An Iterative Approach to Test Time Minimization for Parallel Hybrid BIST Architecture

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

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

Abstract[1]

This paper presents an approach to the test time minimization problem for parallel hybrid BIST with test pattern broadcasting in core-based systems. The hybrid test set is assembled from pseudorandom test patterns that are generated online and deterministic test patterns that are generated off-line and stored in the system. The pseudorandom patterns are broadcasted and applied to all cores in parallel. The deterministic patterns are, on the other hand, generated for particular cores, one at a time, but applied (broadcasted) in parallel to all other cores and used for the rest of the system as pseudorandom patterns. We propose an iterative algorithm to find the optimal combination between those two test sets under given memory constraints, so that the system's testing time is minimized. Our approach employs a fast cost estimation method in order to avoid exhaustive search and to speed-up the optimization process. Experimental results have shown the efficiency of the algorithm to find a near-optimal solution with very few iterations.

1. Introduction

This paper deals with hybrid built-in self-test (BIST) optimization for testing complex systems-on-chip (SoC). In general, testing such systems requires solving a multitude of problems, including test development and scheduling, wrapper design and test access mechanism (TAM) design. Several of these problems have been solved using fixed set of tests that have been developed for every module individually [1]-[4]. Due to different limitations, like test resources and TAM architecture, this may not always lead to the optimal solution for the entire system. Therefore in this work we concentrate on finding the optimal set of tests not for each core individually, but for the entire system, in such a manner that the total system test time is minimized and a set of given design constraints satisfied. All this is done with emphasis towards a parallel hybrid BIST architecture.

A typical self-test approach employs some form of pseudorandom test pattern generators. Due to several reasons, like very long test sequences and random pattern resistant faults [5], this approach alone may not always be efficient. Therefore a hybrid BIST approach, as a possible improvement of a classical logic BIST for testing core-based systems, has been proposed [6]-[10]. In a hybrid BIST pseudorandom test patterns are combined with deterministic test patterns, applied from the ATE or generated on fly by the system itself. We have proposed a different approach were pseudorandom patterns are complemented with deterministic test patterns that are generated off-line and stored inside the system [7]. This requires a very small set of deterministic patterns that have to be especially designed in order to shorten the pseudorandom sequence and to increase fault coverage. The efficiency of this approach depends of the ratio of these two test sets. 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.

We have previously analyzed the efficiency of a hybrid BIST solution for single core designs [11] and for multi-core designs with sequential test application [12], where it is assumed, that every core in the system has its own dedicated self-test structure. The assumptions however may not always be correct as not all cores are equipped with dedicated BIST engines and the test controller may become too complex for multi-core designs. Therefore in this paper we present a parallel hybrid BIST approach where all cores in the system are tested in parallel, using test pattern broadcasting for both, pseudorandom and deterministic patterns via a bus.

The following section gives an overview and a general formulation of the test time minimization problem for parallel hybrid BIST. In Section 3 a cost estimation methodology is described, followed by the detailed description of the hybrid test sequence minimization algorithm in section 4. Finally, the experimental results together with conclusions are presented in Sections 5 and 6, respectively.

2. Parallel hybrid BIST time minimization

Let us assume a system S, consisting of n cores C_1, C_2, …, C_n, that are all connected to a bus. A hybrid test set $TH = \{TP, TD\}$ for parallel testing of all the cores $C_k \hat{I} S$ is composed from the pseudorandom test set TP, and deterministic test set TD. The deterministic test sequence is

[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.

assembled, in general, from deterministic test sequences for each individual core $TD = \{TD_1, TD_2, \ldots, TD_n\}$. 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 $\hat{I}$ S. For the rest of the cores C_j $\hat{I}$ S, $1 \le j \ne k \le n$ this sequence can be considered as a pseudorandom sequence. This form of parallel testing is usually referred to as test pattern broadcasting [13]. 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. An example of such a hybrid test set is presented in Figure 1.

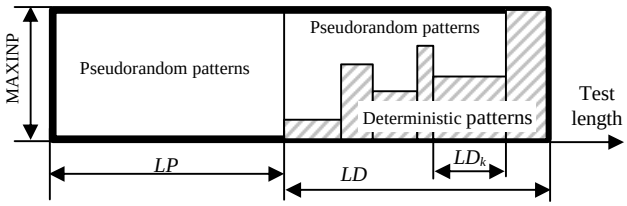

Figure 1. Hybrid test set example

In Figure 1, we denote with LP the length of the pseudorandom test set, with LD the length of the entire deterministic test set, and with LD_k the length of the deterministic test set of core C_k. Since some of the cores may be 100% testable by using only the pseudorandom test sequence and the deterministic test sequences of other cores, the deterministic test sequence TD_k for such a core C_k is not needed and $LD_k = 0$.

The memory size for storing the deterministic part of the hybrid test set can be found from the following formula:

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

The main task of this paper is to minimize the total length

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

of the hybrid test set $TH = \{TP, TD\}$ under given memory constraint $COST_M \le COST_{M,LIMIT}$.

The problem of minimizing the hybrid BIST 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 $n*n$ relationships for n cores to analyse to find the optimal combination; on the other hand the deterministic test sets are not readily available (see Algorithm 3) 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 [11]; but as the optimal time moment for switching from pseudorandom to deterministic testing 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 cope with the high complexity of the problem we propose the following algorithm:

Algorithm 1:

1. Find the best initial state for the LFSR that can generate the shortest common pseudorandom sequence $TP_{INITIAL}$, sufficient for testing simultaneously all the cores with maximum achievable fault coverage. Due to practical reasons the $TP_{INITIAL}$ might be unacceptably long and therefore an adequately long $TP'_{INITIAL}$ should be chosen and complemented with an initial deterministic test set $TD_{INITIAL}$ in order to achieve maximum achievable fault coverage and to satisfy the basic requirements for the test length.
2. Based on our estimation methodology (Section 3) find the length LD_k^E of the estimated deterministic test set TD_k^E and calculate the first iteration of the optimized test structure $TH^E = \{TP*, TD^E\}$, so that the memory constraints are satisfied. $TP*$ denotes here a shortened pseudorandom sequence, found during the calculations.
3. Find the real total test length LH and the real memory cost $COST_M$ of the hybrid test sequence $TH = \{TP*, TD\}$ for the selected pseudorandom sequence $TP*$.
4. If the memory constraints are not satisfied, i.e., $COST_M > COST_{M,LIMIT}$, improve the estimation (see Section 4), choose a new pseudorandom sequence $TP*$, and repeat step 3.
5. If the memory limit has not been reached, i.e., $COST_M < COST_{M,LIMIT}$, reduce the length of TH by moving efficient pseudorandom patterns [12] from the pseudorandom test set to the deterministic test set. A pattern in a pseudorandom test sequence is called *efficient* if it detects at least one new fault for at least one core that is not detected by previous test patterns in the sequence.

We have recently proposed a fast straightforward algorithm for finding TP and TD for parallel hybrid BIST using test pattern broadcasting [14]. This algorithm, in fact, corresponds to the 5th step of the algorithm proposed above and the disadvantage of the straightforward approach is that the deterministic test set TD is generated based on the initial test sequence $TP_{INITIAL}$ and is not minimized. Minimization of TD (test compaction) would be extremely difficult, since TD is assembled simultaneously for all cores in the system and individual deterministic tests for different cores TD_k are difficult to identify.

Differently from [14], the algorithm we have proposed here uses test cost estimates to find the appropriate initial

pseudorandom sequence $TP_{INITIAL}$, and based on the fault coverage of every individual core C_k, achieved by TP, finds an optimal (compacted) deterministic test sequence TD_k, thus reducing significantly the length of the final hybrid test set. In the following the algorithm will be described in detail.

3. Hybrid test sequence computation based on cost estimates

In this section we explain the first two steps of Algorithm 1. It is assumed that we have found the best initial state for the parallel pseudorandom test pattern generator [14]. Let us call this an initial pseudorandom test sequence $TP_{INITIAL}$.

Estimation of the cost of the deterministic test.

By knowing the structure of the hybrid test set TH the total hybrid test length LH at any given memory constraint $COST_M \le COST_{M,LIMIT}$ could be found in a straightforward way. However, calculation of the exact hybrid test structure is a costly procedure, since it assumes that for each possible length of TP the deterministic test sets TD_k for each core should be calculated and compressed while following the broadcasting idea. This can be done either by repetitive use of the automatic test pattern generator or by systematically analyzing and compressing the fault tables [11]. Both procedures are accurate but time-consuming and therefore not feasible for larger designs.

To overcome the high complexity of the problem we propose an iterative algorithm, where the values of LD_k and $COST_{M,k}$ for the deterministic test sets TD_k can be found based on estimates. The estimation method, that is an extension of the method proposed for sequential hybrid BIST [12], is based on the fault coverage figures of TD_k only, and does not require accurate calculations of the deterministic test sets for not yet detected faults.

The estimation method requires the following: a complete deterministic test set for every individual core, TD_k, together with fault simulation results of every individual test vector FD_k and fault simulation results of the pseudorandom sequence $TP_{INITIAL}$ for every individual core, FP_k. Let us denote with $TP_{INITIAL}(i)$ a pseudorandom sequence with length i.

The length of the deterministic test sequence $LD_k(i)$ and the corresponding memory cost $COST_{M,k}(i)$ for any length of the pseudorandom test sequence $i = LP$ can be estimated for every individual core with the following algorithm:

Algorithm 2:

For each $i = 1, 2, \ldots, LD_k$:

1. Find fault coverage value $F(i)$ that can be reached by a sequence of pseudorandom patterns $TP_{INITIAL}(i)$.
2. Find the highest integer value j, where $FD_k(j) \le F(i)$. The value of j is the required length of the deterministic sequence TD_k to achieve fault coverage $F(i)$.
3. Calculate the estimated length of the deterministic test subsequence $TD^E_k(i)$ as $LD^E_k(i) = LD_k - j$. This is the estimated number of deterministic test patterns needed to complement the pseudorandom sequence $TP_{INITIAL}(i)$, so that 100% fault coverage can be achieved.

This algorithm enables us to estimate the memory requirements of the hybrid BIST solution for any length of the pseudorandom sequence for every individual core and by adding the memory requirements of all individual cores $C_k \in S$ also for the entire system. In a similar manner the length of the pseudorandom sequence LP for any memory constraint can be estimated and this defines uniquely the structure of the entire hybrid test set.

4. Computation and minimization of the hybrid test sequence

The memory cost estimation function helps us to find the length LP^* of the pseudorandom test sequence TP^* for the estimated hybrid test sequence $TH^E=\{TP^*; TD^E\}$. The real length LH of the estimated hybrid test sequence TH^E can be found with the following algorithm.

Algorithm 3:

1. Simulate the pseudorandom sequence TP^* for each core $C_k \hat{I} S$ and find a set of not detected faults $F_{NOT,k.}$ Generate the corresponding deterministic test set TD'_k by using any ATPG tool. As a result a preliminary real hybrid test set will be generated: $TH = \{TP^*; TD'\}$.
2. Order the deterministic test set $TD' = (TD'_1, TD'_2, \ldots, TD'_n)$ in such the way that for each $i<n$, $INP_i \le INP_{i+1}$.
3. Perform the analysis of the test pattern broadcasting impact for $i = 2, 3, \ldots n$:
   - calculate a set of not detected faults $F_{NOT,i}$ for the test sequence $(TP^*; TD'_1, TD'_2, \ldots, TD'_{i-1})$;
   - compress the test patterns in TD'_i with respect to $F_{NOT,k}$ by using any test compacting tool.

As a result of the Algorithm 3, the real hybrid test sequence $TH = \{TP^*; TD\} = \{TP^*; TD_1, TD_2, \ldots, TD_n\}$ will be generated. The length of the resulting sequence $LH \le LH^E$ as deterministic test patterns of one core, while broadcasted to the other cores, may detect some additional faults. In general, $LD_k \le LD^E_k$ for every $k = 2, 3, \ldots, n$.

The length of the deterministic test sequence, generated with Algorithm 3, can be considered as a near-optimal solution for the given TAM structure, for all the cores. Ordering of the deterministic test sets, according to the step 2 in Algorithm 3 has the following result: the larger the number of inputs of core C_k the more patterns will broadcasted to C_k from other cores, and hence the chances to reduce its own deterministic test set TD_k are bigger and larger amount of memory can be reduced.

After finding the real deterministic test sequence according to the Algorithm 3, the following three situations may occur:

1. If $COST_M > COST_{M,LIMIT}$ a new iteration of the cost estimation should be carried out. The initial estimation of the pseudorandom test sequence length LP should be updated, and new cost calculation, based on Algorithm 3, should be performed.

2. If $COST_M = COST_{M,LIMIT}$ the best possible solution for the given pseudorandom sequence *TP** is found. $TH = \{TP^*; TD_1, TD_2, \ldots, TD_n\}$.
3. If $COST_M < COST_{M,LIMIT}$ the test length minimization process should be continued by moving efficient test patterns from the pseudorandom test set to the deterministic sequence.

In the following possible steps for further improvement are described in detail.

Iterative procedure for cost estimation.

Let us suppose that our first estimated solution, based on pseudorandom test sequence *TP*, with length *LP*, produces a test structure with total memory requirement "Real $COST_M$" higher than accepted (see Figure 2). A correction of the estimated solution should be made $LP_{NEW} = LP + DLP$ and a new solution "New real $COST_M$" should be calculated based on Algorithm 3. Those iterations should be repeated until the memory constraint $COST_M \leq COST_{M,LIMIT}$ is satisfied.

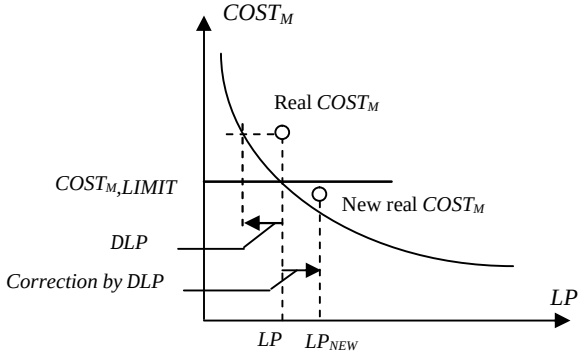

Figure 2. Iterative cost estimation

It should be mentioned that the Algorithm 3 is the most expensive procedure of the whole approach, due to repetitive use of ATPG and test compaction tools. Therefore we cannot start with an arbitrary initial solution and accurate estimation procedure minimizes the number of iterations considerably.

Total test length reduction by reducing the pseudorandom test sequence

Suppose that the real cost of the found solution is below the memory constraint $COST_M < COST_{M,LIMIT}$. There are two alternatives for further reduction of the test length:

1. Additional iterations by using Algorithm 3 to move the solution as close to the memory limit $COST_{M,LIMIT}$ as possible. As mentioned earlier, Algorithm 3 is an expensive procedure and therefore recommended to use as little as possible.
2. It is possible to minimize the length of the hybrid test sequence *TH* by shortening the pseudorandom sequence, i.e. by moving step-by-step efficient patterns from the beginning of *TP* to *TD* and by removing all other patterns between the efficient ones from *TP*, until the memory constraint $COST_M = COST_{M,LIMIT}$ gets violated. This procedure is based on the algorithm used in [14] for straightforward optimization of the parallel hybrid BIST. As a result the final hybrid test sequence is created: $TH_F = \{TP_F; TD_F\} = \{TP_F; TD_1, TD_2, \ldots, TD_n, DTD\}$ where *DTD* is a set of efficient test patterns moved from *TP* to *TD*. This will lead to the situation where the length of the pseudorandom sequence has been reduced by *DLP* and the length of the deterministic test sequence has been increased by *DLD*. The total length LH_F of the resulting hybrid test $TH_F = \{TP_F; TD_F\}$ is shorter, $LH_F < LH$, because in general $DLD << DLP$ (not every pattern in the pseudorandom test set is efficient).

The final hybrid BIST test structure $TH_F = \{TP_F; TD_F\}$ with the total length LH_F is represented in Figure 3.

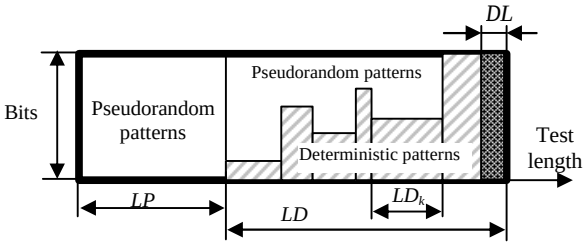

Figure 3. Final hybrid test structure

The accuracy of the solution (proximity of the total length LH_F to the global minimum LH_{MIN}) for the given initial pseudorandom sequence $TP_{INITIAL}$ can be estimated by the length of *DLD*, assuming that the deterministic test set was optimally compacted. Since efficient patterns, moved from *TP* to *TD*, were not taken into account during the compaction procedure for *TD' (algorithm 3)* the new deterministic test sequence $TD_F = \{TD_1, TD_2, \ldots, TD_n, DTD\}$ is not optimal and should be compacted as well. However, since *TD'* was compacted optimally, the upper bound of the gain in test length cannot be higher than *DLD*. Hence, the difference between the exact minimum LH_{MIN} and the current solution LH_F for the given pseudorandom sequence $TP_{INITIAL}$ cannot be higher than $LH_F - LH_{MIN} = DLH$.

5. Experimental results

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	*List of cores*
S1	c5315, c880, c432, c499, c499, c5315
S2	c432, c499, c880, c1355, c1908, c5315, c6288
S3	c880, c5315, c3540, c1908, c880

Table 1. Systems used for experiments

In Table 2 the experimental results for those 3 systems under different memory constraints are presented. In column 3 the estimated length of the hybrid test structure, found by using Algorithm 2, is given. For the systems S1 and S2 only a single iteration was needed (the estimation

System	Memory constraint (bits)	Estimated initial test length (clocks)		Calculated test structure		Final test structure		
				Initial (without broadcasting)	With broadcasting	PR length (clocks)	DET length (clocks)	Total length
S1 (6 cores)	10000	604		596 clocks 8660 bits	548 clocks 4500 bits	145	58+49 = 107	**252 clocks** 9891 bits
S2 (7 cores)	10000	374		399 clocks 12345 bits	335 clocks 8294 bits	163	110+14 = 124	**287 clocks** 9852 bits
	5000	741		740 clocks 4197 bits	717 clocks 3117 bits	469	51+18 = 69	**538 clocks** 4979 bits
	3000	1251		1245 clocks 1547 bits	1240 clocks 1342 bits	783	23+19 = 42	**825 clocks** 2995 bits
S3 (5 cores)	10000	1st iteration	367	412 clocks 13285 bits	379 clocks ***10047*** bits	Need to make another iteration		
		2nd iteration	457	489 clocks 10952 bits	466 clocks 8756 bits	262	130+10 = 140	**402 clocks** 9919 bits

Table 2. Experimental data from three systems

System	Memory constraint (bits)	Straightforward approach				Our approach				Comparison	
		PR length (clocks)	DET length (clocks)	Total length (clocks)	CPU time (sec)	PR length (clocks)	DET length (clocks)	Total length (clocks)	CPU time (sec)	Total test length	CPU time
S1 (6 cores)	10000	232	105	**337**	187,64	145	58+49 = 107	**252**	289,73	– 25.2 %	+54.5 %
S2 (7 cores)	10000	250	133	**383**	718,49	163	110+14 = 124	**287**	1093,5	– 25.1 %	+52.2 %
	5000	598	71	**669**		469	51+18 = 69	**538**	1124,4	– 19.6 %	+56.5 %
	3000	819	48	**867**		783	23+19 = 42	**825**	1109,4	– 4.8 %	+54.4 %
S3 (5 cores)	10000	465	161	**626**	221,48	262	130+10 = 140	**402**	334,28	– 35.8 %	+51.1 %

Table 3. Comparison with straightforward approach

was rather exact), for the system S3 two iterations were needed. In columns 4 and 5 the real length of the hybrid test sequence, found by using Algorithm 3 with cost estimates from column 3, is given. In column 4 the total length of the pseudorandom and deterministic test sequences and the memory cost in bits is given without taking into account broadcasting effect (step 1 in Algorithm 3), and in column 5 together with broadcasting (steps 2 and 3 in Algorithm 3). In the columns 6-8 the results of the final optimization are depicted: the final length of the pseudorandom sequence (column 6), the final length of the deterministic sequence (column 7), and the final length of the hybrid test set together with memory requirements in bits (column 8). In column 7 the first component represents the result of the Algorithm 3, and the second component represents the last improvement, when efficient patterns were moved from the pseudorandom part to the deterministic part.

In Table 3 the results are compared with the straightforward approach [14]. The length of the pseudorandom test sequence (columns 3, 7), deterministic test sequence (columns 4, 8) and the hybrid test sequence (columns 5, 9) together with required CPU time (columns 6, 10) are compared. As it can be seen, the proposed approach gives a noteworthy reduction of the test length while the analysis time is approximately the same.

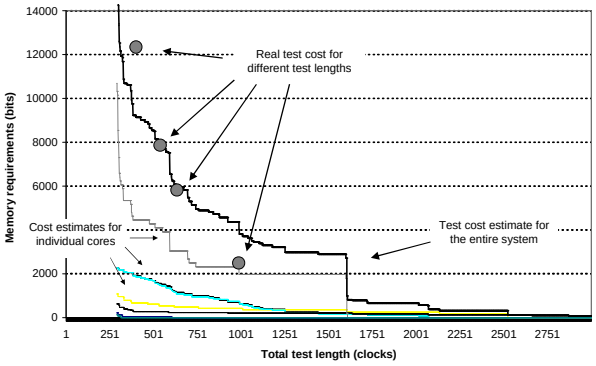

Figure 4. Comparison of estimated and real test costs

In Figure 4 the estimated memory cost as the function of the total test length for different cores in system S2 together with the estimated total memory cost are depicted. For comparison the real costs values for 4 different test lengths are shown as well. As it can be seen the accuracy of the estimation procedure is rather good.

Figure 5 illustrates the estimation curves for the best and worst initial sequences and highlights the three memory constraints given in Table 3. This illustrates the importance of choosing the best possible pseudorandom sequence for testing the system.

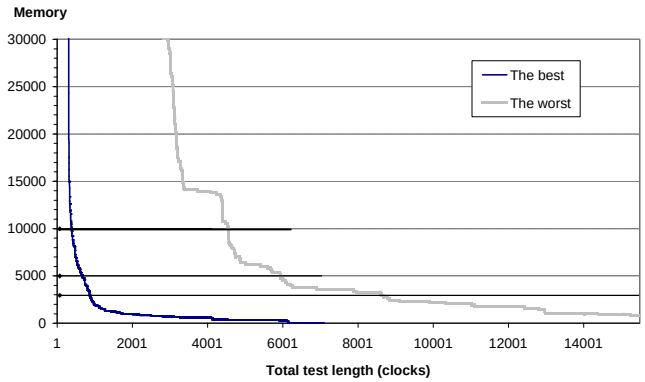

Figure 5. Memory usage as the function of the test length for the best and worst initial pseudorandom sequence

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. Due to the very high switching activity, that occurs during parallel testing, the power dissipation has to be investigated in the future as well and to be included to the analysis process as an additional constraint.

6. Conclusions

We have presented an approach to the test time minimization for parallel hybrid BIST in core-based digital systems under given memory constraints. The approach is based on analysis of different cost relationships as functions of the hybrid test structure. To deal with the high complexity of the problem and to avoid the exhaustive exploration of the design space, a method for estimating the memory cost of the deterministic component of the hybrid test set was proposed. We have 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. This method leads to a near-optimal solution with low computational overhead. We introduced also a method to estimate the proximity of the solution to the exact optimum, and formulated the conditions when the estimation is exact. Experimental results have demonstrated efficiency of the algorithm, and the optimized results are significantly better compared to the solution found by the straightforward approach.

References:

[1] 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.

[2] 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.

[3] V. Iyengar, K. Chakrabarty, E. J. Marinissen, "Test Wrapper and Test Access Mechanism Co-Optimization for System-on-chip," *Journal of Electronic Testing; Theory and Applications (JETTA)*, Vol. 18, No. 2, pp. 213-230, April 2002.

[4] 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.

[5] E. B. Eichelberger, E. Lindbloom, "Random-Pattern Coverage Enhancement and Diagnosis for LSSD Logic Self-Test," *IBM Journal of Research & Development*, Vol. 27, No. 3, pp. 265-272, May 1983.

[6] S. Hellebrand, H.-J. Wunderlich, A. Hertwig, "Mixed-Mode BIST Using Embedded Processors," *Journal of Electronic Testing: Theory and Applications (JETTA)*, Vol. 12, No. 1-2, pp. 127-138, February-April 1998.

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

[8] D. Das, N. A. Touba, "Reducing Test Data Volume Using External/LBIST Hybrid Test Patterns," *International Test Conference*, pp. 115-122, Atlantic City, NJ, USA, October 2000.

[9] S. Hellebrand, H.-G. Liang, H.-J. Wunderlich, "A Mixed Mode BIST Scheme Based on Reseeding of Folding Counters," *International Test Conference*, pp. 778-784, Atlantic City, NJ, USA, October 2000.

[10] R. Dorsch, H.-J. Wunderlich, "Tailoring ATPG for Embedded Testing," *International Test Conference*, pp. 530-537, Baltimore, MD, USA , October 2001.

[11] 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, USA, March 2002.

[12] G. Jervan, P. Eles, Z. Peng, R. Ubar, M. Jenihhin, "Hybrid BIST Time Minimization for Core-Based Systems with STUMPS Architecture," *18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'03)*, pp. 225-232, Cambridge, MA, USA, November, 2003.

[13] 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, pp.1793-1802, December 1999.

[14] 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 (accepted for publication).