

X-Canceling MISR – An X-Tolerant Methodology for Compacting Output Responses with Unknowns Using a MISR

Nur A. Touba

Computer Engineering Research Center
Department of Electrical and Computer Engineering
University of Texas, Austin, TX 78712
touba@ece.utexas.edu

Abstract

A new X-tolerant multiple-input signature register (MISR) compaction methodology is proposed which can compact output streams containing unknown (X) values. Unlike conventional X-masking approaches, it does not require any masking logic at the input of the MISR. Instead it uses symbolic simulation to express each bit of the MISR signature as a linear equation in terms of the X's. Linearly dependent combinations of the signature bits are identified with Gaussian elimination and XORed together using a programmable XOR to cancel out all X values thereby yielding deterministic values that are invariant of what the final values of the X's end up being during the test. These X-canceled values can be compacted in a separate MISR to generate a final X-free signature. Each intermediate signature for an m-bit MISR can tolerate k X's present anywhere in the output stream with error detection capability equivalent to using an m-k bit MISR with no unknowns. The tester storage requirement is a small constant times the total number of unknowns in the test set and thus does not depend on the scan architecture, the number of test vectors, or the distribution of X's which is a key advantage compared with other X-tolerant compaction schemes.

1. Introduction

Compacting output streams that have unknown 'X' values is a major issue for test compression and BIST. There are many sources of unknown values that commonly arise during simulation, for example uninitialized memory elements, bus contention, floating tri-states, etc. X's corrupt the final signature making it unknown. A number of schemes have been developed to deal with the problem of X's in the output response.

One approach is to modify the circuit-under-test (CUT) to eliminate the sources of X values. This is often called *X-bounding* or *X-blocking*. It involves inserting design-for-testability (DFT) hardware into the CUT to prevent X's from propagating to scan cells [Wang 06].

Another approach, which does not require modifying the CUT, is to use *X-masking*. This involves masking out

the X's at the input to the compactor. Mask data is required to specify which scan chain outputs should be masked in each clock cycle. A number of techniques have been developed for designing the masking hardware and compressing the amount of mask data that is required [Barnhart 01], [Wohl 01, 03, 04], [Pomeranz 02], [Chickermane 04], [Vokerink 05], [Chao 05], [Tang 06], [Rajski 06a]. In many cases, the resolution of the masking is reduced in order to keep the amount of mask data at reasonable levels (e.g., an entire scan chain may be masked or an entire scan slice may be masked). This results in some non-X values also getting masked which reduces observability and may impact the coverage, particularly for unmodeled faults.

A third approach is to use an *X-tolerant* compactor that can compact an output stream that contains X's. This is an attractive approach as it eliminates the need for X-masking. Mitra and Kim [Mitra 04a] published the first X-tolerant compactor called X-Compact which consists of a combinational circuit that compacts n scan chains outputs in each clock cycle down to m bits which are compared with the expected response on the tester. When X's arrive at the input of the compactor, they propagate to some of the m output bits thereby corrupting them. The corrupted output bits are masked (i.e., ignored) by the tester. The key idea in X-Compact is to design the combinational circuit so that in the presence of a limited number of X inputs, enough of the compacted outputs are left uncorrupted such that errors on the non-X inputs can still be detected. The compaction ratio of n to m depends on the number of X's that are guaranteed to be tolerated. To maximize the compaction ratio, typically only one X would be guaranteed to be tolerated.

In [Patel 03], the results from [Saluja 83] were applied for compacting output streams with unknowns by treating them as erasures. It was shown that any error correcting code (ECC) with distance d can be used to construct an X-tolerant combinational compactor that can detect e errors in the presence of k X's provided $e+k < d$. However, the number of fault-free signatures would be equal to 2^k and thus would require a one-to-many comparison which current testers do not support. However, this problem can be overcome by using the

“X-filter” approach proposed by Sharma and Cheng in [Sharma 05] which uses on-chip circuitry to cancel out X 's in the compacted response. The X -filter circuitry requires inputs every clock cycle which must be supplied by the tester (similar to X -masking approaches). However, it allows a combinational compactor based on any linear ECC to be used with conventional testers.

In [Rajski 05], a finite memory X -tolerant compactor, called a convolutional compactor, was proposed. It uses a combinational compactor whose outputs are XORed into different stages of multiple shift registers. Each compacted output bit is thus a function of scan chain outputs across a window of consecutive clock cycles. A convolutional compactor allows tradeoffs in terms of the compaction ratio versus the number of X 's that can be tolerated across a window of clock cycles.

In [Mitra 04b], an X -tolerant multiple-input signature register (MISR) approach was proposed based on stochastic coding. Based on the probability of X 's in the output stream, a weighted linear combination of scan chain outputs is fed to each MISR input. The weight logic is designed so as to minimize the expected number of bits in the MISR that get corrupted by X 's. The corrupted bits in the MISR are masked on the tester while the non-corrupted bits are compared with the fault-free signature. The error coverage is probabilistic and will vary depending on the distribution of X 's in the output stream. Error coverage can be improved by using more intermediate signatures or running the test multiple times with different linear combinations. The overhead scales with the product of the MISR size and number of scan chains and can be reduced by using multiple local MISRs.

In [Rajski 06b], an X -tolerant compactor based on using multiple circular registers of relatively prime length was proposed. It generates a signature for each test vector. If the number of scan cells is less than or equal to product of all the circular register lengths, then a single X cannot block observation of any scan cell. For multiple X 's, the probability of error masking increases with the number of X 's. Results in [Rajski 06b] for a particular 101-bit compactor indicated greater than 99% error coverage for up to 10 X 's could be obtained.

This paper presents a new X -tolerant scheme for compacting output streams using a MISR. It allows any number of scan chain outputs to be compacted with a conventional MISR of any size. The overhead scales linearly with the size of the MISR. For an m -bit MISR, k X 's present anywhere in the output stream can be tolerated with error detection capability equivalent to using a $m-k$ bit MISR with no unknowns. For example, with a 256-bit MISR, 236 X 's could be tolerated with less than 1 in a million chance of losing coverage for any error.

The paper is organized as follows: Sec. 2 gives an overview of the main idea in the proposed scheme. Sec. 3 describes the symbolic simulation process and how to

identify X -canceled MISR bit combinations. In Sec. 4, the X -canceling MISR architecture is described. In Sec. 5, one scheme for transferring control data from the tester to the X -canceling MISR is presented. In Sec. 6, the proposed scheme is analyzed and compared with other approaches. Experimental results are shown in Sec. 7, and Sec. 8 is a conclusion.

2. Overview of Proposed Scheme

This section gives an overview of the main idea. Subsequent sections describe the proposed scheme in detail.

In the proposed scheme, each X in the output stream is represented by a unique symbol, and symbolic simulation is used to determine the final state of the MISR in terms of the symbols (this is described in detail in Sec. 3). Each bit of the MISR signature corresponds to a linear combination of the symbols. If there are more bits in the MISR than there are symbols, some combinations of the bits in the MISR signature are guaranteed to be linearly dependent in terms of the symbols corresponding to the X 's. Combinations of linearly dependent bits are identified using Gaussian elimination. If some combination of MISR bits is linearly dependent in terms of the symbols corresponding to the X 's, then when those MISR bits are XORed together, all X 's will cancel out thereby yielding a deterministic value that is invariant of what the final values of the X 's end up being during the test. If this XOR combination of MISR bits mismatches with the fault-free value, then an error is indicated. Errors in approximately half of the output response bits will propagate to each linear dependent combination of the MISR bits. So checking q such combinations will provide an error coverage of approximately $1-2^{-q}$. Thus checking 7 such combinations will give an error coverage of over 99%, and checking 16 such combinations would give an error coverage of 99.998% equivalent to using a 16 bit MISR on an output sequence with no unknowns. Moreover, if fault simulation is used to identify the outputs that faults propagate to, it is possible to guarantee 100% coverage for some fault model by selecting an appropriate set of MISR bit combinations to check.

The MISR bit combinations can be checked using a programmable XOR network which requires very little area overhead and whose design is independent of the CUT. The MISR bit combinations for which all X 's cancel out are pre-computed via symbolic simulation, and then during test application, appropriate control values are supplied to the programmable XOR network so that it XORs together MISR bit combinations for which all X values cancel out. The number of intermediate signatures that need to be checked depends on the density of X 's in the output stream. The X -canceled outputs coming from the programmable XOR can themselves be compacted in a separate MISR such that no tester channels need to be used to transfer data back to the tester during the test

session. This is especially attractive for multi-site testing where broadcasting channels from the tester to all devices-under-test (DUTs) is inexpensive, but bringing data from each individual DUT back to the tester is costly.

As mentioned earlier, the idea of canceling out X 's by XORing combinations of bits was first proposed in [Sharma 05] in the context of constructing an X -filter circuit at the output of a combinational compactor based on any linear ECC. The proposed scheme uses a similar concept, but with a number of important differences. The X -filter method in [Sharma 05] maps an m -bit combinational output with up to k X 's into an m -bit deterministic output each clock cycle using hardware that scales with the product mk and control data equal to the product mkc where c is the total number of scan clock cycles. The proposed scheme is applied in the sequential domain to extract a small number of X -canceled combinations from intermediate MISR signatures. Gaussian elimination is used to simplify the hardware so that only a simple programmable XOR gate is required, hence the hardware scales with only m and the control data is $O(\text{total number of } X\text{'s in output stream})$ as will be shown in Sec. 6 which is much lower than [Sharma 05].

The approach in [Mitra 04b] also uses a MISR, but in a fundamentally different way. It tries to limit the number of bits in the signature that get corrupted and then masks the corrupted signature bits on the tester. Because corrupted signature bits spread fairly quickly in a MISR, the number of intermediate signatures that need to be used in [Mitra 04b] will be much higher than the proposed method for an equivalent error coverage. The proposed method can tolerate a much larger number of X 's in each signature and then simply cancel them out with the programmable XOR network. The method in [Mitra 04b] requires some output response channels going back to the tester plus masking data stored on the tester. The proposed method does not need either of these, but does need control data from the tester that is on the order of $O(\text{total number of } X\text{'s in output stream})$. This control data will be less than the masking data for [Mitra 04b] and hence the test data bandwidth and tester storage will be less for the proposed method.

3. Symbolic Simulation

Consider the output response that has been captured in the scan chains after applying a test vector. Let the value in each scan cell be represented by a symbol (as illustrated in Fig. 1). Symbol simulation can be performed to obtain the final state of the MISR in terms of the symbols after the output response has been shifted in to the MISR. Each bit of the MISR will be equal to a linear combination of the scan cells. This is shown in Fig. 1 where, for example, the final value of the top bit of the MISR will be equal to $X_1 \oplus O_3 \oplus O_8 \oplus O_{13}$.

Without loss of generality, assume all the non- X values in the output response are 0 so that each MISR bit is now simply equal to the linear combination of the X values. In Fig. 1, assume each symbol X_i has an X value and each symbol O_i has a non- X value. The X dependence of the MISR bits in this case is as shown in Fig. 2. Each X can take on any value 0 or 1. Thus, if there are k X 's, then there are 2^k different combinations of values for the X 's. In the example in Fig. 2, there are 4 X 's in the output response (X_1 - X_4), and each could be either 0 or 1 in a fault-free circuit. Thus there are 16 possible signatures that could arise in the MISR for a fault-free circuit. Each possible combination of values for the X 's can be thought of as producing a valid fault-free signature in the MISR. So for an m -bit MISR that compactes k X 's, approximately 2^k signatures out of 2^m possible signatures correspond to valid signatures that could arise in a fault-free circuit (could be slightly less if some combinations map to the same signature). If an error occurs, it will change a fault-free signature into a new signature, and the probability that the new signature is one of the 2^k valid fault-free signatures is $2^k/2^m$. So the probability of aliasing is $2^k/2^m$ assuming all possible signatures are equally likely and all 2^k fault-free signatures are unique. If k is 20 less than m , then the probability of aliasing is 2^{-20} which is less than one in a million. What this illustrates is that an m -bit MISR can quite safely compact an output stream with up to $m-20$ X 's with negligible loss of error coverage.

The problem, however, is how to check whether the MISR is in one of the valid 2^k fault-free signatures. Rather than doing this directly, a different approach can be taken. The linear equations for each MISR bit can be represented as a matrix where each row corresponds to a MISR bit and each column corresponds to an X . Each entry in the matrix is a 1 if the MISR bit corresponding to the row depends on the X corresponding to the column. This is illustrated in Fig. 2. For example, in Fig. 2, the second row of the matrix corresponds to M_2 , and the 1's in the first three columns indicate dependence on X_1 , X_2 , and X_3 , respectively. If the number of X 's is less than the size of the MISR, then there are more rows than columns and hence some combinations of rows are guaranteed to be linearly dependent. Gauss-Jordan elimination [Cullen 97] can be used to identify the linearly dependent row combinations. Gauss-Jordan elimination involves performing row operations that transform a set of columns into an identity matrix. Fig. 3 shows the matrix in Fig. 2 after Gauss-Jordan elimination has been performed. The all-0 rows in the matrix after Gauss-Jordan elimination have no dependence on the value of the X 's. The combination of rows that were XORed together to get the all-0 rows can be kept track of during Gauss-Jordan elimination. In Fig. 3, the first all-0 row is

obtained from $M_1 \oplus M_3 \oplus M_5$. This implies that if the MISR bits M_1 , M_3 , and M_5 are XORed together, all the X 's cancel out and the resulting value will have no dependence on the X 's. This can be seen by looking back at Fig. 1, and computing:

$$\begin{aligned} M_1 \oplus M_3 \oplus M_5 &= (X_1 \oplus O_3 \oplus O_8 \oplus O_{13}) \\ &\oplus (O_2 \oplus X_3 \oplus O_5 \oplus O_{10} \oplus O_{15}) \\ &\oplus (X_1 \oplus O_2 \oplus X_3 \oplus O_{12} \oplus O_{17}) \\ &= O_3 \oplus O_5 \oplus O_8 \oplus O_{10} \oplus O_{12} \oplus O_{13} \oplus O_{15} \oplus O_{17} \end{aligned}$$

As can be seen, after XORing these three bits together, the final equation depends only on non- X values. Similarly, each of the all-0 rows correspond to MISR bit combinations where the X 's cancel out. The values of these X -canceled MISR bit combinations are deterministic and can be predicted through simulation. During test, they can be compared with their fault-free values in order to detect errors.

Each X -canceled combination will depend on roughly half of the scan cells capturing non- X values due to the pseudo-random property of the MISR. Thus if q X -canceled combinations are checked, the error coverage will be approximately equal to $1-2^{-q}$. This provides error coverage equivalent to using a q -bit MISR with no unknowns for signature analysis. The error coverage for different values of q is shown in Table 1. If 7 X -canceled combinations are checked, then over 99% error coverage is obtained (which will generally correspond to even higher fault coverage since most faults are detected multiple times). If 13 X -canceled combinations are checked, then 99.99% error coverage is obtained.

Table 1. Error Coverage versus Number of X -Canceled Combinations (q)

X -Canceled Combinations (q)	Error Coverage
1	50%
2	75%
3	87.5%
4	93.75%
5	96.88%
6	98.44%
7	99.2%
8	99.6%
9	99.8%
10	99.9%
11	99.95%
12	99.97%
13	99.99%
14	99.994%
15	99.997%
16	99.998%

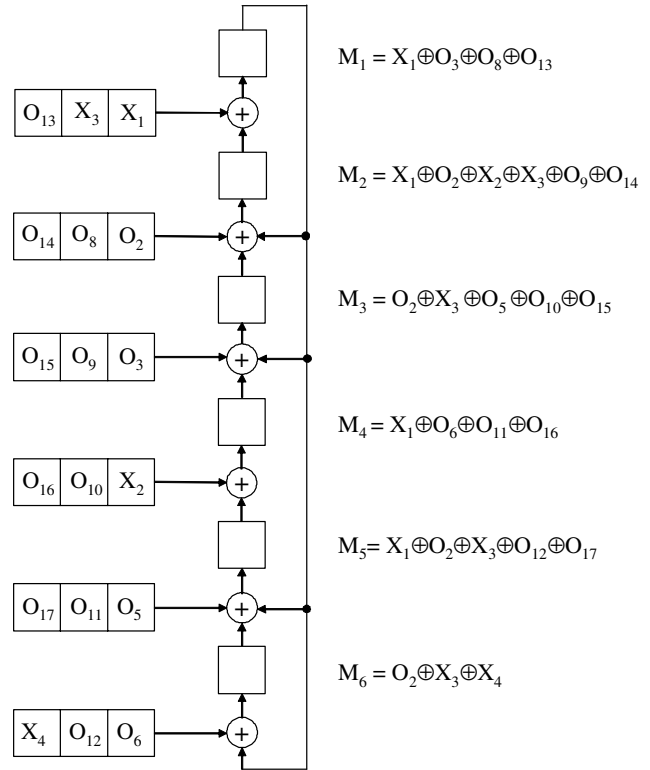


Figure 1. Example of Symbolic Simulation of MISR

$$\begin{aligned} M_1 &= X_1 \\ M_2 &= X_1 \oplus X_2 \oplus X_3 \\ M_3 &= X_3 \\ M_4 &= X_1 \\ M_5 &= X_1 \oplus X_3 \\ M_6 &= X_3 \oplus X_4 \end{aligned} \quad \rightarrow \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Figure 2. Linear Equations for MISR in Fig. 1

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \end{matrix} \quad \rightarrow \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} M_1 \\ M_1 \oplus M_2 \oplus M_3 \\ M_3 \\ M_3 \oplus M_6 \\ M_1 \oplus M_3 \oplus M_5 \\ M_1 \oplus M_4 \end{matrix}$$

Figure 3. Gauss-Jordan Reduction of MISR Equations

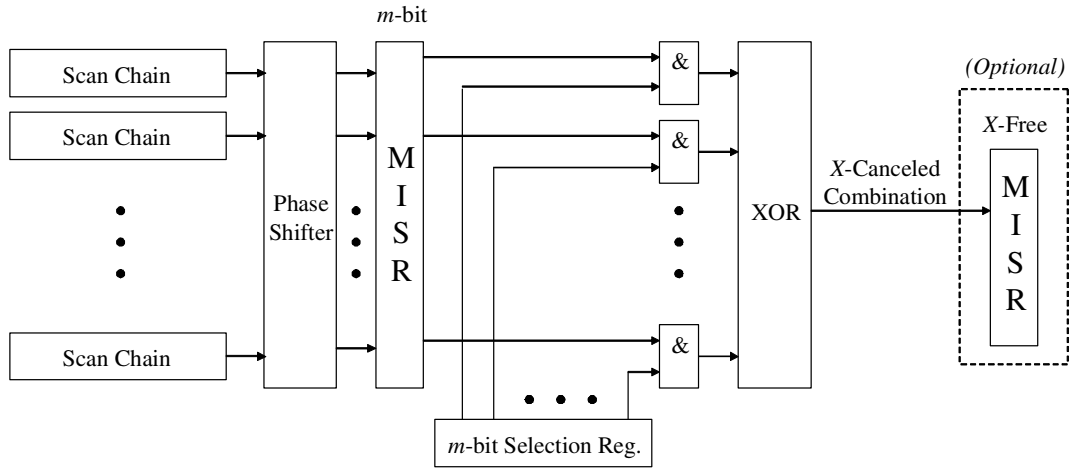


Figure 4. X-Canceling MISR Architecture

4. X-Canceling MISR Architecture

Fig. 4 shows the hardware for an X-Canceling MISR. The output response is scanned into an m -bit conventional MISR through a phase-shifter. The X-canceled combinations are generated using a programmable XOR network. Based on symbolic simulation (as described previously in Sec. 3), linearly dependent combinations of MISR bits are computed. Each combination is stored on the tester as an m -bit vector. The m -bit vector selects a set of MISR bits which are XORed together to generate an “X-free” value that is compacted in the X-free MISR. Note that the X-free MISR would only have a single input if only one programmable XOR is used, but it could also support multiple programmable XORs across multiple CUTs if desired. Since each X-free value has no dependence on the value of any X’s in the output response, the final signature in the X-free MISR at the end of the entire test is deterministic and can be directly compared with the fault-free value.

As shown earlier, the m -bit MISR can compact any number of non-X values, plus up to $m-q$ X’s with an error coverage of $1-2^{-q}$. Consider the case where a 256 bit MISR with a value of $q=10$ is used. In this case, the error coverage is over 99.97%, and up to 244 X’s can be compacted in the MISR along with any number of non-X outputs. The MISR can keep compacting output response data from the scan chains each clock cycle until the number of X’s received reaches up to 244. At that point, 12 X-canceled combinations need to be generated and shifted into the X-free MISR. Then the 256 bit MISR is reset (to all 0 or some constant value), and it can then begin to compact the output response data again. After resetting the MISR, up to another 244 X’s can be compacted. The frequency with which the X-canceled combinations need to be generated and the MISR needs to be reset depends on the density of X’s in the output

response data. For test sets with a low density of X’s, it may be possible to compact the output response for many scan vectors before the MISR has to be reset.

Note that the X-free MISR is optional and could be replaced by one channel going back to the tester. This would provide more diagnostic information.

A phase shifter is placed before the MISR. The purpose of this phase shifter is to eliminate shift correlation among the scan cells feeding into the MISR (and it can also be used to perform space compaction if the MISR is smaller than the number of scan chains). The reason why it is desirable to eliminate shift correlation of the output response in this case is to avoid the situation illustrated in Fig. 5 where an X and non-X value are directly loaded into the same MISR bit because in this case it is not possible to cancel out the X value without also canceling out the non-X value. In Fig. 5, it would not be possible to cancel X_i out without also canceling O_j out because they will always appear together in any final MISR bit equation. More discussion of shift correlation and systematic procedures for designing phase shifters to eliminate it can be found in [Rajski 00]. Any linear phase shifter network can be used with the proposed scheme. The phase shifter is simply included when performing the symbolic simulation described in Sec. 3 so that the final equations for each MISR bit factor in the phase shifter.

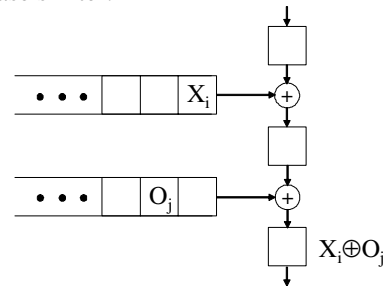


Figure 5. Shift Correlation

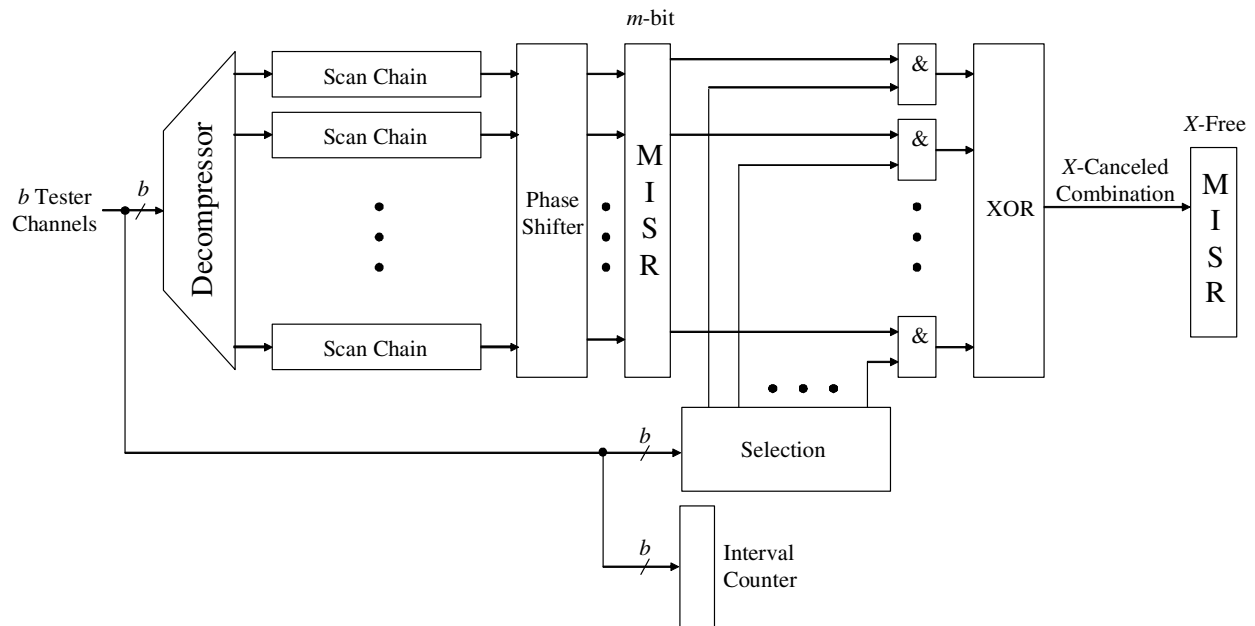


Figure 6. Scheme for Transferring Control Data from Tester via Halting Scan Shifting

Generating the X -canceled combinations requires receiving control data from the tester. There are a number of possible schemes for how to transfer this data in an efficient manner. This is the subject of the next section.

5. Transferring Control Data

For an m -bit MISR, generating each X -canceled combination requires m -bits of control data which must be supplied by the tester. If q X -canceled MISR bit combinations are used, then a total of mq bits of control data need to be transferred from the tester each time before the MISR is reset (i.e., for each intermediate signature).

There are a number of ways this data could be transferred. The simplest approach for generating the X -canceled combinations is to halt scan shifting whenever the MISR has compacted up to $m-q$ X 's and generate all the X -canceled combinations right at that point. After all the X -canceled combinations have been generated and compacted in the X -free MISR, then the MISR is reset and scan shifting can resume as normal. When the scan shifting is halted, the tester channels that are supplying the test stimulus can be utilized to supply the control data that selects the q X -canceled combinations. This is illustrated in Fig. 6. A total of mq bits need to be supplied to generate the q X -canceled combinations. If the number of tester channels is b , then it will take mq/b clock cycles to generate all the X -canceled combinations. An interval counter can be loaded at this time as well which would count down the number of shift cycles until the next time that the scan shifting should be halted. Assuming the

interval counter uses b bits or less, the total number of cycles that the scan shifting has to be halted to generate the X -canceled combinations plus load the interval counter would be $(mq/b)+1$. These cycles would add to the test time. However, no channels are needed to transfer output response back to the tester during the test session, and no channels are needed to transfer mask control data during scan shifting. All available channels can be used for transferring test stimulus during scan shifting, consequently more scan chains can be driven thereby reducing the number of scan shift cycles needed to apply each scan vector. In general, this will more than offset the extra $(mq/b)+1$ cycles needed to generate the X -canceled combinations. Since the tester bandwidth is begin fully utilized at all times, the test time is simply equal to the total amount of data stored on the tester divided by the tester bandwidth. Hence the proposed approach can reduce the overall test time considerably compared with existing output compaction schemes that provide equivalent error coverage as will be shown in Secs. 6 and 7.

If it is not desirable to stop scan shifting to process the intermediate signatures, an alternate approach would be to use a shadow register as shown in Fig. 7. In this case, when an intermediate signature is ready to be processed, the contents of the MISR are transferred to a shadow register, and the MISR is immediately reset so that scan shifting can continue uninterrupted. The saved signature in the shadow register is then processed to extract the X -canceled combinations as the next intermediate signature

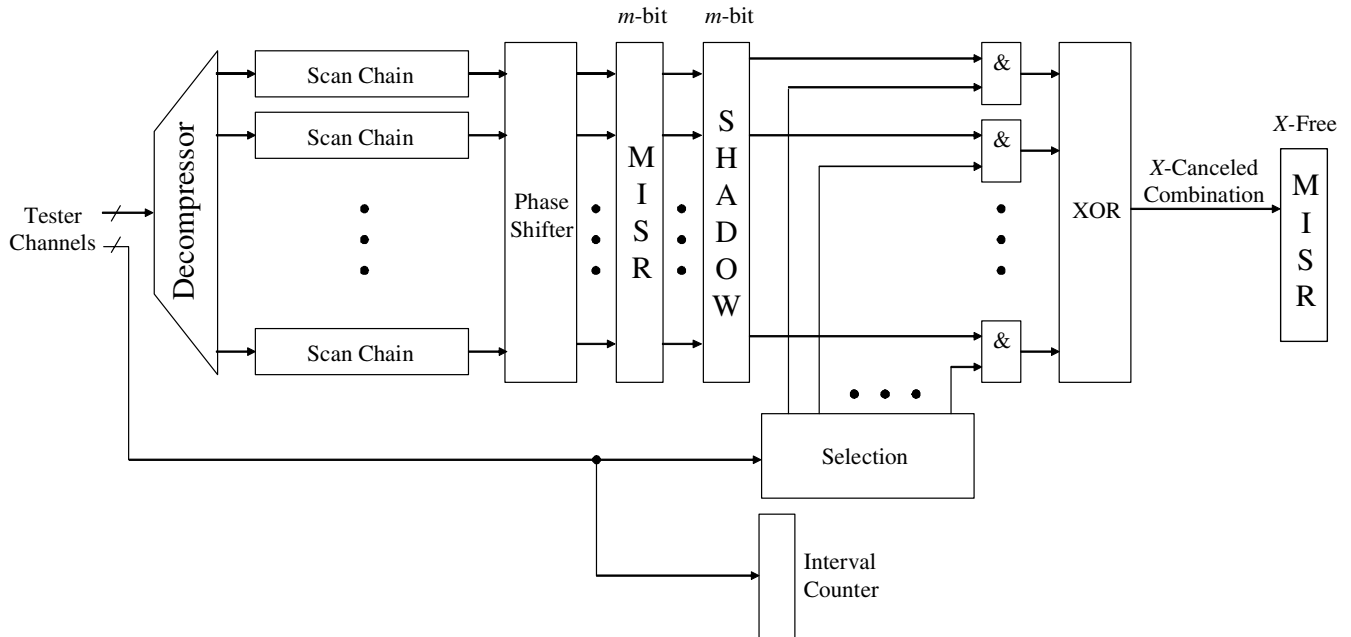


Figure 7. Scheme for Transferring Control Data from Tester using Shadow Register for Continuous Shifting

is being generated. Some tester channels are used to provide the control data that selects the X-canceled combinations. The drawback of this approach compared with the one in Fig. 6 is that a sufficient number of tester channels must be dedicated to supporting the selection logic so that the shadow register can be fully processed before the next intermediate signature is ready to be processed in the worst-case (i.e., the fewest number of clock cycles between intermediate signatures). There is some loss of efficiency depending on the difference between the worst-case and average-case number of clock cycles between intermediate signatures.

6. Analysis and Comparison

The amount of test storage required by the proposed approach depends only on the amount of control data that has to be supplied for generating the X-canceled combinations. No output response data is transferred to the chip during the test. If an m -bit MISR is used, it can compact $m-q$ X's where q is the number of X-canceled combinations used for each intermediate signature. The total number of intermediate signatures will be roughly equal to:

$$\text{Num. Intermediate Signatures} = \frac{\text{Total X's}}{m-q}$$

The total number of X's in the output response for the test set divided by $m-q$. q X-canceled combinations are generated for each intermediate signature, and each X-canceled combination requires m control bits to specify it. Thus, the total amount of control data required is:

$$\begin{aligned} \text{Total Control Data} &= (mq)(\text{Num. Intermediate Signatures}) \\ &= \frac{(mq)(\text{Total X's})}{m-q} \\ &\approx (q)(\text{Total X's}) \quad (\text{when } q \ll m) \end{aligned}$$

Note that the value of q is independent of the size of the scan architecture or the size of the MISR. It depends only on the desired error coverage. A value of 10 gives an error coverage of 99.9% independent of any other parameters. Thus, q is a constant that doesn't scale, and hence the control data is $O(\text{Total X's})$. This is very significant because it indicates that the amount of compression that is provided is independent of the scan architecture or the number of test vectors. The number of scan chains, scan cells, or scan length have no impact on the amount of tester storage. This is a very attractive feature in comparison with other output compaction schemes that can handle unknowns. The tester storage requirement for other schemes does scale with the size of the scan architecture and most also scale with the number of test vectors.

For conventional X-masking techniques, the mask control data will scale with the number of scan chains and/or scan length. If the X-masking resolution is sacrificed in order to reduce the mask control data, then the error coverage may be significantly reduced. For equivalent error coverage, the proposed X-canceling MISR scheme would generally provide much greater reductions in tester storage and bandwidth requirements compared with X-masking techniques.

For X -Compact [Mitra 04a], a combinational compactor with $m \approx \log_2(\text{number of scan chains})$ outputs is used. These outputs need to be transferred to the tester every clock cycle, so the test data bandwidth requirement scales logarithmically with the number of scan chains. However, the tester storage requirement depends on the scan length and number of test vectors. The tester storage requirement for X -Compact is:

$$\text{Tester storage} \approx 2(\text{num_vect})(\text{scan_len})\log_2(\text{scan_chains})$$

The tester storage includes the fault-free response as well as the masking data on the tester which is where the factor of 2 comes from. The error coverage for X -Compact depends on the distribution of X 's and the distribution of errors in the output stream. For maximum compaction, X -Compact would be designed to tolerate only one X per scan slice. The proposed X -canceling MISR scheme provides high error coverage regardless of the distribution of X 's.

Note that the tester storage requirement for other X -tolerant schemes (e.g., convolution compactors [Rajski 05] and X -filtering [Sharma 05]) also scale with the number of test vectors and the scan length.

7. Experimental Results

In the previous section, equations describing the tester storage for the proposed scheme were shown along with those for X -Compact [Mitra 04a]. In Table 2, the amount of output response compression that is obtained for output streams with different percentages of X 's are shown. These results are based on using a 256-bit MISR, however, smaller MISRs could also be used with a relatively minor decrease in test compression (e.g., using a 64-bit MISR would reduce the compression for $q=9$ by 12%). The first two columns show the percentage of X 's in the output stream and the number of scan chains. The third column shows the percentage of scan slices with more than one X . As the percentage of X 's goes up and the number of scan chains goes up, the percentage of scan slices with more than one X goes up as well which is important for X -Compact [Mitra 04a] and other combinational X -tolerant schemes. Generally an X -Compact network is designed only to be able to tolerate one X per scan slice, so as the percentage of scan slices with more than one X goes up, the error coverage is reduced. One of the major advantages of the proposed

Table 2. Results for Different Percentages of X 's

Percentage X 's	Scan Chains	Scan Slices w/more than One X	X -Compact [Mitra 04a]		Proposed X -Canceling MISR		
			Compacted Outputs	Compression Ratio	$q = 7$ 99.2% Cov.	$q = 9$ 99.8% Cov.	$q = 12$ 99.97% Cov.
0.001%	2048	0.02%	14	146x	13,895x	10,720x	7,942x
	4096	0.08%	15	273x	13,895x	10,720x	7,942x
	8192	0.3%	16	512x	13,895x	10,720x	7,942x
0.005%	1024	0.1%	13	79x	2,779x	2,144x	1,588x
	2048	0.5 %	15	146x	2,779x	2,144x	1,588x
	4096	1.8%	16	273x	2,779x	2,144x	1,588x
0.01%	512	0.1%	12	43x	1,390x	1,072x	794x
	1024	0.5%	13	79x	1,390x	1,072x	794x
	2048	1.8 %	14	146x	1,390x	1,070x	794x
0.05%	128	0.2%	10	12.8x	278x	214x	159x
	256	0.8%	11	23.3x	278x	214x	159x
	512	2.8%	12	42.7x	278x	214x	159x
0.1%	64	0.2%	9	7.1x	139x	107x	79x
	128	0.8%	10	12.8x	139x	107x	79x
	256	2.8%	11	23.3x	139x	107x	79x
0.5%	16	0.3%	6	2.7x	27.8x	21.4x	15.9x
	32	1.1%	7	4.6x	27.8x	21.4x	15.9x
	64	4.1%	9	7.1x	27.8x	21.4x	15.9x
1%	8	0.3%	5	1.6x	13.9x	10.7x	7.9x
	16	1.1%	6	2.7x	13.9x	10.7x	7.9x
	32	4.1%	7	4.6x	13.9x	10.7x	7.9x

Table 3. Fault Coverage for Different Numbers of X -Canceled Combinations per Intermediate Signature

Circuit	X -Canceled Combinations (q)	Error Coverage	Fault Coverage
s13207	7	99.2%	99.2%
	9	99.8%	99.8%
	11	99.95%	100%
s15850	7	99.2%	99.2%
	9	99.8%	99.9%
	11	99.95%	100%
s38417	7	99.2%	99.4%
	9	99.8%	99.9%
	12	99.97%	100%
s38584	7	99.2%	99.4%
	9	99.8%	99.95%
	12	99.97%	100%

X -canceling MISR is that the error coverage does not depend on the number of X 's in a scan slice. It depends only on the number of X -canceled combinations that are checked (i.e., q). In Table 2, the results that are shown for X -Compact include the number of compacted outputs and the corresponding compression ratio which is equal to the ratio of the number of scan chains to the number of compacted outputs. It is possible to increase the compression ratio for X -Compact by using more scan chains, however, at some point this begins to significantly reduce the error coverage because too many scan slices have more than one X . Results are shown for cases where the percentage of scan slices with more than one X stays below 5%. The compression for the proposed approach does not depend on the number of scan chains. It only depends on the number of X -canceled combinations that are used (i.e., q). The compression ratio for three different values of q are shown: 7, 9, and 12 which provide an error coverage of 99.2%, 99.8%, and 99.97%, respectively. As can be seen from the results, the proposed method provides much higher amounts of compression with much better error coverage. The proposed method is extremely efficient when the percentage of X 's is low.

In Table 3, results are shown for using the proposed X -Canceling MISR scheme on the output response for the largest ISCAS 89 benchmark circuits [Brglez 89]. X 's were randomly inserted in 0.1% of the output values, and the fault coverage was computed for checking different numbers of X -canceled combinations per intermediate signature. The error coverage is the percentage of errors that are detected in an intermediate signature, and the fault coverage is the percentage of faults that are detected. As can be seen, the fault coverage is generally the same or higher than the error coverage. 100% percent fault coverage was obtained for all circuits by checking no more than $q=12$ X -canceled combinations.

8. Conclusions

The proposed X -canceling MISR scheme offers a number of very attractive features. Its tester storage requirement depends only on the total number of X 's in the output response and is independent of the scan architecture, design size, or number of test vectors. So it scales extremely well and is extremely efficient when the percentage of X 's is small. No output response needs to be shifted to the tester, so all channels from the tester can be directed towards transferring stimulus. This is especially attractive for multi-site testing. The control data for generating the X -canceled combinations can be transferred very efficiently using the test stimulus channels. Since the test data bandwidth is always fully utilized, the test time reduction will scale directly with the test storage reduction that is provided. Lastly, the hardware overhead is very low and scales linearly with the size of the MISR that is used.

One limitation of the proposed X -canceling MISR scheme is that it is not effective when the percentage of X 's becomes large (e.g., greater than 5%). One interesting area for future research would be to look at hybrid approaches combining X -masking with the proposed X -canceling MISR. A potential advantage of a hybrid approach would be that the X -masking need not mask all X 's, it could target only a subset of the X 's that can be easily and efficiently masked. The residual X 's could be handled by the X -canceling MISR.

Even for situations where X -bounding or X -masking can eliminate all X 's, it still may be attractive to incorporate the proposed X -canceling MISR as a safety feature to handle any unexpected X 's or non-determinism that may arise late in the design cycle or in a post-silicon environment.

References

- [Barnhart 01] Barnhart, C., V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, "OPMISR: the Foundation for Compressed ATPG Vectors," *Proc. of International Test Conference*, pp. 748-757, 2001.
- [Brglez 89] Brglez, F., D. Bryan, and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," *Proc. of International Symposium on Circuits and Systems*, pp. 1929-1934, 1989.
- [Chao 05] M. C.-T. Chao, S. Wang, S.T. Chakradhar, and K.-T. Cheng, "Response Shaper: A Novel Technique to Enhance Unknown Tolerance for Output Response Compaction," *Proc. of International Conference on Computer-Aided Design*, pp. 80-87, 2005.
- [Chickermane 04] Chickermane, V., B. Foutz, and B. Keller, "Channel Masking Synthesis for Efficient On-Chip Test Compression," *Proc. of International Test Conference*, pp. 452-461, 2004.
- [Cullen 97] Cullen, C.G., *Linear Algebra with Applications*, Addison-Wesley, ISBN 0-673-99386-8, 1997.
- [Mitra 04a] Mitra, S., and K.S. Kim, "X-Compact: An Efficient Response Compaction Scheme," *IEEE Trans. on Computer-Aided Design*, Vol. 23, No. 3, pp. 421-432, Mar. 2004.
- [Mitra 04b] Mitra, S., S.S. Lumetta, and M. Mitzenmacher, "X-Tolerant Signature Analysis," *Proc. of International Test Conference*, pp. 432-441, 2004.
- [Patel 03] Patel, J.H., S.S. Lumetta, and S.M. Reddy, "Application of Saluja-Karpovsky Compactors to Test Responses with Many Unknowns," *Proc. of VLSI Test Symposium*, pp. 107-112, 2003.
- [Pomeranz 02] Pomeranz, I., S. Kundu, and S.M. Reddy, "On Output Response Compression in the Presence of Unknown Output Values," *Proc. of Design Automation Conference*, pp. 255-258, 2002.
- [Rajski 00] Rajski, J., N. Tamarapalli, and J. Tyszer, "Automated Synthesis of Phase Shifters for Built-In Self-Test Applications," *IEEE Trans. on Computer-Aided Design*, Vol. 19, No. 10, pp. 1175-1188, Oct. 2000.
- [Rajski 05] Rajski, J., J. Tyszer, C. Wang, and S.M. Reddy, "Finite Memory Test Response Compactors for Embedded Test Applications," *IEEE Trans. on Computer-Aided Design*, Vol. 24, No. 4, pp. 622-634, Apr. 2005.
- [Rajski 06a] Rajski, J., J. Tyszer, G. Mrugalski, W.-T. Cheng, N. Mukherjee, and M. Kassab, "X-Press Compactor for 1000x Reduction of Test Data," *Proc. of International Test Conference*, Paper 18.1, 2006.
- [Rajski 06b] Rajski, W., and J. Rajski, "Modular Compactor of Test Responses," *Proc. of VLSI Test Symposium*, pp. 242-251, 2006.
- [Saluja 83] Saluja, K.K., and M. Karpovsky, "Testing Computer Hardware Through Test Data Compression in Space and Time," *Proc. of International Test Conference*, pp. 83-89, 1983.
- [Sharma 05] Sharma M. and W.-T. Cheng, "X-Filter: Filtering Unknowns from Compacted Test Responses," *Proc. of International Test Conference*, Paper 42.1, 2005.
- [Tang 06] Tang, Y., H.-J. Wunderlich, P. Engelke, I. Polian, B. Becker, J. Scholöffel, F. Hapke, and M. Wittke, "X-Masking During Logic BIST and Its Impact on Defect Coverage," *IEEE Trans. on VLSI*, Vol. 14, No. 2, Feb. 2006.
- [Volkerink 05] Volkerink, E.H., and S. Mitra, "Response Compaction with Any Number of Unknowns Using a New LFSR Architecture," *Proc. of Design Automation Conference*, pp. 117-122, 2005.
- [Wang 06] L.T. Wang, C.-W. Wu, X. Wen, *VLSI Test Principles and Architectures*, Morgan Kaufmann, 2006.
- [Wohl 01] Wohl, P., J.A. Waicukauski, and T.W. Williams, "Design of Compactors for Signature-Analyzers in Built-In Self-Test," *Proc. of International Test Conference*, pp. 54-63, 2001.
- [Wohl 03] Wohl, P., J.A. Waicukauski, S. Patel, and M.B. Amin, "X-Tolerant Compression and Application of Scan-ATPG Patterns in a BIST Architecture," *Proc. of International Test Conference*, pp. 727-736, 2003.
- [Wohl 04] Wohl, P., J.A. Waicukauski, and S. Patel, "Scalable Selector Architecture for X-Tolerant Deterministic BIST," *Proc. of Design Automation Conference*, pp. 934-939, 2004.