

Symbolic Techniques for Statistical Timing Analysis of RCL Mesh Networks with Resistor Loops *

Zhigang Hao and Guoyong Shi

School of Microelectronics, Shanghai Jiao Tong University
Shanghai, China

{haozhigang, shiguoyong}@ic.sjtu.edu.cn

Abstract—A symbolic moment calculator for recursive moment computation of RCL interconnect networks involving resistor loops is proposed. Using the tearing technique, the network can be partitioned into a spanning tree and a set of resistor links. Special data structures for symbolic moment analysis are proposed. Applications of this structural computation methodology to symbolic reduced order modeling and statistical timing analysis of mesh networks are investigated.

I. INTRODUCTION

As feature size of VLSI processing technology is continuously scaling down below 65nm, process variations, power supply noise and thermal variations etc. are becoming increasingly significant. Recently, clock mesh [1] and clock tree with crosslinks [2] have been used by designers to alleviate the clock skew variations in deep submicron design.

Due to their scale and complexity, direct analysis of industry-scale mesh networks creates huge challenges to the existing circuit simulators. The existing techniques that work well for large-scale tree networks cannot directly be applied to mesh networks. Many techniques have been developed recently for dealing with analysis and synthesis of mesh networks, such as sliding window [3], model order reduction [4] and moment computation [5], however, none of them are suitable for application under strong circuit parameter variations. In the contemporary design practice, there exist emerging needs on the statistical analysis of timing skews of clock networks (clock mesh) for the purpose of design verification and synthesis. So far, there hardly exists any research work that fully addresses the statistical timing issues of clock mesh networks.

The authors in [6] proposed a symbolic moment calculator for capacitively and inductively coupled interconnect trees and reported its application to statistical analysis of signal integrity effects. This paper extends the previous work [6] to handling interconnects with resistor loops, which are common in RLCC networks or clock mesh. With an underlying structural representation of the moment computation process, design metrics such as timing and crosstalk etc. can be computed symbolically, which renders the sensitivity computation a relatively easy task.

The remainder of the paper is organized as follows. The background of symbolic moment calculation and its extension to handling resistor loops are presented in Section II. Statistical

timing analysis using sensitivity is presented in Section III. Experimental results are reported in Section IV. A conclusion is drawn in Section V.

II. SYMBOLIC MOMENT COMPUTATION FOR MESH NETWORKS

It is well-known that the moments for an RCL interconnect tree can be computed recursively [7], [8]. Let $m_{i,k}$ be the k th order moment at node i . It is well established that any orders of moments at any tree node (called nodal moments) can be computed recursively by the following formula:

$$m_{i,k} = \sum_{R_\ell \in P_i} R_\ell \sum_{j \in T_\ell} C_j \cdot m_{j,k-1} - \sum_{L_\ell \in P_i} L_\ell \sum_{j \in T_\ell} C_j \cdot m_{j,k-2}, \quad (1)$$

where P_i represents the path from the full-tree root to node i , the summation index $R_\ell \in P_i$ means summing over all the resistors R_ℓ on the path P_i , T_ℓ denotes the subtree rooted at node ℓ , and the summation index $j \in T_\ell$ indicates the summation over all nodes belonging to the subtree T_ℓ (including the root of T_ℓ).

For the purpose of numerical computation, equation (1) is considered efficient and has been used in a number of works [5], [7], [8], [9]. However, in certain applications involving statistical analysis, repeated evaluation of this formula is required. Without a good data structure, directly-coded numerical evaluation program (to be invoked repeatedly) would not be efficient. On the other hand, a particular data structure would largely facilitate the computation of moment sensitivity.

The work [6] introduced a dedicated structural computation scheme for the purpose mentioned above; mainly, a novel tree-form binary decision diagram (tree-BDD) is constructed for the recursive computation of moments and moment sensitivities. The key mechanism is to map the arithmetic operations (*multiply* and *add*) involved in (1) to the decision arrows in the tree-BDD.

The seemingly simple tree-BDD data structure offers us the following advantages: 1) The structure follows the circuit topology directly; hence, can be constructed easily given a circuit netlist; 2) The sensitivity computation can be implemented symbolically without altering the data structure; 3) It is easy to incorporate capacitive/inductive couplings for crosstalk analysis; and 4) Extension to dealing with resistive links is also possible, by merging the tree-BDD representation with another

*This research was supported in part by Shanghai Pu Jiang Research Foundation (Grant No. 07pj14053) and the Initiative Research Fund for overseas returnees from the Ministry of Education of China (2008), and by the National Natural Science Foundation of China, Grant No. 60876089.

binary decision diagram used for branch tearing; which is the subject of this work.

A. Extension to Resistor Loops

Rohrer [10] reinterpreted Kron’s branch tearing method in a systematic form which turns out to be a powerful method for partitioning large-scale circuits into tractable forms. A non-tree network involving resistor loops can be partitioned into a spanning tree plus a set of resistor links. Kron’s method has been used for moment computation considering resistor links in [5], [7], [9], but all for numerical computations. For multiple resistor links, the work [7] used matrix formulation by tearing resistor links all together then solved by LU factorization while the work of [5], [9] introduced a recursive computation scheme by tearing the resistors link-wise and stored the computation steps in a BDD. The key discovery of the work [5] is that the computation complexity is reduced to polynomial rather than exponential due to the nice *sharing* property inherent in the branch tearing process. However, the authors of [5], [9] only used this mechanism for numerical moment computation.

We adapt the BDD-based branch tearing scheme proposed in [5] and combine the BDD-based link processing with our tree-BDD based symbolic moment calculator [6]. It turns out that the symbolic moments of non-tree networks can be computed by integrating the two BDD-based moment computation schemes, which forms the key contribution of this paper.

According to Kron’s method, the current that would flow through the resistor link R is

$$I_R = \frac{V_{oc}}{R_{link} + R_{th}} \quad (2)$$

where V_{oc} is the open circuit voltage across the resistor terminals (with the resistor link removed), R_{link} is the value of the resistor link, and R_{th} is the Thevenin equivalent resistor seen by R_{link} . R_{th} can be calculated by: 1) placing a pair of unit current sources at the two terminal nodes p and q of the resistor link with $I_p = -1A$ and $I_q = 1A$ (both connected to the ground), 2) opening all the grounding capacitors, and 3) shorting all voltage sources (see [10] for the details). According to our moment calculator for tree circuits, placing a pair of currents at the relevant nodes can simply be realized by modifying the relevant capacitor values in the C-tree of the tree-BDD. Then R_{th} is obtained as

$$R_{th} = V_{th}/1A = (m_{p,0} - m_{q,0})/1A = m_{p,0} - m_{q,0}, \quad (3)$$

where $m_{p,0}$ and $m_{q,0}$ are the 0th order moments at nodes p and q . An example circuit containing three resistor links is shown in Fig. 1(a) and the Thevenin equivalent circuit for computing R_{th} of the resistor link R_4 is shown in Fig. 1(b).

After the computation of I_R in (2), the voltage of the original circuit (with the link resistor placed back) is computed by [10]

$$V = V^{(O)} - I_R \cdot V^{(A)} \quad (4)$$

where the superscript ‘O’ stands for the *open* circuit with the link resistor removed, while the superscript ‘A’ stands for the Thevenin equivalent circuit with the link replaced by the 1A

and $-1A$ current sources. Similarly, the k th order moment of node i can be computed by [5]

$$m_{i,k} = m_{i,k}^{(O)} - I_R \cdot m_{i,k}^{(A)}, \quad (5)$$

where $m_{i,k}^{(O)}$ and $m_{i,k}^{(A)}$ bear the similar meanings as $V^{(O)}$ and $V^{(A)}$. Following the method in [5], multiple resistor links

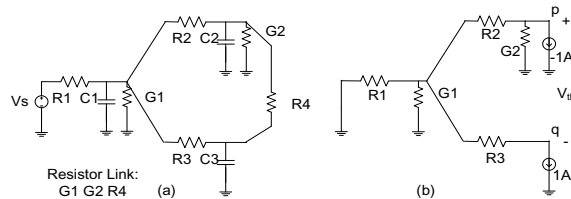


Fig. 1. (a) A circuit with three resistor links. (b) The Thevenin equivalent circuit for computing R_{th} for the link R_4 .

are processed successively by using a BDD data structure, referred to as the link-BDD. The computation process for the example circuit is illustrated in Fig. 2, which is represented in the form different from the way it is presented in [5] for better understanding. Each vertex in the link-BDD represents one of the resistor links for which two operations are taken; the ‘1A’ edge (solid arrow) indicates that the corresponding resistor link in the circuit is replaced by a pair of 1A current sources mentioned earlier for computing the Thevenin equivalent impedance while the ‘open’ edge (dashed arrow) simply indicates that the corresponding resistor link is removed. The triples attached to each link-BDD vertex in Fig. 2 indicates the circuit currently under processing. For example, reading from the link-BDD root downward, (G_1, G_2, R_4) refers to the original circuit; (G_1, G_2, A) refers to the circuit where the link R_4 has been replaced by the 1A current source pair; (G_1, G_2, O) refers to the circuit with the link R_4 removed; and so on. Remember that, whenever computing the Thevenin equivalent impedance for one specific link, only the pair of 1A current sources corresponding to the link is applied while all other sources must be turned off. That is why only one A appears in all triples involving an ‘A’.

We note that all circuits at the leaf vertices of the link-BDD are tree circuits, whose moments are computed by the same tree-BDD. After the moments at the link-BDD leaf vertices are computed, the moments at the other link-BDD vertices are computed bottom-up. That is, the moments at a parent vertex are computed by using the moments at the two offspring vertices pointed by the 1A arrow and the open arrow using a formula similar to (5). The next algorithm summarizes the computation process described so far.

Multiple-link Moment Computation Algorithm

- Step 1. Construct a link-BDD for the set of links. Meanwhile construct a tree-BDD for the spanning tree circuit.
- Step 2. Evaluate the tree circuits at the leaves of link-BDD using the tree-BDD.
- Step 3. Evaluate the link-BDD from bottom-up.
- Step 4. Get the 0th order nodal moments from the root of the link-BDD.

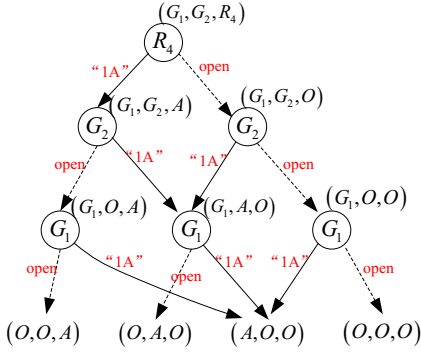


Fig. 2. A link-BDD for moment calculation.

Step 5. Replace the C-values in the tree-BDD by the nodal moments computed in Step 4.

Step 6. Repeat Steps 2 to 4 to get the nodal moments of the next order.

The computation complexity for the nodal moments of one order is estimated as follows. Suppose a network has N nodes and L links. Note that one round of tree-BDD computation is linear in the number of nodes, i.e., $O(N)$. We have $(L + 1)$ leaf nodes in the link-BDD; implying the computation cost of $O((L + 1)N)$. In the bottom-up computation of moments for the link-BDD vertices, a nodal voltage vector of length N is updated by using the formula (5) whose cost is $O(N)$. The link-BDD has a total of $O(L^2)$ vertices; thus the computation cost for one round of link-BDD is of the order of $O(N \cdot L^2)$. If the number of links L is much smaller than the number of nodes N , i.e., $L \ll N$, the computation scheme is of good efficiency. However, the computation cost increases if the number of links is large. For a typical mesh circuit, the number of links L could reach about $N/2$, which would result in a computation complexity of $O(N^3)$. Even for such cases, the BDD-based computation is still worthwhile by using the proposed symbolic computation mechanism because directly inverting a matrix symbolically for a mesh circuit is of exponential complexity in general.

III. STATISTICAL TIMING ANALYSIS

As soon as the moments are obtained via symbolic moment calculator, they can be used in at least two ways: one is to estimate the delay and slew using the moment metrics such as Elmore, D2M [11] and S2M [12] etc.; the other way is to use the computed nodal moments for model order reduction [13], [14] if accurate waveforms are needed.

For tree structured circuits (even capacitively/inductively coupled), an efficient model construction procedure by tree-traversal exists which avoids direct matrix-matrix multiplications [6]. Interestingly, this technique remains valid for non-tree circuits involving resistor links. Due to space limitation, the technical details are omitted for brevity.

A probability density function (pdf) is commonly used for certain statistical circuit quantities with variations [15]. For single parameter variation, the Direct Mapping Method together with the sensitivity analysis (proposed in [6]) is a very efficient method for pdf profiling.

Let X be a real random variable with the probability density function $f_X(x)$. Let $y = g(x)$ be a first-order differentiable function of x . In Probability Theory, the pdf of the random variable Y satisfies

$$f_Y(y) = \sum_k \frac{f_X(x_k)}{|dg(x_k)/dx|}, \quad (6)$$

where x_k , $k = 1, 2, \dots$, are all points satisfying $g(x_k) = y$. The derivative (or sensitivity) in (6) can be computed symbolically using the BDD-based moment calculator or computed numerically using reduced order models.

In principle, the Direct Mapping Method also works for multiple parameters, however, the number of samples would increase exponentially as the number of parameters increases. In the case of a two-variate function $D(W, H)$, it can be linearly approximated by [15]

$$D(W, H) = D_{norm} + f_W(\Delta W) + g_H(\Delta H), \quad (7)$$

where $D_{norm} := D(W_{norm}, H_{norm})$. Then the pdf of $D(W, H)$ can be computed by the convolution

$$pdf(D - D_{norm}) = pdf(f_W(\Delta W)) * pdf(g_H(\Delta H)), \quad (8)$$

where $f_W(\Delta W)$ and $g_H(\Delta H)$ are the sensitivity functions which are computed by the Direct Mapping Algorithm. However, for the number of parameters exceeding three, the computation complexity still increases rapidly. Therefore, pdf profiling for multiple parameters remains an issue.

IV. EXPERIMENTAL RESULTS

In this section, we use six test circuits to demonstrate the application of our proposed approach to statistical interconnect analysis. Designs 1, 2, 3, and 4 listed in Table I are mesh-structured circuits and designs 5 and 6 are tree-structured circuits. We implemented our statistical interconnect simulator (SIS) in C++ and tested it on an Intel 2.83G CPU with 4GB memory running a Redhat Enterprise Linux 4 operating system. The performance was compared to HSPICE (version 2005.09) running on the same machine. The per-unit-length resistor and capacitor values were set to 0.1Ω and $4.11E-16F$, respectively, according to a $0.18\mu m$ process technology. The input source was a $1V$ step input for the designs.

In the first part of experiment, we would like to examine the performance of our proposed moment computation algorithm. The second and third columns in Table I indicate the number of nodes and number of resistor links in the design while the fourth and the fifth columns in the table show the time for constructing the proposed tree-BDD and link-BDD structure, respectively. The time for evaluating the moments for all nodes in the designs up to 4th order is shown in the sixth column. The last column shows the time for generating the reduced 4th order models via traversing the spanning tree of the designs. As can be seen, the time for evaluating the moments for trees is much less than non-trees with resistor links for which most of the time is spent on constructing the link-BDD structure.

On the other hand, the delays measured by simulating reduced order models were also very accurate compared with

TABLE I
PERFORMANCE COMPARISON.

Design	Num of Nodes	Num of R Links	Build tree-BDD (s)	Build link-BDD (s)	Moment Evaluation (s)	Reduced Model (s)
1	2101	100	0.06	1.35	0.16	0.02
2	4301	100	0.13	2.81	0.32	0.04
3	4576	225	0.15	14.41	0.72	0.04
4	9376	225	0.34	30.54	1.49	0.09
5	5000	0	0.12	0	0.05	0.05
6	10000	0	0.24	0	0.11	0.09

HSPICE. We chose the reduced model to be of order 4 for all design cases and used 50% delay of step response for the timing measurement. In general, order 4 is sufficient for most RC cases without considering inductors.

In the second part of the experiment, we analyzed the accuracy and efficiency of our SIS simulator in statistical timing analysis. The per-unit-length resistor and capacitor values were assumed to be Gaussian variables with 30% variation corresponding to 3σ . For designs 2 and 4, two nodes, one with the maximum delay time and the other with the minimum delay time, were selected for 50% delay measurement under variation. Clock skew (defined to be the delay difference of two selected nodes) was also measured. With the Direct Mapping method, only 40 samples were sufficient for timing pdf profiling while the number of samples used for HSPICE Monte Carlo simulation was 5000 for a comparable accuracy.

TABLE II
STATISTICAL MEASUREMENTS OF NODE DELAY AND SKEW

Design	Node	Error		Speed up to HSPICE
		mean	std	
2	node 1	0.01%	2.11%	451
	node 2	0.15%	2.53%	
	skew(1,2)	0.02%	3.35%	
4	node 1	0.02%	2.24%	233
	node 2	0.09%	2.20%	
	skew(1,2)	0.21%	2.70%	

Table II tells us that besides achieving good accuracy, the SIS excels with significant speed up for statistical timing analysis. Shown in Fig. 3 is one simulated pdf result selected for Design 2.

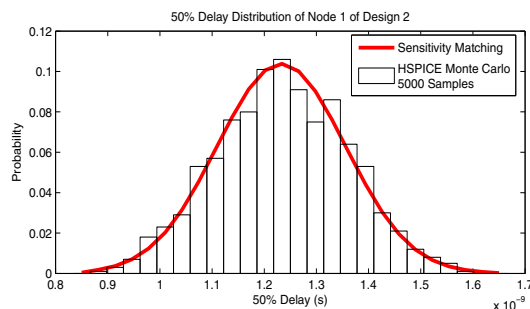


Fig. 3. 50% delay distribution of node 1 for Design 2.

V. CONCLUSION

This paper has proposed a new symbolic moment analysis technique that integrates two recently proposed moment computation schemes, both using BDDs. The technique can be used for analysis of non-tree and mesh networks. Applications to statistical mesh timing analysis in the form of pdf profiling

are studied. Although analyzing fully linked large-scale mesh circuits remains a challenge ahead, the proposed symbolic computation techniques are believed to be promising in many potential applications. The presented technique can further be extended to mesh circuit with multiple sources, which will be the subject of another paper. With the possibility of rapid pdf profiling, physical design automation tools incorporating statistical metrics is expected to be possible in the near future.

REFERENCES

- [1] P. Restle et al, "A clock distribution network for microprocessors," *IEEE J. Solid-State Circuits*, vol. 36, no. 5, pp. 792–799, May 2001.
- [2] A. Rajaram, J. Hu, and R. Mahapatra, "Reducing clock skew variability via crosslinks," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1176–1182, June 2006.
- [3] H. Chen, C. Yeh, G. Wilke, S. Reddy, H. Nugyen, W. Walker, and R. Murgai, "A sliding window scheme for accurate clock mesh analysis," in *Proc. Int. Conf. Computer-Aided Design*, pp. 939–946, Nov. 2005.
- [4] X. Ye, P. Li, M. Zhao, R. Panda, and J. Hu, "Analysis of large clock meshes via harmonic-weighted model order reduction and port sliding," in *Proc. Conf. Computer-Aided Design*, pp. 627–631, Nov. 2007.
- [5] H. J. Lee, M. H. Lai, C. C. Chu, and W. S. Feng, "Moment computations for R(L)C interconnects with multiple resistor loops using ROBDD techniques," in *Proc. Asia Pacific Circuits Syst. Conf.*, vol. 1, pp. 525–528, Dec. 2004.
- [6] Z. Hao and G. Shi, "Sensitivity approach to statistical signal integrity analysis of coupled interconnect trees," in *Proc. IEEE Midwest Symp. Circuits Syst.*, pp. 212–215, Aug. 2009.
- [7] C. Ratzlaff and L. Pillage, "RICE: Rapid interconnect circuit evaluation using AWE," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 6, pp. 763–776, June 1994.
- [8] Q. Yu and E. S. Kuh, "Exact moment matching model of transmission lines and application to interconnect delay estimation," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 3, no. 2, pp. 195–206, April 2002.
- [9] H. J. Lee, M. H. Lai, C. C. Chu, and W. S. Feng, "Applications of tree/link partitioning for moment computations of general lumped RLC networks with resistor loops," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1, pp. 713–716, May. 2004.
- [10] R. Rohrer, "Circuit partitioning simplified," *IEEE Trans. Circuits Syst.*, vol. 35, no. 1, pp. 2–5, Jan 1988.
- [11] C. J. Alpert, A. Devgan, and C. Kashyap, "A two moment RC delay metric for performance optimization," in *Proc. International Symposium on Physical Design (ISPD)*, pp. 73–78, San Diego, CA, 2000.
- [12] K. Agarwal, D. Sylvester, and D. Blaauw, "A simple metric for slew rate of RC circuits based on two circuit moments," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 9, pp. 1346–1354, Sept. 2004.
- [13] L. Pillage and R. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, pp. 352–366, April 1990.
- [14] G. Shi, B. Hu, and C. J. R. Shi, "On symbolic model order reduction," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 7, pp. 1257–1272, July, 2006.
- [15] E. Matoglu, N. Pham, D. N. de Araujo, M. Cases and M. Swaminathan, "Statistical signal integrity analysis and diagnosis methodology for high-speed systems," *IEEE Trans. on Advanced Packaging*, vol. 27, no. 4, pp. 611–629, Nov. 2004.