Study and Development of an Efficient RC-in–RC-out MOR Method

Pekka Miettinen, Mikko Honkala, and Janne Roos
Helsinki University of Technology,
Department of Radio Science and Engineering,
P.O.Box 3000, FI-02015 TKK, Finland.
E-mail: {pekka,mikko,janne}@ct.tkk.fi

Carsten Neff and Achim Basermann
NEC Laboratories Europe,
IT Research Division, NEC Europe Ltd.,
Rathausallee 10, D-53757 St. Augustin, Germany.
E-mail: {neff,basermann}@ccrl-nece.de

***Abstract*— This paper outlines the study and development of an efficient RC-in–RC-out Model-Order Reduction (MOR) method suitable for reduction of very large sized RC circuits or the RC circuit parts of a non-RC circuit. The MOR is carried out on a partitioned circuit, which enables the use of low-order moments and macromodels of few elements. This benefit translates to a typical 10-100 times faster simulation with only a minimal error. The performance of the MOR method is evaluated with simulations and compared with other MOR algorithms.**

I. Introduction

As the size and complexity of industrial circuits increase rapidly, the need for even faster simulation methods becomes substantial. One avenue to speed up transistor-level simulations is to apply MOR to the extracted RC(L) parasitics describing the layout effects. RLC reduction is a favourite research topic, but it is rarely observed in existing commercial tools. This is because the on-chip inductance effect is still not an issue among the top priorities and RLC MOR is far more complicated than RC MOR [1]. Since the RC MOR methods are quite commonly used in fast SPICE-like simulators, more efficient RC MOR methods are of interest.

In order to facilitate the analysis of large circuits, it has often been found useful to partition a circuit into smaller sections, which can then be analyzed separately. By thus reducing the size and order of the corresponding system matrices, the manipulation of the matrices is facilitated.

The starting point for this paper was the RC MOR method proposed by Liao and Dai [2]. The concept behind this method is to partition the circuit into subcircuits, which in turn can be approximated with low-order macromodels. In this paper, the Liao–Dai method is conceptually divided into three steps: circuit partitioning (Sec. II), calculation of y-parameter moments (Sec. III), and macromodel synthesis (Sec. IV). Detailed improvements regarding speed, reduction, and accuracy are developed and implemented at each step. In Sec. V, the simulation results for these implementations are presented and our RC MOR method is compared to the original Liao–Dai [2], PRIMA [3], and TICER [4] MOR methods. Finally, conclusions are detailed in Sec. VI. As a result, the MOR method presented here was used succesfully in a prototype-like industrial netlist-in–netlist-out MOR tool.

II. Circuit partitioning

Since the RC MOR method studied is based on approximating interconnects between port nodes with low-order macromodels, it is necessary to perform a partition on the large circuit prior to macromodel synthesis. The quality of the subcircuits is essential; if the subcircuits are too complex, the low-order macromodel used later is not accurate enough to model the partition, while if the subcircuit is too simple, the order of reduction is small and the MOR is of little use.

A. Original method

The partitioning via the original Liao–Dai method [2] is done by considering the RC netlist as a weighted graph $G(V, E)$. Vertices V consist of circuit elements and circuit nodes that connect more than two elements, and edges E represent the adjacency between elements in the circuit. For each edge, a weight is determined based on the number of ports in each subcomponent the edge is connected to, if the two subcomponents (i.e., preliminary RC connection blocks) were joined together. The partitioning process then combines elements together into larger subcomponents by eliminating edges between the elements until the weight of all remaining edges is greater than a preset maximum weight for an edge. In practise, an edge between two subcomponents is eliminated, if the inequality

$$N_x^2 + N_y^2 > b(N_x + N_y - 2)^2 \tag{1}$$

holds, where N_x and N_y are the number of ports in two subcomponents, x and y, and b is the user defined contracting factor. The equation is derived from the size of the corresponding S-parameter matrix: the left-hand side of the equation describes the number of entries in the matrix before the elimination, and the right-hand side describes the number of entries after the elimination of an edge.

As a result, the circuit is divided into partitions with a number of ports that is equal to or less than the preset value.

B. hMETIS-based method

METIS [5] is an algorithm package for partitioning large irregular graphs, partitioning large meshes, and computing fill-reducing orderings of sparse matrices. The METIS algorithms are based on multilevel graph-partitioning algorithms, which

first reduce the size of the graph by coarsening the graph's details. This takes the form of collapsing adjacent vertices and edges. The smaller graph is then partitioned and refined into the original graph. hMETIS is an extension of METIS that uses hypergraphs instead of graphs [6].

As the partitioning algorithms operate with the reduced-size graph, they are extremely fast compared to traditional partitioning algorithms that compute a partition directly on the original graph. In Ref. [5], extensive testing showed that the partitions provided by METIS are consistently better than those produced by spectral partitioning algorithms.

The use of METIS and hMETIS algorithms especially in circuit partitioning was studied in Ref. [7], where it was noted that they both produced excellent partitionings of equal size. In this paper, we consider the hMETIS algorithm as a partitioning method in the MOR flow.

III. Calculation of y-parameter moments

The MOR is performed on a set of equations. As a typical approach, the y-parameters are used to describe the circuit characteristics. Once the y-parameters are determined, the block moments are calculated as the coefficients of the Taylor expansion around $s = 0$:

$$\mathbf{Y}(s) = \mathbf{M}_0 + \mathbf{M}_1 s + \mathbf{M}_2 s^2 + \cdots. \tag{2}$$

Here, only the first two moments $\mathbf{M}_0$ and $\mathbf{M}_1$, are used for the subcircuit macromodel.

A. Original method

The original method [2] first used S-parameters to generate the subcomponent transfer function and afterwards the y-parameters were calculated using the S-parameters from the relation $\mathbf{Y}(s) = \mathbf{Z}_0^{-1}(\mathbf{I} + \mathbf{S})^{-1}(\mathbf{I} - \mathbf{S})$, where $\mathbf{Z}_0$ and $\mathbf{I}$ are the diagonal reference-impedance and identity matrices, respectively.

B. MNA-based method

The time-domain MNA circuit equations for an N-port can be expressed as

$$\begin{cases} \mathbf{C}\dfrac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = -\mathbf{G}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{i}(t) = \mathbf{L}^{\mathrm{T}}\mathbf{x}(t), \end{cases} \tag{3}$$

where $\mathbf{C}$ and $\mathbf{G}$ are susceptance and conductance matrices (with rows corresponding to the current variables negated [3]), $\mathbf{i}$, $\mathbf{x}$, and $\mathbf{u}$ denote the port currents, voltages, and voltage inputs, and $\mathbf{B} = \mathbf{L}$ is a selector matrix consisting of ones, minus ones and zeroes.

Taking the Laplace transformation of Eq. (3), solving for the port currents, and defining $\mathbf{A} \equiv -\mathbf{G}^{-1}\mathbf{C}$, $\mathbf{R} \equiv \mathbf{G}^{-1}\mathbf{B}$, the y-parameter matrix is given as

$$\mathbf{Y}(s) = \mathbf{L}^{\mathrm{T}}(\mathbf{I} - s\mathbf{A})^{-1}\mathbf{R}, \tag{4}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix, n being the dimension of the $\mathbf{C}$ and $\mathbf{G}$ matrices. $(\mathbf{I} - s\mathbf{A})^{-1}$ may be expanded into a Neumann series to obtain Eq. (2). The block moments of $\mathbf{Y}$ can be calculated using the relation $\mathbf{M}_i = \mathbf{L}^{\mathrm{T}}\mathbf{A}^i\mathbf{R}$ [3]. Note that the dimension of the block moments $\mathbf{M}_i$ is the same as the number of ports in the circuit.

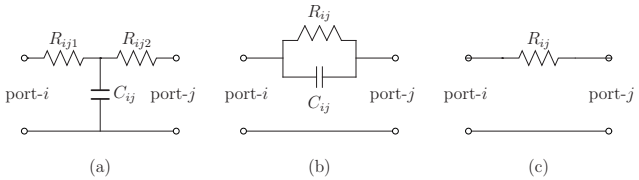

Fig. 1. Circuit macromodels between pairs of ports.

C. Nodal-formulation-based method

Let us write the nodal equations of the RC circuit as $(\mathbf{G} + s\mathbf{C})\,\mathbf{x} = \mathbf{b}$. If the matrices are arranged blockwise as

$$\left(\begin{bmatrix} \mathbf{G}_{\mathrm{p}} & \mathbf{G}_{\mathrm{c}}^{\mathrm{T}} \\ \mathbf{G}_{\mathrm{c}} & \mathbf{G}_{\mathrm{i}} \end{bmatrix} + s \begin{bmatrix} \mathbf{C}_{\mathrm{p}} & \mathbf{C}_{\mathrm{c}}^{\mathrm{T}} \\ \mathbf{C}_{\mathrm{c}} & \mathbf{C}_{\mathrm{i}} \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}_{\mathrm{p}} \\ \mathbf{x}_{\mathrm{i}} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{\mathrm{p}} \\ 0 \end{bmatrix}, \tag{5}$$

where $\mathbf{x}_{\mathrm{p}}$ and $\mathbf{x}_{\mathrm{i}}$ are port and internal node voltages, respectively, and $\mathbf{b}_{\mathrm{p}}$ is the port-current input, then y-parameters can be computed as follows:

$$\mathbf{Y}(s) = \mathbf{G}_{\mathrm{p}} + s\mathbf{C}_{\mathrm{p}} - (\mathbf{G}_{\mathrm{c}} + s\mathbf{C}_{\mathrm{c}})^{\mathrm{T}}(\mathbf{G}_{\mathrm{i}} + s\mathbf{C}_{\mathrm{i}})^{-1}(\mathbf{G}_{\mathrm{c}} + s\mathbf{C}_{\mathrm{c}}). \tag{6}$$

Now, by applying Ref. [8], we get

$$\mathbf{Y}(s) \approx \mathbf{M}_0 + \mathbf{M}_1 s, \tag{7}$$

where $\mathbf{M}_0 = \mathbf{G}_{\mathrm{P}} - \mathbf{G}_{\mathrm{C}}^{\mathrm{T}}\mathbf{V}$, $\mathbf{M}_1 = \mathbf{C}_{\mathrm{P}} - \mathbf{W}^{\mathrm{T}}\mathbf{V} - \mathbf{V}^{\mathrm{T}}\mathbf{C}_{\mathrm{C}}$, and, furthermore, $\mathbf{V} = \mathbf{G}_{\mathrm{I}}^{-1}\mathbf{G}_{\mathrm{C}}$, $\mathbf{W} = \mathbf{C}_{\mathrm{C}} - \mathbf{C}_{\mathrm{I}}\mathbf{V}$. Cholesky factorization can be used for the inversion of the symmetric positive definite matrix $\mathbf{G}_{\mathrm{I}}$ [8].

IV. Macromodel synthesis

For an N-port, the admittance between the ith port and ground is given by the sum of the ith row (or column) of its Y-matrix, $\mathbf{Y}(s)$. The admittance connecting port-i and port-j is $-y_{ij}$. Thus, the circuit synthesis problem amounts to synthesizing admittances between a port and ground and between pairs of ports with lumped R and C elements. Here, a moment-matching technique is used to synthesize the admittance matrices of an RC circuit. Once $\mathbf{M}_0$ and $\mathbf{M}_1$ have been calculated using one of the methods described in the previous section, each element of $\mathbf{Y}(s)$ can be expressed as

$$y_{ij} \approx m_0^{ij} + m_1^{ij} s. \tag{8}$$

A. Original method

Figure 1 shows the macromodel realizations between two ports i and j in different situations [2]. For each port, a terminal macromodel consisting of a parallel resistance and a capacitance is also needed. In case of a one-port, the circuit is replaced with a single port macromodel. Depending on the m_1^{ij}, different macromodels are used between ports i and j:

- The T-circuit in Fig. 1(a) may be used if $m_1^{ij} \geq 0$. Along with the port macromodel this creates a 2Π circuit.
- If m_1^{ij} is negative, Fig. 1(b) must be used instead, which forms, combined with the port models, a Π circuit.

1) T model (2Π model): The creation of the T model shown in Fig. 1(a), where the circuit parameters are determined by matching the moments m_0^{ij} and m_1^{ij} of y_{ij} is discussed first. The T-model together with the capacitances at two ports, to be synthesized with the y_{ii}, forms a 2Π model. By matching m_0^{ij} and m_1^{ij} in Eq. (8) and the relation m_1^{ii}/m_1^{jj} with the admittance matrix of the 2Π model, we obtain the values of R_{ij1}, R_{ij2} and C_{ij}:

$$\begin{cases} R_{ij1} = \dfrac{-\sqrt{m_1^{jj}}}{m_0^{ij}(\sqrt{m_1^{ii}}+\sqrt{m_1^{jj}})}, \\ R_{ij2} = \dfrac{-\sqrt{m_1^{ii}}}{m_0^{ij}(\sqrt{m_1^{ii}}+\sqrt{m_1^{jj}})}, \\ C_{ij} = \dfrac{m_1^{ij}(\sqrt{m_1^{ii}}+\sqrt{m_1^{jj}})^2}{\sqrt{m_1^{ii}m_1^{jj}}}. \end{cases} \tag{9}$$

2) Π model: If the circuit contains floating capacitances, m_1^{ij} may be negative and in this case, a floating capacitance to model the y_{ij} between port-i and port-j (see Fig. 1(b)) can be used. Thus, from the Π model admittance matrix, we have

$$R_{ij} = -\frac{1}{m_0^{ij}} \quad \text{and} \quad C_{ij} = -m_1^{ij}. \tag{10}$$

3) Circuit model of a port: For the diagonal elements y_{ii}, y_{i0} is modeled with a parallel RC model. If there is no DC path from port-i to ground, $m_{i0} = 0$, and the y_{i0} matrix element is modeled with a single capacitance. In general, the parameters of the model are

$$R_{ii} = \frac{1}{m_0^{i0}} \quad \text{and} \quad C_{ii} = m_1^{i0}. \tag{11}$$

B. Simplified method

If m_1^{ij} is positive, but only m_0^{ij} is to be matched, the circuit in Fig. 1(c) can be used. Along with the parallel capacitance and resistace port models this creates the simplified Π circuit.

V. Simulation examples and method comparison

The netlists used in the simulations are shown in Table I, where n, n_e, R, and C mean the number of all nodes, external nodes, resistances, and capacitances, respectively. Both AC and transient analysis was carried out on APLAC [9] on a HP RX5670/1.3 GHz computer with different parameters of which representative samples are shown in the following.

TABLE I
NETLISTS USED IN THE SIMULATIONS.

Name	n	n_e	R	C
`small`	105	10	80	277
`mem1`	1344	4	6997	165
`mem2`	1537	12	10432	197
`mem3`	1539	12	10440	195
`mem4`	8719	91	40974	1696
`clock`	12006	4	12008	12005

The small RC netlist `small` was studied mainly for verification purposes. On the other hand, the netlists `mem1`, ..., `mem4` are real-life netlists derived from a memory chip. They are characterized by a large number of resistances compared to that of capacitances, and a relatively small number of external nodes. Finally, the netlist `clock` [10] represents a grid-type interconnect clock distribution circuit with four resistance loops.

Table II shows some of the MOR results for `mem1`, `mem3`, `mem4`, and `clock` compared with the MOR methods PRIMA and TICER. Here, G, cp/%, $E_{\rm tr}$ and $T_{\rm tr}$ stand for the number of SPICE "G"-elements (voltage-controlled current sources), element-reduction ratio, relative transient-analysis error and transient-analysis CPU time, respectively. From the table it is clear that the partitioning-and-macromodel-based RC MOR method presented in this paper is generally considerably better for RC circuits than the node-eliminamation-based TICER or the RLC-oriented PRIMA. A more thorough analysis of this comparison can be found in [11].

TABLE II
COMPARISON OF PRIMA, MODIFIED LIAO–DAI, AND TICER.

c.	method	n	R	C	G	cp/%	$E_{\rm tr}$/%	$T_{\rm tr}$/s
m1	orig.	1344	6997	165	-	-	-	2.03
	PRIMA	14	12	11	64	99.80	0.005	0.13
	L–D	15	35	13	-	99.33	0.005	0.12
	TICER	982	6800	487	-	-1.74	0.000	5.93
m3	orig.	1539	10440	195	-	-	-	4.30
	PRIMA	24	36	32	576	93.94	0.021	0.37
	L–D	31	110	47	-	98.52	0.008	0.26
	TICER	989	7511	540	-	24.30	0.008	4.76
m4	orig.	8719	40974	1696	-	-	-	60
	PRIMA	273	182	249	33124	34.17	∞	∞
	L–D	169	446	446	-	96.88	0.001	2.04
	TICER	4782	32043	3026	-	17.81	0.000	214
cl.	orig.	12006	12008	12005	-	-	-	164
	PRIMA	-	-	-	-	-	∞	∞
	L–D	608	1230	607	-	92.35	0.357	1.23
	TICER	685	688	684	-	94.29	0.264	1.57

A. Circuit partitioning

While simulating circuits reduced with the original circuit-partitioning method, it soon became apparent that the criterion for subcomponents formation in Eq. (1) was generally too lax for adequate reduction. For example, considering only the number of ports in a subcomponent, a typical ladder subcircuit is always partitioned as one subcomponent regardless of the number of elements it is consisted of. This leads to poor accuracy, and thus, in addition to Eq. (1), it was necessary to add a user defined maximum for the number of elements a subcomponent may contain.

From Table III, it can be seen that the hMETIS algorithm reaches typically equally good, or better partitions as the original method [2]. Here, n_p and P.size stand for the number of partitions used and the approximate size of a partition, respectively. The table presents the best partitionings reached through iteration of user parameters. However, it should be noted that while the results with hMETIS grew better at each step, the partitions with the original Liao–Dai partitioning method fluctuated wildly in size and usability with larger

circuits, such that the majority was notably worse than those reached with hMETIS.

Partitions generated with hMETIS are of the same size by default, whereas the partitioning algorithm deployed by the original Liao–Dai method [2] strives, foremost, to limit the number of ports, and the partitioning is done case-by-case instead of considering the whole circuit topology at once. Since the size of a subcircuit is determined by the number of ports in the subcircuit, different parts of a circuit with different characteristic topologies generate partitions with different number of elements. When a macromodel is then used to approximate these varying partitions, uneven results are reached.

As there was apparently no simple method of determining the usability of the partitioning acquired with the original method beforehand, hMETIS was chosen as a more stable and suitable partitioning algorithm.

TABLE III

CIRCUIT PARTITIONING METHOD COMPARISON; BEST RESULTS.

c.	method	n_p	P.size	R	C	E_tr/%	T_tr/s
small	L–D	2	300	37	106	0.323	0.59
small	hMETIS	2	150	16	65	0.422	0.52
m2	L–D	4	5000	799	145	0.008	0.49
m2	hMETIS	2	5000	62	35	0.008	0.24
m4	L–D	54	1000	3212	834	0.000	2.98
m4	hMETIS	42	1000	887	446	0.001	1.81
cl.	L–D	485	25	971	484	0.522	0.98
cl.	hMETIS	480	50	990	488	0.538	0.85

B. Calculation of y-parameter moments

The advantage of the original S-parameter-based method [2] is tied to the partitioning step. The subcomponent S-parameters are updated with each edge elimination, and the calculation of the transfer function with S-parameters is relatively easier than with the y-parameters, for example.

However, since the S-parameters still need to be translated into y-parameters eventually, the original method has additional calculation steps compared to a more direct method.

As the hMETIS-based partitioning does not benefit from the intermediate S-parameter phase, the two other methods become more reasonable. In this work, the MNA-based method was used. If the the nodal-formulation-based method was used, Cholesky factorization could be employed, which is faster than the LU factorization generally used.

C. Macromodel synthesis

In the course of this work, it was noted that the 2Π method produced a considerable number of negative capacitances, which cause problems with circuit simulators. An alternative approach to using the 2Π model, the simplified Π model, was proposed and proved to be equal or better in accuracy and reduction than the 2Π model with appropriate number of partitions (note that more partitions are needed to reach the same accuracy). Table IV shows the results of comparing the two methods.

In [2], it is noted that the negative capacitances may be negated by scaling down all C_{ij} to keep the total capacitance unchanged. This is, however, a perturbation of the capacitance matrix, and causes an added error to the results.

TABLE IV

MACROMODEL COMPARISON.

method	c.	n_p	P.size	R	C	E_tr/%	T_tr/s
2Π	small	2	150	28	79	0.378	0.55
	small	4	80	50	182	0.278	0.71
Π	small	2	150	16	65	0.422	0.52
	small	4	80	27	122	0.185	0.63
2Π	m2	10	1000	195	127	0.004	0.34
	m2	2	5000	108	77	0.010	0.28
Π	m2	10	1000	113	48	0.008	0.26
	m2	2	5000	62	35	0.008	0.24
2Π	cl.	344	70	1082	716	0.298	1.24
	cl.	267	90	850	562	0.458	0.98
Π	cl.	600	40	1230	607	0.357	1.09
	cl.	480	50	990	487	0.538	0.89

VI. CONCLUSIONS

This paper presented an efficient and stable RC-in–RC-out MOR method suitable for the reduction of large RC circuits. Starting from the MOR concept outlined in [2], alternative solutions were proposed, implemented, and tested for circuit partitioning, y-parameter moment calculation, and macromodel synthesis. As a result, the RC MOR method improved notably in accuracy and reliability of reduction ratio. Additionally, the RC MOR method was compared with two other MOR methods; it was noted that with RC circuits, the partitioning and macromodel-based RC approach was generally superior to the nodal-elimination-based TICER and RLC-oriented PRIMA.

REFERENCES

[1] Z. Zhu, H. Peng, C.-K. Cheng, K. Rouz, M. Borah, and E. S. Kuh, "Two-Satge Newton–Raphson Method for Transistor-Level Simulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 5, pp. 881–895, May 2007.

[2] H. Liao and W. W.-M. Dai, "Partitioning and Reduction of RC Interconnect Networks Based on Scattering Parameter Macromodels," *Digest of Technical Papers of IEEE/ACM International Conference on Computer Aided Design*, pp. 704–709, 1995.

[3] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm," *IEEE Transactions on Computer-Aided design of Integrated Circuits and Systems*, vol. 17, no. 88, pp. 645–654, August 1998.

[4] B. N. Sheehan, "TICER: Realizable Reduction of Extracted RC Circuits," *Proceedings of the 1999 IEEE/ACM International Conference on Computer-Aided Design*, San Hose, CA, 1999, pp. 200–203.

[5] G. Karypis and V. Kumar, *METIS, A software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices (Version 4.0)*.

[6] G. Karypis and V. Kumar, *hMETIS, A Hypergraph Partitioning Package (Version 1.5.3)*.

[7] P. Miettinen, M. Honkala, and J. Roos, *Using METIS and hMETIS Algorithms in Circuit Partitioning*, HUT CTL, Report CT-49, 2006.

[8] K. J. Kerns and A. T. Yang, "Stable and Efficient Reduction of Large, Multiport *RC* Networks by Pole Analysis via Congruence Transformations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol. 16, No 7, pp. 734–744, July 1997.

[9] *APLAC — Circuit Simulation and Design Tool*, Version 8.3 Manuals, AWR–APLAC Corporation, Finland, 2007.

[10] C. L. Ratzlaff, N. Gopal, and L. T. Pillage, "RICE: Rapid Interconnect Circuit Evaluator," *28th ACM/IEEE Design Automation Conference*, 1991.

[11] P. Miettinen, *Hierarchical Model-Order Reduction Tool for RLC Circuits*, TKK CTL, Master's Thesis, 2007.