## **RC Delay Metrics for Interconnect Optimization**

by

Min Ma

#### Department of Electrical and Computer Engineering

McGill University, Montreal



August 2004

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Engineering

© Min Ma, 2004



Library and Archives Canada

Published Heritage Branch

395 Wellington Street Ottawa ON K1A 0N4 Canada Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington Ottawa ON K1A 0N4 Canada

> Your file Votre référence ISBN: 0-494-06571-0 Our file Notre référence ISBN: 0-494-06571-0

#### NOTICE:

The author has granted a nonexclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or noncommercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.



Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

### Abstract

The main challenge for developing accurate and efficient delay *metrics* has been the prediction of delay to points on the interconnect which are relatively close to the source. Those metrics which are relatively successful in meeting this challenge require twodimensional look-up tables and algorithm tuning, and are quite challenging to implement. The simpler explicit metrics only work well on so-called far nodes, which are characterized by all-pole frequency responses.

In this thesis, we first review an existing delay metric for wires and then try to extend it to arbitrary tree networks. Thorough tests demonstrate it to be accurate and efficient for wires only. We then present an explicit delay metric for dealing with near nodes in RC interconnect, which is based on the first three moments of the impulse response. An accurate model for the delay to the internal node of a two-pole one-zero RC circuit serves as the core of the new metric. Since no simplifying assumption is made in the model, it returns excellent accuracy at the internal node in any two-node RC circuit, no matter how close the internal node is to the source. The delay at near nodes in arbitrary RC trees is then computed by order reduction to a two-pole system using the first three moments of the impulse response. A significant further improvement in accuracy is achieved by correcting for the skewness of the impulse response. In parallel, a simple explicit metric is introduced for predicting the delay to far nodes, where order reduction is not needed. This is based on the first moment of the node of interest and the second moment of the slowest node. Furthermore a simple criterion is derived for distinguishing near nodes from far nodes. Tests on RC models of wires and trees demonstrate that the combination of these two metrics is accurate within 2% for far nodes and within 5% for near nodes with delays which are as much as an order of magnitude smaller than that of the slowest node.

I

## Résumé

La difficulté majeure pour le développement de métriques de retard a été la prédiction de retards aux noeuds proches de la source du signal. Cependant, les métriques capables d'une telle performance sont basées sur des méthodes tabulaires ou sur des algorithmes qui sont généralement difficiles à implémenter. Les modèles dit explicites sont performant pour calculer les retards aux noeuds à l'extrémité opposée à la source du signal, lesquels nœuds sont caractérisés pas une fonction de transfert polaire (aucune présence de zéro).

Dans cette dissertation, nous examinons la possibilité d'étendre la validité d'un model de retard, initialement développé pour les lignes RC, au cas d'un arbre RC. Les données expérimentales confirment la validité de ce model uniquement pour les lignes RC. Nous présentons un modèle de retard explicite pour les noeuds proches de la source dans une interconnexion RC basée sur les trois premiers moments de la réponse impulsionelle du système dont l'entrée et la source du signal et la sortie est le noeud en question. Un modèle précis pour mesurer le retard dans un système à deux pôles et un zéro constitues une base pour le nouveau modèle. Etant donné que le modèle n'est basé sur aucune hypothèse quant à la relative position des pôles et du zéro sur l'axe des fréquences, une très bonne précision est obtenue pour le noeud interne d'un système 2RC. Le retard aux noeuds internes sur des réseaux RC arbitraires est alors calculé en usant des trois premiers moments de la réponse impulsionelle. Une amélioration additionnelle est apportée en corrigeant pour la déviance de la normale (skewness) de la réponse impulsionelle. Aussi, un modèle simple capable de prédire le retard aux noeuds extrêmes est présenté (situation où la réduction de l'ordre du système n'est pas requise). Celui ci est basé sur le premier moment associé au noeud en question et du deuxième moment associé au noeud le plus lent dans le réseau. En plus, nous dérivons un simple critère permettant de distinguer, électriquement, un noeud proche d'un noeud lointain. Des simulations sur des lignes RC ainsi que sur des réseaux RC démontrent que la combinaison des deux métriques est précise à hauteur de 2% pour les nœuds lointains et de 5% pour les nœuds dits proches lorsque le retard au noeud en question est, au plus, un ordre de magnitude plus bas que celui du noeud le plus lent.

Π

## Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Nicholas Rumin, who offered me an opportunity to study at McGill University. His continuous guidance, enthusiasm, patience and encouragement have been tremendous. I consider myself very lucky and most honored to have been one of his students.

I would also like to acknowledge the contribution of Mourad Oulmane, for providing important insight into interconnect analysis and delay metrics.

Further, I wish to extend my thanks to the current and former graduate students of MACS lab for their help and friendship over the past two years. To name a few, they are Barry, Eugene, Mona, John, Nisrine, Rong, Tao and Benjamin. To all I wish the best.

Financial support by the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged, as well as CAD software support by the Canadian Microelectronics Corporation.

Last but not least, my deepest personal thanks go to my family. I would like to thank my parents, my husband, my son and my brother. Without their love, patience, understanding, joy, and support, my life would be much poorer and my educational pursuits would probably never be successful. Thanks.

# **Table of Contents**

1	Introduction
1.1	Technology Scaling Implications6
1.2	Motivation and Overview of the Thesis11
1.3	Claim of Originality13
2	Review of Delay Metrics14
2.1	Interconnect and Gate Delay Models14
2.2	Circuit and Central Moments Review18
2.3	Explicit Delay Metrics
2.4	Delay Metrics Based on Probability Distribution Function25
3	Delay Metric Based on Two Moments
3.1	Delay in Wires
3.2	Extension of Metric to Arbitrary Trees
3.3	Experimental Results
3.3.	1 Wire Tests
3.3.2	2 A Simple RC Tree Test41
3.3.	3 Random Tree Tests43
4	Delay via Three Moments46
4.1	Delay to Near Nodes46
4.1.	1 2-pole 1-zero Model (2 <i>p</i> 1 <i>z</i> )46
4.1.2	2 Simplified 2-pole 1-zero Model (2p1zs)
4.1.	3 Delay to Near Nodes in Arbitrary Network
4.1.	3.1 Order Reduction
4.1.	3.2 Skewness Correction
4.2	Delay to Far Nodes
4.3	Experimental Results61
4.3.	1 Two-node Circuit Tests

4.3.2	2 Wire Tests	62
4.3.3	A Simple Tree Test	65
4.3.4	4 Random Tree Tests	66
5	Conclusions and Future Work	69
6	Appendix	71
6.1	Interconnect Order Reduction	71
6.2	Property 1 of Parameters $\alpha$ and $\beta$ in a Two-node Circuit	73
6.3	Property 2 of Parameters $\alpha$ and $\beta$ in a Two-node Circuit	74
6.4	Property 3 of Parameters $\alpha$ and $\beta$ in a Two-node Circuit	77
6.5	Proof of Condition in (4.28)	77
6.6	Node with the Maximum value of $(-m_1)$ Behaves as a Far Node	78
7	REFERENCES	80

## **List of Tables**

Table 1.1 Technology trend according to ITRS'2003[41]
Table 3.1 Average error for 100 random 20-node RC circuits    41
Table 3.2 Error distribution for 100 random 20-node RC circuits       42
Table 3.3 Comparison between D2M and new metric for a simple RC tree
Table 3.4 Average error for the sample 100-node tree    44
Table 3.5 Error distribution for the sample 100-node tree
Table 3.6 Average error for 50 random 100-node trees
Table 3.7 Error distribution for 50 random 100-note trees    45
Table 4.1 Coefficients for the rational function (4.20) and (4.21)
Table 4.2 Comparison errors due to $2p1z$ and $2p1zs$ models and D2M
Table 4.3 Average error of different delay metrics for 100 random 20-node wires 63
Table 4.4 Comparison of error for ITER and New metric without skewness correction,for the nearest 10 nodes in 100 20-node wires.64
Table 4.5 Comparison between different delay metrics for a simple RC tree       65
Table 4.6 Average error of different delay metric for 50 random 100-node trees
Table 4.7 Maximum and minimum errors of different delay metrics for 50 random 100-node trees

# List of Figures

Figure 1.1 Delay for gate, metal1 and global wiring versus feature size [41]9							
Figure 1.2 Estimated and actual wire loads [24]9							
Figure 1.3 Coupling and total capacitances versus feature size [17] 10							
Figure 2.1 (a):An inverter driving a number of similar inverters through interconnect (b): The same inverter driving a Π model of the interconnect load in (a) to compute the voltage waveform at B. (c): A RC model of the interconnect path B to C excited by the voltage transition computed in (b) to compute the interconnect delay							
Figure 2.2 (a): An inverter driving a number of similar inverters through interconnect (b): Thevenin equivalent with linear resistor and time varying voltage source replacing the inverter to calculate stage delay							
Figure 2.3 An interconnect and its RC tree model 17							
Figure 2.4 The impulse response at node $n_1$ and $n_4$ for Figure 2.3(b)							
Figure 3.1 Two-node circuit							
Figure 3.2 Five-node circuit							
Figure 3.2 Five-node circuit							
<ul> <li>Figure 3.2 Five-node circuit</li></ul>							
<ul> <li>Figure 3.2 Five-node circuit</li></ul>							
<ul> <li>Figure 3.2 Five-node circuit</li></ul>							
<ul> <li>Figure 3.2 Five-node circuit</li></ul>							
<ul> <li>Figure 3.2 Five-node circuit</li></ul>							
<ul> <li>Figure 3.2 Five-node circuit</li></ul>							

Figure 3.11 Relative error for the new metric (-) and for D2M() at near nodes
Figure 3.12 A simple RC tree 43
Figure 4.1 Two-node Circuit
Figure 4.2 Relationship between $t_D^{int}/(-m_1)$ and $\alpha$ , $\beta$ , established using two-node circuit. Dots are Hapice results while the solid lines are given by (4.20) and (4.21)
Figure 4.3 Relationship between $t_D^{int}/(-m_1)$ and $\alpha$ , $\beta$ , with $\alpha$ larger than 0.99
Figure 4.4 Relationship between $t_D^{int}/(-m_1)$ and $\alpha$ with three different $\beta$
Figure 4.5 Comparison of normalized delay computed from 2p1zs (-) and actual normalized delay ()
Figure 4.6 Relationship between <i>Err<sub>skew</sub></i> and skewness established by 100 20-node circuits
Figure 4.7 Strong linearity between $\beta$ and $t_D^{far}/(-m_1)$ for one 20-node wire

## **1** Introduction

Advances in integrated circuit technology and, in particular, the continued shrinking of the minimum feature size, have resulted during the past three decades in an exponential growth in circuit density and performance. However as technology scales further down into the deep submicron (DSM) region, system performance is being increasingly determined by the interconnect, not by the transistors. As a result, a lot of research is going into the development of techniques for the modelling and analysis of interconnect.

We present below a brief review of the problems which are being created by the continuing scaling of integrated circuit (IC) feature sizes and die areas. In particular, we examine their impact of interconnect on system performance, and the effect it is having on electrical design automation (EDA) tools. We conclude this chapter with a motivation for the present work, and an outline of the thesis.

#### **1.1 Technology Scaling Implications**

Over the past decades, the semiconductor industry has observed phenomenal advances in the integrated circuit (IC) technology. The key feature of the CMOS IC technology is that the feature sizes have continued to shrink at a rate such that the resulting circuit complexity has followed Moore's law of exponential growth, namely the doubling of circuit density every one or two years. Such a trend may continue for at least another 10-12 years according to the predictions of the 2003 International Technology Roadmap for Semiconductors, shown in Table 1.1 [41]. It is expected that by the year 2016, over eight billion transistors will be integrated on a single chip with chip-to-board speed of 30-40 GHz in a 22 nm technology.

Process Technology Node(nm)	100	90	65	45	32	22
DRAM 1/2 Pitch						
Year of Production	2003	2004	2007	2010	2013	2016
Million transistor number /chip	439	553	1106	2212	4424	8848
Chip-to-board speed (MHz)	2000	2500	4883	9536	18626	36379
V <sub>dd</sub> (high performance)	1.2	1.2	1.1	1.0	0.9	0.8
Number of metal levels	9	10	11	12	12	14
Metal1 wiring pitch (nm)	240	214	152	108	76	54
Minimum global wiring pitch (nm)	475	410	290	205	140	100
Metal1 aspect ratio(Cu)	1.6	1.7	1.7	1.8	1.8	2.0

Table 1.1 Technology trend according to ITRS'2003[41]

The technology feature size scaling presents many fundamental challenges to today's electrical design automation (EDA) tools and design flow. One of the biggest changes that is taking place is that the dominant factor in system performance has shifted from the devices to the interconnect, due to the rising global interconnect delays [11] [31]. For classical transistor scaling, device delay decreases since gate length, gate dielectric thickness, and junction depth are all scaled. The impact of this scaling on the interconnect is that resistance per unit length grows due to the fact that the width and thickness both scale down, although the thickness may scale at a slower rate to prevent the resistance from increasing rapidly [35]. Capacitance per unit length is roughly constant or decreases very slowly. The effect on the interconnect performance differs, depending on the length of the wires. Local wires, which communicate gates or cells locally within blocks, scale down in length with the transistors, so they show delays that in general roughly track the gate delays or grow slowly relative to the gate delays. On the other hand, the global wires such as clock, supply and buses, tend to increase due to the tendency to group more transistors into larger system clocks, and to the growth in the die size. Therefore the global interconnect delay, which is proportional to the square of wire length, grows with technology scaling. Although the global delay can be greatly reduced by using optimization techniques, such as buffer insertion, buffer and wire sizing, optimized global interconnect delays at best remain roughly unchanged [7] [19], which means that, the gap between large interconnect delays and small gate delays is increasing. Figure 1.1 presents delay comparisons among gates and the interconnect, where the parameters of the interconnects and gates are collected from ITRS'2003 [41]. As can been seen, the intrinsic gate delay decreases rapidly from the 180nm to the 32nm technology. The metal1 (local interconnect) delay decreases with the gate delay until 65nm, and then grows slowly compared to the gate delay, while the optimized global delay increases slowly as the technology scales and results in a trend that the global interconnect delay is far beyond the gate delay. Therefore, the system performance is increasingly defined by the global interconnect with each new technology generation. Note that the interconnect and gate delays in Figure 1.1 already include the effect of using copper and low-k materials. Although these new materials are helpful in improving the interconnect performance, using new materials alone is not sufficient to provide a satisfactory solution to the increasing gap between devices and interconnect delay performances [7] [41]. Furthermore, the proportion of global interconnects is increasing dramatically with the size and complexity of ICs. It is the widening delay mismatch between the global interconnects and gates, and the increasing total number of the global interconnects on a chip that result in fundamental problems and difficulties in current IC design methodologies and EDA tools [19] [33].

In a conventional design flow, the synthesis/design of a module occurs prior to the physical layout. The interconnect loads are modelled based on their fan-outs, known as fan-out-based wire model, which is determined by the statistical analysis of past designs in the standard cell library. Due to simplifying interconnect models and ignoring their actual distributed characteristics, such as resistances and capacitances, these wire models can generate large errors. This is illustrated in Figure 1.2 for a small design, where nets are sorted by fanout and post-layout wire load capacitance [24]. As can be seen, wire loads for a particular fanout can vary in a large range of values. The discrepancy between the synthesis estimate and post-placement may be acceptable for short and medium length wires, but the significant underestimation by the synthesis estimate for long wires makes it hard for EDA tools convergence to a final solution that meets timing

constrains.



Figure 1.1 Delay for gate, metal1 and global wiring versus feature size [41]



Figure 1.2 Estimated and actual wire loads [24]

However it is worth noting that in [40], Sylvester and Keutzer investigated the behaviour of the average interconnect, and found that by adequate gates sizing, the average interconnect delay maintains a small part in the total delay with technology scaling. Therefore they concluded that current EDA tools are able to handle future 50K design.

While their analysis applies to the average interconnect, they ignore the fact that most timing problems arise due to the long global wires. Furthermore, as the technology scales down to the next generation, the overall transistor numbers and the complexity of a same-sized die grows exponentially, and therefore we have to deal with much larger and more complicated systems than typically 50K gates. According to ITRS'2003, by the year 2010, the transistor number in one chip can reach 2G, and thus the number and the length of long global wires will increase as well. Current EDA tools and design flow have to be improved or changed to handle these exceptional long wires [20] [24].

Another effect associated with interconnect scaling is the rise of coupling capacitance between adjacent lines. The rise of wire aspect ratio, needed to keep the resistance low, and the decrease of the line spacing, result in an increase of the coupling capacitance, which may reach 70% of the total capacitance by the year 2007 [7] [9] [17] [19]. The coupling capacitance ( $C_{L-L}$ ) and total capacitance ( $C_{Total}$ ) versus feature size are shown in Figure 1.3 [17].



Figure 1.3 Coupling and total capacitances versus feature size [17]

This large coupling capacitance produces two main problems in addition to increasing the interconnect delay directly. One is the delay deterioration, which occurs due to the fact that interconnect delay is no longer a constant value, but depends strongly on its neighbouring signal activities. Another problem is the crosstalk noise, which refers to the fact that the switching of one net, called aggressor net, may induce an undesirable voltage spike or voltage overshoot effect on the neighbouring nets, called victim nets. This undesirable spike may propagate through the circuits and, therefore cause functional signal integrity failures. Crosstalk noise is a larger problem for dynamic circuits in digital systems design, with their two phases of operation, than for static ones. Therefore, the rise of coupling capacitance has a negative impact on the circuit performance and even causes circuit malfunction. It is through the development of new insulating materials such as low-k dielectrics, low resistivity materials, such as copper, and associated processes, design and package innovations, that practical solutions to these noise problems are being found.

The effects of global interconnect delays and coupling capacitances have made interconnect analysis and design one of the most important and challenging problems for high performance IC designs in current and future technology generations. Given the problems in a conventional design flow, which put much concentration on device and logic, a new design methodology, called interconnect-centric design methodology, has been explored in [7] [30]. Here interconnect planning and optimization are considered and emphasized throughout the whole design process, which consists of three major phases: interconnect planning, interconnect synthesis, and interconnect layout. The shift in methodology from device-centric to interconnect-centric provides an efficient way to handle the design complexity in large VLSI systems [7].

#### **1.2 Motivation and Overview of the Thesis**

The design optimization of an integrated circuit (IC) requires millions of delay calculation, especially in deep sub-micron (DSM) technologies, where interconnect delay is often a stage delay determinant. Due to the nature, size and number of interconnects, it is impractical to use the SPICE simulation tool, where every output variable has to be calculated at each time step and then billions of nonlinear equations have to be solved. Therefore, significant approaches in the area of delay estimation have been proposed which require much less computation time.

One popular technique is to model the interconnect by RC or RLC networks and use linear order reduction methods to reduce the original model with a large number of poles

and zeros to a system with a small number of dominant poles. The reduced order system is then used to approximate the response of the original one. Such techniques, for example Asymptotic Waveform Evaluation (AWE) [36], can achieve 10,000 faster than SPICE simulation for the transient analysis of large RLC trees [38], with good accuracy. They provide an efficient and useful way to analyse post layout circuits, where high accuracy is a key requirement. However one of the problems associated with AWE in terms of delay estimation is that it does not produce an explicit relationship between the delay and the interconnect parameters for optimization [34]. Thus, iterations have to be used to solve the transcendental equations for delay estimation. This problem makes order reduction methods inefficient for the early stage of the design flow, where delay estimation is often part of the inner loop of the design optimization algorithms.

In such a case, simple close-form delay expressions, referred to as delay *metrics*, are useful. Furthermore, high accuracy is often not required since the complete information (e.g. detailed routing) is not available for many optimization techniques [6]. These fast and conservative interconnect delay metrics, having reasonable accuracy, are critical to assure the proper and efficient coupling between synthesis and layout, and thus enable the design to converge. The simplicity of delay metrics leads to an efficiency of several orders of magnitude faster than order reduction methods, with only a modest loss in accuracy.

The main challenge in the design of delay metrics has been the prediction of delay to points on the interconnects which are relatively close to the source, called *near* nodes. Those metrics [26] [27] [28] which are relative successful in meeting this challenge require two-dimensional look-up tables and algorithm tuning, and are quite challenging to implement. The simpler explicit metrics[2] [3] [14] [23] [42] only work well on so called *far* nodes, which are characterized by transfer functions having no low frequency zeros. Therefore, there is still a need for more work on delay metrics with good accuracy and efficiency [8].

This thesis concentrates on delay metrics, which can compute delay with good accuracy and high efficiency at far nodes as well as near nodes. It consists of three main chapters. Chapter 2 reviews some of the relevant delay metrics that have been reported.

They are loosely classified into two classes, based on the trade-off between efficiency and accuracy. One class consists of explicit metrics that aim to achieve very high efficiency while the other class involves metrics based on the probability density function, which trade efficiency for accuracy. In Chapter 3, we review a recently reported metric based on two moments [32], and extend it to general trees. The metric can return excellent accuracy both at far node and near nodes of RC models of wires. However, larger errors result at near nodes for RC trees. Given the problems in chapter 3, the core of the present research appears in Chapter 4, where a new explicit metric for predicting the delay to near nodes in RC trees with both efficiency and accuracy is described. It is based on an accurate delay model for the internal node of a 2-pole 1-zero circuit. The delay to *near* nodes in an arbitrary tree network is then developed by order reduction to a 2- pole 1-zero system using the first three moments of the impulse response. A significant further improvement in accuracy is achieved by correcting for the skewness of the impulse response. In parallel, a new and simple, yet accurate delay metric for far nodes is also introduced. The effectiveness and accuracy of the combination of these two metrics are demonstrated on random wires and trees. Finally, conclusions and comments are made in Chapter 5.

#### **1.3 Claim of Originality**

The following results described in this thesis are, to the best of my knowledge, the author's original contributions:

- The development of the 2-pole 1-zero model (2p1z), which predicts the delay to the internal node of a two-node RC circuit, and on which the near node metric, described in Chapter 4, is based.
- The use of the skewness of the impulse response to improve the accuracy of the near node metric.
- The far node metric described in Chapter 4.
- The derivation of the criterion for distinguishing between far and near nodes, from the 2-node RC circuit in Chapter 4.

## **2** Review of Delay Metrics

This chapter is mainly devoted to a review of the techniques that have been reported to evaluate delays in an efficient and accurate manner. We start by outlining gate and interconnect modelling techniques (Section 2.1), followed by a brief review of circuit and central moments (Section 2.2), on which many delay metrics are based. In Section 2.3 and Section 2.4, two classes of existing delay metrics are reviewed: explicit ones and ones based on a probability density function.

In this thesis, delay is defined as the time for the output transient to complete 50% of its transition in response to an ideal step input. The metrics we consider are based on RC models. Despite the growing importance of inductive effects in interconnect networks, metrics based on RC models of the interconnects still appear to be used widely.

#### 2.1 Interconnect and Gate Delay Models

Interconnects are often modelled as RC trees by dividing each long wire into carefully selected number of segments and modelling each wire segment as a  $\Gamma$ -type or a  $\Pi$ -type of RC circuit with a capacitor from each node to ground, no coupling capacitors, no resistor loops and no resistor from a node to ground [36].

In general, we are interested in computing the stage delay from the input gate to one of its interconnect load sinks. For example, considering a simple network in Figure 2.1 (a), to compute the stage delay from node A to C, there are two common approaches to model gate and interconnect delays. One way is to combine the interconnect model with a pre-characterized gate model. The stage delay, which is the total delay due to the gate and the interconnect, is determined by computing the gate and the interconnect delays separately. First, the delay through the first gate (AB) is found by using a reduced model of the loading effect of the interconnect, such as a  $\Pi$ -circuit (C<sub>1</sub>', R<sub>π</sub>, C<sub>2</sub>') as shown in Figure 2.1 (b). The gate output parameters, delay and output transition time, are determined from an empirical look-up table, which characterizes the relationship between the input and output gate parameters. Then the output waveform of B is approximated by a saturated ramp according to the output gate parameters. This calculated waveform is used as the input excitation to the RC interconnect network, path B to C, to compute the interconnect delay as shown in Figure 2.1 (c).



Figure 2.1 (a):An inverter driving a number of similar inverters through interconnect(b): The same inverter driving a II model of the interconnect load in (a) to compute the voltage waveform at B. (c): A RC model of the interconnect path B to C excited by the voltage transition computed in (b) to compute the interconnect delay





Another common way to model the gate delay is to use a Thevenin equivalent comprised of a constant linear resistor and a saturated ramp voltage source, as shown in [12] [13]. To determine the parameters of the Thevenin voltage source waveform, which are characterized by a transition time and a delay, the load interconnect is modelled as an effective capacitance  $C_L$ . The concept of an effective capacitance was proposed in [37] to capture the effect of resistance shielding of the interconnect. Then the parameters are calculated by solving transcendental equations, resulting from matching with the 20% and 50% point of the actual gate response, as a function of the effective capacitance. The value of the resistance in the Thevenin equivalent can be kept roughly constant for

different interconnect load capacitances due to the fact that using a larger or a smaller value of the resistance has little impact on the gate delay approximation [12]. Given the equivalent gate model, the stage delay is computed by connecting the Thevenin model to the interconnect model, as shown in Figure 2.2 (b), where the interaction between the gate and interconnect can be easily captured.





Figure 2.3 An interconnect and its RC tree model

Although the voltage source in the gate delay model is often characterized by a

saturated ramp input, to simplify the problem, in the following chapters, we are dealing with estimating the delays of RC interconnect trees driven by an ideal step input as the one shown in Figure 2.3. The delay expression for a ramp input can be easily found from the delay metric for a step input [25].

#### 2.2 Circuit and Central Moments Review

Since the metrics we are dealing with rely entirely on the concept of circuit moments or cental moments, it is necessary to briefly outline the definition of circuit moments, central moments and explain how they can be computed efficiently in RC or (*RLC*) tree networks.

Assume H(s) is the Laplace transform of a circuit's impulse response h(t), namely its transfer function, then:

$$H(s) = \int_{0}^{\infty} h(t)e^{-st}dt$$
(2.1)

Taking a Taylor series expansion of  $e^{-st}$  about s=0 yields:

$$H(s) = \int_{0}^{\infty} h(t) \left\{ 1 - st + \frac{1}{2!} s^{2} t^{2} - \frac{1}{3!} s^{3} t^{3} + \cdots \right\} dt$$
$$= \sum_{i=0}^{\infty} \frac{(-1)^{i}}{i!} s^{i} \int_{0}^{\infty} t^{i} h(t) dt$$
(2.2)

The *i*-th circuit moment  $m_i$  is defined as the *i*-th coefficient of (2.2), that is,

$$m_{i} = \frac{(-1)^{i}}{i!} \int_{0}^{\infty} t^{i} h(t) dt$$
 (2.3)

Note that  $\int_{0}^{\infty} t^{i} h(t) dt$  is the *i*-th moment of the circuit's impulse response h(t), denoted as

 $\tilde{m}_i$ . Thus the *i*-th circuit moments,  $m_i$ , is related to the *i*-th moment of the impulse response  $\tilde{m}_i$ , by the  $(-1)^i/i!$  term.

$$m_i = \frac{(-1)^i}{i!} \tilde{m}_i \tag{2.4}$$

The circuit moments,  $m_i$ , are generally simply referred to as moments. It follows that the transfer function H(s) can be written as:

$$H(s) = m_0 + m_1 s + m_2 s^2 + m_3 s^3 + m_4 s^4 + \dots$$
(2.5)

In probability distribution-based delay metrics, central moments, instead of circuit moments, are often used. The *i*<sup>th</sup> central moment,  $\mu_i$ , is defined by using the mean of the impulse response as the origin, and has the form of:

$$\mu_{i} = \int_{0}^{\infty} (t - \mu)^{i} h(t) dt$$
(2.6)

where  $\mu$  is the mean of the impulse response and is given by [10]:

$$\mu = \frac{\int_{-\infty}^{\infty} th(t) dt}{\int_{0}^{\infty} h(t) dt}$$
(2.7)

Since the area of the impulse response for RC tree networks is unity, that is:

$$\int_{0}^{\infty} h(t) dt = 1$$
(2.8)

it follows that the mean of the impulse response is equal to the absolute value of the first moment,  $m_1$ ,

$$\mu = \widetilde{m}_1 = -m_1 \tag{2.9}$$

From the definition of circuit moments and central moments, it is easy to show that for RC tree networks, the first few central moments can be calculated from the circuit moments as follows [5]:

$$\mu_{0} = m_{0}$$

$$\mu_{1} = 0$$

$$\mu_{2} = 2m_{2} - m_{1}^{2}$$

$$\mu_{3} = -6m_{3} + 6m_{1}m_{2} - 2m_{1}^{3}$$
(2.10)

The authors of [15] have demonstrated that the second and the third central moments are non-negative for RC tree networks. The second central moment  $\mu_2$  is the variance of the distribution. The third central moment  $\mu_3$  reflects the degree of asymmetry [27]. It is

the geometrical interpretations, such as variance and asymmetry, in the central moments that provide a straightforward way to find a distribution function whose behaviour is similar to the one of the impulse response of RC trees.

Moments can be easily computed for RC (or RLC) trees. This is performed by a limited number of successive DC analyses of the circuit, where each node voltage (capacitor voltage) corresponds to the moments associated with such a node. Furthermore, since the capacitor current is known when we compute the moments, each capacitor in the RC tree is replaced by a DC current source set to its capacitor current (or each inductor is replaced by a DC voltage source for the RLC tree). The computation process starts by replacing input voltage source with 1 volt, the capacitors with current sources set to zero. Then  $m_0$  at each node is determined by solving for its corresponding node voltage in such an equivalent DC circuit. It is obvious that  $m_0$  at any node is equal to 1 for RC circuits. All subsequent moments are computed in the DC equivalent circuits, where the input voltage source is set to zero and each capacitor is replaced with a current source, having the current equal to the product of its previous moment and respective value of capacitance [38].

It is obvious that the analysis of DC equivalent circuits is essential for moment computation. One efficient way to do it for a tree topology is by path-tracing, as presented in RICE [38]. The basic path-tracing involves one complete backward traversal of circuits to generate all branch currents, and another forward traversal to compute all node voltages. In the real implementation of moments calculation, only a one time tracing of the actual path is needed, regardless of the number of moments, because of the similar path-tracing process during each moment generation. For example, recursive computation of the moments at node  $n_2$  via path tracing in Figure 2.3(b) yields:

$$m_0 = 1$$
  

$$-m_{1 - n1} = R_1(C_1 + C_2 + C_3 + C_4 + C_5)$$
  

$$-m_{1 - n2} = R_1(C_1 + C_2 + C_3 + C_4 + C_5) + R_2(C_2 + C_3 + C_4 + C_5)$$
  

$$-m_{1 - n3} = R_1(C_1 + C_2 + C_3 + C_4 + C_5) + R_2(C_2 + C_3 + C_4 + C_5) + R_3(C_3 + C_4)$$
  

$$-m_{1 - n4} = R_1(C_1 + C_2 + C_3 + C_4 + C_5) + R_2(C_2 + C_3 + C_4 + C_5) + R_3(C_3 + C_4) + R_4C_4$$
  

$$-m_{1 - n5} = R_1(C_1 + C_2 + C_3 + C_4 + C_5) + R_2(C_2 + C_3 + C_4 + C_5) + R_5C_5$$

$$m_{2-n2} = R_1(C_1 m_{1-n1} + C_2 m_{1-n2} + C_3 m_{1-n3} + C_4 m_{1-n4} + C_5 m_{1-n5}) + R_2(C_2 m_{1-n2} + C_3 m_{1-n3} + C_4 m_{1-n4} + C_5 m_{1-n5})$$
(2.11)

where  $m_{1-n1}$ ,  $m_{1-n2}$ ,  $m_{1-n3}$ ,  $m_{1-n4}$  denotes the first moment at node  $n_1$ ,  $n_2$ ,  $n_3$ , and  $n_4$  respectively,  $m_{2-n2}$  refers to the second moment at node  $n_2$ , and so on.

Path tracing is very efficient for tree-like structure networks. However for non-tree like networks, it may lose its efficiency. In such a case, moments can be obtained by using MNA (Modified Nodal Analysis) [18], which is more general and can be fairly efficient when taking advantage of sparse matrix techniques and special ordering algorithms [5]. It is estimated that the transient analysis by using moments can be 10,000 faster than Hspice for large RLC trees [4] [38]. Because of the ease and high efficiency in the computation of moments for RC (or RLC) trees, the delay metrics based on moments become very useful for the early stages of the design flow.

#### 2.3 Explicit Delay Metrics

The most desirable delay metrics in terms of efficiency are in explicit form. The Elmore delay [14] is the simplest explicit delay metric even though Elmore arrived at it from a probability distribution point of view. He was interested in circuits with monotonic step response, such as the RC model of interconnect trees. He noted that the impulse response of such circuits is a non-negative function and hence, can be seen as a probability density function. Since the step response is the integral of the impulse response, therefore the 50% delay of the monotonic step response is equal to the median of the impulse response. It is not easy to compute the median of the impulse response, so Elmore proposed to approximate it by the mean of the impulse response, which is the first moment [14]:

$$t_D = ED = \int_0^\infty th(t) dt, \qquad (2.12)$$

where ED denotes the Elmore delay.

Penfield and Rubinstein [39] proved that the step response for an RC tree is monotonic, and demonstrated that the Elmore delay for a node i is given by:

$$t_{D_i} = \sum_{k=1}^{n} R_{ki} C_k , \qquad (2.13)$$

where  $R_{ki}$  is the common resistance between the path from the source to node *i* and the path from the source to node *k*, and  $C_k$  is the capacitance at node *k* [39]. The Elmore delay has been popular because it is a simple, explicit delay approximation and can be computed efficiently in two O(N) traversals of the RC tree.

The main problem with the Elmore delay is that the mean and median are equal only if the response is symmetrical, and in general it is not the case in RC trees. This is illustrated in Figure 2.4, where the impulse response of node  $n_1$  and  $n_4$  for the circuit in Figure 2.3(b) are displayed. It is obvious that the asymmetry of the response at the nearest node  $n_1$  is greater than that at node  $n_4$ , which is farthest from the source. Hence the Elmore formula can not be expected to yield an accurate result for the near node  $n_1$  in particular.



Figure 2.4 The impulse response at node  $n_1$  and  $n_4$  for Figure 2.3(b)

The Elmore delay has been shown to be the first moment of the impulse response [14], which leads to another explicit delay metric; the 'one dominant pole metric'. Assume one of the poles in an RC tree is dominant and there are no low frequency zeros. Then the 50% delay is found to be the Elmore delay in (2.13) scaled by a factor of ln(2), which is often called the scaled Elmore [15] [34]. It has been shown from experiments that the scaled Elmore delay improves the delay estimation under step excitation. However it does not change the relative delay error but simply shifts all the delays. It can be shown that if the input rise time is much larger than the dominant time constant, then the Elmore delay in (2.13) is an accurate estimate of the 50% delay [34].

Kahng and Muddu [23] presented another single dominant pole metric, which involves computing the dominant pole based on matching the 3dB frequency of a twopole transfer function to that of a one-pole one. Assume  $P_1$  and  $P_2$  are the two poles, P is the approximate single pole. Matching the 3dB frequency of the two-pole circuit to the 3dB frequency of the one-pole circuit results in:

$$\frac{1}{P} = -\sqrt{\frac{1}{P_1^2} + \frac{1}{P_2^2}}$$
(2.14)

Note that the transfer function of the two-pole system can be expressed in terms of the circuit moments as:

$$H(s) = \frac{1}{(m_1^2 - m_2)s^2 - m_1s + 1}$$
(2.15)

From (2.14) and (2.15), the delay can be computed by:

$$t_D = \sqrt{2m_2 - m_1^2} \ln(2) \tag{2.16}$$

The main problem with the one dominant pole delay approximation is the difficulty of fitting a two-time constant response with a single time constant response. To improve the accuracy, several two dominant poles models [1] [21] [42] have been proposed. However due to the two exponential terms in the time response expression, it is hard to obtain a closed form delay expression for the two-pole system. A lookup table is one way to avoid nonlinear iterations. To simplify the problem, the authors of [42] assumed that the contribution of the exponential term corresponding to the dominant pole is the dominant factor in the delay, which results in the first order approximation:

$$t_1 = \frac{1}{-p_1} ln(\frac{2r_1}{-p_1}), \qquad (2.17)$$

where  $p_1$  is the dominant pole and  $r_1$  is its corresponding residue <sup>1</sup>. Then a single Newton-Raphson iteration (SNRI) is implemented, using  $t_1$  as the initial guess, which yields an explicit delay expression:

$$t_D = t_1 + \frac{-0.5 + \frac{r_1}{-P_1} e^{P_1 t_1} + \frac{r_2}{-P_2} e^{P_2 t_1}}{r_1 e^{P_1 t_1} + r_2 e^{P_2 t_1}}$$
(2.18)

The SNRI is a very accurate delay metric for those nodes, which are far from source, such as some sink nodes. However, due to the assumption made in (2.17), as the nodes approach the source, the error increases, and when the nodes are close to the source, SNRI frequently returns a negative delay value.

Recently, Alpert el al. [2] proposed an empirical delay metric D2M based on the first two moments. Their goal was to find a metric which is as simple as Elmore, but significantly more accurate. This led them to utilize the Scaled Elmore delay formula in the following way:

$$D2M = -rm_1 ln(2) \tag{2.19}$$

From a large number of experiments, they found that when r is equal to  $\frac{m_1}{\sqrt{m_2}}$ , equation

(2.19) gave the least error. They also found that the relative error at the output nodes for more than 100 one-sink RC circuits was almost always within 2% of the actual delay. Otherwise the metric exhibited Elmore-like behaviour, with the overestimation of the delay to internal nodes increasing with decreasing distance from the source. Similarly in the case of RC trees, D2M performed well only on nodes that were relatively far from the source. The authors of [32] showed that D2M, in fact, is exactly the 50% delay to the

<sup>&</sup>lt;sup>1</sup> The computation of poles,  $p_1$ ,  $p_2$ , and corresponding residues,  $r_1$ ,  $r_2$ , based on the first three moments, is presented in Appendix 6.1.

output of a two-pole RC circuit having one dominant pole, and that the metric remains valid to within 5% error for any ratio of the two poles. They also showed that D2M can be extended to delays other than the 50%.

In the same paper, the authors of D2M [2] present another metric, ECM. It is not based on moments, but has a similar form and structure as the Elmore delay. They take into account the resistive screening of the downstream capacitance, for example  $C_2$  in the circuit in Figure 3.1, by modifying the Elmore delay at node *int* to the form:

$$ECM = R_1(C_1 + kC_2) \tag{2.20}$$

where  $0 \le k \le 1$  is a parameter that captures, in the case of Figure 3.1, the resistive shielding by  $R_2$  of the contribution of  $C_2$  to the delay to node *int*. Although the computation of k is complicated and involves the determination of an effective capacitance from a  $\Pi$  model, ECM is significantly less accurate than D2M according to their tests.

#### 2.4 Delay Metrics Based on Probability Distribution Function

The general idea of delay metrics based on PDF (Probability Distribution Function) method is to extend Elmore's distribution interpretation by matching the probability properties of a representative distribution function with those of the impulse response, which are characterized by central moments. Once the representative function is found, it is treated as the impulse response and the delay can be computed from the median value of the distribution family.

It was shown that one good approximation of the impulse response is the gamma distribution function [26]. The probability density function  $g_{\lambda,n}(t)$  of the gamma distribution is given by:

$$g_{\lambda,n}(t) = \frac{\lambda^n t^{n-1} e^{-\lambda t}}{\Gamma(n)}, \ \Gamma(x) = \int_0^\infty y^{x-1} e^{-y} dy.$$
(2.21)

To better capture the different shapes of the impulse response for different nodes in RC trees, a third parameter  $\Delta$  is added to shift the function. The shifted gamma function is used to approximate the impulse response, which results in:

$$h(t) = g_{\lambda,n} \left( t - \Delta \right) \tag{2.22}$$

The Laplace transform of equation (2.22) is given by:

$$H(s) = e^{-s\Delta} \left(\frac{\lambda}{\lambda+s}\right)^n \tag{2.23}$$

Since the second and third central moments are independent of the shift parameter  $\Delta$ , the first moment, the second central and the third central moments of the gamma function are used to match those of the impulse response respectively, which results in:

$$\lambda = \frac{2\mu_2}{\mu_3}, \ n = \frac{4(\mu_2)^3}{(\mu_3)^2}, \ \Delta = m_1 - \frac{n}{\lambda}$$
 (2.24)

Having three parameters, the step response of the system can be approximated by:

$$y(t - \Delta) = \int_{0}^{t} g_{\lambda,n}(\tau) d\tau$$
(2.25)

Consequently 50% delay  $t_D$  can be found by:

$$y(t_D) = 0.5$$
 (2.26)

The above equation is solved by means of numerical iteration. In order to avoid the complex iteration, pre-compiled tables are suggested to compute the delay. Due to the shift in (2.25), PRIMO [26] returns negative delays for some near nodes, which are unrealistic.

Instead of fitting the impulse response with the shifted gamma distribution, Hgamma [27] uses the same gamma function as in PRIMO to fit the normalized homogenous response, which can be proven to be a probability density function. In fact, H-gamma is an extension of PRIMO. It can achieve more accuracy than PRIMO, and it does not suffer the problem of near nodes as in PRIMO. Its implementation, however, is even more challenging, with a complex precompiled 2-dimension table.

Another possible distribution function is the Weibull function [28] [29] (WED). In this case only two one-dimensional tables are needed, however WED is not as accurate as H-gamma [27].

A lognormal delay metric (LND) was proposed [3] for applying the same PDF matching principle to the lognormal distribution. The probability density function p(t) of the lognormal distribution is given by:

$$p(t) = \frac{1}{tS\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{\ln t - M}{S})^2}$$
(2.27)

where the scale parameter M>0 and the shape parameter S>0. Using a similar matching process as the previous delay metrics, the two parameters M and S can be determined by two moments, which have the forms:

$$M = \ln(m_1^2 / \sqrt{2m_2}), \ S = \sqrt{\ln(2m_2 / m_1^2)}$$
(2.28)

Note that the median of the lognormal distribution, which is the 50% delay, is equal to  $e^{M}$ . In other words, the implementation of the lognormal metric is very easy. It overcomes the problem with previous PDF metrics, which require solving a transcendental equation for the delay. Therefore it can be simply computed by an explicit form:

$$t_d = e^M = e^{\ln(m_1^2 / \sqrt{2m_2})} = m_1^2 / \sqrt{2m_2}$$
(2.29)

As can be seen, this metric (LND) [3] is basically similar to D2M [2], with a slightly different scale factor. It would be expected that they would return similar results for delay estimation.

It is worth noting that in the same paper, the authors of [3] also presented a 10/90 slew metric, which is obtained by matching the mean, variance and skewness of the lognormal function to those of the impulse response and then computing the 10% and 90% delay points.

From this review of the past work on delay metrics, it can be seen that the main challenge for delay estimation is the prediction of delay to points on the interconnect which are relatively close to the source, where zeros play an important role in the transfer function, which are the source of the asymmetry of the impulse response. Those metrics which are relatively successful, such as PRIMO [26], H-gamma [27], require some look-up tables and algorithm tuning, and are quite challenging to implement. It has been shown that H-gamma is by far the most accurate metric [2] [3] [29], especially for near nodes, however, it can significantly underestimate the delays to the very near nodes by as much as 75%, according to the RC tree tests in [2]. The simpler metrics, such as D2M

[2], LND [3], and SNRI [42], only work well on nodes which are far away from the source, where poles are dominant and, therefore the impulse response is relatively symmetric.

## **3 Delay Metric Based on Two Moments**

An interesting delay metric for RC single-sink wires, based on a semi-empirical function of two moments, was described in [32]. Tests on some small RC models of wires demonstrated quite good accuracy even for nodes relatively close to source. In this chapter, we first review this metric and present more thorough test results for wires. We then suggest an extension of this metric to RC trees and present test results on randomly generated RC tree models. In section 3.1, the delay metric used in RC wires is reviewed. The method of extending this metric to RC trees is presented in Section 3.2. Finally in Section 3.3, we describe thorough test results on randomly generated RC wires and trees.

#### 3.1 Delay in Wires

We begin by considering two extreme cases in a simple two-node circuit. Delays in general wires are then computed by an extension of the delay expression for the two-node circuit. Consider the simple two-node circuit in Figure 3.1. The scaled Elmore delay to the internal node *int* is:

$$t_D = R_I(C_1 + C_2) ln(2) \tag{3.1}$$

The accuracy of (3.1) depends on the value of resistance  $R_2$ . When  $R_2$  is zero, the delay in (3.1) is accurate. However when  $R_2$  goes to infinity, it totally "screens" the effect of  $C_2$  on the delay, in which case, the delay to *int* tends to be:



Figure 3.1 Two-node circuit

$$t_D = R_1 C_1 ln(2) \tag{3.2}$$

In the general case,  $C_2$  is partly screened by  $R_2$ , and therefore the delay to the internal node should be larger than (3.2) and smaller than (3.1). This can be viewed as a smaller contribution of the second term in (3.1). Hence it is reasonable to express the delay to node *int* in the form of:

$$t_D = R_I (C_1 + BC_2) ln(2), \tag{3.3}$$

where B is in the range of 0 to 1, and it is a parameter that captures the screening by  $R_2$  of the contribution of  $C_2$  to the delay at the node *int*.

As can be seen, the delay in (3.3) is determined by two terms. The first one, which is  $R_1C_1ln(2)$ , is the delay to *int* overlooking the downstream element,  $R_2$ . The second term captures the additional delay due to  $C_2$ . Thus, for a higher order wire model than Figure 3.1, the first term in (3.3),  $R_1C_1$ , is replaced by the delay to the node of interest, ignoring all the downstream capacitances. For the second term,  $R_1C_2$ , is replaced by the sum of all the capacitances downstream from the node of interest *int*,  $C_{d,int}$ , multiplied by the sum of all resistances upstream of *int*,  $R_{u,int}$ . Since the first term represents the delay through the upstream network to an unloaded node *int*, this delay can be computed using the D2M metric [2], which is accurate for output (unloaded) nodes. Hence for general wire nodes, (3.3) can be replaced by:

$$t_d^{int} = \left(\frac{m_1^2}{\sqrt{m_2}}\right)_{int,ind} + BR_{u,int}C_{d,int} \left(2\right), \qquad (3.4)$$

where the D2M delay is expressed using (2.19) with  $r = \frac{m_1}{\sqrt{m_2}}$ . *B* is the screening factor

and roughly in the range of 0 to  $1^2$ . The index, *int*, in the first term on the right hand side indicates the node with which the moments are associated, while *ind* denotes that the moments are computed from the circuit which is independent of the loading of the node *int* by the downstream network. In other words, the node *int* is treated as the end of a wire, which is obtained by ignoring all downstream elements of the original circuit.

<sup>&</sup>lt;sup>2</sup> B can be slightly larger than 1 because of the scaling factor ln(2) in the equation.

Obviously, how to compute the screening factor B is the essential problem in (3.4). One approach is to express B as an exponential function of the ratio of the first moment at node *int* and the time constant of the right hand side of the  $\Pi$  model. This approach is used in the ECM metric [2], where the computation of B is very complicated and involves determining the  $\Pi$  model of the part of the interconnect downstream from node *int*. The  $\Pi$  model is then replaced by its effective capacitance, which is found by matching the charge to the  $\Pi$  model with the charge to the effective capacitance. However first moments are not sufficient to capture signal transitions under the condition where the screening effect is important [16]. Therefore, a formulation of B based on the first two moments is more appropriate.

A semi-empirical formulation to compute B was proposed in [32] through a function  $f_x(n)$ , which is defined as:

$$f_{X}(n) = n \left( \frac{m_{I}^{2}}{\sqrt[n]{|m_{n}|}} \right)_{X}$$
(3.5)

The function  $f_x(n)$  is very close to being linear with respect to the moment index n. This is illustrated in Figure 3.3 by using a five-node uniform wire in Figure 3.2. Furthermore, the slope of the function monotonically increases with the node distance x to the source. Hence, it appears reasonable to express the screen parameter B at node x in terms of the slope of  $f_x(n)$  normalized by the slope of  $f_{out}(n)$ , where  $f_{out}(n)$  refers to the function at output node *out*, which is the farthest in the circuit, e.g. *out=5* in Figure 3.2. Therefore B is in the form of:



Figure 3.2 Five-node circuit


Figure 3.3 Linear relation between  $f_x(n)$  and the moment index *n* at each node along the five-node circuit in Figure 3.2

$$B \propto \frac{\left(\frac{m_{I}^{2}}{n\sqrt{|m_{n}|}}\right)_{x}}{\left(\frac{m_{I}^{2}}{n\sqrt{|m_{n}|}}\right)_{out}}$$
(3.6)

Although the  $f_x(n)$  do not all intersect at exactly the same point, and especially at the origin, the error resulting from this approximation, as shown below, is acceptable. Thus the screen parameter *B* can be qualified by a parameter  $\beta$ , which results from just letting the *n* equal to 2 in the above equation:

$$\beta(x) = \frac{\left(\frac{m_1^2}{\sqrt{m_2}}\right)_x}{\left(\frac{m_1^2}{\sqrt{m_2}}\right)_{out}}$$
(3.7)

where x is the node in question. To find the relationship between B and  $\beta$ , we use a 40-

node uniform wire. The resistance per section is 1000hm and capacitance is 50fF. Equation (3.7) is used to compute  $\beta$ , and the screening parameter *B* is calculated from (3.4) using delays measured with Hspice. Figure 3.4 shows the behaviours of *B* with  $\beta$  for the 40-node wire. Close examination of the data indicates that it has two nearly linear regions, and that the change of the slope occurs at the point where  $(m_2/m_1^2)_x=1$ . Of the 40 nodes, the first 22 nodes satisfy the condition  $(m_2/m_1^2)_x>1$  and *B* is smaller than 1, indicating strong screening. They are classified as *near* nodes. The remaining nodes have  $(m_2/m_1^2)_x<1$  and *B* is approximately equal to 1, indicating weak screening. They are *far* nodes. The linear fits to the data in the two regions, shown in Figure 3.5, are described as follows:



Figure 3.4 Relationship between B and  $\beta$  established using a 40-node RC circuit

Near nodes region:

$$B = 1.2580 * \beta - 0.0244 \tag{3.8}$$

Far nodes region:

$$B = 0.5707 * \beta + 0.4929 \tag{3.9}$$



Figure 3.5 Relationship between B and  $\beta$  in two regions in Figure 3.4

Note that for the near nodes region, in order to use (3.8),  $\beta$  should be larger than 0.0194, otherwise *B* could be negative. Since negative *B* is unrealistic, we simply round *B* to 0. More explicitly:

For near nodes region  $(m_2/m_1^2 > 1)$ , we have:

$$\begin{bmatrix} \beta > 0.0194 & B = 1.2580 * \beta - 0.0244 \\ \beta <= 0.0194 & B = 0 \end{bmatrix}$$
(3.10)

For far nodes region  $(m_2/m_1^2 < 1)$ , we have:

$$B = 0.5707 * \beta + 0.4929 \tag{3.11}$$

The tests of this metric are presented in subsection 3.3.1

## 3.2 Extension of Metric to Arbitrary Trees

Equation (3.4) and, in particular, the second term can not be used directly for multisink RC trees. This can be seen from the RC circuit in Figure 2.3(b). If we are interested in the delay to node  $n_1$ , we have to consider the screening due to  $R_2$  and  $R_5$  separately from  $R_2$ ,  $R_3$  and  $R_4$ . As in the derivation of (3.4), we investigate two extreme cases. Recall that the Elmore delay to the node of interest *int* is given by (2.13):

$$ED^{int} = \sum_{k=1}^{n} R_{k\,int}C_k , \qquad (3.12)$$

where *ED* denote the Elmore delay value. Equation (3.12) overestimates the actual delay to the node *int* because of the screening effect resulting from other branch elements and downstream elements. In fact, it is the upper bound of the actual delay [15]. However, if those elements producing the screening effect are removed, the original tree is reduced to a single wire, with the node of interest *int* at the end of the wire and thus becoming an independent node, *ind*. In other words, there is no screening effect. In such a case, the independent node behaves like an output node, and the Elmore delay to such a node is given by:

$$ED^{ind} = \sum_{k} \sum_{m} R_m C_k \tag{3.13}$$

The delay to the node *int* should be larger than (3.13) and smaller than (3.12). Thus, in the case of multi-sink trees, it may be reasonable to assume that the delay to the node *int* for arbitrary trees is of the form:

$$t_D = ED^{ind} \ln 2 + B(ED^{int} - ED^{ind}) \ln(2),$$
 (3.14)

where B is roughly in the range of 0 to  $1^3$ . Since the D2M metric is more accurate than the Elmore delay for the output nodes of single sink wires, the first term of (3.14) is replaced by D2M, which results in:

$$t_D = \left[ \left( \frac{m_I^2}{\sqrt{m_2}} \right)_{int,ind} + B(ED^{int} - ED^{ind}) \right] ln(2).$$
(3.15)

The screening parameter *B* is qualified by the parameter  $\beta$ , which is defined in (3.7). Recall that  $\beta$  is defined with reference to the output node. In the case of an RC tree, we take this as the node with the smallest  $m_2/m_1^2$ , i.e, which is screened the least.

As an example, consider the simple RC tree in Figure 2.3(b). The delay to the node  $n_2$  can be computed in two steps. First remove the downstream elements and other branch elements of the node  $n_2$  and reduce the original tree to a wire, which is shown in Figure 3.6 and Figure 3.7. Therefore, node  $n_2$  becomes the output node in Figure 3.7. The D2M

<sup>&</sup>lt;sup>3</sup> B can be slightly larger than 1 because of the scaling factor  $\ln 2$  in the equation.

delay to  $n_2$  is given by :

$$t_{D1} = \left(\frac{m_1^2}{\sqrt{m_2}}\right)_{n_2, ind} ln(2)$$
 (3.16)



Figure 3.6 Circuit to remove screening effect elements of Figure 2.3(b)



Figure 3.7 Remaining circuit after removing screening effect elements

Now, consider the Elmore delay to  $n_2$  in Figure 3.7 is:

$$ED^{ind} = R_1(C_1 + C_2) + R_2C_2 \tag{3.17}$$

For the original tree in Figure 2.3(b), the Elmore delay to  $n_2$  is:

$$ED^{int} = R_1(C_1 + C_2 + C_3 + C_4 + C_5) + R_2(C_2 + C_3 + C_4 + C_5)$$
(3.18)

The second term in (3.15) is therefore:

$$t_{D2} = B(ED^{int} - ED^{ind}) \ln 2$$
  
$$t_{D2} = B(R_1(C_3 + C_4 + C_5) + R_2(C_3 + C_4 + C_5)) \ln(2)$$
(3.19)

Combining (3.16) and (3.19), the delay to the node  $n_2$  is determined by:

$$t_D = t_{D1} + t_{D2} \tag{3.20}$$

To find the relationship between B and  $\beta$  in a tree network, a sample100-node tree was generated with all the resistances of 100 ohm, capacitances of 100fF, as shown in Figure 3.9. Equation (3.7) is used to compute  $\beta$ , and the actual delay is measured using Hspice. The screening parameter B is then computed from (3.15). Figure 3.8 shows the linear fitting used to find the relationship between B and  $\beta$  for both far nodes and near nodes (The figure on the left is for near nodes, the figure on the right is for far nodes). Notice that the linearity at the near nodes in the trees is not as strong as in the wires. One of the reasons could be because we group together different screening effects in different branches in the second term in (3.15). The results of the linear fitting are:

For near nodes region  $(m_2/m_1^2 > 1)$ :

$$B = 1.1185 * \beta + 0.2312 \tag{3.21}$$

For far nodes region  $(m_2/m_1^2 < 1)$ :

$$B = 0.4722 \ *\beta + 0.6636 \tag{3.22}$$



Figure 3.8 Relationship between B and  $\beta$  established using a 100-node tree

Note that in the near nodes region, the smallest *B* is 0.2312 when  $\beta$  is zero. This 37

value of *B* is different from the definition of minimum *B* when the screening is complete. It indicates that the relation between *B* and  $\beta$  is not a simple function at some near nodes which are very close to the source. The tests of (3.21) and (3.22) are presented in subsection 3.3.2 and 3.3.3.



Figure 3.9 100-node tree with 16 branches used in the sample test

## 3.3 Experimental Results

## 3.3.1 Wire Tests

The accuracy of the preceding delay metric for wires can be evaluated by comparing it with the actual delay which is obtained by Hspice. We compute the delay using equation (3.4) (3.7) (3.10) (3.11) and measure the error relative to Hspice, (delay metricactual delay)/actual delay. Considering that the new metric is a simple, explicit two moments expression, we compare its results with the results of an existing two moments metric, D2M [2]. The relative errors for the 40-node uniform wire, from which (3.10) and (3.11) are derived, are presented in Figure 3.10 and Figure 3.11. Positive relative error means overestimation of actual delay. Not surprising the metric results match very well with the actual delay for both far nodes and near nodes and performs consistently much better than D2M. The largest relative error occurs at node 1 and node2, which are closest to the source, in which case the new metric underestimates the actual delay. This indicates that the linearity of B vs.  $\beta$  breaks down somewhat at very near nodes.



Figure 3.10 Relative errors for the new metric (-) and for D2M(-.) at far nodes

Next, we generated 100 random 20-node RC circuits and computed delay and relative error to Hspice. Each resistor and capacitor were randomly chosen between 1-20K $\Omega$  and 1-20 fF, respectively. Table 3.1 and Table 3.2 present the average, maximum, minimum and standard deviation of the error distribution of the new metric compared to D2M [2]. Average errors are taken as absolute value. Positive maximum and minimum relative error indicates overestimation.

We can make the following observations:

(1) Like D2M, the new metric has less than 1% average error for all far nodes. Although the new metric is slightly worse than D2M for some far nodes, the average errors of the new metric at all these nodes are still less than 0.5%.



Figure 3.11 Relative error for the new metric (-) and for D2M(-.) at near nodes

(2) For near nodes, the new metric consistently performs significantly better than D2M. The average errors of the new metric are at least seven times less compared to D2M. Notice that the average error at node1, which is the closest to the source, for D2M is 578.44%, while the new metric is about 20 times less than D2M. Also, the new metric is more stable than D2M as evidenced by the smaller standard deviation.

(3) For at least one extreme case, D2M is about 46.79 times the Hspice result at node 1, while the new method is only 3.88 times (maximum relative error is 288.40%).

(4) The results of the new metric for 100 random 20-node RC wires are similar to the results for one 40-node uniform wire, from which (3.10) and (3.11) are derived. It suggests that the relationship between *B* and  $\beta$  in (3.10) and (3.11) are little influenced by the degree of the wire's non-uniformity.

However we notice that the minimum error at node 1 can reach -91% for the new metric. The reason for so a large underestimation is that we simply round *B* to zero when  $\beta$  is less than 0.0194, a region where the simple relationship between *B* and  $\beta$  does not hold any more. Since the actual delay is very small, a little deviation can cause a large relative error.

node	Average Error (%)			
	D2M	New metric		
1	578.44	28.91		
2	223.44	18.19		
3	146.98	13.84		
4	107.35	9.98		
5	74.91	7.53		
6	55.06	6.73		
7	38.46	5.61		
8	23.79	3.54		
9	13.93	1.90		
10	7.64	1.26		
11	4.34	0.88		
12	2.25	0.70		
13	1.10	0.62		
14	0.47	0.50		
15	0.20	0.49		
16	0.14	0.45		
17	0.14	0.37		
18	0.14	0.30		
19	0.16	0.19		
20	0.17	0.17		

Table 3.1 Average error for 100 random 20-node RC circuits

## 3.3.2 A Simple RC Tree Test

In order to test the accuracy of the delay metric for RC trees in (3.21) and (3.22), we run the new metric and D2M on the same circuit used in [2] and [28], shown in Figure 3.12. The delay is computed from (3.7), (3.15), (3.21) and (3.22). The relative errors of the new metric and D2M are listed in Table 3.3. It turns out that the new metric is almost

as accurate as D2M for far nodes, at which  $m_2/m_1^2$  is smaller than 1. However for most of the near nodes we get a better delay estimation than D2M.

node	Minimum e	rror (%)	Maximum error (%)		Standard deviation (%)	
	D2M	New metric	D2M	New metric	D2M	New metric
1	57.92	-91.01	4579.30	288.40	768.87	40.95
2	67.68	-47.70	791.97	85.44	116.85	14.73
3	41.15	-36.23	331.06	69.11	60.87	13.37
4	40.52	-20.09	254.12	28.11	39.83	6.62
5	24.64	-9.03	176.11	31.05	27.69	6.49
6	20.50	-8.48	136.65	35.86	23.51	6.03
7	10.84	-6.35	96.51	29.48	19.06	5.95
8	7.36	-8.28	69.13	18.76	13.02	3.47
9	3.25	-4.86	40.95	10.39	7.81	1.83
10	0.99	-4.55	28.77	4.88	4.83	1.15
11	0.41	-3.36	15.46	3.51	2.97	0.81
12	0.11	-2.56	11.64	2.14	2.03	0.57
13	-0.14	-1.97	8.41	2.68	1.28	0.52
14	-0.29	-1.95	5.51	2.01	0.73	0.42
15	-0.52	-2.34	1.64	1.01	0.27	0.38
16	-0.58	-1.76	0.46	0.93	0.23	0.33
17	-0.56	-1.64	0.30	0.76	0.11	0.29
18	-0.47	-0.86	0.34	0.65	0.10	0.22
19	-0.39	-0.67	0.37	0.44	0.10	0.14
20	-0.35	-0.35	0.41	0.41	0.11	0.11

Table 3.2 Error distribution for 100 random 20-node RC circuits



Figure 3.12 A simple RC tree

Table 3.3 Comparison between D2M and new metric for a simple RC tree

Node	$m_2/m_1^2$	Relative Error (%)			
- 		D2M	New Metric		
1	1.6371	51.95	22.40		
2	1.1771	7.79	5.78		
3	0.9826	-0.62	2.36		
4	0.8872	-1.78	0.03		
5	0.8439	-1.52	-1.95		
6	1.2702	12.41	4.74		
7	1.1350	8.65	11.07		

## 3.3.3 Random Tree Tests

Next, tests were performed on 50 100-node trees, which have the same topology as the sample tree, shown in Figure 3.9. R and C are randomly chosen from the same range of values as in the wire tests. Since the total number of nodes in the test trees prohibits listing the results node by node, they are classified into five categories. The first one is far nodes with  $m_2/m_1^2$  less than 1, the second one is near nodes with  $m_2/m_1^2$  between 1 and 1.5, the third one is near nodes with  $m_2/m_1^2$  between 2 and 2.5, and the fifth one is near nodes with  $m_2/m_1^2$ 

greater than 2.5. In order to compare the results of the random trees to the results of the sample tree, from which (3.21) and (3.22) are derived, we present the error distribution for both the sample and test trees, listed in Table 3.4, Table 3.5, Table 3.6 and Table 3.7 where the numbers in brackets in the first column denote the total number of nodes in a particular category.

It is clear that overall the new metric performs better than D2M for both far nodes and near nodes in terms of accuracy in the average case, especially at the very near nodes. However one problem associated with the new metric is that it tends to underestimate delay. In the worst case, it can reach 45%. Another problem is that the new metric performs better for the sample tree than for the 50 test trees. This indicates that the relationship between *B* and  $\beta$  is sensitive to the parameters of the trees. Indeed, it turns out that the relationship depends on the topology as well.

Category	Average	e error (%)
	D2M	New Metric
1(44)	1.78	0.42
2(42)	8.07	2.97
3(10)	30.03	16.75
4(2)	69.12	18.26
5(2)	740.15	443.50

Table 3.4 Average error for the sample 100-node tree

Table 3.5 Error distribution for the sample 100-node tree

Category	y Minimum Error (%)		Maximu	ım Error (%)	Standard Deviation (%)	
	D2M	New metric	D2M	New metric	D2M	New metric
1(44)	-2.54	-1.02	0.71	1.16	0.60	0.31
2(42)	-0.61	-13.80	24.47	5.18	4.59	2.77
3(10)	9.89	-16.78	119.33	57.74	33.59	14.49
4(2)	39.60	-3.52	98.65	33.01	452.35	21.13

Category	Minimum Error (%)		Maximum Error (%)		Standard Deviation (%)	
	D2M	New metric	D2M	New metric	D2M	New metric
5(2)	312.09	168.90	1168.21	718.10	428.06	274.60

Table 3.6 Average error for 50 random 100-node trees

Category	Average Error (%)			
	D2M	New Metric		
1(2153)	1.60	1.19		
2 (2180)	8.18	5.07		
3 (421)	32.32	17.18		
4 (128)	130.05	69.93		
5(118)	1185.44	827.48		

Table 3.7 Error distribution for 50 random 100-note trees

Category	Category Minimum Error (%)		Maximum	Error (%)	Standard Deviation (%)	
	D2M	New Metric	D2M	New Metric	D2M	New Metric
1(2153)	-3.14	-2.82	3.64	6.21	0.68	0.94
2 (2180)	-8.83	-39.54	38.25	17.42	6.05	5.19
3 (421)	-8.20	-43.51	194.78	116.44	32.25	15.92
4 (128)	-7.49	-34.89	1074.72	719.90	141.37	89.76
5(118)	46.93	-4.67	17495.62	12582.17	2306.08	1756.00

# **4** Delay via Three Moments

In this Chapter, two new metrics are presented to deal with RC interconnect delay estimations. In Section 4.1, an accurate model, 2p1z, for the delay to the internal node of a two-pole, one-zero RC circuit is developed, and serves as the core of the new delay metric for near nodes. This delay model is expressed in terms of two parameters that represent the ratio of the two poles, and the ratio of the dominant pole and zero, respectively. The delay to near nodes in arbitrary RC trees is then computed by order reduction to a two-pole, one-zero model using the first three moments of the impulse response. A further improvement in accuracy is achieved by correcting for the skewness of the impulse response. In parallel, a simple yet accurate delay metric for far nodes, where order reduction is not needed, is presented in Section 4.2. This new metric is based on the first moment at the node in question and the second moment of the output node. A simple criterion is also included for identifying far nodes, where poles dominate the response, from near nodes, where zeros play an important role. Finally in Section 4.3, tests on random RC models of wires and trees demonstrate that the new metrics are accurate for far nodes and for nodes with delays which are more than an order smaller than that of the slowest node. The tests in fact demonstrate the accuracy limits of modelling near nodes using only three moments.

## 4.1 Delay to Near Nodes

## 4.1.1 2-pole 1-zero delay Model (2p1z)

The delay metric for near nodes is based on an accurate model (2p1z) for the delay to the internal node of a two-pole, one zero network, namely node *int* in the circuit in Figure 4.1. We begin by expressing the delay to *int* in terms of two parameters,  $\alpha$  and  $\beta$ , whence we establish the entire range of values that these two parameters may have, in order to speed up the delay evaluation.



Figure 4.1 Two-node circuit

The transfer function at the output node, *out*, of the two-node circuit shown in Figure 4.1, expressed in circuit moments, is given by:

$$H(s)_{out} = \frac{1}{(m_1^2 - m_2)s^2 - m_1s + 1},$$
(4.1)

where  $m_1$  and  $m_2$  are the first and second moments associated with the output node. The poles of this circuit are therefore:

$$P_{1,2} = \frac{m_1 \pm \sqrt{4m_2 - 3m_1^2}}{2(m_1^2 - m_2)}$$
(4.2)

For convenience, we define a parameter  $\alpha$  as follows:

$$\alpha = \frac{m_2}{m_1^2} \tag{4.3}$$

Using (4.3), the two poles can be recast in the following form:

$$P_{12} = \frac{(1 \pm \sqrt{4\alpha - 3})}{2(1 - \alpha)m_1} \tag{4.4}$$

Let  $P_1$  be the dominant pole,  $P_1 > P_2$  (both are real negative), then the ratio of the two poles is given by:

$$\frac{P_1}{P_2} = \frac{1 - \sqrt{4\alpha - 3}}{1 + \sqrt{4\alpha - 3}}$$
(4.5)

It can be easily shown that if one pole dominates the behaviour of the step response,  $\alpha=1$ , while if the two poles are equivalent,  $\alpha=3/4$ . In general,  $\alpha$  is in the range of 0.75 and

1 for two-node circuits [32].

The transfer function at node *int*, in contrast, also contains a zero. In such a case, Using (4.3) and (4.4), the relative position of the zero and the dominant pole can be expressed in the form:

$$\frac{Z}{P_1} = \frac{2(1-\alpha)m_1 z}{1-\sqrt{4\alpha-3}}$$
(4.6)

Note that  $m_1$  in (4.6) is the first moment associated with the output node, *out*. To account for the effect of the zero, we introduce a parameter  $\beta$ , defined as:

$$\beta = 1/(m_1 z) \tag{4.7}$$

With this definition of  $\beta$ , the relative position of the zero and the dominant pole in (4.6) can be rewritten as:

$$\frac{Z}{P_1} = \frac{2(1-\alpha)}{(1-\sqrt{4\alpha-3})\beta}$$
(4.8)

Multiplying the denominator and numerator of above equation by the factor of  $1+\sqrt{4\alpha-3}$ , we finally have:

$$\frac{Z}{P_{\rm I}} = \frac{(1+\sqrt{4\alpha-3})}{2\beta}$$
(4.9)

Equation (4.5) and (4.9) demonstrate that  $\alpha$  describes the relative position of the two poles, and  $\alpha$  and  $\beta$  together quantify the relative position of the dominant pole and zero.

We will be interested later in treating  $\alpha$  as a constant, in which case the dominant pole in (4.4), and the ratios in (4.5) and (4.9) simplify to, respectively:

$$P_{1} = \frac{1}{k_{2}m_{1}}$$

$$\frac{P_{I}}{P_{2}} = k_{I}$$

$$\frac{Z}{P_{I}} = \frac{k_{2}}{\beta}$$
(4.10)

where:

$$k_1 = \frac{1 - \sqrt{4\alpha - 3}}{1 + \sqrt{4\alpha - 3}}, \quad k_2 = \frac{1 + \sqrt{4\alpha - 3}}{2}$$
 (4.11)

To compute the delay at node *int*, we consider the step response which, when expressed in terms of poles and zero, has the form:

$$v(t)_{int} = 1 - \frac{\frac{1}{p_1} - \frac{1}{z}}{\frac{1}{p_1} - \frac{1}{p_2}} e^{p_1 t} - \frac{\frac{1}{z} - \frac{1}{p_2}}{\frac{1}{p_1} - \frac{1}{p_2}} e^{p_2 t}$$
(4.12)

Substituting (4.10) into (4.12), and evaluating the time for the voltage to complete 50% of its transition yields:

$$0.5 = \frac{1 - \frac{\beta}{k_2}}{1 - k_1} e^{\frac{-1}{k_2} t \frac{1}{(-m_1)}} + \frac{\beta}{\frac{k_2}{1 - k_1}} e^{\frac{-1}{k_1 k_2} t \frac{1}{(-m_1)}}$$
(4.13)

A closed form solution of (4.13) is not possible. Hence we seek a solution based on precomputed constant  $\alpha$  characteristics, relating the delay to  $\beta$ . In view of the form of the exponent in (4.13), we express the delay in the normalized form:

$$\frac{t_D^{int}}{(-m_1)} = f(\beta)|_{\alpha = constant}$$
(4.14)

Equation (4.14) shows that parameters  $\alpha$ ,  $\beta$  and the first moment at node *out* are sufficient to determine the delay to node *int*. Before proceeding to do this, we first need to establish some important properties of  $\alpha$  and  $\beta$ . These are needed in particular to establish the exact range that these two parameters can have in the two-node RC circuit. In Appendices 6.2, 6.3 and 6.4, we demonstrate the following relationships between  $\alpha$  and  $\beta$ :

1: With  $\alpha$  constant, the normalized delay  $t_D^{int}/(-m_1)$  monotonically decreases with respect to  $\beta$ .

2:  $\alpha$  and  $\beta$  are correlated to each other. Moreover, when the circuit parameters satisfy the condition  $R_1C_1 < R_1C_2 + R_2C_2$ ,  $\beta$  has the value:

$$\beta = \frac{1 + \sqrt{1 - 4(1 - \alpha)(1 + \frac{R_1}{R_2})}}{2(1 + \frac{R_1}{R_2})}$$
(4.15)

Therefore, for a given value of  $\alpha$ , the maximum value that  $\beta$  can have is:

$$\beta_{max} = \frac{1 + \sqrt{1 - 4(1 - \alpha)}}{2}$$
(4.16)

On the other hand, when  $R_1C_1 > R_1C_2 + R_2C_2$ ,

$$\beta = \frac{1 - \sqrt{1 - 4(1 - \alpha)(1 + \frac{R_1}{R_2})}}{2(1 + \frac{R_1}{R_2})}$$
(4.17)

So for a given value of  $\alpha$ ,  $\beta$  can not have a value less than:

$$\beta_{min} = \frac{1 - \sqrt{1 - 4(1 - \alpha)}}{2}$$
(4.18)

3. When  $\beta$  reaches its maximum, the delay to node *int* is  $t_D^{int} = ln(2)R_1C_1$ . In such a case, the normalized delay in (4.14) becomes:

$$\frac{t_D^{int}}{(-m_1)}\Big|_{\beta=\beta_{max}} = \frac{1-\alpha}{\beta} \ln(2) = \frac{R_1 C_1 (\ln 2)}{(-m_1)}$$
(4.19)

It is easy to verify from property 2 that for  $\alpha = 1$  (dominant pole),  $\beta$  can vary between 0 and 1, while with  $\alpha = \frac{3}{4}$  (coincident poles),  $\beta = 0.5$ . So, in general,  $\beta$  is in the range of 0 to 1. Thus the expressions for the minimum and maximum values of  $\beta$  yield the complete profile of the relationship in (4.14).

The data point in Figure 4.2 are the delays measured using Hspice, expressed as  $t_D^{int}/(-m_1)$ , and plotted as a function  $\beta$ , over the range defined by  $\beta_{max}$  to  $\beta_{min}$ . These are plotted for 7 values of  $\alpha$ : 0.85, 0.9, 0.92, 0.94, 0.96, 0.98 and 0.99. The solid curves correspond to the following rational functions, which were found to best fit the measured data. The coefficients of the rational functions are listed in Table 4.1.

$$\frac{t_D^{int}}{(-m_1)} = \frac{a_5\beta^4 + a_4\beta^3 + a_3\beta^2 + a_2\beta + a_1}{a_{10}\beta^5 + a_9\beta^3 + a_8\beta^2 + a_7\beta + a_6} \quad \text{for } \alpha = 0.99 \quad (4.20)$$

$$\frac{t_D^{int}}{(-m_1)} = \frac{a_4\beta^3 + a_3\beta^2 + a_2\beta + a_1}{a_8\beta^3 + a_7\beta^2 + a_6\beta + a_5} \quad \text{otherwise} \quad (4.21)$$

Close examination of these characteristics indicates that  $t_D^{int}/(-m_1)$  decreases monotonically with respect to  $\alpha$  when  $\beta$  is constant, which means interpolation and extrapolation can be used to determine the relationship between the normalized delay  $t_D^{int}/(-m_1)$  and  $\beta$  for any value of  $\alpha$ . Given these coefficients, delay to any internal node is computed by interpolation or extrapolation between the two nearest values of  $\alpha$ . Since our two-pole one-zero delay expression is computed without any simplifying approximations regarding the relative position of two poles and the zero, it is extremely accurate, irrespective of the "nearness" of the node to the source. The results for 100 randomly generated sets of values of the parameters of the circuit in Figure 4.1 are presented in Table 4.2 in Section 4.3.1. As can be seen, the average error is less than 1%.



Figure 4.2 Relationship between  $t_D^{int}/(-m_1)$  and  $\alpha$ ,  $\beta$ , established using two-node circuit. Dots are Hapice results while the solid lines are given by (4.20) and (4.21).

а	α =0.85	α =0.90	α =0.92	α=0.94	α =0.96	α =0.98	α =0.99
al	0.2816	0.4544	0.4753	0.4058	0.2805	0.5627	0.2033
a 2	0.0667	-1.4935	-1.7331	-1.5022	-1.1540	-2.4583	-1.0216
a 3	-0.0787	2.0609	2.4350	2.0879	1.6634	3.5516	1.8149
a 4	0.2551	-0.8300	-1.0184	-0.8506	-0.7115	-1.6343	-1.3266
a 5	0.4336	0.6188	0.6536	0.5597	0.3866	0.7962	0.3496
a 6	-0.1127	-1.1689	-1.4004	-1.1087	-0.7412	-2.0174	0.2957
a 7	2.7311	1.2140	0.9513	-0.0503	-1.1475	-0.2387	-1.1598
a 8	2.0998	2.3921	2.9944	4.2975	4.6646	4.4664	2.2469
a 9							-3.0602
a 10							5.2896

Table 4.1 Coefficients for the rational functions (4.20) and (4.21)

## 4.1.2 Simplified 2-pole 1-zero Model (2p1zs)

Although the above delay model is very accurate, 7 curves have to be used, so that the data in Figure 4.2 has to be divided into 8 regions for interpolation. This approach can be simplified, with only a modest loss in accuracy, by using only two curves, with  $\alpha$ =0.99 and  $\alpha$ =0.85. These two values of  $\alpha$  are chosen because it was found that most of the data points in Table 4.2 fell within that range.

We define the delays obtained from (4.20) and (4.21) as *delay1* and *delay2*, respectively, and divide the data in Figure 4.2 into three regions according to the above choice of  $\alpha$ :

#### Region I

This region corresponds to  $\alpha \ge 0.99$ . To obtain an expression for the delay corresponding to this region, we plot Figure 4.3, which shows the relationship between  $t_D^{int}/(-m_1)$  and  $\alpha$ ,  $\beta$ , for  $\alpha = 0.999$  and  $\alpha = 0.99$ , where the normalized delay is computed from the time response in (4.13). As can be seen, the plot can be divided into two sub-regions, *I-1*, *I-2*, corresponding, respectively, to  $\beta \le 0.48$  and  $\beta > 0.48$ . In region *I-1*, where  $\beta < 0.48$ , it is difficult to distinguish  $t_D^{int}/(-m_1)$  with  $\alpha = 0.99$  and  $\alpha = 0.999$ , which suggests that the delay for  $\alpha > 0.99$  can be predicted using (4.20). Furthermore, note that 52

the curve in Figure 4.3 has a very small slope in region *I*-2. According to property 3, the normalized delay to the node with maximum  $\beta$  is given by (4.19). It turns out that a better fit to simulation results in region *I*-2 is obtained by modifying this delay expression under maximum  $\beta$  as follows:

$$\frac{t_D^{int}}{(-m_I)} = \frac{1-\alpha}{\beta^2} ln(2)$$
(4.22)

So, in summary, the delays in Region I are computed from:

$$\beta > 0.48 \quad t_D^{int} = \frac{1-\alpha}{\beta^2} ln(2)(-m_1)$$

$$\beta \le 0.48 \quad t_D^{int} = delay1$$
(4.23)



Figure 4.3 Relationship between  $t_D^{int}/(-m_1)$  and  $\alpha$ ,  $\beta$ , with  $\alpha$  larger than 0.99

#### Region II

This region corresponds to nodes with  $\alpha$  between 0.85 and 0.99. The delay is found by interpolation of *delay1* and *delay2*. Figure 4.4 shows the relationship between  $t_D^{int}/(-m_1)$  and  $\alpha$ , for three different  $\beta$ ,  $\beta=0.2$ ,  $\beta=0.5$ ,  $\beta=0.7$ . Observe that when  $\beta=0.2$ , the slope of the curve decreases with  $\alpha$  increasing. In fact, it turns out that this behaviour exists for  $\beta$  in the range of 0 to 0.4. However, when  $\beta$ =0.5, the slope increases with  $\alpha$  increasing. We therefore divided Region *II* into three sub-regions, *II-1* with  $\beta < 0.4$ , II-2 with  $\beta$  between 0.4 and 0.6 and *II-3* with  $\beta > 0.6$ . An expression is of the form

$$t_D^{int} = delay2 - (delay2 - delay1)(\frac{\alpha - 0.85}{0.99 - 0.85})^x$$
 (4.24)

is used to fit the data with  $\alpha$ =0.9, 0.92, 0.94, 0.96, 0.98. It is found that x=0.8 for region *II-1*, x=1.4 for region *II-2* and x=1.2 for region *II-3*. So, in summary, the delay expressions for Region *II* are:

 $0.85 \le \alpha \le 0.99$ 

$$\beta > 0.6 \quad t_D^{int} = delay2 - (delay2 - delay1)(\frac{\alpha - 0.85}{0.99 - 0.85})^{1.2} 0.4 < \beta \le 0.6 \quad t_D^{int} = delay2 - (delay2 - delay1)(\frac{\alpha - 0.85}{0.99 - 0.85})^{1.4} \beta \le 0.4 \quad t_D^{int} = delay2 - (delay2 - delay1)(\frac{\alpha - 0.85}{0.99 - 0.85})^{0.8}$$

$$(4.25)$$



Figure 4.4 Relationship between  $t_D^{int}/(-m_1)$  and  $\alpha$  with three different  $\beta$ 

Region III

This region corresponds to nodes with  $\alpha$  smaller than 0.85. Instead of using *delay1* and *delay2* to extrapolate the delay in Region *III*, we only use *delay2* since the profile of the curve with  $\alpha$ =0.99 is quite different from that corresponding to  $\alpha < 0.85$ . Consider the smallest  $\alpha$ , which is 0.75, and corresponds to a single point in Figure 4.2 ( $\alpha$ =0.75,  $\beta$ =0.5,  $t_D^{int}/(-m_1)=0.5*\ln(2)$ ). We shift *delay2* up to this single point, yielding another curve which we define by *delay3*. It is obvious that the distance between *delay2* and *delay3* is given by:

$$Df = 0.5*ln(2) - delay_2(0.5)/(-m_1)$$
(4.26)

The delay in Region III is then obtained by linear interpolating between delay2 and delay3, which is given by:  $\alpha < 0.85$ 

$$t_D^{int} = delay2 + \frac{Df^*(0.85 - \alpha)}{(0.85 - 0.75)}(-m_1)$$
(4.27)

Figure 4.5 demonstrates the accuracy of using (4.23) (4.25) (4.27). It is easy to verify that the equations always return positive values of delay.



Figure 4.5 Comparison of normalized delay computed from 2p1zs (-) and actual normalized delay (-.)

#### **4.1.3 Delay to Near Nodes in Arbitrary Networks**

## 4.1.3.1 Order Reduction

Delays to near nodes in arbitrary wires or tree networks are computed by using order reduction to approximate the response at the node in question by a two-pole circuit. Then the two-pole, one-zero model presented above is used to compute the delay. We first present the resulting metric, and follow that by showing that a further improvement in the accuracy can be achieved by including the *skewness* of the impulse response.

Recall that the 2-pole 1-zero model for the delay to the internal node in Figure 4.1 is only valid in the range defined by (4.18) and (4.16). It can be shown from Appendix 6.5 that this constraint can also be expressed as follows:

$$\left(\frac{m_2}{m_1^2}\right)_{int} > 1 \tag{4.28}$$

So, to extend the idea of the internal node to an arbitrary network, we have the following definition: node x is a near node if  $(m_2/m_1^2)_x$  is larger than 1, otherwise it is a far node. Hence we use the above delay metric for a near node if this inequality is satisfied.

In [42], an analytical method was presented for computing the poles and zero of a two-pole approximation to the actual response at a node in an arbitrary tree network from the first three moments of the impulse response. The two poles of the reduced model are given by [42]:

$$P_1 = \left(\frac{m_2}{m_3}\right)_{\chi}$$
 and  $P_2 = P_1 \left|\frac{\frac{1}{m_1} - \frac{m_1}{m_2}}{\frac{m_1}{m_2} - \frac{m_2}{m_3}}\right|_{\chi}$  (4.29)

The residues are:

$$r_{I} = \frac{1 - (m_{I})_{x} P_{2}}{P_{2} - P_{I}} P_{I}^{2} \quad r_{2} = -\frac{1 - (m_{I})_{x} P_{I}}{P_{2} - P_{I}} P_{2}^{2}$$
(4.30)

Having poles and residues, the equivalent  $\alpha$  can be determined by (4.5), and the equivalent  $\beta$  is given by (4.7), where

$$m_1 = \frac{(P_1 + P_2)}{P_1 P_2}, \quad Z = \frac{r_1 p_2 + r_2 p_1}{r_1 + r_2}$$
 (4.31)

#### 4.1.3.2 Skewness Correction

Near nodes are characterized by asymmetrical impulse responses, where the asymmetry increases as one approaches the source. This effect is due to the presence of low frequency zeros in the response, whose number increases as the source is approached. In contrast, far nodes, such as some leaf nodes of trees, which exhibit all-pole behaviour, have symmetric impulse responses. This asymmetry is quantified by the second and third central moment of the impulse response, or the coefficient of *skewness*, which depends on the first three moments as follows [15]:

$$\gamma = \left(\frac{-6m_3 + 6m_1m_2 - 2m_1^3}{(2m_2 - m_1^2)^{3/2}}\right)$$
(4.32)

Clearly, the two poles and one zero in (4.29) and (4.30) which result from the above order reduction, based on the first three moments, can not be expected to capture all the skewness in the response for very near nodes. Indeed the above delay metric tends to overestimate the delay for such nodes because of this. However (4.32) indicates that, nevertheless, those three moments contain all the information one needs to determine the full extent of the skew. This suggests that (4.32) may be used to correct for some of the overestimation of delay in the above metric.

To find an expression for this correction, we generated 100 20-node random *RC* wires and used the above metric to compute the delay to the nodes which satisfied (4.28) namely node 1 to approximately 10, where node 1 is the nearest node. For nodes 8 - 10 the average error is below 7%, which means that there may be no need to correct, Figure 4.6 shows the relative error at nodes 4 to 7 as a function of the skewness  $\gamma$ .

It is obvious that, in general, the relative error (positive value means overestimation) increases with increasing skewness. A linear fit yields:

$$Err=0.18952*\gamma-0.37736$$
 (4.33)

If Err in (4.33) is used as a correction factor, the average relative errors (taken as absolute value) will decrease with a lot of negative relative errors. Since we prefer overestimation than underestimation in delay estimation, the line is shifted down so that 57

the correction factor is zero when the skewness is less than 2.35. More explicitly,

$$Err_{skew} = 0 \qquad \gamma \le 2.35$$
  

$$Err_{skew} = 0.18952 * \gamma - 0.435896 \qquad \gamma > 2.35 \qquad (4.34)$$

So the final delay estimation is given by:

$$\Gamma_{Df}^{int} = t_D^{int} / (1 + Err_{skew})$$

$$(4.35)$$



Figure 4.6 Relationship between *Err<sub>skew</sub>* and skewness established by 100 20-node circuits

Since the correction factor  $Err_{skew}$  is non-negative and  $t_D^{int}$  in (4.23) (4.25) (4.27) is positive, the delay metric for near nodes in (4.35) is always stable.

From the above discussion, our metric for near nodes can be summarized as follows:

Delay Metric at Near Nodes
1. Calculate the three moments $(m_1)_x, (m_2)_x, (m_3)_x$
2. Calculate the approximate poles and residues using three moments from
(4.29) and (4.30).
3. Calculate the equivalent $\alpha$ and $\beta$ using approximate poles and residues
from (4.5), (4.7) and (4.31).

- 4. Using 2p1zs model in (4.23) (4.25) (4.27) to find delay  $t_D^{int}$ .
- 5. Calculate skewness  $\gamma$  in (4.32).
- 6. Compute correction factor Errskew using (4.34).
- 7. Find the delay using (4.35).

## 4.2 Delay to Far Nodes

As explained earlier, the preceding metric is valid for near nodes, i.e. those that satisfy the condition in (4.28). Note that there is a kind of linearity between the normalized delay,  $t_D^{int}/(-m_1)$ , and  $\beta$  for small value of  $\beta$  in Figure 4.2, which suggests that it may be possible to find a simple metric for far nodes that does not require order reduction like we do for near nodes.

Since it is difficult to calculate the zero in (4.7) for arbitrary trees, we introduce another method to compute  $\beta$ , which is equivalent to the one in (4.7) for a two-node circuit, yet based on moments. Expressing the zero, the first moment,  $m_1$ , at the output node *out*, and the first moment,  $(m_1)_{int}$ , at the node *int* in terms of the circuit parameters in Figure 4.1, we have:

$$z = -1/R_2 C_2 \tag{4.36}$$

$$-(m_1)_{int} = R_1(C_1 + C_2) \tag{4.37}$$

$$-(m_1) = R_1(C_1 + C_2) + R_2 C_2 \tag{4.38}$$

Substituting (4.36), (4.37), and (4.38) into (4.7) results in:

$$\beta = (m_1 - (m_1)_{int})/m_1 \tag{4.39}$$

As will be shown in the tests below, the computation of  $\beta$  for a two-node circuit at the node *int* can, in fact, be extended to an arbitrary tree at the node of interest, x, namely:

$$\beta = (m_1 - (m_1)_x)/m_1 \tag{4.40}$$

Given the definition of  $\alpha$  in (4.3) and  $\beta$  in (4.40), a simple metric based on the two parameters can be used when  $\beta$  for the node in question is outside the range defined by (4.28) and, in particular, when  $\beta < \beta_{min}$ . This corresponds to the behavior of a far node. For example, when  $\alpha = 1$ ,  $\beta=0$ , which means that there are no low frequency zeros and,

therefore, a simple metric like D2M should suffice. Tests have been performed on 20node wires. These reveal that the relationship between the normalized delay  $t_D^{far}/(-m_1)$ and  $\beta$  is indeed quite linear at far nodes for a given value of  $\alpha$ , where  $t_D^{far}$  refers to the delay to far nodes. This is illustrated in Figure 4.7 for one 20-node wire with  $\alpha = 0.8608$  at the output node.



Figure 4.7 Strong linearity between  $\beta$  and  $t_D^{far}/(-m_1)$  for one 20-node wire

The metric for far nodes is found by assuming that D2M is accurate at the output node of a wire, in which case  $\beta = 0$  and, the intercept of  $t_D^{far}/(-m_1)$  as a function of  $\beta$  is  $\frac{ln(2)}{\sqrt{\alpha}}$ . It turns out from tests on wire models that the slope of these characteristics varies between -1 and -1.15. Hence, we simply choose the slope to be -1 so that, if anything, a slight overestimate of the delay will result. Therefore the delay to a node where condition (4.28) is not satisfied is computed from:

$$t_D far = (-\beta + \frac{\ln(2)}{\sqrt{\alpha}})(-m_1)$$
 (4.41)

Recall that  $m_1$  is the first moment of the output node of the two-pole circuit in Figure 4.1 or, for that matter, of the sink node of a wire. In the case of an RC tree, it is the node

with the maximum value of  $(-m_1)$  that is taken as the "*output*" node. It is a straightforward matter to show that the output node defined in this manner will always be one of the sink nodes, and that this node behaves as a far node, where the condition in (4.28) is not satisfied (Appendix 6.6). So, the output node can be identified very efficiently.

Thus our metric to far nodes can be summarized as follows:

Delay Metric at Far Nodes					
1. Calculate the first moment $(m_1)_x$ for node x					
2. Search for the maximum of $(-m_1)$ among the sink					
nodes, and define it to be the "output" node					
3. Compute the second moment $m_2$ for the output node.					
4. Calculate $\alpha$ and $\beta$ in (4.3) and (4.40).					
5. find delay using (4.41)					

Note that the delay metric at far nodes only needs the first moment of each node in question and the second moment of the output node, so it is very simple. In fact, as will be seen in the experimental results, its accuracy is somewhat better than that of D2M.

## 4.3 Experimental Results

We present here a comparison of Hspice simulation-based delay evaluations with those obtained using the new metrics described in the preceding section. In every case, either the explicit metric for near nodes in Section 4.2 or the simple metric for far nodes described in Section 4.3 is used, depending on whether condition (4.28) is satisfied or not.

#### 4.3.1 Two-node Circuit Tests

In order to test the accuracy and efficiency of our two-pole and one-zero model, 100 random two-node circuits are generated, The delay to the near node *int* node in Figure 4.1 is computed by using (4.20) and (4.21) for 2p1z, or by (4.23) (4.25) (4.27) for 2p1zs.

Each resistor and capacitor is randomly chosen from between 1-20K $\Omega$  and 1-20 fF, respectively. Table 4.2 lists the average, maximum, minimum relative error distribution and standard deviation of 2p1z and 2p1zs at near nodes, node *int*, compared to D2M. Average errors are taken as absolute value. Positive error in Maximum and Minimum error refers to overestimation of actual delay.

As can be seen from Table 4.2, the two-pole one-zero (2p1z) is very accurate, and 2p1zs performs well too, with an average error of only 0.69% and 2.54% respectively. Predictably D2M overestimates the delay at near nodes, in one case by as much as 460% while ours is only 12.27%. Furthermore, the small standard deviation, 1.71% for 2p1z and 2.73% for 2p1zs, indicates strong stability of the 2-pole 1-zero model. In view of the good performance of the simpler 2p1zs, it is used in all the remaining test results, unless indicated otherwise.

Metric	Average	Minimum Error	Maximum Error	Standard Deviation
2p1z	0.69%	-1.94%	12.27%	1.71%
2p1zs	2.54%	-8.41%	11.84%	2.73%
D2M[2]	42.12%	-20.15%	460.00%	81.97%

Table 4.2 Comparison errors due to 2p1z and 2p1zs models and D2M

#### 4.3.2 Wire Tests

Tests were performed on 100 20-node wire models, with the resistances and capacitances randomly chosen from the same range of values that was used above. Since we use the same method as in [42] to compute the poles and residues of the two pole equivalent model from the first three moments of the impulse response, it is interesting to compare our metrics to the one in [42]. Now what sets the new metrics apart from [42] is how the resulting two pole model is treated. In [42], the delay is evaluated using a single Newton-Raphson (N-R) iteration to find a close form delay expression, with the exponential term associated with the dominant pole to be the dominant factor. Unfortunately, these approximations restrict the metric to far nodes, and in fact this metric yields negative delays for near nodes. In view of the accuracy of the 2plzs model,

we compared our metrics to the approach in [42] but with *full* N-R, namely solving completely the exponential time response equation. The results, labelled ITER, are presented together with the new metrics, D2M, and LND [3] in Table 4.3. Since D2M [2] performs consistently better than Elmore [14], Scaled Elmore and the metric presented by Kahng and Muddu [23], we do not list those results. The following observations can be made:

• Not surprisingly, the new metric performs consistently better than D2M at near nodes. On average, the error at such nodes is ten times smaller than for D2M.

• For far nodes ITER is the most accurate, which we would expect. For near nodes, the new metric is superior. This is due to the fact that the skewness is quite large for very near node, but there was no skewness correction in [42] and we did not include it in ITER. So we should see very close agreement between the new metric and ITER if skewness correction is omitted from the former. This is confirmed in Table 4.4, which shows the delays in the nearest ten nodes in a 20-node wire, and Newwc denotes the new metric without the skewness correction.

• New metric is almost as accurate as D2M at far nodes.

• The results for the lognormal metric (LND) are similar to those of D2M, with about 2% worse than D2M at far nodes and about 5%~10% worse than D2M at near nodes. Because of their similarity, we do not compare to the LND metric in the following tree tests.

• Table 4.4, in particular, demonstrates that the main source of error in our metric is the order reduction.

Node	Average Error (%)							
	New metric ITER [42] D2M [2] LND [3							
1	113.13	325.12	578.44	592.10				
2	27.30	74.68	223.44	229.96				
3	8.81	33.00	146.98	151.96				
4	4.75	18.76	107.35	111.53				

Table 4.3 Average error of different delay metrics for 100 random 20-node wires

Node	Average Error (%)						
	New metric	ITER [42]	D2M [2]	LND [3]			
5	5.40	11.59	74.91	78.43			
6	7.80	9.75	55.06	58.18			
7	7.33	8.30	38.46	41.25			
8	6.36	6.68	23.79	26.28			
9	5.10	5.16	13.93	16.22			
10	4.15	4.06	7.64	9.81			
11	3.77	3.23	4.34	6.44			
12	2.84	1.98	2.25	4.30			
13	1.77	0.81	1.10	3.13			
14	1.06	0.26	0.47	2.47			
15	0.59	0.08	0.20	2.11			
16	0.34	0.08	0.14	1.99			
17	0.23	0.09	0.14	2.00			
18	0.18	0.10	0.14	2.05			
19	0.17	0.10	0.16	2.11			
20	0.17	0.10	0.17	2.15			

Table 4.4 Comparison of error for ITER and New metric without skewness correction,for the nearest 10 nodes in 100 20-node wires.

Node	Average Error (%)				
1997 - A	ITER Newwc				
1	325.12	331.34			
2	74.68	83.19			
3	33.00	35.02			
4	18.76	18.89			
5	11.59	11.29			

64

.

Node	Average Error (%)				
	ITER	Newwc			
6	9.75	9.59			
7	8.30	8.19			
8	6.68	6.58			
9	5.16	5.13			
10	4.06	4.15			

## 4.3.3 A Simple Tree Test

We make a quick comparison with previous metrics [2] [26] [27] and [28] on a frequently used simple tree, shown in Figure 3.12. The relative errors to actual delay are listed in Table 4.5, where positive error indicates overestimation of actual delay. As can been seen from Table 4.5, our new metrics perform consistently better than D2M at both far nodes and near nodes. Since PRIMO, H-gamma and WED [26] [27] [28] require lookup tables, we did not implement them, but simply quote the results given in [28]. For this example at least, our metric appears more accurate than WED, comparable to PRIMO, and slightly worse than H-gamma. Referring to the results for the metric in Chapter 3 in Table 3.3, it is obvious that the new metric presented in Chapter 4 returns better results in terms of accuracy than the one described in Chapter 3.

Node	$m_2/m_1^2$	Relative Delay Error (%)						
		New Metric	D2M[2]	ITER[42]	PRIMO[26]	H-gamma[27]	WED[28]	
1	1.64	-11.16	51.95	-11.34	22.96	-1.02	25.51	
2	1.18	0.19	7.79	-0.96	4.62	2.10	1.89	
3	0.98	0.09	-0.62	-0.05	-0.14	0.14	0.29	
4	0.89	-1.38	-1.78	-0.01	-0.95	-0.47	1.30	
5	0.84	-1.52	-1.52	-0.15	-1.09	-0.76	2.61	

Table 4.5 Comparison between different delay metrics for a simple RC tree

Node	$m_2/m_1^2$	Relative Delay Error (%)						
		New Metric	D2M[2]	ITER[42]	PRIMO[26]	H-gamma[27]	WED[28]	
6	1.27	-2.13	12.41	2.76	0.53	-5.08	3.21	
7	1.14	6.08	8.65	10.71	-0.66	-4.86	3.75	

## 4.3.4 Random Tree Tests

Finally, tests were performed on a 100-node tree with 16 branches, shown in Figure 3.9. 50 trees are generated with R and C randomly chosen like in previous tests. The average, minimum, maximum relative errors and standard deviation are presented in Table 4.6 and Table 4.7. Since the total number of nodes in the test trees prohibits listing results node by node, the total 5000 nodes are classified into five categories:

• Category 1: 2153 far nodes with  $(m_2/m_1^2)_x < 1$  have delay greater or equal to 55% of the maximum delay in the trees

• Category 2: 2180 near nodes with  $(m_2/m_1^2)_x$  between 1 and 1.5 have delay between 15% and 75% of the maximum delay in the trees.

• Category 3: 421 near nodes with  $(m_2/m_1^2)_x$  between 1.5 and 2 have delay between 7% and 28% of the maximum delay in the trees.

• Category 4: 128 near nodes with  $(m_2/m_1^2)_x$  between 2 and 2.5 have delay between 1.5% and 14% of the maximum delay in the trees.

• Category 5: 118 near nodes with  $(m_2/m_1^2)_x > 2.5$  have delay between 0.009% and 7% of the maximum delay in the trees.

As can bee seen from the results, our metrics not only perform much better than D2M at near nodes (on average, the error is five times less than that of D2M), but it is marginally better at far nodes. Again, the comparison with ITER reveals the significant improvement in the delay estimate resulting from the skewness correction for near nodes. Note that even for Category 4 nodes the average error with our metric is only 20%. Referring to Table 3.6 and Table 3.7 the new metric in Chapter 4 returns much smaller errors than the previous one in Chapter 3 in terms of average, maximum, minimum relative errors and standard deviation.

We did not attempt to compare with H-gamma [27], PRIMO [26], or WED [28] on our random trees because of the need to generate large and tuned tables, and the quality of the tables affects the results of the metrics. However we note the results on industry nets presented in table 6 of [28], where the average error at near nodes of WED is 49.3% compared to that of D2M with 56.4%. As pointed out above, on average our metric return delays at near nodes with about five times less error than that for D2M. Furthermore, the authors in [3] pointed out that WED [28] [29] may underestimate the actual delay by 30-40% for near nodes. Although H-gamma [27] is by far the most accurate yet the most complicated delay metric, it may significantly underestimate delay for the nodes which are very close to the source [2], while PRIMO [26] can return negative delay for some near nodes.

100-node trees							
Category	Average Error (%)			Standard Deviation (%)			
	New Metric	D2M[2]	ITER[42]	New Metric	D2M[2]	ITER[42]	
1	1.23	1.60	0.33	0.66	0.68	0.50	
2	3.88	8.18	8.87	3.23	6.05	10.37	
3	4.86	32.32	23.42	4.16	32.35	18.21	
4	20.82	130.05	51.64	23.37	141.37	30.32	
5	379.26	1185.44	662.56	727.34	2306.08	1344.65	

Table 4.6 Average error and standard deviation of different delay metric for 50 random 100-node trees

Table 4.7 Maximum and minimum errors of different delay metrics for 50 random 100-

node trees

Category	Minimum Error (%)			Maximum Error (%)		
	New Metric	D2M[2]	ITER[42]	New Metric	D2M[2]	ITER[42]
1	-5.03	-3.14	-3.05	2.20	3.64	5.72
2	-13.01	-8.83	-13.95	18.64	38.25	65.28
3	-18.26	-8.20	-16.84	30.57	194.78	89.01
Category	Minimum Error (%)			Maximum Error (%)		
----------	-------------------	--------	----------	-------------------	----------	----------
	New Metric	D2M[2]	ITER[42]	New Metric	D2M[2]	ITER[42]
4	-4.94	-7.49	9.08	117.33	1074.72	167.29
5	3.03	46.93	24.95	5379.23	17495.62	8854.72

ſ

### **5** Conclusions and Future Work

We have reviewed an existing metric [32] based on two moments, and have extended it to arbitrary RC tree networks. Tests on randomly generated wires demonstrate that the metric is quite accurate. However large errors result at *near* nodes for random trees. This confirms that at least three moments need to be used for delay estimation at near nodes.

A new explicit delay metric has been presented for dealing with near nodes in RC interconnect. It is based on the first three moments of the impulse response. An accurate model for the delay to the internal node of a two-pole one-zero RC circuit serves as the core of the new metric. This delay model is expressed in terms of two parameters that represent the ratio of the two poles and the dominant pole and zero, respectively. Since no simplifying assumption is made about the relative position of poles and zero, this model can return excellent accuracy at the internal node in any two-node circuit. Therefore, it provides an explicit solution for any two-pole one-zero system without the necessity of solving a transcendental equation. The delay at near nodes in arbitrary RC trees is then computed by order reduction to a two-pole one-zero system using the first three moments of the impulse response. A further improvement in accuracy is achieved by correcting for the skewness of the impulse response.

We have also derived a simple metric for predicting the delay to far nodes, where order reduction is not needed. It is based on the first moment of the node of interest and the second moment of the output node. In parallel, we have introduced a simple criterion for distinguishing a near node, where zeros play an important role in determining the delay, from a far node, where the poles dominate. Tests on RC models of wires and trees demonstrate that the combination of these two metrics is accurate within 2% for far nodes and within 5% for near nodes with delays which are as much as an order of magnitude smaller than that of the slowest node, which is always a sink node. Thus the new metrics appear to be an attractive alternative to metrics such as H-gamma [27], which is by far 69

the most accurate metric but requires large two-dimensional look-up tables and is more challenging to implement. Since the main source of errors in the new metrics is due to the order reduction to a two-pole equivalent circuit, we conclude that any further improvement in delay metrics will require the use of more than three moments of the impulse response.

Future work in this regard could consider delay estimation at any point on the response waveform, not limited to 50% point, e.g. computing the 10% delay point and 90% delay point would result in a 10/90 slew metric. Also we would like to explore explicit delay metrics that take the effect of inductance and coupling capacitance of interconnects into consideration.

# 6 Appendix

#### 6.1 Interconnect Order Reduction

Since the new metrics in Chapter 4 and some existing metrics described in Chapter 2 involve interconnect order reduction, it is appropriate to review this technique and its typical application, Asymptotic Waveform Evaluation (AWE) [36].

Interconnect order reduction is a technique that takes advantage of the fact that although a linear or linearized circuit model of the interconnect may consist of a large number of poles, only a few of these poles are sufficient to determine the behaviour of the circuit model. Assume a transfer function of an RC tree circuit is expressed as follows:

$$H(s) = \frac{1 + a_1 s + a_2 s^2 + \dots + a_n s^m}{1 + b_1 s + b_2 s^2 + \dots + b_n s^n}$$
(6.1)

where the coefficients are all real numbers and  $n \ge m$ . Expanding (6.1) about s=0 results in:

$$H(s) = \sum_{j=0}^{\infty} m_j s^j$$
(6.2)

where  $m_0, m_1, m_2, ..., m_j$ ... are the moments defined in (2.3). Since the original system may have a large number of poles, a reduced order approximation for H(s) is considered:

$$\hat{H}(s) = \frac{a_0 + \hat{a}_1 s + \hat{a}_2 s^2 + \dots + \hat{a}_{q-1} s^{q-1}}{1 + \hat{b}_1 s + \hat{b}_2 s^2 + \dots + \hat{b}_q s^q} (q < n)$$
(6.3)

and its corresponding moments are:

$$\hat{H}(s) = \hat{m}_0 + \hat{m}_1 s + \hat{m}_2 s^2 + \dots$$
 (6.4)

One efficient way to find these approximate q dominant poles is AWE [36], which is

based on moment matching. Moment matching refers to fitting the moments of the approximate system with the moments of the original one, which is  $\hat{m}_i = m_i$ . For example, multiplying (6.3) and (6.4) by the denominator polynomial in (6.3) and then equating the terms with q power of s through 2q-1 power of s, we have the following equations:

$$\begin{bmatrix} m_{q-1} & m_{q-2} & \cdots & m_0 \\ m_q & m_{q-1} & \cdots & m_1 \\ \vdots & \vdots & \vdots & \vdots \\ m_{2q-2} & m_{2q-3} & \cdots & m_{q-1} \end{bmatrix} \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \vdots \\ \hat{b}_q \end{bmatrix} = -\begin{bmatrix} m_q \\ m_{q+1} \\ \vdots \\ m_{2q-1} \end{bmatrix}$$
(6.5)

Solving the above equations would result in the coefficient  $\hat{b}_i$  or the poles  $\hat{p}_i$  of the reduced order model (roots of the denominator). To compute the residue for the purpose of the time response of the circuit, consider that the partial fractions of the approximated transfer function in (6.3) are given by:

$$\frac{\hat{r}_1}{s - \hat{p}_1} + \frac{\hat{r}_2}{s - \hat{p}_2} + \dots + \frac{\hat{r}_q}{s - \hat{p}_q}$$
(6.6)

where  $\hat{p}_i$ ,  $\hat{r}_i$  denotes the poles and residues of the transfer function respectively. Expanding the partial fractions in the above equation and matching the first q moments of the reduced order system to those of the original circuit would result in:

$$\begin{bmatrix} \hat{p}_{I}^{-1} & \hat{p}_{2}^{-1} & \cdots & \hat{p}_{q}^{-1} \\ \hat{p}_{I}^{-2} & \hat{p}_{2}^{-2} & \cdots & \hat{p}_{q}^{-2} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{p}_{I}^{-q} & \hat{p}_{2}^{-q} & \vdots & \hat{p}_{q}^{-q} \end{bmatrix} \begin{bmatrix} \hat{r}_{I} \\ \hat{r}_{2} \\ \vdots \\ \hat{r}_{q} \end{bmatrix} = -\begin{bmatrix} m_{0} \\ m_{I} \\ \vdots \\ m_{q-I} \end{bmatrix}$$
(6.7)

Given poles and residues, the time response of the system is given by:

$$\hat{h}(t) = \hat{r}_{1}e^{\hat{p}_{1}t} + \hat{r}_{2}e^{\hat{p}_{2}t} + \dots + \hat{r}_{q}e^{\hat{p}_{q}t}$$
(6.8)

The moment matching method used in AWE is well known as 'Padé' approximation [5]. This method provides an efficient way to capture the behaviour of linear or linearized circuits, both in time and frequency domains. However, it has an inherent problem, namely unstable poles could be produced even for a stable linear RLC circuit, especially

for high approximation orders more than 6 [5]. Due to the limitation of AWE, there is significant literature on offering stable poles based on moments [4] [22] [42]. Tutuianu el al. [42] developed a provably stable two-pole transfer function based on the first three moments. The two guaranteed stable poles are given by:

$$p_1 = \frac{m_2}{m_3}, \ p_2 = p_1 \left( \frac{\left(\frac{m_0}{m_1} - \frac{m_1}{m_2}\right)}{\left(\frac{m_1}{m_2} - \frac{m_2}{m_3}\right)} \right)$$
(6.9)

and corresponding residue is

$$r_{1} = \frac{1 - (m_{1})_{x} P_{2}}{P_{2} - P_{1}} P_{1}^{2}, r_{2} = -\frac{1 - (m_{1})_{x} P_{1}}{P_{2} - P_{1}} P_{2}^{2}$$
(6.10)

Since  $m_2>0$ ,  $m_3<0$  for RC tree, the dominant pole and second pole are both stable. Given the poles and residues, the output voltage response can be computed by using two exponential terms.

#### 6.2 Property 1 of Parameters $\alpha$ and $\beta$ in a Two-node Circuit

Considering a two-node circuit, as shown in Figure 4.1, we rewrite the 50% delay in (4.13) as:

$$0.5 = \frac{1 - \frac{\beta}{k_2}}{1 - k_1} e^{\frac{-1}{k_2} t \frac{1}{(-m_1)}} + \frac{\beta}{\frac{k_2}{1 - k_1}} e^{\frac{-1}{k_1 k_2} t \frac{1}{(-m_1)}}$$
(6.11)

Taking partial derivative with respect to  $\beta$  when  $\alpha$  is constant, which means  $k_1$ ,  $k_2$  are constant, and considering the fact that  $t_D^{int}/(-m_1)$  is a function of  $\beta$ , we have:

Doing some mathematical manipulation, the above equation becomes:

$$\frac{\partial(\frac{t_D^{int}}{(-m_1)})}{\partial\beta} = \frac{e^{\frac{-1}{k_1k_2}}(-m_1)}{e^{\frac{-1}{k_2}}(-m_1)} - e^{\frac{-1}{k_2}}(-m_1)} \frac{e^{\frac{-1}{k_2}}(-m_1)}{e^{\frac{-1}{k_2}}(-m_1)}$$
(6.13)

Since

$$e^{\frac{-1}{k_{1}k_{2}}\left(-m_{1}\right)} - e^{\frac{-1}{k_{2}}\left(-m_{1}\right)} < 0, \qquad (6.14)$$

and it has been shown in [5] that  $p_1 > z$  (both are negative), then equation (4.10) becomes:

$$\frac{P_l}{Z} = \frac{\beta}{k_2} < l \tag{6.15}$$

Considering the fact that  $k_1$  (4.11) is always smaller than 1, and combining equation (6.13) (6.14) (6.15) we can conclude that:

÷.,

$$\frac{\partial \left(\frac{t_D^{int}}{(-m_1)}\right)}{\partial \beta} < 0 \tag{6.16}$$

Therefore when  $\alpha$  is constant, the normalized delay monotonically decreases with respect to  $\beta$ .

#### 6.3 Property 2 of Parameters $\alpha$ and $\beta$ in a Two-node Circuit

Since  $\alpha$  and  $\beta$  are determined by the first and second moments, we first express the two moments by two-node circuit parameters:

$$(-m_{1}) = (R_{1}C_{1} + R_{1}C_{2} + R_{2}C_{2})$$

$$(6.17)$$

$$m_{2} = (R_{1}C_{1} + R_{1}C_{2} + R_{2}C_{2})^{2} - R_{1}R_{2}C_{1}C_{2}$$

$$(-m_{1})_{int} = (R_{1}C_{1} + R_{1}C_{2})$$

$$(m_{2})_{int} = (R_{1}C_{1} + R_{1}C_{2})^{2} + R_{1}R_{2}C_{2}^{2}$$

Then from the definition of  $\alpha$ , we have

$$\alpha = \frac{(R_1C_1 + R_1C_2 + R_2C_2)^2 - R_1R_2C_1C_2}{(R_1C_1 + R_1C_2 + R_2C_2)^2}$$
(6.18)

Using expression of moments in (6.17), the fact that  $z=-1/R_2C_2$  in two-node circuits and the definition of  $\beta$  in (4.7), we obtain:

$$\beta = \frac{R_2 C_2}{R_1 C_1 + R_1 C_2 + R_2 C_2} \tag{6.19}$$

Dividing the denominator and numerator by a factor of  $R_2C_2$ , we have:

$$\beta = \frac{1}{1 + \frac{R_1}{R_2} + \frac{R_1 C_1}{R_2 C_2}}$$
(6.20)

Using (6.18) and (6.19), we have:

$$\beta = \frac{1}{1 + \frac{R_1}{R_2} + \frac{1 - \alpha}{\beta^2}}$$
(6.21)

Solving the above equation, we obtain:

$$\beta = \frac{1 \pm \sqrt{1 - 4(1 - \alpha)(1 + \frac{R_1}{R_2})}}{2(1 + \frac{R_1}{R_2})}$$
(6.22)

Note that there are two possible values for  $\beta$ , and which one we should use is determined by the expression inside the square root in (6.22). Considering the expression in square root and using (6.18), we have:

$$1 - 4(1 - \alpha)(1 + \frac{R_1}{R_2}) = 1 - \frac{4R_1R_2C_1C_2 + 4R_1^2C_1C_2}{(R_1C_1 + R_1C_2 + R_2C_2)^2}$$
(6.23)

Doing some mathematical manipulation, the right hand side becomes:

$$=\frac{(R_1C_1 - R_1C_2 - R_2C_2)^2}{(R_1C_1 + R_1C_2 + R_2C_2)^2}$$
(6.24)

It follows that when  $R_1C_1 < R_1C_2 + R_2C_2$  the value of the square root in (6.22) is:

$$\sqrt{1 - 4(1 - \alpha)(1 + \frac{R_1}{R_2})} = \frac{R_1 C_2 + R_2 C_2 - R_1 C_1}{R_1 C_1 + R_1 C_2 + R_2 C_2}$$
(6.25)

Since (6.19) and (6.22) should be equal, therefore the sign in (6.22) should be positive, which results in:

$$\beta = \frac{1 + \sqrt{1 - 4(1 - \alpha)(1 + \frac{R_1}{R_2})}}{2(1 + \frac{R_1}{R_2})}$$
(6.26)

It follows from (6.26) that, under the condition that  $\alpha$  is constant,  $\beta$  achieves its maximum at  $R_1/R_2 \rightarrow 0$  (when  $R_1/R_2 \rightarrow 0$  is zero, the numerator is minimum, while the denominator is maximum), which is:

$$\beta_{max} = \frac{1 + \sqrt{1 - 4(1 - \alpha)}}{2}$$
(6.27)

Using a similar process it can be derived that when  $R_1C_1 > R_1C_2 + R_2C_2$ ,  $\beta$  is in the form:

$$\beta = \frac{1 - \sqrt{1 - 4(1 - \alpha)(1 + \frac{R_1}{R_2})}}{2(1 + \frac{R_1}{R_2})}$$
(6.28)

The minimum value of  $\beta$  in (6.28) is not as obvious as the maximum value of  $\beta$  in (6.22). Let us first take the derivative of (6.28) with respect to  $R_1/R_2$  with constant  $\alpha$ . To simplify the denotation, we denote  $k=R_1/R_2$ , therefore:

$$\frac{d\beta}{dk} = \frac{\frac{-4(1-\alpha)(1+k)}{2(1+k)^2} - (1-\sqrt{1-4(1-\alpha)(1+k)})}{2(1+k)^2}$$
(6.29)

Doing some mathematical manipulation, we have:

$$\frac{d\beta}{dk} = \frac{1 - 2(1 - \alpha)(1 + k) - \sqrt{1 - 4(1 - \alpha)(1 + k)}}{2(1 + k)^2 \sqrt{1 - 4(1 - \alpha)(1 + k)}}$$
(6.30)

It is obvious from (6.30) that:

$$\frac{d\beta}{dk} \ge 0 \tag{6.31}$$

It follows that  $\beta$  monotonically increases with k for constant  $\alpha$ . Therefore  $\beta$  has its minimum when k is minimum, which is zero. In such a case, the minimum  $\beta$  with  $\alpha$  constant has the form:

$$\beta_{min} = \frac{1 - \sqrt{1 - 4(1 - \alpha)}}{2}$$
(6.32)

#### 6.4 Property 3 of Parameters $\alpha$ and $\beta$ in a Two-node Circuit

Consider the time for the voltage to complete 50% of its transition in (4.13) when  $\beta$  reaches its maximum:

$$0.5 = \frac{1 - \frac{\beta_{max}}{k_2}}{1 - k_1} e^{\frac{-1}{k_2} t \frac{t_D^{int}}{(-m_1)}} + \frac{\frac{\beta_{max}}{k_2} - k_1}{1 - k_1} e^{\frac{-1}{k_1 k_2} t \frac{t_D^{int}}{(-m_1)}}$$
(6.33)

Note that  $\beta_{\text{max}}$  in (6.27) is equal to  $k_2$  in (4.11), which means the coefficient of the first exponent term in (6.33) is zero, and the coefficient of the second exponent term is 1. Therefore (6.33) becomes:

$$0.5 = e^{\frac{-1}{k_1 k_2} \frac{t_D^{int}}{(-m_1)}}$$
(6.34)

It follows that the normalized delay is of the form:

$$\frac{t_D^{int}}{(-m_I)}\Big|_{\beta=\beta_{max}} = k_1 k_2 \ln(2)$$
(6.35)

Substituting  $k_1$ ,  $k_2$  in (4.11) into (6.35), and consider  $\beta_{max}$  in (6.27), we have:

$$\frac{t_D^{int}}{(-m_I)}\Big|_{\beta=\beta_{max}} = \frac{1-\alpha}{\beta} ln(2)$$
(6.36)

Referring to  $\alpha$  and  $\beta$  in (6.18) and (6.19), (6.36) can be recast as follows:

$$t_D^{int}\Big|_{\beta=\beta_{max}} = R_1 C_1 \ln(2)$$
(6.37)

#### 6.5 **Proof of Condition in (4.28)**

$$\frac{1 - \sqrt{1 - 4(1 - \alpha)}}{2} < \beta < \frac{1 + \sqrt{1 - 4(1 - \alpha)}}{2}$$
(6.38)

It follows that:

$$(\beta - \frac{1 - \sqrt{1 - 4(1 - \alpha)}}{2})(\beta - \frac{1 + \sqrt{1 - 4(1 - \alpha)}}{2}) < 0$$
 (6.39)

Doing some mathematical manipulation, we have:

$$\beta^2 - \beta + (1 - \alpha) < 0 \tag{6.40}$$

Therefore:

$$\frac{(\alpha - \beta)}{(1 - \beta)^2} > 1 \tag{6.41}$$

Combining (6.17), (6.18) and (6.19), we have:

$$\left(\frac{m_2}{m_1^2}\right)_{int} = \frac{(\alpha - \beta)}{(1 - \beta)^2} > 1$$
(6.42)

# 6.6 Node with the Maximum Value of $(-m_1)$ Behaves as a Far Node

The Elmore delay value at node i, or the first moment of impulse response, is in the form of:

$$(-m_{Ii}) = \sum_{k} R_{ki} C_k \tag{6.43}$$

where  $m_{1i}$  represents the first moment associated with node *i*. Assume at a sink node *p*, where the Elmore delay is the largest in the tree, therefore:

$$(-m_{1p}) > (-m_{1k}) (k \neq p)$$
 (6.44)

The second moment of the node p can be computed as:

$$m_{2p} = \sum_{k}^{p} R_{kp} C_k (-m_{1k})$$
(6.45)

Using inequality in (6.44), we have:

$$m_{2p} < \sum_{k}^{p} R_{kp} C_k(-m_{1p})$$
 (6.46)

$$=(-m_{1p})\sum_{k}^{p}R_{kp}C_{k}$$
(6.47)

$$=m_{1p}^{2}$$
 (6.48)

Therefore:

$$\frac{m_{2p}}{m_{1p}^2} < 1 \tag{6.49}$$

## **7 REFERENCES**

- E.Acar, A.Odabasioglu, M.Celik, and L.T.Pileggi, "S2P: A stable 2-pole RC delay and coupling noise metric," Proceedings of Great lakes Symposium on VLSI, 1999.
- [2] C.J Alpert, A.Devgan, and C.V.Kashyap. "RC delay metric for performance optimization", IEEE Transactions on computer-aided design, vol.20 no.5, pp.571-582, May, 2001
- [3] C.J Alpert, F. Liu, C.Kashyap, and A.Devgan, "Delay and Slew Metrics Using the Lognormal Distribution", IEEE/ACM international conference on design automation, June, 2003, pp.382-385.
- [4] D.Anastasakis, N.Gopal, S.Y.Kim and L.T.Pileggi, "On the Stability of Approximations in Asymptotic Waveform Evaluation", Proceedings of the 29<sup>th</sup> ACM/IEEE Design Automation Conference, 1992, pp207-212.
- [5] M.Celik, L.Pileggi, A.Odabasioglu, "IC Interconnect Analysis", Kluwer Academic Publisher, 2002
- [6] C.Cheng, J.Lillis, S.Lin, N.H.Chang, "Interconnect Analysis and Synthesis", A Wiley-Interscience Publication, 2000
- [7] J.Cong "Interconnect-Centric Design Flow for Nanometer Technology," Proceedings of IEEE, vol.89, issue 4, pp.505-528, April, 2002
- [8] J.Cong, L.He, C.-k.Koh, and P.H.Madden, "Performance optimization of VLSI interconnect layout", Integration, the VLSI journal, vol21, pp1-94,1996
- [9] J.Cong, L.He, K.Khoo, C.Koh and Z.Pan, "Interconnect Design for Deep Submicron ICs", IEEE/ACM Int. Conference on Comupute-Aided Design, 1997, pp478-485

- [10] H.Cramer, "Mathematical Methods of Statistics", Princeton University Press, 1946
- [11] W.J.Dally, "Interconnect-limited VLSI architecture', Interconnect Technology, IEEE International Conference, May 1999, pp15-17
- [12] F.Dartu, N.menezes, and L.T.Pileggi, "Performance computation for precharacterized CMOS gate with RC loads," IEEE Transactions on Computer-Aided Design, vol.15, no.5, pp544-553, May,1996
- [13] F.Dartu, N.Menezes, J.Qian, and L.T.Pileggi, "A gate-delay model for high-speed CMOS circuits," in Proc. 31<sup>st</sup> ACM/IEEE Design Automation Conference, pp.576-580, 1994
- [14] W.C Elmore, "The Transient response of damped linear networks with practical regard to wideband amplifiers," Journal of Applied Physics Vol. 19, no.1 pp. 55-93, Jan. 1948.
- [15] R.Gupta, B. Tutuianu and L.T.Pileggi, "The Elmore delay as a bound for RC trees generalized input signals", IEEE Transactions on Computer-Aided Design,vol.16,no.1, January,1997, pp.95-104
- [16] M.Hafed, M.Oulmane, N.Rumin, "Delay and current estimation in a CMOS inverter with an RC load", IEEE Transactions on Computer-Aided Design, vol.20, no.1 Jan.2001 pp80-89.
- [17] R.H.Havemann, J.A. Hutchby, "High-performance interconnects: an integration overview," Proceedings of IEEE, vol.89, issue 5, May 2001, pp586-601.
- [18] C.W.Ho, A.E.Ruehli, and P.A.Brennan, "The modified nodal approach to network analysis," IEEE Transactions on Circuits and Systems, vol.CAS-22, pp.504-509, June, 1975.
- [19] R.Ho, K.W.Mai, and M.A.Horowitz, "The future of wires," Proceedings of the IEEE, vol.89, issue:4, pp.490-504, Apr.2001
- [20] R Ho, R.K.Mai, H.Kapadia, M.Horowitz, "Interconnect Scaling Implications for CAD," IEEE/ACM International Conference on Computer-Aided Design, 1999,

pp425-429.

- [21] M.A.Horowitz, "Timing models for MOS circuits," Ph.D thesis, Stanford University, 1980
- [22] X.Huang, V.Raghavan, R.A.Rohrer, "AWEsim: a program for the efficient analysis of linear(ized)circuits", 1990 IEEE international conference on computeraided design, Nov.1990, pp534-537.
- [23] A.B Kahng and S. Muddu, "Accurate Analytical Delay Models for VLSI Interconnects." Univ. California, Los Angeles, UCLA CS Dept. TR-950034, September 1995
- [24] H.Kapadia and M.Horowitz, "Using Partitioning to Help Convergence in the Standard Cell Design Automation Methodology," Proceedings of the 36<sup>th</sup> ACM/IEEE conference on design automation conference, June 1999, pp592-597
- [25] C.V.Kashyap, C.J.Alpert, F. Liu, and A.Devgan, "Closed Form Expressions for Extending Step Delay and Slew Metrics to Ramp Inputs," Proceedings of the 2003 international symposium on physical design, pp24-31, April, 2003
- [26] R.Kay and L.T.Pileggi "PRIMO: probability interpretation of moments for delay calculation", IEEE/ACM Design Automation Conference, 1998, pp.463-468
- [27] T.Lin, E.Acar, and L.T. Pileggi "H-gamma: an RC delay metric based on Gamma distribution approximation of the homogeneous response," IEEE/ACM international conference on Computer-aided design, 1998. pp.19-25
- [28] F.Liu, C.Kashyap and C.J.Alpert "A delay metric for RC Circuit based on the Weibull Distribution." IEEE/ACM international conference on Computer-aided design, 2002, pp620-624
- [29] F.Liu, C.Kashyap and C.J.Alpert "A delay metric for RC Circuit based on the Weibull Distribution." IEEE Trans. Computer-Aided Design, vol.23,no.3, pp443-447, March 2004
- [30] J.D.Meindl, "Beyond Moore's law: the interconnect era," Computing in Science and Engineering, vol 5, issue 1, pp20-24, Jan.-Feb, 2003

- [31] M.L.Mui, K.Banerjee, and A.Mehrotra, "A global interconnect optimization scheme for Nanometer scale VLSI with implications for latency, bandwidth, and power dissipation." IEEE Transactions on Electronic Devices, vol. 51, no.2, Feb. 2004 pp195-203.
- [32] Oulmane, M. and Rumin, N.C. "Accurate delay metric for on-chip resistive interconnect", European Solid State Device Research Conference, Florence, Italy, September 2002.
- [33] M. Oulmane, "On-chip global interconnect optimization", M.Eng Thesis, McGill University, Nov.2001
- [34] L.T.Pileggi, "Timing Metrics for Physical Design of Deep Submicron Technologies," International Symposium on Physical Design, April 1998, pp28-35
- [35] L.T.Pileggi, "Coping with RC(L) interconnect design headaches", IEEE/ACM International Conference on Computer-Aided Design, Nov. 1995 pp246-253.
- [36] L.T.Pillage, R.A.Rohrer "Asymptotic Waveform Evaluation for Timing Analysis", IEEE Transactions on Computer-Aided Design, Vol.9, No.4, pp.352-366, April 1990.
- [37] J.Qian, S.Pullela, and L.T.Pileggi, "Modeling the effective capacitance for the RC interconnect of CMOS gates," IEEE Transactions on Computer-Aided Design, vol.13, pp1526-1535, Dec.1994.
- [38] C.L.Ratzlaff and L.T.Pillage, "RICE: Rapid interconnect circuit evaluation using AWE,", IEEE Transactions on Computer-Aided Design, pp.763-776, Jun.1994
- [39] J. Rubinstein, P.Penfield and M. Horowitz, "Signal delay in RC networks", IEEE Transactions on Computer-Aided Design, July, pp202-211, 1983
- [40] D.Sylvester and K.Keutzer, "Getting to the bottom of Deep Submicron," Proceedings of the 1998 IEEE/ACM international conference on computer-aided design, Nov 1998, pp203-211.
- [41] The 2003 International Technology Roadmap for Semiconductors, Semiconductor Industry Association, san, Jose, California, 2003

[42] B. Tutuianu, F. Dartu, L.T.Pileggi, "An Explicit RC-Circuit Delay Approximation Based on the First Three Moments of the Impulse Response" Design Automation Conference, 1996, pp.611-616.