MultiCraft

International Journal of Engineering, Science and Technology
Vol. 2, No. 11, 2010, pp. 82-89

INTERNATIONAL JOURNAL OF ENGINEERING, SCIENCE AND TECHNOLOGY
www.ijest-ng.com

Multi-level sequential circuit partitioning for test vector generation for low power test in VLSI

V. H. Prathyush [1] and K. Somasundaram[2]

[1]Amrita Vishwa Vidyapeetham, Coimbatore, INDIA
[2]Department of Mathematics, Amrita Vishwa Vidyapeetham, Coimbatore, INDIA
[1,2]Corresponding Authors' e-mails: s_sundaram@cb.amrita.edu, Tel +91-422-2685600 (K. Somasundaram); prathyushvh@gmail.com (V. H. Prathyush)

Abstract

Sequential graph partitioning algorithms have been developed to fulfill the requirements of emerging multi-phase problems in circuit testing models. In this paper, we present a multi-level graph partitioning algorithm for circuit partitioning, which will minimize the number of test vectors during a low power test in VLSI circuits. By reducing the number of test vectors, we can reduce the energy consumption during the test. Our experimental results with ISCAS bench mark circuits have shown that the power can be reduced up to 55%.

Keywords: Graph partitioning, Circuit partitioning, BIST, Automatic test pattern generation, Low power test.

1. Introduction

Since the days of the first ASIC, improvements in design and manufacturing for Very Large Scale Integration (VLSI) circuit test has been driven by basic economics. The central issue has always been how to apply sufficient test data to a design to ensure that the highest quality is reached, but at the same time to minimize the impact on the cost of manufacturing, and time on the testing. With the continuing escalation of circuit design complexity, testing may soon cost more than its design.

Built-in Self Test (BIST) has been the answer to the test economics trade off in the past (Zorian and Wunderlinch, 1997). However, many traditional solutions have several shortcomings like poor design flow integration, unpredictable coverage and cumbersome diagnostics. All these have hampered designer's attempts to implement BIST. As the complexity of VLSI circuit increases, there is a constant requirement for efficient method to find an Automatic Test Pattern Generation (ATPG). These test patterns must have high fault coverage in identifying faulty chips. Generally, test Pattern Generations (TPG) are classified according to the class of test patterns they produced, like, Deterministic test pattern, Algorithmic test patterns, Exhaustive & pseudo exhaustive test patterns, random & pseudo random test patterns. These are the most common test patterns in BIST applications. Linear Feedback Shift Registrars (LFSR) and Cellular Automata (CA) are the primary hardware's for producing these test patterns. Pseudo-random test patterns have properties similar to those of random pattern sequences, but the sequences are repeatable. The LFSR is one of the most frequently used TPG implementations in BIST applications.

As the complexity of VLSI circuits increases, it has become more important to test VLSI circuits completely. Today's large and complex VLSI circuit in System on Chips (SoC) environments, a need for an enormous amount of test data. When SoCs are tested, data are transferred to the Circuit Under Test (CUT) from Automatic Test Equipment (ATE). Since the channel width and the size of memory for the ATE are limited, the traditional ATE must be modified or more expensive ATE must be developed in order to test an SoC with enormous test data. In addition, if the original test data are reduced to the size of the ATE memory by eliminating useful test patterns, then the accuracy of testing will be diminished. Currently testing ranks among the most expensive and problematic aspects in a circuit design cycle, revealing the ceaseless need for innovative and test-related solutions. As a result, researchers have developed several techniques that enhance design testability through modifications and improve the test generation and application processes. Traditionally, test engineers will evaluate these techniques according to various parameters, like, area overhead, fault coverage, test application time, test development effort, and so forth. Due to the recent development of high-performance and low-power devices in deep-submicron levels, new classes of sophisticated electronic products are emerged.

This new class of systems makes the power management as a critical parameter; test engineers cannot ignore this power during the test. The test power could be as much as twice the power consumed during the normal mode. Excessive power consumption during the test can cause several problems. This power consumption leads to an increasing in peak current and electro migration and this will affect the reliability of the system. In addition, power consumption during the test is more important, since excessive heat dissipation can damage the CUT directly. Moreover, it can create problems such as increasing the product cost, difficulty in performance verification, reduced autonomy of portable systems and decrease of overall yield. In this paper, we proposed a multi-level circuit partitioning algorithm for low power test. Our proposed method is based on a graph partitioning algorithm.

The rest of the paper is organized as follows: Section 2 describes a graph partitioning algorithm with relevant notation and definitions. Section 3 describes the details of low power testing and BIST techniques. Experimental results with ISCAS benchmark circuits are presented in section 4. Finally, the conclusions are given in section 5.

2. Circuit Partition

2.1 Graph Partition

Graph partitioning is one of the important and well-known problems in VLSI circuit design and testing. The objective is to divide the circuit into blocks such that each component falls within prescribed sizes and the complexity of connection between these components will be reduced. The aim of many of the VLSI physical design problems is to minimize the area of the chip occupied by the wires and cells, it can be modelled by embedding a graph into a grid. Developing a good partitioning algorithm for an undirected graph is critical. In general, graph partitioning problem is NP-complete. However, many algorithms have been developed for reasonable partition. Hendrickson and Leland (1993) as well as Karypis and Kumar (1995b, 1998a,b) were introduced a new class of multilevel graph partitioning techniques. Several authors were introduced the matrix partitioning, in particular, sparse matrix partitioning by Riyavong and Karypis and Kumar (1995a). These multilevel schemes will provide an excellent graph partitioning but have moderate computational complexity. Though, these multilevel algorithms are quite fast compared to spectral methods, parallel formulations of multilevel algorithms are needed. Savage and Wloka (1991) have extensively studied a graph embedding heuristic based on parallel Mob heuristic for graph partitioning algorithm. Kernighan and Lin (1970) have developed a heuristic algorithm (K-L algorithm) in polynomial time for two way uniform partitioning. Cong and Wu (2002) proposed a global clustering based multi-level partitioning algorithm for performance optimization. For cut size minimization, hMetis gives a significant solution (Karypis *et al*., 1997). Muthukumar and Selvaraj (2003) gave a comparison of heuristic algorithms for variable partitioning in circuit implementation. Cherng and Chen (2003) as well as Ou and Pedram (2001) have shown a new multi-level bipartitioning algorithm based on the partitioning process, which integrates a clustering technique and an iterative improvement. This clustering algorithm is used to reduce the complexity and to improve the performance.

A k-partition problem is to partition the vertices of a graph into k roughly equal parts, each with m vertices, such that the number of edges connecting the vertices in different parts is to be minimized. That is, for a given weighted graph $G = (V, E)$, let c_{ij} be the weight or cost of an edge e_{ij} in E and k be the number of partitions of G. The k-partition of G is to find a set of disjoint subsets of $V_1, V_2, \ldots, V_k$ such that $\cup V_i = V$, $Vi \cap Vj = \phi$, $i \neq j$ and $C = \Sigma c_{ij}$ is minimum. Here, C is called the cut cost of the partition.

Even the simplest partitioning algorithm will contains all the significant features of larger problems. By iterative (divide-conquer) procedure of K-L algorithm for 2-way partitioning, one can obtain k-partition. But, in this case, we cannot obtain the optimum cut size. The K-L algorithm has a time complexity of $O(n^2 \log n)$. We can generate a k-partition, each with m elements. Starting with a random partitioning of k sets of m vertices, the K-L algorithm is applied for two way partitioning procedure on each pair of partitions. Since there are $\begin{pmatrix} k \\ 2 \end{pmatrix}$ pairs, the time complexity for one pass through all pairs for the $O(n^2)$-procedure is $O(k^2 n^2)$. Hence, the generalization of this procedure leads to a non-polynomial time. Fiduccia and Mattheyses (FM) (1982) presented a K-L inspired algorithm in which iteration can be done in $O(|E|)$ time.

2.2 Multilevel Partitioning

Multilevel partitioning algorithms work with the graph at multiple levels of granularity. We can increase the levels of granularity by combining adjacent vertices into composite vertices at the next coarser level of granularity. This maps the graph into another graph with fewer vertices. Connectivity of the original graph is preserved by ensuring that two vertices at level k are connected by an edge, if and only if, there exists one or more edges between vertices at the next finer level, which will combined to form the vertices at the coarsed level.

The process of coarsening a graph by combining pairs of adjacent vertices repeats until a graph is obtained with sufficiently fewer vertices. An initial partition for the coarsed graph is interpolated at the next finer level. After interpolation, the partition may be improved using a local refinement algorithm and interpolation is repeated until the original level of the graph has reconstructed.

In multi-level paradigm, sequences of successive coarser hyper graphs (See Figure 1) are constructed and bisect the smallest hyper graph (Karypis and Kumar, 1998b; Karypis *et al*. 1997). By successive projection and refining the bisection to the next level of finer hyper graph, we can get the original graph. The problem of computing an optimal bisection of a hyper graph is NP-complete. However, because of the importance of the problem in many application areas, many heuristic algorithms have been developed. In a class of iterative refinement partitioning algorithms, an initial bi-section is computed (often obtained randomly) and then the partition is refined by repeatedly moving vertices between the two parts to reduce the hyper edge cut. These algorithms often use the K-L (Kernighan and Lin, 1970) heuristic. FM refinement heuristic is also used to improve quality of partition. In all these methods, a vertex is moved if it produces the greatest reduction in the edge cuts, which is also called the gain for moving the vertex. The partitions produced by these methods are often poor, especially for larger hyper graphs. Another class of hyper graph-partitioning algorithms performs partitioning in two phases. In the first phase, the hyper graph is coarsened to form a small hyper graph, and then the FM algorithm is used to bisect the small hyper graph. In the second phase, the partitioned graph is reconstructed to the original graph with same number of partition. Since FM refinement is done only on the small coarse hyper graph, this step is usually fast, but the overall performance of such a scheme depends upon the quality of the coarsening method. In many schemes, the projected partition is further improved by using the FM refinement scheme. Recently, a new class of partitioning algorithm was developed based on multilevel paradigm.

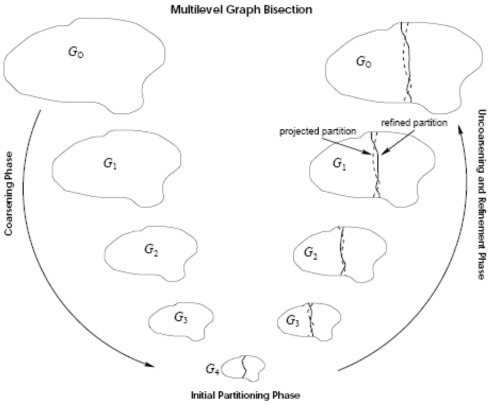
Multilevel Graph Bisection
G0
G0
projected partition
refined partition
G1
G1
Coarsening Phase
Uncoarsening and Refinement Phase
G2
G2
G3
G3
G4
Initial Partitioning Phase

Figure 1: Multi level partitioning phases

One can refer (Karypis and Kumar (1995b, 1998a,b) for more details about coarsening, uncoarsening and refinement phases.

3. Low Power Testing

3.1 Basic Ideas

Low power has emerged as principal theme in today's VLSI industry. The need for low power has caused a major paradigm shift, in which the power dissipation is as important as performance and area (Pedram, 1996; Zorian and Wunderlinch, 1997). Power dissipation in digital CMOS circuits is caused by four sources, namely, the leakage current, the standby current, the short circuit current and the capacitance current. The current due to short circuit and change in the capacitance leads to a dynamic power dissipation in the CMOS circuits. An average power is the total distribution of power over a time period. The peak power is the highest power value at any given instant. It determines the thermal and electrical limits and system packing requirements. If peak power crosses a threshold value (fixed by the designer) then there is a possibility of circuit damage. Energy is the total switching activity generated during a test application. It will affect the lifetime of the battery. Cell internal power is the power consumed by the cell when an input changes, but output does not change. It estimates the short-circuit power.

The energy consumed per switching at a node i is $\frac{1}{2}\, C(i)V^2_{D,}$ where $C(i)$ is the output capacitance at the node i, and V_D is power supply voltage. $C(i)$ is approximately proportional to the fanout at node i. If a signal at node i switches $m(i)$ times during test, then the energy consumed at node i during the test is $\frac{1}{2}\, m(i)\, f_{anout}(i)C(i)V^2_D$. The energy consumed by the CUT during test for a given partition is $\Sigma_p \frac{1}{2}\, m(i)\, f_{anout}(i)C(i)V^2_D$, where the summation over all the nodes in the partition p. Hence, the total energy consumed by the CUT during test is $E = \Sigma\, (\Sigma_p \frac{1}{2}\, m(i)\, f_{anout}(i)C(i)V^2_D),$ the summation over all possible partitions.

If l_p is the length of a test vector generated from a partition p with 100% fault coverage, then the average power is the average energy per clock period. That is, $Power = E/(\Sigma l_p\, .T)$, where T is the clock period. The peak power consumption corresponds to the maximum of the instantaneous power consumed during the test session. Hence, the peak power is a ratio between the highest energy consumed during one clock period and T.

3.2 Low Power BIST Technique

This BIST architecture includes two essential functions as well as two additional functions that are necessary to facilitate execution of the self-testing feature in the system (Stround, 2002). The two essential functions include the Test Pattern Generator (TPG) and Output Response Analyzer (ORA). While the TPG produces a sequence of patterns for testing the CUT, the ORA compacts the output responses of the CUT into some type of *Pass/Fail* indication. The other two functions needed for system-level use of the BIST include a test controller (or BIST controller) and an input isolation circuitry. A simple BIST architecture is shown in Figure 2.

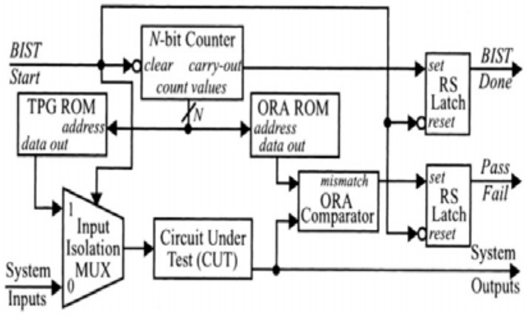

Figure 2:A simple BIST architecture

This type of BIST architecture is rarely used in practical applications, since it requires conventional test vector development and considerable circuit area for the TPG, ORA-ROMs and the counter. This is particularly true when a set of test vectors and expected output responses are large in terms of the number of bits and number of vectors. But a minor design change in the CUT could leads to regeneration of the entire test vectors. The expected output response would require re-programming the ROMs in the best case, and in the worst case, resizing the ROMs and counter. As a result, the minor design change in CUT leads to a major change in the BIST implementation. In addition, the comparator will not be completely tested during the BIST sequence such that additional testing would be required to insure that the comparator is fault-free. Various techniques are available (Stround, 2002; Ou and Pedram, 2001; Girard, 2002; Zhang and Roy, 1999) for low power during BITS. Some of the testing techniques are Toggle suppression, LFSR turing, Vector filtering techniques, low power RAM testing and circuit partitioning (Cong et al., 1999; Ecuyer, 1997; Guiller *et al*., 2002).

In this paper, we focused on circuit partitioning algorithm for BIST. During the circuit partitioning, the original circuit is divided into *k* structural sub circuits, such that the *k* different BIST sessions can successively tested. Based on this strategy, we can partition the circuit into *k* subcircuits and minimize the average and peak powers. Moreover, this strategy also reduces the total energy consumed during BIST because the test length required for the sub circuits is not more than that of the original circuit. The proposed strategy applies to either scan-based or parallel BIST with slight modification to conventional test pattern generator structures. Area overhead is very low and this is almost no penalty on circuit performance.

A given circuit is partitioned in to structured subcirciuts, each sub circuit can be tested with different Built-in Self Tests (BIST). Since the cut size of the partitioning circuit is minimal, the interconnections between each subcircuits are also minimal. Hence, it reduces the area over head. By introducing multiplexers, we can get the connectivity between the inputs and outputs of the each circuit. These multiplexers are laid in between the circuits for better connectivity. Also these multiplexers will keep the other subcirciuts idle while generating the test vectors in one sub circuit. That is, these multiplexers will nullified the test vector generation with other subcircuits. The average and peak powers of the circuit can be calculated from the average and peak power of each subcircuits. Hence, the average and peak powers of the partitioned circuit are minimum than the average and peak powers of unpartitioned circuit. More over the sum of the length of test vectors of these subcircuits is less than the length of test vectors of the circuit, therefore, we can achieve 100% fault coverage during the test.

4. Experimental Results

In this paper, we are using the scan based BIST architecture. Here, the scan registrars are made of scan elements. This BIST architecture reduces the power during CUT. We have experimented this method with various ISCAS'85 benchmark circuits and we implemented in C on Intel Pentium IV 400 Mhz with 1GB memory. We read an ISCAS'85 benchmark circuit and analysis its faulty sites and switching activities. We convert the benchmark circuits into graph files and partition it using Metis (Karypis and Kumar, 1998a,b) and analyzed each partition separately. For each partition, we calculated the number of test vectors and finally we calculated the switching activity at the given inputs form the different partition.

Table 1 shows that the total number of switching activity at a primary level for different test vectors using exhaustive type of testing (without partition). Here, the switching activities are calculated by binary sequence and sequence form gray codes. This results show that gray code sequence gives better result than the binary one. Again for large number of partitions, testing a CUT is a laborious task. Hence, from the Table 1, it is not recommended that the simple exhaustive type of testing with large inputs.

We can calculate the switching activities using random pattern generation. In this case, we cannot guarantee 100% fault coverage. Fault coverage depends on initial seed used for generating random patterns. Fault coverage can be increased by using the genetic algorithm. An alternate solution could be pseudo exhaustive test, here, we test the benchmark circuits using *k*-way partitioning method. For a *k*-way partition, if *n* is number of vertices and *m* is the size of the largest partition produced by the *k*-way partitioning algorithm then the balance of the partitioning is defined as *km*/*n.* Metis (Karypis and Kumar, 1998a,b) produces partitions that are perfectly balanced at each bisection level, some small load imbalance may result due to the log *k* levels of recursive bisection, however, the load imbalance is less than 1%. Table 2 shows that, the edge cut and balance condition by *k*-way partitioning.

Table 1: Total switching activity at primary inputs without partitioning

Benchmar*k* circuit	Number of inputs	Number of outputs	Number of test vectors	
			Binary sequence (bits per seconds)	Gray code sequence (bits per seconds)
C17	5	2	57	31
C499	41	32	4.3 e+12	2.1 e+12
C1355	41	32	4.3 e+12	2.1 e+12
C2670	233	140	2.7 e+73	1.3 e+73
C3540	50	22	2.2 e+15	1.1 e+15
C5315	178	123	7.6 e+56	3.8 e+5
C7552	207	108	4.1 e+63	2.0 e+63

Table 2: Calculation of edge cut and balance condition by *k*-way partitioning

Bench mark Circuit	Number of Partitions	*k*-way partitioning	
		Edge Cut (bits per seconds)	Balance Condition (bits per seconds)
C17	2	2	1.38
	3	4	1.38
C499	15	181	1.69
	100	286	2.18
C1355	150	576	1.94
	200	629	1.94
	250	651	2.02
C2670	50	388	1.15
	200	823	1.92
	800	1360	2.04
C3540	300	1593	2.93
	500	1849	2.3
	800	1981	3.22
C5315	600	2631	2.76
	1000	3011	2.3
C7552	600	2993	2.51
	1500	3940	2.35

Now, results about the power and energy savings achieved by the proposed low power BIST scheme are discussed here. Power consumption in each circuit was estimated by using PowerMill, a dynamic simulator provided by the Epic Technology Group of Synopsys version Y-2006.06, assuming a clock period of 60 nanoseconds, global operating voltage 1.2V, voltage unit 1V, capacitance unit 1 pF, dynamic power unit 1 mW and leakage power unit 1 pW. Based on the circuit partitioning and their switching activities, we calculated the cell internal power, net switching power and total dynamic power. Table 3 gives the switching activity of various benchmark circuits using the multi-level circuit partition algorithm. We observed that, due to circuit partitioning, the switching activity is minimized with 100% fault coverage. In particular, generating test vectors using grey code sequences in multi-level scheme is well suet for low power testing.

Table 3: Switching activity of different benchmark circuits with different partitions

Number of Partitions Bench mark Circuit		Switching activities by multi-level circuit partitioning		Cell internal power (μW)	Net switching power (μW)	Total dynamic power (μW)	Cell leakage power (*n*W)
		Binary sequence (bits per seconds)	Gray code sequence (bits per seconds)				
C17	2	15	10	14.76	4.35	18.31	63.33
	3	9	7	6.96	4.16	12.15	60.75
C499	15	253	147	18.51	5.34	24.02	65.46
	100	53	47	4.25	3.17	7.14	61.49
C1355	150	45	43	20.73	6.32	26.87	70.81
	200	45	43	16.41	4.62	21.10	63.27
	250	41	41	13.92	2.14	16.14	51.86
C2670	50	54269	27251	21.47	5.23	25.91	68.14
	200	741	487	15.64	3.21	18.42	62.65
	800	239	236	6.56	2.13	9.23	53.87
C3540	300	54	52	24.87	6.05	28.32	70.21
	500	50	50	16.25	4.31	21.04	64.83
	800	50	50	9.56	2.17	12.15	50.14
C5315	600	232	205	20.14	5.32	25.84	60.67
	1000	192	185	12.69	2.07	15.23	52.37
C7552	600	409	308	30.15	9.82	40.54	73.89
	1500	239	223	18.37	4.61	12.92	62.38

Table 4 shows that the reductions in total dynamic power and energy consumption expressed in percentages. Our results with ISCAS'85 benchmark circuits show that the average power can be reduced up to 55% and reduction in the energy is up to 85%. Since number of test vectors in each sub circuit is smaller than the original, we can achieve 100% fault coverage during the test.

Table 4: Power and Energy reduction of benchmark circuits

Benchmark Circuit	Power reduction	Energy reduction
C17	54.23 %	88.17 %
C499	53.56 %	88.94 %
C1355	55.45 %	91.38 %
C2670	57.62 %	82.59 %
C3540	50.18 %	80.43 %
C5315	58.22 %	80.91 %
C7552	56.34 %	83.86 %

6. Conclusion

In this paper, we have shown that, testing power can be reduced if we go for multi-level circuit partitioning algorithm. Our experimental results demonstrate that the proposed method will give the minimum number of switching activities during a BIST and hence it reduces the power. It is easy to under standard that, even if we test the technique with sequential benchmark circuits, we will get a better solution than any other techniques. Our results with ISCAS'85 benchmark circuits show that, the average power can be reduced up to 55 % and reduction in the energy can be achieved up to 85% with 100% fault coverage. Finally, we understand that multi-level circuit partitioning algorithm is well suit for low power test of VLSI circuits. Any further improvement can be achieved only by improving the best initial partition and uncoarseing phase.

References

Cong J., Li H., and Wu C., 1999. Simultaneous circuit partitioning/ clustering with retiming for performance optimization, in *Proceedings of ACM/IEEE Design Automation Conference*, pp. 460-465.

Cong J., Wu C., 2002. Global clustering-based performance-driven circuit partitioning, in *Proceedings of ISPD*, pp. 149-154.

Cherng J.S. and Chen S.J., 2003. An efficient multi-level partitioning algorithm for VLSI circuits, in *Proceedings of IEEE Conference on VLSI Design*, pp. 70-79.

Ecuyer P.L., 1997. Uniform random number generators: A review, in *Proceedings of Winter simulation Conference,* pp. 127-134.

Fiduccia C.M and Mattheyses R.M, 1982. A linear-time heuristic for improving network partitions, in *Proceeding of 19th Conference on Design Automation*, pp. 175-181, June.

Girard P., Guiller L., Landrault C. and Pravossoudovitch S., 2002. Low power BIST Design by Hypergraph Partitioning: Method and Architecher, in *Proceedings of ITC International Test Conference*, pp. 652-661.

Girard P., 2002. Survey of low-power testing of VLSI circuits, *IEEE Design and Test of Computers*, Vol. 19, No. 3, pp. 82-92.

Hendrickson B. and Leland R., 1993. A multilevel algorithms for partitioning graphs, Technical report SAND 93-1301.

Karypis G. and Kumar V., 1995a. A parallel algorithm for multilevel graph partitioning and sparse matrix ordering, Technical Report, TR 95-036.

Karypis G. and Kumar V., 1995b. Analysis of multilevel graph partitioning, Technical Report, TR 95-035.

Karypis G. and Kumar V., 1998a. A fast and highly quality multilevel scheme for partitioning irregular graphs, *SIAM journal of Scientific Computing*, Vol.20, No. 1, pp. 359-392.

Karypis G. and Kumar V., 1998b. Multilevel *k*-way partitioning scheme for irregular graphs, *Journal of Parallel and Distributive Computing*, Vol. 48, No. 1, pp. 96-129.

Karypis G., Aggarwal R., Kumar V. and Shekhar S., 1997. Multilevel hyper graph partitioning: applications in VLSI domain, in *Proceedings of ACM / IEEE Design Automation Conference*, pp. 526-529.

Kernighan B.W. and Lin S., 1970. An efficient heuristic procedure for partitioning graphs, *The Bell System Technical Journal.*

Muthukumar V. and Selvaraj H., 2003. Comparison of heuristic algorithms for variable partitioning in circuit implementation, in *Proceedings of IEEE Design Automation Conference*, pp. 256-260.

Ou S.L., Pedram M., 2001. Timing-driven partitioning using iterative quadratic programming, at http://atrak.usc.edu/~massoud/.

Pedram M., 1996. Power minimization in IC design: principles and applications, *ACM Transactions on Design Automation of Electronic Systems*, Vol.1, No.1, pp. 2-56.

Riyavong S., Experiments on Sparse Matrix partitioning, CERFACS working Note WN/PA/03/32, CERFACS, France.

Savage J.E. and Wloka M.G., 1991. Parallelism in graph partitioning, *Journal of Parallel and Distributed Computing*, Vol.13, pp. 257-272.

Stround C.E. 2002., A Designer's Guide to Built-in Self Test, Kluwar Academic Publishers.

Zhang X. and Roy K., 1999. Design and synthesis of low power weighted random pattern generator considering peak power reduction, in *Proceedings of International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 148-156.

Zorian Y. and Wunderlinch H.J., 1997. Built-in Self Test (BIST): Synthesis of self testable systems, Kulwer Academic publishers.

Biographical notes

V.H. Prathyush received M. Tech. from Amrita Vishwa Vidyapeetham, Coimbatore, India. His research interest is Low Power VLSI models.

K. Somasundaram received a M.Sc. degree in Mathematics from St, Joseph's College, Trichy, India, in 1995 and a doctoral degree in Mathematics from Bharathiar University, Coimbatore, India, in 2003. He completed his post doctoral research fellowship from Turku Centre for Computer Science, University of Turku, Finland, in 2009. Currently he is an Associate Professor in the Department of Mathematics, Amrita Vishwa Vidyapeetham, Coimbatore, India. His research interests include Linear Algebra, Graph Theory and Network on Chip.

Received September 2010
Accepted December 2010
Final acceptance in revised form December 2010