System-on-Chip Test-time and Scan-power Minimization Integrating Core and Interconnect Testing

Gautam Das, Santanu Chattopadhyay and Haripada Bhaumik

Indian Institute of Technology, Kharagpur, India

Summary
This paper presents a Genetic Algorithm (GA) based strategy to solve the problem of System-on-Chip testing. It addresses the issues like core and interconnect testing, while most of the previous works reported in the literature takes care of core testing alone. The scheduling results produced show the trade-off between the testing time and power dissipated s while shifting the test patterns and responses through scan chains. This provides a wide range of choice for the designer to select a suitable test architecture.

Key words: SoC testing, Core, Interconnect test, Scan-power, Power optimization

1. Introduction

The integration of a complete system, which until recently consisted of multiple ICs on a PCB, onto one chip is termed as System-on-Chip (SoC) that uses embedded reusable cores. As the technology of microprocessor design and manufacture advances, more and more transistors can be placed on a silicon chip. This continuous increase in the design complexity poses a number of challenges to the system integrators while incorporating the test methodologies. Since cores in an SoC are not directly accessible via chip inputs and outputs, special access mechanisms are required to test them at system level, also known as *Test Access Mechanisms (TAMs).* It is used to deliver the test stimuli from the test source to cores and also to deliver responses from cores to the sink. The efficiency of a TAM depends on to what extent it can reduce the testing time, that is, time to test all cores in the SoC. Optimized architectures are needed to test the System-on-Chip in a cost-effective manner. Apart from the testing of the cores, the interconnects between them also need to be tested. This essentially means some input pattern to be applied at the origin of the interconnect and the value be checked at the other end. A number of interconnects can be tested in parallel, if the test resources are available. Thus, to reduce the total testing time for the chip, it is necessary that we consider the core testing and interconnect testing in an integrated fashion.

Another important issue during testing is the test power consumption. Though constrained scheduling to match the power budget has been proposed, another important component of power reduction is that of the scan chains. As a number of cores are put on a particular TAM, their test patterns will pass through the wrappers scan cells. This is particularly true for wrappers designed without any by-pass mechanism. Thus, the order in which the cores are placed on a TAM determines the switching and the associated power consumptions.

The integrated wrapper/TAM co-optimization and test scheduling problem that we address in this paper is as follows. ***Determine (i) the number of TAMs for the SoC, (ii) a partition of the total TAM width among this number of TAMs, (iii) an assignment of cores to the TAMs of different widths, (iv) a wrapper design for each core such that the SoC testing time is minimized, and, (v) an order of cores assigned to a TAM bus such that switching activity on the bus during testing is minimized.***

Section 2 presents the survey of previous works, Section 3 presents the details of core and interconnect testing, Section 4 discusses about scan-power estimation methods. Section 5 presents the Genetic Algorithm (GA) based solution to the problem, Section 6 presents the experimental results.

2. Prior work

In [13], test scheduling has been modeled as a combinatorial optimization problem of selecting a test set for each core from a set of tests and schedule them in order to minimize the test time. However, a single test bus is available for testing, and external testing can be done for only one core at a time. Test planning, that is, the partitioning of TAM and scheduling the tests has been discussed in [10]. A $O(n^3)$ algorithm has been proposed for test plan by reducing it to the minimum weight perfect bipartite graph matching problem, where n is the number of tests to be scheduled. Here, the experimental results have been presented for SoC having 10 cores only. Simulated Annealing technique has been used in [11] to propose an integrated solution to TAM design and test scheduling. Integrated TAM design and test scheduling has also been attempted in [1,2]. However, the problem of optimizing test bus widths and arbitrating contention among cores for test width is not addressed. The relationship between testing time and TAM widths using

ILP has been examined in [3,5]. The first integrated method for wrapper/TAM co-optimization has been proposed in [6]. TAM optimization is carried out by enumerating over the different partitions of TAM width as well as over the number of TAMs on the SoC. Integer Linear Programming (ILP) was used to calculate the optimal core assignment and resulting testing time for each partition. A drawback of this approach is that the wrapper/TAM designs considered in [6] are limited to small number of TAMs in order to maintain feasible compute time. The methods in [6] are therefore inadequate for large industrial SoCs. The work in [6] has been improved in [7] to include a heuristic method for core assignment. This heuristic core assignment approach forms a part of the TAM optimization method presented in [9]. In this case, a TAM optimization framework based on Lagrange multiplier has been presented. It uses an iterative procedure to obtain the optimal partition widths assuming that the total number of TAMs and the core assignment are given. In [16] a genetic algorithm based approach has been presented to solve the problem of optimal assignment of cores to buses, given the total number of cores, the total number of TAMs, and the partition widths of the TAMs. It also presents a genetic algorithm based approach to get the optimal distribution of total test width among the given total number of TAMs. The results are shown for two hypothetical but non-trivial SoCs. However, it does not provide any solution to the TAM enumeration problem. Recently a number of works have been reported based on rectangle packing that does not partition the TAM explicitly with the intention of minimizing the idle times for the TAMs [4]. But none of the works reported in the literature so far address the problem of interconnect testing, and scan power minimization by core reordering. This paper presents a solution to this integrated problem and shows the possible test time and scan power trade-offs.

3. Testing SoC

One of the major challenges in the system chip realization process is the integration and co-ordination of the on-chip test and diagnosis capabilities. The system chip test is a single composite test. This test is comprised of

- Individual internal tests for each core on-chip
- Test of interconnects between cores and
- The UDL test

3.1 Core Testing

Core testing is the testing of each core present in SoC. To test a core in a SoC, test stimuli for this core must be transmitted right from the chip inputs to the core. Test Access Mechanisms (TAMs) delivers pre-computed test sequences to the cores on the SOC, while Test Wrapper translates these test sequences into patterns that can be applied directly to the. Assuming Test Rail method of TAM, if core *i* is assigned to the test bus *j*, time to test this core is given by,

$$T_i\,(\omega_j) = (1 + max\,(S_i, S_0))\,*\,p_i + min\,(S_i, S_0)$$

Where p_i is the number of test patterns for core *i*, $S_i\,(S_0)$ is the length of the longest wrapper scan-in (scan-out) chain for the core, and ω_j is the width of the test bus *j*.

Schedule-Core Testing

Different test busses can be used simultaneously for delivering test data to the cores, while the cores assigned to the same test rail are to be tested sequentially. Total core testing time for a test rail is the sum of the testing times of the cores assigned to it. The total time to test all cores in the SoC is the maximum of the times taken among all test rails.

Let x_{ij} be a variable defined as follows:

$$x_{ij} = 1 \text{ if core } i \text{ is assigned to bus } j$$
$$= 0 \text{ otherwise}$$

Now, the total testing time needed to test all the cores in the system is

$$C = max_j\,\{\,\Sigma\, T_i\,(\omega_j)\,*\,x_{ij}\,\},\ 1 \le i \le N_c \text{ and } 1 \le j \le N_b$$

Where N_c and N_b are the number of cores and number of test rails respectively, in the system.

3.2 Interconnect Testing

Interconnect testing of a SoC is the testing of all interconnects present in that SoC. Test patterns for the interconnection-under-test are delivered to the source core by its test rail and the destination core delivers the received responses to the test rail assigned to it. To test an interconnection of width *n*, while guaranteeing the complete diagnosis of multiple interconnection faults, $(n+1)$ test patterns are required [17]. The width of each test pattern is equal to the width of the interconnection to be tested. Two cycles are required to test each interconnect, one cycle for the source core and the other for destination core. Serialization is required, when the test rail width is less than the test pattern width (i.e. the width of the interconnection to be tested).

Let

T = testing time for interconnect between core *i* and core *j*
Φ = width of test pattern = width of the interconnect t_i
Wt_i = width of the test rail assigned to core *i*
$W = min\,\{\,Wt_i\,,\,Wt_j\}$
P = number of test patterns for the interconnect
$= \Phi + 1$

Then,

$$T = 2{*}p \text{ cycles if } W \ge \Phi$$
$$= \{2+2{*}\,(\Phi - W)\}{*}\,p \text{ cycles otherwise}$$

This is because, when $W < \Phi$, first W bits of the test patterns can be delivered in parallel while the rest $(\Phi - W)$ bits are delivered serially.

Schedule-Interconnect Testing
Let the tuples $< C_1, C_2 >$ and $< C_3, C_4 >$ denote two interconnects, one from core C_1 to C_2 and the other from core C_3 to C_4 . These two interconnects can be tested in parallel if, test-rail (C_1) ≠ test-rail (C_3) and test-rail (C_2) ≠ test-rail (C_4). Now, we construct a graph where each interconnect is represented by a vertex and there is an edge between two vertices if the corresponding two interconnects can be tested in parallel. If the resultant graph is a clique, all the interconnections can be tested in parallel. But, in general, this will not be the case. We try to minimize the testing time by finding a *minimum clique cover* (partitioning the vertex set of a graph into minimum number of subsets, so that each subset is a clique).
The interconnections are now partitioned into subsets (corresponding to the cliques) and the interconnections in a subset can be tested simultaneously. Testing time of a subset is the maximum of the testing times for the interconnections in it. All such subsets are tested sequentially. Thus the total interconnect testing time is equal to the sum of the times to test each subset. The *minimum clique cover* problem is known to be *NP-complete*.
Let $G(V, E)$ denote a graph, where V is the set of vertices and E is the set of edges. The problem is to partition the vertex set into minimal number of sets such that each node belongs to exactly one set. We have used a modified version of *left-edge algorithm* [18] that solves the clique partitioning problem in $O(n\log n)$ time, using a heuristic approach.

4. Power Dissipation

Excessive switching activity during testing can cause average power dissipation and peak power during test to be much higher than that during normal operation. This obviously can cause damage to the SoC. A single test rail provides access to one or more cores. Testing the cores in a rail must be done sequentially. To test the second core on the bus, all the test data for it must pass through the wrapper scan chain of the first core. In general, if we have n cores on a Rail, to test the n^{th} core, all the test data for this core must be passed through the wrapper scan chains of the cores $1, 2..., n-1$. So, the switching activity, while testing, can be reduced by reordering the position of the cores in the Rail.
To estimate switching activity, *weighted transition* metric has been introduced [19]. *Weighted transition* metric models the fact that the measure of switching activity depends not only on the number of transitions in it but also on their relative positions. For example, consider the test vector $b_1b_2b_3b_4$= 1001 where b_4 is scanned in first and b_1 last. This vector has two transitions - first between b_1 and b_2, and the second between b_3 and b_4. During the scan-in of this vector, the first transition will occur only once but the second transition will be carried throughout the scan chain. Hence the first transition results in less switching activity and thus has less weight than the second. More formally, the number of weighted transitions in a test vector or an output response is given by,
Weighted_Transitions = ∑ (Size_of_scan_chain – Position_of_Transition).
Note that, the intrinsic value of position changes depending upon whether one considers a test vector or an output response. For example, consider the scan vector $b_1b_2b_3b_4 = 0001$. In the case where this is a test vector, the position of the transition between b_3 and b_4 is 1 and weight is (4 – 1) = 3, whereas, when this is an output response, position of the given transition is 3 and the weight is (4 – 3) = 1. This is so because during scan-in, b_4 is scanned in first and during scan-out, b_4 is scanned out first.
Apart from the scan-in transitions and scan-out transitions there is one more kind of transition. This transition occurs when the first bit of the test vector differs from the last bit of the previous output response. In this case, the transition propagates through the entire scan chain, and the weight assigned to it is equal to the size of the scan chain. Thus the total power consumed during scan testing is comprised of scan-in power, scan-out power and power consumed due to the above mentioned transitions.
For example, consider the test sequence shown in Figure 1, which is composed of three test vectors and the corresponding output responses. The scan chain has four flip-flops and hence scan vectors are four bit long. The number of weighted transitions for test vector V1 is (4-3) + (4-1) = 4, and that for output response R1 is (4-1) + (4-2) = 5. For V2, R2 and V3, R3 these are 6, 3 and 1, 5 respectively. In addition, a transition will propagate through the entire scan chain when the first bit of test vector V2 will be scanned in. The weight associated to this transition is 4, which corresponds to the size of the scan chain. Assuming an initial state of 0000 for the scan flip-flops, the total number of weighted transitions produced by the test sequence is (4+5+6+3+1+5) + (4) =28.

Next section discusses about Genetic Algorithmic (GA) based approach to the Test Access Architecture design and minimization of scan-in, scan-out power.

5. GA Formulation

We now describe a genetic algorithm based approach to solve the integrated problem efficiently. Genetic

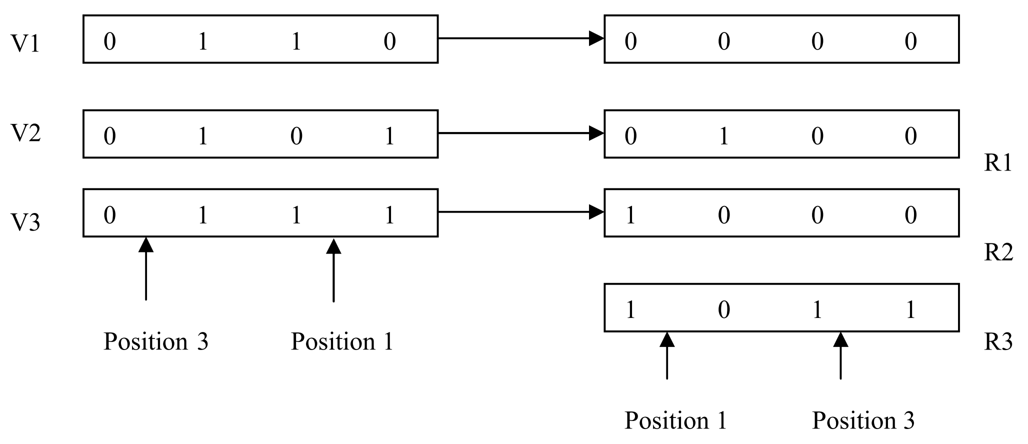

Figure 1: Weighted Transition Metric

algorithms [12, 14] are stochastic optimization search algorithms based on the mechanics of natural selection and natural genetics. The algorithm starts with an initial population usually consisting of a set of randomly generated solutions. Depending upon a reproductive plan, the chromosomes undergo evolution for a number of generations. The reproductive plan consists of applying genetic operators, crossing over two parent chromosomes participating in the operation. This helps in exploring the search space. In order to ensure that the search space explored is not closed under crossover, on the working population, another genetic operator, called mutation is applied to perturb one or more solutions. This operator brings variety within a finite population. After each generation, the chromosomes are evaluated using some fitness criteria. Depending upon the selection policy and the fitness values, the set of chromosomes for the next generation are selected. This evolution process is continued for a number of generations. Depending upon some terminating criteria (which may be the maximum number of generations the GA has run, or the maximum number of successive generations without cost improvement), the algorithm terminates. The best solution at that generation is accepted as the solution produced by GA. The genetic formulation of any problem involves the careful and efficient choice of the following.

- A proper encoding of the solutions to form chromosomes.
- To decide upon a crossover operator.
- To identify a proper mutation operator.
- A cost function measuring the fitness of the chromosomes in a population.

5.1 Solution Representation

A chromosome in our approach consists of four parts namely *enumeration part, assignment part, distribution part, and permutation (ordering) part. Enumeration part* represents the number of TAMs for the SoC, in binary format. *Assignment part* is the assignment of cores to test buses, which is an array of integers, with i^{th} element representing the bus number to which core i is assigned. *Distribution part* is the distribution of total bus width, an array of integers, with j^{th} element representing the width of bus j, such that the sum of these widths is equal to the total TAM width W. The *permutation part*, also an array of integers, represents an ordering of the cores assigned to test buses. These four parts of the chromosome form a solution to the given problem. Figure 2 shows an example chromosome for the above problem. Here W = 16 and number of cores N_c = 9. Cores 2, 3, 4, 5 and 9 are assigned to test bus 1, cores 1 and 4 are assigned to test bust 2 and finally cores 6, 7 and 8 are assigned to 3rd test bus. Bus width W(16) is divided into 7, 3 and 6. The number of TAMs is 3 (binary 0011).

5.2 Crossover

Two-point crossover is applied to the *enumeration part, assignment part* and *permutation part,* while, single point cross over is applied to the *distribution part* of the chromosome. The crossover works on individual parts as follows.

1. For the enumeration part, portion between two points of parent1 are copied to child1, while the similar portion between two points from parent2 are copied to child2. Remaining bits are generated randomly.
2. For the assignment part, portion from the beginning to first crossover point of parent1 is copied to child1, while that portion from parent2 is copied to child2. Similarly, portion from second cross point to end of parent1 is copied to child1, and the portion from parent2 is copied into child2. The portion between the two cross points of the assignment part of child1 and child2 are generated randomly.

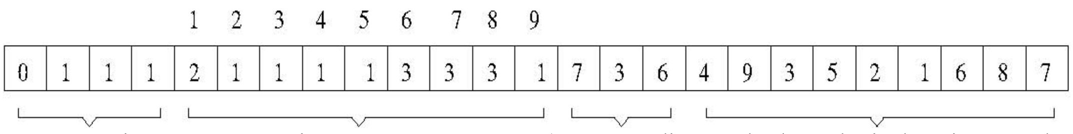

Fig 2: Example chromosome

3. For the distribution part, from the beginning to cross point of parent1 is copied to child1 and that from parent2 is copied to child2. Remaining locations are generated randomly.

4. For the permutation part, crossover is performed similar to the assignment part.

5.3 Mutation

Mutation on the *enumeration part, assignment part* and *permutation part* selects some random number of positions and changes their values. Mutation on the *distribution part*, selects a random number of elements and tries to modify them so that sum of the values on this part remains equals to *W*.

5.4 Fitness Measure

Measure of fitness of chromosomes is useful to rank the obtained solutions at a particular generation. Since our objective is to integrate both the testing time and the scan power consumption, we proceed as follows.

A chromosome gives the TAM distribution, allocation of cores on them and ordering of cores. The core testing time can now be evaluated as shown in Section 3.1. The interconnect testing time is next computed as shown in Section 3.2. Sum of these two gives the total testing time. Power consumption during scan shifting is proportional to the number of transitions occurring in the scan chain as we are shifting in the test data, or shifting out the response. This number can be calculated as discussed in Section 4.
The testing time (*T*) and weighted transitions (*W*) computed above are next combined together. For this purpose, the values are first normalized by dividing them by the maximum testing time and the maximum weighted transitions among all chromosomes in the initial population of the GA. Let, the normalized values for a particular chromosome be *TNORM* and *WNORM* respectively. We compute the fitness of the chromosome as,

$$F = \sqrt{(TNORM^2 + WNORM^2)}$$

That is, if we plot the solutions on a graph with axes *T* and *W*, a better fit chromosome will have lesser distance from the origin (corresponding to the hypothetical optimum solution with zero test time and zero transitions).

6. Experimental Results

In this section, we present our experimentation with the ITC'02 benchmarks [15] for SoCs. The benchmark files in their current format, do not contain the interconnect information. For the sake of our experimentation, we have generated various number of interconnects. For an SoC with *n* cores, we assume that at the most *n(n – 1) /2* possible interconnects can be there. We generate different test cases with different percentages of interconnects. For a particular interconnect density, the actual connections are generated randomly.

Another important information needed to compute scan chain transitions is the test patterns applied to a core and the corresponding responses. The benchmarks, though are having information about the total number of test patterns, the actual patterns are not specified, Hence, we have generated the specified number of input patterns and output responses randomly.

Fig 3 and 4 present the results for SoC d695 with 40% and 75% interconnects respectively. d695 has 10 cores. All chromosomes in the final generation have been plotted on a graph with X-axis representing the switching activity and the Y-axis representing the testing time. A solution is said to be a dominated one, if there exists some other solution with both higher test time and switching activity. Further, a dominated solution should not dominate any other solution. Thus, a dominated solution is a good one, as there does not exist any other solution with both lesser testing time and scan power. The dominated solutions have been marked in the figures. The numbers alongside the dominated solutions mark the number of TAMs for which the solution has been obtained. Fig 5 and 6 present the results for SoC g1023. Fig 7 and 8 correspond to SoC p93791 and Fig 9 is for the SoC p512505. Since all the dominated points are shown, the designer has a set of choices available with test time and scan power trade-off.

7. Conclusion

In this paper we have presented an integrated approach for core and interconnect testing. It combines the problem of scan power reduction with the test time reduction problem.

The designer is provided with a wide range of solutions to use from, for the design of test architecture.

References

[1] E. Larsson and Z. Peng. An integrated system-on-chip test framework. Proc. Design, Automation, and Test in Europe (DATE), pp. 138-144, 2001.
[2] M. Nourani and C. Papachristou. An ILP formulation to optimize test access mechanism in system-on-chip testing. Proc. Int. Test Conf., pp. 902-910, 2000.
[3] K. Chakrabarty. Design of system-on-a-chip test access architectures using integer linear programming. Proc. VLSI Test Symp., pp. 127-134, 2000.
[4] V. Iyengar, K. Chakrabarty, and E.J. Marinissen. On using rectangle packing for SOC wrapper/TAMco-optimization. Proc. VLSI Test Symp., pp. 253-258, 2002.
[5] K. Chakrabarty. Optimal test access architectures for system-on-a-chip. ACM Trans. Design Automation of Electronic Systems, vol. 6, pp. 26-49, January 2001.
[6] V. Iyengar, K. Chakrabarty, and E. J. Marinissen. Test wrapper and test access mechanism co-optimization for system-on-chip. Journal of Electronic Testing: Theory and Applications, 18(2):213-230, 2002.
[7] V. Iyengar, K. Chakrabarty, and E.J. Marinissen. Efficient wrapper/TAM co-optimization for large SOCs. Proc. Design Automation and Test in Europe (DATE) Conf., pp. 491-498, 2002.
[8] V. Iyengar, K. Chakrabarty, and E.J. Marinissen. Wrapper/TAM Co-optimization, Constraint-Driven Test Scheduling, and Tester Data Volume Reduction for SOCs. Proc. IEEE/ACM Design Automation Conference, pp. 685-690, 2002.
[9] A.Sehgal, V. Iyenger, M. Krasniewski and K. Chakrabarty. Test Cost Reduction for SOCs Using Virtual TAMs and Lagrange Multipliers.Proc. IEEE/ACM Design Automation Conference, pp. 738-743, 2003.
[10] Sandeep Korane. On Test Scheduling for core based SOCs. In proceedings of IEEE Asia South Pacific Design Conference, pages 505-510, Bangalore, India, January 2002.
[11] Erik Larsson, Zebo Peng and Gunnar Carlsson. The Design and Optimization of SOC Test Solutions. www.ida.liu.se/~erila/ICCAD01.pdf.
[12] J. H. Holland. Adaptation in Natural and Articial Systems. Ann Arbor MI: University of Michigan press, 1975.
[13] M. Sugihara, H. Date and H. Yasuura. A novel test methodology for core-based system LSIs and a testing time minimization problem. Proc. International Test Conference, pp. 465-472, 1998.
[14] David E. Goldberg, John H. Holland. Genetic Algorithms and Machine Learning. Machine Learning 3: 95-99, 1988.
[15] ITC'02 SOC Test Benchmarks. http://www.extra.research.philips.com/itcsocbenchm/
[16] Ebadi, Z. S. and Ivanov, A. Design of an optimal test access architecture using a genetic algorithm. In Proceedings of IEEE Asian Test Symposium (ATS) (Kyoto, Japan, Nov.), 205--210, 2001.
[17] Park S., A New Complete Diagnosis Patterns for Wiring Interconnects. In Proceedings of 33rd DAC, 203--208, 1996.
[18] Hashimoto, A. and Stevens, J., Wire Routing by Optimizing Channel Assignment within Large Apertures. In Proceedings of 8th DAC, 155--169, 1971.
[19] Oruganti, R., Sankaralingam, R. and Touba, N., Static Compaction Techniques to Control Scan Vector Power Dissipation. In Proceedings of VTS, 35--42, 2000.

Author’s biographies

Mr. Gautam Das is currently an Assistant Professor with the Siliguri Institute of Technology, India, in the ECE department. He did his B.Sc. (Physics honours) from Calcutta University in 1995, received his B.Tech and M.Tech degrees in Electronics & Telecommunication Engineering from Institute of Radiophysics & Electronics under Calcutta University, in 1998 and 2000 respectively. He is currently pursuing his PhD in the North Bengal University. His research interests include digital design, system-on-chip design and test.

Dr. Santanu Chattopadhyay is currently an Assoiate Professor with the Dept. of Electronics & Electrical Communication Engg., Indian Institute of Technology, Kharagpur. He did his B.E. from Bengal Engineering College, Sibpur (Calcutta University) in 1990 in Computer Science & Technology. He received his M.Tech in Computer and Information Technology and PhD in Computer Science & Engineering from Indian Institute of Technology in 1992 and 1996 respectively. He has published more than 110 research papers in international journals and conferences, co-authored a book published by the IEEE Computer Society Press, USA. His research interests include low power system design & test, System-on-chip and Network-on-Chip design & test.

Prof. Haripada Bhaumik obtained his B.E. degree from Jadavpur University, Kolkata in 1962, M.E. from IISc Bangalore in 1965 and PhD from Roorkee University, India. He has served as faculty and other capacities in Rorkee University, also served as Principal in Jalpaiguri Government Engineering College, West Bengal, India, and Regional Engineering College (now NIT) Silchar.

Appendix

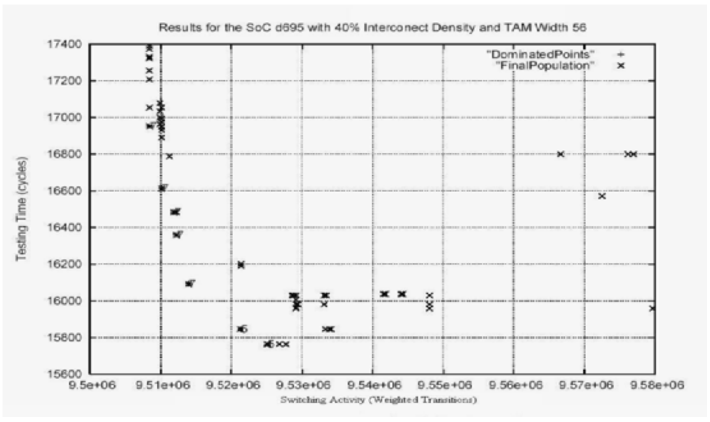

Fig 3: Results for SoC d695

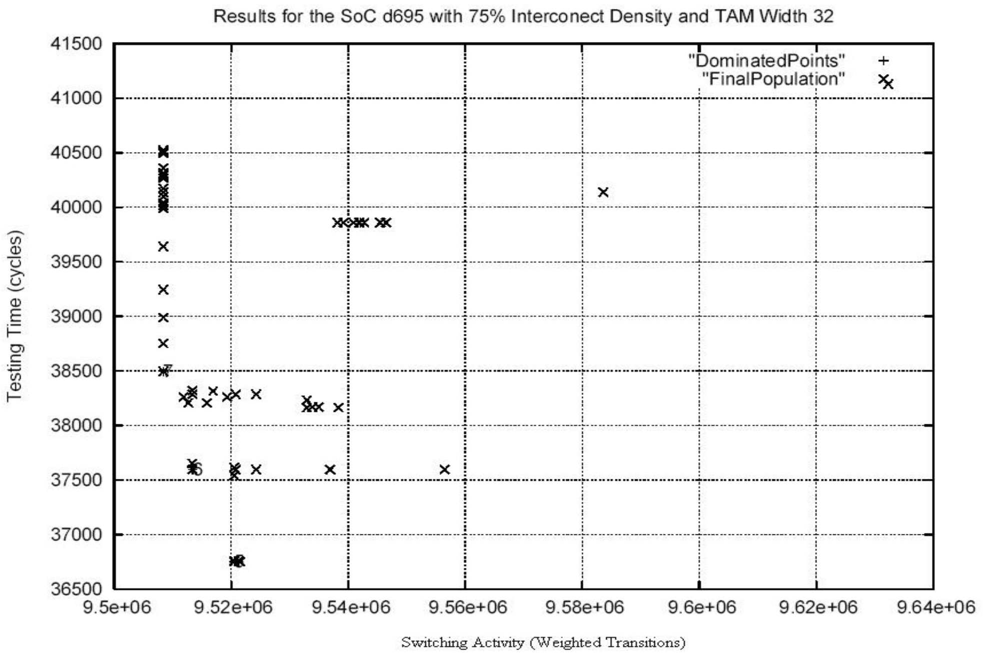

Fig 4: Results for SoC d695

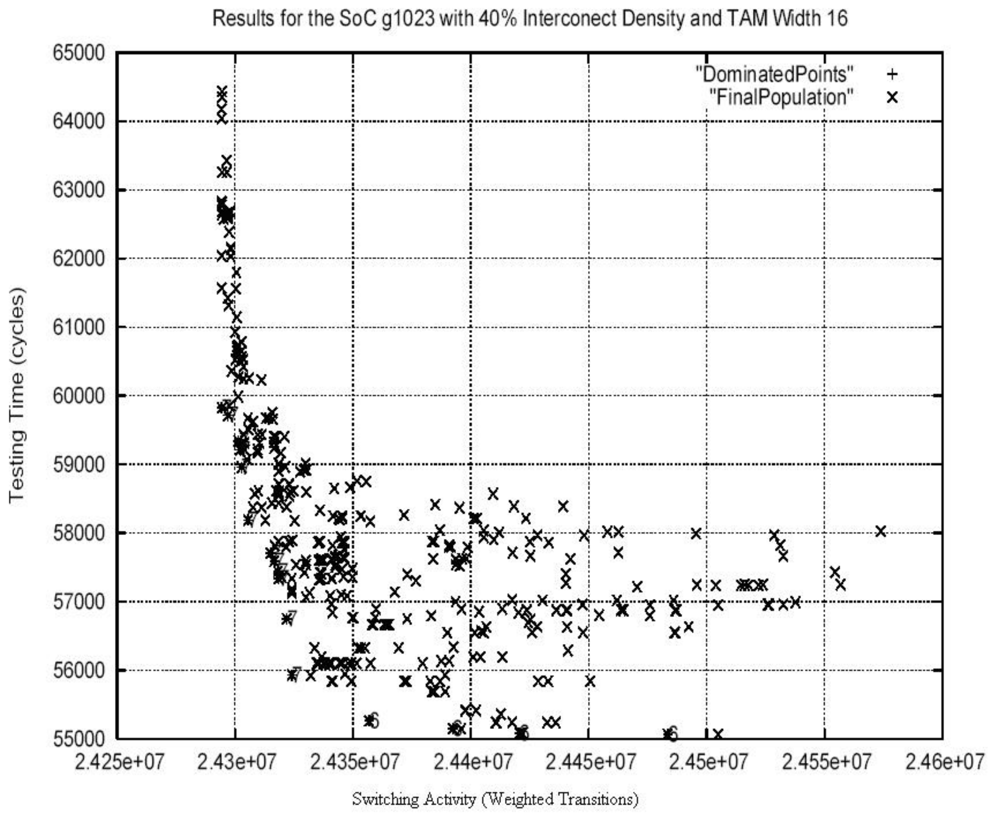

Fig 5: Results for SoC g1023

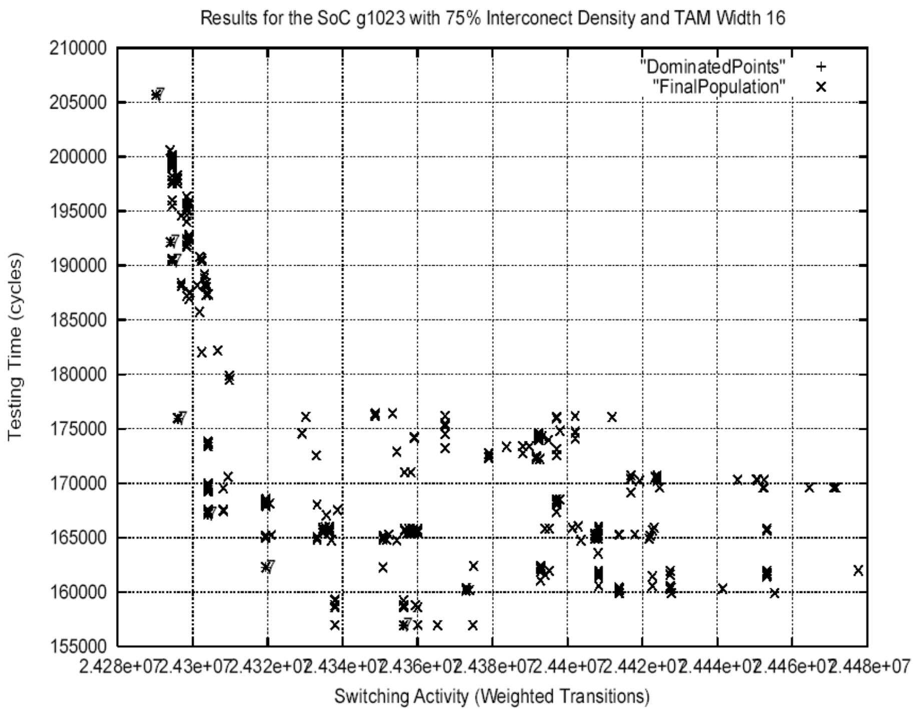

Fig 6: Results for SoC p93791

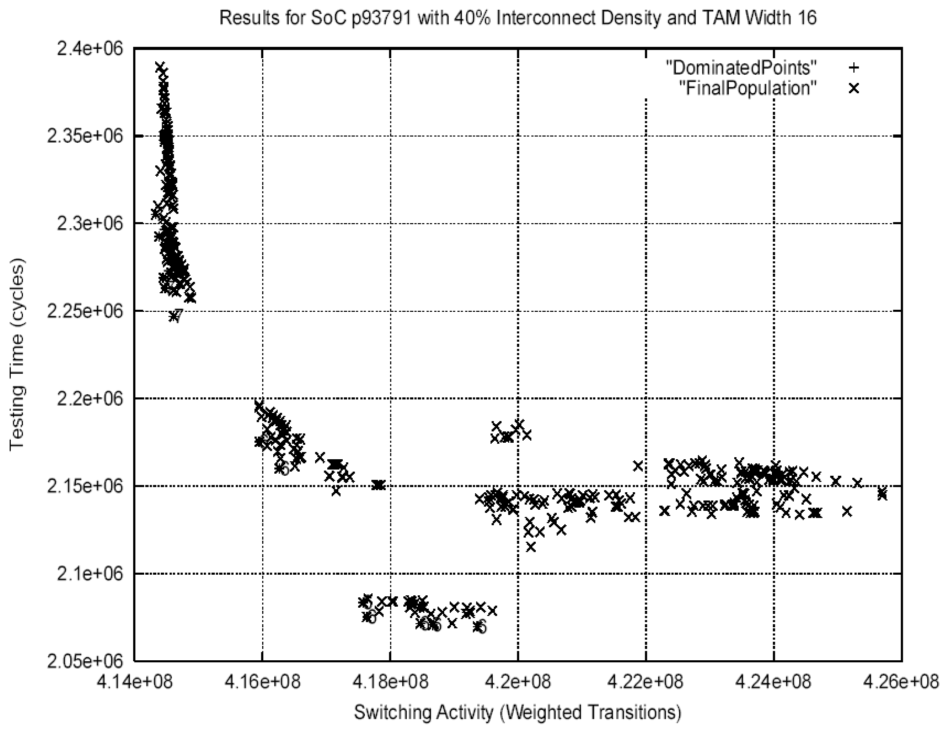

Fig 7: Results for SoC p93791

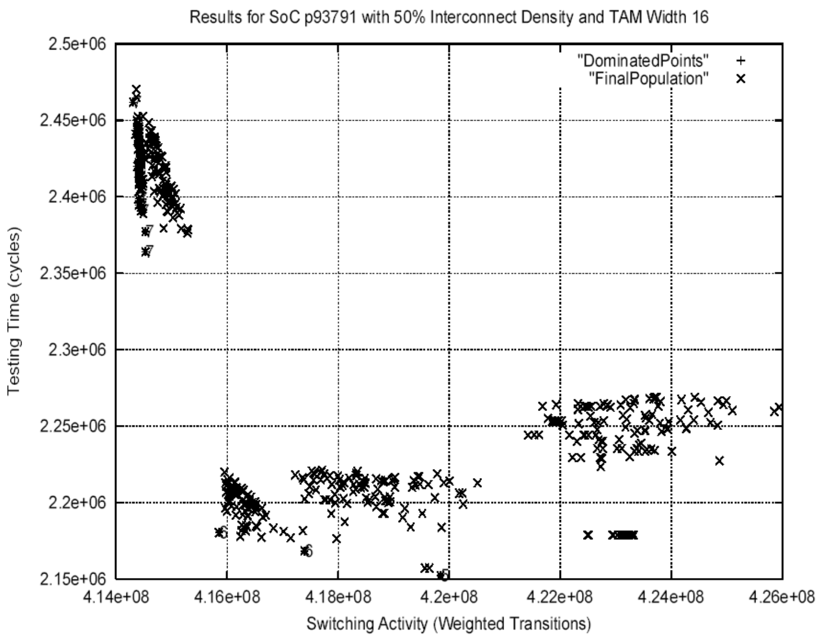

Fig 8: Results for SoC t512505