Delay Fault Testing of Core-Based Systems-on-a-Chip

Qiang Xu and Nicola Nicolici
Department of Electrical and Computer Engineering
McMaster University, Hamilton, ON L8S 4K1, Canada
Email: xuqiang@grads.ece.mcmaster.ca, nicola@ece.mcmaster.ca

Abstract

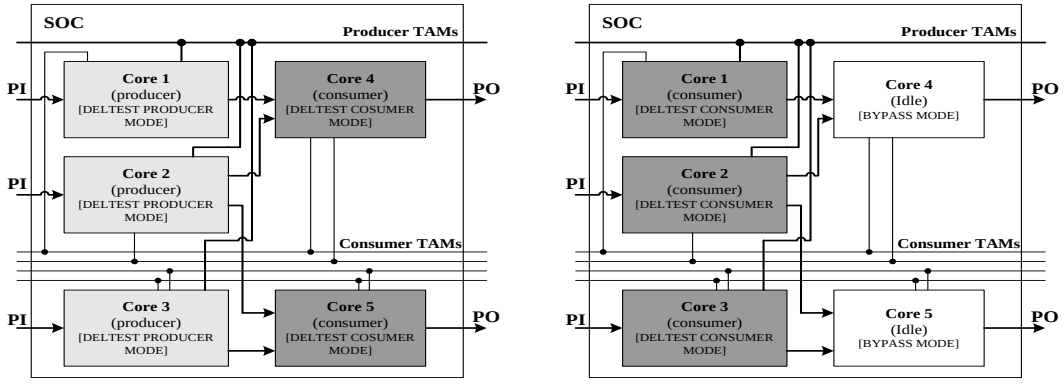
Existing approaches for modular manufacturing testing of core-based systems-on-a-chip (SOCs) do not provide any explicit mechanism for high quality two-pattern tests required for performance validation through delay fault testing. This paper proposes a new approach for broadside delay fault testing of core-based SOCs, by adapting the existing solutions for automatic test pattern generation and design for test support, test access mechanism division and test scheduling.

1 Introduction

One way to model digital integrated circuits' (ICs) physical defects is to abstract them as stuck-at fault models [2]. Regardless of stuck-at fault model's efficiency for several decades, alternative models, such as the delay fault model, need to account for deep submicron manufacturing process variations. Delay fault model is covering physical defects that affect *only* the timing of the circuit without changing its logical operation. To detect these timing-related defects, at least two controlled patterns in *consecutive* clock cycles are necessary: the first one initializes the circuit's nodes and the second one captures the transitions on the sensitizable paths. When applying these two consecutive patterns, the faulty propagation paths will not be detected unless the outputs are sampled at the highest operating frequency. Therefore, since high stuck-at tests consist of a single application pattern, they cannot guarantee the correct functionality at the operational clock frequency. Hence, to increase circuit reliability and manufacturing yield through speed sorting, semiconductor manufacturers are constrained to develop delay fault tests and the associated testing strategies.

While functional testing can address timing verification, its main drawback is low coverage for complex digital ICs, where transistor to pin ratio is continuing to increase. Structural testing using gate-level fault models and design-for-test (DFT) techniques, such as scan, is the remaining alternative which provides a well-structured way of validating high-speed digital ICs. However, the main disadvantage of the standard scan-based test is the need for a scan cycle, which loads/unloads test vectors/responses serially, thus preventing straightforward consecutive application of two test patterns. There are three main approaches to extend scan-based testing to handle delay tests [18]: (a) *enhanced scan* which requires two sequential elements in every scan cell; (b) *skewed-load* which uses the last-shifted pattern in the scan chain as the excitation vector; and (c) *broadside test* based on functional justification where the sequential (pseudo-input) part of the second pattern is generated through the combinational block. On the one hand, test application via enhanced scan and skewed-load delivers tests that may not be sensitizable in the normal operation, which can cause unnecessary yield loss. On the other hand, broadside testing limits the space of the possible consecutive patterns to only those that affect the timing in the normal mode, thus covering the ***worst-case operational behavior*** of the manufactured circuit [18].

Since there has been extensive research on DFT and automatic test pattern generation (ATPG) for delay faults over the last couple of decades [2], the question is how do the existing methods adapt to core-based SOCs? This adaption is an open issue, since, in addition to the standard test-quality problems, core-based SOCs present new challenges, in particular in terms of test development for providers and test access mechanism (TAM) for integrators. One way to deliver at-speed tests to embedded cores is to exploit SOC's architecture-specific information for providing sources and sinks during testing and to reuse on-chip functional interconnect for TAM. Various solutions, either at the core or system level, have been proposed [4, 6, 12, 17]. Regardless of their potential benefits in the long term, unless implemented automatically using a reliable test tool flow, these architecture-specific DFT methodologies do not provide reusability, flexibility and inter-operability. To address these problems in a well-structured modular way, the practical state-of-the-art SOC test approaches, such as the emerging IEEE P1500 standard [16], use separate test and functional communication architectures. Several representative solutions to core wrapper and/or TAM design have been reported in [3, 7, 9–11, 14].



(a) Session 1 for delay fault testing cores 4 and 5

(b) Session 2 for delay fault testing cores 1, 2 and 3

Figure 1. Proposed Approach for Delay Fault Testing of Core-Based SOCs.

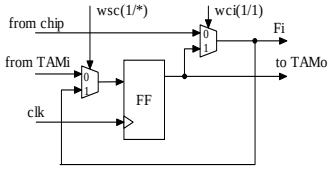
Although recent research advances [3, 7, 9–11, 14] and standardization efforts for modular SOC manufacturing testing (e.g., IEEE P1500 [16]) support structural scan tests, no explicit mechanism for delay fault testing is provided, which is the purpose of this research. To the best of authors' knowledge, all the previous work on TAM design and SOC test scheduling is based on one-pattern test, which, as shown in this paper, is not necessarily applicable to high quality delay fault testing. Therefore, the aim of this paper is to describe a new approach for delay fault testing of *core-based SOCs*, which adapts the existing solutions for ATPG, DFT support, TAM division and test scheduling, to broadside application of two-pattern test vector pairs.

2 Proposed Approach for Delay Fault Testing

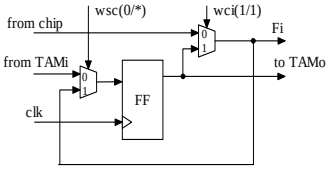
The proposed approach considers, as a starting point, that each core in the SOC is part of the IEEE P1500 architecture [16], which consists of test control lines, test access mechanism and core wrappers. The test control lines set the mode of each core and the test access mechanism transfers data to and from the core-under-test (CUT). The wrapper boundary register (WBR) cells provide a core isolation mechanism used by the test modes (e.g., INTEST or EXTEST). In the INTEST mode, used for testing the core's internal logic, the input WBR cells act as primary inputs (PIs) to the CUT, while the output WBR cells act as primary outputs (POs). In the EXTEST mode, when all the cores are wrapped, the goal is to test the interconnect wires between the cores. Thus the output WBR cells provide patterns and the input WBR cells capture responses from the interconnect, which are blocked in the input WBR cells and do not get propagated to the core's internal logic.

Since broadside delay test vector pairs require full control of the input WBR cells of the CUT in two consecutive clock cycles (for justification see Section 3), the proposed approach uses the input WBR cells of the CUT to control the PIs of the first test vector and it exploits the output WBR cells of the cores connected to the CUT to control the PIs of the second test vector. Therefore, each core tested in the current test session is called a *consumer* and the cores which provide the primary inputs for the second excitation vector are named *producers*. As shown in Figure 1, this test session-level producer-consumer dichotomy leads to a division of the SOC's TAM into producer TAMs and consumer TAMs (for justification see Section 4).

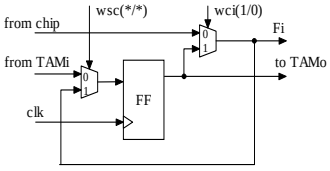
The proposed broadside delay fault testing process can be explained as follows. The P1500 instruction set is extended to support two custom DELTEST modes (one for producer cores and one for consumer cores). In the producer DELTEST mode, the output WBR cells are loaded from the producer TAM lines. In the consumer DELTEST mode, the loading of the initialization vector into the internal scan chains and the application of the first initialization vector are done via INTEST controls using consumer TAM lines. However, since DELTEST requires two capture cycles, an internal control mechanism for applying the second excitation vector must be provided. On the one hand, the pseudo-input part (PSI) of the excitation pattern (the internal scan chain part) is generated through functional justification, i.e., load the pseudo-output (PSO) of the second vector. On the other hand, the PI part is provided using the functional inputs by setting the provider cores in the producer DELTEST mode. This can be seen in Figures 1(a) and 1(b), where two test sessions are required to test a hypothetical SOC for delay faults. To guarantee efficient utilization of the tester's resources, the TAM lines use most of the channels, while the SOC's PIs are controlled via boundary scan. To ensure two consecutive controllable PI vectors, the consumer's WBR multiplexer control signals are switching between the input WBR cells of the consumer for initialization and the output WBR cells of the producer for excitation. It should be noted that by exploiting the existing producer output WBR cells for storing the PI part of the excitation vector, an *emulation of the enhanced-scan is provided only for the PIs of each core*, at no additional cost.



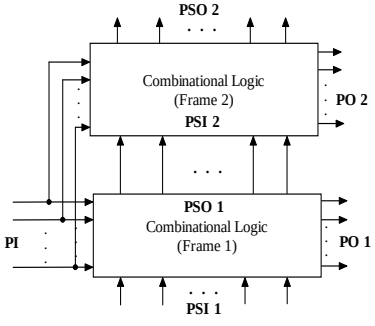
(a) $NC-PI$ Two-Pattern Cell Control



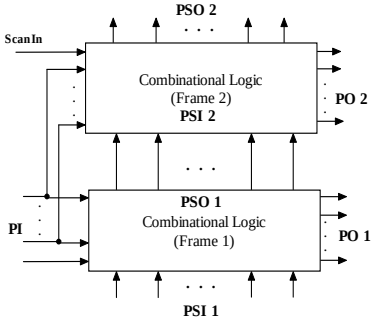
(b) $SC-PI$ Two-Pattern Cell Control



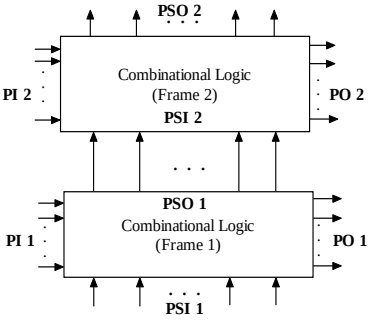
(c) $PC-PI$ Two-Pattern Cell Control



(d) $NC-PI$ Two-Pattern ATPG Model



(e) $SC-PI$ Two-Pattern ATPG Model



(f) $PC-PI$ Two-Pattern ATPG Model [8]

Figure 2. Wrapper Input Cells and Combinational ATPG Models for Broadside Scan Testing When Using Three Different Control Mechanisms for Embedded Core's Primary Inputs.

3 DFT Support and Broadside ATPG models

The additional DFT support is necessary to switch the wrapper cell's multiplexer control signals in the two consecutive capture cycles, when the consumer core is in the DELTEST mode. The associated broadside ATPG models provide an efficient way to compute test vector pairs using combinational ATPG, based on the control mechanism of core's PIs. In broadside delay testing the pseudo-input part of the excitation vector is generated through functional justification, however in order to emulate the functional core behavior, several options for *embedded core's* PI control are analyzed:

- *Non-Controlled Primary Inputs (NC-PI):* this test scenario assumes that PIs are scanned for fully controlling the initialization vector, however they keep the same value (frozen) for the excitation vector; the control of the input WBR cells and the associated broadside ATPG model are shown in Figures 2(a) and 2(d), where labels on the *wsc* and *wci* multiplexer controls show the values during the first/second capture cycle;

- *Serially-Controlled Primary Inputs (SC-PI):* after the PIs are scanned in for the initialization vector, in order to obtain the excitation vector, they are updated through an extra shift (using TAM data as input for the first WBR cell) during the first capture cycle; the control of the input WBR cells and the associated broadside ATPG model are shown in Figures 2(b) and 2(e);

- *Parallelly-Controlled Primary Inputs (PC-PI):* this control mechanism guarantees that *any* primary input value can be justified for both initialization and excitation vectors, either through shifting (first clock cycle) or parallel load from a *producer core* (second clock cycle); the control of the input WBR cells and the associated broadside ATPG model are shown in Figures 2(c) and 2(f); this PI control mechanism leads to a $producer-consumer$ model, which, in each test session, divides the active cores into *producers* (cores that provide PI values for excitation) and *consumers* (cores tested in the current session through broadside).

Since $NC-PI$ and $SC-PI$ control mechanisms can be implemented by the existing TAM design and test scheduling algorithms, an obvious question is why do we need $PC-PI$ with a $producer-consumer$ model, which leads to additional resource conflicts? The answer lies in the *quality of delay tests*. To prove this, we have implemented the three ATPG models for two-pattern broadside tests in ATALANTA [13], and we have calculated the coverage for transition faults. In addition to logic redundancies, and hence redundant faults, caused by the connection of PSO of the first time frame (PSO1) to the PSI of the second time frame (PSI2), which is equally applicable to all the three ATPG models, it was found that $NC-PI$ and $SC-PI$ introduce extra redundant transition faults that reduce the transition fault coverage. While the transition fault coverage loss caused

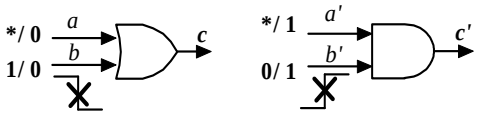


Figure 3. Advantage of $PC-PI$ Over $NC-PI$ and $SC-PI$ for Two Simple Transition Faults.

by PSO1 and PSI2 connection *must* be ignored (these redundant faults will never be activated during functional operation, which is the *key* advantage of broadside over skewload and enhanced-scan [18]), the coverage loss due to PI sharing between the two time frames (see both $NC-PI$ and $SC-PI$ in Figures 2(d) and 2(e) respectively) is intolerable. The limitation of the $NC-PI$ and $SC-PI$ control mechanisms can also be illustrated by testing the b and b' inputs of the OR and AND gates for falling and rising transitions respectively (see Figure 3). For example, if OR input b precedes input a in the wrapper boundary register, then the falling transition on input b is untestable by $NC-PI$ and $SC-PI$, since either freezing ($NC-PI$) or shifting ($SC-PI$) the PI value, $b=1$ required for initialization will conflict with $a=b=0$ necessary for excitation in the following second capture cycle. One might argue that this problem can be solved by structural modifications (i.e., wrapper cell reordering), however, this may be prohibited due to routing constraints. Furthermore, two-pattern test set compaction for $NC-PI$ and $SC-PI$ control mechanisms will introduce additional constraints (caused by freezing and shifting), which are difficult to satisfy when the care-bit density in each vector is increasing (i.e., when test set size decreases). Moreover, an additional advantage of $PC-PI$ is the ability to easily extend broadside scan testing to three-pattern tests [5], which may be required for timing verification of high-performance deep submicron circuits. An original extension of the two-pattern ATPG model proposed in [8] to a new three-pattern ATPG model is shown in Figure 4. When extending $NC-PI$ and $SC-PI$ to three-pattern tests, the number of unsatisfiable PI values will further increase, however, the proposed extension of $PC-PI$, supported by the producer-consumer model, will look for the best possible option when to load or shift the PIs in the second and third capture cycles (see BS/SL signal in Figure 4). It should also be noted that by injecting different necessary assignments in the first two time frames and by using various ATPG search engines, the proposed combinational ATPG model for three-pattern broadside testing can be adapted to generate high quality functionally justifiable delay tests for a large class of timing-related defects (e.g., highly robust tests for path delay faults [5]).

To assess the impact of the DELTEST instruction on the wrapper area, a processor core was synthesized to 0.18 micron TSMC technology [19]. When compared to a standard IEEE P1500 implementation (i.e., INTEST, EXTEST and BYPASS) the additional overhead due to the DFT support for DELTEST decoding is under 1%.

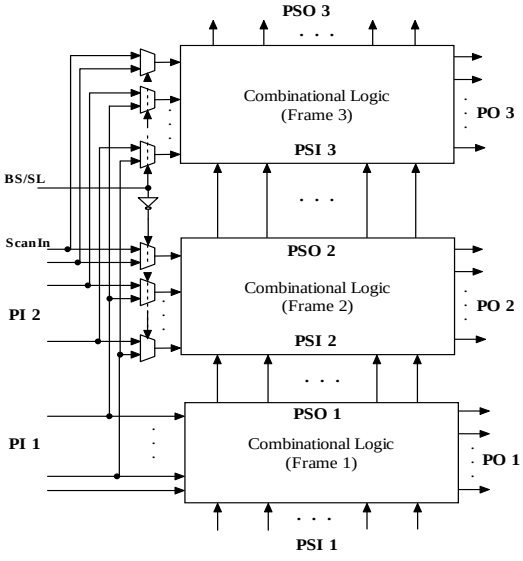


Figure 4. New Three-Pattern ATPG Model Supported by the Producer-Consumer Model.

4 TAM Division and Delay Test Scheduling

Having introduced the *producer – consumer* model supported by a new DELTEST instruction, the DFT support and the broadside delay fault ATPG model required on the core provider's side, this section summarizes some of the considerations necessary to adapt the existing TAM design and test scheduling algorithms for broadside delay fault testing, which are necessary on the system integrator's side:

- *TAM Division Into Producer/Consumer Groups*: producers and consumers should be fed from **two separate** TAM groups. It is known that in the INTEST mode, embedded cores can be tested concurrently as long as they use different TAM lines, however this is not the case for the DELTEST mode. For example, consider $Core_1$ and $Core_2$ connected to separate TAM lines, however $Core_1$ shares its TAM lines with $Core_3$, which is a producer to $Core_2$. In the DELTEST mode, to test $Core_2$, we need to load the output of $Core_3$. This implies that loading a pattern in $Core_1$ is prohibited at this time, although it uses separate TAM lines to $Core_2$. Hence, if we assume that producers and consumers are loaded from the same TAM lines, this *indirect* resource conflict may prohibit test concurrency. A neat solution to this problem, which is essential in particular for complex SOCs with a large number of cores, is to divide the TAM lines into two groups: G_{prod} for loading the excitation patterns in the producers' outputs and G_{cons} for loading the initialization patterns and unloading the test responses from the consumers. Note, to address this we need to use additional multiplexers (one per each TAM line for every core) to facilitate parallel output WBR loading from the producer TAM lines, when the producer is in the DELTEST mode.

- *Producer-Consumer Conflict*: producers and consumers should **not** be tested at the same time. This is justified by the difficulty to implement pipelined testing (e.g., test cores 1 and 4 concurrently in Figure 1(a)). This difficulty arises from the synchronization problems during the capture cycles, which need to occur at the same time for both cores, such that they should not corrupt the excitation data stored in the producer's output WBR cells. In addition, the input WBR cells shown in Figure 2(c), would need to be modified to buffer the output of the producer in the input wrapper cells of the consumer.

- *Shared-Producer Conflict*: two cores which directly connect to the same producer(s) for the excitation vector, **cannot** be tested at the same time, and hence they should not be loaded concurrently. The producers can be connected to consumers in two ways: through functional busses or directly. When sharing the functional busses the test application penalty is insignificant. This is because all the producer cores connected to a functional bus are loaded at the same time and *only* during the two capture cycles, the functional bus needs to be shared. Therefore, the additional testing time penalty is determined by serialized capture cycles for each consumer tested through the functional bus. However, when the producer is connected directly to consumers, the test application time penalty (for each test vector) is given by the loading time of the producer's outputs.

- *Delay Test Scheduling*: unlike the INTEST's case used for stuck-at faults, where the testing time for each core is solely dependent on the TAM width, for DELTEST implementation the testing time for each core is also affected by its schedule. This is due to the serial mechanism (e.g., multiplexing or daisy-chain architecture [1]), which is used for G_{prod} TAM's assignment. This mechanism is necessary since the producer list is *constantly changing from one test session to another*, and hence it is essential to achieve efficient G_{prod} TAM line usage. However, since all the producers used for the consumers tested concurrently need to be loaded serially, the loading time for each tested core is variable (*i.e., it depends on the order of loading all the necessary producer cores*). Hence, although we employ rectangles for core test representation, as in [9, 11], in the proposed DELTEST mode the rectangles cannot be precomputed. Therefore, for delay test scheduling we adapt the TAM_schedule_optimizer algorithm from [11] to fit our particular problems, i.e., accounting for the previously described *Producer-Consumer* and *Shared-Producer* conflicts and dynamic computation of the testing time caused by the serial mechanism for loading the producer's output wrapper cells.

W_{prod}	SOC g1023 T(cc)	SOC p34392 T(cc)	SOC p93791 T(cc)
1	222781	2427864	4567646
2	131620	3098148	4932849
3	135856	3391428	5656494
4	89513	3857560	7587866
5	108827	5216777	9650147
6	157214	7614231	14014672
7	311203	14992022	27993969

Table 1. Optimal Producer TAM Width

5 Experimental Results

To investigate the implication of the proposed approach on testing time, experiments were carried out for three benchmark SOCs: g1023, p34392, p93791. These three SOCs are originally part of the ITC02 *SOC test benchmarking initiative* ([15]). However, since the functional interconnects are not provided in the benchmark files we have randomly generated them to support the proposed approach, including the direct connection between cores and functional busses. We assume that all the cores with internal scan chains are tested using the proposed DELTEST modes, while the remaining non-scan cores were tested at-speed using the INTEST mode. To compare the proposed TAM division and delay test scheduling results with the results reported in [11], we assume that the number of test patterns is equal to the one provided in [15]. However, it should be noted that in reality the number of patterns for delay faults is usually higher than when targeting single stuck-at faults.

Table 1 presents testing times, T(cc), for different widths of the producer TAM (W_{prod}), when the overall TAM width W is fixed to 8. To give an exact TAM width division the functional interconnects are fixed in this experiment. It can be seen that the testing time of g1023 is minimum when W_{prod} is 4, while the testing time for p93791 and p34392 is minimum when W_{prod} is 1. This is due to the relationship between the total number of the producers' outputs and the internal scan flip flops (SFF) in the SOCs. On the one hand, in the case of g1023, the number of the SFF is comparable to the number of the producers' outputs, hence a large amount of testing time is necessary to load producers' outputs, thus leading to a higher number of producer TAM lines. On the other hand, for p93791 and p34392, the SFF number is significantly larger than the number of producers' outputs. As a result, the time necessary to load the internal scan chains dominates the testing time and hence only one TAM line is necessary to load the producers' outputs. It should also be noted that testing time variation with W_{prod} is a convex function, i.e., it decreases until it reaches its minimum for the *optimal* W_{prod}, after which point if W_{prod} is further increased then the testing time will grow as well.

	SOC p93791				
	INTEST [11]	NEW DELTEST			
W	T(cc)	T_{ave}(cc)	$T_{\max}$(cc)	$T_{\min}$(cc)	$\Delta T(\%)$
16	1851135	2289579	2410703	2210286	+23.69
32	975016	1269160	1343325	1156290	+30.17
48	627934	864777	899678	818277	+37.72
64	511286	661803	707815	636578	+29.44

Table 2. Testing Time Comparison for p93791

Table 2 presents results for DELTEST when varying the total TAM width W (note only results for the optimal W_{prod} lines are reported). Since the testing time is dependent on the functional interconnects, we ran the algorithm for 50 randomly generated interconnects. We have assumed that each core has 1 to 3 random producers and the SOC has two functional busses inside. $T_{ave}, T_{\max}, T_{\min}$ denote the average, maximum and minimum testing time. The percentage change in testing time using the new DELTEST is calculated using the formula $\Delta T(\%) = \frac{T_{ave}-T}{T} \times 100$, where T is the result reported in [11]. From Table 2 it can be seen that the average testing time increases on average about %30 when compared to the results in [11]. This increase is due to: (i) W_{prod} TAM lines used to load the excitation vector; (ii) test resource conflicts between cores in DELTEST before TAM optimization; (iii) DELTEST needs two capture cycles, while INTEST needs only one.

6 Conclusion

Motivated by the difficulty to deliver high quality two-pattern tests to embedded cores, this paper has presented a new delay fault testing approach for core-based SOCs. To the best of authors' knowledge this is the first attempt to provide an explicit mechanism for modular broadside delay fault testing of core-based SOCS. It was shown how IEEE P1500 can be extended using DELTEST instructions for consumer and producer cores, which ensure full controllability for every core's primary inputs. Solutions to address TAM division and system-level delay test scheduling have also been summarized. When compared to the existing approaches for stuck-at testing, it was demonstrated that testing time penalty is limited at the benefit of higher test quality achieved by broadside application of two-pattern tests.

References

[1] J. Aerts and E. J. Marinissen. Scan Chain Design for Test Time Reduction in Core-Based ICs. In *Proceedings IEEE International Test Conference (ITC)*, pages 448–457, Oct. 1998.

[2] M. Bushnell and V. Agrawal. *Essentials of Electronic Testing*. Kluwer Academic Publishers, 2000.

[3] K. Chakrabarty. Design of System-on-a-Chip Test Access Architectures Using Integer Linear Programming. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 127–134, Montreal, Canada, Apr. 2000.

[4] L. Chen and S. Dey. DEFUSE: A deterministic functional self-test methodology for processors. In *Proc. IEEE VLSI Test Symposium*, pages 255–262, 2000.

[5] L. Chen, S. Gupta, and M. Breuer. High quality robust tests for path delay faults. In *Proc. IEEE VLSI Test Symposium*, pages 88–93, 1997.

[6] I. Ghosh, S. Dey, and N. K. Jha. A fast and low cost testing technique for core-based system-chips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(8):863–876, August 2000.

[7] P. T. Gonciari, B. M. Al-Hashimi, and N. Nicolici. Useless memory allocation in system-on-a-chip test: Problems and solutions. In *Proc. IEEE VLSI Test Symposium*, pages 423–429, April 2002.

[8] I. Hamzaoglu and J. H. Patel. Compact two-pattern test set generation for combinational and full scan circuits. In *Proc. IEEE International Test Conference*, pages 944 –953, 1998.

[9] 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. In *Proc. IEEE Asian Test Symposium*, pages 265–270, November 2001.

[10] V. Iyengar, K. Chakrabarty, and E. Marinissen. Test wrapper and test access mechanism co-optimization for system-on-chip. *Journal Electronic Testing: Theory and Applications (JETTA)*, 18:211–228, March 2002.

[11] V. Iyengar, K. Chakrabarty, and E. Marrinissen. On using rectangle packing for SOC wrapper/TAM co-optimization. In *Proc. IEEE VLSI Test Symposium*, pages 253–258, 2002.

[12] W.-C. Lai and K.-T. Cheng. Instruction-level DFT for testing processor and IP cores in system-on-a-chip. In *Proc. IEEE/ACM Design Automation Conference (DAC)*, pages 59 –64, 2001.

[13] H. K. Lee and D. S. Ha. On the generation of test patterns for combinational circuits. Technical Report No. 12-93, Department of Electrical Engineering, Virginia Polytechnic Institute and State University, 1991.

[14] E. J. Marinissen, R. Arendsen, G. Bos, H. Dingemanse, M. Lousberg, and C. Wouters. A Structured And Scalable Mechanism for Test Access to Embedded Reusable Cores. In *Proceedings IEEE International Test Conference (ITC)*, pages 284–293, Oct. 1998.

[15] E. J. Marinissen, V. Iyengar, and K. Chakrabarty. ITC'02 SOC Test Benchmarks Web Site. http://www.extra.research.philips.com/itc02socbench/.

[16] P1500 SECT Task Forces. IEEE P1500 Web Site. http://grouper.ieee.org/groups/1500/.

[17] S. Ravi, G. Lakshminarayana, and N. K. Jha. Testing of core-based systems-on-a-chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(3):426–439, March 2001.

[18] J. Rearick. Too much delay fault coverage is a bad thing. In *Proc. IEEE International Test Conference*, pages 624–633, November 2001.

[19] Taiwan Semiconductor Manufacturing Corporation. TSMC 0.18μ CMOS technology. http://www.tsmc.com.