GA BASED TEST SCHEDULING UNDER POWER CONSTRAINTS

Ing. Jaroslav ŠKARVADA, Doctoral Degree Programme (2)
Dept. of Computer Systems, FIT, BUT
E-mail: skarvada@fit.vutbr.cz

Supervised by: Doc. Zdeněk Kotásek

ABSTRACT

This paper deals with test scheduling under power constraints. An approach based on genetic algorithm operating on Test Application Conflict Graph is presented. The main goal of the method is to minimize test application time with considering structural resource allocation conflicts and to ensure that test application schedule does not exceed chip power limits. The proposed method was implemented using C++, experimental results with ITC’02 SOC benchmark suite are presented in the paper together with the perspectives for the future research.

1 INTRODUCTION

Integrated circuits continue to grow in size and complexity. It can be stated that reusable cores are more often used for system on chip (SOC) design. As for any other electronic system, it is very important every core in the design is testable and equipped with predefined test vectors. When building testable system from these components, test scheduling becomes one of main issues in the overall system integration.

Basically, test scheduling is primarily targeted for efficient usage of resources available for the test and for minimization of test application time. But if power dissipation is not taken into account it may happen that the overall chip power dissipation limit during the test application are exceeded. Since much of the power consumed by the circuit is dissipated as heat, the relationship between the test activity and the cooling capacity needs to be taken into account in order to avoid destructive test [8]. Using low power design methodologies such as [6] it is possible to reduce significantly power dissipation during normal functional operation of the system but in the test mode this is not so straightforward. As some experiments reveal, the chip under test can dissipate up to three times higher power during testing in comparison to normal functional operation [1], [5]. Greater power dissipation is caused by significantly higher switching activity (assuming CMOS technology because it is known as dominant fabrication technology for implementation of ICs that contain more than 105 transistors [5]) during testing because there is much more correlation between input patterns in functional mode than in the test mode where correlation between test vectors is decreased in order to reduce test application time.

Another source of additional power dissipation can be traced in scheduling concurrent test of units that don’t operate simultaneously in functional mode. The method proposed in this paper is primarily trying to minimize test application time as possible while not allowing the power dissipation to go over predefined level as a constraint.

Recent research results about test scheduling can be found in [1-4,9-11]. Chakrabarty showed that the test scheduling problem is equal to the open-shop scheduling [10] which is known to be NP complete and the use of heuristics are therefore justified. The optimal test schedule is obtained by using a mixed integer linear programming (MILP). Power constrained MILP model. Precedence or preemptive test scheduling has been presented in [3]. In [11] a technique based on a greedy algorithm for test parallelization under test power consumption is presented and it is shown how it can be used to find the optimal test time for the system under test. In the approach by Chou et al. [9] a resource graph is used to model the system, and from it, a test compatibility graph (TCG) is generated. The test schedule is obtained by solving the minimum cover table problem of the graph. Based on the TCG a method for test scheduling and Test Access Mechanism (TAM) optimization is presented in [4]. The method used Tabu search for solution space exploration.

2 BASIC DEFINITIONS

In our methodology, it is assumed that System Under Test (SUT) contains set of logic cores (blocks) SUT = {B1, B2,..., B_n} that must be all successfully tested in order to achieve faultless operation of the system. Each core B_1, ..., B_n has its own predefined test sequence. The set T is a set of all tests. The test t_i is modeled as a triplet $t_i = (R_i, time_i, pwr_i)$. R_i is set of resources needed by test t_i. That includes test generators and response evaluators as well as data lines and busses used for transports test vectors and responses needed for test t_i. Each test has also assigned its time duration $time_i$ (for instance in global clock cycles) and power dissipation pwr_i during the test. For cores with internal scan chains, it is also possible to reduce test application time and power dissipation by dividing long scan chains into several of shorter length and by extending test access mechanism (TAM) width.

The test resource conflict will occur when two or more tests are scheduled to test the same core or when using the same test resources to test different cores. These types of conflicts can be viewed as structural conflicts. In the case of very complex logical cores it may be possible that simultaneous test of certain two cores may lead to exceeding maximal predefined chip dissipation limit P_D. In this case we can determine another type of conflict. All these conflicts can be easily modeled by using TACG.

TACG is undirected graph $G = (V, E)$, where V is set of all vertices of G and E is set of all edges in G. Each vertex $v_i \in V$ represents a test $ti \in T$. So it is possible to find bijection b: $V \rightarrow$ T. There is an edge $\{v_i, v_j\} \in E$, $i \neq j$ if there exists a conflict between tests ti and tj. It can be written as:

$$(R_i \cap R_j \neq \emptyset) \vee (pwr_i + pwr_j > P_D) \wedge i \neq j \;\Rightarrow \{v_i, v_j\} \in E \tag{1}$$

Further it is necessary to find maximal independent sets of G (with consideration of power constraints) and than to solve vertex coverage problem by selecting only those sets whose combinations lead to shortest test time. It is also possible to transfer this problem to graph coloring problem.

It is well known fact that it is NP-complete to decide whether, for a given graph G and an integer k, there exist a k coloring of G [14]. The graph coloring problem is the problem of finding for a given graph G an optimal coloring that is a coloring with least possible number of colors equal to chromatic number of G. Because we are dealing with power constraint it must be also considered that for every color used, the sum of power dissipations of all vertices colored by same color does not exceed predefined chip dissipation limit P_D. In this case the number of colors used will directly determine the number of test phases and all vertices colored by the same color will be scheduled for simultaneous test.

3 PROPOSED METHOD

The block diagram of the proposed method is depicted in figure 1 and will be described later. As a core of proposed method the genetic algorithm (GA) is used.

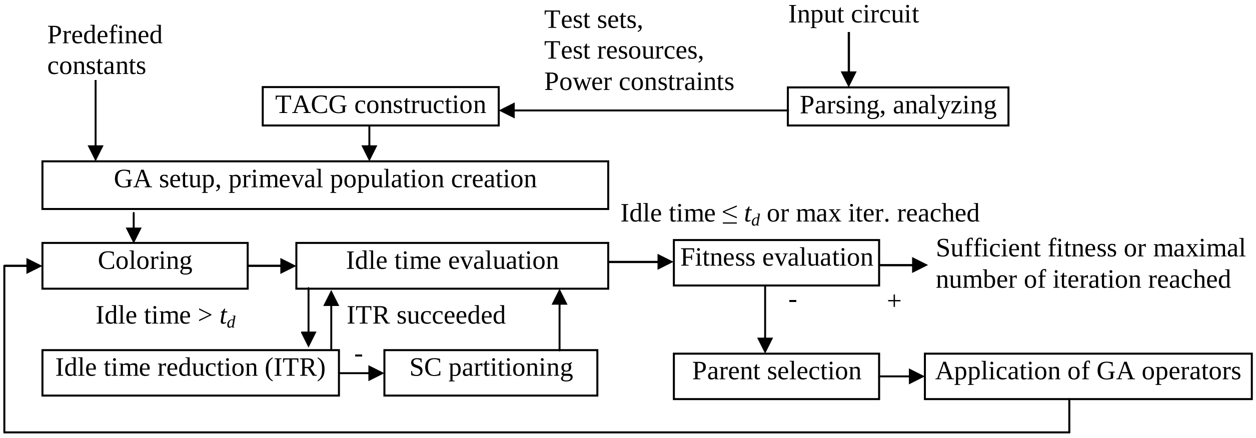

Fig. 1: *Proposed method*

The genetic approach is used to find test scheduling problem solution. It is a frequently discussed problem that genetic algorithms are not able to cope well with the graph coloring problem [12]. Anyway, some challenging attempts still exist such as in [12, 13]. Together with specific variation of the problem (such as in our case the added power constraints and test phase modification) it is possible to get usable results. The proposed method was based on successful general approach from [13] with modifications to our specific case and added constraints. The main goal of the algorithm used is to find suitable ordering of vertices and then coloring these vertices due to proposed ordering in order to find proper test schedule. A chromosome represents an ordering of TACG vertices. If the graph has n vertices, the chromosome will be a vector:

$$chrom = (v_1, v_2, ..., v_n),\ \ v_i \in V, i \in \{1, ... , n\} \qquad (2)$$

The chromosome can be viewed as a permutation of vertices, so the order of vertices in chromosome is important and the chromosomes $(v_1, v_2, ..., v_n)$ and $(v_2, v_1, ..., v_n)$ are two different chromosomes. One crossover and two mutation operators are used. Each mutation operator has its own probability of application. Chromosomes (parents) for crossover are selected by Roulette wheel selection algorithm according to their fitness. Every time when new population is generated the fitness evaluation is performed. Each chromosome is then colored by using naive coloring algorithm due to proposed vertex ordering. The algorithm tries to color sequentially as many successive vertices as possible, starting from position 1. If it is impossible to color successive vertex with the same color as previous one or power constraints would be violated, the new color is chosen. After that the fitness is assessed. The fitness value is based on the evaluation of the time needed to apply the test. Longer test means lower fitness. The test duration is calculated as:

$$\Sigma_{\mathrm{i = 1...n}} \max(\{t_j \mid c(j) = i\}) - \Sigma \text{ eliminated } t_{idle} \tag{3}$$

4 EXPERIMENTAL RESULTS

Proposed method was implemented in C++ language, STL library was used. As C++ compiler, GCC 4.0.2 was used. PC with Pentium 4 processor at 2.0 GHz and 768 MB DDR RAM running Linux OS was used to evaluate the methodology. The experiments were carried on ITC'02 SOC benchmark set [7].

SOC	u226	d281	D695	g1023	p22810
Cores	9	8	10	14	29
Tests	9	15	10	14	30
Inputs	163	1523	584	1689	1999
Outputs	160	1408	1261	1898	1462
Bidirs	0	0	0	0	822
Scans	20	34	137	35	196
Patterns	5148569	8818	881	2349	25112
P_D	871433	78838	134514	2232	527749
$t_{testseq}$	5152582	102015	35946	43951	496224
$t_{testpar}$	2683	1940	2508	1680	8602
t_{CPU} [s]	350	358	284	422	1252

Tab. 1: *Results for selected SOCs*

Table 1 shows results for selected SOCs. In the table *Cores* represents number of cores in SOC, P_D represents overall power dissipation limit obtained from formula (6), $t_{testseq}$ represents time of sequential test, $t_{testpar}$ represents time of scheduled test, t_{CPU} is time of overall computation. Population size was set to 100 and maximal number of generations was set to 1000. The test time for g1023 is so short because this SOC has small number of test patterns. Longest computation time for p22810 SOC is because the solution space for this SOC is huge.

5 CONCLUSIONS

The research and experiments were carried in order to become acquainted with the problems from the area of test scheduling and to try to implement my own test scheduling method operating on TACG resource conflict model. This model is extended by power constraint limitations and analyzed with help of genetic algorithm. Test relocation technique is used for reducing implicit idle time. When idle time ratio is still higher than predefined value, scan chains partitioning and TAM width modification for cores with scan chains are used. The output of the algorithm is a test schedule with shortest recognized test application time that does not break predefined structural and power constraints. It must be noted that the implemented software is compatible with recent framework developed in our department. The next goal is to transfer this method to RT level and try to make experiments with real designs.

ACKNOWLEDGEMENTS

The paper has been prepared as a part of the solution of GAČR project No. 102/04/0737, Modern Methods of Digital Systems Design and with the support of the Czech Ministry of Education – FRVŠ grant No. 3383/2006/G1.

REFERENCES

[1] Schuele, T., Stroele, A. P.: Test Scheduling for Minimal Energy Consumption under Power Constraints. 19th IEEE VLS Test Symposium, pp. 312-318, 2001

[2] Rosinger, P. M., Al-Hashimi, B. M., Nicolici, N.: Power Constrained Test Scheduling Using Power Profile Manipulation, In: Proceedings of Intl. Symposium on Circuits and Systems 2001 V, ISCAS 2001, pp. 251-254, 2001

[3] Iyengar, V., Chakrabarty, K.: Precedence-Based, Preemptive, and Power-Constrained Test Scheduling for System-on-a-Chip, In: Proceedings of the 19th IEEE VLSI Test Symposium (VTS'01), pp. 368-374, 2001

[4] Su, C., Wu, C.: Graph-based power constrained test scheduling for SOC. In. proc. of IEEE design and diagnostics of electronic circuits and system workshop, pp. 61-68, 2002, ISBN 0-2142-0944-0

[5] Nicolici, N., Al-Hashimi, B. M.: Power-Constrained Testing of VLSI Circuits. Kluwer Academic Publishers, Boston, 2003, ISBN 1-4020-7235-X

[6] Schmitz, M. T., Al-Hashimi, B. M., Eles, P.: System-Level Design Techniques for Energy-Efficient Embedded Systems. Kluwer A. P., Boston 2004, ISBN 1-4020-7750-5

[7] Marinissen, E. J., Iyengar, V., Chakrabarty, K.: A Set of Benchmarks for Modular Testing of SOCs. In Proc. of IEEE International Test Conference, Baltimore, pp. 519-528, 2002

[8] Zorian, Y.: A distributed BIST control scheme for complex VLSI devices. In Proceedings of 11th IEEE VLSI Test Symposium, pp. 4-9, 1993

[9] Chou, R. M., Saluja, K. K., Agrawal, V. D.: Scheduling tests for VLSI systems under power constraints. IEEE Transactions on VLSI systems, vol. 5, iss. 2, pp. 175-178, 1997

[10] Chakrabarty, K.: Test scheduling for core based systems using mixed integer linear programming. IEEE Transactions on VLSI systems, vol. 19, iss. 10, pp. 1163-1174, 2002

[11] Larsson, E., Peng, Z.: System-on-chip test parallelization under power constraints, in Proc. of IEEE European Test Workshop, pp. 281-283, 2001

[12] Barbosa, V., Assis, C., Nascimento, J.: Two Novel Evolutionary Formulations of the Graph Coloring Problem. Journal: CoRR: Neural and Evolutionary Computing, pp. 41-63, 2003