

Noise Characterization of Static CMOS Gates

Rouwaida Kanj¹, Timothy Lehner², Bhavna Agrawal² and Elyse Rosenbaum¹

¹ECE Dept. and Coordinated Science Lab., University of Illinois at Urbana-Champaign,

²EDA Labs, IBM Corporation, Hopewell Junction, New York 12533

kanj@uiuc.edu

ABSTRACT

We present new macromodeling techniques for capturing the response of a CMOS logic gate to noise pulses at the input. Two approaches are presented. The first one is a robust mathematical model which enables the hierarchical generation of noise abstracts for circuits composed of the precharacterized cells. The second is a circuit equivalent model which generates accurate noise waveforms for arbitrarily shaped and timed multiple-input glitches, arbitrary loads, and external noise coupling.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aids- Simulation and verification

General Terms

Measurement, Performance, Design, Reliability, Verification

Keywords

Cell model, noise analysis, sensitivity, circuit-equivalent model, mathematical model, simulation

1. INTRODUCTION

With the increasing size, complexity and integration level of VLSI designs, along with faster clock cycles, increased coupling capacitances, and reduced power supply and threshold voltages, signal integrity analysis has become a necessary element of the design cycle. Because dynamic simulations to analyze all noise conditions are impractical, static noise analysis [1] was developed, which typically involves transistor-level simulations.

On large designs, however, it is not always possible to do the detailed transistor-level analysis due to capacity and run-time constraints. Thus, cell-based tools have been developed for further speed and capacity enhancement [1,2,3]. Some of these tools perform cell-specific transistor-level simulations to detect failures [1,3]. Others may rely on rough abstracts of noise immunity to describe a cell's tolerance to injected noise; the load-specific abstracts are generated by transistor-level pre-simulations. Once a failure is detected, nets with highest dynamic gain or injected noise are identified. This, together with information about the ratio between propagated noise and coupled noise, can help circuit designers correct the noise problem on a failed path.

However, approaches which rely on a traditional SPICE-like simulator are still very slow for large circuits, and SPICE-level

models may not be available for ASIC-like (gate-level) designs. Moreover, in advanced SOI technologies, the input noise may propagate, with little attenuation, through several logic stages. This requires noise models which accurately predict the propagated noise waveform as well as the impact of other factors on the propagated noise, such as coupling. The development of fast and simple noise macromodels is therefore a significant improvement for cell-based noise analysis tools.

The authors in [4] proposed an analytical model of a cell's noise stability which is a measure of how much noise a cell can tolerate before functionally failing. However, this method can lead to falsely detecting noise violations [5]. The authors in [5] propose an improved dynamic gain criterion, $S=dV_{out}/dV_{in}$, which is obtained by varying the envelope of the input waveform. However, they linearize the driver models to enable alignment of the coupling and propagated noise. None of the above methods, or other well-known delay models [6][7][8], address the issue of multiple inputs glitching simultaneously and their temporal correlation, or of hierarchical noise analysis.

In this paper, we present new, comprehensive and computationally efficient macromodeling techniques to enable building a noise-rule library. We propose and implement two approaches. The first is a mathematical or behavioral modeling approach. ASIC library-like cells are precharacterized and their behavior in the presence of noise is captured by mathematical models. This model can be used to i) predict the propagated noise and the AC gain ($|dV_{out}/dV_{in}|$, also called sensitivity [5]); ii) to construct the noise abstracts of a single cell for any specified lumped load; iii) to hierarchically construct the abstracts of a group of cells using the macromodels of individual cells. All this can be achieved with significant improvements in speed and accuracy.

Zolotov et al. [5] demonstrated that the interaction between the propagated and the coupling noise is important. The need to adequately model this interaction with high waveform fidelity for arbitrary loads led to the development of our second model, the circuit-equivalent macromodel. This model generates correct output waveforms in the presence of i) arbitrarily shaped and timed input waveforms, ii) multiple-inputs glitching, iii) arbitrary loads, and iv) external noise coupling. Therefore, this model is suitable for computing glitch delays for noise window calculations [3], for computing the effect of coupling on noise propagation in the presence of device non-linearities or for arbitrary loads, and even for generating data for building the mathematical model, as was proposed in [8] for creating timing models.

2. MATHEMATICAL MODEL

This model captures the key features of the response envelope of a circuit to a noisy input. However, to do this efficiently, one must carefully select the input and output variables, perform sampling, and then select a model. Each of these steps are described in detail below. This is followed by an application of this model to hierarchical noise analysis.

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC'04, June 7–11, 2004, San Diego, California, USA

Copyright 2004 ACM 1-58113-828-8/04/0006...\$5.00.

2.1 Design of the Experiment

2.1.1 Variable Selection

To reduce the model complexity, we adopt a simplified representation of a noise glitch. The noise glitch at a given input (and output) is represented by a triangular pulse whose area and amplitude match that of the original waveform (see Figure 1). For simplicity, we ignore the Miller effect region. We also assume a single lumped load capacitance, C_L , per output node.

We select the independent variables to be $X = \{\text{Amp}, t_1, t_2\}$ (per input), $1/C_L$ (per output). This helps avoid any colinearity in the model. Input-output correlation analysis ruled out any further need for input variable transformation (e.g. x to $\exp(x)$); a good correlation between the input and the output helps reduce model complexity [9]. It is worth mentioning that the input amplitude, which is the natural driver, showed strongest correlation with the output. Finally, for more accurate results, we normalize the input variables by mapping their ranges to $[0, 1]$.

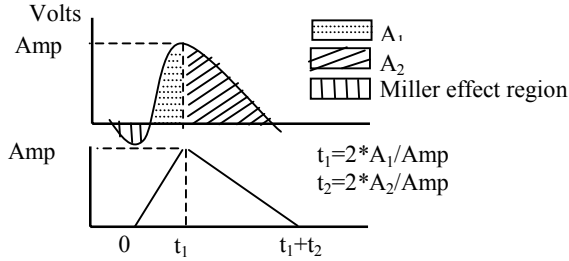


Figure 1. Noise waveform approximation. We assume that all inputs are initially at dc High or Low. Amp corresponds to the maximum deviation-from-nominal of the noise glitch. A_1 is the deviate-from-nominal area; A_2 is the return-to-nominal area.

On the other hand, smoothness of the response is desired to ensure good predictive capabilities [9]. Analysis of families of curves of sampled data (e.g., out_{t_1} vs. in_{Amp} , for fixed in_{t_1} , in_{t_2} , in_{C_L}) indicated that smooth output variables are $Y = \{\text{out}_{\text{Amp}}, \text{out}_A = \text{out}_{A_1} + \text{out}_{A_2}, \text{out}_{A_2}\}$ per output pin. While the amplitude in $A_2 = 2 * t_2 * \text{Amp}$ helps achieve smoothness goal by acting as a weighting filter, it does not completely achieve this goal in A_1 . Many factors like the Miller effect, coupling capacitance, and steepness, lead to non-uniformity of A_1 ; the total area, A , is a smoother function to model.

2.1.2 Obtaining the Training Data Sets

To determine the set of experiments used to train the model, we adopted the adaptive *layered* volume slicing technique [10]. This method helps reduce the total number of nodes (experiments) compared to regular slicing techniques. It utilizes fractional factorial analysis [11] and variable screening techniques to rank and group the variables based on their significance to the output response (see Figure 2).

For purposes of grouping and slicing, and to obtain an optimal set of experiments per output pin, we adopted a single output response metric (as opposed to the three dependent variables). The response metric is not necessarily unique. An example is the weighted sum of the output waveform absolute area, with emphasis on the area above a certain user-defined threshold voltage, or a time-averaged response.

Finally, for very large dimensions, one may rely on randomly sampled data. One would then randomly shed the samples that

result in the abundantly available digital response to avoid biased models.

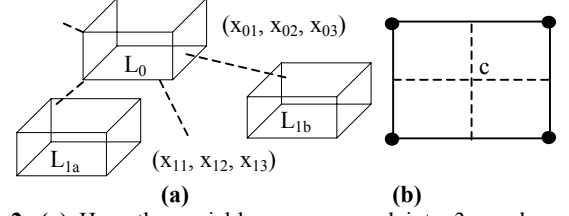


Figure 2. (a) Here the variables are grouped into 3 per layer. Layer 0 variables are most significant and so on. This helps reduce the number of new nodes introduced per slice. (b) Slicing a given layer is terminated when the estimated average response at the center 'c' meets a certain error criteria. We also enforced a limit on the maximum number of slices. Note that the responses in a given layer are averages of those in the lower layers.

2.2 Model Selection and Reduction

2.2.1 Model Basics

In addition to smoothness, residual analysis dictates the necessary dependent variable transformations and, therefore, the model formulation. The main objective is to achieve residuals that have minimal correlation, or maximum random trend with respect to the dependent variables. This is usually detected by analyzing the residual normal probability plot.

To achieve such randomness, we employed a logit transform for the bounded output amplitude, y .

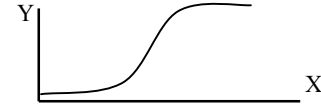


Figure 3. Y is a sigmoidal, bounded s-shape, function of X ; $\text{logit}(Y)$ is a linear (polynomial) function of X .

$$\text{Logit}(y) = \ln\left(\frac{y}{V_{DD}^+ - y}\right) = P_0(x) \quad (1)$$

where $y \in (0, V_{DD}^+)$. P_0 is a polynomial function in x . Residual analysis led to adopting the following representation.

$$\text{out}_{\text{Amp}} = \frac{V_{DD}^+}{1 + e^{-P_1(x)}} * (1 - e^{-P_2(x)}) \quad (2)$$

P_1 and P_2 are polynomial functions in x ; their degree of complexity is discussed in the coming sections. The first term in (2) is the inverse logit transform, and is expected to capture the sigmoidal nature of the output amplitude as well as the logical response. Ideally, we want P_1 to be a function of the input amplitude(s) only. The second term represents a decay function of $(t_1, t_2$ and $1/C_L)$; for larger t_1, t_2 and smaller C_L , the transient response approaches that of the dc transfer function. Note that P_0 is used to obtain an initial guess of P_1 's complexity and parameter values. For constant residual variance, we modeled the remaining variables as follows.

$$\text{Log}(\text{out}_{\text{A}}) = P_3(x) \quad (3)$$

$$\text{Log}(\text{out}_{\text{A}_2}) = P_4(x), \quad (4)$$

where P_3 and P_4 are polynomials in all the input variables.

2.2.2 Regression and Reduction Techniques

Our objective is to find the best models with the least number of parameters and minimum complexity. This in turn depends on the complexity of the embedded polynomials, and can be achieved as follows. First, one would propose various polynomial base functions to assess the degree of nonlinearity of the response, and

then rely on reduction techniques. Figure 4 illustrates the basic model regression and reduction flow diagram.

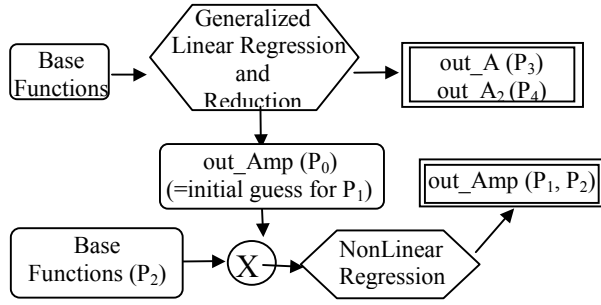


Figure 4. Model regression and reduction flow diagram.

Base functions are polynomials of variable complexity, e.g., product of monomials, 3rd order polynomial, and 2nd order polynomial with amplitude interaction only. We relied on standardized tests of case studies to rule out base functions that failed to capture the general response. We also used the following regression and reduction SAS [12] options.

- *Proc reg*: based on the linear least squares method
 - *Selection=RSQ size=M*: finds the M-term-function with best R² coefficient of correlation [9].
 - *Selection=C_p*: this method is more sensitive to changes in the adequacy of prediction than R²; C_p is defined in (5), n is the number of samples, p is the reduced model target number of parameters, MSE is the full model mean square error, and SSE_p is the reduced model sum of square error. The reduced model that satisfies Mallows' criteria, C_p = p [9], has similar MSE as the full model.

$$C_p = \frac{SSE_p}{MSE} - (n - 2p) \quad (5)$$

- *Selection=stepwise*: suitable for very large problems (many terms in full functions); relies on user-defined variable significance criteria.
- *Proc nlin*: for nonlinear regression

Our goal was to devise a unique base function and reduction method, or a generalizable reduced model function for each dependent variable. As will be illustrated in the following section, the response is highly nonlinear; third order interaction, and therefore third order polynomials, were found favorable.

2.3 Macromodeling Results

2.3.1 Case Studies

We built and tested the model for the case of single input glitching in an inverter, NAND2, NOR2, and NAND4 gates. 700 transient presimulations were performed per gate; this averages to 5 values per variable. The methodology is generalizable to other logic gates. The following discussion summarizes the complexity and generalizability of the polynomial expressions.

- *out_Amp*: The model involved 25 terms (as opposed to 45 terms in the full model). The inverter-based model was generalizable to other gates.

- P₁ (P₀): is a reduced 3rd order polynomial function in all independent variables. It includes 15 terms (compared to 35 terms for the full model); higher order interaction terms were dominant. This ruled out the possibility of amplitude only dependence. In the presence of quadratic P₂, P₁ being a 2nd order full polynomial also provided good results.

- P₂: we adopted a 2nd order polynomial in (t₁, t₂ and C_L). P₂ helped bound the outliers.

- *out_A*: The base model was a 3rd order full model. Due to complexities such as gate-specific parasitics and Miller effect, particularly observable in the A₁ region, significant reduction on a single function representation was not possible. Thus, we relied on Mallows' C_p selection, which resulted in around 28-30 terms (compared to 35 for the full model).
- *out_A₂*: Again the base model was a 3rd order full model. Out_A₂ is a smoother function than out_A, and further reduction in the number of terms was possible. One may rely on Mallows' C_p for term reduction. Our experiments also showed that it is possible to achieve a 20-term model when tolerating a few pessimistic outliers.

Table I compares the results of our reduced models to SPICE. For each gate, we performed 1000 experiments; the amplitudes, widths of the input noise pulses, and fan-out in each experiment were randomly selected over the ranges [0.4V-0.9V], [50ps-600ps], and [1-5], respectively. A single fanout corresponds to C_L=3.5 fF. The experiments were done using 0.25μm PD-SOI-CMOS technology. V_{DD}=1.1V and a static load model were adopted. Figures 5 and 6 compare the inverter model-generated area and amplitude against those of SPICE.

Finally, when the number of inputs glitching exceeds a certain limit beyond which the cost of model characterization or model evaluation, i.e., number of parameters, is undesirable or unfeasible, one may conservatively bound the noise. That is, one may tie together all the inputs, or rely on logic and the responses of single inputs to bound the response of multiple inputs. Also one may rely on the circuit-equivalent model.

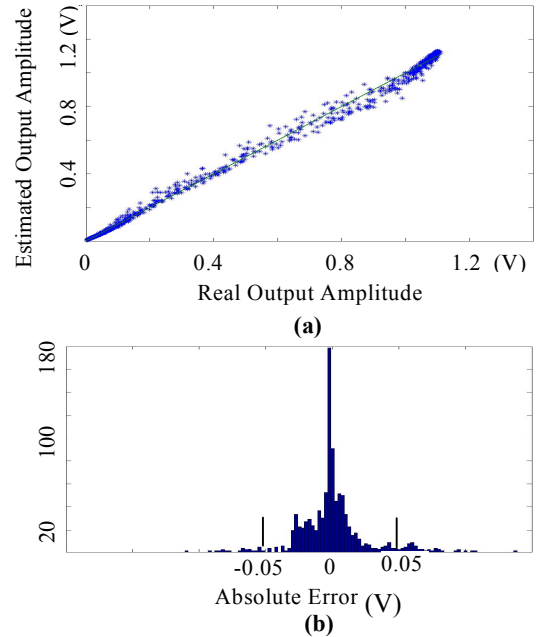


Figure 5. (a) The inverter's real (SPICE) versus estimated (mathematical model) output amplitude (b) The corresponding histogram of the absolute error.

2.3.1 Hierarchical Noise Analysis

We used these single gate models to construct noise rejection curves (abstracts) of multiple gates. Figures 7 and 8 compare our model-generated abstracts to those of SPICE for the two circuits

of Figure 9. The mathematically generated abstracts are accurate and conservative. We adopt the approximation that the inputs to the NOR gate (Figure 9b) are identical. Finally, the results are achieved at several orders of magnitude of speed improvement compared to SPICE, because we are replacing the simulations by simple equations.

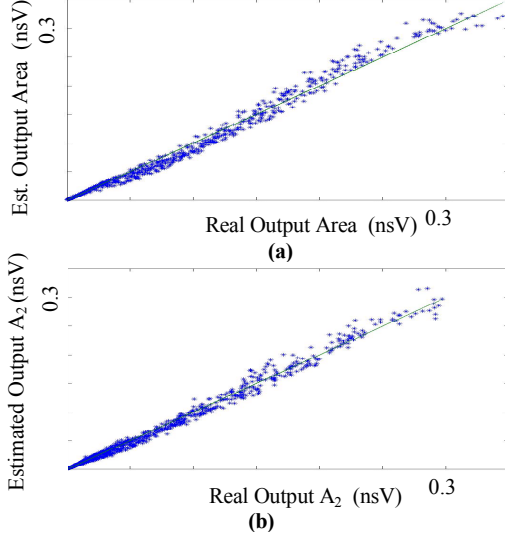


Figure 6. Inverter real versus estimated (a) total Area A, and (b) return-to-nominal area A_2

Table I. Comparing mathematical model results to SPICE

Gate	abs(% Error)					
	output_Amp		output_A		output_A ₂	
	Avg.	Std.	Avg.	Std.	Avg.	Std.
INV	5.6%	5.6%	7.4%	6.7%	7.7%	6.6%
NAND2	4.3%	4.9%	8.3%	6.6%	8.8%	6.5%
NOR2	4.4%	4.6%	8.4%	8.2%	9.1%	8.1%
NAND4	5.1%	5.6%	8.0%	6.0%	7.7%	6.1%

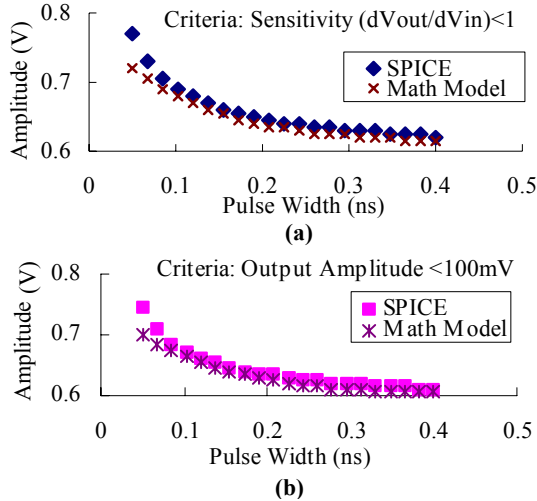


Figure 7. Noise abstracts of a chain of three inverters in Figure 9a with minimum load, fanout=1. For the given pulse widths, we plot the critical input amplitude beyond which a given (a) sensitivity (gain) or (b) out_Amp criteria is violated. We use the mathematical model of a single inverter to predict the response of the chain. The pulses are triangular noise glitches.

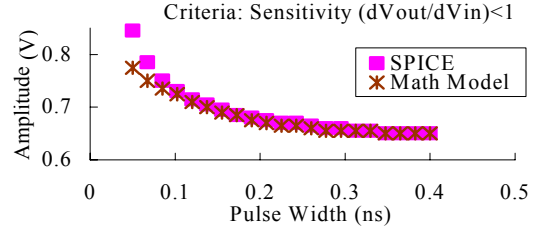


Figure 8. Noise abstract of circuit Figure 9b based on sensitivity criterion. We use the mathematical model of individual NAND2 and NOR2 gates to predict the response. The abstract constrained by output amplitude criteria matched the SPICE results as well.

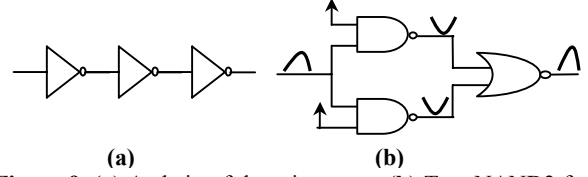


Figure 9. (a) A chain of three inverters. (b) Two NAND2 feeding a NOR2 gate: fanout=1. A LHL noise pulse is injected at the NAND2 gates common input (INV's input). The NAND2 gates have their other inputs connected to V_{DD} .

3. CIRCUIT-EQUIVALENT MACROMODEL

While the mathematical models are robust and able to hierarchically generate noise abstracts of any group of cell, it is sometimes necessary to obtain exact propagated and/or coupling noise waveforms, for example, computing glitch delay for noise windows. This is the focus of the Circuit Equivalent Macromodel.

The basic model consists of voltage dependent capacitors and voltage dependent ideal current sources, and an arbitrary internal impedance. Figure 10 presents the basic model topology and the elements for a single output logic gate. Note that this topology allows for generalization to multiple input and output pins. The gate drive is modeled by the current sources I_p and I_n , one pair per output pin. The gate's intrinsic parasitics are modeled by the capacitors C_m and C_{out} and by the internal impedance Z_{int} . C_m accounts for the Miller effect. C_{in} is the gate input capacitance. C_{out} represents the gate internal capacitance at the output node. For each input (output), there is one C_{in} (C_{out}). There is one C_m for each input-output pair. Z_{int} is introduced to account for the complex internal topology of the modeled gate. The output pin loading Z_{out} is not part of the model.

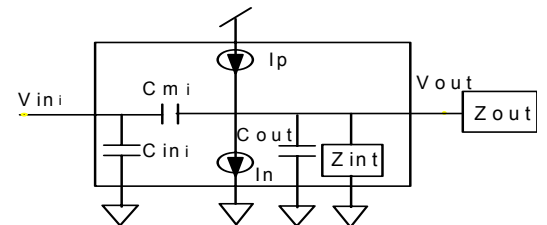


Figure 10. Topology of the noise-sensitive circuit equivalent macromodel.

To achieve noise sensitivity, the model parameters are extracted for all combinations of relevant dc bias voltages at the input and output pins of the gate being modeled. Those combinations span a wide range of operating points sufficient to cover the voltage ranges over which one plans to simulate. Thus,

one has a hyper-grid of the model parameter values at discrete values of all input and out-put bias voltages making our model implicit. Zolotov et al. [5] also found this implicit method beneficial.

Given the circuit topology of Figure 10, Kirchoff's current law (KCL) at the output node can be expressed as an ODE:

$$\frac{dV_{out}}{dt} = \frac{I_p - I_n + (C_m \cdot \frac{dV_{in}}{dt}) - I(Z_{int}) - I(Z_{out})}{C_{out} + C_m}. \quad (6)$$

For the case of multiple inputs, the ODE becomes:

$$\frac{dV_{out}}{dt} = \frac{I_p - I_n + \sum_{i=1}^n (C_m^{(i)} \cdot \frac{dV_{in}^{(i)}}{dt}) - I(Z_{int}) - I(Z_{out})}{C_{out} + \sum_{i=1}^n C_m^{(i)}}. \quad (7)$$

The solution of this implicit ODE is simple, and is much faster than any SPICE-like simulation. During run-time, we know the value of $V_{in}(t)$, and we look-up or interpolate the other parameters.

Thus, one can translate the model into an equation that is differential in nature and implicit with respect to the output. The implicit nature of the ODE greatly improves the accuracy of the solution, and obviates the need for the explicit timing of the delay of the signal across the circuit, allowing us to solve for the output response due to practically any form of input wave, including glitches. Furthermore, the differential nature of the equation, with respect to the outputs and inputs, allows us to account for an output load of variable complexity, without necessitating load dependency in model extraction. Through the C_m , Z_{int} , and C_{in} parameters, it also accounts for all the qualitatively important capacitive effects that influence the accuracy of a transient (noise) waveform. In addition, the model allows for simultaneous input glitches/transitions, and is sensitive to input skew. Finally, without radical changes, other passive devices (resistors, inductors) could be added to the basic model to represent additional physical effects.

3.1. Extracting the Macromodel Parameters

3.1.1. C_m , C_{in} , C_{out} , I_p and I_n for a Given Bias Point

The input and output pins are initially set to the bias point dc values. The internal nodes thus take on their steady-state values. To obtain the capacitor parameter values, we apply small ramps (say, 5 mV) at the output (input) pins and record the resulting capacitive currents. We then solve KCL equations.

To obtain I_p (I_n), we measure the currents flowing from V_{DD} (to ground). Note that we extract I_p and I_n separately for the sake of Z_{int} representation (Section 3.2). Also ($I_p = I_{VDD}$) \neq ($I_n = I_{gnd}$).

3.1.2 Internal Impedance Parameter Z_{int}

Our model accurately predicts an inverter's output response (except for interpolation error). For a complex gate, however, a precise response requires the internal nodes be included in the model and the model parameters be $f(V_{in}, V_{internal-nodes}, V_{out})$. This complicates both the precharacterization and simulation phases, and conflicts with the requirements of an encapsulated high-level model. Thus, alternatively, we rely on a lumped effective parameter, Z_{int} , to account for the internal node effects in order to obtain a conservative estimate of the output response. Since our model drives are derived with no transients on the internal nodes, we treat Z_{int} as a tunable parameter.

We will consider the following possible Z_{int} representations.

1. For the case of "pull-down" noise, set $I(Z_{int}) + I_n = I_{OFF}$, until $t = \tau_{ON}$, i.e., the gate threshold voltage has been reached. The case for pull-up noise is similar. We reduce the problem to that of precharacterizing τ_{ON} in terms of normalized ramps (same slew and skew) and C_L ; this is feasible by mapping the deviate-from-stable part of the noise pulse to a ramp and relying on methods similar to that proposed in [6] to obtain a set of normalized input ramps. For a given placement, we rely on the initial-equivalent C_L value to determine the τ_{ON} value, e.g. C_{NEAR} for a pi-load (see Fig. 12).
2. Set $Z_{int} = C_{lump}$, a tunable lumped capacitance (dependent on the input waveform ranges). One may start with an initial guess equal to the true effective internal node capacitance. Then one may tune it as a function of the input waveform range. This resembles the approach in [7].

3.1.3. Output Load Current $I(Z_{out})$

The output external loading is not part of the model per se; it depends on the placement of the circuit in the larger design. Its presence and details are allowed for by one of two methods.

The first method is the crudest, and fixes the topology of Z_{out} to be that of a static C_L or a pi network. If a "pi model" is used, the $I(Z_{out})$ term in (6) and (7) is determined by (8) and (9). At the beginning of simulation V_2 is set equal to V_{OUT} .

$$I(Z_{out}) = C_{near} \cdot \frac{dV_{out}}{dt} + (V_{out} - V_2) / R_{pi} \quad (8)$$

$$\frac{dV_2}{dt} \cdot C_{far} = (V_{out} - V_2) / R_{pi} \quad (9)$$

The second method supports an interface with other models that supply current sink information due to biases applied at their input pins. At each timestep, the solver will issue a request to the caller, i.e., interface modules, providing the output node voltage and asking it to give back the current, $I(Z_{out})$, sunk into the load circuit. In this case, the ODE presented in (6) is sufficient.

3.2. Simulation Methodology and Results

To build the model, we relied on precharacterized n^p I-V, and C-V tables; n and p are the numbers of steps and pins respectively. For the examples below, $n=10$, and $p=3$ (2-inputs glitching and 1-output). The ODE can be solved numerically by a variety of methods. We chose the 4th order Runge-Kutta method for its high accuracy. We also relied on linear interpolation techniques.

We compared the output responses predicted by our simulator to that of SPICE for NAND2, NAND4, NOR2 and AOI22 gates. The design environment is similar to that of Section 2.3. The gates were biased for worst-case pull-down noise analysis; $C_L = 3.5fF$, and $I(Z_{int}) = 0$. For each gate we performed 100 experiments; the amplitudes and widths of the input noise pulses were randomly selected over the ranges [0.4V-0.9V] and [50ps-600ps], respectively. In all the experiments, our model-generated and the SPICE-generated output waveforms were in good agreement, and our simulation results were conservative (overestimate). The statistics are summarized in Table II, our metrics being absolute non-Miller region (see Figure 1) area and peak amplitude.

For the test cases the level of pessimism was acceptable and there was no need to use Z_{int} . In Figure 11, we present a case where Z_{int} correction, as was suggested in method 2 of Section 3.1.2, helps improve the prediction. Finally, in Figure 12 we compare our model simulation results against those of SPICE when non-identical noise glitches are applied at the inputs of a NOR2 (12a), and when the injected noise due to an aggressor at the output node and propagated noise interact nonlinearly (12b).

Input to output pulse delay is used to align the coupling noise to generate the maximum output noise.

With regard to run-time, our unoptimized (prototype) code was around two orders of magnitude faster than SPICE for the gates tested. We expect that further optimizing the code and relying on the wealth of research available in the simulation area will improve the speed-up substantially.

Table II. Comparing our simulator to SPICE

Gate	#Inputs Glitching	OutAmp		OutArea	
		Avg_err	Std.Dev.	Avg_err	Std.Dev.
NAND2	1	+7.6%	4.9%	+11.7%	3.3%
NAND4	1	+8.7%	4.1%	+12.4%	4.0%
NOR2	2	+2.2%	3.0%	+6.5%	2.45%
AOI22	2	+5.0%	5.6%	+8.8%	3.7%

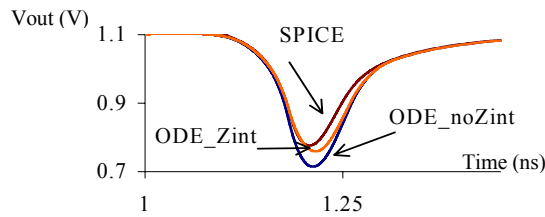


Figure 11. NAND2 gate with the input closer to ground glitching; the internal node is initially fully charge. An illustration of the effect of Z_{int} correction.

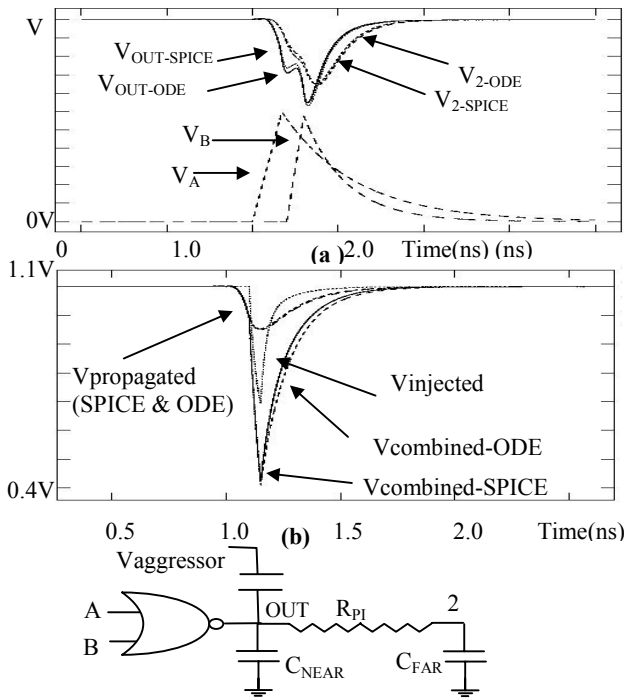


Figure 12. (a) Independent glitches are injected at both inputs of a NOR2 gate with pi-load. (b) The injected noise due to an aggressor at the output node and propagated noise interact nonlinearly (combined effect is greater than that due to linear superposition).

4. SUMMARY AND CONCLUSIONS

We proposed two macromodeling approaches for purposes of building a noise-rule library. Our models capture the cell's output response due to noise at its input. Thus, one may accurately predict the propagated noise, perform sensitivity (stability) analysis, or even hierarchically build the noise abstracts. The first is a mathematical model suitable for static noise analysis. The model can thus be used to predict the propagated noise and the AC gain, to construct the noise abstract for any specified lumped load, and to hierarchically construct the noise abstracts of a group of cells using the macromodels of individual cells. This is done with high accuracy and speed.

The second is a circuit-equivalent model, which can be used to generate almost SPICE-accurate noise waveforms in the presence of coupling, arbitrarily timed and shaped multiple inputs glitching, and arbitrary loads. These models can be used to compute glitch delays for noise windows, and to compute the effect of coupling on total noise in the presence of device nonlinearities and arbitrary loads.

5. ACKNOWLEDGEMENT

The authors would like to thank Ron Rose of IBM Corporation for many useful discussions. The authors would also like to thank IBM Corp. for funding this research through a fellowship grant.

6. REFERENCES

- [1] K. L. Shephard, V. Narayanan, and R. Rose, "Harmony: static noise analysis of deep submicron digital integrated circuits", IEEE Trans. on CAD, Vol. 18, no. 8, pp. 1132-1150, Aug. 1999.
- [2] R. Levy, D. Blaauw, G. Braca, A. Grinshpon, C. Oh, B. Orshav, V. Zolotov, "Clarinet: A noise Analysis Tool and Methodology for Deep-Submicron Design", ACM/IEEE DAC pp. 233-238 June 2000.
- [3] K. Tseng and V. Kariat, "Static Noise Analysis with Noise Windows", 40th ACM/IEEE DAC, pp. 864-868, June 2003.
- [4] K. L. Shepard and K. Chou, "Cell characterization for noise stability", IEEE CICC, pp. 91-94, May 2000.
- [5] V. Zolotov et al., "Noise Propagation and Failure criteria for VLSI designs", IEEE ICCAD, pp.587-594, Nov. 2002.
- [6] A. Chatzigeorgiou, S. Nikolaidis, and I. Tsoukalas, "A modeling technique for CMOS gates", IEEE Trans. on CAD, Vol. 18, Issue 5, pp. 557-575, May 1999.
- [7] A. Nabavi-Lishi and N.C. Rumin, "Inverter models of CMOS gates for supply current and delay evaluation", IEEE Trans. on CAD, Vol. 13, Issue 10, pp. 1271-1279, Oct. 1994.
- [8] J. Croix, and D. F. Wong, "Blade and Razor: Cell and Interconnect Delay Analysis Using Current-Based Models", 40th ACM/IEEE DAC, pp. 386-389, June 2003.
- [9] R. Gunst and R. Mason, *Regression Analysis and Its Application: A Data-Oriented Approach*, New York: Marcel Dekker Inc., 1980.
- [10] J. Shao and R. Harjani, "Macromodeling of Analog Circuits for Hierarchical Circuit Design," IEEE/ACM ICCAD, pp. 656-663, Nov. 1994.
- [11] G. Box, W. Hunter and J. Hunter, *Statistics for Experimenters: an Introduction to Design Data Analysis and Model Building*, John Wiley, 1978.
- [12] SAS v. 8.0, <http://www.SAS.com>.