

Extending JTAG for Testing Signal Integrity in SoCs

N. Ahmed, M. Tehranipour, M. Nourani

Center for Integrated Circuits & Systems

The University of Texas at Dallas

Richardson, TX 75083-0688

{nxa018600,mht021000,nourani}@utdallas.edu

Abstract

As the technology is shrinking and the working frequency is going into multi gigahertz range, the issues related to interconnect testing are becoming more dominant. Specifically, signal integrity loss issues are becoming more important and detection and diagnosis of these losses are becoming a great challenge. In this paper, an enhanced boundary scan architecture with slight modification in the boundary scan cells is proposed to test signal integrity in SoC interconnects. Our extended JTAG architecture: 1) minimizes scan-in operation by using modified boundary scan cells in pattern generation; and 2) incorporates the integrity loss information within the modified observation cells. To fully comply with JTAG standard, we propose two new instructions, one for pattern generation and the other for scanning out the captured signal integrity information.

1. INTRODUCTION

Signal integrity is the ability of a signal to generate correct responses in a circuit. It generally includes all effects that cause design to malfunction due to distortion of the signal waveform. According to this informal definition, a signal with good integrity has: (i) voltage values at required levels and (ii) level transitions at required times. 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.

With fine miniaturization of the VLSI circuits and rapid increase in the working frequency (gigahertz range) of digital system-on-chips (SoC), the signal integrity becomes a major concern for design and test engineers. Although various parasitic factors for transistors can be well controlled during fabrication, the parasitic capacitances, inductances and their cross coupling effects on the interconnects play a significant role in the proper functionality and performance of high-speed SoCs.

The impact of process variations and the way they effect the circuit operation [1] is an important issue. The process variations and manufacturing defects lead to noise and delay effects. The goal of design for deep submicron (DSM) phase is to minimize noise and delay. However, 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 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 is essential to ensure error free operation of the chip and must be addressed in manufacturing testing.

Regardless of the methods to detect integrity loss, we need a mechanism to manage the test session. One of the best choices is boundary scan test methodology that helps test designer to use the capability of accessing interconnects. Boundary scan test methodology was initially introduced to facilitate the testing of complex PC boards. The IEEE

1149.1 Boundary Scan Test Standard [2] 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 and 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. Unfortunately, the standard boundary-scan architecture exposed shortcomings for timing related tests. This drawback is due to the time interval between the update of test stimulus and the response capture, an interval which spans at least 2.5 test clock cycles ($2.5TCKs$). In this paper, the standard boundary-scan architecture is extended to test interconnects for noise and skew violations.

1.1 Prior Work

Various signal integrity problems have been studied previously for radio frequency (RF) circuits and recently for high-speed deep-submicron VLSI chips. The most important ones are: *crossstalk* (signal distortion due to cross coupling effects between signals) [3] [4], *overshoot* (signal rising momentarily above the power supply voltage) [5], *reflection* (echoing back a portion of a signal), *electro-magnetic interference* (resulting from the antenna properties) [7] and *signal skew* (delay in arrival time to different receivers) [8]. 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. These solutions include: 3-D layout modeling and parasitic extraction, accurate RLC simulation of on-chip power grid [8], using decoupling capacitors to limit the maximum dV/dt [9] and to improve IR-drop [8][10], inserting repeaters/buffers on the interconnects and shielding wires (e.g. grounding every other line) [11].

Noise and skew imposed by interconnects have emerged as main concerns in interconnect design of gigahertz SoCs. Buffer insertion and transistor resizing methods [12] are used as two design techniques to achieve better power-delay and area-delay tradeoffs. Self-test methodologies have been developed to test signal integrity in high-speed SoCs. Testing crosstalk in chip interconnects [3][13] and a BIST (built-in self-test) structure using D flip-flops that detects the propagation delay deviation of operational amplifiers [14] are among such methods.

Most of the early work in testing interconnects focused on the development of deterministic test for interconnect faults [15]. Later researches have focused on delay testing [16], at-speed testing [17] and BIST architecture extension [18] in the context of boundary-scan architecture. A modified boundary-scan cell using an additional level sensitive latch (Early Capture Latch) was proposed in [16] 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 [17], 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. A built-in current sensor presented in [19] is used for testing

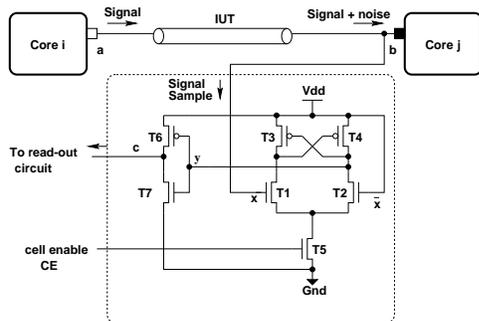


Figure 1: The ND cell using cross-coupled PMOS amplifier.

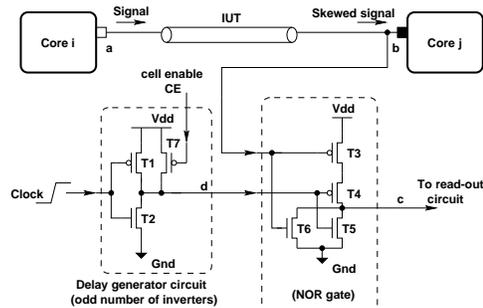


Figure 2: Skew Detector circuitry (SD cell).

timing-related faults in boundary scan architecture for testing buses.

The IEEE P1149.6 working group are studying a solution for testing AC-coupled interconnects between integrated circuits on printed circuit boards and in systems [21]. Our approach is similar to this standard draft in enhancing the JTAG standard and its instructions for testing high-frequency behaviors. However, there are fundamental differences. In contrary to our approach the draft is not intended to consider coupling effects among the interconnect lines. Also, 1149.6 adds a DC blocking capacitor to each interconnect under test to disallow the DC signals. Thus, 1149.6 can not test noise due to low-speed but very sharp-edge signals that are known to cause overshoots and noise. Our sensors can detect such scenarios. Finally, using differential drivers in the modified cells in 1149.6 makes the cells more expensive and less flexible in adopting other type of noise detector/sensors.

1.2 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 noise and skew violations occurring on the interconnects of high-speed SoCs can be tested using JTAG boundary scan architecture. The modified sending end boundary scan cells (PGBSC) used for test pattern generation for cross-talk noise violations is proposed. Special cells (OBSC) to monitor signals received from the system interconnect are incorporated in the boundary scan cell which record the occurrence of signal entering the vulnerable region over a period of operation. Using two new instructions in JTAG architecture the integrity test information is sent out for final test analysis, reliability judgment and diagnosis.

The rest of the paper is organized as follows. Section 2 reviews the ND, SD cells and the multiple aggressor fault model. The enhanced boundary scan cells are proposed in Section 3. Section 4 explains the test architecture to send test patterns and capture and read out the signal integrity information. The experimental results are discussed in Section 5. Finally, the concluding remarks are in Section 6.

2. BACKGROUND

2.1 Noise Detector (ND) cell

The noise detector (ND) cell proposed in [18] is a modified cross-coupled PMOS differential sense amplifier designed to detect integrity loss (noise) relative to voltage violations. Figure 1 shows the noise detector (ND) cell, which sits physically near the receiving core and samples the actual signal plus noise received by Core j. Each time that noise occurs (i.e. $V_b > V_+ = V_{Hthr}$), the ND cell generates a “0” signal that remains unchanged until V_b drops below $V_- = V_{Hmin}$. V_{Hthr} and V_{Hmin} are the voltage limits that represent logic ‘1’. The output of the cell is determined by the cell enable (CE) signal. Briefly the cell is active when $CE=’1’$ and the output V_c generates a $1 \rightarrow 0$ transition when V_b crosses V_{Hthr} .

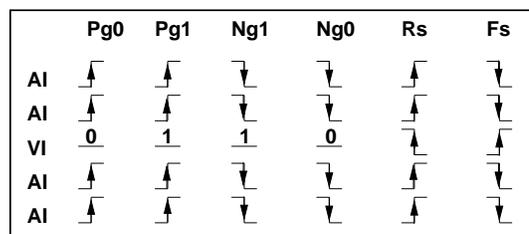


Figure 3: Maximum aggressor fault model.

2.2 Skew Detector (SD) cell

The skew detector (SD) cell proposed in [18] facilitates effective skew violation detection. The skew immune region depends on the maximum storage-to-storage (s-to-s) path delay [20]. Figure 2 depicts the skew detector (SD) cell. A delay generator cell is used to create the desired delay value (i.e. acceptable skew-immune range) as it is defined by designer based on the delay budget of the interconnect. The delayed clock is compared with the interconnect output. If the skew of the signal on the interconnect output is not within the acceptable range, the SD cell issues a pulse. The duration of this pulse depends on the interconnect delay. This pulse is used to trigger to a D flip-flop to store a “1” as indication of a skew violation. Briefly, the cell is active when $CE=’1’$ and the output V_c generates a $0 \rightarrow 1 \rightarrow 0$ pulse when skew violation occurs.

2.3 Integrity Fault Model

In our test methodology, we use the maximum aggressor (MA) fault model [13]. This is a simplified model used by many researchers for noise and delay analysis on long interconnects. The interconnect on which the integrity loss takes place is defined as the victim interconnect (VI). The other wires that act collectively to cause violation on the VI are considered aggressor interconnects (AI). Figure 3 shows the general signal transitions needed on the VI and AIs to produce the strongest error effects on a VI in a five-wire interconnect system. The MA defines six faults based on the resulting noise and skew error effects, i.e., positive glitches P_{g0} , P_{g1} , negative glitches N_{g0} , N_{g1} , and rising/falling skew R_s , and F_s .

Each of the mentioned signal transitions contains two consecutive test vectors. For example, to generate a positive glitch P_{g0} in Figure 3, two test vectors ‘00000’ and ‘11011’ are required. For a set of n interconnects, a total of 6/12 faults/patterns for each victim interconnect need to be tested/applied. Therefore, based on MA model total number of required test vectors for a set of n interconnects system is $12n$.

3. ENHANCED BOUNDARY SCAN CELLS

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 with shift and update stages. The

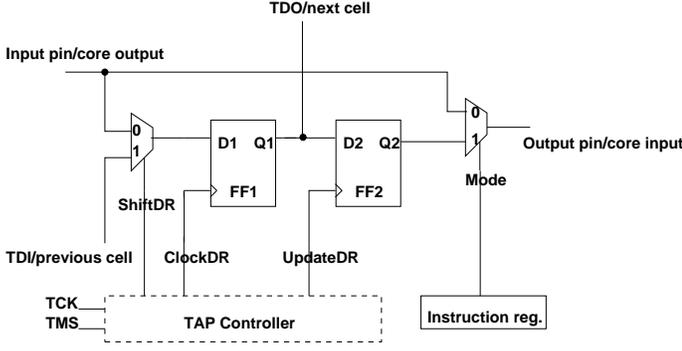


Figure 4: A Standard Boundary Scan Cell.

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 for signal integrity testing of interconnects. We propose new cells and instructions for signal integrity test. For this purpose, some minor modifications are applied to the standard architecture to target the interconnects for signal integrity. Although our approach imposes some area increase, the additional logic inserted inside boundary scan cells is solely on the test path, hence keeping the normal operation intact timing-wise.

3.1 Pattern Generation BSC (PGBSC)

As mentioned before, two test vectors are needed for each of the six integrity faults. Therefore, 12 test patterns should be generated for one victim line. These patterns can be applied to the interconnects in a boundary scan architecture (BSA). 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 MA 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).

Analysis of the MA fault model 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 clock. This important observation helps in reordering patterns such that the amount of data to be scanned in is minimized. The order of the test vectors, applying to the interconnects for only one victim line in a five interconnect system is shown in Figure 5. We consider two initial values for generating the test vectors, i.e., “00000” and “11111”. It shows that after applying the first initial value, “00000”, the generated test patterns cover the P_{g0} , F_s , and P_{g1} faults. The generated test patterns after applying the second initial value, “11111”, cover N_{g1} , R_s , and N_{g0} . Therefore, by such reordering only 8 test patterns are sufficient for covering all faults in the MA fault model. More importantly, only initial values need to be scanned in. This significantly reduces the number of required clock cycle for applying test patterns to the interconnects in a system-chip. One may think that one initial value (e.g. “00000”) is sufficient and

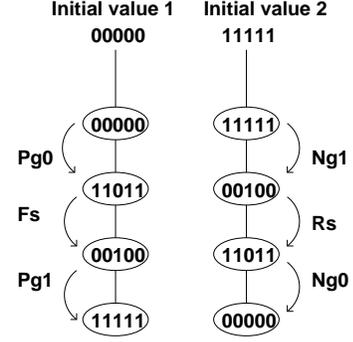


Figure 5: Test vectors generated by a PGBSC.

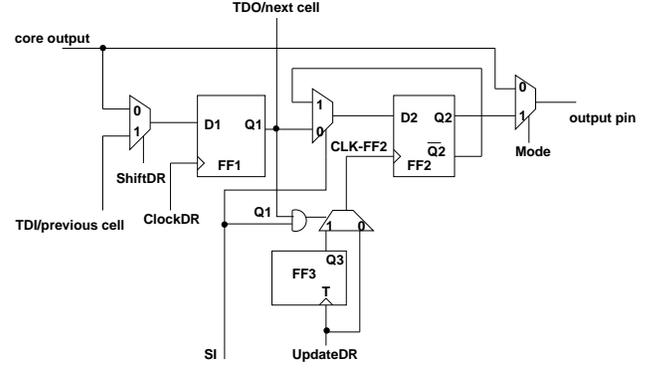


Figure 6: PGBSC design.

N_{g1} can follow P_{g1} . However, a careful examination of that scenario reveals that the victim line goes through $0 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 0 \rightarrow 0$. In such case, the transition frequency of victim line is not half of the aggressor line and hence cannot be used. Having two initial values, as shown in Figure 5, makes the transition frequency of aggressor lines to be always twice as victim line. This significantly simplifies the design of PGBSC cell.

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 6. 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 MA fault model. Table 1 shows the operation modes of the PGBSC. Depending on the select line of the mux attached to FF3, this architecture has three modes:

1. **Victim mode:** Q_3 is selected. *UpdateDR* is divided by two and applied to the FF_2 . By every two *UpdateDR*s, 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 7 shows the operation of a PGBSC. If PGBSC is in victim mode, *UpdateDR* is divided by two and generates $CLK-FF_2$. If the

Table 1: Operational modes of the PGBSC.

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

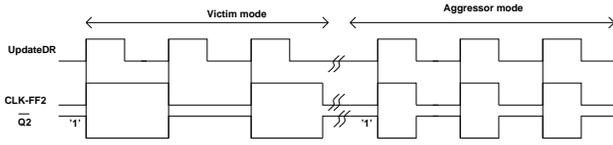


Figure 7: The operation of the PGBSC.

Table 2: One-hot encoded data for victim line.

Victim-select data	Victim line
10000	1
01000	2
00100	3
00010	4
00001	5

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 $UpdateDR$ the content of FF_2 is complemented. As shown in Figure 6, $\overline{Q_2}$ is complemented by each $CLK-FF_2$ while Q_2 is applied to the output (to the interconnect).

Each interconnect acts as victim and aggressor. Therefore, in the test session each time the victim interconnect should be specified. For example, in Figure 3, interconnect 3 is victim. After performing the test process on interconnect 3, it will be an aggressor for other new victims. Briefly, for complete interconnect testing, the victim line rotates. We use one-hot encoded data to specify the victim which is called *victim-select data*. Table 2 shows the scanned in victim-select data, to be stored in FF_1 , for a five interconnects system. 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 2, when we scan in the '10000' to five PGBSCs, the first line is victim and others are aggressors. For changing the second line to the victim, only one '0' is scanned in FF_1 to change victim-select data to '01000'.

The generic behavior of test pattern generation and applying procedure is shown in Figure 8. This behavior will be executed by a combination of automatic test equipment (ATE) and TAP controller. First, "00000" is applied to the BSCs as an initial value and then the cells are set in SI mode. After generating test vectors and applying them to the interconnects, a new victim is selected and the process will be repeated. The same process will be repeated with the second initial value, '11111'.

3.2 Observation BSC (OBSC)

We also propose a new BSC at the receiving side of the interconnects which utilizes the noise and skew detector (ND/SD) cells described in Section 2. Figure 9 shows the new BSC named observation BSC (OBSC). As shown, ND and SD cells are added to the receiving side cells. The ND/SD cells capture signals with noise and delay at the

```

01: for (k= 1 to 2)
02: {
03:   Scan initial value k into FF2
04:   Activate signal integrity test mode (SI=1)
05:   Scan the first victim-select data
06:   For (wire i=1 to n)
07:   {
08:     Apply 3 UpdateDRs. // Pattern generation
09:     Shift one '0' into FF1 // Selecting new victim
10:   }
11: }

```

Figure 8: Test pattern generation procedure using PGBSC.

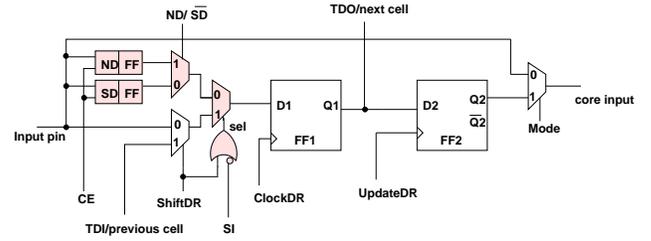


Figure 9: Observation BSC.

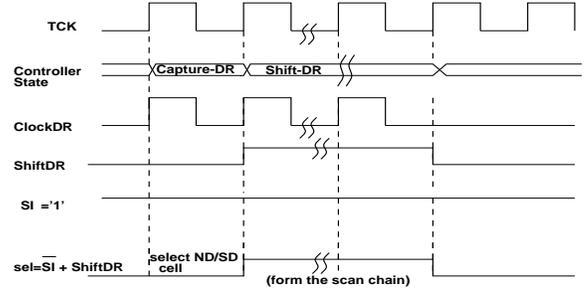


Figure 10: Operation of observation BSC.

end of an interconnect. If they receive a signal with integrity problem (noise or skew violation) they show a pulse at their output and the FFs are set to '1'. The cells are activated by the signal cell enable ($CE = '1'$). If $CE = '0'$, the cells are disabled but the captured data in their flip-flops remain unchanged. The OBSC operates in three modes as summarized in Table 3.

1. **NDFF mode:** ND cell flip-flops are selected. In this case, the captured ND cells data are scanned out every $Shift-DR$ state through the scan chain for final evaluation.
2. **SDFF mode:** SD cell flip-flops are selected. Every $Shift-DR$ state, the data stored in SD cell FFs are scanned out.
3. **Normal mode:** In this mode, the ND/SD cells are isolated and each OBSC acts as a standard BSC.

In the SI test mode, as Figure 9 shows only one of ND or SD cell FFs can be read and scanned out for final evaluation. For reading both ND and SD cells, the scan out process should be repeated twice (once for ND cell FFs and once for SD cell FFs). Before starting the scan out process, we need to send the content of one of the ND/SD FFs to FF_1 . 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. After sending the value of ND or SD FF of each cell 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 FF_1 . Therefore, the ND/SD cells path should be isolated by $sel='1'$ ($SI='1'$ and $ShiftDR='1'$). As shown in Figure 9, SI and $ShiftDR$ are Ored together for selecting the ND/\overline{SD} path for transferring the ND/SD cell FFs to D_1 and the making of the scan chain to scan out. Figure 10 shows the dependency of sel to the SI and $ShiftDR$. As shown, in $Capture-DR$ state, ND/SD cell FFs are 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 4 shows the truth table of signal sel . Additional control signals (i.e. SI , CE and ND/\overline{SD}) are generated by a new instruction, to be explained in Section 4. There are three methods of observation:

Table 3: Operational modes of the OBSC.

Observation mode	ND/SD	SI
NDFF	1	1
SDFF	0	1
Normal	x	0

Table 4: Truth Table of signal *sel*

<i>SI</i>	<i>ShiftDR</i>	<i>sel</i>
1	0	0
1	1	1
0	x	1

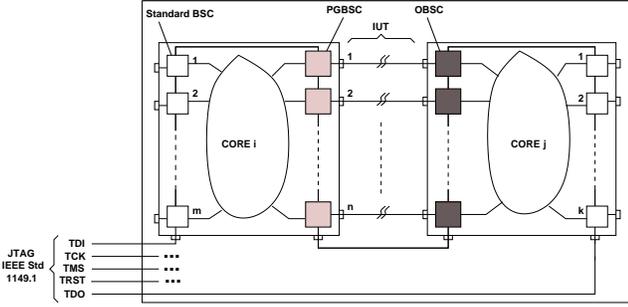


Figure 11: Test Architecture

- Method 1:** To capture and scan out the ND/SD cells data only once after the entire test patterns application covering all the victims.
- Method 2:** To capture and scan out the ND/SD cells data twice, once after the application of patterns covering faults P_{g_0} , F_S and P_{g_1} for all the victims and the next time after the application of patterns covering faults N_{g_1} , R_S and N_{g_0} for all the victims.
- Method 3:** To capture and scan out the ND/SD cells data right after applying each test pattern.

The first method has the advantage of less test time and a disadvantage of not being able to determine which transitions have caused the fault in an interconnect. The second method provides more information to determine as to which set of transitions or faults caused the violation in the interconnects at the expense of more test time. Finally, the third method shows the best information of the test, but it is extremely time consuming. In the experimentations reported in this paper we compare these methods.

4. TEST ARCHITECTURE

Figure 11 shows the overall test architecture with n interconnects between the cores i and j in a two-core SoC. The JTAG inputs (TDI , TCK , TMS , $TRST$ and TDO) are still used without any modification. Two new instructions are defined to be used for signal integrity test, one to activate PGBSCs to generate test patterns and the other for reading out the test results. As shown in Figure 11, 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 OBSCs. The other cells are standard BSCs which are present in the scan chain during the signal integrity test mode.

The ND/SD cells act independently and no special control circuitry is required to control the timing of these cells. After all the patterns for the MA fault model are covered, the signal integrity information stored in the cell FFs is scanned out to determine which interconnect has a problem. This is an efficient method (method 1 as explained in previous section) to decrease the test application time since the information in the cells is scanned out once instead of each time after the application of a pattern.

In conventional BSA, test patterns are scanned in one-by-one and applied on the interconnects. For example in a n interconnect network, 12 test patterns are applied to each victim line and $12.n$ clock is required to apply the test patterns on only one victim line. With rotating victim line among n interconnects, the required clock is $12.n.n$. It shows the complexity of test application time for conventional BSA is $O(n^2)$. In the case of PGBSC, two initial values are applied to the cells and the

```

01: for (k i=1 to 2)
02: {
03:   Load SAMPLE/PRELOAD and Shift initial value k.
04:   Load G-SITEST into IR.
05:   for(l=1 to n)
06:   {
07:     Apply 3 UpdateDRs. //3 Test patterns
08:     Shift victim-select data.
09:   }
10: }
11: Load O-SITEST into IR.

```

Figure 12: Signal integrity test process.

other test patterns are generated by PGBSCs. Having n rotating victim interconnects, the required clocks is $2n$. The order complexity using PGBSC for test pattern generation and application is $O(n)$.

4.1 Instruction set

We propose to add two new instructions $G-SITEST$ and $O-SITEST$ to the IEEE 1149.1 instruction set for our new test architecture.

• $G-SITEST$ Instruction

This instruction is used for test pattern generation using the enhanced architecture. The instruction is loaded after shifting in the initializing data into the PGBSCs. $G-SITEST$ targets the PGBSCs and enables $SI=1$ throughout the instruction. It also enables the ND/SD cells ($CE=1$) to capture the signal integrity information. The victim-select data is then shifted into FF1 of the PGBSCs during the $Shift-DR$ state and the patterns for MA fault model are generated every $Update-DR$ state as explained in Subsection 3.1. Three $UpdateDRs$ are required to generate three test patterns per victim line for each initial value.

• $O-SITEST$ Instruction

This instruction is loaded after the $G-SITEST$ instruction. It is used to capture and scan out the ND/SD FFs data. After the instruction is decoded in the $Update-IR$ state, control signals $SI=1$ and $CE=0$ (to deactivate ND/SD cells) are generated. ND/\overline{SD} is initialized to logic 1 to select ND cell FFs during the first shift operation. The ND/\overline{SD} signal is complemented in the $Update-DR$ state to select SD cell FFs during the next shift operation.

4.2 Test Algorithm

Figure 12 shows the test process in the signal integrity mode. As shown, the new BSCs which target signal integrity test for the interconnects are set in signal integrity mode after loading the $P-SITEST$ instruction. Then, all test patterns generated by PGBSCs are applied to the interconnect and simultaneously ND/SD cells capture the signals at the end of interconnects and detect the violations if any. After test application process, the stored results in the ND/SD cell FFs must be read. This is done using the $O-SITEST$ instruction. First, the ND/SD cells are deactivated because the value of the ND/SD FFs need to be preserved in the reading out process. It is important to deactivate the ND/SD cells because during the scan out operation some new data will be scanned in and it may be applied to the interconnects in the $Update-DR$ state. This may cause a change in the ND/SD cell data changing the previous value. Finally, the scanning out process is performed as explained in the $O-SITEST$ instruction.

5. EXPERIMENTAL RESULTS

As mentioned earlier, we designed the PGBSC to apply the test patterns at-speed. Using the hardware based test generation reduces the number of required clocks. Table 5 shows a comparison between the number of clocks required for applying test patterns to cover all faults in MA fault model. In conventional method, using ClockDR the test patterns are scanned into the cells. Using UpdateDR the test patterns are applied to the interconnects. This process is performed for all twelve test patterns. The last row shows time improvement that our

Table 5: Pattern generation time analysis

Test Architecture	Total Test Time ($m=0$)		
	$n=8$	$n=16$	$n=32$
Conventional	1152	3840	13824
PGBSC	264	520	1032
$\Delta T\%$	77.1	86.5	92.5

Table 6: Test time analysis

Methods	Total Test Time ($k=0$)		
	$n=8$	$n=16$	$n=32$
Method 1	16	32	64
Method 2	32	64	128
Method 3	1536	6144	24576

method achieves by using PGBSC cells. The table shows that compared to conventional scan our method is more efficient for large number of interconnects (n).

Table 6 shows a comparison between three methods described in Subsection 3.2. The required clocks in observation side is equal for both conventional and enhanced BSA. The table shows that the number of clocks required for methods 1 and 2 is significantly lower than method 3. However, method 3 provides much information about type and location of the integrity faults.

The new boundary scan cells are implemented by SYNOPSIS [22]. The total area overhead is shown in Table 7 for a 32-bit wide interconnect. The new cells are almost twice expensive compared to the conventional cells. Practically these cells are used only for those long interconnects susceptible to signal integrity faults.

6. CONCLUSION

We proposed an enhanced boundary scan architecture for testing signal integrity in SoCs. Our architecture detects skew and noise violations using the standard JTAG boundary scan architecture. To do this, additional detector cells, modified scan cells and minor modifications to the TAP controller to handle two new instructions are needed. The advantage of the proposed architecture is that it provides cost effective solution for thorough testing of interconnects with a slight area overhead using the popular JTAG standard.

Acknowledgements

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

REFERENCES

- [1] S. Natarajan, M.A. Breuer, S.K. Gupta, "Process variations and their impact on circuit operation," in Proc. *IEEE International Symposium*, pp. 73-81, 1998.
- [2] IEEE Standard 1149.1-2001, "Standard Test Access Port and Boundary-Scan Architecture", IEEE Standards Board, 2001.
- [3] M. Cuvillo, S. Dey, X. Bai and Y. Zhao, "Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects," in Proc. *Intern. Conf. on Computer Aided Design (ICCAD'99)*, pp. 297-303, 1999.
- [4] W. Chen, S. Gupta and M. Breuer, "Test Generation in VLSI Circuits for Crosstalk Noise," in Proc. *Intern. Test Conf. (ITC'98)*, pp. 641-650, 1998.
- [5] 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.
- [6] Y. Leblebici, "Design Considerations for CMOS Digital Circuits with Improved Hot-Carrier Reliability," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 7, pp. 1014-1024, July 1996.
- [7] S. Kimothi and U. Nandwani, "Uncertainty Considerations in Compliance-Testing for Electromagnetic Interference," In Proc. *Annual Reliability and Maintainability Symp.*, pp. 265-268, 1999.
- [8] H. Chen and L. Wang, "Design for Signal Integrity: The New Paradigm for Deep-Submicron VLSI Design," In Proc. *Intern. Symp. on VLSI Tech.*, pp. 329-333, 1997.
- [9] 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.
- [10] R. Saleh, D. Overhauser and S. Taylor, "Full-Chip Verification of UDSM Designs," in Proc. *Intern. Conf. on Computer Aided Design (ICCAD'98)*, pp. 453-460, 1998.
- [11] A. Kahng, S. Muddu and E. Sarto, "Interconnect Optimization Strategies for High-Performance VLSI Designs," in Proc. *Intern. Conf. on VLSI Design*, pp. 464-469, Jan. 1999.
- [12] G. Tellez and M. Sarrafzadeh, "Minimal Buffer Insertion in Clock Trees with Skew and Slew Rate Constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 4, pp. 333-342, April 1997.
- [13] 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.
- [14] 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.
- [15] C. Chiang and S. K. Gupta, "BIST TPGs for Faults in Board Level Interconnect via Boundary Scan", in Proc. *VLSI Test Symposium (VTS'97)*, 1997.
- [16] K. Lofstrom, "Early Capture for Boundary Scan Timing Measurement", *Proc. ITC*, pp. 417-422, 1996.
- [17] 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.
- [18] M. Nourani and A. Attarha, "Built-In Self-Test for Signal Integrity," in Proc. *Design Automation Conf. (DAC'01)*, pp. 792-797, June 2001.
- [19] 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.
- [20] J. Wakerly, *Digital Design, Principles and Practices*, Prentice Hall, 2000.
- [21] IEEE 1149.6 Working Group, <http://grouper.ieee.org/groups/1149/6/>, 2002.
- [22] Synopsys Design Analyzer, "User Manuals for SYNOPSIS Toolset Version 2000.05-1," Synopsys, Inc., 2000.

Table 7: Cost analysis

Test Architecture	Cost[Nand] ($n=32, m=k=0$)		
	sending	observing	total
Conventional BSA	26	26	1664
Enhanced BSA	44	57	3232