# *X*-Canceling MISR Architectures for Output Response Compaction With Unknown Values

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

***Abstract*—In this paper, an *X*-tolerant multiple-input signature register (MISR) compaction methodology that compacts output responses containing unknown *X* values is described. Each bit of the MISR signature is expressed as a linear combination in terms of *X*s by symbolic simulation. Linearly dependent combinations of the signature bits are identified with Gaussian elimination and XORed to remove *X* values and yield deterministic values. Two *X*-canceling MISR architectures are proposed and analyzed with industrial designs. This paper also shows the correlation between the estimated result based on idealized modeling and the actual data for real circuits for error coverage, hardware overhead, and other metrics. Experimental results indicate that high error coverage can be achieved with *X*-canceling MISR configurations and it highly correlates with actual results.**

***Index Terms*—Gaussian elimination, output response compaction, symbolic simulation, *X*-canceling MISR.**

## I. Introduction

SCAN TEST is a well-established design-for-testability (DFT) technique for digital circuits. With smaller feature sizes, the complexity of integrated circuits is significantly increasing. Growing design size and complexity results in longer testing times and exploding test vector volume. To alleviate the issues, test data compression methods are used. Both test stimulus compression and test response compaction are needed. Unknown *X* values cause issues in compacting output streams for test-response compaction as well as built-in self-test. *X*s arise from things such as uninitialized memory elements, nonscannable flip-flops, analog blocks, bus contention, floating tri-states, and other sources. Because *X* values that propagate indirectly or directly to the output response compactor will corrupt the signature making it unknown, *X* values can directly impact fault coverage [7].

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 DFT logic to prevent *X*-value propagation to scan cells [20]–[22]. In this method, *X* sources are forced to 0 (0-*control point*) or 1 (1-*control point*). However, since this method involves the modification of CUT, the inherent problems of the method are the increase of 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. This adds a blocking logic to the compactor and requires masking signals. Masking signals are transferred through tester channels and they are used to specify which scan chain outputs should be masked during which clock cycles. Many schemes for *X*-masking hardware design and mask control data compression have been developed [1]–[3], [11]–[14], [17], [19], [23]–[26]. 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 or an entire scan slice may be masked). This may result in some non-*X* values also becoming masked out that reduces observability and may impact the coverage, particularly, for unmodeled faults.

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 [8], [10], [15] that are mainly based on the application of systematic linear codes. Convolutional compactors [12] and circular registers [14] 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 [9], [16].

In this paper, a new *X*-tolerant scheme is introduced that removes *X* values in output streams using a MISR. It allows any number of scan chain outputs to be compacted with a conventional MISR of any size. This *X*-canceling MISR methodology can achieve arbitrarily high error coverage of scan cells that are observed in the presence of *X*s. *X*-values are eliminated by linearly dependent combinations of MISR signature bits using Gaussian elimination and XORing. Symbolic simulation helps this process to express each bit of MISR signature as a linear equation in terms of the *X*s. Two different *X*-canceling MISR architectures are presented and two state-of-art industrial designs are used for the experiments. Preliminary results were shown in [18] and [27] and a discussion of the

Manuscript received July 20, 2011; revised October 2, 2011 and December 22, 2011; accepted February 21, 2012. Date of current version August 22, 2012. This work was supported in part by the National Science Foundation, under Grant CCF-0916837. This paper was recommended by Associate Editor J. L. Dworak.

J.-S. Yang is with SungKyunKwan University, Suwon-si, Gyeonggi-Do 440-746, Korea (e-mail: js.yang@skku.edu).

N. A. Touba is with the Computer Engineering Research Center, Department of Electrical and Computer Engineering, University of Texas, Austin, TX 78712 USA (e-mail: touba@ece.utexas.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2012.2193579

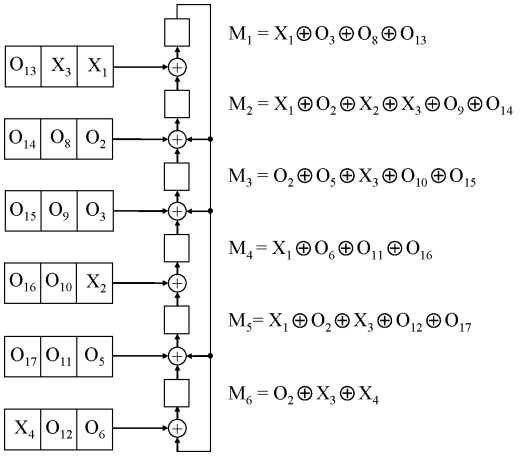


Fig. 1. Example of symbolic simulation of 6-bit MISR.

practical issues in implementing an $X$-canceling method is also given in this paper.

This paper is organized as follows. Section II overviews the symbolic simulation process and the $X$-canceling MISR bit combination identification process. Two $X$-canceling schemes are described in Section III and they are investigated with industrial designs in Section IV. Conclusions are given in Section V.

## II. Overview of Proposed $X$-Canceling MISR

This section gives an overview of the operation of an $X$-canceling MISR.

In the proposed method, the output response compaction is expressed by symbolic simulation to represent each $X$ in the output stream as a unique symbol. Each bit of the final MISR signature is expressed in terms of the symbols that correspond to each scan cell outputs and $X$ values. If there are more bits in the MISR than symbols, there should be some combinations of the MISR signature bits that are linearly dependent in terms of the symbols corresponding to the $X$s. Gaussian elimination is used to identify the combinations of linearly dependent bits and they can be XORed together to cancel out $X$s thereby yielding a deterministic $X$-free signature bit. The following described the $X$-canceling flow in an illustrative manner.

Assume that 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. Through 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 bit of MISR is represented by a linear equation of the scan cell symbols. For example, the final values of the top bit ($M_1$) and the bottom bit ($M_6$) of MISR are $X_1 \oplus O_3 \oplus O_8 \oplus O_{13}$ and $O_2 \oplus X_3 \oplus X_4$, respectively, where $X_i$ denoted an $X$ value and $O_i$ indicates a non-$X$ value in scan cells.

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. Hence, $M_1$ and $M_4$ in Fig. 1 are represented as $X_1$ and $X_3 \oplus X_4$, respectively.

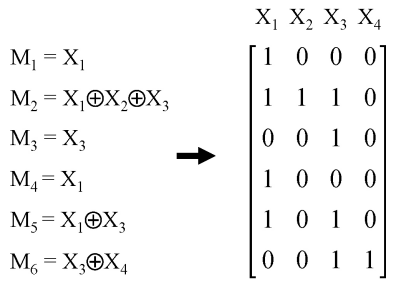


Fig. 2. Simplified linear combinations for MISR and matrix representation.


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

Fig. 2 shows the $X$ dependence of the MISR bits and its representation in the form of a matrix. In the matrix, each entry has a 1 if the MISR bit corresponding to the row depends on the $X$ corresponding to the column. Since there are four $X$s ($X_1$–$X_4$) in the output response, there are 16 possible signatures in a fault-free circuit. In this case, it might be problematic to check the MISR signature whether it is one of the valid 16 fault-free signatures.

To resolve the issue, Gauss–Jordan elimination [4] is used in the proposed method. Gauss–Jordan elimination involves performing rows operations that transform a set of columns into an identity matrix. Using the matrix in Fig. 2, the identity matrix is generated by Gauss–Jordan elimination and this is shown in Fig. 3. The last two rows in Fig. 3 have all 0s and this indicates combinations of MISR bits in which all the $X$s cancel out. The fifth row has all 0s and it implies that XORing MISR bits $M_1$, $M_3$, and $M_5$ generates an "$X$-canceled" signature bit. The $X$-free signature bit ($M_1 \oplus M_3 \oplus M_5$) is composed of scan cell combinations with non-$X$ values ($O_3 \oplus O_5 \oplus O_8 \oplus O_{10} \oplus O_{12} \oplus O_{13} \oplus O_{15} \oplus O_{17}$). Two $X$-free rows are found in Fig. 3. 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.

For an $m$-bit MISR, $k$ $X$s present anywhere in the output stream can be tolerated with error-detection capability equivalent to using an $m-k$-bit MISR with no unknown values. There are four $X$s and 6-bit MISR used in Fig. 1. In this case, the error-detection capability is equivalent to 2-bit MISR and this corresponds to the two $X$-free rows in Fig. 3. The MISR is operated across many clock cycles and may span multiple test vectors. $X$s are accumulated in the MISR and this continues until the MISR is filled up with $X$s that it

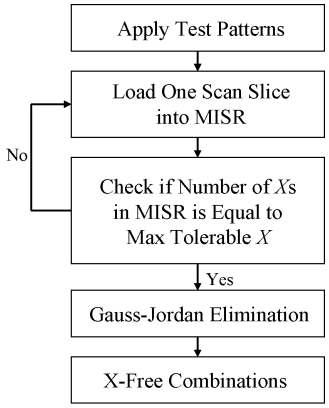


Fig. 4. $X$'s accumulation and canceling flow.

can tolerate up to. Once the MISR fills up with $X$s, the MISR signature is then processed by selectively ORing linearly dependent combinations of MISR bits in terms of the $X$s to generate $X$-free output response. Fig. 4 shows this flow.

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.

## III. $X$-Canceling MISR Schemes

Two $X$-canceling MISR schemes, their architectures, and practical issues in implementing each scheme are described in this section. The key difference is how the test channel is assigned to perform $X$-canceling.

### A. Time-Multiplexing X-Canceling MISR

1) *Architecture Details:* One approach for generating the $X$-canceled combinations is to halt scan shifting whenever the MISR is filled up with the maximum number of $X$s that it can tolerate. This method is called time-multiplexing $X$-canceling MISR; it has two phases that alternate over time: 1) a test-vector application phase, and 2) a signature processing phase. Fig. 5 shows the architecture for time-multiplexing $X$-canceling MISR.

In Fig. 5, in a test-vector application phase, there are $m$ tester channels used to load scan vectors. After the capture cycle, the output response is shifted into an $m$-bit MISR through a phase shifter as the next test vector is loaded. A phase shifter is placed before the MISR. The purpose of this phase shifter is to eliminate shift correlation among the data feeding into the MISR (and it can also be used to perform space compaction if the MISR is smaller than the data word). This proceeds across multiple clock cycles and even multiple scan vectors until the MISR fills up with $X$s. Then, the scan shifting is stopped and the second phase (i.e., a signature processing phase) begins. Linearly dependent combinations of MISR bits are computed by symbolic simulation as shown in Section II. During the signature processing phase, the $m$ tester channels are used to generate $X$-canceled combinations by selecting which of the $m$ bits in the MISR should be XORed

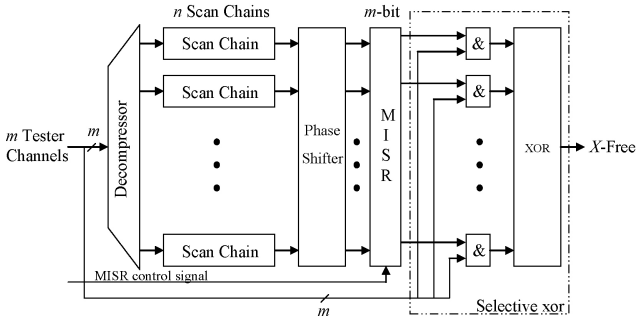


Fig. 5. Time-multiplexing $X$-canceling MISR.

TABLE I
Error Coverage Versus Number of $X$-Canceled Combinations ($q$)

| $X$-Canceled Combinations ($q$) | Error Coverage (%) | $X$-Canceled Combinations ($q$) | Error Coverage (%) |
|---|---|---|---|
| 1 | 50 | 6 | 98.44 |
| 2 | 75 | 7 | 99.2 |
| 3 | 87.5 | 8 | 99.6 |
| 4 | 93.75 | 9 | 99.8 |
| 5 | 96.88 | 10 | 99.9 |

together. The selective XOR network shown in Fig. 5 generates $X$-canceled combinations. Once the MISR signature has been processed, the MISR is reset and the test-vector application phase resumes. In this scheme, $m$ tester channels are fully utilized at all times to drive the scan vector decompressor during the test application phase and to drive the selective XOR during the signature processing phase.

In this approach, few test channels are dedicated for scan vector loading and control data transfer, and a single test channel is assigned for the output response. This leaves other tester channels for providing input stimulus; hence, this architecture is very efficient for multisite testing and for other applications, where it is desirable to have more tester channels for input stimulus and fewer channels for output response.

### B. Error Coverage, Hardware Overhead, and Other Metrics

The error coverage can be estimated based on the number of $X$-canceled combinations checked. Since the MISR with a primitive polynomial has a pseudorandom property, each $X$-canceled combination will depend on roughly half of the scan cells capturing non-$X$ values. Therefore, if $q$ $X$-canceled combinations are checked, the error coverage will be theoretically equal to $1-2^{-q}$. If an $m$-bit MISR is used, it can store up to $m-q$ $X$s and can obtain a $1-2^{-q}$ error coverage by checking $q$ linearly dependent combinations of MISR signature bits obtained via Gauss–Jordan elimination. In Fig. 3, 6-bit MISR ($m$ = 6) is used and two $X$-canceled combinations ($q$ = 2) are identified by Gauss–Jordan elimination. Hence, four $X$s ($m-q$ = 4, where $m$ = 4 and $q$ = 2) are tolerated. In this example, because two $X$-canceled combinations are checked, the error coverage equals to $1-2^{-2}$ = 75%. Note that higher error coverage can be achieved by having more $X$-canceled combinations. Table I shows the theoretical error coverage with $q$ $X$-canceled combinations.

As mentioned in the previous section, the scan shifting is halted for the signature processing phase. This means that additional testing time is required to generate the $X$-canceled combinations. The number of signature processing phases that are required depend on the $X$ density (percentage of output response bits that are $X$s), MISR size, and target error coverage. The following shows how the additional testing time is estimated with given constraints:

Constraints :

$n$ scan chains, $m$-bit MISR, $x\%$ $X$-density, and

$1 - 2^{-q}$ target error coverage.

Based on the given information, assuming a Gaussian $X$ distribution, there would be $n * x$ $X$s in one scan slice. Since the target error coverage is $1-2^{-q}$, $q$ $X$-canceled combinations need to be checked. The MISR can tolerate up to $m - q$ $X$s to achieve the target test coverage. It takes $(m - q)/(n * x)$ cycles to fill up the MISR with $m-q$ $X$s. Hence, the signature needs to be processed at every $(m - q)/(n * x)$ cycles. In the signature processing phase, $q$ cycles are needed to provide the control data for generating the $q$ $X$-canceled combinations. Therefore, if the total number of cycles needed to apply the test patterns without stopping scan shifting is $c$, then the number of additional cycles added for canceling out the $X$s is $[c/(m - q)/(n * x)] * q$. Hence, the total testing time and normalized testing time with respect to the testing time with no compaction is equal to

$$\text{Total testing time} = c + [(c*n*x*q)/(m - q)] \text{ cycles}$$

$$\text{Normalized total testing time} = 1 + [(n*x*q)/(m - q)].$$

Fig. 6 shows the normalized total testing time with different MISR sizes when 100 scan chains and 93.75% target coverage constraints are given. As can be seen, the normalized testing time decreases with a larger MISR and settles very close to 1.

In Table II, the amount of output response compression with 256-bit MISR that is obtained for output streams with different numbers of scan chains and percentages of $X$s are shown. The first two columns show the percentages of $X$s in the output stream and the number of scan chains. One of the major advantages of the proposed method 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 ($q$). The proposed method is extremely efficient when the $X$-density is low.

In this scheme, the same tester channels are used for both test vector decompression and MISR signature processing via time multiplexing. Hence, no additional control tester channels are needed other than one channel to stop and resume MISR operation. For the output response, a single tester channel can be used for transferring the $X$-canceled bits. The requirements can thus be summarized as follows:

Input tester channel : Decompressor channel + 1

Output tester channel : 1.

Note that, while the test time goes up, only one tester channel is needed for the output response; so all the other tester channels could be used for providing test stimulus thereby permitting the use of more scan chains and thereby lowering total testing time ($c$). This actually results in a lower overall testing time.

TABLE II

RESPONSE COMPRESSION WITH DIFFERENT NUMBERS OF SCAN CHAINS

| $X$-Density | Scan Chains | $X$-Canceled Combinations ($q$) | |
|---|---|---|---|
| | | $q$ = 7 (99.2% Cov) | $q$ = 9 (99.8% Cov) |
| 0.001% | 2048 | 13 895x | 10 720x |
| | 4096 | 13 895x | 10 720x |
| | 8192 | 13 895x | 10 720x |
| 0.005% | 1024 | 2779x | 2144x |
| | 2048 | 2779x | 2144x |
| | 4096 | 2779x | 2144x |
| 0.01% | 512 | 1390x | 1072x |
| | 1024 | 1390x | 1072x |
| | 2048 | 1390x | 1072x |

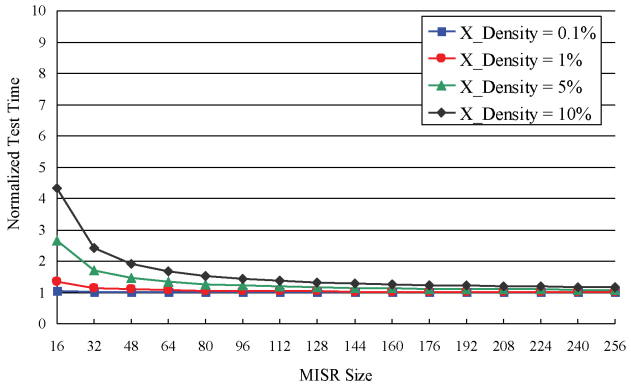


Fig. 6. Normalized Test Time with Different MISR Size.

The required hardware overhead to implement this method is mainly determined by XOR gates for a phase shifter and a selective XOR block. Hence, when there are $n$ scan chains with $f$ fanouts for a phase shifter and $m$-bit MISR, the hardware requirement can be roughly expressed as follows:

Hardware overhead :$[n * f + (m - 1)]$ two input

$XOR$ gates and one $m$-bit MISR.

### C. Shadow Register X-Canceling MISR

For some designs, there could be some complications in using the time-multiplexing $X$-canceling MISR. It is required to pause the scan load or unload operation during the processing of MISR signature and this requires the ability to retain the values in the scan cells. It might be difficult to validate or debug patterns if the cycle count of each load or unload procedure is different; it may also be preferable to have more output response channels to aid in debug and diagnosis. Shadow register $X$-canceling MISR method is presented if it is not desirable to halt scan shifting to process the intermediate MISR signatures.

1) *Architecture Details:* Fig. 7 shows the shadow register $X$-canceling architecture. There are few differences such as a shadow register, input tester channels, and selective XOR networks from the time-multiplexing $X$-canceling MISR architecture. The shadow register is placed after the main

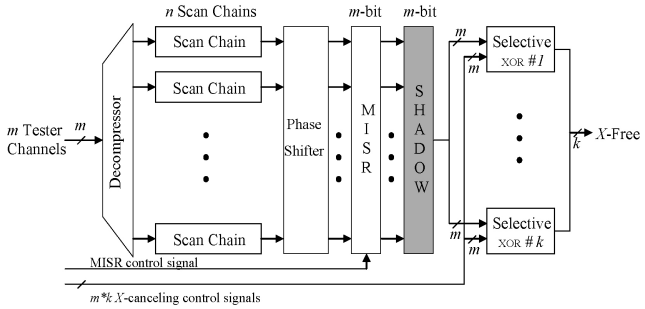


Fig. 7. Normalized testing time with different MISR size.

TABLE III
ERROR COVERAGE FOR X-CANCELING WITH SHADOW REGISTER SCHEME

| k Selective XOR Gates (k Check/Cycle) | s Cycle (Signature Transfer Cycle) | Error Coverage (%) |
|---|---|---|
| 1 | 1 | 50.00 |
| | 2 | 75.00 |
| | 3 | 87.50 |
| | 4 | 93.75 |
| 2 | 1 | 75.00 |
| | 2 | 93.75 |
| | 3 | 98.43 |
| | 4 | 99.60 |
| 3 | 1 | 87.50 |
| | 2 | 98.43 |
| | 3 | 99.80 |
| | 4 | 99.97 |
| 4 | 1 | 93.75 |
| | 2 | 99.60 |
| | 3 | 99.97 |
| | 4 | 99.99 |

MISR and retains the intermediate signature for further processing. This allows the MISR to continue to compress the scan data without halting scan chains. Additional control inputs from the tester are used to provide the control signals to one or multiple-selective XOR networks.

When the MISR fills up with $X$s, 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 intermediate signature in the shadow register is then processed to extract the $X$-canceled combinations as the next signature is being generated in a main MISR. Control signals need to be transferred while both the MISR and shadow register are operating. Therefore, extra tester channels are used to provide the control data that performs Gauss–Jordan elimination for selecting the $X$-canceled combinations. Note that in this scheme, because the shadow register gets rid of the additional test cycles for $X$-canceling, there is no additional testing time penalty.

2) *Error Coverage, Hardware Overhead, and Other Metrics:* In this scheme, because the shadow register eliminates the additional test cycles for $X$-canceling, there is no additional testing time penalty. As shown earlier, the error coverage depends on how many $X$-canceled combinations ($q$) are checked. Time-multiplexing $X$-canceling scheme requires $q$ cycles to reach $1-2^{-q}$ error coverage during each signature

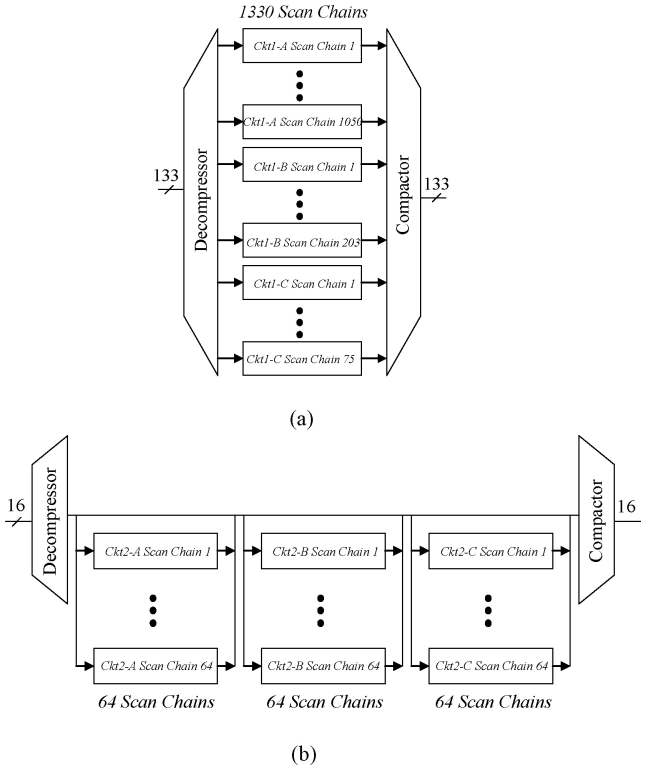


Fig. 8. Details of two industrial designs. (a) *Ckt1* design detail. (b) *Ckt2* design detail.

processing phase. However, the shadow register $X$-canceling only allows extracting $X$-canceled combinations before the next intermediate signature is transferred from the MISR to the shadow register. Because $X$-canceling combinations are extracted differently, calculating the theoretical error coverage is different in this case from what was done in the time-multiplexing $X$-canceling MISR. As shown in Fig. 7, there are $k$-selective XOR gates after the shadow register. This allows $k$ $X$-canceled combinations to be checked each clock cycle. However, the number of clock cycles over which the signature can be processed is limited by the time it takes for the MISR to fill up with $X$s again. Let the "signature transfer period" be defined as the number of clock cycles from when one intermediate signature is transferred from the MISR to the shadow register until the next one is transferred. The number of $X$-canceled combinations that are checked is determined by the number of selective XOR gates that are used times the number of cycles over which the signature is processed, which is the signature transfer period. For $k$-selective XOR gates, the error coverage is $1-2^{-k}$ after the first cycle. In the second cycle, the remaining errors that have not been covered yet are $(1-(1-2^{-k}))$, so the error coverage for them is again $1-2^{-k}$, hence the resulting error coverage after the second cycle is $(1-(1-2^{-k}))*(1-2^{-k})$ plus the error coverage after the first cycle. This is illustrated as

$$\text{Cov}_1 = 1-2^{-k} \qquad \text{(Coverage at first cycle)}$$

$$\text{Cov}_2 = \text{Cov}_1 + (1-\text{Cov}_1)*(1-2^{-k})$$

(Coverage at second cycle)

$$\text{Cov}_s = \text{Cov}_{s-1} + (1-\text{Cov}_{s-1})*(1-2^{-k}) \text{(Coverage at } s\text{th cycle).}$$

TABLE IV
TIME-MULTIPLEXING $X$-CANCELING MISR METHOD: TESTABILITY COMPARISON OF PROPOSED METHODS WITH CONVENTIONAL IMPLEMENTATION

| Circuit | Compactor | | Tester Channel | | No. of XORs | Estimated Test Time (Normalized) | Actual Test Time (Normalized) | Estimated Error Coverage (%) | Actual Error Coverage (%) |
|---|---|---|---|---|---|---|---|---|---|
| | | | Input | Output | | | | | |
| *Ckt1*-A $X$-density =0.07% | $X$-compact | | 133 | 62 | 31 865 | N/A | 1 | N/A | 99.4 |
| | $X$-canceling | $q$=5 | 134 | 1 | 7381 | 1.13 | 1.16 | 96.8 | 96.8 |
| | | $q$=6 | 134 | 1 | 7381 | 1.16 | 1.19 | 98.4 | 98.4 |
| | | $q$=7 | 134 | 1 | 7381 | 1.20 | 1.22 | 99.2 | 99.2 |
| | | $q$=8 | 134 | 1 | 7381 | 1.24 | 1.25 | 99.6 | 99.6 |
| *Ckt1*-B $X$-density =3.35% | $X$-compact | | 133 | 38 | 4135 | N/A | 1 | N/A | 36.9 |
| | $X$-canceling | $q$=5 | 134 | 1 | 1452 | 2.25 | 2.78 | 96.8 | 96.8 |
| | | $q$=6 | 134 | 1 | 1452 | 2.56 | 3.13 | 98.4 | 98.4 |
| | | $q$=7 | 134 | 1 | 1452 | 2.90 | 3.49 | 99.2 | 99.2 |
| | | $q$=8 | 134 | 1 | 1452 | 3.26 | 3.85 | 99.6 | 99.6 |
| *Ckt1*-C $X$-density =3.28% | $X$-compact | | 133 | 31 | 1031 | N/A | 1 | N/A | 86.8 |
| | $X$-canceling | $q$=5 | 134 | 1 | 556 | 1.45 | 1.57 | 96.8 | 96.6 |
| | | $q$=6 | 134 | 1 | 556 | 1.56 | 1.68 | 98.4 | 98.2 |
| | | $q$=7 | 134 | 1 | 556 | 1.68 | 1.80 | 99.2 | 99.0 |
| | | $q$=8 | 134 | 1 | 556 | 1.82 | 1.91 | 99.6 | 99.6 |
| *Ckt2*-A $X$-density =2.01% | $X$-compact | | 16 | 16 | 192 | N/A | 1 | N/A | 95.4 |
| | $X$-canceling | $q$=5 | 17 | 1 | 447 | 1.43 | 1.44 | 96.8 | 96.7 |
| | | $q$=6 | 17 | 1 | 447 | 1.52 | 1.54 | 98.4 | 98.3 |
| | | $q$=7 | 17 | 1 | 447 | 1.62 | 1.64 | 99.2 | 99.1 |
| | | $q$=8 | 17 | 1 | 447 | 1.73 | 1.74 | 99.6 | 99.5 |
| *Ckt2*-B $X$-density =0.67% | $X$-compact | | 16 | 16 | 192 | N/A | 1 | N/A | 97.9 |
| | $X$-canceling | $q$=5 | 17 | 1 | 447 | 1.22 | 1.23 | 96.8 | 96.7 |
| | | $q$=6 | 17 | 1 | 447 | 1.27 | 1.28 | 98.4 | 98.3 |
| | | $q$=7 | 17 | 1 | 447 | 1.32 | 1.33 | 99.2 | 99.1 |
| | | $q$=8 | 17 | 1 | 447 | 1.38 | 1.39 | 99.6 | 99.6 |
| *Ckt2*-C $X$-density =2.74% | $X$-compact | | 16 | 16 | 192 | N/A | 1 | N/A | 92.7 |
| | $X$-canceling | $q$=5 | 17 | 1 | 447 | 1.59 | 1.60 | 96.8 | 96.6 |
| | | $q$=6 | 17 | 1 | 447 | 1.72 | 1.74 | 98.4 | 98.2 |
| | | $q$=7 | 17 | 1 | 447 | 1.86 | 1.87 | 99.2 | 99.0 |
| | | $q$=8 | 17 | 1 | 447 | 2.00 | 2.01 | 99.6 | 99.6 |

Table III shows the error coverage for different values of $k$ and the signature transfer period, $s$.

For example, assume that the signatures are transferred from a MISR to a shadow register every three cycles ($s = 3$) and there are two selective XOR gates ($k = 2$) after a shadow register in Fig. 7. Hence, the error coverage for each cycle can be found as $\text{Cov}_1 = 1-2^{-2} = 75\%$, $\text{Cov}_2 = 0.75+0.75*(1-2^{-2}) = 93.75\%$, and $\text{Cov}_3 = 0.9375 + 0.9375 * (1 - 2^{-2}) = 98.43\%$.

Unlike time-multiplexing $X$-canceling, the shadow register $X$-canceling dedicates tester channels to provide control signals to the selective XORs. Hence, if $k$ XOR gates ($k$ checks/cycle) are used, $m * k$ input tester channels are needed for driving them, where $m$ is the size of the MISR; one input tester channel needs to be assigned to control when the MISR signature is transferred to the shadow register and reset. For the output response, $k$ tester channels are required. The requirements can thus be summarized as follows:

Input tester channels : Decompressor channels + (MISR_size * checks/cycle) + 1

Output tester channels : Checks/cycle.

Shadow register $X$-canceling MISR requires more selective XOR networks and additional one $m$-bit shadow register. Hence, when there are $n$ scan chains with $f$ fanouts and $m$-bit MISR, the hardware requirement can be roughly expressed as follows:

Hardware overhead : $[n*f + k*(m-1)]$ two input XOR gates and two $m$-bit MISR.

## IV. ANALYSIS OF PROPOSED METHODS WITH INTEL DESIGNS

Two industrial designs from Intel were analyzed in detail for the experiments. Fig. 8 shows two designs with scan chain information and tester channels for inputs and outputs.

Fig. 8(a) shows *Ckt1* and it has 133 input and output tester channels, respectively; 133 inputs are expanded into 1330 scan chains and it achieves a 10x compression [6]. *Ckt1* has three subblocks (A, B, and C) and they have 1050, 203, and 75 scan chains, respectively; 62, 38, and 31 output tester channels are assigned to *Ckt1*-A, *Ckt1*-B, and *Ckt1*-C, respectively, and two output channels are used for bypass.

*Ckt2* is shown in Fig. 8(b). *Ckt2* has relatively fewer test channels than *Ckt1* and it has 16 input and output tester channels. There are three partitions (A, B, and C) in the design that are connected in a daisy chain manner. *Ckt2*-A, *Ckt2*-B, and *Ckt2*-C all have 64 scan chains. *Ckt2* has a 4x compression

ratio; 16 inputs are expanded to fill 64 scan chains. Each of *Ckt1* and *Ckt2* submodules has different *X*-density.

The time-multiplexing *X*-canceling and shadow register *X*-canceling schemes are analyzed and compared with *X*-compact [8] that is widely used. *X*-compact is guaranteed to be able to tolerate one *X* per scan slice. However, for these two designs, the fault coverage dropped significantly with *X*-compact from the case where the output response was not compressed. The distribution of *X*s in these designs was such that many scan slices had too many *X*s to be efficiently compacted with *X*-compact. Experimental results from two *X*-canceling schemes are presented, respectively, and compared with *X*-compact.

### A. Time-Multiplexing X-Canceling MISR

Table IV shows the results for time-multiplexing *X*-canceling. A 32-bit MISR is used for each of the three blocks in *Ckt1* to compact the responses from the scan chains and to generate *X*-canceled combinations. The outputs of the scan chains are fed into a phase shifter before going to the MISR to reduce shift correlation [18]. The first column shows the circuits and the second column shows the types of compactors. As shown in Section III-A, the error coverage depends on how many *X*-canceled combinations ($q$) are checked. Results were generated for values of $q$ ranging from 5 to 8. The third column shows the number of input and output tester channels used. The formula for the required number of input and output channels was given in Section III-A. The number of two input XOR gates for hardware overhead estimation is shown in the fourth column. As described earlier, for *Ckt1*-A, a 32-bit MISR, where each scan chain output fans out to seven XOR gates in a phase shifter, is used; so the number of two input XOR gates is 7381 ($7*1050+31$). The fifth column shows the test time for each scheme. The results are normalized with respect to the results for *X*-compact. The additional testing time for control signal transfer is also normalized and shown in the fifth column. The last column shows the error coverage. Unlike other schemes, the error coverage for an *X*-canceling MISR can be estimated based on the number of *X*-canceled combinations that are observed. The experimental results show what the theory would estimate the coverage and testing time to be for purposes of comparison with the actual values. For *Ckt2*, a 64-bit MISR and a phase shifter with five fanouts per scan chain were used. Larger MISRs can hold more *X*s before needing to be processed, however, they also require more data to process each signature; so the net effect is that testing time and storage is relatively constant regardless of the MISR size. The main issue with the MISR size is the number of *X*s in a single scan slice that it can handle. The MISR size should not be smaller than the maximum number of *X*s in any scan slice.

As can be seen from Table IV, the proposed method achieves an error coverage and testing time very close to that predicted by the theoretical formula. The reason for the slight deviation is that the formulas assume the MISR can stop when it takes exactly the full number of *X*s values that it can hold. However, in practice, the *X*s are entering the MISR in clusters scan slice by scan slice; so if the next scan slice puts the number of *X*s over the limit, the MISR signature must first be processed before it can compact that scan slice. This results in some extra testing time in comparison to that predicted by the theoretical formulas.

In comparing the results for *X*-canceling with *X*-compact, many fewer output tester channels are required while arbitrarily higher error coverage can be achieved to whatever the desired level is. For *Ckt1*, less overhead is required for *X*-canceling. For *Ckt2*, the overhead is very low for both methods. Time-multiplexing *X*-canceling does have higher testing time in this scenario because the output tester channels that have been reduced have not been used for providing test stimulus. Effectively, the tester bandwidth allocated for *X*-canceling here is less than that for *X*-compact.

Looking at the individual partitions, it can be seen that *Ckt1*-A has very low *X*-density, and both *X*-compact and *X*-canceling perform very well. *X*-canceling requires many fewer output tester channels and less overhead with a bit more testing time. For *Ckt1*-B and *Ckt1*-C, the *X*-density is over 3% in both cases, and the error coverage provided by *X*-compact is low. This occurs because some scan slices have many *X*s. Note that even though *Ckt1*-B and *Ckt1*-C have similar *X*-densities, the *X*-compact coverage for *Ckt1*-B is much lower. This is because the distribution of *X*s in *Ckt1*-B is such that coverage is lost for a larger percentage of scan slices than in *Ckt1*-C. The *X*s in *Ckt1*-C are more clustered in fewer scan slices, so the percentage of scan slices where coverage is lost is less. *X*-canceling can achieve high error coverage for any distribution of *X*s, so it performs very well in terms of error coverage. The cost of achieving the higher error coverage is additional testing time, but again fewer output tester channels are required. For *Ckt2*, *X*-compact is using 32 tester channels, while *X*-canceling is using only 18 tester channels. If the 14 tester channels that are reduced with *X*-canceling were to be employed in providing test stimulus, then *X*-canceling would have lower testing time in all cases while providing greater error coverage.

As explained and shown in Table IV, time-multiplexing *X*-canceling method almost achieves the estimated error coverage. For the detailed analysis, different MISR sizes (21–32 bits) are chosen to show the testing time estimation correlation when the target error coverage is 99.2%; the error coverage analysis with respect to different *X*-canceled combinations ($q = 1 \sim q = 8$) is also illustrated. Fig. 9 shows results from *Ckt1*-A and *Ckt1*-C. As graphs show, the actual error coverage and the actual testing time correlates very closely with the estimation. Assumption in measuring testing time is that test channels are only used for *X*-canceling. In reality, because only one output channel is needed for the output response in this scheme, there would be extra tester channels available and they could be used to provide test stimulus at other sites. Hence, the actual testing time can be lower than the one shown in Fig. 9(a) and (c).

### B. Shadow Register X-Canceling MISR

Table V shows the results from shadow register *X*-canceling MISR scheme. Unlike time-multiplexing method, the control signals are provided via dedicated channels while scan shift is running. *X*-canceled combinations are generated without introducing additional testing time. Hence, the testing time

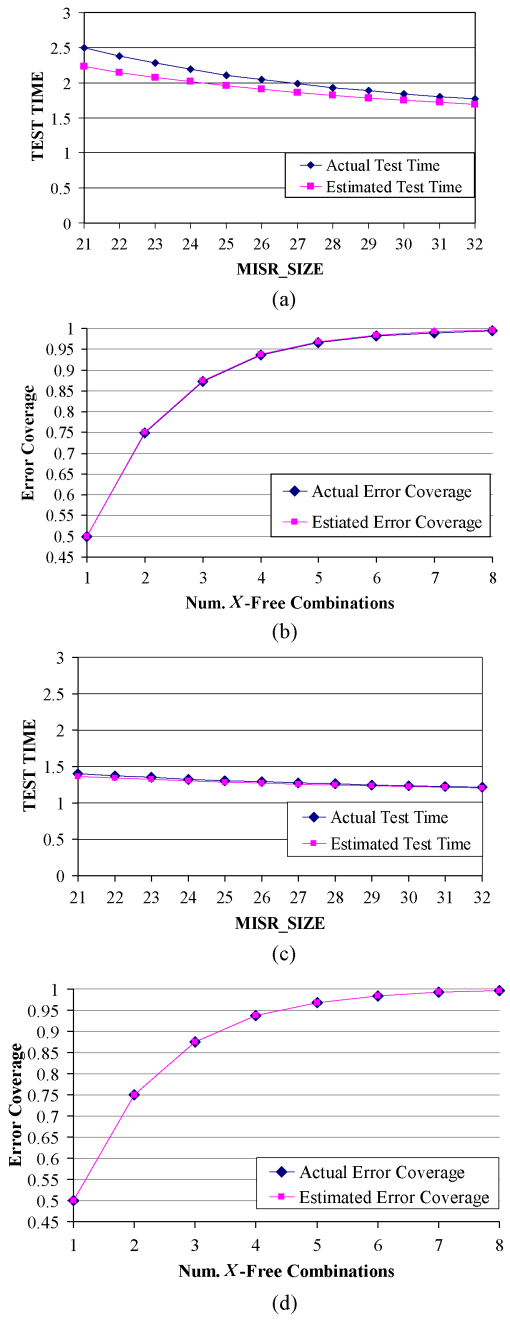


Fig. 9. Testing time and error coverage analysis with different configurations for time-multiplexing $X$-canceling scheme. (a) *Ckt1*-A: Testing time analysis. (b) *Ckt1*-A: Error coverage analysis. (c) *Ckt1*-C: Testing time analysis. (d) *Ckt1*-C: Error coverage analysis.

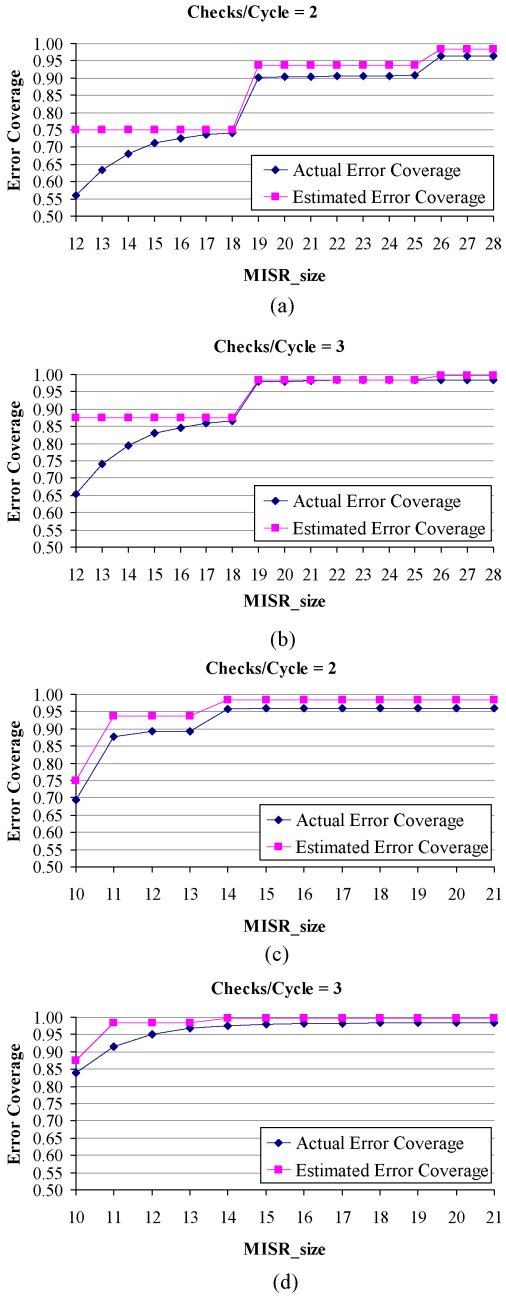


Fig. 10. Testing time and error coverage analysis with different configurations for time-multiplexing $X$-canceling scheme. (a) *Ckt1*-B: Error coverage analysis with two checks/cycle. (b) *Ckt1*-B: Error coverage analysis with three checks/cycle. (c) *Ckt1*-C: Error coverage analysis with two checks/cycle. (d) *Ckt1*-C: Error coverage analysis with three checks/cycle.

is exactly same for $X$-compact and this method. As explained in Section III, the error coverage differs with the numbers of checks/cycle and this is shown in the second column. In column three and four, the number of input and output channels and the number of two input XOR gates are calculated by the estimation equations in Section III. The last column shows the error coverage. The error coverage for $X$-canceling can be made arbitrarily high. In this case, improving the error coverage comes at the cost of requiring more checks/cycle that requires more input tester channels, however, the testing time remains constant.

Fig. 10 shows more data with different MISR size and the three checks/cycle for *Ckt1*-B and *Ckt1*-C (*Ckt*-A is not shown because the low $X$-density shows very good correlation results). The target error coverage can be found in Table III. As explained, the estimated error coverage varies with respect to checks/cycle and signature transfer cycle. Fig. 10 shows increasing error coverage with bigger MISRs because they can tolerate more $X$-values and this increases the signature transfer cycle. For smaller MISRs, there is some correlation gap between the estimated and actual value. Because MISRs are easily filled up with $X$s, the actual signature transfer cycle is shorter than estimation. As can be seen, as the MISR size

TABLE V
SHADOW REGISTER *X*-CANCELING MISR METHOD: TESTABILITY COMPARISON OF PROPOSED METHODS WITH CONVENTIONAL IMPLEMENTATION

| Circuit | Compactor | Check/Cycle | Tester Channel | | No. of XORs | Estimated Error Coverage (%) | Actual Error Coverage (%) |
|---|---|---|---|---|---|---|---|
| | | | Input | Output | | | |
| *Ckt1*-A *X*-density =0.07% | *X*-compact | N/A | 133 | 62 | 31 865 | N/A | 99.4 |
| | *X*-canceling 12-bit MISR | 1 | 146 | 1 | 5261 | 93.7 | 93.7 |
| | | 2 | 158 | 2 | 5272 | 99.6 | 98.2 |
| | | 3 | 170 | 3 | 5283 | 99.9 | 99.1 |
| | | 4 | 182 | 4 | 5284 | 99.9 | 99.4 |
| *Ckt1*-B *X*-density =3.35% | *X*-compact | N/A | 133 | 38 | 4135 | N/A | 36.9 |
| | *X*-canceling 19-bit MISR | 1 | 148 | 1 | 1845 | 75.0 | 74.4 |
| | | 2 | 162 | 2 | 1863 | 93.7 | 90.2 |
| | | 3 | 176 | 3 | 1881 | 98.4 | 97.9 |
| | | 4 | 190 | 4 | 1899 | 99.6 | 98.9 |
| *Ckt1*-C *X*-density =3.28% | *X*-compact | N/A | 133 | 31 | 1031 | N/A | 86.8 |
| | *X*-canceling 14-bit MISR | 1 | 143 | 1 | 1028 | 87.5 | 87.3 |
| | | 2 | 152 | 2 | 1041 | 98.4 | 95.8 |
| | | 3 | 161 | 3 | 1054 | 99.8 | 97.6 |
| | | 4 | 170 | 4 | 1067 | 99.9 | 98.8 |
| *Ckt2*-A *X*-density =2.01% | *X*-compact | N/A | 16 | 16 | 192 | N/A | 95.4 |
| | *X*-canceling 16-bit MISR | 1 | 33 | 1 | 463 | 93.75 | 93.60 |
| | | 2 | 49 | 2 | 478 | 99.60 | 98.01 |
| | | 3 | 65 | 3 | 493 | 99.97 | 98.92 |
| | | 4 | 81 | 4 | 508 | 99.99 | 99.20 |
| *Ckt2*-B *X*-density =0.67% | *X*-compact | N/A | 16 | 16 | 192 | N/A | 97.9 |
| | *X*-canceling 16-bit MISR | 1 | 33 | 1 | 463 | 93.75 | 93.67 |
| | | 2 | 49 | 2 | 478 | 99.60 | 98.07 |
| | | 3 | 65 | 3 | 493 | 99.97 | 98.96 |
| | | 4 | 81 | 4 | 508 | 99.99 | 99.24 |
| *Ckt2*-C *X*-density =2.74% | *X*-compact | N/A | 16 | 16 | 192 | N/A | 92.7 |
| | *X*-canceling 16-bit MISR | 1 | 33 | 1 | 463 | 93.75 | 93.63 |
| | | 2 | 49 | 2 | 478 | 99.60 | 98.20 |
| | | 3 | 65 | 3 | 493 | 99.97 | 99.10 |
| | | 4 | 81 | 4 | 508 | 99.99 | 99.39 |

grows, the actual error coverage shows a very good correlation with estimation.

It should be noted that the proposed methods provide good error coverage estimation and design requirements before the design stage. This would be a great help for planning *X*-canceling architectures and for estimating their error coverage.

### C. Output Response Compression

In Table VI, the amount of output response compression that is obtained for output streams with *Ckt1*-A, B, and C, and with *Ckt2*-A, B, and C are shown. The results are shown with the configurations used for Tables IV and V.

In Table VI(a), for *Ckt1* and *Ckt2*, because the compression ratio depends on the number of *X*-canceled combinations ($q$), the compression ratio with different number of *X*-canceled combinations are shown. The lower *X*-density is required that reduces the number of control bits stored. The experimental results show that the time-multiplexing *X*-canceling MISR method achieves 17x–200x compression ratio depending on different *X*-densities. Note that the compression ratio varies with the number of *X*s and this method can achieve a compression ratio higher than 10 000x when the *X*-density is 0.001% [18].

In Table VI(b), the compression ratios are shown with different check/cycle and different MISR size. Scan shifting is not halted in the shadow register *X*-canceling method, and *X*-canceled combinations are generated from the shadow register with scan shifting. Hence, this requires more control bits compared to the time-multiplexing *X*-canceling method. The number of control bits is determined by the check/cycle and MISR size. This explains why the compression ratios for *Ckt2*-A, B, and C are the same.

### D. Fault Grading Analysis

Fault grading was performed on *Ckt1* to see the actual fault coverage that is achieved by the *X*-canceling methods and *X*-compact. For each block in *Ckt1*, a 32-bit MISR with $q = 8$ configuration is used for time-multiplexing *X*-canceling. For shadow register *X*-canceling, because this adds dedicated tester chandlers, a configuration was selected that has a similar number of tester channels to *X*-compact for a fair comparison. As its configurations, 12-bit MISR with four checks/cycle for *Ckt1*-A, 19-bit MISR with two checks/cycle for *Ckt1*-B, and 14-bit MISR with two checks/cycle for *Ckt1*-C are used. This requires 268 tester channels and *X*-compact needs 266 channels. The fault coverage for 3000 ATPG patterns is shown in Fig. 11. Without any compression, fault coverage slightly over 90% is obtained. As shown in Tables IV and V, the *X*-canceling MISR schemes achieve high error coverage that translates to fault coverage, which is very close to what

TABLE VI
OUTPUT RESPONSE COMPRESSION RESULTS

(a) Time-Multiplexing $X$-Canceling MISR

| *Ckt* | No. $X$-Canceled Comb. ($q$) | Compression Ratio | *Ckt* | No. $X$-Canceled Comb. ($q$) | Compression Ratio |
|---|---|---|---|---|---|
| *Ckt1*-A | 5 | 204.6x | *Ckt2*-A | 5 | 61.9x |
| $X$-density | 6 | 138.5x | $X$-density | 6 | 42.6x |
| =0.07% | 7 | 95.0x | =2.01% | 7 | 30.6x |
| | 8 | 69.3x | | 8 | 22.8x |
| *Ckt1*-B | 5 | 21.3x | *Ckt2*-B | 5 | 120.9x |
| $X$-density | 6 | 14.2x | $X$-density | 6 | 82.1x |
| =3.35% | 7 | 10.0x | =0.67% | 7 | 59.4x |
| | 8 | 7.4x | | 8 | 43.8x |
| *Ckt1*-C | 5 | 59.1x | *Ckt2*-C | 5 | 45.1x |
| $X$-density | 6 | 39.6x | $X$-density | 6 | 30.8x |
| =3.28% | 7 | 27.9x | =2.74% | 7 | 22.1x |
| | 8 | 20.3x | | 8 | 16.6x |

(b) Shadow Register $X$-Canceling MISR

| *Ckt* | Check/Cycle | Compression Ratio | *Ckt* | Check/Cycle | Compression Ratio |
|---|---|---|---|---|---|
| *Ckt1*-A | 1 | 11.1x | *Ckt2*-A | 1 | 8.3x |
| 12-bit | 2 | 5.6x | 16-bit | 2 | 4.2x |
| MISR | 3 | 3.7x | MISR | 3 | 2.8x |
| | 4 | 2.8x | | 4 | 2.1x |
| *Ckt1*-B | 1 | 7.0x | *Ckt2*-B | 1 | 8.3x |
| 19-bit | 2 | 3.5x | 16-bit | 2 | 4.2x |
| MISR | 3 | 2.3x | MISR | 3 | 2.8x |
| | 4 | 1.8x | | 4 | 2.1x |
| *Ckt1*-C | 1 | 9.5x | *Ckt2*-C | 1 | 8.3x |
| 14-bit | 2 | 4.8x | 16-bit | 2 | 4.2x |
| MISR | 3 | 3.2x | MISR | 3 | 2.8x |
| | 4 | 2.4x | | 4 | 2.1x |

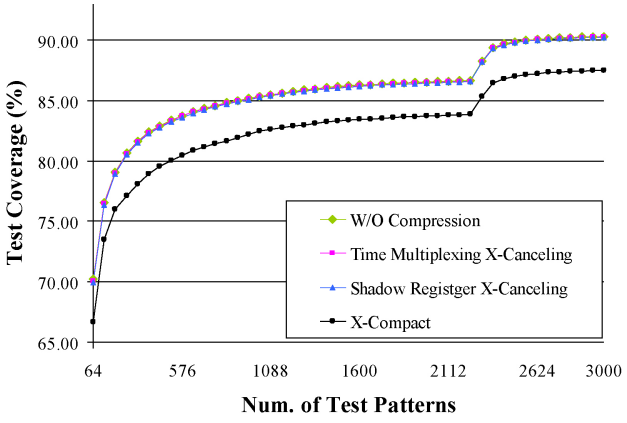


Fig. 11. Fault grading results for *Ckt1* with different schemes.

is obtained without any compression. The fault coverage for $X$-compact, however, approaches about 86%–85%, which is 2%–3% lower than the proposed methods.

## V. CONCLUSION

The proposed $X$-canceling architectures showed the advantages in terms of their scalability and ability to systematically achieve high fault coverage regardless of the distribution of $X$s. Because the tester requirement depends only on the total number of $X$s in the output response and is independent on the design size, number of test vectors, or scan architectures, it scales well and achieves very high coverage. Two different architectures can be selected based on test environments such as tester channel availability, scan shift pause availability, and so on.

In-depth analysis showed that the proposed methods achieve very good correlation between the estimations provided in this paper and the actual data. Hence, the proposed methods can be used in the early design stage with post-silicon for the efficient output response analysis. It should also be noted that the proposed methods can be incorporated with other $X$-handling techniques, such as $X$-masking, and others, as a hybrid $X$-canceling method.

## 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.*, 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 Des.*, 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.*, 2004, pp. 452–461.

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

[5] R. Garg, R. Putman, and N. A. Touba, "Increasing output compaction in presence of unknowns using an $X$-canceling MISR with deterministic observation," in *Proc. IEEE VLSI Test Symp.*, Apr.–May 2008, pp. 35–42.

[6] I. Hamzaoglu and J. H. Patel, "Reducing test application time for full scan embedded cores," in *Proc. Dig. Papers 29th Annu. Int. Symp. Fault Tolerant Comput.*, 1999, pp. 260–267.

[7] 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.

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

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

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

[11] I. Pomeranz, S. Kundu, and S. M. Reddy, "On output response compression in the presence of unknown output values," in *Proc. Des. Automat. Conf.*, 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 Des.*, vol. 24, no. 4, pp. 622–634, Apr. 2005.

[13] J. Rajski J. Tyszer, G. Mrugalski, W.-T. Cheng, N. Mukherjee, and M. Kassab, "$X$-press compactor for 1000x reduction of test data," in *Proc. Int. Test Conf.*, no. 18.1. Oct. 2006, pp. 1–10.

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

[15] M. Sharma and W.-T. Cheng, "$X$-filter: Filtering unknowns from compacted test responses," in *Proc. Int. Test Conf.*, Nov. 2005, pp. 1–9.

[16] O. Sinanoglu and S. Almukhaizim, "$X$-alignment techniques for improving the observability of response compactors," in *Proc. Int. Test Conf.*, Nov. 2009, pp. 1–10.

[17] 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.

[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.*, no. 6.2. Oct. 2007, pp. 1–10.

[19] E. H. Volkerink and S. Mitra, "Response compaction with any number of unknowns using a new LFSR architecture," in *Proc. Des. Automat. Conf.*, 2005, pp. 117–122.

[20] L. T. Wang, C.-W. Wu, and X. Wen, *VLSI Test Principles and Architectures*. San Mateo, CA: Morgan Kaufmann, 2006.
[21] 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.*, 2007, pp. 1–6.
[22] 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.
[23] 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.*, 2001, pp. 54–63.
[24] 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.*, 2003, pp. 727–736.
[25] P. Wohl, J. A. Waicukauski, and S. Patel, "Scalable selector architecture for *X*-tolerant deterministic BIST," in *Proc. Des. Automat. Conf.*, 2004, pp. 934–939.
[26] P. Wohl, J. A. Waicukauski, F. Neuveux, and E. Gizdarski, "Fully *X*-tolerant, very high scan compression," in *Proc. Des. Automat. Conf.*, Jun. 2010, pp. 362–367.
[27] J.-S. Yang, N. A. Touba, S.-Y. Yang, and T. M. Mak, "Industrial case study for *X*-canceling MISR," in *Proc. Int. Test Conf.*, no. 17.2. Nov. 2009, pp. 1–10.

**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, in 2007 and 2009, respectively, all in electrical and computer engineering.

After graduation, he worked for Intel Corporation for three years. He is currently with SungKyunKwan University in Korea. His current research interests include very large-scale integration testing, silicon debugging, and nanometer-scale testing and design methodologies.

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

**Nur A. Touba** (SM'05–F'09) received the B.S. degree from the University of Minnesota, Minneapolis, in 1990, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, 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.

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