Enhancing Superset *X*-Canceling Method With Relaxed Constraints on Fault Observation

Joon-Sung Yang, *Member, IEEE*, Jinsuk Chung, and Nur A. Touba, *Fellow, IEEE*

***Abstract*—An *X*-tolerant multiple-input signature register (MISR) compaction methodology that compacts output streams containing unknown (*X*) values, called *X*-canceling, is an alternative to masking *X* values (i.e., *X*-masking). A number of control bits that is linear in the number of *X*'s to be canceled are required to perform the *X*-canceling operation for existing *X*-canceling approaches. This paper proposes a new *X*-canceling method significantly reducing the number of control bits for *X*-canceling. We exploit the fact that 1) unknown values tend to be highly correlated in the scan cells (i.e., *X*'s tend to be generated in certain portions of design) and 2) fault effects can typically be observed in a multiplicity of scan cells. Instead of custom generating the control bits to cancel out only the *X*'s in one MISR signature, the proposed approach finds a general superset solution which can cancel out the *X*'s for many MISR signatures without losing fault coverage. This allows the same control bits to be reused many times thereby significantly improving the amount of compression that can be obtained. Architectures for implementing superset *X*-canceling are described along with experimental results.**

***Index Terms*—Control bits, output response compaction, superset unknown value generation correlation, *X*-canceling multiple-input signature register (MISR).**

I. Introduction

GROWING design size and complexity in integrated circuits result in longer test times and exploding test vector volume. To alleviate the issues, test stimulus compression and test response compaction are used. Unknown values, "*X*" values, cause a major issue in compacting output streams for test compression and built-in self-test. *X* values are introduced by sources such as uninitialized memory elements, but contention, floating tri-states, etc. In compacting output responses, *X* values make the final signature corrupted and nondeterministic. Hence, *X* values can directly impact fault coverage [27]. There have been a number of approaches proposed to handle the issues caused by *X*'s in the output response.

A number of schemes have been developed to deal with the problem of *X*'s in the output response. One way of controlling *X*'s is to modify the circuit-under-test (CUT) so that it does not generate *X* values. This approach is called *X*-bounding or *X*-blocking and requires adding design-for-testability logic to prevent *X*-value propagation to scan cells [1], [3], [19]. In this method, *X* sources are driven to 0 (0—control point) or 1 (1—control point). However, since this involves the modification of CUT, the inherent problems of the method are the increase in design area and the potential timing issues. Another approach, which does not require modifying the CUT, is *X*-masking that masks out *X*'s at the input to the output response compactor. Masking signals are transferred through tester channels and they are used to specify which scan chain outputs should be masked during which clock cycles [1]–[3], [11], [13], [14], [17], [20]–[22]. A third approach is to design an *X*-tolerant compactor that can compact an output stream that contains *X*'s without the need for *X*-masking. *X*-tolerant compactors have been developed based on linear combinational compactors [6], [7], [15], [16] that are mainly based on the application of systematic linear codes. Convolutional compactors [12] and circular registers [3], [6], [7] can tolerate a certain amount of *X* values. Although multiple-input signature registers (MISRs) are the most efficient for compacting output streams without *X*'s, they present difficulties when *X*'s are present because even a single *X* can corrupt the MISR contents with its sequential nature in accumulating its signature [6], [7].

An *X*-canceling MISR method [18], [26], [27] was proposed using the concept of canceling out *X*'s from MISR signatures. This method can achieve arbitrarily high error coverage very efficiently where error coverage is the percentage of scan cells that are observed in the presence of *X*'s. Symbolic simulation is used to express each bit of the MISR signature as a linear equation in terms of the *X*'s. Linearly dependent combinations of MISR signature bits are identified with Gaussian elimination and are XORed together 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. In this method, the MISR keeps compacting *X*'s until it fills up (i.e., the number of *X*'s approaches the size of the MISR). At that point, Gaussian elimination is performed to remove *X* values in the MISR. Control bits need to be sent from a tester to find a custom solution for *X*-canceling operation.

This paper proposes an *X*-canceling MISR method which can significantly reduce number of control bits (preliminary results were reported in [25]). We exploit the fact that *X*'s

Manuscript received November 7, 2013; revised April 22, 2014 and September 14, 2014; accepted November 15, 2014. Date of publication July 21, 2015; date of current version January 19, 2016. This work was supported by the Samsung Research Fund, Sungkyunkwan University, 2015. This paper was recommended by Associate Editor L.-C. Wang.

J.-S. Yang is with the Department of Semiconductor Systems Engineering, Sungkyunkwan University, Suwon, Korea (e-mail: js.yang@skku.edu).

J. Chung is with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 USA (e-mail: chungdna@gmail.com).

N. A. Touba is with the University of Texas at Austin, Austin, TX 78712 USA (e-mail: touba@ece.utexas.edu).

Digital Object Identifier 10.1109/TCAD.2015.2459035

tend to be generated in certain portions of design such that certain scan cells capture many *X*'s while other scan cells may not capture any. The proposed method tries to find a superset *X*-canceling technique using *X* location correlations. Whereas the preliminary work in [25] constrained *X*-canceling so that all fault effects were guaranteed to be observed; in this paper, we relax the constraints so only one fault effect per fault needs to be guaranteed to be observed. This is sufficient to detect all faults, but allows better optimization for reducing control bits.

Compressing output response with high *X* densities is considered a major challenge for achieving the high compression that will be needed for future generations of designs as technology continues to scale. The superset *X*-canceling technique described here provides a very efficient and elegant way to exploit *X*-correlations to address this problem. There have been some techniques developed recently to exploit the locality of *X*'s for more efficient *X*-masking before a compactor with fewer mask control bits [3], [5]. These methods are coupled with automatic test pattern generation (ATPG) to minimize the number of mask control bits without losing fault coverage. In [5], the terms "intracorrelation" and "intercorrelation" of *X*'s are used to refer to correlations of *X*'s within a single output response and across multiple output responses, respectively. Both are enhanced and exploited to reduce mask control bits. While these methods are performing masking before the compactor and need to consider masking at every clock cycle (i.e., in each scan slice), the proposed approach has an inherent advantage in that no masking is performed before compaction. A large number of scan slices are compacted together into a signature, and then the *X*'s are canceled out of the signature after compaction. The location of the *X*'s within the scan slices and between scan slices is not a concern for the proposed method, and so all the focus can be placed on enhancing and exploiting the intercorrelations of *X*'s between the output responses of different test vectors. The intracorrelations are not relevant.

This paper is organized as follows. Section II provides an overview of the proposed method. Section III discusses details of superset *X*-canceling with scan vectors. Superset *X*-canceling with partial responses is described in Section IV and the hardware architectures are given in Section V. Experimental results and conclusions are addressed in Section VI and Section VII, respectively.

II. Overview of Proposed Method

A. Overview of X-Canceling MISR

This section gives a brief overview of the operation of an *X*-canceling MISR. A more detailed explanation can be found in [27].

Assume the output response has been captured in the scan chains after applying a test vector. As illustrated in Fig. 1, the value in each scan cell can be represented by a symbol. Via symbolic simulation, the final state of the MISR can be expressed in terms of symbols after the output response has been shifted in to the MISR. Each MISR bit is represented by a linear equation of the scan cell symbols. The final value of the top bit (M_1) of the MISR is $X_1 \oplus O_3 \oplus O_8 \oplus O_{13}$, where X_i denotes an X value and O_i indicates a non-X value.

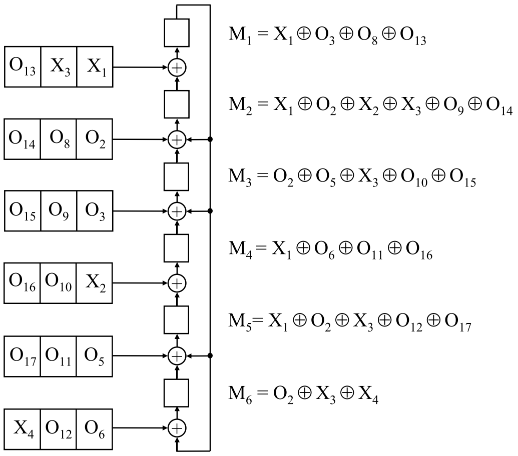

O_{13} X_3 X_1
O_{14} O_8 O_2
O_{15} O_9 O_3
O_{16} O_{10} X_2
O_{17} O_{11} O_5
X_4 O_{12} O_6
$M_1 = X_1 \oplus O_3 \oplus O_8 \oplus O_{13}$
$M_2 = X_1 \oplus O_2 \oplus X_2 \oplus X_3 \oplus O_9 \oplus O_{14}$
$M_3 = O_2 \oplus O_5 \oplus X_3 \oplus O_{10} \oplus O_{15}$
$M_4 = X_1 \oplus O_6 \oplus O_{11} \oplus O_{16}$
$M_5 = X_1 \oplus O_2 \oplus X_3 \oplus O_{12} \oplus O_{17}$
$M_6 = O_2 \oplus X_3 \oplus X_4$

Fig. 1. Example of symbolic simulation of 6-bit MISR.

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

Fig. 2. Linear equations for MISR in Fig. 1.

Without loss of generality, assume all the O_i values in the output response are 0 so that each MISR bit is now simply equal to the linear combination of the X values. These linear combinations can be expressed in the form of a matrix as shown in Fig. 2. Each entry in the matrix has a 1 if the MISR bit corresponding to the row depends of the *X* corresponding to the column.

If the number of columns is less than the number of rows, i.e., the number of *X*'s is less than the MISR size, then some row combinations will be linearly dependent. Gauss–Jordan elimination [4] can be performed on the matrix in Fig. 2 to identify the linearly dependent combinations of rows as illustrated in Fig. 3. The last two rows in Fig. 3 have all 0's and this indicates combinations of MISR bits in which all the *X*'s cancel out. The first all-0 row corresponds to $M_1 \oplus M_3 \oplus M_5$. This implies that XORing MISR bits M_1, M_3, and M_5 generates an "*X*-canceled" signature bit which depends only on scan cells that captured non-*X* values as shown in the following:

$$M_1 \oplus M_3 \oplus M_5 = O_3 \oplus O_5 \oplus O_8 \oplus O_{10} \\ \times \oplus O_{12} \oplus O_{13} \oplus O_{15} \oplus O_{17}.$$

The values of these *X*-canceled MISR bit combinations are deterministic and can be predicted through simulation. Therefore, during test, they can be compared with their fault-free values in order to detect errors.

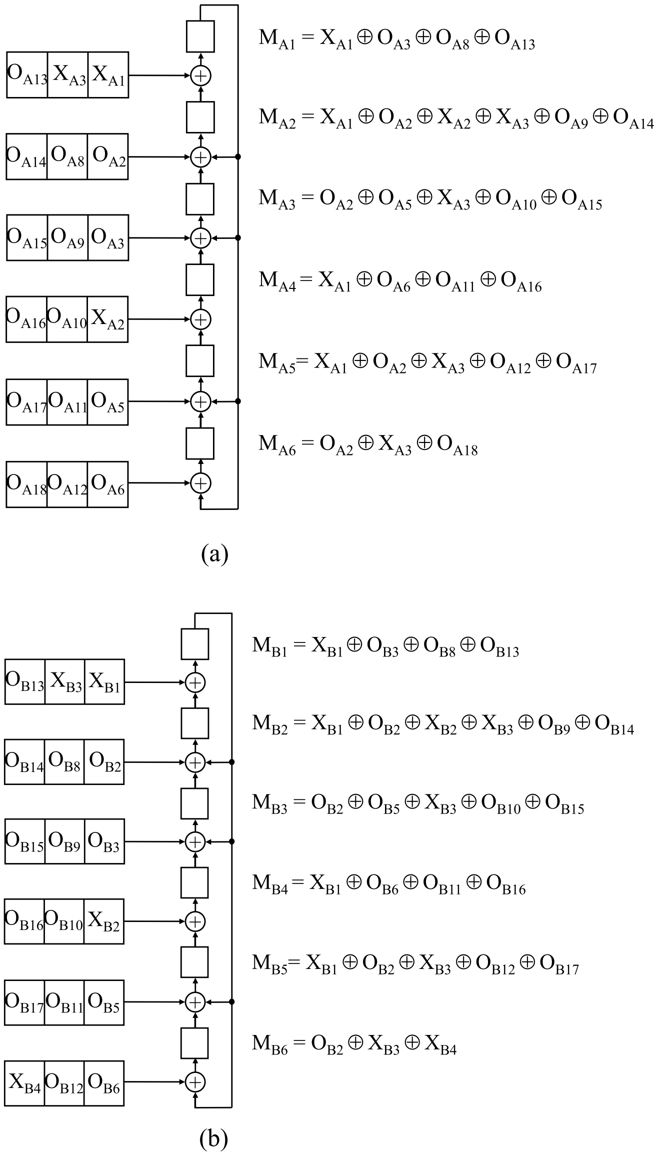
The MISR is operated across many clock cycles and may span multiple test vectors until the MISR fills up with *X*'s. The MISR signature is then processed by selectively XORing

$$\begin{array}{cccc|l} X_1 & X_2 & X_3 & X_4 & \\ \hline 1 & 0 & 0 & 0 & M_1 \\ 1 & 1 & 1 & 0 & M_2 \\ 0 & 0 & 1 & 0 & M_3 \\ 1 & 0 & 0 & 0 & M_4 \\ 1 & 0 & 1 & 0 & M_5 \\ 0 & 0 & 1 & 1 & M_6 \end{array} \xrightarrow{\text{Gaussian Elimination}} \begin{array}{cccc|l} X_1 & X_2 & X_3 & X_4 & \\ \hline 1 & 0 & 0 & 0 & M_1 \\ 0 & 1 & 0 & 0 & M_1 \oplus M_2 \oplus M_3 \\ 0 & 0 & 1 & 0 & M_3 \\ 0 & 0 & 0 & 1 & M_3 \oplus M_6 \\ \hline 0 & 0 & 0 & 0 & M_1 \oplus M_3 \oplus M_5 \\ 0 & 0 & 0 & 0 & M_1 \oplus M_4 \end{array}$$

X-Free Rows

Fig. 3. Gauss–Jordan elimination of MISR equations and *X*-free rows.

linearly dependent combinations of MISR bits in terms of the *X*'s to generate *X*-free output response to send to the tester. The error coverage can be made arbitrarily high by generating and checking a sufficient number of *X*-canceled output responses. The probability of not detecting an error drops by a factor of 2 for each *X*-canceled combination that is checked. If q *X*-canceled combinations are checked, then the error coverage for will be $1 - 2^{-q}$. So if $q = 7$, then the error coverage will be 99.2%, and each MISR signature can capture up to $(m - 7)$ *X*'s where m is the size of the MISR.



O_A13 X_A3 X_A1
O_A14 O_A8 O_A2
O_A15 O_A9 O_A3
O_A16 O_A10 X_A2
O_A17 O_A11 O_A5
O_A18 O_A12 O_A6
$M_{A1} = X_{A1} \oplus O_{A3} \oplus O_{A8} \oplus O_{A13}$
$M_{A2} = X_{A1} \oplus O_{A2} \oplus X_{A2} \oplus X_{A3} \oplus O_{A9} \oplus O_{A14}$
$M_{A3} = O_{A2} \oplus O_{A5} \oplus X_{A3} \oplus O_{A10} \oplus O_{A15}$
$M_{A4} = X_{A1} \oplus O_{A6} \oplus O_{A11} \oplus O_{A16}$
$M_{A5} = X_{A1} \oplus O_{A2} \oplus X_{A3} \oplus O_{A12} \oplus O_{A17}$
$M_{A6} = O_{A2} \oplus X_{A3} \oplus O_{A18}$
(a)
O_B13 X_B3 X_B1
O_B14 O_B8 O_B2
O_B15 O_B9 O_B3
O_B16 O_B10 X_B2
O_B17 O_B11 O_B5
X_B4 O_B12 O_B6
$M_{B1} = X_{B1} \oplus O_{B3} \oplus O_{B8} \oplus O_{B13}$
$M_{B2} = X_{B1} \oplus O_{B2} \oplus X_{B2} \oplus X_{B3} \oplus O_{B9} \oplus O_{B14}$
$M_{B3} = O_{B2} \oplus O_{B5} \oplus X_{B3} \oplus O_{B10} \oplus O_{B15}$
$M_{B4} = X_{B1} \oplus O_{B6} \oplus O_{B11} \oplus O_{B16}$
$M_{B5} = X_{B1} \oplus O_{B2} \oplus X_{B3} \oplus O_{B12} \oplus O_{B17}$
$M_{B6} = O_{B2} \oplus X_{B3} \oplus X_{B4}$
(b)

Fig. 4. Example of output responses and unknown value correlation. Symbolic response representation for (a) Pattern A and (b) Pattern B.

B. General Idea of Superset X-Canceling

The idea in this paper is to exploit the fact that the scan cells that capture *X*'s are highly correlated. *X*'s tend to be generated in certain portions of the design such that some scan cells capture many *X*'s while others may not capture any. To illustrate the concept used in this paper, consider a simple example where the output response for each scan vector is compacted into a single MISR signature. If the location of the *X*'s for two output responses are identical, then the same set of control bits can be repeated to cancel out the *X*'s in both MISR signatures. Now suppose output response A has a subset of the *X*'s in output response B. In this case, again the same set of control bits used for B can also be used for A since it will cancel out a superset of the *X*'s in A. Fig. 4 shows two output responses by two patterns, A and B. As can be seen, the location of *X*'s by A and B has a correlation between patterns. The difference between output responses is found at O_{A18} and X_4 locations, hence, the rest of X values can be canceled by the same control bits for A and B. In this manner, superset *X*-canceling is found.

Now suppose output response A and output response B have 90% of their *X*'s in the same locations, but 10% of their *X*'s are in different locations. In this case, Gaussian elimination can be used to find a superset solution which cancels all the *X*'s in both A and B including the 90% that are the same and the unique 10% for each of A and B. The control bits for this solution can be repeated for both A and B. If using a custom solution for each of A and B required n bits each, then the total number of control bits for both A and B would be $2n$. However, the combined superset solution using the proposed method will contain only $(0.9n + 0.1n + 0.1n = 1.2n)$ bits. Note that when using a superset solution, care must be taken that the non-*X* values that are canceled do not cause a loss of fault coverage. This will be discussed in detail in Section III. Using superset solutions provides a way to significantly reduce the total number of unique control bits needed for output response compaction and thereby reduce tester storage requirements, and depending how it is implemented also test data bandwidth requirements.

III. Overview of Proposed Method

The proposed idea for superset *X*-canceling is to use the same set of control bits for processing multiple MISR signatures. In this section, the simple case where one MISR signature is generated for each scan vector is considered first. This assumes the size of the MISR is larger than the maximum number of *X*'s in any output response for a scan vector. A more general solution for any size MISR and number of *X*'s is described in Section IV.

A. Response Merging Process Without Fault Coverage Consideration

As described in Section II-A, canceling out the X's in one MISR signature is done by using Gaussian elimination to identify MISR bit combinations that are linearly dependent in terms of all the X's captured by the MISR signature. So the control bits that are generated through this procedure will cancel all the X's. To use the same set of control bits for two MISR signatures, the X locations for both MISR signatures can be merged together when performing Gaussian elimination. Consider the following example where the X-locations in the output response for two test vectors are the following.

Scan cell:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Response 1:	1	X	0	1	X	0	1	X	0	1	X	X	0	X
Response 2:	0	X	X	1	X	1	1	X	0	0	X	X	0	0

The X-locations can be merged as follows (only the X's are of concern).

Scan cell:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Merged response:	−	X	X	−	X	−	−	X	−	−	X	X	−	X

Responses 1 and 2 each have 6 X's (Response 1: scan cell position 2, 5, 6, 11, 12, and 14 and Response 2: scan cell position 2, 3, 5, 8, 11, and 12). By ORing the Responses 1 and 2, a merged response is generated. The merged response has 7 X's (scan cell position 2, 3, 5, 8, 11, 12, and 14). As long as the number of X's in the merged response does not exceed the maximum number of X's that can be canceled in the MISR (which depends on the size of the MISR as described in Section II-A), it is possible to find a solution for the control bits which will cancel all the X's in the merged response. Additional responses can be merged in as well provided the total number of X's can still be handled by the MISR. If seven X-canceled combinations are used, then an m-bit MISR can capture up $(m - 7)$ X's and provide more than 99% error coverage of the non-X values.

B. Response Merging With Fault Coverage Consideration

When superset X-canceling response merging process is performed, the number of scan cells whose values are canceled out is greater than the number of X's in the output response, so some non-X values are getting canceled as well. From the example in Section III-A., while merged response has 7 X's, Responses 1 and 2 have 6 X's. If the control bits for merged response are used for Response 1, then there is one scan cell (scan cell 3) that captures a non-X value which will be canceled out for all X-canceled combinations. Hence, no errors can be observed in those cells. So when merging responses together for superset X-canceling, it is necessary to check that fault coverage will not be lost. There are two ways to do this.

1) Perform fault simulation for the test vector assuming no observation for the canceled scan cells and verify that no coverage is lost.
2) Find the necessary scan cells for capturing the fault effects when the original fault simulation is performed. And perform the response merge process if no necessary scan cells get canceled.

In the first approach, fault simulation is performed for the test vector assuming no observation for the canceled scan cells to verify that no coverage is lost. In the second approach, there are some scan cells that must be observed in order to detect the necessary faults for a particular test pattern, and they will be referred as D-values. Hence, the response merging process only can be done if a D-value is not removed by response merge. Consider the following example.

Scan cell:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Response A:	0	0	X	1	X	1	1	X	D	0	X	X	0	0
Response B:	0	X	D	1	X	1	1	X	X	0	X	D	0	0
Response C:	1	X	D	1	X	0	1	X	0	1	X	X	0	X

There are three responses (A, B, C) with X's and D's. Response merging process in the previous section tries to find a superset among three responses; however, it fails to merge the responses due to conflicts between D's and X's. Scan cell position 3 in Responses B and C, scan cell position 9 in Response A, and scan cell position 12 in Response B have a D and X conflict. As addressed above as a second bullet, merging them may result in a fault coverage loss.

C. Enhancing Response Merging With Fault Coverage Consideration by Unique Scan Cells Alive for Fault Detection

Here, we propose a way to further enhance a test compression. Through fault simulation, some scan cells that must be observed for fault detection can be recorded with respect to the corresponding faults. Three responses (A, B, C) shown above have 4 D's; however, we can further distinguish each D using fault simulation information. A scan cell to detect the fault, $f1$, is represented as D_{f1} and the Responses B and C have D_{f1}. In the same manner, D_{f2} for the fault, $f2$, is found from Responses A and B. The following shows responses with D's with respect to the fault lists.

Scan cell:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Response A:	0	0	X	1	X	1	1	X	D_{f2}	0	X	X	0	0
Response B:	0	X	D_{f2}	1	X	1	1	X	X	0	1	D_{f1}	X	0
Response C:	1	X	D_{f1}	1	X	0	1	X	0	1	X	X	0	X

Here, we try to find a superset even though there are D's (D_{f1} and D_{f2}) and X conflict by exploiting a fact that scan cells to detect the necessary fault are very likely to be found multiple times by multiple test patterns. Hence, as long as there is at least one D alive for each fault (D_{f1} and D_{f2} in the above example) after merging, the proposed superset X-canceling method does not lose fault coverage. From the above example, D_{f1} is found at the 12th and the third scan cell for Responses B and C respectively. D_{f2} is found at the ninth scan cell in Response A and the second scan cell in Response B. We exploit the fact that D_{f1} and D_{f2} exist multiple times by different test patterns gives an opportunity to perform a response merging process.

One possible superset is to merge Responses A and B. If they are merged, D_{f1} and D_{f2} disappear because D_{f2} and D_{f1} are merged with X at scan cell location 3, 9, and 12. After the merge process, even though D_{f1} is still alive in Response C,

D_{f2} is not alive in any remaining responses. Therefore, this will result in a fault coverage loss and this merge is discarded. The following shows the merged response and response C.

Scan cell:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Merged response:	–	X	X	–	X	–	–	X	X	–	X	X	X	0
Response C:	1	X	D_{f1}	1	X	0	1	X	0	1	X	X	0	X

Another possible superset is to merge Responses B and C. D_{f1} at 12th scan cell in Response B is removed; however, D_{f1} is still alive at the third scan cell in the merged response. D_{f2} is also found in both Response A and merged response. Hence, Responses B and C can be used to form a superset. D does not disappear as long as D is merged with X. In this manner, we can find a superset X-canceling solution in the presence of scan cells to detect the necessary fault without any loss of fault coverage. The proposed response merge process can be done with entire responses and this helps to find supersets for reducing control bits significantly. From the example, the new merge process gives responses as follows (showing only D and X locations).

Scan cell:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Response A:	0	0	X	1	X	1	1	X	D_{f2}	0	X	X	0	0
Merged Response:	–	X	$D_{f2}D_{f1}$	–	X	–	–	X	X	–	X	X	X	X

In this manner, the proposed method enhances a superset X-canceling method with relaxed constraints on fault observations. We relax the constraints so only one fault effect per fault needs to be guaranteed to be observed. This is sufficient to detect all faults, but allows better optimization for reducing control bits.

D. Response Merging Algorithm With Unique Scan Cells Alive for Fault Detection

The procedure for selecting which output responses to merge together is a basic clustering algorithm where the goal is to minimize the number of merged groups. Output responses can be merged together provided that the number of unique X locations in the merged set is within the amount that the MISR can handle. A greedy procedure can be used as described below:

The merge algorithm selects the output response with the most X's as the seed for the merged cluster. It compares all output responses in the cluster and identifies a candidate when merged with cluster would increase the number of X's the least. On top of the basic principle, the merge algorithm also checks if required D's are alive after merging with a candidate response. Essentially, while the algorithm tries to merge a response with the cluster that will increase the number of X's the least, it always try to maintain the fault coverage by avoiding merging process if the merge destroys any D's that exists only in the source response. To elaborate, while we merge the candidates, they may prevent observation of any D's. To handle this issue, as proposed in the previous section, we keep tracking D's aliveness. In the cluster, if there is a conflict between X and D, the algorithm searches whether there are any D's available in other responses to observe the same fault. If there are, responses are merged regardless of the conflict because the D that is alive in different response can be used to detect the related fault and this will keep the fault coverage unchanged. If D's about to be destroyed due to the merge, the merge algorithm drops the candidate and it moves on to the next candidate responses. Finally, if the number of unique X's in the merged cluster after adding the candidate can be handled by the MISR, then add the candidate to the cluster and we rerun the process for the next candidate. However, if no more output responses can be added to the cluster, the candidate is evaluated with the other clusters already exists. If the candidate cannot be merged with any of the clusters, the merging algorithm creates new cluster with the candidate in it. This flow continues to cover all responses.

Algorithm 1 Finding Merged Response

```
Input: Output Response with X's and D's
Output: Merged Output Responses
 1: Find an output response with the most X's as a seed for
    merged cluster
 2: Compare output responses in the cluster
 3: for compare all output responses in the cluster do
 4:    if merging candidate X cancels D then
 5:       if D is the only observable cell then
 6:          Find other response
 7:       else if other D's found at the same location then
 8:          perform merge response
 9:       end if
10:       else then
11:       perform merge response
12:    end if
13: end for
14: if MISR can handle more X's than merged response then
15:    add more responses to the cluster
16:    go to line 3
17:    else then
18:    identify a merged response candidate
19:    form next cluster
20:    go to line 3
21: end if
```

This procedure will find a good solution, but is not guaranteed to find the best solution. Other heuristic clustering algorithms can be adapted for this problem as well [24].

The amount of merging that can be performed will depend on how correlated the location of the X's are across the output responses. The distribution of X's for industrial circuits used in Section VI is illustrated in Fig. 5. The scan cells are plotted along the x-axis sorted in descending order from the ones capturing the most X's down to the ones capturing the least X's. The y-axis shows how many X's are captured in that scan cell during the application of 3000 test vectors. There are 505 050, 36 075, and 97 643 scan cells in CKT-A, CKT-B, and CKT-C, respectively. For CKT-A, only 2019 scan cells capture X's and 90% of all the X's are captured in 0.09% of the entire scan cells. In CKT-B, X's are capture only by 3903 scan cells and 4.9% of scan cells capture 90% of X's. In CKT-C, 17 073 scan cells capture X's and 90% of all the X's

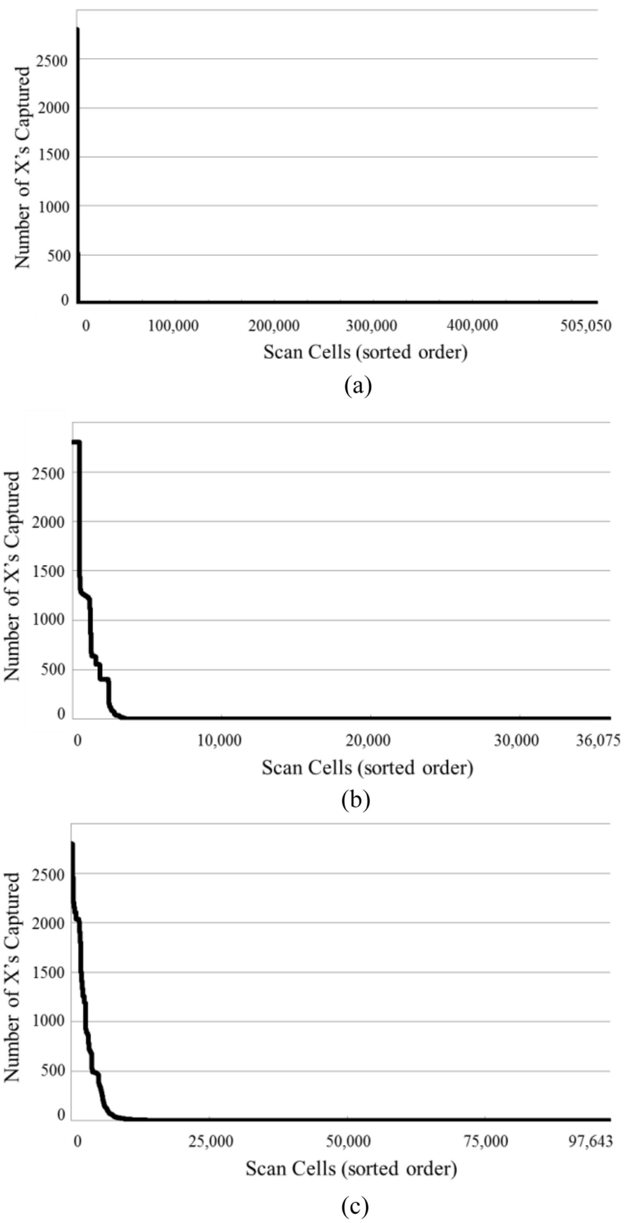


Fig. 5. Distribution of *X*'s captured in scan cell of industrial circuits. (a) CKT-A *X*, (b) CKT-B *X*, and (c) CKT-C *X* capturing scan cell distributions.

are captured in 4.8% of the scan cells. As shown in Fig. 5, there is a high locality correlation in the *X*'s location. This high degree of *X* locality can be efficiently exploited by the proposed method. Obviously, the degree of correlation of the *X*'s is very design dependent, but the structural dependence of *X* generation in circuits generally leads to a highly skewed distribution. Note also that the advanced ATPG methodology described in [5] can be adapted for the proposed method as well to enhance the amount of intercorrelation while avoiding conflicts with *D*'s.

IV. Superset *X*-Canceling for Partial Responses

Section III describes the simple case where one MISR signature is used for each test vector. However, this would work well only if the MISR is sufficiently large enough to be able to store all the *X*'s in any output response. This section proposes a general scheme that can be used for any size MISR, any size scan architecture, and any number of *X*'s and *D*'s. The idea is to use multiple MISR signatures for the output response of each scan vector. Fig. 6 illustrates the superset *X*-canceling for partial responses. The first set of scan slices is compacted in the first MISR signature, then the second set of scan slices is compacted in the next MISR signature, and so forth. The same partitioning of the scan slices into MISR signatures is used for all scan vectors so that correlations in the *X* locations in the *p*th partition of one scan vector will match up with correlations in the *X* locations in the *p*th partition of another scan vector.

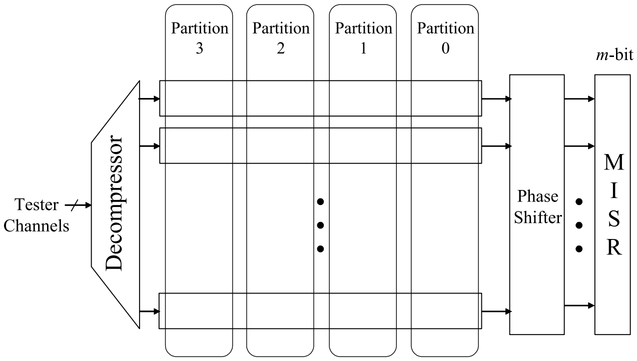


Fig. 6. Partitioning scan slices (one MISR signature generated for each partition).

A. On-Chip Stored Control Bits

Merging of the output responses is done individually for each *p*th partition. For example, the output responses in the first partition of all scan vectors are merged first to minimize the number of merged responses for partition 0. Then the output responses in the second partition are merged independently from all other partitions to minimize the number of merged responses for Partition 1. This is repeated for all the partitions. The control bits for processing each merged partition are loaded from the tester into an on-chip RAM at the start of the test session. During the test session, the tester only needs to supply the index of the set of control bits that should be used for each MISR signature. This is illustrated in Fig. 7. For example, if for some scan vector, the response in the first partition was merged into the third group, then the tester supplies the index value of 3, and on-chip hardware converts that to a pointer into the on-chip RAM and fetches the appropriate control bits from the RAM to use for *X*-canceling of the first MISR signature. If the second partition was merged into the second group, then the tester supplies the index value of 2, and so forth. So the data stored on the tester include one copy of the control bits for each merged partial output response which are used to initialize the RAM, and then it stores one index value for each MISR signature which indicates where to fetch the control bits out of the RAM.

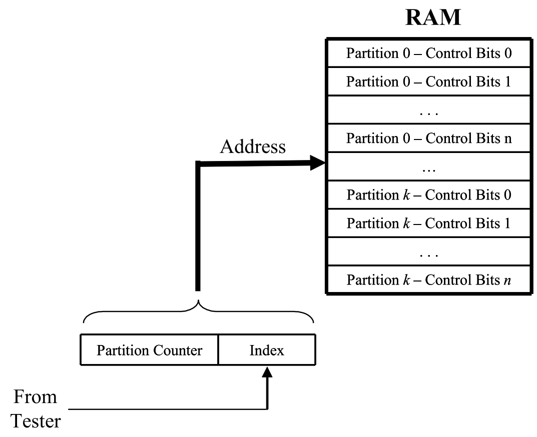
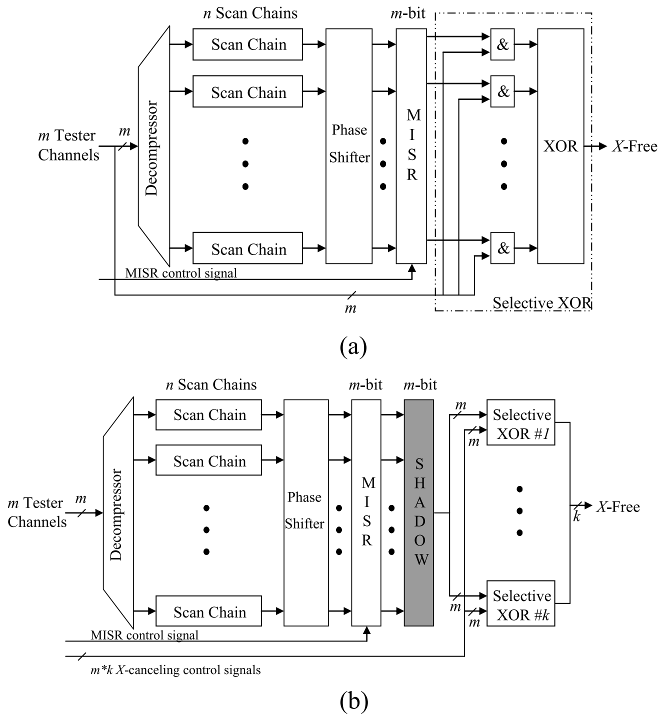

RAM
Partition 0 – Control Bits 0
Partition 0 – Control Bits 1
. . .
Partition 0 – Control Bits n
…
Partition k – Control Bits 0
Partition k – Control Bits 1
. . .
Partition k – Control Bits n
Address
Partition Counter
Index
From Tester

Fig. 7. Storing control bits in on-chip RAM.

Note that if there is a lot of variance in the number of X's in the scan responses, then the scan responses can also first be divided into groups where within each group, the same scan slice partitioning is used. For example, some group of scan responses might have four partitions (i.e., compact into four MISR signatures) while another group might have six partitions. Each group would then be treated as if it were independent, and the RAM would be reloaded before each group of test vectors is applied. This encoding scheme described in this section is very efficient, but it requires the presence of a sizeable on-chip RAM. For processors and other chips that have a cache or other large RAM present for functional purposes, it can be utilized to implement this scheme. If not, then the scheme described in the next section can be used which requires only a small scratch pad RAM be present on-chip.

B. Reducing On-Chip Stored Control Bits

On-chip RAM may not always be available. This section describes a way to perform superset X-canceling for partial output responses without the need of a large on-chip RAM. The idea is to have a small RAM which stores only one set of control bits for each partition. The test vectors are then ordered so that subsequent test vectors use the same control bits for most partitions. Thus, the tester needs to only incrementally update the control bits stored in the on-chip RAM. For example, if test vector i used the same control bits as test vector $i+1$ for all control bits except for the pth partition, then the tester would only need to store and reload the control bits for the pth partition when applying test vector $i+1$. The size of the RAM would be limited to the number of control bits required to process p MISR signatures for scan responses using p-partitions. The data stored on the tester would be limited to an index which points to the partition to be updated along with the control bits to store in that partition. The on-chip hardware would decode the index and then store the control bits in that location for the RAM. When tester has finished updating the necessary partitions, one bit is sent to indicate that the scan response for an entire test vector is ready to be compacted using the control bits stored in the on-chip RAM.



(a)

(b)

Fig. 8. X-canceling MISR architectures. (a) Time multiplexing X-canceling and (b) X-canceling with shadow register MISR architectures.

V. Hardware Architectures for Superset X-Canceling MISR

Two X-canceling MISR architectures are proposed in [26] and [27] and Fig. 8 illustrates them.

In time-multiplexing X-canceling MISR, X-canceled combinations are generated by halting scan shifting whenever the MISR is filled up with the maximum number of X's that it can tolerate. Tester channels are dedicated for scan vector loading and control data transfer. The scan vectors are ordered so that those which are using the same set of control bits for performing the X-canceling are applied consecutively. In this approach, automated test equipment (ATE) vector repeat can be used to reduce the volume of control bits stored in ATE. Vector repeat instruction is used to repeat the control values for all consecutive output responses that use the same control bits. This approach reduces the tester memory storage requirements since only one copy of the control bits needs to be stored in the vector memory, but does not reduce tester bandwidth requirements.

The other approach is an X-canceling with shadow register MISR architecture. This method is used if it is not desirable to halt scan shifting to process intermediate MISR signatures. In this architecture, test channels are assigned for scan vector loading and control data transfer, respectively. Hence, it may be suitable to implement on-chip hardware for superset X-canceling. In this method, a register on-chip drives the controls bits and the tester would simply load the register each time a new set of control bits is required. This approach would also reduce tester bandwidth requirements.

It should be noted that a vector repeat approach and an on-chip hardware approach can also be used in a time-multiplexing and a shadow register X-canceling MISR architectures.

TABLE I
RESULT COMPARISONS FOR PROPOSED SUPERSET *X*-CANCELING WITH RELAXED FAULT OBSERVATION CONSTRAINTS USING DIFFERENT MISR SIZES

Circuits	MISR Size	X-Canceling [18]	Superset X-Canceling [25]			Superset X-Canceling with Relaxed Fault Observation Constraints Control Bits (2 *D*'s in a Group)						
		Control Bits	Partition Size	1% *D*'s	0.5% *D*'s	Partition Size	1% *D*'s			0.5% *D*'s		
				Control Bits	Control Bits		Control Bits	Impv. over [18]	Impv. over [25]	Control Bits	Impv. over [18]	Impv. over [25]
CKT-A	128	5.4M	64	1.37M	1.03M	64	0.49M	11.02	2.80	0.43M	12.56	2.40
X-density = 0.5%	256	5.3M	128	1.78M	1.08M	128	0.46M	11.52	3.87	0.31M	17.10	3.48
	512	5.2M	256	3.20M	1.79M	256	0.77M	6.75	4.16	0.54M	9.63	3.31
CKT-B	128	22.1M	64	3.31M	2.47M	64	1.61M	13.73	2.06	1.41M	15.67	1.75
X-density = 2.75%	256	21.4M	128	3.96M	2.72M	128	1.49M	14.36	2.66	1.14M	18.77	2.39
	512	21.1M	256	5.68M	3.58M	256	1.70M	12.41	3.35	1.19M	17.73	3.01
CKT-C	128	51.6M	64	21.80M	18.03M	64	14.69M	3.51	1.49	14.09M	3.66	1.28
X-density = 2.38%	256	50.2M	128	24.26M	17.54M	128	12.12M	4.14	2.01	11.20M	4.48	1.57
	512	49.5M	256	32.67M	20.71M	256	11.14M	4.44	2.94	9.29M	5.33	2.23

VI. EXPERIMENTAL RESULTS

In this section, experimental results are presented for three industrial circuits. Table I shows the results different sizes of MISRs, namely 128, 256, and 512, with different percentages of *D*'s. The results shown here are for using seven *X*-canceled combinations which provide 99.2% error coverage, i.e., 99.2% of all non-*X* values are observed. Comparisons of *X*-canceling with *X*-compact and other methods can be found in [18], [26], and [27]. The first column shows three industrial circuits with different *X*-density in the output stream. Different MISR sizes used in the experiment are shown in the second column. The third column shows the number of control bits required for *X*-canceling method used in [18]. For superset *X*-canceling, the output response for each test vector was partitioned and one signature was generated for each partition. While the MISR size would need to be selected at design time, the number of partitions could be selected after the output response is already known allowing the opportunity to optimize it. The fourth column shows the number of partitions and the number of control bits in [25]. The fifth column presents the number of partitions, the number of control bits, and the compression ratio over [18] and [25]. The results for superset *X*-canceling depend on how many *D*'s are in the output response data. The actual locations of the *D*'s for these test sets were not available, so the experiments were done by randomly injecting *D*'s. For the proposed superset *X*-canceling with relaxed fault detection constraints, we assume that the effects of each fault propagate to two scan cells and we treat those 2 *D*'s as being in the same group as discussed in Section III-B. The results in Table I show very significant improvements in the amount of compression can be obtained (up to an order of magnitude improvement over [18] and twice or three times the compression ratio achieved in [25]). As can be seen from the results, the proposed method is able to significantly reduce the number of control bits in most cases. The fewer the *D*'s there are, the better the results are since there are less sources of conflict when output responses are merged together considering the proposed approach of keeping one *D*'s alive per fault.

In the proposed method, the compression ratio heavily depends on two factors: 1) the percentage of *D*'s and 2) the multiplicity of *D*'s per fault. Table II shows the compression ratio with respect to these two factors. The first column shows the circuit, and the MISR size is given in the second column. The third and fourth columns show the compression improvement ratio over [18] and [25], respectively, with different *D* percentages and different multiplicity of *D*'s per fault. Because the number of scan cells that the effects of a fault propagate to can vary, we generated results for different multiplicities of *D*'s per fault ranging from 2 to 5. As shown in Table II, compression ratio comparisons with [18] shows a decreasing improvement ratio trend as the percentage of *D*'s increases, whereas an increasing improvement ratio is seen compared to [25]. The reason for this is that in the comparison with [18], the absolute number of control bits for the proposed method increases with higher *D* percentages. However, in the comparison with [25], there are more possibilities in merging responses with higher *D* percentages using the proposed method since we relax fault observation constraints while maintaining the same fault coverage. This gives more merged superset responses, hence, the relative compression ratio over [25] increases with higher *D* percentages.

Note that all of these results could be further improved if this methodology was incorporated into the ATPG procedure along the lines of what was described in [5] to minimize

TABLE II
COMPRESSION IMPROVEMENT RATIO OF THE PROPOSED METHOD OVER [18] AND [25] WITH RESPECT TO TWO FACTORS

Circuits	MISR Size	Improvement Ratio over [18]						Improvement Ratio over [25]					
		Num. of *D*'s	Percentage of *D*'s					Num. of *D*'s	Percentage of *D*'s				
			0.1%	0.5%	1%	2%	3%		0.1%	0.5%	1%	2%	3%
CKT-A	128	2 *D*'s	15.43	12.56	11.02	8.86	7.50	2 *D*'s	2.20	2.40	2.80	3.21	3.61
		3 *D*'s	15.89	14.60	14.22	11.74	10.19	3 *D*'s	2.26	2.78	3.61	4.26	4.91
		4 *D*'s	16.88	15.89	15.43	14.22	13.50	4 *D*'s	2.41	3.03	3.91	5.16	6.50
		5 *D*'s	16.37	15.89	15.89	15.43	14.22	5 *D*'s	2.33	3.03	4.03	5.60	6.84
	256	2 *D*'s	33.13	17.10	11.52	7.69	5.58	2 *D*'s	3.25	3.48	3.87	4.32	4.41
		3 *D*'s	44.17	27.90	20.39	13.25	10.60	3 *D*'s	4.33	5.68	6.85	7.45	8.38
		4 *D*'s	66.25	37.86	29.45	22.09	18.93	4 *D*'s	6.50	7.71	9.89	12.42	14.96
		5 *D*'s	66.25	48.19	40.77	35.34	33.13	5 *D*'s	6.50	9.82	13.69	19.87	26.19
	512	2 *D*'s	21.67	9.63	6.75	4.10	3.03	2 *D*'s	2.46	3.31	4.16	4.29	4.41
		3 *D*'s	40.00	17.94	13.00	8.39	6.05	3 *D*'s	4.54	6.17	8.00	8.79	8.83
		4 *D*'s	47.28	28.89	19.26	14.06	10.62	4 *D*'s	5.36	9.94	11.85	14.73	15.49
		5 *D*'s	57.78	43.34	30.59	21.67	19.26	5 *D*'s	6.56	14.92	18.82	22.71	28.11
CKT-B	128	2 *D*'s	18.58	15.67	13.73	11.17	9.57	2 *D*'s	1.30	1.75	2.06	2.35	2.49
		3 *D*'s	20.10	19.06	17.54	14.84	13.24	3 *D*'s	1.40	2.13	2.63	3.13	3.44
		4 *D*'s	21.05	20.85	20.10	18.73	16.88	4 *D*'s	1.47	2.34	3.01	3.95	4.38
		5 *D*'s	21.46	22.56	22.33	21.25	20.28	5 *D*'s	1.50	2.53	3.35	4.48	5.26
	256	2 *D*'s	26.10	18.77	14.36	10.60	8.53	2 *D*'s	1.64	2.39	2.66	2.98	3.17
		3 *D*'s	30.58	25.48	20.99	16.59	13.99	3 *D*'s	1.92	3.47	3.89	4.66	5.19
		4 *D*'s	34.52	31.48	27.09	23.27	20.39	4 *D*'s	2.17	4.00	5.02	6.53	7.57
		5 *D*'s	36.28	37.55	33.97	31.02	27.44	5 *D*'s	2.28	4.78	6.29	8.70	10.18
	512	2 *D*'s	30.15	17.73	12.41	8.18	6.25	2 *D*'s	2.00	3.01	3.35	3.54	3.67
		3 *D*'s	42.20	28.14	20.49	14.07	12.13	3 *D*'s	2.8	4.77	5.52	6.09	7.13
		4 *D*'s	54.11	39.08	32.47	24.26	19.36	4 *D*'s	3.59	6.63	8.74	10.50	11.37
		5 *D*'s	63.94	57.03	43.96	34.60	31.50	5 *D*'s	4.25	9.68	11.84	14.97	18.5
CKT-C	128	2 *D*'s	3.78	3.66	3.51	3.27	3.03	2 *D*'s	1.09	1.28	1.49	1.87	2.16
		3 *D*'s	3.80	3.77	3.71	3.61	3.51	3 *D*'s	1.12	1.32	1.57	2.07	2.49
		4 *D*'s	3.83	3.82	3.80	3.76	3.70	4 *D*'s	1.13	1.34	1.61	2.15	2.63
		5 *D*'s	3.86	3.84	3.84	3.81	3.80	5 *D*'s	1.14	1.34	1.62	2.18	2.70
	256	2 *D*'s	4.75	4.48	4.14	3.60	3.08	2 *D*'s	1.20	1.57	2.01	2.65	2.94
		3 *D*'s	4.81	4.72	4.63	4.34	4.05	3 *D*'s	1.21	1.65	2.24	3.20	3.86
		4 *D*'s	4.85	4.80	4.76	4.67	4.56	4 *D*'s	1.22	1.68	2.30	3.44	4.35
		5 *D*'s	4.86	4.83	4.82	4.77	4.75	5 *D*'s	1.23	1.69	2.33	3.51	4.53
	512	2 *D*'s	5.98	5.33	4.44	3.29	2.62	2 *D*'s	1.39	2.23	2.94	3.46	3.53
		3 *D*'s	5.97	5.86	5.52	4.80	4.20	3 *D*'s	1.38	2.46	3.64	5.03	5.66
		4 *D*'s	5.98	5.94	5.92	5.68	5.39	4 *D*'s	1.39	2.49	3.91	5.95	7.27
		5 *D*'s	5.99	5.96	6.01	5.90	5.87	5 *D*'s	1.39	2.50	3.97	6.19	7.92

conflicts from *D*'s and enable larger merged output responses. In addition, if the *D*'s show intercorrelation, the results could be further improved.

The required hardware overhead to implement the proposed method without partitioning is mainly determined by XOR gates for a phase shifter and a selective XOR block as shown in Fig. 8 [26], [27]. For n scan chains with f fan-out for a phase shifter, m-bit MISR, and l tester channels, the hardware overhead can be expresses as follows.

1) *Time Multiplexing X-Canceling MISR Architecture (Hardware Overhead):* $[n*f+(m-1)]$ two input XOR gates and one m-bit MISR.
2) *X-Canceling With Shadow Register MISR Architecture (Hardware Overhead):* $[n*f+l*(m-1)]$ two input XOR gates and two m-bit MISR.

In addition to the hardware above, an on-chip RAM, $\lceil \log_2 k \rceil$ and $\lceil \log_2 i \rceil$ bit registers are needed for k partitions and i indexes when response partitioning is used. It should be noted that a cache or on-chip RAM existing for functional purposes can be used in test mode to store control bits. This helps to reduce the area overhead by the on-chip memory requirement. Section IV-B describes a way to perform the proposed method when an on-chip memory is not available or a small RAM is allowed.

TABLE III
PROPOSED METHOD RUN TIME MEASUREMENTS FOR TABLE I

Circuits	MISR Size	Partition Size	CPU Time (sec)	
			1% *D*'s	0.5% *D*'s
CKT-A	128	64	906.44	618.60
	256	128	1126.94	759.97
	512	256	1044.28	847.20
CKT-B	128	64	117.76	111.44
	256	128	81.39	71.92
	512	256	63.81	53.05
CKT-C	128	64	447.41	479.20
	256	128	259.56	303.36
	512	256	160.98	173.22

Table III shows the amount of run time used to determine the control bits for the proposed method. The proposed method is implemented in C++ and the run time is measured on a 2.0 GHz Xeon 6 core Linux machine with 1TB memory in Texas Advanced Computing Center. Control bits are found with 2 *D*'s in a group as in Table I. As can be seen with industrial circuits, the run time overhead for control bits generation is less than other automated flows such as ATPG and fault simulation. Hence, the proposed method can be efficiently integrated with a design flow for an *X*-canceling architecture.

VII. Conclusion

The proposed superset *X*-canceling method exploits two facts: 1) the scan cells that capture *X*'s are highly correlated and 2) the multiplicity of scan cells that the fault effects for each fault propagate to. A merging algorithm for superset *X*-canceling is proposed and its hardware implementations are described which can be employed based on the particular design situation.

In-depth analysis showed that the proposed method achieves significant improvements over the existing techniques [25] and [27]. Various cases with *D*'s in industrial designs are considered to show the effectiveness of the proposed method. With continued increase in design size and complexity, the proposed method provides an avenue for scaling up compression to keep with expected increases in design size and complexity.

Acknowledgment

This work was supported by the Samsung Research Fund, Sungkyunkwan University, 2015.

References

[1] C. Barnhart *et al.*, "OPMISR: The foundation for compressed ATPG vectors," in *Proc. Int. Test Conf.*, Baltimore, MD, USA, 2001, pp. 748–757.

[2] 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," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, 2005, pp. 80–87.

[3] V. Chickermane, B. Foutz, and B. Keller, "Channel masking synthesis for efficient on-chip test compression," in *Proc. Int. Test Conf.*, Charlotte, NC, USA, 2004, pp. 452–461.

[4] C. G. Cullen, *Linear Algebra With Applications*. Reading, MA, USA: Addison-Wesley, 1997.

[5] D. Czysz, G. Mrugalski, N. Mukherjee, J. Rajski, and J. Tyszer, "On compaction utilizing inter and intra-correlations of unknown states," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.,* vol. 29, no. 1, pp. 117–126, Jan. 2010.

[6] E. Gizdarski, "Constructing augmented time compactors," in *Proc. Eur. Test Symp.*, Prague, Czech Republic, 2010, pp. 151–156.

[7] I. Hamzaoglu and J. H. Patel, "Reducing test application time for full scan embedded cores," in *Proc. 29th Annu. Int. Symp. Fault Toler. Comp.*, Madison, WI, USA, 1999, pp. 260–267.

[8] A. Leininger, M. Fischer, M. Braun, M. Richter, and M. Goessel, "Using timing flexibility of automatic test equipment to complement X-tolerant test compression techniques," in *Proc. Int. Test Conf.*, Santa Clara, CA, USA, 2007, Art. ID 6.3.

[9] S. Mitra and K. S. Kim, "X-compact: An efficient response compaction scheme," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.,* vol. 23, no. 3, pp. 421–432, Mar. 2004.

[10] S. Mitra, S. S. Lumetta, and M. Mitzenmacher, "X-tolerant signature analysis," in *Proc. Int. Test Conf.*, Charlotte, NC, USA, 2004, pp. 432–441.

[11] I. Pomeranz, S. Kundu, and S. M. Reddy, "On output response compression in the presence of unknown output values," in *Proc. Design Autom. Conf.*, New Orleans, LA, USA, 2002, pp. 255–258.

[12] J. Rajiski, J. Tyszer, C. Wang, and S. M. Reddy, "Finite memory test response compactors for embedded test applications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 4, pp. 622–634, Apr. 2005.

[13] W. Rajski and J. Rajski, "Modular compactor of test responses," in *Proc. VLSI Test Symp.*, Berkeley, CA, USA, 2006, pp. 242–251.

[14] J. Rajski *et al.*, "X-press compactor for 1000x reduction of test data," in *Proc. Int. Test Conf.*, Santa Clara, CA, USA, 2006, pp. 1–10, Art. ID 18.1.

[15] J. H. Patel, S. S. Lumetta, and S. M. Reddy, "Application of Saluja-Karpovsky compactors to output responses with many unknowns," in *Proc. VLSI Test Symp.*, Napa, CA, USA, 2003, pp. 107–112.

[16] M. Sharma and W.-T. Cheng, "X-filter: Filtering unknowns from compacted test responses," in *Proc. Int. Test Conf.*, Austin, TX, USA, 2005, Art. ID 42.1.

[17] Y. Tang *et al.*, "X-masking during logic BIST and its impact on defect coverage," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 2, pp. 193–202, Feb. 2006.

[18] N. A. Touba, "X-canceling MISR—An X-tolerant methodology for compacting output responses with unknowns using a MISR," in *Proc. Int. Test Conf.*, Santa Clara, CA, USA, 2007, pp. 1–10, Art. ID 6.2.

[19] L. T. Wang, C.-W. Wu, and X. Wen, *VLSI Test Principles and Architectures.* Burlington, ON, Canada: Morgan Kaufmann, 2006.

[20] P. Wohl, J. A. Waicukauski, and T. W. Williams, "Design of compactors for signature-analyzers in built-in self-test," in *Proc. Int. Test Conf.*, Baltimore, MD, USA, 2001, pp. 54–63.

[21] P. Wohl, J. A. Waicukauski, S. Patel, and M. B. Amin, "X-tolerant compression and application of scan-ATPG patterns in a BIST architecture," in *Proc. Int. Test Conf.*, Washington, DC, USA, 2003, pp. 727–736.

[22] P. Wohl, J. A. Waicukauski, and S. Patel, "Scalable selector architecture for X-tolerant deterministic BIST," in *Proc. Design Autom. Conf.*, San Diego, CA, USA, 2004, pp. 934–939.

[23] P. Wohl, J. A. Waicukauski, F. Neuveux, and E. Gizdarski, "Fully X-tolerant, very high scan compression," in *Proc. Design Autom. Conf.*, Anaheim, CA, USA, 2010, pp. 362–367.

[24] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[25] J. Chung and N. A. Touba, "Exploiting X-correlation in output compression via superset X-canceling," in *Proc. VLSI Test Symp.*, Maui, HI, USA, 2012, pp. 182–187.

[26] J.-S. Yang, N. A. Touba, S.-Y. Yang, and T. M. Mak, "Industrial case study for X-canceling MISR," in *Proc. IEEE Int. Test Conf.*, Austin, TX, USA, 2009, pp. 1–10.

[27] J.-S. Yang and N. A. Touba, "X-canceling MISR architectures for output response compaction with unknown values," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.,* vol. 31, no. 9, pp. 1417–1427, Aug. 2012.

Joon-Sung Yang (S'05–M'09) received the B.S. degree from Yonsei University, Seoul, Korea, in 2003, and the M.S. and Ph.D. degrees from the University of Texas at Austin, Austin, TX, USA, in 2007 and 2009, respectively, all in electrical and computer engineering.

He was with Intel Corporation for four years. He is currently an Assistant Professor with SungKyunKwan University, Seoul, South Korea. His current research interests include VLSI testing, silicon debug and nanometer scale test, and design methodologies.

Dr. Yang was a recipient of the Korea Science and Engineering Foundation Scholarship in 2005 and the Best Paper Award at 2008 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, and was nominated for the Best Paper Award at 2013 IEEE VLSI Test Symposium.

Jinsuk Chung received the B.S. degree from Yonsei University, Seoul, Korea, in 2008. He is currently pursuing the Ph.D. degree in computer engineering with the University of Texas at Austin, Austin, TX, USA.

His current research interests include resilience schemes for exascale systems, especially software schemes.

Mr. Chung received an MCD fellowship from the University of Texas at Austin, in 2009. He was nominated for the Best Paper Award at SC 2012 Conference.

Nur A. Touba (SM'05–F'09) received the B.S. degree from the University of Minnesota, Minneapolis, MN, USA, in 1990, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, USA, in 1991 and 1996, respectively, all in electrical engineering.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA.

Dr. Touba was a recipient of the National Science Foundation Early Faulty CAREER Award in 1997, the Best Paper Award at the 2001 VLSI Test Symposium, and the 2008 Defect and Fault Tolerance Symposium. He has served as Program Chair for the 2008 International Test Conference and General Chair for the 2007 Defect and Fault Tolerance Symposium. He currently serves on the program committee for the Design Automation and Test in Europe Conference, International On-Line Test Symposium, European Test Symposium, Asian Test Symposium, and Defect and Fault Tolerance Symposium.