Optimizing *RLC* Tree Delays by Employing Repeater Insertion

Yehea I. Ismail, Eby G. Friedman, and Jose L. Neves[1]

Department of Electrical and Computer Engineering
University of Rochester
Rochester, New York 14627

[1]IBM Microelectronics
1580 Route 52
East Fishkill, New York 12533

Abstract – **The effects of inductance on repeater insertion in *RLC* trees is the focus of this paper. An algorithm is introduced to insert and size buffers within an *RLC* tree to minimize a variety of possible cost functions such as minimizing the maximum path delay, the skew, or a combination of area, power, and delay. The algorithm has a complexity proportional to the square of the number of possible buffer positions and determines a buffer solution that is close to the global minimum. The buffer insertion algorithm is used to insert buffers within several copper-based interconnect trees to minimize the maximum path delay based on both an *RC* model and an *RLC* model. The two buffering solutions are compared using the AS/X dynamic circuit simulator. It is shown that as inductance effects increase, the area and power consumed by the inserted buffers to minimize the path delays of an *RLC* tree decreases. By including inductance in the repeater insertion methodology, the interconnect is modeled more accurately as compared to an *RC* model, permitting average savings in area, power, and delay of 40.8%, 15.6%, and 6.7%, respectively, for a variety of copper-based interconnect trees for a 0.25 μm CMOS technology. The average savings in area, power, and delay increases to 62.2%, 57.2%, and 9.4%, respectively, when using five times faster devices with the same interconnect trees.**

I. Introduction

Repeater insertion has become an increasingly common design methodology for driving long resistive interconnect [1]-[5]. Since the propagation delay has a square dependence on the length of an *RC* interconnect line, subdividing the line into shorter sections by inserting repeaters is an effective strategy to reduce the total propagation delay. A second important advantage of inserting buffers within interconnect trees is to decouble a large capacitance from the critical path in order to minimize the overall delay of the critical path [1], [5].

Currently, inductance is becoming more important with faster on-chip rise times and longer wire lengths [6]-[11]. Wide wires are frequently encountered in clock distribution networks and in upper metal layers. These wires are low resistive lines that can exhibit significant inductive effects. Furthermore, performance requirements are pushing the introduction of new materials for low resistance interconnect [8] and new dielectrics to reduce the interconnect capacitance. These technological advances increase the importance of inductance as has been described in [6] and [7].

The focus of this paper is to characterize the effects of inductance on the repeater insertion process. The paper is organized as follows. The results of applying a repeater insertion CAD tool to insert buffers in several industrial copper-based interconnect trees are presented in section II and interpretations of these results are given. Some conclusions are described in section III. In Appendix A, the buffer insertion algorithm is briefly described. The specific delay models used in this paper for the devices and the interconnect are described in Appendix B

II. Characterizing the Effects of Inductance on the Repeater Insertion Methodology

The results of applying a CAD-based repeater insertion tool on several industrial copper-based interconnect trees are summarized and discussed in this section. The CAD tool is briefly described in Appendix A. The *RLC* trees used in this paper are copper-based interconnect wires based on an IBM 0.25 μm CMOS technology. The depth of the trees (the maximum path length from the input to the sinks) is between 0.5 cm to 1.5 cm and considers a wide range of critical global signals typically encountered in VLSI chips. Long wires within the trees are partitioned with a maximum segment length of 0.5 mm to allow buffer insertion within long wires for improved performance [5].

A buffer solution is determined to minimize the maximum path delay of each tree based on the *RLC* delay model discussed in Appendix B. The total area of the buffers inserted in each tree is described in terms of the area of a minimum size buffer. The CAD tool also generates an AS/X [12] input file which is used to simulate the maximum path delay and the power consumption of the buffered *RLC* tree. The total buffer area, the maximum path delay, and the power consumption of the buffered trees are depicted in Table 1. The tool is also used with AS/X to determine the total buffer area, the maximum path delay, and the power consumption of the buffered *RLC* trees based on an *RC* model. The results based on the *RC* model are also listed in Table 1. Finally, AS/X simulations of the unbuffered *RLC* trees are used to determine the maximum path delay when buffer insertion is not employed which are also listed in Table 1.

Two important trends can be observed from the data listed in Table 1. The first trend is that inserting buffers significantly reduces the maximum path delay as compared to the maximum path delay of an unbuffered tree. This trend illustrates the importance of buffer insertion as an effective methodology to reduce interconnect delay. According to Tables 1 and 2, the average saving in the maximum path delay when inserting buffers based on an *RLC* model as compared to an unbuffered tree is 42.2% and the maximum saving is 76% for TGL1 which is a large asymmetric tree. The second important trend apparent in the data listed in Table 1 is that inserting buffers based on an *RLC* model as compared to an *RC* model consistently introduces savings in all three primary design parameters: area, power, and delay. This trend demonstrates the importance of including inductance in a high speed

This research was supported in part by the National Science Foundation under Grant No. MIP-9610108, a grant from the New York State Science and Technology Foundation to the Center for Advanced Technology - Electronic Imaging Systems, and by grants from the Xerox Corporation, IBM Corporation, and Intel Corporation.

0-7803-5632-2/99/$10.00 © 1999 IEEE

repeater insertion methodology. According to Table 2, including inductance in the interconnect model saves an average 40.8% of the buffer area, 15.6% of the power dissipated by the buffered trees, and 6.7% of the maximum path delay as compared to using an *RC* model.

The reduced buffer area when including inductance in the interconnect model can be explained by the quadratic dependence of the delay on the length of an *RC* wire which tends to a linear dependence as inductance effects increase [11]. The 50% delay of an *RC* line is $0.35RCl^2$ [3], [13] and for an *LC* line is $l\sqrt{LC}$ [11] when the line is driven by an ideal source and has an open-circuit load. *R*, *L*, and *C* are the resistance, inductance, and capacitance per unit length of the line and *l* is the length of the line. These two cases of an *RC* line and an *LC* line are the limiting cases for inductance effects with the *RC* case representing no inductance effects and the *LC* case representing maximum inductance effects. In the *RC* case, the square dependence on the interconnect length causes the delay to increase rapidly with wire length. It is therefore necessary to partition the line into multiple shorter sections by inserting repeaters, thereby reducing the total delay. However, for an *LC* line, the dependence is linear and no gain is achieved by breaking the line into shorter sections. Actually, inserting repeaters would only degrade the delay due to the added gate delay. Thus, an *LC* line requires *zero* repeater area for minimum propagation delay. In the general case of an *RLC* line, the buffer area for minimum propagation delay is between the maximum buffer area in the *RC* case and the zero buffer area in the *LC* case. The buffer area for minimum propagation delay of an *RLC* line decreases as inductance effects increase due to the sub-quadratic dependence of the propagation delay on the length of the interconnect [11]. Hence, inserting buffers based on an *RC* model and neglecting inductance would require larger buffer area than necessary to achieve a minimum delay. The magnitude of the excess buffer area when using an *RC* model depends upon the relative magnitude of the inductance within the tree. For the specific copper interconnect *RLC* trees used here, almost half the buffer area can be saved by including inductance in the interconnect model. Note that a single line analysis can be used to interpret the behavior of a repeater insertion solution in a tree since in both cases repeaters are inserted to break the *RC* delay of long wires (paths and branches in the case of a tree). Additionally, buffers are inserted in a tree to decouple a capacitance from the critical path. The effect of capacitance decoupling on improving the critical path delay is less significant when inductance effects increase. This trend is due to the *LC* time constant at node *i* of a tree, $\sqrt{\sum_k C_k L_{ik}}$ [10], which has a square root behavior as compared to the linear behavior of the *RC* time constant, $\sum_k C_k R_{ik}$. Reducing the capacitance coupling has less effect on the *LC* time constant as compared to the *RC* time constant due to this square root behavior. As inductance effects increase, the square root behavior of the *LC* time constant dominates the behavior of the propagation delay. Thus, as inductance effects increase, the area of the inserted buffers for capacitive decoupling also decreases.

A reduction in the power consumed by the buffered trees when including inductance in the interconnect model as compared to an *RC* model is a direct consequence of the reduced buffer area. The dynamic power consumption, which is linearly dependent on the total capacitance of the interconnect and the buffers, decreases due to the reduced input and output capacitance of the buffers. The short-circuit power consumption is significantly less for a smaller buffer since the short-circuit power consumed by a CMOS inverter is quadratically dependent on the width of the buffer [14], [15].

The optimum number of sections that an *RLC* line should be partitioned into and the size of each inserted buffer to achieve the minimum total propagation delay have been characterized in [11] and are

$$h_{opt} = \sqrt{\frac{R_0 C_t}{R_t C_0}} \frac{1}{\left[1 + 0.16(T_{L/R})^3\right]^{0.24}} \quad \text{and} \quad k_{opt} = \sqrt{\frac{R_t C_t}{2 R_0 C_0}} \frac{1}{\left[1 + 0.18(T_{L/R})^3\right]^{0.3}}, \tag{1}$$

where

$$T_{L/R} = \sqrt{\frac{L_t / R_t}{R_0 C_0}}. \tag{2}$$

R_o and C_o are the output resistance and input capacitance of a minimum size buffer, respectively, and R_t, L_t, and C_t are the total resistance, inductance, and capacitance of the line, respectively. Note in (1) that h_{opt} and k_{opt} are the same as the expressions in [2] and [3] for an *RC* line where $T_{L/R}$ is equal to zero ($L_t = 0$). Both the size and number of the repeaters decrease as inductance effects increase.

Another interesting characteristic in (1) and (2) is that $T_{L/R}$ increases as the time constant R_0C_0 decreases, or alternatively, when faster buffers are used. An increase in $T_{L/R}$ increases the discrepancy between an *RC* model and an *RLC* model as described by (1) even if the same interconnect trees are buffered to minimize the path delay. Thus, the analytical solutions in (1) and (2) anticipate additional savings in the buffer area by including inductance in the interconnect model as compared to an *RC* model when faster devices are used as buffers. To verify this trend, five times faster devices than the 0.25 μm transistors are used as buffers to minimize the maximum path delays of the *RLC* trees listed in Table 1. Note that the savings in area, power, and delay increase when including inductance in the interconnect model rather than using an *RC* model with faster devices as compared to the 0.25 μm CMOS technology. Referring to the results listed in Tables 2 and 3, the average savings increase from 40.8% to 62.2% for the buffer area, from 15.6% to 57.2% for the power consumption, and from 6.7% to 9.4% for the maximum path delay when using five times faster devices as compared to a 0.25 μm CMOS technology. Thus, with a faster technology, the penalty of ignoring inductance increases for all three primary design parameters: area, power, and delay. Therefore, with technology scaling, including inductance in the repeater insertion methodology will become of paramount importance.

This trend can be explained intuitively by examining the special case of a line with large inductance effects. As discussed before, the minimum total propagation delay can be achieved for such a line by inserting no buffers independent of the intrinsic speed of the technology. If inductance is ignored and an *RC* model is used for such a line, the number of buffers that are inserted will increase as the buffers become faster since there is less penalty for inserting more buffers. Thus, the discrepancy between the buffer solutions based on an *RC* and an *RLC* model (zero buffer area for dominant inductance effects) increases as faster buffers are used. In general, the buffer area required to minimize the total propagation delay based on an *RC* model increases more rapidly as the devices become faster as compared to an *RLC* model.

Table 1. Simulation results of unbuffered trees, buffered trees based on an *RLC* model, and buffered trees based on an *RC* model. The area, power, and maximum path delay are compared. Area is determined by the buffer insertion tool while the power and maximum path delay are derived from AS/X.

Tree Name	Area (minimum size inverters)			Power (pJ per Cycle)			Maximum Delay (ps)		
	Un-Buffered Tree	Buffered Tree *RLC* Model	Buffered Tree *RC* Model	Un-Buffered Tree	Buffered Tree *RLC* Model	Buffered Tree *RC* Model	Un-Buffered Tree	Buffered Tree *RLC* Model	Buffered Tree *RC* Model
TSs1	0	352	380	13.86	23.26	25	488	288	297
L1	0	102	250	8.15	11.19	13.76	342	267	272
TS2	0	0	659	25.67	25.67	37.90	193	193	193.5
L2	0	310	337	11.92	20.85	21.55	700	437	454
L3	0	0	422	22.8	22.8	30.3	213	213	237
TSm1	0	1246	1709	95	125	146	389	268	284
TSm2	0	1630	2751	135	211	221.5	343	278	296
TSL	0	1734	2471	147.5	196	227	431	292	304
TSL1	0	2999	4120	164	237	275	781	360	382
TGs1	0	649	842	38	51.2	57.8	262	231	256
TGs2	0	0	553	40.20	40.20	59.80	212	212	247
TGm1	0	1271	1854	89.1	120	139	460	306	344
TGL1	0	3823	7506	201	295	378	1740	442	495

Table 2. The total buffer area, power, and maximum path delay of all the trees. Per cent savings represent the average savings in area, power, and maximum path delay when using an *RLC* model for buffer insertion.

Totals					
	Un-Buffered	Saving *RLC* to un-buffered	Buffered *RLC* Model	Buffered *RC* Model	Saving *RLC* to *RC*
Area (min inverters)	0	-	14116	23854	40.8%
Max delay (ps)	6554	42.2%	3787	4061	6.7%
Power (PJ/Cycle)	-	-	1379	1632	15.6%

Table 3. The total buffer area, power, and maximum path delay of all the trees using five times faster devices. Per cent savings represent the average savings in area, power, and maximum path delay when using an *RLC* model for buffer insertion.

Totals					
	Un-Buffered	Saving *RLC* to un-buffered	Buffered *RLC* Model	Buffered *RC* Model	Saving *RLC* to *RC*
Area (min inverters)	0	-	67960	170227	62.2%
Max delay (ps)	6554	57.17%	2807	3098	9.4%
Power (PJ/Cycle)	-	-	2007	4691	57.2%

III. Summary

Neglecting inductance in the interconnect model for repeater insertion is shown to cause significant errors. Certain VLSI trends will make inductance even more significant, such as:

1- Lower resistivity metal alloys for interconnect, copper interconnect being a primary example [8].

2- Lower permeability dielectrics to insulate the interconnect which reduces the interconnect capacitance. Reducing the interconnect capacitance increases the effects of inductance [7].

3- Higher operating frequencies [6], [7].

4- Faster devices with technology scaling and the increasing use of SOI devices with significantly higher speed. Using faster devices increases the error caused by neglecting inductance in the repeater insertion methodology.

5- Tighter timing constraints for VLSI circuits to meet higher frequency targets which require more accurate delay models.

Appendix A: Algorithm for Buffer Insertion in *RLC* Trees

An *RLC* tree is shown in Fig. 1. The tree has n wires with the input source driving the root wire. Each wire w drives two wires, a left wire and a right wire. A leaf is a wire that has no children. The tree has r leaf wires, each of which drives one of the sinks of the tree. A binary branching factor is used without loss of generality since any tree can be transformed into a binary tree by inserting zero impedance wires [1], [5]. At each sink $1 \leq i \leq r$, the propagation delay t_{di} is defined as the 50% delay of the output signal at sink i with respect to the input signal at the root of the tree. Within a tree, there are m pre-specified buffer positions where buffers can be inserted to minimize a given cost function. The possible buffer positions are shown by circles in Fig. 1 and are placed at the beginning of each wire to allow for maximum capacitive decoupling of the critical paths [1], [5]. Each wire can be subdivided into several shorter wires to insert buffers within the long wires [5]. In some cases, no possible buffer positions can be assigned to some wires due to existing layout constraints. Those wires are labeled to indicate that no buffers can be inserted on those wires.

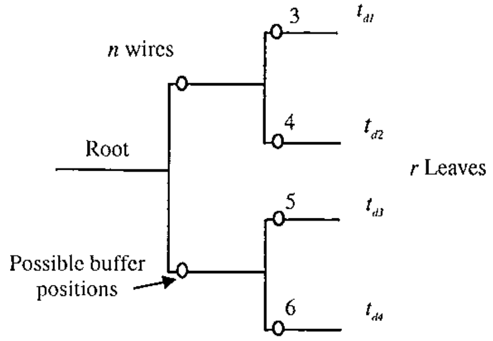

Fig. 1. An *RLC* tree with n wires. The possible buffer positions are shown by circles.

It is necessary to determine the set of buffer sizes h_j, $1 \leq j \leq m$, that minimizes a given cost function $C(h_1, h_2, ..., h_j, ..., h_m)$. The buffers are symmetric inverters with widths h_j and a minimum sized channel length. The buffer sizes h_j are continuous and the buffering solution is not limited to a given buffer library. The buffer size $h_j = 0$ indicates that no buffer is inserted at node j. The sizes of the buffers are in the range $1 \leq h_j \leq h_{max}$ where h_{max} is the maximum allowable size of any buffer. A variety of cost functions can be used. Examples are: minimize(max$_i$ t_{di}) which aims to minimize the maximum path delay, minimize(max$_{i,k}$(t_{di}-t_{dk})) where $1 \leq i,k \leq r$ which is equivalent to minimizing the skew, minimize(t_{dk}) where k is a critical output, or minimize ($f(t_{di})$ + $\sum_{j=1}^{m} h_j$) which considers the area of the buffers. Other cost functions can include power dissipation and transition time.

The algorithm is used to determine the optimum sizes of the buffers that minimize the target cost function. Pseudocode describing this algorithm is shown in Fig. 2. Referring to Fig. 1, the algorithm begins with the initial condition $h_j = 0 \; \forall \; j$ which corresponds to an unbuffered tree. The cost function $C(h_1, h_2, \ldots, h_j, \ldots, h_m)$ is evaluated for several sizes of the buffer at node 1, h_1, with all other buffer sizes $h_2, \ldots, h_m$ equal to zero (no buffers). A binary search is used to determine the value h_1 that minimizes the cost function where in each step a new value for h_1 is chosen and the cost function is evaluated. As is shown in Appendix B, the method used to calculate the delay has a complexity $O(n)$. The number of steps B depends on h_{max} and is typically less than ten. If the case of no buffer at node 1 ($h_1 = 0$) provides the lowest cost, h_1 remains equal to zero. Thus, the algorithm can only improve the cost function at each step. The size of the buffer at node 2, h_2, that minimizes the cost is determined in the same manner with h_1 set to the value calculated from the previous step and all other buffer sizes set to zero. The process is repeated for all m possible buffer positions. At each possible buffer position, the buffer size that minimizes the cost function is determined while all of the previous optimum buffer sizes remain constant. The process of covering all possible m buffer positions is defined as an iteration. The complexity of a single iteration is $O(m{\cdot}n{\cdot}B)$. The memory requirement of the algorithm is proportional to the number of wires, n.

In each step of an iteration, the algorithm improves the cost function. After completing the first iteration, a second iteration starts by changing the sizes of the buffers determined in the previous iteration at the possible buffer positions to determine the buffer sizes $h_1, h_2, \ldots, h_m$ that minimize the cost function. The initial condition is used to calculate the optimum buffer sizes more accurately as compared to a previous iteration. The iterations are repeated until no change in the size of any buffer as compared to the previous iteration occurs. The algorithm typically converges within two or three iterations.

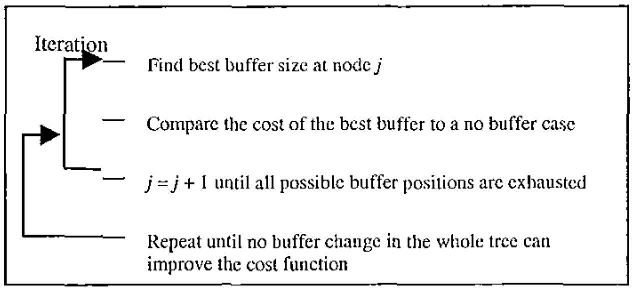
Iteration
Find best buffer size at node j
Compare the cost of the best buffer to a no buffer case
$j = j + 1$ until all possible buffer positions are exhausted
Repeat until no buffer change in the whole tree can improve the cost function

Fig. 2. Algorithm for inserting buffers in an *RLC* tree.

The algorithm terminates when no change in the size of a single buffer can improve the cost function. This behavior can be expressed as

$$\frac{dC(h_1, h_2, \ldots, h_j, \ldots, h_m)}{dh_j} = 0 \qquad \forall \, j, \tag{3}$$

where the algorithm reaches a minimum in the cost function. However, there is no guarantee that this minimum is the global minimum. To improve the final buffer solution, the two buffers at the left and right possible buffer positions of each wire are changed simultaneously. The process of determining two buffer sizes that simultaneously minimize the cost requires B^2 binary search steps. Since there are $m/2$ possible buffer position pairs, the complexity of this modified algorithm is

$$O(m \cdot n \cdot \frac{B^2}{2}). \tag{4}$$

This modified algorithm does not reach a minimum near the initial point. Rather, the modified algorithm searches for a minimum closer to the global minimum, requiring increased computational time. In general, a set of higher order algorithms can be developed by simultaneously changing more buffers. The complexities of these algorithms are

$$O(m \cdot n \cdot B),\; O(m \cdot n \cdot \frac{B^2}{2}),\; O(m \cdot n \cdot \frac{B^3}{3}), \cdots\cdots,\; O(n \cdot B^m). \tag{5}$$

An algorithm that changes m buffers simultaneously is guaranteed to reach the global minimum. However, the computational time is exponential with the number of possible buffer positions and is prohibitively high even for relatively small trees. These algorithms have been examined for small trees (seven to eight possible buffer positions) and compared to the exhaustive algorithm that simultaneously changes all m buffers. The results demonstrate that the second order algorithm consistently reaches the global or near global minimum. The higher order algorithms introduce a slight improvement in the final buffer solution as compared to the second order algorithm. The CPU run time of the second order algorithm for a large tree that has 250 possible buffer positions is 20 sec on an S/490 IBM machine with one gigabyte of RAM. For typical trees with less than fifty possible buffer positions, the CPU time is below one second. Hence, the second order algorithm is used in the work presented here.

Appendix B: Delay Model

The method [9], [10] used to evaluate the delays at the sinks of a buffered *RLC* tree is briefly discussed here. The proposed method approximates the nonlinear transistor characteristics by combining two piecewise linear regions describing the linear and saturation regions of operation [9]. Thus, delays are found for each linear network, denoted t_{pdlin} and t_{pdsat} for the linear approximation and the saturation approximation, respectively.

In the general case, neither t_{pdsat} nor t_{pdlin} can solely characterize the propagation delay of a nonlinear CMOS gate driving an *RLC* tree since the NMOS transistor operates partially in the saturation region and partially in the linear region. However, a combination of both t_{pdsat} and t_{pdlin} can be used to accurately characterize the propagation delay. The resulting delay for the general case in terms of t_{pdsat} and t_{pdlin} is [9]

$$t_{pd} = t_{pdlin} + t_{pdsat} \exp(-1.1 \frac{t_{pdlin}}{t_{pdsat}}). \tag{6}$$

In general, this method is highly accurate (errors within 3%) for fast inputs. Additional errors may result from applying a linear analysis method to determine t_{pdsat} and t_{pdlin}. This piecewise linear approximation has significant accuracy advantages over the commonly used linear transistor model.

To calculate the delay of the linear *RLC* trees resulting from the two region linear transistor model, a second order transfer function that approximates the transfer function at node i of an *RLC* tree as introduced in [10] is

$$g_i(s) = \frac{\omega_{ni}^2}{s^2 + s2\zeta_i\omega_{ni} + \omega_{ni}^2}. \tag{7}$$

The variables ζ_i and ω_{ni} that characterize the second order approximation of the transfer function at node i are

$$\zeta_i = \frac{1}{2}\frac{\sum_k C_k R_{ik}}{\sqrt{\sum_k C_k L_{ik}}} \text{ and } \omega_{ni} = \frac{1}{\sqrt{\sum_k C_k L_{ik}}}, \tag{8}$$

where R_{ik} (L_{ik}) is the common resistance (inductance) from the input to nodes i and k. The accuracy characteristics of this solution is similar to the Elmore [16] (Wyatt [17]) delay model for *RC* trees [10].

The 50% propagation delay of the signal at node i of an *RLC* tree for a step input is [10]

$$t_{pdi} = (1.047e^{-\frac{\zeta_i}{0.85}} + 1.39\zeta_i)/\omega_{ni}\,. \tag{9}$$

The error of these expressions is less than 3% for balanced trees and can reach up to 20% for highly unbalanced trees [10].

Referring to (8) and (9), the delay at node i depends upon evaluating two summations at node i, which are

$$T_{RCi} = \sum_k C_k R_{ik} \quad\text{and}\quad T_{LCi} = \sum_k C_k L_{ik}\,. \tag{10}$$

The first summation is the Elmore delay, which can be calculated efficiently with linear complexity by building the solution at a node in a tree based on the solutions at its immediate children, *e.g.*, [1], [18]. The second summation is calculated in precisely the same manner with the branch resistances replaced by the branch inductances. Thus, this second order approximation [10] preserves the computational properties of the Elmore delay, permitting highly efficient algorithms to characterize the signals within an *RLC* tree. The complexity of calculating the delays at the sinks of a buffered *RLC* tree is linear with the number of wires n in the tree [18].

References

[1] L. V. Ginneken, "Buffer Placement in Distributed RC-tree Networks for Minimal Elmore Delay," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 865 - 868, May 1990.

[2] H. B. Bakoglu and J. D. Meindl, "Optimal Interconnection Circuits for VLSI," *IEEE Transactions on Electron Devices*, Vol. ED-32, No. 5, pp. 903 - 909, May 1985.

[3] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley Publishing Company, 1990.

[4] V. Adler and E. G. Friedman, "Repeater Design to Reduce Delay and Power in Resistive Interconnect," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 2148 - 2151, June 1997.

[5] C. J. Alpert, "Wire Segmenting for Improved Buffer Insertion," *Proceedings of the IEEE/ACM Design Automation Conference*, June 1997.

[6] A. Deutsch, *et al.*, "When are Transmission-Line Effects Important for On-Chip Interconnections?," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 45, No. 10, pp. 1836 - 1846, October 1997.

[7] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Figures of Merit to Characterize the Importance of On-Chip Inductance," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 560-565, June 1998.

[8] P. J. Restle and A. Duetsch, "Designing the Best Clock Distribution Network," *Proceedings of the IEEE VLSI Circuit Symposium*, pp. 2 – 5, June 1998.

[9] Y. I. Ismail and E. G. Friedman, "Optimum Repeater Insertion Based on a CMOS Delay Model for On-Chip *RLC* Interconnect," *Proceedings of* the *IEEE ASIC Conference*, pp. 369-373, September 1998.

[10] Y. I. Ismail, E. G. Friedman, and Jose L. Neves, "Equivalent Elmore Delay for *RLC* Trees," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 715-720, June 1999.

[11] Y. I. Ismail and E. G. Friedman, "Effects of Inductance on the Propagation Delay and Repeater Insertion in VLSI Circuits," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 721-724, June 1999.

[12] *AS/X User's Guide*, IBM Corporation, New York, 1996.

[13] T. Sakurai, "Approximation of Wiring Delay in MOSFET LSI," *IEEE Journal of Solid-State Circuits*, Vol. SC-18, No. 4, pp. 418 - 426, August 1983.

[14] H. J. M. Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and its Impact on the Design of Buffer Circuits," *IEEE Journal of Solid-State Circuits*, Vol. SC-19, No. 4, pp. 468 - 473, August 1984.

[15] Y. I. Ismail, E. G. Friedman, and J. L. Neves," Dynamic and Short-Circuit Power of CMOS Gates Driving Lossless Transmission Lines," *Proceedings of the IEEE Great Lakes Symposium on VLSI*, pp. 39-44, February 1998.

[16] W. C. Elmore, "The Transient Response of Damped Linear Networks," *Journal of Applied Physics*, Vol. 19, pp. 55 - 63, January 1948.

[17] J. L. Wyatt, *Circuit Analysis, Simulation and Design*, Elsevier Science Publishers, North-Holland, 1987.

[18] C. L. Ratzlaff, *A Fast Algorithm for Computing the Time Moments of RLC Circuits*, Masters Thesis, University of Texas at Austin, Austin, Texas, May 1991.