# **On Modeling Cross-Talk Faults**

Sujit T Zachariah, Yi-Shing Chang, Sandip Kundu, Chandra Tirumurti Intel Corporation, 3600 Juliette Lane, Santa Clara, CA 95052, USA Contact: Sandip.Kundu@intel.com

### Abstract

Circuit marginality failures in high performance VLSI circuits are projected to increase due to shrinking process geometries and high frequency design techniques. Capacitive cross coupling between interconnects is known to be a prime contributor to such failures. In this paper, we present novel techniques to model and prioritize capacitive cross-talk faults. Experimental results are provided to show effectiveness of the proposed modeling technique on industrial circuits.

# 1. Introduction

In high performance designs, ensuring signal integrity has assumed an importance comparable to timing closure. Aggressive circuit designs such as domino pipeline, self-resetting circuits and cascode pass-transistor logic attain performance at the expense of reduced tolerance to noise. Settling for less than full potential of silicon performance is not an option in today's highly competitive market place. This eliminates the choice of falling back to overly conservative circuit design practices to solve signal integrity problems. Cutting-edge designers must confront signal integrity problems head on without compromising on performance.

Noise has traditionally been treated purely as a design problem. However, non-design issues such as time-to-market factors have prevented complete debug and resolution of all noise violations during the design phase itself. In today's market place, a design may be fabricated in multiple fabrication sites and may be shrunk optically to take advantage of incremental progress in process technology. Even worse, it could be operated at a slightly lower voltage as a low power part or at a slightly higher voltage as a high performance part. Given this market reality, it is neither possible to guarantee that a part will not suffer from signal integrity issues across the entire spectrum of process changes and supply voltage envelope nor is it wise to hold back a design for complete verification. However, even with the time-to-market constraints, the outgoing product quality still needs to be maintained and this has forced a significant change in the testing strategy of VLSI circuits. Conventional testing of VLSI circuits has focused on manufacturing defects, but the above mentioned design trends have resulted in novel testing strategies for failures resulting from noise and circuit marginality issues.

Any phenomenon that causes the voltage of a circuit node that forms the connection between channel-connected components to deviate from its steady state logic value constitutes a source of noise. Often, a minor process change or supply voltage change can trigger signal integrity violations. The following sources of noise in digital circuits are the most critical from the perspectives of frequency of occurrence and severity of magnitude.



#### Figure 1 Wire aspect scaling with technology.

- Capacitive cross-talk noise results from parasitic coupling between adjacent signal nets and is most seen in nets that have weaker drivers than their adjacent peers [2]. With traditional scaling [16], transistors gain in performance and interconnects become more resistive. To mitigate this effect, interconnects are scaled differently in horizontal and vertical dimensions, resulting in dense lateral packing with larger capacitive exposure to adjacent nets (see Figure 1) [9]. With technology scaling, the noise magnitude increases as the drivers of the coupled nets switch faster. At the same time, the traditional tolerance to noise is eroded by the reduction in supply voltage. The combination of these factors results in glitches and signal delays.
- Power supply noise results from difference in voltage reference levels between a local driver and receiver. The receiver may view this difference as input signal noise. This may result in extra signal transition delay or a catastrophic failure. Differ-

ence in power supply level has average, cyclical and transient components. The average difference is often called IR drop. The low frequency difference may be attributed to package inductance while the high frequency component is often attributed to local simultaneous switching.

- *Leakage noise* results from either the discharge (or accumulation) of charge on dynamic circuit nodes (nodes that sometimes get disconnected from the power rails during normal circuit operation and rely on charge stored on the capacitor) or the substrate noise resulting from minority carrier backinjection due to bootstrapping. Leakage noise is more prominent in circuits with lower threshold voltage (typically mandated by lower power supply voltage to maintain drive strength).
- *Charge sharing noise* results from charge re-distribution between weakly held dynamic evaluation nodes and internal nodes of the circuit. With smaller feature size, the significance of this noise is trending sharply upwards.
- Other sources of electrical noise such as *mutual inductance*, *substrate-coupling noise* and *transient noise* due to radiation occur with varying degree of frequency and magnitude.

As is evident, large perturbations to steady state value, resulting from one or more sources of noise can cause functional failures. Hence, testing for failures resulting from such sources of noise is required to ensure functional correctness of VLSI circuits.

In this paper, we restrict our discussion to failures resulting from cross-talk noise and describe a novel method to model and test capacitive cross-talk faults. We introduce a new fault modeling technique, referred to as *Generalized Fault Model (GFM)*, and describe our infrastructure for identification, modeling using GFM, ranking and pruning of cross-talk faults. Pruning of cross-talk fault list is important as only a subset of all the possible (extracted) cross-talk faults for a given circuit can be actually targeted during automatic test pattern generation (ATPG) and/or fault simulation due to resource and time constraints.

This paper is organized as follows. In Section 2, we discuss prior work done on this topic. Section 3 discusses if scan testing is suitable for signal integrity. We state the basic assumptions of our model in Section 4. Section 5 describes our proposed approach for extraction, ranking and modeling of cross talk faults. In Sections 6 and 7, we present our results and conclusions, respectively.

# 2. Previous Work

Prior literature firmly establishes signal integrity induced failures as not just a design problem but a test one as well. The key learnings that emerge from prior literature are:

- **Impact modeling:** The impact of signal integrity problems can be modeled as a *transition delay* or a *signal hazard* [2,8].
- **Impact size modeling**: Efforts in this area involve trying to represent the resulting noise waveforms in a compact representation [2,6,7].
- Qualitative nature of test: Traditional stuck-at fault test is Boolean. A vector is either a test for a fault or it is not. However, in signal integrity testing, several tests for the same fault may have a qualitative difference in terms of severity of impact at the fault site as well as on an observation node [3,4,10].
- Signal integrity impact propagation issue: Research in this area has wrestled with the question of propagating a signal from fault excitation site to an observable point [3,4,12-15,17]. In traditional stuck-at and transition fault testing, the signal that is propagated is the Boolean difference between a fault-free circuit and the faulty circuit. In more analog-like fault models such as resistive bridging faults, an effort was made to estimate the impact of the bridge fault at the site and propagate an analog voltage difference. However, it either involves propagation of signal difference through spice like circuit simulation or techniques based on pre-characterization of cells through simulation where an input noise can easily be translated to an output noise either by table look up or by evaluating expressions that were curve fitted to simulation results during pre-characterization [1,5]. A simplification of the resulting noise waveform was also proposed based on classification to simpler classes and associating symbols with each class of noise value.
- **Capacity/performance vs. accuracy**: There exists a fundamental duality between a comprehensive analysis and how much of that can be performed within the limitation of computations. There are often no clear answers to that question and when they emerge, it is only after comparing several schemes [1].

# 3. Scan Testing for Signal Integrity?

Within the design/test community, there are some who believe that it is meaningful to consider signal integrity testing only in the context of *functional test*. This is a contested issue at Intel as well. The reasons cited are:

**Simultaneous noise sources**: Unlike stuck-at, transition or bridging faults where a single fault model is assumed, can we really assume single fault model for signal integrity related noise sources? Consider this: if the probability of a via failure was one in a billion, the yield of Pentium<sup>®</sup> 4 class chips will be almost zero. Therefore, such failure mechanisms are indeed very rare

<sup>\*</sup> An application for a patent has been filed on a technique described in this paper [11].

and the probability that there will be two of those failures on a single die without causing highly visible catastrophic failure is even rarer. Hence, it is reasonable to assume a single failure model. However, when it comes to signal integrity, when the inputs change, signal integrity is an issue on every switching node. Therefore, it can be easily argued that all signal integrity related faults must be considered all at once rather than one at a time. Since scan mode of testing introduces non-functional states, additional noise may be introduced into the circuit that may not occur in a functional environment. This has direct bearing on yield loss. It is a well-known fact that so called atspeed scan tests are rarely run at the full clock frequency to minimize the yield impact. Therefore another argument goes that, if the tests are not tight, signal integrity related failures may not be detected and therefore the scan test environment is not good for generating signal integrity test.

**DC versus AC analysis:** Signal integrity problems arise when signals are switched in a specific order. For example, if the aggressors of a capacitively coupled line switch one at a time, the impact is not as severe as when they all switch simultaneously. The order of switching is a strong function of the mode of test application. When a scan control signal is distributed through out a chip, it may have different skews than functional clock. Furthermore, in a multi-cycle test, in a given clock the nodes may switch in an order that can never be reproduced based on scan test sequence. Thus the question of whether scan based test can be used to detect signal integrity problems will be a nagging question until definitive answers from experimental results emerge.

**Combination of different type of noise sources:** Even though failure analysis may point to capacitive cross coupling, the impact of other noise sources may be what causes an overthe edge impact condition. Therefore, when we model and target capacitive cross coupling explicitly, it may be some power supply noise that gives the final push to throw it over the edge. This may not be comprehended in the model but may need functional test to show its impact (or the lack of it to avoid yield loss).

Arguments for scan testing: We layer a number of assumptions to model cross-talk noise. The roster includes a single RC extraction point, single fault model to avoid combinatorial explosion and a single measurement condition. So perhaps, scan testing can also enjoy similar benefit from patterns that trigger more conditions.

The modeling work presented in his paper is test procedure neutral.

## 4. Basic Assumptions

Our modeling work revolves around single fault model or one case at a time approach. While we raised questions about the accuracy of this approach and philosophical issues in using this model, our choice was based upon the alternative: use a combination model for the noise sources where the combinations blow up very quickly.

Secondly, we reduce the capacitive cross-talk fault propagation to a constrained transition fault propagation where instead of propagating a signal waveform we propagate a Boolean difference. This is again an engineering choice based on the chip-size we are targeting (beyond Pentium<sup>®</sup> 4 which already has 42M+ transistors). Had we used a noise waveform propagation based approach, the run time will be unacceptable for the tool.

Thirdly, we are not considering timing effects for signal transitions. There are two reasons for this choice:

- **Process technology is a moving target**: Lithography is shrunk continuously. In fact, a part may never be produced in the technology it was designed for. Therefore, fine tuning for time windows turn out to be an inaccurate approach. Since pessimism about signal integrity is better than being optimistic about it, this choice can be rationalized.
- Analysis at multiple process corners is prohibitively expensive: Even if we have every intention to rule out excessive pessimism, we must run analysis at multiple process corners to avoid being optimistic. This may require months of computation to arrive at a reasonable model.

Having stated our basic assumptions clearly, in the subsequent sections we describe the tool flow and modeling approaches.

# 5. Modeling Cross-Talk Faults

Figure 2 shows the high level view of our proposed methodology. For a given circuit, the list of all nodes that are capacitively coupled is derived using a transistor level noise analysis tool. The overall goal of the modeling methodology is to transform this list into a simplified fault list suitable for fault simulation and/or ATPG Several factors need to be accounted for to make this transformation effective and they help define the goals of a successful modeling methodology. These include the following.

- Given that fault simulation and ATPG are typically performed at the gate level for performance reasons, the final fault information must be specified using nets and gate pins in the gate level net list. This involves handling name-mapping issues across transistor-level and gate-level models.
- Though noise analysis is performed at the transistor level and results in a list of victim sink nodes and associated coupling information, the final fault list should comprise of faults on a net-by-net basis. For nets that have more than one sink node, the ability to rank the sink nodes based on overall noise is important to help achieve superior test quality.
- When multiple attackers couple with a given victim node, often it is very difficult to excite all the attackers to excite the fault effect. Often, the excitation of a subset of these attackers is enough to meet the switching threshold value for the given

node. In such cases, the ability to distinguish the required excitation conditions from the optional ones enables efficient ATPG

 Only a subset of the extracted fault list can be actually targeted using ATPG due to resource and time constraints. Hence, the ability to target the top faults based on user-specified pruning criteria is crucial to help meet time-to-market goals while maintaining good test quality.

In the following sub-sections, we show how these high-level goals are achieved through the extraction, modeling, ranking and pruning steps.



#### Figure 2 High-level view of proposed methodology.

# 5.1 Extraction

The list of nodes susceptible to capacitive coupling is derived using an in-house transistor level noise analysis tool based on circuit simulation. The circuit is divided into channel-connected components (CCC) and each CCC is analyzed separately. The results are then merged using a graph traversal.

The extracted fault list, as generated by the noise analysis tool, is a list of sink nodes (victims). For each sink node, a list of signals with which the node is capacitively coupled (attackers) is also generated. Additional information such as the net to which the sink node belongs, the switching threshold value and the noise contributions (individual and cumulative) of the attacker signals are also available for each victim node.

For the net (assumed labeled N1) driven by gate G0 in the example shown in Figure 3, we show an example of the extracted fault list in Figure 4. The two sink nodes (G1/b and

G2/a) and their associated attacker information are listed. Note that the list of attackers is the same for all victim nodes belonging to a net.



Figure 3 Example of a crosstalk victim node.

## 5.2 Modeling

Next, we show how the faults are represented in a simple, yet flexible manner that is suitable for ATPG We introduce a novel fault modeling technique, hereby referred to as *Generalized Fault Model (GFM)*.

In GFM construct, a fault refers to a physical defect/problematic behavior such as a bridge defect or a cross-talk fault.

```
Victim Node=G1/b
   Net Name = N1.
   Threshold=210mV
   Cumulative Noise=225mV
   Attacker A0: Noise=70mV
   Attacker A1: Noise=60mV
   Attacker A2: Noise=50mV
   Attacker A3: Noise=30mV
   Attacker A4: Noise=10mV
   Attacker A5: Noise=5mV
 Victim Node=G2/a
   Net Name = N1,
   Threshold=175mV
   Cumulative Noise=180mV
   Attacker A0: Noise=70mV
   Attacker A1: Noise=40mV
   Attacker A4: Noise=30mV
   Attacker A5: Noise=30mV
   Attacker A2: Noise=5mV
   Attacker A3: Noise=5mV
```

#### Figure 4 Example of an extracted cross-talk fault list.

A GFM fault consists of one or more *fault atoms*. A fault atom represents a facet of the defective behavior. For example, in Figure 4, a cross-talk fault may be detected at G1/b or G2/a or both. We call out each behavior as a separate atom. Therefore, by definition, if a fault atom is detected, then the fault is de-

tected. The fault atoms within a fault are ranked in terms of their analog behavior. For example, if one atom represents 100 mV noise at certain node and another atom represents 80 mV, then detecting the first atom gives a test of better *quality*. Thus we transform the *analog quality* to a sorted priority order among *atoms*.

A fault atom consists of paired list of excitation conditions and Excitation impact conditions. conditions describe node/pin/value requirements while impact conditions describe node/pin/value effects. If all conditions in an excitation condition list are satisfied, then all impact conditions are enforced regardless of the value at the node. Excitation condition may involve both static and transition signal values. Such a description may have inherent contradictions in them in that if an excitation is triggered, an impact is effected which in turn removes the excitation and that may in turn remove the impact and so on. Therefore, special attention must be paid in ensuring that all such cyclical dependencies are eliminated.

```
fault atom 1:(total noise=225mv)
mandatory conditions: G0=01,A0=01,A1=01,
A2=01,A3=01,A4=01,A5=01
impact: G1/b=slow-to-rise, delay=2
```

fault atom 2:(total noise=220mv) mandatory conditions: G0=01,A0=01,A1=01, A2=01,A3=01,A4=01 optional conditions: A5=01 impact: G1/b=slow-to-rise, delay=2

```
fault atom 3:(total noise=215mv)
mandatory conditions: G0=01,A0=01,A1=01,
A2=01,A3=01,A5=01,
optional conditions: A4=01
```

```
impact: G1/b=slow-to-rise, delay=2
```

```
fault atom 4:(total noise=210mv)
mandatory conditions: G0=01,A0=01,A1=01,
A2=01,A3=01
optional conditions: A4=01,A5=01
impact: G1/b=slow-to-rise, delay=2
```

#### Figure 5 Fault atoms for sink node G1/b.

Thus, GFM enables decoupling of cause and effect components using explicit representation of the excitation conditions and fault impacts. Apart from mandatory excitation conditions, optional conditions may be used to describe conditions that result in an increased fault impact. Satisfying optional conditions leads to a superior test.

A major advantage of this fault modeling technique is its ability to exploit the well-studied test generation and fault simulation algorithms associated with the traditional fault models. At the same time, it is flexible enough to handle defect-based and circuit marginality based fault models.

## 5.3 Modeling Cross-talk Faults Using GFM

In this sub-section, we show how the extracted cross-talk faults are modeled using the Generalized Fault Model. Since

our ultimate goal is to model faults on a net-by-net basis, a preprocessing step is used to bucket the victim sink nodes based on their associated net name. We then process all sink nodes for a given net and their associated attacker information to generate the GFM fault list.

For each sink node, first we determine all the attacker combinations that satisfy the threshold criteria and express them as individual fault atoms. For each such combination, the attacker information is captured in the condition list (polarity considerations are made as appropriate) and the victim information is captured as an impact. The list of attackers comprising the combination is specified as the mandatory condition list and the remaining attackers constitute the optional condition list. We illustrate the concepts using the fault information corresponding to the first sink node (G1/b) in the sample extracted fault list shown in Figure 4. For our example, there are four different attacker combinations that satisfy the minimum threshold criteria of 210 mV. Based on this, we model this victim sink node with four different fault atoms, as shown in Figure 5.

Note that the fault atoms are ordered based on decreasing cumulative noise, thereby providing an effective means for ATPG to target the different representations of the fault starting with the most desirable target.

# 5.4 Pruning

Our modeling methodology enables pruning the fault list based on three different parameters.

- Attacker pruning using minimum per-attacker noise contribution, expressed as a percentage (pa).
- Attacker combination pruning using minimum cumulative attacker noise contribution, expressed as a percentage (a).

Victim sink node pruning using minimum cumulative attacker noise contribution over the threshold noise for the victim, expressed as a percentage (t).

For example, for the GFM fault list shown in Figure 5, if we specify a pruning criteria as a=10%, the fault atoms 1, 2 and 3 are eliminated as attackers A4 and A5 do not meet the minimum per-attacker noise contribution criteria (10% of 225mV = 22.5mV) respectively.

The combination of the above said parameters can be used in a similar manner to effectively reduce the size of the GFM fault list, while ensuring that the top cross-talk sites are being targeted.

# 6. Results

We implemented the proposed cross-talk fault extraction and modeling methodology. Experimental results obtained on four proprietary Intel<sup>®</sup> circuits (0.13µm technology) are shown in Table 1. We chose the following pruning parameters for our experiments: 80% minimum required attacker noise contribution (a = 80), 80% minimum required noise contribution over threshold (t = 80) and 5% for the minimum required noise

contribution for each attacker (pa = 5). These values were chosen based on circuit design styles and were used to limit the number of faults targeted to reasonable sizes. Experiments were run using a Intel<sup>®</sup> Pentium<sup>®</sup> 4 2.0 GHz workstation running Linux OS. Run times shown for modeling and pruning are in CPU seconds.

Note that unlike the traditional fault models, the number of faults extracted from a given circuit bears no direct correlation to the size of circuit but rather depends on the quality and the style of the design. For example, dynamic circuits are more prone to cross-talk related noise as they have smaller threshold values than their static equivalents.

Circuit	Gate count	Fault count	Run time
Circuit 1	812	170	<1 sec
Circuit 2	4522	448	3 sec
Circuit 3	2430	458	2 sec
Circuit 4	1585	394	1 sec

Table 1: Experimental results for sample circuits

We also successfully performed validation of our modeling technique on a complete Intel<sup>®</sup> Pentium<sup>®</sup> design using correlation with real silicon failure data. Our cross-talk extraction and modeling technique was exercised at the full chip level and used to extract potential cross talk fault sites. A failing functional pattern was run to grade cross talk faults using a in-house simulator capable of simulating GFM faults. The cross talk fault detected by simulation matched the silicon failure in every respect i.e. failure node, failure value and failure cycle.

# 7. Conclusion

In conclusion we have presented a methodology for modeling cross-talk noise as faults and described how we retain some of the analog properties of the noise as atoms within our fault model. We have provided a methodology for pruning aggressors when several aggressors may impact a victim node. Finally, we have described results from silicon experiments to validate our result.

Acknowledgement: The authors would like to acknowledge contributions from several members of the Advanced Test Technology group at Intel Corporation. In particular, contributions made by Sanjay Sengupta, Dhiraj Goswami and Rathish Jayabharathi are greatly appreciated.

### References

[1] L. C. Chen, S. K. Gupta and M. A. Breuer, "A new gate delay model for simultaneous switching and its applications," *Proc. Design Automation, Conf.*, June 2001.

[2] W. Y. Chen, S. K. Gupta and M. A. Breuer, "Analytic models for crosstalk delay and pulse analysis for non-ideal inputs," *Proc. Int'l Test Conf.*, 1997, pp. 809-818. [3] W. Y. Chen, S. K. Gupta and M. A. Breuer, "Test generation in VLSI circuits for crosstalk noise," *Proc.Int'l Test Conf.*, 1998, pp. 641–650.

[4] W. Y. Chen, S. K. Gupta and M. A. Breuer, "Test generation for crosstalk-induced delay in integrated circuits," *Proc. Int'l Test Conf.*, 1999, pp. 191-200.

[5] A. El-Zein and S. Chowdhury, "An analytical method for finding the maximum crosstalk in lossless-coupled transmission lines," *Proc. Int'l Conf. on Computed Aided Design*, 1992, pp. 443-448.

[6] D. S. Gao, A. T. Yang and S. M. Kang, "Modeling and simulation of interconnection delays and crosstalk in high-speed integrated circuits," *IEEE Trans. on Circuits and Systems*, Vol. 37, pp.1-9, January 1990.

[7] A. K. Goel and Y. R. Huang, "Modeling of crosstalk among the GaAs VLSI connections," *IEE Proc. Part G*, Vol. 136, pp. 361-368, 1989.

[8] M. A. Breuer and S. K. Gupta, "Process aggravated noise (PAN): new validation and test problems," *Proc. Int'l Test Conf.*, 1996. pp. 914 -923.

[9] R. Kumar, "Interconnect and Noise Immunity Design for the Pentium. 4 Processor,"

http://developer.intel.com/technology/itj/q12001/pdf/art\_5.pdf

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

[11] Sandip Kundu, Sanjay Sengupta and Dhiraj Goswami, "A generalized fault model for defects and circuit marginalities," Pending US Patent.

[12] F. Moll and A. Rubio, "Spurious signals in digital CMOS VLSI circuits: a propagation analysis," *IEEE Tran. On Circuits and Systems –II: Analog and Digital Signal Processing*, Vol. 39, No. 10, pp. 749-752, October 1992.

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

[14] A. Rubio, N. Itazaki, X. Xu and K. Kinoshita, "An approach to the analysis and detection of crosstalk faults in digital VLSI circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol.13, No.3, pp.387-394, March 1994.

[15] S. Voranantakul and J. L. Prince, "Crosstalk analysis for high-speed pulse propagation in lossy electrical interconnections," *IEEE Trans. on Components, Hybrids, and Manufacturing Technology*, Vol. 16, No. 1, pp. 127-136, February 1993.

[16] R. K Watts, *Submicron Integrated Circuit*, New York: Wiley, pp. 317-318, 1989.

[17] H. You and M. Soma, "Crosstalk and transient analysis of high-speed interconnects and packages", *IEEE Trans. On Solid State Circuits*, Vol. 26, pp. 319-30, March 1991.