Multiple Transition Model and Enhanced Boundary Scan Architecture to Test Interconnects for Signal Integrity

M. H. Tehranipour, N. Ahmed, M. Nourani

Center for Integrated Circuits & Systems

The University of Texas at Dallas, Richardson, TX 75083

{mht021000, nxa018600, nourani}@utdallas.edu

ABSTRACT

As the technology is shrinking toward 50 nm and the working frequency is going into multi gigahertz range, the effect of interconnects on functionality and performance of system-on-chips is becoming dominant. More specifically, distortion (integrity loss) of signals traveling on high-speed interconnects can no longer be ignored. In this paper, we propose a new fault model, called multiple transition, and its corresponding test pattern generation mechanism. We also extend the conventional boundary scan architecture to allow testing signal integrity in SoC interconnects. Our extended JTAG architecture collects and outputs the integrity loss information using the enhanced observation cells. The architecture fully complies with the JTAG standard and can be adopted by any SoC that is IEEE 1149.1 compliant.

I. INTRODUCTION

The number of cores and modules on a system-on-chip (SoC) is rapidly growing and therefore, the number of interconnects is intensively increased. Use of nanometer technology in SoCs magnifies the cross coupling effects between the interconnects. These effects are coupling capacitance and mutual inductance and they may affect the integrity of a signal by creating noise and delay. The noise effect can cause overshoot and ringing. Slowdown and performance degradation are the result of delay effect. If signal integrity losses (noise and delay) on an interconnect are between the defined safe margin, they are acceptable. Otherwise, they may cause an intermittent logic-error, performance degradation, shorter life time and reliability concern [1].

Process variations and manufacturing defects lead to noise and delay effects [2]. The goal of design for deep submicron (DSM) phase is to minimize noise and delay. However, due to its complexity it is impossible to check and fix all possible signal integrity problems during the DSM design validation/analysis phase. Process variations and manufacturing defects may lead to an unexpected increase in coupling capacitances and mutual inductances between interconnects. It results in loss of signal integrity as glitches and delay effects, which may intermittently cause logic-error and failure of the chip.

In recent years, there have been some research in the signal integrity area to model and test noise and delay [3] [4] [5] [7]. Regardless of the methods to detect integrity loss, we need a mechanism to manage the test session within or independent of other test sessions for a SoC. The signals carrying noise and delay at the end of the interconnect should be carefully tested. Therefore, at least appropriate sensor/detector cells are needed to test the signal. There are thousands of short, medium and long interconnects in an SoC and managing the test process of the interconnects is very important. One of the best choices is boundary scan test methodology, IEEE 1149.1 [6], that helps test designer to use the capability of accessing interconnects, applying test patterns and reading out the test results.

A. Prior Work

●*Signal Integrity Analysis:* Various signal integrity problems have been studied previously for radio frequency circuits and recently for high-speed deep-submicron VLSI chips. Maximum aggressor (MA) fault model [7] is one of the fault models proposed for crosstalk. Analysis of crosstalk is described in [8] [9]. Analysis of interconnect defects coverage of test sets is explained in [8]. They address the problem of evaluating the effectiveness of test sets to detect crosstalk defects in interconnections of deep submicron circuits. Several researchers have worked on test pattern generation for crosstalk noise and delay and signal integrity [10] [11] [12]. 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 [1].

●*Test methodologies:* Several self-test methodologies have been developed to test interconnects for signal integrity in high-speed SoCs. At-speed test of crosstalk in chip interconnects [3], testing interconnect crosstalk defects using on-chip processor [5], a BIST to test long interconnects for signal integrity [4] and using boundary scan and I_{DDT} for testing bus [13] are some of the proposed methods. The experiments show that short interconnects as well as long interconnects are susceptible to the integrity problem. Therefore, in near future methodologies for testing both short and long interconnects are required.

●*Integrity Loss Sensor (ILS) Cell:* Due to more and more concerns about signal integrity loss in gigahertz chips, researchers presented various on-chip sensors. Many of such integrity loss sensors (ILS) are amplifier-based circuits capable of detecting violation of voltages and delay thresholds. A BIST (built-in self-test) structure using D flip-flops has been proposed to detect the propagation delay deviation of operational amplifiers [16]. In [13] a built-in sensor is integrated within the system. The sensor is an on-chip current mirror converting the dissipated charges into the associated test time.

The authors in [17] presented a more expensive but more accurate circuits to measure jitter and skew in the range of few picoseconds. The authors in [18] 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. These cells sit physically near the end of an interconnect and samples the actual signal plus noise. To detect delay violation, an integrity loss sensor (called ILS) has been designed in [20] which is flexible and tunable. The acceptable delay region is defined as the time period from the triggering clock edge during which all output transitions must occur.

●*Boundary Scan Application:* Most of the early work in testing interconnect using boundary scan method focused on the development of deterministic test for interconnect faults at board level. BIST test pattern generators for board level interconnect testing and delay testing are proposed in [14] and [15], respectively. A modified boundary-scan cell using an additional level sensitive latch (called Early Capture Latch or ECL) was proposed in [15] for delay fault testing.

IEEE 1149.4 mixed-signal test bus standard [23] was purposed to allow accesses to the analog pins of a mixed-signal device. IEEE std. P1149.6 provides a solution for testing AC-coupled interconnects between integrated circuits on printed circuit boards and in systems [24].

A test methodology targeting bus interconnects defects using I_{DDT} and boundary scan has been presented in [13]. An extending JTAG

COMPUTER SOCIETY

is proposed to test SoC interconnects for signal integrity in [19] [20]. The maximum aggressor (MA) fault model was used in [19]. MA test patterns are generated and applied to the interconnects by modified boundary scan cells placed at the output of a core. The other modified boundary scan cells cited at the input of a core (at the end of interconnects) collect the integrity loss information. The drawback is that MA fault model is not included inductance.

We completed our work in [20] in which we assumed that test patterns are already generated based on a fault model including inductance. The test patterns are scanned by an external tester into the boundary scan cells and applied to the interconnects. The drawback is that the proposed method is time consuming because of scanning the test patterns through scan cells. In this paper we propose a new test pattern generation architecture which generates and applies test patterns almost at the speed of test clock (TCK). We also propose a new fault model which covers all possible transitions on the interconnects under test. This model also covers MA and some specific cases presented in [25].

B. Contribution and Paper Organization

Our main contribution is a new fault model, called multiple transition (MT), and its corresponding test mechanism using boundary scan application. MT essentially is a superset of MA patterns that covers all possible transitions on the interconnects known so far to stimulate integrity losses. An on-chip mechanism is proposed to extend JTAG standard to include testing interconnects for signal integrity based on MT model. Upon this extension delay violations occurring on the interconnects of high-speed SoCs can be tested using JTAG boundary scan architecture. Using new instructions in JTAG architecture the MT-patterns are generated and applied to the interconnects and the integrity test information is collected and sent out for final test analysis, reliability judgment and diagnosis.

The rest of the paper is organized as follows. Section II describes the MT fault model and test pattern generation. The enhanced boundary scan cells are proposed in Section III. Section IV explains the test architecture and experimental results. Finally, the concluding remarks are in Section V.

II. Multiple Transition: Fault Model and Pattern Generation

A. MT Fault Model

MA fault model [7] is a simplified model used by many researchers often for crosstalk analysis and testing of long interconnects. This model, shown in Figure 1, assumes the signal traveling on a line *V* (victim) may be affected by signals/transitions on other line(s) *A* (aggressor) in its neighborhood. The coupling can be generalized by a generic coupling component *Z*. The effect, in general, could be noise (causing ringing and functional error) and delay (causing performance to degrade). However, there is a controversy as to what patterns trigger the maximal integrity loss. Specifically, in the traditional MA model that takes only coupling *C* into account, all aggressors make a 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). When mutual inductance comes into play, some researchers showed that MA may not reflect the worst case and presented other ways (pseudorandom or deterministic) to generate test patterns to create maximal integrity loss [10] [11] [12].

As reported in [25], a chip failed when the nearest aggressor line changes in one direction and the other aggressors are in the opposite direction. This cannot be covered by MA fault model and some of the above pseudorandom and deterministic test set generated with different models. Exhaustive testing covers all situations but it is very time consuming because the number of test patterns is huge. Additionally,
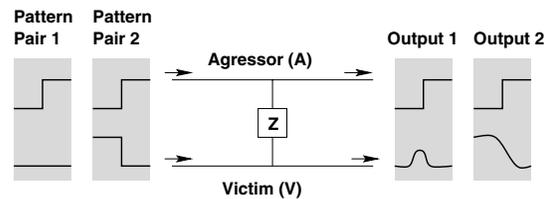

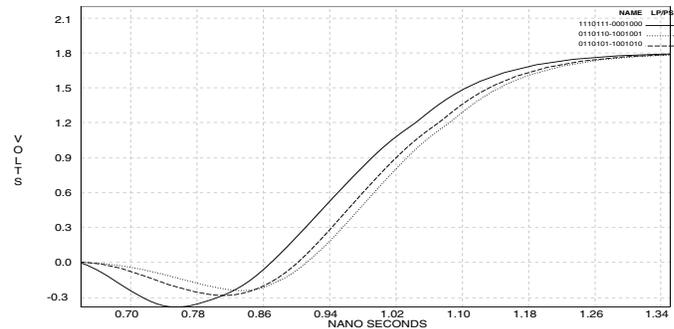Fig. 1. Signal integrity fault model.

Fig. 2. Comparison between MT and MA models.

exhaustive patterns include some cases that aggressors are in quiescent mode and obviously do not affect the victim line for noise and delay. Therefore, they need not to be considered in the model or pattern generators. Based on these observations and empirical evidence by researchers, we define a new fault model and test set which covers *all* transitions on victim and *multiple* transition on aggressors. We acknowledge that our *multiple transition* strategy refers to a specific pattern generation mechanism. However, throughout this paper, we call it *multiple transition fault model* for two reasons. First, we wanted a terminology similar to *maximum aggressor fault model* widely used [7] [8]. Second, because MT pattern set covers the high-speed interconnect coupling faults comprehensively.

Figure 2 shows the simulation results of two MT-patterns (*i*) 0110110 → 1001001 and (*ii*) 0110101 → 1001010 and one MA-pattern i.e. 1110111 → 0001000 applied to a seven interconnect system while the middle one is victim and the others are aggressors. Extraction and simulation are done by OEA tool (BUSAN) [22] and TISPICE [21], respectively. The interconnect model is distributed RLC and coupling capacitance and mutual inductances are considered between the lines using OEA tool for $0.18\mu m$ technology. As shown, MT-patterns create more delay compared to MA-pattern, ranging from 35 to 70ps depending on the buffer size. Therefore, MA-patterns may not be able to generate maximum noise/delay on the victim line when inductance is included. Another scenario reported in [25] which applies a test pattern not covered by MA failed a chip.

The main idea behind our proposed fault model called multiple transition (MT) is having single victim, limited number of aggressors, full transition on victim and multiple transition aggressors. The basis of MT is still the effect of coupling components as shown in Figure 1. In this case all the possible transitions on the victim and aggressors are applied, while only a subset of these transitions are applied in MA fault model. Another difference between MT and MA is that the aggressors in MA always change in the same direction. Briefly, MA-patterns (see Figure 3) is a subset of MT-patterns. MT is not an exhaustive model because it does not cover quiescent cases of aggressor lines. For example, assume that we have three interconnects and the middle one is victim. Figure 4 shows all possible transitions on the aggressors and victim line based on MT fault model. The test patterns for signal integrity are vector-pairs. As shown, when victim line is kept quiescent at '0' (column 1), four possible transitions on the aggressors are exam-

Fig. 3. MA fault model and test patterns.

Fig. 4. Transitions that MT and MA (shaded) models generate for a 3-line interconnect.

Fig. 5. Simulation results for different *k*.

Fig. 6. Resorted test pattern for a three interconnect system.

ined. For example, the first pair is '000' and '101' in which aggressors change from '0' to '1'. The MA-patterns (a subset of MT-patterns) are shaded in Figure 4.

Four cases are examined for each victim line when victim line is quiescent at '0', '1' or changes from '0' to '1' or '1' to '0'. As shown in Figure 4, the number of required test patterns to cover all possible transitions on the three interconnects is $4*2^{3-1}=16$ when the second 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. The number of test patterns for a group of *m* interconnects is $N_{Pattern} = m \cdot 4 \cdot 2^{m-1} = m \cdot 2^{m+1}$. When *m*, the total number of interconnects, increases the number of test patterns increases exponentially. Simulation show that in an interconnect system the lines which are far away from the victim cannot affect much on the victim line. Therefore, the number of lines (aggressor) after and before the victim line can be limited. We define *k* as locality factor that is determined empirically showing how far the effect of aggressor lines remain significant. Figure 5 shows the simulation results of different number of aggressors in the victim neighborhood while victim line is quiescent at 0. As shown, when the number of interconnects on either side of the victim increases the noise on the victim increases. The noise voltage difference between line k=3 and k=4 is $V_{noise}(k=4) - V_{noise}(k=3)=0.048v$ which for many systems can be assumed negligible. Therefore, k=3 is the locality factor in our simulation. We acknowledge that finding such locality factor is technology and application dependent. However, once a user based on application and accurate simulation provides it, the total number of pattern and time to test integrity faults will be significantly reduced.

B. Test Pattern Generator

Analysis of the MT fault model test vector-pair shows that in some transitions the value of the victim line should be fixed, while aggressor lines change. In some other transitions, both victim and aggressors lines change. It shows that in all cases the aggressor lines change from one value to another ('0' to '1' or '1' to '0') with every clock, while in some cases, victim line value changes with every two clocks. This important observation helps us to design a circuit to generate these test patterns. Figure 6 shows resorted test vectors shown in Figure 4. Each row shows that victim changes every two clocks and aggressors change every clock.

Figure 7 shows the required circuit for a three interconnect system. The victim and aggressor lines are selected by using select lines of
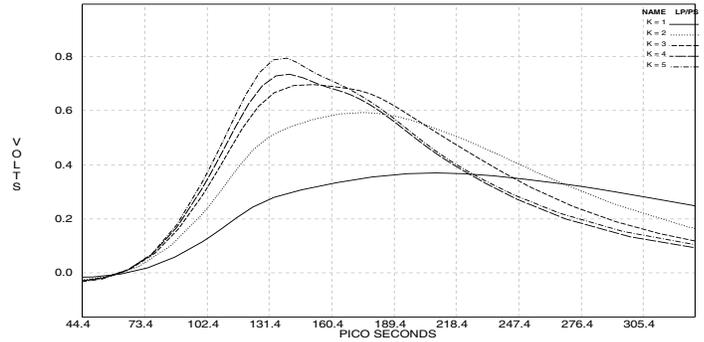
MUXs (s_1, s_2 and s_3). For example, when $s_1s_2s_3$='010', the second line is victim and others are aggressors. All four cases of victim line, $0 \rightarrow 0$ (quiescent at 0), $1 \rightarrow 1$ (quiescent at 1), $0 \rightarrow 1$ and $1 \rightarrow 0$, are generated by this circuit. When an interconnect is in victim mode, the related flip flop is clocked by $clk/2$ and the other aggressor flip flops are clocked by clk. It means that the content of the aggressor flip flops change every clock and the victim flip flop's content changes every two clocks.

The flip flops are initialized through I_1, I_2 and I_3. Assume that $I_1I_2I_3$='000' as a seed value for the circuit and $s_1s_2s_3$='010' (the second line is victim), the test patterns are shown in the second row of Figure 6. Four pairs of test patterns are generated with seed='000' that are ('000','101'), ('101','010'), ('010','111') and ('111','000'). For covering all possible transitions as shown in Figure 6, four seeds are required. In the above three interconnect system, the seeds are '000', '001', '100' and '101'. The total number of required seeds to cover all lines in victim mode in a three interconnect system is $3*4=12$. For a group of *m* interconnects, which $m = 2k+1$, the total number of seeds are $N_{Seed} = m \cdot 2^{m-1} = (2k+1) \cdot 2^{2k}$.

The MT fault model covers MA fault model. We consider two seeds for generating test vectors for MA fault model, i.e, '000' and '101'. In Figure 6, the shaded pattern shows the MA patterns which are generated based on these two seeds. It shows that after applying the first seed, '000', the generated test patterns cover P_{g_0}, d_f, and P_{g_1} faults. The generated test patterns after applying the second seed, '101', cover
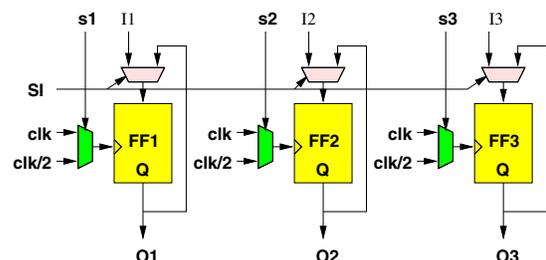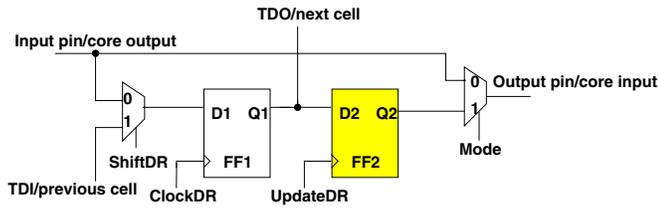

Fig. 7. Concept of MT test pattern generator.

Fig. 8.   A Standard Boundary Scan Cell.

d_r, N_{g_0} and N_{g_1}. Therefore, by such reordering only two seeds are sufficient for covering all 12 test patterns in the MA fault model.

III. ENHANCED BOUNDARY SCAN CELL

Boundary scan is a widely used test technique that requires boundary scan cells to be placed between each input or output pin and the internal core logic. The standard provides an efficient test methodology to test the core logic and the interconnects. Figure 8 shows a conventional standard boundary scan cell (*BSC*) with shift and update stages. *Mode* = 1 puts the cell in the test mode. The data is shifted through the shift register (*Shift-DR* state) during scan operation. Test patterns scanned into the boundary scan cells through the scan in port (*TDI*) are applied in parallel during the *Update-DR* state (*UpdateDR signal*). Circuit response is captured in parallel by the boundary scan cells connected between internal logic and output pins and is scanned out through the scan out port (*TDO*).

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

A. Pattern Generation BSC (PGBSC)

As mentioned before, a pair of test vectors are required to test interconnects for signal integrity. These patterns can be applied to the interconnects in a boundary scan architecture. For applying each pair, the first pattern is scanned into the conventional BSCs and then the second pattern is scanned into the BSCs. Using *UpdateDR*, they are applied onto the interconnects. Scanning and applying patterns in this way is very straightforward but needs a large number of clocks which increases the overall test time. We propose a hardware-based method for test pattern generation based on MT fault model. Test pattern generation is performed at the input side of the interconnects, that is the output side of a core which drives the interconnects. The new BSC that generate test patterns is called pattern generation BSC (PGBSC).

Boundary scan cell can be utilized to support the proposed circuit in Figure 7. FF_2 in boundary scan cell (see Figure 8) can be used as FFs in Figure 7. A T flip flop generates half of a clock, which *UpdateDR* plays the same role of *clk* in Figure 7. First, the initial values come from TDI and are sent into the FF_2. The select signals are also scanned through TDI and select victim and aggressor lines.

In addition to its normal mode, PGBSC should work in two new operational modes, victim and aggressor in signal integrity test mode. The PGBSC architecture is shown in Figure 9. The shaded components are those discussed in Figure 7. Only one extra control signal (SI) is needed for this architecture. This signal is generated by a new instruction, to be explained in Section 4. The PGBSC generates the required test patterns for covering the MT fault model. Table I shows the operation modes of the PGBSC. Depending on the select line of the mux attached to FF3, this architecture has three modes:
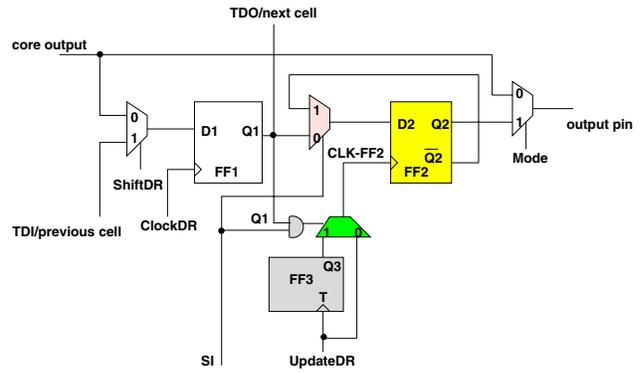

Fig. 9.   PGBSC design.

TABLE I
OPERATIONAL MODES OF THE PGBSC.

PGBSC Mode	Q_1	SI
Victim	1	1
Aggressor	0	1
Normal	x	0

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 the FF_2. By each *UpdateDR*, the complemented data is generated in $\overline{Q_2}$ and it is transferred to the output pin.
3) **Normal mode:** *UpdateDR* is selected. It is the normal mode of the PGBSC and *UpdateDR* is applied to the FF_2.

Figure 10 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 be '0', then $\overline{Q_2}$ is '1' and is applied to D_2 through the feedback. By every two *UpdateDR*, the content of the FF_2 is complemented. On the other hand, if PGBSC is in aggressor mode, *CLK-FF*$_2$ has the same frequency of *UpdateDR* and by each *UpdateDR* content of FF_2 is complemented. As shown in Figure 9, $\overline{Q_2}$ is complemented by each *CLK-FF*$_2$ while Q_2 is applied to output (i.e. interconnect).

Each interconnect acts as victim and aggressor. Therefore, in the test session each time the victim interconnect should be specified. After performing the test process on a victim, 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 the interconnects is possible. Because, in each step of test we need at most k lines as aggressor before and after the victim line. We use an encoded data to specify the victim which is called *victim-select data*.

Table II shows the scanned in victim-select data for a n-bit interconnect system, to be stored in FF1 while the locality factor k=2. After specifying the victim, the test vectors are generated by the 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 II, when we scan in the '100100...100' to n PGBSCs, the first line
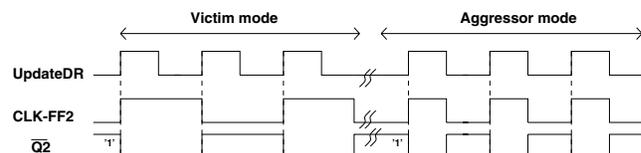

Fig. 10.   The operation of the PGBSC.

TABLE II

ENCODED DATA FOR VICTIM LINE IN A n INTERCONNECT SYSTEM.

Victim location	Victim-select data
VAAVAA...VAA	100100...100
AVAAVAA...VA	0100100...10
AAVAAVAA...V	00100100...1

```
01: for (k= 1 to N_Seed)
02: {
03:      Scan seed k into FF2
04:      Activate signal integrity test mode (SI=1)
05:      Scan the first victim-select data
06:      For (i=1 to k)   //Total # of shifts for victim-select data
07:      {
08:          Apply 4 UpdateDRs.        // Pattern generation
09:          Shift one '0' into FF1        // Selecting new victim
10:      }
11: }
```

Fig. 11.   Test pattern generation procedure using PGBSC.

is victim and the next two lines are aggressors for first interconnect and the fourth one. Therefore, $[n/(k+1)]=[n/3]$ victims are tested simultaneously. As shown, with one clock the victims locations change '0100100...10'. Only two rotates are enough for covering the whole interconnects to act as victim and aggressor for each initial value.

The generic behavior of test pattern generation and applying procedure is shown in Figure 11. This behavior will be executed by a combination of automatic test equipment (ATE) and TAP controller. The first seed is applied to the new BSCs as an initial value into FF_2. The victim and aggressors are selected with 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 the interconnects, a new victim is selected and the process will be repeated with the same initial value. Note carefully that at the end of the first step again the same seed would be in FF_2. The same process will be repeated for all initial values.

Equation 1 shows the number of required clocks to generate and apply MT test patterns by using the enhanced boundary scan architecture. This formula is extracted based on the test pattern generation procedure shown in Figure 11. The number of required clocks to scan and apply MT test patterns by conventional boundary scan architecture through TDI is shown in Equation 2. The test application time reduction ratio (TR) is shown in Equation 3.

$$N_{TCK}(Enhanced\_BS) = N_{Seed}(2n+8k) \qquad (1)$$

$$N_{TCK}(Conv.\_BS) = m \cdot N_{Pattern}(n+4) \qquad (2)$$

$$TR\% = \frac{N_{TCK}(Conv.\_BS) - N_{TCK}(Enhanced\_BS)}{N_{TCK}(Conv.\_BS)} \cdot 100\% \qquad (3)$$

While we keep the MT model, our test methodology does not depend on the test patterns. Any other test patterns (pseudorandom or deterministic) generated with other models can be applied by the external tester through the new boundary scan cells in the conventional mode of the boundary scan architecture.

B. Observation BSC (OBSC)

In [20], we proposed a new BSC at the receiving side of the interconnects which employs the ILS cell. Figure 12 shows the new BSC named observation BSC (OBSC). As shown, ILS is added to the receiving side cells. The ILS captures signals with noise and delay at the end of the interconnect. If it receives a signal with integrity problem (eg. delay violation) it shows a $0 \rightarrow 1 \rightarrow 0$ pulse at the output and the FF is set to '1'. If SI=1, the signal F is selected. The captured integrity data is scanned out every *Shift-DR* state through the scan chain for final evaluation. When SI=0, the ILS is isolated and each OBSC acts as a standard BSC.
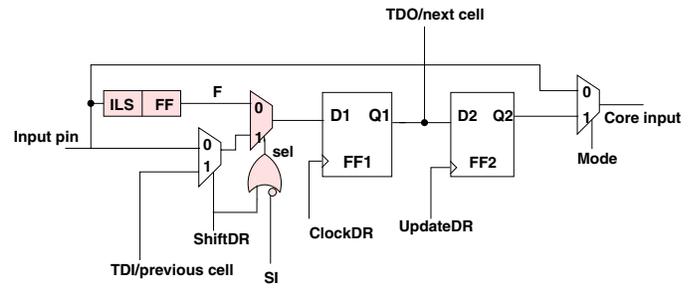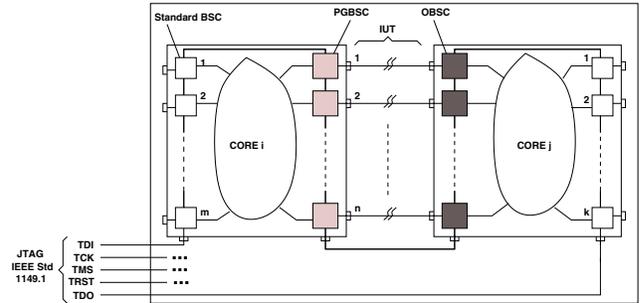

Fig. 12.   Observation BSC.

Fig. 13.   Test Architecture

The observation of the signal integrity information can be performed in three methods are: 1) *Method 1:* Reading out after applying each test patterns, 2) *Method 2:* Reading out after applying a subset of test patterns, and 3) *Method 3:* Reading out once after applying the entire test patterns. Selecting the methods depends on the acceptable time overhead. The first method is very time consuming, but it shows maximum integrity information as to which pattern caused the violation on each interconnect. The third method is very fast with minimum integrity information because the obtained information shows only the type of fault but not which pattern or which set of patterns have caused the integrity fault. Method 2 can help user to do a tradeoff between test time and accuracy.

IV. BOUNDARY SCAN IMPLEMENTATION

A. Test Architecture

Figure 13 shows the overall test architecture for a small SoC. The JTAG inputs (TDI, TCK, TMS, $TRST$ and TDO) are still used without any modification. A new instruction is defined to be used for signal integrity test for reading out the test results. As shown in Figure 13, the sending side cells of the IUTs are changed to PGBSCs and the receiving end cells to OBSCs. For bidirectional interconnects, the PGBSC and OBSC cell are used for both sides as shown between $Core_j$ and $Core_l$. The other cells are standard BSCs which are present in the scan chain during the signal integrity test mode. The integrity information, after applying one, some or all the patterns the signal integrity information shown by F is scanned out to determine which interconnect has a problem.

Two new instructions are used for test pattern generation and integrity test information reading out. G-SITEST instruction is used for test pattern generation using the enhanced architecture and allows testing interconnects between the chips in an SoC. O-SITEST instruction is similar to the EXTEST instruction with an additional control signal, *SI* activated. See more details of the instructions in [19].

B. Simulation Results

The enhanced boundary scan cell and architecture is implemented by Synopsys synthesizer [26]. The total area overhead for conventional BSA cell and enhanced BSA cell (ILS) is shown in Table III.

TABLE III

COST ANALYSIS FOR BOUNDARY SCAN CELLS.

Test Architecture	Cost[NAND]		
	Sending	Observing	Bidirectional
Conventional Cells	26	26	78
Enhanced Cells	36	38	100

TABLE IV

MT-PATTERN APPLICATION TIME.

Method	MT-Pattern Application Time [Cycle]					
	n=8		n=16		n=32	
	k=2	k=3	k=2	k=3	k=2	k=3
N_{TCK} Enhanced BS	2560	17920	3840	25088	6400	39424
N_{TCK} Conv. BS	19200	150528	32000	250880	57600	451584
TR%	86.1%	88.5%	88.3%	90.3%	88.9%	91.8%

The enhanced cells are almost 40% more expensive compared to the conventional one. Considering the overall cost of boundary scan architecture (cells, controller, etc.) additional overhead of components is still negligible (less than 5%).

Table IV shows the comparison between using enhanced boundary scan to generate and apply MT-patterns and conventional boundary scan to scan in and apply the MT-patterns. As shown, the application time reduction ratio is between 86 to 92%.

Table V shows a comparison between three methods described in Subsection III-B for different number of interconnects under test *n* and locality factor *k*. The table shows that the number of clocks required for methods 3 is significantly lower than method 1. However, method 1 provides much information about type and location of the integrity faults. In method 2, we have performed one scan-out operation per victim line. Method 2 can be used to tradeoff test time versus accuracy.

Table VI shows maximum noise voltage and delay comparison between different *k's* for MT and MA fault model. As shown, for *k*=2, MT and MA show almost the same maximum noise and delay. The simulation for *k*=3,4 shows more delay for MT model compared to MA model while the noise is almost the same for these two models.

V. CONCLUSION

We target enhancing the IEEE 1149.1 JTAG boundary scan standard that is the most widely used test methodology in industry. The enhancement allows testing SoC interconnects for integrity loss. The importance of distorted signals in gigahertz systems justifies the cost

TABLE V

OBSERVATION TEST TIME COMPARISON.

Methods	Observation Test Time [Cycle]					
	n=8		n=16		n=32	
	k=2	k=3	k=2	k=3	k=2	k=3
Method 1	2560	14336	5120	28672	10240	57344
Method 2	16	16	32	32	64	64
Method 3	8	8	16	16	32	32

TABLE VI

SIMULATION RESULTS.

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	0.415	440.9

overhead for the integrity loss sensors. The proposed MT fault model and patterns cover all possible transitions on the interconnects to stimulate integrity loss. MT patterns are generated and applied using enhanced boundary scan cells.

REFERENCES

[1] L. Green, "Simulation, Modeling and Understanding the Importance of Signal Integrity," *IEEE Circuit and Devices Magazine*, pp. 7-10, Nov. 1999.

[2] S. Natarajan, M. Breuer, S.K. Gupta, "Process variations and their impact on circuit operation," in Proc. *IEEE International Symposium*, pp. 73-81, 1998.

[3] X. Bai, S. Dey and J. Rajski, "Self-Test Methodology for At-Speed Test of Crosstalk in Chip Interconnects," in Proc. *Design Automation Conf. (DAC'00)*, pp. 619-624, 2000.

[4] M. Nourani and A. Attarha, "Built-In Self-Test for Signal Integrity," in Proc. *Design Automation Conf. (DAC'01)*, pp. 792-797, June 2001.

[5] L. Chen, X. Bai and S. Dey, "Testing for Interconnect Crosstalk Defects Using On-Chip Embedded Processor Cores," in Proc. *Design Automation Conf. (DAC'01)*, pp. 317-322, 2001.

[6] IEEE Standard 1149.1-2001, *"Standard Test Access Port and Boundary-Scan Architecture"*, IEEE Standards Board, 2001.

[7] M. Cuviello, S. Dey, X. Bai and Y. Zhao, "Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects," in Proc. *Int. Conf. on Computer Aided Design (ICCAD'99)*, pp. 297-303, 1999.

[8] Y. Zhao and S. Dey, "Analysis of Interconnect Crosstalk Defect Coverage of Test Sets," in Proc. *Int. Test Conf. (ITC'00)*, pp. 492-501, 2000.

[9] W. Chen, S. Gupta and M. Breuer, "Analytic Models for Crosstalk Delay and Pulse Analysis Under Non-Ideal Inputs," in Proc. *Int. Test Conf. (ITC'97)*, pp. 809-818, 1997.

[10] W. Chen, S. Gupta and M. Breuer, "Test Generation for Crosstalk-Induced Delay in Integrated Circuits," in Proc. *Int. Test Conf. (ITC'99)*, pp. 191-200, 1999.

[11] W. Chen, S. Gupta and M. Breuer, "Test Generation in VLSI Circuits for Crosstalk Noise," in Proc. *Int. Test Conf. (ITC'98)*, pp. 641-650, 1998.

[12] Attarha and M. Nourani, "Test Pattern Generation for signal Integrity Faults on Long Interconnects," in Proc. *VLSI Test Symp. (VTS'02)*, pp. 336-341, 2002.

[13] S. Yang, C. Papachristou, and M. Tabib-Azar, "Improving Bus Test Via I_{DDT} and Boundary Scan," in Proc. *Design Automation Conf. (DAC'01)*, pp. 307-312, 2001.

[14] C. Chiang and S. K. Gupta, "BIST TPGs for Faults in Board Level Interconnect via Boundary Scan", in Proc. *VLSI Test Symposium (VTS'97)*, pp. 417-422, 1997.

[15] K. Lofstrom, "Early Capture for Boundary Scan Timing Measurement", in Proc. *Int. Test Conf. (ITC'96)*, pp. 417-422, 1996.

[16] 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)*, pp. 304-310, 1999.

[17] S. Tabatabaei and A. Ivanov, "An Embedded Core for Sub-Picosecond Timing Measurements," in Proc. *Int. Test Conf. (ITC'02)*, pp. 129-137, Oct. 2002.

[18] F. Caignet, S. Delmas-Bendhia and E. Sicard, "The Challenge of Signal Integrity in Deep-Submicrometer CMOS Technology," in Proc. of the IEEE, vol. 89, no. 4, pp. 556-573, April 2001.

[19] N. Ahmed, M. H. Tehranipour and M. Nourani, "Extending JTAG for Testing Signal Integrity in SoCs," in Proc. *Design, Automation and Test in Europe (DATE'03)*, pp. 218-223, 2003.

[20] M. H. Tehranipour, N. Ahmed and M. Nourani, "Testing SoC Interconnects for Signal Integrity Using Boundary Scan," in Proc. *VLSI Test Symposium (VTS'03)*, pp. 158-163 , 2003.

[21] TI-SPICE3 User's and reference manual, *1994 Texas Instrument Incorporation*, 1994.

[22] International OEA, Inc., http://www.oea.com.

[23] IEEE 1149.4 standard, *http://grouper.ieee.org/groups/1149/4/*.

[24] IEEE 1149.6 Working Group, *http://grouper.ieee.org/groups/1149/6/*, 2002.

[25] S. Naffziger, "Design Methodologies for Interconnects in GHz+ ICs," Tutorial Lecture in *Int. Solid-State Conf.*, 1999.

[26] Synopsys Design Analyzer, "User Manuals for SYNOPSYS Toolset Version 2000.05-1," Synopsys, Inc., 2000.

IEEE COMPUTER SOCIETY