X-TOLERANT SIGNATURE ANALYSIS

Subhasish Mitra
Intel Corporation
Folsom, CA
subhasish.mitra@intel.com

Steven S. Lumetta
Dept. of ECE
Univ. of Illinois, Urbana-Champaign
steve@crhc.uiuc.edu

Michael Mitzenmacher
Dept. of EECS
Harvard University
michaelm@eecs.harvard.edu

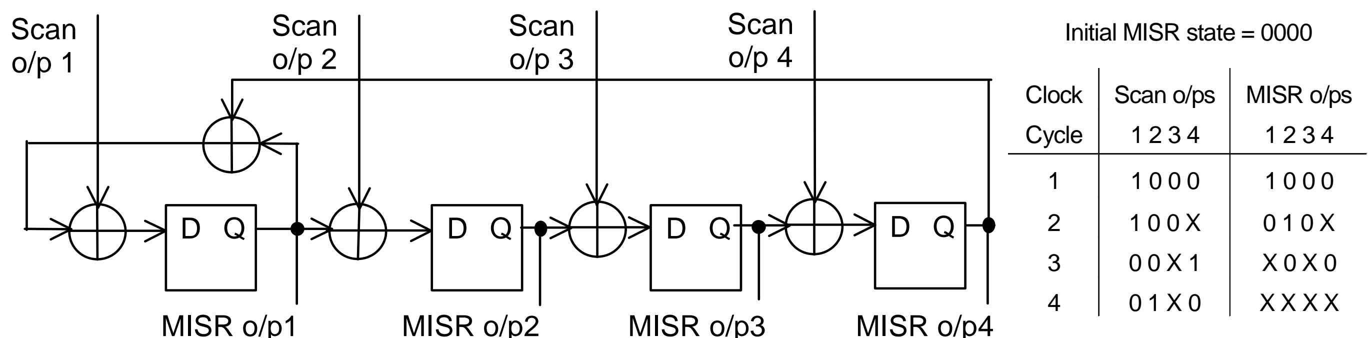
Abstract

Stochastic coding is used to design X-tolerant signature analyzers that can detect defective chips even in the presence of unknown logic values (X's). These signature analyzers can be used for Built-In-Self-Test applications and test data compression. Application of this technique to industrial designs shows that thousands of X's can be tolerated while reducing test response data volume by 50 to 2,000 times compared to traditional scan, with practically no impact on test quality.

1. Introduction

Scan Design for Testability (DFT) techniques have become *de facto* test standards in the industry. The major contributors to scan test cost are: (1) test data volume, which translates to tester memory requirement; and (2) test time, which also translates to the maximum number of flip-flops in a scan chain. The number of scan chains is constrained by the number of scan channels on the tester, the number of pins available for scan purposes, and also scan routing. For large designs with several millions of logic gates, the cost of traditional scan has become prohibitively high [McCluskey 03]. Built-In-Self-Test (BIST) and test compression techniques target scan test cost reduction by obtaining massive reductions in test data volume and test time. Any such solution consists of two basic components – test stimulus compression and test response compaction. Depending on the situation and characteristics of a design, either or both of them may be employed. The major differences between BIST and test compression is that the former requires minimal support of external test equipment. Hence, BIST is often described as the ultimate test compression technique. This paper focuses on signature analyzer designs for test response compaction using stochastic coding. These signature analyzers can be used for both BIST and test compression purposes. Unlike traditional BIST, these signature analyzers are tolerant to unknown logic values (X's) in test responses with practically no impact on test quality. Unlike conventional test compression approaches, these signature analyzers require minimal external tester support.

Classical signature analyzers such as Multiple Input Signature Registers (MISRs) are probably the best response compactors. The major problem with classical signature analysis is that the signature can be corrupted by signals whose logic values are not known during fault-free simulation, also called X's. Sources of X's include un-initialized and uncontrollable sequential elements, bus contention, floating buses, interactions among multiple clock domains, and modeling inaccuracies. Figure 1.1 illustrates the problem. The outputs of four scan chains are connected to the inputs of the MISR. The initial MISR state is 0000. The states of the MISR during the first four clock cycles are shown in Fig. 1.1. Unknown logic values (X's) appearing at scan chain outputs corrupt the MISR contents. After four clock cycles, the expected MISR signature obtained from fault-free simulation consists of all X's. Signature bits whose expected values are X's are ignored during comparison of the expected signature with the actual signature. In this example, no comparison can be made. Note that, for this example, there are eight possible expected MISR signatures since there are three X's entering the MISR, and each X may be 0 or 1.



Clock Cycle	Scan o/ps 1 2 3 4	MISR o/ps 1 2 3 4
1	1 0 0 0	1 0 0 0
2	1 0 0 X	0 1 0 X
3	0 0 X 1	X 0 X 0
4	0 1 X 0	X X X X

Figure 1.1. MISR example.

0-7803-8580-2/04 $20.00 Copyright 2004 IEEE

Sources of X's must be minimized by employing DFT techniques. However, it is not practical to eliminate all X-sources due to timing constraints, area overhead, inefficiencies of the simulation engine used for fault-free simulation (e.g., 0-delay), and inaccuracies in modeling the behaviors of certain circuit blocks (e.g., memory blocks, custom logic blocks, analog circuit blocks). Some of these problems become visible late in the design, or after the ICs have already been manufactured, when it is very difficult or impossible to insert additional DFT structures.

Any response compaction technique must be able to detect a defective chip in the presence of the residual X's that cannot be eliminated by DFT or by accurate modeling. There are three ways of addressing this problem: (1) fixing the X's to 0s and 1s before they enter the compaction hardware [Barnhart 01, Rajski 02, Naruse 03, Wohl 03a, Volkerink 03]; (2) post-processing the response data to determine whether the tested IC is defective [Patel 03]; and, (3) designing compaction hardware that can tolerate X's and identify defective parts without requiring any fixing of X's or any post-processing [Mitra 02, Mitra 04]. The first two approaches require significant tester support and knowledge of the exact positions of X's in test responses. The third approach imposes no such requirements, and is ideal for BIST and test compression purposes. Hence, it serves as the fundamental principle behind the signature analysis techniques described in this paper. Of course, other techniques and tester features can complement the presented techniques.

The X-Compact technique [Mitra 02, Mitra 04] was the first published work on designing X-tolerant response compactors. X-Compactors are combinational circuits that are used to compact the scan chain outputs at every scan cycle. An application of the scan clock to perform a shift operation is referred to as a *scan cycle*. Hence, the test equipment is required to collect the X-Compactor outputs at every scan cycle and compare with the expected response. The X-tolerant signature analyzers described in this paper are sequential circuits that can be used for compacting test responses over multiple scan cycles, similar to Single-Input-Signature-Registers (SISRs) and Multiple-Input-Signature-Registers (MISRs). However, unlike SISRs and MISRs, these signature analyzers provide a high degree of tolerance to X's. After the test responses from multiple scan cycles are compacted into a signature, the tester compares the actual signature with the expected signature. During comparison, the bit positions that are X's in the fault-free signature are ignored. This comparison can be performed on the tester or on-chip. Hence, these signature analyzers are useful for BIST applications with minimal tester support.

Depending on the number of X's in a design, test response data volume can be reduced by up to three orders of magnitude. No assumptions about defect behaviors are necessary (e.g., "all defects behave as single stuck-at faults"). Hence, the design techniques are independent of any fault models, and the compaction hardware is unaffected by any engineering change orders or changes in test patterns after tape-out.

The major contributions of this paper are: (1) a probabilistic analysis of the relationships between the amount of X-tolerance and the amount of compaction that can be achieved using stochastic coding, which forms the basis for X-tolerant signature analyzer designs; (2) simple yet effective X-tolerant signature analyzer designs using stochastic coding; (3) an analysis of the effectiveness of these techniques for profiles of X's obtained from actual industrial designs; and, (4) test compression and BIST architectures using X-tolerant signature analyzers.

As will be clear to the reader, the benefits of stochastic coding for X-tolerant signature analyzer designs include: (1) simple hardware; (2) practically no impact on test quality; (3) no complex algorithms for construction of deterministic codes is required; and (4) any mistakes in the compaction hardware design can be easily overcome.

Section 2 of this paper presents a theoretical framework that forms the basis for X-tolerant signature analyzer designs. Section 3 describes X-tolerant signature analyzer designs. Section 4 discusses design of scan and BIST architectures using X-tolerant signature analyzers. An overview of previously published test response compaction techniques together with comparisons with our techniques are presented in Sec. 5.

2. X-Tolerance and Stochastic Codes

Suppose that n bits of test response are to be compacted into m bits. This situation can be represented by a matrix with n rows and m columns, where each row represents a bit in the uncompacted test response and each column represents a bit in the compacted test response. The matrix entry corresponding to row i and column j is 1 if and only if bit j of the compacted test response depends on bit i of the uncompacted test response. Otherwise, the entry is 0. For consistency with earlier publications, we call this matrix an *X-Compact Matrix* [Mitra 02, Mitra 04]. A bit in the compacted response is obtained by calculating the XOR of those bits in the uncompacted test response that have 1s in the compacted response bit's column.

The fundamental problem in designing response compactors is to determine how to fill the X-Compact matrix entries with 1s and 0s. Here, we consider a *stochastic coding approach*, in which we fill the entries in the matrix with 1s and 0s at random, such that the probability of assigning 1 to an entry is p and the probability of assigning 0 to an entry is $1-p$. The expected number of 1s in a row, the *expected weight* of the row, is then $m \times p$.

A bit in the compacted response is a *non-X bit* if the logic value of that bit in the expected compacted response is not X. An error in the uncompacted test response is *masked* if none of the non-X bits in the compacted response are erroneous. Next, we ask the following question: what is the probability that an error in an

uncompacted test response bit gets masked when there are X's in k other uncompacted test response bits? This probability is given by the following expression:

$$[1 - p \times (1-p)^k]^m \qquad (1)$$

Expression (1) is derived from the fact that if an entry in the row corresponding to the erroneous bit is 0, the error does not affect the corresponding bit in the compacted response. If an entry in this row is 1, the error affects the corresponding compacted response bit (column) if and only if none of the k rows producing X's have 1s in that column – otherwise, the error is masked. An error must affect at least one of the non-X bits in the compacted response in order to detect a defective part.

As shown below, expression (1) reaches its minimum when the following relationship is satisfied: $p = \frac{1}{k+1}$.

$$f(p) = [1 - p \times (1-p)^k]^m$$

$$\frac{df(p)}{dp} = m[1 - p \times (1-p)^k]^{m-1}[p \times k \times (1-p)^{k-1} - (1-p)^k]$$

For minimizing the value of the function f, $\frac{df(p)}{dp}$ must be 0. The first two factors in the product cannot be 0 for m, $k > 0$. Hence, the third factor must be 0. This equality implies $p = \frac{1}{k+1}$, since $p = 1$ is not a desired solution. For this value of p, $\frac{d^2 f(p)}{dp^2} > 0$.

At this optimum probability, the expected weight of each row is $\frac{m}{k+1} = \frac{m}{n \times \left(\frac{k}{n} + \frac{1}{n}\right)} \approx \frac{1}{\frac{n}{m} \times \frac{k}{n}} = \frac{1}{r \times d}$, where $r = \frac{n}{m}$ is the compaction ratio, and $d = \frac{k}{n}$ is the proportion of X's, which is often referred to as the *X-density*.

If there are t error bits and k X's in the uncompacted test response, the probability that no non-X bit of the compacted response is erroneous is:

$$[1 - \sum_{1 \le i \le t,\, i \text{ is odd}} \binom{t}{i} p^i (1-p)^{t-i+k}]^m \qquad (2)$$

The derivation of expression (2) is similar to that of expression (1). The only difference is that, a bit in the compacted response is erroneous if and only if it is derived from an odd number of error bits in the uncompacted response, and it doesn't depend on any bit producing X in the uncompacted response. An even number of error bits affecting a given output is not detectable, a phenomenon called *error cancellation.* From practical experience, a pessimistic value of t is between 3 and 5.

Let us look at these parameters for some industrial ASIC designs, as shown in Table 2.1. The X-density in the test response is calculated from fault-free simulation of single stuck-at test patterns. Tables 2.2a, 2.2b, 2.2c and 2.2d show the corresponding probability that errors in the test response get masked due to the presence of X's, using expressions (1) and (2) with the value of p for which expression (1) results in the minimum error masking probability.

Using our approach, even the design with ill-managed X's (Design 4) achieves more than 50x reduction in test response data volume. For designs with well-managed X's, reduction by three orders of magnitude is possible. Section 3 presents techniques for designing X-tolerant signature analyzers using this principle. Although not discussed in this paper, the stochastic coding approach can also be used for designing X-tolerant combinational compactors for which deterministic codes are either unknown or are too expensive to construct systematically (e.g., solutions to NP-complete problems are required).

Table 2.1. X-densities in industrial ASIC designs.

Name	X-management	X-density (% test response bits with X's)	Average number of test response bits per X (= 1/X-density)
Design 1	Well-managed	0.004%	25,000
Design 2	Moderately-managed	0.02%	5,000
Design 3	Ill-managed	0.15%	666
Design 4	Ill-managed	0.3%	333

Table 2.2a. Response compaction for Design 1 (well managed X's).

Number of response bits	Number of X's	Number of compacted bits	Compaction ratio	Average compacted bits per X	Probability of errors masked by X's		
					$t = 1$	$t = 3$	$t = 5$
100,000	4	100	1,000	25	2×10^{-4}	2.5×10^{-8}	8×10^{-10}
		50	2,000	12.5	1.3×10^{-2}	1.5×10^{-4}	3×10^{-5}
1,000,000	40	2,000	500	50	1.2×10^{-8}	$< 10^{-22}$	$< 10^{-36}$
		1,000	1,000	25	10^{-4}	$< 10^{-11}$	$< 10^{-18}$
		500	2,000	12.5	10^{-2}	1.9×10^{-6}	$< 10^{-9}$

Table 2.2b. Response compaction for Design 2 (moderately managed X's).

Number of response bits	Number of X's	Number of compacted bits	Compaction ratio	Average compacted bits per X	Probability of errors masked by X's		
					$t = 1$	$t = 3$	$t = 5$
100,000	20	600	170	30	2×10^{-5}	$< 10^{-13}$	$< 10^{-20}$
		250	400	12.5	10^{-2}	3.6×10^{-6}	$< 10^{-8}$
		125	800	6.25	10^{-1}	1.9×10^{-3}	6×10^{-5}
1,000,000	200	6,500	153	32.5	6.5×10^{-6}	$< 10^{-15}$	$< 10^{-25}$
		2,400	420	12	10^{-2}	2×10^{-6}	$< 10^{-9}$
		1,250	800	6.25	10^{-1}	10^{-3}	1.2×10^{-5}

Table 2.2c. Response compaction for Design 3 (ill-managed X's).

Number of response bits	Number of X's	Number of compacted bits	Compaction ratio	Average compacted bits per X	Probability of errors masked by X's		
					$t = 1$	$t = 3$	$t = 5$
100,000	150	4,000	25	26.7	5×10^{-5}	$< 10^{-12}$	$< 10^{-20}$
		1,700	60	11.3	10^{-2}	4×10^{-6}	$< 10^{-8}$
		900	110	6	1.1×10^{-1}	1.4×10^{-3}	2×10^{-5}
1,000,000	1,500	38,000	26	25.3	8×10^{-5}	$< 10^{-12}$	$< 10^{-20}$
		16,000	63	10.6	2×10^{-2}	$< 10^{-5}$	$< 10^{-8}$
		9,000	110	6	1.1×10^{-1}	1.3×10^{-3}	1.6×10^{-5}

Table 2.2d. Response compaction for Design 4 (ill-managed X's).

Number of response bits	Number of X's	Number of compacted bits	Compaction ratio	Average compacted bits per X	Probability of errors masked by X's		
					$t = 1$	$t = 3$	$t = 5$
100,000	300	8,000	13	26.6	5×10^{-5}	$< 10^{-12}$	$< 10^{-21}$
		3,000	34	10	2.5×10^{-2}	1.7×10^{-5}	1.2×10^{-8}
		1,800	56	6	1.1×10^{-1}	1.4×10^{-3}	1.8×10^{-5}
1,000,000	3,000	76,000	13	25.3	8×10^{-5}	$< 10^{-12}$	$< 10^{-20}$
		30,000	34	10	2.5×10^{-2}	1.6×10^{-5}	$< 10^{-8}$
		18,000	56	6	1.1×10^{-1}	1.3×10^{-3}	1.6×10^{-5}

3. X-Tolerant Signature Analyzer Design using Stochastic Codes

For simplicity, we first consider the case of serial signature analysis in which case test response bits from a single source (e.g., a single scan chain) are compacted into a signature. We call this structure an X-SISR, which stands for X-tolerant Single Input Signature Register. Next, we describe the design of an X-MISR – X-tolerant Multiple Input Signature Register.

3.1. X-SISR Design

Suppose that there is a source producing *n* bits of test response data serially and that the signature consists of *m* bits. Here *n* and *m* are design parameters. As explained in Sec. 2, the stochastic coding approach fills the X-Compact matrix (with *n* rows and *m* columns) with 1s and 0s – the probability of assigning 1 is *p*, a parameter based on the X-density of the design.

The next question is: How can such a coding scheme be implemented in hardware? Given the parameter *p*, a simple weighted-random pattern generator can be used to generate 1s with probability *p*. Design of logic structures to generate weighted random patterns has been extensively covered in testing literature [Eichelberger 91] and is not repeated here. The X-SISR design is shown in Fig. 3.1.

At each clock cycle, a test response bit is AND-ed with the weighted random pattern generator outputs. Depending on where the 1s appeared, the corresponding signature bits will depend on this particular test response bit – the XOR gate and feedback loop achieves that. The reader can prove the equivalence between this structure and the matrix structure.

The area overhead of the X-SISR structure is analyzed next. Each signature bit has a flip-flop, one two-input XOR gate, and one two-input AND gate associated with it. This results in *m* flip-flops, *m* two-input XOR gates and *m* two-input AND gates. In addition, there is area overhead due to the LFSR, the phase shifter and the weight logic. For the designs analyzed in Sec. 2, the minimum weight used is of the order of $\frac{1}{3000}$. This implies that approximately 12 random bits (with probability of 1 = 0.5) will be AND-ed generate a single bit of the given weight for those designs. Hence, the size of LFSR is required to be greater than roughly $\log_2(12 \times n)$, where *n* is the number of uncompacted test response bits. Hence, the maximum length of an LFSR that is required for the previously analyzed designs is roughly 24 bits for 1 million uncompacted test response bits. The area overhead of the weight logic will be of the order of 12 two-input gates for each signature bit. This adds extra area of roughly 12*m* two-input gates for all outputs. There is a fanout of *m* from the serial test data input.

For designs using LFSRs and/or phase shifters for data compression or test pattern generation for Built-In-Self-Test, the same LFSRs and phase-shifters can be reused for X-SISR designs.

The unique features of the X-SISR design are: (1) no complicated controller is required to implement the signature analysis – no complex seed computation is required and the LFSR can start from any non-zero state; (2) properties of LFSR are utilized to achieve the effect of random filling of the X-Compact matrix; (3) unlike traditional weighted-random testing, a single weight is enough; however, multiple weights can always be built in to create programmability; (4) depending on the design, thousands of X's are tolerated with high probability.

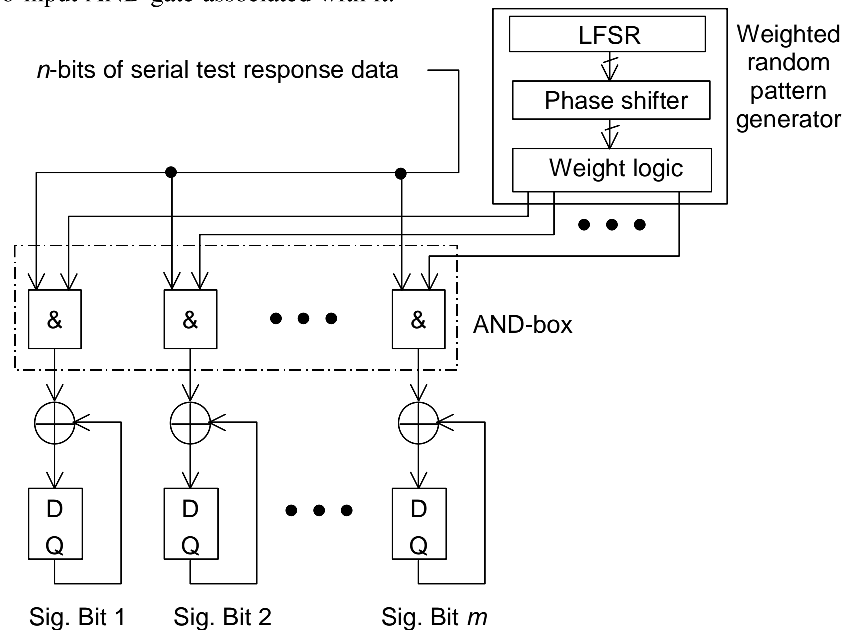


Figure 3.1. X-SISR design.

Depending on the value m, the fanout in Fig. 3.1 may be a cause of concern. For example, suppose that the value of m is 20. Implementation of the design of Fig. 3.1 means that we need a fanout of 20. Generally scan chain outputs are compacted and scan shifting is performed at slow speed. Hence, this may not be problem because signal paths to the response compaction circuitry aren't timing critical. However, simple pipelining minimizes the fanout problem. Figure 3.2 shows such an example where four flip-flops are added and the output of each of these flip-flops fans out to a group of five signature flip-flops. This organization reduces the fanout problem. The fanout associated with the weighted random generators can be reduced by using multiple weighted random pattern generator circuits.

3.2. X-MISR Design

The X-SISR design of Sec. 3.1 is converted into an X-MISR design in the following way. Suppose that there are s parallel test response inputs (which can be scan chains). For each parallel test response input, there are m AND gates as shown in Fig. 3.1 – this structure is referred to as the *AND-box*. However, unlike Fig. 3.1, for each flip-flop corresponding to each signature bit, there is an (s+1)-input XOR gate instead of a 2-input XOR gate. The overall design is shown in Fig. 3.3. Relative to the X-SISR design, the X-MISR requires s 2-input XOR gates for each signature bit, and m 2-input AND gates for each AND box.

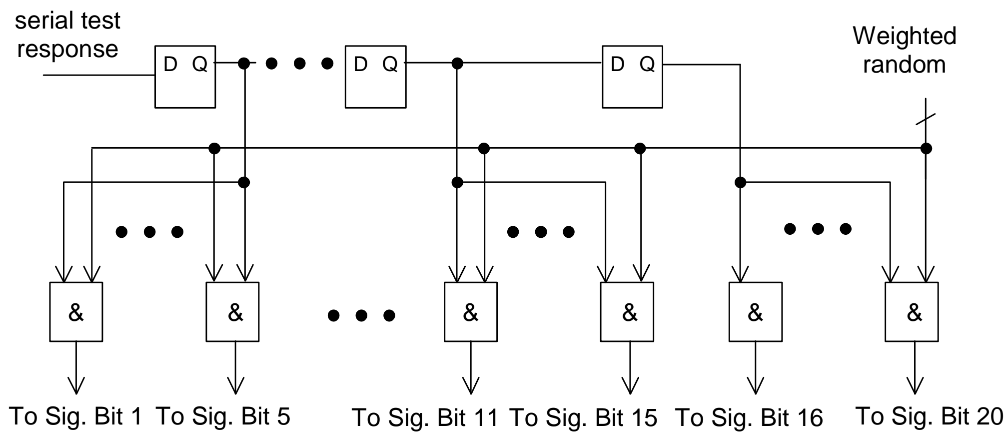


Figure 3.2. Fanout control in X-SISR design.

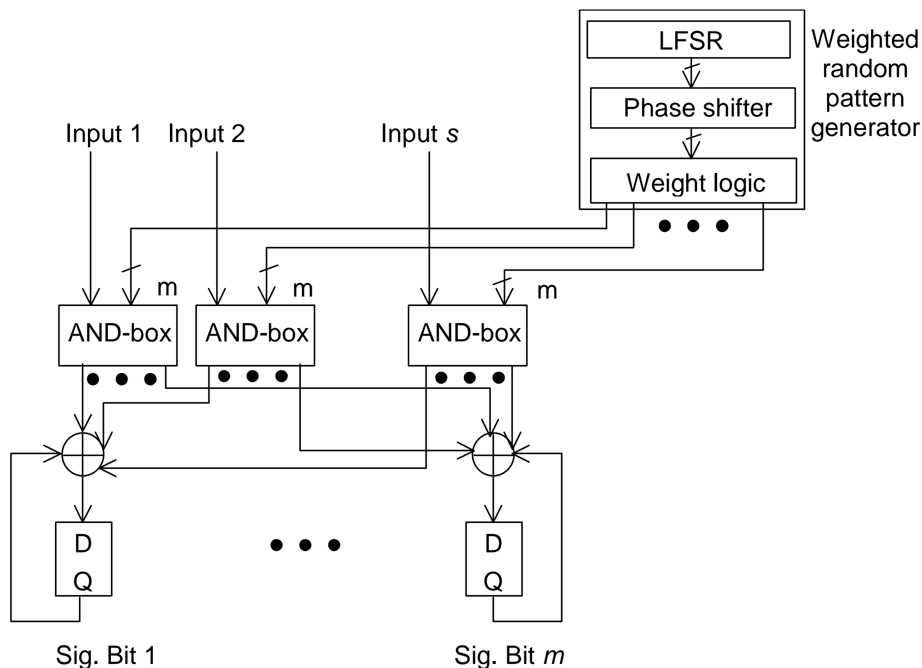


Figure 3.3. X-MISR design.

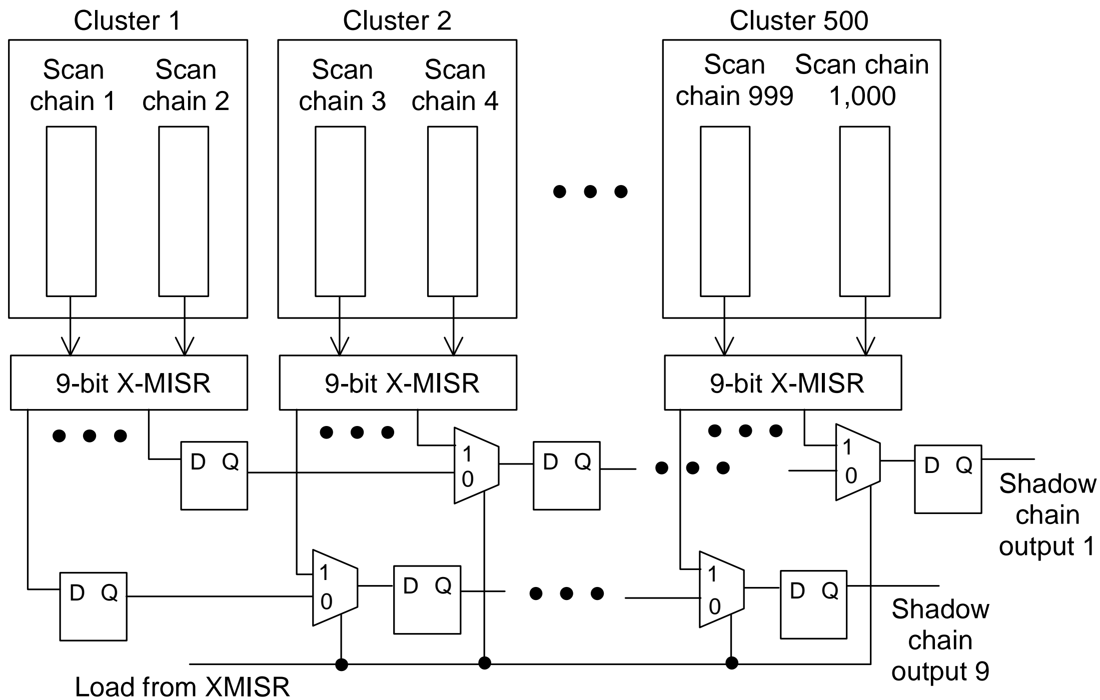


Figure 3.4. Scan architecture with local X-MISRs and intermediate signatures.

Depending on where the signature analyzer is implemented (e.g., on the tester to reduce test response data volume or on-chip to reduce test data volume, test time, and test pins), the routing overhead must be suitably managed. One way of accomplishing this is to use local X-MISRs and intermediate signatures. We illustrate this point using an actual design example – the Design 3 of Sec. 2. Suppose that Design 3 has 1,000 scan chains and each chain is 1,000 bits long. We break the design into 500 clusters, for example, such that each cluster contains two scan chains. For each cluster, we build an X-MISR of nine bits. Next, we run the X-MISR for 500 cycles – this corresponds to 1,000 response bits per cluster, which can be compacted into nine bits of signature with a high error detection probability (Table 2.2c). Every 500 cycles, these nine-bit intermediate signatures from each cluster are transferred to a shadow register. The shadow registers of all clusters are configured into nine scan chains. It takes 500 cycles to scan these intermediate signature bits from all clusters out of the nine scan chains before another intermediate signature from the clusters is loaded into the shadow register. The overall scan architecture is shown in Fig. 3.4. This approach leads to an X-tolerant signature analysis technique that tolerates thousands of X's but outputs only nine bits of response for observation on the tester in every scan cycle, although there are 1,000 scan chains. Thus, the test data volume and test time reduction obtained is more than two orders of magnitude even for a design with ill-managed X's. The shadow scan chains of Fig. 3.4 do not necessarily need to be connected in the way shown in the figure. For example, all bits of the first X-MISR can be connected in a single scan chain, followed by the bits in the second X-MISR, and so on.

4. Scan and BIST Architecture Design with X-Tolerant Signature Analyzers

In this section, we show how the probabilistic analysis of Sec. 2 can be used to make decisions about designing test architectures using X-tolerant signature analyzers. Suppose that we decide to use a certain signature register design, which is characterized by the number of signature bits (m) and the weight logic. For a specified maximum probability of error masking, the number of X's (k) that can be tolerated can be derived directly from this signature analyzer design using expression (1) of Sec. 2. The X-density and the distribution of X's in the design then decide the number of uncompressed response bits (n) that contain k X's. This result implies that the signature register contents must be scanned out for every n uncompressed signature bits. Hence, the compaction ratio is n/m.

The graph in Fig. 4.1 illustrates the associated trade-offs. For X-tolerant signature analyzers with between 50 and 500 bits and several weights, the graph presents the compacted response volume per X in the uncompacted response as a function of the probability of not detecting an error. This makes the graph independent of X-density.

Each line in the graph corresponds to designs with a specific signature length, but combines several weight values. The weighting, which decides the average number of 1s in every row of the X-Compact matrix, is restricted to inverse powers of two (1/2, 1/4, etc.), as they are readily generated from a pseudo-random source using only AND gates. The curves in the graph are then obtained by selecting the optimal weight for each error masking probability.

The analysis used to generate the graph of Fig. 4.1 treats uncompressed response bits as independent random variables for the purposes of calculating the exact number of X's. In particular, given an expected number of X's, λ, we use a Poisson distribution to calculate the probability of seeing each possible value of k. The probability of having k X's is thus given by $\frac{e^{-\lambda}\lambda^k}{k!}$. For signature register length (m) and weighting p, expression (2) of Sec. 2 gives the error masking probability due to X's. The value of t, the number of error bits, is assumed to be 3. The *total expected error masking probability* for a given expected number of X's (λ) is:

$$\sum_{k=0}^{\infty} \frac{e^{-\lambda}\lambda^k}{k!} \times \text{Prob. of error masking with } k \text{ X's}.$$

The log-scale horizontal axis of Fig. 4.1 represents the total expected error masking probability.

For a given signature analyzer design, each point corresponds to a certain value of λ, ranging from 0 to about 73 (the rightmost point on the 500-bit signature).

Since the X-SISR and X-MISR structures allow dynamic generation of random X-Compact matrices, test responses with too many X's can easily be repeated (by applying the same test set or vector multiple times) to allow targeted reduction of failure probabilities. Each such run creates independent X-Compact matrices and, hence, the probability of error masking due to X's is given by the product of error masking probabilities from individual runs. Any dependence of the weighted-random generator on initial seeds can be eliminated by loading multiple seeds into the generator during these runs. Depending on the application (e.g., tester supported or BIST in the field), the number of cycles of compaction between intermediate signatures may or may not be fixed in advance. Controlling this timing more carefully based on the X's in the uncompacted response reduces the error masking probability to those given by expressions (1) and (2) by eliminating the need for a Poisson distribution.

The curves in Fig. 4.1 can be used to select an appropriate signature size and to guide the application of DFT techniques to reduce X-density to an acceptable level. For example, consider the use of a 100-flip-flop design with an assumption of three errors when a defect is detected (t=3). Suppose that the target probability of not detecting the errors is set to one in ten million. In Fig. 4.1, this corresponds to the 10^{-7} point on the horizontal axis. The best weight in this case is 1/8, and the design requires a compacted response volume of 50 bits per X in the uncompacted test response. If the compacted response volume must be limited to 10,000 bits, X-density must be reduced to the point that the input contains no more than 200 X's. If the goal is instead a compaction ratio of 100, X-density must be reduced to about 0.02%, because 1 X per 50 bits of compacted response translates to 1 X every $50 \times 100 = 5{,}000$ bits of uncompacted test response – this value corresponds to the X-density of Design 2 discussed earlier.

It is important to note that selecting such a design does not fix the operating point, which can be scaled as necessary (after the chip is manufactured) by simply modifying the scanout interval between intermediate signatures. Using the same X-tolerant design just discussed, for example, the probability can be further reduced to less than two in a billion by doubling the number of intermediate signatures loaded into the shadow register. Such retroactive changes are difficult with the deterministic X-Compact matrices, forcing designers to estimate X-density much more carefully in advance.

The overall advantages of stochastic coding for designing X-tolerant signature analyzers are: (1) the compaction hardware consisting of weighted random generators is simple; (2) no complicated sequential controllers need to be designed, unlike deterministic approaches; (3) a huge number of independent X-Compact matrices can be generated out of a single compaction hardware; repeating the same stimulus for response compaction with multiple independent X-Compact matrices can efficiently reduce the error masking probability; (4) no complicated and time-consuming deterministic algorithms need to be executed to construct X-Compact matrices; and (5) any mistakes in the design of the compaction hardware can be easily overcome.

Application of X-MISR for built-in-self-test in the field might involve the use of an on-chip memory (e.g., cache memory or other storage areas available for BIST use) that can be preloaded with test response data from an off-chip store. This test response data would both indicate which bits should be masked as well as the expected responses for the remaining bits, allowing the chip itself to compute a pass/fail result with no need to output any test data whatsoever. As with other types of test data, the pre-loaded data could also be compressed for storage in a small ROM or FLASH memory.

5. Related Work

Previous work on response compaction can be classified into two broad categories: response compaction in the absence of any X's, and response compaction in the presence of X's. Several publications belonging to the first category that discuss signature analysis and combinational compaction include [Bardell 87, Chakrabarty 95, Chakrabarty 98, McCluskey 86, Pradhan 91, Saluja 83,

Saxena 92, Saxena 97]. These techniques do not address X's. Some of them assume that actual defects behave as faults from a chosen fault model that is not supported by experimental data [McCluskey 04].

Recent publications discuss response compaction in the presence of X's. The X-Compact technique described in [Mitra 02, Mitra 04] uses combinational circuits for X-tolerant response compaction. The techniques presented in [Rajski 03 Wang 03, Wohl 03b] build on the X-Compact concept. These techniques guarantee tolerance of a single X. The techniques in [Rajski 03, Wang 03] use the X-Compactor circuitry as the basic combinational logic block but reduce the number of bits to be observed at the expense of the number of X's tolerated over multiple scan cycles. The technique in [Wohl 03b] introduces a graph-theoretic formulation, in contrast with the matrix-theoretic formulation used in [Mitra 02, Mitra 04], to generate X-Compact matrices such that each row of the X-Compact matrix contains two 1s. The response compaction technique in [Sinanoglu 03] uses a parity bit for every scan chain in addition to XOR-trees – however, a single X in a scan chain will corrupt the parity bit.

The technique presented in this paper is unique for the following reasons: (1) It is capable of tolerating even thousands of X's; (2) uses signature analysis to tolerate X's; (3) presents a novel architecture that can dynamically generate response compaction circuits without requiring circuit implementations with high controller overhead in signature analysis applications; and, (4) extends the concept of weight distribution in coding theory and aliasing analysis to include X's.

Techniques such as those presented in [Barnhart 01, Naruse 03, Rajski 02, Wohl 03a, Volkerink 03] use DFT and/or active masking from the tester to reduce the number of X's. Of course, these techniques can be used in conjunction with the X-tolerant signature analysis technique presented in this paper.

The I-Compact technique in [Patel 03] treats X's as erasures and performs post-processing of the compacted responses on the tester to identify defective parts. The architecture described in this paper is entirely compatible with the I-Compact approach. Combinations of the two techniques can also be used, even within a single test vector, by treating some X's as erasures and others as unknowns that corrupt certain outputs. This approach can further reduce the probability that the errors are masked by unknowns, compared to the X-tolerant signature analysis approach alone, at the expense of extra processing on the tester.

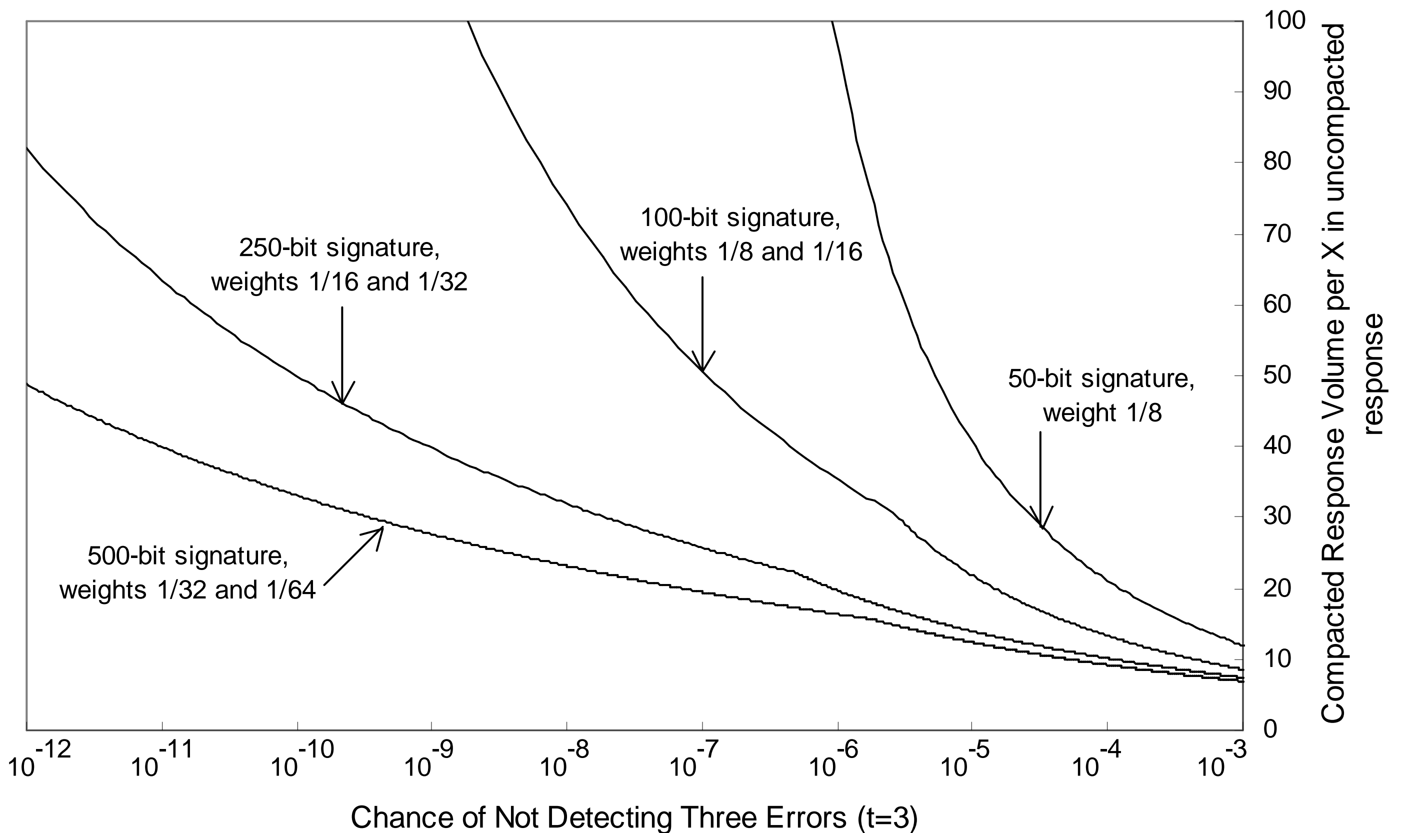


Figure 4.1. Pareto curve for X-tolerant signature analyzer designs.

6. Conclusions

X-tolerant signature analyzer designs using stochastic coding approaches are capable of massive reduction in test response data volume with negligible impact on test quality. For the industrial designs analyzed in this paper, these signature analyzers have capabilities of tolerating thousands of X's while reducing test response data volume by 50 to 2,000 times. These signature analyzer designs are independent of test sets, fault models, engineering change orders and do not impose any restrictions on the accuracy of fault models. Several novel test compression and BIST architectures can be designed using such signature analyzers. In conclusion, X-tolerant signature analyzers provide a novel solution for test compression and Built-In-Self-Test.

7. Acknowledgments

We acknowledge Dr. Kee Sup Kim of Intel Corporation for his support. We would also like to thank Mr. Shyam Kallepalli of Intel Corporation for providing us with data on the characteristics of industrial designs used in this paper. Steve Lumetta was supported in part by National Science Foundation grant ACI-99-84492-CAREER. The content of the information does not necessarily reflect the position or the policy of that organization.

8. References

[Bardell 87] Bardell, P.H., W.H. McAnney, and J. Savir, *Built-In Test for VLSI: Pseudo-Random Techniques*, Wiley-Interscience, 1987.

[Barnhart 01] Barnhart, C., V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, "OPMISR: The Foundation for Compressed ATPG Vectors," *Proc. Intl. Test Conf.*, pp. 748-757, 2001.

[Chakrabarty 95] Chakrabarty, K., B.T. Murray, and J.P. Hayes, "Optimal Space Compaction of Test Responses," *Proc. Intl. Test Conf.*, pp. 834-843, 1995.

[Chakrabarty 98] Chakrabarty, K., "Zero-aliasing Space Compaction using Linear Compactors with Bounded Overhead," *IEEE Trans. CAD*, Vol. 17, No. 5, pp. 452-457, May 1998.

[Eichelberger 91] Eichelberger, E.B., E. Lindbloom, J. Waicukauski, and T.W. Williams, *Structured Logic Testing*, Prentice Hall, 1991.

[McCluskey 86] McCluskey, E.J., *Logic Design Principles with Emphasis on Testable Semicustom Circuits*, Prentice Hall, 1986.

[McCluskey 03] McCluskey, E.J., *et al.*, "Test Compression Roundtable," *IEEE Design and Test of Computers*, Vol. 20, Issue 2, pp. 76-87, March-April, 2003.

[McCluskey 04] McCluskey, E.J., *et al.*, "ELF-Murphy Data on Defects and Test Sets," *Proc. IEEE VLSI Test Symp.*, pp. 16-22, 2004.

[Mitra 02] Mitra, S., and K.S. Kim, "X-Compact: An Efficient Response Compaction Technique for Test Cost Reduction," *Proc. Intl. Test Conf.*, pp. 311-320, 2002.

[Mitra 04] Mitra, S., and K.S. Kim, "X-Compact: An Efficient Response Compaction Technique," *IEEE Trans. CAD*, Vol. 23, Issue 3, pp. 421-432, March 2004.

[Naruse 03] Naruse, M., I. Pomeranz, S.M. Reddy and S. Kundu, "On-chip Compression of Output Responses with Unknown Values using LFSR Reseeding," *Proc. Intl. Test Conf.*, pp. 1060-1068, 2003.

[Patel 03] Patel, J.H., S.S. Lumetta and S.M. Reddy, "Application of Saluja-Karpovsky Compactors to Test Responses with Many Unknowns," *Proc. IEEE VLSI Test Symp.*, pp. 107-112, 2003.

[Pradhan 91] Pradhan, D.K., and S.K. Gupta, "A New Framework for Designing and Analyzing BIST Techniques and Zero Aliasing Compression," *IEEE Trans. Computers*, Vol. 40, No. 6, pp. 743-763, June 1991.

[Rajski 02] Rajski, J., *et al.*, "Embedded Deterministic Test for Low Cost Manufacturing Test," *Proc. IEEE Intl. Test Conf.*, pp. 301-310, 2002.

[Rajski 03] Rajski, J., J. Tyszer, C. Wang and S.M. Reddy, "Convolutional Compaction of Test Responses," *Proc. Intl. Test Conf.*, pp. 745-754, 2003.

[Saluja 83] Saluja, K.K., and M. Karpovsky, "Testing Computer Hardware through Data Compression in Space and Time," *Proc. Intl. Test Conf.*, pp. 83-89, 1983.

[Saxena 92] Saxena, N.R., P. Franco, and E.J. McCluskey, "Simple Bounds on Signature Analysis Aliasing for Random Testing," *IEEE Trans. Computers,* pp. 638-645, May 1992.

[Saxena 97] Saxena, N.R., and E.J. McCluskey, "Parallel Signature Analysis Design with Bounds on Aliasing," *IEEE Trans. Computers*, Vol. 46, No. 4, pp. 425-438, April 1997.

[Sinanoglu 03] Sinanoglu, O., and A. Orailoglu, "Compacting Test Responses for Deeply Embedded SOC Cores," *IEEE Design & Test of Computers*, Vol. 20, Issue 4, pp. 22-30, July-Aug., 2003.

[Volkerink 03] Volkerink, E., and S. Mitra, "Test Response Compaction for Any Number of Unknowns," *IEEE Intl. Workshop Infrastructure IP*, 2003.

[Wang 03] Wang, C., S.M. Reddy, I. Pomeranz, J. Rajski and J. Tyszer, "On Compacting Test Response Data Containing Unknown Values," *Proc. Intl. Conf. Computer-Aided Design*, pp. 855-862, 2003.

[Wohl 03a] Wohl, P., J.A. Waicukauski, S. Patel and M.B. Amin, "Efficient Compression and Application of Deterministic Patterns in a Logic BIST Architecture," *Proc. Design Automation Conf.*, pp. 566-569, 2003.

[Wohl 03b] Wohl, P., and L. Huisman, "Analysis and Design of Optimal Combinational Compactors," *Proc. IEEE VLSI Test Symp.*, pp. 101-106, 2003.