Recent Advances in Test Planning for Modular Testing of Core-Based SOCs[1]

Vikram Iyengar*, Krishnendu Chakrabarty† and Erik Jan Marinissen‡

*IBM Microelectronics, Essex Jct, VT 05452, USA, vikrami@us.ibm.com

†Duke University, ECE Department, Durham, NC 27708, USA, krish@ee.duke.edu

‡Philips Research Laboratories, 5656 AA Eindhoven, The Netherlands, erik.jan.marinissen@philips.com

Abstract

Test planning for core-based system-on-a-chip (SOC) designs is necessary to reduce testing time and test cost. In this paper, we survey recent advances in test planning that address the problems of test access and constrained test scheduling for core-based SOCs. We describe several test access architectures proposed by research groups in industry and academia, as well as a wide range of methodologies for the optimization of such architectures. An extensive list of references to prior and current work in the SOC test planning domain is included.

1 Introduction

Modular testing of embedded cores in a system-on-a-chip (SOC) is being increasingly advocated to simplify test access and test application [49]. To facilitate modular test, an embedded core must be isolated from surrounding logic, and test access must be provided from the I/O pins of the SOC. Test wrappers are used to isolate the core, while test access mechanisms (TAMs) transport test patterns and test responses between SOCs pins and core I/Os [49].

Effective modular test requires efficient management of the test resources for core-based SOCs. This involves the design of core test wrappers and TAMs, the assignment of test pattern bits to ATE channels, the scheduling of core tests, and the assignment of ATE channels to SOCs. The challenges involved in the optimization of SOC test resources for modular test can be divided into three broad categories.

1. Wrapper/TAM co-optimization: Test wrapper design and TAM optimization are of critical importance during system integration since they directly impact hardware overhead, testing time and tester data volume. The issues involved in wrapper/TAM design include wrapper optimization, core assignment to TAM wires, sizing of the TAMs, and routing of TAM wires. As shown in [8, 21, 35], most of these problems are $\mathcal{NP}$-hard. Figures 1(a) and (b) illustrate the position of TAM design and test scheduling in the SOC design for test (DFT) and test generation flows.

2. Constraint-driven test scheduling: The primary objective of test scheduling is to minimize testing time, while addressing one or more of the following issues: (a) resource conflicts between cores arising from the use of shared TAMs and BIST resources, (b) precedence constraints among tests, and (c) power dissipation constraints. Furthermore, testing time can often be decreased further through the selective use of test preemption [27]. As discussed in [6, 27], most problems related to test scheduling for SOCs are also $\mathcal{NP}$-hard.

3. Minimizing ATE re-load under memory depth constraints: Given test data for the individual cores, the entire test suite for the SOC must be made to fit in a minimum number of ATE memory loads (preferably one memory load). This is important because, while the time required to apply digital vectors is relatively small, the time required to load several gigabytes of data to the ATE memory from workstations is significant [3, 36]. Therefore, to avoid splitting the test into multiple ATE load–apply sessions, the number of bits required to be stored on any ATE channel must not exceed the limit on the channel's memory depth. While the problems of minimizing ATE re-load have not been studied in depth, their relation to the $\mathcal{NP}$-hard Bin Packing problem is intuitively obvious and we conjecture that these problems are also $\mathcal{NP}$-hard.

In addition, the rising cost of automatic test equipment (ATE) for system-on-chip (SOC) devices is a major concern [20]. Due to the growing demand for pin counts, speed, accuracy and vector memory, the cost of high-end ATE for full-pin, at-speed functional test is predicted to rise to over $20M by 2010 [20]. As a result, the use of low-cost ATE that perform structural rather than at-speed functional test is increasingly being advocated for reducing test costs. Multi-site testing, in which multiple SOCs are tested in parallel on the same ATE, can significantly increase the efficiency of ATE usage, as well as reduce testing time for an entire production batch of SOCs. The use of low-cost ATE and multi-site test involve test data volume reduction and test pin count (TAM width) reduction, such that multiple SOC test suites can fit in a single ATE [36, 46].

As a result of the intractability of the problems involved in test planning, test engineers have adopted a series of simple ad hoc solutions in the past [36]. For example, the problem of TAM width optimization is often simplified by stipulating that each core on the SOC have the same number of internal scan chains, say W; thus, a TAM of width W bits is laid out and cores are simply daisy-chained to the TAM. However, with the growing size of SOC test suites and rising cost of ATE, more aggressive test resource optimization techniques that enable effective modular test of highly-complex next-generation SOCs using current-generation ATE is critical.

The pressing need for new techniques in test resource optimization has led to the recognition of SOC test planning as an important area of research in both industry and universities. In addition, several research collaborations have emerged between SOC test research groups in universities and industry. In this paper, we survey a wide range of methods for test wrapper and TAM design, constraint-driven test scheduling, and tester memory reduction developed in recent research.

The remainder of this paper is organized as follows. In Section 2, we introduce the design of core test wrappers and describe methods for wrapper optimization. In Section 3, we describe a range of TAM architectures. In Section 4, we review several techniques proposed for TAM optimization. In Section 5, we discuss methods for test scheduling. In Section 6, we survey a range of methods proposed to integrate TAM design with test scheduling. Section 7 concludes the paper.

[1] This research was supported in part by the National Science Foundation under grants CCR-9875324 and CCR-0204077, and by an IBM Graduate Fellowship. This research was performed while Vikram Iyengar was at Duke.

Proceedings of the 11 th Asian Test Symposium(ATS'02)
1081-7735/02 $17.00 © 2002 IEEE

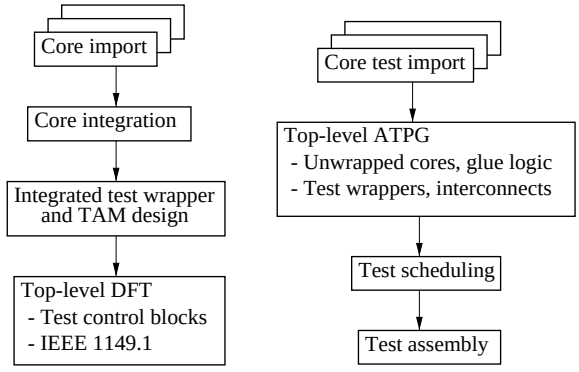

Figure 1. The (a) DFT generation flow and (b) test generation flow for SOCs.

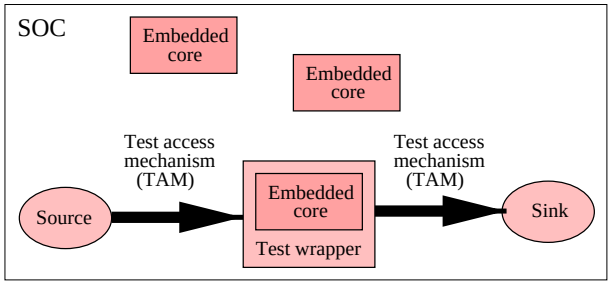

Figure 2. Embedded core test architecture.

2 Core test wrappers

A core test wrapper is a layer of logic that surrounds the core and forms the interface between the core and it's SOC environment. In this section, we first describe wrapper design and then wrapper optimization.

2.1 Wrapper design

Wrapper design is related to the well-known problems of circuit partitioning and module isolation, and is therefore a more general test problem than its current instance (related to SOC test using TAMs). For example, earlier proposed forms of circuit isolation (precursors of test wrappers) include boundary scan and BILBO [1].

The test wrapper and TAM model of SOC test architecture was presented in [49]; see Figure 2. In this paper, three mandatory wrapper operation modes listed were a) normal operation, b) core-internal test, and c) core-external test. Apart from the three mandatory modes, two optional modes are "core bypass" and "detach".

Two proposals for test wrappers have been the "test collar" [45] and TestShell [34]. The test collar was designed to complement the Test Bus architecture [45] and the TestShell was proposed as the wrapper to be used with the TestRail architecture [34]. In [45], three different test collar types were described: combinational, latched, and registered. For example, a simple combinational test collar cell consisting of a 2-to-1 multiplexer can be used for high-speed signals at input ports during parallel, at-speed test. The TestShell described in [34] is used to isolate the core and perform TAM width adaptation. It has four primary modes of operation: function mode, IP test mode, interconnect test mode and bypass mode. These modes are controlled using a test control mechanism that receives two types of control signals: pseudo-static signals (that retain their values for the duration of a test) and dynamic control signals (that can change values during a test pattern).

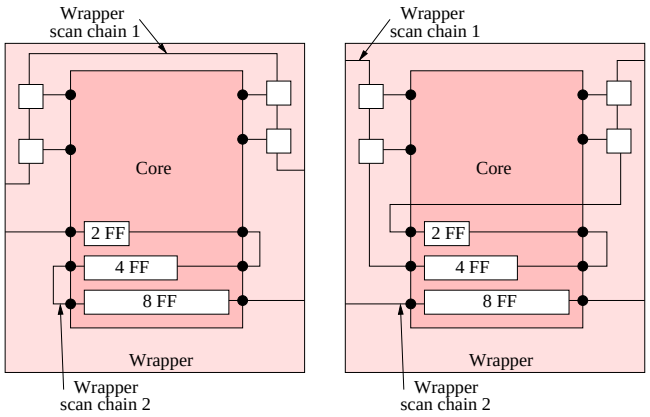

Figure 3. Wrapper chains: (a) unbalanced, (b) balanced.

The IEEE P1500 working group was established to draft a test standard for digital logic and memory cores [18, 40]. The activities of the P1500 working group include developing a standardized core test language and test wrapper interface from cores to on-chip TAMs. The P1500 wrapper contains a wrapper boundary register with wrapper cells at core I/O terminals, a wrapper instruction register, a wrapper interface port for serial access, and a wrapper bypass register. A parallel access port is currently in the process of standardization.

2.2 Wrapper optimization

An important function of the wrapper is to adapt the TAM width to the core's I/O terminals and internal scan chains. This is done by partitioning the set of core-internal scan chains and concatenating them into longer wrapper scan chains, equal in number to the TAM wires. Each TAM wire can now directly scan test patterns into a single wrapper scan chain. TAM width adaptation directly affects core testing time and has been the main focus of research in wrapper optimization. Note that to avoid problems related to clock skew, internal scan chains in different clock domains must either not be placed on the same wrapper scan chain, or anti-skew (lock-up) latches must be placed between scan flip-flops belonging to different clock domains.

The issue of designing balanced scan chains within the wrapper was addressed in [7]; see Figure 3. The first techniques to optimize wrappers for test time reduction were presented in [35]. To solve the problem, the authors proposed two polynomial-time algorithms that yield near-optimal results. The LPT (Largest Processing Time) algorithm is taken from the Multi-Processor Scheduling literature and solves the wrapper design problem in very short computation times. At the expense of a slight increase in computation time, the COMBINE algorithm yields even better results. It uses LPT as a start solution, followed by a linear search over the wrapper scan chain length with the First Fit Decreasing heuristic.

To perform wrapper optimization, the authors in [21] proposed *Design_wrapper*, an algorithm based on the Best Fit Decreasing heuristic for the Bin Packing problem. The algorithm has two priorities: (i) minimizing core testing time , and (ii) minimizing the TAM width required for the test wrapper. These priorities are achieved by balancing the lengths of the wrapper scan chains designed, and identifying

Proceedings of the 11 th Asian Test Symposium(ATS'02)
1081-7735/02 $17.00 © 2002 IEEE

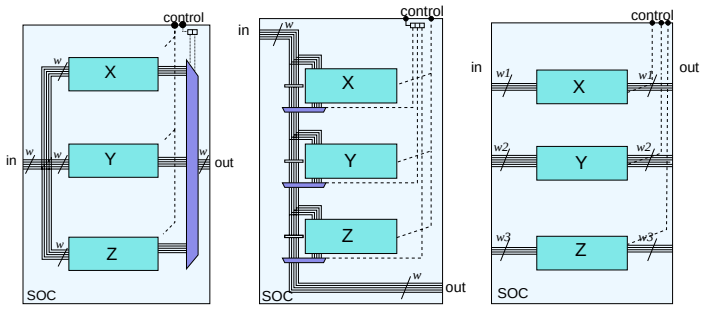

Figure 4. The (a) Multiplexing, (b) Daisychain, and (c) Distribution (c) architectures [2].

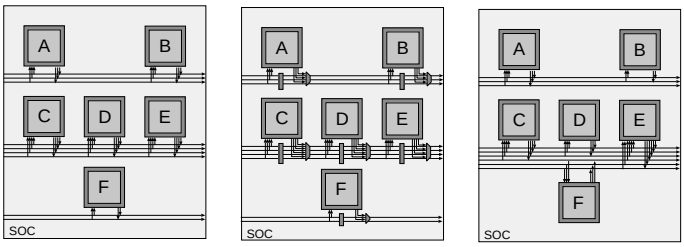

Figure 5. The (a) fixed-width Test Bus architecture, (b) fixed-width TestRail architecture, and (c) flexible-width Test Bus architecture.

the number of wrapper scan chains that actually need to be created to minimize testing time. Priority (ii) is addressed by the algorithm since it has a built-in reluctance to create a new wrapper scan chain, while assigning core-internal scan chains to the existing wrapper scan chains [21].

3 TAM Design

Many different TAM designs have been proposed. TAMs have been designed based on direct access to cores multiplexed onto the existing SOC pins [19], reusing the on-chip system bus [15], searching transparent paths through and/or around neighboring modules [14, 37, 42], and one-bit Boundary Scan rings around cores [47, 44].

Recently, the most popular appear to be the dedicated, scalable TAMs such as Test Bus [45] and TestRail [34]. Despite the fact that their dedicated wiring adds to the area costs of the SOC, their flexible nature and guaranteed test access have proven successful. Three basic types of such scalable TAMs have been described in [2] (see Figure 4): (a) the *Multiplexing* architecture, (b) the *Daisychain* architecture, and (c) the *Distribution* architecture. In the Multiplexing and Daisychain architectures, all cores get access to the total available TAM width, while in the Distribution architecture, the total available TAM width is distributed over the cores. Note that the multiplexer in the Multiplexing Architecture is conceptual, and hence could also be implemented by means of tri-state buffers with appropriate control signals.

In the Multiplexing architecture, only one core wrapper can be accessed at a time. Consequently, this architecture only supports serial schedules, in which the cores are tested one after the other. An even more serious drawback of this architecture is that testing the circuitry and wiring in between cores is difficult with this architecture; interconnect test requires simultaneous access to multiple wrappers. The other two basic architectures do not have these restrictions; they allow for both serial as well as parallel test schedules, and also support interconnect testing.

The *Test Bus* architecture [45] (see Figure 5(a)) is a combination of the Multiplexing and Distribution architectures. A single Test Bus is in essence the same as what is described by the Multiplexing architecture; cores connected to the same Test Bus can only be tested sequentially. The Test Bus architecture allows for multiple Test Buses on one SOC that operate independently, as in the Distribution architecture. Cores connected to the same Test Bus suffer from the same drawback as in the Multiplexing architecture, i.e., their wrappers cannot be accessed simultaneously, hence making core-external testing difficult or impossible.

The *TestRail* architecture [34] (see Figure 5(b)) is a combination of the Daisychain and Distribution architectures. A single TestRail is in essence the same as what is described by the Daisychain architecture: scan-testable cores connected to the same TestRail can be tested simultaneously, as well as sequentially. A TestRail architecture allows for multiple TestRails on one SOC, which operate independently, as in the Distribution architecture. The TestRail architecture supports serial and parallel test schedules, as well as hybrid combinations of those.

In most TAM architectures, the cores assigned to a TAM are connected to *all* wires of that TAM. We refer to this as *fixed-width* TAMs. A generalization of this design, is one in which the cores assigned to a TAM each connect to a (possibly different) subset of the TAM wires [24]. The core–TAM assignments are made at the granularity of TAM wires, instead of considering the entire TAM bundle as one inseparable entity. We call these *flexible-width* TAMs. We can apply this concept both for Test Bus as well as for TestRail architectures. Figure 5(c) shows an example of a flexible-width Test Bus architecture.

4 TAM Optimization

In this section, we describe recent research in TAM optimization from a "TAM architecture" perspective. The algorithms presented in several of the papers reviewed here are discussed further in Section 6 from a "scheduling" viewpoint.

Most SOC test architecture optimization algorithms proposed have concentrated on fixed-width Test Bus architectures and assume cores with fixed-length scan chains. In [8], the author describes a Test Bus architecture optimization approach that minimizes testing time using integer linear programming (ILP). ILP is replaced by a genetic algorithm in [11]. In [23], the authors extend the optimization criteria of [8] with place-and-route and power constraints, again using ILP. In [16], Test Bus architecture optimization is mapped to the well-known problem of two-dimensional bin packing and a Best Fit algorithm is used to solve it. Wrapper design and TAM design both influence the SOC testing time, and hence their optimization needs to be carried out in conjunction in order to achieve the best results. The authors in [21] were the first to formulate the problem of integrated wrapper/TAM design; despite its $\mathcal{NP}$-hard character, it is addressed using ILP and exhaustive enumeration. In [22], the authors presented efficient heuristics for the same problem.

Idle bits exist in test schedules when parts of the test wrapper and TAM are under-utilized leading to idle time in the test delivery architecture. In [39], the authors first formulated the testing time mini-

Proceedings of the 11 th Asian Test Symposium(ATS'02)
1081-7735/02 $17.00 © 2002 IEEE

mization problem both for cores having fixed-length as well as cores having flexible-length scan chains. Next, they presented lower bounds on the testing time for the Test Bus and TestRail architectures and then examined three main reasons for under-utilization of TAM bandwidth, leading to idle bits in the test schedule and testing times higher than the lower bound [39].

Flexible-width Multiplexing architecture optimization (i.e., for one TAM only) was proposed in [24]. This work again assumes cores with fixed-length scan chains. The paper describes a heuristic algorithm for co-optimization of wrappers and Test Buses based on rectangle packing. In [25], the same authors extended this work by including precedence, concurrency, and power constraints, while allowing a user-defined subset of the core tests to be preempted.

Fixed-width TestRail architecture optimization was investigated in [12, 13]. These papers describe heuristic algorithms for co-optimization of wrappers and TestRails. The algorithms work both for cores with fixed-length and flexible-length scan chains. TR-ARCHITECT, the tool presented in [13] is currently in actual industrial use.

5 Test scheduling

Test scheduling for SOCs involving multiple test resources and cores with multiple tests is especially challenging, and even simple test scheduling problems for SOCs have been shown to be $\mathcal{NP}$-hard [6]. In this section, we review several recently-proposed test scheduling methods. These methods do not consider TAM design issues during scheduling.

In [43], a method for selecting tests from a set of external and BIST tests (that run at different clock speeds) was presented. Test scheduling was formulated as a combinatorial optimization problem. Re-ordering tests to maximize defect detection early in the schedule was explored in [28]. The entire test suite was first applied to a small sample population of ICs. The fault coverage obtained per test was then used to arrange tests that contribute to high fault coverage earlier in the schedule. The authors used a polynomial-time algorithm to re-order tests based on the defect data as well as execution time of the tests [28].

Macro Test is a modular testing approach for SOC cores in which a test is broken down into a *test protocol* and list of test patterns [4]. A test protocol is defined at the terminals of a macro and describes the necessary and sufficient conditions to test the macro [37]. The test protocols are expanded from the macro-level to the SOC pins and can either be applied sequentially to the SOC, or scheduled to increase parallelism. In [37], a heuristic scheduling algorithm based on pairwise composition of test protocols was presented. The algorithm determines the start times for the expanded test protocols in the schedule, such that no resource conflicts occur and test time is minimized [37].

SOCs in test mode can dissipate up to twice the amount of power they do in normal mode, since cores that do not normally operate in parallel may be tested concurrently [48]. *Power-constrained* test scheduling is therefore essential in order to limit the amount of concurrency during test application to ensure that the maximum power budget of the SOC is not exceeded. In [9], a method based on approximate vertex cover of a resource-constrained test compatibility graph was presented. In [41], the use of list scheduling and tree-growing algorithms for power-constrained scheduling was discussed. The authors presented a greedy algorithm to overlay tests such that the power constraint is not violated. A constant additive model is employed for power estimation during scheduling [41]. The issue of re-organizing scan chains to trade-off testing time with power consumption was investigated in [32]. The authors presented an optimal algorithm to parallelize tests under power and resource constraints. The design of test wrappers to allow for multiple scan chain configurations within a core was also studied.

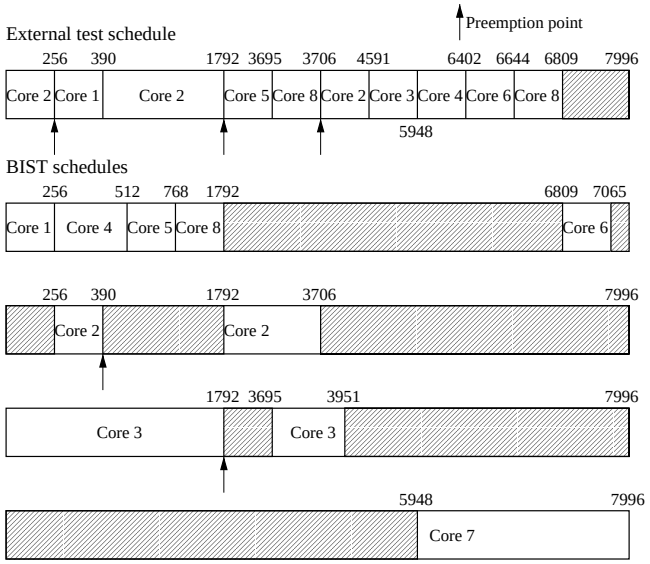

Figure 6. Preemptive test schedule for d281 incorporating precedence and power constraints.

In [27], an integrated approach to test scheduling was presented. Optimal test schedules with precedence constaints were obtained for reasonably-sized SOCs. For precedence-based scheduling of large SOCs, a heuristic algorithm was developed. The proposed approach also includes an algorithm to obtain preemptive test schedules in $O(n^3)$ time, where n is the number of tests [27]. Parameters that allow only a certain number of preemptions per test can be used to prevent excessive BIST and sequential circuit test preemptions. Finally, a new power-constrained scheduling technique was presented using which power-constraints can be easily embedded in the scheduling framework in combination with precedence constraints, thus delivering an integrated approach to the SOC test scheduling problem. An example test schedule incorporating precedence relations, selective preemption as well as power constraints for d281 (a benchmark SOC presented in [27]) is illustrated in Figure 6.

6 Integrated TAM optimization and test scheduling

Both TAM optimization and test scheduling significantly influence the testing time, test data volume and test cost for SOCs. Furthermore, TAMs and test schedules are closely related. For example, an effective schedule developed for a particular TAM architecture may be inefficient or even infeasible for a different TAM architecture. Integrated methods that perform TAM design and test scheduling *in conjunction* are therefore required to achieve low-cost, high-quality test.

In [31], an integrated approach to test scheduling, TAM design, test set selection and TAM routing was presented. The SOC test architecture was represented by a set of functions involving test generators, response evaluators, cores, test sets, power and resource constraints,

Proceedings of the 11 th Asian Test Symposium(ATS'02)
1081-7735/02 $17.00 © 2002 IEEE

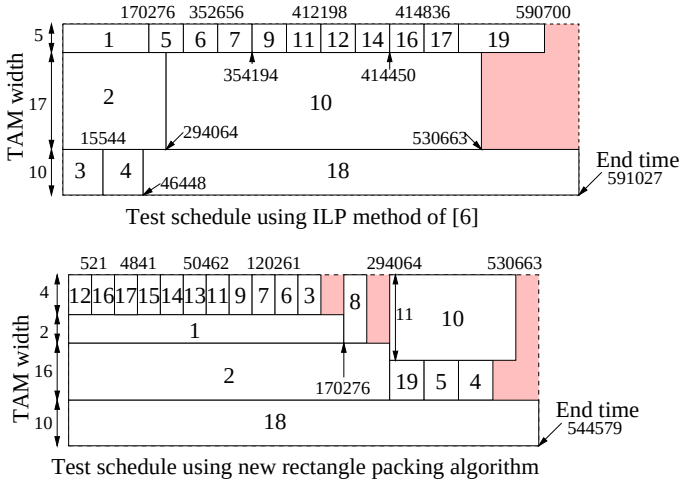

Figure 7. Test schedules for p34392 using [21] and the new method (figures not drawn to scale).

and start and end times in the test schedule modeled as boolean and integral values [31]. A polynomial-time algorithm was used to solve these equations and determine the test resource placement, TAM design and routing, and test schedule, such that the specified constraints are met.

The mapping between core I/Os and SOC pins during the test schedule was investigated in [16]. TAM design and test scheduling was modeled as two-deminsional bin-packing, in which each core test is represented by a rectangle. The height of each rectangle corresponds to the testing time, the width corresponds to the core I/Os, and the weight corresponds to the power consumption during test. The objective is to pack the rectangles into a bin of fixed width (SOC pins), such that the bin height (total testing time) is minimized, while power constraints are met. A heuristic method based on the Best Fit algorithm was presented to solve the problem [16]. The authors next formulated the constraint-driven pin mapping and test scheduling problem as a chromatic number problem in graph theory and as a dependency matrix partitioning problem [17]. Both problem formulations are $\mathcal{NP}$-hard. A heuristic algorithm based on clique partitioning was proposed to solve the problem.

The problem of TAM design and test scheduling with the objective of minimizing the *average* testing time was formulated in [29]. The problem was reduced to one of minimum-weight perfect bipartite graph matching, and a polynoimial-time optimal algorithm was presented. A test planning flow was also presented.

In [25], a new approach for wrapper/TAM co-optimization and constraint-driven test scheduling using rectangle packing was described. Flexible-width TAMs that are allowed to fork and merge were designed. Rectangle packing was used to develop test schedules that incorporate precedence and power constraints, while allowing the SOC integrator to designate a group of tests as preemptable. Finally, the relationship between TAM width and tester data volume was studied to identify an effective TAM width for the SOC. In Figure 7, we illustrate the difference between the test schedules obtained for the p34392 benchmark SOC for $W = 32$ using the method of fixed-width TAMs in [21], and using flexible-width TAMs in [25].

The research in [25] was extended in [26] to address the minimization of ATE buffer re-loads and include multi-site test. The ATE is assumed to contains a pool of memory distributed over several channels, such that the memory depth assigned to each channel does not exceed a maximum limit. Furthermore, the sum of the memory depth over all channels equals the total pool of ATE memory. Idle bits appear on ATE channels whenever there is idle time on a TAM wire. These bit positions are filled with don't-cares if they appear between useful test bits; however, if they appear only at the end of the useful bits, they are not required to be stored in the ATE.

The SOC test resource optimization problem for multi-site test was stated as follows. Given the test set parameters for each core, and a limit on the maximum memory depth per ATE channel, determine the wrapper/TAM architecture and test schedule for the SOC, such that (i) the memory depth required on any Channel is less than the maximum limit, (ii) the number of TAM wires is minimized, and (iii) the idle bits appear only at the end of each channel. A rectangle packing algorithm was developed to solve this problem [26].

Finally, a new method of representing SOC test schedules using k-tuples was discussed in [30]. The authors presented a p-admissible model for test schedules that is amenable to several solution methods such as local search, two-exchange, simulated annealing and genetic algorithms that cannot be used in a rectangle-representation environment.

ITC 2002 SOC test benchmarks. The ITC 2002 SOC Test Benchmark Initiative [38] is a joint initiative between Philips Research Laboratories in Eindhoven, The Netherlands and Duke University in Durham, NC to put together a common set of SOC benchmarks in consultation with other universities and companies. The benchmarks are aimed at stimulating research in new tools and methodologies in SOC test automation. These benchmarks will be formally presented at the *International Test Conference* in October 2002.

7 Conclusion

We have described a wide range of test planning methods that address the challenges of test access to embedded cores, test scheduling and ATE memory reduction. These advances in test research have led to a new paradigm in test planning for SOCs. System integrators are now moving away from simple ad hoc solutions that treat each test flow challenge as an independent problem that must be "fixed", towards a more comprehensive understanding of test resources, their relationship to test cost and how trade-offs between resources affect test quality. With the growing complexity of SOC designs, new and broader challenges that remain to be addressed include hierarchical TAM design, TAM design for mixed-signal cores and ATE reload minimization (scheduling tests among several ATE buffer loads). The wide range of methodologies described in this paper effectively investigate the potential for further research in this domain.

References

[1] M. Abramovici, M.A. Breuer and A.D. Friedman. *Digital Systems Testing and Testable Design*. Computer Science Press, New York, NY, 1990.

[2] J. Aerts and E.J. Marinissen. Scan chain design for test time reduction in core-based ICs. *Proc. Int. Test Conf.*, pp. 448–457, 1998.

[3] C. Barnhart et al. OPMISR: The foundation for compressed ATPG vectors. *Proc. Int. Test Conf.*, pp. 748–757, 2001.

[4] F. Beenker, B. Bennetts and L. Thijssen. *Testability Concepts for Digital ICs - The Macro Test Approach.* Frontier in Electronic Testing, vol. 3. Kluwer Academic Publishers, Boston, MA.

Proceedings of the 11 th Asian Test Symposium(ATS'02)
1081-7735/02 $17.00 © 2002 IEEE

[5] M. Benabdenbi, W. Maroufi and M. Marzouki. CAS-BUS: A scalable and reconfigurable test access mechanism for systems on a chip. *Proc. DATE Conf.*, pp. 141–145, 2000.

[6] K. Chakrabarty. Test scheduling for core-based systems using mixed-integer linear programming. *IEEE Trans. CAD*, vol. 19, pp. 1163–1174, October 2000.

[7] T.J. Chakraborty, S. Bhawmik and C.-H. Chiang. Test access methodology for system-on-chip testing. *Proc. Int. Workshop on Testing Embedded Core-Based System-Chips*, pp. 1.1-1–1.1-7, 2000.

[8] K. Chakrabarty. Optimal test access architectures for system-on-a-chip. *ACM Trans. Design Automation of Electronic Sys.*, vol. 6, pp. 26–49, January 2001.

[9] R.M. Chou, K.K. Saluja and V.D. Agrawal. Scheduling tests for VLSI systems under power constraints. *IEEE Trans. VLSI*, vol. 5, no. 2, pp. 175–184, June 1997.

[10] E. Cota et al. Test planning and design space exploration in a core-based environment. *Proc. DATE Conf.*, pp. 478–485, 2002.

[11] Z.S. Ebadi and A. Ivanov. Design of an optimal test access architecture using a genetic algorithm. *Proc. Asian Test Symp.*, pp. 205–210, 2001.

[12] S.K. Goel and E.J. Marinissen. Cluster-based test architecture design for system-on-chip. *Proc. VLSI Test Symp.*, pp. 259–264, 2002.

[13] S.K. Goel and E.J. Marinissen. A novel test time reduction algorithm for test architecture design for core-based system chips. *Proc. European Test Workshop*, 2002, in press.

[14] I. Ghosh, S. Dey and N.K. Jha. A fast and low cost testing technique for core-based system-on-chip. *Proc. Design Automation Conf.*, pp. 542–547, 1998.

[15] P. Harrod. Testing re-usable IP: A case study. *Proc. Int. Test Conf.*, pp. 493–498, 1999.

[16] Y. Huang et al. Resource allocation and test scheduling for concurrent test of core-based SOC design. *Proc. Asian Test Symp.*, pp. 265–270, 2001.

[17] Y. Huang et al. On concurrent test of core-based SOC design. *J. Electronic Testing: Theory and Applications*, vol. 18, Aug. 2002, to appear.

[18] IEEE P1500 Standard for Embedded Core Test. *http://grouper.ieee.org/groups/1500*

[19] V. Immaneni and S. Raman. Direct access test scheme - Design of block and core cells for embedded ASICs. *Proc. Int. Test Conf.*, pp. 488–492, 1990.

[20] Semiconductor Industry Association. *Int. Tech. Roadmap for Semiconductors, http://public.itrs.net*

[21] V. Iyengar, K. Chakrabarty and E.J. Marinissen. Test wrapper and test access mechanism co-optimization for system-on-chip. *J. Electronic Testing: Theory and Applications*, vol. 18, pp. 213–230, April 2002.

[22] V. Iyengar, K. Chakrabarty, and E.J. Marinissen. Efficient wrapper/TAM co-optimization for large SOCs. *Proc. DATE Conf.*, pp. 491–498, 2002.

[23] V. Iyengar and K. Chakrabarty. Test bus sizing for system-on-a-chip. *IEEE Trans. Computers*, vol. 51, pp. 449–459, May 2002.

[24] V. Iyengar, K. Chakrabarty, and E.J. Marinissen. On using rectangle packing for SOC wrapper/TAM co-optimization. *Proc. VLSI Test Symp.*, pp. 253–258, 2002.

[25] V. Iyengar, K. Chakrabarty, and E.J. Marinissen. Integrated wrapper/TAM co-optimization, constraint-driven test scheduling, and tester data volume reduction for SOCs. *Proc. Design Automation Conf.*, pp. 685–690, 2002.

[26] V. Iyengar, S.K. Goel, E.J. Marinissen and K. Chakrabarty. Test resource optimization for multi-site testing of SOCs under ATE memory depth constraints. *Proc. Int. Test Conf.*, 2002, to appear.

[27] V. Iyengar and K. Chakrabarty. System-on-a-chip test scheduling with precedence relationships, preemption, and power constraints. *IEEE Trans. CAD*, 2002, in press.

[28] W. Jiang and B. Vinnakota. Defect-oriented test scheduling. *Proc. VLSI Test Symp.*, pp. 433–438, 1999.

[29] S. Koranne. On test scheduling for core-based SOCs. *Proc. Int. Conf. VLSI Design*, pp. 505–510, 2002.

[30] S. Koranne and V. Iyengar. A novel representation of embedded core test schedules. *Proc. Int. Test Conf.*, 2002, to appear.

[31] E. Larsson and Z. Peng. An integrated system-on-chip test framework. *Proc. DATE Conf.*, pp. 138–144, 2001.

[32] E. Larsson and Z. Peng. Test scheduling and scan-chain division under power constraint. *Proc. Asian Test Symp.*, pp. 259–264, 2001.

[33] J.-F. Li et al. A hierarchical test scheme for system-on-chip designs. *Proc. DATE Conf.*, pp. 486–490, 2002.

[34] E.J. Marinissen et al. A structured and scalable mechanism for test access to embedded reusable cores. *Proc. Int. Test Conf.*, pp. 284–293, 1998.

[35] E.J. Marinissen, S.K. Goel and M. Lousberg. Wrapper design for embedded core test. *Proc. Int. Test Conf.*, pp. 911–920, 2000.

[36] E.J. Marinissen and H. Vranken. On the role of DfT in IC - ATE matching. *Int. Workshop on TRP*, 2001.

[37] E.J. Marinissen. The role of test protocols in automated test generation for embedded-core-based system ICs. *J. Electronic Testing: Theory and Applications*, vol. 18, pp. 435–454, Aug. 2002, to appear.

[38] E.J. Marinissen, V. Iyengar and K. Chakrabarty. A Set of Benchmarks for Modular Testing of SOCs. *Proc. Int. Test Conf.*, 2002, to appear. (Benchmark SOC data available at *http://www.extra.research.philips.com/itc02socbenchm*)

[39] E.J. Marinissen and S.K. Goel. Analysis of test bandwidth utilization in test bus and TestRail architectures in SOCs. *Digest of papers of DDECS*, pp. 52–60, 2002.

[40] E.J. Marinissen et al. On IEEE P1500's standard for embedded core test. *J. Electronic Testing: Theory and Applications*, vol. 18, pp. 365–383, Aug. 2002, to appear.

[41] V. Muresan et al. A comparison of classical scheduling approaches in power-constrained block-test scheduling. *Proc. Int. Test Conf.*, pp. 882–891, 2000.

[42] M. Nourani and C. Papachristou. An ILP formulation to optimize test access mechanism in system-on-chip testing. *Proc. Int. Test Conf.*, pp. 902–910, 2000.

[43] M. Sugihara, H. Date and H. Yasuura. A novel test methodology for core-based system LSIs and a testing time minimization problem. *Proc. Int. Test Conf.*, pp. 465–472, 1998.

[44] N.A. Touba and B. Pouya. Using partial isolation rings to test core-based designs. *IEEE Design and Test of Computers*, vol. 14, pp. 52–59, October–December 1997.

[45] P. Varma and S. Bhatia. A structured test re-use methodology for core-based system chips. *Proc. Int. Test Conf.*, pp. 294–302, 1998.

[46] E. Volkerink et al. Test economics for multi-site test with modern cost reduction techniques. *Proc. VLSI Test Symp.*, pp. 411–416, 2002.

[47] L. Whetsel. An IEEE 1149.1 based test access architecture for ICs with embedded cores. *Proc. Int. Test Conf.*, pp. 69–78, 1997.

[48] Y. Zorian. A distributed BIST control scheme for complex VLSI devices. *Proc. VLSI Test Symposium*, pp. 6–11, 1993.

[49] Y. Zorian, E.J. Marinissen and S. Dey. Testing embedded-core-based system chips. *IEEE Computer*, vol. 32, pp. 52–60, June 1999.

Proceedings of the 11 th Asian Test Symposium(ATS'02)
1081-7735/02 $17.00 © 2002 IEEE