Enhancement of the Illinois Scan Architecture for Use with Multiple Scan Inputs *

Mihir A. Shah† and Janak H. Patel
Center for Reliable & High-Performance Computing
1308 W Main Street
University of Illinois, Urbana, IL 61801
{mashah, jhpatel}@uiuc.edu

Abstract

Testing cost is becoming increasingly important as System-on-Chip circuits continue to become more complex. In this paper, we address the issue of reducing test cost by shortening test application time and reducing the volume of data that needs to be stored on a tester. The number of scan channels on a tester and/or the number of pins on an SOC are limited. We propose a method to enhance the Illinois Scan Architecture for use with a small number of scan-in pins. Pin reduction is achieved by connecting a single pin to several scan chains together depending on their compatibility relations. With the use of an incompatibility graph and graph coloring algorithm, the number of pins needed is minimized.

1. Introduction

Many techniques have been suggested to lower the overall test cost by reducing test data volume and tester pins. One such method is using built-in-self-test (BIST), which provides on-chip test pattern generation and output comparison. In order to achieve sufficient fault coverage, using BIST alone is normally insufficient, and must be combined with standard scan approaches. Several proposed methods [1, 2, 3] have shown that using a hybrid of test pattern generation (ATPG) and BIST are effective in reducing tester data volume while still maintaining high fault coverage. Implementing BIST, however, introduces complications. There is an increase in area and routing when adding the BIST controller. Logic and signal changes/additions are needed to accommodate the additional hardware. All these factors greatly increase the complexity of the design, which leads to an increase in the development time of the chip [4].

An alternative to BIST is the Illinois Scan Architecture (ILS) [5]. Originally proposed as a technique for embedded cores, ILS can be used on standalone chips as well. Under the ILS methodology, there are two modes, broadcast and serial. In broadcast mode, the entire scan chain of the circuit is broken up into smaller scan chains, which all get scanned in with identical data supplied on a single scan-in pin. In serial mode, conventional scan patterns are created to test any faults not covered by broadcast mode. A case study of the Illinois Scan Architecture [4] on an industrial circuit showed that because of the parallelism involved in broadcast mode, there is a significant reduction in test data volume and application time, while introducing less of an area and routing penalty than other BIST implementations.

Other methods have been proposed to improve the original ILS methodology. An incremental algorithm [6] for ILS test generation was proposed which created an efficient way for finding the most optimal ILS configuration. A reconfigurable technique used with ILS was shown to reduce test data volume by decreasing the number of patterns applied in serial mode [7]. In this paper, we propose a new technique involving ILS to further reduce test data volume by further eliminating the need for serial patterns, accomplished by the intelligent use of multiple groups of scan chains. Previous studies [4, 8] have demonstrated that using multiple groups instead of the one group used in traditional ILS broadcast mode can be beneficial, but the groups were predetermined without using any compatibility analysis. Recently, a reconfigurable scan-chain architecture [9] was proposed which used compatibility analysis of scan chains for the assignment of groups. It was assumed that the order of scan cells was not predetermined, and the algorithm used a fixed number of groups. Our proposal assumes a pre-determined order of scan cells, and determines group configurations without constraints on the number of groups.

2. Introduction to Illinois Scan Architecture

The Illinois Scan Architecture (ILS) was first introduced in [5]. An overview of the architecture is provided here. The architecture is shown in Figure 1, which represents the

*This research was supported by the Semiconductor Research Corporation under contract SRC 99-TJ-717.

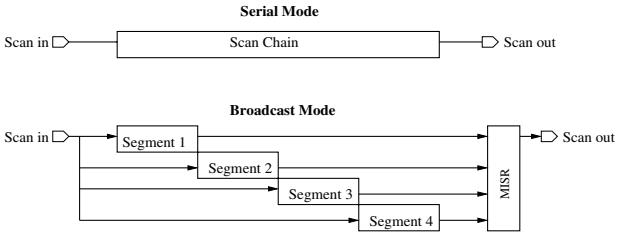

Figure 1. Two Modes of Illinois Scan Architecture

two modes of operation that it consists of.

The top part of the figure shows a regular scan chain, which is known as Serial Mode. The bottom part of the figure shows the scan chain broken up into segments, called the Broadcast Mode. Here the scan-in pin that originally went into the entire scan chain now feeds into each of the scan chain segments. Thus, each segment will be inputted the same data in parallel. Note this shared scan-in idea has been reported in previous work [10]; however, this was limited to testing several independent circuits in parallel. As shown by the figure, the outputs of the scan chains are compressed into a multiple input signature register (MISR). This is similar to the output response calculation found in BIST implementations. Like BIST implementations, in order to prevent corrupting the MISR signature, certain design rules must be followed, such as avoiding unknown states, internal bus conflicts, among others. Alternately, combinational compactors [11, 12] that tolerate X values, could be used in place of MISR.

The additional hardware required for this architecture includes multiplexers for each scan chain segment, which are needed to switch between the two modes of operation. In addition, a MISR or combinational compactor is needed. It was shown that even with pessimistic assumptions, the addition of this hardware and the additional routing needed only caused a minor increase in the design area of a chip, less than what is needed for previously reported BIST implementations [4].

Testing is accomplished by first using an ATPG to generate vectors in broadcast mode. Since broadcast mode imposes constraints on test patterns, many faults become untestable. To cover these untestable faults, an ATPG is used to generate test vectors under serial mode.

2.1. ILS with Multiple Groups

In a large practical design, it is common to use multiple groups of chains, each with their own scan-in pin. The ILS architecture can then be modified for use with multiple groups of scan chains. An example with two groups is shown in Figure 2. The advantage of using multiple groups is that the ATPG is less constrained than using just one group, reducing the number of untestable faults in this mode. This will in turn reduce the amount of vectors needed in serial mode, which is beneficial since these vectors are far more expensive than those in broadcast mode. However, having multiple groups of scan chains may also be costly, as test data needs to be stored for each group. It is imperative that a proper grouping of chains is used in order to achieve test data volume reduction. Therefore, we present a technique to find an optimal grouping of scan chains by performing compatibility analysis on the scan chains. By using this optimal grouping of scan chains, it is shown that the need for serial vectors is eliminated altogether, which further reduces both the test data volume and test application time. The procedure to optimally group also minimizes the need for the scan-in pins.

3. Compatibility Analysis

In order to perform compatibility analysis of scan chains, a post-processing procedure is used. First test patterns are generated for broadcast mode using a single group (scan-in pin). Then, for all remaining undetected faults , U, a *nonfilling* ATPG is used to produce a partially specified test set, S, of serial mode patterns. It is from the analysis of these vectors in S that the compatibility of scan chains is determined. Before this analysis procedure is described, it is first necessary to provide some definitions related to compatibility, as shown in the following subsection.

3.1. Compatibility Definitions

1. Two scan cells are said to be *compatible* if and only if no fault becomes untestable as a result of tying the two cells to a single input [13].

2. Two scan chains are said to be *compatible* if and only if every pair of scan-cells that receive the same logic value are compatible [5].

Since the exact analysis for determining all pairwise compatibilities is computationally complex, we resort to a fast analysis which obtains a subset of all compatibilities. This analysis is based on the partially specified *complete* test set S. This procedure was used as a first step in the more thorough procedures of [13] and [14]. A complete test set detects *all* detectable faults. If the test set is incomplete, the procedure given here is still applicable, but can

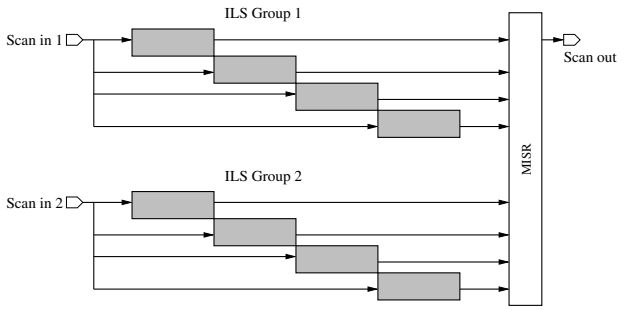

Figure 2. ILS with two groups

guarantee the fault coverage only up to what the incomplete test gives.

In our analysis, a "folding" process is used, similar to the one described in [15]. By folding a given test set on an ILS organization, one can determine if two scan chains are compatible by simply noting an absence of value conflicts. Any two chains not found to be compatible by this procedure are termed *potentially incompatible*. Since the test set based compatibility analysis does not find *all* compatibilities, any incompatible pairs found can only be said to be potentially incompatible. A further analysis, such as in [14], may determine that potentially incompatible pairs are indeed compatible. In other words, our procedure finds a super-set of actual incompatibilities. It is sufficient (but not necessary) to remove these incompatibilities to achieve complete fault coverage. This is the basis of our procedure for altering the ILS structure such that all potential incompatibilities are removed. In subsequent sections, we will drop the adjective "potential" for incompatibilities.

3.2. Compatibility Analysis Procedure

The objective of this procedure is to form the minimal amount of groups so that all scan chains within any group are compatible, as this will result in lowering of test data volume that needs to be stored. To accomplish this, first an incompatibility graph, with all the scan chains as nodes, needs to be created from the serial patterns found in S. It is essential not to have any randomly-filled inputs as this will lead to extraneous incompatibilities between scan chains. The process is simple: for each serial pattern generated, we 'fold over' the pattern into the scan chains used in broadcast mode and for every bit position, determine if there are conflicting values. When a conflict arises, we note which chains are conflicting, and add an edge between the two scan chains, represented as nodes in the incompatibility graph.

After the incompatibility graph has been created, a graph coloring algorithm is applied, which will assign a specific color to every node such that no two nodes connected by an edge will have the same color. Here, assigning colors to nodes is equivalent to assigning a group number, or scan-in pin, to a scan chain. The graph coloring algorithm will attempt to assign the minimal number of colors (groups) while following the restriction that two adjacent nodes (two incompatible scan chains) cannot be of the same color (group number).

For the incompatibility graphs generated in our procedure, a graph coloring algorithm based upon the DSATUR algorithm [16] was utilized. The algorithm first attempts to find the maximum clique size in the graph, which corresponds to finding the maximum number of nodes that are all connected to each other. For example, in the graph in Figure 3, the maximum clique is of size 3, since SC1, SC2, and SC3 are all interconnected.

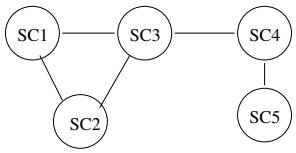

Figure 3. Graph with Maximum Clique of Size 3 (SC1,SC2,SC3)

This maximum clique number then represents the lower bound of the number of colors needed, as every node in that clique needs to be a different color, since they are all interconnected. Those nodes are then colored first, as suggested in [17].

Then a heuristic method is applied for the rest of the graph. A simple explanation of the algorithm is provided here. For more details on the exact algorithm, see [16, 18, 19]. The algorithm works by looking at node in the graph with the highest *saturation degree* first, which is defined as the number of different colors found for all adjacent nodes. If more than one node shares this highest saturation degree, then the algorithm attempts to find the node with the highest unlabeled degree, defined as the largest number of uncolored adjacent nodes. If there is still more than one node left, then one is chosen randomly. After a node is chosen, it is assigned the lowest feasible color, depending on the color of its adjacent nodes. This will result in a configuration of groups for which there are no incompatibilities, which will now be referred to as *groups mode*.

4. Test Generation Procedure

By using groups mode, there are no incompatibilities between the scan chains within a group, which will allow all detectable faults for the given circuit to be tested. There are two options for test generation that can be used in order to take advantage of this fact. The first is using groups mode as a replacement for serial mode in the traditional ILS procedure, in which case both broadcast and groups modes will be used. The second option would be to use groups mode by itself to detect all faults. This will eliminate the multiplexers needed for the traditional ILS approach, and also reduce routing. In addition, the use of only a single mode reduces the time for test generation compared to the dual modes used in the traditional ILS approach. However, since broadcast mode is not used, there will be less data volume reduction than using the first option.

Both options will be discussed in the following subsections; we will refer to these options as *dual mode* and *single mode*, respectively.

4.1. Dual Mode: Broadcast and Groups Modes

The test generation procedure for dual mode is shown in Figure 4. Steps 1-3 were previously described for compatibility analysis: generate a compact test set B under broadcast mode, and generate an partially specified compact test

1. Regenerate the netlist for broadcast mode. Generate a test set B under Broadcast Mode. Perform static compaction on B. Identify the set of faults, U, that are undetectable.
2. Regenerate the netlist for serial mode. Using a *non-filling* ATPG, generate a partially specified test set S tagerting only the faults in U. Perform static compaction on S to remove redundant vectors.
3. Perform compatability analysis for all patterns in S to find a group configuration.
4. Regenerate the netlist for groups mode. Generate a test set G under Groups Mode targeting only the faults in U. Perform static compaction on G.
5. Fault simulate test set G for all faults, and remove the detected faults from the complete fault list.
6. Regenerate the netlist for broadcast mode. Perform static compaction on test set B using the remaining faults from the previous step or regenerate B under Broadcast Mode using only these faults.
7. Output the final test set $T = B \cup G$.

Figure 4. Test Procedure for Using Broadcast and Groups Mode

set S and perform compatibility analysis. Test generation for groups mode is then performed for only the undetectable faults in broadcast mode. Since groups mode will test more faults than what it is needed for, reverse-order fault simulation (RFS) for groups mode is then performed and all detected faults are removed from the complete fault list. Test patterns for broadcast mode are then re-generated, or compacted, using this new fault list.

4.2. Single Mode: Groups mode

The test procedure for using a single mode is the same as in steps 1 through 4 in Figure 4, with the exception that in step 4, we perform test generation for *all* faults, not just the undetectable faults in broadcast mode. After this step, the procedure is complete, and thus the test generation time for single mode is much less than that for dual mode.

5. Experimental Results

Experiments using this test procedure were carried out on four different ISCAS89 benchmark circuits [20]. Test pattern generation with random-fill was performed by using ATOM [21], while generation without random-fill was accomplished by using a modified version of ATOM. Static compaction for fully specified patterns was performed using a double detection and reverse order fault simulation technique [22], while compaction for partially specified patterns was performed using a simpler forward and reverse order single detection scheme [23]. All tests were run an AMD 900 MHz with 256 MB of memory. The longest overall run time was approximately 35 min for ten different configurations of circuit s38417. Test generation time will be longer than the traditional ILS procedure because of a few reasons. First, smaller chain length configurations are used, which causes the ATPG to run longer as there are more redundancies in the circuit. Second, extra time is needed for compatibility analysis and the extra serial pattern ATPG run required for this analysis. In addition, compaction for partially specified vectors takes more time than compaction for fully specified vectors.

Table 1 shows experimental results for various configurations of the circuit s38584.1. The columns represent the ILS configuration, the number of scan chains, the number of broadcast mode patterns generated before reverse-order fault simulation (RFS), the additional undetectable faults caused by this ILS configuration, the number of vectors created by the nonfilling ATPG to cover these undetectable faults, and the number of groups found after compatibility analysis. An ILS configuration that divides the scan chain into multiple chains each of length k, is labeled as an ILS-k configuration.

Table 1. Results for Multiple Group ILS for s38584.1

Config	Num Chains	Pre-RFS Broadcast Vectors	Addnl Red Faults	Num Topoff Vectors	Num Groups
ILS-200	8	863	293	195	5
ILS-128	12	902	503	310	5
ILS-100	15	846	576	374	7
ILS-64	23	726	1513	914	6
ILS-50	29	770	1017	650	5
ILS-32	45	655	2239	1311	6
ILS-25	58	606	2170	1316	6
ILS-20	72	551	2681	1631	7
ILS-16	90	458	3883	2267	7
ILS-12	119	341	4990	2886	8

From this table, it is evident that as the length of the scan chain (column 1) becomes shorter, and the number of internal chains (column 2) increases, more faults become undetectable (column 4). This causes the number of broadcast vectors (column 3) to decrease, and also increases the number of 'top-off' vectors (column 5) needed to cover these undetectable faults. In general, as there are more top-off vectors and larger numbers of scan chains, this will cause more incompatibilities between the scan chains. This results in more groups (external scan-pins) being formed (column 6).

Table 2 shows the experimental results for test data volume needed for various configurations of the four tested cir-

cuits using both dual mode and single mode. Recall that an ILS-k configuration has k cells in each internal chain except one. The columns in this table represent the circuit being tested or which mode is being applied to the circuit, the ILS configuration, the number of groups(external scan-in pins) used for groups mode, the number of groups mode patterns required, the amount of test data that needs to be stored for these patterns, the number of scan chain segments in broadcast mode, the final number of broadcast patterns needed after RFS, the amount of test data needed to be stored to apply these patterns, the total test data needed to be stored including groups and broadcast modes, and finally the data volume reduction factor. The reduction factor is calculated as the ratio of the test data volume needed for the conventional full scan circuit (indicated by 'Baseline*') to the test data volume needed for a particular ILS configuration.

Comparing the results from this table to the traditional ILS procedure described in [24], there is a far more significant reduction in test data volume. This is due to the fact that groups mode has replaced serial mode. In terms of data volume, using serial mode is the equivalent as having one scan-in pin, or group, for each internal scan chain. Thus, as long as there are less groups than internal scan chains, groups mode will require less data to be stored than serial mode. This is most prevalent when using the smallest chain lengths, as there is a much larger difference between the number of groups and the number of scan chains. As a result, the maximum data volume reduction factor is found with smaller chain lengths. For larger industrial circuits, this will be beneficial in terms of tester depth. By using the optimal configuration using the Multiple Group ILS procedure, it is much less likely that test data will exceed tester scan channel depth, in which case access to slower mass storage would have been required. Although not shown, test application time reduction follows the same trends as the data volume reduction as both calculations use the same variables.

6. Minimizing Scan-in Pins

In the previous sections we minimized the number of external scan-in pins with respect to the available test set. If an application requires further reduction in number of scan-in pins, there are several alternatives. Any procedure that finds more exact compatibility information will help find larger compatibility classes which in turn reduces the number of compatibility classes and hence the number of pins. A simple extension of our procedure to find more compatibility information using only an ATPG is as follows.

Using the procedure of Figure 4, find compatibility classes. Pick the largest class among this to form one ILS group, leaving all other chains free to be grouped later. Temporarily connect the free chains to a single scan-in pin. Run ATPG on the configuration with previously formed groups and the temporary group. Remove all detected faults and update the fault list. Undo the temporary group to restore the free chains. Using the configuration of already formed groups and the remaining free chains rerun the ATPG to generate a new unfilled test set, a new incompatibility graph and a new set of compatibility classes from the remaining chains. Pick the largest class among these to form a new group. Repeat this greedy selection procedure until all chains are selected. This greedy procedure is attractive since it permits any off-the shelf ATPG. The run time of ATPG decreases in each iteration since the fault list shrinks in each iteration.

Additionally, we can use *Combinational Compatibility* used in [14]. Combinational compatibility makes use of inverters and two-input gates (e.g. AND, XOR, NOR) at the input of the internal scan chains. Computation for such compatibility is computationally expensive but very effective in reducing the number of pins. Alternatively we can use two or more different Illinois Scan configurations using muxes at the input of each internal chain.

Finally, recall that Illinois Scan Architecture works with an arbitrarily small number of scan-in pins since any undetected fault in group mode can be detected with Serial Mode.

7. Conclusion

Illinois Scan Architecture has been widely accepted in industry for test cost reduction by reducing test time, volume and pins. This paper enhances the effectiveness of Illinois Scan using multiple groups of scan chains. A method for compatibility analysis of scan chains was described using an incompatibility graph. This analysis led to a new configuration of the scan-in pins for the scan chains, called *groups mode*. Two methods were described for test generation: using both broadcast and groups modes and using just groups modes. Both methods were shown to have a significant improvement in both test data volume and test application time reduction over the ad-hoc assignment of chains to pins in Illinois Scan.

References

[1] D. Das and N. A. Touba, "Reducing test data volume using external/LBIST hybrid test patterns," in *Proc. Int. Test Conf.*, October 2000, pp. 115-122.

[2] G. Kiefer, H. Vranken, E. J. Marinissen, and H-J. Wunderlich, "Applications of deterministic logic BIST on industrial circuits," in *Proc. Int. Test Conf.*, October 2000, pp. 105-114.

[3] A. Jas, B. Pouya, and N. A. Touba, "Virtual scan chains: A means for reducing scan length in cores," in *Proc. of the IEEE VLSI Test Symp.*, April 2000, pp. 73-78.

[4] F. F. Hsu, K. M. Butler, and J. H. Patel, "A case study on the implementation of the Illinois scan architecture," in *Proc. of the Int. Test Conf.*, October 2001, pp. 538-547.

[5] I. Hamzaoglu and J. H. Patel, "Reducing test application time for full scan embedded cores," in *Dig. Papers, 29th Int. Symp. Fault-Tolerant Comp.*, June 1999, pp. 260-267.

Table 2. Test Data Volume Reduction Using Multiple Group ILS

		Group Mode			Broadcast Mode				
Circuit/ Mode	**Config**	**Num Groups**	**Num Pats.**	**Mem bits**	**Num Chains**	**Num Pats.**	**Mem bits**	**Total bits**	**Red Factor**
s13207.1	Baseline*	-	468*	327 600	-	0	0	327 600	1.00
dual	ILS-10	8	385	54 670	64	55	3960	58 630	5.59
mode	ILS-6	9	425	49 300	107	18	1224	50 524	6.48
single	ILS-10	8	466	66 172	-	-	-	66 172	4.95
mode	ILS-6	9	449	52 084	-	-	-	52 084	6.29
s15850.1	Baseline*	-	432*	253 952	-	0	0	253 952	1.00
dual	ILS-10	7	303	44 541	54	115	10 005	54 546	4.66
mode	ILS-6	11	315	45 045	89	98	8134	53 179	4.78
single	ILS-10	7	432	63 504	-	-	-	63 504	4.16
mode	ILS-6	11	423	60 489	-	-	-	60 489	4.36
s38417	Baseline*	-	921*	1 532 544	-	0	0	1 532 544	1.00
dual	ILS-32	5	252	47 376	52	594	35 640	83 016	18.46
mode	ILS-20	7	379	63 672	82	344	16 512	80 184	19.11
single	ILS-20	7	885	148 680	-	-	-	148 680	10.31
mode	ILS-14	10	856	143 808	-	-	-	143 808	10.66
s38584.1	Baseline*	-	634*	928 176	-	0	0	928 176	1.00
dual	ILS-16	7	455	68 250	90	155	8370	76 620	12.11
mode	ILS-12	8	509	68 206	119	105	5250	73 456	12.64
single	ILS-16	7	629	94 350	-	-	-	94 350	9.84
mode	ILS-12	8	634	84 956	-	-	-	84 956	10.93

* indicates Serial Mode

[6] A. R. Pandey and J. H. Patel, "An incremental algorithm for test generation in Illinois scan architecture based designs," in *Proc. Design, Automation, and Test in Europe*, March 2002, pp. 368-375.

[7] A. R. Pandey and J. H. Patel, "Reconfiguration technique for reducing test time and test data volume in Illinois scan architecture based designs," in *Proc. IEEE VLSI Test Symp.*, April 2002, pp. 9-15.

[8] A. R. Pandey, "Test time and test data volume reduction techniques for VLSI circuits," M.S. thesis, University of Illinois at Urbana-Champaign, 2001.

[9] S. Samaranayake, E. Gizardski, N. Sitchinava, F. Neuveux, R. Kapur, and T. W. Williams, "A reconfigurable shared scan-in architecture," in *Proc. IEEE VLSI Test Symp.*, April 2003, pp. 9-14.

[10] K-J. Lee, J-J. Chen, and C-H. Huang, "Using a single input to support multiple scan chains," in *Dig. Tech. Papers, 1998 IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1998, pp. 74-78.

[11] S. Mitra and K. S. Kim, "X-compact: an efficient response compaction technique for test cost reduction, " in *Proc. IEEE Intl. Test Conf.*, Oct 2002, pp. 311-320.

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

[13] C. Chen and S. K. Gupta, "A methodology to design efficient BIST test pattern generators," in *Proc. of the Int. Test Conf.*, October 1995, pp. 814-823.

[14] I. Hamzaoglu and J. Patel, "Reducing test application time for built-in-self-test pattern generators," in *Proc. of the IEEE VLSI Test Symp.*, April 2000, pp. 369-375.

[15] M. Sharma, J. H. Patel and J. Rearick, "Test data compression and test time reduction of longest-path-per-gate tests based on Illinois Scan Architecture," in *Proc. of the IEEE VLSI Test Symp.*, April 2003, pp. 15-21.

[16] D. Brelaz, "New methods to color the vertices of a graph," *Communications of the ACM*, vol. 22, no. 4, pp. 251-256, April 1979.

[17] E. C. Sewell, "An improved algorithm for exact graph coloring," in *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 26, 1996, pp. 359-373.

[18] A. Mehrotra and M. A. Trick, "A column generation approach for graph coloring," *Journal on Computing*, vol. 8, no. 4, pp. 344-354, Fall 1996.

[19] W. Klotz, "Graph coloring algorithms," Mathematics Report, Technical University Clausthal, May 2002, pp. 1-9.

[20] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *Proc. Int. Symp. on Circuits and Systems*, May 1989, pp. 1929-1934.

[21] I. Hamzaoglu and J. H. Patel, "New techniques for deterministic test pattern generation," in *Proc. IEEE VLSI Test Symp.*, April 1998, pp. 446-452.

[22] X. Lin, J. Rajski, I. Pomeranz, and S. M. Reddy, "On static test compaction and test pattern ordering for scan designs," in *Proc. Int. Test Conf.*, October 2001, pp. 1088-1097.

[23] V. Agrawal and M. Bushnell, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Norwell: Kluwer Academic Publishers, 2000.

[24] M. A. Shah, "Enhancement of the Illinois Scan Architecture for multiple scan inputs and transition faults" M.S. thesis, University of Illinois at Urbana-Champaign, 2003.