

# X-Stacking – A Method for Reducing Control Data for Output Compaction

Rudrajit Datta and Nur A. Touba

Computer Engineering Research Center  
University of Texas at Austin, Austin, TX 78712  
[rudrajit@utexas.edu](mailto:rudrajit@utexas.edu), [touba@ece.utexas.edu](mailto:touba@ece.utexas.edu)

## Abstract

*This paper proposes an X-stacking technique for efficiently reducing the number of unknown values ( $X$ 's) in the output response, thereby, in turn reducing the amount of control data required for output compaction without losing fault coverage. Output response streams containing  $X$ 's are fed into circular registers, which stack  $X$ 's on top of one another. The contents of the circular registers are then passed on to conventional output compaction circuitry where each stack of  $X$ 's can then be treated as a single  $X$  thus substantially reducing the overall number of  $X$ 's present in the output data thereby requiring less control data to perform output compaction. Some loss of observability may occur due to non- $X$  values getting stacked with  $X$  values, but this can be compensated by adding some additional top-up test vectors. Results indicate that the overall test compression even with factoring in the additional test vectors is significantly improved with the proposed method. Different ways for implementing the circular registers, including configuring them from the scan cells themselves, are investigated for keeping the overhead for the proposed method very small.*

**Keywords:** X stacking; output compression; output masking;

## 1. Introduction

Compacting output streams that have unknown ' $X$ ' values is a major issue for test compression and BIST. Uninitialized memory elements, bus contention, floating tri-states, and other sources introduce unknown values.  $X$  values corrupt the final signature making it unknown. A number of schemes have been developed to deal with the problem of  $X$ 's in the output response.

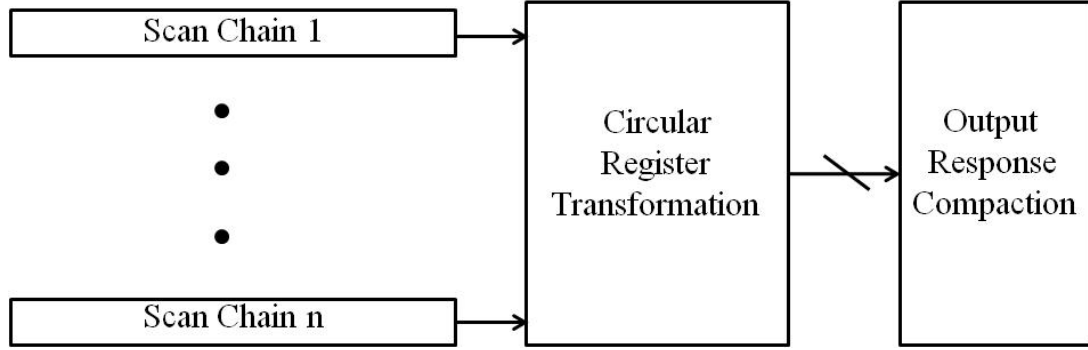
One approach is to modify the circuit-under-test (CUT) to eliminate the sources of  $X$ -values. This involves blocking sources of  $X$  within the circuit by inserting design-for-testability (DFT) hardware to prevent  $X$ 's from propagating into scan cells [Wang 06]. Another approach, which does not require modifying the CUT, is  $X$ -masking which masks out  $X$ 's at the input to the compactor. Mask control data is 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 [Barnhart 01], [Wohl 01, 03, 04], [Pomeranz 02], [Chickermane 04], [Volkerink 05], [Chao 05], [Tang 06], [Rajski 06a], [Rajski 08], [Mrugalski 09]. A third approach is to use an  $X$ -tolerant compactor which 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 [Mitra 04a], [Patel 03], [Sharma 05], convolutional compactors [Rajski 05], and circular registers [Rajski 06b], [Gizdarski 10].

In [Touba 07], the concept of canceling out  $X$ 's from MISR signatures was proposed. An  $X$ -canceling MISR methodology was described which 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.

For  $X$ -masking and  $X$ -tolerant compactors, the amount of control data required to compact output streams is generally proportional to the density of  $X$ 's in the output stream. For  $X$ -canceling [Touba 07], the amount of control data is directly proportional to the total number of  $X$ 's. In this paper, an  $X$ -stacking methodology is proposed which uses circular registers to stack  $X$ 's on top of one another such that the stacked  $X$ 's can be treated as a single  $X$  by the subsequent output compaction circuitry. By so doing, the control data required for output compaction can be substantially reduced. The proposed methodology can be used to transform the raw output response streams shifted out of the scan chains into a new one with a smaller number of  $X$ 's that is then passed on to the output compaction circuitry, as illustrated in Fig. 1. The output compaction circuitry can be any of the existing  $X$ -masking or  $X$ -tolerant compaction schemes. The methodology is very efficient for use with an  $X$ -canceling MISR [Touba 07] where the

amount of compression depends directly on the number of  $X$ 's in the output stream. Results are shown in Sec. 4 illustrating that a factor of 2 improvement in compaction can be achieved using the proposed methodology combined with an  $X$ -canceling MISR.

The rest of the paper has been organized as follows. Sections 2 and 3 describe the proposed scheme and its implementation in detail. Section 4 presents the simulation results and section 5 concludes the paper.



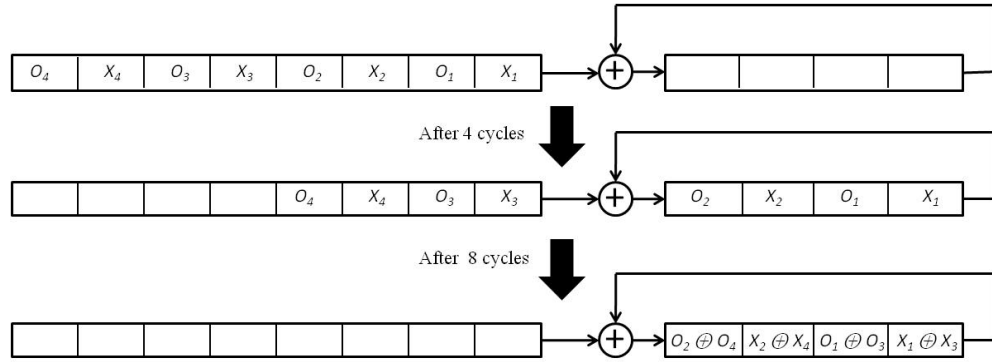
**Figure 1: Block Diagram of Proposed Scheme**

## 2. Proposed Scheme

A circular register [David 80] is a feedback register with  $f(x) = 1 + x^n$  as the feedback polynomial. So, if a stream  $S$  of length  $L$ , is fed into a circular register of length  $r$  ( $L > r$ ), every  $i^{th}$  value of the stream  $S$ , would XOR with the  $(i+r)^{th}$  value. Each scan chain shifting out an output stream feeds into circular registers as illustrated in Fig. 1. The circular register have the nice property that the  $X$ 's do not spread as they would in a MISR, but simply cycle around and get XORed with additional data coming from the scan chains. If two  $X$ 's are "stacked" together (i.e., XORed together) in the same cell of the circular register, then they are effectively merged into a single  $X$  thereby eliminating one  $X$  from the output response. The length of the circular registers need not be the same for each scan chain, but could be programmed to be different lengths. Depending on the distribution of  $X$ 's in the output data, the length of each circular register can be optimized to get the best results. After a certain number of clock cycles of compaction which could be equal to the full length of a scan chain or a partial length of a scan chain, the contents of the circular registers are fed out to the output response compaction circuitry and the circular register is reset to all-0. When a non- $X$  value gets stacked with an  $X$  value, then it can no longer be observed. So this limits how aggressively the output response can be compacted in the circular register. If the compaction is done too aggressively, the observability will degrade to the point where the fault coverage is reduced significantly necessitating a lot of additional test vectors to be added to restore the fault coverage. The amount of compaction in the circular registers can be adjusted in two ways. One is by programming the length of the circular register used for each scan chain, and the other is in selecting the number of cycles in which the scan output is fed into the circular register before flushing the circular register. Both of these things can be adjusted at the end of the design flow. The hardware that is inserted for the proposed method is generic, and only after the actual scan vectors are known, the length of the circular registers and the number of cycles that it is loaded can be programmed to optimize the overall results.

Figure 2 illustrates the working of the proposed scheme with a cycle by cycle example. In this case, the output stream in the scan chain contains four  $X$ -values and four non- $X$  values, denoted here by  $O$ . We use a circular register of length four for the purpose of this simple example. At the beginning, the circular register is reset to all-0. After the first four values of the output stream have been shifted into the circular register, the state of the circular register is as shown in the figure. Now the next element of the output stream being shifted in would be XORed with the rightmost element in the circular register, which in the example in Fig. 2 is  $X_1$ . So,  $X_3$  which is the next element from the output stream being shifted in, is XORed with  $X_1$ . Subsequently, as the shifting continues,  $O_1$  is XORed with  $O_3$ ,  $X_2$  with  $X_4$  and  $O_3$  with  $O_4$ . The final state of the circular register, which is that after eight cycles of shifting, is also shown in Fig. 2. As can be seen, the first and the third cells (from the right) of the circular register contain two  $X$ 's XORed with each other. For the subsequent compaction circuitry, these XORed  $X$  values can be considered as a single  $X$ . So, in this example from four  $X$ 's in the output stream in the scan chain, the number of  $X$ 's is effectively reduced to two in the circular registers. For an  $X$ -canceling MISR [Touba 07] where the control data is directly proportional to the number of  $X$ 's, this would achieve a factor of 2 better compaction. This explains the basic premise of the proposed scheme – that when an output stream filled with  $X$ 's is run through a circular register of

shorter length than the scan chain containing the output stream, some of the  $X$ 's would stack up on top of each other. These stacked up  $X$ 's can henceforth be treated as a single entity for the remainder of the test operation.



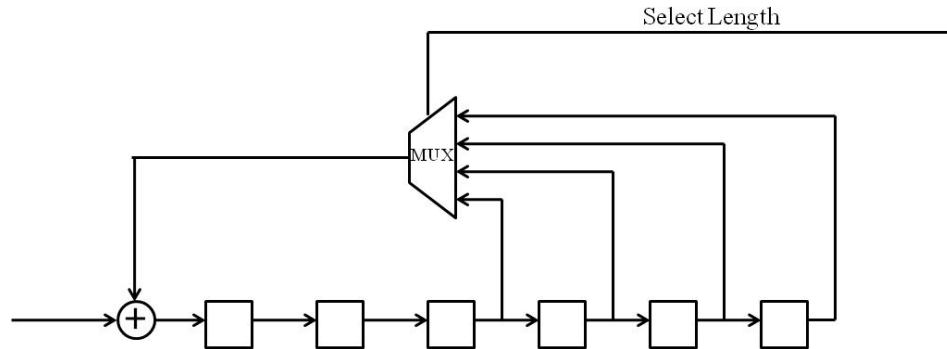
**Figure 2:** Example of Cycle-by-Cycle Operation

Note that, although not shown in the example in Fig. 2, it is also possible for non- $X$  values to be XORed together (i.e., stacked) with  $X$  values which would make the non- $X$  value unobservable. If some fault is only detectable in that particular scan cell for that particular test vector, coverage of that fault would be lost. Fault coverage can be restored by adding some additional test vectors at the end to top-up the fault coverage. The additional test vectors would subtract from the compression gains using the proposed method, so it is important to optimize the aggressiveness of  $X$ -stacking to maximize the overall net compression achieved. As  $X$ -stacking is increased through using a smaller circular register and compacting over more clock cycles, more  $X$ 's can be reduced, but then more top-up vectors are required. By programming the size of the circular registers and choosing the compaction period appropriately, experimental results show that the reduction in  $X$ 's through  $X$ -stacking improves the overall compression much more than the data added for the additional top-up test vectors.

### 3. Implementation

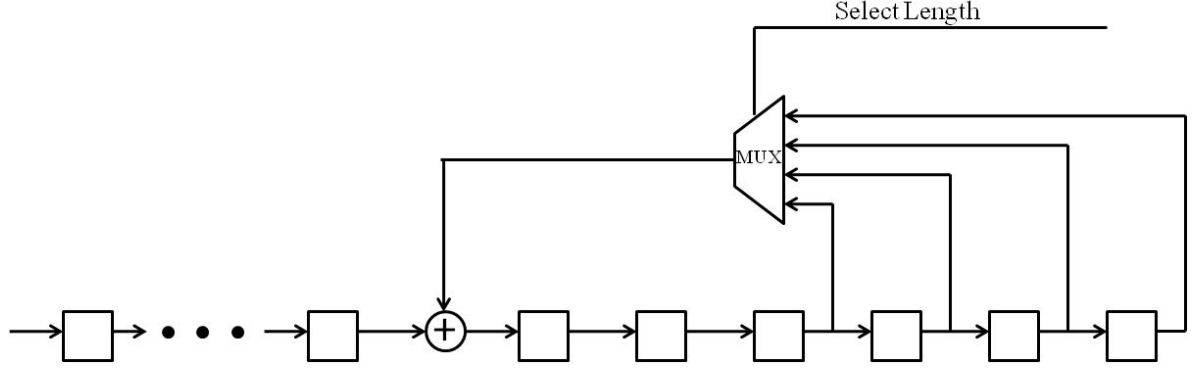
As discussed in the previous section, the best results are achieved when the lengths of the circular registers for each scan chain are customized depending on the  $X$ -density of the scan chain feeding it. This requires the need for programmable length circular registers which uses MUXes and corresponding control bits to adjust the shift length as illustrated in Fig. 3. For the proposed scheme, the lengths of the circular registers can be programmed once at the start of the test session. Once the sources of  $X$ 's in the circuit and the  $X$ -density of the scan chains are known, each circular register's length can be suitably optimized. They would not require any more tuning for the remainder of the test session thereby keeping the number of control bits for selecting the size of the circular registers very minimal.

Alternatively, it is possible to customize the circular register length for partitions of test vectors. At the start of each partition of test vectors, the control data would be loaded to set the circular register length. This will require more control data, but it allows for better optimization of the compaction and can reduce the number required top-up vectors. It can be cost-effective in situations where the reduction in the top-up vectors is sufficiently large to justify the cost of the additional control bits for programming the length of the circular register.



**Figure 3:** Circular register of programmable length with one control input

The main hardware cost of the proposed method is the circular registers. However, this overhead can be reduced by implementing them via reconfiguring the scan chains themselves into programmable length circular registers instead of using separate hardware. This is illustrated in Fig. 4. The last  $n$  scan cells can be configured as a programmable circular register of up to length  $n$ . After the capture cycle, as the whole scan chain is shifted to load in the next test vector, the last  $n$  flip-flops would be performing the circular register compaction operation.



**Figure 4:** Scan Chain Reconfigured as Circular Register of Programmable Length

The DFT hardware that is inserted for the proposed method consists of the feedback lines and MUX needed to implement the programmable length circular registers at the end of each scan chain. If a 4-to-1 MUX is used, then 4 different size circular registers can be configured for each scan chain. Moreover, the number of clock cycles that the circular register should compact before its contents are flushed to the output compaction circuit can also be customized. These can be optimized by either using a hill climbing procedure or simulated annealing. Starting for an initial default configuration where the number of cycles for compaction with the circular register is equal to the length of each scan chain, and each circular register has a midway length, the initial compression achieved can be determined. Then the length of each circular register can be incrementally changed and if the compression improves, the new configuration is kept, otherwise it reverts back to the previous value. The same can be done with the number of cycles of compaction. This is done until no further improvement can be made with incremental changes.

## 4. Experimental Results

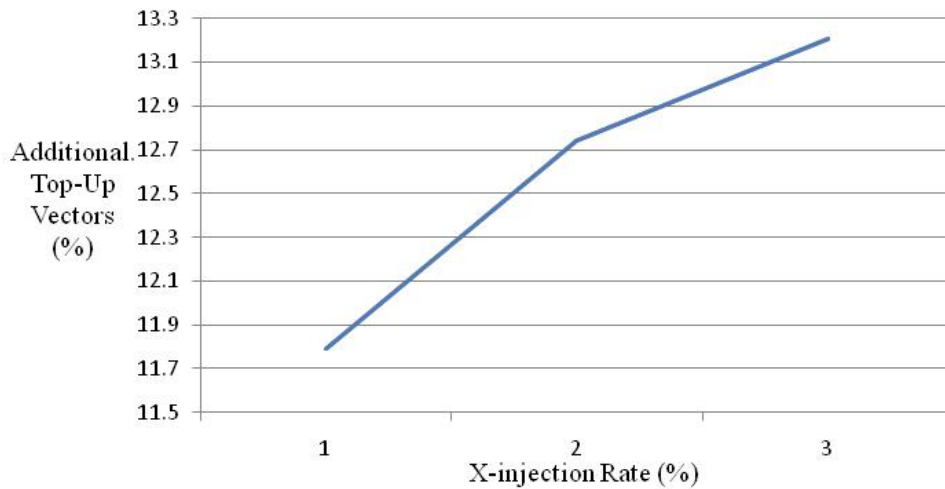
Experiments were performed to evaluate the proposed method when used in conjunction with an X-canceling MISR [Touba 07]. Table 1 shows the simulation results for the proposed method.  $X$ 's were injected with an exponential distribution in the number of  $X$ 's captured across the scan cells to model the typical distribution seen in industrial circuits. The results in Table 1 are for an injection rate of 1%  $X$ 's. For each circuit, the number of scan chains is shown followed by the drop in fault coverage that occurred using the circular registers as an intermediate compaction step before X-canceling. The improvement factor for the compaction is shown in the 4<sup>th</sup> column. This shows the ratio of  $X$ 's entering the MISR without and with the circular registers. As can be seen, approximately 2x more output compaction was achieved using the proposed method. The fifth column shows the amount of additional test vectors that needed to be added to top-up the fault coverage back to 100% of detectable faults as a percentage of the original number of test vectors. This ranged from roughly 9% to 22%, with an average of 15%. These additional test vectors need to be stored in compressed form on the chip and the output response needs to be compacted as well. Factoring the cost for the additional test vectors, the last column shows the overall improvement in test compression considering both test vectors and output response. The overall improvement factor is calculated as  $(total\ bits\ on\ tester\ with\ no\ X\text{-}stacking) / (total\ bits\ on\ tester\ with\ X\text{-}stacking)$ . As can be seen, the overall compression improves by a factor of around 1.5x on average. This is a significant improvement in overall test compression. In terms of hardware overhead, as discussed in Sec. 3, the circular registers can be configured from the scan cells themselves, so the cost would be only the additional feedback and MUXes needed to implement the programmable length circular registers plus two flip-flops per scan chain to store the configuration bits for selecting the length of the circular register.

The test vectors were generated using *Atalanta* [Lee 93] and fault simulation was performed using *fsim* [Lee 91]. While generating test vectors using *Atalanta* [Lee 93], the random vector count was set to zero, so that unnecessary random vectors did not alter the vector count. Figure 5 shows the impact of increasing the percentage of  $X$ 's in the

circuit on test vector overhead (i.e., percentage of additional top-up test vectors required for 100% coverage of detectable faults). As shown in Fig. 5, with higher percentage of  $X$ 's, more non- $X$  values are corrupted leading to drop in fault coverage and a consequent need for additional test vectors. This data is shown for a constant output compaction improvement ratio of 2x.

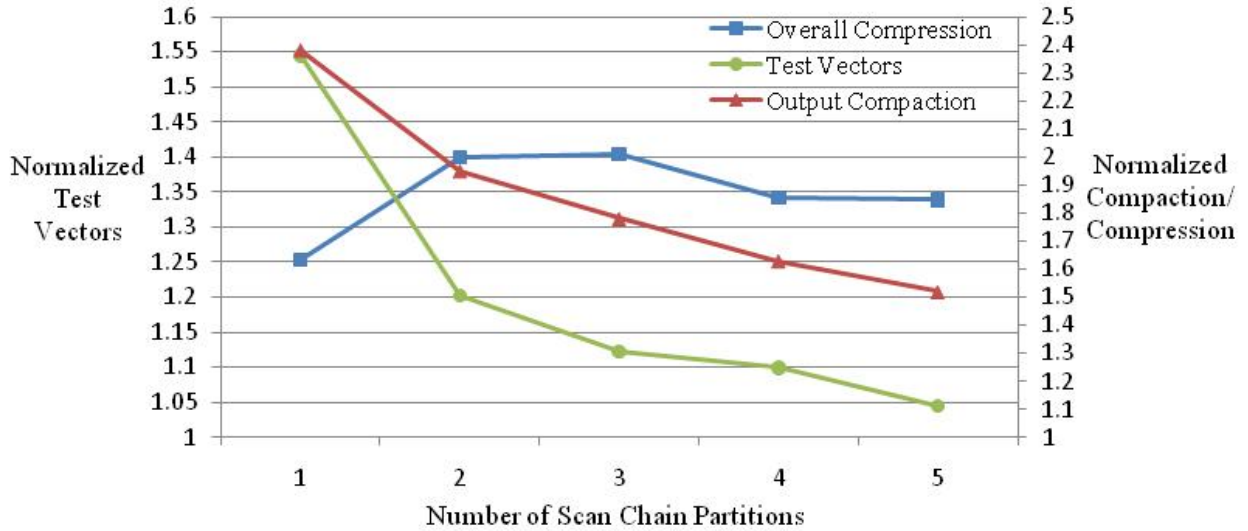
**Table 1:** Experimental Results using Proposed Method for ISCAS'89 Benchmark Circuits with 1%  $X$  Injection

| Benchmark Name | Number of Scan Chains | Fault Coverage Drop (%) | Improvement Ratio for Compaction | Additional Top-Up Vectors for 100% Coverage | Improvement Ratio for Overall Compression |
|----------------|-----------------------|-------------------------|----------------------------------|---------------------------------------------|-------------------------------------------|
| s5378          | 4                     | 1.60                    | 1.9x                             | 9.7%                                        | 1.48x                                     |
| s9234          | 4                     | 1.84                    | 2.0x                             | 10.8%                                       | 1.52x                                     |
| s13207         | 4                     | 2.56                    | 2.0x                             | 22.7%                                       | 1.39x                                     |
| s15850         | 5                     | 1.32                    | 2.1x                             | 14.9%                                       | 1.56x                                     |
| s38417         | 5                     | 0.72                    | 2.0x                             | 10.4%                                       | 1.56x                                     |
| s38584         | 4                     | 0.83                    | 2.1x                             | 20.7%                                       | 1.47x                                     |



**Figure 5:** Test Vector Overhead vs  $X$ -injection Rate for a Fixed Improvement Ratio of 2x for *s15850*

Figure 6 shows how the overall compaction varies as the compaction period (number of clock cycles that circular registers compact before flushing). The x-axis shows the compaction period as a fraction of the scan chain length, i.e., 1 corresponds to the compaction period equal to the scan chain length, 2 corresponds to the compaction period equal to  $\frac{1}{2}$  of the scan chain length, and so forth. So going from left to right on the graph, the aggressiveness of the compaction is being reduced. As expected, the amount of top-up vectors reduces as observability improves. With the longest compaction period, the test set size is 1.55 larger than the original test set size, however, as less aggressive compaction is used, this reduces all the way down to 1.5 in the shortest compaction period. The overall test compression on the far left is the lowest even though compaction is highest because there are so many top-up vectors. Moving from left to right, the compaction goes down, but the number of top-up vectors goes down even faster so that overall test compression improves. This happens up until we reach a period of  $\frac{1}{3}$  of the scan chain length where the overall compression is maximum. After that point, the compaction is dropping faster than the amount of top-up vectors, so the overall compression starts to decline. If the graph were extended all the way to the point where the compaction period was equal to a single clock cycle, then the compaction would go to 1 (i.e., no compaction) and the top-up vectors would also go to 1 (i.e., no additional top-up vectors). So, for every circuit there exists an optimal point which maximize the overall compression.



**Figure 6:** Overall Compaction for Different Circular Register Compaction Periods for *s15850*

## 5. Conclusions

Handling  $X$ 's in the output response is very costly, and the proposed  $X$ -stacking scheme provides an efficient way to reduce the number of  $X$ 's that the output response compaction circuitry needs to handle. It can significantly improve the overall compression by using a very small amount of additional logic inserted in the scan chains to allow the last  $n$  scan cells to be configured into a programmable length circular register. As  $X$ -densities continue to rise in future generations of technology, the propose approach provides a way to reduce the cost of handling them.

## Acknowledgments

This research was supported in part by the National Science Foundation under Grant No. CCF-0916837.

## References

- [Barnhart 01] Barnhart, C., V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, "OPMISR: the Foundation for Compressed ATPG Vectors", *Proc. of International Test Conference*, pp. 748-757, 2001.
- [Chao 05] M. C.-T. Chao, S. Wang, S.T. Chakradhar, and K.-T. Cheng, "Response Shaper: A Novel Technique to Enhance Unknown Tolerance for Output Response Compaction", *Proc. of International Conference on Computer-Aided Design*, pp. 80-87, 2005.
- [Chickermane 04] Chickermane, V., B. Foutz, and B. Keller, "Channel Masking Synthesis for Efficient On-Chip Test Compression", *Proc. of International Test Conference*, pp. 452-461, 2004.
- [David 80] R. David, "Testing by Feedback Shift Register", *IEEE Trans. Comput.*, vol. 29, No. 7, pp. 668-673, 1980.
- [Gizdarski 10] Gizdarski, E., "Constructing Augmented Time Compactors," *Proc. of European Test Symposium*, pp. 151-156, 2010.
- [Lee 91] H.K. Lee, D.S. Ha, "An Efficient Forward Fault Simulation Algorithm Based on the Parallel Pattern Single Fault Propagation," *Proc. of Int. Test Conference*, pp. 946-955, Oct. 1991.
- [Lee 93] H.K. Lee and D.S. Ha, "Atalanta: An efficient ATPG for combinational circuits," *Technical Report 12*, Dep't of Electrical Eng., Virginia Polytechnic Institute and State University, 1993.
- [Mitra 04a] Mitra, S., and K.S. Kim, "X-Compact: An Efficient Response Compaction Scheme", *IEEE Trans. on Computer-Aided Design*, Vol. 23, No. 3, pp. 421-432, Mar. 2004.
- [Mrugalski 09] Mrugalski, G., N. Mukherjee, J. Rajske, D. Czysz, and J. Tyszer, "Highly X-Tolerant Selective Compaction of Test Responses," *Proc. of VLSI Test Symposium*, pp. 245-250, 2009.

- [Patel 03] Patel, J.H., S.S. Lumetta, and S.M. Reddy, "Application of Saluja-Karpovsky Compactors to Test Responses with Many Unknowns", *Proc. of VLSI Test Symposium*, pp. 107-112, 2003.
- [Pomeranz 02] Pomeranz, I., S. Kundu, and S.M. Reddy, "On Output Response Compression in the Presence of Unknown Output Values", *Proc. of Design Automation Conference*, pp. 255-258, 2002.
- [Rajski 06a] Rajski, J., J. Tyszer, G. Mrugalski, W.-T. Cheng, N. Mukherjee, and M. Kassab, "X-Press Compactor for 1000x Reduction of Test Data", *Proc. of International Test Conference*, Paper 18.1, 2006.
- [Rajski 06b] Rajski, W., and J. Rajski, "Modular Compactor of Test Responses", *Proc. of VLSI Test Symposium*, pp. 242-251, 2006.
- [Rajski 08] Rajski, J., J. Tyszer, G. Mrugalski, W.-T. Cheng, N. Mukherjee, and M. Kassab, "X-Press: Two-Stage X-Tolerant Compactor with Programmable Selector," *IEEE Trans. on Computer-Aided Design*, Vol. 27, Issue 1, pp. 147-159, Jan. 2008.
- [Sharma 05] Sharma M. and W.-T. Cheng, "X-Filter: Filtering Unknowns from Compacted Test Responses", *Proc. of International Test Conference*, Paper 42.1, 2005.
- [Tang 06] Tang, Y., H.-J. Wunderlich, P. Engelke, I. Polian, B. Becker, J. Scholöf, F. Hapke, and M. Wittke, "X-Masking During Logic BIST and Its Impact on Defect Coverage", *IEEE Trans. on VLSI*, Vol. 14, No. 2, Feb. 2006.
- [Touba 07] Touba, N.A., "X-Canceling MISR – An X-Tolerant Methodology for Compacting Output Responses with Unknowns Using a MISR", *Proc. of International Test Conference*, paper 6.2, 2007
- [Volkerink 05] Volkerink, E.H., and S. Mitra, "Response Compaction with Any Number of Unknowns Using a New LFSR Architecture", *Proc. of Design Automation Conference*, pp. 117-122, 2005.
- [Wang 06] L.T. Wang, C.-W. Wu, X. Wen, *VLSI Test Principles and Architectures*, Morgan Kaufmann, 2006.
- [Wohl 01] Wohl, P., J.A. Waicukauski, and T.W. Williams, "Design of Compactors for Signature-Analyzers in Built-In Self-Test", *Proc. of International Test Conference*, pp. 54-63, 2001.
- [Wohl 03] Wohl, P., J.A. Waicukauski, S. Patel, and M.B. Amin, "X-Tolerant Compression and Application of Scan-ATPG Patterns in a BIST Architecture", *Proc. of International Test Conference*, pp. 727-736, 2003.
- [Wohl 04] Wohl, P., J.A. Waicukauski, and S. Patel, "Scalable Selector Architecture for X-Tolerant Deterministic BIST", *Proc. of Design Automation Conference*, pp. 934-939, 2004.