Nov. 2006, Vol.21, No.6, pp.907–912 J. Comput. Sci. & Technol.

Test Time Minimization for Hybrid BIST of Core-Based Systems

Gert Jervan[1,2], Petru Eles[1], Zebo Peng[1], Raimund Ubar[2], and Maksim Jenihhin[2]

[1] *Embedded Systems Laboratory (ESLAB), Linköping University, Sweden*

[2] *Department of Computer Engineering, Tallinn University of Technology, Estonia*

E-mail: {gerje, petel, zebpe}@ida.liu.se; {gerje, raiub, maksim}@pld.ttu.ee

Received June 15, 2004; revised May 19, 2006.

Abstract 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 near optimal solutions.

Keywords SoC, self-test, hybrid BIST

1 Introduction

Today's microelectronics technology provides designers with 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, mainly due to the resulting IC's complexity and the high integration density[1].

There exists extensive work for testing core-based systems[2−9]. 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)[10]. In order to apply at-speed tests and to keep the test costs under control, on-chip test solutions, usually referred to as built-in self-test (BIST), are becoming a mainstream technology. Different BIST strategies are known, among them hybrid BIST[11].

Our earlier work[11−13] 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 time is minimized, and the maximum achievable fault coverage is guaranteed.

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 the basic definitions, cost functions and detailed 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.

2 Hybrid BIST Architecture

In this paper we assume the following test architecture: 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. The deterministic tests are applied from the external source (either on-chip memory or ATE), one core at a time; and in the current approach we have assumed for test data transportation an AMBA-like test bus[14]. AMBA (Advanced Microcontroller Bus Architecture) integrates an on-chip test access technique that reuses the basic bus infrastructure. An example of a multi-core system, with such a test architecture is given in Fig.1.

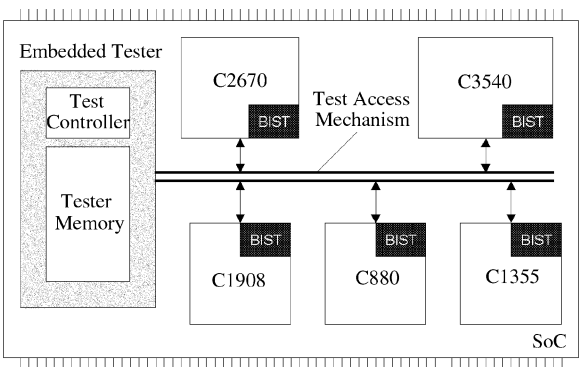

Fig.1. Core-based system example with the proposed test architecture.

Short Paper

Supported by the Estonian Science Foundation grants G6829 and G5910, Enterprise Estonia project Technology Development Centre ELIKO, and the Swedish Foundation for Strategic Research (SSF) under the Strategic Integrated Electronic Systems Research (STRINGENT) program.

This example system consists of 5 cores (different ISCAS benchmarks). Using our hybrid BIST optimization methodology for single core[11] we can find the optimal combination between pseudorandom and deterministic test patterns for every individual core (Fig.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 Fig.2.

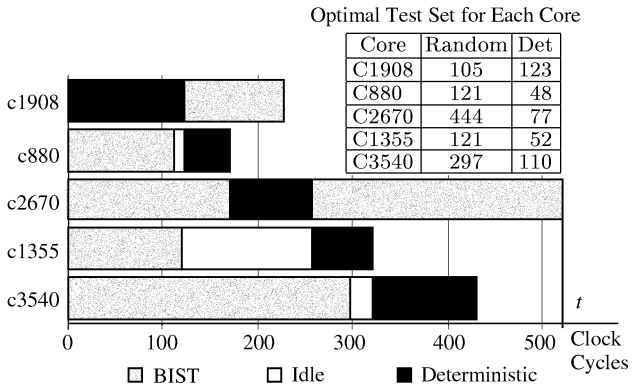

Fig.2. Ad-hoc test schedule for hybrid BIST of the core-based system example.

As seen from Fig.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.

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 $C_1, C_2, \ldots, C_n$. 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 $TP_k \subseteq TP_k^F$ of the pseudorandom test sequence, and a deterministic test sequence $TD_k \subseteq TD_k^F$. The sequences TP_k and TD_k complement each other to achieve the maximum achievable fault coverage.*

Definition 2. *A pattern in a pseudorandom test sequence is called efficient if it detects at least one new fault that is not detected by the previous test patterns in the sequence. The ordered sequence of efficient patterns form an efficient pseudorandom test sequence $TPE_k = (P_1, P_2, \ldots, P_n) \subseteq TP_k$. 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.*

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:*

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

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 $(k = 1, 2, \ldots, n)$ 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. *$\boldsymbol{J} = (j_1, j_2, \ldots, j_n)$ is called the characteristic vector of a hybrid test set $TH = \{TH_1, TH_2, \ldots, TH_n\}$, 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 $TP_k(j_k) \subseteq TP_k^F$, and ac-

cording 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 $\boldsymbol{J}$ determines entirely the structure of the hybrid test set TH_k for all cores $C_k \in S$.

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

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

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

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

Definition 6. *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 $j = 1, 2, \ldots, |TPE_k^F|$, 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 $j = f_{T,k}^{-1}(COST_{T,k})$, and inserting it into the $f_{M,k}(j)$ we get the generic cost function $COST_{M,k} = f_{M,k}(f_{T,k}^{-1}(COST_{T,k})) = f_k(COST_{T,k})$ 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 $\boldsymbol{J}$, 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, i.e., $COST_M \leqslant COST_{M,LIMIT}$.

4 Hybrid Test Sequence Computation Based on Cost Estimates

The test time minimization problem under memory constraints can be solved in a straightforward way if the supplementary deterministic test set for every possible length of the pseudorandom set is available. This can be achieved either by repetitive use of the automatic test pattern generator or by systematically analyzing and compressing the fault tables for each j[13]. Both procedures are accurate but time-consuming and therefore not feasible for larger designs. To overcome the complexity explosion problem we have developed 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[15]. 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 $F_{NOT,k}(j)$.

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. 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 1. Test Length Minimization

1. Given the memory constraint $COST_{M,LIMIT}$, find the estimated total test length $COST_T^{E*}$ as a solution to the equation $f^E(COST_T^E) = COST_{M,LIMIT}$.
2. Based on $COST_T^{E*}$, find a candidate solution $\boldsymbol{J}^* = (j_1^*, j_2^*, \ldots, j_n^*)$, where each j_k^* is the maximum integer value that satisfies the equation $COST_{T,k}^E(j_k^*) \leqslant COST_T^{E^*}$.
3. To calculate the exact value of $COST_M^*$ for the candidate solution $\boldsymbol{J}^*$, 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 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 $COST_T^{E,N}$ from the equation $f_k^E(COST_T^E) = COST_M^*$, and find the difference $\Delta t = COST_T^{E,*} - COST_T^{E,N}$.
7. Calculate a new cost estimate $COST_T^{E,*} = COST_T^{E,*} + \Delta t$ for the next iteration.
8. If the value of $COST_T^{E,*}$ is the same as in an earlier iteration, go to Step 9, otherwise go to Step 2.
9. END: the vector $\boldsymbol{J}^* = (j_1^*, j_2^*, \ldots, j_n^*)$ is the solution.

To illustrate the above procedure, in Figs.3 and 4 an example of the iterative search for the shortest length of the hybrid test is given. Fig.3 represents all the basic cost curves $COST_{D,k}^E(j)$, $COST_{P,k}^E(j)$, and $COST_{T,k}^E(j)$, 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]. Fig.4 represents the estimated generic cost function $COST_M^E = f^E(COST_T^E)$ for the whole system. At first (Step 1), the estimated $COST_T^{E^*}$ for the given memory constraints is found (point 1 on Fig.4). Then (Step 2), based on $COST_T^{E^*}$ the length j_k^* of TPE_k for the core C_k in Fig.3 is found. This procedure (Step 2) is repeated for all the cores to find the characteristic vector $\boldsymbol{J}^*$ of the system as the first iterative solution. After that the real memory cost $COST_M^{E^*}$ is calculated (Step 3, point 1* in Fig.4). As we see in Fig.4, the value of $COST_M^{E^*}$ in point 1* violates the memory constraints. The difference Δt_1 is determined by the curve of the estimated cost (Step 5). After correction, a new value of $COST_T^{E^*}$ is found (point 2 on Fig.4). Based on $COST_T^{E^*}$, a new $\boldsymbol{J}^*$ is found (Step 2), and a new $COST_M^{E^*}$ is calculated (Step 3, point 2* in Fig.4). An additional iteration via points 3 and 3* can be followed in Fig.4.

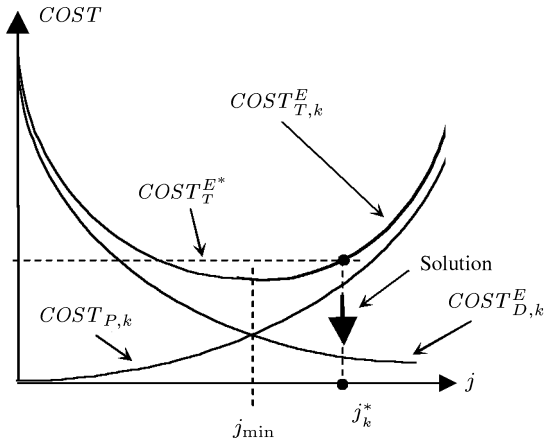

Fig.3. Cost curves for a given core C_k.

It is easy to see that Procedure 1 always converges. By each iteration we get closer to the memory constraints level, and also closer to the minimal test length with the given constraints. However, the solution may be only near-optimal, since we only evaluate solutions derived from the estimated cost functions.

6 Experimental Results

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

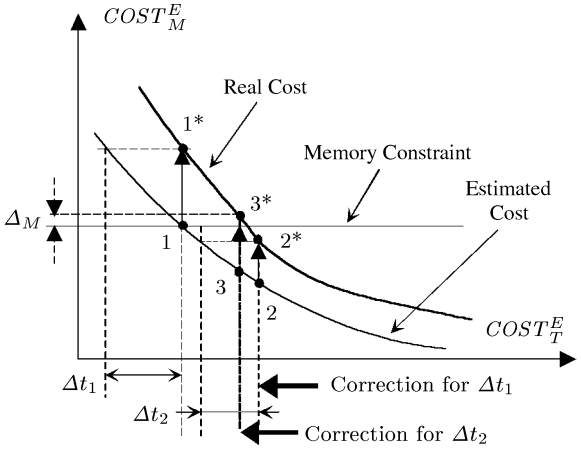

Fig.4. Minimization of the test length.

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. For the exact approach the CPU time includes time needed to calculate the complete deterministic test set and to simulate the pseudorandom sequence, as well as time needed for fault table manipulations. For the proposed approach the CPU time includes time for creating the initial estimates as well as time needed for the final iterations (Procedure 1). As seen from the results, our approach can give significant speedup (more than order of magnitude), while retaining acceptable accuracy (biggest deviation is less than 9% from the exact solution, in average 2.4%).

In Fig.5 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 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 Fig.6. In Fig.7 we show another test schedule, when the memory constraints are different ($M_{LIMIT} = 14000$ bits).

Table 1. Experimental Results

System	Number of Cores	Memory Constraint (bit)	Exact Approach		Our Approach	
			Total Test Length (clock)	CPU Time (s)	Total Test Length (clock)	CPU Time (s)
$S1$	6	20000	222	3772.84	223	199.78
		10000	487		487	57.08
		7000	552		599	114.16
$S2$	7	14000	207	3433.10	209	167.3
		5500	540		542	133.84
		2500	1017		1040	200.76
$S3$	5	7000	552	10143.14	586	174.84
		3500	3309		3413	291.40
		2000	8549		8556	407.96

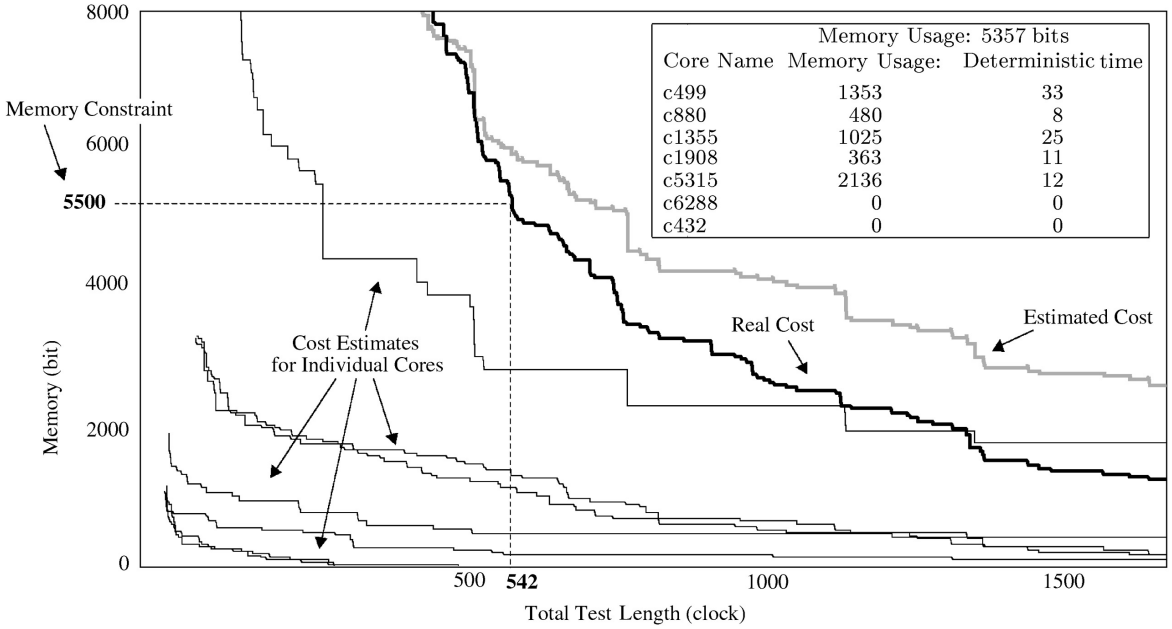

Fig.5. Final test solution for the system $S2$ (M_{LIMIT} = 5500 bits).

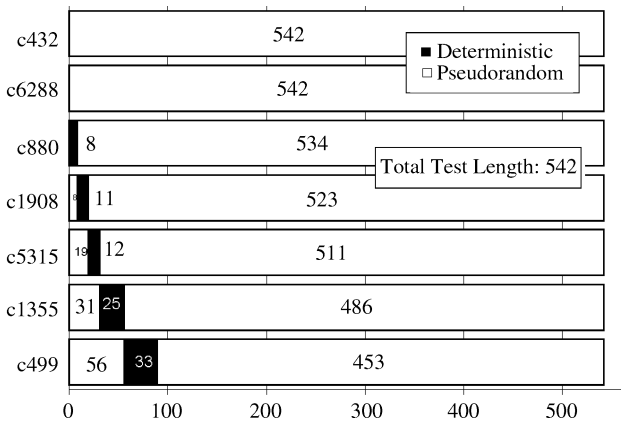

Fig.6. Test schedule for the system $S2$ (M_{LIMIT} = 5500 bits).

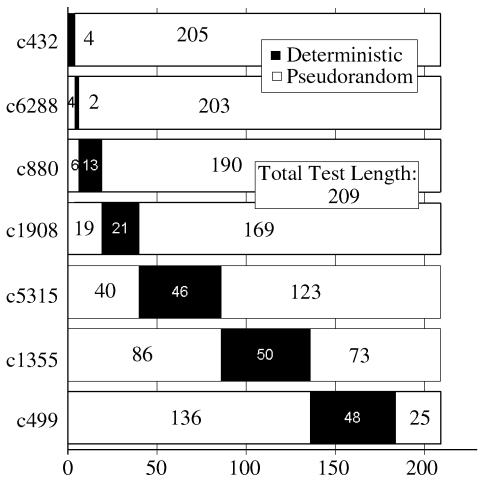

Fig.7. Test schedule for the system $S2$ (M_{LIMIT} = 14000 bits).

7 Conclusions

We have presented an approach to the test time minimization problem for multi-core systems. 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 cost estimation of the deterministic component of the hybrid test set has been 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 the very high speed of the algorithm compared to the exact calculation method.

References

[1] Murray B T, Hayes J P. Testing ICs: Getting to the core of the problem. *IEEE Computer*, Nov. 1996, 29(11): 32–39.

[2] Garg M, Basu A, Wilson T C *et al.* A new test scheduling algorithm for VLSI systems. In *Proc. CSI/IEEE Symposium on VLSI Design*, New Delhi, 1991, pp.148–153.

[3] Zorian Y. A distributed BIST control scheme for complex VLSI devices. In *Proc. IEEE VLSI Test Symposium* (*VTS'93*), Atlantic City, NJ, 1993, pp.4–9.

[4] Chou R, Saluja K, Agrawal V. Scheduling tests for VLSI systems under power constraints. *IEEE Trans. VLSI Systems*, June 1997, 5(2): 175–185.

[5] Chakrabarty K. Test scheduling for core-based systems. In *Proc. IEEE/ACM Int. Conf. Computer Aided Design* (*ICCAD'99*), San Jose, CA, 1999, pp.391–394.

[6] Sugihara M, Date H, Yasuura H. Analysis and minimization of test time in a combined BIST and external test approach. In *Proc. Design, Automation & Test in Europe Conference* (*DATE 2000*), Paris, 2000, pp.134–140.

[7] Muresan V, Wang X, Muresan V *et al.* A comparison of classical scheduling approaches in power-constrained block-test

scheduling. In *Proc. IEEE Int. Test Conference* (*ITC'2000*), Atlantic City, NJ, 2000, pp.882–891.

[8] Chakrabarty K. Test scheduling for core-based systems using mixed-integer linear programming. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, Oct. 2000, 19(10): 1163–1174.

[9] Larsson E, Peng Z. An integrated framework for the design and optimization of SoC test solutions. *Journal of Electronic Testing; Theory and Applications* (*JETTA*), *Special Issue on Plug-and-Play Test Automation for System-on-a-Chip*, Aug. 2002, 18(4/5): 385–400.

[10] Zorian Y, Marinissen E J, Dey S. Testing embedded core-based system chips. In *Proc. IEEE Int. Test Conf.* (*ITC'98*), Washington DC, 1998, pp.130–143.

[11] Jervan G, Peng Z, Ubar R. Test cost minimization for hybrid BIST. In *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems* (*DFT'00*), Yamanashi, 2000, pp.283–291.

[12] Jervan G, Peng Z, Ubar R, Kruus H. A hybrid BIST architecture and its optimization for SoC testing. In *Proc. IEEE 3rd Int. Symp. Quality Electronic Design* (*ISQED'02*), San Jose, CA, 2002, pp.273–279.

[13] Ubar R, Jervan G, Peng Z *et al.* Fast test cost calculation for hybrid BIST in digital systems. *Euromicro Symp. Digital Systems Design*, Warsaw, 2001, pp.318–325.

[14] Flynn D. AMBA: Enabling reusable on-chip designs. *IEEE Micro*, 1997, 17(4): 20–27.

[15] Jervan G, Eles P, Peng Z *et al.* Test time minimization for hybrid BIST of core-based systems. *Asian Test Symposium*, Xian, 2003, pp.318–323.