Delay Modelling for Buffered RLC/RLY Trees *

Sheng-Lung Wang and Yao-Wen Chang

Department of Electrical Engineering & Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan

Abstract

For deep-submicron, high-performance circuits, the inductive effect plays a very important role in determining the circuit delay. In this paper, we derive accurate formulae for modeling the delays of buffered RLY/RLC wires and trees. *Our formulae can handle balanced and un-balanced trees and consider buffer insertion. Extensive simulations with HSPICE show that the formulae have high fidelity, with an average error of within 5.51% based on the 180 nm technology. The simulations show that our formulae are more accurate than previous works.*

1 Introduction

As technology advances into the very deep-submicron era, interconnect delay dominates overall circuit performance. Therefore, accurately modeling the interconnect delay becomes a major challenge in high performance IC design. For deep-submicron, high-performance circuits, ignoring inductance effects may incur a large amount of error, since an RC model as compared to an RLC model may create errors of up to 30% in the total propagation delay of a repeater system [9]. As technology improves and die size increases, short rise/fall times of signals and long wires make inductive effects more significant than before [15]. Therefore, it is very important to consider the effects of inductance.

Timing is a crucial concern in the design of high-performance circuits. Arunachalam et al. in [3] proposed accurate CMOS gate delay models for general RLC loads. The waveform resulted from their delay model excellently agrees with SPICE results; however, they do not present any formula for propagation delay. Many delay models have been proposed to calculate delay (e.g., [4, 5, 9, 14, 18]); however, these models cannot apply to tree structures. Modeling and analysis techniques for timing calculation under tree structures have been studied extensively in the literature [1, 2, 7, 8, 10, 11, 12]. Previous work in [1] proposed a method (Fitted Elmore Delay) for delay estimation by using the curve fitting technique. However, their work does not consider inductance. The work in [2] only considered the RC delay model, and did not include the inductance effect. The works in [7, 8] extended the Elmore delay to include the inductance effect, but they did not consider buffer insertion/sizing. Ismail and Friedman in [10] proposed an algorithm for buffer insertion/sizing in an RLC tree. However, if the tree is unbalanced, as pointed out in the paper, the delay estimation may incur significantly larger errors. The works in [11, 12] adopted two-pole simulation of interconnect trees via the moment matching technique, and used non-uniform lumped segments to model the distributed lines. However, they did not apply buffer insertion/sizing to reduce the delay. Kahng and Muddu in [13] provided an analytic delay model for interconnection lines under the step input, and extended their model to estimate the delay in arbitrary interconnect trees. However, their model does not consider buffer insertion/sizing, and cannot calculate for any percentage of delay time. Ismail and Friedman in [10] presented an algorithm to insert and size buffers in an RLC tree for minimizing the delay. However, their empirical formulae obtained by curve-fitting with circuit simulation were only for the 50% propagation delay and the 10%–90% rise time. Therefore, their works cannot treat any percentage of delay time. Banerjee in [4] considered buffer insertion/sizing for an RLC interconnection line and did not handle the problem with the tree structure. Table 1 compares the features of important related works.

	Buffer Insertion/Sizing	Interconnection Trees	Calculation for any percentage delay
[4]	√		√
[10]	√	√	
[13]		√	
Our Work	√	√	√

Table 1: Comparison of features with the related previous works.

In this paper, we derive accurate formulae for modeling the delays of *buffered RLY/RLC wires and trees.* The RLY model not only can model RLC interconnect, but also can consider off-path subtree effects. Our formulae can handle balanced and un-balanced trees and consider buffer insertion. Extensive simulations with HSPICE show that the formulae have high fidelity, with an average error of within 5.51% based on the 180 nm technology. The simulations show that our formulae are more accurate than related previous works.

The remainder of this paper is organized as follows. Section 2 introduces some notations. Section 3 gives the delay models for RLY/RLC wires and trees. Section 4 shows the experimental results, and finally concluding remarks are given in Section 5.

2 Preliminaries

We use the following notations throughout this paper.

- h_i: the length of wire i.
- r: the unit-length resistance of a wire.
- l: the unit-length inductance of a wire.
- c: the unit-length capacitance of a wire.
- c_b: the input capacitance of a minimum sized buffer.
- r_b: the output resistance of a minimum sized buffer.
- c_p: the output parasitic capacitance of a minimum sized buffer.
- k: the size of a buffer.
- R_S: the resistance of the driver.
- C_L: the capacitance of the load.

3 Accurate Delay Model

3.1 Delay Model for RLY Wires

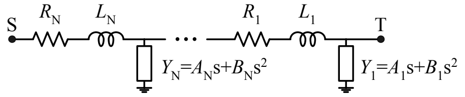

Figure 1: An equivalent single "RLY" line for an RLC tree.

Sriram and Kang in [16] developed an equivalent single RLY line (see Figure 1) for an RLC tree, where N is the number of RLY sections in an RLC tree. As shown in Figure 2, we can model a distributed RLC line of length h_i as a single RLY segment. The transfer function for the structure of Figure 1 is given by

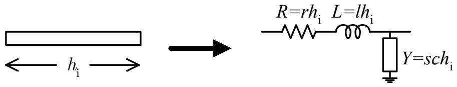

Figure 2: An interconnect wire can be modeled as a RLY segment.

$$H(s) = \frac{V_O(s)}{V_I(s)} = \frac{1}{1+b_1s+b_2s^2+\ldots}. \quad (1)$$

By the approximation method proposed by Gao et al. in [7], we can approximate Equation (1) as follows:

$$H(s) \approx \frac{1}{1+b_1s+b_2s^2},$$

where

$$\begin{aligned} b_1 &= \sum_{j=1}^{N} A_j \sum_{i=j}^{N} R_i, \\ b_2 &= \sum_{j=1}^{N} A_j \sum_{l=j}^{N} L_l + \sum_{j=1}^{N} B_j \sum_{l=j}^{N} R_l \\ &+ \sum_{j=2}^{N} A_j \sum_{l=j}^{N} R_l \sum_{i=1}^{j-1} A_i \sum_{d=i}^{j-1} R_d. \end{aligned}$$

The first and second moments of the transfer function from Equation (1) can be obtained by the coefficients b_1 and b_2, i.e., $M_1 = b_1$ and $M_2 = {b_1}^2 - b_2$. The two poles s_1 and s_2 of the transfer function could be real or complex depending on the sign of $({b_1}^2 - 4b_2)$. Thus, we separately discuss the results from two poles response for each of these cases classified in [13].

* Yao-Wen Chang's work was partially supported by the National Science Council of Taiwan under Grant No's. NSC 92-2215-E-002-016 and NSC 93-2752-E-002-008-PAE.

Case I. Real Poles: The condition for this case is $({b_1}^2 - 4b_2) > 0$. The step response, which is the inverse Laplace transform of $\frac{1}{s}H(s)$, is given by

$$v(t) \quad = \quad V_O(1 - \frac{s_2}{s_2 - s_1}e^{s_1 t} + \frac{s_1}{s_2 - s_1}e^{s_2 t}),$$

where

$$s_{1,2} \quad = \quad \frac{-b_1 \pm \sqrt{{b_1}^2 - 4b_2}}{2b_2}.$$

For a step input, the $f \times 100\%$, (where $0 \leq f < 1$) delay, τ, (i.e., $v(\tau) = fV_O$) is the solution of the following equation [4]:

$$1 - f - \frac{s_2}{s_2 - s_1}e^{s_1 \tau} + \frac{s_1}{s_2 - s_1}e^{s_2 \tau} \quad = \quad 0. \tag{2}$$

Therefore, we can use the Newton-Raphson method to numerically solve the delay that was calculated by Equation (2).

Case II. Complex Poles: The condition for this case is $({b_1}^2 - 4b_2) < 0$. The time-domain response for this case is given by

$$v(t) \quad = \quad V_O(1 - \sqrt{1 + (\frac{\alpha}{\beta})^2}e^{-\alpha t}\sin(\beta t + \rho)),$$

where

$$\begin{aligned} \alpha &= \frac{M_1}{2({M_1}^2 - M_2)}, \\ \beta &= \frac{\sqrt{3{M_1}^2 - 4M_2}}{2({M_1}^2 - M_2)}, \\ \rho &= \tan^{-1}(\frac{\beta}{\alpha}). \end{aligned}$$

We consider a step input. Thus, $v(\tau) = fV_O$ is the solution of the following equation.

$$1 - f - \sqrt{1 + (\frac{\alpha}{\beta})^2}e^{-\alpha t}\sin{(\beta t + \rho)} \quad = \quad 0. \tag{3}$$

Similarly, we also use the Newton-Raphson method to solve the delay that was calculated by Equation (3).

Case III. Double Poles: The condition for this case is $({b_1}^2 - 4b_2) = 0$. The time-domain response is given by

$$v(t) \quad = \quad V_O(1 - e^{s_1 t} - \frac{2t}{b_1}e^{s_1 t}), \tag{4}$$

where

$$s_1 \quad = \quad -\frac{b_1}{2b_2}.$$

Similarly, the Newton-Raphson method can be applied to calculate the delay that was calculated by Equation (4).

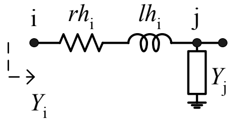

Figure 3: Computation of Subtree Admittance

For a series section of RLC segments, Kahng and Muddu in [12] presented an expression for the coefficient of s and s^2 of admittance. As shown in Figure 3, the admittance at node i can be expressed in terms of the admittance at node j.

$$Y_i \quad = \quad \frac{1}{rh_i + slh_i + \frac{1}{Y_j}} = Y_j - Y_j^2 rh_i - slh_i Y_j^2 + \ldots \tag{5}$$

Using the above recursive equation, the admittance of the off-path subtrees can be computed.

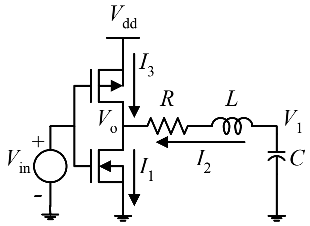

Figure 4: A CMOS inverter drives an RLC load.

3.2 Delay Model for Buffered RLC Load

A CMOS inverter driving an RLC load is shown in Figure 4. For an interconnect wire of length h_i, its total resistance is $R = rh_i$, total inductance is $L = lh_i$, and total capacitance is $C = ch_i$, where r, l, and c are the resistance, inductance, and capacitance per unit length of the interconnect, respectively. To consider the velocity saturation effects in short-channel devices, a CMOS inverter is modeled by using the alpha power law [17]. V_o and V_1 are the output voltage of the CMOS inverter and the output voltage at the end of the interconnect wire, respectively. The input voltage V_{in} is a fast ramp signal that can be approximated by a step signal:

$$V_{in}(t) = \quad \tfrac{t}{\tau_r}V_{dd} \quad \text{for } 0 \leq t \leq \tau_r,$$

where τ_r is the input transition time.

Because V_o and V_1 depend on V_{in} and the operation region of NMOS transistor, we separately discuss three different conditions in the following [18]:

Case I. $\tau_n \leq t \leq \tau_r$**:** The NMOS transistor is ON and operates in the saturation region. We have the following equations:

$$\begin{aligned} V_1(t) &= L\frac{dI_{DS}}{dt} + RI_{DS} + V_o(t), \\ C\frac{dV_1(t)}{dt} &= -I_{DS} = -B_n(\frac{t}{\tau_r}V_{dd} - V_T). \end{aligned}$$

Therefore, the solution of $V_o(t)$ is

$$\begin{aligned} V_o(t) &= V_{dd} - V_c(t) - V_r(t) - V_l(t), \\ V_r(t) &= RB_n(\frac{t}{\tau_r}V_{dd} - V_T), \\ V_l(t) &= LB_n\frac{V_{dd}}{\tau_r}, \\ V_c(t) &= \frac{B_n\tau_r}{2CV_{dd}}(\frac{t}{\tau_r}V_{dd} - V_T)^2, \end{aligned}$$

where V_T is the switching threshold voltage and τ_n is the time for V_{in} to reach V_T.

Case II. $\tau_r \leq t \leq \tau_{nsat}$**:** V_{in} is fixed at V_{dd} and the NMOS transistor continues to operate in the saturation region. The discharge current is equal to the saturated drain-to-source current of the NMOS transistor:

$$I_{DS} \quad = \quad I_{nsat} = B_n(V_{dd} - V_T) = constant.$$

Therefore,

$$\begin{aligned} V_o(t) &= V_1(\tau_r) - RB_n(V_{dd} - V_T) \\ &\quad - \frac{B_n}{C}(V_{dd} - V_T)(t - \tau_r), \end{aligned} \tag{6}$$

where

$$V_1(\tau_r) \quad = \quad V_{dd} - \frac{B_n\tau_r}{2CV_{dd}}(V_{dd} - V_T)^2,$$

τ_{nsat} is the time when the NMOS transistor leaves the saturation region, V_{nsat} is the drain saturation voltage and is usually around $0.7V_{dd}$ in short-channel devices [14]. As $V_o = V_{nsat}$, $t = \tau_{nsat}$, where τ_{nsat} is determined from Equation (6). Therefore, we have

$$\begin{aligned} \tau_{nsat} &= \tau_r \\ &+ \frac{C}{I_{D0}}\left(V_{dd} - V_{nsat} - \frac{\tau_r I_{D0}}{2CV_{dd}}(V_{dd} - V_T) - RI_{D0}\right). \end{aligned}$$

Case III. $t \geq \tau_{nsat}$**:** After V_o drops below V_{nsat}, the NMOS transistor enters the linear region:

$$\begin{aligned} C\frac{dV_1(t)}{dt} &= -\gamma_n V_{DS} = -\gamma_n V_o, \\ V_1(t) &= L\frac{dI_{DS}}{dt} + RI_{DS} + V_o, \end{aligned}$$

where γ_n is the effective output conductance.

Therefore,

$$V_o = K_1 e^{-\alpha_1 t} + K_2 e^{-\alpha_2 t},$$

where

$$\alpha_{1,2} = \frac{\frac{1+R\gamma_n}{Lr_n} \pm \sqrt{(\frac{1+R\gamma_n}{L\gamma_n})^2 - \frac{4}{LC}}}{2},$$

K_1 and K_2 can be determined from $V_o(\tau_{nsat})$ an $V_o^{'}(\tau_{nsat})$. Because α_1 is typically much greater than α_2, we have

$$V_o = V_{nsat} e^{-\alpha_2(t-\tau_{nsat})}.$$

Therefore, the propagation delay time (50%) of a CMOS inverter is

$$t_{0.5} = \frac{1}{\alpha_2} \ln \frac{2V_{nsat}}{V_{dd}} + \tau_{nsat} - \frac{\tau_r}{2}.$$

3.3 Delay Model for Buffered RLC Trees

In this section, we extend our delay model to handle arbitrary balanced and un-balanced buffered RLC trees. For instance, consider an un-balanced buffered RLC tree with a root (or a source) and a set of leaves (or sinks) as shown in Figure 5. The buffer is inserted in an arbitrary location of the tree. Our delay model not only can handle different wire lengths but also can compute any percentage of delay time. Suppose we are to compute the delay from the source S to node 7 (critical path) in Figure 5. Buffer insertion divides the path into three stages. The path can be represented by the equivalent circuit shown in Figure 6. In order to calculate the delay time of stage 2, we show the equivalent circuits of Figure 6 (b) in Figure 7. We apply the method presented in Sections 3.1 and 3.2. Assume that the delay times of stage 1, stage 2, and stage 3 are τ_1, τ_2, and τ_3, respectively. The total delay between the source S and node 7 is

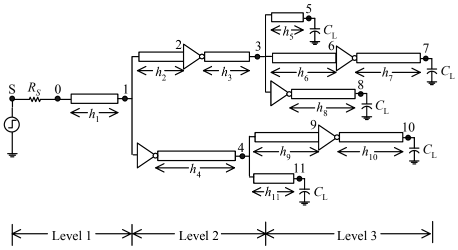

Figure 5: An un-balanced buffered RLC tree.

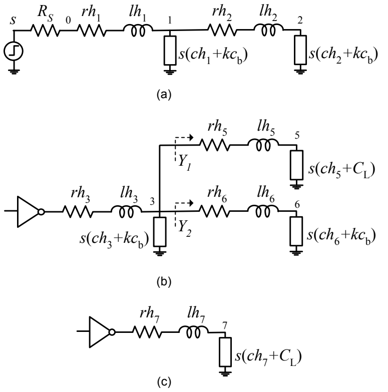

Figure 6: Representation of the path between the source S and node 7. (a) Stage 1 of the path; (b) Stage 2 of the path; (c) Stage 3 of the path.

$$\tau_t = \sum_{i=1}^{3} \tau_i. \quad (7)$$

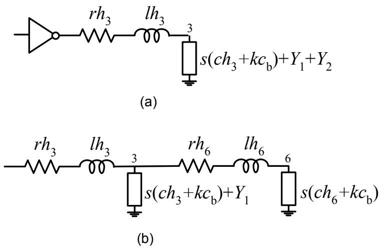

Figure 7: The equivalent circuits of Figure 6(b). (a) The equivalent RLY circuit for calculating gate delay; (b) The equivalent RLY circuit for calculating wire delay.

4 Simulations

To test the accuracy of our model, we used a set of parameters based on the 180 nm technology in the ITRS'99 roadmap [15]. The parameters were also used in [5]. We list the technology parameters in Table 2. Level 49 HSPICE models were used for the buffer drivers that are 1.27x larger than a minimum sized buffer. Our delay model can deal with different wire lengths of a tree, and can also compute any percentage of delay time.

We applied our delay model for a balanced buffered RLC tree with a root and a set of leaves (see Figure 8). The topology was used in [8, 10]. Each edge of the tree is modeled by a buffer driving an interconnect wire segment. Meanwhile, buffer insertion was also used to reduce the delay between the source and node 6 (critical path) in this experiment. For this experiment, we computed the 50% delay time. In Table 3, columns 1-3 (Levels 1-3) list several different lengths of a wire in the buffered trees. Column 4 gives the capacitances of the load. Columns 5-8 show the delay times calculated by the HSPICE simulation, the Elmore delay model [6] (with the π-model to model a wire segment), the delay model in [10], and our delay model, respectively. Columns 9-11 give the respective percentages of errors compared with HSPICE.

The simulations show that our formulae are more accurate than related previous works. Compared to HSPICE, the maximum error calculated by the Elmore delay model is 32.35% and the absolute average error is 22.47%, the maximum error calculated by the [10] is -21.28% and the absolute average error is 15.51%, and the maximum error calculated by our delay model is only -6.87% and the absolute average error is only 4.19%. As shown in Figure 9, the delays are plotted as a function of the total path length.

Finally, we applied our delay model for an un-balanced buffered RLC tree with a root and a set of leaves (see Figure 5). The buffer is inserted in an arbitrary location of the tree. Buffer insertion was also used to reduce the delay between the source and node 7 (critical path) in this experiment. For this experiment, we computed the 50% delay time. In Table 4, columns 1-11 (Levels 1-3) list several different lengths of a wire in the buffered trees. Column 12 gives the capacitances of the load. Columns 13-15 show the delay times calculated by the HSPICE simulation, the delay model in [10], and our delay model, respectively. Columns 16-17 give the respective percentages of errors compared with HSPICE.

The simulations show that our formulae are more accurate than related previous works. Compared to HSPICE, the maximum error calculated by the [10] is -26.91% and the absolute average error is 25.16%, and the maximum error calculated by our delay model is only -8.77% and the absolute average error is only 6.83%.

According to the above two experiments, the average error of our delay model is 5.51%. Therefore, our delay model is more accurate than the Elmore delay model and that proposed by [10]. The Elmore delay model always overestimates the delay [6]. The delay model proposed by [10] may incur larger errors when the tree is unbalanced. Its error may exceed 20% for unbalanced trees [8]. HSPICE is very accurate but computationally very expensive. For a buffered RLC tree shown in Figure 5 (Figure 8), the CPU run time of HSPICE is about 20 minutes. On the contrary, the CPU run time of our method is less than 1 second, which is close to that under Elmore delay model and the delay model proposed by [10]. Therefore, our method is very efficient.

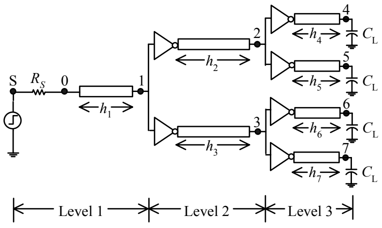

Figure 8: A balanced buffered RLC tree used in [8, 10].

5 Conclusion

We have presented accurate formulae for modeling the delays of *buffered RLY/RLC wires and trees*. Our formulae can handle balanced and un-

Tech. node (nm)	width (nm)	height (nm)	t_{ins} (nm)	ϵ_r	r ($k\Omega/m$)	c (pF/m)	l (nH/mm)	r_b ($k\Omega$)	c_b (fF)	c_p (fF)
180	525	1155	7699	3.75	36.3	269	4.6	8	1.9	4.8

Table 2: Technology parameters in ITRS'99.

Level 1	Level 2	Level 3	Load Cap.	50% Delay Time (ps)				Error (%)		
(um)	(um)	(um)	(pF)	HSPICE	Elmore	I&F	Ours	Elmore	I&F	Ours
300	200	100	0.1	1060	1262	882	990	19.06	-16.76	-6.60
			0.5	2920	3784	2635	2830	29.59	-9.77	-3.08
			1	5240	6935	4825	5125	32.35	-7.91	-2.19
600	500	400	0.1	2010	2298	1609	1879	14.33	-19.97	-6.52
			0.5	3840	4824	3364	3723	25.63	-12.40	-3.05
			1	6130	7981	5558	6022	30.20	-9.32	-1.76
900	800	700	0.1	2970	3336	2338	2766	12.32	-21.28	-6.87
			0.5	4810	5866	4096	4610	21.95	-14.83	-4.16
			1	7100	9029	6295	6910	27.17	-11.34	-2.68
1000	1000	1000	0.1	3728	4195	2939	3481	12.53	-21.15	-6.63
			0.5	5568	6729	4701	5326	20.85	-15.57	-4.35
			1	7862	9897	6903	7626	25.88	-12.20	-3.00
600	800	1000	0.1	3326	3831	2678	3144	15.18	-19.47	-5.47
			0.5	5180	6366	4440	4988	22.90	-14.28	-3.71
			1	7498	9534	6642	7288	27.15	-11.42	-2.80
Absolute average								22.47	15.51	4.19

Table 3: Experimental results for the accuracy of related delay models; balanced tree, technology node = 180 nm, R_S = 180 Ω.

Level 1	Level 2			Level 3							Load	50% Delay Time (ps)			Error (%)	
$h1$ (um)	$h2$ (um)	$h3$ (um)	$h4$ (um)	$h5$ (um)	$h6$ (um)	$h7$ (um)	$h8$ (um)	$h9$ (um)	$h10$ (um)	$h11$ (um)	Cap. (pF)	HSPICE	I&F	Ours	I&F	Ours
1500	2000	2000	2200	1200	4000	4000	5000	4000	4000	1400	0.1	19730	14682	18281	-25.59	-7.34
											0.5	24530	18245	22579	-25.62	-7.95
											1	29980	22699	27943	-24.28	-6.79
1000	1500	1500	3000	1000	3000	3000	4000	3500	3500	1300	0.1	14830	11292	14151	-23.86	-4.56
											0.5	19490	14840	18453	-23.86	-5.32
											1	25320	19276	23819	-23.87	-5.93
600	1200	1200	2400	900	2300	2300	3500	3000	3000	1100	0.1	11960	9055	11417	-24.29	-4.54
											0.5	16680	12594	15718	-24.50	-5.77
											1	22570	17017	21087	-24.61	-6.57
1000	500	500	1000	1000	500	500	1000	500	500	1000	0.1	5455	3987	5093	-26.91	-6.64
											0.5	10200	7500	9399	-26.47	-7.85
											1	16030	11891	14770	-25.82	-7.86
2000	800	800	1600	500	700	700	1500	700	700	400	0.1	5819	4315	5360	-25.85	-7.89
											0.5	10600	7833	9670	-26.11	-8.77
											1	16480	12230	15045	-25.79	-8.71
Absolute average															25.16	6.83

Table 4: Experimental results for the accuracy of related delay models; un-balanced tree, technology node = 180 nm, R_S = 180 Ω.

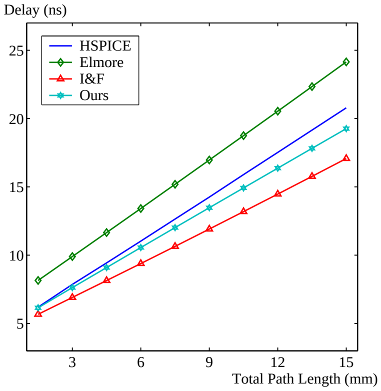

Figure 9: Comparison of the delays calculated by HSPICE, [6] (denoted by *Elmore*), [10] (denoted by *I&F*), and our delay model for a balanced buffered RLC tree.

balanced trees and consider buffer insertion. Extensive simulations with HSPICE have shown that our formulae achieve the best accuracy than related previous works. Future work lies in the delay optimization for buffered RLY/RLC trees based on the formulae.

References

[1] A. I. Abou-Seido, B. Nowak, and C. Chu, "Fitted Elmore Delay: A Simple and Accurate Interconnect Delay Model," *Proc. ICCD*, pp. 422–217, Sep. 2002.

[2] V. Adler and E. G. Friedman, "Uniform Repeater Insertion in RC Trees," *IEEE Trans. on CAS-I*, vol. 47, no. 10, pp. 1515–1523, Oct. 2000.

[3] R. Arunachalam, F. Dartu, and L. T. Pileggi, "CMOS Gate Delay Models for General RLC Loading," *Proc. ICCD*, pp. 224–229, Oct. 1997.

[4] K. Banerjee and A. Mehrotra, "Analysis of On-Chip Inductance Effects using a Novel Performance Optimization Methodology for Distributed RLC," *Proc. DAC*, pp. 798–803, June 2001.

[5] K. Banerjee and A. Mehrotra, "Analysis of On-Chip Inductance Effects for Distributed RLC Interconnects," *IEEE Trans. on CAD*, vol. 21, no. 8, pp.904–915, Aug. 2002.

[6] W. C. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wide Band Amplifiers," *J. Applied Physics*, Vol. 19, No. 1, 1948.

[7] D. S. Gao and D. Zhou, "Propagation Delay in RLC Interconnection Networks," *Int. Symposium on Circuit and Systems*, pp. 2125–2128, May 1993.

[8] Y. I. Ismail and E. G. Friedman, "Equivalent Elmore Delay for RLC Trees," *IEEE Trans. on CAD*, vol. 19, no. 1, pp. 83–97, Jan. 2000.

[9] Y. I. Ismail and E. G. Friedman, "Effects of Inductance on the Propagation Delay and Repeater Insertion in VLSI Circuits," *IEEE Trans. on VLSI*, vol. 8, no. 2, pp. 195–206, Apr. 2000.

[10] Y. I. Ismail and E. G. Friedman, "Repeater Insertion in Trees Structured Inductive Interconnect," *IEEE Trans. on CAS-II*, vol. 48, no. 5, pp. 471–481, May. 2001.

[11] A. B. Kahng and S. Muddu, "Optimal Equivalent Circuits for Interconnect Delay Calculations Using Moments," *Proc. European Design Automation Conf.*, pp. 164–169, Sep. 1994.

[12] A. B. Kahng and S. Muddu, "Two-pole Analysis of Interconnection Trees," *Proc. MCMC Conf.*, pp. 105–110, Jan. 1995.

[13] A. B. Kahng and S. Muddu, "An Analytical Delay Model for RLC Interconnects," *IEEE Trans. on CAD*, vol. 16, no. 12, pp. 1507–1514, Dec. 1997.

[14] Ankireddy Nalamalpu and Wayne Burleson, "Repeater Insertion in deep sub-micron CMOS: Ramp-based Analytical Model and Placement Sensitivity Analysis," *Proceedings of International Symposium on Circuits and Systems*, vol.3, pp. 766 -769, June. 2000.

[15] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors 1999 Edition*, 1999.

[16] M. Sriram and S. M. Kang, "Performance Driven MCM Routing Using a Second Order RLC Tree Delay Model," *Proc. IEEE Intl. Conf. on Wafer Scale Integration*, pp. 262-267, Jan. 1993.

[17] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 2, pp. 584-594, Apr. 1990.

[18] K. T. Tang and E. G. Friedman, "Delay and Power Expressions Characterizing a CMOS Inverter Driving an RLC Load," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. III.283-III.286, May 2000.