

Design and Analysis of Multiple Weight Linear Compactors of Responses Containing Unknown Values

Thomas Clouqueur¹, Kamran Zarrineh², Kewal K. Saluja³, Hideo Fujiwara¹

¹Graduate School of Information Science, Nara Institute of Science and Technology, Japan.

²AMD Corporation, USA.

³University of Wisconsin-Madison, USA.

Abstract

Occurrence of unknown values in scan chains in response to test vectors is a common phenomenon. This paper presents a method for designing matrices for linear test output compactors by using rows of multiple weights. Compared to previously proposed compactors, the method reduces the masking caused by unknowns by an order of magnitude provided that the unknowns are non-uniformly distributed among the scan chains. Also, using multiple rather than single weight compactors increases the compaction ratio and reduces the hardware overhead. The effectiveness of multiple weight compactors is demonstrated through analysis, simulations and experiments with test response from an industrial design.

1 Introduction

Testing of VLSI circuits is challenged by the increasing test volume and number of scan cells, contrasting with the limited bandwidth available at the input and output pins. To reduce the test application time, hundreds or thousands of scan chains are required to be loaded and unloaded in parallel while only a limited amount of data can be loaded from and unloaded to the tester [14]. Such test volume reduction also cuts the memory and computing requirements for the tester.

On the input side, lossless compression techniques can be used to reduce the amount of data stored on the tester and loaded on the chip to generate the test stimuli. As test cubes generated by ATPG tools have generally a low fill rate, coding schemes can achieve high compression ratios by properly choosing the unspecified bits. The techniques for test stimuli compression include the statistical coding [11], LFSR reseeding [12] and parallel serial scan [6] schemes. These approaches have been shown to achieve up to two orders of magnitude compression.

On the output side, the data unloaded from the chip is used to decide if the chip is faulty and to localize the errors in the case of diagnosis. To perform such analysis, a signature of the test response, rather than the full data, is sufficient in general. Therefore lossy compaction techniques can be used with the desired property that errors propagate through the compactor to avoid any loss in coverage. Two types of masking, namely *error masking* and *X masking*, can prevent errors from propagating to the signature. Error masking occurs when a combination of errors cancel each other while X masking occurs when unknown values mask errors.

Unknown values are results of the circuit simulation that occur when the value of a line cannot be predetermined. X states have various sources such as bus contentions, uninitialized states, inaccurate simulation models, and timing uncertainties during at-speed testing for delay faults.

While error masking is relatively unlikely for well designed compactors, X states severely impact the observability of scan cells at the outputs of the compactors. To prevent X masking, solutions can be proposed at different levels. First, sources of X states can be removed or blocked through design modification [14]. Such a technique requires substantial design effort and it cannot be guaranteed in general to fix all sources of X states. Second, test sets can be modified to reduce the number of X states produced during test [24]. Such modification selectively fills test vectors to prevent X states to propagate to the scan cells. That extra filling has the drawback of reducing the freedom used for test set compression and also it does not guarantee that no X state reaches the scan cells. Third, a circuit can be inserted in front of the compactor to mask the unknown values that reached the scan cells, by replacing them with arbitrary but known values. Such circuit can be designed for a given test set [19] or need to receive inputs from the tester to know which position to mask [5, 17]. Also, while aiming at masking unknown values, known values can also be allowed to be masked to reduce the complexity of the masking circuitry. However, that masking of known values can result in a drop of coverage of unmodeled faults [25]. Last, compactors can be designed to reduce the impact of X states, which is one of the objectives of this study.

In this paper, we investigate a class of compactors called *multiple weight compactors* based on the linear block compactors [26]. We establish that, compared to single weight compactors, multiple weight compactors have reduced hardware overhead and higher compaction ratios. Also, we demonstrate that the two types of compactors have comparable error masking performance. Finally, we show that multiple weight compactors reduce the X masking effect given that the X states are not uniformly distributed among the scan chains. Before introducing the underlying model of linear block compactors and the multiple weight compactors in Section 3, the paper presents an overview of compaction schemes previously proposed in Section 2. The properties of the multiple weight compactors are derived in Section 4. To demonstrate the effectiveness of the compactors, experimental results are presented in Section 5. Experiments were conducted to evaluate the X masking, error masking, and the combination of X and error masking using industrial circuits. The paper concludes with Section 6.

2 Previous work

Test response compactors have three main characteristics: their space and time compaction capability, their dependence on the circuit or the test set, and their linearity or nonlinearity. A number of schemes were developed to compact a given test response [1, 2, 4, 10, 13, 23]. They can achieve high compaction ratios with minimum impact on fault coverage, but they can be designed only once the test set has been derived. Furthermore, they need to be redesigned every time the circuit or the test set is modified. Such schemes can be used for testing of IP cores as the knowledge of the circuits may not be required for compacting a given test response. Other schemes [3, 20] are only circuit dependent, thus constraining that the compactor can be designed only after the circuit is fully designed. Also the compactor needs to be modified every time the circuit is modified. This study focuses on circuit independent compactors whose design only requires coarse information about the circuit such as the number of outputs. Such compactors are usually linear to maximize the observability of the scan outputs, thus we also restrain this study to linear compactors.

Regarding space and time compaction capabilities, compactors can be divided into three classes: *infinite memory compactors*, *finite memory compactors*, and *zero memory compactors*. Each class of compactors can also combine some space compaction with time compaction, zero memory compactors being purely space compactors. When a scan output sequence is fed to an infinite memory compactor, the signature produced depends on all the scan-out values, no matter how long the sequence is. Thus, these compactors are used to achieve very high compaction ratios in the time dimension. They are frequently used, for example in BIST environments, and well developed techniques for infinite time compaction include linear feedback shift register (LFSR), multiple input shift register (MISR), and counting based techniques. Note that, while performing time compaction, a MISR also performs space compaction when the number of memory elements used is smaller than the number of scan chains. When producing signatures, compactors should propagate errors occurring at the scan outputs. Although linear compactors have good signal propagation property, some errors can cancel each other during compaction, possibly leading to a drop in fault coverage. That effect, referred to as *error masking* (or aliasing) can be marginalized in MISR designs by increasing the size of the shift register. However, the loss of information due to compaction still impacts severely the fault diagnostic based on compacted outputs. Similarly, the presence of X states in the scan chains also affects the compactor performance. The impact of X states, referred to as *X masking*, is severe because each X state in the output sequence multiplies by two the number of possible signatures corresponding to the fault free behavior. That makes signature analysis impossible when the test response sequence contains more X states than the size of the shift register. As mentioned in the introduction, methods to contain X values can be used at several levels but each method alone may not guarantee to fix all the X values or may achieve that goal only at a high cost. Therefore one objective of this work is to study how compactors can handle the X values present at the scan outputs. Since infinite memory compactors have very limited X tolerance, we focus on zero memory and finite memory compactors.

Zero memory compactors are combinational circuits that only

achieve space compaction. Some zero memory compactors, such as Saluja-Karpovsky compactors, are based on check matrices of error correcting codes [22]. Their error masking properties and diagnostic capabilities depend directly on the distance of the code corresponding to the check matrix used. In the presence of X states, these compactors can be used with post-processing of the compacted output to compare it with a set of possible compacted outputs for the fault free circuit, each possible compacted output corresponding to a different realization of the X states [18]. Other compactors named X-compactors avoid that post-processing by building compaction matrices guaranteeing that a given number of errors are directly observable, i.e. by simple comparison with the fault free compacted output, in the presence of a given number of X states at the scan output [15]. Zero memory compactors are limited in the compaction ratio attainable by the relative small number of scan chains. Indeed, much higher compaction ratio can usually be obtained by compacting more values at a time. Finite memory compactors can improve the compaction ratio of zero memory compactors by expending compaction in the time domain. When a scan output sequence is fed to a finite memory compactor of depth d , the signature produced at a given time t depends on the output values between time $t - d + 1$ and t . For example, a MISR whose content is checked and reset every d cycles is a finite memory compactor. Recently, two block compactors were proposed, namely *convolutional compactor* [8, 21] and *block compactor* [26]. They are able to provide one output compaction while preserving low error masking and X masking. Also, when compared with the X-compact scheme for equal compaction ratio, they demonstrate lower X masking probability. Another finite memory compactor produces *X tolerant signatures* by generating random rows for the linear operation and compacting in the time dimension [16].

3 Model and main idea

The compactors developed in this paper are based on block compactors. This section describes the block compactor model in detail and presents the main idea of multiple weight compactors.

3.1 Block compactor model

A block compactor of depth d compacts n scan chains into m outputs by considering blocks of time depth d . It can be described with a binary matrix with $d.n$ rows and $d.m$ columns. Each row is associated with one of the last d cells of the scan chains, the set of $d.n$ such cells forming an input data block as shown in Figure 1. The matrix compacts the input data block into an output data block of size $d.m$, which is then scanned out in d cycles through the m outputs as the next input data block is formed during regular shifting of the scan chains. The entry in row i and column j of the compaction matrix is “1” if and only if the input block cell associated with row i is connected to the XOR tree feeding the output block cell associated with column j . Otherwise, the entry is “0”. The circuit realizing the linear operation is described in Figure 2. It corresponds directly to the compaction matrix. Note that this model of data blocks is valid for memoryless compactors, such as X-compact, in which case a data block consists of scan cells lo-

cated at the scan chain outputs which are observed during a single scan-out cycle, i.e. depth $d = 1$. Note also that Figure 1 suggests that block compactors are intrusive of the scan chains. However, there exist non-intrusive designs for which the compactor inputs are merely the scan chain outputs. Of course, these designs have extra memory elements within the linear compactor.

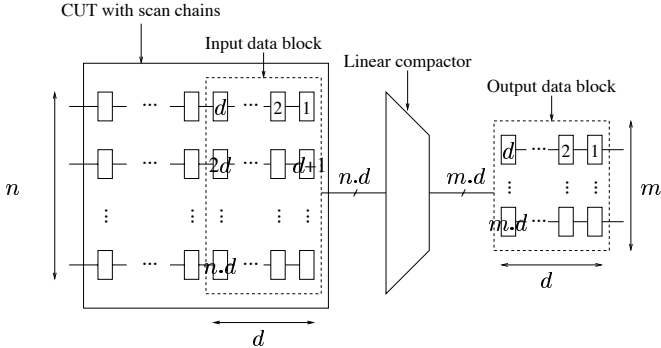


Figure 1: Block compactor architecture.

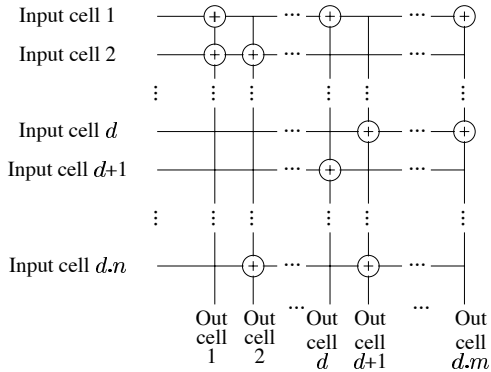


Figure 2: Linear compactor.

The error masking and X masking properties of block compactors depend essentially on the properties of their matrices. The block compactors previously proposed use matrices with rows of equal and odd weights, the weight being the number of ones in a row [15, 26]. Such single weight matrices guarantee to detect 1, 2 or any odd number of errors occurring within the same data block if all the matrix rows are different. In the presence of one X state in a data block, the matrices of equal weight guarantee detection of any single error.

3.2 Multiple weight compactors

Using multiple weights when choosing the rows of the compactor matrices can improve the compaction ratio and/or the hardware cost, and can also reduce the X masking of the compactors. From the compaction ratio point of view, considering multiple weights increases the number of different rows to choose from when building the matrix. Indeed, for a given weight w , the number of different rows of weight w is $\binom{d.m}{w}$. Furthermore, rows of differ-

ent weight are always different. If more rows can be found while maintaining the error and X masking probabilities, then more scan chains can be compacted into the same number of outputs, thus increasing the compaction ratio while keeping the same compactor performance. From the overhead point of view, the number of XOR gates needed to implement the tree feeding the compactor output i is $w_i - 1$ where w_i is the weight of the column i (assuming that XOR trees do not share logic). Therefore, overhead decreases when low weights are used in general. By using rows of multiple weights instead of single weight, one may achieve the same compaction ratio at a lower cost.

Although multiple weight compactors were considered in pioneer work on compaction [9, 22] and in compactors with random rows [16], recent schemes for block compaction in the presence of X values only use single weight matrices. From the X masking point of view, non identical rows of same weight cannot fully cover each other. Therefore, a single X value cannot mask a single error when single weight matrices are used. That property may not hold anymore for multiple weight matrices but, nevertheless, multiple weight matrices can have lower X masking on average. Considering X masking, it was observed that matrices with low weight outperformed matrices with higher weight when scan chains produced a large number of X states [21]. But the opposite observation was made when scan chains produced only few X states. Indeed, increasing the weight makes scan chains outputs propagate to more compactor outputs, and become more observable. However, increasing the weight also increases the number of compactor outputs that are made unobservable by each X state produced. Another observation is that the X state distribution among the scan chains is not uniform in general, i.e. the majority of the X states is produced by a small fraction of the scan chains. Indeed, the sources of X states are often localized within a circuit and, therefore, affect some scan chains more than others. For example, observation of some industrial circuit showed that over 90% of the X states were produced by only 10% of the scan chains [21]. We also make the same observation for the designs considered in this paper and presented in Section 5.

Therefore the main idea of this study is to build compactors using matrices with multiple weights. Rows of small weight are assigned to scan chains producing many X states, while rows of larger weight are assigned to scan chains producing fewer X states. Note that a method to build multiple weight matrices considering convolutional compactors rather than block compactors has also been proposed recently [7]. Also, the scheme proposed for X-tolerant signature analysis allows rows of multiple weights but it does not assign the rows according to the X distribution.

4 Compactor properties

In this section, we study the general properties of multiple weight block compactors. In particular, we prove that, in the absence of X states, the error masking is identical for the compactors with rows of multiple weights and of single weight. The diagnostic capability is also proven identical for both types of compactors. In the presence of X states, we describe the X masking properties of multiple weight compactors. Finally, we study the achievable compaction ratio and overhead of multiple weight compactors.

4.1 Properties in the absence of X states

First, we look at cases for which no X states are produced in a data block.

Property 1: A multiple weight compactor detects errors from any one or two scan cells in a data block, provided that no scan cell produces X states in the data block, if the rows of its matrix are nonzero and different.

Property 1 simply states that any error can propagate to the compacted outputs if the rows are nonzero, and two errors cannot cancel each other if two rows cannot add up to zero, i.e. two rows are different. That property corresponds to well known results from error correcting code theory since a check matrix with nonzero and different rows corresponds to a Hamming code (shortened or not) of distance three [22].

Property 2: A multiple weight compactor detects errors from any one, two, or k scan cells, with k odd, in a data block, provided that no scan cell produces X states in the data block, if the rows of its matrix are nonzero, different, and all the entries of one column of its matrix are "1".

Such a matrix corresponds to an augmented Hamming code with overall parity check, which has distance four. Therefore, detection of 1, 2, and any odd number of errors is guaranteed. Note that this property is identical to the one derived for the single weight compactors previously proposed. In general, multiple weight matrices can be built using check matrices of error correcting codes of any given distance. The code needs to be chosen so that its check matrix has enough rows of each desired weight. Code with overall parity check require that all the scan cells are connected to one of the compactor outputs, thus creating a very deep XOR tree. To solve timing issues that such a design may create, it is possible to pipeline the linear compactor, thus introducing latency and extra overhead. Therefore, the following approach using only odd weights may be preferred and it is the scheme used in the remainder of this paper.

Property 3: A multiple weight compactor detects errors from any one, two, or k scan cells, with k odd, in a data block, provided that no scan cell produces X states in the data block, if the rows of its matrix are nonzero, different, and have odd (but not necessary equal) weight.

Adding the condition that all the weights are odd gives again identical detection performance for multiple weight and single weight compactors. That property derives from the fact that errors cancel in pairs, therefore if an odd number of errors are propagated an odd number of times, they cannot fully cancel out. Note that detection of four errors can also be guaranteed by choosing the rows in a manner such that four rows cannot add up to zero [21]. We will demonstrate such compactors in the following section.

Regarding error diagnosis ability, the following property holds.

Property 4: A multiple weight compactor uniquely identifies any single erroneous cell in a data block, provided that no scan cell produces X states in the data block, if the rows of its matrix are nonzero and different.

Property 4 also relates to error correcting code theory, the matrix corresponding to a code of distance three. The result of Property 4 is again equivalent to the diagnostic capability of single weight compactors.

4.2 Properties in the presence of X states

When using rows of multiple weights, it is possible that one row r_1 with high weight covers a row r_2 with low weight, i.e. the entry in any column of r_1 is "1" whenever the entry in the same column of r_2 is "1". In such a case, an X state produced by the scan cell associated with r_1 will make the scan cell associated with r_2 unobservable. Therefore, no error detection is guaranteed in the presence of X states for multiple weight compactors when considering all the scan cells together. Note that single error detection in the presence of single X state is guaranteed by single weight compactors. Nevertheless, some properties hold for multiple weight compactors when considering groups of scan cells separately. For simplicity of the presentation, we assume that only two different weights are used in the compactor matrix, namely w_l for weight low and w_h for weight high.

Property 5: A multiple weight compactor using weights w_l and w_h detects any single erroneous cell in a data block, provided that an X state is produced by one of the scan cells associated with a row of low weight in the data block, if the rows of its matrix are nonzero and different.

Indeed, a row of low weight cannot cover another row of low weight (since the rows are different) and cannot either cover a row of large weight.

Property 6: A multiple weight compactor using weights w_l and w_h detects any single erroneous cell in a data block associated with a row of high weight, provided that an X state is produced by any scan cell in the data block, if the rows of its matrix are nonzero and different.

Indeed, a row of high weight cannot be covered by any other row in the matrix.

Property 7: A multiple weight compactor using weights w_l and w_h with $w_h > 2w_l$ detects any single erroneous cell in a data block associated with a row of high weight, provided that two X states are produced by any two scan cells associated with rows of low weight in the data block, if the rows of its matrix are nonzero and different.

Indeed, the condition $w_h > 2w_l$ prevents any two rows of low weight to cover a row of high weight. Note that such a property does not hold for single weight compactors.

The intrinsic properties of the matrices presented here only partially indicate the X masking performance of the compactors. Although multiple weight matrices do not guarantee to detect any error in the presence of one X value, it can still outperform single weight compactors when X values are nonuniformly distributed among the scan chains. Indeed, it has weaker properties for unlikely events, e.g. occurrence of an X in a scan chain associated with a row of high weight. But it has stronger properties for likely events, e.g. occurrence of Xs in two scan chains associated with rows of low weight. In Section 5, we demonstrate the effectiveness of multiple weight compactors in reducing X masking by evaluating the X masking probability in various cases including some industrial scenarios.

4.3 Compaction ratio and cost analysis

The compaction capability of a compactor can be measured by the maximum number of scan chains supported for a given number of

outputs m and a given compactor depth d . Let $w_i, 1 \leq i \leq p$ be p different odd weights used to build the rows of the compactor matrix. Then, the maximum number of scan chains $N(d, m)$ that can be supported, is given by the following equation.

$$N(d, m) = \left\lfloor \frac{1}{d} \sum_{i=1}^p \binom{d \cdot m}{w_i} \right\rfloor \quad (1)$$

The maximum number of scan chains supported by a single weight compactor would be only one of the terms in the sums specified. Therefore, the maximum number of scan chains supported by the multiple weight compactor is substantially greater. That property can be used to achieve higher compaction ratio for a given output size m . Note however that single weight block compactors are able to compact any number of scan chains into a single output by increasing the depth d . In that sense, compaction ratio can be improved for a given d . Also, multiple weight compactors can reduce the depth d necessary to achieve a given compaction ratio. In that sense, the complexity can be reduced for a given compaction ratio. More importantly, these benefits from multiple weight compactors are obtained while keeping the same properties for error masking and we will also show that the X masking probability decreases.

m	3	4	5	6	7	8
max n single weight	10	28	126	396	1716	5720
max n mult. weight	16	64	256	1024	4096	16384

(a) $d = 2$

m	2	3	4	5	6
max n single weight	6	42	264	2145	16206
max n mult. weight	10	85	682	5461	43690

(b) $d = 3$

Table 1: Maximum number of scan chains for varying number of outputs and depth.

Table 1 shows the maximum number of scan chains supported by single and multiple weight compactors for two different depths ($d = 2, d = 3$) and for varying number of outputs m . In the case of single weight, the maximum number of supported scan chains is reached for weight $w = \lfloor \frac{d \cdot m}{2} \rfloor$. For multiple weight, the maximum is reached when using all the odd weights ranging from 1 to $d \cdot m$. The table shows that for a given depth d and output size m , the multiple weight compactor can support substantially more scan chains. Also, for a given number of scan chains n and output size m , the depth can be reduced when using multiple weight compactors. For example, to compact 60 scan chains into 4 outputs, depth of two is sufficient for multiple weights whereas a depth of at least three is required for single weight compactors.

With respect to overhead, multiple weight compactors can reduce the number of XOR gates needed to realize the linear operator. Assuming that the XOR trees corresponding to the m outputs

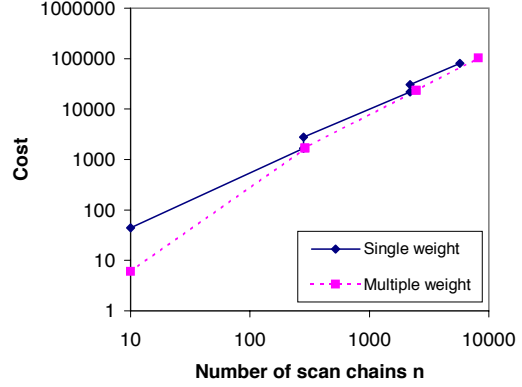


Figure 3: Cost for varying number of scan chains with $d = 2, m = 8$.

do not share logic, the total number of XOR gates, say C , is given by the following expression.

$$C = \sum_{i=1}^{d \cdot n} w(i) - m \cdot d \quad (2)$$

where $w(i)$ is the weight of the row associated with the input cell i . Starting with a given compactor of single weight w_0 , the cost can be reduced by replacing rows of weight w_0 with rows of weight w_1 with $w_1 < w_0$. Figure 3 shows the cost C as a function of the number of scan chains n using single and multiple weight compactors. The figure shows clearly that using multiple weights always results in lower cost.

5 Experimental results

The properties described in the previous section indicate that multiple weight compactors have the same guarantees of error detection in the absence of X values but different guarantees in the presence of X values. In this section, we further study the error and X masking characteristics of multiple weight compactors by evaluating the average masking probability. First, we measure the X masking probability assuming non uniform distribution of X values among the scan chains. Second, we evaluate the error masking probability when four errors occur simultaneously in the input data block. Last, we provide a study of the combination of error and X masking of an industrial design.

5.1 X masking

5.1.1 Experimental evaluation

The properties described in the presence of X states show that in some cases more X states can be supported by multiple weight compactors than by single weight compactors, and vice versa. Therefore, it is essential to evaluate the average X masking probability in order to compare the schemes. The X masking probability depends on the distribution of X values in the test response. As argued earlier, that distribution was observed to be non uniform among the scan chains of industrial designs. In this experiment,

we assume that each input scan cell i has a stationary probability p_i of producing X values. To model the non uniformity, we allow $p_i \neq p_j$ whenever scan cell i and j do not belong to the same scan chain. We evaluate the X masking probability by measuring the average number of scan cells masked, including the scan cells producing X values, when each scan cell i produces an X value with probability p_i . The matrices used have 1600 rows and 16 columns so they can be used to compact $\frac{1600}{d}$ scan chains into $\frac{16}{d}$ outputs with a depth d . We considered four different distributions of X values among the scan chains.

- Scenario S₁: the X values are uniformly distributed among the scan chains and single weight $w = 7$ is used.
- Scenario S₂: 90% of the X values are produced by 10% of the scan chains. Two weights are used, $w = 3$ for rows associated with scan cells belonging to scan chains producing many X values and $w = 7$ for the remaining rows.
- Scenario S₃: scan chains are divided into four different groups described in the following table. For each group, the table specifies the size of the group as a percentage of the total number of scan chains, the percentage of the total number of X values produced by the group, and the weight of rows associated with scan cells belonging to scan chains from the group.

Group #	1	2	3	4
Size (%)	1.0	3.0	6.0	90.0
X produced (%)	50.0	20.0	20.0	10.0
Weight	1	3	5	7

For example, Group 1 consists of 1% of the scan chains which produce 50% of the X values and have associated weight $w = 1$.

- Scenario S₄: scan chains are divided into five groups described in the following table. Note that the scan chains in the last group do not produce any X values and a relatively high weight is associated with these scan chains.

Group #	1	2	3	4	5
Size (%)	1.0	3.0	6.0	20.0	70.0
X produced (%)	50.0	20.0	20.0	10.0	0.0
Weight	1	3	5	7	11

The X masking probability is evaluated for the four scenarios using Monte Carlo simulation. For each scenario, the average portion of scan cells producing X states considering all the scan cells (named X probability) varies from 0.01% to 1%. For every scan cell i , the probability p_i of producing X values is computed according to the X probability p_X and the group of scan chains to which cell i belongs. Precisely, $p_i = \frac{a_i}{b_i} \times p_X$ where a_i is the percentage of X values produced by the group to which cell i belongs, and b_i is the relative size of that group. For example, for scan cell i belonging to group 1 of scenario S₄ and for an X probability of 1%, $p_i = \frac{.5}{.01} \times .01 = .5$ so that scan cell i produces an X value with 50% probability. To build the matrices used in the four scenarios, the desired number of rows of a given weight are

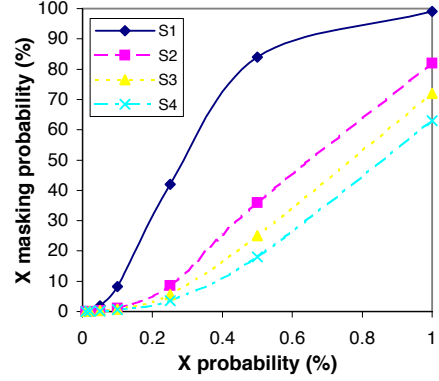


Figure 4: X masking probability for four scenarios.

randomly picked among all the possible rows of that weight while making sure that all the rows are different. Repeating the experiment with different randomly picked matrices showed that the X masking probability is robust to the selection process. However, note that other row selection process, such as following the lexicographic order, may result in different X masking probability. Such selection scheme is not considered in this paper.

Figure 4 shows the X masking probability for the four scenarios when the X probability varies from 0.01% to 1%. As expected, the graph indicates that the X masking probability increases with the X probability. Comparing the different scenarios, we see that X masking is highest for scenario S₁, and then reduces for scenarios S₂, S₃ and S₄. Although scenario S₁ corresponds to uniform distribution of X values among the scan chains, the X masking probability with single weight $w = 7$ for the non uniform distributions of scenarios S₂, S₃ and S₄ was also considered and found to be identical to that of S₁. Indeed, the rows of identical weight being picked and ordered randomly, the distribution of X values does not affect the X masking probability. Therefore, scenario S₁ gives the performance of single weight compactors for any X value distribution. The graph shows the effectiveness of increasing the number of weights in reducing the X masking. When only rough knowledge of the X value distribution is known as in scenario S₂, the X masking effect can already be reduce by using only two different weight. As more detailed information is known about the X value distribution, more weights can be used to further reduce X masking. In particular, using rows of higher weights for scan chains that produce no X values reduces X masking even further.

5.1.2 Analytical expression

An analytical expression of the X masking probability can help find the optimal weight selection for a given X value distribution. Here we derive the X masking probability $P_{X \text{ masking}}$ of single weight compactors assuming that X values are uniformly distributed among the scan chains. We consider one row i of the matrix that contains a total of n rows. Let $P_c(j)$ be the conditional probability that row i is masked given that j scan cells contain X values. Also, let $Q(j)$ be the probability that j scan cells contain X values. Then,

$$P_{X \text{ masking}} = \sum_{j=1}^{n-1} P_c(j) \cdot Q(j) \quad (3)$$

We can derive $Q(j)$ as a function of the X probability p_X :

$$Q(j) = \binom{n-1}{j} p_x^j (1-p_x)^{n-1-j}$$

Let $r(j, k)$ be the probability that j rows cover a given set of k bits out of m . Then, $P_c(j) = r(j, w)$ where w is the weight of the rows. Let $s(l, k)$ be the probability that one row covers exactly a given set of l bits out of k . By considering that a single row out of j can cover from 0 to k bits, leaving the remaining bits to be covered by the remaining $j-1$ rows, we derive the following recursive relation for $r(j, k)$:

$$r(j, k) = \sum_{l=0}^{k-1} \binom{k}{l} s(l, k) r(j-1, k-l) + r(1, k) \quad (4)$$

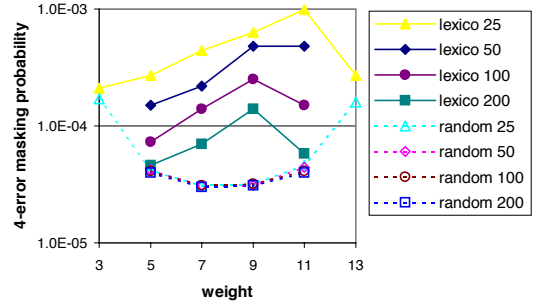
Furthermore, $r(1, k)$ and $s(l, k)$ can be found by counting the number of rows of weight w that cover a given set of k bits and the number of rows that cover exactly l out of k bits, with respect to the total number of rows of weight w but different from row i .

$$r(1, k) = \frac{\binom{m-k}{w-k}}{\binom{m}{w} - 1}, \quad s(l, k) = \frac{\binom{m-k}{w-l}}{\binom{m}{w} - 1}$$

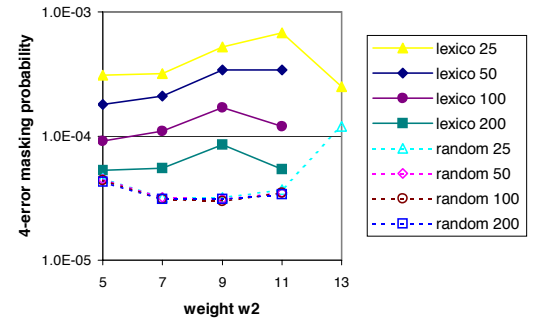
By using the recursive expression in (4) to find $p_c(j)$, we evaluated the X masking probability given by Equation 3 for the X probability varying from 0.01% to 1%. We found that the experimental results corresponding to the scenario S_1 formerly presented match the analytical results since they are within the confidence level of the Monte Carlo simulation used. However, the expression derived here for the X masking probability may be impractical for determining the optimal weight due to its recursive nature. Also, it only considers single weight compactors and uniform X value distribution. Finding useful and general expression for the X masking probability remains an open problem.

5.2 Error masking

We showed in Section 4 that single and multiple weight compactors have identical error detection properties in absence of X values when only odd weights are used. For both types of compactors, errors of multiplicity four, six, eight, etc, can be masked and we investigate here if the error masking probability increases when using multiple rather than single weight compactors. When evaluating the error masking probability, the different multiplicities of errors bear different importance. Indeed, the probability of masking of six, eight, or more even number of errors has been reported to be much smaller than the probability of 4-error masking [21]. Furthermore, it is much less likely that a defective chip is detected only through errors of multiplicity six or more rather than multiplicity four [15]. Therefore, we focus here on 4-error masking. We conduct two experiments to compare 4-error masking for single and multiple weight compactors. In the first experiment, we measure the probability of 4-error masking and in the second experiment we evaluate the compaction ratio reachable when building matrices that eliminate 4-error masking.



(a) Single weight.



(b) Two weights ($w_1=3$, w_2 varies).

Figure 5: 4-error masking probability with $d.m = 16$.

5.2.1 Masking probability of 4 errors

The 4-error masking probability is evaluated through Monte Carlo simulation by injecting random sets of four errors in each data block and checking if the errors cancel each other through compaction. We consider various matrices that have different number of rows, columns and different building scheme. In the first experiment, matrices have 16 columns and varying number of rows to achieve compaction ratios ranging from 25 up to 200. Two schemes are considered to build the matrices. In the first scheme, rows are chosen following a lexicographic order. For example, if rows are six bit long with weight three, the order is 111000, 110100, 110010... In the second scheme, rows are chosen randomly.

Figure 5(a) shows the 4-error masking probability for single weight matrices lexicographically and randomly picked, the weight varying from 3 up to 13. We observe that the two schemes for building matrices result in curves of opposite shapes. Matrices built following the lexicographic order have higher 4-error masking probability because of the high correlation between the different rows. Indeed, the rows picked tend to have entry "1" in their first few positions and it is even more true as the total number of possible rows of weight w , $\binom{16}{w}$, is large, which happens for $w = 8$. On the other hand, randomly picked matrices have lowest 4-error masking when the weight is around 8. For compaction ratio of 25 and weight of three, the two types of matrices perform almost equally since most of the rows of weight three are being used (400 out of 560 possible) so that the selection process has less impact. Similar observation holds for weight 13 and compaction ratio of 25. Note that there are not enough rows of weight

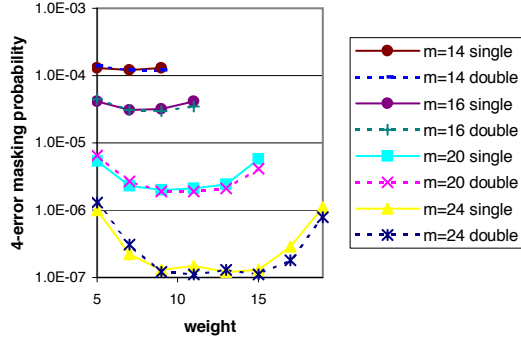


Figure 6: 4-error masking probability for 100x compaction.

3 or 13 to achieve compaction ratio of 50 and higher.

We run a similar experiment with matrices using two different weights. In each case, ten percents of the rows have weight three and the remaining rows have a higher weight that ranges from 5 to 13 in the experiment. The results presented in Figure 5(b) show a similar trend than for single weight matrices. We conclude from the results that lexicographically picked matrices result in higher 4-error masking probability. We also observe that the compaction ratio has a very small impact on the 4-error masking probability for randomly built matrices. Finally, we observe by comparing the two graphs that single and double weight matrices result in comparable 4-error masking probability.

Low weights	High weight				
	5	7	9	11	13
\emptyset (single)	4.1e-5	3.1e-5	3.2e-5	4.1e-5	
1	4.1e-5	3.1e-5	3.1e-5	4.0e-5	
1,3	5.9e-5	3.6e-5	3.2e-5	3.5e-5	
1,3,5		4.3e-5	3.6e-5	3.5e-5	4.9e-5
1,3,5,7			3.4e-5	3.3e-5	3.6e-5
1,3,5,7,9				3.2e-5	3.2e-5
1,3,5,7,9,11					3.2e-5

Table 2: 4-error masking probability for varying number of weights

To further compare single and multiple weight matrices, we evaluated the 4-error masking probability for a compaction ration of 100 and number of columns varying from 14 up to 24. Only randomly built matrices are considered. The weight of single weight matrices varies from 5 to 19. Multiple weight matrices have ten percents of the rows of weight three and the remaining rows of a higher weight that ranges from 5 to 19. The results are presented in Figure 6 and they show that single and multiple weight matrices result in almost identical 4-error masking probability.

Finally, we evaluated the 4-error masking probability while increasing further the number of weights used in the matrices. We built matrices with 1600 rows and 16 columns with up to seven different weights. In each case, the matrices have the same number of rows of each weight except for rows of weight one since there are only 16 such rows. For matrices with p weights, we set the first $p - 1$ weights to be the lowest possible, i.e. 1, 3, ..., $2p-3$ and the last weight varies.

The 4-error masking probabilities are shown in Table 2. The first column of the table specifies the low weights used and the highest weight varies from 5 to 13. The results show that the 4-error masking probability remains at the same level when increasing number of weights are used.

5.2 Matrices free of 4-error masking

Another measure of the impact of multiple weights on error masking is the compaction ratio achievable with matrices that are free of 4-error masking. To build such matrices, let all the different rows of all desired weights form a set S_p . The matrix building process consists of randomly picking rows in S_p one at a time. At each step, the row picked is moved to the set of chosen rows S_c , then the linear sum of all the combinations of three rows in S_c is computed and the sum obtained is rejected from S_p if it belongs to S_p . That process was initially proposed to build 4-error masking free matrices of single weight [21]. We implemented it for building matrices of multiple weight by modifying the random selection of rows from S_p so that every weight is as likely to be picked.

Single weight		Multiple weight	
weight	# rows	weights	# rows
3	103	1-3	82
5	276	1-5	232
7	425	1-7	381
9	489	1-9	472
11	497	1-11	504
13	492	1-13	516

Table 3: Number of rows in 4-error masking free matrices for $m = 22$.

Table 3 shows the number of rows found when building 4-error masking free matrices of single and multiple weight. For each row in the table, we consider a given weight w for the single weight matrix and a weight range from 1 to w (and only odd weights) for the multiple weight matrix. That choice guarantees that the multiple weight matrix can be realized with less hardware than the single weight matrix. We observe that for small weights, more rows can be found with a single weight matrix. However, the highest compaction ratio is obtained with a multiple weight matrix. Space limitation prevents us from presenting more results here but we observed similar results with various numbers of columns.

5.3 Combination of X and error masking

The combination of X and error masking is evaluated using the X distribution in the test response of a microprocessor design. The design has 4.4M gates and 147K flip flops organized in 40 scan chains of maximum length 3853. The test set consists of 5814 test patterns and the test response contains 2.1% of X values. We considered various scan chain configurations by breaking the original scan chains into s smaller scan chains. Figure 7 shows the distribution of X values among the scan chains for $s = 1, 4$ and 10. The lowest curve ($s = 1$) corresponds to the original configuration with 40 scan chains and the other curves correspond to configurations with 160 ($s = 4$) and 400 ($s = 10$) scan chains.

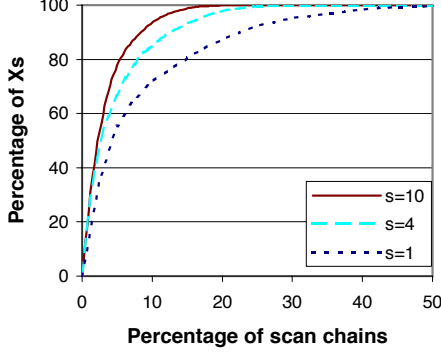


Figure 7: Distribution of X values among the scan chains.

As assumed throughout this paper, we observe that most of the X values are produced by only a few scan chains and that non uniformity increases with the number of scan chains considered. For the configuration with 400 scan chains, 10% of the scan chains produce 95% of the X values and 80% of the scan chains do not produce any X value.

The experiment consists in injecting errors with a given probability in the test response and measuring how many of these errors are unobserved at the output of the compactor. As one block of cells is compacted at a time, we consider that if at least one bit in the compacted block is erroneous, all the errors within the input block are observed. Indeed, detection of one error is sufficient to detect a faulty device. On the other hand, if all the bits in a compacted block are non erroneous or X, we consider that all the errors within the input block are masked. We first study the original scan configuration with 40 scan chains and consider three compactor settings.

- Compactors for setting A_1 have depth $d = 1$, number of outputs $m = 8$ for a compaction ratio of 5.
- Compactors for setting A_2 have depth $d = 2$, number of outputs $m = 5$ for a compaction ratio of 8.
- Compactors for setting A_3 have depth $d = 3$, number of outputs $m = 3$ for a compaction ratio of 13.

For each setting, single weight compactors use weight $w = 5$ and multiple weight compactors use weights $w = 1, 3$ and 5 . The percentage of errors masked is presented in Figure 8 as a function of the error probability that varies from 0.01% up to 1.0%. We observe that the percentage of errors masked is almost constant as the error probability varies. Also, we observe that the use of multiple weights reduces the percentage of errors masked by a factor of around 10 for setting A_1 , 5 for setting A_2 , and 3 for setting A_3 .

Next, we create three configurations, namely B_1 , B_2 and B_3 , by subdividing the 40 scan chains into 160 ($s = 4$), 400 ($s = 10$), and 800 ($s = 20$) scan chains respectively and study the error masking for each configuration. The compactor settings used for each configuration are described in Table 4.

Again, we measure the percentage of errors masked as the error probability varies from 0.01% up to 1.0%, as presented in Figure 9. We observe that the masking probability drops drastically as the error probability increases. Indeed, for higher error probability,

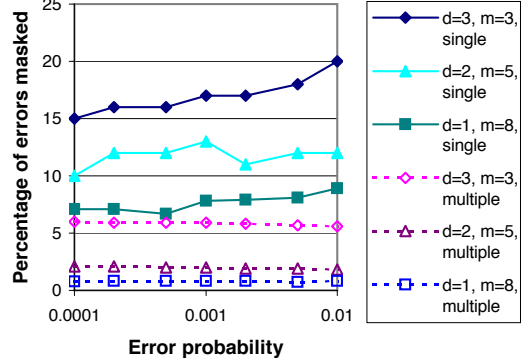


Figure 8: Error masking for original scan chain configuration.

conf.	d	m	comp. ratio	weights used	
				single	multiple
B_1	1	20	8	3	odd from 1 to 11
B_2	1	30	13	3	odd from 1 to 17
B_3	1	40	20	3	odd from 1 to 21

Table 4: Compactors settings

the error patterns in the input block are more likely to have higher multiplicity and such error patterns have lower masking probability. Note that the faults hardest to detect typically produce patterns with very few errors and these patterns are the most likely to be masked. Therefore, lower error probabilities are of more interest when evaluating masking. We observe that multiple weight compactors reduce the percentage of errors masked by a factor around 7 for configuration B_1 , 10 for configuration B_2 , and 10 for configuration B_3 for error probabilities smaller than 0.5%. That reduction factor increases because the X distribution is less and less uniform as the number of subdivisions increases. Note that we measured the X masking of single weight compactors for all possible single weights and only the best performance is reported here. On the other hand, not every multiple weight assignment could be evaluated and better performance may be achievable. A scheme for finding the optimal multiple weight assignment remains to be developed.

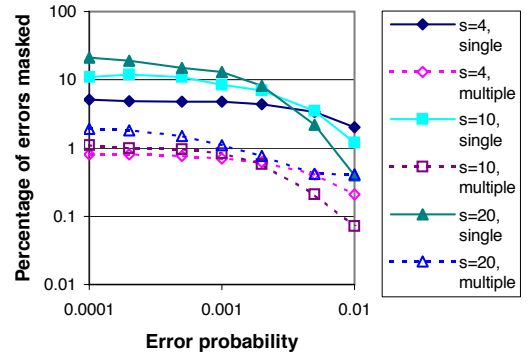


Figure 9: Error masking for subdivided scan chains.

6 Conclusion

This paper presented a scheme for compacting test responses containing X values. The linear compactors proposed use matrices with rows of multiple weights and were compared to the previously proposed single weight compactors. They were shown to reduce the hardware overhead and increase the compaction ratio of the compactor. Also, the X masking of multiple weight compactors is reduced by properly assigning the rows according to the distribution of X values among the scan chains. Meanwhile, the error masking of multiple and single weight compactors remain equivalent. The effectiveness of multiple weight compactors was demonstrated through analysis, simulations and experiments conducted with industrial designs.

Multiple weight compactors were shown to reduce X masking even when very coarse information is known about the X value distribution among the scan chains. Therefore, they can perform circuit independent compaction. Furthermore, we showed that X masking can be further reduced when very detailed information is known about the X value distribution. Therefore, multiple weight compactors are most effective for circuit or test set dependent compaction, e.g. in system on chip environment. High performance multiple weight compactors can also be implemented for circuit independent compaction by using programmable logic.

The performance of multiple weight compactors strongly depends on the weight assignment and a systematic mechanism for such assignment is under investigation. Also, the masking of non uniformly distributed errors remains to be investigated.

Acknowledgments

We would like to thank AMD DFT teams in Sunnyvale, CA and Boxboro, MA for providing us with test pattern data of the designs used in this paper. This work was supported in part by 21st Century COE Program and in part by Japan Society for the Promotion of Science (JSPS) under Grants-in-Aid for Scientific Research B(2)(No. 15300018) and the grant of JSPS Research Fellowship (No. L04509).

References

- [1] B. Bhattacharya, A. Dmitriev, M. Gossel, and K. Chakrabarty, "Synthesis of single-output space compactors for scan-based sequential circuits," *IEEE Trans. on CAD*, vol. 21, pp. 1171–1179, October 2002.
- [2] K. Chakrabarty, "Zero-aliasing space compaction using linear compactors with bounded overhead," *IEEE Trans. on CAD*, vol. 17, pp. 452–457, May 1998.
- [3] K. Chakrabarty and J. P. Hayes, "Zero-aliasing space compaction of test responses using multiple parity signatures," *IEEE Trans. on VLSI Systems*, vol. 6, pp. 309–313, June 1998.
- [4] K. Chakrabarty, B. T. Murray, and J. P. Hayes, "Optimal zero-aliasing space compaction of test responses," *IEEE Trans. on Computers*, vol. 47, pp. 1171–1187, November 1998.
- [5] V. Chickermane, B. Foutz, and B. Keller, "Channel masking synthesis for efficient on-chip test compression," in *Proc. ITC*, pp. 452–461, 2004.
- [6] I. Hamzaoglu and J. H. Patel, "Reducing test application time for full scan embedded cores," in *Proc. of FTCS*, pp. 260–267, 1999.
- [7] 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. Symp. on Defect and Fault Tolerance in VLSI Systems*, pp. 298–305, 2004.
- [8] Y. Han, Y. Xu, H. Li, X. Li, and A. Chandra, "Test resource partitioning based on efficient response compaction for test time and tester channels reduction," in *Proc. ATS*, pp. 440–445, 2003.
- [9] M. Hsiao, "A class of optimal minimum odd-weight-column sec-ded codes," *IBM Journal of Research and Development*, vol. 14, pp. 395–401, July 1970.
- [10] A. Ivanov, B. Tsuji, and Y. Zorian, "Programmable bist compactors," *IEEE Trans. on Computers*, vol. 45, pp. 1393–1404, December 1996.
- [11] A. Jas, J. Ghosh-Dastidar, and N. Toubia, "Scan vector compression/decompression using statistical coding," in *Proc. VTS*, pp. 114–120, 1999.
- [12] B. Koenemann, "Lfsr-coded test patterns for scan designs," in *Proc. European Test Conference*, pp. 237–242, 1991.
- [13] Y. K. Li and J. P. Robinson, "Space compression methods with output data modification," *IEEE Trans. on CAD*, vol. 6, pp. 290–294, March 1987.
- [14] E. McCluskey, D. Burek, B. Koenemann, S. Mitra, J. Patel, J. Rajski, and J. Waicukauski, "Test data compression," *Design & Test of Computers, IEEE*, vol. 20, pp. 76–87, Mar-Apr 2003.
- [15] S. Mitra and K. S. Kim, "X-compact an efficient response compaction technique," *IEEE Trans. on CAD*, vol. 23, pp. 421–432, March 2004.
- [16] S. Mitra, S. S. Lumetta, and M. Mitzenmacher, "X-tolerant signature analysis," in *Proc. ITC*, pp. 432–441, 2004.
- [17] M. Naruse, I. Pomeranz, S. M. Reddy, and S. Kundu, "On-chip compression of output responses with unknown values using lfsr reseeding," in *Proc. ITC*, pp. 1060–1068, 2003.
- [18] J. H. Patel, S. S. Lumetta, and S. M. Reddy, "Application of saluja-karpovsky compactors to test responses with many unknowns," in *Proc. VTS*, pp. 107–112, 2003.
- [19] I. Pomeranz, S. Kundu, and S. M. Reddy, "On output response compression in the presence of unknown output values," in *Proc. DAC*, pp. 255–258, 2002.
- [20] B. Pouya and N. A. Toubia, "Synthesis of zero-aliasing elementary-tree space compactors," in *Proc. VTS*, pp. 70–77, 1998.
- [21] J. Rajski, J. Tyszer, C. Wang, and S. M. Reddy, "Convolutional compaction of test responses," in *Proc. ITC*, pp. 745–754, 2003.
- [22] K. K. Saluja and M. Karpovsky, "Testing computer hardware through data compression in space and time," in *Proc. ITC*, pp. 83–88, 1983.
- [23] O. Sinanoglu and A. Orailoglu, "Efficient construction of aliasing-free compaction circuitry," *IEEE Micro*, vol. 22, pp. 82–92, Sept-Oct 2002.
- [24] H. Tang, C. Wang, J. Rajski, S. M. Reddy, J. Tyszer, and I. Pomeranz, "On efficient x-handling using a selective compaction scheme to achieve high test response compaction ratios," in *Proc. VLSI Design*, pp. 59–64, 2005.
- [25] Y. Tang, H.-J. Wunderlich, H. Vranken, F. Hapke, M. Witke, P. Engelke, I. Polian, and B. Becker, "X-masking during logic bist and its impact on defect coverage," in *Proc. ITC*, pp. 442–451, 2004.
- [26] C. Wang, S. M. Reddy, I. Pomeranz, J. Rajski, and J. Tyszer, "On compacting test response data containing unknown values," in *Proc. ICCAD*, pp. 855–862, 2003.