Fast and Accurate Simulation of Tree Structured Interconnect[1]

Yehea I. Ismail
Department of Electrical and Computer Engineering
Northwestern University
Evanston, IL 60208

Eby G. Friedman
Department of Electrical and Computer Engineering
University of Rochester
Rochester, New York 14627

***Abstract* – A method is introduced to evaluate time domain signals within *RLC* trees with arbitrary accuracy in response to any input signal. This method depends on finding a low frequency reduced order transfer function by direct truncation of the exact transfer function at different nodes of an *RLC* tree. The method is numerically accurate for any order of approximation, which permits approximations to be determined with a large number of poles appropriate for approximating *RLC* trees with underdamped responses. The method is computationally efficient with a complexity linearly proportional to the number of branches in an *RLC* tree. A common set of poles are determined that characterize the responses at all of the nodes of an *RLC* tree which further enhances the computational efficiency. Stability is guaranteed by the DTT method for low order approximations with less than 5 poles. Such low order approximations are useful for evaluating monotone responses exhibited by *RC* circuits.**

I. Introduction

It has become well accepted that interconnect delay dominates gate delay in current deep submicrometer VLSI circuits [1]-[5]. With the continuous scaling of technology and increased die area, this situation is becoming worse [6]-[11]. In order to properly design complex circuits, accurate characterization and simulation of the interconnect behavior and signal transients are required. This high accuracy is necessary for analyzing performance critical modules and nets and to accurately anticipate possible hazards during switching. Also, increasing performance requirements have forced a reduction of the safety margins used in worst case design, requiring more accurate interconnect delay characterization. Thus, the process of characterizing signal waveforms in tree structured interconnect (or nearly tree structured) is of primary importance since most interconnect in a VLSI circuit is tree structured [12]-[14].

AWE (Asymptotic Waveform Evaluation) based algorithms [15]-[21] have gained popularity as a more accurate delay model as compared to the Elmore delay model. AWE uses moment matching to determine a set of low frequency dominant poles that approximate the transient response at the nodes of an *RLC* tree. However, AWE suffers two primary problems [16]-[20]. The first problem is that the AWE method can lead to an approximation with unstable poles even for low order approximations [16]-[20]. The second problem is that AWE becomes numerically unstable for higher order approximations which limits the order of the approximations determined using AWE to less than approximately eight poles (of which some poles may be unstable and are discarded) [16]-[20]. This limited number of poles is inappropriate for evaluating the transient response of an underdamped *RLC* tree which requires a much greater number of poles to accurately capture the transient response at all of the nodes. To overcome this limitation, a set of model order reduction algorithms have been developed to determine higher order approximations appropriate for *RLC* circuits based on the state space representation of an *RLC* network. Examples are Pade via Lanczos (PVL) [22], Matrix Pade via Lanczos (MPVL) [23], Arnoldi Algorithms [24], Block Arnoldi Algorithms [25], Passive Reduced-Order Interconnect Macromodeling Algorithm (PRIMA) [26], [27], and SyPVL Algorithm [28]. However, these model order reduction techniques have significantly higher computational complexity than AWE. The complexity of these techniques is super linear with *n*, where *n* is the order of the *RLC* tree and is equal to the total number of capacitors and inductors in the tree. This high complexity is due to these model order reduction techniques solving *n* linear equations in *n* variables several times [22]-[28]. This complexity is much higher than the complexity of AWE which is linearly proportional to *n* for an *RLC* tree [16]-[20]. Note that *n* can be on the order of thousands for a typical large industrial *RLC* circuit.

The objective of this paper is therefore to introduce a new method [29] for evaluating the transient response at the nodes of a general *RLC* tree capable of determining high order approximations appropriate for underdamped *RLC* trees in a computationally efficient manner (complexity linear with *n*). A single line as a special case of a tree with only one output (or sink) is included in this tree analysis methodology. This new method also has improved stability properties for low order approximations as compared to AWE, a useful feature with *RC* trees which do not require higher order approximations. The rest of the paper is organized as follows. A description of the DTT method is provided in section II. The transient responses based on the DTT method for several *RC* and *RLC* trees are compared to SPICE simulations in section III. Finally, some conclusions are offered in section IV.

II. The DTT Method

The concepts used to develop the DTT method are explained in this section. The rules governing the poles and zeros in an *RLC* tree are defined in subsection A. The method used to calculate the exact transfer functions at the nodes of an *RLC* tree is introduced in subsection B. The use of transfer function truncation to determine a reduced order approximation is discussed in subsection C as well as the stability and complexity of the reduced order system.

A. Pole-Zero Behavior in *RLC* Trees

The poles and zeros of an *RLC* tree maintain specific relations to the poles and zeros of the subtrees forming the *RLC* tree. These rules are established in this subsection and are used in the following subsection to develop an algorithm to determine the poles and zeros of a general *RLC* tree by recursively subdividing the tree into smaller subtrees.

Rule 1: *The poles of an RLC circuit are zeros of the impedance seen at the input of the circuit.*

This rule can be understood by referring to Fig. 1 and noting that the transfer functions describing the capacitor voltages and inductor currents have a common denominator (the characteristic equation of the tree) [30]-[34]. Thus, the transfer function at an arbitrary node *i* of an *RLC* tree and the input admittance of the tree are given by

$$\frac{V_i(s)}{V_{in}(s)} = \frac{N_i(s)}{D(s)}, \tag{1}$$

$$Y_{in}(s) = \frac{I_{in}(s)}{V_{in}(s)} = \frac{N_{I_{in}}(s)}{D(s)}, \tag{2}$$

respectively, where $N_i(s)$ and $N_{Iin}(s)$ are functions of *s* dependent on the circuit structure and *D*(*s*) is the common denominator of the circuit. The input impedance is

$$Z_{in}(s) = \frac{V_{in}(s)}{I_{in}(s)} = \frac{D(s)}{N_{Iin}(s)}. \tag{3}$$

Thus, the common denominator of an *RLC* circuit is also the numerator of the input impedance, proving rule 1.

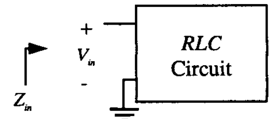


Fig. 1. A general *RLC* circuit.

Rule 2: *The poles of an RLC circuit driven at node x are zeros of the transfer function at node x.*

This rule can be explained by referring to Fig. 2. Note that the *RLC* circuit 2 is driven by the *RLC* circuit 1 at node *x*. Applying rule 1, Z_{in2} is a short-circuit between node *x* and the ground at frequencies equal to the poles of circuit 2. Hence, $V_x(s)$ is equal to zero when *s* is equal to the poles of circuit 2, *i.e.*, the poles of circuit 2 are zeros of the transfer function at node *x*.

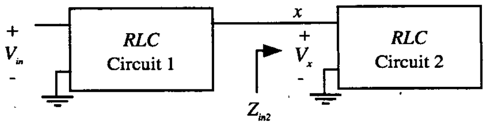


Fig. 2. A general *RLC* circuit composed of two *RLC* subcircuits connected together.

[1] Patent pending

Proc. 43rd IEEE Midwest Symp. on Circuits and Systems, Lansing MI, Aug 8-11, 2000
0-7803-6475-9/00/$10.00 ©IEEE 2000

Rule 3: *The poles of an RLC circuit driven at node x are zeros of the transfer functions at all of the nodes of parallel RLC circuits driven at x.*

This rule can be explained by referring to Fig. 3. The *RLC* subcircuits 2, 3, ..., *k* are driven by *RLC* subcircuit 1 at node *x*. Applying rule 1, Z_{in2} is a short-circuit at frequencies equal to the poles of circuit 2. Hence, $V_x(s)$ is equal to zero and all of the current supplied by circuit 1 is sunk to ground by Z_{in2} when *s* is equal to the poles of circuit 2. Since $V_x(s)$ is equal to zero and no current is supplied to the subcircuits 3, ..., *k* when *s* is equal to the poles of circuit 2, the voltages at all of the nodes of subcircuits 3, ..., *k* are equal to zero. Alternatively, the poles of circuit 2 are zeros of the transfer functions at all of the nodes of the parallel subcircuits driven at node *x*. The same is true for the poles of subcircuits 3, ..., *k* which are zeros of the transfer functions at all of the nodes of the parallel subcircuits driven at node *x*.

As an example, consider the *RLC* tree shown in Fig. 4. The *RLC* section 1 drives the two parallel *RLC* sections 2 and 3. The transfer functions at nodes *x*, 2, and 3 are given by

$$\frac{V_x(s)}{V_{in}(s)}=\frac{(1+R_2C_2s+L_2C_2s^2)(1+R_3C_3s+L_3C_3s^2)}{D}, \qquad (4)$$

$$\frac{V_2(s)}{V_{in}(s)}=\frac{(1+R_3C_3s+L_3C_3s^2)}{D}, \qquad (5)$$

$$\frac{V_3(s)}{V_{in}(s)}=\frac{(1+R_2C_2s+L_2C_2s^2)}{D}, \qquad (6)$$

respectively, where *D* is the common denominator and is a polynomial in *s* of order six. The specific form of *D* is not of interest here. The denominators of subcircuits 2 and 3 are $1 + R_2C_2s + L_2C_2s^2$ and $1 + R_3C_3s + L_3C_3s^2$, respectively. Note that both denominators appear as multipliers in the numerator of the transfer function at node *x* showing that the poles of subcircuits 2 and 3 are zeros of the transfer function at the driving node *x* in accordance with rule 2. Note also that the poles of subcircuit 2 are zeros of the transfer function at node 3 and vice versa, verifying rule 3.

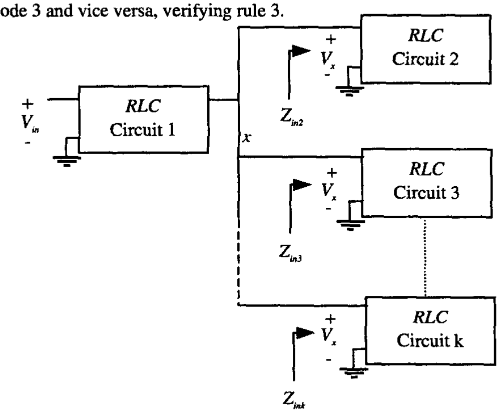


Fig. 3. A general *RLC* circuit composed of an *RLC* subcircuit driving several subcircuits connected in parallel.

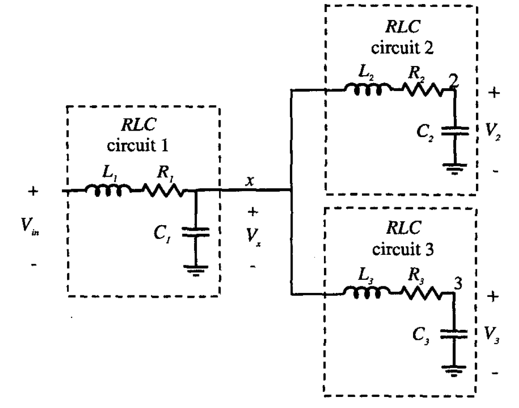


Fig. 4. An *RLC* tree composed of three *RLC* sections.

B. Calculating the Transfer Functions at the Nodes of an *RLC* Tree

It is illustrated in this subsection how to recursively calculate the transfer functions at the nodes of an *RLC* tree using the concepts developed in the previous subsection. Consider the general *RLC* tree shown in Fig. 5. The current sunk to ground by a capacitor *k* is given by $C_k dv_k(t)/dt$ where $v_k(t)$ is the voltage across C_k. Thus, the current passing through the resistance R_1 and the inductance L_1 is given by

$$i_1(t)=\sum_k C_k\frac{dv_k(t)}{dt}, \qquad (7)$$

where the summation index *k* operates over all of the capacitors in the tree. The voltage drop across R_1 and L_1 is given by

$$v_{in}(t)-v_1(t)=R_1i_1(t)+L_1\frac{di_1(t)}{dt}=R_1\sum_k C_k\frac{dv_k(t)}{dt}+L_1\sum_k C_k\frac{d^2v_k(t)}{dt^2}. \qquad (8)$$

In the frequency domain, this relation transforms to

$$V_{in}(s)-V_1(s)=(sR_1+s^2L_1)\sum_k C_kV_k(s). \qquad (9)$$

Dividing (9) by $V_{in}(s)$, the following relation results,

$$1-T_1(s)=(sR_1+s^2L_1)\sum_k C_kT_k(s), \qquad (10)$$

where $T_1(s)$ is the transfer function at node 1 and $T_k(s)$ is the transfer function at node *k*. Note that determining the transfer function at node 1 is sufficient to determine the poles of the entire circuit since the transfer functions at all of the nodes of an *RLC* tree have a common denominator (as mentioned previously).

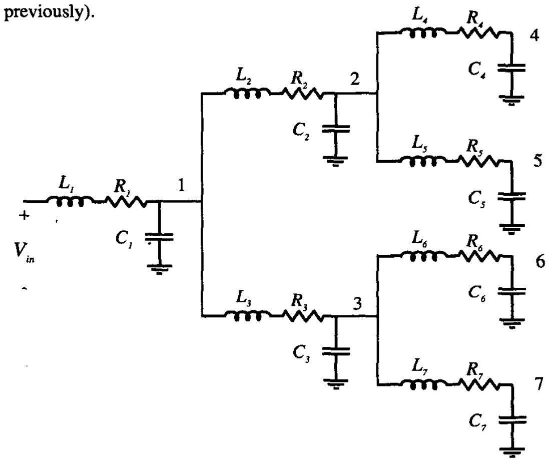


Fig. 5. General *RLC* tree.

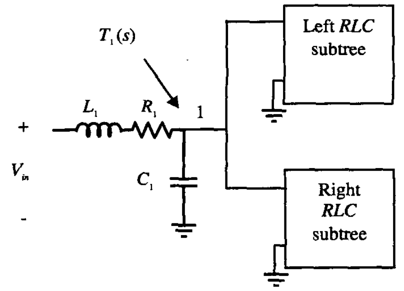


Fig. 6. Building block of a general *RLC* tree.

Now consider the structure shown in Fig. 6 which depicts an *RLC* section driving left and right subtrees. Without loss of generality, a binary branching factor is used here since a general tree with an arbitrary branching factor can be transformed into a binary tree by inserting zero impedance branches [35], [36]. The structure shown in Fig. 6 can be used recursively to completely represent any *RLC* tree since the left and right subtrees can in turn be represented by the same structure. The transfer function at node 1 of Fig. 6 is given by (10), which can be reformulated by using the rational representations of the transfer functions, $T_1(s)=N_1(s)/D(s)$ and $T_k(s)=N_k(s)/D(s)$, and is

$$D(s) - N_1(s) = (sR_1 + s^2L_1)\sum_k C_k N_k(s). \tag{11}$$

Assume that the transfer functions at all of the nodes of the left and right *RLC* subtrees (when the trees are disconnected) are known and are given by $T_{lk1}(s)=N_{lk1}(s)/D_l(s)$ at node k_1 of the left subtree and $T_{rk2}(s)=N_{rk2}(s)/D_r(s)$ at node k_2 of the right subtree. The numerator at node 1, $N_1(s)$ of Fig. 6, can be directly calculated by applying rule 2 described in the previous subsection and is

$$N_1(s) = D_l(s) \bullet D_r(s) \cdot \tag{12}$$

The "•" operator above represents a · polynomial multiplication. The denominator $D(s)$ can be determined from (11) as

$$D(s) = N_1(s) + (sR_1 + s^2L_1)M_1, \tag{13}$$

where M_1 is defined as

$$M_1 = \sum_k C_k N_k(s), \tag{14}$$

and characterizes the summation of the numerators of the transfer functions across the capacitors in the tree multiplied by the corresponding capacitances. The summation in M_1 operates over all of the capacitors in the tree and can be divided into three components,

$$M_1 = C_1N_1(s) + \sum_{k1} C_{k1}N_{k1}(s) + \sum_{k2} C_{k2}N_{k2}(s)' \tag{15}$$

where k_1 covers the capacitors in the left subtree and k_2 covers the capacitors in the right subtree. By applying rule 3, the numerators in the left subtree can be described in terms of the parameters of the disconnected left and right subtrees as $N_{k1}(s)=N_{lk1}(s)\bullet D_r(s)$. Similarly, $N_{k2}(s)=N_{rk2}(s)\bullet D_l(s)$. Thus, (15) can be reconfigured as

$$M_1 = C_1N_1(s) + \left(\sum_{k1} C_{k1}N_{lk1}(s)\right)\bullet D_r(s) + \left(\sum_{k2} C_{k2}N_{rk2}(s)\right)\bullet D_l(s). \tag{16}$$

Note that the two summations above are M_l and M_r of the disconnected left and right subtrees, respectively. Hence, M_l can be fully calculated in terms of the disconnected left and right subtree parameters as

$$M_1 = C_1N_1(s) + M_l(s) \bullet D_r(s) + M_r(s) \bullet D_l(s) \cdot \tag{17}$$

Thus, by knowing the parameters of the left and right subtrees, $M_l(s)$, $D_l(s)$, $M_r(s)$, and $D_r(s)$, (12), (17), and (13) can be used in that order to determine $N_1(s)$, $M_l(s)$, and $D(s)$, respectively. The parameters of the left and right subtrees, $M_l(s)$, $D_l(s)$, $M_r(s)$, and $D_r(s)$, can be determined in turn in terms of their left and right subtrees by using the structure shown in Fig. 6 and (12), (17), and (13). This process is repeated recursively until the left and right subtrees are non-existent. If the left subtree does not exist, then $M_l(s) = 0$ and $D_l(s) = 1$. If the right subtree does not exist, then $M_r(s) = 0$ and $D_r(s) = 1$.

After this recursion process terminates, the denominator and numerator across each capacitance C_k in the tree represent the transfer function for the subtree rooted at the *RLC* section *k*. For example, for the tree shown in Fig. 5, D(s) and *N*(s) at node 1 represent the transfer function at node 1 for the entire tree. However, D(s) and *N*(s) at node 2 represent the transfer function at node 2 for the subtree composed of the *RLC* sections, 2, 4, and 5. Also, *D*(s) and *N*(s) at node 4 represent the transfer function at node 4 for the subtree composed of *RLC* section 4. Thus, after the recursion process terminates, the only relevant parameters for the entire *RLC* tree are *D*(s) and *N*(s) across the capacitor closest to the input (C_1 in the case of the tree shown in Fig. 5). The denominators and numerators at all of the other nodes are incorrect. The denominators at these nodes need not be corrected since these denominators are the same as the denominator at the node closest to the input. However, the numerators differ at each node and need to be corrected. According to rule 3, all of the numerators in the left subtree have to be multiplied by $D_r(s)$ and all of the numerators in the right subtree have to be multiplied by $D_l(s)$. This process is repeated recursively starting at the root of the tree and advancing towards the sinks.

Thus, the process of determining the transfer function at all of the nodes of an *RLC* tree consists of two steps. The first step is to calculate the common denominator of the *RLC* tree using the recursive equations in (12), (17), and (13). The common denominator is the denominator at the node closest to the input of the *RLC* tree after the recursion terminates. The second step is to correct the numerators of the transfer functions at the nodes of the *RLC* tree.

C. Transfer Function Truncation and Approximation Order

The process of calculating the exact transfer functions at all of the nodes of an *RLC* tree has been described in the previous subsection. However, calculating the exact transfer function can be time consuming since *n* can be on the order of thousands for typical large industrial *RLC* trees. In practice, there is no need to calculate the thousands of poles characterizing an *RLC* tree since the transient behavior can be accurately characterized by a few number of low frequency dominant poles [15]-[21] (typically several tens of poles). Thus, a low frequency approximation is required that can correctly anticipate the set of dominant poles without calculating the exact high order transfer function. Assume that the exact transfer function at a specific node of the *RLC* tree is given by

$$T(s) = \frac{1 + a_1s + a_2s^2 + \ldots + a_ms^m}{1 + b_1s + b_2s^2 + \ldots + b_ns^n}, \tag{18}$$

where $b_1 - b_n$ and $a_1 - a_m$ are positive real constants. The system order *n* is equal to the total number of capacitors and inductors in the tree. The order of the numerator polynomial *m* is less than *n* and is dependent on the node at which the transfer function is calculated. A q^{th} order approximate transfer function is found by direct truncation of the exact transfer function *T*(*s*) in (18) and is given by

$$T_q(s) = \frac{1 + a_1s + a_2s^2 + \ldots + a_xs^x}{1 + b_1s + b_2s^2 + \ldots + b_qs^q}, \tag{19}$$

where $q < n$. The numerator order $x = m$ if $m \le q - 1$, otherwise $x = q - 1$. The order of the numerator has to be less than the order of the denominator for a causal approximation. If *s* (or the frequency) is sufficiently small, the terms with higher power of *s* in the denominator and numerator polynomials ($b_{q+1}s^{q+1} - b_ns^n$, $a_{x+1}s^{x+1} - a_ms^m$) are negligible with respect to the lower power terms in $T_q(s)$. Thus, for low frequencies, $T_q(s)$ is an accurate representation of $T(s)$. The range of frequencies for which $T_q(s)$ is accurate increases as *q* increases.

The calculation of a q^{th} order approximation for the transfer functions at all of the nodes of an *RLC* tree can be accomplished by an order limited polynomial multiplication. To better understand this concept, assume that *A* and *B* are two polynomials of orders n_a and n_b, respectively. The polynomial *C* given by *A*•*B* has an order of $n_c = n_a + n_b$. The polynomials *A*, *B*, and *C* are given by

$$A = \sum_{i=0}^{n_a} a_is^i,\ B = \sum_{i=0}^{n_b} b_is^i,\ \text{and}\ C = \sum_{i=0}^{n_c} c_is^i, \tag{20}$$

respectively, where the coefficients c_i are

$$c_i = \sum_{j=0}^{n_a} a_jb_{i-j}. \tag{21}$$

Note that b_{i-j} is equal to zero if $i - j$ is out of the range of 0 to n_b. For a *q* limited polynomial multiplication, the highest desired power of *s* in *C* is *q* rather than n_c and the coefficients of higher powers of *s* do not need to be calculated. Also, *A* and *B* can be limited by *q* since higher powers than s^q in both polynomials cannot produce powers of *s* in *C* less than or equal to *q*. Hence, if a q^{th} order approximation is sought, all of the polynomial multiplications of the DTT method described in the previous subsection are *q* limited. These *q* limited polynomial multiplications are much less expensive than full polynomial multiplications since *q* is typically much less than *n*.

Once the common denominator of order *q*, $D_q(s)$, is determined as described in the previous subsections, the first *q* dominant low frequency poles of the *RLC* tree can be calculated as the roots of the polynomial $D_q(s)$. A numerical method for evaluating the roots of a polynomial can be used to determine the *RLC* tree poles, $p_1 - p_q$, *e.g.*, [37], [38]. The residues corresponding to each pole at a specific node can be efficiently calculated by direct substitution of the poles into the numerator of the transfer function at this node and calculating the partial fraction equivalent of the reduced order transfer function.

The DTT method has a complexity linearly proportional to the order of the tree *n*, which is twice the number of *RLC* sections in the tree since each *RLC* section has one capacitor and one inductor. This linear complexity occurs because the DTT method traverses each section in the tree only once as illustrated in the previous section. At each section of the *RLC* tree, polynomial multiplications are required to calculate the common denominator as given by (12), (17), and (13). The number of scalar multiplications required for a *q* limited polynomial multiplication is at most *q*(*q*+1)/2 when the polynomial orders, n_a and n_b, are equal to *q*. However, the actual number of scalar multiplications performed by the DTT method is much less than the number of multiplications anticipated using the *q*(*q*+1)/2 complexity of a polynomial multiplication. The average number of scalar multiplications required per section is almost linear with *q* for an *RLC* tree. This lower complexity is because only two multiplications are required at each leaf of the tree independent of *q*. Note that half the nodes of a binary tree are leaves and that the percentage of leaves are higher with higher branching factors. Also, for a single line (with no branching), there are no polynomial multiplications in (12), (13), and (17). Thus, the DTT method has an almost linear complexity with the approximation order *q*. This linear complexity is in comparison to the complexity of AWE which is proportional to q^3. By analyzing the stability of the DTT approximations, it can be shown that a DTT approximation with an order less than five is guaranteed to be stable.

III. Experimental Results

The DTT method is applied in this section to calculate the transient response of several *RC* and *RLC* trees. The resulting transient responses are compared to SPICE simulations to evaluate the accuracy of the DTT method. The *RC* tree shown in Fig. 7 is simulated using the DTT method. The transient response at two nodes of the tree are calculated based on the DTT method and compared to SPICE in Fig. 8. A fourth order DTT approximation is used to calculate the transient responses shown in Fig. 8. Note that a fourth order approximation is accurate as compared to SPICE simulations. In general, a fourth order approximation is sufficiently accurate for most *RC* trees. The guaranteed stability of a fourth order approximation is therefore a valuable feature for *RC* circuits.

The circuit shown in Fig. 9 represents an *RLC* transmission line with a lumped source resistance and a load capacitance and is simulated using the DTT method. Several simulations of the circuit shown in Fig. 9 are shown in Fig. 10 with different line parameters and source and load impedances. Note that an approximation order between 20 and 35 is required for an underdamped response with second order oscillations to achieve SPICE-like accuracy. Such high order approximations cannot be achieved by AWE [16]-[20] due to the numerical instability of AWE with high approximation orders. Other methods capable of calculating such high order approximations [22]-[28] have a much higher computational complexity as compared to the DTT method. The computational efficiency of the DTT method and the numerical accuracy of DTT for very high orders of approximation makes DTT suitable for accurately simulating *RLC* trees.

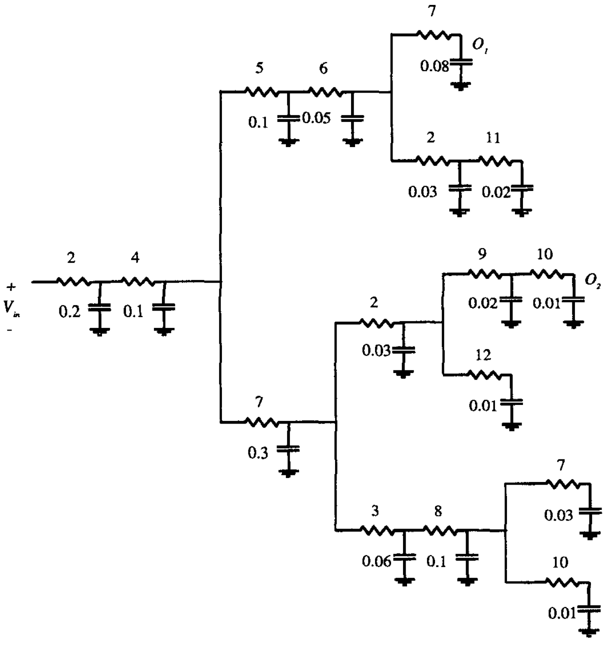
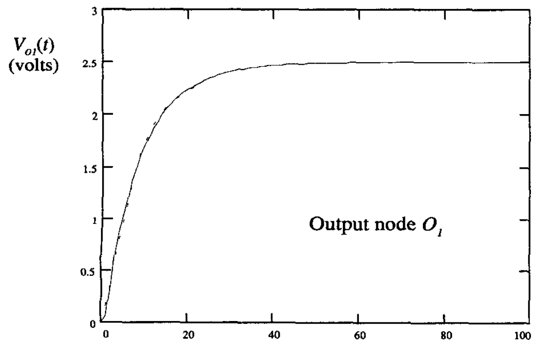


Fig. 7. A general *RC* tree. The resistance values shown are in ohms and capacitance values are in pF.



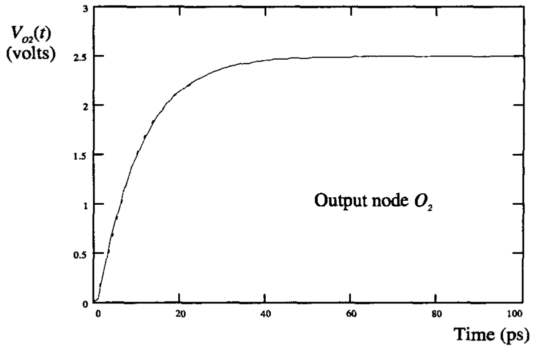


Fig. 8. Transient response evaluated using the DTT method as compared to SPICE simulations at different nodes of the *RC* tree depicted in Fig. 7. SPICE simulations are represented by a solid line and the DTT simulations are represented by a dashed line. A fourth order approximation is used.

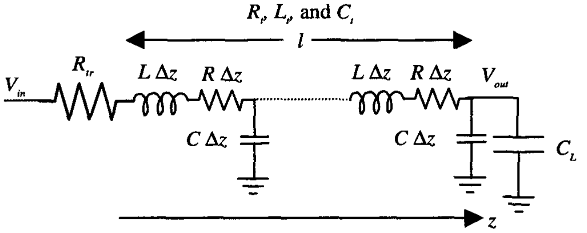


Fig. 9. An *RLC* transmission line with a source resistance and a load capacitance.

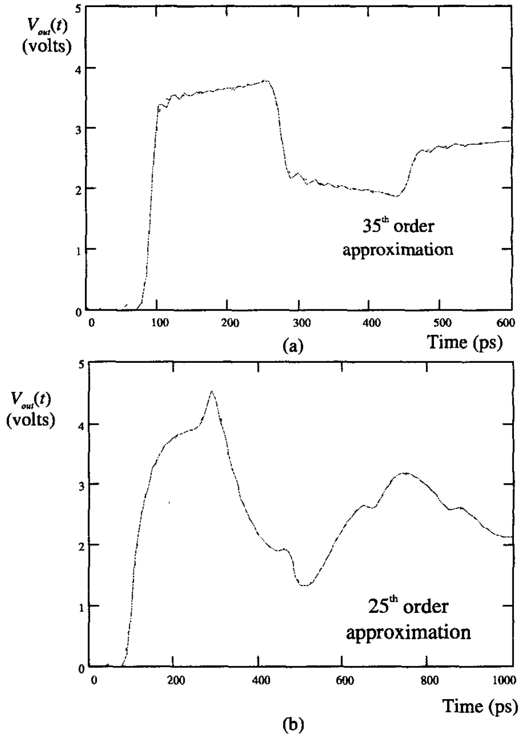


Fig. 10. Transient response evaluated using the DTT method as compared to SPICE simulations for the circuit shown in Fig. 9 using different line parameters. SPICE simulations are represented by a solid line and the DTT simulations are represented by a dashed line. R_t = 40 Ω, L_t = 7 nH, C_t = 1 pF, R_{tr} = 10 Ω, and C_L = 0.1 pF, and approximation order = 35. (b) R_t = 20 Ω, L_t = 8 nH, C_t = 1 pF, R_{tr} = 10 Ω, C_L = 0.4 pF, and approximation order = 25.

IV. Conclusions

The DTT method has been introduced to evaluate the transient responses within *RLC* trees with arbitrary accuracy for any input signal. The DTT method is numerically accurate for any order of approximation, permitting solutions to be determined with a large number of poles appropriate for approximating *RLC* trees with underdamped responses. The DTT method is computationally efficient with a complexity linearly proportional to the number of branches in the tree. A common set of poles is determined that characterizes the responses at all of the nodes of an *RLC* tree, further enhancing the computational efficiency of the proposed method. The stability is guaranteed by the DTT method for low order approximations with less than five poles, a useful characteristic when analyzing *RC* circuits.

References

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

[2] 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.

[3] 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.

[4] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Figures of Merit to Characterize the Importance of On-Chip Inductance," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 7, No. 4, pp. 442 – 449, December 1999.

[5] 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 and Techniques*, Vol. MTT-42, No. 8, pp. 1514 - 1523, August 1994.

[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] G. Y. Yacoub, H. Pham, and E. G. Friedman, "A System for Critical Path Analysis Based on Back Annotation and Distributed Interconnect Impedance Models," *Microelectronic Journal*, Vol. 18, No. 3, pp. 21 - 30, June 1988.

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

[9] C. F. Webb *et al.*, "A 400MHz S/390 Microprocessor," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 448 – 449, February 1997.

[10] 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.

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

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

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

[14] Y. I. Ismail, E. G. Friedman, and Jose L. Neves, "Equivalent Elmore Delay for *RLC* Trees," *IEEE Transactions on Computer Aided Design of Circuits and Systems*, Vol. TCAD-19, No. 1, pp. 83 – 97, January 2000.

[15] L. T. Pillage and R. A. Rohrer, "Delay Evaluation with Lumped Linear RLC Interconnect Circuit Models," *Proceedings of the Caltech Conference on VLSI*, pp. 143-158, May 1989.

[16] L. T. Pillage, R. A. Rohrer, and C. Visweswariah, *Electronic Circuit and System Simulation Methods*, McGraw-Hill, Inc., USA, 1994.

[17] L. T. Pillage and R. A. Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-9, No. 4, pp. 352 - 366, April 1990.

[18] D. F. Anastasakis, N. Gopal, S. Y. Kim, and L. T. Pillage, "On the Stability of Approximations in Asymptotic Waveform Evaluation," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 207 – 212, June 1992.

[19] 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.

[20] C. L. Ratzlaff, N. Gopal, and L. T. Pillage, "RICE: Rapid Interconnect Circuit Evaluator," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 555 – 560, June 1991.

[21] T. K. Tang and M. S. Nakhla, "Analysis of High-Speed VLSI Interconnects Using the Asymptotic Waveform Evaluation Techniques," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-11, No. 3, pp. 341 - 352, March 1992.

[22] P. Feldmann and R. W. Freund, "Efficient Linear Circuit Analysis by Pade Approximation via the Lancozos Process," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-14, No. 5, pp. 639 - 649, May 1995.

[23] P. Feldmann and R. W. Freund, "Reduced-Order Modeling of Large Linear Subcircuits via Block Lanczos Algorithm," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 474 – 479, June 1995.

[24] M. Silveira, M. Kamon, and J. White, "Efficient Reduced-Order Modeling of Frequency-Dependent Coupling Inductances Associated with 3-D Interconnect Structures," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 376 – 380, June 1995.

[25] D. L. Boley, "Krylov Space Methods on State-Space Control Models," *Journal of Circuits, Systems, and Signal Processing*, Vol. 13, No. 6, pp. 733 - 758, May 1994.

[26] A. Odabasioglu, M. Celik, and L. T. Pillage, "PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-17, No. 8, pp. 645 - 654, August 1998.

[27] A. Odabasioglu, M. Celik, and L. T. Pillage, "PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 58 – 65, November 1997.

[28] P. Feldmann and R. W. Freund, "Reduced-Order Modeling of Large Passive Linear Circuits by Means of the SyPVL Algorithm," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 280 – 287, November 1996.

[29] "Fast and Accurate Method for Simulating VLSI Interconnect," *Patent Application Pending.*

[30] B. C. Kuo, *Automatic Control Systems, A Design Perspective,* Prentice Hall of India, New Delhi 1989.

[31] B. J. Ley, S. G. Lutz, and C. F. Rehberg, *Linear Circuit Analysis*, McGraw-Hill, Inc., New York, 1959.

[32] C.A. Desoer and E. S. Kuh, *Basic Circuit Theory*, McGraw-Hill, Inc., New York, 1969.

[33] G. F. Paskusz and B. Bussell, *Linear Circuit Analysis*, Prentice Hall, Inc., New Jersey, 1963..

[34] R. E. Scott and W. Essigman, *Linear Circuits*, Addison-Wesley Publishing Co., Mass., 1960.

[35] L. P. P. P. van 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.

[36] C. J. Alpert, "Wire Segmenting for Improved Buffer Insertion," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 588-593, June 1997.

[37] P. Rabinowitz, *Numerical Methods for Non-Linear Algebraic Equations*, Gordon and Breach Science Publishers, London, England, 1970.

[38] M. Daehlen and A. Tveito, *Numerical Methods and Software Tools in Industrial Mathematics*, Birkhauser, Boston, 1997.

[39] S. Winograd, *Arithmetic Complexity of Computations,* J. W. Arrowsmith Ltd.., Bristol, England, 1980.

[40] R. C. Agarwal and J. W. Cooley, "New Algorithms for Digital Convolution," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 25, No. 5, pp. 392 - 410, May 1995.

[41] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms,* Addison-Wesley Publishing Co., Mass., 1976.