ELEN 689-603: Project Progress Report

Second Order Delay Computation for RC networks with Non-Tree Topology

Rajeshwary Tayade
000 07 1573
12/2/2003

1: Introduction

With DSM technologies, Interconnect delay is turning out to be a more dominating component of the total system delay. A lot of work has been devoted towards accurate and efficient computation of interconnect delay. The most popular interconnect model is the RC-π model which will be employed in this project. A comprehensive coverage of the timing metrics that are being traditionally used for delay estimation is given in [2]. These methods are summarized below:

1) **Elmore Delay:** Estimates the delay in terms of the step response of the equivalent RC circuit. Elmore observed that the step response of a RC circuit can be viewed as the CDF function of a random variable. Then the 50% delay point t_d corresponds to the median of the CDF function. The Elmore delay approximates the median to the mean of the distribution and is given by

$$T_D = G^{-1}C \qquad (1)$$

where G and C are the path conductance matrix and node capacitance vector respectively.

Thus for a generalized network, Elmore delay computation requires a single matrix inversion. In [1], it was shown that for RC-tree topologies, Elmore delay can be computed using simple path tracing algorithm instead of a matrix inversion. In [12] it was proved that Elmore delay forms the upper bounds on the delay for both ramp and step input signal. Due to its simplicity Elmore delay has been the most popular metric for interconnect delay estimation for years. The main problem of Elmore delay is that it neglects the resistive shielding effect [2] for long nets. Thus the Elmore delay can be seen as a good bound, but the estimates obtained are good only for some regular-structure circuits such as clock tree.

2) **AWE**: Automatic Waveform Evaluation [3] is a highly accurate but computationally intensive method, in which the system transfer function is first obtained using the system moments. Each moment computation requires just the DC-analysis of the equivalent RC-circuit and thus the transfer function of any order can be obtained. Once the transfer function is obtained, the system response and delay can be observed for any generalized input function. (Note that moment computation for a RC-tree topology is simple, but for non-tree topologies it requires matrix inversion.)

3) **Model Order Reduction**: These methods try to achieve a trade-off between the accuracy of AWE and the simplicity of Elmore delay by approximating 2^{nd} or 3^{rd} order systems as in [4] [5]. The problem with these methods that it system function obtained can be unstable and/or can produce complex poles. Also after obtaining the system impulse response, non-linear operations have to be used to determine the system delay.

4) **Probability Interpretation**: This method is a continuation of Elmore's concept of viewing the system impulse response as a PDF of a statistical distribution. The main idea is that we want to find the median of the step response. So we try to match the system step-response to some well-known CDF and the problem of delay calculation now translates into the problem of computing the median of this distribution. The matching in turn is done by equating the first three central moments of the distribution (mean, variance and skewness) to the first three circuit moments. This approach was proposed in [6] in which a time-shifted incomplete Gamma function was used for modeling. In [7] this same modeling is extended to match the homogenous portion of the step response to the gamma distribution. In [12] the modeling was further simplified by using the Weibull distribution for matching; the advantage being that the median of the Weibull distribution is a direct function of its parameters. These methods involves non-linear mapping and hence is based on look-up-table evaluation. One more advantage of this method is that after the system moments are obtained, delay calculation is independent of the circuit topology (no instability issues) which makes this method a better choice for generalized topology.

In this project the objective is to obtain the second-order delay estimates for a generalized RC-mesh (i.e. tree or non-tree). All capacitances are grounded and there are no resistive paths to ground. As mentioned above a tree structure provides a huge computational gain and hence most methods are focused towards solving for tree structures. In this project we use the Tree-Link partitioning algorithm to calculate the first two moments of the non-tree circuit and then use the WED (Weibull based delay) approximation in [12] to compute the second-order delay.

The report is organized as follows: Section 2 gives describes the Tree-link Partitioning method and how it has been implemented in the project. Section 3 describes the Weibull Distribution and WED method. Section 4 gives the complete procedure in brief and the implementation details, problems faced etc. Section 5 describes the examples on which the method was implemented and section 6 summarizes the project.

2: Non-Tree Topology and Tree-Link Partitioning

As mentioned in the first section, most delay estimation methods need to compute the circuit moments. Each moment calculation can be represented as a simple dc-analysis procedure and hence can be solved using matrix inversion. For a RC Tree network, i.e. a topology where there is a unique path from any node to the source, we need not calculate the matrix inverse. The solution is obtained using a simple tree-traversal algorithm [1] in *O(n)* time-complexity. However consider a general RC network also called a RC mesh:

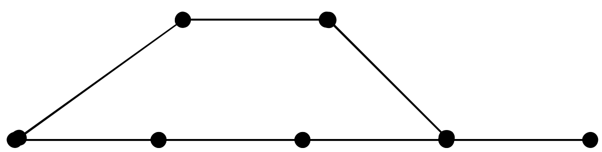


Figure 1: Non Tree-circuit topology

Here there are two paths from source *S* to node *B*, namely *S-A-B* and *S-E-F-F-B*. Thus if we want to calculate the delay at node *B*, it is not clear what links to use for the load-side capacitance and which ones to use for the driver-side Resistance values. For such cases we cannot use simple tree traversal procedure and might need to use a direct matrix-inverse procedure for delay calculation.

A simple solution for this problem is proposed in [9], called a Tree-link Partitioning algorithm. The basic idea is to first obtain a spanning tree for the given network using Depth-First-Search and use the edges not part of the tree as links. The node voltages (in our case moments) are computed for the spanning tree first using the tree-traversal procedure and then each link is added while updating the node voltages using Large Change Sensitivity analysis. The Tree/Link Partitioning method proposed in [9] obtains the first order delay (Elmore) as shown below:

$$t_{di} = \frac{\sum_{j=1}^{n} R_{i,j} C_j [v_i(\infty) - v_i(0)]}{v_i(\infty) - v_i(0)} \qquad (2)$$

Now using Large Change sensitivity if the node voltages (which correspond to the moments in our case) of a circuit are known, then the result of adding a resistive branch is given by:

$$\hat{v} = v - \frac{v_k - v_l}{R + R_{th}} Y^{-1} x_{kl} \qquad (3)$$

Assuming the initial and final voltages are same in both the cases, we obtain:

$$\hat{v}(\infty) - \hat{v}(0) = v(\infty) - v(0) - \frac{(v_k(\infty) - v_k(t) - (v_l(\infty) - v_l(t)))}{R + R_{th}} Y^{-1} x_{kl} \qquad (4)$$

Thus the Elmore delay for the non-tree circuit becomes:

$$\hat{t}_d = t_d - \frac{(t_{d,k} - t_{d,l})}{(R + R_{th})} Y^{-1} x_{kl} \qquad (5)$$

We need to estimate the second order delays for higher accuracy and that would require both first and second moments. The details of the method used for moment calculation is given in section 4.

3: Weibull Based Delay (WED)

The PDF of a 2-parameter Weibull distribution is given by:

$$P_{WEI}(t) = ab^{-a} t^{a-1} e^{-(t/b)} \text{ for t,}\alpha\text{,}\beta > 0 \qquad (6)$$

The CDF is given by

$$C_{WEI}(t) = 1 - e^{-(t/b)^a} \qquad (7)$$

Here α and β are the two parameters, α is called the shape parameter and β is the location parameter. Figure 2 shows plots of Weibull Distribution for different values of α and same β. The idea is to match the step-response of the circuit to the CDF of a Weibull distribution. This can be done by matching the statistical moments of the Weibull CDF to the circuit moments. The first two moments (namely the mean and variance) of the Weibull distribution is given by:

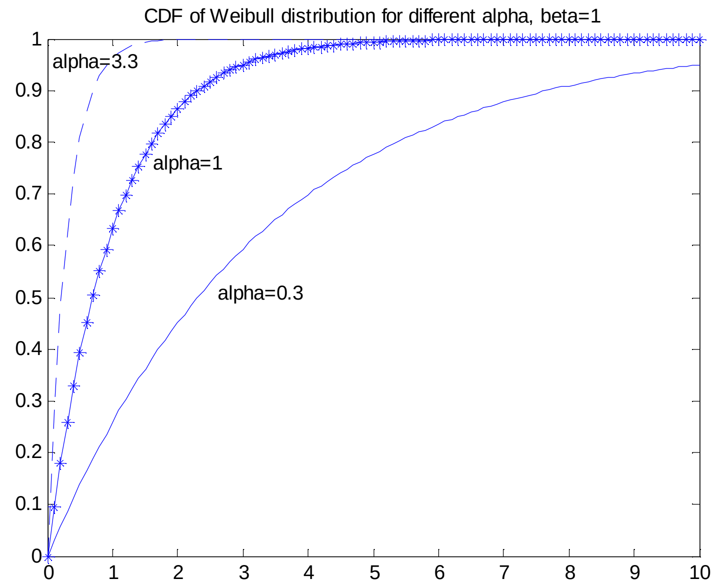
$$m_{WEI} = b\Gamma(1+q) \text{ and } s^2{}_{WEI} = b^2(\Gamma(1+2q) - \Gamma^2(1+q)) \qquad (8)$$

where $\theta = 1/a$

Given the circuit moments *m1* and *m2*, the central moments (mean and variance) are obtained as

$$m = -m_1 \text{ and } s^2 = 2m_2 - m_1{}^2 \qquad (9)$$

Now, we have $\frac{s^2{}_{WEI}}{m^2{}_{WEI}} = \frac{\Gamma(1+2q)}{\Gamma^2(1+q)} - 1,$ (10)



Then Matching the Circuit mean and variance to that of the Weibull distribution, we obtain:

$$\frac{\Gamma(1+2q)}{\Gamma^2(1+q)} = \frac{2m_2}{m_1{}^2}; \tag{11}$$

Now, Let $r = \frac{m_2}{m_1{}^2} = \frac{1}{2}\frac{\Gamma(1+2q)}{\Gamma^2(1+q)}$ (12)

Then Given the circuit moments, the value of θ can be obtained using the inverse Gamma function. Since this is a non-linear relation and is complicated to solve, it is implemented using a simple Look-up Table. After θ is obtained, then from the mean matching, we get

$$b = \frac{-m_1}{\Gamma(1+q)} \tag{13}$$

The 50% delay point is then obtained using the formula for the median of the Weibull distribution:

$$t_{0.5} = b(\ln(2))^q \tag{14}$$

Thus the main advantage of using this method is that once the two circuit moments are obtained at each node, a simple algorithm (described later) can be used to get the delay irrespective of the circuit topology.

4: Procedure

In the above two sections we have discussed two different papers/concepts. The first one is a method to obtain the first order delay for a non-tree RC circuit using Tree-Link partitioning, and the second one is a procedure to obtain circuit delay using Weibull distribution approximation of the step-response. In this project, we combine the two ideas to get the second-order delay of a generalized RC circuit.

The WED method requires the first two circuit moments and after that the delay calculation is just based on Look-up Tables. So the main issue is to compute the moments of the RC-non-tree circuit correctly. As mentioned in section 2, we first get a spanning tree of the circuit and calculate the first and second order moments for this tree-circuit. Let the four moments be denoted as $m_1^{tree}, m_2^{tree}, m_1^{non-tree}, m_2^{non-tree}$. From equation (5) it is clear that the first order moment for the non-tree can be easily computed by adding one link at a time and using simple large-change-sensitivity analysis.

Thus the first moment for node *i* after adding a link *k-l* is given by,

$$m_{1,i}^{non-tree} = m_{1,i}^{tree} - \frac{m_{1,k}^{tree} - m_{1,l}^{tree}}{R + R_{th}} Y^{-1} x_{kl} \qquad (15)$$

The question now was to choose between the following two methods for computing $m_2^{non-tree}$

1) **Method 1:**

$$\vec{v}(s) = \vec{m}_0 + \vec{m}_1 s + \vec{m}_2 s^2 + \mathbf{K}$$

$$\vec{\hat{m}}_1 s + \vec{\hat{m}}_2 s^2 = \vec{m}_1 s - \frac{m_1^k - m_1^l}{R + R_{th}} s Y^{-1} x_{kl} + \vec{m}_2 s^2 - \frac{m_2^k - m_2^l}{R + R_{th}} s^2 Y^{-1} x_{kl}$$

Thus $\quad \vec{\hat{m}}_1 = \vec{m}_1 - \dfrac{m_1^k - m_1^l}{R + R_{th}} Y^{-1} x_{kl}$ and $\vec{\hat{m}}_2 = \vec{m}_2 - \dfrac{m_2^k - m_2^l}{R + R_{th}} Y^{-1} x_{kl}$

This method was presented in the progress-report and it makes an underlying assumption that the large-change-sensitivity analysis is applicable to ac signal. The method is shown in Figure 2a.

2) **Method 2:** Figure (2b) shows the second method used. We know that value of m_2 depends on that of m_1. In the first method the m_2^{tree} is computed using the m_1^{tree} values, while the second method uses the new values i.e. $m_1^{non-tree}$ values.

The results in section 5 shows the delay values obtained using both the methods.

$$\begin{tikzcd} m_1^{tree} \arrow[r] \arrow[d] & m_2^{tree} \arrow[d] \\ m_1^{non-tree} & m_2^{non-tree} \end{tikzcd}$$

Figure (2a): Method 1 to compute second-order non-tree moment

$$m_1^{tree} \longrightarrow m_1^{non-tree} \longrightarrow m_2^{tree} \longrightarrow m_2^{non-tree}$$

Figure (2b): Method 2 to compute the second-order non-tree moment

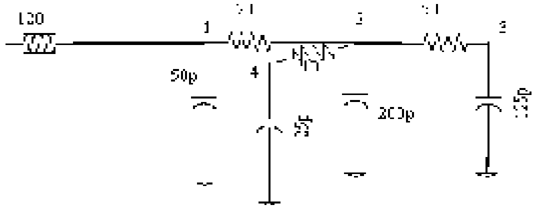
The entire procedure requires the following steps:

1) Given an input interconnect geometry in rectilinear space; create a RC-circuit using the pi-model. In my program, the input was taken as a file and each node object knows its node capacitance and connections to other nodes. Note that in rectilinear space, each node has only 4 possible connections, namely left, top, right and bottom. The input file also has the total number of nodes, the unit r and unit c values and the driver resistance value.
2) This collection of node now forms a graph. Perform DFS on this graph with the source node as the root. The DFS traverses only certain edges which form the DFS tree (or spanning tree). The rest of the edges are recorded as the 'Links' of the graph.
3) For the above obtained DFS tree, calculate the moments for each node. Note that the node capacitance for the tree nodes is the same as the node capacitance of the original (non-tree) circuit. The first moments for the Tree-circuit were calculated using simple path traversal procedure. The moments obtained for each node of the DFS tree are then updated by adding one link at a time. The first moment is updated directly using equation (15). The second moments are computed using both Method1 and Method 2 as explained above.
4) The basic algorithm for WED is as follows:
    - Calculate r = m2/m1^2
    - Use a look-up table to find θ from log10(r)
    - Use a look-up table to obtain Γ(1 + θ) from θ
    - Let β = -m1/(Γ(1 + θ))
    - The 50% delay td = β(ln2)^ θ
5) The Look-up table for finding θ was taken directly from [12], while that for obtaining Γ(1+ θ) was generated using MATLAB. First the WED algorithm was tested for Tree circuits and then was used for Non-tree circuits. In the examples below, circuit 1, 2 are from the homework problems and the results show that the delay obtained using the WED method is within acceptable range of the spice results. Circuits 3-6 are arbitrarily chosen non-tree topology circuits.

5: Results

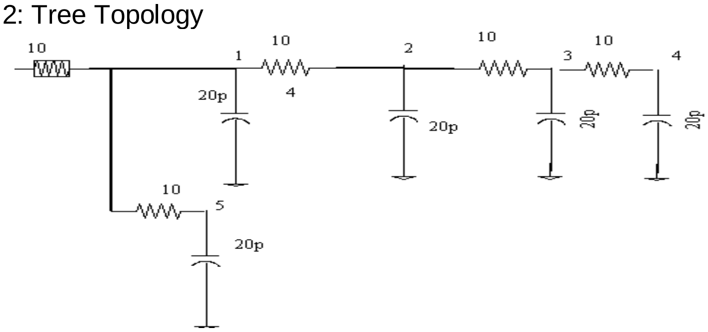
Here we show the actual RC circuits instead of the interconnect geometry.

1) Circuit 1: Tree Topology



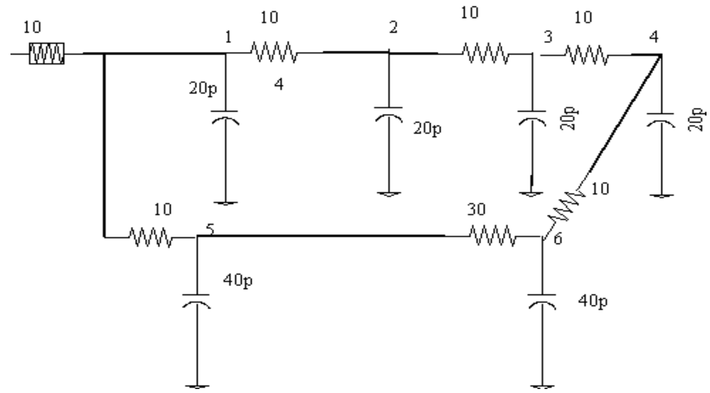
Node	Spice	WED	% error
1	2.97619e-8	2.78199e-8	6.53
2	3.7619e-8	3.81231e-8	1.34
3	4.41905e-8	4.24553e-8	3.93
4	3.77619e-8	3.82964e-8	1.42

2) Circuit 2: Tree Topology



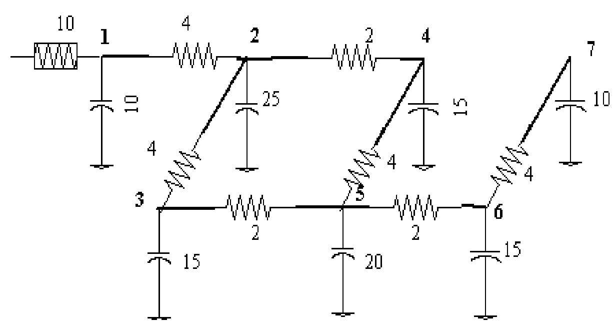
Node	Spice	WED	% error
1	3.9626e-10	4.75329e-10	19.95
2	1.01887e-9	1.10904e-9	8.85
3	1.47467e-9	1.38629e-9	5.99
4	1.6909e-9	1.78763e-9	5.72
5	6.3477e-9	6.95497e-10	8.94

3) Circuit 3: Non-tree, Links : (1-5)



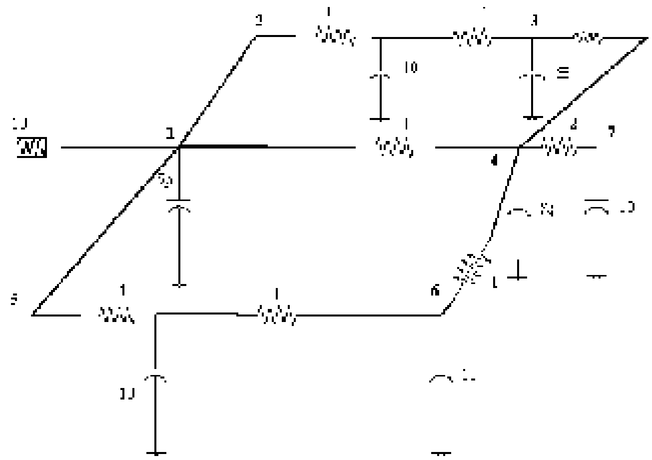
Node	Spice	Program 1	% error	Program 2	%error
1	5.34848e-10	3.94606e-10	26.22	7.838e-10	46.55
2	1.375e-9	9.07273e-10	34.02	1.3736e-9	0.1
3	1.96154e-9	1.09868e-9	43.99	1.98933e-9	1.42
4	2.27885e-9	1.50679e-9	33.88	2.19728e-9	3.58
5	1.36538e-9	7.18434e-10	47.38	1.37361e-9	0.6
6	2.39423e-9	1.55433e-9	35.08	2.65706e-9	10.98

4) Circuit 4: Non-Tree, Links: (2-3)



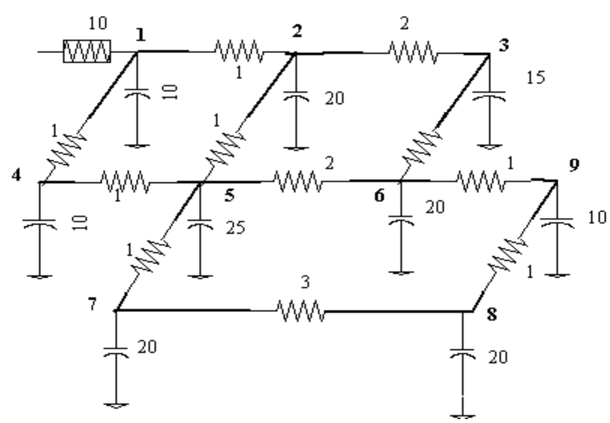
Node	Spice	Program 1	% error	Program 2	% error
2	4.95455e-10	5.39499e-10	8.89	5.39499e-10	8.89
3	9.9733e-10	8.89656e-10	10.80	1.06398e-9	6.68
4	1.15e-9	1.16102e-9	9.6	1.16102e-9	9.6
5	1.09394e-9	1.11943e-9	2.33	1.11943e-9	2.33
6	1.19467e-9	1.188875e-9	0.49	1.188875e-9	0.49
7	1.25472e-9	1.2234e-9	2.5	1.2234e-9	2.5
8	1.2924e-9	1.25113e-9	3.19	1.25113e-9	3.19

5) Circuit 5: Links (1-4), (1-5)



Node	Spice	Program 1	% error	Program 2	% error
2	6.13208e-10	5.04235e-10	17.77	6.23832e-10	1.73
3	6.00377e-10	5.09306e-10	15.17	6.28165e-10	4.63
4	6.79245e-10	5.08582e-10	25.13	6.25565e-10	7.90
5	6.98113e-10	5.02062e-10	28.08	6.16035e-10	11.76
6	6.69811e-10	5.09306e-10	23.96	6.28165e-10	6.22
7	6.79245e-10	5.08582e-10	25.13	6.25565e-10	7.90
8	7.16981e-10	5.13653e-10	28.36	6.2989e-10	12.15

6) Circuit 6: Links (1-4) ,(2-5), (6-9)



Node	Spice	Program 1	% error	Program 2	% error
2	2.0384e-9	1.79784e-9	11.80	2.13111e-9	4.55
3	2.13462e-9	1.81988e-9	14.74	2.17648e-9	1.96
4	2.19231e-9	1.81896e-9	17.03	2.19161e-9	0.03

5	2.09615e-9	1.70365e-9	18.72	2.0929e-9	0.16
6	2.14423e-9	1.75695e-9	18.06	2.13773e-9	0.30
7	2.19231e-9	1.80981e-9	17.45	2.18877e-9	0.16
8	2.1923e-9	1.78025e-9	18.80	2.16356e-9	1.31
9	2.23077e-9	1.81537e-9	18.62	2.19948e-9	1.40
10	2.33e-9	1.81549e-9	22.08	2.19759e-9	5.68

5: Summary and Conclusion

From the results it is clear that the delay computation using Method 1 have really high percentage error. Method 2 gives considerably accurate results except of a couple of cases which can be thought of as outliers. Thus we can safely conclude that Method 2 should be used for higher order moment calculation using the Tree-Link Partitioning algorithm. In summary second-order delay for RC-non tree circuits can be obtained using the Tree-link Partitioning Algorithm along with the Weibull based delay approximation with quite good accuracy. The computation time is considerably low since we do a simple path tracing for moment calculation and use look-up tables for delay calculation. This is a huge saving as compared to the model-order reduction techniques that requires non-linear solution methods and may run into stability issues.

Reference:

[1] Paul Penfield and Jorge Rubinstein, "Signal Delay in RC Tree Networks", IEEE Trans. on Computer-Aided Design, CAD-2, (July 1983), pp. 202-211.

[2] L. Pileggi, "Timing Metrics for Physical Design of Deep Sub Micron Technologies" , Int'l Symposium on Physical Design, April 1998

[3] L. T. Pillage and R.A. Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis", IEEE Transactions on Computer Aided Design, 9(4), pp. 352-366, 1990.

[4] K. J. Kerns, I. L. Wemple, A. T. Yang, "Stable and Efficient Reduction of Substrate Model Network using Congruence Transformation", ICCAD, Nov. 1995.

[5] B. Tutuianu, F. Dartu and L. Pileggi, "An Explicit RC-Circuit Delay Approximation Based on the First Three Moments of the Impulse Response", Proc. ACM/IEEE Design Automation Conf., 1996.

[6] Rony Kay and Lawrence Pileggi , "PRIMO: Probability Interpretation of Moments for Delay Calculation", *Proc. IEEE/ACM Design Automation Conference, 1998*

[7] Tao-Lin, Emrah Acar, L. Pileggi, "h-gamma: An RC Delay Metric Based on a Gamma Distribution Approximation of the Homogeneous Response", *Proc. IEEE Transactions on Computer-Aided Design*, 16:1507-1514, December 1997

[8] Pak K. Chan and Martine Schlag, "Bounds on Signal Delay in RC Mesh Network", *Proc. IEEE Transactions on Computer-Aided Design*, Vol 8. No. 6, June 1989

[9] Pak K. Chan and Kevin Karplus, "Computing Signal Delay in General RC networks by Tree/Link Partitioning", *Proc. IEEE Transactions on Computer-Aided Design*, Vol 9. No. 8, June 1990

[10] Lin, Tzu-Mu and Mead, Carver A. (1983) Signal Delay in General RC Networks with Application to Timing Simulation of Digital Integrated Circuits. http://resolver.caltech.edu/CaltechCSTR:1983.5089-tr-83

[11] R. Gupta, B. Tutuianu and L. Pileggi, "The Elmore Delay as a Bound for RC Trees Generalized input signals" *Proc. IEEE Transactions on Computer-Aided Design*, Vol 16. No. 1, June 1997

[12] F. Liu , C. V. Kashyap and C. J. Alpert, "A Delay Metric for RC Circuits Based on the Weibull Distribution", ICCAD 2002