High-level Crosstalk Defect Simulation for System-on-Chip Interconnects*

Xiaoliang Bai, Sujit Dey
ECE Department
UC San Diego
La Jolla, CA 92037
{xibai, dey}@ ece.ucsd.edu

Abstract

For system-on-chips (SoC) using deep submicron (DSM) technologies, interconnects are becoming critical determinants for performance and reliability. Buses and long interconnects are susceptible to crosstalk defects and may lead to functional and timing failure. Hence, testing for crosstalk errors on interconnects and buses in a SoC has become critical. To facilitate development of new crosstalk test methodologies and to efficiently evaluate crosstalk defect coverage for existing tests, there is a need for efficient crosstalk defect coverage analysis techniques. In this paper, we present an efficient high-level crosstalk defect simulation methodology. By using a novel high-level DSM error model for the interconnects, together with HDL models for the cores, our methodology enables fast crosstalk defect simulation to be conducted at high level. We validate the high-level interconnect DSM error model by comparing its outputs with HSPICE simulation results. The fast and accurate high-level crosstalk defect simulation methodology will enable evaluation and exploration of new crosstalk test techniques, as well as existing tests, leading to the development of low-cost crosstalk test.

Keywords: Crosstalk, System-on-Chip, Interconnect test, Defect simulation, High level

1. Introduction

Due to the relatively increased vertical/lateral ratio of interconnects, increased circuit frequency, and the increased number of layers of interconnects, cross-coupling capacitances between interconnects can significantly affect the performance of the circuit and even cause malfunction of the chip. On-chip interconnects have become a critical determinant of performance and reliability in high performance system-on-chip (SoC). Several design techniques [1] [2] [3] [4] and analysis techniques [5] [6] [7] have been developed to help design for margin and minimize signal integrity problems. Even so, process variations and manufacturing defects may lead to an unexpected increase in cross-coupling capacitance between interconnects, resulting in glitches and delay effects, which may cause logic error and failure of the chip. Any process variation and manufacturing defects, which may lead to crosstalk errors, are *crosstalk defects*. Since it is impossible to predict the occurrences of crosstalk defects, testing of crosstalk defects is essential to ensure error free operation of the SoC.

Previous work has shown that crosstalk effects are most significant in long interconnects [8] [9]. Therefore, it is important to develop test and defect simulation methodologies for crosstalk errors on long interconnects. Recently test methodologies and fault models have been developed for testing crosstalk defects [9] [10] [11] [12] [13] [14] [15], such as the Maximal Aggressor Fault Model (MAFM) [9]. The effectiveness of the MA fault model has been validated using a SPICE based crosstalk defect simulation methodology. However, it may not be possible to apply the MA tests from chip I/Os to achieve high fault coverage of the deeply embedded chip interconnects. Moreover, crosstalk tests need to be applied at-speed, requiring expensive external testers. To facilitate high-quality at-speed testing of chip interconnects, a self-test method was proposed in [16], which uses embedded Built-in Self-Test (BIST) structures to generate MA tests for the chip interconnects and detect any potential errors. However, implementation of the BIST structure may result in high area and delay overheads, imposing limitations upon its application.

A less expensive alternative may be to use legacy tests, such as functional, delay, BIST, or boundary scan tests, if the existing test vectors have high crosstalk defect coverage. To evaluate the defect coverage for crosstalk errors, SPICE based defect simulation can be used. Although SPICE based defect simulation is accurate, it may be prohibitively time consuming for complex SoCs. To efficiently evaluate the crosstalk defect coverage of a given test set, an analytical methodology has been developed[17]. Although it is faster than SPICE based defect simulation, the analysis graph size grows exponentially with the complexity of interconnects, and the analysis methodology cannot handle all types of test patterns. Hence, it is critical to develop an alternative method, which can accurately evaluate the crosstalk defect coverage of a given test set for arbitrarily complex SoC interconnect architectures. At the same time, the new method needs to be fast, so that it can be used iteratively to evaluate alternative crosstalk test methodologies.

In this paper, we propose a high-level crosstalk defect simulation methodology for fast crosstalk fault simulation and defect coverage evaluation. This methodology uses a novel high level crosstalk model of the interconnect, together with high level HDL models of the cores, to achieve fast yet accurate defect simulation of the interconnects in a SoC. We develop the high-level interconnect DSM error model, and validate it by comparing its result with HSPICE simulation results. The high-level crosstalk defect simulation methodology has

* This work is supported by the Semiconductor Research Corporation under contract #98-TJ-648

been applied to validate a self-test methodology for crosstalk errors, and has also been used to evaluate crosstalk defect coverage for an existing test set.

In section 2, we discuss the interconnect crosstalk effects and properties. Section 3 introduces the high-level crosstalk defect simulation methodology and describes the derivation of a high-level interconnect DSM error model, which is used by the high-level crosstalk defect simulation methodology. Section 4 presents the validation of the high-level interconnect DSM error model by comparing its outputs with HSPICE simulation results. Section 5 reports two applications of the high-level crosstalk defect simulation methodology. Section 6 concludes this paper.

2. Interconnect Crosstalk Effects and Properties

The adverse effects of increased cross-coupling capacitance and inductance on signal integrity can be threefold. When cross-coupled capacitance becomes a first order parameter between two interconnects, two basic signal anomalies can take place as a result of step inputs. In the first case, when one signal is switched (for example, Y_1 switched high in Figure 1) and the other is driven steadily (Y_2 driven low) the energy is transferred through cross-coupling capacitance and results in a voltage glitch on the steady signal (Y_2). This is shown in Figure 1(a). The second anomaly, when the two lines are switched to opposite values (for example, Y_1 switched to high and Y_2 switched to low), the transition time increases, as shown in Figure 1 (b).

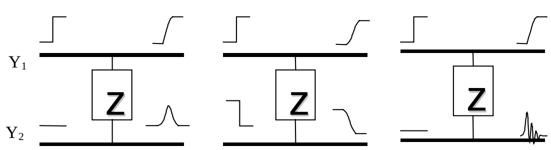

Figure 1 - Three signal anomalies (a) Glitch , (b) Delay, and (c) Oscillations

When inductance is combined with the other elements of the circuit model, the voltage relationship generally results in high-order differential equations. In addition to glitches and delays, damped voltage oscillations will be superimposed on top of a glitch or delay, as illustrated in Figure 1(c). If the damping is large enough, the effect of this case may be approximated using one of the first two cases. In this paper, we consider the error effects induced by the cross-coupling capacitances.

In this work, we consider four different crosstalk effects, as shown in Figure 2. When the signal on one interconnect is steadily high, such as Y_2 in Figure 2 (a), and signals on neighboring interconnects (Y_1, $Y_3 \ldots Y_n$) are largely falling transitions, they cause a glitch towards low on Y_2. According to the direction of the glitch, we name it as negative glitch (g_n). Since the signal on Y_2 is affected, and may cause an erroneous operation, we name Y_2 as the victim, and other interconnects as the aggressors. Similarly, as shown in Figure 2 (b), since a glitch towards high is generated on interconnect Y_2, it is named positive glitch (g_p). If a rising transition is delayed because of cross-coupling capacitances, it is named as a rising delay (d_r), as shown in Figure 2 (c). In Figure 2 (d), a falling transition is delayed, and hence is named as falling delay (d_f).

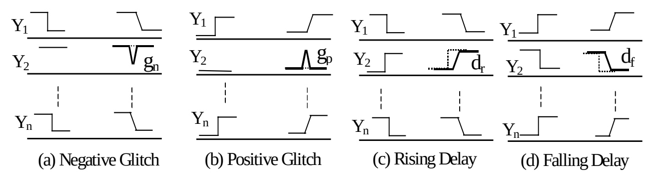

Figure 2- Four different effects induced by cross-coupling capacitances

To simulate crosstalk defects, which may results in error effects described above, the first step is to define crosstalk errors. What constitutes an error depends on the receiving logic, and the technologies used. Next we will describe the error definitions that are used in our proposed high-level crosstalk defect simulation methodology.

Take positive glitch (g_p) as an example. In order for the receiver to sample an erroneous value, the glitch's voltage should be higher than a threshold value Vp_{th}, for a duration Δt_g, as shown in Figure 3 (a). This means the glitch should be large (magnitude) and wide enough, and should fit into the sampling window of the receiver. Here, the Vp_{th} and Δt_g are determined by the threshold value, setup time and hold time of the receiver. Figure 3 (b) shows the definition for falling delay (d_f). For an erroneous value to be sampled by the receiver, the voltage of the delayed signal should be less than Vp_{th} at time T, for a time period Δt_d. Similarly we define errors for negative glitch (g_n) and rising delay (d_r).

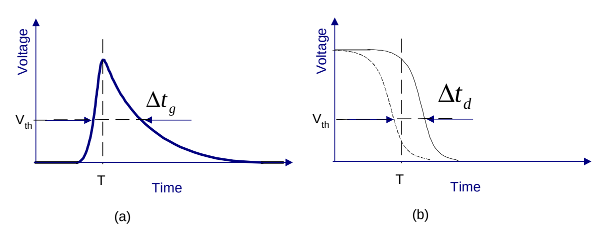

Figure 3- Definition of errors (a) Definition for g_p (b) Definition for d_f

Next we restate two properties related to crosstalk defects and crosstalk error effects, and define a *threshold cross-coupling capacitance C_{th}*, which will help in deriving a high-level DSM error model. For a given interconnect system, crosstalk effects are related to the summation value of cross-coupling capacitances, as described in [9]:

- *When the driver strength is sufficiently high, (driver resistance sufficiently low), the difference in effects (glitch or delay) between different capacitance distributions with the same summation value, becomes negligible.*
- *Glitch and delay effects increase monotonically as the total cross coupling capacitance between the*

victim and all the aggressors increases monotonically.

Based on the above properties, for each type of crosstalk effects on a victim wire, a corresponding threshold capacitance (C_{th}) can be defined such that if any cross-coupling capacitance distribution exceeds the $C_{th,}$ it may cause an error [9]. The C_{th} values can be determined from the error definitions by circuit simulation with monotonically increased cross-coupling capacitances [9]. Next we will discuss the high-level DSM crosstalk defect simulation methodology, which uses the threshold capacitance for efficient crosstalk effects estimation.

3. High-level Crosstalk Defect Simulation Methodology

The goal in developing the high-level crosstalk defect simulation methodology is to enable very fast simulation of interconnects in a SoC, without compromising the accuracy of the crosstalk defect coverage analysis. Our simulation methodology uses HDL models for the cores and new high-level interconnect error model for the interconnects of the SoC. An example of the high-level simulation environment of a SoC with cores and interconnect hierarchy is shown in Figure 4, consisting of cores 1 to 5. Core 1, 2, 4 and 5 communicate through one bus, while core1 and core3 communicate via a dedicated local bus. The crosstalk defects on buses are simulated by using the high-level interconnect DSM error model, which will be explained next.

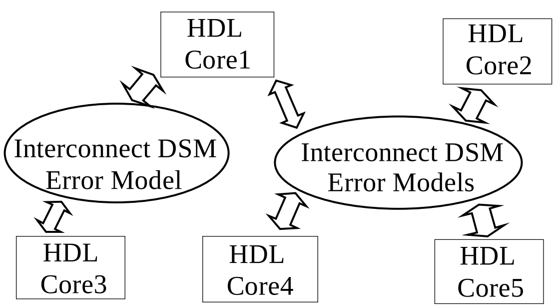

Figure 4 – High-level crosstalk simulation environment

The most accurate way of modeling of interconnects for glitch and delay error effects, is to use transmission line model consisting of the R, L, C parameters. However, the use of this model would make simulation very slow. Therefore, we developed an interconnect DSM error model based on the lumped sum representation of cross-coupling capacitances, and use of threshold capacitances to characterize crosstalk errors, as explained in next section. Figure 5 (a) shows the operations of the interconnect DSM error model. A set of interconnects has its own parameter file, consisting of the cross-coupling capacitances. The error model reads the parameter file with perturbed cross-coupling capacitances, and then depending upon the input vectors, the model produces an output, which resembles the response of the interconnects. Figure 5 (b) shows that if a given set of perturbed cross-coupling capacitances is a defect, which should result in a positive glitch on interconnect 3, the DSM error model will generate a digitized error effect when an appropriate test vector is applied. Note that this methodology needs extraction of coupling capacitance once. Subsequently instead of using time consuming SPICE simulation, crosstalk defect simulation of SoC interconnect can be performed at high level and iteratively using HDL simulation. Next we describe the mechanism used by the interconnect DSM error model to estimate the error effects.

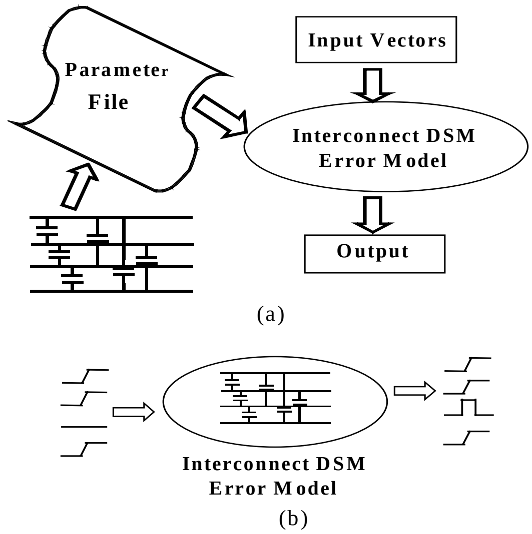

Figure 5 - High-level interconnect DSM error model
(a) Operations (b) Generation of error effect

3.1 Derivation of the High-level Interconnect DSM Error Model

In this section, we will derive our high-level interconnect DSM error model. To explain the mechanism for estimating crosstalk error effects, we use a simple interconnect system as an example. For simplicity, we assume the drivers are balanced and have sufficient strength to allow representation by a piece-wise voltage source.

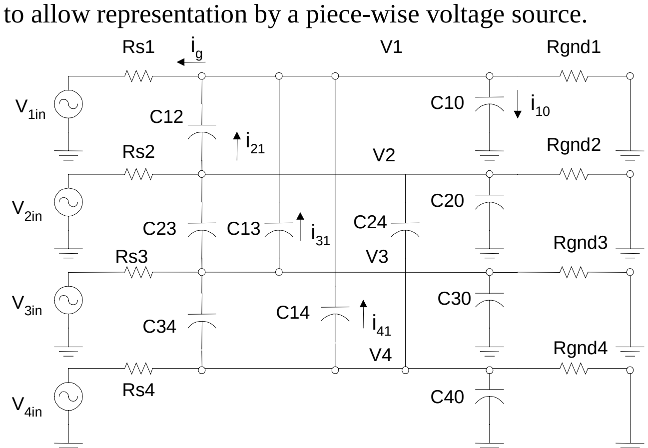

Figure 6 -An example four-wire interconnect system

Consider the four-wire interconnect set shown in Figure 6. Suppose wire 1 is the victim, and 2, 3, and 4 are aggressors. We use Thevenin equivalent model for drivers. V_{1in}, V_{2in}, V_{3in}, V_{4in} are voltage sources for drivers. V_1, V_2, V_3, and V_4 denote the voltages of the receiver end on wire 1, 2, 3, and 4 respectively. C_{10}, C_{20}, C_{30}, and C_{40} are intrinsic and load capacitances to ground, and R_{gnd1}, R_{gnd2}, R_{gnd3}, and R_{gnd4} are the resistances to ground for each interconnect respectively. To derive a high-speed simulation model, we assume that drivers are aligned (transition occurs at the same time) and strong. Thus, the input transitions on the interconnects have either same slope or stay unchanged, and the drivers' voltage sources can be characterized as follows:

$$\frac{dv_2}{dt} = S_2\,|V_a'|$$
$$\frac{dv_3}{dt} = S_3\,|V_a'| \qquad (1)$$
$$\frac{dv_4}{dt} = S_4|V_a'|$$

$|V_a'|$ is the absolute value of the input signal transition slope. S takes the value +1 for rising transitions, -1 for falling transitions, and 0 when the input signal steady.

Aggressors contribute currents to victim wire. The current flow from interconnect 2 to 1 is:

$$i_{21} = C_{12}\frac{d(v_2 - v_1)}{dt} \qquad (2)$$

and the currents to ground on victim wire are:

$$i_g = \frac{v_1}{R_{gnd1} \parallel R_{s1}} \qquad (3)$$

$$i_{10} = C_{10}\frac{dv_1}{dt} \qquad (4)$$

From KCL, we know that

$$i_{10} + i_g - i_{21} - i_{31} - i_{41} = 0 \qquad (5)$$

Let us consider a positive glitch generated on the victim. In this case, the input for the victim does not change. The input vectors on the aggressors contribute charges to victim through cross-coupling capacitances.
From equations (1) – (5), we get

$$v_1(t) = (R_{gnd1} \parallel R_{s1})(S_2C_{12} + S_3C_{13} + S_4C_{14})|V_a'| * \left(1 - e^{\frac{-t}{(C_{10}+C_{12}+C_{13}+C_{14})(R_{gnd1}\parallel R_{s1})}}\right)$$
$$(0 \le t \le T) \qquad (6)$$

Here "T" is the aggressor signal transition time.

From (6), we noticed that the induced voltage is approximately proportional to ($S_2C_{12}+S_3C_{13}+S_4C_{14}$), and can be written as:

$$v_1 = m(S_2C_{12} + S_3C_{13} + S_4C_{14}) \qquad (7)$$

where

$$m = (R_{gnd1} \parallel R_{s1})|V_a'|\left(1 - e^{\frac{-t}{(C_{10}+C_{12}+C_{13}+C_{14})(R_{gnd1}\parallel R_{s1})}}\right) \qquad (8)$$

To minimize the computation for simulation, we approximate the noise as linearly proportional to ($S_2C_{12}+S_3C_{13}+S_4C_{14}$). For an n-wire interconnect system, the $\sum_{0\le j\le N,\, j\ne i}^{N} S_jC_{ij}$ can be named *as effective summation value* for Cross-coupling Capacitances to wire i, denoted as CC_{eff}.

Recalling the definition of threshold capacitance, we know that if the summation value of coupling capacitances between the victim interconnect and the aggressors equals or exceeds the threshold capacitances, C_{th}, a glitch error will be generated. Thus if V_1=V_{pth}, from (7) we have:

$$V_{th} = V_1 = mC_{th} \qquad (9)$$

From (9) and (11), for positive glitch we now have

$$\frac{V_1}{Vp_{gth}} = \frac{(S_2C_{12} + S_3C_{13} + S_4C_{14})}{Cp_{gth}} = \frac{CC_{eff}}{Cp_{gth}} = CR_{pg} \qquad (10)$$

Here the C_{pgth} is the threshold capacitance for positive glitch, and CR_{pg} stands for *capacitance ratio* for positive glitch. Thus we can use CR_{pg} as a criterion to determine whether the noise is large enough or not. For example, assuming the cross-coupling capacitances among wire1, 2, 3 and 4 are C_{12}, C_{13}, C_{14}, C_{23}, C_{24} and C_{34} respectively, and the input vectors are V:0010->0101, we have:

$$\frac{V_1}{V_{pgth}} = \frac{(C_{12} - C_{13} + C_{14})}{C_{pgth}} = CR_{pg} \qquad (11)$$

If CR_{pg} is equal to or greater than 1, and the input signal on the victim is steadily low, a positive glitch error effect should be generated on victim 1, since the crosstalk induced voltage is large enough. Similarly, for other error effects, such as negative glitch, if CR_{ng} is equal to or less than –1, and the input signal is steadily high, a negative glitch error effect should be generated on the victim.

Input Vectors	CR	Error Effect
"0"→"0"	$CR_{pg} \ge 1$	g_p: "0"→"1" →"0"
"1" →"1"	$CR_{ng} \le -1$	g_n: "1"→"0" →"1"
"0"→"1"	$CR_{rd} \le -1$	d_r: "0"→"1" delayed
"1" →"0"	$CR_{fd} \ge 1$	d_f: "1" →"0" delayed

Table 1 - Input vectors, CR values, and corresponding error effects of high-level interconnect DSM error model

In general, given the input vectors for interconnect, the DSM error model calculates the capacitance ratios for each wire. Based on cross-coupling capacitances and input vector pairs, the DSM error model generates corresponding digitized error effects.

4. Validation of the DSM Error Model

The high-level interconnect DSM error model has been validated by comparing the outputs of the interconnect DSM error model with HSPICE results. Defects are generated by randomly perturbing the cross-coupling capacitances in the RC network of interconnects under test. HSPICE simulation and the DSM error model are then each used to evaluate the given set of vectors. By comparing the outputs of HSPICE and interconnect DSM error model, we can evaluate the accuracy of the high-level interconnect DSM error model.

Margins	C_{13}(pf)	C_{23}(pf)	C_{34}(pf)	C_{35}(pf)	C_{36}(pf)	C_{th}(pf)	Δt
0%	0.2	0.3	0.3	0.2	0.098	1.098	1.30E-11
5%	0.21	0.315	0.315	0.21	0.1029	1.1529	1.37E-11
10%	0.22	0.33	0.33	0.22	0.1078	1.208	1.42E-11
15%	0.23	0.345	0.345	0.23	0.1127	1.2627	1.40E-11

Table 2 – Definition of delay error (Δt) and C_{th} for different design margins

To compare the defect simulation results of interconnect DSM error model with HSPICE, we need V_{th}, T and Δt values of the error definitions and corresponding C_{th}. Let us consider how these values can be defined, and take delay error as an example. For a given design, the receiver determines the V_{th} and T. Since process variation may lead to increased coupling capacitance, designers may leave corresponding slack time and noise margin, which is named as *design margin*, to overcome the extra delay and noise induced by cross-coupling capacitances. As shown in Table 2, for a given design margin, we perturb the cross-coupling capacitances for wire 3, in such a way that they reach the design margin's limitation. The vector applied will activate all the cross-coupling capacitors collectively to generate worse case delay on wire 3. Hence, the summation value of the perturbed cross-coupling capacitances equals C_{th} exactly. After simulating the circuit with these cross-coupling capacitances, we can find out the largest tolerable delay (Δt) and the corresponding threshold capacitance (C_{th}) for wire 3. Table 2 shows for design margin from 0% to 15%, the error definitions (Δt), and threshold capacitances (C_{th}).

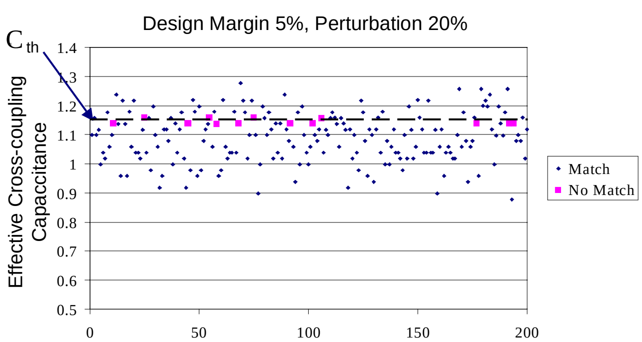

Figure 7- Comparison of interconnect DSM error model outputs and HSPICE results

Based on the definition of error (Δt) in Table 2, we can determine whether a crosstalk error occurs from the HSPICE simulation. The interconnect DSM error model is used with threshold capacitance (C_{th}) obtained from Table 2. By comparing the outputs of the interconnect DSM error model with the HSPICE simulation, we evaluated the accuracy of the interconnect DSM error model.

Design Margin 5%					
	Perturbation Range				
Vector	10.0%	15.0%	20.0%	25.0%	30.0%
Vector0(++++++)	100.0%	100.0%	99.5%	100.0%	100.0%
Vector1(+++0++)	100.0%	99.0%	96.0%	96.0%	91.0%
Vector2(+++-++)	100.0%	100.0%	99.0%	98.0%	96.0%
Vector3(++-+-+)	100.0%	100.0%	100.0%	100.0%	100.0%
Vector4(+++++0)	100.0%	100.0%	100.0%	99.0%	97.0%

Table 3-Matching percentage between HSPICE result and DSM error model for design margin 5%

We simulated 6-wire interconnects for different design margins, with different cross-coupling defect perturbation ranges. Figure 7 shows the comparison of HSPICE with the DSM error model. The X-axis denotes different defect perturbations, and the Y-axis denotes the effective cross-coupling capacitance CC_{eff} for each perturbation. Diamonds indicate the perturbations where the results of the DSM error model matched with HSPICE, and the large rectangles indicate where results did not match. The vectors used in this simulation were "000100-111011", and design margin was 5%. All the cross-coupling capacitances were randomly perturbed, by using uniform distribution within 80%~120% (20% perturbation) of the original values. The horizontal line in Figure 7 shows the threshold capacitance (C_{th}). From Figure 7, we notice that when CC_{eff} of the cross-coupling defects is far from C_{th} (larger or smaller), the DSM error model is accurate. The accuracy of the DSM error model is compromised when CC_{eff} of the crosstalk defect is very close or equal to C_{th}, because of the intrinsic estimation error introduced by the approximation in the interconnect DSM error model.

Design Margin 10%					
	Perturbation Range				
Vector	10.0%	15.0%	20.0%	25.0%	30.0%
Vector0(++++++)	100.0%	100.0%	100.0%	100.0%	100.0%
Vector1(+++0++)	100.0%	99.0%	98.0%	93.0%	91.0%
Vector2(+++-++)	100.0%	99.0%	96.0%	93.0%	87.0%
Vector3(++-+-+)	100.0%	100.0%	100.0%	100.0%	100.0%
Vector4(+++++0)	100.0%	100.0%	100.0%	100.0%	99.0%

Table 4 - Matching percentage between HSPICE result and DSM error model for design margin 10%

Design Margin 15%					
	Perturbation Range				
Vector	10.0%	15.0%	20.0%	25.0%	30.0%
Vector0(++++++)	100.0%	100.0%	100.0%	100.0%	100.0%
Vector1(+++0++)	100.0%	100.0%	100.0%	100.0%	99.0%
Vector2(+++-++)	100.0%	100.0%	100.0%	100.0%	100.0%
Vector3(++-+-+)	100.0%	100.0%	100.0%	100.0%	100.0%
Vector4(+++++0)	100.0%	100.0%	100.0%	100.0%	100.0%

Table 5- Matching percentage between HSPICE result and DSM error model for design margin 15%

We have simulated the interconnect RC network with different cross-coupling defect perturbation range for different design margins. Table 3 shows the percentage of cases where the HSPICE and interconnect DSM error model gave matching results. This simulation is conducted with a design margin 5%, and a perturbation range 10%~30% (process parameters can vary as much as +/- 30% [18]), and vectors V0~V4. Table 4 and Table 5 shows the matching percentage of the simulations with design margin 10% and 15% respectively.

The simulation results show that for different cross-coupling capacitance perturbation ranges and design margins, in almost all the defect simulations, the outputs of interconnect DSM error model match with the HSPICE simulation result, hence demonstrating the accuracy of the high-level interconnect DSM error model.

5. Applications

We have applied the high-level crosstalk defect simulation methodology to: 1) validate a self-test methodology for crosstalk of SoC interconnects; 2) evaluate crosstalk defect coverage for an existing test set. Both applications are based on a digital signal processor (DSP) circuit, CMU DSP [19], which is shown in Figure 8. The CMU DSP consists of components such as Address Generate Unit (AGU), Program Control Unit (PCU), Bus Switch and Arithmetic Logic Unit (ALU). These components communicate through a set of address buses and data buses, such as XAB, YAB, PAB, XDB, YDB, PDB and GDB.

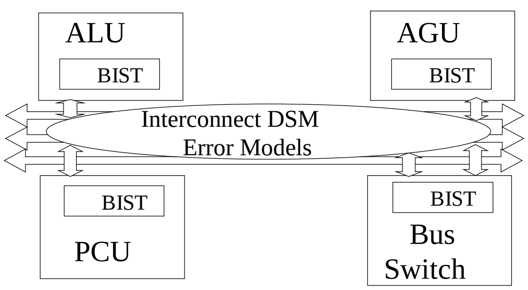

Figure 8 – CMU DSP with embedded self-test structures and high-level interconnect DSM error models

5.1 Validation of a Crosstalk Self-Test Methodology

To enable at-speed testing of SoC interconnects, a self-test methodology was proposed [16], which uses on-chip test generator (TG) to generate MA test and uses on-chip error detector (ED) to analyze interconnect responses. Figure 8 shows the DSP circuit with self-test structures inserted. Test generators are inserted in each core, behind the bus interface, to generate MA test vectors, and error detectors are inserted in each core, behind the bus interface, to detect possible crosstalk errors. A test controller is used to initiate and manage test transactions by activating the appropriate test generators and error detectors. To validate the self-test methodology, interconnect DSM error models are substituted for each bus. Figure 8 shows the CMU DSP with appropriate self-test structures inserted for each bus under test.

We have evaluated the effectiveness of the self-test methodology using the proposed high-level crosstalk defect simulation technique. For each bus under test, randomly perturbed coupling capacitances were generated. Using the high-level defect simulation methodology, the crosstalk self-test methodology has been simulated and validated. As expected, 100% defect coverage is obtained when the crosstalk self-test methodology is simulated.

5.2 Evaluation of Crosstalk Defect Coverage for Existing Tests

The high-level crosstalk defect simulation methodology can be used to evaluate crosstalk defect coverage for existing test sets. This is useful to identify if the different tests, which typically exist for a chip, can be combined and used to provide the desired crosstalk defect coverage before developing any new crosstalk test. As shown in Figure 9, we implemented logic BIST in CMUDSP to generate test vectors for stuck-at faults. Logic BIST generates test vectors on-chip using a Linear Feedback Shift Register (LFSR). The outputs of the LFSR are connected to the scan chains through a phase shifter. The outputs of the scan chains are compressed on-chip using a Multiple Input Shift Register (MISR).

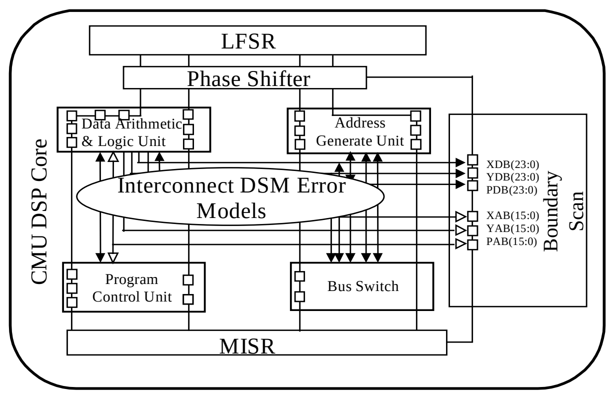

Figure 9 – Application to evaluate crosstalk defect coverage of existing test set

During a normal logic BIST test, the on-chip buses experience signal activities, due to the application of the logic BIST test to logic components, and due to the responses generated by logic components. The resulting transitions to buses can act as test vectors for crosstalk defects on those buses.

To evaluate the crosstalk defect coverage of the logic BIST test, each bus in the DSP is replaced by the interconnect DSM error model, and high-level defect simulation is conducted with randomly generated crosstalk defects. Defects are generated by perturbing the cross-coupling capacitances using uniform distribution and gaussian distribution. Table 6 shows the defect coverage on the different buses, with defects generated by using uniform distribution with different perturbation values. Table 7 shows the defect coverage with cross-coupling capacitances perturbed by using gaussian distribution with different standard deviations. The results show that tests generated by the logic BIST have very high crosstalk defect coverage on the PAB bus, and moderately high coverage on XDB and YDB. On the other hand, crosstalk defect coverage on other buses is low, so additional tests need to be developed for these buses.

Perturb range	PAB	PDB	XAB	XDB	YA B	YDB	GDB
20%	99.5%	0%	1%	47.5%	0	44.5%	1.5%
30%	100%	0..5%	1.5%	74%	1%	67%	2.5%

Table 6 - Crosstalk defect coverage of logic BIST generated tests on the buses (perturbation using uniform distribution)

Standard deviation	PAB	PDB	XAB	XDB	YAB	YDB	GDB
0.2	99%	0.5%	1%	55%	0.5%	46%	1.5%
0.3	100%	1%	2%	74%	2.3%	65%	2%

Table 7 - Crosstalk defect coverage of logic BIST generated tests on the buses (perturbation using gaussian distribution)

The above two applications demonstrate the usefulness of the high-level crosstalk defect simulation technique. It can be used to validate new crosstalk test methodologies or evaluate the crosstalk defect coverage of existing tests. It provides designers a fast way to evaluate different tests and thereby determine low cost crosstalk test methods.

6. Conclusion

In this paper, we have presented a methodology that enables simulation of crosstalk defects in SoC interconnects at high level, instead of having to resort to the time consuming, and sometimes infeasible SPICE-level simulation. The high-level simulation methodology is based on a novel high-level DSM error model. As demonstrated, the proposed methodology is fast and accurate, and can be used to evaluate the effectiveness of crosstalk test methods as well as the defect coverage of existing tests, thereby leading to low-cost crosstalk test techniques.

Acknowledgment

We would like to thank Sreejit Chakravarty for helpful discussions and suggestions in developing the ideas in this paper.

References

[1] H. Zhou, and D. F.Wang, “Global Routing with Crosstalk Constaints”, *Proceedings 1998 Design and Automation Conference 35th DAC*, pages 374-377, 1998.

[2] Z. Chen, and I. Koren, “Crosstalk Minimization in Three-Layer HVH Channel Routing”, *Proceeding IEEE International Symposium on Defect and Fault Tolerance in VLSI System*, pages 38-42, 1997.

[3] K. L. Shepard, “Design Methodologies for Noise in Digital Integrated Circuits”, *Proceedings Design Automation Conference*, pages 94-99, 1998.

[4] T. Stohr H. Alt, A. Hetzel and J. Koehl, “Analysis, reduction and avoidance of crosstalk on VLSI chips’, in Proc. *Int. Symp. Physical Design*, pages 211-218, 1998.

[5] K. Rahmat, J. Neves, J.Lee, “Methods for calculating coupling noise in early design: a comparative analysis”, *Proceeding International Conference on Computer Design VLSI in Computers and Processors*, pages 76-81, 1998.

[6] H. Kawaguchi and T. Sakurai, “Delay and Noise Formulas for Capacitively Coupled Distributed RC Lines”, Proceedings of the Asian and South Pacific Design Automation Conference, Pages 35-43, 1998.

[7] A. Vittal and M. Marek-Sadowska, “Crosstalk Reduction for VLSI”, *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 16, No. 3, pages 290-298, 1997.

[8] P. Nordholz, D. Treytnar, J. Otterstedt, H. Grabinski, D. Niggemeyer, and T. W. Williams, “Signal Integrity Problems in Deep Submicron arising from Interconnects between Cores”, *Proceedings IEEE VLSI Test Symposium*, pages 28-33, 1998.

[9] M. Cuviello, S. Dey, X. Bai, Y. Zhao, “Fault Modeling and Simulation for Crosstalk in System-On_chip Interconnects”, *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, pages 297-303, 1999.

[10] A. Rubio, N. Itazaki, X, Xu, and K. Kinoshita, “An approach to the analysis and detection of cross talk faults in digital VLSI circuits”, *IEEE Trans. CAD of Int. Cir. Sys.*, Vol. 13, no. 3, pp. 387-395, 1994.

[11] K. T. Lee, C. Nordquist, and J. A. Abraham, “Automatic test pattern generation for crosstalk glitches in digital circuits”, *Proceeding VLSI Test Symposium*, pp. 34-39, 1998.

[12] W. Chen, S. Gupta, M. Breuer, “Test generation for crosstalk-induced delay in integrated circuits”, *Proceeding International Test Conference*, pp. 191-200, 1999.

[13] F. Moll and A. Rubio, “Methodology of detection of spurious signals in VLSI circuits”, *Proceeding Europe Test Conference*, pp. 491-496, 1993.

[14] W. Y. Chen, S. K. Gupta, and M. A. Breure, “Test Generation in VLSI circuits for crosstalk noise”, *Proceeding International Test Conference*, 641-650, 1998.

[15] A. Sinha, S. K. Gupta and M.A. Breuer, “Validation and test generation for oscillatory noise in VLSI interconnects”, *Proceeding International Conference on Computer-Aided Design (ICCAD)*, 1999.

[16] X. Bai, S. Dey, J. Rajski, “Self-Test Methodology for At-Speed Test of Crosstalk in Chip Interconnects”, *Proceedings of Design Automation Conference 2000*.

[17] Y. Zhao, S. Dey, “Fault Coverage Analysis Techniques of Crosstalk in Chip Interconnects”, *Proceeding of International Test Conference*, 2000.

[18] B. Choi, D.M. H. Walker, “Timing Analysis of Combinational circuit Including Capacitive Coupling and Statistical Process Variation”, Proceedings 18th *IEEE VLSI Test Symposium*, 2000.

[19] C. Inacio, The CMU DSP Team, “The Carnegie Mellon Synthesizable Digital Signal Processor Core”, 1999.