Reducing Control Bit Overhead for X-Masking/X-Canceling Hybrid Architecture via Pattern Partitioning

Jin-Hyun Kang
Semiconductor Systems Department
Sungkyunkwan University
Suwon, Korea, 16419
kangjin13@skku.edu

Nur A. Touba
ECE Department
University of Texas at Austin
TX, USA, 78712
touba@utexas.edu

Joon-Sung Yang
Semiconductor Systems Department
Sungkyunkwan University
Suwon, Korea, 16419
js.yang@skku.edu

ABSTRACT

An *X*-masking scheme prevents unknown (*X*) values from shifting into an output response compactor, whereas an *X*-canceling MISR methodology allows *X*'s to enter the compactor, but then cancels them out through selective XORing. However, both approaches require significantly high volume of the control bits to remove *X* values to generate *X*-free output signatures. This paper proposes a method to reduce the control bit overhead by combining *X*-masking and *X*-canceling methodologies and exploiting the fact that unknown values tend to have high correlation in the scan cells. In this paper, correlation is considered across whole patterns in order to enhance reuse of control bits. The proposed hybrid method of *X*-canceling and *X*-masking reduces test time without losing fault coverage. The experimental results show that the proposed method significantly reduces control bits and test time compared to a conventional *X*-canceling MISR methodology.

Categories and Subject Descriptors

B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance

General Terms

Algorithms, Design, Reliability, Experimentation

Keywords

Control Bits, Test Pattern Partitioning, Hybrid of *X*-masking and *X*-canceling MISR.

1. INTRODUCTION

As design size and complexity in integrated circuits grows, test volume continues to increase considerably. As a result, test time and test cost grow rapidly. Test stimuli compression and test output response compaction are used to alleviate these problems.

DAC '16, June 05-09, 2016, Austin, TX, USA
 ISBN 978-1-4503-4236-0/16/06$15.00
DOI: http://dx.doi.org/10.1145/2897937.2898078

In test output response compaction, unknown values, *X* values cause a significant problem. *X* values are introduced by sources such as uninitialized memory elements, bus contention, floating tri-states, etc., and they corrupt the final signature. *X* values directly degrade fault coverage [1], hence, *X* values should be eliminated for a proper test process [2]. *X*-masking removes *X* values at the input to the output response compactor [5-7]. This requires control signals from test channels for each clock cycle. Since *X*-masking needs mask control signals for each clock cycle, the control signal volume becomes an overhead. An *X*-canceling MISR (multiple input signature register) is another *X*-handling scheme that cancels out *X*'s from MISR signatures [11, 12]. In this method, Gaussian elimination is used to identify linearly dependent combinations of MISR signature bits and they are XORed together to cancel out all *X* values. The volume of control bits for the canceling process is dependent on the *X*-density (percentage of output response bits that are *X*'s).

In [13], the terms "intra (or spatial)-correlation" and "inter-correlation" of *X*'s are introduced to refer to correlations of *X*'s within a single output response and across multiple output responses, respectively. Superset *X*-canceling [17, 18] proposes a method to reduce control bits for an *X*-canceling MISR considering the correlation. It finds a general superset of the control bits which can be reused across multiple output responses. The control bits are considerably reduced by this method. This approach tries to find other output responses which can be merged with the *selected output response*. It allows merging *X*'s with non-*X* values among output responses. However, because the superset *X*-canceling merges output responses with respect to the *selected output response*, the effectiveness of the merge process depends on how much the other output responses are spatially correlated with the *selected output response*. The superset *X*-canceling method tries to find inter-correlation with a specific spatial correlation. It may limit the number of output responses that can be merged. This method also requires iterative fault simulations to check the fault coverage loss by the disappearance of non-*X* values.

In this paper, we propose a control-bit test data volume reduction approach for a hybrid *X*-handling using both *X*-masking and *X*-canceling MISR architectures. In this work, because *X*-masking removes *X* values shifting into the MISR, the control bits for the *X*-canceling MISR are significantly reduced. Because *X*-masking requires a significant volume of control bits, this paper tries to reduce the control bits required for *X*-masking operation. To reduce the test data, we exploit the inter-correlation of *X*-locations in the entire output responses. Unlike [17, 18], the proposed method tries to reduce the number of control bits for *X*-masking considering an *X*-canceling MISR. Using the inter-correlation, the

proposed method partitions the set of test patterns that each mask is used for. The partitioning allows us to reduce the control bit test data volume by sharing control bits for multiple test patterns without any fault coverage loss. Because the non-*X* values are not removed or merged, the proposed method guarantees no fault coverage loss without the need for fault simulations.

The paper is organized as follows. In section 2, the overview of *X*-masking and *X*-canceling MISR is given. Section 3 explains the correlation of *X*'s, and Section 4 shows a proposed procedure of partitioning output responses to reduce control bits. Experimental results and conclusion are provided in Section 5 and 6 respectively.

2. REVIEW OF *X*-MASKING AND *X*-CANCELING MISR METHODS

In this section, we briefly review *X*-masking and *X*-canceling MISR and show how the control data can be calculated, and explain why reducing the number of *X* values helps to reduce the control data volume.

Figure 1 shows an *X*-masking architecture. This method places AND gate (or OR gate) at the input of the output response compactor (MISR). Control bits are transferred from ATE every clock cycle to remove *X* values before they are shifted into MISR. Hence, this requires a number of control bits. The total number of control bits is the same as the number of total scan cells. For example, if there are *n* scan cells in one scan chain, *m* scan chains and *k* test patterns, *n* x *m* x *k* control bits are required. Hence, the total number of control bits for *X*-masking can be expressed as follows:

*Total Number of Control Bits = Longest Scan Chain Length * Number of Scan Chains * Number of Test Patterns*

Figure 2 shows the final state of the MISR through symbolic simulation of each scan cell value with 14 deterministic values and 4 *X* values. Each MISR bit is expressed by a linear equation of the scan cell symbols, O_i and X_j where O_i indicates a non-*X* value and X_j denotes an *X* value. The linear combinations of each MISR bit can be expressed as a form of a matrix as shown in Figure 3. Each row maps to each MISR bit and the matrix element is *1* if the MISR bit corresponding to the row depends on the *X* corresponding to the column. If the number of *X*'s is less than the MISR size, some row combinations are linearly dependent and this can be found by Gaussian elimination. Since there are 4 *X*'s in a 6 bit MISR, 2 *X*-free rows can be found as shown in Figure 3. Two *X*-free rows can be generated by XORing M_1, M_3, M_5 and M_1, M_4 bits. The following shows the *X*-canceled signatures:

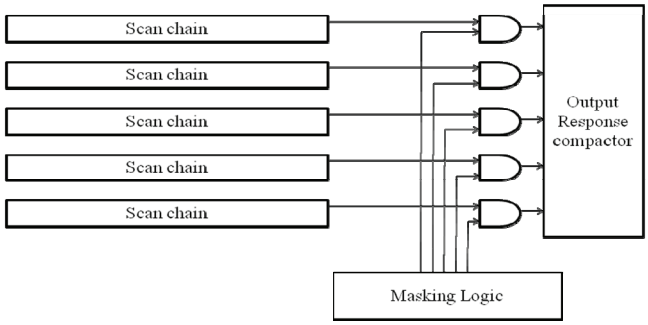

Figure 1. *X*-Masking Architecture

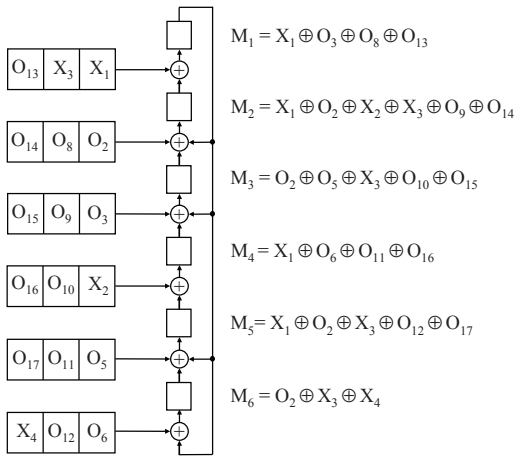

Figure 2. Example of Symbolic Simulation of MISR

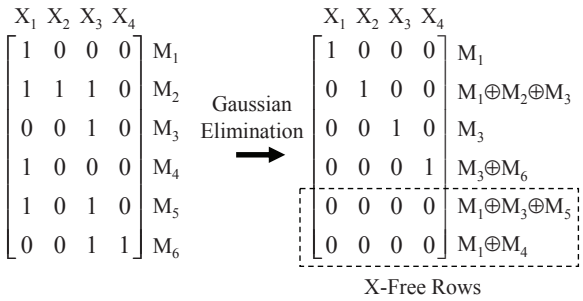

Figure 3. Gaussian Elimination

$$M_1 \oplus M_3 \oplus M_5 = O_3 \oplus O_5 \oplus O_8 \oplus O_{10} \oplus O_{12} \oplus O_{13} \oplus O_{15} \oplus O_{17}$$
$$M_1 \oplus M_4 = O_3 \oplus O_6 \oplus O_8 \oplus O_{11} \oplus O_{13} \oplus O_{16}$$

An *X*-canceling MISR requires control bits to perform selective XORing of signature bits as in the example in Figure 3. Since two *X*-free signatures are generated, it needs two cycles and each cycle requires 6 bits of control data. A total of 12 bits of control data is needed. The amount of control data for an *X*-canceling MISR can be expressed as a general form. If the size of MISR is *m* and the number of *X*-free combination is *q*, the MISR stops *(Total Number of X's) / (m – q)* times to extract *X*-free signatures. Each time, *m*q* bits of control data needs to be transferred for Gaussian elimination. Hence, the total number of control bits can be expressed as:

$$Total\ Number\ of\ Control\ Bits = mq * \frac{Total\ Number\ of\ X's}{m-q}$$

The total number of control bits required for an *X*-canceling MISR depends on the total number of *X*'s shifted into the MISR. If both the *X*-masking and *X*-canceling MISR methods are combined and *X*-masking removes a number of *X*'s and leaves only a few *X* values to be shifted to the *X*-canceling MISR, we can significantly reduce the amount of control bits required for *X*-canceling MISR. This also means that the MISR is halted less times, therefore, the hybrid approach of *X*-masking and *X*-canceling MISR helps to reduce the test time as well. However, a problem still remains since the total number of control bits for the hybrid method is (*Total Number of Control Bits for X-masking*) + (*Total Number of Control Bits for X-canceling MISR*) and *X*-

masking imposes a significant control bit overhead for the hybrid method. Hence, the key is to reduce the control bits for *X*-masking.

3. *X*-VALUE CORRELATION ANALYSIS

The approach presented in this paper tries to reduce the control data volume of *X*-masking by considering *X*-correlation. [13] addresses "*X states are not randomly distributed in test responses. On the contrary, they have identical or similar patterns occurring in contiguous and adjacent areas of scan chains for many test patterns*". These are referred as intra and inter-correlation of *X* locations. In this paper, we focus on the inter-correlation. Since the intra-correlation finds an adjacent region of *X* value sources in scan chains within one pattern, the effect of reducing *X*'s may be constrained compared to an inter-correlation. The inter-correlation is found across multiple test patterns and thus it has a potential to remove a higher number of *X*'s from multiple patterns.

To analyze the *X*-value correlation, we use an industrial circuit example as an illustration. It has 36,075 scan cells and only 3,903 scan cells capture *X*'s. 90% of *X*'s are captured in 4.9% of the scan cells. This shows a strong *X* value correlation. For the inter-correlation analysis, we count the number of *X*'s captured in each scan cell for 3000 test patterns. One of scan cells captures 406 *X*'s and there are 176 more scan cells which have the same number of *X*'s. Totally, 177 scan cells have the same number of *X*'s, 406, in 3000 patterns. To examine the inter-correlation in this example, we can check the patterns whether the same 406 test patterns produce *X*'s in the same 177 scan cells. The analysis results show that 172 scan cells out of 177 have the 406 *X*'s by the same 406 test patterns. Even though there are 5 scan cells which are not sharing the same test patterns, the same 405 test patterns out of 406 generate 405 *X*'s and one other test pattern produces one *X* for the 5 scan cells. This means that the same 405 test patterns out of 406 patterns produce *X* values for 177 scan cells and one pattern differentiate 172 scan cells and 5 scan cells (one pattern is shared by 172 scan cells and the other pattern is shared by 5 scan cells). However, this analysis shows a significant inter-correlation of *X* values.

There could be a case where the inter-correlation of *X* values is not high. Let's assume that there are 100 scan cells capturing 200 *X*'s with 3000 test patterns. Via *X*-value correlation analysis, only 10 scan cells capture *X*'s for the entire 200 test patterns. This does not seem to be helpful. However, if 10 scan cells are properly handled, 10 * 200 *X*'s can be removed from the scan responses.

The proposed method exploits the *X* value inter-correlation to reduce control bits. In *X*-masking, if *X* values occur at the same location, *scan_cell_A*, across entire test patterns, one control pattern can remove a number of *X* values. The single control pattern, which masks only *scan_cell_A* and leaves the other scan cell values to be shifted into the MISR, can get rid of *Number of Test Patterns X*'s. From the *X*-canceling MISR point of view, this is a significant help to reduce the amount of control bits for *X*-canceling MISR.

4. DETAILS OF PROPOSED METHOD

This section describes the steps to reduce the control bit overhead for the hybrid approach of *X*-masking and *X*-canceling MISR.

Figure 4 shows an example of some simple logic. There are 5 scan chains (SC1 ~ SC5) and each chain has 3 scan cells. The circuit responses are captured in the scan chain for 8 test patterns (P1 ~ P8). By performing the *X*-value correlation analysis for 8

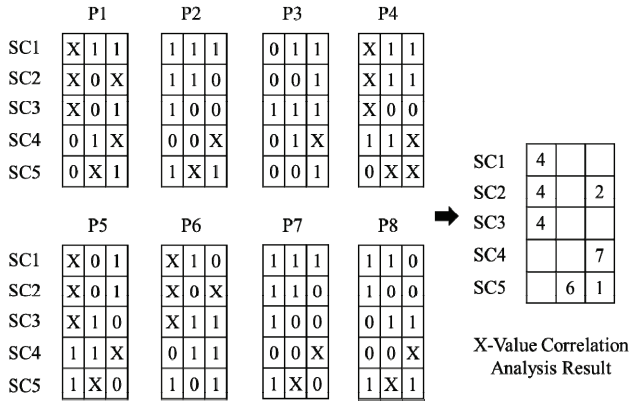

Figure 4. Example of *X*-Value Correlation Analysis with 8 Test Patterns, 5 Scan Chains with 3 Scan Cells per Each Chain

test patterns, the inter-correlation for *X*'s can be found. There are 3 scan cells capturing 4 *X*'s (the first scan cell in SC1, the first scan cell in SC2, the first scan cell in SC3), 1 scan cell with 1 *X* (the third scan cell in SC5), 1 scan cell with 2 *X*'s (the third scan cell in SC2), 1 scan cell with 6 *X*'s (the second scan cell in SC5) and 1 scan cell with 7 *X*'s (the third scan cell in SC4). In this example, the most number of *X*'s captured in one scan cell is 7 and the largest number of scan cells having the same number of *X*'s is 3 (3 scan cells capturing 4 *X*'s). From *X*-value correlation analysis results, removing 7 *X*'s by *X*-masking looks to be a better choice than choosing one of three scan cells to get rid of 4 *X*'s. However, as explained in Section 3, there is a high chance of inter-correlation when there are many scan cells with the same number of *X*'s. This means that if we trace test patterns giving the largest number of scan cells with the same number *X*'s (the scan cell with 4 *X*'s in Figure 4), owing to the inter-correlation, there will be a chance that they are handled together for control bit reduction. Based on this assumption and the *X*-value inter-correlation property, we partition the test patterns using the *X*-value correlation analysis result.

As the first step of the partitioning, one scan cell is chosen among the largest number of scan cells having the same number of *X*'s in the *X*-value correlation analysis result. In Figure 5, since the largest number of scan cells capturing the same number of *X*'s (i.e., there are 3 scan capturing 4 *X*'s), then one of scan cells is chosen among the first scan cell in SC1, the first scan cell in SC2, or the first scan cell in SC3. Once the scan cell is chosen, the first partition can be performed with respect to the selected scan cell. In this example, we randomly select one of 3 scan cells and the first scan cell in SC1 is picked. We perform the first partitioning with respect to the selected scan cell. The selected first scan cell in SC1 captures *X* from P1, P4, P5 and P6. The other patterns (P2, P3, P7 and P8) generate a non-*X* value for the selected first scan cell in SC1. If we partition the entire set of test patterns with respect to the first scan cell in SC1, there are two partitions – one partition (*Partition 1*) has *X*'s in the first scan cell in SC1 and the other partition (*Partition 2*) has non-*X* values in the first scan cell in SC1. As shown in Figure 5, the partitioning divides the *X*-value correlation analysis result in (a) into *Partition 1* and *Partition 2* in (b). This example shows that the 3 scan cells capturing 4 *X*'s (the first scan cell in SC1, the first scan cell in SC2, the first scan cell in SC3) in the *X*-value correlation analysis result still belong to the same partition. They have the inter-correlation across multiple test patterns.

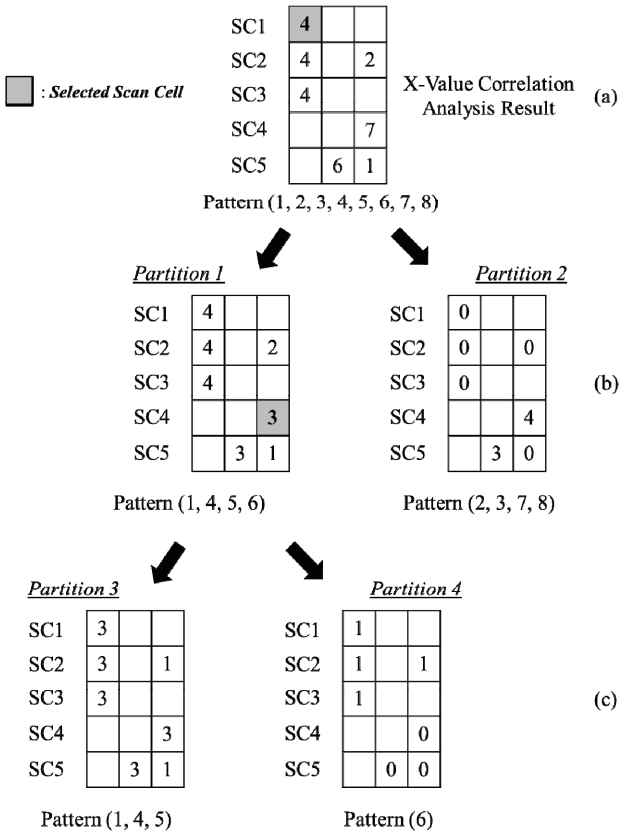

Figure 5. Test Pattern Partitioning Process

(a) Before Partitioning : *X*-Value Correlation Analysis Result

(b) 1st Partitioning : Remaining Partitions : Partition 1 & 2

(c) 2nd Partitioning : Remaining Partitions : Partition 2, 3 & 4

In the same fashion, the partitioning process is iteratively performed with the generated partitions. Now, there are two partitions, *Partition 1* and *Partition 2* in Figure 5 (b). In *Partition 1*, since the scan cell with 4 *X*'s is already chosen by the first partitioning process, we need to find the second largest number of scan cells having the same number of *X*'s. There are 3 *X*'s found from 2 scan cells (the third scan cell in SC4 and the second scan cell in SC5 of *Partition 1*). We choose the third scan cell in SC4 and the second round of partitioning is performed with respect to the selected third scan cell in SC4. The binary partitioning makes new partitions, *Partition 3* and *Partition 4*, which are shown in Figure 5 (c). *Partition 3* is composed of test pattern 1, 4, 5 and *Partition 4* includes test pattern 6.

In *Partition 2*, there are only two scan cells where one captures 3 *X*'s and the other captures 4 *X*'s. Since there is no largest number of scan cells capturing the same number of *X*'s, the partitioning is not carried out in *Partition 2*. Even for *Partition 3* and *Partition 4*, there is no largest number of scan cells capturing the same number of *X*'s. The iterative partition process stops.

The example with 8 Patterns, 5 scan chains with 3 scan cells per each chain is finally divided into 3 partitions, *Partition 2, Partition 3 and Partition 4*. *Partition 2* shares 4 test patterns (pattern 2, 3, 7, 8) and 3 test patterns (pattern 1, 4, 5) belong to *Partition 3*. *Partition 4* has one pattern (pattern 6). Because the test patterns are shared within partitions, the control bits for *X*-masking operation can also be shared. Figure 6 shows how control bits are generated for each partition. In *Partition 2*, there is only one scan cell, the third scan cell in SC4, has the same number of *X*'s with the number of patterns in the partition (i.e., 4 *X*'s found at the third scan cell in SC4 with 4 test patterns). Hence, by setting 1 at the corresponding scan cell location in the control bits, *X*-masking logic can remove 4 *X*'s. It should be noted that the proposed method does not mask the second bit in SC5 to remove 3 *X*'s. If the second bit in SC5 is masked, there will be one non-*X* value being masked (i.e., non-*X* value of the second bit in SC5 from P3 in Figure 4 will be lost). It loses observability and potentially impacts the fault coverage. To avoid any concerns about fault coverage loss, the proposed method does not mask any scan cells if it loses non-*X* values. In the same manner, the control bits for *Partition 3* and *4* can be found. This method removes 23 *X*'s out of total 28 *X*'s. The unmasked 5 *X*'s are shifted into the *X*-canceling MISR which can take care of them. A conventional *X*-masking method requires total 120 control bits (*Longest Scan Chain Length (3) * Number of Scan Chains (5) * Number of Test Patterns (8)*). The proposed partitioning method reduces 120 control bits to 45 bits (i.e., 15 control bits for each partition) for *X*-masking. However, it should be noted that the proposed method allows *X* values shifted into the *X*-canceling MISR. The amount of control bits required for selective XORing in the *X*-canceling MISR has to be considered for finding the total control bits. Hence, the total number of control bits for the proposed hybrid *X*-masking and *X*-canceling MISR approach is as follows:

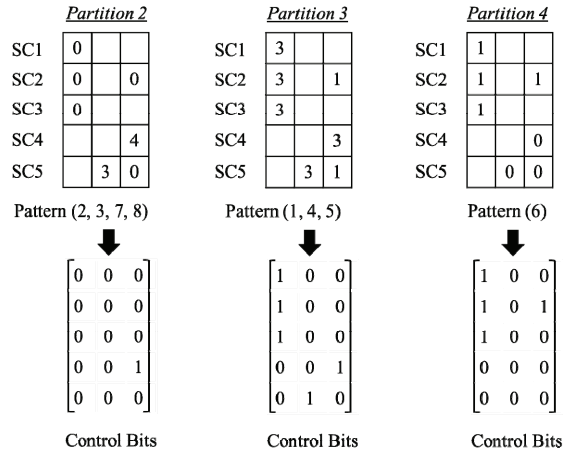

Figure 6. Control Bit Generation for Test Pattern Partitions in *X*-Masking

$$\begin{aligned}&\textit{Total Number of Control Bits}\\&= \textit{Longest Scan Chain Length} * \textit{Number of Scan Chains}\\&\quad * \textit{Number of Partitions}\\&\quad + mq * \frac{\textit{Total Number of X's Shifted to X-canceling MISR}}{m-q}\end{aligned}$$

where m is the size of MISR and the number of *X*-free combination is q.

In the hybrid *X*-handling method, *X* values are first removed by *X*-masking and the unremoved *X*'s are shifted into and taken care by *X*-canceling MISR. The more partitions that are generated by the proposed partitioning process, the more *X* values that can be removed in *X*-masking. As can be seen from the equation, using more partitions introduces an increase of control bits. A smaller number of partitions reduces the volume of control bits, however,

Algorithm 1. Finding Partitions

Input : Entire Output Response with *X*'s
Output : Output Response Partitions, Test Pattern Groups

1: Find *X*-value Correlation Analysis Results by Checking *X* Capturing Scan Cell Locations
2: Choose *Selected Scan Cell*
3: Partition *Output Responses w.r.t Selected Scan Cell*
4: Count control bits for generated *output response partitions*
5: **if** control bit data volume for current partitions is smaller than that of previous partition results **then**
6: Stop partitioning
7: **else then**
8: **go to** line 2
9: **end if**

it makes a number of *X*'s unmasked by *X*-masking which are shifted to the *X*-canceling MISR. This means that the *X*-canceling MISR requires more control bits to perform selective XORing due to a higher number of *X*'s. This in turn increases the total number of control bits for the hybrid *X*-handling method. To reduce the control bit overhead of the hybrid approach of *X*-masking and *X*-canceling MISR, it is necessary to find partitions which minimizes the total number of control bits. The partitions balances the number of reduced *X*'s by *X*-masking and the number of shifted *X*'s into *X*-canceling MISR. To make partitions, we introduce a cost function.

$$cost\ function = (Total\ Num.\ of\ Control\ Bits)_i - (Total\ Num.\ of\ Control\ Bits)_{i+1}, \quad when\ i = 1, 2, 3, 4, \cdots$$

In the cost function, i denotes an i^{th} partition round. The cost function compares the total number of control bits (control bits for *X*-masking and *X*-canceling MISR) of the current partition with the total number of control bits by the next partition round. Hence, the proposed partitioning process uses the cost function to decide whether more partitions need to be found iteratively or the partitioning should stop. If the value of the cost function is positive, the total number of control bits at the i^{th} partitioning round requires more control bits than $(i+1)^{th}$ round, hence, the partitioning continues. On the other hand, if the cost function gives a negative value, the partitioning needs to stop since the next partitioning requires more control bits than the current partitions. The proposed partitioning process continues as long as the control bit reduction from the *X*-masking stage is greater than the control bit increase from the *X*-canceling MISR. From the example in Figure 5, the first round of partitioning creates *Partition 1* and *Partition 2*. This removes 16 *X* values (4 *X*'s in the first scan cell of SC1, SC2 and SC3 with *Partition1*, the third scan cell in SC4 in *Partition 2*) and leaks 12 *X*'s to *X*-canceling MISR. If the *X*-canceling MISR has the m = 10, q = 2 configuration, the total number of control bits is 60 bits (3*5*2+10*2*12/(10-2) bits). The second partitioning process generates three partitions (*Partition 2*, *Partition 3* and *Partition 4*) and this masks 23 *X*'s and unmasks 5 *X*'s. The total number of control bits is ⌊57.5⌋ bits, 58 bit (3*5*3+10*2*5/(10-2) bits). Hence, the partitioning continues to the next round. However, if the *X*-canceling MISR configuration has m = 10 and q = 1, it does not give the same result. With this configuration, the first round of partitioning requires ⌊43.3⌋ bits, 44 bits (3*5*2+10*1*12/(10-1) bits) and the required total control bits is ⌊50.5⌋ bits, 51 bits (3*5*3+10*1*5/(10-1) bits) at the second round. In this case, the partitioning process stops at the first round since it requires the smallest control data volume.

The proposed method significantly reduces the number of *X*'s being shifted to *X*-canceling MISR. In the time-multiplexing *X*-canceling MISR [11], the scan shifting is halted for the *X*-free signature processing. This means that additional test time is required to generate the *X*-canceled combinations. Since the proposed method significantly reduces the number of *X*'s from *X*-masking, the time-multiplexing *X*-cancel MISR needs to be halted less. This can considerably reduce the test time.

Fault coverage is an important factor in test. The proposed method does not allow any non-*X* values to be masked in *X*-masking. This guarantees that observable values are not lost and additional fault simulation is not required. Hence, with the proposed method, the fault coverage is kept with less volume of control bits.

5. SIMULATION RESULTS

In this section, experimental results are presented for three industrial designs. CKT-A, CKT-B, and CKT-C having 505,050, 36,075 and 97,643 scan cells respectively. For output response compaction, 3000 test patterns are applied and the hybrid *X*-handling architecture (*X*-masking and *X*-canceling MISR) is used with 32 tester channels. For the *X*-canceling MISR configuration, we assume that the MISR size is 32 (m = 32) and the *X*-free combination size is 7 (q = 7).

Three circuits are given with their *X*-density in the first column of Table 1. The required control bits for *X*-Masking only [5], *X*-canceling MISR only [12], and the proposed method are shown

Table 1. Control Bit Data Volume and Test Time Comparisons

Circuit (X-density)	Control Bit Data Volume					Test Time		
	X-Masking Only [5]	*X*-Canceling MISR Only [12]	Proposed Method	Impv. over [5]	Impv. over [12]	*X*-Canceling MISR Only [12]	Proposed Method	Impv. over [12]
CKT-A (0.05%)	1515.15M	6.54M	5.35M	283.21	1.22	1.14	1.09	1.05
CKT-B (2.75%)	108.23M	26.57M	12.22M	8.86	2.17	1.58	1.26	1.26
CKT-C (2.38%)	292.93M	62.22M	41.13M	7.12	1.51	2.35	1.88	1.25

the second, third, and fourth columns, respectively. The fifth and sixth columns give the control bit reduction ratios by the proposed method over *X*-masking only and *X*-canceling MISR only. As can be seen for CKT-A, the *X*-canceling MISR scheme relatively reduces the number of control bits well with a low *X*-density. With higher *X*-densities, the proposed method provides a factor of 7 to 8 times of control bit reduction over *X*-masking and it achieves 1.51 to 2.17 times of control data volume reduction over *X*-canceling MISR.

The time-multiplexing *X*-canceling MISR [11] requires a scan shifting operation to be halted to find *X*-free combinations. Hence, there is a test time overhead. Note that the shadow register *X*-canceling MISR [11] is not considered. Since it requires additional input tester channels, it does not provide fair comparison results. The normalized test time can be expressed as follows:

*Normalized total testing time = 1 + [(n*x*q)/(m-q)]*

where *n* is the number of scan chains and *x* is the percentage of *X*-density. The total test time is normalized by the test time required for *X*-masking. Using the test time estimation equation [11], the normalized test times for the time-multiplexing *X*-canceling MISR and the proposed method are given. The last column shows the test time reduction ratio. The proposed method significantly reduces the test time overhead required for *X*-canceling MISR.

The fault coverage is not given in the experimental results. Because the proposed hybrid *X*-handling method does not lose any observable non-X values in the partitioning process, the fault coverage is the same as that from the *X*-canceling MISR method. In summary, the proposed method significantly reduces the output response compaction overheads, control bit data volume and test time, without losing any fault coverage.

6. CONCLUSION

The proposed hybrid *X*-handling method combines both the *X*-masking and *X*-canceling MISR architectures. This approach uses a partitioning process to exploit the *X* value inter-correlation to reduce output response compaction overheads. The experimental results show that the proposed method achieves a significant control bit reduction over existing techniques [5-7], [11] and test time reduction over [11]. With continuing 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.

7. ACKNOWLEDGMENTS

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015R1D1A1A01058856), and in part by the National Science Foundation under Grant No. CCF-1217750.

8. REFERENCES

[1] C.Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann. *OPMISR: The foundation for compressed ATPG vectors*, in Proc. Int. Test Conf, pp. 748-757, 2001.

[2] M. Hilscher, M. Braun, M. Richter, A. Leininger, and M. Gossel. *X-tolerant test data compaction with accelerated shift registers*. J. Electron Test., vol. 25, nos. 4–5, pp. 247–258, Aug. 2009.

[3] S. Wang, W. Wei, and S. T. Chakradhar. *Unknown blocking scheme for low control data volume and high observability*. in Proc. Des., Automat. Test Eur, pp. 1–6, 2007.

[4] S. Wang, K. J. Balakrishnan, and W. Wei. *X-block: An efficient LFSR reseeding-based method to block unknowns for temporal compactors.* IEEE Trans. Comput., vol. 57, no. 7, pp. 978–989, Jul. 2008.

[5] Y. Tang, 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. VLSI, vol. 14, no. 2, pp. 193–202, Feb. 2006.

[6] 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 Des. pp. 80–87, 2005.

[7] I. Pomeranz, S. Kundu, and S. M. Reddy. *On output response compression in the presence of unknown output values.* in Proc. Des. Automat. Conf, pp. 255–258, 2002.

[8] P. Wohl, J. A. Waicukauski, F. Neuveux, and E. Gizdarski. *Fully X-tolerant, very high scan compression.* in Proc. Des. Automat. Conf, pp. 362–367, Jun. 2010.

[9] S. Mitra, S.S. Lumetta, and M.Mitzenmacher. *X-Tolerant Signature Analysis.* Proc. Of International Test Conference, pp. 432-441, 2004.

[10] J. Rajski, 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.

[11] W. Rajski, J. Rajski. *Modular Compactor of Test Responses. Proc. of VLSI Test Symposium*, pp. 242-251, 2006.

[12] J.-S. Yang, N. A. Touba. *X-Canceling MISR Architectures for Output Response Compaction With Unknown Values.* IEEE Trans on Computer-Aided Design, Vol. 31, No. 9, Sep. 2012.

[13] N. A. Touba. *X-Canceling MISR – An X-Tolerant Methodology for Compacting Output Responses with Unknowns Using a MISR*. International Test Conference. IEEE, 2007.

[14] D. Czysz, G. Mrugalski, N. Mukherjee, J. Rajski, and J. Tyszer. *On Compaction Utilizing Inter and Intra-Correlation of Unknown States.* IEEE Trans on Computer-Aided Design, Vol. 29, No. 1, Jan 2010.

[15] A. Ramdas, O. Sinanoglu. *Toggle-Masking Scheme for X-Filtering*. IEEE European Test Symposium, 2012.

[16] O. Sinanoglu. *Toggle-Based Masking Scheme for Clustered Unknown Response Bits.* IEEE European Test Symposium, pp.105-110, 2011.

[17] J.-S. Yang, N.A. Touba. *Enhancing Superset X-Canceling Method with Relaxed Constraints on Fault Observation*. DOI: 10.1109/TCAD.2015.2459035

[18] J. Chung and N. A. Touba. *Exploiting X-Correlation in Output Compression via Superset X-Canceling*. Proc. of VLSI Test Symposium, pp. 182-187, 2012.