# Systematic Scan Reconfiguration

Ahmad A. Al-Yamani
Computer Engineering
KFUPM, Dhahran, Saudi Arabia
aaa@ieee.org

Narendra Devta-Prasanna
Electrical and Computer Eng.
U. of Iowa, Iowa City, IA 52242
ndevtapr@engineering.uiowa.edu

Arun Gunda
LSI Logic Corp.
Milpitas, CA 95035
arun@lsil.com

*Abstract* - **We present a new test data compression technique that achieves 10x to 40x compression ratios without requiring any information from the ATPG tool about the unspecified bits. The technique is applied to both single-stuck as well as transition fault test sets. The technique allows aggressive parallelization of scan chains leading to similar reduction in test time. It also reduces tester pins requirements by similar ratios. The technique is implemented using a hardware overhead of a few gates per scan chain.**

## I. Introduction

The quality of structural testing for digital circuits is a function of the accessibility to the internal nodes of the circuit. The most widely used design for testability (DFT) technique to improve accessibility is scan-path, which is based on serialization of test data [1]. The main advantage of scan is improving the controllability and observability of the circuit under test by having direct access to the states of the flip-flops. Scan-based testing causes some challenges resulting in significant increase in test cost. These challenges are:

**(1) Test time and pin count trade off**: every test pattern needs to be shifted into these shift registers before being applied. For example, a circuit with 128K flip-flops organized into 32 balanced scan chains will have a chain length of 4,000 flip-flops. For every pattern to be applied, 4,000 clock cycles are spent loading that pattern into the scan chains. Increasing the number of scan chains to reduce the loading time causes an increase in another costly parameter, which is the number of tester pins available for loading and unloading the scan chains.

**(2) Test power consumption and shift speed trade off:** Because all flip-flops are clocked while shifting patterns in and out of the scan chains, the power consumption of the circuit is much higher during test than it is during normal operation. Since the circuit is designed to work within the functional power budget, power consumption during shift operations causes major test validity concerns. One of the solutions for this problem is reducing the frequency at which patterns are shifted in and out but that negatively contributes to the previous problem.

Another fundamental problem with test today is the test data volume. The major cause for the problem is accessibility limitations. The problem exists both in scan and sequential test.

Existing solutions in the industry often address some but not all of the above challenges simultaneously. The most popular solution includes several compression techniques used to reduce the data volume and the tester channel requirements. In such techniques, a compressed vector is loaded from the tester into the decompression circuitry, which expands the vector into a test pattern in the scan chains. The test response is also compressed into a smaller vector using the output compression circuitry. To name a few, [2], [3], [4] and [5] discuss such compression techniques.

Illinois Scan Architecture (ISA) is another class of solutions that was introduced in [6] to reduce data volume and test application time by splitting the scan chain into multiple segments and broadcasting the data to all of them as long as the segments data are compatible.

Very recently [7], we presented a new architecture and circuitry for significantly reducing test data volume, test application time, test power consumption and tester channel requirements. The new architecture, called segmented addressable scan (SAS), is based on ISA but it enables much more aggressive segmentation of the scan chains by enabling many different compatibility configurations among multiple segments.

This paper presents Systematic Scan Reconfiguration (SSR). SSR is a compression solution that does not require any information about don't care bits. Yet, it achieves 10x to 40x reduction in test data volume, test application time, and tester channel requirements. With the same minimal hardware overhead as SAS, SSR achieves this major cost reduction through modifying the ATPG process instead of utilizing the don't care bits.

Section 2 of this paper briefly presents segmented addressable scan. Section 3 explains systematic scan reconfiguration. Section 4 gives appropriate credit to previous work. Sections 5 shows initial experimental results and Sec. 6 concludes the paper.

 

## II. Segmented Addressable Scan

This section is a review for segmented addressable scan (SAS) architecture, which incorporates some of the basic concepts from Illinois scan [6] and from scan segment decoding [8] [9]. Combining these concepts with an efficient design of a multiple-hot decoder operating based on positional cube encoding [10], SAS addresses all challenges of digital core testing raised in the previous sections. The basic blocks of the SAS architecture are shown in Figure 1.

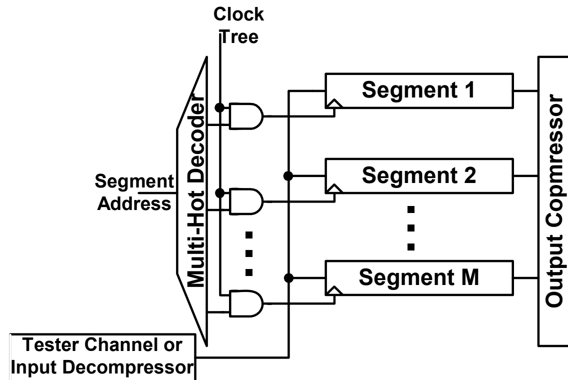


Figure 1 Segmented Addressable Scan (SAS)

A given address is loaded into the multiple-hot decoder (MHD) to refer to a single or multiple segments. A regular decoder scheme like the ones in [8] and [9] would take advantage of the compatibility for data volume reduction only. Because of the MHD used in the SAS architecture, test time can also be optimized based on this compatibility since the compatible classes will be loaded in parallel.

For regular one-hot decoders, the input to the decoder is an address of the selected output. For the MHD, the address can include don't care bits (d's) allowing multiple outputs to be activated.

As explained in [7], positional cube encoding scheme results in an implementation for the multiple-hot decoder that requires the same hardware as a regular one-hot address decoder. In general, if we have $S$ segments, we need $S$ AND gates each with $2\lceil \log_2 S \rceil$ inputs for the multiple-hot decoder. For clock gating, we need $S$ 2-input AND gates.

As an example for SAS hardware overhead, the number of transistors needed for the additional hardware for 128 segments is less than 3000 transistors i.e., less than 1000 gates.

Using SAS, we reported an order of magnitude or more of reduction in test data volume, test application time, tester channel requirements and test power consumption.

## III. Systematic Scan Reconfiguration

As it is obvious from the previous section, we need the information about the don't care bits to generate the compatibility classes needed for SAS decoder address generation. We had two issues with this requirement: (1) Some ATPG vendors don't provide don't care bits information as they consider them confidential. (2) A fault can be detected by multiple patterns. With the ATPG unaware of the SAS architecture, the selection of which patterns to generate by the ATPG tool will not be driven by higher compatibility but rather by ease of generation.

As a result of the above two issues, we were not only forced to come up with an algorithm that doesn't require don't care bits but we were also convinced that we could drive the ATPG tool to generate more highly compatible patterns that would require the fewer addresses or configurations with SAS.

The SSR algorithm is based on the same SAS hardware presented in Sec. 2. It works by configuring the scan chains in the circuit such that they appear to be tied together to the ATPG tool with multiple configurations. The selection of which segments to tie together is done such that the number of addresses required to be loaded into the multiple-hot decoder is minimized. Basically, an address corresponds to a subset of the segments. For example, for a 2-to-4 multiple-hot decoder, the address 00 corresponds to segment 0, and so on. Also, the address 0d (d = don't care) corresponds to segments 0 and 1. Finally, the address *dd* corresponds to all 4 segments. Without the SAS architecture, we could choose a multiplicity of configurations and generate patterns with them tied together. However, this would require many multiplexers at the inputs and outputs of the scan segments to reconfigure them. It would also either cause problems with engineering changes or require these multiplexers to be highly reconfigurable which leads to high hardware overhead. The high flexibility and simplicity of the SAS architecture allows for a very large number of configurations ($3^{\lceil \log_2 S \rceil}$, where $S$ is the number of scan segments) with very simple hardware that doesn't need to be changed with engineering changes.

Physically, all segments in the architecture are tied together. The decoder controls which segments to load together by activating a subset of the clocks to these segments based on the address loaded.

Our SSR algorithm selects a set of configurations for combining scan segments together and then fakes to the ATPG tool that these segments are tied together to generate compatible patterns for them. It continues with such configurations until complete fault coverage is achieved.

```
1. Classify all detectable faults as undetected
2. Start with the configuration dd…d
3. While (there are undetected faults)
4.      Generate ATPG patterns
5.      If the address care bit(s) are not the least significant
6.         Move address care bits to lower significance
7.      Else
8.         Increase the number of care bits in address
9.         Make the care bits the most significant
10.     Endif
11.Endwhile
12.End
```

Algorithm 1 Systematic Scan Reconfiguration Algorithm.

The algorithm is best explained by an example. Take a SAS architecture with 8 segments (the addresses for the individual segments are 000 through 111). First, we tie all segments together and we call this *Category* 0. There is only one configuration in this category, which corresponds to the address *ddd*. We run the ATPG tool with this configuration to detect as many faults as it can. Notice that during test application, all we need to do is load the address *ddd* in the decoder and then start loading the patterns in category 0. Also note that every pattern generated with this configuration is $1/8^{th}$ (generally $1/S$) of the size of the regular pattern (assuming segments are balanced). Most of the time, there will be undetected faults with this configuration. So, we switch to category 1. In category 1, only one of the address bits is specified and the remaining bits are all *d*'s. Notice that there are 3 possible configurations (generally $\lceil \log_2 S \rceil$ configurations) where only one bit is specified. We start with the configuration *cdd*, where *c* stands for a care bit. The care bit will take the values 0 and 1. This means that we use the addresses 0*dd* and 1*dd*. These two addresses correspond to tying segments 0, 1, 2, and 3 together and segments 4, 5, 6, and 7 together. We invoke the ATPG tool to generate patterns and load only the faults that were not detected with category 0 patterns. The next configuration within category 1 is *dcd*, which corresponds to segments 0, 1, 4, and 5 tied together and segments 2, 3, 6, and 7 tied together. We again invoke the ATPG tool with the undetected faults. After the last configuration in category 1, we go to category 2 where we have two care bits instead of one. The first configuration will be *ccd*, which corresponds to tying the segments in four groups (0 with 1, 2 with 3, 4 with 5, and 6 with 7). We continue with these categories and configurations until all detectable faults are detected. The general algorithm for SSR is shown below in Algorithm 1. Experiments show that we normally don't need to go beyond category 1.

By going through the example above the reader will feel that the ATPG runtime will be very long and that's true. However, there are multiple solutions that could be used for this problem. Here are some of them:

(1) The first solution is not to try all configurations but to cut the process in the middle and jump to the configuration *ccc*. This configuration will detect all remaining detectable faults at any step.

(2) Another solution is not to start with the configuration *dd…d* but rather with *cd…d* or *ccd…d*. This will cut the runtime significantly because the first configuration is the hardest for the ATPG tool.

(3) A third solution is to reduce the effort level with the first few configurations to the minimum such that the ATPG tool starts with the easily detectable faults.

Not surprisingly, the price for all of the above solutions is reduction in the compression ratio. It's well-known to the reader by now that the SSR ATPG runtime is a one time cost while the SSR compression ratio is a recurrent saving.

Proper credit should be given to [11] in which the idea of using multiple configurations of Illinois Scan was presented. SSR has the following distinguishing features: (1) The architecture in [11] is based on mapping logic, multiplexers-based added hardware that combines multiple subsets together. The hardware is designed based on reducing the number of compatibilities required because more compatibilities will require more multiplexers and more scan inputs. In addition to the processing time required for these compatibilities, such information about which faults are detectable with which is only available to ATPG vendors. Our SSR hardware does not require any such information and does not need such extensive processing time. Furthermore, it allows $3^{\lceil \log_2 S \rceil}$ different configurations without any additional overhead. For example, an SSR configuration of 256 segments will automatically allow more than 6500 configurations. For such flexibility, the technique in [11] will require 256 6500-input multiplexers. SSR will require 256 8-input AND gates and 256 2-input AND gates. (2) For the same example above, the number of tester pins required for SSR is 17. For their technique to allow similar flexibility, the number of tester channels is more than 6500 tester pins. It can be argued that not all such configurations are needed to achieve an

acceptable compression ratio. However, these configurations can be used to reduce runtime too (3) Any engineering change orders may alter the compatibilities based on which the hardware in [11] was synthesized. With SSR, all we need is to select a different set of compatibilities. No hardware changes are needed. (4) SSR inherently offers power reductions by selective activation. (5) The technique in [11] is heavily based on broadcasting mode, which as will be shown in the results section is very time-consuming for the ATPG tool and it gets worse with more aggressive parallelization. Their results show up to 50x increase in ATPG runtime. As shown in the experimental results, we found that it is very helpful in terms of runtime to use configurations with fewer chains in broadcast mode. This is something that SSR automatically allows.

## IV. Related work

Illinois Scan Architecture (ISA) was introduced to reduce data volume and test application time [6]. Since a majority of the bits in ATPG patterns are don't care bits, there are chances that these segments will have compatible vectors. In this case, all segments of a given chain are configured in broadcast mode to read the same vector. In case if the segments within a given scan chain are incompatible, the test vector needs to be loaded serially. Several enhancements to the Illinois scan architecture have been proposed and discussed in the literature for multiple reasons.

Lee et. al. presented a broadcasting scheme where ATPG patterns are broadcasted to multiple scan chains within a core or across multiple cores [12]. This scheme seems to have been concurrently developed with ISA.

[13] introduced a token scan architecture to gate the clock to different scan segments while taking advantage of the regularity and periodicity of scan chains. Another scheme for selective triggering of scan segments was proposed in [14].

A novel scheme was presented in [15] to reduce test power consumption by freezing scan segments that don't have care bits in the next test stimulus. By only loading the segments that have care bits, data volume, application time, and test power consumption are all reduced at once.

[16] presented a scheme for resolving conflicts between care bits in different segments of an ISA architecture to improve the compression ratio.

The X-pand scheme presented in [17] also presented a mapping scheme for an ISA based compression. The paper discussed compression using don't care bits and using ATPG configurations. X-pand, which was a major first step in the right direction for compression, differs from SSR in two major ways: (1) it doesn't offer any power reduction. (2) it's a combinational compactor, so shadow registers cannot be used for further reduction in tester channel requirements.

A new scan architecture was proposed in [18] to order the scan cells and connect them based on their functional interaction.

A circular scan scheme was presented in [8] to reduce test data volume. The basic concept is to use a decoder to address different scan chains at different times. This increases the number of possible scan chains ($2^{N-1}$ for an $N$-input decoder). Also, the output of each scan chain is reconnected to its input. This enables reusing the contents of the response captured in the chain as a new test stimulus if they are compatible.

The previous schemes are either limited in how much they can benefit from compatibility between some of the segments or don't address the issue of power consumption during scan or both.

Another attempt for using decoder-based segmentation is available in [9]. In this scheme the authors control the clocks to various segments through a regular decoder. The main advantage of the scheme is power reduction during scan and capture. The solution doesn't address data volume, or test application time.

SAS hardware enhances the benefit from all scan segmentation schemes by avoiding the limitation of having to have all segments compatible to benefit from the segmentation. In other words, any combination of segments can be compatible to lead to reduction in the test stimuli loaded. This is done with minimal overhead due to the multiple-hot decoder. The scheme simultaneously addresses data volume, test time, power, and tester channel requirement.

Recently, a scan chain segmentation technique was presented in [19]. The technique is a BIST solution that selectively inserts inversions at some locations in the scan path based on the ATPG patterns to minimize the number of weights required for weighted random patterns.

The technique in [20] is a recent attempt for test cost reduction through scan reconfiguration. The technique is based on finding the matches between the test response of pattern $n$ and the bits of pattern $n$+1. This technique requires high routing overhead just like random access scan presented in [21] and enhanced in [22]. Although the titles are close to each other, these two recent solutions are in essence very different from SSR.

## V. Experiments and Results

SSR experiments were performed on the circuits in TABLE I, both of which are 180 nm designs.

TABLE I
Circuit Characteristics.

| | flip-flops | Gate count | Clock domains | Test Patterns |
|---|---|---|---|---|
| Ckt1 | 29K | 350K | 10 | 1.5K |
| Ckt2 | 35.5K | 450K | 26 | 3.4K |

It has been evident to us from our experiments in [7] as well as these experiments that SSR achieves better results with bigger designs.

TABLE II shows the compression ratio achieved by SSR for stuck patterns using different segmentations.

TABLE II
Stuck-at Tests Data Volume Compression.

| **Ckt1** | | | |
|---|---|---|---|
| Total data volume | | 40 Mb | Comp Ratio |
| SSR data volume | 32 Segments | 3.3 Mb | 12x |
| | 64 Segments | 2.4 Mb | 16x |
| | 128 Segments | 2.0 Mb | 19x |
| | 256 Segments | 1.9 Mb | 21x |
| **Ckt2** | | | |
| Total data volume | | 120 Mb | Comp Ratio |
| SSR data volume | 32 Segments | 7.5 Mb | 16x |
| | 64 Segments | 5.8 Mb | 20x |
| | 128 Segments | 4.8 Mb | 25x |
| | 256 Segments | 3.7 Mb | 32x |

Similar data for transition fault patterns is shown in TABLE III. The results are slightly better. It's obvious that the compression ratio increases as the number of segments increases for both single-stuck and transition patterns. The price for increasing the segments is the runtime, which we will discuss.

Similar reduction ratios are achieved for test time. Furthermore, the fact that the cost for additional scan chains is minimal (just a few gates per chain), promises for significant reduction in test time. With only 21 scan input pins, our technique can support 1,024 scan chains. such parallelization considers parallel loading into the decoder without any shadow registers. Using shadow registers allows for more parallelization.

To give an idea about how much fault coverage can be achieved while tying multiple segments together, we show the fault coverage progressive improvement of SSR together with the normal fault coverage achieved with basic ATPG. Figure 2 shows the fault coverage vs. the categories and configurations used for Ckt1 with 32 segments (the other segmentations behaved similarly). The figure delivers 2 significant messages: (1) The first category (all segments tied together) achieved more than 99% of the achievable coverage (achievable = 97.3, achieved = 96.3). (2) We don't need more than the first two categories to achieve the achievable coverage. In fact, we even slightly exceeded it. Ckt2 exhibited a similar behavior.

TABLE III
Transition Data Volume Compression.

| **Ckt1** | | | |
|---|---|---|---|
| Total data volume | | 98 Mb | Comp Ratio |
| SSR data volume | 32 Segments | 7.7 Mb | 12x |
| | 64 Segments | 5.3 Mb | 18x |
| | 128 Segments | 4.5 Mb | 22x |
| | 256 Segments | 3.6 Mb | 27x |
| **Ckt2** | | | |
| Total data volume | | 300 Mb | Comp Ratio |
| SSR data volume | 32 Segments | 21.7Mb | 14x |
| | 64 Segments | 14.1Mb | 21x |
| | 128 Segments | 11.8Mb | 25x |
| | 256 Segments | 7.7Mb | 39x |

Figure 3 shows similar results to those in Figure 2 but for transition test instead of the single-stuck test. The observations for transition patterns are consistent with those for single-stuck patterns.

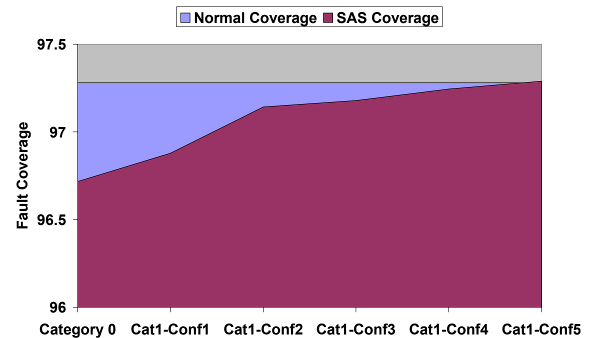


Figure 2 Progressive SSR coverage with 2 categories of single-stuck patterns.

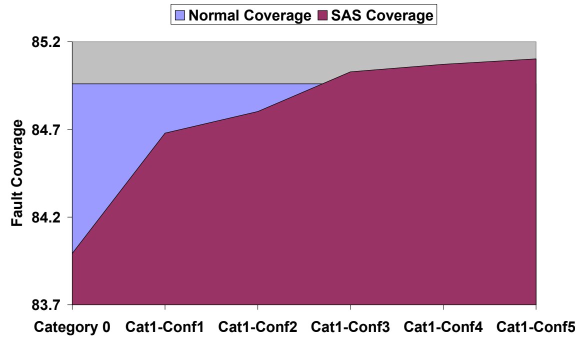


Figure 3 Progressive SSR coverage with 2 categories of transition patterns.

## VI. Conclusions

"Necessity is the mother of invention". We could not implement our previous test data compression solution due to the unavailability of the unspecified bits information. This paper presents our solution to this problem. The solution is a compression technique that satisfies the test data and test time reduction requirements of all of our designs without requiring any information about the unspecified bits. It also reduces tester pin requirements while requiring minimal hardware overhead.

## Acknowledgment

The authors acknowledge the support of LSI Logic Corporation, King Fahd University of Petroleum and Minerals, and the University of Iowa.

## References

[1] E.J. McCluskey, *Logic Design Principles with Emphasis on Testable Semicustom Circuits*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1986.

[2] B. Koenemann "LFSR-Coded Test Patterns for Scan Designs," *European Test Conference (ETC'91)*, pp. 237-242, 1991.

[3] E.J. McCluskey, D. Burek, B. Koenemann, S. Mitra, J. Patel, J. Rajski and J. Waicukauski, "Test Data Compression," *Design & Test of Computers*, Vol. 20, No. 2, pp. 76 – 87, March-April 2003.

[4] A. Al-Yamani and E.J. McCluskey, "Seed Encoding for LFSRs and Cellular Automata," *40th Design Automation Conference (DAC'03)*, June 2003.

[5] J. Rajski, J. Tyszer, M. Kassab and N. Mukherjee, "Embedded Deterministic Test," *IEEE Transactions on Computer-Aided Design (TCAD)*, Vol. 23 , No. 5 , pp. 776-792, May 2004.

[6] I. Hamzaoglu and J. Patel, "Reducing Test Application Time for Full Scan Embedded Cores" *IEEE International Symposium on Fault Tolerant Computing (FTC'99),* pp. 260-267, 1999.

[7] A. Al-Yamani, Erik Chmelar, and Mikhail Grinchuk, "Segmented Addressable Scan Architecture," *VLSI Test Symposium (VTS'05)*, May 2005.

[8] A. Arslan and A. Orailoglu, "CircularScan: A Scan Architecture for Test Cost Reduction," *Design, Automation and Test in Europe Conference and Exhibition (DATE'04),* Vol. 2, pp. 1290-1295, Feb. 2004.

[9] P. Rosinger, B.M. Al-Hashimi, and N. Nicolici, "Scan Architecture With Mutually Exclusive Scan Segment Activation for Shift- and Capture-Power Reduction," *IEEE Transactions on Computer-Aided Design (TCAD)*, Vol. 23 , No. 7 , pp. 1142-1153, July 2004

[10] G. De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, 1994.

[11] S. Samaranayake, E. Gizdarski, N. Sitchinava, F. Neuveux, R. Kapur and T. Williams, "A Reconfigurable Shared Scan-in Architecture" *VLSI Test Symposium (VTS'03),* Apr. 2003.

[12] K-J. Lee, J-J. Chen and C-H. Huang, "Broadcasting Test Patterns to Multiple Circuits," *IEEE Transactions on Computer-Aided Design (TCAD),* Vol. 18, No. 12, pp. 1793-1802, Dec. 1999.

[13] T-C. Huang and K-J. Lee, "A Token Scan Architecture for Low Power Testing," *International Test Conference (ITC'01),* pp. 660-669, Oct. 2001.

[14] S. Sharifi, M. Hosseinabadi, P. Riahi and Z. Navabi, "Reducing Test Power, Time and Data Volume in SoC Testing Using Selective Trigger Scan Architecture," *International Symposium on Defect and Fault Tolerance (DFT'03),* 2003.

[15] O. Sinanoglu and A. Orailoglu, "A Novel Scan Architecture for Power-Efficient, Rapid Test," *International Conference on Computer-Aided Design (ICCAD'02)*, pp. 299-303, Nov. 2002.

[16] N. Oh, R. Kapur, T. Williams, and J. Sproch, "Test Pattern Compression Using Prelude Vectors In Fan-out Scan Chain with Feedback Architecture," *Design, Automation, and Test in Europe Conference (DATE'03),* pp. 110-115, 2003.

[17] S. Mitra, and K. Kim, "XMAX: X-Tolerant Architecture for MAXimal Test Compression," *International Conference on Computer Design (ICCD'03),* pp. 326-330, Oct. 2003.

[18] D. Xiang, J. Sun, M. Chen and S Gu, "Cost-Effective Scan Architecture and a Test Application Scheme for Scan Testing with Non-scan Test Power and Test Application Cost" *US Patent Application 20040153978*, Aug. 2004.

[19] L. Lay, J. Patel, T. Rinderknecht, and W-T. Cheng, "Logic BIST with Scan Chain Segmentation," *International Test Conference (ITC'04)*, pp. 57-66, Nov. 2004.

[20] B. Arslan and A. Orailoglu, "Test Cost Reduction Through a Reconfigurable Scan Architecture," *International Test Conference (ITC'04)*, pp. 945-952, Nov. 2004.

[21] H. Ando, "Testing VLSI with Random Access Scan," *IEEE Computer Society Conference (COMPCON'80)*, pp. 50-52, Feb, 1980.

[22] D. H. Baik, K. K. Saluja, and S. Kajihara, " Random Access Scan: A solution to test power, test data volume, and test time," *International Conference on VLSI Design (VLSID'04)*, pp. 883-888, Jan. 2004.