

Short Papers

Second-Order Approximations for *RLC* Trees

Ching-An Lin and Chien-Hsien Wu

Abstract—We propose two-pole one-zero second-order approximations for transfer functions in resistance–inductance–capacitance trees. The approximation matches the first three moments of the original transfer function. Formulas for computing step-response parameters such as delay time, rise time, overshoot, etc., are given. Simulation results show that adding the zero improves accuracy of the approximation.

Index Terms—Model reduction, reduced-order systems, resistance–inductance–capacitance (*RLC*) trees.

I. INTRODUCTION

Resistance–inductance–capacitance (*RLC*) trees are useful in modeling interconnect lines in very large scale integrated (VLSI) circuits [3]. Step-response parameters, such as delay time, at capacitor nodes in the tree are important for routing and wire-sizing optimization. Low-order approximations for the corresponding transfer functions are required for estimating the parameters without solving the complete *RLC* tree equations.

Ismail *et al.* [3] proposed a second-order approximation that matches the first two moments of the original transfer function. The second-order transfer function, with unit dc-gain, is completely characterized by the damping ratio ζ and undamped natural frequency ω_n . Estimates of various step-response parameters such as delay time, rise time, overshoot, etc., are proposed. In an effort to improve the accuracy of the second-order approximation, we propose a slightly more general two-pole one-zero second-order approximation. The three parameters of the transfer function are determined by matching the first three moments of the original transfer function. We give necessary and sufficient conditions under which the two-pole one-zero approximation is stable. Simulation results show that the additional degree of freedom in the second-order transfer function indeed improves the accuracy of the approximation, in term of frequency response and step response, and thus also improves the accuracy of estimates of step-response parameters.

The paper is organized as follows. We describe the *RLC* tree and give formulas for computing the moment matrices in Section II. The formula for damping ratio, undamped natural frequency, and zero location of the second-order approximation together with stability conditions are given in Section III. Explicit formulas for delay time, rise time, and overshoot are given in Section IV. Simulation examples and comparisons are given in Section V. Finally, brief conclusions are given in Section VI.

Manuscript received August 20, 2003; revised November 17, 2003. This work was supported by National Sciences Council under Grants NSC-90-2215-E009-055 and NSC-91-2215-E009-069. This paper was recommended by Associate Editor S. Sapatnekar.

The authors are with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: calin@cc.nctu.edu.tw).

Digital Object Identifier 10.1109/TCAD.2004.829794

II. MOMENTS OF TRANSFER MATRIX

For an *RLC* tree, the tree graph is uniquely defined and it consists of the independent voltage source branch and the series RL branches. The grounded capacitor branches are the links that define the fundamental loops [1]. KVL equations of these fundamental loops and KCL equations at the capacitor nodes completely specify the interconnection of the tree.

We call a series connection of an RL branch and a capacitor (link) branch a section. Consider an *RLC* tree of n sections. Let $v_s(t)$ be the voltage source input and $v(t)$ be the capacitor voltage vector. Since there are n sections, there are n fundamental loops and $v(t) \in \mathbf{R}^n$. Suppose that sections and loops are numbered from 1 to n . It can be shown that the (vector) differential equation relating the input voltage source $v_s(t)$ and the capacitor voltage vector $v(t)$ is [6]

$$\text{FRF}^T C \frac{dv_l}{dt} + \text{FLF}^T C \frac{d^2 v_l}{dt^2} + v_l = E v_s \quad (1)$$

where $E = [1 \dots 1]^T$, $R = \text{diag}(R_1, R_2, \dots, R_n)$, $L = \text{diag}(L_1, L_2, \dots, L_n)$, $C = \text{diag}(C_1, C_2, \dots, C_n)$, and the ik th element of $F \in \mathbf{R}^{n \times n}$, f_{ik} , is

$$f_{ik} = \begin{cases} 1, & \text{if the } k\text{th tree branch is in the } i\text{th loop} \\ 0, & \text{if the } k\text{th tree branch is not in the } i\text{th loop.} \end{cases}$$

The transfer matrix from $v_s(t)$ to the capacitor voltage vector $v(t)$ is

$$H(s) = \frac{V(s)}{V_s(s)} = (\text{FLF}^T C s^2 + \text{FRF}^T C s + I)^{-1} E \quad (2)$$

where $V(s) = \mathcal{L}(v(t))$ and $V_s(s) = \mathcal{L}(v_s(t))$. The moment matrices of transfer matrix $H(s)$ are

$$m_k = \frac{1}{k!} H^{(k)}(0) \quad k = 0, 1, 2, \dots$$

By computations [6], we have $m_0 = H(0) = E$ and the first three moment matrices

$$m_1 = -\text{FRF}^T C m_0 \quad (3)$$

$$m_2 = -\text{FRF}^T C m_1 - \text{FLF}^T C m_0 \quad (4)$$

$$m_3 = -\text{FRF}^T C m_2 - \text{FLF}^T C m_1. \quad (5)$$

We note that the recursive formula holds true for other high-order moments as well. The i th component of m_k is the k th moment of the transfer function to the i th capacitor node. We note that formulas for moments of transfer functions of *RLC* trees are available [3]. The above formulas are convenient for computing moment matrices. Note also that each component of the first moment matrix m_1 is negative.

III. SECOND-ORDER APPROXIMATIONS

We now consider matching the first three moments to obtain a second-order approximation. The three parameters of the second-order approximation we will determine are the damping ratio ζ , undamped natural frequency ω_n , and zero location $-z$. We consider scalar transfer function in this section, since an approximation of a transfer matrix is obtained component by component. Suppose the first three moments of an *RLC* tree transfer function m_1 – m_3 have been computed, say, using the formula in the previous section.

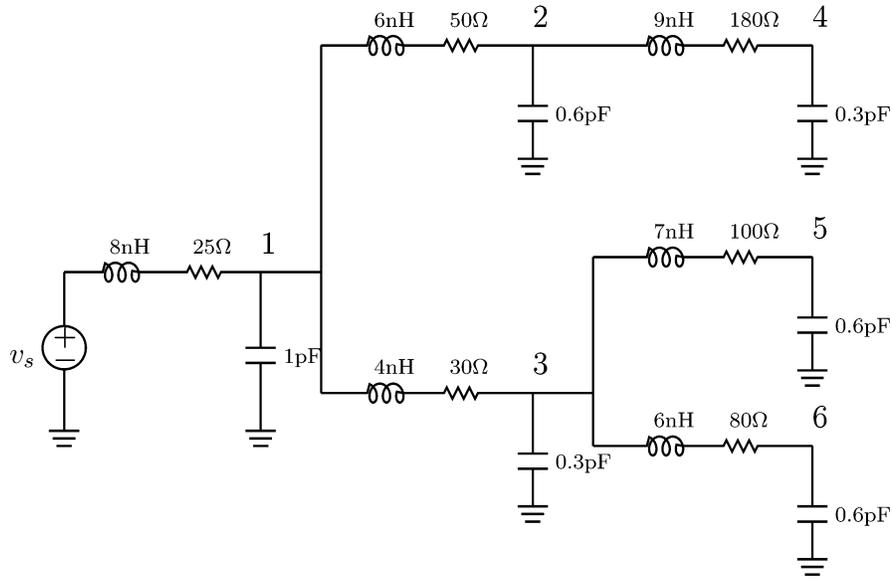


Fig. 1. Example of an RLC tree.

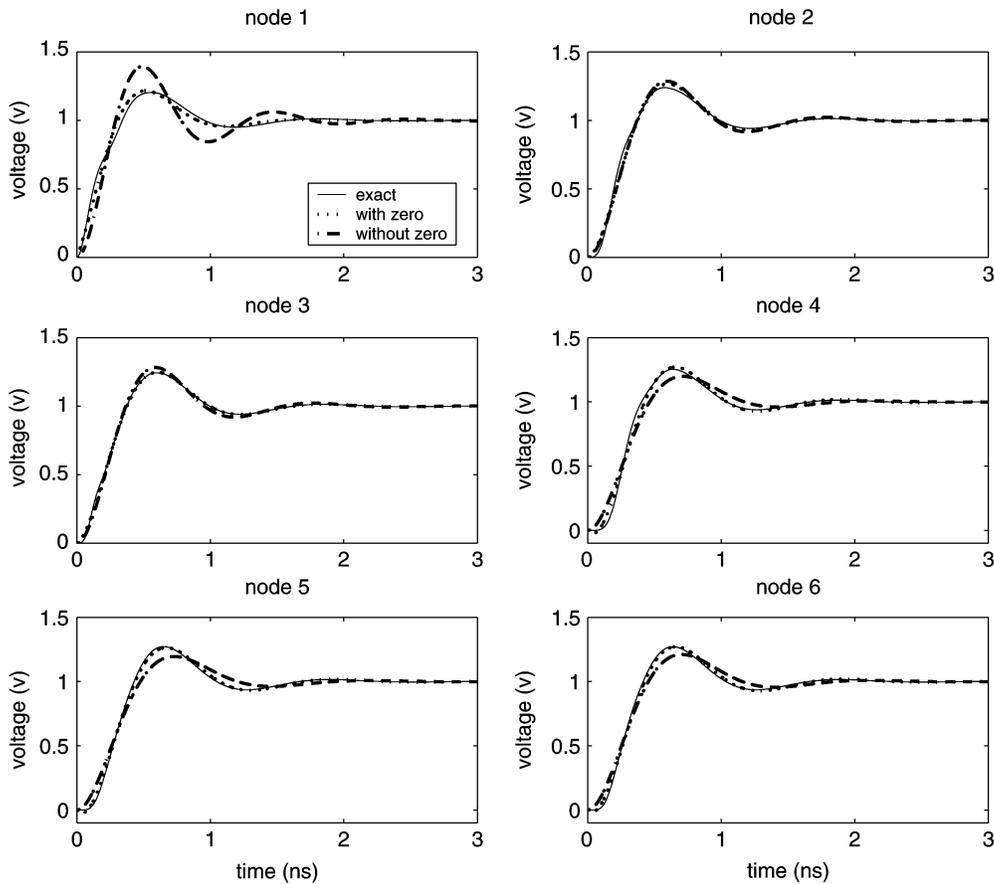


Fig. 2. Step response of each node.

The transfer function of the two-pole one-zero approximation, with unit dc-gain, has the form

$$H(s) = \frac{\omega_n^2(s+z)}{z(s^2+2\zeta\omega_n s + \omega_n^2)} = \frac{1 + \frac{1}{z}s}{1 + \frac{2\zeta}{\omega_n}s + \frac{1}{\omega_n^2}s^2}. \quad (6)$$

The power series expansion of $H(s)$ can be obtained by long division

$$H(s) = 1 + \frac{\omega_n - 2\zeta z}{z\omega_n} s + \frac{4\zeta^2 z - 2\zeta\omega_n - z}{z\omega_n^2} s^2 + \frac{-\omega_n + 4\zeta z + 4\zeta^2\omega_n - 8\zeta^3 z}{z\omega_n^3} s^3 + \dots$$

The moment-matching equations are

$$m_1 = \frac{\omega_n - 2\zeta z}{z\omega_n} \quad (7)$$

$$m_2 = \frac{4\zeta^2 z - 2\zeta\omega_n - z}{z\omega_n^2} \quad (8)$$

$$m_3 = \frac{-\omega_n + 4\zeta z + 4\zeta^2\omega_n - 8\zeta^3 z}{z\omega_n^3}. \quad (9)$$

Equations (7)–(9) uniquely determine ω_n , ζ , and z . The solutions are

$$\omega_n = \sqrt{\frac{m_1^2 - m_2}{m_2^2 - m_1 m_3}} \quad (10)$$

$$\zeta = -\frac{1}{2m_1\omega_n} (m_2\omega_n^2 + 1) \quad (11)$$

$$z = \frac{\omega_n}{m_1\omega_n + 2\zeta}. \quad (12)$$

Hence, given the moments m_1 – m_3 , (10)–(12) uniquely determine the undamped natural frequency ω_n , the damping ratio ζ , and zero location $-z$.

It is well known that a reduced-order transfer function obtained via Padé approximation may be unstable. It is possible that for some *RLC* trees, the second-order approximation obtained by matching the first three moments has poles with positive real parts. In such cases, the approximation is of course useless. In the following, we give conditions on the moments m_1 – m_3 under, which the second-order approximation, is stable. The second-order transfer function in (6) is stable if and only if $\zeta\omega_n > 0$. From (10) and (11) and the fact that $m_1 < 0$, it can be easily checked that the second-order approximation is stable if and only if:

- 1) $m_1^2 - m_2 \neq 0$, $m_2^2 - m_1 m_3 \neq 0$, $m_3 - m_1 m_2 \neq 0$;
- 2) the three values in 1) have the same sign.

IV. STEP-RESPONSE PARAMETERS

The unit-step response of the second-order transfer function $H(s)$ in (6) is

$$s(t) = 1 - \frac{e^{-\sigma t}}{\sqrt{1-\zeta^2}} [(\zeta - r) \sin \omega_d t + \sqrt{1-\zeta^2} \cos \omega_d t] \quad (13)$$

where $\sigma = \zeta\omega_n$, $r = \omega_n/z$, and $\omega_d = \omega_n \sqrt{1-\zeta^2}$. The delay time t_d is defined as the time it takes the response to rise to 50% of its final value. The rise time t_r is defined as the time it takes the response to rise from 10% to 90% of its final value. To simplify expressions, let $t' = \omega_n t$, and $g(t') = s(t)$. Hence, (13) becomes

$$g(t') = 1 - \frac{e^{-\zeta t'}}{\sqrt{1-\zeta^2}} [(\zeta - r) \sin(\sqrt{1-\zeta^2} t') + \sqrt{1-\zeta^2} \cos(\sqrt{1-\zeta^2} t')]. \quad (14)$$

We note that (14) has two variables, ζ and r , only and that explicit formulas for delay time and rise time are hard to find.

We use least squares curve fitting [5] to obtain approximation formulas for the delay time t_d and rise time t_r . We briefly describe curve fitting for t_d , that for t_r is similar. The normalized delay time $t'_d = t_d/\omega_n$ is first determined for a set of values of ζ and r via simulations.

For each r , we compute the parameters a and b so that

$$\frac{1}{t'_d(\zeta, r)} \approx a\zeta + b$$

in the least squares sense. The parameters a and b , which depend on r , are then least squares fitted by polynomials of degree 2. The result is

$$t'_d(r, \zeta) = \frac{1}{p_d(r)\zeta + q_d(r)} \quad (15)$$

TABLE I
DELAY TIME t_d IN NANoseconds

node	exact	formula (17)	error (%)	formula in [2]	error (%)
node 1	0.1235	0.1473	19.3	0.1715	38.9
node 2	0.1953	0.2112	8.1	0.2094	7.2
node 3	0.2021	0.2066	2.2	0.2080	2.9
node 4	0.2608	0.2658	1.9	0.2512	3.7
node 5	0.2720	0.2717	0.1	0.2572	5.4
node 6	0.2575	0.2624	1.9	0.2482	3.6

TABLE II
RISE TIME IN NANoseconds

node	exact	formula (18)	error (%)	formula in [2]	error (%)
node 1	0.2561	0.2296	10.3	0.2158	15.7
node 2	0.2134	0.2545	19.3	0.2759	29.3
node 3	0.2575	0.2601	1.0	0.2745	6.6
node 4	0.1943	0.2521	29.7	0.3526	81.5
node 5	0.2385	0.2590	8.6	0.3626	52.0
node 6	0.2295	0.2592	12.9	0.3449	50.3

where $p_d(r) = -0.0051r^2 - 0.5989r - 0.3652$ and $q_d(r) = 0.5355r^2 + 0.9136r + 0.9542$. The same curve-fitting method is used for the formula for t'_r . The result is

$$t'_r(r, \zeta) = \frac{1}{p_r(r)\zeta + q_r(r)} \quad (16)$$

where $p_r(r) = -0.3886r^2 - 0.1123r - 0.6959$ and $q_r(r) = 0.6064r^2 + 0.0762r + 0.9707$. The delay time t_d and rise time t_r are thus

$$t_d = \frac{t'_d}{\omega_n} \quad (17)$$

$$t_r = \frac{t'_r}{\omega_n}. \quad (18)$$

The peak time can be found by differentiating $s(t)$ in (13) and then set $s'(t) = 0$. The peak time at which the maximum overshoot occurs is

$$t_p = \frac{\pi - \theta}{\omega_d} \quad \text{where } \theta = \tan^{-1} \frac{r\sqrt{1-\zeta^2}}{1-r\zeta} \quad \text{and } \omega_d = \omega_n \sqrt{1-\zeta^2}. \quad (19)$$

The maximum overshoot M_o , obtained by substituting t_p into (13), is

$$M_o = \sqrt{1-2r\zeta+r^2} e^{-\frac{\zeta(\pi-\theta)}{\sqrt{1-\zeta^2}}}. \quad (20)$$

V. SIMULATION RESULTS

To examine the effectiveness of the proposed second-order approximation, we consider two *RLC* tree examples.

Example 1: Consider the *RLC* tree shown in Fig. 1, which has six sections and is considered in [3]. Step response of each capacitor node for the original transfer function, the two-pole one-zero approximation, and the two-pole approximation [3] are computed using Matlab. The results are shown in Fig. 2. We note that, for this example, the two-pole one-zero approximations give step responses very close to the original twelfth-order transfer function and the improvement provided by the additional zero can be clearly seen from the plots.

The delay time t_d and rise time t_r for step response of each node are shown in Tables I and II, respectively. Formula (17) gives an average error of 4.2% for delay time t_d , while the formula in [2] gives an average error of 7.8%. Formula (18) gives an average error of 12.8% for rise time t_r , while the formula in [2] gives an average error of 37.3%. For this example, the proposed two-pole one-zero approximation, on

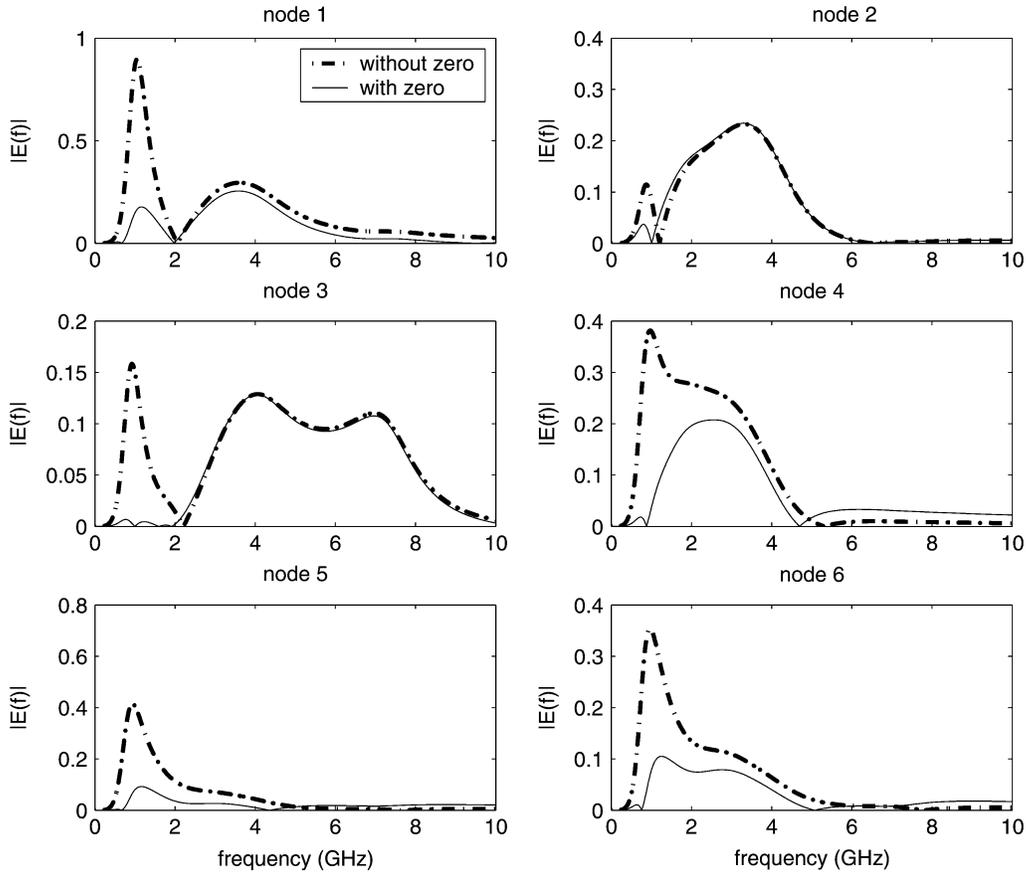


Fig. 3. Magnitude response of approximation error transfer function.

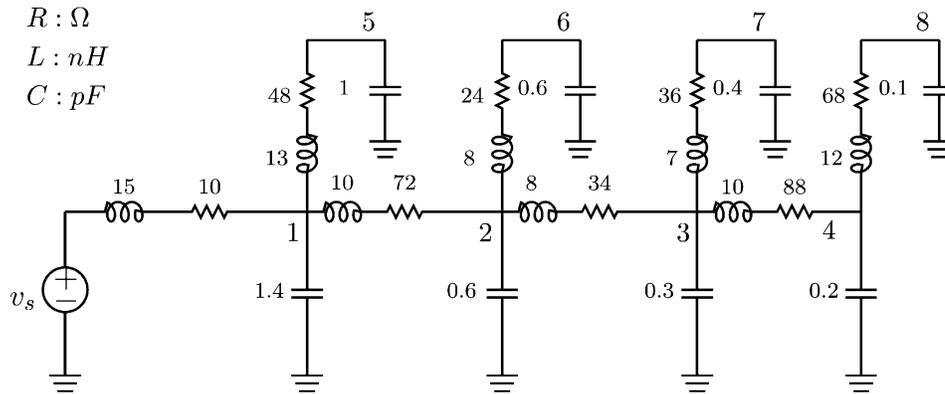


Fig. 4. RLC tree.

average, gives better results than the approximation in [2]. We note, however, that the approximation in [2] gives better estimate for delay time to node 2. Fig. 3 shows the frequency-magnitude response of the approximation-error transfer function at each node. It can be seen that adding the zero reduces the approximation error, especially at low frequencies. The plots are all in linear scale and are done using Matlab.

Example 2: Consider the RLC tree shown in Fig. 4, which has eight sections. The delay time and rise time for step response of each node are shown in Tables III and IV, respectively. Formula (17) gives an average error of 4.4% for delay time t_d , while the formula in [2] gives an average error of 12.0%. Formula (18) gives an average error of 12.3% for rise time t_r , while the formula in [2] gives an average error of 43.0%. For this example, the proposed two-pole one-zero approximation, on average, also gives better results than the approximation in [2]. We note

TABLE III
DELAY TIME t_d IN NANoseconds

node	exact	formula (17)	error (%)	formula in [2]	error (%)
node 1	0.1802	0.2140	18.7	0.2709	50.0
node 2	0.3847	0.3747	2.6	0.3519	8.5
node 3	0.4337	0.4201	3.1	0.3844	11.4
node 4	0.4642	0.4453	4.1	0.4060	12.5
node 5	0.3007	0.2934	2.4	0.3083	2.5
node 6	0.3826	0.3995	4.4	0.3684	3.7
node 7	0.4507	0.4372	3.0	0.3980	11.7
node 8	0.4737	0.4523	4.5	0.4123	13.0

that approximation in [2] gives better estimate for delay time at node 6 and rise time at node 1, although the differences are small. The step

TABLE IV
RISE TIME t_r IN NANoseconds

node	exact	formula (18)	error (%)	formula in [2]	error (%)
node 1	0.3141	0.3035	3.4	0.3118	0.7
node 2	0.4005	0.3764	6.0	0.4557	13.8
node 3	0.3583	0.3735	4.2	0.5063	41.3
node 4	0.3312	0.3671	10.8	0.5432	64.0
node 5	0.2197	0.3491	58.9	0.3710	68.9
node 6	0.3568	0.3739	4.8	0.4796	34.4
node 7	0.3266	0.3697	13.2	0.5280	61.7
node 8	0.3173	0.3650	15.0	0.5536	74.5

responses of the approximations and frequency magnitude responses for the approximation error transfer functions show similar results as in Example 1 and thus are not shown again.

VI. CONCLUSION

We propose a method to obtain second-order approximations for transfer functions in *RLC* trees. Examples show that the two-pole one-zero approximations give improved accuracy over the existing second-order approximations in terms of step response, frequency response, estimated delay time, and rise time. The results can be used to quickly estimate signal delay and other parameters in *RLC* trees.

ACKNOWLEDGMENT

The authors thank the reviewers, whose comments improved the paper.

REFERENCES

- [1] L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits*. New York: McGraw Hill, 1987.
- [2] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Equivalent elmore delay for *RLC* trees," in *Proc. IEEE/ACM Design Automation Conf.*, June 1999, pp. 715–720.
- [3] ———, "Equivalent elmore delay for *RLC* trees," *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 83–97, Jan. 2000.
- [4] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352–366, Apr. 1990.
- [5] G. Lindfield and J. Penny, *Numerical Methods Using MATLAB*. Englewood Cliffs, NJ: Prentice Hill, 2000.
- [6] C. H. Wu, "Model order reduction for *RLC* trees," M.S. thesis, Dept. Elect. Contr. Eng., National Chiao-Tung Univ., June 2003.

Indirect Test Architecture for SoC Testing

Mohsen Nahvi and André Ivanov

Abstract—A generic model for test architectures in the core-based system-on-chip (SoC) designs consists of source/sink, wrapper, and test access mechanism (TAM). Current test architectures for digital cores assume a direct connection between the core and the tester. In these architectures, the tester establishes a physical link between itself and the core, such that it can directly control the core's design-for-testability (DFT), such as the scan chains or primary inputs. This direct connection undermines the modularity in the generic test architecture by tightly coupling its elements. In this paper, we propose a network-oriented indirect and modular architecture (NIMA) for postfabrication test in an SoC design methodology. In NIMA, test stimuli and expected results for digital cores are first compiled into new formats and subsequently encapsulated into packets. These packets are augmented with control and address bits such that they can autonomously be transmitted to their destination through a switching fabric. Owing to the indirect nature of the connection, embedded autonomous blocks at each core are used to apply the test to the core and compare the test results with expected values. This indirect access to the core decouples test data processing at the core from its communication providing the basis for flexible and modular test design and programming. Moreover, NIMA facilitates remote-access of single or multiple testers to an SoC, and enables the sending of test data to an SoC in-field in order to test the chip in its target system. Finally, NIMA serves in contributing toward the development of new test architectures that benefit from network-centric SoCs. We present a first implementation of NIMA when applied to a number of SoC benchmarks.

Index Terms—Core-based testing, design-for-testability (DFT), networks-on-chip (NoC), system-on-chip (SoC).

I. INTRODUCTION

The productivity gap is one that exists between the productivity of chip designers and the available resources on today's complex chips [1]. An effective way to overcome the productivity gap is to reuse previously designed/verified functional blocks, or semiconductor intellectual properties (IPs), as embedded cores in a system-on-chip (SoC) design [2]. The integration of IP blocks into a design results in increased complexity of the design-for-testability (DFT) aspects and manufacturing test [3]. Testing core-based SoCs presents major challenges especially in regards to the limited accessibility of the embedded cores and the generation of a system DFT [2]. To reduce test development time and, hence, keep up with shorter time-to-market and time-to-volume pressures, a similar productivity improvement technique, i.e., reuse of the DFT and test program for each core, needs to be applied [3]. In the modular model of SoC testing, the core-user treats individual core test programs as distinct components and integrates/schedules these components into a system test program with limited knowledge of the core's internal detail [2].

To address the core-based SoC testing challenges, a more structured and systematic approach than the traditional DFT is required. Zorian *et al.* [2] proposed a generic test architecture consisting of source/sink, wrapper, and test access mechanism (TAM). In this model, the source

Manuscript received September 5, 2002; revised August 2, 2003 and October 28, 2003. This work was supported in part by the Canadian Microelectronics Corporation (CMC), in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), in part by Micronet Research and Development, in part by PMC-Sierra, and in part by the Gennum Corporation. This paper was recommended by Associate Editor K. Chakrabarty.

The authors are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: mohsenn@ece.ubc.ca; ivanov@ece.ubc.ca).

Digital Object Identifier 10.1109/TCAD.2004.829796