Hybrid BIST Time Minimization for Core-Based Systems with STUMPS Architecture

Gert Jervan, Petru Eles, Zebo Peng
Embedded Systems Laboratory (ESLAB)
Linköping University, Sweden
{gerje, petel, zebpe}@ida.liu.se

Raimund Ubar, Maksim Jenihhin
Department of Computer Engineering
Tallinn Technical University, Estonia
{raiub, maksim}@pld.ttu.ee

Abstract[1]

This paper presents a solution to the test time minimization problem for core-based systems that contain sequential cores with STUMPS architecture. We assume a hybrid BIST approach, where a test set is assembled, for each core, from pseudorandom test patterns that are generated online, and deterministic test patterns that are generated off-line and stored in the system. We propose a methodology to find the optimal combination of pseudorandom and deterministic test sets of the whole system, consisting of multiple cores, under given memory constraints, so that the total test time is minimized. Our approach employs a fast estimation methodology in order to avoid exhaustive search and to speed-up the calculation process. Experimental results have shown the efficiency of the algorithm to find near optimal solutions.

1. Introduction

Testing of systems-on-chip (SoC) is a problematic and time consuming task, mainly due to their complexity and high integration density [1]. To test the individual cores of a SoC the test pattern source and sink have to be available together with an appropriate test access mechanism (TAM) [2]. Due to the rapid increase of chip speed and test data volume, the traditional Automatic Test Equipment (ATE) based solution is becoming increasingly expensive and inaccurate. Therefore, in order to apply at-speed tests and to keep the test costs under control, built-in self-test (BIST) solutions are becoming a mainstream technology for testing such complex systems.

BIST for digital logic (logic BIST) uses mostly pseudorandom tests. Due to several reasons, like very long test sequences, and random pattern resistant faults, this approach may not always be efficient. One solution to the problem is to complement pseudorandom test patterns with deterministic test patterns, applied from the on-chip memory or, in special situations, from the ATE. This approach is usually referred to as hybrid BIST [3].

One of the important parameters influencing the efficiency of a hybrid BIST approach is the ratio of pseudorandom and deterministic test patterns in the final test set. As the amount of resources on the chip is limited, the final test set has to be designed in such a way that the deterministic patterns fit into the on-chip memory. At the same time the testing time must be minimized in order to reduce testing cost and time-to-market.

[1] This work has been supported by the Swedish Foundation for Strategic Research (SSF) under the Strategic Integrated Electronic Systems Research (STRINGENT) program, by the EC project EVIKINGS (IST-2001-37592) and by the Estonian Science Foundation Grants 4300 and 5649.

There exists extensive work for testing core-based systems. The main emphasis has been so far on test scheduling, TAM design and testability analysis. The earlier test scheduling work has had the objective to determine start times for each test such that the total test application time is minimized. This assumes a fixed set of tests and test resources together with a test access architecture. Some approaches can also take into account test conflicts and different constraints, e.g. power [4] - [11]. Sugihara et al. [8] have addressed the problem of selecting a test set for each core from a set of pre-determined test sets provided by the core vendor and scheduling these tests in order to minimize the testing time. Although this approach can find the best possible selection of tests from a given set, it doesn't provide a mechanism for finding the test set in first place, which is the objective of this paper.

Our earlier work, [3], [12] and [13], has been concentrating on test cost calculation and hybrid BIST optimization for single-core designs. Recently we have proposed a methodology for test time minimization, under memory constraints, for multi-core systems, where only combinatorial cores were assumed [14]. We have proposed for this purpose a method for estimating the cost of the deterministic component in the hybrid test set. Based on this methodology, we have developed an iterative algorithm to minimize the total length of the hybrid BIST solution under given memory constraints. In this paper we will extend this approach to provide a methodology for sequential cores with full scan, where the memory constraints are not violated, the total test time is minimized, and maximum achievable fault coverage is guaranteed.

The rest of the paper is organized as follows. In Section 2 a hybrid BIST approach for sequential cores with STUMPS architecture is described and a general problem description is given. Section 3 is devoted to basic definitions, cost functions and problem formulation. Section 4 describes our test cost estimation methodology and the algorithm for test length minimization, based on our estimates is illustrated in Section 5. Finally, the experimental results are presented in Section 6, and Section 7 concludes the paper together with directions to the future work.

2. Hybrid BIST Architecture

In general a hybrid BIST approach combines two different types of tests. It starts with a pseudorandom test sequence of length L and continues with precomputed deterministic test patterns, stored in the system, in order to reach the desirable fault coverage. For off-line generation of the deterministic test patterns, arbitrary software test generators may be used, based on deterministic, random or genetic algorithms.

In a hybrid BIST technique the length of the pseudorandom test is an important parameter that determines the behavior of the whole test process. It is assumed here that for the hybrid BIST the best polynomial for the pseudorandom sequence generation will be chosen. By using the best polynomial, we can achieve the maximal fault coverage of the CUT. In most cases this means that we can achieve 100% fault coverage if we run the pseudorandom test long enough. With the hybrid BIST approach we terminate the pseudorandom test in the middle and remove the latter part of the pseudorandom sequence, which leads to lower fault coverage. The loss of fault coverage should be compensated by additional deterministic test patterns. In general a shorter pseudorandom test set implies a larger deterministic test set. This requires additional memory space, but at the same time, shortens the overall test process, since deterministic test vectors are more efficient in covering faults than the pseudorandom ones. A longer pseudorandom test, on the other hand, will lead to longer test application time with reduced memory requirements.

Therefore it is crucial to determine the optimal length of the pseudorandom test sequence, in order to minimize the total testing cost.

There are two widely used BIST schemes: test-per-clock and test-per-scan. Our earlier work concentrated on systems with combinatorial cores and therefore a test-per-clock scheme was used. In this paper our objective is to provide a solution to the test time minimization problem in case of sequential cores. As testing of sequential cores is very complex and development of efficient test pattern generation algorithm for sequential cores is outside the scope of this paper, it is assumed here that every core contains one or several scan paths (full scan). Therefore a test-per-scan scheme has to be used, and in this paper, for every individual core, the Self-Test Using MISR and Parallel Shift Register Sequence Generator (STUMPS) [15] architecture is assumed.

While every core has its own STUMPS architecture, at the system level we assume the following architecture: Every core's BIST logic is capable of producing a set of independent pseudorandom test patterns, i.e. the pseudorandom test sets for all the cores can be carried out simultaneously. The deterministic tests, on the other hand, can only be carried out for one core at a time, which means only one test access bus at the system level is needed. An example of a multi-core system, with such a test architecture is given in Figure 1.

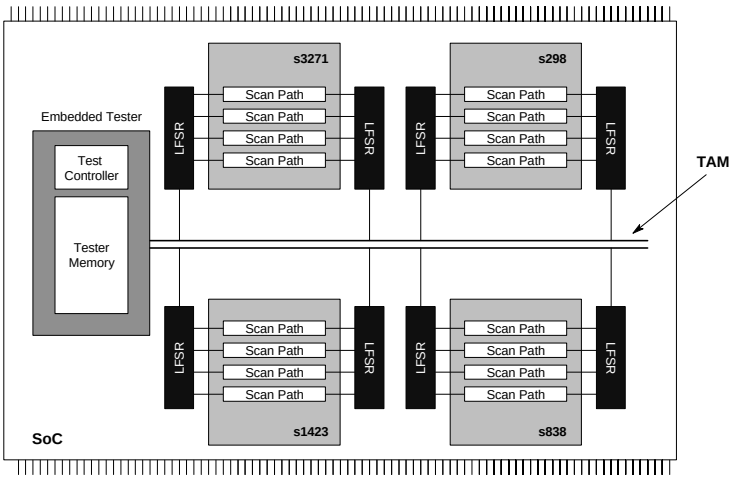

Figure 1. A core-based system example with the proposed test architecture

The example system given in Figure 1 consists of 4 cores (different ISCAS benchmarks). We have shown earlier that the solution where every individual core has the best possible combination between pseudorandom and deterministic patterns usually does not lead to the best system-level test solution. There are several problems:

- The total test length of the system is determined by the single longest individual test set, while other tests may be substantially shorter;
- The resulting deterministic test sets do not take into account the memory requirements, imposed by the size of the on-chip memory or the external test equipment;
- The ad hoc test schedule may introduce idle periods, due to the test conflicts between the deterministic tests of different cores;

There are several possibilities for improvement. For example the ad-hoc solution can easily be improved by using a better scheduling strategy. This, however, does not necessarily lead to a significantly better solution as the ratio between pseudorandom and deterministic test patterns for every individual core is not changed. Therefore we have to explore different combinations between pseudorandom and deterministic test patterns for every individual core in order to find a solution where the total test length of the system is minimized and memory constraints are satisfied. In the following sections we will define this problem more precisely, and describe a methodology for calculating the optimal combination between different test sets for the whole system.

3. Basic Definitions and Problem Formulation

Let us assume that a system S consists of n cores $C_1, C_2, \ldots, C_n$. For every core C_k $\hat{I}$ S a complete sequence of deterministic test patterns TD^F_k and a complete sequence of pseudorandom test patterns TP^F_k will be generated. It is assumed that both test sets can obtain by itself maximum achievable fault coverage F_{max}.

Definition 1: A *hybrid BIST* set $TH_k = \{TP_k, TD_k\}$ for a core C_k is a sequence of tests, constructed from the subsets of pseudorandom test sequence $TP_k \subseteq TP^F_k$, and a deterministic test sequence $TD_k \subseteq TD^F_k$. The sequences TP_k and TD_k complement each other to achieve the maximum achievable fault coverage.

Definition 2: A pattern in a pseudorandom test sequence is called *efficient* if it detects at least one new fault that is not detected by the previous test patterns in the sequence. The ordered sequence of efficient patterns form an *efficient pseudorandom test sequence* $TPE_k = (P_1, P_2,\ldots,P_n) \subseteq TP_k$. Each efficient pattern $P_j \in TPE_k$ is characterized by the length of the pseudorandom test sequence TP_k, from the start to the efficient pattern P_j, including P_j. Efficient pseudorandom test sequence TPE_k, which includes all efficient patterns of TP^F_k is called *full efficient pseudorandom test sequence* and denoted by TPE^F_k.

***Definition 3*:** The cost of a hybrid test set TH_k for a core C_k is determined by the total length of its pseudorandom and deterministic test sequences, which can be characterized by their costs, $COST_{P,k}$ and $COST_{D,k}$ respectively:

$$COST_{T,k} = COST_{P,k} + COST_{D,k} = a|TP_k| + b_k|TD_k|$$

and by the cost of recourses needed for storing the deterministic test sequence TD_k in the memory:

$$COST_{M,k} = g_k|TD_k|$$

The parameters a and b_k are introduced here to align the application times of different test sequences. When test-per-scan scheme is assumed then it takes so many clock cycles to apply one test pattern, as long is the longest scan path in the design (a). The parameter b_k for a particular core C_k is equal to the total number of clock cycles needed for applying a deterministic test pattern from the memory. This includes the number of clock cycles needed for test data transportation as well as the number of clock cycles needed for scanning in the pattern. In a special case, when deterministic test patterns are applied by an external test equipment, application of deterministic test patterns may be up to one order of magnitude slower than applying BIST patterns. The coefficient g_k is used to map the number of test patterns in the deterministic test sequence TD_k into the memory recourses, measured in bits.

Definition 4: When assuming the test architecture described above, a hybrid test set $TH = \{TH_1, TH_2, \ldots, TH_n\}$ for a system $S = \{C_1, C_2, \ldots, C_n\}$ consists of hybrid tests TH_k for each

individual core C_k, where pseudorandom components of the *TH* can be scheduled in parallel, whereas the deterministic components of *TH* must be scheduled in sequence due to the shared test resources.

Definition 5: The test length of a hybrid test $TH = \{TH_1, TH_2, \ldots, TH_n\}$ for a system $S = \{C_1, C_2, \ldots, C_n\}$ is given by:

$$COST_T = \max\{\max_k(a|TP_k| + b_k|TD_k|), \sum_k b_k|TD_k|\}$$

The total cost of resources needed for storing the patterns from all deterministic test sequences TD_k in the memory is given by:

$$COST_M = \sum_k g_k|TD_k|$$

The objective of this paper is to find a shortest possible ($min(COST_T)$) hybrid test sequence TH_{opt} such that the memory constraints are not violated, i.e. $COST_M \leq COST_{M,LIMIT}$. In the following we will describe a methodology to solve the problem and present the experimental results to confirm our approach.

4. Estimation of Hybrid Test Sequence Characteristics

The test time minimization problem under memory constraints can be solved in a straightforward way if the supplementary deterministic test set for every possible length of the pseudorandom set is available. This can be achieved either by repetitive use of the automatic test pattern generator or by systematically analyzing and compressing the fault tables for each *j* [13]. Both procedures are accurate but time-consuming and therefore not feasible for larger designs. To overcome the complexity explosion problem we have proposed an iterative algorithm, where costs $COST_{M,k}$ and $COST_{D,k}$ for the deterministic test sets TD_k can be found based on estimates [14]. The estimation method is based on fault coverage figures and does not require accurate calculations of the deterministic test sets for not yet detected faults $F_{NOT,k}(j)$. In the following we will illustrate the use of the estimation methodology in case of test-per-scan scheme, based on realistic design example.

The estimation methodology requires a complete pseudorandom and deterministic test sequences. In Figure 2 we have presented an extract of the fault simulation results for both of those test sets. The length of the pseudorandom sequence has to be only so long as potentially necessary. By knowing the length of the complete deterministic test set and fault coverage figures for every individual pattern we can estimate the size of the additional deterministic test set for any length of the pseudorandom test sequence, as illustrated in the Figure 2. Here we can see that for a given core 60 deterministic test cycles are needed to obtain the same fault coverage as 524 pseudorandom test cycles and it requires additional 30 deterministic test cycles to reach 100% fault coverage. Based on this information we assume, that if we will apply those 30 deterministic test cycles on top of the 524 pseudorandom cycles, we can obtain close to the maximum fault coverage. This assumption is the basis of the cost estimation procedure. For calculating the real cost (length) of the deterministic set, we have to start the deterministic ATPG, to generate the deterministic test patterns that are really needed for detecting the rest of the faults, not yet covered by those 524 pseudorandom patterns.

|TP|	FC%	|TD|	FC%
1	21.9	1	43.3
2	34.7	2	45.6
	…		
524	97.5	60	97.5
	…		
1000	98.9	90	100

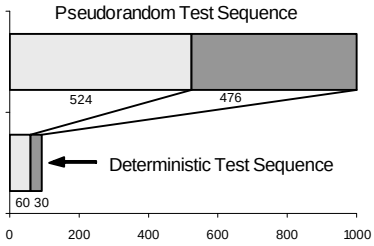

Figure 2. Estimation of the length of the deterministic test sequence (core s1423)

The resulting curve of cost estimates, that illustrates the memory requirements for a core in case of different test lengths, is depicted in Figure 3. For comparison we have also depicted the curve of real values, obtained by repetitive use of test pattern generator. By adding the cost estimates of all cores in the system we can obtain the hybrid BIST cost estimate for the whole system. This is illustrated in Figure 4.

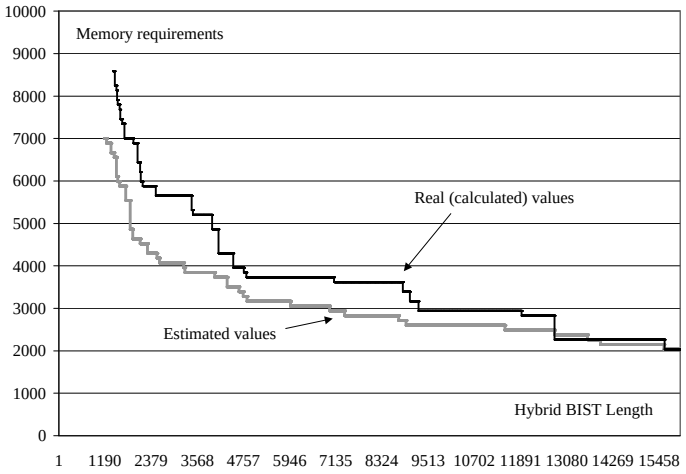

Figure 3. Test cost estimation accuracy

5. Test Length Minimization Under Memory Constraints

As described above, the exact calculations for finding the cost of the deterministic test set are very time-consuming. Therefore we have proposed to use cost estimates to find the minimal test length under given memory constraints [14]. Using the estimates can give us a quasi-minimal solution for the test length of the hybrid test at given memory constraints. After obtaining a quasi-minimal solution, the cost estimates can be improved and another, better, quasi-minimal solution can be calculated. This iterative procedure will be continued until we reach the final solution.

Figure 4 illustrates this procedure. At first the estimated test length for the given memory constraints is found. Then, based on the estimate the real memory cost is calculated. As can be seen in Figure 4, this particular solution would violate the memory constraint and therefore an iteration to find another solution has to be made. A new solution (test length) based on the difference between the estimated and real memory costs is found on the estimated cost curve and the search process is determined by the characteristics of the curve [14]. After iteration a new test length is found and a new real solution is calculated. It has

been shown that this procedure always converges. By each iteration we get closer to the memory constraints level, and also closer to the minimal test length at given constraints. However, the solution may be only near-optimal, since we only evaluate solutions derived from estimated cost functions.

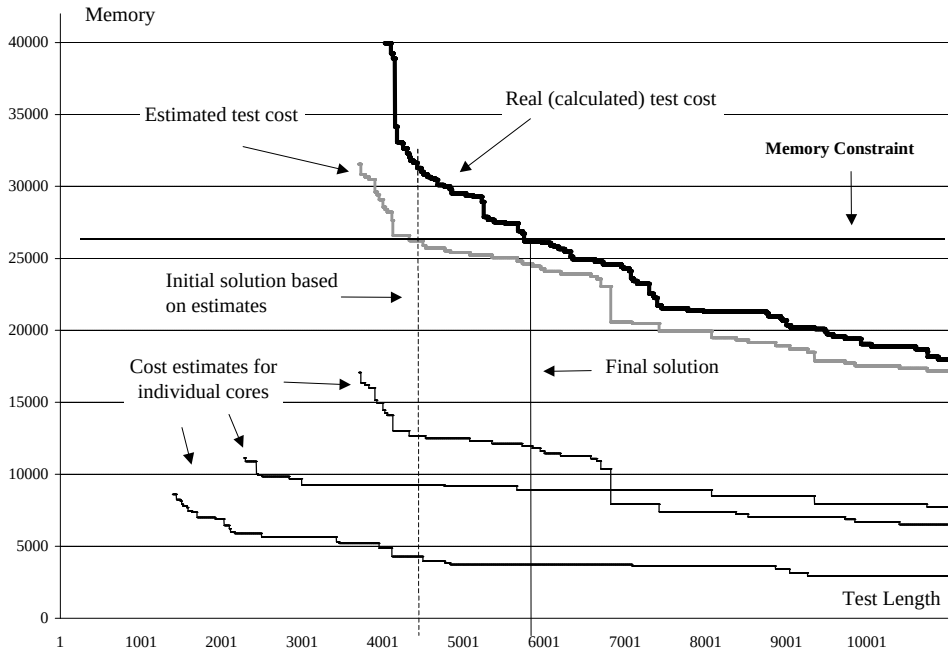

Figure 4. Test cost calculation based on estimates

6. Experimental Results

We have performed experiments with several systems composed from different ISCAS'89 benchmarks as cores. All cores have been redesigned to include full scan path (one or several). The STUMPS architecture was simulated in software and for deterministic test pattern generation a commercial ATPG tool from Mentor Graphics was used. The results are presented in Table 1.

Table 1. Experimental results

System Name	Number of Cores	Memory Constraint (bits)	Exhaustive Approach		Optimized Approach	
			Total Test Length (clocks)	CPU Time (seconds)	Total Test Length (clocks)	CPU Time (seconds)
J	6	25 000	5750	57540	5775	270
		22 000	7100		7150	216
		19 000	9050		9050	335
K	6	22 000	5225	53640	5275	168
		17 000	7075		7075	150
		13 000	9475		9475	427
L	6	15 000	3564	58740	3570	164
		13 500	4848		4863	294
		12 200	9350		9350	464

In Table 1 we compare our approach where the test length is found based on estimates, with an exact approach where deterministic test sets have been found by a brute force method (repetitive use of test pattern generator) for every possible switching point between pseudorandom and deterministic test patterns. As it can be seen from the results, our approach can give significant speedup (several orders of magnitude), while retaining very high accuracy.

7. Conclusions

We have presented an approach to the test time minimization problem for multi-core systems. We apply this to systems containing sequential cores with STUMPS architecture. To avoid the exhaustive exploration of solutions, the cost estimation method for the deterministic component of the hybrid test set is used. An iterative algorithm, based on cost estimates is thereafter applied in order to minimize the total test length of the hybrid BIST solution under the given memory constraints. Experimental results show the very high speed and accuracy of the proposed method compared to the exact calculation approach.

As a future work we would like to investigate the possibilities to apply the same approach also to the cores with partial scan. Additionally we would like to investigate more complex test architectures and include power constraints into the test time minimization algorithm.

References:

[1] B. T Murray, J. P. Hayes, "Testing ICs: Getting to the core of the problem," *IEEE Transactions on Computer*, Vol. 29, pp. 32-39, November 1996.

[2] Y. Zorian, E. J. Marinissen, S. Dey, "Testing Embedded Core-Based System Chips," *IEEE International Test Conference (ITC'98)*, pp. 130-143, 1998.

[3] G. Jervan, Z. Peng, R. Ubar, "Test Cost Minimization for Hybrid BIST," *IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT'00)*, pp.283-291, 2000.

[4] M. Garg, A. Basu, T. C. Wilson, D. K. Banerji, and J. C. Majithia, A New Test Scheduling Algorithm for VLSI Systems, *Proceedings of the CSI/IEEE Symposium on VLSI Design*, pp. 148-153, 1991.

[5] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices", *Proceedings of the IEEE VLSI Test Symposium (VTS)*, pp. 4-9, 1993.

[6] R. Chou, K. Saluja, and V. Agrawal, Scheduling Tests for VLSI Systems Under Power Constraints, *IEEE Transactions on VLSI Systems*, Vol. 5, No. 2, pp. 175-185, June 1997.

[7] K. Chakrabarty, "Test Scheduling for Core-Based Systems", *Proceedings of IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, page 391-394, 1999.

[8] M. Sugihara, H. Date, H. Yasuura, "Analysis and Minimization of Test Time in a Combined BIST and External Test Approach," *Design, Automation & Test In Europe Conference (DATE 2000)*, pp. 134-140, 2000.

[9] V. Muresan, X. Wang, V. Muresan, and M. Vladutiu, A Comparison of Classical Scheduling Approaches in Power-Constrained Block-Test Scheduling, *IEEE International Test Conference (ITC'2000)*, pp. 882-891, 2000.

[10] K. Chakrabarty, "Test Scheduling for Core-Based Systems Using Mixed-Integer Linear Programming", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 19, No. 10, pp. 1163-1174, October 2000.

[11] E. Larsson, Z. Peng, "An Integrated Framework for the Design and Optimization of SOC Test Solutions", *Journal of Electronic Testing; Theory and Applications (JETTA), for the Special Issue on Plug-and-Play Test Automation for System-on-a-Chip*, Vol. 18, no. 4/5, pp. 385-400, August 2002.

[12] G. Jervan, Z. Peng, R. Ubar, H. Kruus, "A Hybrid BIST Architecture and its Optimization for SoC Testing," *IEEE 2002 3rd International Symposium on Quality Electronic Design (ISQED'02)*, pp. 273-279, 2002.

[13] R. Ubar, G. Jervan, Z. Peng, E. Orasson, R. Raidma, "Fast Test Cost Calculation for Hybrid BIST in Digital Systems", *Euromicro Symposium on Digital Systems Design*, pp. 318-325, 2001.

[14] G. Jervan, P. Eles, Z. Peng, R. Ubar, M. Jenihhin, "Test Time Minimization for Hybrid BIST of Core-Based Systems", *accepted to the Asian Test Symposium 2003*, 2003.

[15] P. H. Bardell, W. H. McAnney, J. Savir, "Built-In Test for VLSI Pseudorandom Techniques," *John Wiley and Sons*, 1987.