

Modeling Delay and Noise in Arbitrarily Coupled RC Trees

Dinesh Pamunuwa, *Member, IEEE*, Shauki Elassaad, and Hannu Tenhunen

Abstract—Closed-form equations for second-order transfer functions of general arbitrarily coupled resistance–capacitance (RC) trees with multiple drivers are reported. The models allow precise delay and noise calculations for systems of coupled interconnects with guaranteed stability and represent the minimum complexity associated with this class of circuits. Their accuracy is extensively compared against other relevant models and is found to be better or comparable to more expensive models. All results are derived from a theoretical approach, and their physical basis is examined. The simplicity, accuracy, and generality of the models make them suitable for use in early signal integrity analyses of complex systems and incremental physical optimization.

Index Terms—Crosstalk, delay and noise modeling in VLSI circuits, interconnect modeling, timing analysis, transfer function.

I. INTRODUCTION

IMPROVEMENTS in lithography have seen continuously shrinking line widths and increasing die sizes, resulting in unprecedented opportunities to implement entire systems on a single chip. This trend of decreasing feature sizes can be expected to continue for the next 8–10 years [1]. With decreasing gate delays and increasing wiring density, noise modeling and its impact on performance and functionality has become very important. The majority of signal wires are typically lossy, and higher aspect ratios to control the resistance result in increased capacitive coupling. This, together with smaller signal rise times, results in heavy crosstalk, which couples a noise voltage onto the victim net. A distinction is usually made between the coupled noise amplitude and the effect of noise on delay. The former can cause functional failures by causing the voltage to swing above or below the logic threshold, while the latter has an impact on the cycle time.

The ability to put billions of transistors on a single die has also imposed severe restrictions on the computational complexity of noise and delay models used in an iterative design flow. While more accurate modeling is necessary, the sheer

Manuscript received December 6, 2003; revised May 21, 2004 and October 14, 2004. This work was funded partly by Cadence Design Systems, and partly by the Swedish governmental funding agencies Sida, Vinnova, and Exsite. The bulk of this work was carried out when D. Pamunuwa was a graduate intern at Cadence Berkeley Laboratories, Cadence Design Systems, Berkeley, CA 94704 USA. This paper was recommended by Associate Editor D. Blaauw.

D. Pamunuwa is with the Centre for Microsystems Engineering, Department of Engineering, Faculty of Applied Sciences, University of Lancaster, Lancaster LA14YW, U.K.

S. Elassaad is with the Rio Design Automation, Santa Clara, CA 95054-1137 USA.

H. Tenhunen is with the Laboratory of Electronics and Computer Systems, Department of Microelectronics and Information Technology, Royal Institute of Technology (KTH), Stockholm, SE 16440, Sweden.

Digital Object Identifier 10.1109/TCAD.2005.852279

size of the systems prohibits expensive dynamic simulation. Consequently, the subject of delay and noise modeling for very large scale integration (VLSI) circuits has received a vast amount of attention in the literature. The three attributes of accuracy, computational simplicity, and generality are, however, difficult to encompass in a single integrated model. Most reported models that consider the effect of crosstalk on noise and delay either use heuristics that are tailored for specific topologies or use multiple moments that make them expensive.

In particular, there is a huge need for an efficient model that can be used in the inner loop of place-and-route algorithms for global signal planning and incremental physical optimization in circuits with heavy coupling. In addition to minimal computational complexity, some other specific characteristics are necessary in such a model; it should allow a one-to-one correspondence between the physical structure being modeled, and the electrical circuit being analyzed, so that changes can be quickly and efficiently incorporated, and it should also exhibit fidelity. In incremental physical optimization, the absolute accuracy of the delay at any given node predicted by the metric is of secondary importance to the generation of a solution that is the same, or as close as possible, to the solution that would be generated by using a much more accurate delay metric. The Elmore delay [10] possesses this attribute of fidelity, which, coupled with its peerless efficiency, is why it has been and still is very popular in place-and-route algorithms. However, in the face of heavy coupling, its usefulness diminishes, fundamentally because it can only address grounded capacitors.

The contribution of this paper is as follows. Closed-form models for generating second-order transfer functions from each driver to the receiver in arbitrarily coupled resistance–capacitance (RC) trees (such as that shown in Fig. 1) representing the minimum complexity for this class of circuits were recently reported [2], [3]. The summation of all waveforms results in the complete response to all switching events at the node of interest, with no restriction on arrival times, and allows both delay and noise estimations. These models are intended for global signal planning, and incremental physical optimization in CAD algorithms. In this paper, the model and constituent expressions are derived from first principles, and formal proofs provided; also, statistical data of extensive comparisons against other relevant models are included. It is also clearly shown how this model saves at least two full tree traversals over the second-order model that bears the closest resemblance to it, while providing comparable accuracy.

This manuscript is organized as follows. The next section presents a detailed overview of the current work that puts the contribution of the reported model in context. Section III

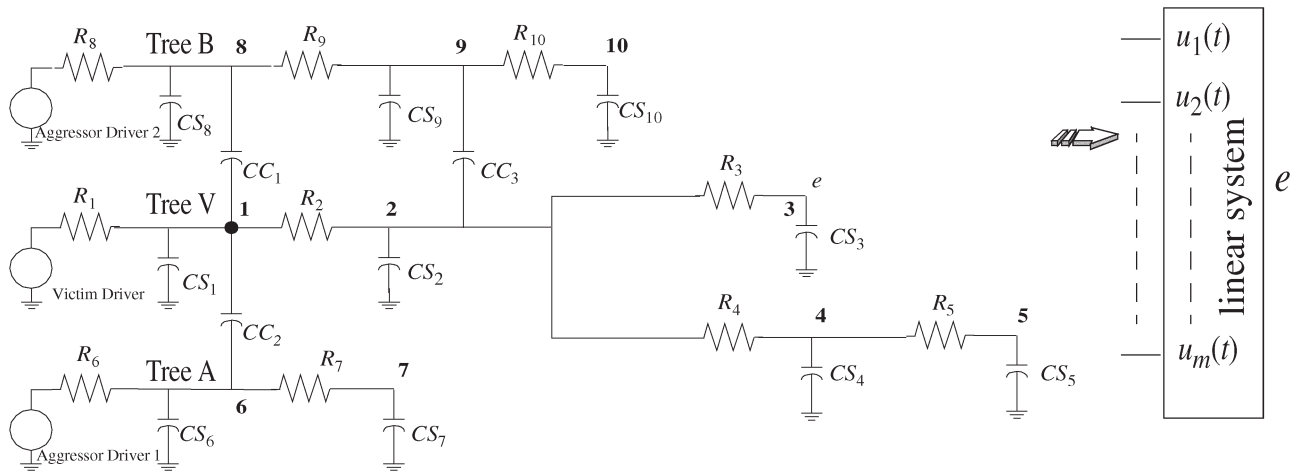


Fig. 1. Example of coupled RC tree.

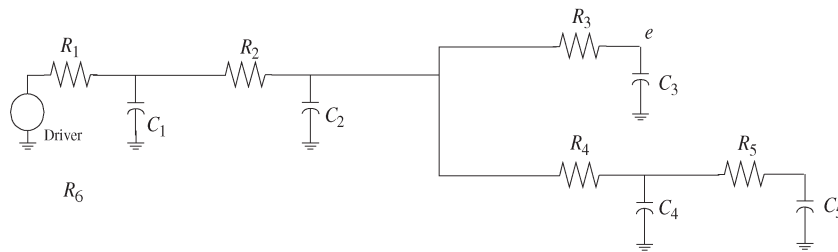


Fig. 2. Example of simple RC tree.

explains the proposed modeling in detail, which is based on the moments of the circuit, and follows a strict theoretical approach. The proposal for guaranteeing stability and its physical ramifications, and the computational complexity of the models are discussed. Next, the results of running the proposed models on numerous test circuits are presented. The paper ends with a discussion.

II. BACKGROUND

An accurate analysis of signal propagation in interconnects requires solving Maxwell's equations in three dimensions, which is prohibitively expensive in terms of computation time. However, it is possible to use simplified models in most cases to capture the important effects in the regime of interest [5], [6]. A particular concern with falling rise times is how important inductive effects are, and when and how they should be modeled. A growing body of literature that addresses this issue now exists [7]–[9]. They propose metrics that relate the physical dimensions of the wire to the signal rise time by assuming transverse-electromagnetic (TEM) propagation to determine when neglecting inductive effects results in significant errors. The general consensus now is that modeling inductance is necessary for special nets such as clock and power lines, and that the majority of signal lines can be accurately modeled by networks of resistors and capacitors, even with very small rise times. It is important, however, to consider the effect of capacitive crosstalk, which is exacerbated by sharper slew rates. In this paper, interconnects that can be modeled by coupled RC lines are considered.

A. Delay Modeling

Timing analysis in VLSI circuits has long been carried out using the simplified model of an RC tree where all capacitors are connected to ground, whose circuit model is called a simple tree (Fig. 2). There is a large body of literature that deals with delay modeling in simple trees. One of the most important and widely used metrics, the first moment of the impulse response, was proposed back in 1948 as an upper bound for the delay in valve circuits [10] and is known as the Elmore delay. Its attraction is that it uses minimum information and has unmatched algorithmic simplicity and elegance, explicitly matches the circuit elements to the delay, and yet exhibits good fidelity, giving results as good as more expensive models when used in interconnect optimization algorithms. However, as mentioned, it becomes less accurate as heavy coupling comes into play, and noise waveforms cannot be modeled, as a minimum of two time constants are required to model a voltage spike. Bounds and metrics that give an indication of when the Elmore delay is a poor approximation were developed in [11].

A stable approximation to the second-order transfer function for simple trees based on the first and second moment of the impulse response, and the sum of the open-circuit time constants, was proposed in [4] and expanded in [12] to encompass charge sharing networks. Later, generic moment-based techniques applicable to any circuit comprising linear elements that allowed the calculation of an arbitrary number of poles were developed in [13]. An implementation that is optimized for the tree-like structures of interconnects was proposed in [14]. These techniques depend on the Padé approximation,

which typically requires $2q$ moments for a q th-order approximation, and though very efficient, are too expensive for incremental physical optimization. Other estimators based on the Arnoldi algorithm [15] match a lesser number of moments to a q th-order approximation. An example is [16], which gives reduced order models for linear systems. However, at least one LU decomposition of the admittance matrix (which has a cubic complexity) is necessary, in addition to qN backward and forward substitutions for q block moments (for a q th-order approximation) of an N -port linear circuit. For a second-order approximation, the complexity is much more than the model proposed here, as shall be shown in Section III-D.

Hence, numerous models that occupy some position in the spectrum defined by the accurate though expensive solution offered by generic moment matching techniques such as AWE [13] (and similar methods) at one end, and the simplicity offered by the Elmore delay at the other, have been proposed. For simple trees, the models of [4] represent the minimum computational complexity for a second-order model. Alternate second-order models for the transfer function include those reported in [17] and [18], which involve generating equivalent circuits and are more suited for highly inductive lines, that reported in [19], which yields a stable model from the first three moments, and also the one reported in [20]. The latter two models will be revisited as they are relevant in discussing the performance of the proposed model.

Now, a two (or higher)-pole model cannot be solved explicitly for the delay at a given threshold. Hence, there are quite a few works that attempt to garner more information than the first moment (Elmore delay) from the circuit, and match it explicitly to the delay via some heuristic, such as in [17]–[20]. Alpert *et al.* [21] also present two heuristic delay metrics, one based on the first two moments, and another based on an effective capacitance model, which seeks to overcome the effect of resistive shielding that makes the Elmore delay inaccurate at near-end nodes. Explicit delay models for inductive lines were proposed in [22]. Different approaches were suggested in [23] and [24], where the moments of the circuit are matched to parameters of probability density functions to yield the delay, but are less accurate than moment matching of a similar order. The issue of explicit expressions for calculating the delay from a two-pole waveform is not addressed here, as the activity that dominates runtime for any reasonably sized circuit is the generation of the moments. Processing a two-pole waveform for the input of choice (whether step, ramp, or exponential) is independent of the circuit, and the complexity is hence trivial in comparison to traversing the circuit and generating the moments.

In today's circuits, as mentioned, considering the effect of noise is important. Finding the response of such systems involves solving circuits with multiple drivers and coupling capacitors, consisting of simple trees coupled to each other via series capacitors, whose circuit model is called a coupled tree (Fig. 1). General moment-matching techniques can of course be applied to solve coupled trees, but, again, simplified techniques are necessary for use early in the design flow. Timing analyzers often use the concept of worst, average, and best case delay, using a switch factor that takes the value of 2, 1, or 0 to modify

the Elmore delay. The capacitance for a line is modeled as the sum of two components, one of which represents the capacitance to ground, and the other the capacitance to adjacent nets. This second component is multiplied by a factor depending on whether the coupled net is expected to be quiet or not, and if not, on the direction of switching. This method of modeling is not accurate except in certain very simple situations, such as uniform structures or simultaneously switching nets, and indeed it was recently shown to not even represent an upper bound on the delay [25]. A lot of research has focused on certain simplified configurations of interest. In [26], the authors use the first moment of the impulse response to generate single-pole responses for uniformly coupled RC lines, while [27] presents a two-pole response for a single section of a coupled π circuit with arbitrary ramp inputs. They extend it to accommodate multiple segmented aggressors in [28], but the allowed topology is still very limited.

B. Noise Modeling

Now, as mentioned, it is often necessary to know the coupled noise amplitude explicitly in order to check for spurious errors caused by switching nets disturbing the logic state of a quiescent net. A single-pole noise metric for coupled trees was proposed in [29]. Although computationally efficient, some simplifying assumptions in the formulation of the metric cause the results to be mostly very pessimistic. Some of the works mentioned above, which present models for estimating the effect of noise on delay, also report noise metrics [20], [26], [28]. In [30], the authors use circuit transformations to simplify a general tree to a two- π model when analytic formulas can be used, but intermediate steps require the calculation of admittances at each branch point and the estimation of equivalent capacitances, which increase runtime and impact on the accuracy, respectively.

When dealing with multiple-driver systems such as the one depicted in Fig. 1, the concept of superposition is very useful, as the coupled RC network is still a linear system. The effect of multiple aggressors switching at different times can be estimated by considering one input at a time with all other inputs grounded, and then adding up the individual waveforms as is done in all moment-based methodologies. Tong and Marek-Sadowska [31] also adopt superposition, where transfer functions are generated from each driver to the receiver. However, the only concession to different switching events (and hence different charging paths) is calculating a unique zero; the poles of the transfer function for all switching events are the same, and they are the two lowest frequency poles of the system. These poles are estimated from the methodology proposed in [32], which gives closed-form expressions for the poles of systems with storage elements, and is a technique that has long been used in analog design to estimate the bandwidth of amplifiers. However, using the same two lowest frequency poles in all of the transfer functions can result in large errors, as the significant poles, which determine the response for different switching events, can be far apart on the frequency axis. The reason is that though these poles are part of the natural response of the system, and hence do appear in the transfer function

from each driver to the receiver, there will always be partial pole-zero cancellation in systems that have signal paths with widely differing time constants. Since the transfer function is limited to two poles, it is important that for each path, the two-pole-one-zero model that best fits that particular charging path is calculated. If the same poles are used, the results will be skewed by the highest parasitics in the coupled tree, regardless of their influence on the particular switching event. For this same reason, [20], which uses the driving point moment to calculate the poles and uses the same poles for all the responses, is less accurate than [19].

The model that best merits comparison with the proposed model for both delay and noise estimation is [19], being able to handle coupled RC trees and being the closest in computational complexity when the moments are generated by the algorithms of [14], though still more expensive by virtue of the fact that it uses one extra moment. An accuracy comparison against this model (and others) are carried out in Section IV, while the complexity is discussed in detail in Section III-D.

III. MODELING THE TRANSFER FUNCTION

Consider Fig. 1, which shows an arbitrary network comprising a victim net and several aggressors coupled to the victim net through banks of series capacitances. Such a network can be represented by an m -input-one-output system as shown. $E_e(t)$ is defined as the voltage waveform at the node of interest e on the victim. The total waveform at e can always be represented by the n th-order linear differential (1), where n is the order of the system and u_i the inputs

$$a_n \frac{d^n}{dt^n} E_e(t) + a_{n-1} \frac{d^{n-1}}{dt^{n-1}} E_e(t) + \cdots + a_1 \frac{d}{dt} E_e(t) + E_e(t) = b_1 u_1(t) + b_2 u_2(t) + \cdots + b_m u_m(t). \quad (1)$$

Setting the right-hand side to zero results in the homogeneous equation, the solution to which gives the natural or transient response of the circuit. For a second-order approximation, the complementary equation becomes

$$\lambda_2 s^2 + \lambda_1 s + 1 = 0.$$

Assuming that the roots (which are always real and negative for an RC tree) are $s = -1/\tau_1$ and $s = -1/\tau_2$, the complete response of the circuit is given by the following two time constant model, where $f(t)$ is the particular solution corresponding to the forcing functions

$$E_e(t) = A e^{-\frac{t}{\tau_1}} + B e^{-\frac{t}{\tau_2}} + f(t).$$

For the purpose of analysis, it is possible to assume that $f(t)$ or the steady-state value is always zero or one (for normalized supply rails). The coefficients A and B will depend on the inputs. In the proposed methodology, linear superposition is used where the response for each input is considered with all other inputs grounded, and all those responses are summed up to generate the complete solution (as in all moment-based approaches).

A coupled RC tree is characterized by a resistive path from the output node e to the forcing (victim—denoted by v) driver, and series capacitive elements to other (aggressor—denoted by a) drivers. Hence, the output for the victim driver switching will always change rails, while it will start and end at the same rail for an aggressor switching. Consequently, the transfer function characterizing the response in the former case will have a zero on the negative part of the real axis

$$H^v(s) = \frac{1 + s\tau_z^v}{(1 + s\tau_1^v)(1 + s\tau_2^v)} \quad (2)$$

while that for the latter will have a zero at the origin

$$H^{a_i}(s) = \frac{s\tau_z^{a_i}}{(1 + s\tau_1^{a_i})(1 + s\tau_2^{a_i})}. \quad (3)$$

A. Calculation of Moments

Fig. 1 can be referred to in the following descriptions. First, the notation used is described in some detail below.

CS_k^p	Capacitance to ground at node k on p th tree.
CC_{kj}^{pq}	Capacitance between node k on p th tree and node j on q th tree where first sub(super)script refers to reference tree.
E_k^p	Voltage at k th node on p th tree.
E_{kj}^{pq}	Voltage between node k on p th tree and node j on q th tree.
E_k^{pq}	Voltage between node k on p th tree and corresponding coupled node on q th tree; i.e., second subscript is omitted as it is a more convenient notation when permissible.
R_{ko}^p	Shared resistance from source to nodes o and k on p th tree.
Υ_k^n	n th moment of the impulse response at the k th node.

It should be noted that superscripts always refer to trees, while subscripts always refer to nodes, except in the definition for moments, where the superscript refers to the order of the moment, and in the time constants, where the subscripts refer to an identifier. This distinction is always obvious in the context. Additionally, rail voltages are normalized to zero and one, and the expressions are always derived for a positive step without loss of generality. For negative transitions, the waveforms are simply mirrored. The usage of the notation is illustrated in Fig. 3. Where it is possible to do so without introducing ambiguity, the second subscript will be dropped for convenience. For example, if tree 1 is the reference tree in Fig. 3, CdE_k^{12}/dt refers to $i(t)$, and node j is implicit in the expression.

Finally, the following quantity is defined

$$\tau_{D_e}^{t_r t_i} = \sum_{k \in t_r} R_{ke}^{t_r} C_k^{t_r t_i}. \quad (4)$$

This is the summation over the reference tree t_r , of resistance-capacitance products at each node k , where R_{ke} is the shared resistance between node k and sink e , on the path from source

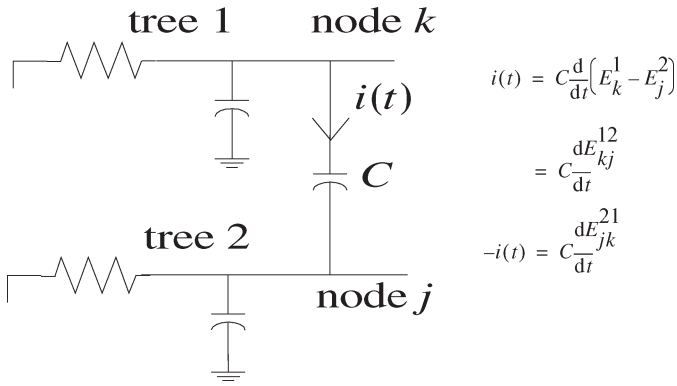


Fig. 3. Illustration of usage of notation.

to sink. The capacitance term $C_k^{t_r t_i}$ is the capacitance between trees t_r and t_i at node k on t_r . For example with reference to Fig. 1, $C_1^{v b}$ is CC_1 . If the second tree t_i is omitted, the capacitance refers to the total capacitance at node k ; for example, C_1^v is $(CS_1 + CC_1 + CC_2)$. In that case, the second tree would also be omitted in the name, i.e., $C_k^{t_r}$ would be with respect to $\tau_{D_e}^{t_r}$. This notation is used because it makes for a compact description, and also to make it consistent with that adopted in the famous Penfield–Rubinstein metrics [11], and in [4], which describes second-order models for simple trees. The lowercase subscript in τ_{D_e} , which is e in this case, always refers to the output. If the output node is omitted, the only quantity, which is with respect to the output R_{ke} , becomes R_{kk} .

The derivation of the following expressions is given in Appendix I. The first moment of the impulse response at the output node e for the victim driver switching, while all other inputs are grounded, is given by the following expression, where a_1, a_2, \dots are the aggressors

$$\Upsilon_{e,v}^1 = \sum_{k \in v} R_{ke}^v [CS_k^v + CC_k^{va_1} + CC_k^{va_2} + \dots] = \tau_{D_e}^v \text{ say.} \quad (5)$$

The second moment is given by

$$\begin{aligned} \Upsilon_{e,v}^2 &= 2 \sum_{k \in v} R_{ke}^v \left\{ CS_k^v \tau_{D_k}^v + CC_k^{va_1} \left[\tau_{D_k}^v + \sum_{K \in a_1} R_{Kj}^{a_1} CC_K^{a_1 v} \right] \right. \\ &\quad \left. + CC_k^{va_2} \left[\tau_{D_k}^v + \sum_{K \in a_2} R_{Kj}^{a_2} CC_K^{a_2 v} \right] + \dots \right\} \\ &= 2 (\tau_{G_e}^v)^2 \text{ say.} \end{aligned} \quad (6)$$

The first moment of the impulse response at node e on the victim tree for aggressor a_i switching can be shown to be

$$\Upsilon_{e,a_i}^1 = - \sum_{k \in v} R_{ke}^v CC_k^{va_i} = -\tau_{D_e}^{a_i} \text{ say.} \quad (7)$$

The second moment is

$$\begin{aligned} \Upsilon_{e,a_i}^2 &= -2 \sum_{k \in v} R_{ke}^v \\ &\quad \times \left\{ (CS_k^v + CC_k^{va_1} + CC_k^{va_2} + \dots) \tau_{D_k}^{a_i} + CC_k^{va_i} \right. \\ &\quad \left. \times \left[\sum_{K \in a_i} R_{Kj}^{a_i} (CS_K^{a_i} + CC_K^{a_i v} + CC_K^{a_i b_1} + \dots) \right] \right\} \\ &= -2 (\tau_{G_e}^{a_i})^2 \text{ say.} \end{aligned} \quad (8)$$

The expressions in (5)–(8) form the basis of the proposed models, along with the sum of the open-circuit time constants, introduced later in (21).

B. Matching Moments to the Characteristic Time Constants in the Circuit

Now, generating the best two-pole–one-zero transfer function for the response at the output node for any given switching event is of interest here. The moments can be matched to the characteristic time constants in the circuit by considering the power series expansion of e^x in the definition of the Laplace transform. The Laplace transform of the impulse response is

$$\begin{aligned} H(s) &= \int_0^\infty h(t) e^{-st} dt \\ &= \int_0^\infty h(t) \left[1 - st + \frac{s^2}{2!} t^2 - \dots \right] dt \\ &= \int_0^\infty h(t) dt - s \int_0^\infty t h(t) dt + \frac{s^2}{2} \int_0^\infty t^2 h(t) dt - \dots \end{aligned}$$

From this equality, the following identity can be observed

$$\Upsilon^n = (-1)^n \frac{d^n}{ds^n} H(s) \Big|_{s=0}. \quad (9)$$

This equation can be used to match the moments to the poles and zeroes of the circuit directly. Using (2), (5), (6), and (9), it can be seen that

$$\tau_1^v + \tau_2^v - \tau_z^v = \tau_{D_e}^v \quad (10)$$

$$(\tau_1^v + \tau_2^v - \tau_z^v) (\tau_1^v + \tau_2^v) - \tau_1^v \tau_2^v = (\tau_{G_e}^v)^2. \quad (11)$$

Now, additional information is necessary to solve for the three unknowns in (10) and (11). If a third equation is assumed for the reciprocal pole sum

$$\tau_1^v + \tau_2^v = \tau_{\text{sum}}. \quad (12)$$

These three equations can be combined to form the following quadratic, which yields two time constants

$$\tau^2 - \tau_{\text{sum}}\tau + \tau_{D_e}^v \tau_{\text{sum}} - (\tau_{G_e}^v)^2 = 0. \quad (13)$$

Other than τ_{sum} , the other metrics in (13), the first and second moment, are with reference to the victim. At this point, it is helpful to look at the physical interpretation of the first and second moments of the impulse response. The first moment always considers resistances of the switching line, and either all capacitances connected to the switching line (in the case of the victim driver switching) or capacitances connecting it to a particular line (for the switching of an aggressor driver). The second moment propagates outwards another level and considers the resistances and capacitances of immediately adjacent lines as well. This intuition is valuable in generating a solution with minimum computational complexity; namely, (13) can be used to generate the pole time constants for all switching events by using the appropriate reciprocal pole sum.

Now, since (13) can in general yield complex poles or a positive pole, some care is necessary to ensure stability. Potential instability can take one of two forms: If the sign under the radical in the solution for the roots of (13) is negative, complex poles can result; if the magnitude of the square root is greater than the reciprocal pole sum, a negative time constant results. Using these as limiting conditions, a methodology that always yields stable and accurate results can be formulated. The time constants are

$$\tau_{1,2} = \tau_{\text{sum}} \pm \sqrt{\tau_{\text{sum}}^2 - 4 \left[\tau_{\text{sum}} \tau_{D_e}^v - (\tau_{G_e}^v)^2 \right]}. \quad (14)$$

One limiting condition is that the sign under the radical should be positive. This leads to

$$\tau_{\text{sum}}^2 + 4 \left[(\tau_{G_e}^v)^2 - \tau_{\text{sum}} \tau_{D_e}^v \right] > 0. \quad (15)$$

This inequality is satisfied if

$$(\tau_{G_e}^v)^2 > \tau_{\text{sum}} \tau_{D_e}^v. \quad (16)$$

However, this would violate the second condition, which is that the magnitude of the square root should be greater than the reciprocal pole sum

$$\tau_{\text{sum}} > \sqrt{\tau_{\text{sum}}^2 - 4 \left[\tau_{\text{sum}} \tau_{D_e}^v - (\tau_{G_e}^v)^2 \right]}. \quad (17)$$

If (16) is true, (17) will never be true. Hence, the stability condition has to be more stringent. It can be guaranteed that (17) is true if the following holds

$$(\tau_{G_e}^v)^2 < \tau_{\text{sum}} \tau_{D_e}^v$$

or

$$\tau_{\text{sum}} > \frac{(\tau_{G_e}^v)^2}{\tau_{D_e}^v}. \quad (A)$$

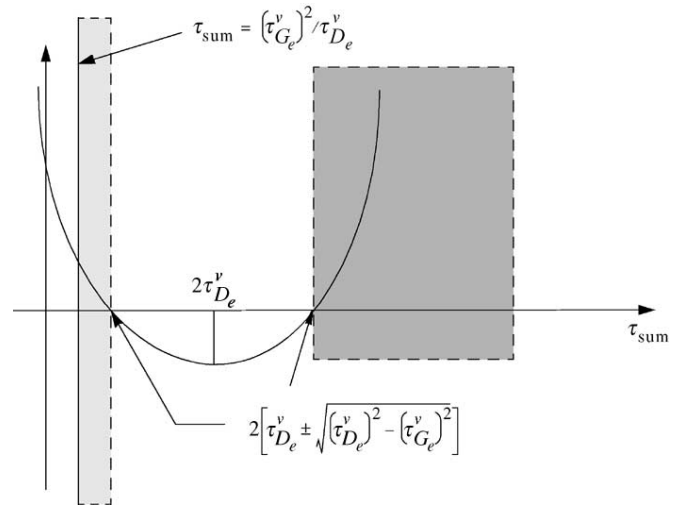


Fig. 4. Variation of quadratic which determines stability with τ_{sum} .

That is to say, the reciprocal pole sum must be large enough. However, when (A) is fulfilled, the second term in (15) is negative. Rewriting the left-hand side of it gives

$$\text{LHS} = \tau_{\text{sum}}^2 - 4\tau_{D_e}^v \tau_{\text{sum}} + 4(\tau_{G_e}^v)^2. \quad (18)$$

The function designated LHS is called the stability function, and is a quadratic in τ_{sum} . By considering the first and second derivatives, this parabola can be shown to have a minimum at $2\tau_{D_e}^v$. The zero-crossing points are given by

$$\tau_{\text{sum}} = 2 \left[\tau_{D_e}^v \pm \sqrt{(\tau_{D_e}^v)^2 - (\tau_{G_e}^v)^2} \right]. \quad (19)$$

Obviously, both of these points are on the right-hand side of the vertical axis, hence the shape of the parabola (Fig. 4). Now, if the sign under the radical in (19) is negative, its roots are complex, or in other words LHS will never become negative and (15) is always true. Hence, for potential instability to occur, the following must always be true

$$(\tau_{D_e}^v)^2 > (\tau_{G_e}^v)^2. \quad (20)$$

As proven in Appendix II, the line corresponding to the equality of (A) should appear to the left of the first zero-crossing as shown. Then for stability, τ_{sum} has to appear in the lightly hatched area, or to the right of the second zero-crossing point. If τ_{sum} is too small, the sign under the radical is positive, but a negative time constant results. If τ_{sum} is situated between the zero-crossing points, complex poles are derived. Finally, if τ_{sum} is to the right of the second zero-crossing point, represented by the darkly hatched area, again, a stable solution results. Hence, from the zero-crossing points, the next condition is derived

$$\tau_{\text{sum}} < 2 \left[\tau_{D_e}^v - \sqrt{(\tau_{D_e}^v)^2 - (\tau_{G_e}^v)^2} \right]$$

or

$$\tau_{\text{sum}} > 2 \left[\tau_{D_e}^v + \sqrt{(\tau_{D_e}^v)^2 - (\tau_{G_e}^v)^2} \right]. \quad (B)$$

Now, the stability conditions have been identified, the values for τ_{sum} that give the best response for different switching events can be derived. Firstly, for the case of the victim driver switching, since all aggressors are grounded, the metric that gives the best solution is the sum of the open circuit time constants with reference to the victim driver, which is called τ_p^* . This is simply the summation of the products of all capacitors connected to the victim line with the driving point resistance to each of those capacitors

$$\tau_p^* = - \sum_{k \in v} [R_{kk}^v C S_k^v + (R_{kk}^v + R_{jj}^{a1}) C C_k^{va1} + (R_{kk}^v + R_{jj}^{a2}) C C_k^{va2} + \dots] \quad (21)$$

This is a good approximation for the sum of the pole time constants [32], giving

$$\tau_1^v + \tau_2^v = \tau_p^* \quad (22)$$

Substituting (22) for τ_{sum} in (10) and (13) results in the zero time constant and pole time constants respectively, for the victim switching. Now, an inspection of (5) and (21) shows that $\tau_p^* > \tau_{D_e}^v$. Since (20) has to be true for instability to occur, this means that

$$\tau_p^* > \frac{(\tau_{G_e}^v)^2}{\tau_{D_e}^v} \quad (23)$$

Therefore, (A) is always true, and the only possible stability violation in this case is (B); i.e., very occasionally, using τ_p^* can result in complex poles. The physical interpretation of such an occurrence is that the sum of the open circuit time constants underestimates the reciprocal pole sum, which has been unusually escalated by an aggressor or aggressors with exceptionally high parasitics. Because both exponential waveforms are either additive or subtractive unlike when an aggressor switches (where one is additive and the other is subtractive), the higher frequency pole does not have a significant impact. In fact, this form of instability is usually an indication of a very-low-frequency pole, which makes the prediction of the waveform straightforward. The simplest remedy therefore is to consider a single pole response, with the pole time constant being given by $\tau_{D_e}^v$. This results in good accuracy as shall be shown in the Section IV.

Secondly, to solve for the poles and zeros associated with an aggressor switching, (3), (7), (8), and (9) are combined to give

$$\tau_{D_e}^{a_i} = \tau_z^{a_i} \quad (24)$$

$$(\tau_{G_e}^{a_i})^2 = \tau_z^{a_i} (\tau_1^{a_i} + \tau_2^{a_i}) \quad (25)$$

Now, the zero time constant is available immediately from (24), and dividing (25) by (24) results in the reciprocal pole sum

$$\frac{(\tau_{G_e}^{a_i})^2}{\tau_{D_e}^{a_i}} = \tau_1^{a_i} + \tau_2^{a_i} \quad (26)$$

The pole time constants can be obtained by substituting (26) as τ_{sum} in (13). It can be seen from an inspection of the relevant expressions that potentially either of (A) or (B) can be violated. The solution without generating extra information about the circuit is to accept the next best approximation. That is to say, if τ_{sum} is so small that it violates inequality (A), the simplest and most logical remedy is to increase τ_{sum} so that it is in the lightly hatched area. When inequality (B) is violated, if τ_{sum} is less than the minima, it should be decreased so that it falls into the lightly hatched region; if it is greater than the minima, it should be increased so that it falls into the darkly hatched region. Since the equality will generate coincident poles that are not acceptable, the exact value should be chosen so that it is slightly greater than or less than the equality, which can be achieved with a percentage factor. From empirical evidence over a range of testbeds, it was seen that 1% provides the best accuracy. This is the only constant that cannot be predicted by the theory, and is unfortunately unavoidable, as any approach to preserve stability without resorting to higher order moments has to be approximate in nature.

Using this approach, the values that τ_{sum} should take in the different cases are summarized in Table I. Of the two, (A) being violated is by far the more common form of instability. This occurs when the dominant poles for the victim and the particular aggressor are very far apart on the frequency axis. Physically, this translates to a situation where the receiver node is charged extremely rapidly by a very strong aggressor (i.e., through a relatively very small time constant), and decays with a very long tail, dictated by the much larger time constant of the victim. Such behavior is common for far end coupling. The instability in the solution predicted by (13) occurs because the pole sum given by (26) accurately reflects the high-frequency nature of the poles in the aggressor's charging path, but $\tau_{D_e}^v$ and $(\tau_{G_e}^v)^2$ reflect the much lower frequency content of the victim's dominant poles, and the gap is too much to bridge. The remedy proposed to this situation is to increase the reciprocal pole sum just beyond the threshold of the equality. Now, this yields accurate results, because the intention is to generate the best two-pole-one-zero model; in other words, the poles and zeros need not equate to the actual poles and zeros of the system, and indeed should differ for a second-order approximation. Using the factor of 1% beyond the threshold, which yields coincident poles, ensures that both the high- and low-frequency behavior is matched. It must be emphasized that conditions (A) and (B) are violated infrequently, and when they do, the values proposed above result in a simple yet accurate solution, which requires no extra information. The expressions for the reciprocal pole sum in body rows three to five of Table I represent the best approximations that guarantee stability when the first-choice approximations in rows one and two prove to be incompatible with the quadratic (13).

C. Physical Basis of the Model

Computing the first and second moments of the impulse response of the circuit, and using them to generate a transfer function with two poles and one zero, results in the matching of the boundary conditions at time zero and infinity, and geometric

TABLE I
VALUES THAT τ_{sum} SHOULD TAKE

Condition		Value of τ_{sum}
Victim Switching	no violation	τ_p^k
	(B) violated	N/A (use a single pole response)
Aggressor Switching	no violation	$\left(\frac{a_i}{\tau_{G_e}}\right)^2 / \tau_{D_e}$
	(A) violated $\tau_{\text{sum}} < \left(\frac{\tau_{G_e}^v}{\tau_{D_e}^v}\right)^2$	$0.99\left(\tau_{G_e}^v\right)^2 / \tau_{D_e}^v +$ $0.02\left[\tau_{D_e}^v - \sqrt{\left(\tau_{D_e}^v\right)^2 - \left(\tau_{G_e}^v\right)^2}\right]$
	(B) violated $2\left[\tau_{D_e}^v - \sqrt{\left(\tau_{D_e}^v\right)^2 - \left(\tau_{G_e}^v\right)^2}\right]$ $< \tau_{\text{sum}} < 2\tau_{D_e}^v$	$0.01\left(\tau_{G_e}^v\right)^2 / \tau_{D_e}^v +$ $1.98\left[\tau_{D_e}^v - \sqrt{\left(\tau_{D_e}^v\right)^2 - \left(\tau_{G_e}^v\right)^2}\right]$
	(B) violated $2\tau_{D_e}^v < \tau_{\text{sum}} <$ $2\left[\tau_{D_e}^v + \sqrt{\left(\tau_{D_e}^v\right)^2 - \left(\tau_{G_e}^v\right)^2}\right]$	$2.02\left[\tau_{D_e}^v + \sqrt{\left(\tau_{D_e}^v\right)^2 - \left(\tau_{G_e}^v\right)^2}\right]$

properties—namely the area and first moment—of the actual waveform (step response) with the estimated waveform. The boundary conditions are already considered in the particular formulation of the transfer function (i.e., that the waveform starts and ends on a specific rail). Hence, matching the first and second moment of the impulse response does not define a unique solution, as a two-pole–one-zero transfer function has three unknowns. The necessary third equation is obtained by matching circuit components to the reciprocal pole sum.

For the switching of the victim driver with the other inputs grounded, the sum of the open circuit time constants provides a good approximation to the reciprocal pole sum, and combining it with the moments of the circuit for the victim driver switching has a straightforward intuitive motivation. For the switching of an aggressor driver, the geometric properties of the actual waveform (via the first and second moments of the impulse response for an aggressor driver switching) are used to obtain the precise reciprocal pole sum. Since the quadratic (13) obtained from the moments of the impulse response for the victim driver switching contains relevant information about the victim net, combining it with the reciprocal pole sum for an aggressor switching gives a good approximation to the best two-pole–one-zero model. This is a procedure that works for the vast majority of circuits; however, some adjustments are necessary to the reciprocal pole sum for certain pathological cases, which was analyzed in a systematic manner, resulting in Table I.

D. Computational Complexity

1) *Background—Incremental Computation of the Elmore Delay:* The Elmore delay, as mentioned before, has been and is used extensively as a delay metric in VLSI interconnection circuits modeled by a tree where all capacitors are grounded, which is termed a simple tree. Simple trees are characterized by nodes that may have multiple children but only one parent. The Elmore delay is defined as

$$\tau_{D_e} = \sum_{k \in \text{tree}} R_{ke} C_k.$$

Consider the simple tree given in Fig. 2, where the output node is designated as e . According to the definition, the Elmore delay is

$$\tau_{D_e} = R_1 C_1 + (R_1 + R_2) C_2 + (R_1 + R_2 + R_3) C_3 + (R_1 + R_2) C_4 + (R_1 + R_2) C_5. \quad (27)$$

This can be rearranged so that the expression is in terms of the product of the downstream capacitance and resistance at each node on the path from the source to the sink

$$\tau_{D_e} = R_1(C_1 + C_2 + C_3 + C_4 + C_5) + R_2(C_2 + C_3 + C_4 + C_5) + R_3(C_3). \quad (28)$$

All downstream capacitances can be stored at each node by traversing the tree once. Hence, after one traversal of the complete tree, the computation of the Elmore delay at any node requires only that the path from source to sink for that particular instance be traversed, with the product of the resistance and downstream capacitance at each node being summed up. Because of this property, any changes to the capacitance values at any node in the tree require only that those changes be propagated upstream of those nodes where the changes took place. This is known as incremental computation, as only those cached values that are stale need to be updated. Any change to a resistance need only be considered when the metric with respect to a particular node is required, and the path from the root to that node is traversed. Incremental computation bestows considerable savings, and it is one of the principal reasons for the popularity of the Elmore delay.

2) *Computational Complexity of Proposed Metrics:* Altogether, five metrics that depend on the circuit topology are required for the proposed models. The first-order metrics are (5), (7), and (21), and the second-order metrics are (6) and (8).

First-order metrics: An inspection of the first-order metrics (5) and (7) clearly shows their similarity to the Elmore delay. These can be rearranged so that the expressions are formulated as the sum of the products of resistance and downstream capacitance at each node on the path from source to sink. Because of the extra complexity introduced by the coupling capacitances, it is necessary to keep track of individual coupling capacitances at each node. This can be achieved by caching the sum of the downstream self (or total) capacitances, and the sum of the individual downstream coupling capacitances with associated root information at each node. Hence, similar to the Elmore delay, all downstream capacitances are cached from a full tree traversal, and then the output with respect to a particular node e only requires a traversal from the source to e . Also similar to the Elmore delay, any changes to the tree require only that the capacitance changes be propagated to the upstream nodes, resulting in incremental computation being possible.

The final first-order metric (21), the sum of the open circuit time constants, requires that at each node in the summation, that node should be treated as the output. Since the output node is therefore always defined for a given victim net (unlike in the previous metrics where the output can be any node in the tree), the incremental components of the summation in τ_p^* can be cached along with the downstream capacitance. For example, in Fig. 1, node 4 should have CS_5 as downstream self capacitance, and $R_5 \cdot CS_5$ as downstream τ_p^* information. Therefore, this metric requires no extra traversals at all, but instead, it can be computed along with the downstream capacitances. Again, changes to the tree require only that the changes be propagated to upstream nodes.

Second-order metrics: The second-order metrics require the capacitances at each node be weighted individually by a first-order time constant, which is basically expression (4) (in one of the three forms used) for the path defined from the root of the relevant simple tree to the current node, or its coupled counterpart. There are now three issues related to complexity:

1) How much work is needed to calculate the weights for the original tree?

- 2) When the weights are known, how much work needs to be done to calculate the second-order metrics with respect to a particular node?
- 3) Finally, how much work needs to be done to recalculate all the weights once a change or changes have been made to the tree?

Calculation of the weights is demonstrated on the victim net of Fig. 1. The weights required are different for the two expressions, and also for the types of capacitances (i.e., coupling capacitance between two trees, or the total capacitance, at a particular node), but are always characterized in a generic sense by the expression (4). Hence, any technique that works for one will always work for all the weights. For the sake of explanation, assume that the weight consists of $\tau_{D_k}^v$, where only self capacitances are considered, and that the weights at nodes 1 and 2 are τ_1 and τ_2 , etc. Then

$$\tau_1 = R_1(CS_1 + CS_2 + CS_3 + CS_4 + CS_5) \quad (29)$$

and

$$\tau_2 = R_1(CS_1 + CS_2 + CS_3 + CS_4 + CS_5) + R_2(CS_2 + CS_3 + CS_4 + CS_5). \quad (30)$$

The rest of the metrics are calculated in a similar manner. Now, since the weight is always with respect to the root, it is necessary to visit all the nodes once after the downstream capacitance information has been stored on the initial pass. (It is useful also to store the upstream resistance at each node on this pass, so that in future visits to the node, the τ information can be updated instantly, as will be shown later.) All weights can be calculated in one pass by using the property that

$$\tau_{D_n}^v = \tau_{D_m}^v + \tau_{D_{m \rightarrow n}}^v \quad (31)$$

where node m is situated on the path between the root and node n . At branch points, a depth first traversal of all child branches preserves the linearity of the traversal. Hence, the weights for all nodes can be calculated by one full tree traversal once the downstream capacitance information has been stored.

The answer to the second question is straightforward; an inspection of (6) and (8) shows that the form that the outer (second order) summation takes is exactly similar to the inner (first order) summation, which is characterized in a generic way by the expression (4). Therefore, it is possible to cache the downstream $\tau \cdot C$ information (just as the downstream C information was cached for the first-order metrics) and obtain the metrics from the root to a particular node by visiting only the nodes along the path from the root to that node.

So far, two complete traversals have been necessary, one bottom-up pass to store the downstream capacitance information, and one top-down pass, beginning at the root to store the τ information (and the upstream resistance information, which is necessary, later, to minimize computation when changes are made). Now, to calculate the second-order metric to any node, rearranging the terms in the summation exactly as in the first-order calculation allows the downstream $\tau \cdot C$ to be cached in one full traversal. Subsequently, the second-order metric to any node can be calculated simply by visiting all the nodes on the

path from the root to that node. Again, if an imaginary second-order metric is defined to consist only of the self capacitances for simplicity of explanation, the value that would be cached at node 5 on the third (bottom-up) traversal would be $T_5 = T_5 \cdot CS_5$, that at node 4 would be $T_4 = T_5 + \tau_4 \cdot CS_4$, and so on.

Hence, three full traversals are necessary, one bottom-up traversal to store the downstream capacitance information, one top-down traversal to store the weights, and a final bottom-up traversal to store the downstream $\tau \cdot C$ information. None of these passes can be combined as the necessary order is bottom-up, top-down, and bottom-up.

The only remaining question is also the most important; if it is necessary to traverse the entire tree three times each time a change is made, the incremental computation property is lost. However, after a modification to a component, since only the resulting changes in the stored values need to be accounted for, the calculations that required three traversals for the original tree can be accomplished in one traversal. Consider, for example, that the component value CS_2 is changed to CS'_2 . This immediately causes:

- 1) the downstream capacitance values cached at node 2 and all nodes upstream of node 2 to be stale;
- 2) the cached weight (τ) information at all nodes to be stale;
- 3) the cached downstream $\tau \cdot C$ information at all nodes to be stale.

In node 5, for example, the stored downstream capacitance is current (since the changed capacitor is upstream of it), but the weight and downstream $\tau \cdot C$ information is stale. The old weight is

$$\begin{aligned} \tau_5 = & R_1(CS_1 + CS_2 + CS_3 + CS_4 + CS_5) \\ & + R_2(CS_2 + CS_3 + CS_4 + CS_5) \\ & + R_4(CS_4 + CS_5) + R_5(CS_5). \end{aligned} \quad (32)$$

The new weight is

$$\begin{aligned} \tau'_5 = & R_1(CS_1 + CS'_2 + CS_3 + CS_4 + CS_5) \\ & + R_2(CS'_2 + CS_3 + CS_4 + CS_5) \\ & + R_4(CS_4 + CS_5) + R_5(CS_5). \end{aligned} \quad (33)$$

The change is

$$\tau'_5 - \tau_5 = (R_1 + R_2)(CS'_2 - CS_2). \quad (34)$$

Therefore

$$\tau'_5 = \tau_5 + (R_1 + R_2)(CS'_2 - CS_2). \quad (35)$$

This is simply the change in the capacitance multiplied by the resistance that is upstream of the changed capacitance. This is true of all nodes downstream of node 2. At the nodes upstream of node 2, the capacitance change is multiplied by the upstream resistance from that node. Similarly, the downstream $\tau \cdot C$ information can also be calculated and stored. Hence, all stale information can be updated by doing a single bottom-up traversal by considering the difference introduced by the change to the component. First, the changed component is

located, and its upstream resistance ($R_1 + R_2$), which has been stored earlier, is noted. Now, starting from a leaf node, say node 5, for example, a bottom-up traversal is initiated, where both the weight information and the downstream $\tau \cdot C$ information are updated at once. From node 2 upwards, the downstream capacitance also needs to be updated. Hence, the original requirement of three passes for the unprocessed tree has been reduced to a single pass. This principal also applies for resistor changes, and also multiple component changes. That is, the effect of multiple changes can be considered in one pass.

Summary: It was shown that all the metrics have a very simple and small core, which exactly resembles the Elmore delay. The second-order expressions can be described as a weighted Elmore delay; each term in the summation is weighted by either (5) or (7) for that particular node. These are similar to the second moment of the impulse response proposed for simple trees in [4]. Just as the models of [4] represent the minimum computational complexity for second-order models for the class of circuits that were called simple trees, these models represent the minimum complexity for coupled trees. In fact, if the coupling capacitance terms are put to zero (the entire capacitance is lumped into a ground component), the model for the victim tree reverts to the model proposed in [4].

One of the major attractions of the Elmore delay is its incremental computational property. This is a very useful feature, and is the mainstay of several interconnect optimization algorithms. It should be noted that this is independent of the output node. Whatever node is chosen as e in the tree, this hierarchical property holds true. Now, since the constituent summations in the proposed metrics have exactly the same form as the Elmore delay, whose format is basically that of (4), incremental computation is possible for the proposed metrics.

3) *Comparison With Other Metrics:* As shown in Section II, there are an enormous number of models both of a general nature that are not restricted to second-order estimations, and ones that are explicitly second-order. In this section, it is shown that the proposed model has the least complexity of all two-pole-one-zero models in both categories. It is useful to start the discussion with a look at the general techniques AWE [13] and RICE [14].

The latter is basically an efficient implementation of the former, and both require $2q$ moments to generate a q th-order approximation. The computation of the moments in [14] is not less efficient than the computation outlined above. The so-called path-tracing algorithm (with recursive solutions of a dc circuit) presented there can be used to compute the first and second moment with exactly the same efficiency, including caching of values at nodes, though not explicitly articulated in the same manner. The point is, [14] requires four moments to generate a second-order response, and each higher order moment requires a computation that involves a backward and forward substitution of the system matrix, which is equivalent to two tree traversals. Hence, the proposed model saves at least four tree traversals in the processing of the initial tree, by not requiring the third and fourth moment of the impulse response. Each time the circuit is changed, as would be done innumerable times in incremental physical optimization, the

computation of the higher order moments that require four extra traversals is avoided. For component value changes, the general property of incremental computation would hold, but it would necessarily be more expensive for four moments. The first moment (Elmore delay) requires component values to be propagated upstream only; the second moment requires one full tree traversal as has been shown; similarly, the third and fourth moments would require two and three traversals. Therefore, the proposed model saves two full tree traversals for component value modifications.

As mentioned earlier, the model that best merits comparison with the proposed model is [19], the other second-order models either being restricted in topology (such as [4]—simple trees; [17], [18], and [22]—inductive trees with no series capacitors; [26], [27]—parallel lines), restricted in accuracy (such as [20] and [31], which use the same poles for all responses), or lacking a clear physical basis (such as [23] and [24]), which do not translate to a two-pole–one-zero model, and hence cannot be used to generate the responses for aggressors switching, and are less accurate than direct moment matching for the victim switching. When the moments are computed with the algorithms of [14], the explicit two-pole–one-zero model of [19] is the least expensive second-order model that can handle arbitrarily coupled RC trees reported in the literature. Because the proposed model does not require the third moment of the impulse response, it is cheaper than this model by the equivalent of at least two full tree traversals for the unprocessed tree, and at least one full tree traversal for component value changes. Because incremental physical optimization requires many changes to the interconnection tree, the proposed model will often save two complete tree traversals, a huge saving for any reasonably sized circuit. The comparison carried out in Section IV shows that this saving is not at the expense of loss in accuracy.

Finally, it is worth looking at the complexity of circuit collapsing techniques such as passive PRIMA [16], as they only require q moments to generate a q th-order response. In PRIMA, the most critical steps in terms of runtime are an LU decomposition and qN backward and forward substitutions for an N -port circuit, which corresponds to a q th-order response, both of which must be performed on the modified nodal analysis (MNA) matrix of the original network. The LU factorization already makes the complexity at least as much as the three complete traversals required for the unprocessed tree in the proposed methodology. The most efficient manner in which the second can be accomplished is by recursively solving an equivalent DC circuit as in [13]. Hence, the computation of each block moment requires at the minimum one backward and forward substitution, which is equivalent to two tree traversals. Furthermore, there is an overhead introduced by the orthogonalization and orthonormalization procedures, which makes the complexity of a block moment computation more than the equivalent of two tree traversals. This is neglecting the overhead of setting up the MNA matrix, which has to follow the setting up of the spanning tree (unless the netlist is already appropriately ordered, which is not the general case). Hence, the complexity of a second-order Krylov space projection-based approximation does not really bear comparison to the proposed methodology. Such

network collapsing methods are suitable for macromodeling linear blocks by higher order approximations, not for incremental physical optimization.

E. Peak Noise and Delay to Peak Noise

In this paper, the generation of the transfer function, which is the most important aspect of the modeling, is the main concern. Choice of input waveform, driver modeling, and subsequent processing of the waveform depend on the application, and are not covered for the most part. Knowing the transfer function allows the output to be generated for the appropriate input by a number of approaches, including table look-up and heuristic expressions. Explicit expressions are, however, derived for step inputs, which are sufficiently accurate for quite a number of applications. First, the step response at node e when the victim driver switches is given by

$$E_e^v(t) = 1 - \frac{\tau_1^v - \tau_z^v}{\tau_1^v - \tau_2^v} e^{-\frac{t}{\tau_1^v}} - \frac{\tau_z^v - \tau_2^v}{\tau_1^v - \tau_2^v} e^{-\frac{t}{\tau_2^v}}. \quad (36)$$

The step response when an aggressor driver switches is

$$E_e^{a_i}(t) = \frac{\tau_z^{a_i}}{\tau_1^{a_i} - \tau_2^{a_i}} \left(e^{-\frac{t}{\tau_1^{a_i}}} - e^{-\frac{t}{\tau_2^{a_i}}} \right). \quad (37)$$

It is not possible to solve a two-pole (or higher order) waveform explicitly for the delay at a given threshold. Closed-form heuristics for a two-pole waveform can be derived—such as in [19]—but since the complete response for an m driver system will consist of $2m$ exponential waveforms, some iterative procedure needs to be adopted in the general case.

However, (37) can be solved explicitly for the peak noise, and the time at which it occurs. Equating the first time derivative to zero and doing some trivial algebra results in

$$t_{pk} = \frac{\tau_1^{a_i} \tau_2^{a_i}}{(\tau_1^{a_i} - \tau_2^{a_i})} \ln \frac{\tau_1^{a_i}}{\tau_2^{a_i}}. \quad (38)$$

Substituting (38) for t in (37) results in the peak noise from the switching of the driver of aggressor a_i . Knowing the times at which the individual peaks occur is extremely useful in algorithms to process the waveform for delays at a given threshold, when multiple drivers switch at different times. A temporal window can be defined, which is bounded by the first peak and last peak; these peak values can be used as initial guesses for fast and true convergence with the Newton–Raphson algorithm, even for multiple-time constant waveforms.

IV. RESULTS

The proposed metrics were implemented in a tool called DeANo (for Delay And Noise Modeler), and tested on many test beds that cover a wide range of topologies, by comparing the step response against a circuit simulator, Spice, and other moment-based models. Some plots for instructive testbeds are shown in [3] including networks consisting of primary and secondary aggressors where inequalities (A) and (B) are violated. Comparisons are carried out against moment-based techniques such as the two-pole–one-zero model from three moments described in [19], the gamma and hgamma probability

TABLE II
AVERAGE PERCENTAGE ERRORS IN DELAYS AT DIFFERENT THRESHOLDS FOR VICTIM SWITCHING

Testbed Information	Model	Thresholds		
		0.1	0.5	0.9
Cluster 1 No. of Tests: 220	Elmore	-2.1367	-2.5774	-0.2012
	DeANo	-1.1893	-1.1653	-0.1840
	3M model	-1.3722	-1.1161	-0.3358
Cluster 2 No. of Tests: 21	Elmore	-0.9386	-0.9831	-0.0873
	DeANo	-0.1743	-0.3000	-0.0460
	3M model	-0.2704	-0.2960	-0.0220
Cluster 3 No. of Tests: 79	Elmore	-2.5981	-1.6843	-0.0067
	DeANo	-0.7233	-0.4885	0.0108
	3M model	-0.5958	-0.5489	-0.0836
Cluster 4 No. of Tests: 120	Elmore	-1.8112	-1.9938	-0.0143
	DeANo	-0.6458	-0.8451	0.0290
	3M model	-1.4991	-2.1379	-0.1394

distribution models from three moments described in [33] and [24], and the gamma probability distribution model from two moments, also described in [33]. It should be noted that the probability distribution models are valid only for the victim switching. Here, the model is extensively tested against the Elmore delay, the closest less expensive model, and the two-pole-one-zero model from three moments, the closest more expensive model, in thousands of comparisons with the response from a Spice3 simulation serving as the yardstick of the actual waveform. Since the actual and predicted delay at a single threshold can agree very well, and still result in significant deviations along the full waveform, the accuracy was tested at three points along the waveform. For the victim switching, the thresholds are 10%, 50%, and 90%, while for the aggressors they are 25%, 100%, and 25% of the peak amplitude. This is to ensure that three points, with two being on either side of the peak, are tested. The average results are shown in Tables II and III. The numbers shown in each case is the error calculated as a percentage against the actual value as given by Spice; a negative number means that the predicted value was greater than the actual and vice versa. Hence, the smaller the magnitude of the number, the better; a value of zero means zero error. These numbers do not have much significance as indicators of absolute error, but are very useful for comparisons between models.

Different kinds of circuits were grouped into the four cluster categories. The test nodes comprise all nodes in each circuit in each case, covering all possible circumstances such as near-end and far-end coupling and stiff nodes. It can be seen that the proposed model performs very well against the more expensive model computed from three moments, and is in fact often better in the delay predictions. The only area in which it is consistently outperformed by the three moment model is in peak noise

TABLE III
AVERAGE PERCENTAGE ERRORS IN DELAYS AT DIFFERENT THRESHOLDS AND AVERAGE ERROR IN PEAK NOISE FOR AGGRESSOR SWITCHING

Testbed Information	Model	Thresholds			Peak noise
		0.25	1	0.25	
Cluster 1 No. of Tests: 599	DeANo	-0.5190	-0.3649	-0.2101	0.1901
	3M model	-1.0186	-10.7072	-1.4887	0.1414
Cluster 2 No. of Tests: 43	DeANo	-0.6399	-0.9515	-0.2231	0.1131
	3M model	-0.6456	-1.0979	-0.3508	0.1102
Cluster 3 No. of Tests: 198	DeANo	-0.4191	-0.5207	-0.1808	0.1923
	3M model	-0.6635	-1.1880	-0.5218	0.0943
Cluster 4 No. of Tests: 240	DeANo	0.2292	-0.0589	-0.0276	0.1349
	3M model	-0.2668	-0.5847	-0.2277	0.1084

prediction, to varying degrees. However, as the point of the proposed model is to use as a metric of comparison between different topologies, this need not be any kind of limitation in its intended applications of incremental physical optimization and global signal planning.

V. SUMMARY AND CONCLUSION

Closed-form expressions for the first two moments of the impulse response for general arbitrarily coupled RC trees with multiple drivers were presented and used to generate stable and accurate second-order approximations to the transfer function for any switching event. For the victim driver switching, the sum of the open-circuit time constants with regard to the victim net is used to avoid using higher order moments. For the case of an aggressor driver switching, the moments for the victim and aggressor switching are combined, which can be described as an averaging of the dominant poles along the victim and that aggressor. This procedure provides the best opportunity to match the waveform along the charging and discharging paths while saving on higher order moments. The summation of all waveforms result in the complete response at the node of interest. This model represents the minimum complexity for a general second-order model for this class of circuits, and is hence proposed as being useful for initial estimates of delay and noise in complex systems early in the design flow, and particularly for incremental physical optimization.

The point of doing early signal integrity analyses is that where problems are identified, some change in the circuit graph is required. As has been shown in Section III-D-3, the model proposed here saves at a minimum, the equivalent of two complete traversals for the unprocessed (i.e., structurally changed) tree, and one full traversal at the minimum for component value changes in the tree. Since tree changes will occur very often in physical optimization, this is a big saving in computational complexity over other published models. By running extensive tests, it was shown that this saving in computational complexity was achieved without compromising on accuracy, the proposed model having comparable or better accuracy than more expensive models.

APPENDIX I
COMPUTATION OF MOMENTS

When the victim driver switches while all other inputs are grounded, the first moment of the impulse response at the output node e is given by

$$\Upsilon_e^1 = \int_0^\infty t h_e^v(t) dt. \quad (39)$$

Now, the following expression describes the voltage drop from the source to e where $E_e^v(t)$ is the step response at e , and a_1, a_2, \dots are the aggressors. This is obtained by summing up the capacitor currents and adding the drops across each resistor, or in other words, using Kirchoff's voltage and current laws

$$1 - E_e^v(t) = \sum_{k \in v} R_{ke}^v \left[CS_k^v \frac{dE_k^v}{dt} + CC_k^{va_1} \frac{dE_k^{va_1}}{dt} + CC_k^{va_2} \frac{dE_k^{va_2}}{dt} + \dots \right]. \quad (40)$$

The impulse response $h_e^v(t)$ is the first time derivative of the step response. Hence, (39) can be integrated by parts, and (40) substituted in it to yield the following expression

$$\Upsilon_e^1 = \sum_{k \in v} R_{ke}^v [CS_k^v + CC_k^{va_1} + CC_k^{va_2} + \dots] = \tau_{D_e}^v \text{ say.} \quad (41)$$

The second moment of the impulse response at the output node e is given by

$$\Upsilon_e^2 = \int_0^\infty t^2 h_e^v(t) dt.$$

Integrating by parts, it can be shown that this is equivalent to

$$\Upsilon_e^2 = 2 \int_0^\infty t(1 - E_e^v(t)) dt.$$

Using the above expression for the step response and again integrating by parts, this can be shown to be

$$\Upsilon_e^2 = 2 \sum_{k \in v} R_{ke}^v \left[CS_k^v \int_0^1 t dE_k^v + CC_k^{va_1} \int_0^1 t dE_k^{va_1} + CC_k^{va_2} \int_0^1 t dE_k^{va_2} + \dots \right]. \quad (42)$$

The constituent integrals can be evaluated by integrating by parts and using Kirchoff's laws to obtain expressions for the voltages. The first integral is basically the first moment at

node k , for which an expression can be obtained by simply substituting k for e in (41). The other integrals are of the form

$$I_{a_i} = \int_0^1 t dE_k^{va_i}.$$

Integrating by parts, this simplifies to

$$I_{a_i} = \int_0^\infty [1 - E_k^{va_i}(t)] dt.$$

The voltage can be decomposed into two components thus

$$E_k^{va_i} = E_k^v - E_j^{a_i}.$$

Now, circuit laws can be used to obtain expressions for the individual voltages. This first is

$$1 - E_k^v(t) = \sum_{K \in v} R_{Kk}^v \left[CS_K^v \frac{dE_K^v}{dt} + CC_K^{va_1} \frac{dE_K^{va_1}}{dt} + CC_K^{va_2} \frac{dE_K^{va_2}}{dt} + \dots \right]$$

and the second

$$E_j^{a_i} = \sum_{K \in a_i} R_{Kj}^{a_i} \left[CS_K^{a_i} \frac{dE_K^{a_i}}{dt} + CC_K^{a_i v} \frac{dE_K^{a_i v}}{dt} + CC_K^{a_i b_1} \frac{dE_K^{a_i b_1}}{dt} + CC_K^{a_i b_2} \frac{dE_K^{a_i b_2}}{dt} + \dots \right].$$

The superscripts $a_i b_1, a_i b_2 \dots$ in the CC terms indicate the coupling capacitances to tree a_i 's own aggressors. Considering the fact that all nodes not on the victim tree start and end at the same voltage, this simplifies to

$$\begin{aligned} \Upsilon_e^2 = & 2 \sum_{k \in v} R_{ke}^v \\ & \times \left\{ CS_k^v \sum_{K \in \text{vic}} R_{Kk}^v (CS_K^v + CC_K^{va_1} + CC_K^{va_2} + \dots) \right. \\ & + CC_k^{va_1} \left[\sum_{K \in a_1} R_{Kj}^{a_1} CC_K^{a_1 v} + \sum_{K \in v} R_{Kk}^v \right. \\ & \quad \left. \times (CS_K^v + CC_K^{va_1} + CC_K^{va_2} + \dots) \right] \\ & + CC_k^{va_2} \left[\sum_{K \in a_2} R_{Kj}^{a_2} CC_K^{a_2 v} + \sum_{K \in v} R_{Kk}^v \right. \\ & \quad \left. \times (CS_K^v + CC_K^{va_1} \right. \\ & \quad \left. + CC_K^{va_2} + \dots) \right] + \dots \left. \right\}. \end{aligned}$$

This can be expressed in more succinct form by using the first moment of the impulse response when

$$\begin{aligned} \Upsilon_e^2 &= 2 \sum_{k \in v} R_{ke}^v \left\{ CS_k^v \tau_{D_k}^v + CC_k^{va_1} \left[\tau_{D_k}^v + \sum_{K \in a_1} R_{Kj}^{a_1} CC_K^{a_1 v} \right] \right. \\ &\quad \left. + CC_k^{va_2} \left[\tau_{D_k}^v + \sum_{K \in a_2} R_{Kj}^{a_2} CC_K^{a_2 v} \right] + \dots \right\} \\ &= 2 (\tau_{G_e}^v)^2 \text{ say.} \end{aligned}$$

Following an approach identical to that in the former case, the first moment of the impulse response at node e on the victim tree for aggressor a_i switching can be shown to be

$$\Upsilon_e^1 = - \sum_{k \in v} R_{ke}^v CC_k^{va_i} = -\tau_{D_e}^{a_i} \text{ say.}$$

The second moment can also be calculated from an approach similar to the former case, resulting in

$$\begin{aligned} \Upsilon_e^2 &= -2 \sum_{k \in v} R_{ke}^v \\ &\quad \times \left\{ (CS_k^v + CC_k^{va_1} + CC_k^{va_2} + \dots) \tau_{D_k}^{a_i} + CC_k^{va_i} \right. \\ &\quad \left. \times \left[\sum_{K \in a_i} R_{Kj}^{a_i} (CS_K^{a_i} + CC_K^{a_i v} + CC_K^{a_i b_1} + \dots) \right] \right\} \\ &= -2 (\tau_{G_e}^{a_i})^2 \text{ say.} \end{aligned}$$

APPENDIX II BEHAVIOR OF STABILITY FUNCTION

In this section, it is proven that the equality of (A) is always to the left of the first zero-crossing point of LHS, which is important in analyzing the physical basis of potential instability. Since the following is true [where the last condition is necessary for potential instability to occur—see the explanation for (20) in the main text]

$$\tau_{D_e}^v > 0 \quad (\tau_{G_e}^v)^2 > 0 \quad (\tau_{D_e}^v)^2 > (\tau_{G_e}^v)^2$$

the following holds true

$$4 \left((\tau_{D_e}^v)^2 - (\tau_{G_e}^v)^2 \right) + \frac{(\tau_{G_e}^v)^4}{(\tau_{D_e}^v)^2} > 4 \left((\tau_{D_e}^v)^2 - (\tau_{G_e}^v)^2 \right)$$

i.e.,

$$\left((2\tau_{D_e}^v)^2 - \frac{(\tau_{G_e}^v)^2}{\tau_{D_e}^v} \right)^2 > \left(2\sqrt{(\tau_{D_e}^v)^2 - (\tau_{G_e}^v)^2} \right)^2$$

$$2\tau_{D_e}^v - \frac{(\tau_{G_e}^v)^2}{\tau_{D_e}^v} > 2\sqrt{(\tau_{D_e}^v)^2 - (\tau_{G_e}^v)^2}.$$

Rearranging the terms results in

$$\frac{(\tau_{G_e}^v)^2}{\tau_{D_e}^v} < 2 \left[\tau_{D_e}^v - \sqrt{(\tau_{D_e}^v)^2 - (\tau_{G_e}^v)^2} \right].$$

Hence, the equality of (A) is always to the left of the first zero-crossing point of LHS.

ACKNOWLEDGMENT

Very productive discussions with Dr. L.-R. Zheng of the Laboratory of Electronics and Computer Systems, Department of Microelectronics and Information Technology, Royal Institute of Technology, Sweden, are gratefully acknowledged.

REFERENCES

- [1] SEMATECH. (2003). International Technology Roadmap for Semiconductors (ITRS). [Online]. Available: <http://public.itrs.net/Files/2003ITRS/Home2003.htm>
- [2] D. Pamunuwa, S. Elassaad, and H. Tenhunen, "Modelling noise and delay in VLSI circuits," *Electron. Lett.*, vol. 39, no. 3, pp. 269–271, Feb. 2003.
- [3] —, "Analytic modeling of interconnects for deep sub-micron circuits," in *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, San Jose, CA, Nov. 2003, pp. 835–842.
- [4] M. A. Horowitz, "Timing models for MOS circuits," Ph.D. dissertation, Stanford Electr. Lab., Stanford Univ., Stanford, CA, Jan. 1984.
- [5] E. Chiprout, "Interconnect and substrate modelling and analysis: An overview," *IEEE J. Solid-State Circuits*, vol. 33, no. 9, pp. 1445–1452, Sep. 1998.
- [6] D. Pamunuwa, L. R. Zheng, and H. Tenhunen, "Maximizing throughput over parallel wire structures in the deep submicrometer regime," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 2, pp. 224–243, Apr. 2003.
- [7] A. Deutsch *et al.*, "When are transmission lines important for on-chip interconnects," *IEEE Trans. Microw. Theory Tech.*, vol. 45, no. 10, pp. 1836–1846, Oct. 1997.
- [8] I. Ismail, E. G. Friedman, and J. L. Neves, "Figures of merit to characterize the importance of on-chip inductance," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 4, pp. 442–449, Dec. 1999.
- [9] K. Banerjee and A. Mehrotra, "Analysis of on-chip inductance effects using a novel performance optimization methodology for distributed RLC interconnects," in *Proc. Design Automation Conf. (DAC)*, Las Vegas, NV, 2001, pp. 798–803.
- [10] W. C. Elmore, "The transient response of linear damped circuits," *J. Appl. Phys.*, vol. 19, no. 1, pp. 55–63, Jan. 1948.
- [11] J. Rubinstein, P. Penfield, and M. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. CAD-2, no. 3, pp. 202–211, Jul. 1983.
- [12] C. Y. Chu and M. A. Horowitz, "Charge-sharing models for switch level simulation," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. CAD-6, no. 6, pp. 1053–1061, Nov. 1987.
- [13] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 9, no. 4, pp. 352–366, Apr. 1990.
- [14] C. L. Ratzlaff, N. Gopal, and L. T. Pillage, "RICE: Rapid interconnect circuit evaluator," in *Proc. Design Automation Conf. (DAC)*, San Francisco, CA, 1991, pp. 555–560.
- [15] L. M. Silveira, M. Kamon, and J. White, "Efficient reduced-order modeling of frequency-dependent coupling inductances associated with

- 3-D interconnect structures,” in *Proc. Design Automation Conf. (DAC)*, San Francisco, CA, 1995, pp. 376–380.
- [16] A. Odabasioglu, M. Celik, and L. T. Pileggi, “PRIMA: Passive reduced-order interconnect macromodeling algorithm,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 17, no. 8, pp. 645–654, Aug. 1998.
- [17] A. B. Kahng and S. Muddu, “Two-pole analysis of interconnection trees,” in *Proc. Multi-Chip Module Conf. (MCMC)*, Santa Cruz, CA, 1995, pp. 105–110.
- [18] —, “An analytic delay model for RLC interconnects,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 16, no. 12, pp. 1507–1514, Dec. 1997.
- [19] B. Tutuianu, F. Dartu, and L. T. Pillage, “An explicit RC-circuit delay approximation based on the first three moments of the impulse response,” in *Proc. Design Automation Conf. (DAC)*, Las Vegas, NV, 1996, pp. 611–616.
- [20] E. Acar, A. Odabasioglu, M. Celik, and L. T. Pillage, “S2P: A stable 2 pole RC delay and coupling noise metric,” in *Proc. Great Lakes Symp. (GLS) VLSI*, Ypsilanti, MI, 1999, pp. 60–63.
- [21] C. J. Alpert, A. Devgan, and C. V. Kashyap, “RC Delay metrics for performance optimization,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 20, no. 5, pp. 571–582, May 2001.
- [22] Y. I. Ismail and E. G. Friedman, “Effects of inductance on the propagation delay and repeater insertion in VLSI circuits,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 2, pp. 195–206, Apr. 2000.
- [23] R. Kay and L. T. Pillage, “PRIMO: Probability interpretation of moments for delay calculation,” in *Proc. Design Automation Conf. (DAC)*, San Francisco, CA, 1998, pp. 463–468.
- [24] T. Lin, E. Acar, and L. T. Pillage, “ h -gamma: An RC delay metric based on a gamma distribution approximation to the homogeneous response,” in *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, San Jose, CA, 1998, pp. 19–25.
- [25] A. B. Kahng, S. Muddu, and E. Sarto, “On switch factor based analysis of coupled RC interconnects,” in *Proc. Design Automation Conf. (DAC)*, Los Angeles, CA, Jun. 2000, pp. 79–84.
- [26] H. Kawaguchi and T. Sakurai, “Delay and noise formulas for capacitively coupled distributed RC lines,” in *Proc. Asian and South Pacific Design Automation Conf.*, Yokohama, Japan, Jun. 1998, pp. 35–43.
- [27] A. B. Kahng, S. Muddu, and D. Vidhani, “Noise and delay uncertainty studies for coupled RC interconnects,” in *Proc. Application Specific Integrated Circuit/System-on-Chip (ASIC/SOC)*, Washington, DC, 1999, pp. 3–8.
- [28] A. B. Kahng, S. Muddu, N. Pol, and D. Vidhani, “Noise model for multiple segmented coupled RC interconnects,” in *Proc. Int. Symp. Quality Electronic Design (ISQED)*, San Jose, CA, 2001, pp. 145–150.
- [29] A. Devgan, “Efficient coupled noise estimation for on-chip interconnects,” in *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, San Jose, CA, 1997, pp. 147–153.
- [30] M. Takahashi, M. Hashimoto, and H. Onodera, “Crosstalk noise estimation for generic RC trees,” in *Proc. Int. Conf. Computer Design (ICCD)*, Austin, TX, 2001, pp. 110–116.
- [31] X. Tong and M. Marek-Sadowska, “Efficient delay calculation in presence of crosstalk,” in *Proc. Int. Symp. Quality Electronic Design (ISQED)*, San Jose, CA, 2000, pp. 491–497.
- [32] B. L. Cochrun and A. Gabel, “On the determination of the transfer function of electronic circuits,” *IEEE Trans. Circuit Theory*, vol. CT-20, no. 1, pp. 16–20, Jan. 1973.
- [33] M. Celik, L. Pileggi, and A. Odabasioglu, *IC Interconnect Analysis*. Boston, MA: Kluwer, May 2002.



Dinesh Pamunuwa (M'04) received the B.Sc. degree in engineering with honors from the University of Peradeniya, Peradeniya, Sri Lanka, in 1997, and the Ph.D. degree in electronic system design from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 2003.

He is currently a faculty member of the Department of Engineering at Lancaster University, Lancaster, U.K. In 2002, he was also at Cadence Berkeley Laboratories in Berkeley, CA. He has worked extensively on interconnect design and signal integrity issues and methodologies for electronic system design. He is the author and coauthor of many papers in this area. Dinesh is also the cofounder of PsiTech, an electronics and software consultancy company based in Sweden and Sri Lanka. His current research interests include nanoelectronic devices and circuits and tera-scale integration in the nanometer regime.



Shauki Elassaad received the B.S. and M.S. degrees in computer engineering from North Carolina State University, Raleigh, in 1988 and 1991, respectively.

He is involved in research and development related to all aspects of Signal Integrity at Rio Design Automation. From 1999 to 2004, he worked at Cadence Berkeley Laboratories, Berkeley, CA, where his research included various aspects of design planning timing, signal integrity, interconnect design, and hierarchical floorplanning. From 1997 to 1999, he architected and designed timing algorithms for hierarchical floorplanning methodologies. During his tenure at High-Level Design Systems from 1993 to 1997, he was involved in various products related to physical design—clock design, delay calculation, and global routing. From 1991 to 1993, he worked at Intel Corporation where he designed algorithms for extraction and delay analysis for high-end microprocessors. His research interests include design planning, signal and power integrity, and digital design.



Hannu Tenhunen received the Diploma Engineer degree in electrical engineering and computer sciences from Helsinki University of Technology, Helsinki, Finland, in 1982, and the Ph.D. degree in microelectronics from Cornell University, Ithaca, NY, in 1986.

During 1978–1982, he was with Electron Physics Laboratory, Helsinki University of Technology, and from 1983 to 1985 at Cornell University as a Full-bright scholar. Since September 1985, he has been with the Signal Processing Laboratory, Tampere University of Technology, Tampere, Finland, as an Associate Professor. He was also a coordinator of the National Microelectronics Programme of Finland during 1987–1991. Since January 1992, he has been with the Royal Institute of Technology (KTH), Stockholm, Sweden, where he is a Professor of electronic system design. His current research interests are in very large scale integration (VLSI) circuits and systems for wireless and broadband communication, and related design methodologies and prototyping techniques. He has made over 400 presentations and publications on IC technologies and VLSI systems worldwide, and has over 16 patents pending or granted.