Notice of Violation of IEEE Publication Principles

“Time Minimization of hybrid BIST for systems-on-chip”
by I. Popa, D. Cazacu
in the 31st International Spring Seminar on Electronics Technology (ISSE 2008), 2008, pp. 153 – 158

After careful and considered review of the content and authorship of this paper by a duly constituted expert committee, this paper has been found to be in violation of IEEE's Publication Principles.

This paper contains significant portions of original text from the paper cited below. The original text was copied with insufficient attribution (including appropriate references to the original author(s) and/or paper title) and without permission.

Due to the nature of this violation, reasonable effort should be made to remove all past references to this paper, and future references should be made to the following article:

“Test Time Minimization for Hybrid BIST of Core-Based Systems”
by Gert Jervan, Petru Eles, Zebo Peng, Raimun Ubar, Maksim Jenihhin
in the 12th IEEE Asian Test Symposium (ATS03), 2003, pp. 318 - 323

Time Minimization of Hybrid BIST for Systems-on-chip

Ilie POPA, Dumitru CAZACU
University of Pitesti, Electronics and Computers Department,
Street Targul din Vale, No. 1, Pitesti, Arges, 110040, Romania
e-mail: cipopa@upit.ro

Abstract. *In this paper, we concentrate on hybrid BIST optimization for multi-core designs. As total cost minimization for multi-core systems is an extremely complex problem and is rarely used in reality, the main emphasis here is on test time minimization under memory constraints with different test architectures. The memory constraints can be seen as limitations of on-chip memory or ATE memory, where the deterministic test set will be stored, and therefore with high practical importance. We will concentrate on one large classes of test architectures and we assume that every core is equipped with its own pseudorandom pattern generator and only deterministic patterns have to be transported to the cores. For this architecture we will describe test-per-clock as well as test-per-scan application schemes.*

Keywords*: Design-For-Test (DFT); Build-In-Self-Test (BIST); Test Pattern Generator (TPG); BIST*

1. INTRODUCTION

Now, microelectronics technology provides designers the possibility to integrate a large number of different functional blocks, usually referred as cores, in a single Integrated Circuit. 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.

The main idea of a BIST approach is to eliminate or reduce the need for an external tester by integrating active test infrastructure onto the chip. The test patterns are not any more generated externally, as it is done with ATE, but internally, using special BIST circuitry. BIST techniques can be divided into *off line* and *on-line* techniques . On-line BIST is performed during normal functional operation of the chip, either when the system is in idle state or not. Off-line BIST is performed when the system is not in its normal operational mode but in special test mode. A typical BIST architecture, presented in Figure 1, consists of a TPG, a TRA, and a *BIST control unit* (BCU), all implemented on the chip [3, 4, 5, 6]. This hybrid BIST approach is based on the intelligent combination of pseudorandom and deterministic test sequences that would provide a high-quality test solution under imposed constraints.

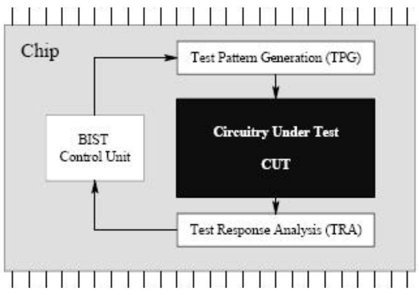

Figure 1. A typical BIST architecture.

There are following main criteria used for optimization of BIST design in one-core or in multi-core design: total cost minimization, time minimization and energy minimization. In this paper we concentrate on hybrid BIST optimization for multi-core designs.

As total cost minimization for multi-core systems is an extremely complex problem and is rarely used in reality, the main emphasis here is on test time minimization under memory constraints with different test architectures. The memory constraints can be seen as limitations of on-chip memory or ATE memory, where the deterministic test set will be stored, and therefore with high practical importance. We will concentrate on two large classes of test architectures. In one case we assume that every core is equipped with its own pseudorandom pattern generator and only deterministic patterns have to be transported to the cores. In the second case we assume test pattern broadcasting, where both pseudorandom and deterministic test patterns have to be transported to the cores under test. For both architectures we will describe test-per-clock as well as test-per-scan application schemes.

2. Parallel Hybrid BIST Architecture and Possibilities for Time Minimization

It is start with a test architecture where every core has its own dedicated BIST logic that 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 [1, 2]. At the system level, however, only one test access bus is assumed, thus the deterministic tests can only be carried out for one core at a time. Such architecture assumes that all patterns from the same test set (pseudorandom or deterministic) for the same CUT have the same test application time, thus simplifying test cost calculations. An example of a multicore system, with such a test architecture is given in Figure 2. In order to explain the test time minimization problem for multicore systems, let us use an example design, consisting of 5 cores (Figure 2).

Using the hybrid BIST optimization methodology, described in [2], Chapter 5, is possible to find the optimal combination between pseudorandom and deterministic test patterns for every individual core. 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 adhoc scheduling method can be used.

For portable systems, one of the most important test constraints is the total amount of on-chip test memory. In the previous chapters it is introduced several methods, based on different test architectures, for test time minimization. Those algorithms were able to find a shortest possible test time under given test memory constraint for test-per-clock and test-per-scan schemes. These algorithms, however, do not require explicit specification, whether the deterministic test set has to be stored in the ATE or on-chip memory (ROM). In the latter case hybrid BIST solutions can be used not only for manufacturing test but also for periodical field maintenance tests in portable devices.

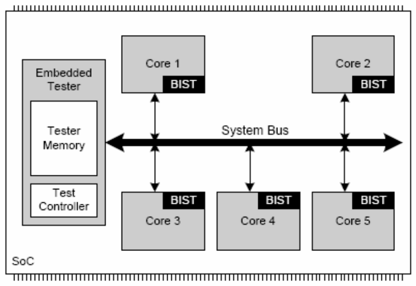

Figure 2. An example of a core-based system, with independent BIST resources.

In the following were assumed a hybrid BIST test architecture where all cores have their own dedicated BIST logic that is capable of producing a set of independent pseudorandom test patterns. The deterministic tests, on the other hand, are applied from an on-chip memory, one core at a time.

If the objective is only test time minimization and power/energy is not taken into account then the shortest test schedule for such a test architecture, is the one where all cores are tested concurrently and have the same tests lengths. It is important to note that the exact composition of pseudorandom and deterministic test patterns defines not only the test length and test memory requirements but also the energy consumption. In general, since a deterministic test pattern is more effective in detecting faults than a pseudorandom pattern, using more deterministic test patterns for a core will lead to a short test sequence, and consequently less energy on the average case. However, the total number of deterministic test patterns is constrained by the test memory requirements, and at the same time, the deterministic test patterns of different cores of a SOC have different energy and fault detection characteristics. Namely, some cores might require only few test patterns, while these patterns have large memory requirements, due to big number of inputs or excessively long scan chains. These few patterns might generate less switching activity than a long sequence of test patterns with smaller memory requirements. A careful trade-off between the deterministic pattern lengths of the cores must therefore be made in order to produce a globally optimal solution. Moreover, as deterministic test patterns are stored in the memory, energy dissipation during memory access and test data transportation has to be taken into account as well.

In a hybrid BIST approach the test set is composed of pseudorandom and deterministic test patterns, where the ratio of these patterns is defined by different design constraints, such as test memory and test time. From a energy perspective, different cores have different energy dissipation while applying the same amount of test patterns. Furthermore, the pseudorandom and deterministic test sequences for the same core have different power characteristics. Therefore, for total energy minimization it is important to find, for every individual core, such a ratio of the pseudorandom and deterministic test patterns that leads to the overall reduction of switching energy. At the same time the basic design constraints, such as test memory, should not be violated. In the following some basic definitions together with the problem formulation are given.

3. Basic definitions and Test Set Generation Based on Cost Estimates

Let us assume that a system S consists of n cores $C_1, C_2, \ldots, C_n$. For every core $C_k \in S$ a complete sequence of deterministic test patterns TD_k^F and a complete sequence of pseudorandom test patterns TP_k^F can be generated.

Definition 1: A *hybrid BIST* set $TH_k = \{TP_k, TD_k\}$ for a core C_k is a sequence of tests, constructed from a subset $TP_K \subseteq TD_k^F$ of the pseudorandom test sequence, and a deterministic test sequence $TP_K \subseteq TD_k^F$. The sequences TP_k and TD_k complement each other to achieve the maximum achievable fault coverage [2, 5].

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. An efficient pseudorandom test sequence TPE_k, which includes all efficient patterns of TP_k^F is called *full efficient pseudorandom test sequence* and denoted by TPE_k^F [2].

Definition 3: The function cost of a hybrid test set TH_k for a core C_k is determined by the total length of its pseudorandom (*TP*) and deterministic (*TD*) test sequences, which can be characterized by their costs, $COST_{P,k}$ and $COST_{D,k}$ respectively [1, 2]:

$$COST_{T,k} = COST_{D,k} + COST_{D,k} = \sigma|TP_k| + \varphi_k|TD_k| \quad (1)$$

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

$$COST_{M,k} = \gamma_k|TD_k| \quad (2)$$

The parameters σ and φ_k (k=1, 2, ..., n) 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 $\sigma = 1$. The parameter φ_k for a particular core C_k is equal to the total number of clock cycles needed for applying one 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 $TH = \{TH1, TH2, \ldots, THn\}$ for a system $S = \{C1, C2, \ldots, Cn\}$ consists of hybrid tests THk for each individual core Ck, where the pseudorandom components of 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 *characteristic vector* of a hybrid test set $TH = \{TH_1, TH_2, \ldots, TH_n\}$ is $J = (j_1, j_2, \ldots, j_n)$, where $j_k \in J$ is the length of the pseudorandom test sequence $TP_k \subseteq TH_k$. and n is number of cores on the chip [1, 2, 4].

According to *Definition 2*, for each j_k corres-ponds a pseudorandom subsequence $TP_K(j_k) \subseteq TD_k^F$, 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 THk for all cores $C_k \in S$.

Definition 6: 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 [2]:

$$COST_T = \max\left\{\max_k(\sigma|TP_k| + \varphi_k|TD_k|), \sum_k \varphi_k|TD_k|\right\} \quad (3)$$

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

$$COST_M = \sum_k COST_{M,k} \qquad (4)$$

Definition 7: 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 $j = 1, 2, \ldots, |TPE^F{}_k|$, is possible to create the cost functions $COST_{T,k} = f_{T,k}(j)$, and $COST_{M,k} = f_{M,k}(j)$. By taking the inverse function $j = f^{-1}{}_{T,k}(COST_{T,k})$, and inserting it into the $f_{M,k}(j)$ is possible to get the generic cost function $COST_{M,k} = f_{M,k}(f^{-1}{}_{T,k}(COST_{T,k})) = f_k(COST_{T,k})$ 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 $C_k \in S$. The objective here is to find a shortest possible ($min(COST_T)$) hybrid test sequence TH_{OPT} when the memory constraints is respected, $COST_M \leq COST_{M,LIMIT}$.

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.

By knowing the generic cost function $COST_M = f(COST_T)$, the total test length $COST_T$ at any given memory constraint $COST_M \leq COST_{M,LIMIT}$ 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 $j = 1, 2, \ldots, |TPEF_k|$ has to be available. This assumes that after every efficient pattern $Pj \subseteq TPE_k \subseteq TP_k$, $j = 1, 2, \ldots, |TPEF_k|$ a set of not yet detected faults $F_{NOT,k}(j)$ should be calculated. This can be done by repetitive use of the automatic test pattern generator or by systematically analyzing and compressing the fault tables for each j. Both algorithms are time-consuming and therefore not feasible for larger designs. To overcome the complexity explosion problem we propose an iterative algorithm, where costs $COST_{M,k}$ *an*d $COST_{D,k}$ 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 description will use $FD_k(i)$ and $FPE_k(i)$ to denote the fault coverage figures of the test sequences $TD_k(i)$ and $TPE_k(i)$, respectively, where i is the length of the test sequence.

***Procedure* 1 -** Estimation of the length of the deterministic test set TD_k [2]:

1. Calculate, by fault simulation, the fault coverage functions $FD_k(i)$, $i = 1, 2, \ldots, \left|TD_k^F\right|$, and $FPE_k(i)$, $i = 1, 2, \ldots, \left|TPE_k^F\right|$. The patterns in $\left|TD_k^F\right|$ are ordered in such a way that each pattern put into the sequence contributes with maximum increase in fault coverage;

2. For each $i^* \leq \left|TPE_k^F\right|$, find the fault coverage value F^* that can be reached by a sequence of patterns $(P_1, P_2, \ldots, P_{i^*}) \subseteq TPE_k$;

3. By solving the equation $FD_k(i) = F^*$, find the maximum integer value j^* that satisfies the condition $FD_k(j^*) \leq F^*$. 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 $\left|TD_k^E(i^*)\right| = \left|TD_k^F\right| - j^*$ which is the number of test patterns needed from the TD_k^F to reach to the maximum achievable fault coverage. The value $\left|TD_k^E(i^*)\right| = \left|TD_k^F\right| - j^*$,calculated by *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 $|TDE_k(j)|$ for all $j = 1, 2, \ldots, \left|TPE_k^F\right|$ it is possible to determine the cost function $COST_{D,k}^E(j)$. Using $COST_{D,k}^E(j)$, other cost function estimates $COST_{M,k}^E(j)$, $COST_{T,k}^E$ and $COST_{M,k}^E = f_k^E(COST_{T,k}^E)$ can be created according to the *Definition3* and *Definition7*. Finally, by adding cost estimates $COST_{M,k}^E = f_k^E(COST_{T,k}^E)$ of all cores, it is possible to determine the hybrid BIST cost function estimate $COST_M^E = f^E(COST_T^E)$ for the whole system.

4. 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 timeconsuming. Therefore, will use the cost estimates, calculated by *Procedure* 1 in the previous section, instead. Using estimates can give us a close to minimal solution for the test length of the hybrid test at given memory constraints. After obtaining this solution, the cost estimates can be improved and another, better solution can be calculated. This iterative procedure will be continued until we reach the final solution.

Procedure 2 - Test length minimization [2]:

1. Given the memory constraint $COST_{M,LIMIT}$, find the estimated total test length $COST_k^{E*}$ as a solution to the equation $f^E(COST_T^E) = COST_{M,LIMIT}$;

2. Based on $COST_T^{E*}$, find a candidate solution $J^* = (j^*_1, j^*_2, \dots, j^*_n)$ where each j^*_k is the maximum integer value that satisfies the equation $COST_{T,k}^E(j_k^*) \le COST_T^{E*}$;

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 $\left|COST_M^* - COST_{M,LIMIT}\right|$ 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 $COST_T^{E,N}$ 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 $J^* = (j^*_1, j^*_2, \dots, j^*_n)$ is the solution.

To illustrate the above procedure, in Figure 3 and Figure 4, an example of the iterative search for the shortest length of the hybrid test is given. Figure 3 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 [7].

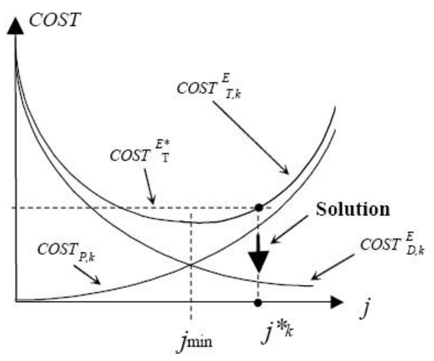

Figure 3. Cost curves for a given core C_k.

Figure 4 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 4). Then (Step 2), based on $COST^{E*}_T$ the length j^*_k of TPE_k for the core C_k in Figure 3 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 4). As it can see in Figure 4 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 4). 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 4). An additional iteration via points 3 and 3^* can be followed in Figure 4.

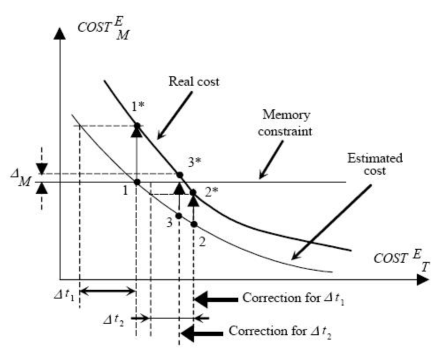

Figure 4. Minimization of the test length.

It is easy to see that Procedure 2 always converges. By each iteration it is 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 the estimated cost functions.

5. Experimental Results

We have performed experiments with several systems composed from different ISCAS benchmarks as cores. Circuits used are for [10]: core 1 -C2670; core 2 - C3540; core 3 -C1908; core 4 -C880; core 5 - C1355; core 6-C432. For the BIST part a Parallel Shift Register Sequence Generator (STUMPS) [8, 9] architecture was used. The results are presented in Table 1. In Table 1 is presented the comparison between the approach from this paper, where the test length is found based on estimates, and an 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, the approach from this paper can give significant speedup (more than one order of magnitude), while retaining acceptable accuracy (biggest deviation is less than 10.5% from the fault table based solution, in average 2.77%).

6. Conclusions

The paper present algorithms for SOC test time minimization with several different implementations of the hybrid BIST architecture.

The algorithms are based on the analysis of different cost relationships as functions of the hybrid BIST structure. It is propose straightforward algorithms and more sophisticated iterative heuristics for this purpose.

The presented methods can minimize test time under given test memory constraint for test-per-clock and test-per-scan schemes. The optimization algorithms have been presented together with experimental results to demonstrate their efficiency.

REFERENCES

[1] Popa Ilie, I. Liță – *„Hibrid BIST Energy Minimization”*, 13th SIITME 2007,Baia Mare, 20-23 Sept. 2007, pp 244-248

[2] G. Jervan, *“Hybrid Built-In Self-Test and Test Generation Techniques for Digital Systems”,* Department of Computer and Information Science, Linköpings Universitet, SE-581 83 Linköping, Sweden, Linköping 2005

[3] Ilie Popa, Ioan Liță – *„Usage of Desin-For-Test Concept in Designing of the Electronics Modules”*, The 12-th International Symposium for Design and Technology of Electronic Packages, Iasi, 21-24 Sept. 2006, Proceedings pp. 263 – 267.

[4] I. Popa, I. Liță, A. M. Chiță, „*The Concepts, Methods and Technics for The Electronic Testability Implementation*, The 24th ISSE 2001, 5–9 May, 2001, Calimănești-Căciulata, Romania, Proceedings pp. 263 – 267.

[5] www.eecs.umich.edu/~jhayes/iscas

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

[7] G. Jervan, *“Test Cost Minimization forHybrid BIST*” IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 283-291, 2000.

[8] G. Jervan, P. Eles, Z. Peng, R. Ubar, M. Jenihhin, “Hybrid BIST Time Minimization for Core-Based Systems with STUMPS Architecture,” *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 225-232, 2003.

[9] P. H.Bardell, W. H.McAnney, “*Self-Testing of Multichip, Logic Modules*” IEEE International Test Conference,pp.200-204,1982.

[10] www.eecs.umich.edu/~jhayes/iscas.restore/

Table 1. Experimental simulated results

System	Number of cores	Memory Constraints	Fault table based approach		In approach from this paper	
			Total test length (clocks)	CPU[1] time (seconds)	Total test length (clocks)	CPU[1] time (s)
S1	5	2048	8349	8723.10	8355	408.25
		4096	2827		2916	300.57
		8192	472		501	185.58
S2	6	4096	323	2704.19	351	163,66
		8192	399		399	46,76
		16384	182		183	66,80

[1] CPU time for calculating all possible hybrid BIST solutions.