Test Time Minimization for Hybrid BIST of Core-Based Systems

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. 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. In this paper we propose an iterative algorithm 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 a near optimal solutions.

1. Introduction

Today's microelectronics technology provides designers the possibility to integrate a large number of different functional blocks, usually referred as cores, in a single IC. Such a design style allows designers to reuse previous designs and will lead therefore to shorter time-to-market and reduced cost. Such a system-on-chip (SoC) approach is very attractive from the designers' perspective. Testing of such systems, on the other hand, is a problematic and time consuming task, mainly due to the resulting IC's complexity and the high integration density [1].

To test the individual cores in such systems the test pattern source and sink have to be available together with an appropriate test access mechanism (TAM) [2]. Traditional approaches implement both test source and sink off-chip and require therefore the use of external Automatic Test Equipment (ATE). As the requirements for the ATE speed and memory size are continuously increasing, the ATE solution may be unacceptably expensive and inaccurate. Therefore, in order to apply at-speed tests and to keep the test costs under control, on-chip test solutions are becoming a mainstream technology for testing such complex systems. Such a solution is usually referred to as built-in self-test (BIST).

Different test scenarios are possible while using BIST. Sometimes the embedded cores may be tested using only internally generated pseudorandom test patterns. Due to several reasons, like very long test sequences, and random pattern resistant faults, this approach may not always be efficient. One solution to this 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.

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]. However there hasn't been any work to find the optimal test sets for testing every individual core in such a manner that the total system test time is minimized and the different ATE constraints satisfied. 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.

This paper deals with the problem of core-based system testing where hybrid BIST approach is used. Our earlier work, [3], [12] and [13], has been concentrating on test cost calculation and hybrid BIST optimization for single-core designs. In this paper we propose a methodology for test time minimization, under memory constraints, for multi-core systems. We propose an algorithm for calculating the best combination between pseudorandom and deterministic tests, where the memory constraints are not violated, the total test

[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.

time is minimized, and maximum achievable fault coverage is guaranteed.

Relations between different cost components of the test sets, as functions of the hybrid BIST structure, are introduced to find the optimal solution. To avoid exhaustive search, a method for estimating the cost of the deterministic component in the hybrid test set is introduced. Finally, based on our estimation methodology, we have developed an iterative algorithm to minimize the total length of the hybrid BIST solution under given memory constraints.

The rest of the paper is organized as follows. In Section 2 a hybrid BIST 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 presented 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

Recently we have proposed a hybrid BIST optimization methodology for a single core designs [3]. Such a hybrid BIST approach starts with a pseudorandom test sequence of length L. At the next stage, the stored test approach takes place: precomputed deterministic test patterns are applied to the core under test 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 achievable by the pseudorandom test. 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 L_{OPT} of the pseudorandom test sequence, in order to minimize the total testing cost. Our previously proposed methodology enables us to find the most cost-effective combination of the two test sets not only in terms of test time but also in terms of tester/on-chip memory requirements. The efficiency of such approach has been demonstrated so far for individual cores. In this paper we propose an approach to extend our methodology also for complex systems containing more than one core. We take into account the constraints (memory size) imposed by the system and minimize the testing time for the whole system with multiple cores, while keeping the high fault coverage.

In this paper we assume the following test architecture: Every core has its own dedicated BIST logic that is capable to produce 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.

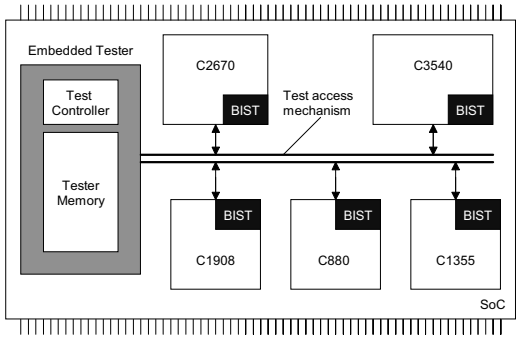

Figure 1. A core-based system example with the proposed test architecture

This example system consists of 5 cores (different ISCAS benchmarks). Using our hybrid BIST optimization methodology for single core [3] we can find the optimal combination between pseudorandom and deterministic test patterns for every individual core (Figure 2). Considering the assumed test architecture, only one deterministic test set can be applied at any given time, while any number of pseudorandom test sessions can take place in parallel. To enforce the assumption that only one deterministic test can be applied at a time, a simple ad-hoc scheduling can be used. The result of this scheduling defines the starting moments for every deterministic test session, the memory requirements, and the total test length t for the whole system. This situation is illustrated on Figure 2.

As it can be seen from Figure 2, 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. In the example we have illustrated three potential 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 proposed test schedule may introduce idle periods, due to the test conflicts between the deterministic tests of different cores;

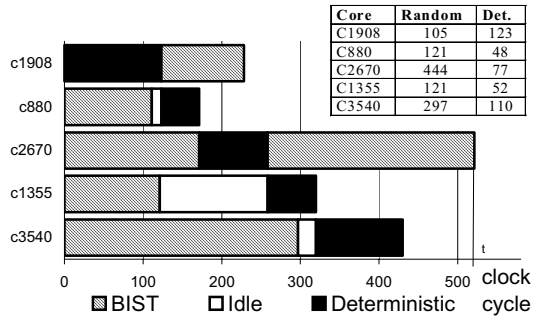

Figure 2. Ad-hoc test schedule for hybrid BIST of the core-based system example

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 propose a fast iterative algorithm 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 $C_1, C_2, ..., C_n$. For every core $C_k \in S$ 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* $TPE_k = (P_1, P_2, ..., P_n) \subseteq TP_k$. 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:

$$COST_{T,k} = COST_{P,k} + COST_{D,k} = \alpha|TP_k| + \beta_k|TD_k|$$

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

$$COST_{M,k} = \gamma_k|TD_k|.$$

The parameters α and β_k can be introduced by the designer to align the application times of different test sequences. For example, when a test-per-clock BIST scheme is used, a new test pattern can be generated and applied in each clock cycle and in this case $\alpha = 1$. The parameter β_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. 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 γ_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 $TH = \{TH_1, TH_2, ..., TH_n\}$ for a system $S = \{C_1, C_2, ..., C_n\}$ 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: $J = (j_1, j_2, ..., j_n)$ is called the *characteristic vector* of a hybrid test set $TH = \{TH_1, TH_2, ..., TH_n\}$, where $j_k = |TPE_k|$ is the length of the efficient pseudorandom test sequence $TPE_k \subseteq TP_k \subseteq TH_k$.

According to Definition 2, for each j_k corresponds a pseudorandom subsequence $TP_k(j_k) \subseteq TP^F_k$, and according to Definition 1, any pseudorandom test sequence $TP_k(j_k)$ should be complemented with a deterministic test sequence, denoted with $TD_k(j_k)$, that is generated in order to achieve the maximum achievable fault coverage. Based on this we can conclude that the characteristic vector J determines entirely the structure of the hybrid test set TH_k for all cores $C_k \in S$.

Definition 6: The test length of a hybrid test $TH = \{TH_1, TH_2, ..., TH_n\}$ for a system $S = \{C_1, C_2, ..., C_n\}$ is given by:

$$COST_T = \max\{\max_k(\alpha|TP_k| + \beta_k|TD_k|), \sum_k \beta_k|TD_k|\}.$$

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

$$COST_M = \sum_k \gamma_k|TD_k|.$$

Definition 7: Let us introduce a generic cost function $COST_{M,k} = f_k(COST_{T,k})$ for every core $C_k \in S$, and an integrated generic cost function $COST_M = f_k(COST_T)$ for the

whole system S.

The functions $COST_{M,k} = f_k(COST_{T,k})$ will be created in the following way. Let us have a hybrid BIST set $TH_k(j) = \{TP_k(j), TD_k(j)\}$ for a core C_k with j efficient patterns in the pseudorandom test sequence. By calculating the costs $COST_{T,k}$ and $COST_{M,k}$ for all possible hybrid test set structures $TH_k(j)$, i.e. for all values $j = 1, 2, \ldots, |TPE^F_k|$, we can create the cost functions $COST_{T,k} = f_{T,k}(j)$, and $COST_{M,k} = f_{M,k}(j)$. By taking the inverse function $j = f'_{T,k}(COST_{T,k})$, and inserting it into the $f_{M,k}(j)$ we get the generic cost function $COST_{M,k} = f_{M,k}(f'_{T,k}(COST_{T,k})) = f_k(COST_{T,k})$ where the memory costs are directly related to the lengths of all possible hybrid test solutions.

The integrated generic cost function $COST_M = f(COST_T)$ for the whole system is the sum of all cost functions $COST_{M,k} = f_k(COST_{T,k})$ of individual cores $C_k \in S$.

From the function $COST_M = f(COST_T)$ the value of $COST_T$ for every given value of $COST_M$ can be found. The value of $COST_T$ determines the lower bound of the length of the hybrid test set for the whole system. To find the component j_k of the characteristic vector J, i.e. to find the structure of the hybrid test set for all cores, the equation $f_{T,k}(j) = COST_T$ should be solved.

The objective of this paper is to find a shortest possible ($min(COST_T)$) hybrid test sequence TH_{opt} when the memory constraints are not violated $COST_M \le COST_{M,LIMIT}$.

4. Hybrid Test Sequence Computation Based on Cost Estimates

By knowing the generic cost function $COST_M = f(COST_T)$, the total test length $COST_T$ at any given memory constraint $COST_M \le COST_{M,LIMIT}$ can be found in a straightforward way. However, the procedure to calculate the cost functions $COST_{D,k}(j)$ *and* $COST_{M,k}(j)$ is very time consuming, since it assumes that the deterministic test set TD_k for each $j = 1, 2, \ldots, |TPE^F_k|$ has to be available. This assumes that after every efficient pattern $P_j \in TPE_k \subseteq TP_k, j = 1, 2, \ldots, |TPE^F_k|$ a set of not yet detected faults $F_{NOT,k}(j)$ should be calculated. This can be done 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 propose 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. 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 use $FD_k(i)$ and $FPE_k(i)$ to denote the fault coverage figures of the test sequences $TD_k(i)$ and $TPE_k(i)$, correspondingly, where i is the length of the test sequence.

Procedure 1: Estimation of the length of the deterministic test set TD_k.

1. Calculate, by fault simulation, the fault coverage functions $FD_k(i)$, $i = 1, 2, \ldots, |TD^F_k|$, and $FPE_k(i)$, $i = 1, 2, \ldots, |TPE^F_k|$. The patterns in TD^F_k are ordered in such the way that each pattern put into the sequence contributes with maximum increase in fault coverage.
2. For each $i^* \le |TPE^F_k|$, find the fault coverage value F^* that can be reached by a sequence of patterns $(P_1, P_2, \ldots, P_{i^*}) \subseteq TPE_k$ (see Figure 3).
3. By solving the equation $FD_k(i) = F^*$, find the maximum integer value j^* that satisfies the condition $FD_k(j^*) \le F^*$. The value of j^* is the length of the deterministic sequence TD_k that can achieve the same fault coverage F^*.
4. Calculate the value of $|TD^E_k(i^*)| = |TD^F_k| - j^*$ which is the number of test patterns needed from the TD^F_k to reach to the maximum achievable fault coverage.

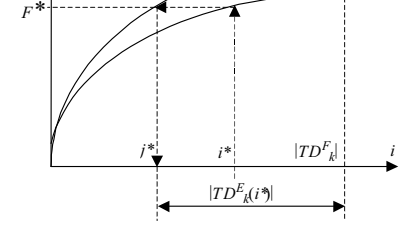

Figure 3. Estimation of the length of the deterministic test sequence

The value $|TD^E_k(i^*)| = |TD^F_k| - j^*$, calculated by the Procedure 1, can be used to estimate the length of the deterministic test sequence TD_k in the hybrid test set $TH_k = \{TP_k, TD_k\}$ with i^* efficient test patterns in TP_k, ($|TPE_k| = i^*$).

By finding $|TD^E_k(j)|$ for all $j = 1, 2, \ldots, |TPE^F_k|$ we get the cost function estimate $COST^E_{D,k}(j)$. Using $COST^E_{D,k}(j)$, other cost function estimates $COST^E_{M,k}(j)$, $COST^E_{T,k}(j)$ and $COST^E_{M,k} = f^E_k(COST^E_{T,k})$ can be created according to the Definitions 3 and 7.

Finally, by adding cost estimates $COST^E_{M,k} = f^E_k(COST^E_{T,k})$ of all cores, we get the hybrid BIST cost function estimate $COST^E_M = f^E(COST^E_T)$ for the whole system.

5. Test Length Minimization Under Memory Constraints

As described above, the exact calculations for finding the cost of the deterministic test set $COST_{M,k} = f_k(COST_{T,k})$ are very time-consuming. Therefore we will use the cost estimates, calculated by Procedure 1 in Section 4, instead. Using 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.

Procedure 2: Test length minimization.

1. Given the memory constraint $COST_{M,LIMIT}$, find the estimated total test length $COST^{E*}_{T}$ as a solution to the equation $f^{E}(COST^{E}_{T}) = COST_{M,LIMIT}$.
2. Based on $COST^{E*}_{T}$, find a candidate solution $J^* = (j^*_1, j^*_2, ..., j^*_n)$ where each j^*_k is the maximum integer value that satisfies the equation $COST^{E}_{T,k}(j^*_k) \le COST^{E*}_{T}$.
3. To calculate the exact value of $COST^*_M$ for the candidate solution J^*, find the set of not yet detected faults $F_{NOT,k}(j^*_k)$ and generate the corresponding deterministic test set TD^*_k by using an ATPG algorithm.
4. If $COST^*_M = COST_{M,LIMIT}$, go to the Step 9.
5. If the difference $|COST^*_M - COST_{M,LIMIT}|$ is bigger than that in the earlier iteration make a correction $\Delta t = \Delta t/2$, and go to Step 7.
6. Calculate a new test length $COST^{E,N}_{T}$ from the equation $f^{E}_{k}(COST^{E}_{T}) = COST^{*}_{M}$, and find the difference $\Delta t = COST^{E,*}_{T} - COST^{E,N}_{T}$.
7. Calculate a new cost estimate $COST^{E,*}_{T} = COST^{E,*}_{T} + \Delta t$ for the next iteration.
8. If the value of $COST^{E,*}_{T}$ is the same as in an earlier iteration, go to Step 9, otherwise go to Step 2.
9. *END:* The vector $J^* = (j^*_1, j^*_2, ..., j^*_n)$ is the solution.

To illustrate the above procedure, in Figures 4 and 5 an example of the iterative search for the shortest length of the hybrid test is given. Figure 4 represents all the basic cost curves $COST^{E}_{D,k}(j)$, $COST^{E}_{P,k}(j)$, and $COST^{E}_{T,k}(j)$, as functions of the length j of TPE_k where j_{min} denotes the optimal solution for a single core hybrid BIST optimization problem [3]. Figure 5 represents the estimated generic cost function $COST^{E}_{M} = f^{E}(COST^{E}_{T})$ for the whole system. At first (Step 1), the estimated $COST^{E*}_{T}$ for the given memory constraints is found (point 1 on Figure 5). Then (Step 2), based on $COST^{E*}_{T}$ the length j^*_k of TPE_k for the core C_k in Figure 4 is found. This procedure (Step 2) is repeated for all the cores to find the characteristic vector J^* of the system as the first iterative solution. After that the real memory cost $COST^{E*}_{M}$ is calculated (Step 3, point 1* in Figure 5). As we see in Figure 5, the value of $COST^{E*}_{M}$ in point 1* violates the memory constraints. The difference Δt_1 is determined by the curve of the estimated cost (Step 6). After correction, a new value of $COST^{E*}_{T}$ is found (point 2 on Figure 5). Based on $COST^{E*}_{T}$, a new J^* is found (Step 2), and a new $COST^{E*}_{M}$ is calculated (Step 3, point 2* in Figure 5). An additional iteration via points 3 and 3* can be followed in Figure 5.

It is easy to see that Procedure 2 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.

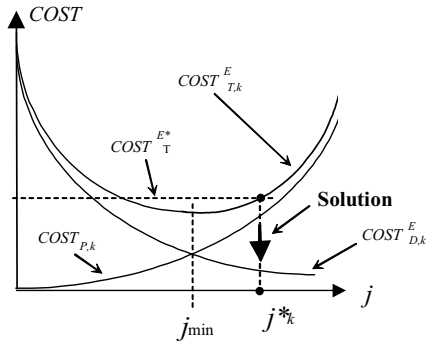

Figure 4. Cost curves for a given core C_k

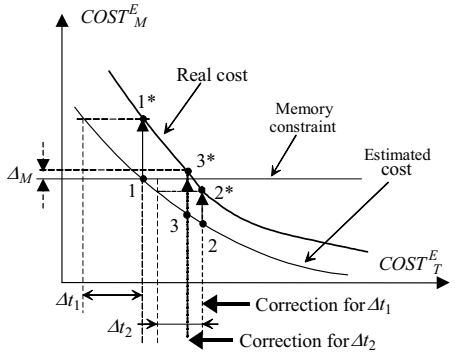

Figure 5. Minimization of the test length

6. Experimental Results

We have performed experiments with several systems composed from different ISCAS benchmarks as cores. The results are presented in Table 1.

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 manipulating the fault tables 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 (more than order of magnitude), while retaining acceptable accuracy (the biggest deviation is less than 9% from the exact solution, in average 2.4%).

Table 1. Experimental results

System	Number of cores	Memory Constraint (bits)	Exact approach		Our approach	
			Total Test Length (clocks)	CPU Time (seconds)	Total Test Length (clocks)	CPU Time (seconds)
S1	6	20 000	222	3772.84	223	199.78
		10 000	487		487	57.08
		7 000	552		599	114.16
S2	7	14 000	207	3433.10	209	167.3
		5 500	540		542	133.84
		2 500	1017		1040	200.76
S3	5	7 000	552	10143.14	586	174.84
		3 500	3309		3413	291.40
		2 000	8549		8 556	407.96

In Figure 6 we present the estimated cost curves for the individual cores and the estimated and real cost curves for the system S2. We also show in this picture a test solution point for this system under given memory constraint that has been found based on our algorithm. In this example we have used a memory constraint M_{LIMIT} = 5500 bits. The final test length for this memory constraint is 542 clock cycles and that gives us a test schedule depicted in Figure 7.

7. Conclusions

We have presented an approach to the test time minimization problem for multi-core systems that are tested with a hybrid BIST strategy. A heuristic algorithm was proposed to minimize the test length for a given memory constraint. The algorithm is based on the analysis of different cost relationships as functions of the hybrid BIST structure. To avoid the exhaustive exploration of solutions, a method for the cost estimation of the deterministic component of the hybrid test set was proposed. We have also proposed an iterative algorithm, based on the proposed estimates, to minimize the total test length of the hybrid BIST solution under the given memory constraints. Experimental results show very high speed of the algorithm, compared to the exact calculation method.

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

8. 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 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.

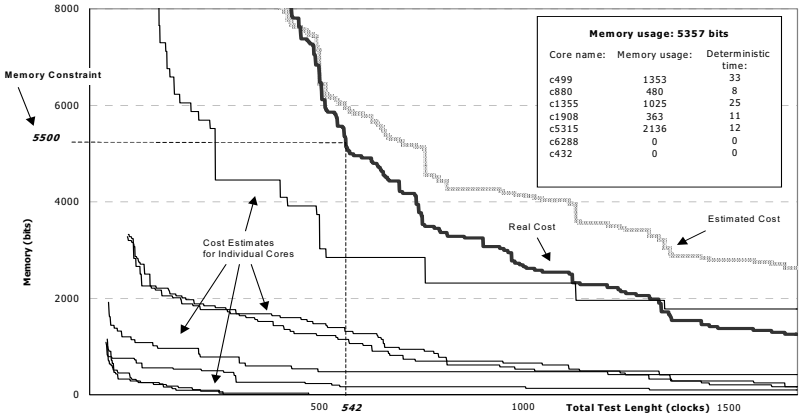

Figure 6. The final test solution for the system S2 (M_{LIMIT} = 5 500)

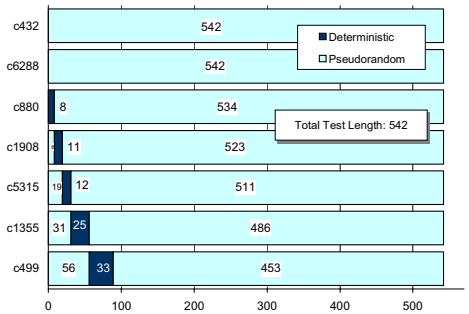

Figure 7. Test Schedule for the system S2 (M_{LIMIT} = 5 500)