# Repeater Insertion in Tree Structured Inductive Interconnect: Underlying Theory

Yehea I. Ismail
ECE Department
Northwestern University
Evanston, IL 60208

Eby G. Friedman
ECE Department
University of Rochester
Rochester, New York 14627

***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 repeaters within an *RLC* tree to optimize a variety of possible cost functions such as minimizing the maximum path delay, the skew between branches, or a combination of area, power, and delay. The algorithm has a complexity proportional to the square of the number of possible repeater positions and determines a repeater solution that is close to the global minimum. The repeater insertion algorithm is used to insert repeaters 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 repeaters 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 from 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

It has become well accepted that interconnect delay dominates gate delay in current deep submicrometer VLSI circuits [1]-[7]. With the continuous scaling of technology and increased die area, the crosssectional area of the interconnect decreases while the length of the global interconnect increases which quadratically increases the resistance of the interconnect with technology scaling. Meanwhile, the gate parasitic impedances decrease due to the shrinking of the minimum feature size [4]. The combined effect of these trends is that interconnect has become the primary performance bottleneck, contributing an increasingly significant portion to the total cycle delay. Furthermore, this situation is expected to become worse [4]-[7].

Repeater insertion is becoming an increasingly common design methodology for driving long resistive interconnect [8]-[14]. Since the propagation delay has a square dependence on the length of an *RC* interconnect line, subdividing the line into shorter sections is an effective strategy to reduce the total propagation delay. The interconnect can be subdivided into shorter sections by inserting repeaters, which breaks the quadratic dependence of the delay on the interconnect length but adds additional parasitic impedances due to the inserted repeaters. Thus, an optimum number and size of repeaters exist that minimizes the total propagation delay of the line [10]-[11]. As the gate parasitic impedances decrease with respect to the interconnect parasitic impedances, more repeaters are inserted to further minimize the overall interconnect delay. In that sense, the repeater insertion methodology can be viewed as an effective means for exploiting the decreasing gate delay so as to minimize the increasing interconnect delay. Another reason to insert repeaters within interconnect trees is to decouble large capacitances from the critical path so as to minimize the overall delay of the critical path [8], [13].

Currently, inductance is becoming more important with faster on-chip rise times and longer wire lengths [15]-[28]. Wide wires are frequently encountered in clock distribution networks, data buses, and other structures that use upper metal layers [29]. These wires are low resistance lines that can exhibit significant inductive effects. Furthermore, performance requirements are pushing the introduction of new materials for low resistance interconnect [30]-[32] and new dielectrics to reduce the interconnect capacitance. These technological advances increase the effects of inductance, as has been described in [19]-[21], [27].

The focus of this paper is twofold: to describe a CAD system for repeater insertion in *RLC* trees in order to optimize a variety of cost functions and to characterize the effects of neglecting inductance on the repeater insertion process. The results from applying the repeater insertion tool to several industrial trees are also interpreted. The paper is organized as follows. In section II, the basic repeater insertion algorithm which can be used with any delay model for the interconnect and transistor devices is described. The specific models used in this paper for the transistors and the interconnect are described in section III. The results of applying the tool to insert repeaters in several practical copper-based interconnect trees are presented in section IV. Finally, a summary is given in section V.

## II. Algorithm for Repeater Insertion in *RLC* Trees

A generic algorithm to insert repeaters in a general *RLC* tree is presented in this section. The algorithm can be used with different delay models such as the Elmore delay, moment matching methods, and/or the effective capacitance model to evaluate the transient response of the buffered *RLC* tree. The algorithm has a quadratic complexity with the number of possible repeater positions in an *RLC* tree and achieves a repeater solution that is reasonably close to the global optimum repeater solution. In subsection A, the repeater insertion problem is defined. The algorithm for repeater insertion used in this paper is discussed in subsection B. The complexity and optimality of the algorithm are discussed in subsection C.

### *A.* Problem Definition

The problem of inserting repeater in an *RLC* tree to minimize a given cost function is formulated and defined in this subsection. The terms and mathematical notations used in this paper are also defined. An arbitrary 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 $left(w)$ and a right wire $right(w)$. If a left (right) wire does not exist then $left(w)=0$ ($right(w)=0$). A leaf is a wire that has $left(w)=0$ and $right(w)=0$. 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 [8], [13]. 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 repeater positions where repeaters can be inserted to minimize a given cost function. The possible repeater positions are represented by the circles shown in Fig. 1 and are placed at the beginning of each wire to allow for maximum capacitive decoupling of the critical paths [8], [13]. Each wire can be subdivided into several shorter wires to permit repeater insertion within long wires [13]. In some cases, no possible repeater positions can be assigned to some wires due to layout constraints. Those wires are labeled to indicate that no repeaters can be inserted along the wires.

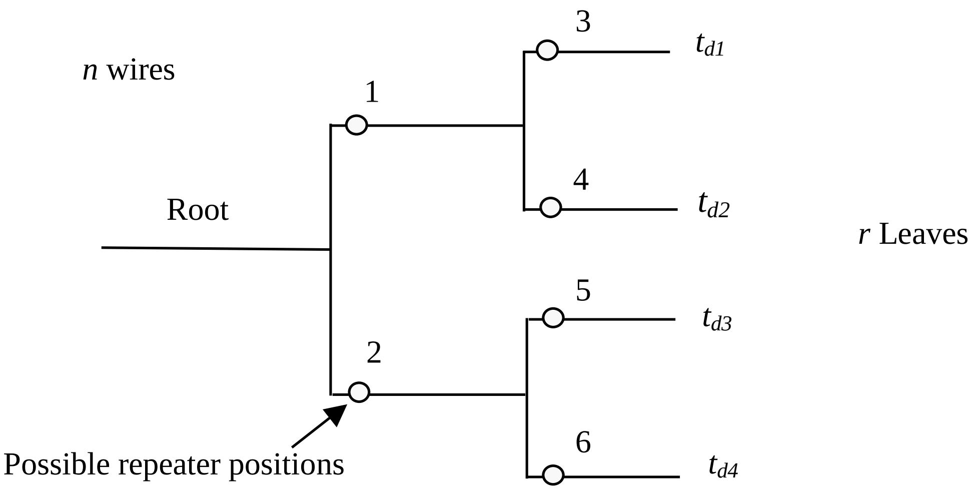


Fig. 1. An arbitrary tree with *n* wires. The possible repeater positions are represented by circles.

The repeater insertion problem can be defined as: determine the set of repeater sizes $h_j$, $1 \leq j \leq m$, that minimizes a given cost function $C(h_1, h_2, \ldots, h_j, \ldots, h_m)$. The repeaters are considered to be symmetric inverters with widths $h_j$ and a minimum sized channel length. The repeater sizes $h_j$ are continuous numbers. The special repeater size $h_j = 0$ indicates that no repeater is inserted at node *j*. The sizes of the repeaters are to be found in the range $1 \leq h_j \leq h_{max}$ where $h_{max}$ is the maximum allowable size of any repeater. 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 between branches *i* and *k*, 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 repeaters. Other cost functions can include power and slew rate.

### *B.* Repeater Insertion Algorithm

According to the problem definition described in the previous subsection, the sizes $h_j$ that minimize the cost function $C(h_1, h_2, \ldots, h_j, \ldots, h_m)$ need to be calculated. The algorithm to calculate the optimum sizes of the repeaters to minimize the cost function is provided in Fig. 2. Referring to Fig. 1, the algorithm starts 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 repeater at node 1, $h_1$, with all other repeater sizes $h_2, \ldots, h_m$ equal to zero (no repeaters). A binary search is applied which permits the value of $h_1$ that minimizes the cost function to be reached within a few steps where each step involves choosing a new value for $h_1$ and evaluating the cost function. The number of steps depends on $h_{max}$ and is typically less than ten steps. If the case of no repeater 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. Next, the size of the repeater 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 repeater sizes set to zero. The process is repeated for all $m$ possible repeater positions. At each possible repeater position the size that minimizes the cost function is determined while all of the previous optimum repeater sizes remain constant. The process of covering all possible $m$ repeater positions is defined as an iteration. Since in each step (determining the best repeater at node $j$) of an iteration the algorithm improves the cost function, the repeater solution at the end of an iteration generates a lower cost than at the beginning of an iteration. After the first iteration is completed, a second iteration starts by changing the sizes of the repeaters at the possible repeater positions to determine the repeater sizes $h_1$, $h_2$, …, $h_m$ that minimize the cost function. However, in the second iteration, the initial repeater solution is the output of the previous iteration. Thus, at the second iteration (as compared to the first iteration), the capacitive loading and driving resistance at the node at which the best repeater size is sought are closer to the values for minimum cost, enabling the optimum repeater sizes to be more accurately calculated. The iterations are repeated until there is no change in the size of any repeater as compared to the previous iteration. The algorithm typically converges within two or three iterations.

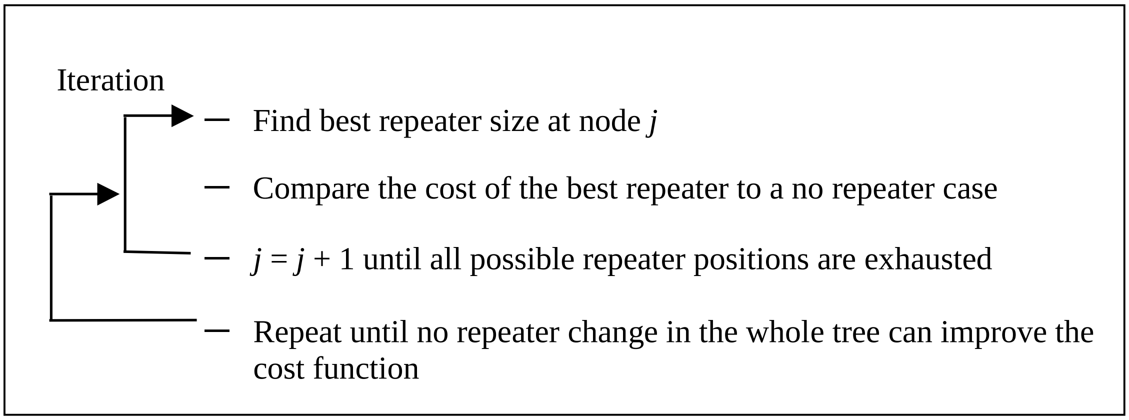

Iteration
– Find best repeater size at node $j$
– Compare the cost of the best repeater to a no repeater case
– $j = j + 1$ until all possible repeater positions are exhausted
– Repeat until no repeater change in the whole tree can improve the cost function

Fig. 2. Proposed algorithm for inserting repeaters in an *RLC* tree.

## *C.* Complexity and Optimality of Proposed Algorithm

The algorithm consists of several iterations. Each iteration scans the $m$ possible repeater positions to determine the repeater sizes that minimize the objective cost. The number of necessary steps to find the repeater size at a possible repeater position which minimizes the cost is denoted $B$ and is on average ten for the typical range of allowable repeater size ($1 \leq h_j \leq h_{max}$). The cost function is evaluated each time the repeater size is changed at each of the $B$ step. Thus, the complexity of an iteration is

$$\Theta(iteration) = O(m \cdot B) \cdot O(evaluating\ the\ cost\ function)\,. \tag{1}$$

The complexity of evaluating the cost function depends upon the delay model used for the drivers and the interconnect. As shown in section III, for the specific delay model used here, the cost function can be evaluated in a time proportional to the number of wires in the tree, $n$. Thus, the complexity of a single iteration is

$$\Theta(iteration) = O(m \cdot n \cdot B)\,. \tag{2}$$

As mentioned previously, the number of iterations for convergence is typically two or three. The memory requirement of the algorithm is proportional to the number of wires, *n*.

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

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

This relation means that the algorithm reaches a minimum in the cost function. There is no guarantee, however, that this minimum is the global minimum. To improve the final repeater solution, the two repeaters at the left and right possible repeater positions of each wire are simultaneously changed. The process of determining two repeater sizes that minimize the cost simultaneously requires $B^2$ steps with the binary search algorithm used here. Since there are *m* / 2 possible repeater position pairs, the complexity of this modified algorithm is

$$\Theta(2^{nd}\ order\ alg) = O(m \cdot n \cdot \frac{B^2}{2})\,. \tag{4}$$

This modified algorithm does not reach the first minimum near the initial point. Rather, the modified algorithm searches for a minimum closer to the global minimum. The price is increased processing time. In general, a set of higher order algorithms can be achieved by simultaneously changing more repeaters. The complexities of these algorithms are

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

The algorithm that changes *m* repeaters simultaneously is guaranteed to reach the global minimum. However, the processing time is exponential with the number of possible repeater positions and is prohibitively high even for relatively small trees. The set of algorithms above has been examined for small trees (seven to eight possible repeater positions) and compared to the exhaustive algorithm that changes all *m* repeaters simultaneously. The results demonstrate that the second order algorithm consistently reaches the global or a near global minimum. The higher order algorithms introduced no or only a slight improvement in the final repeater solution as compared to the second order algorithm. The CPU run time of the second order algorithm is 20 sec on an S/490 IBM machine with one giga byte of RAM for a large tree with 250 possible repeater positions. For typical trees with less than fifty possible repeater positions, the CPU time is less than one second. Hence, the second order algorithm is used in the examples discussed in this paper.

## III. Delay Model

As mentioned in the previous section, the repeater insertion algorithm can be used with any delay model. The specific delay model used in this paper is discussed in this section. In subsection A, the model of the devices (the repeaters) used here is discussed. The method used to combine the repeater model with an *RLC* tree and to calculate the delay is discussed in subsection B.

### *A*. Repeater Model

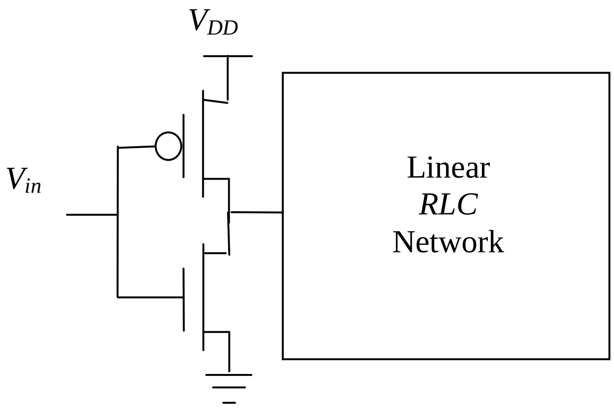


Fig. 3. A symmetric CMOS inverter driving an *RLC* Network.

The problem of evaluating the delay at a sink of a buffered tree simplifies to adding the delay of several structures as shown in Fig. 3 along the path from the input to the sink. The structure shown in Fig. 3 is a symmetric inverter (repeater) driving an *RLC* tree (which is a sub-tree of the original *RLC* tree). Evaluating the delay of such a structure is complicated by a combination of linear and nonlinear elements constituting the circuit. It is common to replace the nonlinear transistors by equivalent linear resistors, *e.g.*, [6], [8], [10], [11], [13]. However, such an approximation strongly affects the final repeater solution, significantly increasing the final cost achieved by the repeater insertion algorithm. Thus, in this subsection, a method [33] is discussed that significantly improves the accuracy of the transistor model as compared to a linear resistor approximation. The proposed method approximates the nonlinear transistor characteristic by a two piecewise linear curve as shown in Fig. 4. Assuming a step input, the input signal is constant at the supply voltage $V_{DD}$ for the entire switching time. Thus, the gate-to-source voltage of the NMOS transistor is $V_{DD}$ and the PMOS transistor is off for the entire switching time. The curve shown in Fig. 4 is the drain-to-source current $I_{DS}$ versus the source-to-drain voltage of the NMOS transistor $V_{DS}$ where $V_{GS}$ is equal to $V_{DD}$.

The method used here calculates the delay of two linear networks, one assuming the transistor operates in the linear region for the entire switching time and the other assuming the transistor operates in the saturation region for the entire switching time. The two linear circuit models used for approximating the transistor in the linear and saturation regions are shown in Fig. 5 (a) and (b), respectively. These linear and saturation transistor models are combined with the *RLC* tree driven by the repeater, resulting in two linear *RLC* networks. A delay value is found for each *RLC* network using a linear network analysis method and are denoted $t_{pdlin}$ and $t_{pdsat}$ for the linear and saturation regions of operation, respectively. The parameters used to define the device model in the linear and saturation regions are $C_{in}$, $R_{lin}$, $C_{out}$, $I_{sat}$, and $R_{sat}$. $I_{sat}$, $R_{sat}$, and $R_{lin}$, respectively, and are shown in Fig. 4. These parameters describe the saturation current of a transistor with $V_{GS}$ equal to $V_{DD}$ and the equivalent output resistance of a transistor in the saturation and linear regions, respectively. $C_{in}$ and $C_{out}$ are the input and output capacitances of the repeater. These parameters are calculated in terms of the corresponding parameters, $C_{in0}$, $R_{lin0}$, $C_{out0}$, $I_{sat0}$, and $R_{sat0}$, of a minimum size symmetric inverter. An inverter $h$ times wider than a minimum size inverter has $C_{in}$=$C_{in0}h$, $R_{lin}$=$R_{lin0}$ / $h$, $C_{out}$=$C_{out0}h$, $I_{sat}$=$I_{sat0}h$, and $R_{sat}$=$R_{sat0}$ / $h$.

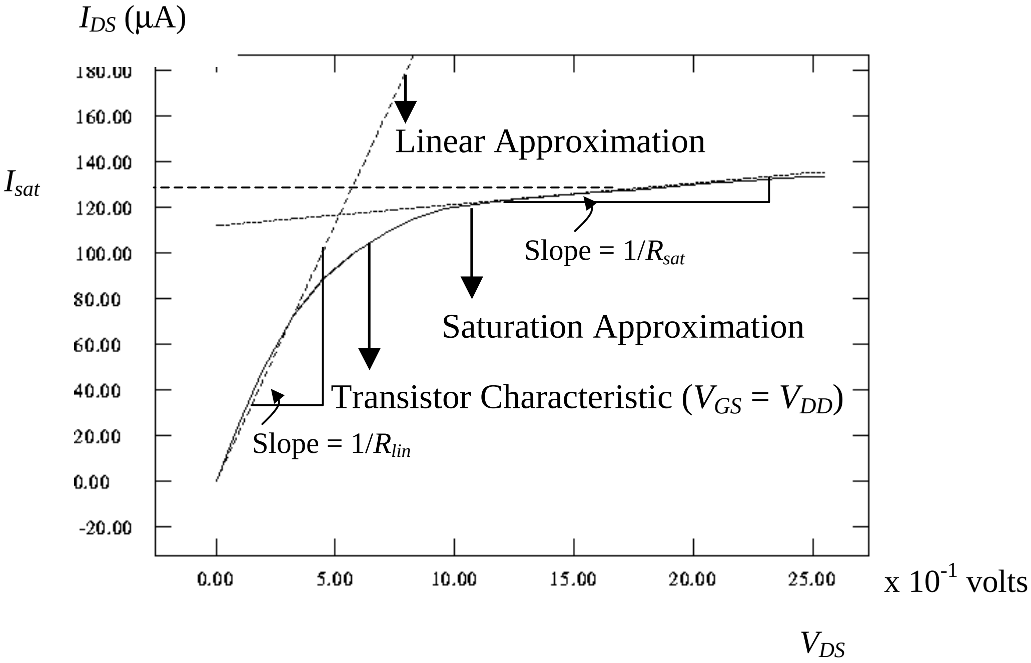


Fig. 4. Piecewise linear approximation of an NMOS transistor for $V_{GS}$ = $V_{DD}$.

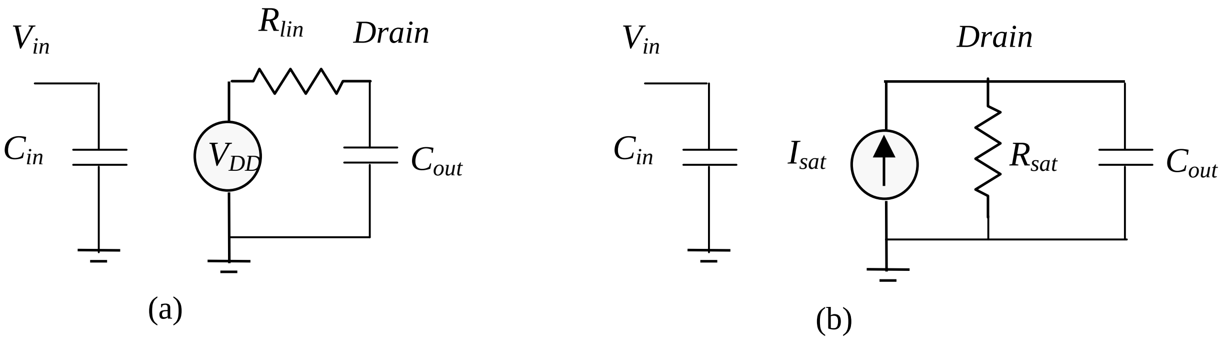


Fig. 5. Equivalent circuit models of an NMOS transistor when operating (a) in the linear region and (b) in the saturation region for $V_{GS}$ = $V_{DD}$.

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 $t_{pdlin}$

has been shown to accurately characterize the propagation delay [33]. The resulting delay for the general case in terms of $t_{pdsat}$ and $t_{pdlin}$ is [33]

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

In general, this method is highly accurate (errors within 3%) for fast input signals. Additional error may result from the linear analysis method used to determine $t_{pdsat}$ and $t_{pdlin}$ of an *RLC* network.

## *B.* Delay of an *RLC* Tree

The linear analysis method used to evaluate the delays $t_{pdsat}$ and $t_{pdlin}$ of the two *RLC* trees resulting from the saturation and linear region approximations, respectively, is described in this subsection. A second order transfer function that approximates the transfer function at a node *i* of an *RLC* tree is introduced in [34] and is

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

The variables $\zeta_i$ and $\omega_{ni}$ that characterize the second order approximation of the transfer function at node *i* are

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

$$\omega_{ni} = \frac{1}{\sqrt{\sum_k C_k L_{ik}}} , \tag{9}$$

where $R_{ik}$ ($L_{ik}$) is the common resistance (inductance) from the input to nodes *i* and *k*. For example, in Fig. 6, $R_{77} = R_1 + R_3 + R_7$, $R_{67} = R_1 + R_3$, and $R_{27} = R_1$. The summation variable *k* operates over all of the capacitors in the circuit.

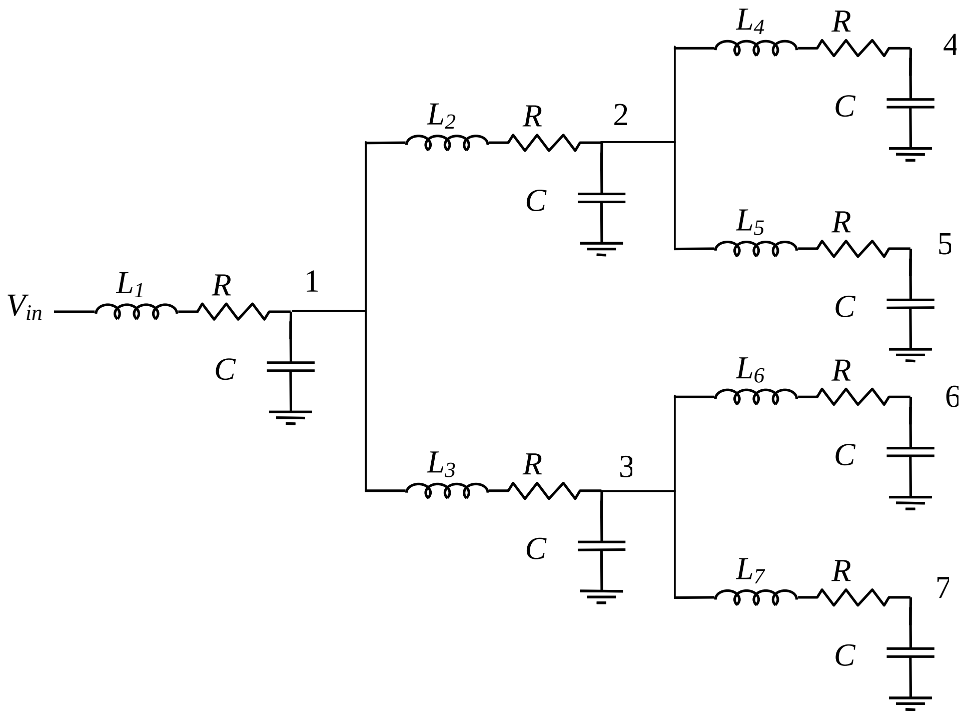


Fig. 6. General *RLC* tree.

The second order approximation is compared in Fig. 7 to AS/X [35] simulations of the output node 7 of the tree shown in Fig. 6. A balanced tree with equal left and right branch impedances is used. The supply voltage is 2.5 volts. Note the accuracy that the second order approximation exhibits as compared to AS/X simulations for the case of a balanced tree. If the tree is unbalanced, the second order approximation is less accurate. The accuracy characteristics of this solution is similar to the Elmore [36] (Wyatt [37]) delay model for *RC* trees [34].

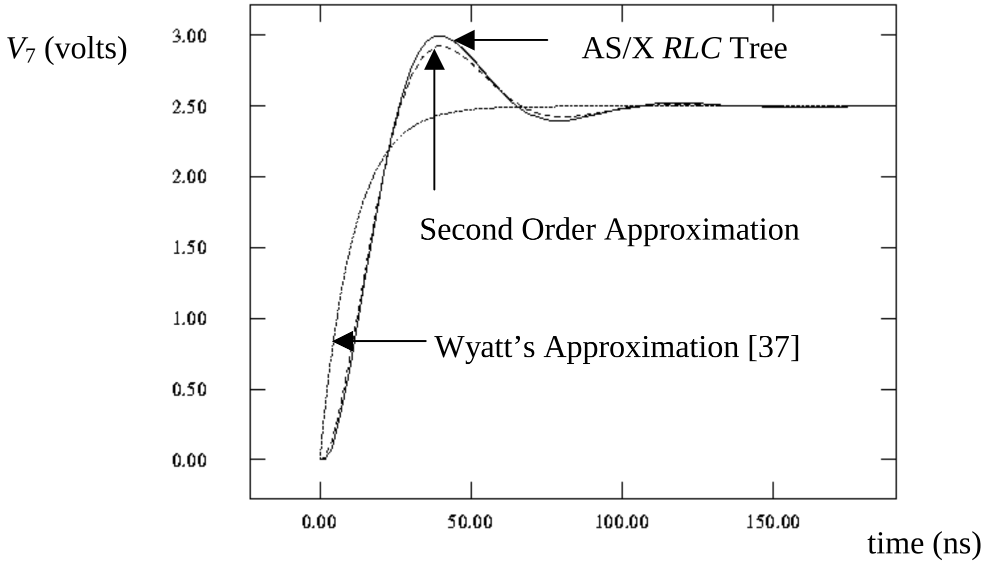


Fig. 7. AS/X simulations of the *RLC* tree shown in Fig. 6 as compared to the second order approximation and the Wyatt *RC* model.

The 50% propagation delay and the 10% to 90% rise time of the signal at node *i* of an *RLC* tree are given in closed form in [34] for a step input and are

$$t_{pdi} = (1.047 e^{-\frac{\zeta_i}{0.85}} + 1.39\zeta_i) / \omega_{ni} , \tag{10}$$

$$t_{ri} = (6.017 e^{-\frac{\zeta_i^{1.35}}{0.4}} - 5 e^{-\frac{\zeta_i^{1.25}}{0.64}} + 4.39\zeta_i) / \omega_{ni} . \tag{11}$$

The error in these expressions is less than 3% for balanced trees. The error can exceed 20% for highly unbalanced trees [34].

Referring to (8) and (9), evaluating the delay and rise time at node *i* depends on evaluating two summations at node *i* which are

$$T_{RCi} = \sum_k C_k R_{ik} , \tag{12}$$

$$T_{LCi}^2 = \sum_k C_k L_{ik} . \tag{13}$$

These two summations can be rewritten as

$$T_{RCi} = \sum_k C_{Tk} R_k , \tag{14}$$

$$T_{LCi}^2 = \sum_k C_{Tk} L_k , \tag{15}$$

where the summation index *k* operates over all of the wires that belongs to the path from the input to node *i*. $R_k$ and $L_K$ are the resistance and inductance of wire *k*. $C_{Tk}$ is the total capacitance seen at the beginning of wire *k*. For example, in Fig. 6, $T_{RC7} = R_1(C_1+C_2+\ldots+C_7) + R_6(C_3+C_6+C_7) + R_7C_7$. This form of expressing the summations is computationally efficient since these summations can be calculated recursively at all of the nodes of an *RLC* tree in a time linearly proportional with the number of branches in the tree [8], [38], [39].

The summations in (14) and (15) of a tree rooted at section $w_1$ are calculated in two steps. The first step is to calculate the total load capacitance of each section. Pseudo-code of the procedure that performs this task is provided in Fig. 8.

```
float Cal_Cap_Loads (section w)
{
        if(right(w)=0 and left(w)=0)     /* if w is a leaf */
                return w.C;

        if(right(w)≠0)
                $C_{TR}$=Cal_Cap_Loads(right(w));
        else
                $C_{TR}$=0;        /* No right branch is driven by w */

        if(left(w)≠0)
                $C_{TL}$=Cal_Cap_Loads(left(w));
        else
                $C_{TL}$=0;        /* No left branch is driven by w */

        w.$C_T$=$C_{TR}$+$C_{TL}$;

        return w.$C_T$;
}
```

Fig. 8. Pseudo-code for calculating the total load capacitance at all of the sections of a tree

The function is initially called by Cal_Cap_Loads($w_1$) and recursively calculates the capacitive load at each section. *w.C* is the capacitance of the section *w*. The functions, left(*w*) and right(*w*), return the left and right sections driven by *w*, respectively. If no left (right) section is driven by *w*, left(*w*)=0 (right(*w*)=0). If *w* is a leaf, left(*w*)=0 and right(*w*)=0. The time required to calculate the total capacitive load of each section is proportional to the number of *RLC* sections in the tree *m* and requires no multiplication operations. Note that a binary branching factor is assumed without loss of generality since any general tree can be transformed into a binary tree by inserting wires with zero impedances [8], [13].

The second step is to calculate and store the summations in (14) and (15) at the nodes of the tree. The function performing this task is described in Fig. 9. The function is initially called by Cal_Summations($w_1$,0,0). *w.R* and *w.L* are the resistance and inductance of section *w*, respectively. The computational time required to calculate the summations is proportional to the number of *RLC* sections in the tree, *m*. The total number of multiplications required to evaluate the second order approximation at all of the nodes of an *RLC* tree is 2*m*. Alternatively, the number of multiplications is equal to the order of the characteristic equation describing the *RLC* tree since the order of an *RLC* tree with *m* *RLC* sections is 2*m* (each *RLC* section has an inductor and a capacitor).

```
Cal_Summations(section w, float $T_{Rcprev}$, float $T_{LCprev}$)
{
    $T_{RC}=T_{Rcprev}+w.R*w.C_T$;
    $T_{LC}=T_{Rcprev}+w.L*w.C_T$;


    if(right(w)$\neq$0)
        Cal_ Summations (right(w),$T_{RC}$,$T_{LC}$);

    if(left(w)$\neq$0)
        Cal_ Summations (left(w),$T_{RC}$,$T_{LC}$);
}
```

Fig. 9. Pseudo-code for calculating the delays at the sinks of an *RLC* tree.

## IV. Results and Discussion

The results of applying the CAD-based repeater insertion tool to several industrial copper-based interconnect trees are summarized and discussed in this section. The *RLC* trees described in this paper are copper 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 circuits. Long wires within the trees are partitioned with a maximum segment length of 0.5 mm to permit repeaters to be inserted within these long wires for improved performance [13].

A repeater solution is determined to minimize the maximum path delay of each tree based on the *RLC* delay model discussed in the previous section. The total area of the repeaters inserted within each tree is described in terms of the area of a minimum size repeater. The tool also generates an AS/X [35] input file which is used to simulate the maximum path delay and the power consumption of the buffered *RLC* tree. The total inserted repeater 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 repeater area, the maximum path delay, and the power consumption of the buffered *RLC* trees when inductance is neglected and repeaters are inserted 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 repeater insertion is not employed. These results are listed in Table 1 as well.

Two important trends can be observed from the data listed in Table 1. The first trend is that inserting repeaters significantly reduces the maximum path delay as compared to the maximum path delay of an unbuffered tree. This behavior illustrates the importance of repeater insertion as an effective methodology to reduce interconnect delay. According to Tables 2 and 3, the average saving in the maximum path delay when inserting repeaters based on an *RLC* model as compared to an unbuffered tree is about 40% where 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 repeaters based on an *RLC* model as compared to an *RC* model consistently introduces savings in all of the three primary design criteria: area, power, and delay. This behavior demonstrates the importance of including inductance in a high speed repeater insertion methodology. According to Table 3, including inductance in the interconnect model saves an average 40.8% of the repeater 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 repeater area when including inductance in the interconnect model is due to the quadratic dependence of the delay on the length of an *RC* wire which tends to a linear dependence as inductance effects increase [40]. The 50% delay of an *RC* line is given by $0.35RCl^2$ [1], [6], [11] and by $l\sqrt{LC}$ [40] for an *LC* line when the line is driven by an ideal source with 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. Inserting repeaters in an *LC* line only degrades 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 repeater area for minimum propagation delay is between the maximum repeater area in the *RC* case and the zero repeater area in the *LC* case. The repeater 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 [40]. Hence, inserting repeaters based on an *RC* model and neglecting inductance results in larger repeater area than necessary to achieve a minimum delay. The magnitude of the excess repeater area when using an *RC* model depends upon the relative magnitude of the inductance within the tree. For the specific copper-based interconnect *RLC* trees used here, almost half the repeater 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, repeaters are inserted in a tree to decouple 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 ($\sqrt{\sum_k C_k L_{ik}}$) [34], which has a square root behavior as compared to the linear behavior of an *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 repeaters 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 repeater area. The dynamic power consumption, which is linearly dependent on the total capacitance of the interconnect and the repeaters, decreases due to the reduced input and output capacitance of the repeaters. The short-circuit power consumption is significantly less for a smaller repeater since the short-circuit power consumed by a CMOS inverter is quadratically dependent on the width of the repeater [41]-[43]. The decreased delay achieved by including inductance is due to more accurate modeling of the interconnect thereby enabling improved repeater insertion which eliminates the excess repeater area that would result when using an *RC* interconnect model. This excess repeater area increases the total delay due to the increased gate capacitance.

The optimum number of sections $k_{opt}$ that an *RLC* line should be partitioned into and the size of each inserted repeater $h_{opt}$ to achieve the minimum total propagation delay have been characterized in [40] and are

$$h_{opt} = \sqrt{\frac{R_0 C_t}{R_t C_0}} \frac{1}{\left[1 + 0.16 (T_{L/R})^3\right]^{0.24}}, \tag{16}$$

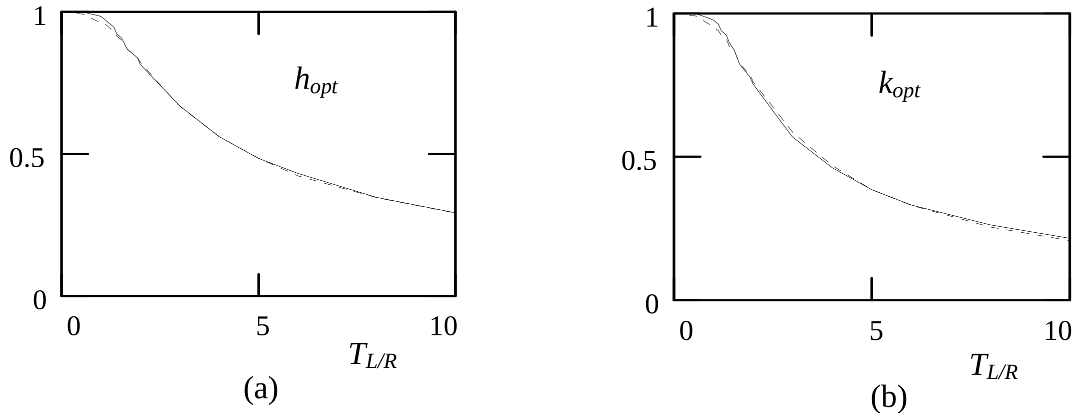
and

$$k_{opt} = \sqrt{\frac{R_t C_t}{2 R_0 C_0}} \frac{1}{\left[1 + 0.18 (T_{L/R})^3\right]^{0.3}}. \tag{17}$$

where

$$T_{L/R} = \sqrt{\frac{L_t / R_t}{R_0 C_0}}. \tag{18}$$

$R_0$ and $C_0$ are the output resistance and input capacitance of a minimum size repeater, respectively, and $R_t$, $L_t$, and $C_t$ are the total resistance, inductance, and capacitance of the line, respectively. Note in (16) and (17) that $h_{opt}$ and $k_{opt}$ are equivalent to the expressions in [10], [11] for an *RC* line when $T_{L/R}$ is equal zero ($L_t$ = 0). The error factors in the optimum size of each repeater and the optimum number of sections as compared to the corresponding optimum repeater expressions based on an *RC* interconnect model are plotted in Fig. 10. Both the size and number of the repeaters decrease as the inductance effects increase.



(a)

(b)

Fig. 10. The error factors in a) the optimum size of each repeater $h_{opt}$ and b) the optimum number of sections $k_{opt}$, respectively, as compared to the corresponding optimum repeater expressions based on an *RC* interconnect model.

Another interesting aspect of (16), (17), and (18) is that $T_{L/R}$ increases as the time constant $R_0C_0$ decreases, or alternatively, as faster repeaters are used. An increase in $T_{L/R}$ increases the discrepancy between an *RC* model and an *RLC* model as described by (16) and (17) even if the same interconnect trees are buffered to minimize the path delay. Thus, the analytical solutions in (16), (17), and (18) anticipate additional savings in repeater area by including inductance in the interconnect model as compared to an *RC* model for technologies with faster devices. To verify this trend, five times faster devices than the 0.25 μm devices are used as repeaters to minimize the maximum path delays for the same set of trees listed in Table 1. The results corresponding to the data listed in Table 1 are listed in Table 4. Note that the savings in area, power, and delay increases 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. The average savings increases from 40.8% to 62.2% for the repeater 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 criteria: area, power, and delay. Therefore, with technology scaling, the issue of 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 previously discussed, the minimum total propagation delay can be achieved for such a line by not inserting repeaters 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 repeaters that are inserted will increase as the repeaters become faster since there is less of a penalty for inserting more repeaters. Thus, the discrepancy between the repeater solutions based on an *RC* and an *RLC* model (zero repeater area for dominant inductance effects) increases as faster repeaters are used. In general, the area required by the repeaters to minimize the total propagation delay based on an *RC* model as compared to an *RLC* model increases more rapidly as the devices become faster.

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. The area is generated by the repeater insertion program while the power and maximum path delay are simulated using AS/X.

| Tree Name | Area<br>(minimum size inverters) | | | Power<br>(pJ per Cycle) | | | Maximum Delay<br>(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. Percentage savings in area, power, and maximum path delay introduced by inserting repeaters based on an *RLC* model rather than an *RC* model. The percentage savings in delay when inserting repeaters as compared to an unbuffered tree are also listed.

| Tree Name | Per cent savings in delay of a buffered tree based on an *RLC* model as compared to an un-buffered tree | Per cent savings in the area of repeaters inserted based on an *RLC* model as compared to using an *RC* model | Per cent savings in power dissipation when repeaters are inserted based on an *RLC* model as compared to using an *RC* model | Per cent savings in delay when repeaters are inserted based on an *RLC* model as compared to using an *RC* model |
|---|---|---|---|---|
| TSs1 | 40.9 | 7.3 | 6.9 | 3 |
| L1 | 21.9 | 59.1 | 18.6 | 1.8 |
| TS2 | 0 | 100 | 32.26 | 0.26 |
| L2 | 37.6 | 8 | 3.2 | 3.7 |
| L3 | 0 | 100 | 24.75 | 10.4 |
| TSm1 | 31 | 27 | 14.3 | 5.9 |
| TSm2 | 18.9 | 40.7 | 4.7 | 6.2 |
| TSL | 32.2 | 29.8 | 13.6 | 4 |
| TSL1 | 51 | 27 | 13.8 | 5.9 |
| TGs1 | 11.8 | 22.9 | 11.4 | 2.1 |
| TGs2 | 0 | 100 | 32.77 | 14.5 |
| TGm1 | 37.9 | 31.4 | 13.6 | 11.5 |
| TGL1 | 76 | 49.5 | 21.9 | 11.9 |

Table 3. The total repeater area, total power, and total maximum path delay of all of the trees. The per cent savings shown here represent the average savings in area, power, and maximum path delay when using an *RLC* model for repeater insertion.

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

Table 4. Simulation results of unbuffered trees, buffered trees based on an *RLC* model, and buffered trees based on an *RC* model with five times faster devices. The area, power, and maximum path delay are compared. The area is generated by the repeater insertion program while the power and maximum path delay are simulated using 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 | 1349 | 1997 | 13.86 | 21.4 | 24.4 | 488 | 144 | 145 |
| L1 | 0 | 569 | 1168 | 8.15 | 12.2 | 14.44 | 342 | 164 | 166 |
| TS2 | 0 | 740 | 2738 | 25.67 | 34 | 62 | 193 | 154 | 165 |
| L2 | 0 | 1137 | 1862 | 11.92 | 19.4 | 22.4 | 700 | 248 | 258 |
| L3 | 0 | 534 | 1799 | 22.8 | 28 | 40 | 213 | 206 | 218 |
| TSm1 | 0 | 5150 | 13468 | 95 | 177 | 348 | 389 | 222 | 240 |
| TSm2 | 0 | 7107 | 21654 | 135 | 482 | 1516 | 343 | 238 | 262 |
| TSL | 0 | 12819 | 26674 | 147.5 | 382 | 832 | 431 | 220 | 240 |
| TSL1 | 0 | 9358 | 35844 | 164 | 242 | 688 | 781 | 268 | 308 |
| TGs1 | 0 | 2152 | 6392 | 38 | 60.8 | 115 | 262 | 198 | 224 |
| TGs2 | 0 | 2402 | 4410 | 40.20 | 77.6 | 138.8 | 212 | 187 | 262 |
| TGm1 | 0 | 5738 | 15184 | 89.1 | 141 | 302 | 460 | 212 | 232 |
| TGL1 | 0 | 18905 | 37037 | 201 | 330 | 588 | 1740 | 346 | 378 |

Table 5. Percentage savings in area, power, and maximum path delay introduced by inserting repeaters based on an *RLC* model rather than an *RC* model. The devices used for the repeaters are from a five times faster technology as compared to the 0.25 μm CMOS technology used to generate the data listed in Table 2. The percentage savings in delay when inserting repeaters as compared to an unbuffered tree are also listed.

| Tree Name | Per cent savings in delay of a buffered tree based on an *RLC* model as compared to an un-buffered tree | Per cent savings in the area of repeaters inserted based on an *RLC* model as compared to using an *RC* model | Per cent savings in power dissipation when repeaters are inserted based on an *RLC* model as compared to using an *RC* model | Per cent savings in delay when repeaters are inserted based on an *RLC* model as compared to using an *RC* model |
|---|---|---|---|---|
| TSs1 | 70.5 | 32.4 | 12.2 | 0.68 |
| L1 | 52 | 51 | 15.5 | 1.2 |
| TS2 | 20 | 72.9 | 45 | 6.6 |
| L2 | 37.6 | 38.9 | 13.39 | 3.8 |
| L3 | 3.2 | 70.31 | 28 | 5.5 |
| TSm1 | 43 | 61.7 | 49.1 | 7.5 |
| TSm2 | 30.6 | 67 | 68 | 9.2 |
| TSL | 49 | 52 | 54 | 8.3 |
| TSL1 | 65.7 | 74 | 64.8 | 21.4 |
| TGs1 | 24.4 | 66.3 | 47 | 11.6 |
| TGs2 | 11.7 | 45.5 | 44.1 | 28.62 |
| TGm1 | 53.9 | 62.2 | 53.3 | 8.6 |
| TGL1 | 80 | 49 | 43 | 10.8 |

Table 6. The total repeater area, total power, and total maximum path delay of all of the trees using five times faster devices. The per cent savings shown here represent the average savings in area, power, and maximum path delay when using an *RLC* model for repeater insertion.

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

## V. Summary

The effect of inductance on repeater insertion in *RLC* trees is investigated in this paper. An algorithm is introduced to insert and size repeaters within an *RLC* tree to minimize a variety of possible cost functions. The algorithm has a polynomial complexity proportional to the square of the number of possible repeater positions and determines a repeater solution that is reasonably close to the global minimum. It is shown that as inductance effects increase, both the number of repeaters and the size of each repeater decrease. This trend means significantly less repeater area and power consumption due to decreased repeater capacitance. Also, less cost can be attained by including inductance in the design methodology rather than using an *RC* model since the interconnect is modeled more accurately. Hence, it is shown that including inductance in a repeater insertion design methodology as compared to using an *RC* model improves the overall repeater solution in terms of area, power, and delay. The average savings in area, power, and delay for the set of trees used in this paper are 40.8%, 15.6%, and 6.7%, respectively, when inserting repeaters based on an *RLC* delay model as compared to an *RC* delay model with repeaters from a 0.25 μm CMOS technology and copper interconnect. The average savings in area, power, and delay increases to 62.2%, 57.2%, and 9.4%, respectively, when using repeaters from a five times faster technology with the same set of interconnect trees.

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

1- Lower resistivity metal alloys for interconnect, copper interconnect being a primary example [30]-[32].
2- Lower permeability dielectrics to insulate the interconnect which reduces the interconnect capacitance. Reducing the interconnect capacitance increases the effects of inductance [27].
3- Higher operating frequencies [19]-[21], [27].
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 in VLSI circuits to meet higher frequency targets which require more accurate delay models.

Therefore, it is imperative that inductance be included in the interconnect impedance model when inserting repeaters to drive *RLC* trees in high speed circuits.

## References

[1] J. M. Rabaey, *Digital Integrated Circuits, A Design Perspective,* Prentice Hall, Inc., New Jersey, 1996.

[2] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design,* Addison-Wesley Publishing Co., New York, 1993.

[3] S. M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits Design and Analysis*, McGraw-Hill, Inc., New York, 1996.

[4] S. Bothra, B. Rogers, M. Kellam, and C. M. Osburn "Analysis of the Effects of Scaling on Interconnect Delay in ULSI Circuits," *IEEE Transactions on Electron Devices*, Vol. ED-40, No. 3, pp. 591 - 597, March 1993.

[5] R. J. Antinone and G, W. Brown, "The Modeling of Resistive Interconnects for Integrated Circuits," *IEEE Journal of Solid-State Circuits*, Vol. SC-18, No. 2, pp. 200 - 203, April 1983.

[6] 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.

[7] J. Cong, L. He, C-K. Koh, and P. Madden, "Performance Optimization of VLSI Interconnect," *Integration*, Vol. 21, pp. 1 - 94, November 1996.

[8] 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.

[9] V. Adler and E. G. Friedman, "Delay and Power Expressions for a CMOS Inverter Driving a Resistive-Capacitive Load," *Analog Integrated Circuits and Signal Processing*, Vol. 14, No. 1/2, pp. 29 - 39, September 1997.

[10] 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.

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

[12] 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.

[13] C. J. Alpert and A. Devgan, "Wire Segmenting for Improved Buffer Insertion," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 649-654, June 1997.

[14] S. Dhar and M. A. Franklin, "Optimum Buffer Circuits for Driving Long Uniform Lines," *IEEE Journal of Solid-State Circuits*, Vol. SC-26, No. 1, pp. 32 - 40, January 1991.

[15] D. A. Priore, "Inductance on Silicon for Sub-Micron CMOS VLSI," *Proceedings of the IEEE Symposium on VLSI Circuits*, pp. 17-18, May 1993.

[16] D. B. Jarvis, "The Effects of Interconnections on High-Speed Logic Circuits," *IEEE Transactions on Electronic Computers*, Vol. EC-10, No. 4, pp. 476 - 487, October 1963.

[17] M. P. May, A. Taflove, and J. Baron, "FD-TD Modeling of Digital Signal Propagation in 3-D Circuits with Passive and Active Loads," *IEEE Transactions on Microwave Theory*

3-D Circuits with Passive and Active Loads," *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-42, No. 8, pp. 1514 - 1523, August 1994.

[18] Y. Eo and W. R. Eisenstadt, "High-Speed VLSI Interconnect Modeling Based on S-Parameter Measurement," *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, Vol. CHMT-16, No. 5, pp. 555 - 562, August 1993.

[19] A. Deutsch, *et al.*, "High-Speed Signal Propagation on lossy transmission lines," *IBM Journal of Research and Development*, Vol. 34, No. 4, pp. 601 - 615, July 1990.

[20] A. Deutsch, *et al.*, "Modeling and Characterization of Long Interconnections for High-Performance Microprocessors," *IBM Journal of Research and Development*, Vol. 39, No. 5, pp. 547 - 667, September 1995.

[21] 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.

[22] M. Shoji, *High-Speed Digital Circuits,* Addison Wesley, Massachusetts, 1996.

[23] A. Duetsch, A. Kopcsay, and G. V. Surovic, "Challenges Raised by Long On-Chip Wiring for CMOS Microprocessors," *Proceedings of the IEEE Topical Meeting on Electrical Performance of Electronic Packaging,* pp. 21 – 23, October 1995.

[24] Y. Massoud, S. Majors, T. Bustami, and J. White, "Layout Techniques for Minimizing On-Chip Interconnect Self Inductance," *Proceedings of the IEEE/ACM Design Automation Conference,* pp. 566 – 571, June 1998.

[25] B. Krauter and S. Mehrotra, "Layout Based Frequency Dependent Inductance and Resistance Extraction for On-Chip Interconnect Timing Analysis," *Proceedings of the IEEE/ACM Design Automation Conference,* pp. 303 – 308, June 1998.

[26] A. Duetsch, *et al.*, "Design Guidelines for Short, Medium, and Long On-Chip Interconnect," *Proceedings of the IEEE Topical Meeting on Electrical Performance of Electronic Packaging,* pp. 30 – 32, October 1996.

[27] 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.

[28] L. T. Pillage, "Coping with RC(L) Interconnect Design Headaches," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 246 – 253, September 1995.

[29] E. G. Friedman, *High Performance Clock Distribution Networks*, Kluwer Academic Publishers, Massachusetts, 1997

[30] J. Torres, "Advanced Copper Interconnections for Silicon CMOS Technologies," *Applied Surface Science*, Vol. 91, No. 1, pp. 112 - 123, October 1995

*Surface Science,* Vol. 91, No. 1, pp. 112 - 123, October 1995.

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

[32] K. K. Likharev and V. K. Semenov, "RSFQ Logic/Memory Family: A New Josephson-Junction Technology for Sub-Terahertz-Clock Frequency Digital System," *IEEE Transactions on Applied Superconductivity,* Vol. AS-1, No. 1, pp. 3 - 28, March 1991.

[33] 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.

[34] 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.

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

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

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

[38] J. Rubinstein, P. Penfield, and M. Horowitz, "Signal Delay in *RC* Tree Networks," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-2, No. 3, pp. 202 - 211, July 1983.

[39] 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.

[40] 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.

[41] 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.

[42] S. R. Vemuru and N. Scheinberg "Short-Circuit Power Dissipation Estimation for CMOS Logic Gates," *IEEE Transactions on Circuits and Systems,* Vol. CAS-41, No. 11, pp. 762 - 765, November 1994.

[43] Y. I. Ismail, E. G. Friedman, and J. L. Neves," Dynamic and Short-Circuit Power of CMOS Gates Driving Lossless Transmission Lines," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,* Vol. CAS-46, No. 8, pp. 950 - 961, August 1999.