Driver Models For Timing And Noise Analysis

Bogdan Tutuianu and Ross Baldick

Abstract:

In the recent years, the impact of noise on chip level signals has become a significant source of static timing errors. This paper presents a new technique to generate accurate non-linear driver models which can be used for static timing and noise analysis, with inductive interconnect and multi-source nets. The new technique is efficient because it relies on existent gate characterization for timing, does not require additional non-linear circuit simulations and generates re-usable models.

Introduction:

One of the problems that has gathered much attention recently is the effect of switching noise on chip level timing (delay noise or dynamic noise). Static timing analysis determines the extremes of signal propagation, being the main tool used for predicting the speed performance of the digital IC's. Since switching noise can overlap with and affect logic signals, it will directly impact the chip level timing and the reliability of the final product. A good description of the different types of noise, their impact on circuit activity and ways to model and analyze it is given in [23]. Other tools and methodologies for functional noise analysis are proposed in [19], [10] and [1]. Special circuit modeling techniques to asses global noise impact have been proposed in [12], [13] and [18].

The impact of switching noise on chip level timing is generally split into functional noise and delay noise. Functional noise is noise induced in quiet nets (victims) by switching neighbors (aggressors). For high levels of induced currents, it can cause unwanted logic activity and even functional failures. The delay noise is caused by the same switching activity on the neighboring nets, but it happens while the victim net is itself active. In this case the noise can modify the time of flight and slew-rate of the useful signal and it can cause delay (timing) errors.

Switching noise analysis is performed in two steps: a first stage where all possible aggressors are considered, some being filtered based on functional constraints, clock domains, timing windows, etc., and a second stage where the actual effect of noise on delay is being determined through circuit simulation. Most of the research in this area has been focused on the first step,

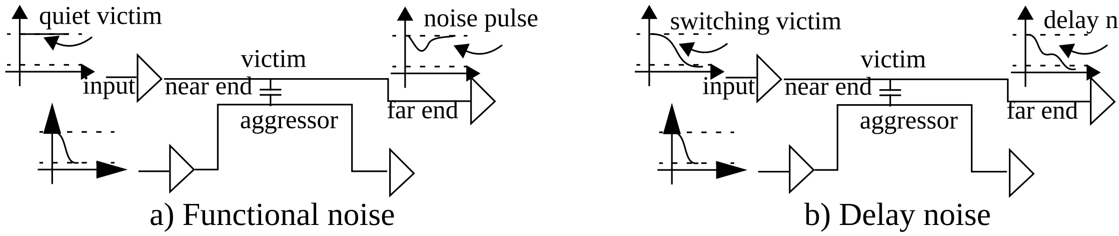


Figure 1**:** Capacitively and/or inductively coupled nets interact with each other. a) interaction with quiet victim net driver: the aggressor will induce a noise pulse on the victim net (functional noise). b) interaction with active victim net driver: the aggressor will induce a variation in the victim net signal shape (delay noise).

mainly on the alignment of the aggressor noise signals for worst/best case analysis and convergence of the timing analysis in the presence of noise [3], [4], [7], [22], [24], [15], [25] and [26]. In this work, our attention is focused on the second step, mainly on the derivation of efficient and accurate logic gate models for noise analysis. In this area, the existing models can be separated into two groups:

a) linear timing models: in [9], [2] and [15] the authors have developed linear gate models that include current injected by aggressors, based on the static timing gate models developed earlier by [21] and [8], and,

b) best-fit resistance models: an analytic model based on the equivalent resistance of the pull up/down transistor chain proposed in [6] and a “transient holding” resistive model proposed in [24].

In the case of functional noise, since the victim net driver is holding low or high, the driver is correctly approximated by a linear (RC) model and the analysis is reduced to linear circuit simulation. In the case of delay noise, functional noise-*like* analysis is used to determine a worst-case alignment of the aggressor noise pulses which are then “merged” with the victim net logic signal. In the “merging” step it is crucial to take into account the very complex non-linear interaction between the driver gate and noise injected from aggressors. This complex interaction is modeled by an iterative process which tries to match the current (charge) injected into the driver. In the case of [9], [2] and [15] the delay/charge measurements with an effective capacitance must match the delays of the perturbed circuit. In the case of [24] the area under the noise pulse must be matched by the area obtained with a “transient holding” resistance model of the driver. In the case of [6], the driver is modeled by a simple pull-up/down resistance derived from the physical devices.

In this paper we present a new logic gate modeling technique well suited for the basic static timing and functional noise analysis as well as accurate delay noise analysis. Our proposed solution is a non-linear dynamic model of the gate driving port, controlled by both input and output signals. Some of the distinguishing features of our modeling technique are: a) our models are derived for a range of input and output conditions so they are re-usable, b) the modeling process is based on the existent delay measurements taken during the pre-processing step of the static timing analysis flow and no extra characterization work is needed, c) the modeling technique allows the user to control the accuracy of the models being generated.

In Section 1 we are discussing the differences between functional noise and delay noise. In Section 2 we are giving a brief presentation of the existing gate models used in static timing and noise analysis and an introduction to the Finite Elements Method (FEM) used at the core of our modeling technique. The new modeling technique is described in detail in Section 3 followed by results (Section 4) for various test cases in which the new models are used to determine the impact of noise on delay, the propagation of signals in nets with multiple drivers, the response of gates with inductive output loads and others. In Section 5 we are reviewing the major contributions of the proposed modeling technique and setting goals for future work.

1. Functional noise vs. delay noise.

In all the examples shown in the paper we have used the same victim driver, a medium sized 4-input NAND gate from a Motorola MPC755 PowerPC-Compatible microprocessor. The gate has 48 transistors, 6 diodes, 725 parasitic capacitors and 322 resistors. In all examples, the active input signal has been A_1 while all others (A_0, A_2 and A_3) were tied to Vdd. The gate is driving a long wire routed on the upper layers of metal (M5). For the aggressor net, we have used a strong inverter as driver and it has been routed along to victim at minimum spacing (about 40% of total wire capacitance is coupling to its neighbor). The two nets were coupled for 75% of the victim net’s length. The aggressor signal has been offset such that its effect is overlapping the far end victim signal obtained in the absence of noise. In Figure 3, the victim input signal and the victim far end signal without noise are shown. In addition, the functional noise has been determined with the victim driver holding low. Then the far end signal in the presence of noise has been determined using accurate spice simulation of the full circuit (“with delay noise” waveform) and by adding the functional noise to the far end signal without noise (“with functional noise” waveform). It is apparent from figure 3 that functional noise can be a very poor estimate of delay noise.

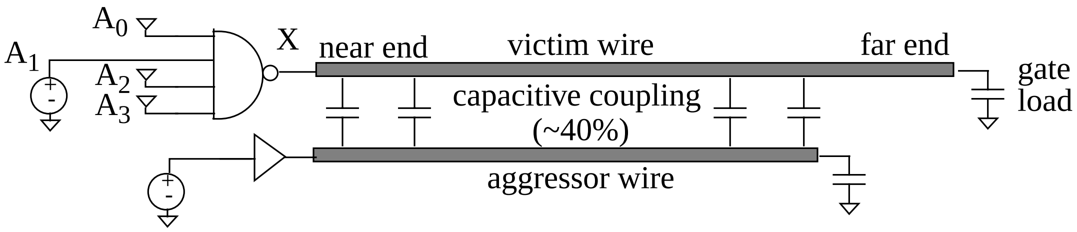
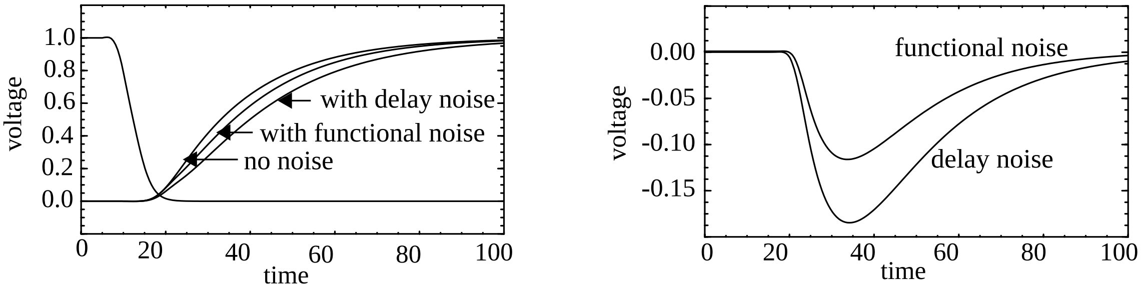


Figure 2**:** Test circuit used throughout this paper. A NAND4 gate driving a very long wire coupled to an aggressor. Each net’s load is within the range specified for its driving gate.



a) far end signals

b) delay and functional noise

Figure 3**: a)** The effect of switching noise on delay **b)** Comparison between functional noise and delay noise.

The last stage of the gate, the driving port, can be seen as comprised of two variable resistors: one modeling the P FETs and one for the N FETs. These two resistors will have opposite variation during the transition of the output and, as a consequence, any noise pulse injected in the output pin will see this variable resistive path to ground. In Figure 4 we are showing the equivalent driver resistance seen at the output port during the output transition (for the input-output signals pair shown in Figure 3). Note how the resistance varies between 100 and 1000 ohms during the active interval. Since one of the existing methods to model the driver for delay noise [24] relies on computing a “transient holding resistance”, it is clear from Figure 4 that such a single value resistor model will not be able to accurately capture the complex driver behavior.

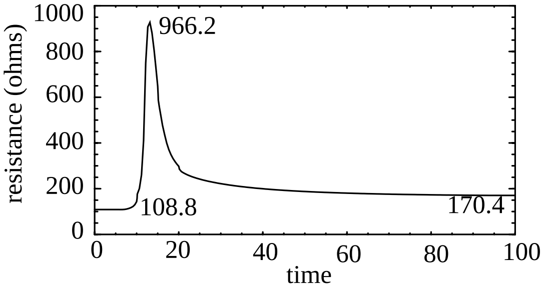


Figure 4: The equivalent resistance seen at output pin during output transition.

2 Background

In section 2.1 we are giving a succinct presentation of the different linear driver models currently used in timing and noise analysis. In Section 2.2 we give a brief introduction to the Galerkin method for Finite Elements, which is being used at the core of our modeling technique.

2.1 Logic gate models for static timing and noise analysis.

In the timing pre-characterization process of a logic block, detailed simulations of all the possible signal paths are performed for different input signals and output loads. The delay measurements are stored in table format or even post-processed as delay equations. This delay data (equations) is usually generated for simple output capacitive loads. However, due to interconnect resistance and inductance, the output load of the gate is modeled by complex RLC circuits which vary from simple Π models to high order models. During timing analysis, the simple delay data (equations) is used to generate driver equivalent circuit models using an iterative process (often called C-effective algorithm). These models were first developed by [21], later their accuracy has been greatly improved by [8].

In the C-effective algorithm, the driver is modeled by a Thevenin like circuit: an ideal voltage source - step or saturated ramp- and a driver equivalent resistance (Figure 5). The iterative procedure tries to determine an “effective” output capacitance load such that for a specific time interval the total charge stored on the simple capacitance is the same as the total charge stored on the complex load and the delays and rise times derived from pre-characterized data for this simple effective capacitance match the ones obtained through the simulation of the linear driver model and the Π RC load:

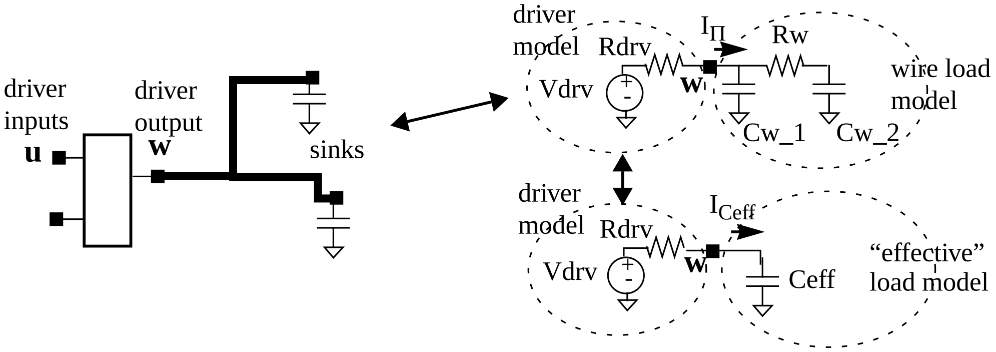


Figure 5: Driver logic gate is modeled (using C-effective) by a Thevenin like linear circuit while the interconnect input impedance is modeled (using AWE) by a Π model and transfer functions between driver output and sink pins.

$$Q_\Pi = \int_{t_0}^{t_1} I_\Pi(t)dt = Q_{Ceff} = \int_{t_0}^{t_1} I_{Ceff}(t)dt \quad ,$$

$$TD_\Pi = GateTD(TX_u, Ceff) \qquad TX_\Pi = GateTX(TX_u, Ceff)$$

where *Q* is charge, *I* is current, *TD* is delay time, *TX* is rise time, *GateTD* and *GateTX* are the gate delay and rise time coming from pre-characterized data.

In practice, the C-effective technique is stable and converges rapidly and it has been in use for almost a decade with different flavors in most static timers, commercial as well as corporate

EDA tools. Switching noise effects, on-chip interconnect inductance and multiple source nets are relatively recent issues for static timing and there has been significant work done to extend this simple algorithm to these cases. In [2] the authors are extending the algorithm to general output load models in reduced order format [20][14]. Later [17] and [16] developed an extension of the RC Π model to a stable RLC Π model with good accuracy for chip level timing.

For delay noise analysis there are a couple of models proposed in the literature:

1) An extension of the C-effective algorithm to model the injected current as additional capacitive load [9]. The C-effective algorithm is applied simultaneously to the victim and the aggressor resulting in a system of non-linear equations solved efficiently using the successive-chord method. It is worth noting that the same algorithm can be applied to the situation of a single net with multiple sources.

2) The “transient holding resistance” model proposed in [24] which models the reaction of the gate to the injected current with the help of a fitted resistance. Each iteration contains the following steps: a) for each aggressor in isolation (with victim and other aggressors grounded) the currents injected in the victim is recorded; b) a non-linear simulation is performed to determine the response of the gate with the induced current at its output. From the comparison of this output with the one obtained in the absence of noise a delay noise pulse is obtained; c) a “transient” resistance value for the victim driver is then computed to match the area of the delay noise with a functional noise pulse.

In [24] the authors have compared the two methods and reported much better results for the later technique. It is important to note that, in order to be accurate, the second modeling technique requires a non-linear circuit simulation at each iteration.

2.2 Introduction to finite elements method

The finite elements method is being used extensively in engineering (e.g. for solving field equations, in various civil and mechanical engineering problems and electronic device parameter modeling). The success of FEM comes from its simplicity and flexibility. Furthermore, the method can be used very efficiently (such as the Galerkin method) by reducing the non-linear dynamic problems to simple linear systems of equations. In this section we are giving a very brief introduction to finite elements tailored to the Galerkin method. This introduction follows closely the treatise of the subject from [5].

To introduce the basic concepts from FEM we take a simple one-dimensional differential equation with essential boundary conditions:

Find $g = g(t)$*, a real valued function defined on a finite domain* $\Gamma = [t_m, t_M] \subset \Re$*,* $g:\Gamma \to \Re$*, which satisfies the following differential equation:*

$$g''(t) + c_1 \cdot g'(t) + c_0 \cdot g(t) = f(t) \tag{1}$$

with the given boundary conditions:

$$g(t_m) = g_m \text{ and } g(t_M) = g_M. \tag{2}$$

The finite elements method relies on the possibility to approximate any function within a desired accuracy limit as a combination of certain building functions also called basis functions. For example, any polynomial of order three or less $P_3(x) = a_3x^3 + a_2x^2 + a_1x + a_0$ can be exactly and uniquely represented using the following family of basis functions: $p_3(x) = x^3$, $p_2(x) = x^2$, $p_1(x) = x$ and $p_0(x) = 1$.

Let us assume that we have such a family of basis functions $\boldsymbol{B} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ and that the solution to our problem is sought to be in the following form: $\tilde{g}(t) = \sum_{i=1}^{n} a_i \cdot \alpha_i(t)$. In order for this set of basis functions to provide a reasonable approximation of the solution, each basis function must be continuous, bounded and twice differentiable on Γ. In order to simplify the interpolation, the "solution" is defined only on a set of nodes $T = \{t_1, t_2, \ldots, t_{n-1}, t_n\}$ within Γ with $t_1 = t_m$ and $t_n = t_M$. As our natural choice for a set of basis functions we use the Lagrange interpolating polynomials. For our domain Γ with *n* nodes we define *n* basis functions such that each one is 1 in one node and 0 in all others. The family of basis functions is defined as:

$$\alpha_i(t) = \left(\prod_k (t - t_k)\right) \Big/ \left(\prod_k (t_i - t_k)\right) \qquad 1 \le k \le n \text{ with } k \ne i. \tag{3}$$

In Figure 6 we give examples of Lagrange interpolating polynomials for 2, 3 and 4 nodes in the domain which corresponds to first, second and, respectively, third order polynomials. If, for example we have measurements of a function $h = h(t)$ in every point t_i, i.e $h(t_1) = h_1$, $h(t_2) = h_2$, etc., the approximation of *h* using the Lagrange polynomials is simply: $\tilde{h}(t) = \sum_{i=1}^{n} h_i \cdot \alpha_i(t)$.

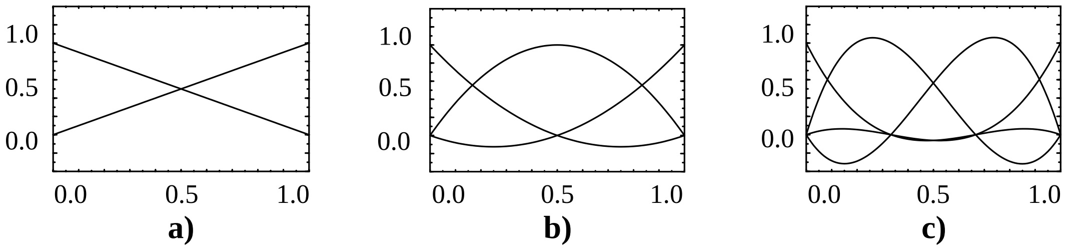


Figure 6**:** First, second and third order Lagrange polynomials as basis functions.

The FE method first transforms the differential equation into an integral equation by noting that if equation (1) is identically satisfied by the solution then the following form:

$$\int_{t_m}^{t_M} \tau(t) \cdot (g''(t) + c_1 \cdot g'(t) + c_0 \cdot g(t) - f(t))dt = 0 \tag{4}$$

holds for any test function τ(*t*), $\tau(t) = \sum_{i=1}^{m} b_j \cdot \beta_j(t)$ defined also over a set of basis functions $\boldsymbol{B_{test}} = \{\beta_1, \beta_2, \ldots, \beta_m\}$ which must be continuous, bounded and at least once differentiable on Γ. Integrating by parts the second order derivative term of equation (4) we get:

$$\int_{t_m}^{t_M} (-\tau'(t) \cdot g'(t) + \tau(t) \cdot (c_1 \cdot g'(t) + c_0 \cdot g(t) - f(t)))dt + \tau(t) \cdot g'(t)\Big|_{t_m}^{t_M} = 0 \tag{5}$$

Equation (5) must hold for any choice of a test function $\tau(t)$ and that gives us the possibility to choose test functions that are identically equal to 0 at the boundary points which will cancel the extra term $\tau(t) \cdot g'(t)\big|_{t_m}^{t_M}$. At the same time, since the test function can be expressed as a combination of basis functions β_j we can rewrite equation (5) as:

$$\sum_{j=1}^{m} b_j \cdot \left(\int_{t_m}^{t_M} (-\beta_j'(t) \cdot g'(t) + \beta_j(t) \cdot (c_1 \cdot g'(t) + c_0 \cdot g(t) - f(t)))dt \right) = 0 \qquad \forall b_j \in \Re \quad (6)$$

which is valid for any b_j values and that gives us m independent equations:

$$\int_{t_m}^{t_M} (-\beta_j'(t) \cdot g'(t) + \beta_j(t) \cdot (c_1 \cdot g'(t) + c_0 \cdot g(t) - f(t)))dt = 0 \qquad j = 1, \ldots, n \qquad (7)$$

If the approximation of the actual solution is $\tilde{g}(t) = \sum_{i=1}^{n} a_i \cdot \alpha_i(t)$, equation (7) becomes:

$$\int_{t_m}^{t_M} \left(-\beta_j'(t) \cdot \sum_{i=1}^{n} a_i \cdot \alpha'_i(t) + \beta_j(t) \cdot \left(c_1 \cdot \sum_{i=1}^{n} a_i \cdot \alpha'_i(t) + c_0 \cdot \sum_{i=1}^{n} a_i \cdot \alpha_i(t) - f(t) \right) \right) dt = 0 \qquad (8)$$
$$j = 1, \ldots, m$$

$$\sum_{i=1}^{n} a_i \cdot \left(\int_{t_m}^{t_M} (-\beta_j'(t) \cdot \alpha'_i(t) + \beta_j(t) \cdot (c_1 \cdot \alpha'_i(t) + c_0 \cdot \alpha_i(t)))dt \right) = \int_{t_m}^{t_M} \beta_j(t) \cdot f(t)dt \qquad (9)$$
$$j = 1, \ldots, m$$

Equation (4) has been reduced to a system of linear equations (9) which can be written as $\boldsymbol{Q} \times \boldsymbol{a} = \boldsymbol{r}$ where $\boldsymbol{Q} = [q_{ij}]\big|_{i \le n \ \& \ j \le m}$ is a $m \times n$ matrix with its entries defined as:

$$q_{ij} = \int_{t_m}^{t_M} (-\beta_j'(t) \cdot \alpha'_i(t) + \beta_j(t) \cdot (c_1 \cdot \alpha'_i(t) + c_0 \cdot \alpha_i(t)))dt, \qquad (10)$$

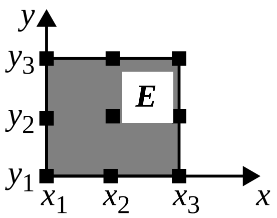
$\boldsymbol{a} = \begin{bmatrix} a_1 & \ldots & a_n \end{bmatrix}^T$ is the vector of scalar coefficients for the solution and $\boldsymbol{r} = \begin{bmatrix} r_1 & \ldots & r_m \end{bmatrix}^T$ is the vector of right hand side terms with each entry defined as:

$$r_j = \int_{t_m}^{t_M} \beta_j(t) \cdot f(t)dt. \qquad (11)$$

At this point, the remaining step is to evaluate the integrals of equations (10) and (11), usually done through Gaussian integration.

There is a trade-off between the number of nodes and the accuracy of the approximation. In order to keep the computational cost low, the Γ domain is usually split into elements as $\Gamma = \bigcup_p \boldsymbol{E}_p$. This allows us to use low order basis functions and low order Gaussian integration.

In the case when there are two variables (as in our models), the most flexible domain partition is obtained with triangular elements. However, this is cumbersome for the automated modeling process and rectangular elements are used instead. The Lagrange interpolating polynomials in two dimensions are also straightforward to obtain as products of one-dimensional polynomials. In Figure 7 we are showing a rectangular element with 9 nodes and the expression of a two-



$$\psi_{2,3}(x,y) = \alpha_2(x)\cdot\alpha_3(y)$$
$$= \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}\cdot\frac{(y-y_1)(y-y_2)}{(y_3-y_1)(y_3-y_2)}$$

Figure 7: An example of a rectangular element (***E***) for the two-dimensional case. The basis function ($\psi(x,y)$) that takes value 1 in the point (x_2,y_3).

dimensional basis function derived using second order one-dimensional basis functions. In Figure 8 we are plotting two of these basis functions, $\psi_{1,1}(x,y)$ and $\psi_{2,2}(x,y)$. The use of two-dimensional Lagrange interpolating polynomials on rectangular elements (bi-linear, bi-quadratic basis functions, etc.) guarantees the continuity of the approximation at the boundary between elements.

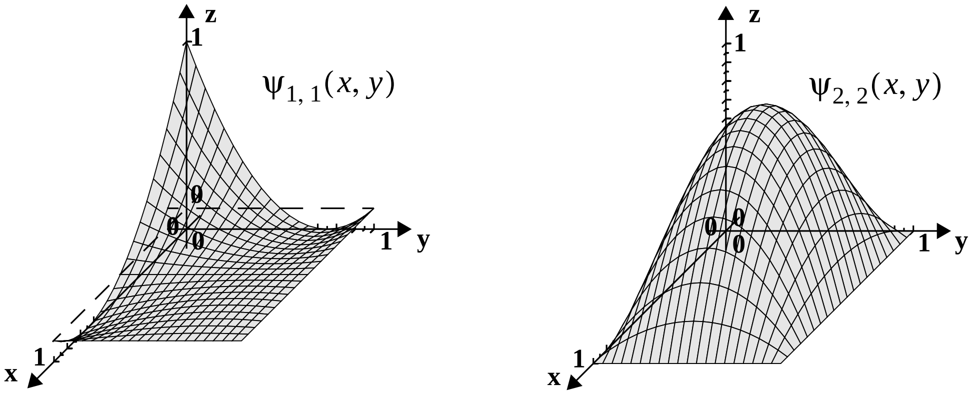


Figure 8: Two dimensional basis functions using second order one-dimensional Lagrange interpolating polynomials.

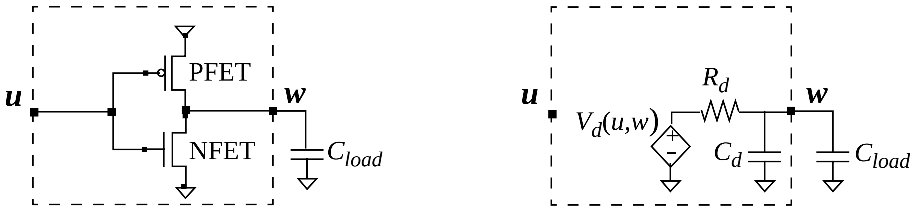
3. Non-linear driver models for timing and noise analysis.

The switching noise pulses inject/draw charge in/from the victim net, effectively changing the size of the interconnect load seen by the victim net driver. As a consequence, the driver response depends simultaneously on the input signal and the noise pulse and it is not possible to separate these effects without incurring errors. Our solution is a simple non-linear model which has either a Thevenin or a Norton form. In the following, the Thevenin type model is used to present the modeling process and its properties.

This Section is divided in two sub-sections: the main steps of our modeling technique are presented in section 3.1 followed by a discussion on the properties of our models in section 3.2.

3.1 The proposed modeling technique

In the Thevenin form, the driver model is comprised of a non-linear voltage source, controlled simultaneously by the input pin voltage and the output pin voltage, and a fixed value impedance (resistance and capacitance) (Figure 9).



a) Simplified real driver b) Proposed model

Figure 9**:** a) Real driver (in its simplest form as an inverter) and b) its non-linear model (shown here in Thevenin form).

For any input signal (u) and any output capacitive loads (C_{load}) we can determine from the pre-characterized data the response of the gate (w) as delay values on pre-defined voltage levels (usually the 10%, 50% and 90% delays). This pre-characterized data is stored in delay tables or curve-fitted delay equations. If we were to simulate the circuit from Figure 9.b, we would have a single node with the following Kirckhoff current equation:

$$V_d(u, w) - w = R_d C \frac{dw}{dt} \qquad (12)$$

where $C = C_{load} + C_d$. R_d and C_d are modeling the holding high/low output port admittance and are considered known for the rest of this section. It remains to determine the expression of $V_d(u,w)$ such that the output voltage that satisfies the above equation *is similar at the measurement points* with the pre-determined output data. We assume that V_d is fully described by a collection of points V_{ij} in a domain $\boldsymbol{D}$ defined by: $\boldsymbol{D} = \{(u, w) | u \in (u_{min}, u_{max}), w \in (w_{min}, w_{max})\}$ (Figure 10).

The current equation of the output node can be re-written in integral form (equation (4)):

$$\int_{tmin}^{tmax} \tau \cdot \left(V_d(u, w) - w - R_d C \frac{dw}{dt} \right) dt = 0 . \qquad (13)$$

At this point we must explain an important difference between the traditional FE method and our process: the former is applied to solve a differential equation (i.e. to find w, the function

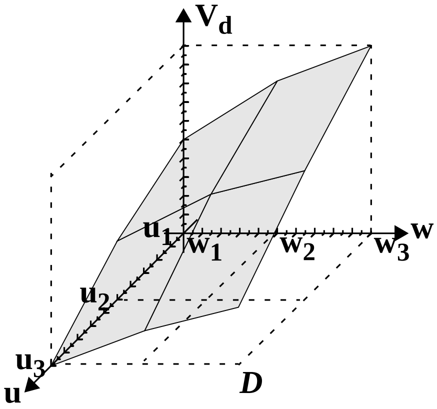


Figure 10: A simple voltage source model defined on a grid as a function of the input (**u**) and output (**w**) signal values as a PWL function of **u** and **w**.

under the difference operator) while we actually have the solution, but we do not know the functional coefficients of the equation (i.e. V_d). In some sense, we are applying the FE method in reverse. Our goal is to find a representation of $V_d(u, w)$ which satisfies the differential equation at the measurement points (u_i,w_j). If the values $V_{ij} = V_d(u_i, w_j)$ are known, then its approximation is:

$$\tilde{V}_d(u, w) = \sum_{i,j} V_{ij} \cdot \psi_{ij}(u, w) \qquad i = 1, \ldots, N_u \text{ and } j = 1, \ldots, N_w \quad , \tag{14}$$

where each ψ_{ij} is the two-dimensional Lagrange interpolation polynomial. The same interpolation process is applied to all other time dependent functions: u, w and τ. For a particular input-output signal pair, Figure 11, the time domain is partitioned by the measurement points. For the end points of the time domain, *tmin* is defined by the starting point of the input signal (ti_0) and *tmax* is usually defined by the end point of the output signal (to_{100}). For example we can express the input signal as:

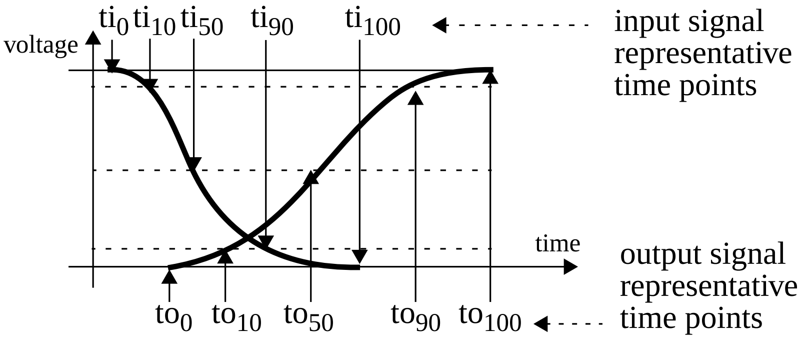


Figure 11: An input-output signal pair with representative time points. In addition to the three measurement points, we have the start and end time points.

$$u(t) = \sum_i u_i \cdot \alpha_i(t) \qquad u_i = u(t_i) \qquad t_i \in \{TI\} = \{ti_0, ti_{10}, ti_{50}, ti_{90}, ti_{100}\}, \tag{15}$$

and the output as:

$$w(t) = \sum_j w_j \cdot \alpha_i(t) \qquad w_j = w(t_j) \qquad t_j \in \{TO\} = \{to_0, to_{10}, to_{50}, to_{90}, to_{100}\}\,. \tag{16}$$

Note that both input and output are defined by measurements (u_i,w_j) which are taken on predefined voltage levels, i.e. the values u_i and w_j are known a priori.This is why the Galerkin method is perfectly complemented by the pre-characterization process for timing: the former needs point values which is exactly what the later provides.

The test function is also expressed using basis functions (β) which may be different from the α basis functions:

$$\tau(t) = \sum_k \tau_k \cdot \beta_k(t) \qquad \tau_k = \tau(t_k) \qquad t_k \in \{TI \cup TO\} \qquad k = 1, \ldots, I + J\,. \tag{17}$$

When all the functions are expressed using basis functions, equation (13) becomes:

$$\sum_k \tau_k \cdot \left(\int_{tmin}^{tmax} \beta_k(t) \cdot \left(\sum_{i,j} V_{ij} \cdot \psi_{ij}(u, w) - \left(\sum_j w_i \cdot \alpha_i + R_d C \sum_j w_i \cdot \alpha'_j \right) \right) dt \right) = 0\,. \tag{18}$$

Since we can choose any test functions, equation (18) must be identically satisfied for any choice of τ_k coefficients, being equivalent to a system of equations. With some more algebraic manipulation, each equation of the system can be described as:

$$\sum_{i,j} V_{ij} \cdot \left(\int_{tmin}^{tmax} \beta_k \cdot \psi_{lm}(u, w) dt \right) - \sum_j w_j \cdot \left(\int_{tmin}^{tmax} \beta_k \cdot \alpha_j dt - R_d C \cdot \int_{tmin}^{tmax} \beta_k \cdot \alpha'_j dt \right) = 0 \tag{19}$$

where $k = 1, \ldots, I + J$. Every equation can be concisely written as: $\sum_{i,j} V_{ij} \cdot \phi_{ij}^k - \nu^k = 0$

which is part of the system of equations: $\Phi \times \boldsymbol{V} = \vartheta$ where all the ϕ coefficients are ordered in the matrix Φ, all the unknown voltage points V_{ij} are ordered in the vector $\boldsymbol{V}$ and all the free terms are in ϑ.

$$\phi_{ij}^k = \int_{tmin}^{tmax} \beta_k \cdot \psi_{ij}(u, w) dt \qquad \nu^k = \sum_j w_j \cdot \left(\int_{tmin}^{tmax} \beta_k \cdot \alpha_j dt - R_d C \cdot \int_{tmin}^{tmax} \theta_k \cdot \alpha'_j dt \right) \tag{20}$$

By solving the system of equations (20) we can obtain the set of voltage points that define our voltage source model.

The number of equations obtained in this process must be related to the number of elements needed for the V_d function. Since one input-output signal pair will provide a limited number of equations, we have to extend the analysis to more than one pair. It is easier to understand that by visualizing every input-output signal pair as a path in the input-output domain ***D***. In Figure 12.a we are showing a typical set of paths for an inverter with rising input and falling output. These paths can be obtained by varying the input signal and/or the output pin capacitive loads. In order to cover the lower left region of the domain we need to take other paths into account with falling input and rising output (Figure 12.b). In order to better model the hold-up and hold-down resistances of our model we need to better cover the lower right and upper left corners of the domain which can be done with static noise signals on the input. Their distinctive paths are shown in Figure 12.c. Note that the points known from measurements (marked with black squares) are situated on one or both of the measurement thresholds. In the case of noise characterization, other

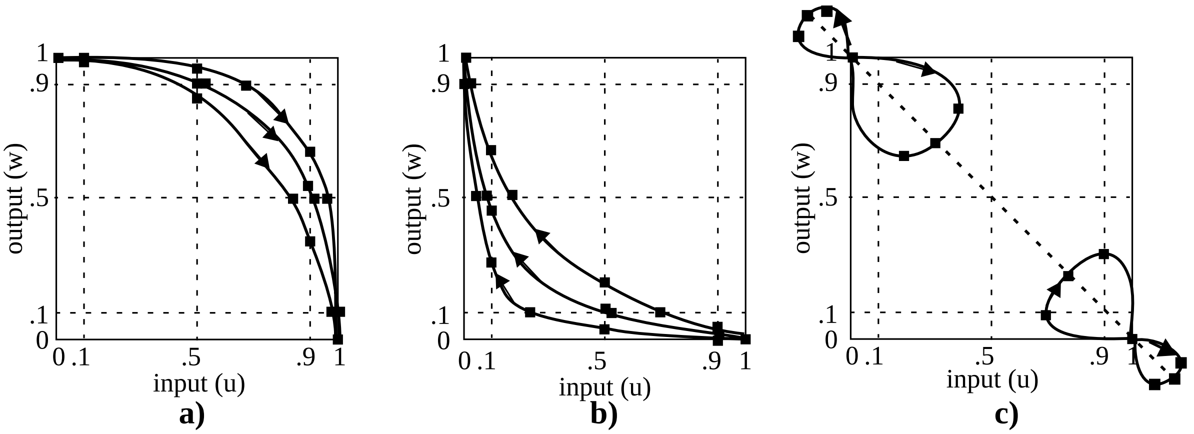


Figure 12**:** Input-output signal pairs as paths through the domain of an invertor: a) rising input falling output paths, b) falling input rising output paths and c) static (positive and negative) noise paths for output holding high and low.

measurement rules can be applied. For example we are interested in the peak value of noise both on input and output. Another point easy to describe is the point where input noise and output noise pulses have the same height (points situated on the diagonal of the domain).

It is apparent from the distribution of points that we may need more accurate models of the gate in some regions of the domain while others are sparsely populated and/or used. The $\boldsymbol{D}$ domain can be split into elements in various ways. It is more efficient and more accurate to use the measurement thresholds as boundaries between elements because in that case we have precise information about the time-points at which the paths are traversing the element boundaries.

The variety of basis functions and the flexibility in the choice of a domain partition provides us with the adequate means of controlling the accuracy of our models. Depending on the application, the user can choose to fit a model to a larger number of data points (equations) and can use curve-fitting techniques such as Singular Value Decomposition to generate optimal (in the least square sense) driver models.

3.2 Properties of the proposed non-linear driver model

In a practical implementation of our driver models in the delay noise analysis flow, one must pay attention to the stability and convergence properties.

We will define our model as follows:

Definition: Given a domain $\boldsymbol{D} = \{(u, w) | u \in (u_{min}, u_{max}), w \in (w_{min}, w_{max})\}$ we define the driver model to be the port current function:

$$\boldsymbol{I_{out}}:\boldsymbol{D} \rightarrow \mathfrak{R} \qquad \text{with} \qquad \boldsymbol{I_{out}}(u, w) = \frac{\boldsymbol{V_d}(u, w) - w}{\boldsymbol{R_d}} - \boldsymbol{C_d} \cdot \frac{dw}{dt} \tag{21}$$

given $\boldsymbol{R_d}$ and $\boldsymbol{C_d}$ and $\boldsymbol{V_d}(u, w) = \sum_{i, j} V_{ij} \cdot \psi_{ij}(u, w)$.

In Figure 13 the dc output port current of the NAND4 gate is plotted with respect to input and output pin voltages. In Figure 14 we are showing the points of convergence of the NAND4 gate output (the points where the output port current is zero).

The non-linear model that we have generated is not going to match the dc port current of the original gate because it models the transient behavior rather than the steady state one. In Fig-

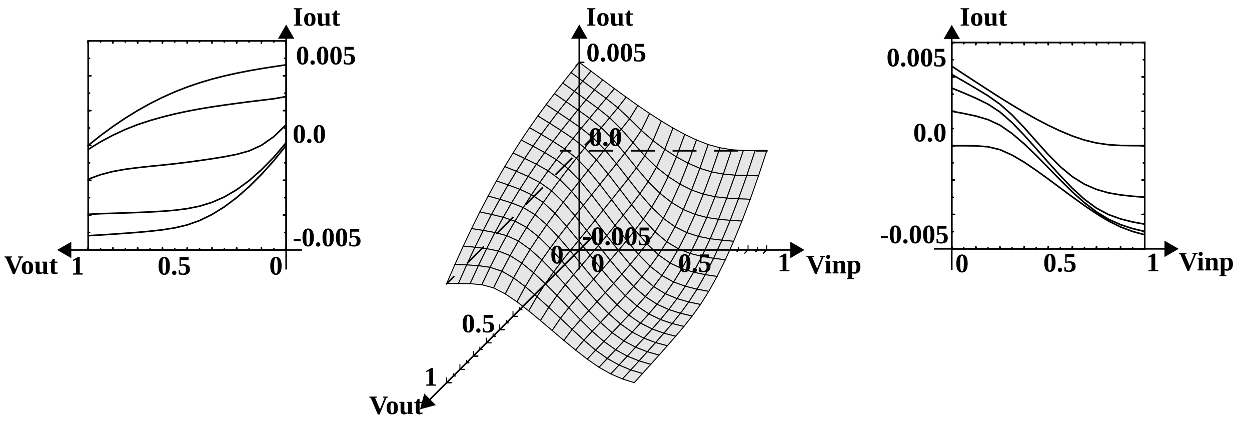


Figure 13**:** Variation of output port current w.r.t. input and output pin voltages (center). Contour plots of the output current for fixed input pin voltage levels (left) and fixed output pin voltage levels (right).

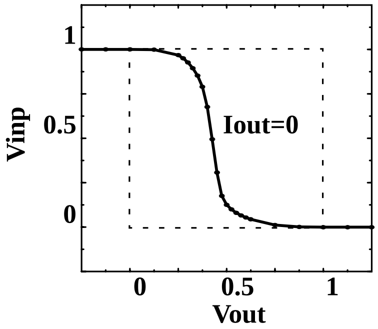


Figure 14**:** The convergence points of the original NAND4 gate which correspond to the points where the dc output gate current is 0.

ure 15 we are showing the port current of our model which has been obtained for a one element

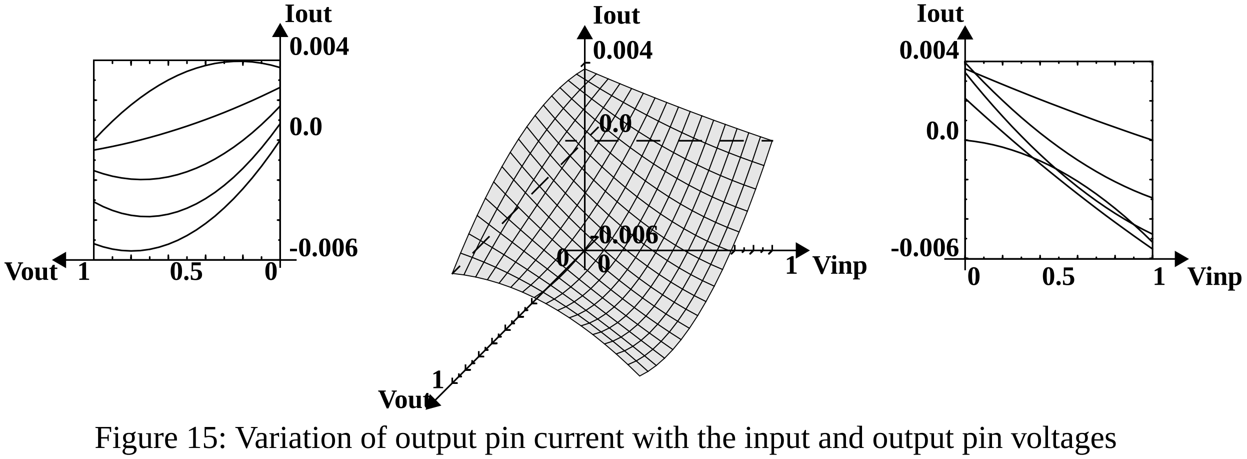


Figure 15: Variation of output pin current with the input and output pin voltages (center) for the non-linear driver model of the NAND4 gate. Contour plots of the output current for fixed input pin voltage levels (left) and fixed output pin voltage levels (right).

partition of the domain and using second order Lagrange interpolating polynomials as base functions. From the contour plots it can be seen that the port current is not monotonic inside the domain and that results in multiple operating points for the same input-output voltage pair.

In Figure 16 the convergence curve of the driver port model and stability region(s) are shown.

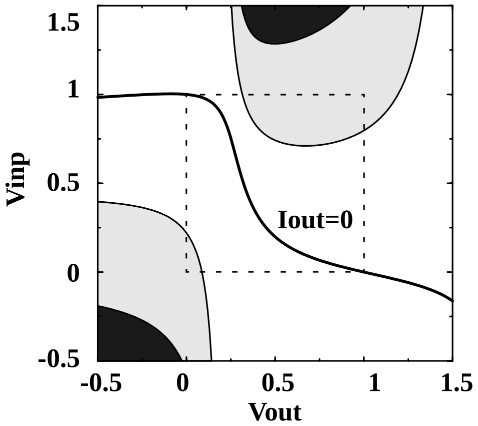


Figure 16: The convergence points of the driver model, the stable model domain (dark shaded area) and the absolute stability domain (light shaded area).

It is in general desirable to have a close match between the original convergence curve and the model because that impacts the steady state accuracy which is important in cases when multiple drivers are driving simultaneously the interconnect (see Example 5 from Section 4). From Figure 16 it is also apparent the impact of the holding resistance value. Our choice for R_d was the hold down resistance value (108ohms) and the model tries to compensate with current in the hold high case where the actual resistance is larger. However, in Example 2 of Section 4 showing the hold high and hold low functional noise pulses, we can see (Figure 19) that the accuracy in both cases is comparable.

Another important issue for our model is the domain of stability. For example, in our case the basis functions are second order Lagrange polynomials and for any input voltage value there are exactly two points where the port current is zero. One point is the convergence point and is characterized by:

$$I(u, w) = 0 \qquad \text{and} \qquad \frac{d}{dw}I(u, w) < 0\,, \tag{22}$$

and the other one is the limit of the stable region and is characterized by:

$$I(u, w) = 0 \qquad \text{and} \qquad \frac{d}{dw}I(u, w) > 0\,. \tag{23}$$

The stable region boundary is marked in Figure 16 by the border of the dark shaded areas. The light shaded areas are marking the boundary of the absolutely stable region. The points in this region are characterized by:

$$\frac{d}{dw}I(u, w) \le 0 \tag{24}$$

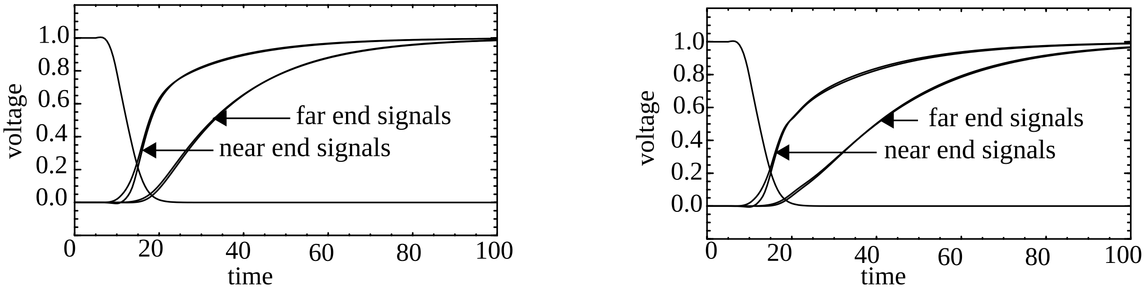
in which the port current source offers a negative feedback with respect to the output voltage variation. In general, the situation in which the model has regions of instability inside its domain is the result of sparse measurements data present in those regions.

4. Results

In this section we are presenting some results obtained with our proposed model. We are showing for comparison the performance of our model in the case of basic timing signal propagation, functional noise and delay noise. We are also exemplifying the robustness of our model in the case when input signals are outside the characterization range (over-shoot and under-shoot) and with highly inductive interconnect. We present the performance of the model in a multi-source net case and how the steady state is captured.

Using the test case described in Section 2, the new modeling technique has been used to characterize the timing arc of our NAND4 gate from the input pin A_1 to output pin X. We have used a Thevenin type model on a domain with 1 two-dimensional element (similar to the one shown in figure 9) characterized by 9 points (a 3x3 grid), two of them with known values, the hold high $V_d(0, 1) = 1$ and hold low $V_d(1, 0) = 0$ conditions. So, 7 points were unknowns in the characterization process. We have used 8 input-output signal pairs, 4 for rising output and 4 for falling output, with 2 equations for each pair (one for 0% to 50% and one for 10% to 90%). The R_d and C_d values have been determined using a simple small signal analysis on the output port with the gate set-up to hold low.

Example 1: The first example is the test used in Figure 3. The near and far end waveforms in the case “without noise” are shown in Figure 17.a. The near and far end waveforms in the case with delay noise are shown in Figure 17.b. The actual “delay noise” waveforms are shown in more detail in Figure 18 for the far end signals.



a) without noise **b)** with delay noise

Figure 17: The output pin of the NAND4 gate (near end) and the sink pin (far end).

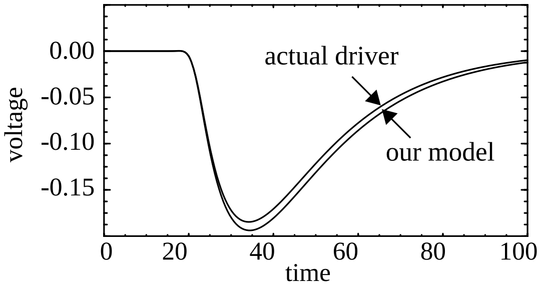


Figure 18**:** Delay noise comparison between our model and actual driver.

Example 2: In Figure 19 we are showing the accuracy of the model for functional noise estimation. Our model is compared with the actual gate and the hold-up/down resistors.

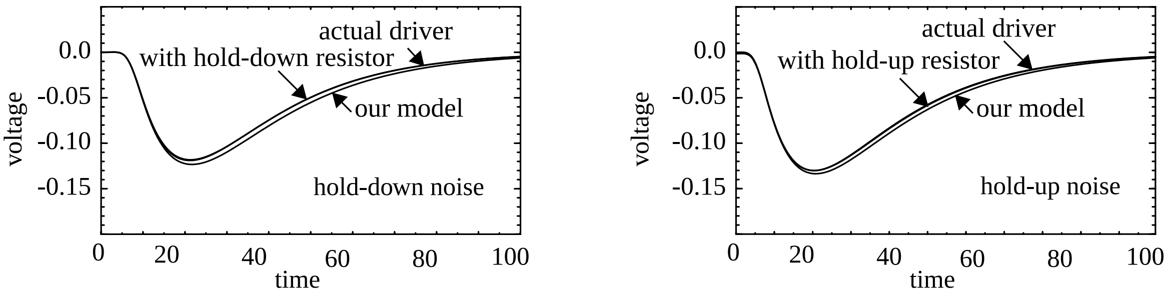


Figure 19: Functional noise at sink pin (far end) with the actual driver, holding resistance model and our model for holding down (left) and holding up (right).

Example 3: One of the more difficult cases to model in static timing (for the C-effective algorithm) is the gate response with inductive output loads. In order to show the robustness of the driver model, a large amount of inductance has been added to the interconnect RC model. The simulation results for the near and far end nodes of the interconnect wire are shown in Figure 20.

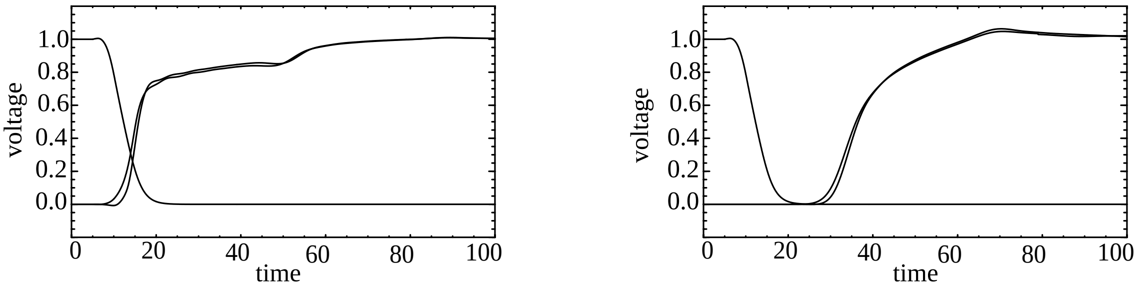
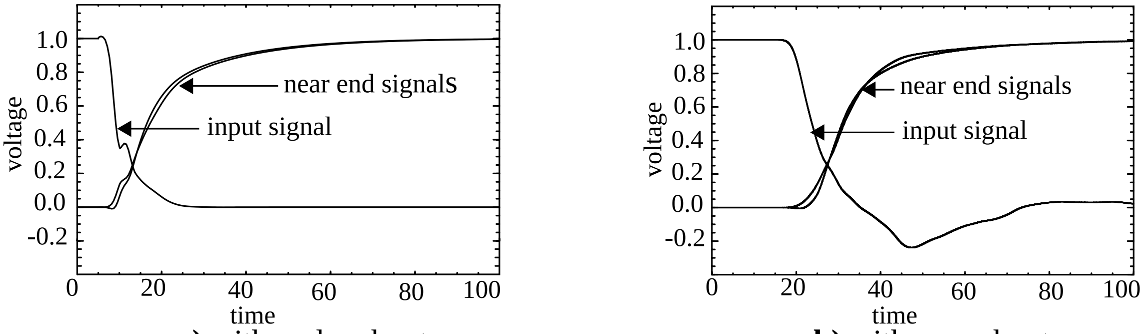


Figure 20: Signal propagation through highly inductive interconnect: near end signals (left) and far end signals (right).

Example 4: It is often the case that chains of gates are analyzed together with their interconnect. If the signals are heavily altered by noise or other effects, the simple ramp-like truncations performed on signals during static timing will result in significant errors. Since our models are characterized for a range of input signals, they are better suited for propagating a large class of input signals. In Figure 21 we used signals obtained at the beginning and end points of an RLC line to test the response of the model with signals that have less common shapes. Note that in Figure 21.b the input signal has an over-shoot going outside the range of input values for which the gate was characterized.



a) with under-shoot

b) with over-shoot

Figure 21: The response of the gate to less common input signals.

Example 5: In the recent years, one trend in the microprocessor clock design has been to generate grid-like clock distribution networks with multiple drivers to reduce the clock skew across the chip. Coupled with the higher impact of inductance on the long wide clock wires, the analysis of these nets in the static timing flow has been very difficult and inaccurate, forcing designers to perform extensive detailed circuit simulations. Our models capture very well the driver behavior on nets with multiple sources as shown in Figure 22 on a 5x5 grid network with 40 wire sections driven from the four corners. Two of the drivers have a significant input offset to magnify the voltage division across the interconnect. In Figure 22.b the waveforms at the first and last driver outputs and in the center of the grid are shown, both with real drivers and with our models. During the first part of the response, the two active drivers are driving each an amount of capacitance outside their model characterization range but good accuracy is maintained.

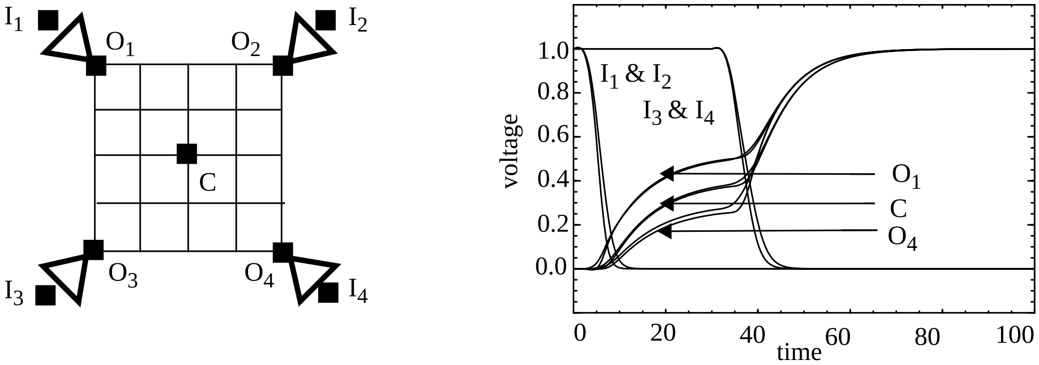


Figure 22: Signal propagation in a multi-source large RC grid (left). The driver inputs, outputs and the signal in the center of the grid are shown (right).

Example 6: We have used the algorithm proposed in [24] to generate the “transient holding” resistance for comparison with our model. Through full net simulation, a “transient holding” resistance has been determined (712.8ohms) such that the area of the noise pulse with resistance model matches within 0.004% the area of the real delay noise pulse. The superposition of the quiet response and the noise pulse with resistance model produces the approximation of the noise impact on delay. The errors are tabulated in Table 1:

Table 1: Comparison with “transient holding” resistance model

Pin	measurement type	our model	“transient” resistance
Near end	50% delay	-1.8%	-10.78%
	10%-90% rise time	8.10%	18.68%
Far end	50% delay	0.54%	-6.88%
	10%-90% rise time	3.99%	8.45%

In Figure 23, all the near end point “delay noise” pulses (using actual driver model, our non-linear model and “transient holding” resistance model) are shown. The plot spans 150 time units which is the interval used for matching the noise pulse areas.

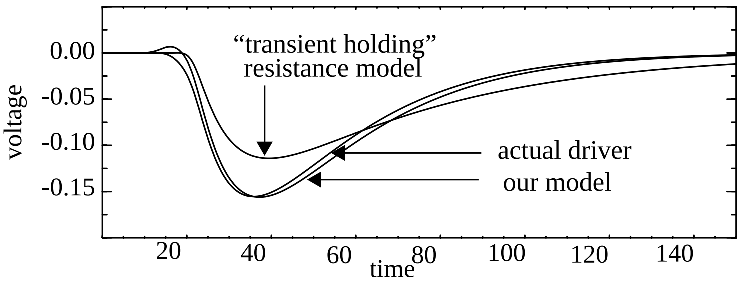


Figure 23: Comparison between the "delay noise" pulse obtained using our nonlinear driver model and the "transient holding" resistance model at the near end.

5. Conclusions and future work:

In this paper we have proposed a new technique to model logic gates for timing and noise analysis. The proposed models have quite a few distinct advantages over the existent driver models for timing and/or noise:

- The modeling process is using the already existent measurements data generated for static timing analysis for each logic block. No new data or special characterization work is needed.
- No non-linear spice simulations are required in the modeling process.
- The models are simple Thevenin/Norton-like circuits with voltage/current sources dependent on the input and output pin voltages and are represented using elementary functions (polynomials). This makes their simulation extremely efficient.
- The models have variable accuracy both in terms of the range of input rise time and output capacitance load that is being covered and in terms of the error with respect to the actual measurement values used in the process.
- The models are covering large ranges of input rise time and output capacitive load and they are re-usable (do not depend on a particular input-output situation or noise pulse).
- The models are very robust and maintain accuracy outside the characterization range.

The examples presented in Section 4 are showing the versatility of the proposed model. We have demonstrated the accuracy of our model in different situations of practical interest:

- normal signal propagation (static timing analysis) with very good behavior throughout the characterization domain,
- simulation of the driver response with complex output load models including inductance,
- computation of the delay variation due to switching noise,
- functional noise analysis,
- simulation of special cases such as nets with multiple drivers with significant time offsets and complex interconnect models.

As future work, our attention is focused on circuit simulation. One draw-back of our models is that in order to simulate them we need a non-linear circuit simulator. The driver models that we have used are piece-wise polynomial models. These models can be simulated very efficiently by available special purpose simulators (such as ACES [11]). One can also make a simple observation that the same FE method used to generate the models can be used to simulate them and, in conjunction with reduced order interconnect models, one can develop a very efficient simulation engine.

References:

[1][Ain00] - Aingaran K. *et al.* “Coupling noise analysis for VLSI and ULSI circuits”, IEEE First International Symposium on Quality Electronic Design 2000, page(s) 485-489.

[2][Aru97] - Arunachalam R., Dartu F. and Pileggi L.T. “CMOS gate delay models for general RLC loading” IEEE 1997 page(s) 224-229.

[3][Aru00] - Arunachalam R., Rajagopal K. and Pileggi L.T. “TACO: timing analysis with coupling” Design Automation Conference 2000, page(s) 266-269.

[4][Aru01] - Arunachalam R., Blanton R.D. and Pileggi L.T. “False coupling interactions in static timing analysis” Design Automation Conference 2001, page(s) 726-731.

[5][Bec81] - Becker E.B., Carey G.F. and Oden J.T. “Finite elements - an introduction -” Prentice-Hall Inc. 1981.

[6][Che97] - Chen W., Gupta S.K. and Breuer M. “Analytic models for crosstalk delay and pulse analysis under non-ideal inputs” International Test Conference 1997, page(s) 809-818.

[7][Che99] - Chen P. and Keutzer K. “Towards true crosstalk analysis” International Conference on CAD 1999, page(s) 132-137.

[8][Dar96] - Dartu F., Menezes N. and Pileggi L.T. “Performance computation for pre-characterized CMOS gates with RC loads” IEEE Transactions on CAD, vol. 15, issue 5, May 1996, page(s) 544-553.

[9][Dar97] - Dartu F. and Pileggi L.T. “Calculating worst-case gate delays due to dominant capacitance coupling” Design Automation Conference 1997, page(s) 46-51.

[10][Del00] - Delaurenti M. *et al.* “Switching noise analysis framework for high speed logic families” 14th International Conference on VLSI Design 2000, page(s) 524-530.

[11][Dev94] - Devgan A. and Rohrer R.A. “Adaptively controlled explicit simulation” IEEE Transactions on CAD, vol.13, no.6, Jun.1994, page(s) 746-762.

[12][Dev97] - Devgan A. “Efficient coupled noise estimation for on-chip interconnects” Digest of Technical Papers, International Conference on CAD 1997, page(s) 147-151.

[13][Fel97] - Feldman P. and Freund L.W. “Circuit noise evaluation by Pade approximation based model-reduction techniques” Digest of Technical Papers, International Conference on CAD 1997, page(s) 132-138.

[14][Fre98] - Freund R.W. “Reduced-order modeling techniques based on Krylov sub-spaces and their use in circuit simulation” Numerical analysis manuscript No. 98-3-02, Bell Laboratories, Feb. 1998.

[15][Gro98] - Gross P.D. *et.al.* “Determination of worst-case aggressor alignment for delay calculation” International Conference on CAD 1998, page(s) 212-219.

[16][Kas00] - Kashyap, C.V. and Krauter, B.L. “A realizable driving point model for on-chip interconnect with inductance” Design Automation Conference, 2000, page(s) 190-195.

[17][Kra99] - Krauter, B.L., Mehrotra S. and Chandramouli V. “Including inductive effects in interconnect timing analysis” IEEE Custom Integrated Circuits Conference 1999, page(s) 445-452.

[18][Kuh01] - Kuhlmann M. and Sapatnekar S.S. “Exact and efficient crosstalk estimation” IEEE Transactions on CAD, vol.20, no.7, July 2001, page(s) 858-866.

[19][Lev00] - Levy R. *et al.* “ClariNet: a noise analysis tool for deep sub-micron design” Design Automation Conference 2000, page(s) 233-238.

[20][Oda98] - Odabasioglu A., Celik M. and Pileggi L.T. “PRIMA: Passive reduced-order interconnect macro-modeling algorithm” IEEE Transactions on CAD, vol. 17, issue 8, Aug.1998, page(s) 645-654.

[21][Qia94] - Qian J., Pullela S. and Pillage L.T. “Modeling the ‘effective capacitance’ of RC interconnect” IEEE Transactions on CAD, vol. 13, issue 12, Dec. 1994, page(s) 1526-1535.

[22][Sas00] - Sasaki Y. and DeMicheli G. “Crosstalk delay analysis using relative window method” ASIC/SOC Conference 1999, page(s) 9-13.

[23][She99] - Shepard K.L., Narayanan V. and Rose R. “Harmony: static noise analysis of deep submicron digital integrated circuits” IEEE Transactions on CAD, vol.18, no.8, Aug.1999, page(s) 1132-1150.

[24][Sir01] - Sirichotiyakul S. *et al.* “Driver modeling and alignment for worst-case delay noise” Design Automation Conference 2001, page(s) 720-725.

[25][Xia00.1] - Xiao T., Chang C.-W. and Marek-Sadowska M. “Efficient static timing analysis in presence of crosstalk” ASIC/SOC Conference 2000, page(s) 335-339.

[26][Xia00.2] - Xiao T. and Marek-Sadowska M. “Worst delay estimation in crosstalk aware static timing analysis” International Conference on Computer Design 2000, page(s) 115-120.