社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2004－SLDM－117（27）
2004／12／2

畳込み型テスト応答圧縮器における誤り見逃し率の解析

新井 雅之† 福本 聡† 岩崎 一彦†

†東京都立大学大学院工学研究科 〒192-0397 東京都八王子市南大沢 1-2

E-mail: {arai, fukumoto, iwasaki}@eei.metro-u.ac.jp

あらまし 畳込み型テスト応答圧縮器は，不定値を含むテスト応答圧縮に対して有効な技術である．欠点の一つとして，誤り見逃しが挙げられる．すなわち，テスト応答に現れる複数の誤りが互いにマスクされ，検出不可能となる場合が存在する．本稿では，畳込み圧縮器における誤り見逃し率の解析的な評価について議論する．まず，双対符号の重み分布をもとにした計算方法について述べる．次に，4 重および 6 重誤り見逃し率を高速に導出する手法について示す．数値例より，解析結果がモンテカルロシミュレーションによって得られた結果とほぼ等しいことを示す．また，一定の誤り発生率のもとで，誤り見逃し率が極大値を持つことを示す．

キーワード 畳込み圧縮器，テスト応答圧縮器，誤り見逃し率，双対符号，重み分布

Analysis on Error Masking Rate for Convolutional Compactors

Masayuki ARAI† Satoshi FUKUMOTO† Kazuhiko IWASAKI†

†Graduate School of Engineering, Tokyo Metropolitan University

1-2 Minami-osawa, Hachioji, Tokyo 192-0397, Japan

E-mail: {arai, fukumoto, iwasaki}@eei.metro-u.ac.jp

Abstract Convolutional compactors offer a promising technique of compacting test responses that include unknown values. One drawback of this compaction technique is error masking, i.e., some errors appearing in the test responses cannot be detected due to mutual cancellation. In this work, we theoretically analyze error-masking probability. First, we apply weight distributions of binary linear error-correcting codes to derive the error-masking probability. We then present a fast calculation scheme for 4- and 6-error-masking probabilities. Numerical examples reveal that they are about the same as those obtained by Monte-Carlo simulations. We also found that the masking probability as a function of test length has a peak value for constant error probabilities, such as 0.01, and 0.001.

Keyword convolutional compactor, test response compactor, error masking rate, dual code, weight distribution

1. Introduction

With the advance in semiconductor technology, the cost of testing VLSI chips has been increasing [1]. Testing them is requiring exponential amounts of test data and longer testing times due to their faster clock frequency and increasing scale, resulting in higher costs. Therefore, it is important not only to reduce test stimuli but also to reduce the number of test responses that needed to be compared. To overcome these difficulties, many testing techniques have been proposed [2-5].

One promising testing technique is the built-in self-test (BIST), where a linear feedback shift–register (LFSR) is used as a pseudo-random test generator and a multiple-input signature register (MISR) is used as a test response compactor [2]. Some drawbacks of the BIST structure are: (a) It is quite difficult for LFSRs to generate test patterns that provide 100% or very close to 100% fault coverage. (b) More than two erroneous responses cancel one another resulting in a fault-free signature. Many technique including reseeding was proposed [2.3] to improve fault coverage. Aliasing probabilities for MISRs have also been analyzed applying various techniques [6].

Testing methods that combine scan chains and BIST have also been proposed [7]. However, it is still difficult to attain 100% fault coverage with this technique. Consequently, many hybrid BIST schemes [8-10] have been proposed to provide better BIST patterns to circuits under test (CUTs). In other words, BIST has been considered a practical solution to overcoming the difficulties with VLST testing [11-12].

As the number of transistors being integrated into chips increases, more attention has been focused on SOC testing [5]. As a result, many commercial hybrid BIST tools such as EDT [13-15], smart BIST [16], Encounter Test [17], and BAST [18] have appeared on the market. One key technique for hybrid BISTs is their handling of unknown values. Masking logic is usually inserted between the test response and compaction circuits. A masking technique using an N-detection test set has also been proposed [19].

An X-compactor uses an EOR network that compacts test responses and unknown values are propagated in limited test cycles [20]. Furthermore, the convolutional compactor proposed by Rajski et al. is a promising technique to compact test responses even if there are

some unknown values [21]. It comprises non-feedback shift registers and an EOR network that connects scan chains of the CUT to shift registers. By arranging the EOR network, it can effectively compact responses in excess of 100 times, while simultaneously detecting double as well as odd-numbered errors. Convolutional compactors still have difficulties in that they cause error masking, i.e., some errors appearing in the test responses cannot be detected due to mutual cancellation. Rajski et al. [21] evaluated error-masking probabilities through extensive Monte-Carlo simulations.

In this paper, we theoretically analyze the exact and approximate error-masking probabilities for convolutional compactors. First, we apply weight distributions of binary linear error-correcting codes to derive error-masking probability. We then show a fast calculation technique of calculating 4-error and 6-error-masking probabilities. Peak error-masking probabilities as well as asymptotical error-masking values are also evaluated [22].

This paper is organized as follows. We briefly explain the basic concept behind convolutional compactors and their error-masking properties in Sec. 2. In Sec. 3, we discuss our analysis of error-masking probability where we apply weight distributions. Section 4 presents a faster calculation scheme for 4- and 6-error-masking probabilities and various numerical examples. Section 5 concludes this paper.

2. Convolutional compactors

Here, we will briefly explain the basic arrangement of convolutional compactors, as well as error masking [21].

2.1 Basic arrangement of convolutional compactors

Figure 1 outlines the basic arrangement of a convolutional compactor. Scan chains in the CUT are connected to any of the M flip-flops (FFs) via the network of EOR gates, called an injector network. Let S denote the number of scan chains in the CUT, and let b denote the number of outputs from the convolutional compactor. The FFs are therefore divided into b groups, and each of them is arranged as an shift register with length M/b. The FFs are driven by the same scan-shift clock as the scan chains in the CUT. Then, during N scan-shift clock cycles, the CUT outputs $S{\cdot}N$-bit test responses, while the convolutional compactor outputs $b{\cdot}N$-compacted test responses. Therefore, the compaction rate for test responses on the convolutional compactor is b/S.

Scan chain 1 is connected to FFs 1, 2, and 3 in Fig. 1, and is not connected to FF 4. When an erroneous value is output from scan chain 1, it is propagated into the three connected FFs. During

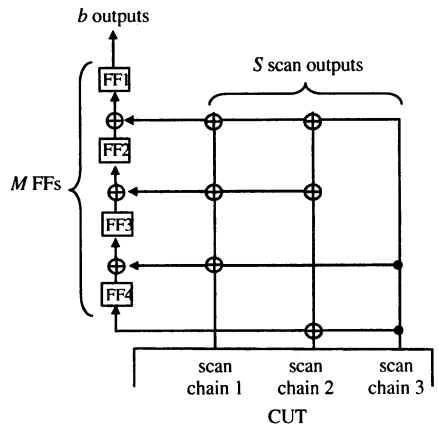

Fig. 1. Example arrangement for convolutional compactor. ($b = 1, M = 4, S = 2$)

succeeding shift clock cycles, the convolutional compactor outputs errors three times.

We describe the connectivity of the injector network by S by b matrix $\mathbf{CN}(x)$,

$$\mathbf{CN}(x) = \begin{bmatrix} CN_{1,1}(x) & \cdots & CN_{b,1}(x) \\ \vdots & \ddots & \vdots \\ CN_{1,S}(x) & \cdots & CN_{b,S}(x) \end{bmatrix}, \tag{1}$$

where $CN_{i,j}(x)$ is a polynomial over $GF(2)$ at a degree of $M/b - 1$,

$$CN_{i,j}(x) = c_{i,j,1} + c_{i,j,2} \cdot x + \ldots + c_{i,j,M/b} \cdot x^{M/b-1}. \tag{2}$$

Parameter $c_{i,j,k}$ is 1 if the scan chain j is connected to the k-th FF in the i-th shift register. For example, the injector network in Fig. 1 is described as

$$\mathbf{CN}(x) = \begin{bmatrix} 1 + x + x^2 \\ 1 + x^2 + x^3 \\ 1 + x + x^3 \end{bmatrix}. \tag{3}$$

We also describe the injector network as M by S binary matrix, $\mathbf{CN}$,

$$\mathbf{CN} = \begin{bmatrix} \mathbf{CN}_1 \\ \vdots \\ \mathbf{CN}_b \end{bmatrix}. \tag{4}$$

Each of $\mathbf{CN}_1$, ..., $\mathbf{CN}_b$ is an M/b by S binary matrix, i.e.,

$$\mathbf{CN}_i = \begin{bmatrix} c_{i,1,1} & \cdots & c_{i,S,1} \\ \vdots & \ddots & \vdots \\ c_{i,1,M/b} & \cdots & c_{i,S,M/b} \end{bmatrix}. \tag{5}$$

In Fig. 1, $\mathbf{CN}$ is arranged as:

$$\mathbf{CN} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \tag{6}$$

External testers can detect errors when at least one erroneous value is output, while a convolutional compactor may output more than one erroneous value even if the error is only input from scan chains once. This characteristic contributes to detecting multiple errors and handling unknown values output from scan chains, despite greatly decreasing the compaction rate of test responses.

2.2 Error masking and injector network

A convolutional compactor cannot always output erroneous values, because there is the possibility that more than one erroneous bit will occur to cancel each other. Let us explain this error-masking property. Let $\mathbf{EF}(x) = [EF_1(x) \ \ldots \ EF_S(x)]$ denote a response vector that represents the expected (error-free) values output from scan chains during N shift clock cycles. Also, let $\mathbf{ER}(x) = [ER_1(x) \ \ldots \ ER_S(x)]$ denote an error vector, which represents the positions of errors. Elements in $\mathbf{EF}(x)$ and $\mathbf{ER}(x)$ are polynomials at a degree of $N - 1$. Thus, an element in $\mathbf{ER}(x)$, $ER_i(x)$, which is for the i-th scan chain, is described as

$$ER_i(x) = e_{i,1} + e_{i,2} \cdot x + \ldots + e_{i,N} \cdot x^{N-1}, \tag{7}$$

where $e_{i,j}$ is 1 if an erroneous value is output from scan chain i at the j-th shift clock cycle. Let $\mathbf{RS}(x) = [RS_1(x) \ \ldots \ RS_b(x)]$ denote the output response vector that represents bit sequences output from the convolutional compactor. Each element $RS_i(x)$, representing responses

from the i-th shift register, is a polynomial at a degree of $N + M/b - 2$. Then, the output sequence is denoted as

$$RS(x) = \{EF(x) + ER(x)\} \cdot CN(x). \quad (8)$$

If $\mathbf{RS}(x)$ is equal to $\mathbf{EF}(x)\cdot\mathbf{CN}(x)$, the external tester determines that errors do not occur. That is, if $\mathbf{ER}(x)\cdot\mathbf{CN}(x) = \mathbf{0}$ despite $\mathbf{ER}(x) \neq \mathbf{0}$, errors will be masked. We call $\mathbf{ER}(x)\cdot\mathbf{CN}(x)$ the error indicator vector, $\mathbf{EI}(x)$.

For example, consider the injector matrix of $M = 4$, $b = 1$, $S = 3$, and $N = 2$, where $\mathbf{CN}(x)$ is given by Eq. (3), and quadruple errors $e_{1,1}$, $e_{1,2}$, $e_{2,2}$, and $e_{2,3}$ occur. Here, $\mathbf{ER}(x)$ is described as $\mathbf{ER}(x) = [1 \quad 1+x \quad x]$. Then,

$$\mathbf{ER}(x)\cdot\mathbf{CN}(x) = \begin{bmatrix}1 & 1+x & x\end{bmatrix}\begin{bmatrix}1+x+x^2\\ 1+x+x^3\\ 1+x^2+x^3\end{bmatrix} = [0], \quad (9)$$

i.e., these quadruple errors are masked.

To reduce the chances of errors being masked, the convolutional compactor imposes some restriction on the arrangement of injector networks [21].

(R1) Each scan chain has to be connected to the equal odd number of q FFs. That is, out of $c_{1,i,1}$, ..., $c_{b,i,M/b}$, q elements must be 1 and the others must be 0, for each i for $1 \le i \le S$.

(R2) A set of i, j, k ($1 \le i, j \le b$, $i \neq j$, $k \ge 0$) must not exist that satisfies $CN_{i,1}(x) = x^k CN_{j,1}(x)$, ..., and $CN_{i,S}(x) = x^k CN_{j,S}(x)$.

The first restriction guarantees that any odd number of errors can be detected. The second restriction guarantees that any combinations of double errors input at different clock cycles can be detected. For example, consider double errors $e_{i1,j1}$ and $e_{i2,j2}$ occuring, where $j_1 \neq j_2$. Here, the rows of $\mathbf{EI}(x)$, $EI_1(x)$, ..., $EI_b(x)$, are arranged as

$$EI_k(x) = x^{j1-1}CN_{k,i1}(x) + x^{j2-1}CN_{k,i2}(x) \quad (1 \le k \le b). \quad (10)$$

Then, if restriction (R2) holds, at least one from $EI_1(x)$, ..., $EI_b(x)$ will satisfy $EI_k(x) \neq 0$. Consequently, these errors can be detected. Because of these restrictions, convolutional compactors can always detect any single error and 2, 3, 5, 7, ... errors.

3. Exact analysis of error-masking probability

Although the injector network satisfies restrictions (R1) and (R2), convolutional compactors still present the possibility of masking an even number of errors, and this reduces the reliability of test compaction. In this section, we discuss an exact analysis of error-masking probability for a given injector network.

As discussed in Sec. 2.2, whether t multiple errors $e_{i1,j1}$, ..., $e_{it,jt}$ are masked or not depends on the output bit sequence, which is calculated by the summation of shifted rows of $\mathbf{CN}(x)$, $x^{j'}CN_{i',k}(x)$ ($1 \le t' \le t$, $1 \le k \le b$). Because the number of scan-shift clocks is N, we need to first arrange error-masking matrix $\mathbf{EM}(x)$ that contains all possible rows, each of which corresponds to one error. $\mathbf{EM}(x)$ is an $S{\cdot}N$ by b matrix, and is described as

$$\mathbf{EM}(x) = \begin{bmatrix}\mathbf{CN}(x)\\ x\mathbf{CN}(x)\\ \vdots\\ x^{N-1}\mathbf{CN}(x)\end{bmatrix}. \quad (11)$$

Based on $\mathbf{EM}(x)$, we can determine whether error masking occurs for a given set of t errors.

[Theorem]
The t-bit error output during N shift-clock cycles, $e_{i1,j1}$, ..., $e_{it,jt}$ is masked by the convolutional compactor if and only if corresponding t rows in $\mathbf{EM}(x)$ are linearly dependent, i.e.,

$$\mathbf{EM}_{(j1-1)\cdot S+i1}(x) + \ldots + \mathbf{EM}_{(jt-1)\cdot S+it}(x) = 0. \quad (12)$$

[Proof] When t errors $e_{i1,j1}$, ..., $e_{it,jt}$ occur, each element of the error indicator vector $\mathbf{EI}(x)$ is calculated as

$$EI_k(x) = \sum_{t'=1}^{t} x^{jt'-1}CN_{k,it'}(x). \quad (1 \le k \le b) \quad (13)$$

The u-th row of $\mathbf{EM}(x)$ is arranged as

$$\mathbf{EM}_u(x) = \left[x^{\lceil u/S\rceil}CN_{1,(u \bmod S)} \quad \cdots \quad x^{\lceil u/S\rceil}CN_{1,(u \bmod S)}\right] \quad (14)$$

Then, if t rows of $\mathbf{EM}(x)$, which are $\mathbf{EM}_{(j1-1)\cdot S+i1}(x)$, ..., $\mathbf{EM}_{(jt-1)\cdot S+it}(x)$, are linearly dependent,

$$\sum_{t'=1}^{t} x^{jt'-1}CN_{k,it'}(x) = 0 \quad (15)$$

holds for each k ($1 \le k \le b$). Thus,

$$EI_k(x) = 0. \quad (1 \le k \le b) \quad (16)$$

If these rows are linearly independent, a parameter k exists that satisfies $EI_k(x) \neq 0$, and thus errors can be detected. [Q. E. D]

We can also describe the error-masking matrix as binary matrix $\mathbf{EM}$. Matrix $\mathbf{EM}$ is $(M + b\cdot(N-1))$ by $S{\cdot}N$ matrix, and described as

$$\mathbf{EM} = \begin{bmatrix}\mathbf{CN}_1 & \mathbf{0} & & \mathbf{0}\\ \mathbf{0} & \mathbf{CN}_1 & & \vdots\\ \vdots & \mathbf{0} & & \\ & \vdots & \ddots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & & \mathbf{CN}_1\\ & & \vdots & \\ \mathbf{CN}_b & \mathbf{0} & & \mathbf{0}\\ \mathbf{0} & \mathbf{CN}_b & & \vdots\\ \vdots & \mathbf{0} & & \\ & \vdots & \ddots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & & \mathbf{CN}_b\end{bmatrix}. \quad (17)$$

For example, when $N = 3$ for the $\mathbf{CN}$ in Eq. (6), $\mathbf{EM}$ is arranged as

$$\mathbf{EM} = \begin{bmatrix}1&1&1&0&0&0&0&0&0\\ 1&1&0&1&1&1&0&0&0\\ 1&0&1&1&1&0&1&1&1\\ 0&1&1&1&0&1&1&1&0\\ 0&0&0&0&1&1&1&0&1\\ 0&0&0&0&0&0&0&1&1\end{bmatrix}. \quad (18)$$

Considering $\mathbf{BM}$, Theorem 1 is rewritten as:

[Corollary]
The t errors during N shift-clock cycles, $e_{i1,j1}$, ..., $e_{it,jt}$ are masked if and only if t columns in $\mathbf{EM}$, which are $((j1-1)\cdot S+i1)$-th, ..., and $((jt-1)\cdot S+it)$-th columns, are linearly dependent.

By calculating all the combinations of t rows of $\mathbf{EM}(x)$ that satisfy Eq. (12), we can obtain the number of combinations for t-error masking, A_t. This problem of obtaining A_t is equivalent to calculating the weight distribution of an (n, k) error-correcting code whose parity check matrix is given by $\mathbf{EM}$. Parameter n denotes the length of code words in bits, and is equal to the width of $\mathbf{EM}$, i.e., $S{\cdot}N$. Parameter k is the length of information bits, and is calculated as $k = n - (M + b\cdot(N-1))$. However, it becomes more difficult to check all combinations for a larger N, because the number of combinations exponentially increases as N increases.

It is known as MacWilliams identity [10] that the weight distribution function of an error-correcting code, $A(z) = A_0 + A_1 z + \ldots + A_n z^n$, can be calculated from the weight distribution function of its

dual code, $B(z) = B_0 + B_1 z + \ldots + B_n z^n$. The dual code is also an error-correcting code whose generator matrix is given by the original code's parity check matrix.

[MacWilliams identity]

$$A(z) = 2^{-(n-k)}(1+z)B\left(\frac{1-z}{1+z}\right). \tag{19}$$

The number of code words for the dual code is 2^{n-k}, while that for the original code is 2^k. In convolutional compactors, $b < S$ always holds so that $2^{n-k} < 2^k$ when N becomes larger. Thus, it is computationally feasible to calculate the number of combinations for error masking with B_i and Eq. (19) than directly calculating A_i.

After calculating weight distribution A_i, we can obtain error-masking probability as

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i}, \tag{20}$$

where p is the bit-error probability. Also, substituting Eq. (20) into Eq. (19), we can directly calculate $P_u(E)$ from B_i, as

$$P_u(E) = 2^{-(n-k)} \cdot \sum_{i=0}^{n} B_i (1-2p) - (1-p)^n \tag{21}$$

Figure 2 plots the calculation results for error-masking probability for the injector network given by Eq. (3). The bit error probability p was set to 0.01, 0.05, 0.1, and 0.5. Except for $p = 0.01$, error-masking probability has a maximum value for $N < 20$. For example, when p = 0.05, the error-masking rate has a maximum value of 5.5E-5 at $N = 8$. The figure also shows that when p is higher, the error-masking probability decreases faster as N increases. This is because the chance of at least one error being detected without masking was increased when p and N were higher.

Figure 3 plots the calculation results for error-masking probability for $b = 2$, $M = 6$, $S = 16$, and $q = 3$. The injector network is given as

$$\mathbf{CN} = \begin{bmatrix} 1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0 \\ 1&1&1&1&0&0&0&0&0&0&1&1&1&0&0&0 \\ 1&0&0&0&1&1&1&0&0&0&1&0&0&1&1&0 \\ 0&1&0&0&1&0&0&1&1&0&1&1&1&1&1&1 \\ 0&0&1&0&0&1&0&1&0&1&0&1&0&1&0&1 \\ 0&0&0&1&0&0&1&0&1&1&0&0&1&0&1&1 \end{bmatrix}, \tag{22}$$

which is the same as in Fig. 1 in Ref. [21]. Figure 3 has similar tendencies as Fig. 2, i.e., the error-masking probability has maximum values and decreases more quickly as N increases. The maximum values were observed at $N = 1$ except for $p = 0.01$.

4. Analysis of 4- and 6-error masking probabilities

Analysis of the error-masking probability described in the previous section gives an exact one for given parameters and injector matrix **CN**. However, this still has the problems with computational complexity because the number of code words for the dual code also exponentially increases as N increases, although this is less than those for the original code. The characteristics of the convolutional compactor and Eq. (20), on the other hand, indicate that 4- and 6-error-masking occur most and second-most frequently, and that the 4-error-masking probability is significantly higher when p is small, because 1, 2, 3, and 5 errors are not masked. In this section, we will present a fast scheme to calculate 4-error-masking probability $P_4(E)$ and 6-error-masking probability $P_6(E)$, which is intended to approximate overall error-masking probability as $P_u(E) \approx P_4(E)$ or $P_u(E) \approx P_4(E) + P_6(E)$.

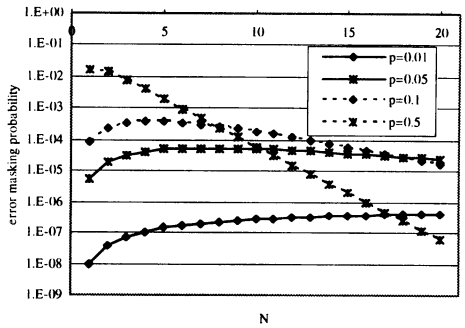

Fig. 2. Calculation results of error masking probability. ($b = 1, M = 4, S = 2, q = 3$)

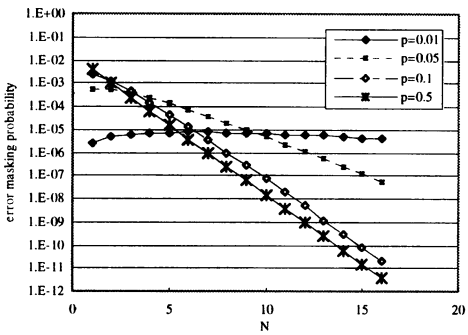

Fig. 3. Calculation results of error masking probability. ($b = 2, M = 6, S = 16, x = 3$)

4.1 Fast calculation of 4- and 6-error-masking probabilities

Let $A_4(N)$ denote the number of combinations for 4-error-masking for a given N, i.e., the number of combinations of four columns in **EM** that are linearly dependent. We can see that there is an interesting correlation between $A_4(N)$ and $A_4(N+1)$. For example, let us consider the case where $b = 1$ and $N = 1$. Here, **EM** is equal to **CN** and the number of combinations for 4-error masking is $A_4(1)$. We have to calculate $A_4(1)$ and assume $A_4(1) = \gamma$. Then, when considering $A_4(2)$, **EM** is arranged as

$$\mathbf{EM} = \begin{bmatrix} \mathbf{CN}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{CN}_1 \end{bmatrix}, \tag{23}$$

As we can see from Fig. 4, the leftmost S columns are exactly the same as $N = 1$ and thus there are $A_4(1)$ combinations for 4-error masking in these columns. Similarly, the rightmost S columns have the same number of combinations. In addition, some new combinations might exist. The number of new combinations for 4-error masking, γ, must be checked, but one of the four columns has to be taken up from the leftmost S columns, and another has to be taken up from the rightmost S columns. Then, the combinations that have to be checked is less than all possible ones, and $A_4(2)$ is calculated as the summation of $A_4(1) + \gamma$ and newly-discovered combinations for 4-error masking.

Let $\gamma_4(N)$ denote the number of newly-discovered combinations for 4-error masking at N. Then, $A_4(N)$ can be described as

$$A_4(N)=\begin{cases}\gamma_4(1), & (N=1)\\ A_4(N-1)+\sum_{i=1}^{N}\gamma_4(i). & (N>1)\end{cases} \quad (24)$$

We can also say that newly-discovered combinations $\gamma_4(N)$ are equal to zero for large N.

[Lemma]

$\gamma_4(N) = 0$ if $N > 2M/b - 2$.

[Proof] Without loss of generality, let i_1 denote the scan-shift clock where the earliest error is output from a scan chain. This error is output q times during succeeding M/b scan-shift clock cycles. To mask this error, at least two further errors have to occur until $i_{M/b}$, because two errors cannot mask each other. When the third error occurs at $i_{M/b}$, the convolutional compactor outputs at least one error during the scan shift clocks between $i_{M/b+1}$ and $i_{2M/b-1}$. For the errors to be masked, the third and the fourth errors have to share at least one clock cycle, i.e., the fourth error has to occur until $c_{2M/b-2}$. [Q. E. D]

Table 1 shows calculation results for $\gamma_4(N)$ and $A_4(N)$ for the injector network of Eq. (18). The number of combinations that have to be checked is much less than all possible combinations. $\gamma_4(N)$ becomes 0 at $N = 4$. The bound in the lemma takes the worst case into consideration, and thus $\gamma_4(N)$ usually becomes 0 at an N less than $2M/b - 2$, depending on **CN**.

From the lemma, Eq. (24) can be rewritten as

$$A_4(N)=\begin{cases}\sum_{i=0}^{N-1}(N-i)\gamma_4(i), & (N\le 2M/b-2)\\ (N-2M/b-2)\sum_{i=0}^{2M/b-2}\gamma_4(i) & \\ +\sum_{i=0}^{2M/b-2}(2M/b-2-i)\gamma_4(i). & (N>2M/b-2)\end{cases} \quad (25)$$

That is, when $N > 2M/b - 2$, $A_4(N)$ increases constantly by $\gamma_4(1) + ... + \gamma_4(2M/b - 2)$, and we can calculate $A_4(N)$ quickly only by calculating $\gamma_4(1), ..., \gamma_4(2M/b - 2)$.

Based on $A_4(N)$, 4-error-masking probability $P_4(E)$ can be calculated as

$$P_4(E) = A_4(N)p^4(1-p)^{n-4}, \quad (26)$$

where $n = S \cdot N$.

Similarly, 6-error-masking probability $P_6(E)$ can be derived as

$$P_6(E) = A_6(N)p^6(1-p)^{n-6}, \quad (27)$$

where

$$A_6(N)=\begin{cases}\sum_{i=0}^{N-1}(N-i)\gamma_6(i), & (N\le 3M/b-2)\\ (N-3M/b+2)\sum_{i=0}^{3M/b-2}\gamma_6(i) & \\ +\sum_{i=0}^{3M/b-2}(3M/b-2-i)\gamma_6(i). & (N>3M/b-2)\end{cases} \quad (28)$$

For 6-error masking, the calculation for $\gamma_6(N)$ requires more computation than those for $\gamma_4(N)$, because we have to check combinations in the 6 columns. Also, the bound of N, where $\gamma_6(N) = 0$ holds, becomes larger than in the lemma.

4. 2 Numerical Examples

Figure 4 plots the calculation results for 4-error-masking probability for the injector network given by Eq. (3). The results for error-masking probability, which are the same as in Fig. 2 have also been plotted for comparison. Four-error-masking probability can approximate overall error-masking probability well. Note that we can calculate 4-error-masking probability for any N, once $\gamma_4(2M/b - 2)$ can be obtained. Thus, the figure is plotted within the range of $N <$ 500, indicating there are also maximum values for p lower than 0.01.

Figure 5 plots the calculation results for 4-error-masking probability $P_4(E)$ for the injector matrix given by Eq. (22), i.e., for b = 2, M = 6, S = 16, and q = 3. Error-masking probability $P_u(E)$, 6-error-masking probability $P_6(E)$, and $P_4(E) + P_6(E)$ have also been shown for comparison. When p is lower than or equal to 0.01, $P_4(E)$ can approximate $P_u(E)$, although they do not match well for $p = 0.1$. Although $P_6(E)$ is much lower than $P_4(E)$, $P_4(E) + P_6(E)$ significantly fits closer to $P_u(E)$ under a higher p. We can also see that the existence of maximum values for error-masking probabilities.

We can obtain N that corresponds to a maximum value for 4-error masking probability by solving

$$A_4(N+1)\cdot p^4(1-p)^{S(N+1)-4} - A_4(N)\cdot p^4(1-p)^{SN-4} = 0. \quad (29)$$

Table 2 lists the calculation results for maximum 4-error-masking

Table 1. Calculation results of error masking probability.
(b = 2, M = 6, S = 16, x = 3)

N	all combinations	combinations checked	$\gamma_4(N)$	$A_4(N)$
1	1820	1820	63	63
2	35960	32320	211	337
3	194580	124480	102	713
4	635376	282176	0	1089
5	1581580	505408	0	1465
6	3321960	794176	0	1841

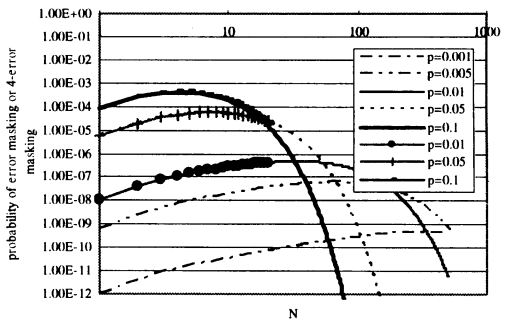

Fig. 4. Calculation results of 4-error masking probability.
(b = 1, M = 4, S = 3, x = 3)

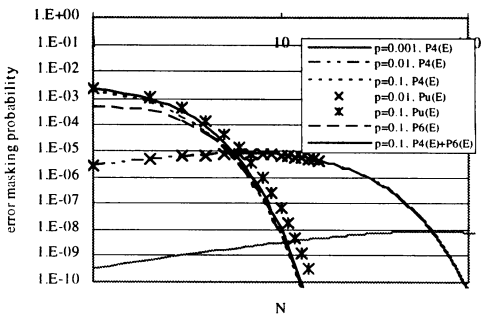

Fig. 5. Calculation results of 4-error masking probability.
(b = 2, M = 6, S = 16, x = 3)

probability and corresponding N. As p becomes larger, N corresponding to maximum 4-error-masking probability becomes smaller, and the maximum value increases. Also, as p increases, the difference between $P_4(E)$ and $P_4(E) + P_6(E)$ becomes greater.

Finally, we will compare the 4-error-masking probability in our analysis to the one shown in Ref. [21]. In Ref. [21], frequency of 4-error masking was derived as the ratio of a combination that caused 4-error masking from 100 million random combinations of four errors, without regarding bit error probability. Then we also calculated the frequency of 4-error masking in a similar way. Injector matrices were randomly generated for given parameters under (R1) and (R2) restrictions. Table 3 shows the frequency of 4-error masking under $N = 1$, $q = 3$, and $S/b = 100$. Table 4 shows the frequency of 4-error masking under $b = 1$, $M = 16$, $S = 100$, and $q = 3$. The applied injector matrix might differ for analysis and simulations. Thus, our analysis did not always accurately fit the simulation-based results. However, it did have similar tendencies of masking errors.

5. Conclusions

In this paper we analyzed error-masking probability for convolutional compactors. First, we explicitly derived the error-masking probability for a given injector network by applying weight distributions of binary linear error-correcting codes. Then, we presented a fast calculation scheme for 4- and 6-error-masking probabilities. Numerical examples revealed that the calculated error-masking probabilities were about the same as those obtained through Monte-Carlo simulations, that 4-error-masking probability could approximate overall error-masking probability when bit-error probability was low, and that masking probability as a function of test length had a peak value for constant error probabilities.

Table 2. Calculation results of error masking probability. ($b = 2, M = 6, S = 16, x = 3$)

p	N (peak)	P4(E)	P4(E) + P6(E)
0.00001	6250	8.64E-15	8.64E-15
0.0001	625	8.63E-12	8.63E-12
0.001	63	8.52E-09	8.52E-09
0.01	6	7.49E-06	7.55E-06
0.1	1	1.76E-03	2.22E-03

Table 2. Calculation results of frequency of error masking. ($N = 1, q = 3, S/b = 100$)

M		b			
		1	2	4	8
16	Ref. [21]	2.40E-04	2.43E-04	2.70E-04	
	analysis	2.51E-04	2.19E-04	1.85E-04	
20	Ref. [21]	6.05E-05	6.65E-05	5.46E-05	4.05E-05
	analysis	6.83E-05	5.59E-05	5.37E-05	4.62E-05
24	Ref. [21]	4.39E-05	2.08E-05	2.02E-05	1.67E-05
	analysis	3.44E-05	2.29E-05	2.00E-05	1.24E-06
28	Ref. [21]	1.21E-05	8.17E-06	8.90E-06	7.46E-06
	analysis	9.95E-06	9.80E-06	7.56E-06	6.81E-06
32	Ref. [21]	5.13E-06	5.81E-06	3.77E-06	3.61E-06
	analysis	5.87E-06	8.12E-06	3.55E-06	3.60E-06
36	Ref. [21]	3.08E-06	2.73E-06	2.27E-06	1.17E-06
	analysis	2.55E-06	2.24E-06	2.30E-06	1.73E-06
40	Ref. [21]	1.54E-06	2.00E-06	9.70E-07	8.80E-07
	analysis	1.02E-06	1.47E-06	1.03E-06	8.68E-07
44	Ref. [21]	5.13E-07	9.69E-07	7.70E-07	5.10E-07
	analysis	5.10E-07	7.42E-07	7.96E-07	4.94E-08

Table 3. Calculation results of frequency of error masking. ($b = 2, M = 6, S = 100, q = 3$)

N	Ref. [21]	analysis
1	2.40E-04	2.51E-04
5	5.66E-05	6.17E-05
9	1.69E-05	2.82E-05
13	5.61E-06	1.51E-05
17	2.13E-06	8.70E-06

References

[1] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 2003 Edition, 2003.

[2] M. L. Bushnell and V. D. Agrawal, Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits, Kluwer, 2000.

[3] C. E. Stroud, A Designer's Guide to Built-In Self-Test, Kluwer, 2002.

[4] B. Nadeau-Dostie, Design for At-Speed Test; Diagnosis and Measurement, Kluwer, 2000.

[5] R. Rajsuman, System-on-a-Chip Design and Test, Artech House, 2000.

[6] R. David, Random Testing of Digital Circuits, Theory and Applications, Marcel Dekker, 1998.

[7] B. N. Destie, D.Burek, and A. Hassan, "Scan BVIST: A Multifrequency Scan-Based BIST Method," IEEE Design & Test of Comput., Vol. 11, No. 1, pp. 7-17, 1997.

[8] K. H. Tsai, S. Hellebrand, J. Rajski, and M. Marek-Sadowska, "A STARBIST: Scan autocorrelated random pattern generation," DA Conf., pp. 472-477, 1997.

[9] D. Das and N. A. Touba, "Reducing test data volume using external/LBIST hybrid test patterns," Proc. ITC, pp. 115-122, 2000.

[10] N. A. Touba and E. J. McCluskey, "Altering a pseudo-random BIT sequence for scan-based BIST," Proc. ITC, pp. 167-175, 1996.

[11] L. Y. Ungar and T. Ambler, "Economics of Built-in Self-Test," IEEE Design & Test of Comput., pp. 70-79, Sept.-Oct. 2001.

[12] J. Bedsole, R. Raina, A. Crouch, and M. S. Abadir, "Very Low Cost Testers: Opportunities and Challenges," IEEE Design & Test of Comput., pp. 60-69, Sept.-Oct. 2001.

[13] J. Rajski, and J. Tyszer, M. Kassab, N. Mukhaerjee, R. Thompson, K.-H. Tsai, Andre Hertwig, N. Tamarapalli, G. Mrugalski, G. Eide, and J. Quin, "Embedded Deterministic Test for Low Cost Manufacturing Test," Proc. ITC, pp. 301-310, 2002.

[14] F. Poeh., M. Beck, R. Arnold, P. Muhmenthaler, N. Tamarapalli, M. Kassab, N. Mukhaerjee, and J. Rajski, "Industiral Experimence with Adoption of EDT for Low-Cost Test without Concession," Proc. ITC, pp. 1211-1220, Sept. 2003.

[15] G. Kiefer, H. Vranken, E. J. Marinissen, and H.-J. Wunderlich, "Application of Deterministic Logic BIST on industrial circuits," Proc. ITC, pp. 105-114, 2001.

[16] B. Koenemnn, C. Barnhart, B. Keller, T. Snethen, O. Farnsworth, and D. Wheater, " A SmartBIST Variant with Guaranteed Encoding," Proc. Asian Test Symposium, pp. 325-330, 2001.

[17] C. Barnhart, V. Brunkhorst, F. Distler, W. Farnsworth, A. Ferko, B. Keller, D. Scott, B. Koenemann, and T. Onodera, "Extending OPMIDR beyond 10xScan test efficiency," IEEE Design & Test of Comput., pp. 65-73, Sept.-Oct. 2002.

[18] T. Hiraide, K. O. Boateng, H. Konishi, K. Itaya, M. Emori, H. Yamanaka, T. Mochiyama, "BIST-Aided Scan Test - A New Method for Test Cost Reduction," Proc. VTS, pp. 359-364, 2003.

[19] M. Naruse, I. Pomerantz, S. M. Reddy, and S. Kundu, "On-Chip Compression of Output Responses with Unknown Values Using LFSR Reseeding," Proc. ITC, pp. 1060-1068, 2003.

[20] S. Mitra and K. S. Kim, "X-Compact: an efficient response compaction technique for test cost reduction," Proc. ITC, pp. 311-320, 2002.

[21] J. Rajski, J. Tyszer, C. Wang, and S. M. Reddy, "Convolutional Compaction of Test Responses," Proc. ITC, pp. 745-754, 2003.

[22] M. Arai, S. Fukumoto, and K. Iwasaki, "Study on Error Masking Rate for Convolutional Compactors," Forum on Information Technology, C-025, pp. 279-280, Sept. 2004. (in Japanese)

[23] S. Lin and D. J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall. 1983.