Analytic Modeling of Interconnects for Deep Sub-Micron Circuits

Dinesh Pamunuwa[*] Shauki Elassaad[**] Hannu Tenhunen[*]

*Laboratory of Electronics and Computer Systems
Dept. of Microelectronics and Information Technology
Royal Institute of Technology, Kista 16440, Sweden
dinesh|hannu@imit.kth.se

**Cadence Berkeley Laboratories
Cadence Design Systems
Berkeley, CA. 94704, USA
shauki@cadence.com

ABSTRACT

Closed form equations for second order transfer functions of general arbitrarily-coupled *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. The simplicity, accuracy and generality of the models make them suitable for use in early delay and noise planning of global signals in complex systems.

INTRODUCTION

Accurately analyzing the impact of delay and noise on performance and functionality has become very important in modern VLSI circuits. The majority of signal wires are typically very lossy, with a high degree of capacitive coupling. This, together with smaller signal rise times, results in heavy cross-talk, which couples a noise voltage onto the victim net. 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 size of the systems prohibits expensive dynamic simulation. Consequently the subject of delay and noise modeling for 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 cross-talk on either use heuristics that are tailored for specific topologies, or use multiple moments that make them expensive.

Our contribution in this paper is as follows. We present new closed form models for generating second order transfer functions from each driver to the receiver in coupled trees such as shown in Fig. 1, with guaranteed stability. These equations, which are derived from a rigorous theoretical treatment, define the poles and zeros *explicitly* in terms of the circuit elements. They are based on the first two moments of the impulse response, and are linear in complexity, resulting in a saving over other explicit moment-based 2-pole-1-zero models. With our closed form models, the intuition and algorithmic simplicity of the Elmore delay are retained as will be shown. Our work can best be described as an extension of the work for *simple trees* (an *RC* tree where all capacitors are grounded) reported in [8], to *coupled trees* (an *RC* tree consisting of simple trees connected together via series capacitors - Fig. 1). Just as that model represents the minimum complexity associated with a second order response for simple trees, our model represents the minimum complexity for a second order response for coupled trees, when no compromise is made on generality.

Related Work

There is a large body of literature that deals with delay and noise modeling in simple and coupled trees. One of the most important metrics for simple trees, the first moment of the impulse response, is known as the Elmore delay [7]. Its attraction is that it has unmatched algorithmic simplicity and elegance, explicitly matches the circuit elements to an upper bound on the delay, and yet exhibits good *fidelity* in interconnect optimization algorithms [22]. When used as the dominant time constant however, its error can be as high as several hundred percent, especially for near-end nodes. Also noise effects cannot be included, as a minimum of two time constants are required to model a noise-voltage spike.

To consider the effect of noise, timing analyzers often use the concept of worst-case, 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 whose value is dependent 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 was recently shown to not even represent an upper bound on the delay [13]. A lot of research has focused on simplified configurations of interest. In [15] the authors use the first moment of the impulse response to generate single pole responses for uniformly coupled *RC* lines, while [14] presents a two pole response for a single section coupled π circuit with arbitrary ramp inputs. They extend it to accommodate multiple segmented aggressors in [12], but the allowed topology is still very limited.

Historically, a landmark paper that established bounds which serve as indicators for poor prediction by the Elmore model is [22]. Then 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 [8] and extended to encompass charge sharing networks in [4]. Later, generic moment-based techniques that allowed the calculation of an arbitrary number of poles for any kind of linear circuit were developed in [20]. An implementation that is optimized for the tree like structures of interconnects was proposed in [21]. These techniques depend on the Pade approximation, which typically requires $2q$ moments for a q^{th} order approximation. Hence obtaining a second order model requires the calculation of four moments. Reduced-order models based on the Arnoldi algorithm [23] match q moments to a q^{th} order approximation. An example is [19], which gives reduced order models for linear systems. However the nodal matrices of the system need to formed, and at least one LU decomposition of the admittance matrix (which has a cubic complexity) is necessary. For initial analysis of complex systems which involves many iterations, such techniques are best avoided when possible.

There are several explicit 2-pole-1-zero models that have been

ICCAD'03, November 11-13, 2003, San Jose, California, USA.

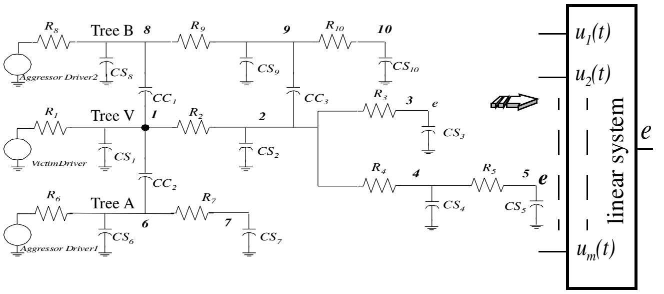

Figure 1. Example of *coupled* RC tree

reported in the literature. For simple trees, the analytic models of [8] represent the minimum computational complexity for a second-order model. Stable models based on the first three moments were proposed in [26] and [1]. Different approaches were suggested in [16] and [17], where the moments of the circuit are used to match a probability density function (gamma distribution) to the impulse response and step response respectively. The underlying circuit transfer function in these models have coincident poles with a real number order. In [18], the authors match the first two moments of the circuit to a Weibull distribution. Alternate second order models for the transfer function include those reported in [10] and [11], which involve generating equivalent circuits and are more suited for highly inductive lines.

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 [10], [11], [26] and [1]. The authors of [2] present two 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 [9].

Now as mentioned, it is often necessary to know the coupled noise amplitude explicitly, to check for spurious errors caused by switching nets disturbing the logic state of a quiescent net. A single pole noise metric for general circuits was proposed in [6]. Although computationally efficient, some simplifying assumptions in the formulation of the model may cause the results to be very pessimistic. Some of the works mentioned above which present models for estimating the effect of noise on delay also report noise metrics ([1], [15] and [12]). In [24] the authors use circuit transformations to simplify a general tree to a 2-π model when analytic formulae can be used, but intermediate steps require the calculation of admittances at each branch point and the estimation of equivalent capacitances which increase run time and impact on the accuracy.

When dealing with multiple driver systems such as 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. The authors of [25] and [3] adopt such a methodology, where an attempt is made to generate transfer functions from each driver to the receiver. However the only concession to different inputs (and hence different charging/discharging paths) is calculating a unique zero; the poles of the transfer function for all switching events are the same, and are the two lowest frequency poles of the system. They are estimated from the methodology proposed in [5], 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 to model interconnect systems can give rise to large errors as the results will be skewed by the highest parasitics in the coupled tree, regardless of their influence on the particular switching event.

DERIVATION OF PROPOSED MODEL

In this paper we are only concerned with the generation of the transfer function, which is the most important aspect of the modeling. Processing of the composite waveform and linearization of non-linear elements can be accomplished in a variety of ways that are suitable for the specific application. Linearization can be accomplished either through substitution of equivalent linear elements or by using some form of convolution in the time domain; non-ideal input waveforms can be similarly modelled. The task which dominates run time for any circuit with more than a few hundred nodes is computation of the moments. There is a clear delineation between linearization of the actual circuit, and solving of the linearized circuit. We concentrate on the latter part here.

Consider Fig. 1 which shows an example 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 single output system as shown. In our methodology we use linear superposition 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). Now in general, all the natural poles of the system contribute to the step response for any switching event where the other inputs are grounded, but their relative contribution varies greatly according to the zeros for a particular switching event. Since the transfer function is limited to two poles, it is important that for each path the two-pole-single-zero model that *best fits that particular charging or discharging path* is calculated.

A coupled *RC* tree is characterized by a resistive path from the output node *e* to the forcing (victim) driver, and series capacitive elements to other (aggressor) 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. Therefore the transfer function characterizing the response to the victim switching has a zero on the negative part of the real axis:

$$H_v(s) = \frac{1 + s\tau_{z,v}}{(1 + s\tau_{1,v})(1 + s\tau_{2,v})} \qquad (1)$$

while that for an aggressor switching has a zero at the origin.

$$H_{a_i}(s) = \frac{s\tau_{z,a_i}}{(1 + s\tau_{1,a_i})(1 + s\tau_{2,a_i})} \qquad (2)$$

Computation of Moments

In the following sections, expressions are presented for the first and second moment of the impulse response for general coupled trees, which form the core of our models. The derivation is based on Kirchoff's laws and integration by parts, and omitted due to lack of space. Fig. 1 can be referred to in the following descriptions. First our notation is described below.

CS_k^p = capacitance to ground at node k in pth tree

$CC_{kj}^{pq} = C_{kj}^{pq}$ = capacitance between node k on pth tree and node j on qth tree where first sub(super) script refers to reference tree

C_k^p = total capacitance at node k on pth tree

R_{ke}^p = shared resistance from source to nodes e and k on tree p

Υ_k^n = nth moment of the impulse response at the kth node

$$\tau_{D_e}^{t_r t_i} = \sum_{k \in t_r} R_{ke}^{t_r} C_k^{t_r t_i} \qquad (3)$$

Superscripts always refer to simple trees while subscripts always refer to nodes, except in the definition for moments, where the superscript refers to the order of the moment. Rail voltages are normalized to 0 and 1 and the expressions derived for a positive step, without loss of generality. The quantity (3) 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 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^{vb} 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 [8]. The lower case subscript in τ_{D_e} (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 first moment of the impulse response at the receiver node e for the victim driver switching is defined as:

$$\Upsilon_{e,v}^1 = \int_0^\infty t h_e^v(t)\,dt \qquad (4)$$

Now the impulse response is the first time derivative of the step response, for which an expression can be formulated by summing up the capacitor currents, or in other words by applying Kirchoff's current and voltage laws. This can then be integrated by parts to yield (5), where a_1, a_2.. are the aggressors.

$$\Upsilon_{e,v}^1 = \sum_{k \in victim} R_{ke}^v [CS_k^v + CC_k^{va_1} + CC_k^{va_2} + \ldots] = \tau_{D_e}^v \qquad (5)$$

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

$$\Upsilon_e^2 = \int_0^\infty t^2 h_e^v(t)\,dt \qquad (6)$$

Following the procedure described above in two stages, this can be shown to be equivalent to (7).

$$\Upsilon_{e,v}^2 = 2\sum_{k \in v} R_{ke}^v \left\{ C_k^v \tau_{D_k}^v + CC_k^{va_1} \tau_{D_j}^{a_1 v} + CC_k^{va_2} \tau_{D_{j'}}^{a_2 v} + \ldots \right\} = 2(\tau_{G_e}^v)^2 \text{ say} \qquad (7)$$

From 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:

$$\Upsilon_{e,a_i}^1 = -\sum_{k \in v} R_{ke}^v CC_k^{va_i} = -\tau_{D_e}^{a_i} \qquad (8)$$

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

$$\Upsilon_{e,a_i}^2 = -2\sum_{k \in v} R_{ke}^v \left\{ C_k^v \tau_{D_k}^{va_i} + CC_k^{va_i} \tau_{D_j}^{a_i} \right\} = -2\left(\tau_{G_e}^{a_i}\right)^2 \text{ say} \qquad (9)$$

The expressions in (5), (7), (8), (9) and (19) described later, form the basis of our proposed models.

Matching Moments to Time Constants

Now the interest is in generating the best two pole single zero transfer function for the response at the output node for any given switching event. 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. From the expansion, the following identity can be observed:

$$\Upsilon^n = (-1)^n \frac{d^n}{ds^n} H(s) \bigg|_{s=0} \qquad (10)$$

Using (1), (10), (5) and (7), it can be seen that:

$$\tau_{1,v} + \tau_{2,v} - \tau_{z,v} = \tau_{D_e}^v \qquad (11)$$

$$(\tau_{1,v} + \tau_{2,v} - \tau_{z,v})(\tau_{1,v} + \tau_{2,v}) - \tau_{1,v}\tau_{2,v} = (\tau_{G_e}^v)^2 \qquad (12)$$

Now additional information is necessary to solve for the three unknowns in (11) and (12). If the reciprocal pole sum is designated as τ_{sum}, these two equations can be combined to form the following quadratic, which yields two time constants.

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

Other than τ_{sum}, the other metrics in the equation, 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, equation (13) can be used to generate the pole time constants for *all* switching events, by using the appropriate reciprocal pole sum.

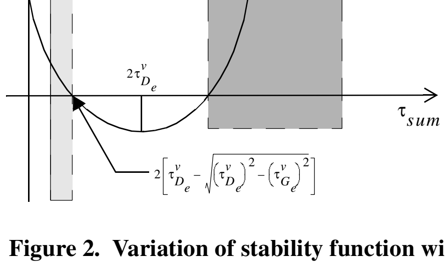

Figure 2. Variation of stability function with τ_{sum}

Proceedings of the International Conference on Computer Aided Design (ICCAD'03)
1092-3152/03 $ 17.00 © 2003 ACM

Now first, 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 first limiting condition is that the magnitude of the square root should be greater than the reciprocal pole sum:

$$\tau_{sum} > \sqrt{\tau_{sum}^2 - 4[\tau_{sum}\tau_{D_e}^{v} - (\tau_{G_e}^{v})^2]} \tag{14}$$

This is true if the following holds:

$$\tau_{sum} > (\tau_{G_e}^{v})^2 / \tau_{D_e}^{v} \tag{A}$$

That is to say, the reciprocal pole sum must be large enough.

The second limiting condition is that the sign under the radical should be positive.

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

When (A) is fulfilled the second term in (15) is negative, and hence the inequality is not guaranteed. The left hand side (*LHS*) 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:

$$\tau_{sum} = 2\left[\tau_{D_e}^{v} \pm \sqrt{(\tau_{D_e}^{v})^2 - (\tau_{G_e}^{v})^2}\right] \tag{16}$$

Obviously, both of these points are on the right hand side of the vertical axis. Now first, if the sign under the radical 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:

$$(\tau_{D_e}^{v})^2 > (\tau_{G_e}^{v})^2 \tag{17}$$

Using this property and the fact that both of these quantities are positive, it can easily be proved that:

$$(\tau_{G_e}^{v})^2 / \tau_{D_e}^{v} < 2\left[\tau_{D_e}^{v} - \sqrt{(\tau_{D_e}^{v})^2 - (\tau_{G_e}^{v})^2}\right] \tag{18}$$

Hence the equality of (A) is always to the left of the first zero crossing point of *LHS*, and we have the shape of the parabola (Fig. 2). 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 we end up with one negative time constant. If τ_{sum} is situated between the zero crossing points, we get complex poles. 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, we get the next condition:

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

Now the stability conditions have been identified, the values for τ_{sum} that give the best response for the 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 we shall call τ_p^k. This is simply the summation of the products of all capacitances connected to the victim line with the driving point resistance to each of those capacitors:

$$\tau_p^k = \sum_{k \in v} [R_{kk}^{v} CS_k^{v} + (R_{kk}^{v} + R_{jj}^{a_1}) CC_k^{va_1} + (R_{kk}^{v} + R_{jj}^{a_2}) CC_k^{va_2} + \ldots]$$

which can be simplified to the following using (3):

$$\tau_p^k = \tau_D^{v} + \tau_D^{a_1 v} + \tau_D^{a_2 v} + \ldots \tag{19}$$

This is a good approximation for the sum of the pole time constants [5], giving:

$$\tau_{1,v} + \tau_{2,v} = \tau_p^k \tag{20}$$

Substituting (20) for τ_{sum} in (11) and (13) result in the zero time constant, and pole time constants respectively, for the victim switching. Since $\tau_p^k > \tau_{D_e}^{v}$ and (17) always has to be true for instability to occur, the following has to hold:

$$\tau_p^k > (\tau_{G_e}^{v})^2 / (\tau_{D_e}^{v})^2 \tag{21}$$

Therefore (A) is always true and the only possible stability violation in this case is (B); i.e. very occasionally, using τ_p^k 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 we shall show in the results section.

Secondly, to solve for the poles and zeros associated with an aggressor switching, (2), (10), (8) and (9), are combined to give:

$$\tau_{D_e}^{a_i} = \tau_{z,a_i} \tag{22}$$

$$\left(\tau_{G_e}^{a_i}\right)^2 = \tau_{z,a_i}(\tau_{1,a_i} + \tau_{2,a_i}) \tag{23}$$

Now the zero time constant is available immediately in (22), and dividing (23) by (22) results in the reciprocal pole sum:

$$\left(\tau_{G_e}^{a_i}\right)^2 / \tau_{D_e}^{a_i} = \tau_{1,a_i} + \tau_{2,a_i} \tag{24}$$

The pole time constants can be obtained by substituting (24) as τ_{sum} in (13). Now it can be seen from an inspection of the relevant expressions that 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 (A), the simplest and most logical remedy is to increase it so that is in the lightly hatched area. When (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 inequality will generate coincident poles which is 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, such as 1%. Using this approach, the values that τ_{sum} should take in the different cases are summarized in Table 1.

Of the two, (A) being violated is by far the most 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, as shown in Fig. 5. The instability in the solution predicted by (13) occurs because the reciprocal pole sum given by (24) accurately reflects the high frequency nature of the poles in the aggressor's charging path, but $\tau^{v}_{D_e}$ and $\left(\tau^{a_i}_{G_e}\right)^2$ reflect the much lower frequency content of the victim's dominant poles, and the gap is too much to bridge. 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 of Fig. 2. Since the equality will generate coincident poles which is not acceptable, the exact value should be chosen so that it is slightly greater than the equality, which can be achieved with a percentage factor, such as 1%. This yields accurate results, because the intention is to generate the *best two pole single zero model*; in other words the poles and zero need not equate to 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. Following this approach, the values that τ_{sum} should take in the different cases are summarized in Table 1. It must be emphasized that conditions (A) and (B) are violated infrequently, and when they do, the solutions proposed above result in a simple yet accurate solution, which requires no extra information.

COMPUTATIONAL COMPLEXITY

An inspection of the first order metrics (5) and (8) 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.

Table 1. Different Values for τ_{sum}

Condition		Value of τ_{sum}
Victim switching	no violation	τ^{k}_{p}
	(B) violated	N/A (use a single pole response)
ggressor switching	no violation	$\left(\tau^{a_i}_{G_e}\right)^2 / \tau^{a_i}_{D_e}$
	(A) violated: $\tau_{sum} < \left(\tau^{v}_{G_e}\right)^2 / \tau^{v}_{D_e}$	$0.99\left(\tau^{v}_{G_e}\right)^2 / \tau^{v}_{D_e} + 0.02$ $\left[\tau^{v}_{D_e} - \sqrt{\left(\tau^{v}_{D_e}\right)^2 - \left(\tau^{v}_{G_e}\right)^2}\right]$
	(B) violated: $2\left[\tau^{v}_{D_e} - \sqrt{\left(\tau^{v}_{D_e}\right)^2 - \left(\tau^{v}_{G_e}\right)^2}\right]$ $< \tau_{sum} < 2\tau^{v}_{D_e}$	$0.01\left(\tau^{v}_{G_e}\right)^2 / \tau^{v}_{D_e} + 1.98$ $\left[\tau^{v}_{D_e} - \sqrt{\left(\tau^{v}_{D_e}\right)^2 - \left(\tau\tau^{v}_{G_e}\right)^2}\right]$
	(B) violated: $2\tau^{v}_{D_e} < \tau_{sum} < 2\left[\tau^{v}_{D_e} + \right.$ $\left.\sqrt{\left(\tau^{v}_{D_e}\right)^2 - \left(\tau^{v}_{G_e}\right)^2}\right]$	$2.02\left[\tau^{v}_{D_e} + \right.$ $\left.\sqrt{\left(\tau^{v}_{D_e}\right)^2 - \left(\tau^{v}_{G_e}\right)^2}\right]$

The final first order metric (19), 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^{k}_{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 $R_5 \bullet CS_5$ as downstream τ^{k}_{p} information. Therefore this metric requires no extra traversals at all, but instead can be computed along with the downstream capacitances. Again, changes to the tree require only that the changes be propagated to upstream nodes.

The second order metrics require the capacitances at each node be weighted individually by a first order time constant, which is basically expression (3) (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 the 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 any node?
3. 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 are demonstrated on the victim net of Fig. 1. The weights required are different for the two expressions, and also different for types of capacitances (i.e coupling capacitance between two trees, or the total capacitance, at a particular node), but always characterized in a generic sense by the expression (3). Hence any technique that works for one will always work for all the weights. For the sake of explanation, let us assume that the weight consists of $\tau^{v}_{D_k}$ where only self capacitances are considered, and that the weights at nodes *1*, *2* are τ_1, τ_2 etc. Then:

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

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

The rest of the metrics are calculated similarly. 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:

$$\tau^{v}_{D_n} = \tau^{v}_{D_m} + \tau^{v}_{D_{m \to n}}$$

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 (7) and (9) shows that the form 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 (3). Therefore it is possible to cache the downstream $\tau \bullet C$ 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 $\tau \bullet C$ 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 $T_5 = \tau_5 \bullet CS_5$, that at node *4* would be $T_4 = T_5 + \tau_4 \bullet CS_4$, and so on. Hence three full traversals are necessary, one bottom-up pass to store the downstream capacitance information, one top-down pass to store the weights, and a final bottom-up pass to store the downstream $\tau \bullet C$ 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 $\tau \bullet C$ 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 $\tau \bullet C$ information is stale. The old weight is:

$$\tau_5 = R_1(CS_1 + CS_2 + CS_3 + CS_4 + CS_5) + R_2(CS_2 + CS_3 + CS_4 + CS_5) \\ + R_4(CS_4 + CS_5) + R_5(CS_5)$$

The new weight is:

$$\tau_5' = R_1(CS_1 + CS_2' + CS_3 + CS_4 + CS_5) + R_2(CS_2' + CS_3 + CS_4 + CS_5) \\ + R_4(CS_4 + CS_5) + R_5(CS_5)$$

The change is:

$$\tau_5' - \tau_5 = (R_1 + R_2)(CS_2' - CS_2)$$

Therefore:

$$\tau_5' = \tau_5 + (R_1 + R_2)(CS_2' - CS_2)$$

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 $\tau \bullet C$ 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 which has been stored earlier, $(R_1 + R_2)$ 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 $\tau \bullet C$ information is updated at once. From node *2* upwards, the downstream capacitance also needs to be updated. Hence the original requirement of three passes for the virgin 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.

RESULTS

The proposed metrics were tested on several different test beds which cover a wide range of topologies, by comparing the step response against a circuit simulator, Spectre. Due to space restrictions, only the results pertaining to three which illustrate all the corner cases are shown; the tree of Fig. 3 consisting of the victim, three primary aggressors, and three secondary aggressors (representing an arbitrarily-coupled circuit, where inequality (B) is violated when solving for the poles of the victim switching), the circuit of Fig. 5 (with far end coupling where inequality (A) is violated when solving for the poles of the aggressor switching), and the tree of Fig. 6, with four primary and four secondary aggressors (representing global distributed interconnects). Shown in Fig. 4 are the waveforms at node *e* of the circuit in Fig. 3, for each driver switching. It can be seen that the model prediction is very close to the Spectre simulation. Since the actual and predicted delay at a single threshold can agree very well, and still result in significant deviations along the full waveform, we tested the accuracy 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. For the aggressors, the error at different thresholds is given as a fraction of the pulse width between the first and last threshold. The waveforms for the circuit of Fig. 5 are shown alongside, and those of Fig. 6 in Fig. 7.

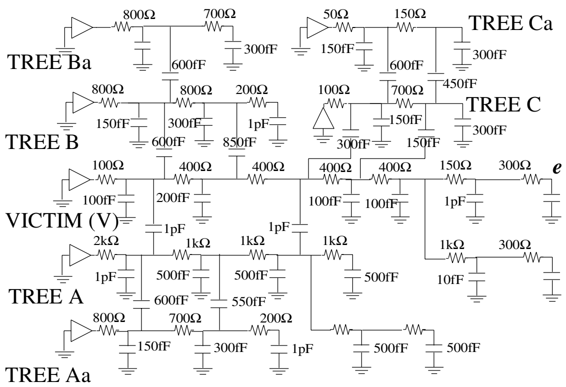

Figure 3. Testbed 1: arbitrarily-coupled tree

SUMMARY AND CONCLUSIONS

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. The summation of all waveforms results in the complete response at the node of interest. These represent new models for estimating delay and noise in complex systems, with no compromise on generality, and in fact subsume a lot of models that address simplified structures.

Computing the first two 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 boundary conditions at time zero and infinity, and geometric 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 contain 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-single-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 1.

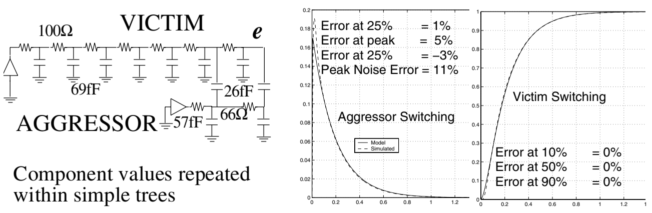

Figure 5. Example of far-end coupling

The proposed models have an Elmore-like flavour, and the algorithm outlined here allows the moments to be calculated with the absolute minimum computational effort. However when the moments are computed in a hierarchical manner, starting from the solution to the DC circuit (a procedure known as *path tracing*) the same refinements are possible. Our claim that these models represent the minimum complexity associated with a two-pole-one-zero model for this class of circuits is based instead on the fact that the sum of the open circuit time constants is used instead of the third moment, which results in a saving of at least one complete tree traversal (or equivalent arithmetic operations). Two moments can be used to map the response to a probability function, but then the model reverts to a coincident pole transfer function, which reduces the degree of freedom, or the generality of the model.

For testing purposes, the models we proposed were used to derive the time domain waveform for the step response. For the delay at a given threshold, the accuracy was found to be more than 90% on average, even for complex circuits such as shown in Fig. 3 and Fig. 6. The time at which the peak noise occurs was predicted with even better accuracy. The peak noise itself was predicted with an accuracy of about 85% or higher in general. These figures cannot be claimed as being hard bounds for all possible circuit topologies as it is always possible to create a circuit which is poorly represented by a two pole response. However the models did perform very well when tested over a wide range of circuits that are representative of coupled interconnect structures in nano-meter technologies.

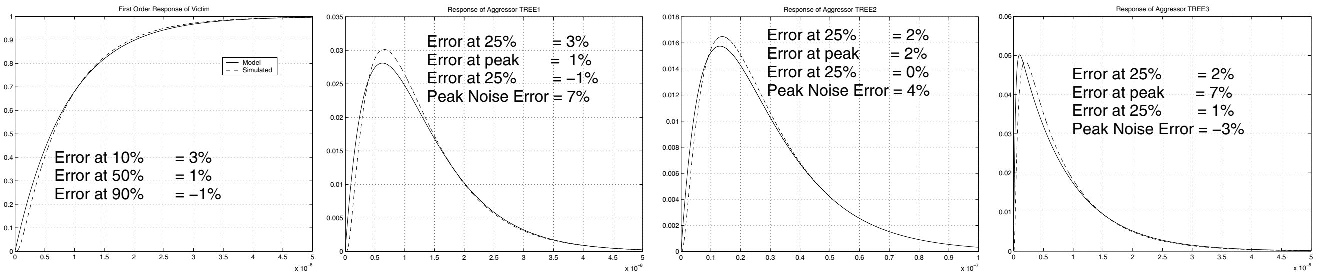

Fig.4.a Victim Switching

Fig.4.b Tree B Switching

Fig.4.c Tree A Switching

Fig.4.d Tree C Switching

Figure 4. Waveforms for testbed 1 of Figure 3.

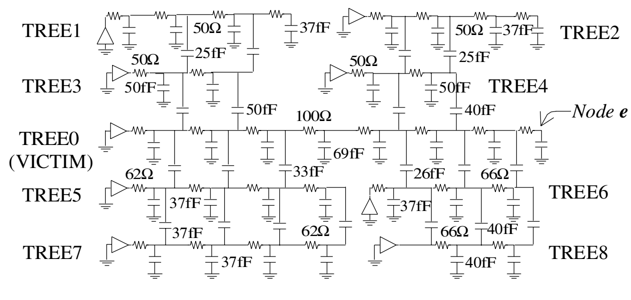

Figure 6. Testbed 2 (values repeated within simple trees)

Explicit second-order noise models can be derived easily by equating the first derivative of the waveforms to zero. These can be used with the single-pole bounds reported in [6] to gain an idea of the uncertainty of the estimate. Calculating the delay requires a few iterations for a multiple-pole waveform, which presents a negligible overhead in comparison with the moment computation for any reasonably sized circuit. The simplicity and accuracy of these models combined with their generality, in comparison with other reported work, should make them useful in delay and noise estimations in complex systems, early in the design flow.

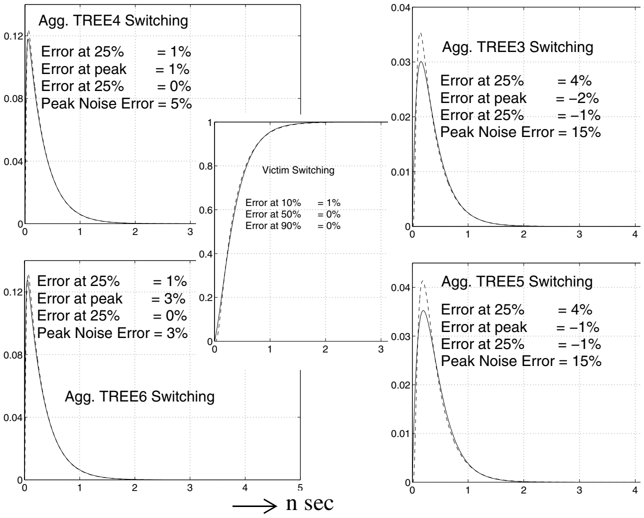

Figure 7. Waveforms for Testbed 2

REFERENCES

[1] E. Acar, A. Odabasioglu, M. Celik, and L. T. Pileggi, "S2P: A stable 2 pole RC delay and coupling noise metric," in *Proc. GLSVLSI*, 1999, pp. 60 -63.

[2] C. J. Alpert, A. Devgan, and C. V. Kashyap, "RC Delay metrics for performance optimization," *IEEE Trans. Comp.-Aided Design of ICs and Sys.*, vol. 20, no. 5, pp. 571-582, May 2001.

[3] L. H. Chen and M. Marek-Sadowska, "Efficient closed-form cross-talk delay metrics," in *Proc. ISQED*, 2002, pp. 431-6.

[4] C. Y. Chu and M. A. Horowitz, "Charge-sharing models for switch level simulation," *IEEE Trans. Comp.-Aided Design*, vol. CAD-6, no. 6, pp. 1053-1061, Nov. 1987.

[5] B. L. Cochrun and A. Grabel, "On the determination of the transfer function of electronic circuits," *IEEE Trans. Circuit Theory*, vol. CT-20, pp.16-20, Jan. 1973.

[6] A. Devgan, "Efficient coupled noise estimation for on-chip interconnects," in *Proc. ICCAD*, 1997, pp. 147-153.

[7] W. C. Elmore, "The transient response of linear damped circuits," *J. Appl. Physics*, vol. 19, pp-55-63, Jan. 1948.

[8] M. A. Horowitz, "Timing models for MOS circuits," Ph.D. dissertation, Stanford Electronics Laboratories, Stanford University, Stanford, CA, Jan. 1984.

[9] Y. I. Ismail and E. G. Friedman, "Effects of inductance on the propagation delay and repeater insertion in VLSI circuits," *IEEE Trans. VLSI Systems*, vol. 8, pp. 195-206, Apr. 2000.

[10]A. B. Kahng and S. Muddu, "Two-pole analysis of interconnection trees," in *Proc. MCMC*, 1995, pp. 95-104.

[11]–, "An analytic delay model for RLC interconnects," *IEEE Trans. Comp.-Aided Design of ICs and Sys.*, vol. 16, no. 12, pp. 1507-14, Dec. 1997.

[12]A. B. Kahng, S. Muddu, N. Pol, and D. Vidhani, "Noise model for multiple segmented coupled RC interconnects," in *Proc. ISQED*, 2001, pp. 145-150.

[13]A. B. Kahng, S. Muddu, and E. Sarto, "On switch factor based analysis of coupled RC interconnects," in *Proc. DAC*, 2000, pp. 79-84.

[14]A. B. Kahng, S. Muddu, and D. Vidhani, "Noise and delay uncertainty studies for coupled RC interconnects," in *Proc. ASIC/SOC*, 1999, pp. 3-8.

[15]H. Kawaguchi and T. Sakurai, "Delay and Noise Formulas for Capacitively Coupled Distributed RC Lines", in *Proc. Asian and South Pacific DAC*, 1998, pp. 35-43.

[16]R. Kay and L. T. Pileggi, "PRIMO: Probability interpretation of moments for delay calculation," in *Proc. DAC*, 1998, pp. 463-8.

[17]T. Lin, E. Acar, and L. T. Pileggi, "h-gamma: An RC delay metric based on a gamma distribution approximation to the homogeneous response," in *Proc. ICCAD*, 1998, pp. 19-25.

[18]F. Liu, C. Kashyap, and C. J. Alpert, "A delay metric for RC circuits based on the Weibull distribution," in *Proc. ICCAD*, 2002, pp. 620-4.

[19]A. Odabasioglu, M. Celik and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Comp.-Aided Design of ICs and Sys.*, vol. 17, no. 8, pp. 645-654, Aug. 1998.

[20]L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Comp.-Aided Design of ICs and Sys.*, vol. 9, no. 4, pp. 352-366, Apr. 1990.

[21]C. L. Ratzlaff, N. Gopal and L. T. Pillage, "RICE: Rapid interconnect circuit evaluator," in *Proc. DAC*, 1991, pp. 555-60.

[22]J. Rubinstein, P. Penfield, and M. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. Comp.-Aided Design*, vol CAD-2, no. 3, pp. 202-211, Jul. 1983.

[23]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. DAC*, 1995, pp. 376-80.

[24]M. Takahashi, M. Hashimoto, and H. Onodera, "Crosstalk noise estimation for generic RC trees", in *Proc. ICCD*, 2001, pp. 110-6.

[25]X. Tong and M. Marek-Sadowska, "Efficient delay calculation in presence of crosstalk," in *Proc. ISQED*, 2000, pp. 491-7.

[26]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. DAC*, 1996, pp. 611-6.