Signal Integrity Fault Analysis Using Reduced-Order Modeling

Amir Attarha and **Mehrdad Nourani**

Center for Integrated Circuits & Systems
The University of Texas at Dallas
Richardson, TX 75083-0688

{attarha,nourani}@utdallas.edu

Abstract

This paper aims at analysis of signal integrity for the purpose of testing high speed interconnects. This requires taking into account the effect of inputs as well as parasitic RLC elements of the interconnect. To improve the analysis/simulation time in integrity fault testing, we use reduced-order modeling that essentially performs the analysis in the frequency domain. To demonstrate the generality and usefulness of our method, we also discuss its application for test pattern generation targeting signal integrity loss.

1. INTRODUCTION

As we approach 100*nm* technology, noise and skew imposed by interconnects have emerged as main concerns in the interconnect design of gigahertz SoCs. Voltage distortion (noise) and delay violations (skew) contribute to signal integrity loss causing time-dependent dielectric breakdown (TDDB) [1], functional error [2], performance degradation [3] and reliability problems [4]. Therefore, new test development and coverage techniques for interconnects susceptible to noise and skew are substantial. Unfortunately, the existing methods analyzing the interconnect for integrity are either too slow or too inaccurate [5]. This motivated us to develop an analysis method that is not only accurate for purpose of integrity fault testing but also much faster than the existing methods.

Due to the complexity of the interconnect model, an efficient simulation method is essential for test pattern generation. We have adopted reduced-order model to alleviate computation complexity with slight loss of accuracy. Reduced-order model methods were developed as an alternative for circuit-level simulators to approximate the behavior of long interconnect, power and clock networks [6].

On the application side of our method, we use the model order reduction methods such as the PVL [7] and SyMPVL [8] to evolve a test pattern generation approach for detecting intermittent failures due to integrity fault on long interconnects. These methods are chosen because of their high accuracy, numerical stability and ability to identify the true poles and zeros of the original system efficiently. Poles and zeros are the roots of denominator and numerator of the system transfer function, respectively [9].

This paper is organized as follows. In Section 2, we show how the reduced-order model can be used to partially analyze the interconnect network in frequency domain and how integrity loss problem (e.g. signal overshoot and skew) is formulated. In Section 3 we comment on few techniques that can further improve the integrity fault analysis in terms of running time. Section 4 discusses application of our analysis tool in a test pattern generation method that targets integrity faults. The experimental results are presented in Section 5. Finally, concluding remarks are in Section 6.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DAC 2002, June 10-14, 2002, New Orleans, Louisiana, USA.
Copyright 2002 ACM 1-58113-461-4/02/0006 ...$5.00.

2. SYSTEM REPRESENTATION

2.1 Effect of Interconnect on Integrity

The origin of signal integrity problems lies in the circuit interconnects. In deep-submicron SoC's, an interconnect behaves like a lossy transmission line. In low and mid-range frequencies, common in the past, the *RC* delays have been the dominating factors in the global interconnect delay and distortion. As technology shrinks, the effects of coupling capacitances and mutual/self inductances will increase significantly, jeopardizing the efficiency of the SoCs.

We present a new formulation for the interconnect network which facilitates various test analysis targeting integrity loss including pattern generation [10]. The most important novelty of this formulation is the fact that the effects of inputs are parameterized into the behavior of the interconnect model. This leads to a unique formula which annotates the effects of input patterns within the behavior of the interconnect network.

2.2 Input Representation

In a System-on-Chip (SoC), the interconnect drivers are designed very strong to diminish any possible signal disturbance [11]. Therefore, the transitions on the input signals can be reckoned as step functions. On the other hand, some of the signals are quiescent at "1" or "0" depending upon the driver circuitry. We define an input $x_i(t)$ of the interconnect network as:

$$x_i(t) = a_i u(t) + b_i$$

where $u(t)$ is a step function; a_i and b_i are constants such that $a_i \in \{-1,0,1\}$ and $b_i \in \{0,1\}$.

2.3 System Representation

An interconnect network can be represented by a set of transfer functions in the S-domain [9]. As shown in Figure 1, an interconnect network with n inputs and m outputs is represented by transfer

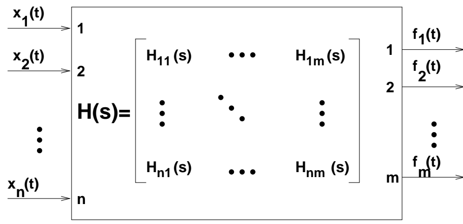

Figure 1: S-domain representation of interconnect network

function $H(s)$. $H(s)$ itself contains $n \times m$ partial transfer functions relating inputs to outputs. Note that, in general for an interconnect network, we have $m \geq n$ due to the possibility of fanout on some wires. The transfer function of a specific output f_r ($1 \leq r \leq m$) can be represented as:

$$H_{f_r}(s) = \sum_{i=1}^{n} X_i(s) H_{ir}(s) \tag{1}$$

where $X_i(s)$ is the Laplace transform of $x_i(t)$ (the ith input).

Using an order reduction method (e.g. PVL [7] or ENOR [12]), the transfer function of output f_r becomes of order q:

$$H_{f_r}(s) = \sum_{i=1}^{n} \sum_{j=1}^{q} \frac{k_{ij}}{(s - p_{ij})} \tag{2}$$

where k_{ij} and p_{ij} are zeros and poles of the output f_r, respectively. Note that $H_{f_r}(s)$ is obtained based on the superposition of the effects of all n inputs on f_r. The output of a system can be computed in the frequency domain by multiplying the transfer function and the input function. After transferring it to time domain to observe the timing behavior of the output for $t > 0$, $f_r(t)$ is calculated by:

$$f_r(t) = \sum_{i=1}^{n} \left(a_i \sum_{j=1}^{q} \frac{k_{ij}}{p_{ij}} \left(e^{p_{ij}t} - 1\right) + b_i \sum_{j=1}^{q} k_{ij} e^{p_{ij}t} \right) \tag{3}$$

The practical use of Equation 3 is that if poles and zeros are available for a given set of inputs, $f_r(t)$ can be obtained efficiently using numerical methods. Therefore, it can be used to find patterns that cause maximal delay or overshoots. In providing this equation, we need to estimate q (the order of model reduction). Full discussion on empirical and heuristic methods to determine q is beyond the scope of this paper. We just point out that it plays an important role in the accuracy of the equation and computational complexity of the reduced model. We will briefly comment on such methods in the next section.

2.3.1 *Finding Input Patterns for Maximal Delay*

In Equation 3, all the timing information of the output signal is available and the impact of inputs $x_i(t)$ are considered within a_i and b_i coefficients. As the result, by defining delay as the time taken to reach certain voltage (e.g. 50% of V_{dd} or $V_d = V_{dd}/2$ as shown in Figure 2), the delay can be computed from the following equation:

$$f_r(t) - 0.5V_{dd} = 0 \tag{4}$$

Using numerical methods, we can solve Equation 4 to obtain the maximal delay t_d associated with an input set.

2.3.2 *Finding Input Patterns for Maximal Overshoot*

Overshoot occurs when a signal momentarily exceeds V_{dd}. In other words, overshoot is the global maximum of the output waveform of the system. Finding the global maximum of a nonlinear equation (e.g. Equation 3) is a complicated problem in general. However, overshoot is the maximum voltage value taking place in the neighborhood of the root of Equation 4. More specifically, the neighborhood information (i.e. a point around t_d) has already been obtained by solving Equation 4. The following equation finds when the overshoot happens:

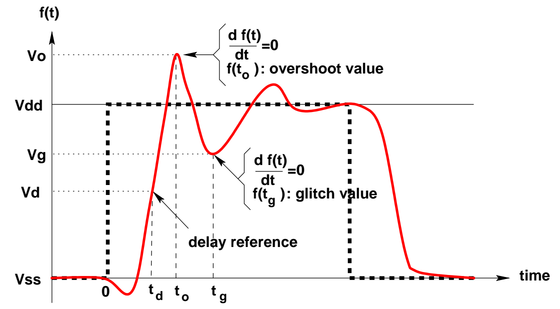

Figure 2: Calculating delay and overshoot values

$$\frac{df_r(t)}{dt} = 0 \ni t \in (t_d, t_d + \Delta] \tag{5}$$

By solving Equation 5, t_o is found and the overshoot value ($f_r(t_o) = V_o$) is computed. Figure 2 graphically demonstrates the calculation of overshoot value.

2.3.3 *Finding Input Patterns for Glitches*

Glitches (ringings) occur when a signal momentarily falls below V_{Hmin}. Similar to what we explained for overshoots, glitches taking place in the neighborhood of t_o obtained already by solving Equation 5. The following equation finds when the glitch happens:

$$\frac{df_r(t)}{dt} = 0 \ni t \in (t_o, t_o + \Delta] \tag{6}$$

By solving Equation 6, t_g is found and the glitch voltage value ($f_r(t_g) = V_g$) is computed for further processing. Figure 2 demonstrates this graphically.

3. IMPROVING PERFORMANCE

Success of the integrity fault analysis model that we presented depends on numerical techniques that can solve the equations efficiently. In this section, we present few techniques that significantly enhance this analysis.

3.1 Numerical Methods to Solve Equations

Theoretically, we are able to analyze the output for any possible indication of signal integrity loss such as delay, overshoot, or undershoot. However, Equations such as 3, 4 and 5 must be solved efficiently. There are several numerical methods for solving nonlinear equations such as Bisection, Secant, and Newton-Raphson [13]. Although such numerical methods can solve these Equations, they may require long running time. Moreover, care must be given to the proper choice of a starting point that may not be easy to determine. If we are successful in choosing a good starting point, methods such as Newton-Raphson method converges quadratically to the root, as proven in [13].

3.2 Using SyMPVL Algorithm

The interconnect network is modeled as a large RLC circuit. Such a circuit can be described and characterized in terms of square and symmetric matrix transfer functions. Moreover, being composed of passive components, RLC networks are always stable and passive. In our work we used PVL/SyMPVL [7]/[8], an algorithm for the approximation of the symmetric multi-port transfer function of an RLC network. This algorithm takes advantage of these special structures of RLC networks. SyMPVL computes symmetric

matrix-Pade approximations of the circuit matrix transfer function, using a symmetric block-Lanczos technique. It produces more accurate results compared to other models and even in special cases of RC and LC networks, guarantees stability and passivity at any order of approximation [8].

More specifically, in our application this method computes the reduced-order model for large multi-port RLC networks. This leads to considerably less execution time to provide Equation set 3. Instead of running PVL algorithm $n \times m$ (see Figure 1) and performing superposition, the desired matrices (e.g. poles and zeros) can be obtained by SyMPVL in one run. This significantly improves the preprocessing time to construct Equation set 3.

3.3 Defining the Order of Reduction

As discussed earlier, q (the order of model reduction) should be much smaller than the the order of original model. Choosing an appropriate q enhances accuracy of the model while keeping the computational complexity under control. Unfortunately, to the best of our knowledge, there is no analytical method to find the optimum value of q. In almost all works reported in the literature, the value of q is selected heuristically or based on trial and error [6] [8] [12]. The selection is done based on the application and desired level of accuracy and computation complexity. A key feature of the PVL [7] algorithm is that the accuracy of approximation can be guaranteed for a special frequency range. The authors in [7] have shown that a quality measure (Q) for the poles of $H(s)$ can be defined to determine q. Calculation of Q is based on defining bounds for eigenvalues of the reduced model. Finally, Q is used for determining the optimum order of reduction (q) so that to satisfy the desired accuracy. Details can be found in [7].

In our work, we have simply started from a very small q (e.g. 1/50 of the original order) and checked its accuracy by comparing to the simulation results of SPICE (i.e. the golden model). If the accuracy was above 8%, we have increased q and repeated the process. We acknowledge that such selection process is ad-hoc and may require few iterations. However, since the selection of q is done only once the effort, compared to the running time of the signal integrity analysis, is negligible. We are currently working on a mixed analytical-empirical heuristic method to choose the optimal (or near-optimal) value of q and we hope to report the outcome in near future.

3.4 Defining the Locality Metric

Considering the property of the interconnect network and the effect of parasitic elements, the search space of possible patterns can be restricted without decreasing the chance of finding effective patterns. To be more specific, we need to elaborate on the roles of different parasitic elements in the behavior of an interconnect network. In spite of coupling capacitances between all wires, the effect of capacitive coupling is considered local, in the sense that the coupling effects of adjacent wires are quite dominant compared to the capacitive coupling effects of far off wires [14] [15]. However, the inductance has larger range effect and thus the effect of mutual inductance could be significant. Furthermore, the effect of coupling inductances and capacitances on a wire oppose each other [14]. When the signal on a wire switches in one direction, the noise due to capacitive coupling affects other nearby signals in the same direction as that of switching. However, the noise due to inductive coupling is in the opposite direction.

To take advantage of above observation, we define the *locality* metric. Locality k is a parameter showing the proximity of neighboring wires in each side that actively affect one wire due to coupling capacitances. In other words, the victim is affected by the signals carried on k wires on each side of it mainly due to the repercussion effects of coupling capacitances. The signals traveling on wires beyond the locality border exert adverse effects on the victim mainly due to their mutual inductances.

Using the locality metric, we can effectively consider RC model for the victim wire and $2k$ wires next to it and RLC model for the rest. Practically, this reduces the complexity of the network and shortens the analysis time for constructing and solving Equation sets 3 and 5. Briefly, if the interconnect network contains n lines, the total number of patterns for integrity analysis shrinks from $n * 4^n$ to $n * 4^{n-2k-1}$, where 4 is the possible states for an input (falling, rising, quiescent at 0, and quiescent at 1). Also, the term n appears because the analysis of all patterns has to be done for all wires. Note that depending upon the targeted fault (overshoot/delay), the input of the specific line subject to fault is determined. We can further restrict the search space of patterns by exploiting the fact that the effects of mutual inductances oppose coupling capacitance effects. Beyond the locality region, the effects of mutual inductances are considered dominant. Therefore, the number of patterns can be significantly reduced, that is: $(n-2k) * 4^{(n-2k-1)} + k\sum_{i=0}^{k-1} 4^{(n-2k+i)}$, where, the summation term covers the asymmetry of k lines at the sides of interconnect network. Note that depending on the available information, test time budget and desired accuracy the candidate set of patterns will be chosen. For example, for application of test pattern generation, the method eventually selects a very small subset of the candidate set.

4. TEST GENERATION

The signal integrity model and analysis procedure presented in the previous section is a general technique that can be used in many signal integrity simulation and analysis applications. In this section we show how our technique can be used for signal integrity fault (unacceptable overshoot and skew) analysis. More specifically, we use this technique to find deterministic patterns that stimulate maximal integrity loss on the interconnects.

Conventional test pattern generation algorithms cannot be used for signal integrity faults due to fundamental differences between the nature of classic and integrity faults [10]. Classic faults such as stuck-at, bridging, and open faults, are often assumed to cause permanent adverse effects on system in terms of logic level of signals. On the other hand, integrity faults are intermittent, appear mostly as disturbances, heavily depend on the working frequency and are hard to categorize.

For generating the deterministic test patterns, accurate interconnect model is required which can be obtained using extraction tools such as NETAN [16]. Then, the locality factor k and the order of model order reduction q need to be decided. In this work, we use the PVL/SyMPVL algorithm [7]/[8] to get the reduced order model and find the poles and zeros. For a selected line, Equation 3 is formed. Then, the integrity loss regarding is calculated for an input pattern chosen from the candidate set. If the pattern causes integrity loss (i.e. overshoot and skew exceed the acceptable range) the pattern will be stored. Otherwise, another pattern will be tried until the whole candidate set is exhausted.

5. SIMULATION RESULTS

MATLAB [17] was used for generating the reduced order model and performing analytical computations. To test the accuracy, interconnects with various lengths and bitwidths are analyzed using our model and also simulated in SPICE [18]. The results of our reduced model were consistently within 8% of accuracy compared to SPICE simulation. Specifically, our method identifies the same

Line #	Time-Dependent Loss		Voltage-Dependent Loss	
	Avg. Iterations	Delay [ps]	Avg. Iterations	Overshoot [Volt]
1	424.6	25.4	208.7	2.76
2	320.0	32.9	132.3	2.69
3	129.0	25.8	159.3	2.91
4	106.3	31.3	84.4	3.04
5	13.7	52.7	61.0	3.19
6	276.3	33.5	131.7	2.73
7	180.3	41.5	70.7	2.85
8	168.3	28.3	365.4	2.63
9	1000	Not Found	330.7	2.71

Table 1: The results of integrity fault simulation

test patterns to stimulate the maximal integrity loss as SPICE finds but runs much faster by 2 to 3 order of magnitude.

We have employed an interconnect network containing nine parallel lines with RLC segments, coupling capacitances, and mutual inductances to perform the experimentations. Before starting the test pattern generation, we need to restrict the search space of possible test patterns for signal integrity faults. For instance, if maximum delay of $n = 9$ interconnect lines are targeted, the total number of possible patterns is equal to $9 * (4^9) = 2,359,296$ as we explained in previous section. While if we consider locality factor of $k = 2$, as described in Section 4, the number of patterns in candidate set will reduce to $(5 * 4^4) + (2 * 4^5) + (2 * 4^6) = 11,520$.

After defining k, the test pattern generation procedure is used. q is calculated along with the model order reduction. In our experimentation the selected order $q = 14$, while the original model is order of 100 since it contains more than 300 memory elements such as capacitances and inductances. The frequency and the source voltage are $f = 1GHZ$ and $V_{dd} = 2.5\ Volt$. The thresholds for unacceptable skew, overshoot and ringing are assumed to be $T_d = 25\ ps$, $V_o = 2.62\ Volt$ and $V_g = 1.75\ Volt$, respectively.

Table 1 demonstrates the simulation results for capturing time-dependent and voltage-dependent integrity faults, respectively. As seen, the average number of MATLAB iterations to solve the equations numerically vary depending upon the layout information extracted and position of the line. The lines in the middle are more subject to coupling capacitance and mutual inductance interferences compared to the lines at (or close to) the side of interconnect network. Therefore, as a general trend, more efforts are required to find the desired test pattern at (or close to) sides because the candidate set is smaller. For instance, as Table 1 shows, MATLAB finds the test patterns for line 5 (in the middle) after only 14 iterations on the average while 424 iterations on average were needed to find the test patterns for line 1.

Note that in test generation, the patterns are chosen from the candidate set randomly. Additionally, the process is terminated after: 1) a test pattern causing unacceptable integrity loss (e.g. delay larger than 25 *ps* or overshoot larger than 2.62 *Volt*) is found; or 2) after 1000 MATLAB iterations. Obviously, such termination conditions can be revised to check more patterns from the candidate set or find more patterns violating the integrity thresholds. In general, there might be also some wires that no pattern can exceed T_d, V_o or V_g which means no unacceptable integrity loss will occur in that particular wire.

6. CONCLUSION

High-speed interconnects in deep-submicron need to be treated as transmission lines. This is a challenge for SoC testing due to the computational demand of distributed RLC model. We proposed an analysis technique based on the reduced-order model to improve the performance of signal integrity analysis. Our analysis method is quite general and can be employed by various integrity test methods. We applied this technique to find deterministic test patterns that can stimulate maximal integrity loss (overshoot and skew) in each line of the interconnect network.

Acknowledgements

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

REFERENCES

[1] W. Hunter, "The Statistical Dependence of Oxide Failure Rates on Vdd and Tox Variations with Applications to Process Design, Circuit Design and End Use," in Proc. *Reliability Physics Symposium*, pp. 72-81, 1999.

[2] H. Chen and L. Wang, "Design for Signal Integrity: The New Paradigm for Deep-Submicron VLSI Design," in Proc. *Intern. Symp. on VLSI Tech.*, pp. 329-333, 1997.

[3] P. Nordholz, D. Treytnar, J. Otterstedt, H. Grabinski, D. Niggemeyer and T. Williams, "Signal Integrity Problems in Deep Submicron Arising from Interconnects Between Cores," in *Proc. VLSI Test Symposium (VTS'98)*, pp. 28-33, 1998.

[4] M. Nourani and A. Attarha, "Detecting Signal Overshoots for Reliability Analysis in High-Speed SoCs," to appear in *IEEE Trans. on Reliability*, 2002.

[5] A. Kahng, S. Muddu and E. Sarto, "Interconnect Optimization Strategies for High-Performance VLSI Designs," in *Proc. Intern. Conference on VLSI Design*, pp. 464-469, Jan. 1999.

[6] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm," *IEEE Conf. on CAD* , 1997.

[7] Peter Feldmann, and Ronal W. Freund, "Efficient Linear Circuit Analysis by Pade Approximation via the Lanczos Process," *IEEE Trans. on CAD*, vol. 14, no. 5, pp. 639-649, 1995.

[8] R. Freund and P. Feldmann, "Reduced-Order Modeling of Large Linear Passive Multi-Terminal Circuits Using Matrix-Pade Approximation," in Proc. , 1998.

[9] Alan V. Oppenheim, Signals and systems, second Edition, Prentice Hall, 1997.

[10] A. Attarha, and M. Nourani, "Test Pattern Generation for Signal Integrity Faults on Long Interconnects," in *Proc. VLSI Test Symposium (VTS'02)*, April 2002.

[11] J. Rabaey, *Digital Integrated Circuits*, Prentice Hall, 1996.

[12] Bernard N. Sheehan, "ENOR: Model Order Reduction of RLC Circuits Using Nodal Equations for Efficient Factorization," *Design Automation Conference*, pp. 17-21, 1999.

[13] E. W. Cheney, *Numerical Mathematics and Computing*, Brooks Cole publishing Company, 1985

[14] C. Cheng, J. Lillis, S. Lin and N. Chang, *Interconnect Analysis and Synthesis*, John Wiley & Sons, 2000.

[15] L. He, N. Chang, S. Lin and O. Nakagawa, "An Efficient Inductance Modeling for On-chip Interconnects," *IEEE Custom Integrated Circuits Conference*, pp. 457-460, 1999

[16] NET-AN: Multi-net three-dimensional field solver extraction tool, Users reference manual, OEA International, Inc., 2001.

[17] MATLAB User's and reference manual, *2000 The MathWorks, Inc., Version 6.0.0.88*,2000.

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