Scan Chain Clustering for Test Power Reduction

Melanie Elm, Hans-Joachim Wunderlich,
Michael E. Imhof, Christian G. Zoellin
Institut fuer Technische Informatik,
Universitaet Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
{elm, imhof, wu, zoellin}@iti.uni-stuttgart.de

Jens Leenstra, Nicolas Maeding
IBM Deutschland Entwicklung
Schoenaicherstr. 220,
D-71032 Boeblingen, Germany
{leenstra, nmaeding}@de.ibm.com

ABSTRACT

An effective technique to save power during scan based test is to switch off unused scan chains. The results obtained with this method strongly depend on the mapping of scan flip-flops into scan chains, which determines how many chains can be deactivated per pattern.
In this paper, a new method to cluster flip-flops into scan chains is presented, which minimizes the power consumption during test. It is not dependent on a test set and can improve the performance of any test power reduction technique consequently. The approach does not specify any ordering inside the chains and fits seamlessly to any standard tool for scan chain integration.
The application of known test power reduction techniques to the optimized scan chain configurations shows significant improvements for large industrial circuits.

Categories and Subject Descriptors

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

General Terms

Algorithms, Reliability

Keywords

Test, Design for Test, Low Power, Scan Design

1. INTRODUCTION

Dynamic power consumption of a circuit under test is of great concern. The elevated switching activity during the test leads to increased power by almost a magnitude compared to functional mode [1]. High *instantaneous* power consumption may lead to increased noise, IR-drop and ground bounce, resulting in undesired yield loss. High *average* power requires additional cooling, may cause increased electro-migration or change the temperature distribution in the die. In addition to yield loss, this may result in early-life failures or impact reliability in general [2].
Techniques such as the reduction of shift speed, circuit partitioning and extensive cooling have negative impact on test cost and test quality. Hence, numerous methods for design for test (DFT), automated test pattern generation (ATPG) and test planning have been proposed and are employed in high-volume manufacturing to reduce test power consumption [3]. For systems-on-a-chip and multi-core chips, test scheduling and test planning methods efficiently test all modules of a chip while keeping a constrained power budget [1, 4, 5]. In scan-based testing, special attention is paid to the switching activity during shifting. DFT techniques include special flip-flops which suppress output toggling during shift and masking of useless patterns [3, 6].
Usually, multiple scan chains are employed for implementing a built-in self-test (BIST), embedded deterministic testing (EDT) or external test. State-of-the-art DFT allows to disable the clocks of individual scan chains for diagnosis of scan-cell failures [7]. This feature may also be used to reduce power consumption significantly. Methods to automatically disable some of the scan chains during certain times have been proposed for deterministic test [8] and built-in self-test [9, 10].
For deterministic tests, the switching activity during shifting can be reduced by filling don't-care values in the test-cubes. A common technique is to repeat the value of the most recent care-bit [11, 12]. The same effect can be achieved by adding a mask register [13, 14] or by using a shadow register [15].
The clustering of the scan-cells into chains has significant impact on the aforementioned techniques. In scan chain disabling, the set of chains that has to be enabled to detect a specific fault should be as small as possible. Similarly, in X-filling, the specified bits per pattern should be concentrated in as few scan chains as possible to allow for long constant runs in the remaining chains.
Up to now, scan chain organization for low power has mainly concentrated on flip-flop reordering [16], which has severe impact on the routing overhead, and special care has to be taken such as in [17].
The approach presented here does not imply any scan chain reordering but partitions the flip-flops into sets. Each set is passed to a standard scan insertion tool which keeps all the degrees of freedom for place and route. Scan insertion tools partition the flip-flops at topological level anyway before layout synthesis for complexity reasons and allow the specification of constraints, e.g. [18]. The layout-aware minimization of the routing overhead [19, 20] is done post-placement in the

DAC 2008, June 8–13, 2008, Anaheim, California, USA

usual way. Hence, scan chain clustering does not introduce any additional impact in the tool flow.
The clustering is independent of the test set and may be combined with any of the power optimization techniques in [8, 9, 10, 11, 12, 13, 14, 15]. In the sequel, the BIST-based power optimization method from [10] is used to evaluate the synthesized scan configurations for benchmark circuits and large industrial circuits. Test power can be reduced by up to 60% compared to scan chain configurations generated by using a standard flow. The proposed algorithm has linear complexity and is suitable for large industrial circuits.
The rest of the paper is organized as follows: Chapter 2 formalizes the partitioning problem for scan chain clustering to maximize the power saving during scan test. Chapter 3 presents an efficient hyper-graph based partitioning algorithm. In chapter 4 the conducted experiments show an additional power reduction of up to 50% on top of standard low power techniques.

2. FORMALIZING SCAN CHAIN CLUSTERING

All the approaches mentioned in section 1 reduce the switching activity by disabling as many scan chains as possible for each test pattern. The amount of disabled scan chains mainly depends on the location of the flip-flops necessary to sensitize and to observe the faults detectable by the pattern. If these flip-flops are distributed over a minimum number of chains a maximum number of chains can be disabled.
In order to detect a fault, an erroneous value has to be observed at one of the flip-flops of its output cone (Fig. 1). Here we consider faults in the conditional stuck-at fault model, which is capable to model any single defect [21].

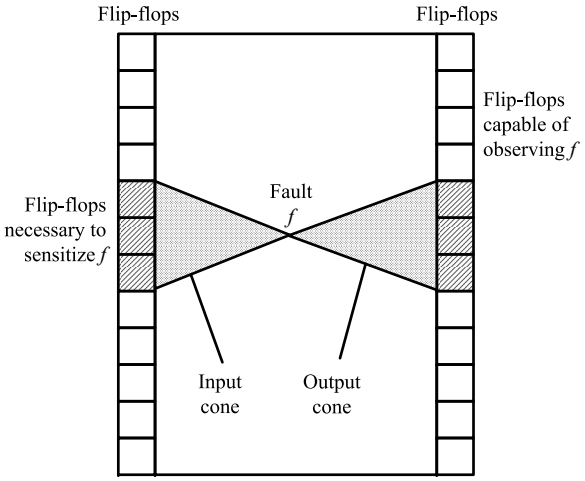

Figure 1: Input cone and output cone of fault f

Each flip-flop of the output cone of fault f defines its input cone which covers f (Fig. 2). In general, each of these cones is smaller than the so-called support of f [22], which is just the union of all of these input cones.
It is sufficient to detect the fault f at just one observer flip-flop o, and o and the flip-flops of its input cone should be distributed across as few scan chains as possible.
In the sequel, we denote the input cone of a flip-flop o together with o itself as the cone of o. Figure 2 shows three cones including their observer flip-flops. The construction of cones does only depend on the circuit structure and not on the applied test set.

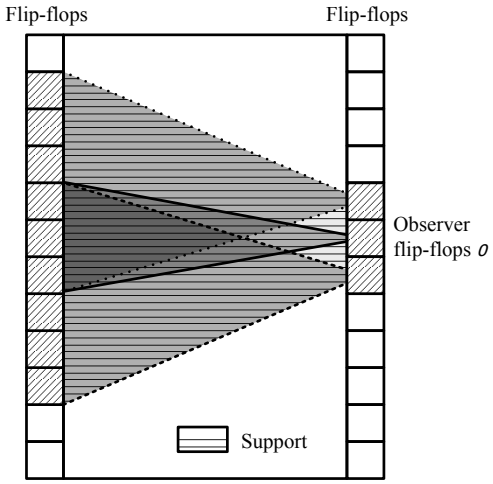

Figure 2: Input cones of observers o

The goal of this work is to provide a scan chain clustering, where the amount of chains spanned by a cone is minimal. To optimize the scan-configurations, we model the circuit topology by using a hypergraph. In this hypergraph $H = (V, E)$ the set of vertices $V = \{v_1, v_2, v_3, \ldots, v_n\}$ represents the scan elements of the circuit. The set of hyperedges $E \subseteq \mathcal{P}(V)$ represents the set of cones derived from the circuit structure.
Using this representation, for every observing flip-flop o a hyper-edge is derived which includes the vertex v_o corresponding to the flip-flop itself and all the vertices $v_{i1}, v_{i2}, ...,$ v_{in} derived from its input cone. Since every flip-flop is a potential observer, the number of hyperedges $|E|$ equals the number of vertices $|V|$.
Let k be the maximum number of scan chains, and t be the maximum number of scan elements in one scan chain (Fig. 3). The scan chain clustering problem is solved by partitioning the hypergraph into k disjoint sets of vertices not larger than t. The optimization objective for this partitioning is to minimize the amount of partitions spanned by each edge, which is achieved by minimizing the global edge cut (GEC):

$$GEC = \sum_{e \in E} (\#spanned\_partitions(e) - 1) \qquad (1)$$

3. EFFICIENT SCAN CHAIN CLUSTERING

The problem of partitioning a graph into k balanced sets of vertices while minimizing the edge cut is NP-complete [23]. For multi-million gate circuits it is impossible to solve this NP-complete problem exactly. Instead, we present an efficient heuristic procedure which exploits application specific properties of the posed problem.
Up to now, substantial research effort has been dedicated to fast and reliable partitioning heuristics for graphs and hypergraphs aiming at different optimization criteria. Exam-

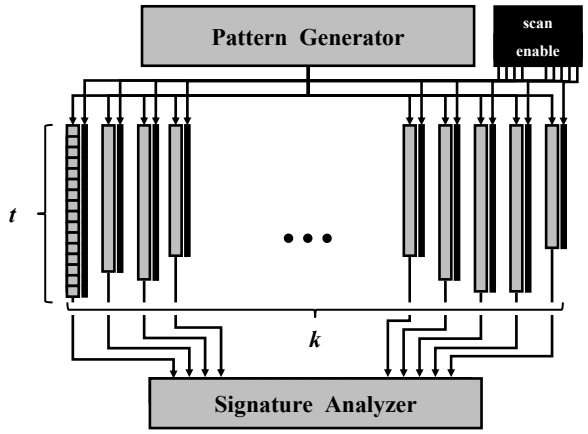

Figure 3: Embedded test scheme with k scan chains of maximum size t

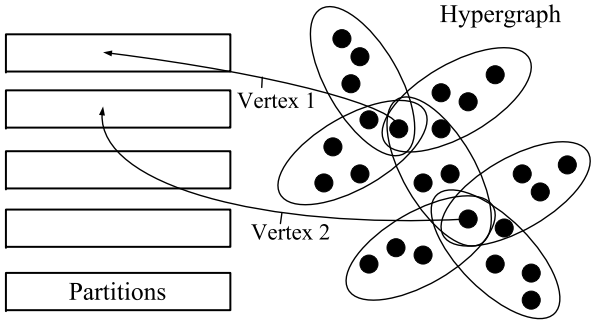

Figure 4: Prepartitioning the two most strongly connected vertices.

ples are spectral methods [23], min cut [24, 25], coarsening [26] or variations of the Fiduccia Mattheyses algorithm [24]. All of these general heuristics do not target large problem instances and are thus not applicable to the problem established here.
In the next sections we will introduce a linear time partitioning heuristic which works out for several hundred thousands of vertices and hyperedges. First a cost function is presented to evaluate partial solutions, the subsections deal with the three phases of the algorithm, and finally some complexity issues are discussed.

3.1 Evaluating Configurations

A configuration Π of the partitioning process is a state, where s vertices were already assigned to partitions and $n - s$ vertices are left. A configuration is evaluated by a cost function, which displays its quality and the difficulty to complete the configuration. This is achieved by assigning a label $L(e)$ to each hyperedge $e \in E$. The label of an edge is the number of partitions this edge is spanning in the current configuration. The cost function is defined as the sum of all these labels.

$$C(\Pi) \quad = \quad \sum_{e \in E} L(e) \tag{2}$$

3.2 Initial Configurations

If the hypergraph is not connected, each of its connected components can be processed independently in order to reduce complexity. All vertices in components smaller than t are simply assigned to a single arbitrary partition, as this can be done without cutting edges. Without loss of generality we consider the hypergraph as a connected component.
Figure 4 depicts some vertices, which induce a natural, initial partitioning of two vertices of the hypergraph as they are the centers of two different strongly connected areas of the hypergraph. If vertex 1 and vertex 2 are assigned to two different partitions, the number of cut hyperedges will still be minimal, and a partitioning algorithm to assign the remaining vertices can be guided.
The initial configuration identifies k vertices like those in figure 4 and assigns them to different, empty partitions. These preassigned vertices will then attract all vertices in their neighborhood to the same partition in the next steps.
The vertices to be prepartitioned are determined by evaluating the connectivity of each vertex. The method selects the vertex with the most incident hyperedges and puts it into the first partition. It then marks all incident hyperedges and proceeds to find another vertex, with the most incident, not yet marked, hyperedges. This is repeated until a vertex has been assigned to all the k partitions.

3.3 Graph Partitioning

The main algorithm adds one vertex after the other to the configuration obtained so far. The cost function evaluates for each vertex, to which of the k partitions it should be added. After the best partition has been found, the vertex is assigned to this partition, and the labels and the cost function are updated accordingly.
Not only the choice of the initial partitioning is important, but also the order in which the vertices are processed. For this reason, we also define a label $L(v)$ for each vertex $v \in V$.

$$L(v) \quad = \quad \max_{\{e|v \in e\}} L(e) \tag{3}$$

The label of a vertex is the maximum label of all the edges incident to this vertex. The next vertex to choose for partitioning is one with maximum label and maximum incidence. As the label cannot exceed k, we can find the next critical vertex fast by storing the vertices in buckets according to their label. The evaluation of the incidence can be done, when the graph is built. Again the vertices are put into buckets, and the next vertex is found in constant time.

3.4 Post-Partitioning

The process described above touches each vertex only once, and some vertices may be partitioned based on incomplete information. The post-processing step removes some of the vertices considering their cost impact by re-evaluating the corresponding partitioning decision.
The contribution of a vertex to the overall cost can be estimated by

$$C(v) \quad = \quad \sum_{\{e|v \in e\}} L(e). \tag{4}$$

The vertices are ranked by their cost $C(v)$, and according to this ranking, the first 10% of vertices are removed. In order to gain more freedom in repartitioning vertices, also the 10% vertices ranked lowest are removed. These vertices are partitioned as before a second time. But here, the order in which the vertices are selected is based on the incidence of a vertex, starting with the vertices with highest incidence.
The post-partitioning step can be applied to any configuration, which may be the original scan design or which may be constructed by the clustering heuristic described above. In cases where the original scan design has already been optimized and should not be changed completely, the post-partitioning step may provide further gain.

3.5 Complexity Considerations

During each of the three steps — initial configuration, graph partitioning, post processing — each of the n vertices is only touched once. However, the number of operations during a step depends on the cardinality of a hyperedge or on the number of hyperedges the vertex is a member of.
The cardinality of a hyperedge corresponds to the size of an input cone, and the number of hyperedges incident to one vertex corresponds to the size of its output cone. In general, the size of an input or output cone is independent of the circuit size n and is limited by a constant, say c_1. However, often we find a few signals connected to a very large number of flip-flops. Examples are buffer trees or asynchronous signals. Again, the number of signals of this kind is limited in a real circuit for timing and regularity reasons to say c_2. Putting this together, we get a complexity estimation of $c_1 \cdot n + c_2 \cdot n$, which indicates a linear complexity with circuit dependent constants c_1, c_2. The results reported below confirm this estimation.

4. RESULTS

This section describes the evaluation of the presented method by using a method for "Scan Test Planning" similar to [10]. All described steps and methods were implemented in Java with an in-house electronic design automation framework and experiments were conducted for a large number of circuits described below.

4.1 Benchmarks and Industrial Circuits

Low power techniques with scan chain disabling are only useful and required for rather large circuits, and only the largest designs are selected from the well known benchmark sets. Table 1 shows the circuits, their basic properties and the results.
The designs from ISCAS89 (denoted by $s*$) and ITC99 ($b*$) do not contain any design for test (DFT) structures and were extended for this article. The circuits provided by NXP ($p*$) already contained parallel scan chains generated by a standard design flow. They represent typical properties of industrial circuits, namely shorter paths and smaller output cones as a consequence of the stronger optimization for high clock rates and low area.
As an example for the application of the presented method to a circuit with partial scan, the Synergistic Processing Element (SPE) of the CELL/B.E. chip was used. The self-test architecture of the CELL/B.E. consists of 15 self-test domains (so called BIST-satellites), each with its own STUMPS instance [27].

4.2 Experimental Setup

For all the designs, the test strategy is a built-in self-test based on reseeding. All the test plans are generated for a test set consisting of 200 seeds, each generating 1024 patterns.
Three different types of scan chain organizations were investigated. The *original* scan organization has been generated with standard tools for the industrial designs and with topological algorithms for the benchmarks. For comparison reasons, the results of a *random* scan insertion are reported. Finally, the outcome of the presented clustering algorithm is denoted as *partition*. As input for this clustering the number k and length t of scan chains were extracted from the *original* scan organization.
The same set of seeds and the same pattern count were applied to all the configurations. With this setup, a scan chain disabling technique similar to [10] was applied to all the configurations.

4.3 Results

The three columns denoted by "# Detected Faults" show the fault coverage obtained for the random, original and partitioned clustering of scan chains by the same test setup. As expected, there is no significant difference in the fault coverage obtained for the three structures, the variations seem to be just random.
The next column of table 1 presents the run time of the clustering algorithm. Even for the largest circuits, the run

Circuit	# Scan		# Faults	# Detected Faults			Partition	% Power rel. original	
	Chains	FFs		Random	Original	Partition	Runtime (s)	Random	Partition
s38417	32	1770	32320	31556	31589	31663	7	212.89	37.72
s38584	32	1742	38358	36385	35989	36389	9	395.21	60.22
b17	32	1549	81330	70535	70300	70238	127	181.09	75.78
b18	32	3378	277976	239594	240385	239599	261	153.06	84.56
b19	32	6693	560696	479901	479834	479037	538	134.88	91.14
p286k	55	17713	648044	609609	610070	610072	10079	109.77	59.09
p330k	64	17226	547808	491079	491528	491464	4534	169.64	48.64
p388k	50	24065	856678	839075	839352	839004	2171	237.39	59.29
p418k	64	29205	688808	639787	639786	640146	6340	182.05	72.85
p951k	82	104624	1590490	1545320	1544906	1544512	6516	185.68	48.87
SPE	32	40027	1065190	904534	904544	904303	637	378.82	93.56
						904434	1653		* 102.55

Table 1: Test results with 200 seeds and 1024 patterns each

time is less than three hours and could be reduced by another order of magnitude by an industrial implementation style instead of Java prototyping. The depicted run times confirm the expected linear scaling and the dependence on the incidence factors c_1 and c_2. p286k is an outlier regarding the incidence factors and besides the impact on the run time this also coincides to the power results, which are relatively high if compared to results for other circuits but are still dramatically reduced by the clustering.

The last two columns of table 1 report the impact on power consumption. As a precise power consumption estimation would require a circuit simulation for each shift cycle, the power is estimated by computing the switching activity of all the flip-flops for the sake of computation time. The numbers are given in relation to the original scan configuration, i.e. 100% is the power after applying scan chain disabling to the original configuration. The power required by a random scan chain clustering exceeds this up to a factor of 4, and the scan design of standard tools already leads to significantly lower power consumption.

However, applying scan chain disabling to the clustering presented here outperforms the original configuration by far. Just 30% to 60% of the power of the original configuration are required for nearly all of the industrial circuits, only for the SPE of the CELL/B.E. the savings seem to be moderate. The number with an asterisk $*$ denotes the power, if the clustering algorithm is performed with the proposed initial configurations, graph partitioning and post-partitioning steps. This fully automated flow reaches nearly the efficiency of the hand-crafted solution. The original scan configuration of the SPE has been created using a custom design flow to meet special design targets. In contrast to most scan designs, here the scan path was timed for full functional speed [27]. Hence, the scan clustering was part of the functional logic and circuit design process and was conducted in a hierarchical manner. For each unit of the circuit, the scan configuration has been created using specially crafted tools or even created and optimized manually. Consequently, the scan clustering already exploits as much of the topological information as possible in order to meet timing targets. If this original clustering is taken as the initial configuration for the post-partitioning step power consumption is reduced further to 93.56%.

Summarizing, we presented a fully automated solution which nearly reaches the quality of sophisticated, hand-crafted designs. If the procedure is applied to such a well optimized structure, even further improvements are obtained. For a standard scan design, the presented solution cuts down power during test by another factor of 2 on average.

Circuit	% Power
s38417	10.49
s38584	2.63
b17	38.35
b18	55.03
b19	62.93
p286k	53.83
p330k	28.40
p388k	24.50
p418k	39.95
p951k	26.27
SPE	23.83

Table 2: Power after scan clustering and disabling

Table 2 shows the results of combining scan clustering and disabling with respect to standard BIST. Now, 100% power is the activity of the original scan configuration without disabling. The fault coverage is not changed.

The power required now lies inbetween 2.63% and 62.93% of the standard test. In many cases, the values obtained are below the power consumption in system mode. Further reduction is not reasonable in order to have comparable operating conditions during test and system mode. Further research will investigate the impact of the presented scheme on the detection of non-target faults and defect coverage [28].

5. CONCLUSION

The clustering of flip-flops into scan chains has significant impact on the power savings obtainable during test. While standard scan insertion methods already provide viable results, the presented clustering algorithm achieves substantial power reductions over standard methods. The algorithm scales to large, industrial designs, and the power during the test is reduced by more than a factor of 2 on average. The resulting clustering fits perfectly into standard scan insertion design flows and does not constrain the optimization techniques employed there.

6. ACKNOWLEDGMENT

This work was supported by the IBM CAS project „Improved Testing of VLSI Chips with Power Constraints“, as well as by the „Deutsche Forschungsgemeinschaft“ within the DFG project „Realtest“ Wu245/5-1. The circuits from NXP were provided in the context of „Realtest“.

CELL/B.E. is a trademark of Sony Computer Entertainment Inc.

7. REFERENCES

[1] Y. Zorian, “A distributed BIST control scheme for complex VLSI devices,” in *Proceedings of the 11th IEEE VLSI Test Symposium (VTS ’93)*, 1993, pp. 4–9.

[2] C. F. Hawkins and J. Segura, “Test and reliability: Partners in IC manufacturing,” *IEEE Design & Test of Computers*, vol. 16, no. 3 and 4, pp. 64–71, 1999.

[3] P. Girard, “Survey of low-power testing of VLSI circuits,” *Design & Test of Computers, IEEE*, vol. 19, no. 3, pp. 80–90, 2002.

[4] Y. Huang, S. M. Reddy, W.-T. Cheng, P. Reuter, N. Mukherjee, C.-C. Tsai, O. Samman, and Y. Zaidan, “Optimal core wrapper width selection and SOC test scheduling based on 3-D bin packing algorithm,” in *Proceedings IEEE International Test Conference, Baltimore, MD, USA, October 7-10*, 2002, pp. 74–82.

[5] N. Nicolici and B. M. Al-Hashimi, “Power-conscious test synthesis and scheduling,” *IEEE Design & Test of Computers*, vol. 20, no. 4, pp. 48–55, 2003.

[6] S. Gerstendoerfer and H.-J. Wunderlich, “Minimized power consumption for scan-based BIST,” in *IEEE International Test Conference (ITC ’99), NJ, USA, 27-30 Sept.*, 1999, pp. 77–84.

[7] Y. Huang, W. Cheng, and J. Rajski, “Compressed pattern diagnosis for scan chain failures,” in *IEEE International Test Conference (ITC ’05), 8-10 Nov., Austin TX*, 2005, p. 30.3.

[8] R. Sankaralingam, N. A. Touba, and B. Pouya, “Reducing power dissipation during test using scan chain disable,” in

19th IEEE VLSI Test Symposium (VTS '01), 29 April - 3 May, Marina Del Rey, CA, USA, 2001, pp. 319–325.

[9] C. Zoellin, H.-J. Wunderlich, N. Maeding, and J. Leenstra, "BIST power reduction using scan-chain disable in the Cell processor," in *IEEE International Test Conference (ITC '06), Santa Clara, CA, USA, Oct. 24 - 26*, 2006.

[10] M. E. Imhof, C. G. Zoellin, H.-J. Wunderlich, N. Maeding, and J. Leenstra, "Scan test planning for power reduction," in *Proceedings of the 44th Design Automation Conference (DAC '07), San Diego, CA, USA, June 4-8*, 2007, pp. 521–526.

[11] K. M. Butler, J. Saxena, T. Fryars, and G. Hetherington, "Minimizing power consumption in scan testing: Pattern generation and DFT techniques," in *IEEE International Test Conference (ITC '04), Oct. 26-28, Charlotte, NC, USA*, 2004, pp. 355–364.

[12] S. Kajihara, K. Ishida, and K. Miyase, "Test vector modification for power reduction during scan testing," in *20th IEEE VLSI Test Symposium (VTS '02), 28 April - 2 May, Monterey, CA, USA*, 2002, pp. 160–165.

[13] P. M. Rosinger, B. M. Al-Hashimi, and N. Nicolici, "Low power mixed-mode BIST based on mask pattern generation using dual LFSR re-seeding," in *20th International Conference on Computer Design (ICCD '02), VLSI in Computers and Processors, 16-18 Sept., Freiburg, Germany*, 2002, pp. 474–479.

[14] J. Lee and N. A. Touba, "Low power test data compression based on LFSR reseeding," in *22nd IEEE International Conference on Computer Design: VLSI in Computers & Processors (ICCD '04), 11-13 Oct., San Jose, CA, USA*, 2004, pp. 180–185.

[15] G. Mrugalski, J. Rajski, D. Czysz, and J. Tyszer, "New test data decompressor for low power applications," in *Proceedings of the 44th Design Automation Conference (DAC '07), San Diego, CA, USA, June 4-8*, 2007, pp. 539–544.

[16] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, "Techniques for minimizing power dissipation in scan and combinational circuits during test application," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 12, pp. 1325–1333, 1998.

[17] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, and A. Virazel, "Design of routing-constrained low power scan chains," in *Design, Automation and Test in Europe (DATE '04), 16-20 Feb., Paris, France*, 2004, pp. 62–67.

[18] J. Beausang, C. Ellingham, and M. Robinson, "Integrating scan into hierarchical synthesis methodologies," in *Proceedings IEEE International Test Conference (ITC '96), Test and Design Validity, Washington, DC, USA, October 20-25*, 1996, pp. 751–756.

[19] M. Hirech, J. Beausang, and X. Gu, "A new approach to scan chain reordering using physical design information," in *Proceedings IEEE International Test Conference (ITC '98), Washington, DC, USA, October 18-22*, 1998, pp. 348–355.

[20] S. Makar, "A layout-based approach for ordering scan chain flip-flops," in *Proceedings IEEE International Test Conference (ITC '98), Washington, DC, USA, October 18-22*, 1998, pp. 341–347.

[21] O. E. Cornelia and V. K. Agarwal, *Conditional Stuck-at Fault Model for PLA Test Generation.* VLSI Design Laboratory, McGill University, 1989.

[22] I. Hamzaoglu and J. H. Patel, "New techniques for deterministic test pattern generation," in *16th IEEE VLSI Test Symposium (VTS '98), 28 April - 1 May , Princeton, NJ, USA*, 1998, pp. 446–452.

[23] P. K. Chan, M. Schlag, and J. Zien, "Spectral k-way ratio-cut partitioning and clustering." in *30th Conference on Design Automation, 14-18 June*, 1993, pp. 749–754.

[24] C. Fiduccia and R. Mattheyses, "A linear-time heuristic for improving network partitions." in *19th Conference on Design Automation, 14-16 June*, 1982, pp. 175–181.

[25] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs." *Bell System Technical Journal, February*, vol. 49, no. 2, pp. 291–307, 1970.

[26] G. Karypis and V. Kumar, "Multilevel k-way hypergraph partitioning." in *Proceedings 36th Design Automation Conference, 21-25 June, New Orleans, LA, USA*, 1999, pp. 343–348.

[27] M. Riley, L. Bushard, N. Chelstrom, N. Kiryu, and S. Ferguson, "Testability features of the first-generation Cell processor," in *Proceedings of the IEEE International Test Conference (ITC '05), 8-10 Nov., Austin TX*, 2005, p. 6.1.

[28] Y. Tang, H.-J. Wunderlich, H. Vranken, F. Hapke, M. Wittke, P. Engelke, I. Polian, and B. Becker, "X-Masking during logic BIST and its impact on defect coverage," in *IEEE International Test Conference (ITC '04), Oct. 25-28, Charlotte, NC, USA*, 2004, pp. 442–451.