# Test Time Minimization for Hybrid BIST with Test Pattern Broadcasting

Raimund Ubar, Maksim Jenihhin
Department of Computer Engineering
Tallinn Technical University
EE-126 18 Tallinn, Estonia
{raiub, maksim}@pld.ttu.ee

Gert Jervan, Zebo Peng
Embedded Systems Laboratory (ESLAB)
Linköping University
SE-581 83 Linköping, Sweden
{gerje, zebpe}@ida.liu.se

**Abstract[1]:**

*This paper describes a hybrid BIST architecture for testing core-based systems together with a method for test time minimization. The approach uses test pattern broadcasting for both pseudorandom and deterministic patterns. To overcome the high complexity of the test time minimization problem we propose a fast algorithm to find an efficient combination of pseudorandom and deterministic test sets under given memory constraints. The efficiency of the approach is demonstrated by experimental results.*

## 1. Introduction

The advances in microelectronics technology has made it possible to integrate a large number of different functional blocks, usually referred as cores, in a single IC. Testing of such systems-on-chip (SoC) is a problematic and time consuming task, mainly due to the resulting IC's complexity and the high integration density [1]. One of the solutions to this problem is on-chip test, usually referred to as built-in self-test (BIST). In our earlier work we have proposed a hybrid BIST technique [2], [3] as one of the promising methodologies for self-test in SoCs. The key issue for the hybrid BIST is to find the best balance between pseudorandom and deterministic test patterns, such that the system design constraints are satisfied and test cost is minimized.

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]-[8]. 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 design constraints are satisfied.

In our earlier work it has been assumed that every core has its own dedicated BIST logic that is capable to produce a set of independent pseudorandom test patterns [9]. This however may lead to high area overhead and may require redesign of the cores (to include the BIST logic). In this paper we propose a novel self-test approach that is based on test pattern broadcasting [10]. The proposed architecture enables us to generate pseudorandom pattern simultaneously for all cores by using a single pseudorandom pattern generator. These patterns will be complemented with dedicated deterministic patterns for every individual core and the whole test process is carried out in such a manner that the total testing time is kept minimal without violating the design constraints, in particular, the amount of on-chip resources.

In the following section we will describe the hybrid BIST architecture with test pattern broadcasting in detail. It is followed by the test time minimization procedure that is finally demonstrated with experimental results.

## 2. Hybrid BIST Architecture Using Test Pattern Broadcasting

In our earlier work we have proposed a hybrid BIST methodology and its optimization for single core designs [2]-[3]. We have also proposed a hybrid BIST architecture for multi-core designs, where it is assumed, that every core in the system has it's own dedicated BIST structure [3]. This type of architecture however may not always be feasible as not all cores may be equipped with self-test structures. It may also introduce a significant area overhead and the performance degradation, as some cores may require excessively large self-test structures (LFSRs).

To avoid those problems a single pseudorandom test pattern generator for the whole system gives a better solution. It can be implemented as a dedicated hardware block or in software. In the latter case the test program, together with test data (LFSR polynomial, initial state, pseudorandom test length, signature), is kept in a ROM and executed in a test mode. This, however, may lead to a very complex test controller, as every core requires pseudorandom patterns with different characteristics (polynomial, initial state and length, for example) and therefore may lead to very complex and expensive solutions. In this paper we propose a novel methodology, where only a single set of pseudorandom test patterns that is broadcasted to all cores simultaneously will be used. This universal pseudorandom test set is followed by additional deterministic vectors applied to every individual core, if needed. Those deterministic test vectors are generated

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

during the development process and are stored in the system. For this purpose arbitrary software test generators may be used, based on deterministic, random or genetic algorithms. This architecture together with appropriate test access mechanism is depicted in Figure 1.

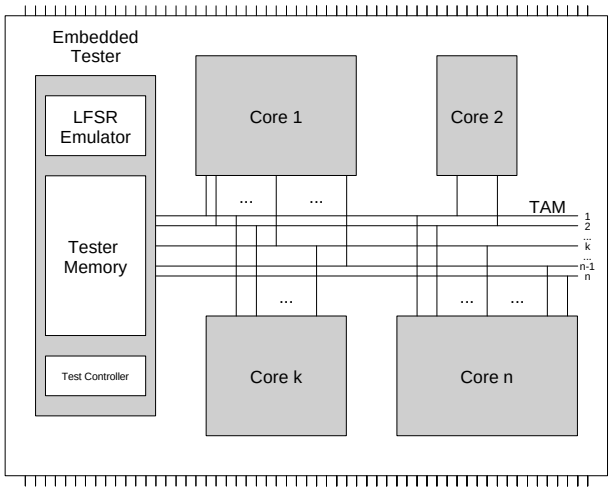


*Figure 1. Hybrid BIST architecture with test pattern broadcasting*

Testing of all cores is carried out in parallel, i.e. all pseudorandom patterns as well as each deterministic test sequence $TD_k$ is applied to all cores in the system. The deterministic test sequence $TD_k$ is a deterministic test sequence generated only by analyzing the core $C_k$. For the rest of the cores this sequence can be considered as a pseudorandom sequence. The width of the hybrid test sequence *TH* is equal to $MAXINP=\max\{INP_k\}$, $k=1, 2, \ldots, n$, where $INP_k$ is the number of inputs of the core $C_k$. For each deterministic test set $TD_k$, where $INP_k < MAXINP$, the not specified bits will be completed with pseudorandom data, so that the resulting test set $TD_k$* can be applied in parallel to the other cores in the system as well.

In case of hybrid BIST, we can dramatically reduce the length of the initial pseudorandom sequence by complementing it with deterministic stored test patterns, and achieve 100% fault coverage. The method proposed in this paper helps to find tradeoffs between the length of the best pseudorandom test sequence and the number of stored deterministic patterns, under given memory constraints. The problem of finding the exact solution is NP-complete. To overcome the high complexity of the problem we will propose in the following a simple and fast algorithm that gives us a quasioptimal solution with low computational cost. Although the solution is not optimal it can be used successfully for design space exploration.

## 3. Formulation of the Test Time Minimization Problem

Let us assume a system *S*, consisting of cores $C_1, C_2, \ldots, C_n$, that are all connected to the bus. For this system a pseudorandom test sequence *TP* with length *LP* is generated and applied in parallel to all cores. This sequence should preferably achieve 100% fault coverage for all cores. In this sequence we can specify subsequences $TP_k$ with length $LP_k$, $k = 1, 2, \ldots, n$, for each core, so that all the subsequences start in the beginning of *TP*, and by the last pattern of a subsequence $TP_k$ the 100% fault coverage for the core $C_k$ is reached.

In a case when $LP_k$ is too long, we restrict the length of the pseudorandom sequence to the maximum acceptable length $LP_{max}$, thus reducing the length of the whole pseudorandom sequence to $LP_{max}$. For all cores where 100% fault coverage has not been achieved with this test set *TP* we generate complementary joint set of deterministic test patterns *TD*, so that by applying to the system both test sequences *TP* and *TD* with total length *L*, the 100% fault coverage for all cores is achieved.

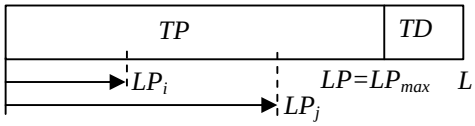


*Figure 2. Initial test sequence for multi-core system*

As an example, in Figure 2 a hybrid test sequence $TH = \{TP, TD\}$ is shown consisting of a pseudorandom test set *TP* with length *LP* and a deterministic test set *TD* with length $LD$ $(L=LP+LD)$. Here $LP_i$ denotes a moment where 100% fault coverage is reached for the core $C_i$, and $LP_j$ denotes a moment where 100% fault coverage is reached for the core $C_j$. In this example we assume that not for all cores 100% fault coverage is achieved by the pure pseudorandom test sequence *TP* and an additional deterministic test set *TD* has to be applied to achieve 100% fault coverage. Those deterministic test patterns are precomputed and stored in the system.

The main problem of the hybrid BIST is to find the optimal balance between the pseudorandom test part *TP* and the deterministic test part *TD*, so that the total testing time is minimal, and that the memory constraints $COST_{M.LIMIT}$ for storing deterministic test patterns are satisfied, $COST_M \le COST_{M,LIMIT}$. The memory cost can be calculated as follows:

$$COST_M = \sum_{k=1}^{n} (LD_k * INP_k),$$

where $INP_k$ is the number of inputs of the core $C_k$ and $LD_k$ is the length of the deterministic test set of the core $C_k$. If the same deterministic pattern is needed simultaneously for a subset $S' \subseteq S$ of cores, we say that it is dedicated for the core $C_k \in S'$ with the highest number of inputs. The efficiency of the hybrid BIST approach is achieved by minimizing the total test length

$$LH = LP + \sum_{k=1}^{n} LD_k$$

for a given memory constraint $COST_M \le COST_{M,LIMIT.}$

As all cores are tested in parallel, the problem is to find a time moment when to switch from the parallel pseudorandom test to the parallel deterministic test. The problem of minimizing the hybrid test length at the given memory constraints for parallel multi-core testing is extremely complex. The main reasons of this complexity are the following:

- The deterministic test patterns of one core are used as pseudorandom test patterns for all other cores; unfortunately there will be $n(n\text{-}1)$ relationships for $n$

cores to analyse for finding the optimal interaction; on the other hand the deterministic test sets are not readily available and calculated only during the analysis process;

- For a single core an optimal combination of pseudorandom and deterministic patterns can be found by rather straightforward algorithms [9]; but as the optimal time moment for switching from pseudorandom to deterministic test will be different for different cores the existing methods cannot be used and the parallel testing case is considerably more complex.
- For each core the best initial state of the LFSR can be found experimentally, but to find the best LFSR for testing all cores in parallel is a very complex and time consuming task.

To overcome the high complexity of the problem we have proposed a straightforward algorithm for calculating *TP* and *TD*, where we neglect the optimal solutions for individual cores in favour of finding a quasioptimal solution for the whole system [11].

## 4. Test Time Minimization Procedure

We solve the test time minimization problem in three consecutive steps: first, we find as good as possible initial state for the LFSR for all cores; second, we generate a deterministic test sequence if the 100% fault coverage cannot be reached by a pure pseudorandom test sequence for all cores; and third, we update the test sequence by finding the quasi-optimal time moment for switching from parallel pseudorandom testing to parallel deterministic testing at the given memory constraint.

*Finding the initial state for the LFSR.*

To find the best initial state for the parallel pseudorandom test generator, we carry out $m$ experiments, with randomly chosen initial states, for all $n$ cores. Let us denote with $INP_k$ the number of inputs of core $C_k$. Within each $j^{th}$ experiment we calculate for each core $C_k$ the weighted length $LP_{k,j} * INP_k$ of the test sequence which achieves the 100% fault coverage for the core $C_k$. Then, for all the experiments we calculate the average weighted length

$$L_j = \frac{1}{n}\sum_{k=1}^{n} LP_{k,j} * INP_k$$

as the quality merit of pseudorandom sequences for parallel testing of all cores. The best pseudorandom sequence is the one that gives as shortest $L_j$, $j = 1,2,\ldots, m$. Let us call this initial pseudorandom test $TP^0$.

*Generation of the initial deterministic test set.*

Suppose there are $k \le n$ cores where 100% fault coverage cannot be achieved with $TP^0$ because of the practical constraints to the pseudorandom test length. Let us denote this subset of cores with $S' \subseteq S$. Let us denote with $FP_i^0$ fault coverage of the core $C_i$, achieved by $TP^0$. Let us order the cores in $S'$ as $C_1, C_2, \ldots, C_k$, so that for each $i < j$, $1 \le i,j \le k$, we have $FP_i \le FP_j$. We assume here that every deterministic test pattern, to be propagated to the system, has to be as wide as the maximum width of the TAM. If the core under test has less inputs than the width of the TAM, all unused bits in the TAM are filled with pseudorandom data. The initial deterministic test set can be found by generating deterministic test patterns for every core in $S'$ individually, starting from the core $C_1$, in order to achieve 100% fault coverage for this particular core. The generated test patterns are simulated with the remaining cores and their respective fault coverages $FP_j$ will be updated. This process is carried out for every cores in $S'$ and guarantees 100% fault coverage for all cores in the system.

*Optimization of the test sequence.*

After the previous 2 steps we have obtained a hybrid BIST sequence $TH^0 = \{TP^0, TD^0\}$ with length $LH^0$, consisting of the pseudorandom part $TP^0$ with length $LP^0$, and of the deterministic part $TD^0$ with length $LD^0$.

In special case $TD^0$ may be an empty set.

Let us denote with $COST_M(TD^0)$ the memory cost of the deterministic test set $TD^0$. We assume that the memory constraints are at this moment satisfied: $COST_M(TD^0) < COST_{M,LIMIT}$. In a opposite case, if $COST_M(TD^0) > COST_{M,LIMIT,}$ the length of the pseudorandom sequence has to be extended and the second step of the procedure has to be repeated.

If $COST_M(TD^0) = COST_{M,LIMIT}$ the third step is unnecessary, and the procedure is finished.

Under optimization of $TH^0$ we mean the minimization of the test length $LH^0$ at the given memory constraints $COST_{M,LIMIT}$.

It is possible to minimize $LH^0$ by shortening the pseudorandom sequence, i.e. by moving step-by-step efficient patterns from the beginning of $TP^0$ to $TD^0$ and by removing all other patterns between the efficient ones from $TP^0$, until the memory constraints will become violated, $COST_M(TD^0) > COST_{M,LIMIT}$.

To find the efficient test patterns in the beginning of the $TP^0$ we have to fault simulate the whole test sequence $TH^0$ for all the cores in the opposite way from the end to the beginning. As a result of the fault simulation we get for each pattern the increments of fault coverage in relation to each core $D = \{D_1, D_2,\ldots, D_n,\}$.

As the result of this procedure we create a new hybrid BIST sequence $TH = \{TP,TD\}$ with total length $LH$ and with lengths $LP \le LP^0$ and $LD \ge LD^0$ for the new pseudorandom and deterministic parts correspondingly. Due to removal of all non-efficient patterns $LP - LP^0 >> LD^0 - LD$. Hence, the total length of the new hybrid BIST sequence will be considerably shorter compared to its initial length, $LH < LH^0$.

The proposed approach doesn't guarantee absolute minimum of the test length, however, the procedure is rather straightforward (similar to the greedy algorithm) and fast and therefore suitable for use in the design process. The method can be used to find a cheap practical solution as well as for a fast reference for comparison with more sophisticated optimization algorithms to be developed in the future.

## 5. Experimental data

We have performed experiments with three systems composed from different ISCAS benchmarks as cores. The data of these systems are presented in Table 1 (the lists of used cores in each system) .

| System name | S1 6 cores | S2 7 cores | S3 5 cores |
|---|---|---|---|
| List of used cores | c5315 | c432 | c880 |
| | c880 | c499 | c5315 |
| | c432 | c880 | c3540 |
| | c499 | c1355 | c1908 |
| | c499 | c1908 | c880 |
| | c5315 | c5315 | |
| | | c6288 | |

*Table 1. Systems used for experiments*

The experimental results for three different systems are presented in Table 2. The total length of the hybrid test sequence is calculated for three different memory constraints.

The CPU time used for carrying out the procedure for each system is depicted in the last column. For the first two systems S1 and S2 the cost of the procedure is determined only by the CPU time for pseudorandom pattern generation and subsequent simulation of test patterns for all the cores in the system. For the third system S3 the CPU time includes also the time needed for generation of deterministic patterns. In this case the length of the pseudorandom test, to reach 100% fault coverage, would be too long to be practically feasible. The pseudorandom test was stopped at the length of 14524 patterns with 98,26% fault coverage for one of the cores and 26 deterministic patterns were generated in order to achieve 100% fault coverage.

In Figure 3 the test schedules for all systems are presented when memory constraint is at 10000 bits. The left part represents the pseudorandom test, and the right part represents the deterministic test.

The full overview about the all possible hybrid BIST solutions for the three systems is presented in Figure 4 representing the memory cost as the function of total test length. By these curves for an arbitrary memory constraint the corresponding total testing time can be found. The three constraints illustrated in Table 2 are also highlighted in Figure 4.

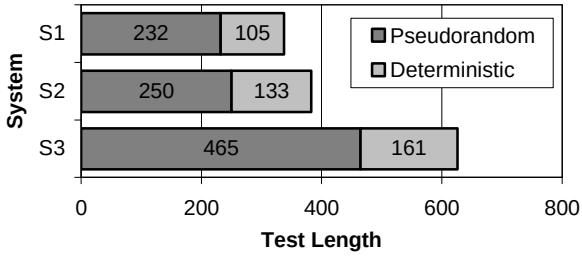


*Figure 3. Test Schedules*

## 6. Conclusions

We have presented a new architecture for the hybrid BIST in multi-core systems where for the first time the hybrid BIST idea is extended by the concept of test pattern broadcasting, where the deterministic test set of each core is applied in parallel to all other cores in a similar way as the pseudorandom test patterns. For this new architecture we have formulated the task to minimize the total test time of the hybrid BIST at given memory limitations for storing deterministic test patterns.

The problem of finding the exact solution for the formulated task is NP-complete. To overcome the high complexity of the problem we have proposed a straightforward algorithm for calculating a possible combination between pseudorandom and deterministic test sequences, where we neglect the optimal solutions for individual cores in favour of finding a quasioptimal solution for the whole system. The described procedure doesn't guarantee minimal test length, however, the procedure is simple (similar to the greedy algorithm) and fast. The latter is demonstrated also by corresponding experimental results.

Although the current work covers only combinatorial circuits, it can easily be extended also for full-scan sequential circuits and can be considered as a future work.

The method proposed can be used first, as a cheap practical solution, and second, as a quickly computable reference for comparison with more sophisticated optimization algorithms to be developed in future.

| System Name | Number of cores | Memory Constraint | Pseudorandom Test Length | Deterministic Test Length | Total Test Length | CPU Time Used (sec) |
|---|---|---|---|---|---|---|
| S1 | 6 | 20 000 | 85 | 181 | **266** | 187, 64 |
| | | 10 000 | 232 | 105 | **337** | |
| | | 5 000 | 520 | 55 | **575** | |
| S2 | 7 | 20 000 | 92 | 222 | **314** | 718.49 |
| | | 10 000 | 250 | 133 | **383** | |
| | | 5 000 | 598 | 71 | **669** | |
| S3 | 5 | 20 000 | 142 | 249 | **391** | 221,48 |
| | | 10 000 | 465 | 161 | **626** | |
| | | 5 000 | 1 778 | 88 | **1866** | |

*Table 2. Experimental Results*

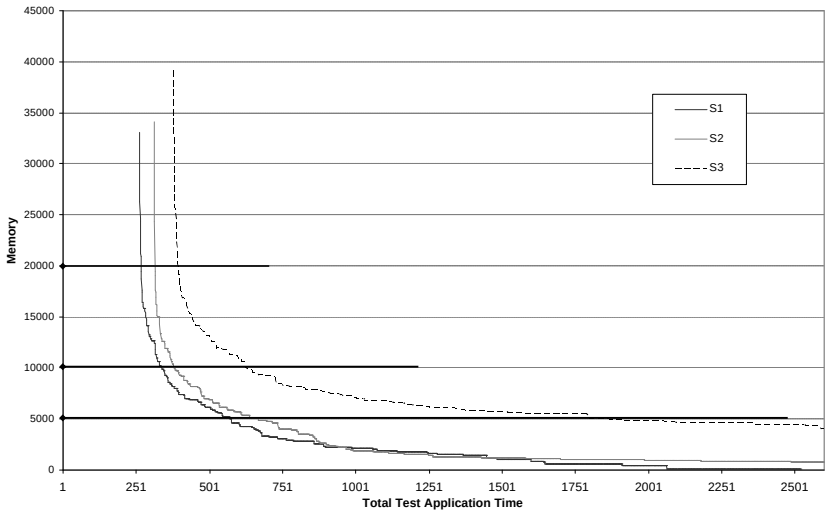


*Figure 4. Memory cost as the function of total test length*

## 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] 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, Yamanashi, Japan, October 2000.

[3] G. Jervan, P. Eles, Z. Peng, R. Ubar, M. Jenihhin, "Test Time Minimization for Hybrid BIST of Core-Based Systems," *IEEE Asian Test Symposium 2003 (ATS'03)*, Xian, China, November 2003 (accepted for publication).

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

[5] 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.

[6] 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, Paris, France, March 2000.

[7] 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.

[8] 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.

[9] 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, San Jose, CA, March 2002.

[10] K-J. Lee, J-J. Chen, C-H. Huang, "Broadcasting Test Patterns to Multiple Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.18, No.12, December 1999, pp.1793-1802.

[11] R. Ubar, M. Jenihhin, G. Jervan, Z. Peng, "Hybrid BIST Optimization for Core-based Systems with Test Pattern Broadcasting," *IEEE International Workshop on Electronic Design, Test and Applications (DELTA 2004)*, Perth, Australia, January 2004 (submitted).