Testing SoC Interconnects for Signal Integrity Using Boundary Scan

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 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. We also propose a simple yet efficient compression scheme that can be employed by an ATE to minimize the scan-in delivery time.

Keywords: *Boundary Scan, Data Compression, Integrity Loss Sensor, Signal Integrity, System-on-Chip Interconnects.*

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 as 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. Since it is impossible to predict the occurrence of defects causing noise and delay, signal integrity loss testing especially on long interconnects has became essential to ensure error free operation of the chip and must be addressed in manufacturing testing.

In recent years, there have been some research in the signal integrity area to test noise and delay [3] [4] [5]. 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 that helps test designer to use the capability of accessing interconnects, applying test patterns and reading out the test results. Boundary scan test methodology was initially introduced to facilitate the testing of complex PC boards. The IEEE 1149.1 boundary scan test standard [6] known as JTAG has been widely accepted in the test community. The standard, nevertheless, provides excellent testing features with less complexity but was not intended to address high-speed testing, delay testing or signal integrity loss. The standard provides testing of core logic and the interconnects between them. Interconnects can be tested for stuck-at, open and short faults. As an efficient solution, we propose integration of integrity loss detectors with the popular JTAG architecture to make it practical, economical and easy to adopt. Our test architecture is an on-chip mechanism to extend JTAG standard to include testing interconnects for signal integrity. Upon this extension noise and skew violations occurring on the interconnects of high-speed SoCs can be tested using JTAG boundary scan architecture. The 5-pin standard interface to the ATE remain unchanged. However, the TAP controller will change to accommodate for at least one new instruction for reading out the test results. When needed test pattern compression/decompression is possible to save delivery time.

A. Prior Work

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 [4] [10] [12]. Authors in [11] and [10] proposed test pattern generation for crosstalk-induced noise and delay, respectively. A test pattern generation algorithm based on considering the effect of inputs and parasitic RLC elements of the interconnect has been proposed in [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].

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.

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

Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)
1093-0167/03 $17.00 © 2003 IEEE

IEEE COMPUTER SOCIETY

sensitive latch (called Early Capture Latch or ECL) was proposed in [15] for delay fault testing. The motive was to latch the data at the core input pins as soon as the output cells are updated for delay analysis and to capture the input pin data in the capture state. An additional control circuitry is designed in [16], Early Capture Control Register, to control the relative timing between the update in output cells and the falling edge of Early Capture. The area overhead of the special control circuitry is a drawback of this method.

There are two ways to send test patterns on the interconnects. First, conventional scan method which sends test patterns one by one serially. The second method is compressing test patterns and scanning them in then decompress them on chip. There are thousands interconnects in large SoCs and using conventional method is very time consuming. Researches showed that compressing the test patterns will considerably reduce the test application time. There are several compression techniques for scan based testing [17], [18], and [19]. Test data compression using don't cares [17], test data compression using Golomb codes for SoCs [18] and Finding the minimum compacted test sequence without considering the don't cares for continuous scan [19] are some of the proposed techniques.

B. Contribution and Paper Organization

Our main contribution is an on-chip mechanism to extend JTAG standard to include testing interconnects for signal integrity. Upon this extension delay violations occurring on the interconnects of high-speed SoCs can be tested using JTAG boundary scan architecture. Special cells (observation boundary scan cell or OBSC) to monitor signals received from the system interconnect are incorporated within the boundary scan cell which record the occurrence of signals entering the vulnerable region over a period of operation. Using a new instruction in JTAG architecture the integrity test information is sent out for final test analysis, reliability judgment and diagnosis. To show another flexibility of our approach, we also propose a technique to compact the test patterns by exploiting the don't cares in test patterns often used in integrity loss testing.

The rest of the paper is organized as follows. Section II reviews the signal integrity fault model and briefly discusses the ILS cell. The enhanced boundary scan cell is proposed in Section III. Section IV explains the test architecture to send test patterns and capture and read out the signal integrity information. Our test pattern compression technique is summarized in Section V. The experimental results are discussed in Section VI. Finally, the concluding remarks are in Section VII.

II. Background

A. Integrity Loss on Interconnects

Many researchers have recognized that signal integrity loss through interconnects is critical for multi-gigahertz highly complex SoCs. They often take ad-hoc approaches during design and test to alleviate the problems. From our point of view, integrity loss (also called integrity fault in this paper) happens when the the voltage distortion (noise) and delay violations (skew) go beyond an acceptable threshold. The threshold depends on the technology used in fabrication. Such violations occur due to many unpredictable reasons including: 1) process variations that cause parasitic values (e.g. transistor dimensions, transconductance, threshold voltage, values of parasitic R/L/C, etc.) [20] [21], 2) transmission line effects (e.g. crosstalk, overshoot, reflection, electro-magnetic interference, etc.) [22], 3) coupling effects (e.g. coupling capacitors and mutual inductances) among interconnects which is not easy to analyze and are subject to change during fabrication [23] [24] and 4) ground bounce due to simultaneous switchings in SoC causing change in noise margins [25].

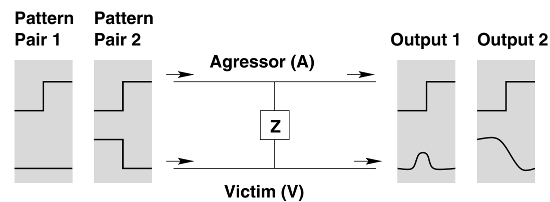

Fig. 1. Signal integrity fault model.

B. Integrity Fault Model

The most widely-used model is the maximum aggressor (MA) fault model [7]. This is a simplified model used by many researchers often for crosstalk analysis and testing on 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).

We use the same model in our work. However, we need to emphasis that 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 while the victim line is kept quiescent (for maximal ringing) or makes an opposite transition (for maximal delay). When mutual inductance comes to play, some researchers presented other ways (pseudorandom or deterministic) to generate test patterns to create maximal integrity loss [12] [11] [10]. While we keep the MA model, our test methodology does not depend on the test patterns. In this work we assume the test patterns are determined a priori and we show how they can be efficiently fed to the interconnect through an enhance JTAG architecture.

C. Integrity Loss Sensor (ILS) Cell

Due to more and more concerns about signal integrity loss in gigahertz chips, researchers presented variations of 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 [26]. During test mode, the Op Amp under test is placed in a voltage follower configuration in order to detect its slew-rate deviation, or in a comparator configuration in order to detect its signal propagation delay deviation. A test methodology targeting bus interconnects defects using I_{DDT} and boundary scan has been presented in [13]. In this work 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 work presented in [4] offers inexpensive 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. Each time the noise or skew goes above an acceptable limit the cells generate a $1 \rightarrow 0$ transition which can be stored in a flipflop for further analysis. The authors in [27] presented a more expensive but more accurate circuits to measure jitter and skew in the range of few picoseconds. This circuit, called EDTC, samples signals in non-intrusive way and sends out the test information through its low speed serial information. When the cost is not a concern, the concept of accurate signal monitoring has been followed up by researchers even through idea of on-chip oscilloscope [28]. The authors presented a sample and hold circuit that probes the voltage directly within the interconnects. While expensive, calibration and waveform measurements and even reconstruction all are possible.

IEEE COMPUTER SOCIETY

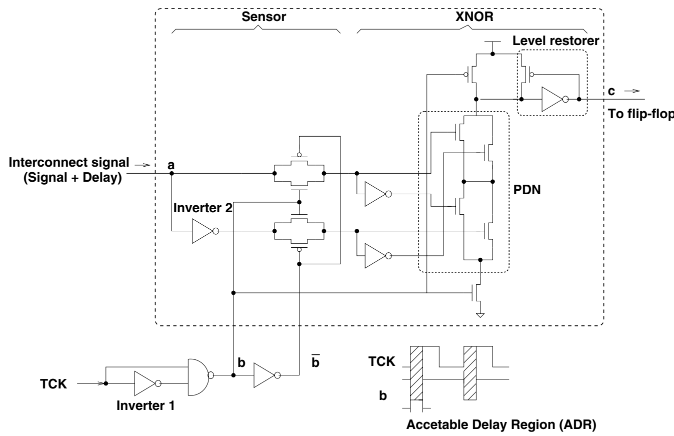

Fig. 2. Delay Violation Sensor.

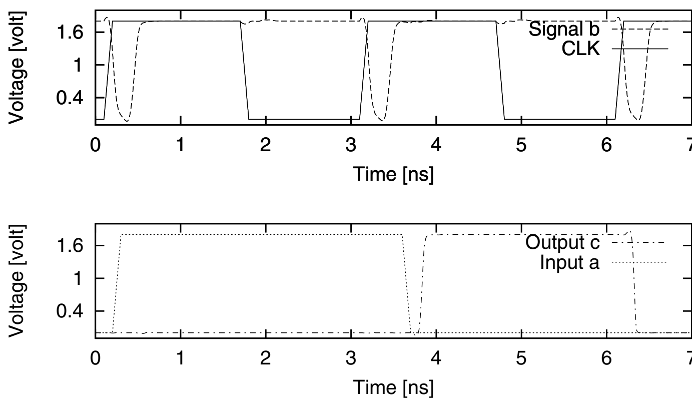

Fig. 3. Spice Simulation of the Delay Cell.

D. ILS Cell Used in Our Work

While any ILS sensor 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 explain the circuit and functionality of this cell briefly. The details of this cell is beyond the scope of this paper.

Our ILS is a delay violation sensor shown in Figure 2. The acceptable delay region (*ADR*) is defined as the time period from the triggering clock edge during which all output transitions must occur. The test clock is used to create a window which determines the acceptable skew region. The 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 logic '1' is passed through the transmission gates to the XNOR gate. It is implemented using dynamic precharged logic. The $Inverter1$ is tuned for desirable delay range. The output $c = 1$ when a signal transition occurs during $b = 1$ and remains unchanged till $b = 0$, the next precharge cycle. The output is used to trigger a flip-flop.

Figure 3 shows the SPICE [29] simulation of the 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 the output $c = 1$ till b goes to zero. The delay sensor also detects transition faults due to crosstalk. The pulse can be fed to a flipflop to store delay occurrence for further readout/analysis.

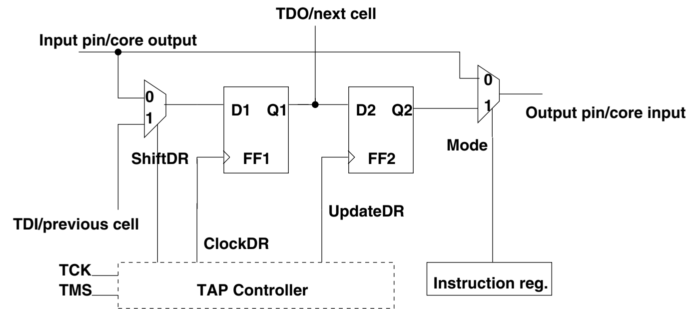

Fig. 4. A Standard Boundary Scan Cell.

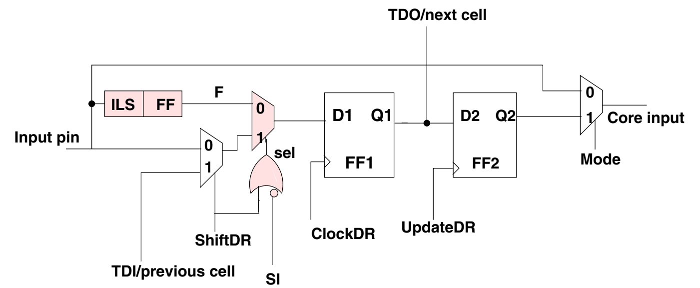

Fig. 5. Observation BSC.

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 4 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 *"EXTEST"* 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 a new cell and an instruction 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. Observation BSC (OBSC)

We propose a new BSC at the receiving side of the interconnects which employs the ILS cell. Figure 5 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 an interconnect. If it receives a signal with integrity problem (eg. delay violation) it shows a pulse at its output and the FF is set to '1'. The OBSC operates in two modes:

1) **Integrity mode (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.
2) **Normal mode (SI=0):** In this mode, the ILS is isolated and each OBSC acts as a standard BSC.

Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)
1093-0167/03 $17.00 © 2003 IEEE
IEEE COMPUTER SOCIETY

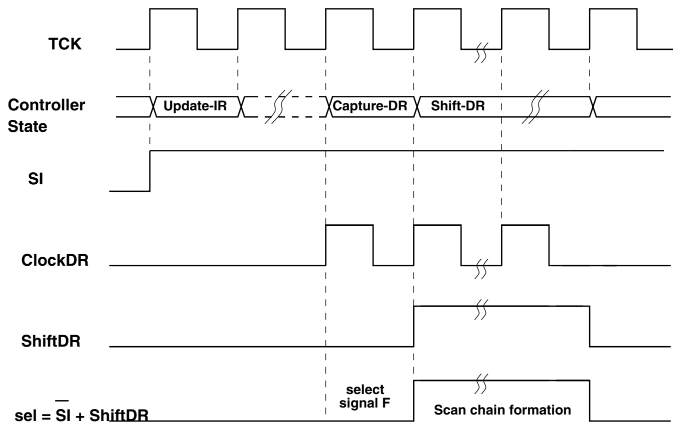

Fig. 6. Operation of observation BSC.

TABLE I
TRUTH TABLE OF SIGNAL *sel*

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

In the scan out process, we need to capture the output F into FF1. In this case, *sel* should be zero. Therefore *SI* and *ShiftDR* should be one and zero, respectively. When the scanning out process is started, D_1 is transferred to Q_1 to be used as a TDI for the next cell. The ILS flip-flop is reset after the signal integrity information is captured into FF1. After sending the value of F to the Q_1, the scan chain must be formed. In this case, during the *Shift-DR* state the TDI input must be connected to the FF1. Therefore, the ILS path should be isolated by *sel*='1' (*SI*='1' and *ShiftDR*='1'). As shown in Figure 5, $\overline{SI}$ and *ShiftDR* are ORed together for selecting and transferring the signal F to D_1 and the making of the scan chain to scan out. Figure 6 shows the dependency of *sel* to the *SI* and *ShiftDR*. As shown, in *Capture-DR* state, signal F is selected and then in *Shift-DR* state scan chain is formed and data is scanned out depending on how many wires are under test. Table I shows the truth table of signal *sel*. There is only one additional control signal (i.e. *SI*) which is generated by a new instruction, to be explained in Section IV.

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 on each interconnect. The third method is very fast with minimum integrity information because the obtained information shows only which pattern or which set of patterns have caused the integrity fault and not the type of fault. Method 2 can help user to do a tradeoff between test time and accuracy.

IV. TEST ARCHITECTURE

Figure 7 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 7, only the cells at the receiving end of each interconnect is changed to OBSCs. For bidirectional interconnects, the OBSC cell is 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

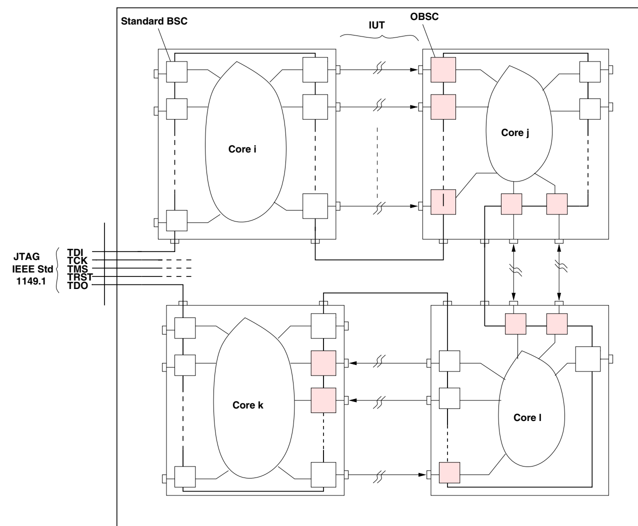

Fig. 7. Test Architecture

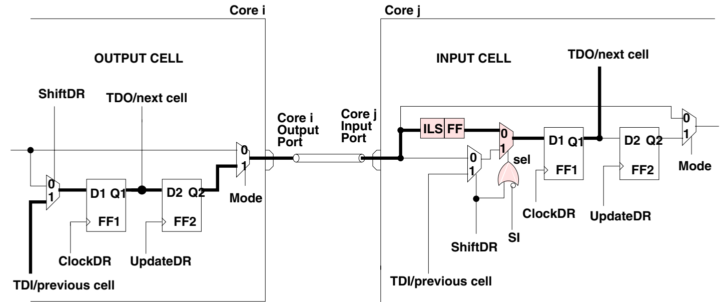

Fig. 8. Test data flow for EX-SITEST instruction.

integrity test mode. The ILS acts independently and no special control circuitry is required to control the timing of this cell. 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.

A. EX-SITEST Instruction

We propose to add a new instruction, i.e. *EX-SITEST*, to the IEEE 1149.1 instruction set for our new test architecture. This instruction is similar to the EXTEST instruction with an additional control signal, *SI* activated. In the *Update-IR* state, the instruction is decoded and ($SI = 1$) is generated. The output cells act as standard BSCs and the input cells act as OBSCs. The signal F is captured during the *Capture-DR* state and shifted out every clock cycle during the *Shift-DR* state. In this case, TAP controller sates will not change. Some changes are required in instruction decoding. Figure 8 shows the data flow of the EX-SITEST instruction between the cores.

B. Test Process

The tap controller IR is loaded with EX-SITEST instruction. Then, all test patterns are applied to the interconnects and simultaneously ILS cells capture the signals at the end of interconnects and detect the violation if any. After test application process, the stored results in the ILS cell FFs must be read. One of the three methods mentioned in Section III-A can be used for the observation process. For example, using method 3, all the test patterns are applied and then the integrity information is read out once.

Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)
1093-0167/03 $17.00 © 2003 IEEE
IEEE COMPUTER SOCIETY

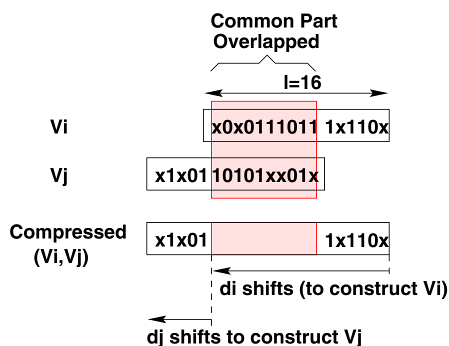

Fig. 9. Basic compression of two vectors (V_i, V_j).

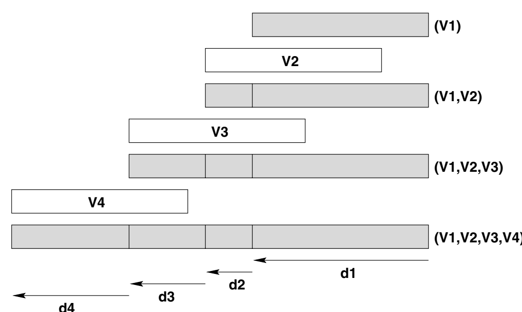

Fig. 10. The constructive compression technique.

TABLE II

COST ANALYSIS FOR BOUNDARY SCAN CELLS.

Test Architecture	*Cost[NAND]*		
	Sending	*Observing*	*Bidirectional*
Conventional Cell	26	26	78
ILS Cell	26	45	97

V. TEST DATA COMPRESSION

In conventional boundary scan architecture (BSA), test patterns are scanned in one-by-one and applied to the interconnects. For example in a n-bit interconnect using the maximum aggressor (MA) fault model, 12 test patterns are applied to each victim line and $12n$ clocks are required to apply the test patterns on only one victim line. With rotating victim line among n interconnects, the overall number of clock (test application time) is $12n^2$. Of course, MA is a simplistic model. Using more sophisticated models [12] or having large number of interconnects in an SoC results in huge number of test patterns and thus, the compression becomes a necessity. In this section we explain our simple yet efficient compression technique for the enhanced boundary scan architecture. Due to lack of space, we explain it briefly to show the flexibility of our enhanced JTAG architecture.

The key points in our approach is twofold. First, our method is a straightforward lossless compression that constructs the compressed bit stream by identifying the maximum similarity between two adjacent patterns and overlapping them. Second, since this compression is neither destructive nor reorders patterns, no additional decompressor hardware is needed. The decompression process is only performed by the automatic test equipment (ATE) by controlling the JTAG TMS control input.

When test patterns are generated, often plenty of don't cares (up to 90% reported in [31]) exist in the test pattern set. This is also true for patterns generated for signal integrity, especially if a locality metric (to limit exploring the pattern space) is considered [12]. In any case, we assume that the test set consists of same-length patterns (l) that include don't cares. Figure 9 shows our basic idea of compressing two patterns V_i and V_j (of length $l = 16$) by overlapping their bits as much as possible by taking advantage of their don't cares. In this example, the compressed data (V_i, V_j) requires only 21 clocks to scan in as opposed to 16+16=32 clocks if no compression is applied. Note carefully that to decompress a given stream, we need to have one number per pattern (e.g. d_i and d_j in our example) to be able to construct (decompress) the patterns. For purpose of boundary scan test, these numbers are the number of shifts (i.e. clocks) required before updating the content of BSC cells. We assume the ATE stores the decompression data (d values such that $0 \leq d \leq l$) and during scanning in the bit stream, it activates TMS (Test Mode Select) signal after d number of clocks. TMS signal enables the TAP controller to generate appropriate controls for the signal integrity test (e.g. EX-SITEST). Therefore, there is no additional decompression hardware needed in our architecture.

The compression process, as explained before, continues by compressing one pattern at a time to the bit streams obtained up to that point. Figure 10 shows the concept of this constructive process for four patterns $V_1, \cdots, V_4$. The process is repeated until all of the patterns compressed. The decompression data (e.g. $d_1, \cdots, d_4$) are stored in the ATE and will be used to control the TMS signal in the test mode. When there are n patterns $(V_1, \cdots, V_n)$ each of length l, the total length of the compressed data (i.e. total number of clock cycles) needed to deliver them to the interconnects under test will be $\sum_{i=1}^{n} d_i$ which is expected to be much smaller than $l \cdot n$. Thus, for a give set of n patterns the compression rate will be $\frac{l \cdot n - \sum_{i=1}^{n} d_i}{l \cdot n}\%$.

VI. EXPERIMENTAL RESULTS

The enhanced boundary scan cell and architecture is implemented by Synopsys synthesizer [32]. The total area overhead for conventional BSA cell and enhanced BSA cell (ILS) is shown in Table II. The ILS cell is almost twice expensive compared to the conventional one.

Table III shows a comparison between three methods described in Subsection III-A. 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 IV summarizes the compression rate statistics for our technique. In this table, the compression rate for various interconnects are reported. Without judging the quality of the patterns, for comparison reason we have used three pattern sets based on MA model (12 patterns per victim line), deterministic (based on the locality metric reported in [12]) and pseudorandom. The compression rate (as defined in Section V) reflects the application time reduction compared to conventional (uncompressed) method. The compression rate is higher for pseudorandom (with large number of don't cares) and is lower for MA (with no don't cares).

TABLE III

OBSERVATION TEST TIME COMPARISON.

Methods	*Total Test Time [Cycle]*		
	n=8	*n=16*	*n=32*
Method 1	768	3077	12288
Method 2	64	256	1024
Method 3	8	16	32

1093-0167/03 $17.00 © 2003 IEEE

IEEE COMPUTER SOCIETY

TABLE IV
COMPRESSION RATE FOR DIFFERENT TEST PATTERN SETS.

Application Method	*Compression Rate [%]*		
	n=8	*n=16*	*n=32*
MA	37.5	37.8	38.3
Deterministic	46.7	57.2	59.8
Pseudorandom	58.1	61.2	63.8

VII. CONCLUSION

We target enhancing the IEEE 1149.1 standard boundary scan that is the most widely used test methodology in industry. The enhancement allows testing SoCs for integrity loss. The importance of distorted signals in gigahertz systems justifies the cost overhead for the integrity loss sensors. We proposed an inexpensive delay/skew sensor to be added to the boundary scan cells in the receiving sides of interconnects. However, our method is flexible to adopt almost any such sensor. When large test pattern sets are used for integrity testing the test time becomes a matter of concern. We suggested an efficient compression process executed by the ATE to overlap test patterns as much as possible. Our method does not need any on-chip decompressor while it shrinks the scan-in delivery time up to 60% for the examples that we have tried so far.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation CAREER Award #CCR-0130513.

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] J. Shin, H. Kim and S. Kang, "At-Speed Boundary-Scan Interconnect Testing in a Board with Multiple System Clocks", in Proc. *Design, Automation and Test in Europe (DATE'99)*, pp. 473-477, 1999.

[17] S. Kajihara, K. Taniguchi, I. Pomeranz and S. M. Reddy, "Test Compression using Don't care Identification and Statistical Encoding," in Proc. *IEEE Int. Workshop on Electronic Design, Test and Application (DELTA'02)*, pp. 413-416, 2002.

[18] A. Chandra and K. Chakrabarty, "Test Data Compression for System-on-Chip Using Golomb Codes," in Proc. *VLSI Test Symp. (VTS'00)*, pp. 113-120, 2000.

[19] S. Wang and S. Chiou, "Generating Efficient Tests for Continuous Scan," in Proc. *Design Automation Conf. (DAC'01)*, pp. 162-165, 2001.

[20] S. Natarajan, M. Breuer, and S. Gupta, "Process Variations and their Impact on Circuit Operation," in *Proc. IEEE Intern. Symp. on Defect and Fault Tolerance in VLSI Systems*, pp. 73-81, 1998.

[21] Nagaraj NS, P. Balsara, C. Cantrell, "Crosstalk Noise Verification in Digital Designs with Interconnect Process Variations," in *Proc. 14th Intern. Conf. on VLSI Design*, pp. 365-370, 2001.

[22] P. Fang, J. Tao, J. Chen and C. Hu, "Design in Hot Carrier Reliability For High Performance Logic Applications," in Proc. *IEEE Custom Integrated Circuits Conf.*, pp. 25.1.1-25.1.7, Oct. 1998.

[23] R. Downing, P. Gebler and G. Katopis, "Decoupling Capacitor Effects on Switching Noise," *IEEE Transactions on Components, Hybrids and Manufacturing Technology*, vol. 16, no. 5, pp. 484-489, Aug. 1993.

[24] L. He, N. Chang, S. Lin and O. Nakagawa, "An Efficient Inductance Modeling for On-chip Interconnects," in Proc. *IEEE Custom Integrated Circuits Conf. (CICC'99)*, pp. 457-460, 1999

[25] Y. Chang, S. Gupta and M. Breuer, "Test Generation for Maximizing Ground Bounce for Internal Circuitry with Reconvergent Fan-outs," in *Proc. VLSI Test Symp. (VTS'01)*, pp. 191-200, 2001.

[26] 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.

[27] 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.

[28] 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.

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

[30] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison-Wesley, 1998.

[31] F. Wolff and C. Papachristou, "Multiscan-Based Test Compression and Hardware Decompression Using LZ77," in Proc. *Int. Test Conf. (ITC'02)*, pp. 331-339, 2002.

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

IEEE COMPUTER SOCIETY