

of layers. For example, for three-layer routing, the circuit s38584 would need about $(496/2) \times 3 = 744$ MB.

V. CONCLUSION

We have proposed a novel multilevel routing framework MR considering both routability and performance. Unlike the previous multilevel routing, MR integrates global routing, detailed routing, and resource estimation together at each level of the framework, leading to more accurate routing resource estimation during coarsening and thus facilitating the solution refinement during uncoarsening. The exact routing information at each level makes our framework more flexible in dealing with various routing objectives (such as crosstalk, power, etc.). Experimental results have shown that MR is very promising. Future work lies in the development of a timing-driven multilevel router considering signal integrity.

ACKNOWLEDGMENT

The authors would like to thank the authors of [5], Prof. J. Cong, J. Fang, and Y. Zhang, for providing the benchmark circuits. Special thanks go to Y. Zhang for her prompt explanations of their data and very helpful discussions. They also thank the anonymous reviewers for their very constructive comments.

REFERENCES

- [1] C. Albrecht, "Global routing by new approximation algorithms for multicommodity flow," *IEEE Trans. Computer-Aided Design*, vol. 20, pp. 622–632, May 2001.
- [2] C. J. Alpert, J.-H. Huang, and A. B. Kahng, "Multilevel circuit partitioning," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 655–667, Aug. 1998.
- [3] Y.-W. Chang, K. Zhu, and D. F. Wong, "Timing-driven routing for symmetrical-array-based FPGAs," *ACM Trans. Design Automation Electron. Syst.*, vol. 5, no. 3, pp. 433–450, 2000.
- [4] T. Chan, J. Cong, T. Kong, and J. Shinnerl, "Multilevel optimization for large-scale circuit placement," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 2000, pp. 171–176.
- [5] J. Cong, J. Fang, and Y. Zhang, "Multilevel approach to full-chip gridless routing," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 2001, pp. 396–403.
- [6] J. Cong, J. Fang, and K. Khoo, "DUNE: A multi-layer gridless routing system with wire planning," in *Proc. ACM Int. Symp. Physical Design*, 2000, pp. 12–18.
- [7] J. Cong, A. Kahng, and K. Leung, "Efficient algorithms for the minimum shortest path steiner arborescence problem with applications to VLSI physical design," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 24–39, Jan. 1998.
- [8] J. Cong, S. Lim, and C. Wu, "Performance driven multilevel and multiway partitioning with retiming," in *Proc. ACM/IEEE Design Automation Conf.*, June 2000, pp. 274–279.
- [9] J. Cong and P. H. Madden, "Performance driven global routing for standard cell design," in *Proc. ACM Int. Symp. Physical Design*, Apr. 1997, pp. 73–80.
- [10] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong, "Provably good performance driven global routing," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 739–752, June 1992.
- [11] T. Deguchi, T. Koide, and S. Wakabayashi, "Timing-driven hierarchical global routing with wire-sizing and buffer-insertion for VLSI with multi-routing-layer," in *Proc. Asia South Pacific Design Automation Conf.*, June 2000, pp. 99–104.
- [12] M. Hayashi and S. Tsukiyama, "A hybrid hierarchical global router for multi-layer VLSI's," *IEICE Trans. Fundamentals*, vol. E78-A, no. 3, pp. 337–344, 1995.
- [13] J. Heisterman and T. Lengauer, "The efficient solutions of integer programs for hierarchical global routing," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 748–753, June 1991.
- [14] D. Hightower, "A solution to line routing problems on the continuous plane," in *Proc. Design Automation Workshop*, 1969, pp. 1–24.
- [15] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: application in VLSI domain," *IEEE Trans. VLSI Syst.*, vol. 7, pp. 69–79, Mar. 1999.
- [16] R. Kastner, E. Bozorgzadeh, and M. Sarrafzadeh, "Predictable routing," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 2000, pp. 110–114.
- [17] C.-Y. Lee, "An algorithm for path connection and its application," *IRE Trans. Comput.*, vol. EC-10, pp. 346–365, Sept. 1961.
- [18] Y.-L. Lin, Y.-C. Hsu, and F.-S. Tsai, "Hybrid routing," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 151–157, Feb. 1990.
- [19] J. Lillis, C.-K. Cheng, T.-T. Y. Lin, and C.-Y. Ho, "New performance driven routing techniques with explicit area/delay tradeoff and simultaneous wiresizing," in *Proc. Design Automation Conf.*, June 1996, pp. 395–400.
- [20] H.-C. Lee, Y.-W. Chang, J.-M. Hsu, and H. Yang, "Multilevel floorplanning/placement for large-scale modules using B*-trees," in *Proc. ACM/IEEE Design Automation Conf.*, Anaheim, CA, June 2003, pp. 812–817.
- [21] S.-P. Lin and Y.-W. Chang, "Novel framework for multilevel routing considering routability and performance," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 2002, pp. 44–50.
- [22] M. Marek-Sadowska, "Router planner for custom chip design," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1986, pp. 246–249.
- [23] G. Meixner and U. Lauther, "A new global router based on a flow model and linear assignment," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1990, pp. 44–47.
- [24] J. Soukup, "Fast maze router," in *Proc. ACM/IEEE Design Automation Conf.*, June 1978, pp. 100–102.
- [25] D. Wang and E. Kuh, "A new timing-driven multilayer MCM/IC routing algorithm," in *Proc. Multi-Chip Module Conf.*, Feb. 1997, pp. 89–94.

Testing SoC Interconnects for Signal Integrity Using Extended JTAG Architecture

Mohammad H. Tehranipour, Nisar Ahmed, and Mehrdad Nourani

Abstract—As technology shrinks and working frequency reaches the multigahertz range, designing and testing interconnects are no longer trivial issues. In this paper, we propose an enhanced boundary-scan architecture to test high-speed interconnects for signal integrity. This architecture includes: 1) a modified driving cell that generates patterns according to multiple transitions fault model and 2) an observation cell that monitors signal integrity violations. To fully comply with the conventional Joint Test Action Group Standard, two new instructions are used to control cells and scan activities in the integrity test mode.

Index Terms—Boundary-scan test, integrity loss, interconnect testing, Joint Test Action Group (JTAG) Standard, signal integrity, system-on-chip.

I. INTRODUCTION

A. Motivation

The number of cores in a system-on-chip (SoC) is rapidly growing, which leads to a significant increase in the number of interconnects. With fine miniaturization of the very large scale integrated (VLSI) circuits, existence of long interconnects in SoCs and rapid increase in the working frequency (currently in the gigahertz range), signal integrity

Manuscript received June 23, 2003. This work was supported in part by the National Science Foundation under CAREER Award #CCR-0130513. This paper was recommended by Associate Editor K. Chakrabarty.

The authors are with the Center for Integrated Circuits and Systems, The University of Texas at Dallas, Richardson, TX 75083-0688 USA (e-mail: mht021000@utdallas.edu; nxa018600@utdallas.edu; nourani@utdallas.edu).

Digital Object Identifier 10.1109/TCAD.2004.826540

has become a major concern for design and test engineers. Use of nanometer technology in SoCs magnifies cross-coupling effects among the interconnects. The coupling capacitance and mutual inductance affect the integrity of a signal by adding noise and delay. The noise effect can appear as overshoot and ringing. The former is known to shorten transistor lifetime and the latter to cause intermittent functional errors. Slowdown and performance degradation are often the result of excessive delay. Various parasitic factors such as parasitic capacitances, inductances, and their cross-coupling effects on the interconnects, are difficult to control during fabrication. These parasitic factors play a significant role in the ultimate functionality and performance of high-speed SoCs.

Signal integrity is the ability of a signal to generate correct responses in a circuit. It generally includes all effects that cause the design to malfunction due to distortion of the signal waveform. According to this informal definition, a signal with good integrity has: 1) voltage values at required levels and 2) level transitions at required times. If signal-integrity losses (i.e., noise and delay) on an interconnect are within the defined safe margin, they are acceptable, since they do no harm. Otherwise, they may cause intermittent logic-error, performance degradation, shorter life time, and reliability concern. For example, an input signal to a flip-flop with good signal integrity arrives early enough to guarantee the setup and hold time requirements and it does not have spikes causing undesired logic transition (ringing).

The impact of process variation on circuit operation is an important issue in deep submicron (DSM). Process variation and manufacturing defects both may lead to unacceptable levels of noise and delay. The goal of design for DSM is to minimize noise and delay. However, it is impossible to check and fix all possible signal-integrity problems during DSM design by only design rule checking (DRC), validation, and analysis. Process variations and manufacturing defects may lead to unexpected changes in coupling capacitances and mutual inductances between interconnects. They in turn result in loss of signal integrity (e.g., glitches and excessive delay), which may eventually cause logic error and failure of the chip. The impact of spot defects and process variations on the magnitude of inductance induced noise are reported in [1]. The authors reported that the maximum crosstalk pulse considering process variation is almost twice the value for the nominal set of parameters. Since it is impossible to predict the occurrence of defects that cause noise and delay, signal-integrity testing is essential to ensure error-free operation of the chip and must be addressed in manufacturing testing.

In recent years, various methodologies to test signal noise and skew on interconnects, due to different sources, are reported in literature. Regardless of the method used to detect integrity loss, we also need a mechanism to coordinate activities in an integrity test session. We believe that one of the best choices is the boundary scan test methodology that includes the capability of accessing interconnects. Boundary scan test methodology was initially introduced to facilitate testing complex printed circuit boards (PCBs). The IEEE 1149.1 Boundary Scan Test [5], also known as the Joint Test Action Group (JTAG) Standard, has been widely accepted and practiced in the testing community. The standard provides excellent testing features with low complexity, but it was not intended to address high-speed testing and signal-integrity loss. The standard, nevertheless, provides a mechanism to test core logic and the interconnects among them. Interconnects can be tested for stuck-at, open, and short faults [6]. In this paper, the standard boundary-scan architecture is extended to test interconnects for noise and skew violation. While we focus on interconnects, any nonmodeled fault (inside or outside cores) that manifests itself as integrity loss on interconnects will also be detected by our method.

B. Prior Work

1) *Signal Integrity Modeling and Analysis*: Maximum aggressor (MA) fault model [7] is one of the fault models proposed for crosstalk.

Various approaches to analyze the crosstalk are described in [8]–[10]. Interconnect design for multigigahertz integrated circuits is discussed in [11]. The author observed that chips failed, when a specific test pattern (not included in the MA model) is applied to the interconnects, due to the overall effect of coupling capacitances and mutual inductances. Similarly, according to [12], the worst case switching pattern to handle inductive effects for multiple signal lines may not be included in the MA fault model. Several researchers have worked on test pattern generation for crosstalk noise/delay and signal integrity [13]–[15].

2) *Test Methodologies*: There is a long list of possible design and fabrication solutions to reduce signal-integrity problems on the interconnect. None guarantees to resolve the issue perfectly. A double sampling data checking (DSDC) technique is used to capture noise-induced logic failures in on-chip buses [16]. At-speed testing of crosstalk in chip interconnects and testing interconnect-crosstalk defects using an on-chip processor are reported in [3], respectively. A built-in self-test (BIST)-based architecture to test long interconnects for signal integrity [4], and the use of boundary scan and I_{DDT} for testing buses [17] are other proposed methods. Even short interconnects, especially those located near long interconnects, are also susceptible to integrity problems. Therefore, in the near future, methodologies are required for testing both short and long high-speed interconnects [18].

3) *Integrity Loss Sensor Cell*: Due to increasing concerns about signal-integrity loss in gigahertz chips and the fact that their occurrence must be captured on the chip, researchers presented various on-chip integrity loss sensors (ILS). A BIST structure using D flip-flops has been proposed to detect deviation of propagation delay in operational amplifiers [19]. In [17], a built-in sensor is integrated within the system. This sensor is an on-chip current mirror converting dissipated charges into associated test time. Reference [20] presented a more expensive but more accurate circuit to measure jitter and skew in the range of a few picoseconds. The authors in [21] presented a sample-and-hold circuit that probes the voltage directly within the interconnects. The work presented in [4] proposed two cells called noise detector (ND) and skew detector (SD) cells based on a modified cross-coupled PMOS differential sense amplifier. To detect delay violation, an integrity-loss sensor (ILS) has been designed in [22], which is flexible and tunable for various delay thresholds and technologies. A double sampling technique is applied by online-error detector circuit to test multiple-source noise-induced errors on the interconnects and buses [16].

4) *Modified Boundary Scan and IEEE Standards*: BIST-based test pattern generators for board-level interconnect and delay testing are proposed in [23] and [24], respectively. A test methodology targeting defects on bus structures using I_{DDT} and boundary scan has been presented in [17]. P1500 proposes standardization of a core test wrapper and core test language (CTL) [25]. This method is similar to boundary scan in terms of serial data transfer and therefore, the extended serial interface layer (SIL) architecture can be used for various test applications at the system level in general, and in integrity test in particular.

IEEE 1149.4 mixed-signal test bus standard [26] was proposed to allow access to the analog pins of a mixed-signal device. In addition to the ability to test interconnects using digital patterns, IEEE 1149.4 includes the ability to measure actual passive components, such as resistors and capacitors. This standard cannot support high-frequency phenomena such as crosstalk on the interconnects. Reference [27] proposes a method to simplify the development of a mixed-signal test standard by adding the analog interconnect test to IEEE 1149.1. IEEE 1149.6 provides a solution for testing ac-coupled interconnects between integrated circuits on printed circuit boards and systems [28]. Our approach is similar to this standard draft in strategy of enhancing the JTAG standard and its instructions for testing high-frequency behaviors. However, there are fundamental differences. Contrary to our approach, the standard is not intended to consider coupling effects among the inter-

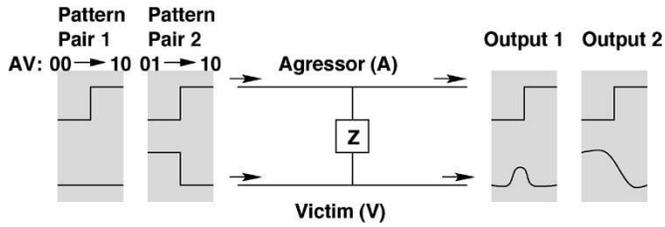


Fig. 1. Signal integrity fault model.

connect lines. Also, 1149.6 adds a dc-blocking capacitor to each interconnect under test to disallow the dc signals. Thus, IEEE 1149.6 cannot test integrity loss due to low-speed, but very sharp-edge signals that are known to cause overshoots and noise. The sensors in our architecture, sitting in the observation boundary-scan cells (BSCs), can detect such scenarios. Finally, using differential drivers in the modified cells in IEEE 1149.6 makes the cells more expensive and less flexible in adopting other types of noise detectors/sensors. Various issues on the extended JTAG architecture to test SoC interconnects for signal integrity are reported in [22], [29], and [30], using the MA and multiple transition (MT) fault models, respectively.

C. Contribution and Paper Organization

Our main contribution is an on-chip mechanism to extend the JTAG standard to include testing interconnects for signal integrity. The modified driving-end pattern generation BSCs (called PGBSCs) receive a few seeds and generate MT patterns at-speed to stimulate integrity violations. The MT pattern set is a superset of the MA set and is much more capable of testing the capacitive and inductive coupling among interconnects. The modified receiving-end BSCs (called OBSC) record the occurrence of signals entering the vulnerable region over a period of operation. Using two new instructions in JTAG architecture, the integrity test information is sent out for a final test analysis, reliability judgment, and diagnosis.

The rest of this paper is organized as follows. Section II describes the MT fault model and its corresponding test patterns. The enhanced BSCs are detailed in Section III. Section IV explains the test architecture to send test patterns and capture and read out the signal integrity information. The experimental results including implementation and simulation of our own ILS are discussed in Section V. Finally, concluding remarks are in Section VI.

II. MT-FAULT MODEL

The MA fault model [7] is a simplified model used by many researchers mainly for crosstalk analysis and testing. This model, shown in Fig. 1, assumes the signal traveling on a victim line V may be affected by signals/transitions on other aggressor line(s) A in its neighborhood. The coupling can be represented by a generic coupling component Z . In general, the result could be noise (causing ringing and functional error) and delay (causing performance degradation). However, there is controversy as to what patterns trigger maximal integrity loss. Specifically, in the traditional MA model that takes only coupling C into account, all aggressors make the same simultaneous transition in the same direction, while the victim line is kept quiescent (for maximal ringing) or makes an opposite transition (for maximal delay). Fig. 2 shows the test patterns for detecting faults according to MA model. When mutual inductance comes into play, some researchers have shown that the MA model may not reflect the worst case, and presented other ways (pseudorandom, weighted pseudorandom, or deterministic) to generate test patterns to create maximal integrity loss [13]–[15].

	P_{g0}	P_{g1}	N_{g1}	N_{g0}	d_r	d_f
A	↑	↑	↓	↓	↑	↓
V	0	1	1	0	↓	↑
A	↑	↑	↓	↓	↑	↓
Vector 1: AVA	000	010	111	101	010	101
Vector 2: AVA	101	111	010	000	101	010

Fig. 2. MA fault model and test patterns.

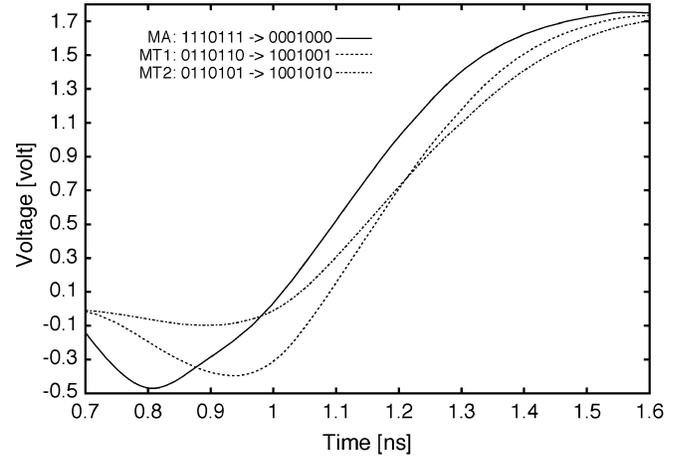


Fig. 3. Comparison between the MT and MA models.

As reported in [11], a chip fails when the nearest aggressor lines change in one direction and the other aggressors in the opposite direction. This and many similar carefully chosen scenarios are not covered by the MA fault model. Exhaustive testing covers all situations, but it is very time consuming because of the huge number of test patterns. Additionally, exhaustive or pseudorandom patterns include some cases where aggressors are in quiescent mode and obviously do not maximally affect the victim line for noise and delay. Therefore, they need not be considered in model or pattern generators. Based on these observations reported by researchers, we define a new fault model and corresponding test set which covers *all* transitions on victim and *multiple* transitions on aggressors. The main idea behind our proposed fault model, MT, is having a single victim, a limited number of aggressors, a full transition on victim, and multiple transition aggressors. In the MT model, all possible transitions on the victim and aggressors are applied, while in the MA model, only a subset of these transitions are generated. Another difference is single versus double direction change of aggressors in MA and MT, respectively. Briefly, the MA pattern set is a subset of the MT pattern set. Note carefully that the MT model is not an exhaustive model because it does not cover quiescent cases of aggressor lines for which integrity loss will not be obviously maximal.

As a motivating example, Fig. 3 shows the simulation results of two MT patterns: 1) 0110110 \rightarrow 1001001 and 2) 0110101 \rightarrow 1001010 and one MA pattern, i.e., 1110111 \rightarrow 0001000 applied to a seven interconnect system while the middle wire is the victim and the others are aggressors. Extraction and simulation are done by an OEA tool (BUSAN) [31] and TISPICE [32], respectively. We extracted the distributed resistance–inductance–capacitance (RLC) and coupling capacitances and mutual inductances using OEA tool for 0.18- μ m technology. As shown, MT patterns create more delay as compared with the MA pattern, ranging from 35 to 70 ps, depending on the buffer size. Therefore, MA patterns may not be able to generate maximum noise/delay on the victim line, especially when inductance is included. In our opinion, this is the main reason why some researchers (e.g., [11])

Quiescent at 0 x0x x0x	Transition 0 → 1 x0x x1x	Quiescent at 1 x1x x1x	Transition 1 → 0 x1x x0x
000 101	000 111	010 111	010 101
001 100	001 110	011 110	011 100
100 001	100 011	110 011	110 001
101 000	101 010	111 010	111 000

Fig. 4. All transitions on a three-interconnect system that the MT and MA (shaded) models generate.

reported scenarios in which test patterns not covered by the MA model failed a chip. Fig. 4 shows all possible transitions on three line interconnects, where the middle one is victim based on the MT fault model. Test patterns for signal integrity are vector pairs. For example, when the victim line is kept quiescent at “0” (column 1), four possible transitions on the aggressors are examined. The MA patterns (a subset of MT patterns) are shaded in Fig. 4.

Four cases are examined for each victim line when victim line is quiescent at “0,” “1,” changes from “0” to “1” or “1” to “0.” As shown in Fig. 4, the number of required test patterns to cover all possible transitions on the three interconnects is $4 * 2^{3-1} = 16$, when the middle line is in victim mode. The total number of required test patterns is $3 * 16 = 48$, when all three lines are examined in victim mode. In general, the number of test patterns for a group of m interconnects is $N_P = m \cdot 4 \cdot 2^{m-1} = m \cdot 2^{m+1}$. When m increases, the number of test patterns increases exponentially. Simulations show that in an interconnect system, the lines which are far away from the victim cannot affect the victim line much [11], [15]. Therefore, the number of lines (aggressor) before and after the victim line can be limited. We define k as the locality factor that is empirically determined showing how far the effect of aggressor lines remain significant. Briefly, the total number of patterns to be generated will be $m \cdot 2^{m+1}$, where $m = 2k + 1$. By choosing k (e.g., $k < n$ or $k \ll n$, where n is the total number of interconnects under test) users can do a tradeoff between time and accuracy.

Fig. 5 shows the simulation results, using OEA and TISPICE tools [31], [32] of a different number of aggressors in the victim neighborhood while the victim line is quiescent at 0 for $V_{dd} = 1.8$ V. As shown, when the number of interconnects on either side of the victim increases, the noise on the victim increases. The glitch height and period mainly depend on the driver strength. As the driver strength increases, the glitch height increases and the period decreases. The peak noise difference corresponding to two locality factors of $k = 3$ and $k = 4$ is $V_{peak}(k = 4) - V_{peak}(k = 3) = 0.048$ V which, for many applications, can be assumed negligible. If that is the case, $k = 3$ can be used as the locality factor in the simulation and pattern generation. The percentage difference between glitches of different distributions with respect to k decreases with an increase in driver strength [7]. We acknowledge that finding such a locality factor is technology-dependent (e.g., parasitic RLC values) and application-dependent (e.g., depending on driven core or shielding techniques). However, once provided based on the application and accurate simulation, the total number of patterns and time to test integrity faults will be significantly reduced. Our observation is that, for current technologies and simulation based on the

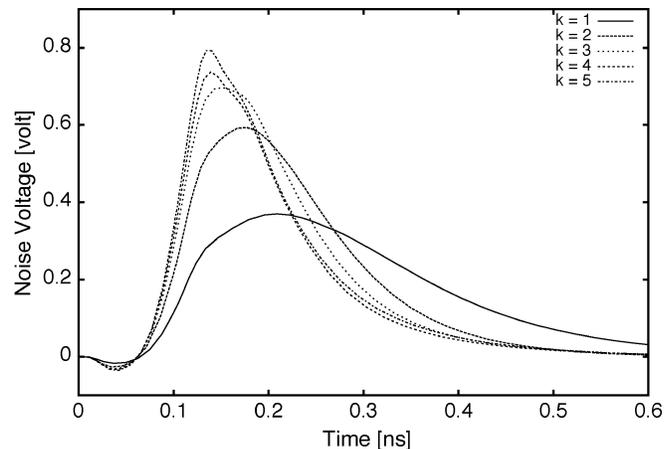


Fig. 5. Simulation results for different k .

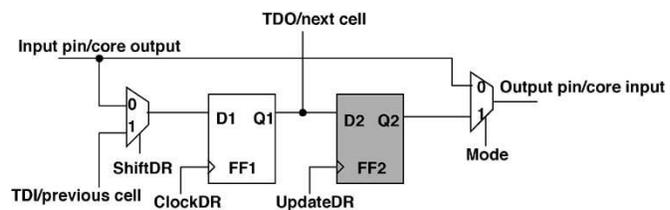


Fig. 6. Standard BSC.

reduced order models, a small k (often less than 4) can produce accurate behavior of the interconnects. In any case, by choosing k the user can always tradeoff between longer simulation time and accuracy.

III. ENHANCED BSCs

Boundary scan is a widely used test technique that requires BSCs to be placed between each input or output pin and the internal core logic. The standard provides an efficient mechanism for functional testing of core logic and interconnects. Fig. 6 shows a conventional standard BSC with shift and update stages. $Mode = 1$ puts the cell in test mode. The data is transferred into the shift register (in Shift-DR state) during scan operation. Test patterns scanned into the BSCs through the scan-in port (TDI) are applied in parallel in Update-DR state (using UpdateDR signal). Circuit response is captured in parallel by the BSCs connected between internal logic and output pins, and is scanned out through the scan-out port (TDO) for observation [5].

Using the JTAG standard (IEEE 1149.1), interconnects can be tested for stuck-at, open, and short faults. This is possible by EXTEST instruction, by which the test access port (TAP) controller isolates the core logic from the interconnects using the BSCs. However, EXTEST was not intended to test interconnects for signal integrity. We propose new cells and instructions for signal integrity testing. For this purpose, some minor modifications are applied to the standard architecture to target interconnects for signal integrity.

A. Pattern Generation BSC (PGBSC)

Analysis of the MT fault model test vector-pair shows that in some transitions the victim line should remain quiescent while the aggressor lines change. In some other transitions, both the victim and aggressor lines change. Additionally, in all cases the aggressor lines change from one value to another (“0” to “1” or “1” to “0”) at every clock, while the victim line value changes every two clocks. This important observation helps us design a circuit to efficiently generate MT test patterns. Fig. 7 shows test vectors of Fig. 4 resorted to make the point clear. Each row

Seed	Quiescent at 0 x0x x0x	Transition 0 → 1 x0x x1x	Quiescent at 1 x1x x1x	Transition 1 → 0 x1x x0x
000	000 101	101 010	010 111	111 000
001	001 100	100 011	011 110	110 001
100	100 001	001 110	110 011	011 100
101	101 000	000 111	111 010	010 101

Fig. 7. Resorted test pattern for a three interconnect system.

clearly shows that the victim changes every two clocks, and the aggressors change every clock.

Each row in Fig. 7 needs one seed. For example, for seed = “000,” these test patterns (shown in the first row) are generated: (“000,” “101”), (“101,” “010”), (“010,” “111”), and (“111,” “000”). To cover all possible transitions as shown in Fig. 7, four seeds are required. In the above three interconnect system, these seeds are “000,” “001,” “100,” and “101.” The total number of required seeds to cover all of the lines in victim mode in a three interconnect system is $3 * 4 = 12$. For a group of m interconnects, where $m = 2k + 1$, the total number of seeds are $N_S = m \cdot 2^{m-1} = (2k + 1) \cdot 2^{2k}$, in which 2^{m-1} shows all the possible combinations of $m - 1$ aggressor lines. Note carefully that as we discussed in Section II, since the locality factor k is very small, the total number of seeds remains very limited. For example, two seeds are sufficient to stimulate maximum delay in the interconnect according to the results in [12].

The MT test patterns cover the MA test patterns. In Fig. 7, the shaded patterns show MA patterns which are generated based on only two seeds. It shows that after applying the first seed, “000,” the generated test patterns cover the P_{g0} , d_f , and P_{g1} faults. The generated test patterns after applying the second seed, “101,” cover the d_r , N_{g0} and N_{g1} faults (see Fig. 3). Therefore, by such reordering, two seeds are sufficient for covering all 12 test patterns in the MA fault model.

A pair of test vectors are required to test the interconnects for signal integrity. These patterns can be applied to the interconnects using conventional boundary scan architecture. To apply each pair, the first and second patterns are scanned into the conventional BSCs and stored in FF_2 and FF_1 , respectively (see Fig. 6). Then, using UpdateDR, they are applied to the interconnects. Scanning and applying patterns in this way are very straightforward but need a large number of clocks which increase the overall test time. We propose a hardware-based method for test-pattern generation for the MT fault model. Test-pattern generation is performed at the input side of interconnects, that is, the output side of a core which drives the interconnects. The new BSC that generates test patterns is called PGBSC.

BSC can be utilized to support the MT model. FF_2 in conventional BSC (shaded flip-flop in Fig. 6) can be used to generate test patterns based on a given seed. FF_1 is used to initialize FF_2 . A T flip-flop divides the clock in half and UpdateDR plays the same role as the clock in driving FF_2 (see Fig. 8). First, the seed comes from TDI and is stored in FF_2 through FF_1 . Victim/aggressor select signals are then scanned into FF_1 through TDI to select the victim and aggressor lines. Note carefully that only the seeds need to be scanned in instead of exact test patterns. This significantly reduces the number of required clock cycles for applying test patterns to the interconnects in a system-chip.

In addition to its normal (nonintegrity) mode, PGBSC should work as victim or aggressor in signal integrity test mode. The PGBSC architecture is shown in Fig. 8(a). Note that additional components in PGBSC are located in the test path and, therefore, no additional delay is imposed on the normal mode. Only one extra control signal (SI) is needed in this architecture. The SI signal is generated by a new instruction, to be explained in Section IV. PGBSC generates required test patterns to cover the MT fault model. Fig. 8(b) shows the operation modes of the PGBSC. Depending on the select line of the MUX attached to FF_3 , this architecture has three modes:

- 1) **Victim Mode:** Q_3 is selected. UpdateDR is divided by two and applied to FF_2 . By every two UpdateDRs, the complemented data is generated in $\overline{Q_2}$ and it is transferred to the output pin.
- 2) **Aggressor Mode:** UpdateDR is selected, but PGBSC is in signal integrity mode. UpdateDR is applied to FF_2 . By each UpdateDR, the complemented data is generated in $\overline{Q_2}$ and it is transferred to the output pin through feedback and Q_2 .
- 3) **Nonintegrity Mode:** UpdateDR is selected. It is the normal test mode of the PGBSC (e.g. conventional boundary scan test) and UpdateDR is applied to FF_2 .

Fig. 9 shows the operation of a PGBSC. If PGBSC is in victim mode, UpdateDR is divided by two and generates $CLK - FF_2$. If the initial value in Q_2 is “0,” then $\overline{Q_2}$ is “1” and is applied to D_2 through the feedback. Every two activations of UpdateDR, the content of FF_2 is complemented. On the other hand, if PGBSC is in aggressor mode, $CLK - FF_2$ has the same frequency as UpdateDR and by each UpdateDR the content of FF_2 is complemented. As shown in Fig. 8, $\overline{Q_2}$ is complemented by each $CLK - FF_2$, while Q_2 is applied to the interconnect.

Each interconnect can act as a victim or an aggressor. Therefore, in each test session, the victim interconnect should be specified. After the victim is tested, it will become an aggressor for other new victims. Briefly, for complete interconnect testing, the victim line rotates. One of the major advantages of limiting the number of aggressor lines is that parallel testing of interconnects becomes possible. In other words, in each step of the test we deal with at most k lines as aggressors before and k lines after the victim line. We use encoded data, called *victim-select data*, to specify the victim.

Table I shows the victim-select data for an n -bit interconnect system, to be scanned and stored in FF_1 when the locality factor is $k = 2$. After specifying the victim, test vectors are generated by PGBSC and applied to the interconnects. Then, the new victim line is specified and the process will be repeated for the new victim. As shown in Table I, when we scan “100100...100” into n PGBSCs, the first line is the victim and the next two lines are aggressors for the first interconnect as well as for the fourth one. Therefore, $\lceil n/(k+1) \rceil = \lceil n/3 \rceil$ victims are tested simultaneously. As shown, with one clock the victims locations rotates (“0100100...10”). For $k = 2$, two rotations (clocks) are enough to set victim/aggressor nets and cover the entire interconnect.

The generic behavior of test-pattern generation and applying the procedure is shown in Fig. 10. This behavior will be executed by a combination of the automatic test equipment (ATE) and the TAP controller. The first seed is applied to new BSCs as an initial value into FF_2 . Victim and aggressors are selected with the victim-select data scanned into FF_1 , and then the cells are set in SI mode to start generating test patterns. After generating test vectors and applying them to interconnects, a new victim is selected and the process will be repeated with the same initial value. Note carefully that at the end of testing one victim again (line 8) the same seed would be in FF_2 (See also Fig. 7 for an example). Therefore, we do not need to scan in the seed for the next victim. This significantly reduces the overall test time. The same process will be repeated for all the seeds.

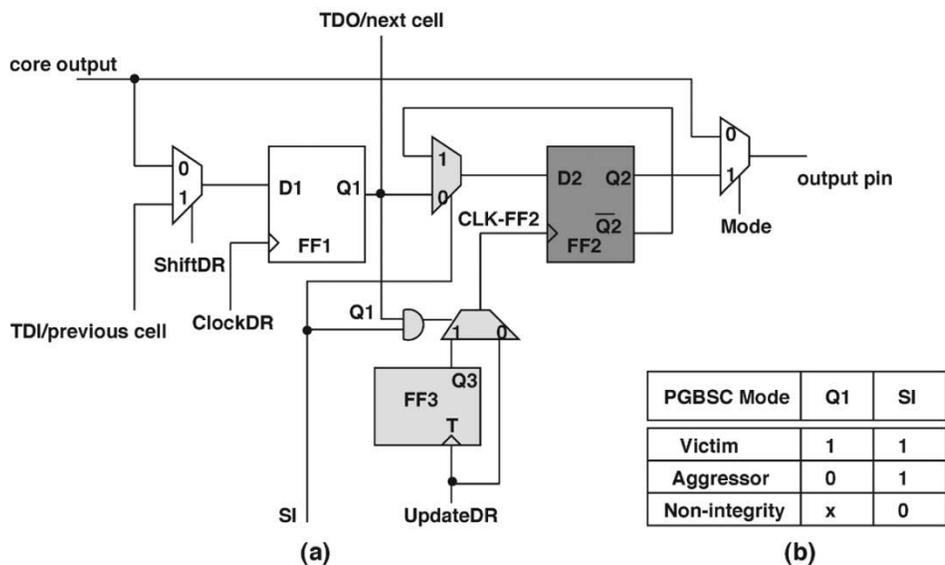


Fig. 8. PGBSC structure.

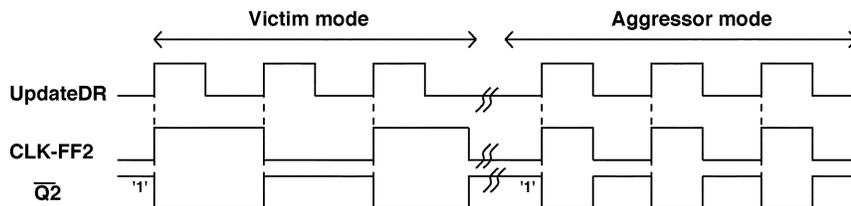


Fig. 9. Operation of the PGBSC.

TABLE I
VICTIM-SELECT DATA IN AN n -bit INTERCONNECT SYSTEM WHEN $k = 2$

Victim Location	Victim-Select Data
VAAVAA...VAA	100100...100
AVAAVAA...VA	0100100...10
AAVAVAA...V	00100100...1

```

01: for ( $i=1$  to  $N_S$ )
02: {
03:   Scan seed  $i$  into  $FF_2$ 
04:   Activate signal integrity test mode ( $SI=1$ )
05:   Scan the first victim-select data
06:   For ( $j=1$  to  $k$ ) //Total # of shifts for victim-select data
07:   {
08:     Apply 4  $UpdateDRs$ . // Pattern generation
09:     Shift one '0' into  $FF_1$  // Selecting new victim
10:   }
11: }
    
```

Fig. 10. Test-pattern generation procedure using PGBSC.

1) *Comment on External Test Patterns:* PGBSC is implemented based on a slight modification of the BSC. The MT test patterns are generated and applied by PGBSC, which can be also efficiently used for external (deterministic or weighted-pseudorandom) patterns. These patterns are applied by an external tester through TDI input pin. More specifically, PGBSC cells provide extra control and flexibility to scan in patterns and apply them to interconnects, and they even mix the MT model with other fault models. The flexibility of a PGBSC cell can be further explored for various tradeoffs. For example, a simple yet efficient compression technique for the enhanced boundary-scan architec-

ture using the PGBSC cells to minimize delivery time is discussed in [22].

B. Observation BSC (OBSC)

We propose a new BSC at the receiving side of interconnects which can employ any ILS such as the one presented in [22]. Fig. 11(a) shows the new BSC, called OBSC. As shown, the ILS circuits, added to the receiving cells, capture signals with noise and delay at the end of an interconnect. When it receives a signal with an integrity problem (noise or skew violation) it produces a pulse at its output and the FF is set to "1." The ILS is activated by the signal cell enable ($CE = "1"$). If $CE = "0"$, the ILS is disabled but captured data in its flip-flop remain unchanged. The OBSC operates in two modes as summarized in Fig. 11(b).

- 1) **ILS Mode:** ILS flip-flop is selected. In this case, every time the *shiftDR* signal becomes active the captured ILS data is scanned through the scan chain for final evaluation.
- 2) **Nonintegrity Mode:** In this mode, ILS is isolated and each OBSC acts as a conventional (nonintegrity) BSC.

In SI test mode, as Fig. 11 shows, ILS FF can be read and scanned out for final evaluation. Before starting the scan out process, we need to send the content of ILS FF to FF_1 . In this case, sel should be zero. Therefore, the SI and ShiftDR signals should be "1" and "0," respectively. When the scanning out process starts, D_1 is transferred to Q_1 to be used as a TDI for the next cell. After sending the value of ILS FF to Q_1 , a scan chain must be formed. In this case, during the Shift-DR state, TDI input must be connected to FF_1 . Therefore, the ILS path should be isolated by sel = "1" ($SI = "1"$ and ShiftDR = "1"). As shown in Fig. 11, \bar{SI} and ShiftDR are ORED together for selecting the ILS path for transferring ILS FF to D_1 and scanning out data in the chain. Fig. 12 shows the dependency of sel on SI and ShiftDR.

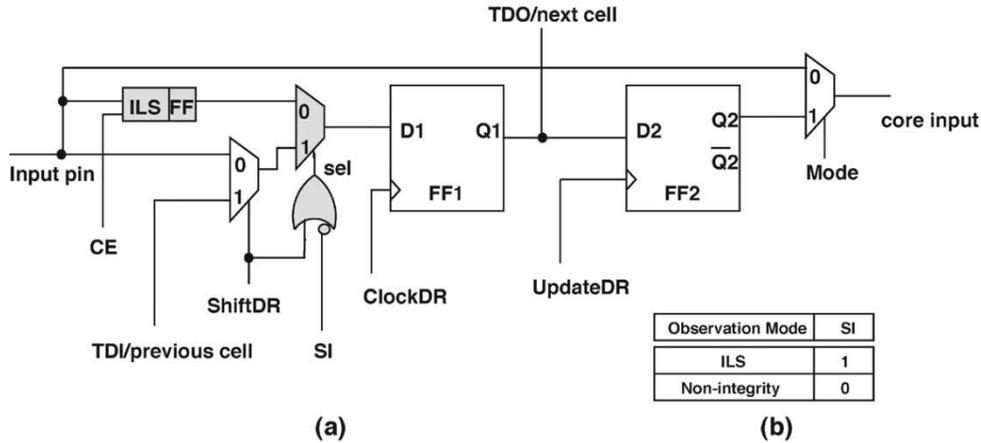


Fig. 11. OBSC structure.

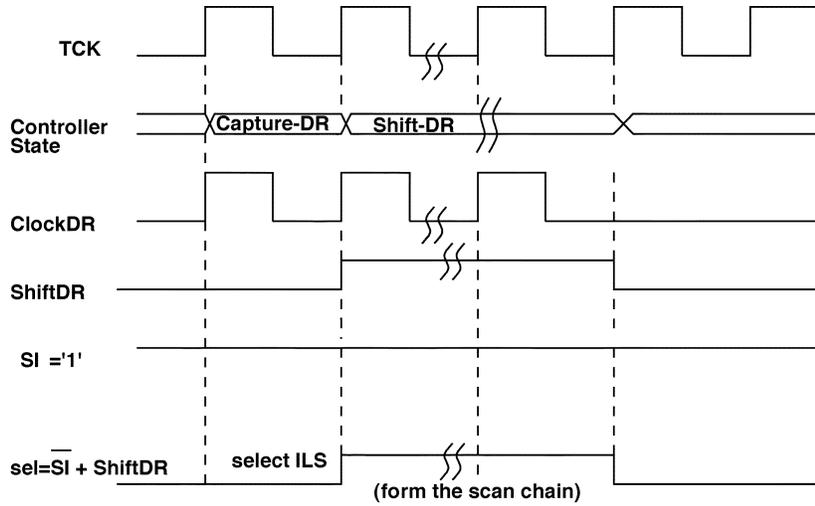


Fig. 12. Operation of the OBSC.

TABLE II
TRUTH TABLE OF SIGNAL sel

SI	ShiftDR	sel
1	0	0
1	1	1
0	x	1

As shown, in Capture-DR state, ILS FF is selected and then in the Shift-DR state a scan chain is formed and data is scanned out depending on how many wires are under test. Table II shows the truth table of signal sel. Additional control signals (i.e., SI and CE) are generated by a new instruction to be explained in Section IV. There are three methods of observation:

- 1) **Method 1:** to capture and scan-out ILS data only once, after applying the entire test pattern set and covering all victims;
- 2) **Method 2:** to capture and scan-out ILS data N_S times, after the application of a group of patterns (e.g., the patterns generated based on a specified seed) for each victim;
- 3) **Method 3:** to capture and scan-out ILS data right after applying each test pattern.

The first method has the least test time and a disadvantage for not being able to determine which transitions have caused the fault in an interconnect. The second method provides more information,

to determine which set of transitions or faults caused the violation in the interconnects, at the expense of more test time. Finally, the third method is the most informative one, but it is extremely time consuming. In Section V, we will report the experimentation and compare these methods. Although power issue is beyond the scope of this paper, we would like to comment that extra switching activities, due to the MT model, will be confined in BSCs. In other words, as Fig. 11 shows, cores sitting on the two sides of interconnects are isolated in signal integrity test mode and do not suffer from excessive power.

IV. TEST ARCHITECTURE

Fig. 13 shows the overall test architecture with n interconnects between cores i and j in a two-core SoC. The five standard JTAG interfaces (TDI, TCK, TMS, TRST, and TDO) are still used without any modification. The TAP controller and instruction register provide various control signals to BSCs (Fig. 6). The mandatory instruction PRELOAD is initially used to scan data into BSCs. Two new instructions are defined for the signal-integrity test, one to activate PGBSCs to generate test patterns and the other to read out the test results. As shown in Fig. 13, the cells at the output pins of core i are changed to PGBSCs and the cells at the input pins of core j are changed to the OBSCs. The remaining cells are standard BSCs, which are present in the scan chain

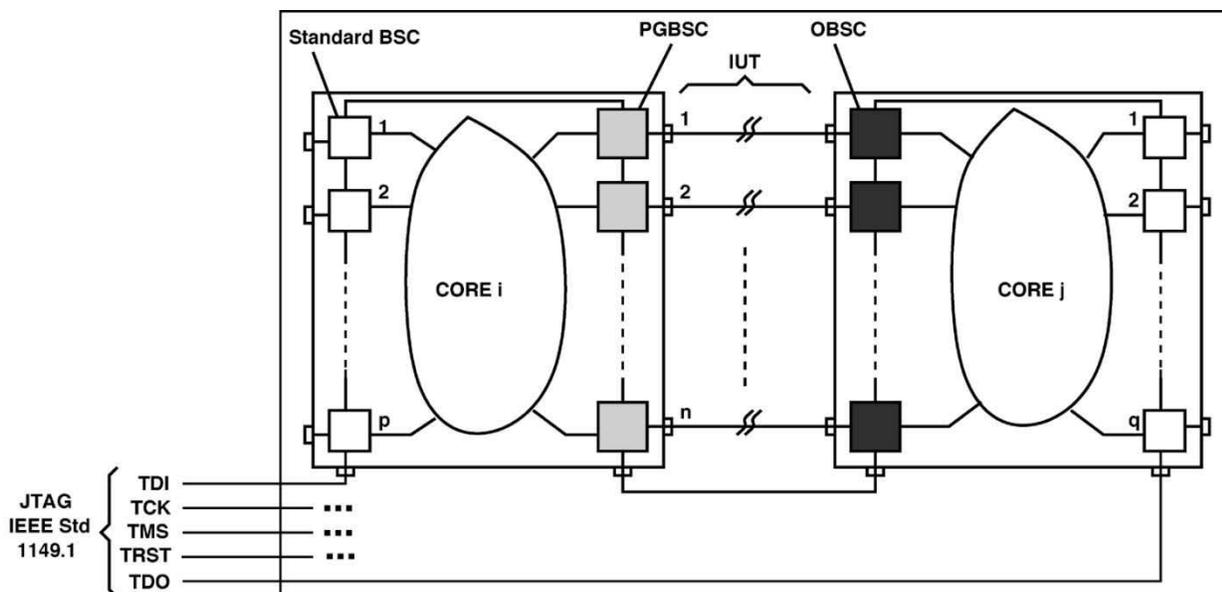


Fig. 13. Test architecture.

during signal integrity test mode. In the case of bidirectional interconnects, BSCs used at both ends are a combination of PGBSC and OBSC to test the interconnects in both directions.

The ILS part of OBSCs does not need any special control and automatically captures the occurrence of integrity loss. After all, patterns are generated and applied, signal-integrity information stored in the ILS FF is scanned out to determine which interconnect has a problem. In Section III-B, we have presented various methods to readout this information. For example, Method 1 can be used to minimize the overall test-application time, since the information in the cells is scanned out once per interconnect as opposed to once per pattern.

A. Clock Analysis

In this section, we analyze the required clocks to apply the test patterns and read out the test results for different methods. Assume that n is the number of interconnects under test, p is the number of BSCs in the scan chain before the first PGBSC, and q is the number of BSCs in the scan chain after the last OBSC (see Fig. 13).

In conventional BSC, test patterns are scanned in and applied to the interconnects one-by-one. In a n -wire interconnect network, the number of required clocks to scan and apply the MT test patterns by conventional boundary scan architecture through TDI is $N_{clk_BS} = m \cdot N_p(n+4) + p$, where N_p is the total number of MT patterns and $m = 2k + 1$. When PGBSC is used, only seeds are scanned into the cells and the other test patterns are generated internally. Having n interconnects, the required clocks in the enhanced boundary scan method is $N_{clk_EBS} = N_S(2n+8k) + p$, where k is the locality factor defined in Section II. This formula essentially counts clocks when the architecture executes the test generation algorithm shown in Fig. 10. Tables III and IV summarize the exact number of required clocks for test generation/application and readout, respectively. Comparing conventional and enhanced boundary scan, test application time reduction (TR) will be

$$TR\% = \frac{N_{clk_BS} - N_{clk_EBS}}{N_{clk_BS}} \cdot 100\%. \quad (1)$$

B. New Test Instructions

IEEE 1149.1 allows the user to define some optional instructions. We propose to add two new instructions G-SITEST and O-SITEST to

TABLE III
PATTERN GENERATION AND APPLICATION-CLOCK ANALYSIS

Test Architecture	N_{clk}
Conventional BSC	$m \cdot N_p(n+4) + p$
Enhanced BSC	$N_S(2n+8k) + p$

TABLE IV
CLOCK ANALYSIS OF THE READOUT METHODS

Readout Methods	N_{clk}
Method 1	$n+2q$
Method 2	$N_S \cdot n+2q$
Method 3	$N_p(n-1)/2+2q$

the mandatory instruction set. These new instructions are only used in signal integrity mode for testing interconnects.

1) *G-SITEST Instruction*: This instruction is used for test pattern generation using the enhanced boundary scan architecture. Seeds typically would be loaded onto the BSCs using the PRELOAD instruction before loading the G-SITEST instruction. Thus, when the G-SITEST instruction takes place in *Update-IR* controller state, the related signals will be activated. G-SITEST targets PGBSCs and enables $SI = 1$ during execution of instruction. It also enables the ILS cells (i.e., $CE = 1$) to capture signal integrity information. The victim-select data is then shifted into FF_1 of PGBSCs during the Shift-DR state and patterns for the MT fault model are generated every Update-DR state as explained in Section III-A. Three UpdateDRs are required to generate three test patterns per victim line for each initial value. Each BSC at the output of a core whose interconnect is under test will behave according to G-SITEST instruction. The flow of data through PGBSC is shown in Fig. 14. Core i executes the PRELOAD instruction and the cores before core i execute the BYPASS instruction to scan in initial data (seeds) from the system input. The seeds are scanned in (broken lines in Fig. 14) while other necessary changes to generate patterns are done internally using additional components (shaded components in Fig. 14) in the PGBSC cell.

2) *O-SITEST Instruction*: This instruction is loaded after execution of the G-SITEST instruction and when the loss information has been stored in ILS FF. Assuming all test patterns have been applied to in-

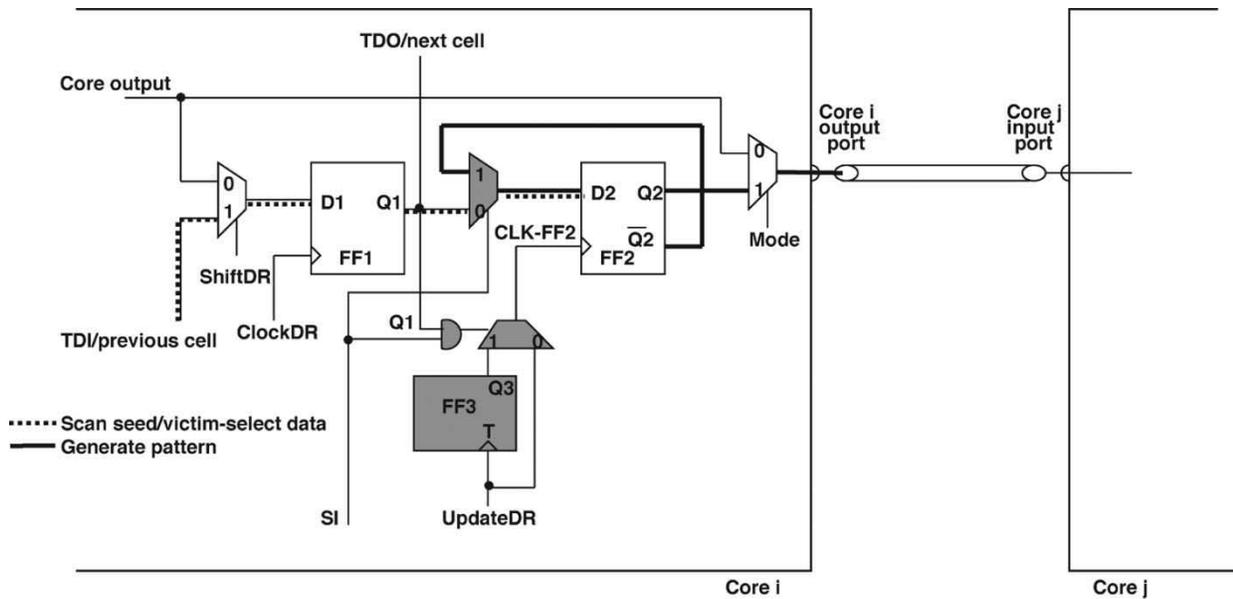


Fig. 14. Test-data flow while the G-SITEST instruction is executed.

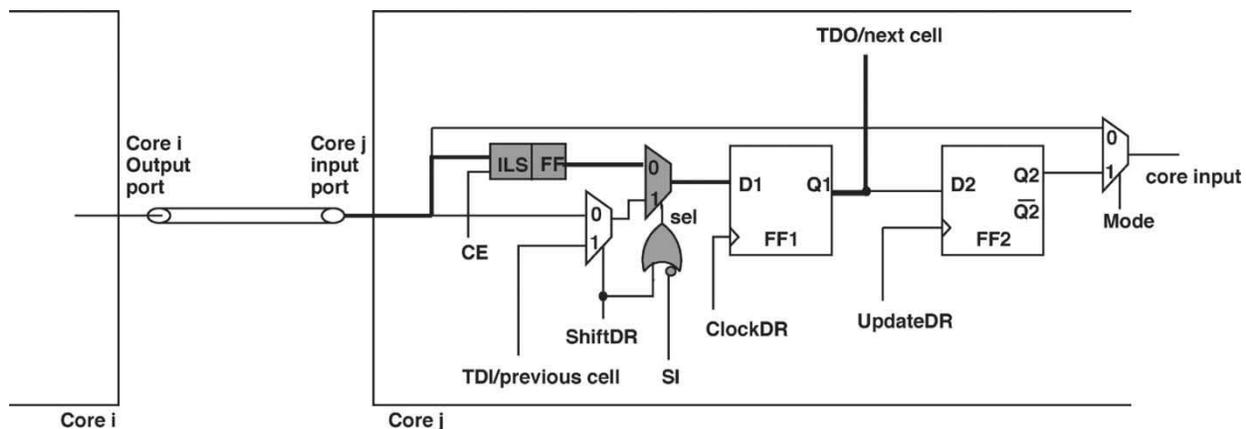


Fig. 15. Test-data flow while the O-SITEST instruction is selected.

terconnects, O-SITEST instruction is used to capture and scan out the content of ILS FF. After the instruction is decoded in the Update-IR state, control signals $SI = 1$ and $CE = 0$ (to deactivate ILS) are generated. With $CE = 0$, ILS will not receive input data anymore as the O-SITEST instruction is executed. The flow of data through the OBSC cell is shown in Fig. 15. All cores after core j execute the BYPASS instruction to scan out data to the system output.

Having two instructions in the integrity-test mode is necessary in enhanced boundary-scan architecture. G-SITEST enables PGBSC to generate and apply test patterns and, simultaneously, OBSC captures signals at the receiving end of the interconnects. After test application is performed, reading out the integrity information is begun by using the O-SITEST instruction. These two different operations cannot be done by only one instruction because FF_1 is reused for different purposes. Specifically, the content of ILS FF, which is captured into FF_1 and then shifted to the next cells FF_1 , will be overwritten by the next data captured by ILS FF cells during the Capture-DR state in the next test pattern application cycle.

Fig. 16 summarizes the test process and instructions used in the signal-integrity mode. First, the seed is scanned into BSCs using the PRELOAD instruction. By loading the G-SITEST instruction into the instruction register, the new BSCs, which target signal integrity test

```

01: RESET
02: FOR ( $j=1$  to  $N_S$ )
03: {
04:   PRELOAD seed  $j$  // Initialize the PGBSCs
05:   G-SITEST //Pattern generation and application
06:   FOR each victim line
07:     Apply 3 UpdateDRs
08: }
09: O-SITEST //Reading out the integrity loss data

```

Fig. 16. Signal integrity test process.

for interconnects are set in signal-integrity mode. Update-IR makes the G-SITEST instruction operational and sets $SI = 1$ and $CE = 1$. Then, test patterns are generated by PGBSCs and applied to interconnects and, simultaneously, ILS captures the signals at the end of interconnects and detects violations, if any. After the test-application process, integrity-loss data is stored in ILS FF and must be read out. This is done using the O-SITEST instruction. The instruction first deactivates the ILS cell because during the scan-out operation, some new data will be scanned in and it may be applied to interconnects in the Update-DR controller state before the scan-out phase is complete. This may cause ILS to lose integrity-loss data stored in the previous cycle. Finally, the scanning out process is performed as explained in Section IV-B.

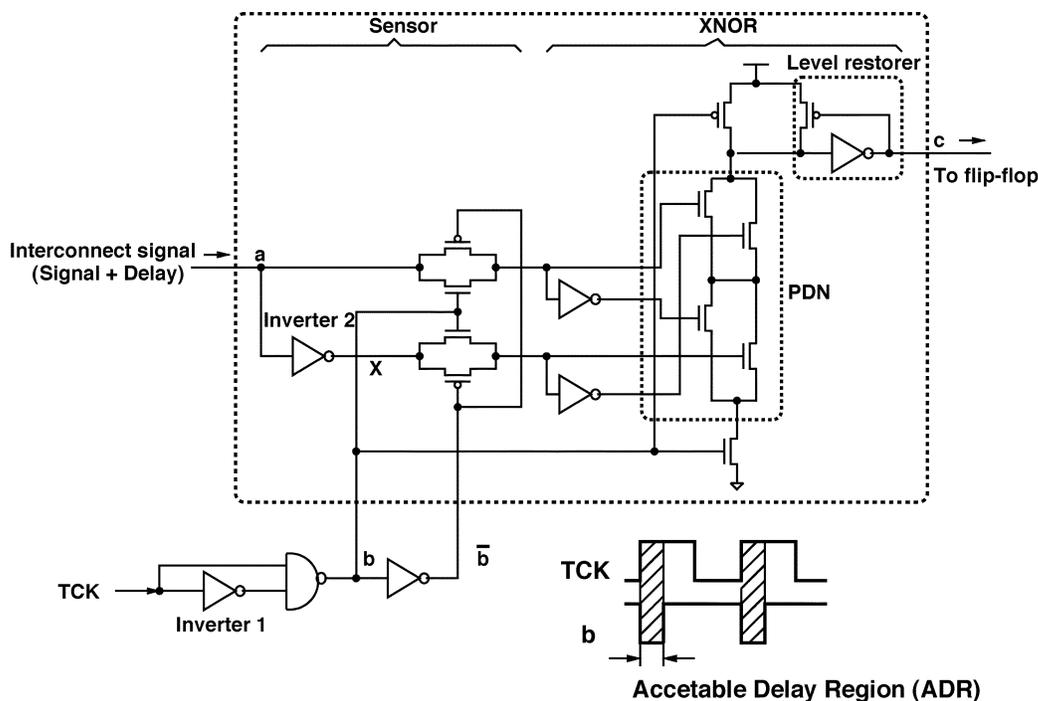


Fig. 17. ILS cell.

V. IMPLEMENTATION AND SIMULATION RESULTS

A. ILS Cell Used in Our Work

While any ILS sensor, such as those presented in [4], [17], and [19], can be used for the purpose of integrity-loss detection, for the purpose of simplicity, cost, and experimentation, we have developed our own ILS cell. In what follows, we briefly explain the functionality and characteristics of this cell.

Our ILS is a delay-violation sensor shown in Fig. 17. The cell consists of two parts, the sensor and the detector. The acceptable delay region (ADR) is defined as the time interval from the triggering clock edge during, in which all output transitions must occur. The test clock is used to create a window, which determines the acceptable skew region. Signal input a is in the acceptable delay period if its transition occurs during the period when b is at logic “0.” Any transition that occurs during the period when b is at logic “1” and is passed through the transmission gates to the detector XNOR gate. The XNOR gate is implemented using dynamic precharged logic. Output c becomes 1 when a signal transition occurs during $b = 1$ and remains unchanged until $b = 0$, the next precharge cycle. Output c is used to trigger a flip-flop.

Fig. 18 shows timing behavior of different input signals to the cell. Transmission gates are closed when signal $b = 0$ and their outputs hold to the previous values. Output of the ILS cell is determined by the timing of signals a and X . Three different cases for transition on signals a and X are illustrated as follows.

- 1) **Case 1:** Transition on signals a and X occurs during $b = 0$, i.e., inside the period t_{Inv1} . The transmission gate outputs change simultaneously and the XNOR gate output c is at logic “0.”
- 2) **Case 2:** Signal a makes a transition when $b = 0$, and X changes when $b = 1$. The transmission gate outputs have a delay difference of $(t_a + t_{Inv2}) - (t_{NAND} + t_{Inv1})$. Signal c changes to “1” when this delay difference is greater than or equal to the XNOR gate delay. Equation (2) determines the minimum delay of signal a from the clock edge detected by the cell, i.e., the ADR period.
- 3) **Case 3:** The transition on signals a and X occurs during $b = 1$, i.e., outside the period t_{Inv1} . The transmission gate outputs

have a delay difference equal to t_{Inv2} and the XNOR gate output c changes to logic “1.”

$$ADR = t_{a,\min} = t_{XNOR} + t_{Inv1} + t_{NAND} - t_{Inv2} \quad (2)$$

Equation (2) has two parameters t_{Inv1} and t_{Inv2} which can be adjusted to tune the cell to detect required delay. These two parameters are independent of each other. The minimum detectable delay ($t_{a,\min}$) can be decreased by increasing t_{Inv2} or decreasing t_{Inv1} . t_{Inv2} can be decreased until the duration is enough to precharge the dynamic logic. Table V shows the minimum delay detectable by changing the number of cascaded inverters with delay of t_{NOT} , each to form Inverter 1 and 2. We acknowledge that depending on the application and threshold for delay and noise, we may need to retune ILS cells for different designs. Fortunately, most of the sensors [4], [17], [20] are easily tunable. When needed, the delay of $Inv2$ can be tuned to control ADR for individual interconnects.

Fig. 19 shows the SPICE [32] simulation of the ILS cell, implemented using 0.18- μm technology, for two transitions of signal at input a . The first transition of the signal occurs at 0.2 ns when $b = 0$ and the output remains zero. The second transition occurs at 3.5 ns when $b = 1$ and exceeds the acceptable delay period and output c goes to 1 until b goes to zero. This delay sensor can detect any transition faults (e.g., due to crosstalk) that result in excessive delay. The pulse produced on c can be fed to a flip-flop to store delay occurrence for further readout/analysis.

B. Simulation Results

The enhanced BSC and architecture is implemented using Synopsis design analyzer [33]. The total area overhead for conventional BSC cell and the enhanced BSC cell (including ILS) is shown in Table VI. Enhanced cells are 28%–46% more expensive compared to the conventional one. This is negligible compared to the overall cost of boundary scan architecture (cells, controller, etc.). Moreover, to lower the cost,

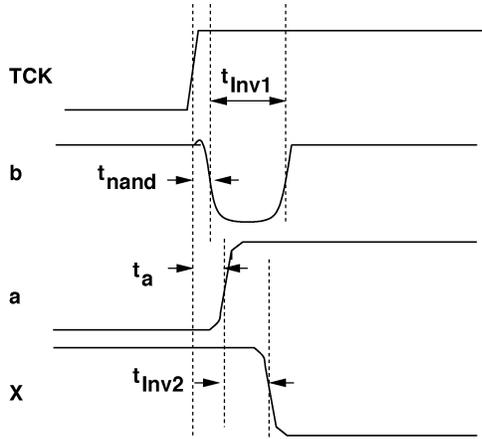


Fig. 18. ILS signals and their timing behavior.

TABLE V
MINIMUM DETECTABLE DELAY OF ILS CELL

Inverter1 (t_{Inv1}) [$\times 10t_{NOT}$]	Inverter2 (t_{Inv2}) [$\times t_{NOT}$]	Minimum Detectable Delay (ADR) [ps]
5	1	450
3	3	250
1	5	100

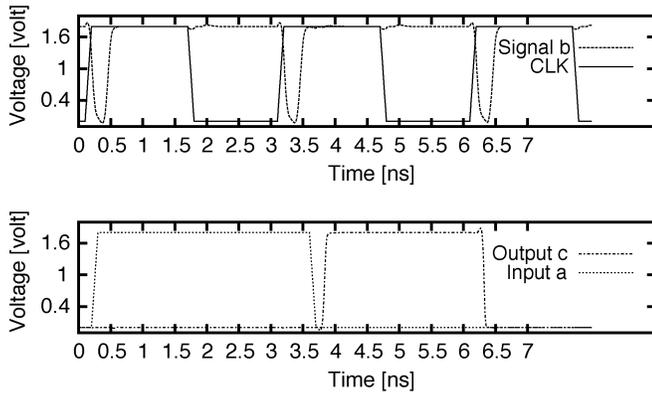


Fig. 19. SPICE simulation of the ILS cell.

the user still has the option of inserting/changing cells on only interconnects of interest (e.g., those that are susceptible to noise or those that feed glitch-/delay-sensitive circuits).

Table VII shows the comparison between using enhanced boundary scan to generate and apply MT patterns and conventional boundary scan to scan in and apply the same set of patterns. The application TR ratio is between 86% and 92% for various scenarios. Table VIII compares three methods described in Section III-B for different number of interconnects under test (n) and locality factor (k) assuming $q = 0$. As expected, the number of clocks required for Method 1 is significantly lower than Method 3. However, Method 3 provides much more information about the type and location of integrity faults. In Method 2, we have performed one scan-out operation per victim line. Essentially, Method 2 can be used to tradeoff test time versus accuracy.

Table IX compares the maximum noise voltage and skew between the MT and MA fault models for different k s. For $k = 2$, both methods show almost the same maximum noise and delay. The simulation for $k = 3, 4$ shows that a considerable increase in delay can be stimulated by the MT model compared to the MA model while the peak noise is slightly higher.

TABLE VI
COST ANALYSIS FOR BSCs

Test Architecture	Cost[NAND]		
	Sending	Observing	Bidirectional
Conventional Cells	26	26	78
Enhanced Cells	36	38	100
Area Increase %	38.5	46.2	28.2

TABLE VII
MT PATTERN-APPLICATION TIME

Methods	MT-Pattern Application Time [Cycle] ($p=0$)					
	$n=8$		$n=16$		$n=32$	
	$k=2$	$k=3$	$k=2$	$k=3$	$k=2$	$k=3$
N_{clk_EBS}	2560	17920	3840	25088	6400	39424
N_{clk_BS}	19200	150528	32000	250880	57600	451584
TR%	86.1%	88.5%	88.3%	90.3%	88.9%	91.8%

TABLE VIII
OBSERVATION-TEST TIME FOR THREE METHODS

Methods	Observation Test Time [Cycle] ($q=0$)					
	$n=8$		$n=16$		$n=32$	
	$k=2$	$k=3$	$k=2$	$k=3$	$k=2$	$k=3$
Method 1	8	8	16	16	32	32
Method 2	640	3584	1280	7168	2560	14336
Method 3	2560	14336	5120	28672	10240	57344

TABLE IX
MT AND MA COMPARISON

k	MT		MA	
	$V_{noise}[V]$	Delay[ps]	$V_{noise}[V]$	Delay[ps]
$k=2$	0.351	470.5	0.348	468.5
$k=3$	0.408	472.3	0.404	450.7
$k=4$	0.420	483.4	0.411	440.9

VI. SUMMARY

We proposed an enhanced boundary-scan architecture for testing signal integrity in SoC interconnects. The enhanced architecture is utilized to generate and apply MT patterns. The proposed MT fault model is a superset of the MA model and is much more capable of testing the capacitive and inductive couplings among interconnects. Our architecture detects integrity loss violations on interconnects based on the widely-used JTAG boundary-scan architecture. To do this, additional detector cell, modified scan cells, and minor modifications to the TAP controller to handle two new instructions are needed. The advantage of this proposed architecture is that it provides cost-effective solution for thorough testing of interconnects using the popular JTAG standard.

REFERENCES

- [1] A. Sinha, S. K. Gupta, and M. A. Breuer, "Validation and test issues related to noise induced by parasitic inductances of VLSI interconnects," *IEEE Trans. Adv. Packaging*, pp. 329–339, 2002.
- [2] X. Bai, S. Dey, and J. Rajski, "Self-test methodology for at-speed test of crosstalk in chip interconnects," in *Proc. Design Automation Conf.*, 2000, pp. 619–624.
- [3] L. Chen, X. Bai, and S. Dey, "Testing for interconnect crosstalk defects using on-chip embedded processor cores," in *Proc. Design Automation Conf.*, 2001, pp. 317–322.
- [4] A. Attarha and M. Nourani, "Testing interconnects for noise and skew in gigahertz SoCs," in *Proc. Int. Test Conf.*, 2001, pp. 305–314.
- [5] *Standard Test Access Port and Boundary-Scan Architecture*, IEEE Standard 1149.1-2001, 2001.

- [6] M. Bushnell and V. Agrawal, *Essentials of Electronics Testing*. Norwell, MA: Kluwer, 2000.
- [7] M. Cuvillo, S. Dey, X. Bai, and Y. Zhao, "Fault modeling and simulation for crosstalk in system-on-chip interconnects," in *Proc. Int. Conf. Computer-Aided Design*, 1999, pp. 297–303.
- [8] Y. Zhao and S. Dey, "Analysis of interconnect crosstalk defect coverage of test sets," in *Proc. Int. Test Conf.*, 2000, pp. 492–501.
- [9] N. S. Nagaraj, P. Balsara, and C. Cantrell, "Crosstalk noise verification in digital designs with interconnect process variations," in *Proc. Int. Conf. VLSI Design*, 2001, pp. 365–370.
- [10] W. Chen, S. Gupta, and M. Breuer, "Analytic models for crosstalk delay and pulse analysis under nonideal inputs," in *Proc. Int. Test Conf.*, 1997, pp. 809–818.
- [11] S. Naffziger, "Design methodologies for interconnects in GHz+ ICs," presented at the Int. Solid-State Conf., 1999.
- [12] Y. Cao *et al.*, "Effective on-chip inductance modeling for multiple signal lines and application to repeater insertion," *IEEE Trans. VLSI Syst.*, vol. 10, pp. 799–805, Dec. 2002.
- [13] W. Chen, S. Gupta, and M. Breuer, "Test generation for crosstalk-induced delay in integrated circuits," in *Proc. Int. Test Conf.*, 1999, pp. 191–200.
- [14] —, "Test generation in VLSI circuits for crosstalk noise," in *Proc. Intern. Test Conf.*, 1998, pp. 641–650.
- [15] M. Attarha and M. Nourani, "Test pattern generation for signal integrity faults on long interconnects," in *Proc. VLSI Test Symp.*, 2002, pp. 336–341.
- [16] Y. Zhao, L. Chen, and S. Dey, "On-line testing of multi-source noise-induced errors on the interconnects and buses of system-on-chips," in *Proc. Int. Test Conf.*, 2002, pp. 491–499.
- [17] S. Yang, C. Papachristou, and M. Tabib-Azar, "Improving bus test via I_{DDT} and boundary scan," in *Proc. Design Automation Conf.*, 2001, pp. 307–312.
- [18] International Technology Roadmap for Semiconductors (2001). Available: <http://public.itrs.net> [Online]
- [19] I. Rayane, J. Velasco-Medina, and M. Nicolaidis, "A digital BIST for operational amplifiers embedded in mixed-signal circuits," in *Proc. VLSI Test Symp. (VTS'99)*, 1999, pp. 304–310.
- [20] S. Tabatabaei and A. Ivanov, "An embedded core for sub-picosecond timing measurements," in *Proc. Int. Test Conf.*, Oct. 2002, pp. 129–137.
- [21] F. Caignet, S. Delmas-Bendhia, and E. Sicard, "The challenge of signal integrity in deep-submicrometer CMOS technology," *Proc. IEEE*, vol. 89, pp. 556–573, Apr. 2001.
- [22] M. H. Tehranipour, N. Ahmed, and M. Nourani, "Testing SoC interconnects for signal integrity using boundary scan," in *Proc. VLSI Test Symp.*, 2003, pp. 163–168.
- [23] C. Chiang and S. K. Gupta, "BIST TPG's for faults in board level interconnect via boundary scan," in *Proc. VLSI Test Symp.*, 1997, pp. 376–382.
- [24] K. Lofstrom, "Early capture for boundary scan timing measurement," in *Proc. Int. Test Conf.*, 1996, pp. 417–422.
- [25] IEEE P1500 Standard Available: <http://grouper.ieee.org/groups/1500/> [Online]
- [26] IEEE 1149.4 Standard Available: <http://grouper.ieee.org/groups/1149/4/> [Online]
- [27] L. Whetsel, "Proposal to simplify development of a mixed signal test standard," in *Proc. Int. Test Conf.*, 1996, pp. 400–409.
- [28] IEEE 1149.6 Working Group (2002). Available: <http://grouper.ieee.org/groups/1149/6/> [Online]
- [29] N. Ahmed, M. H. Tehranipour, and M. Nourani, "Extending JTAG for testing signal integrity in SoCs," in *Proc. Design, Automation, Test Eur.*, 2003, pp. 218–223.
- [30] M. H. Tehranipour, N. Ahmed, and M. Nourani, "Multiple transition model and enhanced boundary scan architecture to test interconnects for signal integrity," in *Proc. Int. Conf. Computer Design*, 2003.
- [31] OEA International, Inc. Available: <http://www.oea.com> [Online]
- [32] *TI-SPICE3 User's and Reference Manual*, 1994 Texas Instruments Inc., 1994.
- [33] *User Manual for SYNOPSIS Toolset Version 2003.12*, Synopsys, Inc., 2003.