–73, Sep./Oct. 2002.
[17] P. Elias, "Coding for noisy channels," *IRE Int. Conv. Rec.,* pp. 37–46, 1955.
[18] J. L. Massey, D. J. Costello, and J. Justesen, "Polynomial weights and code constructions," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 1, pp. 101–110, Jan. 1973.
[19] K. Larson, "Short convolutional codes with maximal free distance for rates 1/2, 1/3, and 1/4," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 3, pp. 371–372, May 1973.
[20] J. Justesen, "New convolutional code constructions and a class of asymptotically good time-varying codes," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 2, pp. 220–225, Mar. 1973.
[21] G. D. Forney, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 6, pp. 720–738, Nov. 1970.
[22] H. G. Luerssen and W. Schmale, *Distance Bounds for Convolutional Codes and Some Optimal Codes*. [Online]. Available: http://www.uni-oldenburg.de/math/personen/gluesing/OptCodes.pdf
[23] J. Rosenthal, J. M. Schumacher, and E. V. York, "The behavior of convolutional codes," Nat. Res. Inst. Math. Comput. Sci., Amsterdam, The Netherlands, Rep. BS-R9533, 1995.
[24] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, Oct. 1, 1982.
[25] R. J. Mceliece, *The Theory of Information and Coding*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2002.
[26] R. A. Brualdi, *Introductory Combinatorics*, 3rd ed. Upper Saddle River, NJ: Pearson Education, 1999.
[27] H. K Lee and D. S. Ha, "On the generation of test patterns for combinational circuits," Dept. Elect. Eng., Virginia Tech., Blacksburg, Tech. Rep. No. 12-93, 1993.
[28] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, and J. Rajski, "Logic BIST for large industrial designs: Real issues and case studies," in *Proc. Int. Test Conf.*, Atlantic City, NJ, 1999, pp. 358–367.
[29] Tetramax Manual. [Online]. Available: http://www.synopsys.com/products/test/tetramax_wp.html
[30] P. Wohl, J. A. Waicukauski, S. Patel, and G. Maston, "Effective diagnostics through interval unloads in a BIST environment," in *Proc. Design Automation Conf.*, New Orleans, LA, 2002, pp. 54–63.
[31] Y. Han, Y. Hu, H. Li, and X. Li, "Theoretic analysis and enhanced X-tolerance of test response compact based on convolutional code," in *Proc. IEEE Asia and South Pacific Design Automation Conf.*, Shanghai, China, 2005, pp. 53–58.
[32] Y. Han, Y. Hu, H. Li, X. Li, and A. Chandra, "Response compaction for test time and test pins reduction based on advanced convolutional codes," in *Proc. IEEE Int. Symp. Defect and Fault Tolerance VLSI Systems*, Cannes, France, 2004, pp. 298–305.

Yinhe Han (M'02) received the B.Eng. degree in electrical engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1997. He is currently working toward the Ph.D. degree in computer science at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

His research interests include very-large-scale-integration/system-on-a-chip design and design for testability.

Mr. Han received the IEEE Test Technology Technical Council Best Paper Award of the Asian Test Symposium in 2003.

Xiaowei Li (M'00–SM'04) received the B.Eng. and M.Eng. degrees from Hefei University of Technology, Hefei, China, in 1985 and 1988, respectively, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing China, in 1991, all in computer science.

In 1991, he joined the Department of Computer Science and Technology, Peking University, Beijing, as a Postdoctoral Research Associate and was promoted as an Associate Professor in 1993. From 1997 to 1998, he was a Visiting Research Fellow in the Department of Electrical and Electronic Engineering, University of Hong Kong. In 1999 and 2000, he was a Visiting Professor in the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. In 2000, he was a Professor in the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include very-large-scale-integration/system-on-a-chip design verification and test generation, design for testability, low-power design, and dependable computing.

Dr. Li received the Natural Science Award from the Chinese Academy of Sciences in 1992 and the Certificate of Appreciation from IEEE Computer Society in 2001. He is an Area Editor of the *Journal of Computer Science and Technology* and an Associate Editor-in-Chief of the *Journal of Computer-Aided Design and Computer Graphics* (in Chinese).

Huawei Li (S'00–A'01–M'04) received the B.S. degree in computer science from Xiangtan University, Hunan, China, in 1996 and the M.S. and Ph.D. degrees in computer architecture from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 1999 and 2001, respectively.

She is currently an Associate Professor at the Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include very-large-scale-integration/system-on-a-chip design verification and test generation, delay test, and dependable computing.

Anshuman Chandra (S'98–M'04) received the B.Eng. degree in electrical engineering from the University of Roorkee, Roorkee, India in 1998 and the M.S. and Ph.D. degrees in electrical and computer engineering from Duke University, Durham, NC, in 2000 and 2002, respectively.

He is currently a Senior R&D Engineer at Synopsys, Inc., Mountain View, CA. His research interests include very-large-scale-integration (VLSI) design, digital testing, and computer architecture.

Dr. Chandra is a member of the Association for Computing Machinery Special Interest Group on Design Automation. He received the Test Technology Technical Council James Beausang Student Paper Award for a paper in the IEEE VLSI Test Symposium in 2000 and Best Paper Award at the Design, Automation, and Test in Europe Conference in 2001.