

Power-Constrained Hybrid BIST Test Scheduling in an Abort-on-First-Fail Test Environment

Zhiyuan He, Gert Jervan, Zebo Peng, Petru Eles
Embedded Systems Laboratory (ESLAB)
Linköping University, Sweden
{zhihe, gerje, zebpe, petel}@ida.liu.se

Abstract¹

This paper presents a method for power-constrained system-on-chip test scheduling in an abort-on-first-fail environment where the test is terminated as soon as a fault is detected. We employ the defect probabilities of individual cores to guide the scheduling, such that the expected total test time is minimized and the peak power constraint is satisfied. Based on a hybrid BIST architecture where a combination of deterministic and pseudorandom test sequences is used, the power-constrained test scheduling problem can be formulated as an extension of the two-dimensional rectangular packing problem and a heuristic has been proposed to calculate the near optimal order of different test sequences. The method is also generalized for both test-per-clock and test-per-scan approaches. Experimental results have shown that the proposed heuristic is efficient to find a near optimal test schedule with a low computation overhead.

1. Introduction

The complexity of electronic systems is continuously increasing, raising the cost and prolonging the production period of such systems. A significant proportion in the total cost is related to testing and therefore the reduction of the test cost contributes a lot to the overall cost reduction.

During recent years we have seen an advent of complex electronic systems where multiple pre-designed and pre-verified blocks, usually referred to as cores, are integrated into one single die. Such complex system-on-chip (SoC) designs pose great challenges to the test engineers since inefficient test methodologies may significantly increase the testing time of the system [1].

To test individual cores in a SoC, a set of test resources, such as test pattern source and sink together with an appropriate test access mechanism (TAM), have to be available [2]. Depending on the TAM architecture, the tests can be scheduled in parallel or sequentially. In this context, efficient scheduling can have significant impact on the reduction of total test time.

In a production test environment, an abort-on-first-fail (AOFF) approach is usually utilized. It means that the test process is stopped as soon as a fault is detected. This

approach leads to the reduction of test time and also production costs, as faulty dies can be eliminated before completing the entire test process. It should be noted here, that this approach has especially high significance during the early phases of the production, when the yield is low and defects are more likely to appear.

In this paper we propose a heuristic for test scheduling in an abort-on-first-fail test environment, considering defect probabilities of individual cores. These defect probabilities are usually derived from statistical analysis of the production process or generated based on inductive fault analysis. We assume a hybrid built-in self-test (BIST) architecture, where the test set is composed of internally generated pseudorandom test patterns and externally applied deterministic test patterns.

In our earlier work [3] we have proposed a method for concurrent test scheduling using a test-per-clock approach based on defect probabilities such that the Estimated Total Test Time (ETTT) is minimized. The approach, however, did not take into account the power consumption of tests. Scheduling too many tests concurrently might simply burn the circuit. Therefore in this paper we propose a heuristic for hybrid BIST test scheduling, such that the peak power constraint of the system is not exceeded and the ETTT is minimized.

The rest of this paper is organized as follows. In the next section the backgrounds and motivation of our work are given. In Section 3 the power constrained test scheduling problem is formulated based on the hybrid BIST architecture in the AOFF test environment and the calculation of estimated total test time is explained. Section 4 demonstrates the proposed heuristic. In section 5 the experimental results are presented and the paper is concluded in Section 6.

2. Backgrounds and Motivation

As the number of cores in a chip increases, the amount of total test data grows tremendously. This may pose serious problems because of the cost and technological limitations of the automated test equipment (ATE). One of the solutions to this problem is to use BIST and perform pseudorandom test pattern generation and test response compaction on the chip. However, due to several reasons, like very large numbers of test patterns and random pattern resistant faults, this approach may not be efficient. Therefore different approaches have been proposed to complement pseudorandom test patterns with deterministic test patterns which are applied from an ATE or on-chip memory. These approaches are generally

¹ This work has been partially supported by the Swedish Foundation for Strategic Research (SSF) under the Strategic Integrated Electronic Systems Research (STRINGENT) program.

referred to as hybrid BIST [4, 5] which can reduce memory requirements compared to pure deterministic test, and can provide higher fault coverage and shorter test application time compared to the stand-alone BIST solution.

In the current approach we have assumed that all cores have their own dedicated BIST logic that can carry out pseudo-random tests concurrently. The deterministic tests, on the other hand, are applied from external source to one core at a time. We have also assumed that an AMBA-like test bus [6] is used for test data transportation. An example of a multi-core system with such a test architecture is given in Figure 1.

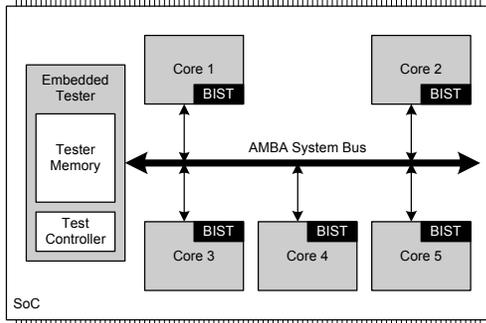


Figure 1. AMBA bus-based hybrid BIST architecture

The SoC test scheduling problem has been studied recently [7, 8]. In most of these approaches, an assumption has been made that all tests are applied to the completion. However, in a production environment, a test can be aborted as soon as a fault has been detected. Therefore the likelihood of a core to fail during the test should be considered to improve the efficiency of the test scheduling [9, 10].

In this paper an abort-on-first-fail test scheduling approach in a hybrid BIST environment is proposed, taking into account the power constraints. During the scheduling process we use the defect probability of individual cores to calculate the expected total test time. In our earlier work [3] we proposed a heuristic for AOFF test scheduling using test-per-clock approach without considering power constraints. In this paper, a generic method for ETTT calculation is proposed such that both test-per-clock and test-per-scan approaches can be employed. A new AOFF test scheduling heuristic is also proposed to minimize the ETTT without violating the power constraint.

3. Problem Formulation

Suppose that a system S , consisting of n cores C_1, C_2, \dots, C_n , has a test architecture as depicted in Figure 1. For every individual core C_i ($1 \leq i \leq n$), a deterministic test sequence DT_i and a pseudorandom test sequence PR_i is given, consisting of deterministic and pseudorandom test patterns, respectively. With DT_{ij} ($1 \leq j \leq d_i$) we denote the j -th deterministic test pattern, and with PR_{ij} ($1 \leq j \leq r_i$) the j -th pseudorandom test pattern, where d_i and r_i are respectively the total number of deterministic and pseudorandom test patterns for core C_i . For every individual core C_i a defect probability $DF(C_i)$ defined as the probability of a core to be detected faulty during production tests is given.

Based on the given hybrid BIST architecture, pseudorandom tests can be generated and applied to all cores

in parallel, thus reducing the total test time. However, testing consumes more power than normal operations, and may cause the chip to burn if too many tests are applied concurrently. Hence there exists a peak power constraint POW_c defining the amount of instant power a circuit can tolerate without getting damaged. In a SoC testing framework this means that at any time the total power of the concurrent test sequences should be below POW_c .

Efficient test scheduling which decides the order and starting time of test sequences can lead to the reduction of test time and consequently test cost. Figure 2 shows an example of a test schedule for five deterministic test sequences DT_i ($1 \leq i \leq 5$) and five pseudorandom test sequences PR_i ($1 \leq i \leq 5$). The peak power constraint is denoted with POW_c . The height and width of a rectangle in Figure 2 indicate the peak power and the length of a test sequence, respectively.

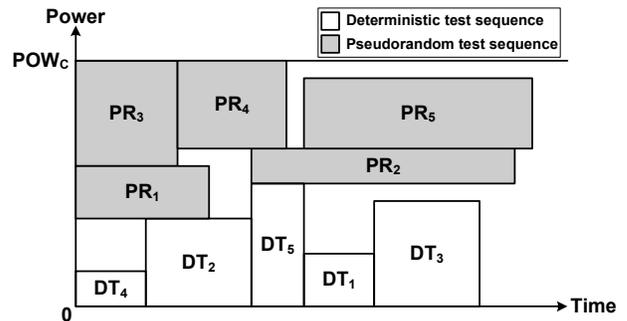


Figure 2. An example of power constrained test schedules

Note that test sequences belonging to the same core, like DT_5 and PR_5 , cannot be scheduled concurrently due to the test conflict.

In an AOFF environment we have two possible scenarios where tests can be aborted prematurely. During deterministic test application, the fault effect can be observed and analyzed at the end of every individual test pattern. Thus the test process can be aborted as soon as a fault is detected. On the other hand, if a fault is detected by a pseudorandom test sequence, the fault effect can only be analyzed at the end of the entire test sequence, when the signature is available. In our approach we can treat deterministic test patterns and pseudorandom test sequences in a similar way, since pseudorandom test sequence can be treated as a single test pattern with a length corresponding to the length of the particular pseudorandom test sequence. Therefore in the following discussion, a “test pattern” is used to denote both individual deterministic test patterns and pseudorandom test sequences, if not mentioned otherwise. It is also important to note that if a test-per-clock approach is assumed, the application time of a test pattern is one clock cycle, while with a test-per-scan approach, it equals to the period of an entire scan cycle.

The precedent discussion leads us to a set of possible test termination points: after every individual deterministic test pattern and at the end of every pseudorandom test sequence, as illustrated in Figure 3. The possible test termination points in deterministic test sequences are marked with gray lines and the test termination points at the end of pseudorandom test sequences are marked with black dotted lines in Figure 3.

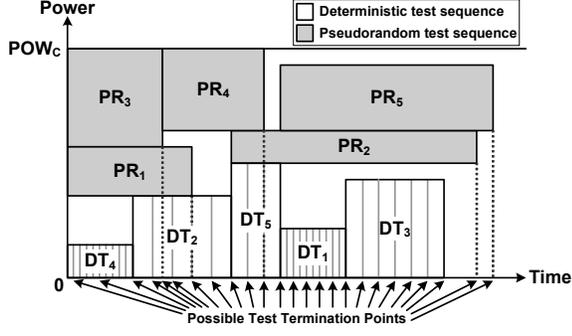


Figure 3. Possible test termination points

Note that some of these points overlap and therefore are treated as one identical possible test termination point.

We are interested in the expected total test time (ETTT) as the expectation of the total test application time in the AOFF environment. In Equation 1, we give a generic formula for ETTT calculation. In contrast to the formula presented in [3], this formula can be used for both test-per-clock and test-per-scan approach.

$$ETTT = \sum_{x \in X} (t_x \times p(A_x)) + L \times p(T) \quad (1)$$

Equation 1 is presented as the sum of two literals. The first corresponds to the situation when a test is terminated prematurely and the second one corresponds to the case where all tests are passed to the completion. At every possible test termination point $x \in X$, we can calculate a test abortion probability $p(A_x)$ together with a test length t_x at this test termination point x . With A_x we denote the event that the test has been aborted at test termination point x . Similarly we can also calculate the probability $p(T)$ that no faults are detected and all tests (T) are exercised till their completion. The length of the complete test set is denoted with L .

At every test termination point $x \in X$ we can distinguish two different sets of tests – the tests that have failed and the tests that have passed. The failed set Y_x consists of all test patterns y that have finished exactly at this point. They are supposed to detect at least one fault otherwise the test would not have been stopped at this point. The passed set Z_x consists of all test sequences that have successfully finished before this point x . They are all supposed to be passed otherwise the test would have been aborted before this point. This leads us to the following formula:

$$A_x = \left(\bigcup_{y \in Y_x} F_y \right) \cap \left(\bigcap_{z \in Z_x} P_z \right) \quad (2)$$

Here F_y is an event that the test pattern y detects at least one fault and P_z is an event that the test sequence z is passed. Thus the event A_x can be described as an event at test termination point x , such that any of the test patterns in the failed test set Y_x detect at least one fault, and all test sequences in the passed test set Z_x have passed. Please note that if a test pattern is included to the failed set then all other patterns testing the same core (including both the pseudorandom patterns and deterministic patterns) should be removed from the passed set, since the probability of these patterns passing the test has been already considered due to the use of incremental fault coverage (see Equation 6 below).

We assume that defect occurrences in different cores are independent of each other. Thus we can calculate the probability that the test is terminated at a possible termination point x as follows:

$$p(A_x) = p \left(\bigcup_{y \in Y_x} F_y \right) \prod_{z \in Z_x} p(P_z) = \left(1 - \prod_{y \in Y_x} (1 - p(F_y)) \right) \prod_{z \in Z_x} p(P_z) \quad (3)$$

Similarly we can also calculate the probability $p(T)$ that no faults are detected and all tests are exercised till their completion, as follows:

$$T = \bigcap_{z \in Z_x} P_z ; \quad p(T) = p \left(\bigcap_{z \in Z_x} P_z \right) = \prod_{i=1}^n (1 - DP(C_i)) \quad (4)$$

where $DP(C)$ denotes the defect probability of core C .

This leads us to the refined version of Equation 1:

$$ETTT = \sum_{x \in X} \left[t_x \times \left(1 - \prod_{y \in Y_x} (1 - p(F_y)) \right) \prod_{z \in Z_x} p(P_z) \right] + L \times \prod_{i=1}^n (1 - DP(C_i)) \quad (5)$$

The probability of the event that at least one fault is detected by a test pattern $y \in Y_x$ at the test termination point x can be calculated as

$$p(F_y) = IFC(y) \times DP(C) \quad (6)$$

where the incremental fault coverage $IFC(y)$ of a single test pattern y is defined as a percentage of the faults only detected by y and not detected by any previous test pattern.

Similarly the probability of the event that no faults are detected by a test sequence $z \in Z_x$ can be calculated as

$$p(P_z) = 1 - \sum_{j=1}^n IFC(v_j) \times DP(C) \quad (7)$$

where n is the total number of test patterns in the test sequence $z \in Z_x$, and v_j is the j -th test pattern.

In order to shorten the total test application time in the AOFF environment we have to reduce the ETTT. When a peak power constraint is taken into account, the test scheduling problem is similar to the classical two-dimensional rectangular packing problem [11, 12] which is NP complete. However, the need to take into account the given core defect probability and the test conflict constraints makes it different from the classical rectangular packing problem. Our objective is to use a heuristic to find a near optimal test schedule for both deterministic and pseudorandom test sequences so that the ETTT of the entire system is minimized while the power constraint is not exceeded.

4. Proposed Heuristic for Test Scheduling

Since usually deterministic test sequences are more efficient in detecting faults, it is natural to give them higher scheduling priority than pseudorandom test sequences. In this work, deterministic test sequences are scheduled sequentially as soon as possible. When all the deterministic test sequences have been scheduled, pseudorandom test sequences are scheduled into the rest of the space under the power constraint, as many in parallel as possible.

The scheduling of pseudorandom test sequences can employ a similar principle developed for the two-dimensional rectangular packing problem [11, 12]. Here we use the Bottom-Left-Decreasing (BLD) approach [12] to sort pseudorandom test sequences before scheduling them. First,

all the pseudorandom test sequences are sorted decreasingly by core defect probability. Then they are divided into several groups each of which has test sequences sorted decreasingly by the ratio of the peak power to the sequence length. Sorted pseudorandom test sequences are thereafter scheduled with the Bottom-Left (BL) strategy. Different from the classical two-dimensional rectangular packing problem, here test sequences do not have to be moved to the “left” since in the power dimension the position of a test sequence does not have any impact on the schedule. Thus pseudorandom test sequences are scheduled to the first available time moment.

Based on the principles described above, a heuristic is proposed to find a near optimal scheduling order for deterministic test sequences in an iterative way. Starting from the initial state that no test sequence is scheduled, one deterministic test sequence is scheduled per iteration step until all the deterministic test sequences are scheduled. At every iteration step, an unscheduled deterministic test sequence is selected to be inserted into the scheduled list. Then all pseudorandom test sequences are scheduled into the rest of the space and the partial ETTT is calculated within the range of scheduled deterministic test sequences. By exploring different solutions, a newly scheduled list with minimum partial ETTT is generated in this iteration step. The heuristic terminates when all the deterministic test sequences have been scheduled with an near optimal order and all the pseudorandom test sequences have been scheduled into the rest of the space under the power constraint.

5. Experimental Results

We performed two sets of experiments. In the first set, all the designs of cores were obtained from ISCAS’85. In the second set, ISCAS’89 benchmarks were used and all cores were redesigned in order to include a scan chain. For simplicity we assumed that all flip-flops are connected into one single scan chain and the STUMPS architecture was used for BIST.

For both cases we made experiments with 5 different design sizes, from 5 to 50 cores. For each design size we used 5 different SoC designs (different cores with different defect probability), and for every design we used 5 different levels of peak power constraint. The experimental results listed in Table 1 and Table 2 are the average of 25 experiments. The defect probabilities for individual cores are generated randomly, while keeping the total system defect probability at the value 0.6 (40% system yield).

Table 1. Experimental results of systems containing combinatorial cores

Number of Cores	5		10		20		30		50	
	ETTT	CPU Time (s)								
BLD Scheduling	440	<0.1	975	<0.1	1465	0.2	1695	<0.1	2820	0.2
Our Heuristic	377	0.7	884	16.8	1277	211.3	1534	947.4	2446	7306.7
SA	354	204.0	770	571.2	1111	2899.3	1289	9447.9	2118	16511.6

Table 2. Exp. results of systems containing sequential cores with full scan

Number of Cores	5		10		20		30		50	
	ETTT	CPU Time (s)	ETTT	CPU Time (s)	ETTT	CPU Time (s)	ETTT	CPU Time (s)	ETTT	CPU Time (s)
BLD Scheduling	45025	0.1	101330	0.1	151570	<0.1	181193	0.1	261948	0.3
Our Heuristic	37978	1.0	84258	12.7	128903	169.7	155202	740.3	226147	5561.6
SA	37045	4776.6	83525	4776.6	123451	7209.8	138668	9525.3	200056	12700.7

In order to show the efficiency of our heuristic, a classical bottom-left-decreasing scheduling algorithm is taken for comparison. It sorts both deterministic and pseudorandom test sequences decreasingly by the peak power times the test sequence length and schedules them using the bottom-left approach [12]. As shown in Tables 1 and Table 2, by employing our heuristic the ETTT can be reduced around 10%-17% compared to the bottom-left-decreasing scheduling, with an acceptable increase of computation time. On the other hand, when compared with a simulated annealing algorithm, which generates near-optimal solutions, our heuristic has a significantly lower execution time.

6. Conclusion

In this paper a power constrained SoC test scheduling method based on hybrid BIST architecture is presented. Different from other approaches, the defect probability of a individual core is introduced and a peak power constraint is taken into account. Based on the calculation of ETTT in an AOFF environment, a scheduling heuristic for test time minimization is proposed to produce good solutions with low computational overhead. Experimental results have shown the proposed method is effective to shorten the total test time.

References

- [1] B. T. Murray, and J. P. Hayes. Testing ICs: Getting to the core of the problem. *IEEE Transactions on Computer*, Vol. 29, No. 11, 1996, pp. 32-38.
- [2] Y. Zorian, E. J. Marinissen, and S. Dey. Testing Embedded Core-Based System Chips. *IEEE International Test Conference*, 1998, pp. 130-143.
- [3] Z. He, G. Jervan, Z. Peng, and P. Eles. Hybrid BIST Test Scheduling Based on Defect Probabilities. *IEEE Asian Test Symposium*, 2004, pp. 230-235.
- [4] G. Jervan, P. Eles, Z. Peng, R. Ubar, and M. Jenihhin. Test Time Minimization for Hybrid BIST of Core-Based Systems. *IEEE Asian Test Symposium*, 2003, pp. 318-323.
- [5] M. Sugihara, H. Date, and H. Yasuura. Analysis and Minimization of Test Time in a Combined BIST and External Test Approach. *Design, Automation and Test in Europe*, 2000, pp. 134-140.
- [6] D. Flynn. AMBA: Enabling Reusable On-Chip Designs. *IEEE Micro*, Vol. 17, No. 4, 1997, pp. 20-27.
- [7] Y. Huang, W.-T. Cheng, C.-C. Tsai, N. Mukherjee, O. Samman, Y. Zaidan, and S. M. Reddy. Resource Allocation and Test Scheduling for Concurrent Test of Core-based SOC Design. *IEEE Asian Test Symposium*, 2001, pp. 265-270.
- [8] E. Larsson, and Z. Peng. An Integrated Framework for the Design and Optimization of SOC Test Solutions. *Journal of Electronic Testing: Theory and Applications*, Vol. 18, No. 4/5, 2002, pp. 385-400.
- [9] W. J. Jiang, and B. Vinnakota. Defect-Oriented Test Scheduling. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 9, No. 3, 2001, pp. 427-438.
- [10] E. Larsson, J. Pouget, and Z. Peng. Defect-Aware SOC Test Scheduling. *IEEE VLSI Test Symposium*, 2004, pp. 359-364.
- [11] B. S. Baker, E.G. Coffman, Jr., and R. L. Rivest. Orthogonal Packings in Two Dimensions. *SIAM Journal of Computing*, Vol. 9, Issue 4, 1980, pp. 846-855.
- [12] N. Lesh, J. Marks, A. McMahon, and M. Mitzenmacher. Exhaustive Approaches to 2D Rectangular Perfect Packings, *Elsevier Science Direct, Information Processing Letters*, Vol. 90, Issue 1, 2004, pp. 7-14.