# High-Level Crosstalk Defect Simulation Methodology for System-on-Chip Interconnects

Xiaoliang Bai and Sujit Dey

***Abstract*—For system-on-chips (SoC) using nanometer technologies, buses and long interconnects are susceptible to crosstalk defects that may lead to functional and timing failures. Testing for crosstalk defects is becoming important to ensure error-free operation of an SoC. To efficiently evaluate crosstalk-defect coverage of existing tests and facilitate the development of new crosstalk test methodologies, effective crosstalk-defect coverage-analysis techniques are needed. In this paper, we present an efficient high-level crosstalk-defect simulation methodology for interconnects dominated by capacitive coupling effects. A novel coupling defect-simulation model was developed and implemented in hardware description languages. The high-level crosstalk-defect simulation methodology was examined by SPICE simulations. Experimental results show the crosstalk defect simulation methodology efficiently provides high-fidelity defect-coverage results. The proposed methodology enables fast exploration and evaluation of different tests, leading to high-quality, low-cost manufacturing tests for crosstalk-induced ac failures.**

***Index Terms*—Crosstalk, defect, interconnect, simulation, system-on-chip (SoC).**

## I. Introduction

Due to large coupling to ground capacitance ratio, gigahertz frequencies, and dense interconnect, coupling capacitance between interconnects can significantly affect circuit performance and even cause a chip to malfunction [1], [2]. On-chip interconnect is becoming a critical determinant for the performance and reliability of high-frequency low-power system-on-chip (SoC) designs. Several design techniques [3]–[6] and analysis techniques [7]–[9] have been developed to minimize signal integrity problems in various design phases. However, process variations together with manufacturing defects may lead to an unexpected increase in crosstalk noise. Process variations and manufacturing defects that can cause crosstalk errors are *crosstalk defects*. Since it is impossible to accurately predict the occurrences of crosstalk defects in nanometer circuits, testing for crosstalk defects is becoming essential to ensure error-free operation of SoCs.

Previous works have shown that crosstalk effects are significant in long interconnects [1], [10]. Therefore, it is important to develop testing solutions and defect-coverage analysis methodologies targeting crosstalk errors on global interconnects. Several testing methodologies and fault models have been developed for crosstalk errors in gate-level interconnect [11]–[14]. For global interconnect, a behavior-level fault model, maximal aggressor (MA) fault model, was developed [10]. Frequently, it is difficult to apply tests to on-chip interconnects directly from chip input/outputs (I/Os), since those system-level interconnects are embedded between various cores. Additionally, crosstalk tests need to be applied at operational speed, which may require expensive external testers. To achieve high-quality at-speed testing of on-chip interconnects, a self-test method was proposed that uses embedded built-in self-test (BIST) structures to generate tests for on-chip interconnects [15]. However, the insertion of the BIST structures introduces area and delay overhead. A less expensive alternative is to reuse legacy tests, such as functional, scan, and BIST to achieve good crosstalk-defect coverage.

In searching and developing an optimal test solution for crosstalk faults, all new and existing tests need to be evaluated and validated. SPICE-based defect simulation provides golden accuracy. However, SPICE-based methods are prohibitively time consuming and are unsuitable for iterative evaluations of large SoCs. An analytical methodology has been developed to evaluate crosstalk-defect coverage for a given test set [16]. It uses hypercubes to represent covering relationships among test patterns. Although faster than SPICE simulation, this method has several limitations. First, the size of the hypercube grows exponentially with the bus width. Second, the test patterns it can evaluate are limited to a subset of test patterns. Third, it assumes that all errors propagated to receivers are observable. All these limitations render this analytical method unsuitable in practice. Therefore, it is critical to develop an alternative method that can efficiently evaluate crosstalk-defect coverage for complex SoC interconnect architectures. The desired crosstalk-defect coverage-analysis method should be fast, so that it can be used iteratively to explore different test solutions. To ensure the accuracy, it should have good correlation with detailed low-level noise simulation methods, such as the SPICE-based defect simulation framework developed in previous work [10].

In this paper, we present a high-level crosstalk-defect simulation methodology that enables fast crosstalk-defect coverage analysis for interconnects dominated by capacitive coupling effects. This methodology uses a novel coupling defect-simulation model for interconnects to achieve fast crosstalk-defect simulation. The coupling defect-simulation model is independent of fault model and test methodology to be evaluated. It estimates noise effects based on coupling defect information and interconnect signal transitions. The coupling defect simulation model is implemented in hardware description languages (HDL) and validated by extensive transistor-level simulations. The proposed crosstalk-defect simulation methodology enables a complex SoC design, including HDL modules for different cores, testing solutions, and physical defects to be simulated in an integrated environment.

The rest of the paper is organized as follows. Section II describes interconnect crosstalk noise effects and their properties. Section III presents an efficient interconnect coupling defect simulation model and the high-level crosstalk defect simulation methodology. Section IV validates the high-level crosstalk defect simulation methodology. Section V concludes this paper.

## II. Interconnect Crosstalk Effects and Properties

The adverse effects of coupling capacitance and inductance on signal integrity can cause timing and logic errors. When coupling capacitance is the first-order parameter between two interconnects, two basic signal anomalies will take place as a result of switching signals on neighboring wires. Fig. 1(a) shows the first case. When one of the wires switches ($\mathrm{Y}_1$ switches from logic "0" to "1") and the other is stable ($\mathrm{Y}_2$ is stable logic "0"), energy will be transferred through the coupling capacitance. This interference causes a glitch generated on the original stable wire ($\mathrm{Y}_2$). In the second case, when the two wires are switching in opposite directions, the transition time will increase, as shown in Fig. 1(b).

When inductance is combined with *RC* elements, damped voltage oscillations will be generated on top of a glitch or delay, as shown in Fig. 1(c). Unlike coupling capacitance, mutual inductance can be

Manuscript received October 11, 2002; revised February 21, 2003 and September 12, 2003. This work was supported by the Semiconductor Research Corporation (SRC) under contract #98-TJ-648. This paper was recommended by Associate Editor K. Chakrabarty.

X. Bai was with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093 USA. He is now with Magma Design Automation, Inc., Santa Clara, CA 95054 USA (e-mail: xibai@ece.ucsd.edu).

S. Dey is with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: dey@ece.ucsd.edu).

Digital Object Identifier 10.1109/TCAD.2004.833612

0278-0070/04$20.00 © 2004 IEEE

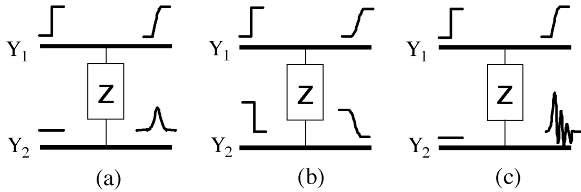


Fig. 1. Three signal anomalies (a) glitch, (b) delay, and (c) oscillations.

a source of long-range noise. Analytical models for *RLC* noise effects have been developed in [17] and [18]. Inductance effect is important for some interconnects [19], [20]. It has been shown in previous studies that inductance effects need to be considered for interconnects of length $l, (t_r/2\sqrt{LC}) < l < (2/R)\sqrt{L/C}$, where $\mathrm{t_r}$ is the signal rise time of the driver. The interconnect must be long enough for the time-of-flight to be comparable to rise time, or the interconnect must be short enough such that attenuation does not eliminate inductive effects [19]. Therefore, although inductance needs to be considered under some circumstances, its effect can be safely approximated by capacitance noise for many on-chip interconnects. In this paper, we focus on crosstalk effects that are caused by capacitive coupling defects and address the problem of fast simulation and test coverage evaluation of such defects.

In this work, six different crosstalk effects are considered, as shown in Fig. 2. When signal on one interconnect is logic "0" such as $\mathrm{Y_2}$ in Fig. 2(a), and most neighboring signals are rising transitions, a glitch toward logic "1" will be generated. We name this noise a *positive glitch* $(\mathrm{g_p})$. Since the signal on $\mathrm{Y_2}$ is affected and may cause an erroneous data on receiver, we refer $\mathrm{Y_2}$ as the *victim* and other interconnects as *aggressors*. Similarly, as shown in Fig. 2(b), a glitch toward logic "0" is generated on $\mathrm{Y_2}$. This error effect is a *negative glitch* $(\mathrm{g_n})$. If a rising transition is delayed because of crosstalk-induced noise, it is a *rising delay* $(\mathrm{d_r})$, as shown in Fig. 2(c). In Fig. 2(d), a falling transition is delayed and the noise effect is a *falling delay* $(\mathrm{d_f})$. Crosstalk-induced noise effects can also speedup signal transitions and cause hold-time violations. Fig. 2(e) and (f) show *rising speedup* $(\mathrm{s_r})$ and *falling speedup* $(\mathrm{s_f})$ error effects, respectively.

In digital circuits, what constitutes an error depends on the receiving logic's characteristics, such as the threshold voltage and sampling window. Not all crosstalk noises lead to errors. Using positive glitch $(\mathrm{g_p})$ as an example, for the receiver to get an erroneous value the noise should be higher than a threshold voltage $(\mathrm{Vp_{th}})$ for a duration time longer than $\Delta\mathrm{t_g}$ around the sampling time $T$, as shown in Fig. 3(a). This means that the glitch should be large, wide enough, and should fall into the sampling window. Here, $\mathrm{Vp_{th}}$ and $\Delta\mathrm{t_g}$ are determined by threshold voltage, setup time, and hold time requirements of the receiver. $V_{\mathrm{gmax}}$ is the magnitude of a glitch noise that just satisfies the positive glitch-error criterions. Fig. 3(b) shows error characterization for the falling delay $(\mathrm{d_f})$. For an erroneous value to be sampled by the receiver, the voltage of the delayed signal should be higher than $\mathrm{Vp_{th}}$ at the sampling time $T$ for a time period $\Delta\mathrm{t_d}$. Based on receiver's characteristics, negative glitch $(\mathrm{g_n})$, rising delay $(\mathrm{d_r})$, rising speedup $(\mathrm{s_r})$ and falling speedup $(\mathrm{s_f})$ can also be characterized in a similar way.

For an interconnect wiring network, when all the coupling capacitances are triggered, the total coupling noise is closely related to the sum of all the coupling capacitance values [10].

- *When driver strength is sufficiently high, noise effect is strongly related with the summation value of coupling capacitances. The differences of noise effects become negligible for cases that have the same summation value, but different distributions of coupling capacitances among aggressors*. This can be attributed to the fact that in an ideal case, with zero driver resistance, the resulting distribution is similar to a lumped sum equivalent of the parallel combination of all coupling capacitances.
- *Monotone property: Glitch and delay effects increase monotonically as the total coupling capacitance value increases while other parameters remain the same*. This can be attributed to the fact that noise current injected into the victim is proportional to the value of coupling capacitance. When all coupling capacitances are triggered to contribute to noise generation, the total noise current will increase with the increasing of total coupling capacitance value.

Based on the above properties, a *threshold capacitance* $(C_{\mathrm{th}})$ can be derived for a corresponding type of crosstalk fault. When the summation value of coupling capacitances exceeds $C_{\mathrm{th}}$, it is possible to cause an error on the receiver if sufficient coupling capacitances are successfully triggered by input vectors [10].

## III. High-Level Crosstalk-Defect Simulation Methodology

The goal of developing a high-level crosstalk defect simulation methodology is to achieve fast crosstalk-defect coverage analysis for global interconnects and enable efficient exploration of different test methodologies. In this section, we propose a high-level crosstalk-defect simulation methodology. It uses a simple yet effective coupling defect-simulation model, which is implemented in HDL. Due to limitations of HDL simulators, noise estimation has to be simple. Yet, it has to be sufficiently accurate for defect-coverage analysis purpose. To achieve these goals, an efficient noise-estimation mechanism is developed.

### A. Interconnect Coupling Defect-Simulation Model

In this section, we use a four-wire interconnect system as an example to explain the crosstalk-defect estimation mechanism used in the coupling defect-simulation model. For simplicity, we assume that bus drivers are balanced and have sufficient strength. Therefore, they can be accurately modeled by saturated ramp voltage sources and linear driver resistances. The accuracy of the model will be evaluated in the next section.

Fig. 4 shows a four-wire interconnect system. To illustrate the noise estimation process, a positive glitch error on wire 1 will be analyzed. In coupling defect simulation, noise estimation will be performed for each wire. However, in this example, wire 1 is the wire-under-concern (the victim) and wires 2–4 are the aggressors. $V_{1\mathrm{in}}$, $V_{2\mathrm{in}}$, $V_{3\mathrm{in}}$, and $V_{4\mathrm{in}}$ are voltage sources of the drivers. $V_1$, $V_2$, $V_3$, and $V_4$ are voltages at the receiver end of wires 1–4, respectively. $C_{10}$, $C_{20}$, $C_{30}$, and $C_{40}$ are wire and load capacitances. $R_{\mathrm{line1}}$, $R_{\mathrm{line2}}$, $R_{\mathrm{line3}}$, and $R_{\mathrm{line4}}$ are wire resistances. We assume that for buses, the drivers are strong, balanced, and all have the same slew rate for rising and falling transitions. The switching drivers can be characterized as follows:

$$\frac{dv_2}{dt} = S_2\left|V_a'\right|$$
$$\frac{dv_3}{dt} = S_3\left|V_a'\right|$$
$$\frac{dv_4}{dt} = S_4\left|V_a'\right| \tag{1}$$

where $|V_a'|$ is the absolute value of the slew rate and $S$ is a transition direction factor. $S$ is " $+1$" for rising transitions, " $-1$" for falling transitions and "0" for stable signals.

Aggressors inject noise current into the victim wire. The current flow from wire 2 to 1 is

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

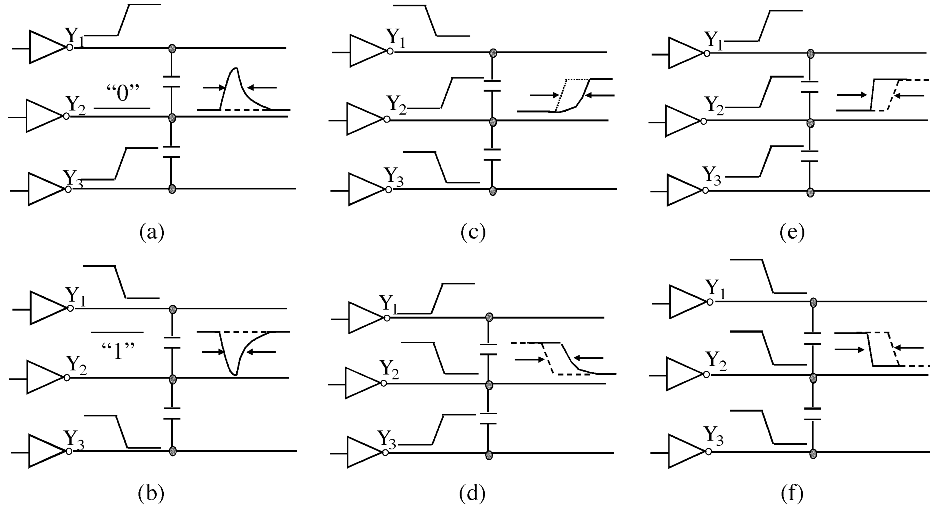


Fig. 2. Error effects induced by coupling capacitances. (a) Positive glitch. (b) Negative glitch. (c) Rising delay. (d) Falling delay. (e) Rising speedup. (f) Falling speedup.

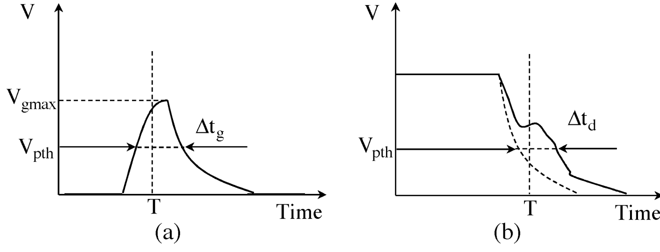


Fig. 3. Characterization of errors: (a) positive glitch and (b) falling delay.

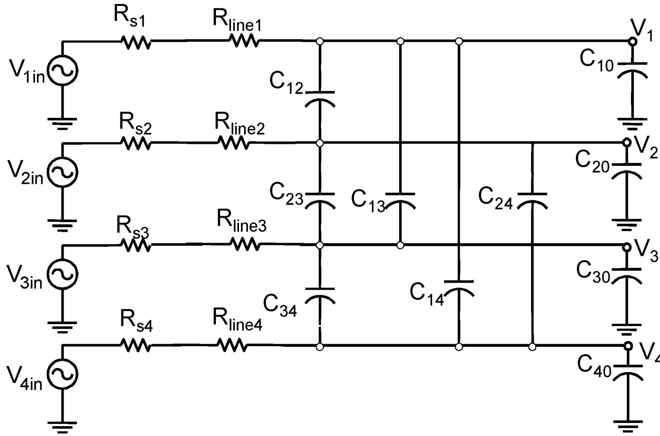


Fig. 4. An example of 4-wire interconnect system.

and currents from the victim wire to ground are

$$i_g = \frac{v_1 - v_{1\text{in}}}{R_{\text{line}1} + R_{s1}} \tag{3}$$

$$i_{10} = C_{10}\frac{dv_1}{dt}. \tag{4}$$

Based on Kirchoff's current law (KCL), we know that

$$i_{10} + i_g - i_{21} - i_{31} - i_{41} = 0. \tag{5}$$

For a positive glitch, the input of the victim does not change and $v_{1\text{in}} = 0$. Input signals on the aggressors affect the victim signal through the coupling capacitances.

From (1)–(5), we have

$$v_1(t) = (R_{\text{line}1} + R_{s1})(S_2C_{12} + S_3C_{13} + S_4C_{14})\left|V_a'\right|^* \times \left(1 - e^{\frac{-t}{(C_{10}+C_{12}+C_{13}+C_{14})(R_{line1}+R_{s1})}}\right)_{(0\leq \text{t} \leq \text{tsettle})} \tag{6}$$

where $\text{t}_{\text{settle}}$ is the aggressor's switching duration time.

From (6), we know that the magnitude of $v_1$ is proportional to $(S_2C_{12} + S_3C_{13} + S_4C_{14})$. Hence, it can be written as

$$v_{1_{\max}} = m * (S_2C_{12} + S_3C_{13} + S_4C_{14}) \tag{7}$$

where

$$m = (R_{\text{line}1} + R_{s1})\left|V_a'\right|\left(1 - e^{\frac{-T}{(C_{10}+C_{12}+C_{13}+C_{14})(R_{\text{line}1}+R_{s1})}}\right). \tag{8}$$

For an $n$-wire interconnect system, $\sum_{0\leq j\leq N, i\neq j}^{N} S_jC_{ij}$ is an effective coupling capacitance $(\text{CC}_{\text{eff}})$ of the wire $i$ under a given pair of input vectors. From the definition of the threshold capacitance, we know that if the summation value of the coupling capacitances between the victim and aggressors is equal to or larger than the threshold capacitance $C_{\text{th}}$, a glitch error will occur. Thus, if $V_{1\max} = V_{g\max}$, then from (7) we have

$$V_{g\max} = m^*(C_{12} + C_{13} + C_{14}) = mC_{\text{pg}-\text{th}}. \tag{9}$$

From (7) and (9), we can compare the estimated noise with threshold voltage by

$$\frac{V_{1\max}}{V_{g\max}} = \frac{(S_2C_{12} + S_3C_{13} + S_4C_{14})}{C_{\text{pg}-\text{th}}} = \frac{\text{CC}_{\text{eff}}}{C_{\text{pg}-\text{th}}} = \text{CR}_{\text{pg}} \tag{10}$$

$C_{\text{pg}-\text{th}}$ is the threshold capacitance of positive glitch and $\text{CR}_{\text{pg}}$ stands for positive glitch's *effective capacitance ratio*, which is the ratio of effective capacitance to threshold capacitance. Based on the monotone property, we know that the larger the triggered coupling capacitance

TABLE I
ERROR CRITERION FOR DIFFERENT CROSSTALK ERRORS

| Crosstalk Error | Input (wire-under-concern) | Error Criteria | Digitized Error Effect |
|---|---|---|---|
| Positive Glitch | "0"→"0" | $CR_{pg} \geq 1$ | $g_p$: "0"→"1" →"0" |
| Negative Glitch | "1" →"1" | $CR_{ng} \leq -1$ | $g_n$: "1"→"0" →"1" |
| Rising Delay | "0"→"1" | $CR_{rd} \leq -1$ | $d_r$: "0"→"1" delayed |
| Falling Delay | "1" →"0" | $CR_{fd} \geq 1$ | $d_f$: "1" →"0" delayed |
| Rising Speedup | "0" →"1" | $CR_{sr} \geq 1$ | $s_r$: "0" →"1" speedup |
| Falling Speedup | "1" →"0" | $CR_{rd} \leq -1$ | $s_f$: "1" →"0" speedup |

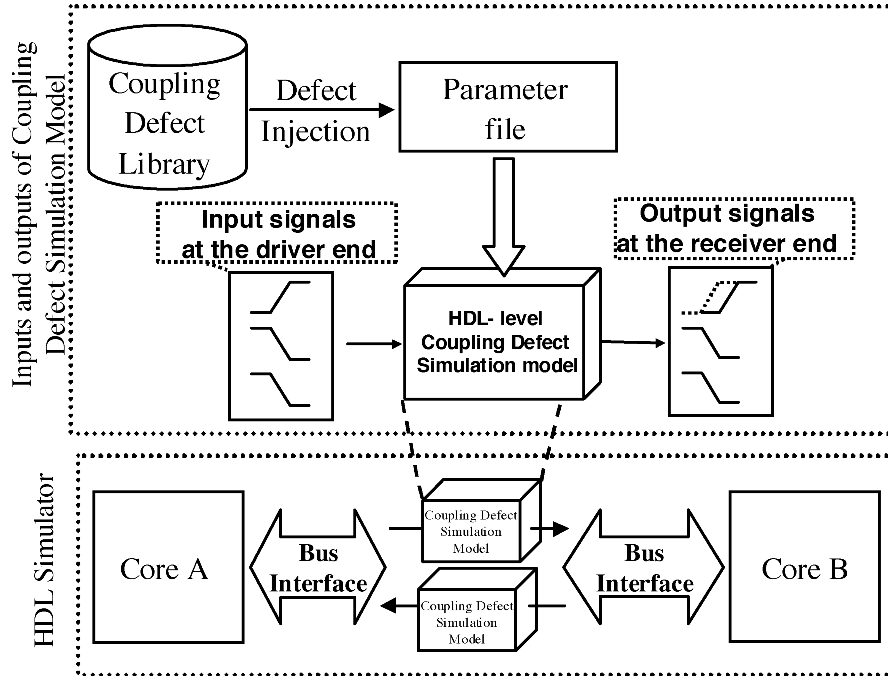


Fig. 5. High-level interconnect coupling defect simulation.

is, the larger a crosstalk noise will be generated. Hence, we do not explicitly check the crosstalk error conditions of $\Delta t_g$ and $V_{th}$. Instead, $CC_{eff}$ is compared with the threshold capacitance $C_{pg-th}$. $CR_{pg}$ is used as a criterion in determining whether the noise is large enough to cause an error. For example, suppose the coupling capacitances between wire1–4 are $C_{12}$, $C_{13}$, $C_{14}$, $C_{23}$, $C_{24}$, and $C_{34}$, and the input pattern is V: 0010 → 0101, then we have

$$CR_{pg} = \frac{(C_{12} - C_{13} + C_{14})}{C_{pg-th}}. \tag{11}$$

If $CR_{pg}$ is equal to or larger than 1 and the original driving signal on the victim is stable logic "0," a positive glitch error effect will occur on the victim's receiver. Similarly, other crosstalk error effects can be estimated and generated. Table I shows the error criteria and digitized error effects for all types of crosstalk errors considered in this interconnect coupling defect simulation model.

*B. High-Level Defect Simulation Methodology*

The interconnect coupling defect simulation model is implemented in HDL and a high-level coupling defect-simulation methodology is developed. For a set of interconnects, an appropriate number of coupling defect-simulation models will be inserted to replace the original interconnect signals. The coupling defect simulation model estimates noise effects and generates digitized crosstalk errors. The high-level defect-simulation methodology enables various test solutions to be evaluated and simulated together with register-transfer level (RTL) and gate-level models in a complex SoC

Fig. 5 shows an example of the high-level coupling defect-simulation methodology. All models are simulated in a commercial HDL simulator. Coupling defect-simulation models are used for a bidirectional bus between cores A and B. For different communication directions of the bus, interconnects have different sets of drivers and receivers. Hence, two coupling defect-simulation models are needed. The upper part (surrounded by the dotted line) of Fig. 5 shows the inputs and outputs of a coupling defect-simulation model. Each coupling defect-simulation model is configured by a physical parameter file, which contains information about coupling capacitances and $C_{th}$ values of all faults for each receiver. SPICE simulation is used to calibrate the $C_{th}$ values. Coupling defects can be pregenerated and stored in a defect library. During defect simulation, defects are injected by perturbing coupling capacitance values in the parameter file.

In the HDL-simulation environment, each coupling defect-simulation model is a process that monitors input events on the corresponding bus interface. For bidirectional buses, two coupling defect-simulation processes will be needed. The flowchart of an HDL coupling defect-simulation process is shown in Fig. 6(a). When any input event occurs, the coupling defect simulation process will be activated. For an $N$-wire bus, this process estimates noise for each of the $N$ wires, one wire at a time. The noise effects are evaluated and digitized. After all the

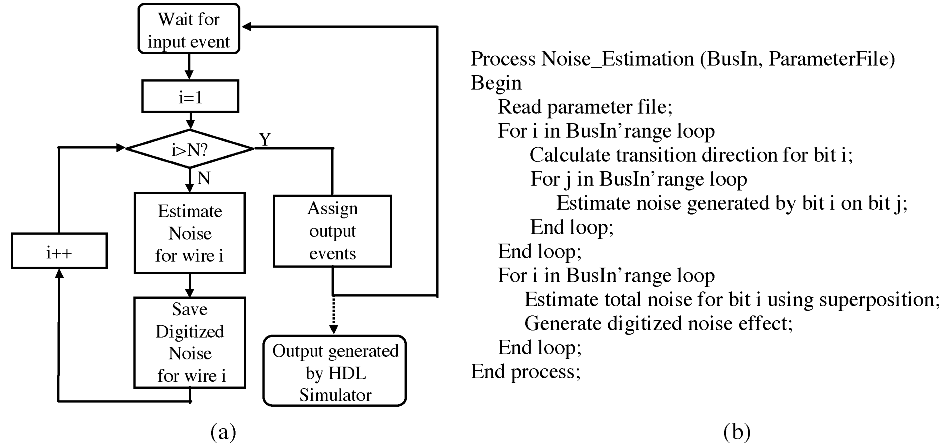


Fig. 6. Flowchart of HDL coupling defect simulation process. (a) Flow chart. (b) Pseudocode.

TABLE II
SLACK $(\Delta t)$ AND $C_{th}$ FOR DIFFERENT DESIGN MARGINS

| Margins | $C_{12}$(pf) | $C_{23}$(pf) | $C_{34}$(pf) | $C_{35}$(pf) | $C_{36}$(pf) | $C_{th}$(pf) | Slack ($\Delta$t) |
|---|---|---|---|---|---|---|---|
| 0% | 0.20 | 0.300 | 0.300 | 0.20 | 0.0980 | 1.0980 | 1.30E-11 |
| 5% | 0.21 | 0.315 | 0.315 | 0.21 | 0.1029 | 1.1529 | 1.37E-11 |
| 10% | 0.22 | 0.330 | 0.330 | 0.22 | 0.1078 | 1.2080 | 1.42E-11 |
| 15% | 0.23 | 0.345 | 0.345 | 0.23 | 0.1127 | 1.2627 | 1.40E-11 |

TABLE III
MATCHING PERCENTAGE BETWEEN HSPICE RESULT AND DEFECT-SIMULATION MODEL (DESIGN MARGIN 5%)

| | Design Margin 5% | | | | |
|---|---|---|---|---|---|
| | Perturbation Range | | | | |
| Input Vector Pair | 10.0% | 15.0% | 20.0% | 25.0% | 30.0% |
| $V_0$ | 100.0% | 100.0% | 99.5% | 100.0% | 100.0% |
| $V_1$ | 100.0% | 99.0% | 96.0% | 96.0% | 91.0% |
| $V_2$ | 100.0% | 100.0% | 99.0% | 98.0% | 96.0% |
| $V_3$ | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| $V_4$ | 100.0% | 100.0% | 100.0% | 99.0% | 97.0% |

$N$ wires are processed, the digitized error effects are assigned as new events to the HDL simulator. These events will then generate output waveforms on the bus interfaces at the receiver end. The pseudocode of the noise-estimation process is shown in Fig. 6(b).

The high-level simulation methodology needs to be calibrated only once using SPICE with layout-extracted parasitic parameters. After the calibration, crosstalk-defect simulation can be performed iteratively using HDL simulators. A parameterized template of the bidirectional coupling defect-simulation model has been developed that enables simulation of a bus with arbitrary bus-width. The high-level simulation methodology facilitates efficient defect simulations and test-methodology evaluations. Instead of recording test vectors, performing SPICE simulation, and then transferring SPICE outputs back to HDL simulator, the high-level coupling defect-simulation methodology allows software (memory image), hardware (HDL modules like embedded processor), and coupling defects to be simulated in an integrated environment. Hence, the defect-coverage evaluation process speeds up tremendously.

## IV. VALIDATION

The high-level crosstalk-defect simulation methodology has been validated by extensive SPICE simulations. The experiments were conducted on a six-wire bus (with drivers and receivers). We will examine the quality of the coupling defect-simulation model by focusing on crosstalk-induced delay errors on wire 3 in the six-wire bus.

A SPICE netlist was extracted from layout using IC station and Calibre [25]. Threshold capacitances were calibrated by SPICE simulations. Then, both SPICE-based [10] and the proposed coupling defect simulation methods were used to evaluate a set of random vectors.

Since process variations may cause coupling capacitances to be larger than expected, designers may leave some degree of slack to tolerate extra delay and noise. This design tolerance to accommodate process variations and small defects is called *design margin*. Using SPICE simulation, coupling capacitance threshold values can be calibrated for different design margins. In our experiments, test vectors were applied to activate all the coupling capacitances collectively to generate a rising delay on wire 3. The value of the coupling capacitances was incrementally increased until the crosstalk-induced delay becomes large enough to cause an erroneous data on the output of the receiver. Then, the summation value of the perturbed coupling capacitances was used as the $C_{th}$ for the delay error on the receiver. Table II shows the largest tolerable delays $(\Delta t)$, and the corresponding threshold capacitances $(C_{th})$ for different design margins (from 0% to 15%).

To assess the quality of the coupling defect-simulation model, we performed extensive simulation with different design margins, perturbation ranges and random input vectors. Simulations were conducted with design margins from 5% to 15% and perturbation ranges from 10% to 30% (process parameters can vary as much as $\pm 30\%$ [22]).

Four vector pairs were randomly selected as input stimulus. Table III shows the experimental results with a design margin 5%. In the first

TABLE IV
MATCHING PERCENTAGE BETWEEN HSPICE RESULT AND DEFECT-SIMULATION MODEL (DESIGN MARGIN 10%)

| | Design Margin 10% | | | | |
|---|---|---|---|---|---|
| | Perturbation Range | | | | |
| Input Vector Pair | 10.0% | 15.0% | 20.0% | 25.0% | 30.0% |
| $V_0$ | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| $V_1$ | 100.0% | 99.0% | 98.0% | 93.0% | 91.0% |
| $V_2$ | 100.0% | 99.0% | 96.0% | 93.0% | 87.0% |
| $V_3$ | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| $V_4$ | 100.0% | 100.0% | 100.0% | 100.0% | 99.0% |

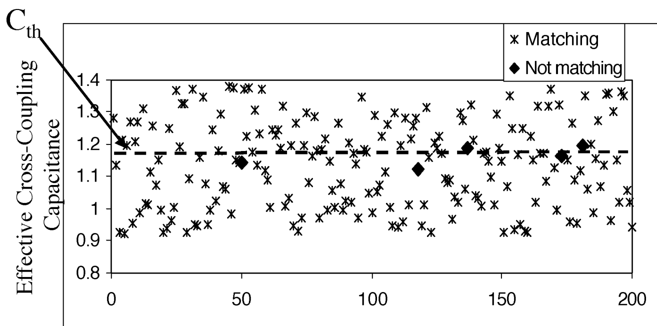


Fig. 7. Comparison of interconnect coupling defect simulation model outputs and HSPICE results.

column of this table, vectors are shown symbolically. "+" represents a rising transition. "−" represents a falling transition and "0" represents a stable signal. On average, 98.8% of the coupling defect-simulation model's outputs match with the SPICE simulation results. Table IV shows the results with a design margin 10%. The average matching rate was 98.2%. Experiments were also conducted for a design margin 15%. The matching rate was 99% for one set of simulations with a perturbation range 30% and vector $v_2$. In all the other experiments the high-level defect simulation model generated same output as the SPICE simulation framework.

The coupling defect simulation model is accurate for defect coverage analysis in most of the randomly generated test cases. However, the high-level coupling defect model uses a lumped RC network to approximate distributed coupling effect of interconnects and uses a linear model to approximate nonlinear drivers. As a result of these approximations, inaccuracy is introduced. Fig. 7 shows the experimental results of 200 randomly generated cases. The $Y$ axis is for the $C_{\rm eff}$ value and $X$ axis is for the different simulation cases. Diamonds represent inaccurate results and stars represent accurate results (matching with SPICE simulation). The horizontal dotted line corresponds to the threshold capacitance of $C_{\rm th} = 1.1529$ pf. It is clear that when the $\rm CC_{eff}$ value of the coupling defects falls into a narrow range (±9% in this case) around the $C_{\rm th}$, the quality of the coupling defect simulation model will become inferior. Nevertheless, for most of the test cases, the coupling defect simulation model is accurate and efficient.

The simulation results show that over a wide range of process variations and design margins, the proposed high-level crosstalk-defect simulation methodology provides accurate and efficient crosstalk defect-coverage analysis.

## V. CONCLUSION

In this paper, we presented an efficient high-level methodology that enables fast crosstalk-defect coverage analysis for system interconnects. Instead of resorting to time-consuming SPICE-level simulations, defect coverage analysis can be performed in an HDL simulation environment. The high-level crosstalk defect simulation methodology has been applied to various test solutions, including validating a BIST methodology for crosstalk errors [15], facilitating the development of a software-based self-test for crosstalk errors on interconnects [23], evaluating crosstalk defect coverage for existing tests and developing a new low-cost test solution [24].

The proposed crosstalk defect simulation methodology is efficient and scalable. It enables software (memory image), hardware (HDL modules), coupling defects and various test solutions (scan, BIST, functional test) to be evaluated and validated in an integrated environment. The high-level coupling defect simulation methodology can evaluate the effectiveness of new crosstalk test methods and existing test sets. It leads to the development of low-cost crosstalk test techniques for complex SoC designs.

## REFERENCES

[1] 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," in *Proc. IEEE VLSI Test Symp.*, 1998, pp. 28–33.
[2] A. Krstic, J.-J. Liou, Y.-M. Jiang, and K.-T. Cheng, "Delay testing considering crosstalk-induced effects," in *Proc. Int. Test Conf.*, Oct. 2001, pp. 558–567.
[3] H. Zhou and D. F. Wang, "Global routing with crosstalk constaints," in *Proc. Design Automation Conf.*, 1998, pp. 374–377.
[4] Z. Chen and I. Koren, "Crosstalk minimization in three-layer HVH channel routing," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, 1997, pp. 38–42.
[5] K. L. Shepard, "Design methodologies for noise in digital integrated circuits," in *Proc. Design Automation Conf.*, 1998, pp. 94–99.
[6] T. Stohr, H. Alt, A. Hetzel, and J. Koehl, "Analysis, reduction and avoidance of crosstalk on VLSI chips," in *Proc. Int. Symp. Physical Design*, 1998, pp. 211–218.
[7] K. Rahmat, J. Neves, and J. Lee, "Methods for calculating coupling noise in early design: A comparative analysis," in *Proc. Int. Conf. Comput. Design VLSI Comput. Process.*, 1998, pp. 76–81.
[8] H. Kawaguchi and T. Sakurai, "Delay and noise formulas for capacitively coupled distributed RC lines," in *Proc. Asian South Pacific Design Automation Conf.*, 1998, pp. 35–43.
[9] A. Vittal and M. Marek-Sadowska, "Crosstalk reduction for VLSI," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 290–298, Mar. 1997.
[10] M. Cuviello, S. Dey, X. Bai, and Y. Zhao, "Fault modeling and simulation for crosstalk in system-on-chip interconnects," in *Proc. Int. Conf. Computer-Aided Design*, 1999, pp. 297–303.
[11] 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. Computer-Aided Design*, vol. 13, pp. 387–395, Mar. 1994.
[12] K. T. Lee, C. Nordquist, and J. A. Abraham, "Automatic test pattern generation for crosstalk glitches in digital circuits," in *Proc. VLSI Test Symp.*, 1998, pp. 34–39.
[13] W. Chen, S. Gupta, and M. Breuer, "Test generation for crosstalk-induced delay in integrated circuits," in *Proc. Int. Test Conf.*, 1999, pp. 191–200.
[14] F. Moll and A. Rubio, "Methodology of detection of spurious signals in VLSI circuits," in *Proc. Eur. Test Conf.*, 1993, pp. 491–496.
[15] X. Bai, S. Dey, and J. Rajski, "Self-test methodology for at-speed test of crosstalk in chip interconnects," in *Proc. Design Automation Conf.*, 2000, pp. 619–224.
[16] Y. Zhao and S. Dey, "Fault coverage analysis techniques of crosstalk in chip interconnects," in *Proc. Int. Test Conf.*, 2000, pp. 492–501.

[17] A. Sinha, S. K. Gupta, and M. A. Breuer, "Validation and test generation for oscillatory noise in VLSI interconnects," in *Proc. Int. Conf. Computer-Aided Design*, 1999, pp. 289–296.
[18] A. B. Kahng and S. Muddu, "An analytical delay model for RLC interconnects," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 1507–1513, 1997.
[19] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Fig.s of merit to characterize the importance of on-chip inductance," *IEEE Trans. VLSI Syst.*, vol. 7, pp. 442–449, Dec. 1999.
[20] A. Deutsch *et al.*, "When are transmission-line effects important for on-chip interconnections?," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1836–1846, Oct. 1997.
[21] *IC Station User's Manual, Version v8.7_3, Mentor Graphics*, 1991–1999.
[22] B. Choi and D. M. H. Walker, "Timing analysis of combinational circuit including capacitive coupling and statistical process variation," in *Proc. 18th IEEE VLSI Test Symp.*, 2000, pp. 49–54.
[23] L. Chen, X. Bai, and S. Dey, "Testing for interconnect crosstalk defects using on-chip embedded processor cores," in *Proc. Design Automation Conf.*, 2001, pp. 317–320.
[24] K. Sekar and S. Dey, "LI-BIST: A low-cost self-test scheme for SoC logic cores and interconnects," *J. Electron. Testing: Theory Applicat.*, vol. 19, no. 2, pp. 113–123, 2003.
[25] *IC Station User's Manual*, 1991–1999.

# IDAP: A Tool for High-Level Power Estimation of Custom Array Structures

Mahesh Mamidipaka, Kamal Khouri, Nikil Dutt, and Magdy Abadir

***Abstract*—While array structures are a significant source of power dissipation, there is a lack of accurate high-level power estimators that account for varying array circuit implementation styles. We present a methodology and a tool, the implementation-dependent array power (IDAP) estimator, that model power dissipation in SRAM-based arrays accurately based on a high-level description of the array. The models are parameterized by the array operations and various technology dependent parameters. The methodology is generic and the IDAP tool has been validated on industrial designs across a wide variety of array implementations in the e500[1] processor core. For these industrial designs, IDAP generates high-level estimates for dynamic power dissipation that are accurate with an error margin of less than 22.2% of detailed (layout extracted) SPICE simulations. We apply the tool in three different scenarios: 1) identifying the subblocks that contribute to power significantly; 2) evaluating the effect of bitline-voltage swing on array power; and 3) evaluating the effect of memory bit-cell dimensions on array power.**

***Index Terms*—Estimation, high-level power estimation, implementation-dependent array power (IDAP), implementation styles.**

Manuscript received July 24, 2003; revised December 15, 2003. This work was done in collaboration with Motorola Inc. and was supported in part by the National Science Foundation under Grant CCR 0203813. This paper was recommended by Associate Editor M. Pedram.

M. Mamidipaka and N. Dutt are with the Center for Embedded Computer Systems, University of California, Irvine, CA 92697 USA (e-mail: maheshmn@cecs.uci.edu; dutt@cecs.uci.edu).

K. Khouri and M. Abadir are with the High-Performance PowerPC Platforms Group, Motorola Inc., Austin, TX 78729 USA (e-mail: kamal.khouri@motorola.com; m.abadir@motorola.com).

Digital Object Identifier 10.1109/TCAD.2004.833609

[1]e500 is the Motorola processor core that is compliant with the PowerPC Book E architecture.

## I. Introduction

Many factors have contributed to the increased demand for lowering power consumption in today's semiconductor designs. Market demand for portable electronics has driven the need for low-power devices, which rely on a battery for operation and, hence, the aim is to increase the lifetime of the battery between recharges. While performance has traditionally been the main driver for high-end desktop and network processors, the need for reducing power consumption has become a serious issue as these devices operate at maximum tolerance levels. For desktop computing, lower yields and higher cooling-system costs increase the overall cost of the system. Similarly, in the network-processor domain, heat-removal systems in a switch farm have a fixed capacity and, hence, a limit is imposed on the number of processors that can be placed on a single board.

Array structures, such as register files, branch-target buffers, tag arrays, and caches consume up to 70% of the overall power in a SoC [4]. It has also been shown that caches alone consume up to 40% of total power [9]. In this paper, we focus on the dynamic power estimation in CMOS-based array structures. Fig. 1 shows the typical design flow for custom memory structures. To meet the stringent power constraints, it is important to obtain power estimates at each level of design hierarchy. As we go down in the design hierarchy, the level of detail in the design increases, leading to more accurate estimates. Although there has been a sizable body of work on power estimation in array structures (Section II summarizes this research), the focus has either been toward modeling at the microarchitectural level or modeling through characterization after the availability of transistor level design. Models at the microarchitectural level lack accuracy because of the nonavailability of design-specific information, such as sense-amplifier type (differential or inverter based), decoder-style type (static CMOS or dynamic CMOS based) etc. On the other hand, models at the transistor level, while accurate, are available only in the latter stages of the design cycle. Hence, there is a need for accurate power models which bridge the gap between the microarchitecture level models and characterization-based models. To the best of our knowledge, this is the first attempt which tries to bridge this gap.

In this paper, we propose a methodology for accurate estimation of power dissipation in CMOS-based arrays using a high-level description of the design, which contains microarchitecture-level parameters and subblock circuit-implementation styles of the array structures. The main contributions of this work are: 1) the ability to represent various organization and implementations of arrays; 2) the ability to abstract parameters which define the power consumption in arrays for a given implementation style; and 3) a methodology to generate accurate power models based on these parameters at a higher level in the design flow. This technology provides designers with the ability to perform a wide variety of tasks, as follows.

- Conduct "what–if" studies on the effects that implementation changes may have on power, without the need to redesign at the transistor-level, and hence, avoid time-consuming SPICE simulations.
- Conduct accurate power-dissipation studies for new process technologies to better understand the impact a new transistor may have on a design.
- Generate highly accurate power models for register transfer level (RTL) design-space exploration.

The remainder of the paper is organized as follows. Section II summarizes the related work in the area of array-power modeling and estimation. Section III provides a brief background in array structures

0278-0070/04$20.00 © 2004 IEEE