TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

SparseRC : Sparsity preserving model reduction for RC circuits with many terminals

Citation for published version (APA):
Ionutiu, R., Rommes, J., & Schilders, W. H. A. (2011). SparseRC : Sparsity preserving model reduction for RC circuits with many terminals. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *30*(12), 1828-1841. https://doi.org/10.1109/TCAD.2011.2166075

DOI:
10.1109/TCAD.2011.2166075

Document status and date:
Published: 01/01/2011

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 07. Nov. 2026

SparseRC: Sparsity Preserving Model Reduction for RC Circuits with Many Terminals

Roxana Ionuţiu, *Associate Member, IEEE,* Joost Rommes, and Wil H. A. Schilders

***Abstract*—A novel model order reduction (MOR) method, SparseRC, for multiterminal RC circuits is proposed. Specifically tailored to systems with many terminals, SparseRC employs graph-partitioning and fill-in reducing orderings to improve sparsity during model reduction, while maintaining accuracy via moment matching. The reduced models are easily converted to their circuit representation. These contain much fewer nodes and circuit elements than otherwise obtained with conventional MOR techniques, allowing faster simulations at little accuracy loss.**

***Index Terms*—Circuit simulation, graphs, model reduction, moment matching, partitioning, passivity, sparsity, synthesis.**

I. Introduction

DURING THE design and verification phase of very large-scale integrated circuits, coupling effects between various components on a chip have to be analyzed. This requires simulation of electrical circuits consisting of many nonlinear devices together with extracted *parasitics*. Due to the increasing amount of parasitics, full device-parasitic simulations are too costly and often impossible. Hence, reduced models are sought for the parasitics, which when recoupled to the devices can reproduce the original circuit behavior.

Parasitic circuits are very large network models containing millions of nodes interconnected via basic circuit elements: R, RC, or RLC(k). Of the circuit nodes, a special subset form the *terminals*. These are the designer specified input/output nodes and the nodes connecting the parasitics to the nonlinear devices. Parasitic networks with millions of nodes, RC elements, and thousands of terminals are often encountered in real chip designs. A reduced order model for the parasitics ideally has fewer nodes and circuit elements than the original, and preserves the terminal nodes for reconnectivity. The presence of many terminals introduces additional structural and computational challenges during model order reduction (MOR). Existing MOR methods may be unsuitable for circuits with many terminals as they produce *dense* reduced models. These correspond to circuits with fewer circuit nodes, but more circuit elements (Rs, Cs) than the original circuit, and may even require longer simulation times than originally. Furthermore, if terminal connectivity is affected, additional elements such as current/voltage-controlled sources must be introduced to reconnect reduced parasitics to other devices.

The emerging problem is to develop *efficient* model reduction schemes for large multiterminal circuits that are accurate, sparsity preserving, and also preserve terminal connectivity. The method, SparseRC, proposed here achieves these goals by efficiently combining the strengths of existing MOR methodology with graph-partitioning and fill-reducing node reordering strategies, achieving tremendous reduction rates even for circuits with terminal numbers exceeding thousands. Reduced RC models thus obtained are sparser than those computed via conventional techniques, have shorter simulation times, and also accurately approximate the input/output behavior of the original RC circuit. In addition, the reduced RC parasitics can be reconnected directly via the terminal nodes to remaining circuitry without introducing new circuit elements.

A comprehensive coverage of established MOR methods is available in [1], while [2] collects more circuit simulation-specific contributions. Mainly, MOR methods are classified into truncation-based (modal approximation [3]/balancing [4]) and Krylov-based methods, from which we mention PRIMA [5], SPRIM [6], or the dominant spectral zero method [7] as they are passivity preserving.[1] Generally, however, traditional MOR techniques cannot be applied to very large circuits with many terminals, due to computational limitations together with the aforementioned sparsity and reconnectivity considerations. While the multiterminal problem has been addressed in numerous works such as [8] and [9], it is usually less clear whether their performance scales with the number of ports, especially as this exceeds thousands.

Recent developments in model reduction for large multiterminal R-networks were achieved in [10] (denoted here as ReduceR), which uses graph tools, fill-in minimizing node reorderings, and node elimination to obtain sparse reduced R-networks. Toward obtaining sparse reduced models for multiterminal RC(L) networks, the sparse implicit projection

Manuscript received April 11, 2011; revised June 24, 2011; accepted July 29, 2011. Date of current version November 18, 2011. This work was supported in part by the COMSON European project, under Contract MRTN-CT 2005-019417, and by the Deutshcen Forschungsgemeinschaft (DFG) research program "AN-1 Modellreduktion." This paper was recommended by Associate Editor R. Suaya.

R. Ionuţiu is with the School of Engineering and Science, Jacobs University, Bremen, Germany, and with the Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, Eindhoven 5600 MB, The Netherlands (e-mail: r.ionutiu@gmail.com).

J. Rommes is with NXP Semiconductors, Eindhoven 5656 AE, The Netherlands (e-mail: joost.rommes@nxp.com).

W. H. A. Schilders is with the Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven 5600 MB, The Netherlands (e-mail: w.h.a.schilders@tue.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2011.2166075

[1] Only passive reduced order models guarantee stable results when recoupled to other circuit blocks in subsequent simulation stages [5].

0278-0070/$26.00 © 2011 IEEE

(SIP) method [11] also proposes reordering actions prior to eliminating unimportant internal nodes, and makes several important analogies between related node elimination-based methods (e.g., TICER [12] and [13]) and moment-matching MOR by projection (e.g., PRIMA [5]). In fact, the fundamental projection behind SIP can be traced back in the pole analysis via congruence transformations (PACT) methods of [14] and [15] for reducing multiterminal RC(L) networks.

As will be shown, SparseRC combines the advantages of ReduceR and SIP/PACT into an efficient procedure, while overcoming some of their computational limitations; using graph partitioning, circuit components are identified which are reduced individually via a PACT-like projection [denoted here as the *extended moment matching projection* (EMMP)] while appropriately accounting for the interconnection between components. The reduction process is simplified computationally, as smaller components are reduced individually. The final SparseRC reduced circuit matches by default two moments at DC of the original circuit's multiport admittance, and can be extended with dominant poles [3] or additional moments to improve accuracy at higher frequency points, if needed. Through partitioning, the relevant nodes responsible for fill-in are identified *automatically*; SparseRC preserves these along with the terminals to ensure the sparsity of the reduced model. This feature makes SparseRC more efficient than ReduceR or SIP; it avoids the unnecessary computational cost of monitoring fill-in at each node elimination step.

A related method is PartMOR [16], which is based on the same partitioning philosophy, but constructs the reduced models in a different manner. PartMOR realizes selected moments from each subnet into a netlist equivalent, while SparseRC is implemented as block moment matching projection operating on the matrix hierarchy which results from partitioning. This construction enables SparseRC to match admittance moments per subnet as well as for the recombined network. With global accuracy thus ensured, the approximation quality of the final SparseRC reduced model is guaranteed irrespective of the partitioning tool used or the number/sizes of partitions.

This paper focuses on RC reduction. For RC, a reducing transformation which matches multiport admittance moments is sufficient to ensure accuracy *and* can be so constructed as to improve sparsity. For RLC, however, additional accuracy considerations have to be accounted for as to capture oscillatory behavior. Hence, constructing a projection which simultaneously ensures accuracy and sparsity is more involved. RLC circuits can be partitioned with the same framework using a second order susceptance-based system formulation which reveals the network topology [17]. On the other hand, the dense nature of inductive couplings for RLCK circuits may prevent finding a good partition. These topics are subject to ongoing research; for an existing RLC partitioned-based approach, we refer to PartMOR [16].

The remainder of this paper is structured as follows. Section II formulates the multiterminal model reduction problem. The SparseRC partitioning-based strategy is detailed in Section III, the main focus of this paper. Numerical results and circuits simulations are presented in Section IV. Section V concludes this paper. Some conventions on notation and terminology follow next. *Matrices*: $\mathbf{G}$ and $\mathcal{G}$ are used interchangeably for the conductance matrix, depending on whether the context refers to unpartitioned or partitioned matrices, respectively (similarly for the capacitance matrix $\mathbf{C}$, $\mathcal{C}$, or the incidence matrix $\mathbf{B}$, $\mathcal{B}$). *Graphs*: G (nonbold, noncalligraphic) is a graph associated with the nonzero pattern of the circuit matrices, C (nonbold, noncalligraphic) is a component of G, and *nzp* is the nonzero-pattern of a matrix, i.e., its graph topology. *Dimensions*: p, number of circuit terminals (external nodes), the same for the original and reduced matrices/circuit, n, the number of internal nodes of the original circuit, k, the number of internal nodes of the reduced circuit, and N, number of matrix partitions. *Nodes*: circuit nodes prior to partitioning are classified into terminals and internal nodes; separator nodes are a subset of the original nodes identified through partitioning as communication nodes among individual components. *Terminology*: a partition/subnet/block describes the same concept. An individual graph/circuit/matrix component identified from partitioning; similarly, a separator, border is a component containing only separator nodes.

II. Problem Formulation

This section provides the preliminaries for model reduction of general RC circuits and identifies the challenges emerging in multiterminal model reduction. The building block for SparseRC is described: EMMP for reducing multiterminal RC circuits.

A. Model Reduction

Similarly to [14], consider the modified nodal analysis (MNA) description of an RC circuit as follows:

$$(\mathbf{G} + s\mathbf{C})\mathbf{x}(s) = \mathbf{B}\mathbf{u}(s) \tag{1}$$

where MNA matrices $\mathbf{G}$, $\mathbf{C}$ are symmetric, nonnegative definite, corresponding to the stamps of resistor and capacitor values, respectively. $\mathbf{x} \in \mathbb{R}^{n+p}$ denote the node voltages (at the n internal nodes and the p terminals) and $n + p$ is the dimension of (1). $\mathbf{u} \in \mathbb{R}^p$ are the currents injected into the terminals. The outputs are the voltage drops at the terminals: $\mathbf{y}(s) = \mathbf{B}^T\mathbf{x}(s)$. The underlying matrix dimensions are $\mathbf{G}, \mathbf{C} \in \mathbb{R}^{(n+p)\times(n+p)}$, $\mathbf{B} \in \mathbb{R}^{(n+p)\times p}$. In model reduction, an appropriate $\mathbf{V} \in \mathbb{R}^{(n+p)\times(k+p)}, k \geq 0$ is sought, such that the system matrices and unknowns are reduced to

$$\begin{aligned} \widehat{\mathbf{G}} &= \mathbf{V}^T\mathbf{G}\mathbf{V}, \quad \widehat{\mathbf{C}} = \mathbf{V}^T\mathbf{C}\mathbf{V} \quad \in \mathbb{R}^{(k+p)\times(k+p)} \\ \widehat{\mathbf{B}} &= \mathbf{V}^T\mathbf{B} \in \mathbb{R}^{(k+p)\times p}, \quad \widehat{\mathbf{x}} = \mathbf{V}^T\mathbf{x} \in \mathbb{R}^{k+p} \end{aligned}$$

and satisfy $(\widehat{\mathbf{G}} + s\widehat{\mathbf{C}})\widehat{\mathbf{x}}(s) = \widehat{\mathbf{B}}\mathbf{u}(s)$.

The transfer function $\mathbf{H}(s) = \mathbf{B}^T(\mathbf{G} + s\mathbf{C})^{-1}\mathbf{B}$ characterizes the system's behavior at the input/output ports (here, at the terminal nodes) over the frequency sweep s. After reduction, this becomes $\widehat{\mathbf{H}}(s) = \widehat{\mathbf{B}}^T(\widehat{\mathbf{G}} + s\widehat{\mathbf{C}})^{-1}\widehat{\mathbf{B}}$. A "good" reduced model/circuit generally satisfies the following:

1) gives a small approximation error $\|\mathbf{H} - \widehat{\mathbf{H}}\|$ in a suitably chosen norm, for instance, by ensuring that $\widehat{\mathbf{H}}$ matches moments of the original $\mathbf{H}$ at selected frequency points;
2) preserves passivity (and stability implicitly);

3) can be computed efficiently.

For multiterminal circuits especially, new conditions emerge:

4) for reconnectivity purposes, the incidence of current injections into terminal nodes is preserved (i.e., $\widehat{\mathbf{B}}$ is a submatrix of $\mathbf{B}$);
5) $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{C}}$ are sparse.

SparseRC is a multiterminal RC reduction method which meets targets 1)–5), as will be shown.

B. Multiterminal RC Reduction with Moment Matching

The EMMP is a moment matching reduction method for multiterminal RC circuits derived from PACT [14] (and conceptually similar to SIP [11]). Being suitable for multiterminal RC circuits with relatively few terminals only, this projection will be applied, after partitioning, in a block-wise manner inside SparseRC, as described in Section III-B. The description here covers only material from [14] that is relevant for SparseRC.

1) *Original Circuit Model* (1): $\mathbf{G}, \mathbf{C} \in \mathbb{R}^{(n+p)\times(n+p)}, \mathbf{B} \in \mathbb{R}^{(n+p)\times p}$: Recalling (1), let the nodes $\mathbf{x}$ be split into *selected nodes* $\mathbf{x}_S$ (terminals and separator nodes[2]) to be preserved, and internal nodes to be eliminated $\mathbf{x}_R$, revealing the following structure:

$$\left(\begin{bmatrix} \mathbf{G}_R & \mathbf{G}_K \\ \mathbf{G}_K^T & \mathbf{G}_S \end{bmatrix} + s\begin{bmatrix} \mathbf{C}_R & \mathbf{C}_K \\ \mathbf{C}_K^T & \mathbf{C}_S \end{bmatrix}\right)\begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_S \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_S \end{bmatrix}\mathbf{u}. \quad (2)$$

Note that [14] uses a simple block separation into "purely" terminal nodes $\mathbf{x}_S$ and internal nodes $\mathbf{x}_R$. Promoting separator nodes along with terminals inside $\mathbf{x}_S$ will ultimately positively influence the sparsity of the reduced model. The congruence transform applied to (2), $\mathbf{X}^T\mathbf{G}\mathbf{X}$, $\mathbf{X}^T\mathbf{C}\mathbf{X}$, $\mathbf{X}^T\mathbf{B}$, where [14]

$$\mathbf{X} = \left[\begin{array}{c|c} \mathbf{I} & -\mathbf{G}_R^{-1}\mathbf{G}_K \\ \mathbf{0} & \mathbf{I} \end{array}\right], \quad \mathbf{x}' = \mathbf{X}^T\mathbf{x} \quad (3)$$

yields

$$\left(\left[\begin{array}{c|c} \mathbf{G}_R & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{G}_S' \end{array}\right] + s\left[\begin{array}{c|c} \mathbf{C}_R & \mathbf{C}_K' \\ \hline \mathbf{C}_K'^T & \mathbf{C}_S' \end{array}\right]\right)\left[\begin{array}{c} \mathbf{x}_R \\ \hline \mathbf{x}_S' \end{array}\right] = \left[\begin{array}{c} \mathbf{0} \\ \hline \mathbf{B}_S \end{array}\right]\mathbf{u} \quad (4)$$

where

$$\mathbf{G}_S' = \mathbf{G}_S - \mathbf{G}_K^T\mathbf{G}_R^{-1}\mathbf{G}_K, \quad \mathbf{W} = -\mathbf{G}_R^{-1}\mathbf{G}_K \quad (5)$$

$$\mathbf{C}_S' = \mathbf{C}_S + \mathbf{W}^T\mathbf{C}_R\mathbf{W} + \mathbf{W}^T\mathbf{C}_K + \mathbf{C}_K^T\mathbf{W} \quad (6)$$

$$\mathbf{C}_K' = \mathbf{C}_K + \mathbf{C}_R\mathbf{W}.$$

Expressing $\mathbf{x}_R$ in terms of $\mathbf{x}_S'$ from the first block-row of (4), and replacing it in the second gives

$$[\underbrace{(\mathbf{G}_S' + s\mathbf{C}_S')}_{\mathbf{Y}_S'(s)} - s^2\underbrace{\mathbf{C}_K'^T(\mathbf{G}_R + s\mathbf{C}_R)^{-1}\mathbf{C}_K'}_{\mathbf{Y}_R'(s)}]\mathbf{x}_S' = \mathbf{B}_S\mathbf{u}. \quad (7)$$

The expression (7) represents the circuit's multiport admittance, defined with respect to the selected nodes $\mathbf{x}_S$. $\mathbf{Y}'(s)$ captures the first two multiport admittance moments at $s = 0$ [14]. This is formalized as Proposition 1.

Proposition 1: For a multiterminal RC circuit of the form (2), the first two moments at $s = 0$ of the multiport admittance are given by $\mathbf{G}_S'$ and $\mathbf{C}_S'$ from (5) to (6).

[2]See Sections I and III-A for the definition of separator nodes.

Proof: See Appendix A. ■

The practical consequence of Proposition 1 is that, as with ReduceR, the path resistance of the original circuit is precisely (5) and, as shown next, is preserved by the reduced model. In addition to the path resistance, the slope of an RC circuit's response is captured by the second moment, namely, (6).

2) *Reduced Circuit Model:* $\widehat{\mathbf{G}}, \widehat{\mathbf{C}} \in \mathbb{R}^{(k+p)\times(k+p)}$, $k \geq 0$: By Proposition 1, the reduced model which preserves the first two admittance moments of the original (2) is revealed: eliminate nodes $\mathbf{x}_R$ (and the contribution $\mathbf{Y}_R'$) and retain nodes $\mathbf{x}_S'$. The corresponding moment matching projection is obtained by removing from $\mathbf{X}$ of (3) the columns corresponding to $\mathbf{x}_R$ as follows:

$$\mathbf{V} = \begin{bmatrix} -\mathbf{G}_R^{-1}\mathbf{G}_K \\ \mathbf{I} \end{bmatrix}, \quad (8)$$

$$\widehat{\mathbf{G}} = \mathbf{V}^T\mathbf{G}\mathbf{V} = \mathbf{G}_S', \quad \widehat{\mathbf{C}} = \mathbf{V}^T\mathbf{C}\mathbf{V} = \mathbf{C}_S' \quad (9)$$

$$\widehat{\mathbf{B}} = \mathbf{V}^T\mathbf{B} = \mathbf{B}_S, \quad \widehat{\mathbf{x}} = \mathbf{V}^T\mathbf{x} = \mathbf{x}_S'. \quad (10)$$

For simplicity, the reducing projection $\mathbf{V}$ from (8) shall be referred to further-on as the EMMP. It matches multiport admittance moments defined by terminals *and* the preserved internal nodes, rather than, as in PACT, by terminal nodes only. In other words, EMMP is the extension of the PACT [14] or SIP [11] projection to include the separator nodes.

a) *On the Singularity of* $\mathbf{G}$: Conductance $\mathbf{G}$ and capacitance $\mathbf{C}$ matrices describing parasitic RC circuits in MNA form are often singular, thus one must ensure that the EMMP projection (8) inverts only nonsingular $\mathbf{G}_R$ blocks. This is easily achieved by exploiting the properties of MNA matrices (e.g., definiteness, diagonal dominance), and a simple grouping of nodes so that internal nodes (i.e., rows/columns) responsible for the singularity of $\mathbf{G}$ are excluded from $\mathbf{G}_R$ (and promoted to $\mathbf{G}_S$) without any accuracy loss. Similar actions for ensuring the invertibility of $\mathbf{G}_R$ are detailed in [14].

Reduction via the EMMP already meets some of the challenges defined at the beginning of Section II. 1) Two multiport admittance moments are preserved irrespective of the separation level of $\mathbf{x}$ into $\mathbf{x}_R$ and $\mathbf{x}_S$, provided that $\mathbf{x}_R$ are internal nodes (thus the incidence matrix $\mathbf{B}_R = \mathbf{0}$); this ensures that accuracy is maintained via moment matching also when EMMP is later applied in the partitioned framework (see Section III-B). 2) Passivity is preserved [14], as $\mathbf{V}$ is a congruence transformation projecting the original positive semidefinite matrices $\mathbf{G}$ and $\mathbf{C}$ into reduced matrices $\widehat{\mathbf{G}}$, $\widehat{\mathbf{C}}$ which remain positive semidefinite, and 4) the input/output incidence matrix $\mathbf{B}_S$ remains unaltered after reduction. Consequently, the reduced model can be reconnected directly via the terminal nodes to remaining circuitry (e.g., nonlinear devices), without introducing new circuit elements such as controlled sources. The efficiency 3) and sparsity 5) considerations however are not met by EMMP alone when the original circuits have nodes, circuit elements, and terminals exceeding thousands. First, constructing $\mathbf{G}_R^{-1}\mathbf{G}_K$ is either too costly or unfeasible. Second, $\widehat{\mathbf{G}}$, $\widehat{\mathbf{C}}$ from (9) may become too dense.

C. Fill-In and Its Effect in Synthesis

Usually, $\mathbf{G}$ and $\mathbf{C}$ describing circuits from real chip designs are large and sparse, while the $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{C}}$ as obtained from (9)

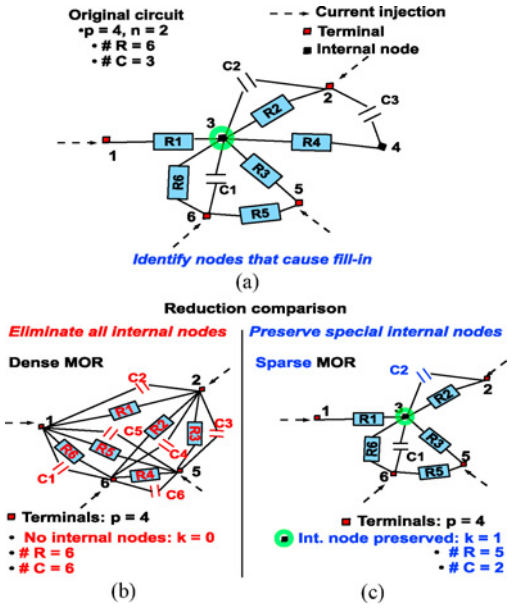

Fig. 1. (a) RC circuit to be reduced, containing $p = 4$ terminals and $n = 2$ internal nodes. Node 3 is a special internal node with many connections to other nodes. (b) Dense reduced model, where all internal nodes (3 and 4) were eliminated, but more circuit elements are generated. (c) Sparse reduced model (with fewer circuit elements) obtained from keeping node 3 and eliminating only node 4.

are small, but dense. Furthermore, they are only mathematical constructions, thus a *synthesis* procedure is required to convert the reduced matrix representation back into a RC netlist. This is obtained by unstamping the nonzero entries of $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{C}}$ into the corresponding resistor and capacitor topology, respectively, while $\widehat{\mathbf{B}}$ (being a submatrix of $\mathbf{B}_S$) is mapped directly into the original current injection at terminals. Unstamping is done via RLC equivalent circuit synthesis (RLCSYN) [17]. The dimension of $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{C}}$ gives the number of nodes, while the number of their nonzero entries dictates how many Rs and Cs are present in the reduced netlist. Therefore, while limiting the size of $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{C}}$, it is critical to also ensure their sparsity.

The simple example in Fig. 1 compares two reduced netlists derived from a small circuit. The dense reduced model has fewer nodes but more R, C elements than the original, while the sparse reduced model has both fewer nodes and R, C elements. The sparse model was obtained by preserving a node which would introduce fill-in if eliminated. Identifying such nodes by inspection is no longer possible for very large circuits. In practice, such nodes can be identified using reordering techniques, as explained next.

1) *Improving Sparsity with Node Reorderings:* At the basis of sparsity preserving MOR lies the following observation: the congruence transform $\mathbf{X}$ from (3) is a partial Cholesky factorization [18] of $\mathbf{G}$ [11]. Just as fill-reducing matrix reorderings are used for obtaining sparser factorizations, so can they be applied for sparsity preserving model reduction. These lie at the heart of ReduceR [10] and SIP [11], where the idea is to preorder the matrix for instance with constrained approximate minimum degree (CAMD) [19], [20], so that nodes responsible for fill-in are placed toward the end of the elimination sequence, along with the terminals. By eliminating the nodes one by one and keeping track of the fill-in generated

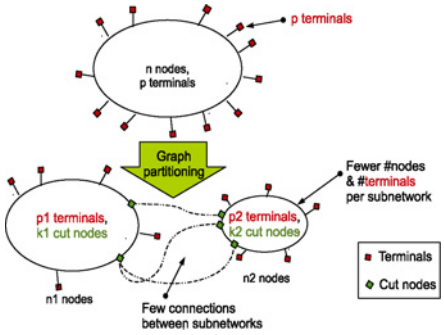

Fig. 2. Graph partitioning with separation of terminals.

at each step, the circuit with the fewest number of elements can be determined. For circuits with challenging topologies, however (i.e., with more internal nodes, terminals, or circuit elements), these actions are either too costly or even unfeasible (see the results in Section IV-B). SparseRC avoids them by exploiting graph partitioning and an appropriate matrix structure which allow for separator (fill-creating) nodes to be identified and skipped automatically in the reduction process, thus ensuring a desirable level of sparsity.

The following sections show how SparseRC, building upon the EMMP in combination with graph-partitioning and sometimes additional fill-reducing orderings, maintains 1), 2), 4) and in addition meets 3) efficiency and 5) sparsity requirements. These are crucial for successfully reducing challenging networks arising in industrial problems.

III. SparseRC Reduction Via Graph Partitioning

The analogy between circuits and graphs is immediate; the circuit nodes are the graph vertices, while the connections among them via R, C elements form the edges in the graph. For very large multiterminal circuits to be manageable at all with limited computational resources, a global partitioning scheme is proposed, visualized with the help of Fig. 2.[3] Essentially, an original large multiterminal circuit is split into subnetworks which have fewer nodes and terminals and are minimally connected among each other via separator (cut) nodes. SparseRC reduces each subnet individually, up to terminals and separator nodes. As all separator nodes are automatically preserved, sparsity is improved.

A. Partitioning and the BBD Form

Implemented as a divide and conquer reduction strategy, SparseRC first uses graph decompositions (based on the nzp of $\mathbf{G} + \mathbf{C}$) to identify individual subnets, as well as the separator nodes through which these communicate. Through partitioning, the original circuit matrices are reordered into the bordered block diagonal (BBD) [21] form; individual blocks form the subnets to be reduced, while the border blocks collect the separator nodes which are *all* preserved. The separator nodes are identified with the help of a separator tree indicating which of the subnets resulting from partitioning are in fact separators (this is usually a direct functionality of the partitioning algorithm, see for instance the MATLAB [22], [25] manual

[3]The two-way partitioning is presented here for simplicity; a natural extension is partitioning into $N > 2$ subnets.

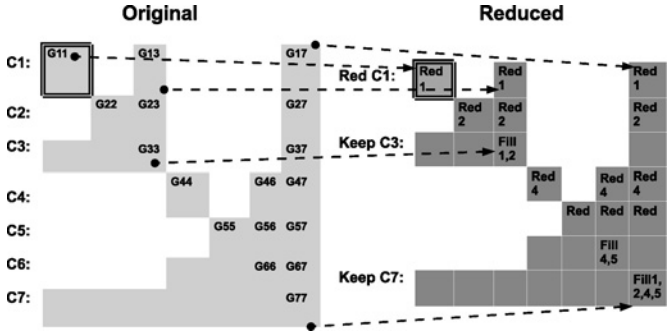

Fig. 3. Circuit matrices after partitioning, in BBD form (original-left versus reduced-right). For clarity, block dimensions are not drawn to scale; in practice, the separators are much smaller than the independent components, thus the borders are "thin." The individual blocks are reduced up to terminals, the borders are retained and updated. The number inside each independent component denotes the reduction step; this number is also stamped into the corresponding border blocks to mark fill-in. Example: reducing C_1 also updates the separators C_3 and C_7 and the corresponding borders. The "root" separator C_7 is updated from reducing all individual blocks $C_{1,2,4,5}$.

pages of `nesdis` reordering). In the *conquer* phase, individual blocks are reduced with the EMMP from Section II-B and the border blocks are correspondingly updated to maintain the moment matching property. A graphical representation of the 7-component BBD partitioning and reduction is shown in Fig. 3.[4] It is emphasized that, as reduction progresses, fill-in is isolated to the reduced parts of C_1, C_2, C_4, C_5, the separator blocks C_3, C_6, C_7, and the corresponding connection borders. As components are minimally connected, fill-in generated on the border is minimized.

B. Mathematical Formulation

The mathematical derivation of SparseRC follows, showing how reduction progressively traverses the BBD matrix structure, reducing individual components and updating the connectivity among them. Herein, $\mathcal{G}$ and $\mathcal{C}$ shall denote the original circuit matrices, while $\mathbf{G}$, $\mathbf{C}$ shall directly refer to matrix blocks associated with the EMMP reduction from Section II-B. Reconsider the original RC circuit in MNA form as follows:

$$(\mathcal{G} + s\mathcal{C})\mathbf{x}(s) = \mathcal{B}\mathbf{u}(s) \tag{11}$$

of dimension $n + p$, where n are internal nodes and p are terminals. The appropriate projection $\mathcal{V} \in \mathbb{R}^{(n+p)\times(k+p)}$, $k \geq 0$, is sought, which reduces (11) to

$$\widehat{\mathcal{G}}{=}\mathcal{V}^T\mathcal{G}\mathcal{V} \in \mathbb{R}^{(k+p)\times(k+p)},\ \widehat{\mathcal{C}}{=}\mathcal{V}^T\mathcal{C}\mathcal{V} \in \mathbb{R}^{(k+p)\times(k+p)} \tag{12}$$

$$\widehat{\mathbf{x}}{=}\mathcal{V}^T\mathbf{x} \in \mathbb{R}^{(k+p)},\ \widehat{\mathcal{B}}{=}\mathcal{V}^T\mathcal{B} \in \mathbb{R}^{(k+p)\times p}. \tag{13}$$

As illustrated in Section III-A, $\mathcal{V}$ is constructed step-wise using the BBD matrix reordering. Mathematically, this is shown via the simplest example of a BBD partitioning: a *bisection* into two independent components communicating via one separator (border) block. General reduction for a multilevel BBD partitioned system follows similarly. Consider the bisection of (11) as follows:

$$\begin{bmatrix} \mathcal{G}_{11} & 0 & \mathcal{G}_{13} \\ 0 & \mathcal{G}_{22} & \mathcal{G}_{23} \\ \mathcal{G}_{13}^T & \mathcal{G}_{13}^T & \mathcal{G}_{33} \end{bmatrix} + s \begin{bmatrix} \mathcal{C}_{11} & 0 & \mathcal{C}_{13} \\ 0 & \mathcal{C}_{22} & \mathcal{C}_{23} \\ \mathcal{C}_{13}^T & \mathcal{C}_{13}^T & \mathcal{C}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} \mathcal{B}_1 \\ \mathcal{B}_2 \\ \mathcal{B}_3 \end{bmatrix}. \tag{14}$$

Reducing (14) amounts to applying the EMMP from Section II-B on the individual components [here, $C_1 := \text{nzp}(\mathcal{G}_{11} + \mathcal{C}_{11})$ and $C_2 := \text{nzp}(\mathcal{G}_{22} + \mathcal{C}_{22})$]. The separator [here, $C_3 := \text{nzp}(\mathcal{G}_{33} + \mathcal{C}_{33})$] is kept and updated twice with the projections reducing C_1 and C_2, respectively. Naturally, the reduction is applied to the communication blocks $\mathcal{G}_{13}, \mathcal{C}_{13}, \mathcal{G}_{23}, \mathcal{C}_{23}$. Updating the separator and communication blocks at each individual reduction step ensures the preservation of admittance moments for the total recombined circuit (see Theorem 1 in Section III-C).

1) *Step 1:* Consider the reduction of subnetwork C_1 with EMMP, based on splitting $\mathbf{x}_1$ of (14) into $\mathbf{x}_{1_R}$ and $\mathbf{x}_{1_S}$, i.e., into the internal nodes (to be eliminated) and selected nodes (to be preserved) of subnet C_1 as follows:

$$\left[\begin{array}{c|ccc} \mathcal{G}_{11_R} & \mathcal{G}_{11_K} & \mathbf{0} & \mathcal{G}_{13_R} \\ \hline \mathcal{G}_{11_K}^T & \mathcal{G}_{11_S} & \mathbf{0} & \mathcal{G}_{13_S} \\ \mathbf{0} & \mathbf{0} & \mathcal{G}_{22} & \mathcal{G}_{23} \\ \mathcal{G}_{13_R}^T & \mathcal{G}_{13_S}^T & \mathcal{G}_{23}^T & \mathcal{G}_{33} \end{array}\right] + s \left[\begin{array}{c|ccc} \mathcal{C}_{11_R} & \mathcal{C}_{11_K} & \mathbf{0} & \mathcal{C}_{13_R} \\ \hline \mathcal{C}_{11_K}^T & \mathcal{C}_{11_S} & \mathbf{0} & \mathcal{C}_{13_S} \\ \mathbf{0} & \mathbf{0} & \mathcal{C}_{22} & \mathcal{C}_{23} \\ \mathcal{C}_{13_R}^T & \mathcal{C}_{13_S}^T & \mathcal{C}_{23}^T & \mathcal{C}_{33} \end{array}\right] \tag{15}$$
$$\mathbf{x}^T = [\mathbf{x}_{1_R}^T, \mathbf{x}_{1_S}^T, \mathbf{x}_2^T, \mathbf{x}_3^T]^T, \quad \mathcal{B}^T = [\mathbf{0}, \mathcal{B}_{1_S}^T, \mathcal{B}_2^T, \mathcal{B}_3^T].$$

Recognizing in (15) the structure (2), the EMMP-based transformation which reduces the network (15) by eliminating the internal nodes $\mathbf{x}_{1_R}$ is given by

$$\left[\begin{array}{c|cc} -\mathcal{G}_{11_R}^{-1}\mathcal{G}_{11_K} & \mathbf{0} & -\mathcal{G}_{11_R}^{-1}\mathcal{G}_{13_R} \\ \hline \mathbf{I}_{S_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{array}\right] = \mathbf{V}_1. \tag{16}$$

As with (8)–(10), the reduced model for (15) is computed from $\mathbf{G}_S^{'} = \mathbf{V}_1^T\mathcal{G}\mathbf{V}_1$, $\mathbf{C}_S^{'} = \mathbf{V}_1^T\mathcal{C}\mathbf{V}_1$. The reduced system is

$$\mathbf{G}_S^{'} = \begin{bmatrix} \widehat{\mathcal{G}}_{11} & \mathbf{0} & \widehat{\mathcal{G}}_{13} \\ \mathbf{0} & \mathcal{G}_{22} & \mathcal{G}_{23} \\ \widehat{\mathcal{G}}_{13}^T & \mathcal{G}_{23}^T & \widetilde{\mathcal{G}}_{33} \end{bmatrix}, \qquad \mathbf{C}_S^{'} = \begin{bmatrix} \widehat{\mathcal{C}}_{11} & \mathbf{0} & \widehat{\mathcal{C}}_{13} \\ \mathbf{0} & \mathcal{C}_{22} & \mathcal{C}_{23} \\ \widehat{\mathcal{C}}_{13}^T & \mathcal{C}_{23}^T & \widetilde{\mathcal{C}}_{33} \end{bmatrix} \tag{17}$$

$$\mathbf{B}_S = \begin{bmatrix} \widehat{\mathcal{B}}_1 \\ \mathcal{B}_2 \\ \mathcal{B}_3 \end{bmatrix}, \qquad \mathbf{x}_S^{'} = \begin{bmatrix} \widehat{\mathbf{x}}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} \tag{18}$$

where

$$\widehat{\mathcal{G}}_{11} = \mathcal{G}_{11_S} - \mathcal{G}_{11_K}^T\mathcal{G}_{11_R}^{-1}\mathcal{G}_{11_K} \tag{19}$$

$$\widehat{\mathcal{G}}_{13} = \mathcal{G}_{13_S} - \mathcal{G}_{11_K}^T\mathcal{G}_{11_R}^{-1}\mathcal{G}_{13_R} \tag{20}$$

$$\widetilde{\mathcal{G}}_{33} = \mathcal{G}_{33} - \mathcal{G}_{13_R}^T\mathcal{G}_{11_R}^{-1}\mathcal{G}_{13_R} \tag{21}$$

$$\widehat{\mathcal{C}}_{11} = \mathcal{C}_{11_S} + \mathcal{W}_{11}^T\mathcal{C}_{11_R}\mathcal{W}_{11} + \mathcal{W}_{11}^T\mathcal{C}_{11_K} + \mathcal{C}_{11_K}^T\mathcal{W}_{11} \tag{22}$$

$$\widehat{\mathcal{C}}_{13} = \mathcal{C}_{13_S} + \mathcal{W}_{11}^T\mathcal{C}_{13_R} + \mathcal{C}_{11_K}^T\mathcal{W}_{13} + \mathcal{W}_{11}^T\mathcal{C}_{11_R}\mathcal{W}_{13} \tag{23}$$

$$\widetilde{\mathcal{C}}_{33} = \mathcal{C}_{33} + \mathcal{W}_{13}^T\mathcal{C}_{11_R}\mathcal{W}_{13} + \mathcal{W}_{13}^T\mathcal{C}_{13_R} + \mathcal{C}_{13_R}^T\mathcal{W}_{13} \tag{24}$$

$$\widehat{\mathcal{B}}_1 = \mathcal{B}_{1_S}, \quad \widehat{\mathbf{x}}_1 = \mathbf{x}_{1_S}^{'} \quad \text{with} \tag{25}$$

$$\mathcal{W}_{11} = -\mathcal{G}_{11_R}^{-1}\mathcal{G}_{11_K}, \quad \mathcal{W}_{13} = -\mathcal{G}_{11_R}^{-1}\mathcal{G}_{13_R}. \tag{26}$$

The BBD form provides an important structural advantage, both in terms of identifying fill-in, as well as in implementation: reducing one subnet only affects the entries of the corresponding separator and border blocks, leaving the rest

[4]In implementation both $\mathbf{G}$ and $\mathbf{C}$ are in BBD form, in Fig. 3 G denotes simultaneously the corresponding blocks from both matrices.

of the independent subnets intact. Notice from (17) to (18) how reducing subnet C_1 has only affected its corresponding connection blocks to C_3, and the separator block C_3 itself. The blocks of subnet C_2 are not affected. This is because C_1 communicates with C_2 only via the separator C_3. Therefore, while reducing C_2, the already computed blocks of the reduced C_1 will no longer be affected. Only the connection blocks from C_2 to C_3 and the separator C_3 itself will be updated. Mathematically, this is shown next.

2) *Step 2:* Partition now the reduced $\mathbf{G}_S^{'}$, $\mathbf{C}_S^{'}$ matrices (17), (18) by splitting component C_2 according to $\mathbf{x}_{2_R}$ and $\mathbf{x}_{2_S}$ as follows:

$$\left[\begin{array}{cc|cc} \widehat{\mathcal{G}}_{11} & \mathbf{0} & \mathbf{0} & \widehat{\mathcal{G}}_{13} \\ \mathbf{0} & \mathcal{G}_{22_R} & \mathcal{G}_{22_K} & \mathcal{G}_{23_R} \\ \hline \mathbf{0} & \mathcal{G}_{22_K}^T & \mathcal{G}_{22_S} & \mathcal{G}_{23_S} \\ \widehat{\mathcal{G}}_{13}^T & \mathcal{G}_{23_R}^T & \mathcal{G}_{23_S}^T & \widetilde{\mathcal{G}}_{33} \end{array}\right] + s \left[\begin{array}{cc|cc} \widehat{\mathcal{C}}_{11} & \mathbf{0} & \mathbf{0} & \widehat{\mathcal{C}}_{13} \\ \mathbf{0} & \mathcal{C}_{22_R} & \mathcal{C}_{22_K} & \mathcal{C}_{23_R} \\ \hline \mathbf{0} & \mathcal{C}_{22_K}^T & \mathcal{C}_{22_S} & \mathcal{C}_{23_S} \\ \widehat{\mathcal{C}}_{13}^T & \mathcal{C}_{23_R}^T & \mathcal{C}_{23_S}^T & \widetilde{\mathcal{C}}_{33} \end{array}\right] \tag{27}$$

$$\mathbf{x}_S^{'T} = [\widehat{\mathbf{x}}_1^T, \mathbf{x}_{2_R}^T, \mathbf{x}_{2_S}^T, \mathbf{x}_3^T]^T, \quad \mathbf{B}_S^{'T} = [\widehat{\mathcal{B}}_1^T, \mathbf{0}, \mathcal{B}_{1_S}^T, \mathcal{B}_3^T]. \tag{28}$$

As before, the EMMP-based transformation which reduces the network (27) by eliminating nodes $\mathbf{x}_{2_R}$ is given by

$$\left[\begin{array}{ccc} \mathbf{I}_{S_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathcal{G}_{22_R}^{-1}\mathcal{G}_{22_K} & -\mathcal{G}_{22_R}^{-1}\mathcal{G}_{23_R} \\ \hline \mathbf{0} & \mathbf{I}_{S_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{array}\right] = \mathbf{V}_2. \tag{29}$$

The reduced model is obtained by projecting (27) and (28) with $\mathbf{V}_2$. $\widehat{\mathcal{G}} = \mathbf{V}_2^T \mathbf{G}_S^{'} \mathbf{V}_2$, $\widehat{\mathcal{C}} = \mathbf{V}_2^T \mathbf{C}_S^{'} \mathbf{V}_2$, $\widehat{\mathcal{B}} = \mathbf{V}_2^T \mathbf{B}_S$, $\widehat{\mathbf{x}} = \mathbf{V}_2^T \mathbf{x}_S^{'}$:

$$\widehat{\mathcal{G}} = \begin{bmatrix} \widehat{\mathcal{G}}_{11} & \mathbf{0} & \widehat{\mathcal{G}}_{13} \\ \mathbf{0} & \widehat{\mathcal{G}}_{22} & \widehat{\mathcal{G}}_{23} \\ \widehat{\mathcal{G}}_{13}^T & \widehat{\mathcal{G}}_{23}^T & \overline{\mathcal{G}}_{33} \end{bmatrix}, \qquad \widehat{\mathcal{C}} = \begin{bmatrix} \widehat{\mathcal{C}}_{11} & \mathbf{0} & \widehat{\mathcal{C}}_{13} \\ \mathbf{0} & \widehat{\mathcal{C}}_{22} & \widehat{\mathcal{C}}_{23} \\ \widehat{\mathcal{C}}_{13}^T & \widehat{\mathcal{C}}_{23}^T & \overline{\mathcal{C}}_{33} \end{bmatrix} \tag{30}$$

$$\widehat{\mathcal{B}} = \begin{bmatrix} \widehat{\mathcal{B}}_1 \\ \widehat{\mathcal{B}}_2 \\ \mathcal{B}_3 \end{bmatrix}, \qquad \widehat{\mathbf{x}} = \begin{bmatrix} \widehat{\mathbf{x}}_1 \\ \widehat{\mathbf{x}}_2 \\ \mathbf{x}_3 \end{bmatrix} \tag{31}$$

where (19)–(26) hold and

$$\widehat{\mathcal{G}}_{22} = \mathcal{G}_{22_S} - \mathcal{G}_{22_K}^T \mathcal{G}_{22_R}^{-1} \mathcal{G}_{22_K} \tag{32}$$

$$\widehat{\mathcal{G}}_{23} = \mathcal{G}_{23_S} - \mathcal{G}_{22_K}^T \mathcal{G}_{22_R}^{-1} \mathcal{G}_{23_R} \tag{33}$$

$$\overline{\mathcal{G}}_{33} = \widetilde{\mathcal{G}}_{33} - \mathcal{G}_{23_R}^T \mathcal{G}_{22_R}^{-1} \mathcal{G}_{23_R} \tag{34}$$

$$\widehat{\mathcal{C}}_{22} = \mathcal{C}_{22_S} + \mathcal{W}_{22}^T \mathcal{C}_{22_R} \mathcal{W}_{22} + \mathcal{W}_{22}^T \mathcal{C}_{22_K} + \mathcal{C}_{22_K}^T \mathcal{W}_{22} \tag{35}$$

$$\widehat{\mathcal{C}}_{23} = \mathcal{C}_{23_S} + \mathcal{W}_{22}^T \mathcal{C}_{23_R} + \mathcal{C}_{22_K}^T \mathcal{W}_{23} + \mathcal{W}_{22}^T \mathcal{C}_{22_R} \mathcal{W}_{23} \tag{36}$$

$$\overline{\mathcal{C}}_{33} = \widetilde{\mathcal{C}}_{33} + \mathcal{W}_{23}^T \mathcal{C}_{22_R} \mathcal{W}_{23} + \mathcal{W}_{23}^T \mathcal{C}_{23_R} + \mathcal{C}_{23_R}^T \mathcal{W}_{23} \tag{37}$$

$$\widehat{\mathcal{B}}_2 = \mathcal{B}_{2_S}, \quad \widehat{\mathbf{x}}_2 = \mathbf{x}_{2_S}^{'} \text{ with} \tag{38}$$

$$\mathcal{W}_{22} = -\mathcal{G}_{22_R}^{-1} \mathcal{G}_{22_K}, \quad \mathcal{W}_{23} = -\mathcal{G}_{22_R}^{-1} \mathcal{G}_{23_R}. \tag{39}$$

As seen from (34) and (37), separator blocks $\overline{\mathcal{G}}_{33}$ and $\overline{\mathcal{C}}_{33}$ are the further updated blocks $\widetilde{\mathcal{G}}_{33}, \widetilde{\mathcal{C}}_{33}$ (previously obtained from reducing C_1). The reduced model retains the BBD form, and the separator nodes are preserved in the blocks corresponding to $\overline{\mathcal{G}}_{33}$ and $\overline{\mathcal{C}}_{33}$. The p terminals are distributed among C_1, C_2, C_3 as seen from the form of $\widehat{\mathcal{B}}$ in (31). Equations (25), (31), and (38) together show that the input/output incidence matrix is preserved after reduction, thus the reduced netlist obtained from RLCSYN [17] unstamping preserves connectivity via the terminal nodes. In the general case, block-wise reduction of finer BBD partitions (into $N > 2$ subnets) follows in the same manner as the bisection framework presented here, with the appropriate projections of separator and border blocks. The moment-matching, terminal connectivity, and passivity requirements remain satisfied.

3) *Options for Further Improving Sparsity:* Partitioning provides an additional structural and computational advantage; if necessary, additional reordering and minimum-fill tracking options such as those employed by ReduceR/SIP can be applied per subnet. Naturally, such operations come at additional computational cost, but are still more efficient than monitoring fill-in directly from the unpartitioned circuit. So, while separator nodes already improve sparsity globally and automatically, fill-monitoring actions may further identify additional internal nodes to be preserved *locally* in each subnet. For instance, in the reduction scenario of Step 1 (see Section III-B1), the $\mathcal{G}_{11}, \mathcal{C}_{11}$ blocks of (14) would be reordered with CAMD and fill-tracking would identify which additional internal nodes should be preserved along with terminals in $\mathbf{x}_{1_S}$. This may improve sparsity inside $\widehat{\mathcal{G}}_{11}$, $\widehat{\mathcal{C}}_{11}$ (and correspondingly in $\widehat{\mathcal{G}}_{13}$, $\widehat{\mathcal{C}}_{13}$, $\widetilde{\mathcal{G}}_{33}$, $\widetilde{\mathcal{C}}_{33}$) even beyond the level already achieved by preserving the separators nodes. In Section IV, examples are provided to illustrate this effect. Nevertheless, such fill-monitoring operations are not always necessary; the sparsity level achieved directly from partitioning and preserving separator nodes is often sufficient. This is discussed in Section III-D1.

C. Moment Matching, Passivity, and Synthesis

Note that as $\widehat{\mathcal{G}} = \mathbf{V}_2^T \mathbf{G}_S^{'} \mathbf{V}_2 = \mathbf{V}_2^T \mathbf{V}_1^T \mathcal{G} \mathbf{V}_1 \mathbf{V}_2$ (similarly for $\widehat{\mathcal{C}}$, $\widehat{\mathcal{B}}$), the reducing projection from (12) to (13) is $\mathcal{V} := \mathbf{V}_1 \mathbf{V}_2$, with $\mathbf{V}_1$, $\mathbf{V}_2$ as deduced in (16) and (29), respectively. In efficient implementations, however, $\mathbf{V}_1$, $\mathbf{V}_2$, and $\mathcal{V}$ are never formed directly, rather they are formed block-wise as just shown. Only $\mathcal{W}_{11}$, $\mathcal{W}_{13}$, $\mathcal{W}_{22}$, $\mathcal{W}_{23}$ from (26) and (39), respectively, are explicitly formed. Next, it is shown that the $\mathcal{V}$ constructed from successive EMMPs matches the first two admittance moments at the end of the reduction procedure.

Theorem 1: Consider the original circuit (14) with matrices partitioned and structured in BBD form, which is reduced by applying successive EMMP projections (see Section II-B) on each subnet. The final reduced model (30), (31) preserves the first two multiport admittance moments around s=0 of each individual subnet and of the entire recombined circuit (14).

Proof: See Appendix A. ■

1) *Matching Behavior at Higher Frequency Points:* Although SparseRC is mainly based on matching the first two admittance moments at $s = 0$ (which proved sufficient for most experiments of Section IV), it is possible, when necessary, to additionally improve approximation at higher frequency points. One possibility is to include contributions from the otherwise neglected term $\mathbf{Y}_R^{'}(s)$ of the admittance response (7). Let for simplicity $\mathbf{Q}$ be a transformation which reduces $\mathbf{Y}_R^{'}(s)$. One can for instance perform the traditional PRIMA [5] reduction of $\mathbf{Y}_R^{'}(s)$ which constructs, for a chosen expansion point s_i, the Krylov subspace as follows: $\mathcal{K}_m(\mathbf{A}^{-1}\mathbf{C}_R, \mathbf{A}^{-1}\mathbf{C}_K^{'}) = \text{span}\left[(\mathbf{A}^{-1}\mathbf{C}_R)^{m-1}\mathbf{A}^{-1}\mathbf{C}_K^{'}\right]$, where $\mathbf{A} = \mathbf{G}_R + s_i\mathbf{C}_R$. If $\mathcal{K}_m \subseteq \mathbf{Q}$,

then $\mathbf{Q}$ matches $2m$ multiport admittance moments of $\mathbf{Y}'_R(s)$ at s_i. As a special case, if $s_i = 0$, then $\mathbf{Q}$ matches $2m$ multiport admittance moments at zero of (7), additionally to the 0th and 1st which are matched by default. Another option is to include in $\mathbf{Q}$ eigenvectors associated with the dominant eigenvalues of $\mathbf{Y}'_R(s)$, similarly to the pole matching proposed in [14]. Be it obtained either from moment or pole matching (or a combination of both), it is easily verified that the projection $\mathbf{Q}$ which reduces $\mathbf{Y}'_R(s)$ enters the transformation (3) as follows:

$$\mathbf{X}_Q = \left[\begin{array}{c|c} \mathbf{Q} & -\mathbf{G}_R^{-1}\mathbf{G}_K \\ \mathbf{0} & \mathbf{I} \end{array}\right].$$

To form $\mathbf{X}_Q$ within the partitioned framework, reconsider the reduction of subnet 1 from Section III-B1. The reducing transformation there is $\mathbf{V}_1$ (16), which preserved the first two admittance moments at $s = 0$ and eliminated entirely the contribution of internal nodes $\mathbf{x}_{1_R}$. Rather than eliminating $\mathbf{x}_{1_R}$, let $\mathcal{Q}_1$ be the transformation which reduces the internal matrices $\mathcal{G}_{11_R}$ and $\mathcal{C}_{11_R}$. Similarly, during the reduction of subnet 2 (Section III-B2), let $\mathcal{Q}_2$ be the transformation which reduces $\mathcal{G}_{22_R}$ and $\mathcal{C}_{22_R}$. $\mathcal{Q}_1$ and $\mathcal{Q}_2$ enter the projection for the recombined network as follows:

$$\mathcal{X}_Q = \begin{bmatrix} \mathcal{Q}_1 & \mathcal{W}_{11} & \mathbf{0} & \mathbf{0} & \mathcal{W}_{13} \\ \mathbf{0} & \mathbf{I}_{S_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{Q}_2 & \mathcal{W}_{22} & \mathcal{W}_{23} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{S_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \tag{40}$$

where (26) and (39) hold. Note that (40) is the extension with $\mathcal{Q}_1$ and $\mathcal{Q}_2$ of the default projection $\mathcal{V}$ of (41) which matches the first two multiport admittance moments at $s = 0$.

In a similar manner, a projection can be constructed which matches directly moments at $s_i \neq 0$ of the entire multiport admittance (7) (see [15] for details, which pertains more to RLC reduction). Given that a circuit's offset and slope at DC are precisely the first two admittance moments at $s = 0$ [14], matching these is a natural choice. We emphasize that, in contrast with SparseRC, direct moment matching at $s = 0$ cannot be achieved via PRIMA [5], or SPRIM [6] since the original $\mathcal{G}$ matrix is singular. To summarize, should additional accuracy be necessary when reducing RC circuits with SparseRC, it is safer to match the first two admittance moments at $s = 0$ and improve the response with contributions from the internal $\mathbf{Y}'_R(s)$ term as described above. This approach was implemented in the partitioned framework for two examples in Section IV-A2; the RC transmission line (TL) example where moments of $\mathbf{Y}'_R(s)$ are matched in addition, and the low noise amplifier (*LNA*) where dominant eigenmodes of $\mathbf{Y}'_R(s)$ are additionally matched.

SparseRC matches two moments at $s = 0$ by default. Hence, extra poles/moments can be added, if needed, after the default reduction, by rearranging the projection $\mathcal{X}_Q$ so that the extra $\mathcal{Q}_i$ columns are formed in a separate reduction run. The decision on whether to add extra poles or moments is difficult to make *a priori*, as is the case for any moment-based reduction method. One approach would be to compare the response of the original and default reduced order model for large frequencies. Should significant deviations be observed, then the addition of extra poles/moments is recommended. In PACT [14], poles are added based on a specified error and maximum operating frequency.

2) *Passivity and Synthesis:* As with PACT [14], the reducing projection $\mathcal{V}$ is a congruence transformation applied on original symmetric, nonnegative definite matrices, which gives reduced symmetric, nonnegative definite matrices (30). Consequently [14], the final reduced model (30), (31) is passive and stable, and the reduced netlist obtained from RLCSYN [17] unstamping remains passive irrespective of the values of the resulting R, C elements. If reduction is performed with the default two-moment matching at $s = 0$ (which was sufficient in all experiments except the two-port RC line of Section IV-A2b), the projection $\mathcal{V}$ of (41) guarantees that the unstamping of $\widehat{\mathcal{G}}$ generates only positive resistors. This is ensured by the special form of $\mathcal{V}$ which performs a Schur-complement operation on the original matrix $\mathcal{G}$ [23] (note that a standard moment matching projection as in PRIMA [5] does not guarantee positive resistors from unstamping, even though the reduced $\widehat{\mathcal{G}}$ is symmetric positive definite). While there may be negative capacitances resulting from unstamping $\widehat{\mathcal{C}}$, they do not violate the passivity of the netlist, nor its direct usability inside a simulator such as Spectre [24]. Furthermore, they do not prejudice the quality of the resimulation results as confirmed by Section IV. In fact, as also motivated in SIP [11], dropping negative capacitors from the reduced netlist is, in practice, a dangerous operation, so all capacitors are safely kept in. In the case of reduction with additional accuracy as in Section III-C1, the unstamping of $\widehat{\mathcal{G}}$ may generate negative resistors; these again posed no difficulties in the simulations performed (e.g., AC, transient, periodic steady state). PartMOR [16] presents an alternative reduction and synthesis strategy which ensures positive-only elements.

D. SparseRC Algorithm

The SparseRC pseudocode is outlined in Algorithms 1 and 2, which describe the most relevant reduction case of matching the first two admittance moments at $s = 0$. To summarize Algorithm 1, from the original circuit matrices $\mathcal{G}$, $\mathcal{C}$ and a vector of indices $\mathbf{e}$ denoting the original location of terminals (external nodes), SparseRC outputs the reduced circuit matrices $\widehat{\mathcal{G}}, \widehat{\mathcal{C}}$, and the vector $\widetilde{\mathbf{e}}$ denotes the new terminal locations. As an advanced option, "do_minfill" specifies whether additional fill-reducing orderings should be employed per partition. The graph G defined by the circuit topology (the nzp of $\mathcal{G}+\mathcal{C}$) is partitioned into N components. A permutation P (which reorders the circuit matrices in BBD form) is obtained, together with a vector *Sep* indicating which of the N components is a separator. For each nonseparator component C_k, defined by nodes $\mathbf{i}_k$, the corresponding matrix blocks are reduced with EMMP while accounting for the communication of C_k to the remaining components via the separator nodes $\mathbf{i}_{sep}$. All separator components are kept, after having been appropriately updated inside EMMP.

The $\mathcal{G}$, $\mathcal{C}$ matrices supplied at each step to EMMP (line 8 of Algorithm 1) are updated in place, and the reduction follows according to Algorithm 2. The $\mathbf{i}_k$ index selects the component to be reduced (say, C_k) from the supplied $\mathcal{G}$, $\mathcal{C}$, while $\mathbf{i}_{sep}$ are the indices of separator nodes through which

Algorithm 1 $(\widehat{\mathcal{G}}, \widehat{\mathcal{C}}, \widetilde{\mathbf{e}})$ = SparseRC($\mathcal{G}, \mathcal{C}$, **e**, do_minfill)

Given: original $\mathcal{G}, \mathcal{C}$, original vector of terminal indices **e**, do_minfill (0/1) option for minimum-fill reordering per subnet
Output: reduced $\widehat{\mathcal{G}}, \widehat{\mathcal{C}}$, updated vector of terminal indices $\widetilde{\mathbf{e}}$
1: Let graph $G := \text{nzp}(\mathcal{G} + \mathcal{C})$
2: (P, Sep) =partition(G, N)
3: $\mathcal{G} = \mathcal{G}(P, P), \mathcal{C} = \mathcal{C}(P, P), \mathbf{e} = \mathbf{e}(P)$ ▷ reorder in BBD
4: **for** component $C_k = 1 \ldots N$ **do**
5: **if** $C_k \notin Sep$ **then** ▷ C_k is not a separator
6: $\mathbf{i}_k$ = index of nodes for component C_k
7: $\mathbf{i}_{sep}$ = index of separator nodes connecting C_k to components $C_{k+1} \ldots C_N$
8: $(\mathcal{G}, \mathcal{C}, \mathbf{e})$ = EMMP($\mathcal{G}, \mathcal{C}, \mathbf{i}_k, \mathbf{i}_{sep}$, **e**, do_minfill) ▷ reduce C_k with EMMP
9: **else** keep separator component C_k
10: **end if**
11: **end for**
12: $\widehat{\mathcal{G}} = \mathcal{G}, \widehat{\mathcal{C}} = \mathcal{C}, \widetilde{\mathbf{e}} = \mathbf{e}$

Algorithm 2 $(\widehat{\mathcal{G}}, \widehat{\mathcal{C}}, \widetilde{\mathbf{e}})$ = EMMP($\mathcal{G}$, $\mathcal{C}$, $\mathbf{i}_k$, $\mathbf{i}_{sep}$, **e**, do_minfill)

Given: initial $\mathcal{G}, \mathcal{C}$, corresponding vector of terminal indices **e**, do_minfill (0/1) option for minimum-fill node reordering
Output: reduced $\widehat{\mathcal{G}}, \widehat{\mathcal{C}}$, corresponding vector of terminals $\widetilde{\mathbf{e}}$
1: **if** do_minfill **then** ▷ find additional internal nodes to preserve
2: $(\mathbf{i}_k, \mathbf{i}_{sep}, \mathbf{e})$ = reorderCAMD$(\mathcal{G}, \mathcal{C}, \mathbf{i}_k, \mathbf{i}_{sep}, \mathbf{e})$
3: ▷ find optimal minimum fill ordering per subnet
4: **end if**
5: $(\mathbf{i}_{int}, \mathbf{i}_{ext})$ = split$(\mathbf{i}_k, \mathbf{e})$ ▷ split $\mathbf{i}_k$ into internal and external nodes
6: $\mathbf{i}_R = \mathbf{i}_{int}$ ▷ internal nodes to eliminate
7: $\mathbf{i}_S = [\mathbf{i}_{ext}, \mathbf{i}_{sep}]$ ▷ selected nodes to keep
8: $\mathbf{G}_R = \mathcal{G}(\mathbf{g}i_R, \mathbf{i}_R), \mathbf{C}_R = \mathcal{C}(\mathbf{i}_R, \mathbf{i}_R)$
$\mathbf{G}_K = \mathcal{G}(\mathbf{i}_R, \mathbf{i}_S), \mathbf{C}_K = \mathcal{C}(\mathbf{i}_R, \mathbf{i}_S)$
$\mathbf{G}_S = \mathcal{G}(\mathbf{i}_S, \mathbf{i}_S), \mathbf{C}_S = \mathcal{C}(\mathbf{i}_S, \mathbf{i}_S)$
9: $\mathbf{W} = -\mathbf{G}_R^{-1}\mathbf{G}_K$ ▷ construct reducing projection
10: $\mathcal{G}(\mathbf{i}_S, \mathbf{i}_S) = \mathbf{G}_S - \mathbf{G}_K^T\mathbf{W}$ ▷ update entries for selected nodes
11: $\mathcal{C}(\mathbf{i}_S, \mathbf{i}_S) = \mathbf{C}_S + \mathbf{C}_K^T\mathbf{W} + \mathbf{W}^T\mathbf{C}_K + \mathbf{W}^T\mathbf{C}_R\mathbf{W}$
12: $\mathcal{G}(\mathbf{i}_R, \mathbf{i}_R) = \{\}, \mathcal{C}(\mathbf{i}_R, \mathbf{i}_R) = \{\}$ ▷ eliminate $\mathbf{i}_R$ nodes
$\mathcal{G}(\mathbf{i}_R, \mathbf{i}_S) = \{\}, \mathcal{C}(\mathbf{i}_R, \mathbf{i}_S) = \{\}, \quad \mathbf{e}(\mathbf{i}_R) = \{\}$
13: $\widehat{\mathcal{G}} = \mathcal{G}, \widehat{\mathcal{C}} = \mathcal{C}, \widetilde{\mathbf{e}} = \mathbf{e}$

$\mathbf{i}_k$ communicate with the rest of the circuit. If desired, at the entry of EMMP these nodes are reordered with CAMD, as to identify additional internal nodes which may further improve sparsity from reducing C_k (this operation, however, is only an advanced feature and often unnecessary). Internal and external nodes of component C_k are identified. Internal nodes $\mathbf{i}_R$ will be eliminated and selected nodes $\mathbf{i}_S$ will be preserved (i.e., terminals of C_k, corresponding separator nodes, and possibly some additional internal nodes obtained from step 2). The corresponding matrix blocks are identified and the update matrix $\mathbf{W}$ is formed. The blocks corresponding to selected nodes $\mathbf{i}_S$ are updated, while those corresponding to the eliminated $\mathbf{i}_R$ nodes are removed. At output, $\widehat{\mathcal{G}}$, $\widehat{\mathcal{C}}$ are reduced: internal nodes were eliminated *only* from the component defined by node indices $\mathbf{i}_k$; nodes corresponding to the other components are untouched. The terminals of the reduced model are indexed by $\widetilde{\mathbf{e}}$.

As a few computational remarks, to ensure numerical stability while forming the reduced matrix blocks at step 11 of Algorithm 2, we apply rescaling to $\mathcal{C}$ (and/or $\mathcal{G}$). Based on Theorem 1, it is also ensured that the global moment matching projection which underlies SparseRC inherits the proved [11] full-column-rank properties of the SIP/PACT projection.

1) *On the Partitioning Strategy:* It was shown how preserving internal nodes along with terminals improves the sparsity of the reduced model. Good reduced models are *sparse and small*, i.e., have minimum fill and few preserved internal nodes. Toward achieving a suitable tradeoff between sparsity and dimension, one may naturally ask: a) what are the optimal partitioning criteria and the number of generated subnets N, and b) when are additional fill-reducing node reorderings and fill-monitoring actions needed aside from partitioning?

a) *Choice of N:* Through partitioning, the aim is to minimize the communication among subnets (via a small number of separator nodes) *and* spread the terminals across partitions, as to minimize the fill-in generated from reducing each partition (up to its terminals) and inside the separator blocks. Toward achieving this goal, this paper relies on the general-purpose partitioner nested dissection (NESDIS, part of [25]) the choice however is by no means exclusive. In [26], for instance, the usage of the hypergraph partitioner Mondriaan [27] is documented. NESDIS partitions a graph so that communication among subnets is minimized, however it cannot control explicitly the distribution of terminals across parts. With NESDIS, terminals get spread indirectly, as a consequence of partitioning. While estimating an optimal N is an open problem which must simultaneously account for multiple factors (number of terminals, internal nodes, elements, potential fill-in), we provide some guidelines as to quickly determine a satisfactory value to be used with NESDIS. For our experiments, N was mostly determined immediately by inspecting the ratio of terminals to internal nodes in the original graph, $\frac{p}{n}$. For netlists with small $\frac{p}{n}$ [for instance, $\frac{p}{n} < \frac{1}{10}$], a coarse partitioning is already sufficient to achieve few terminals per subnet and ensure sparsity (certainly, as long as the resulting number of nodes per subnet enables the computation of the corresponding block projection). Such circuits are the ideal candidates for SparseRC based on NESDIS partitioning alone, without extra fill-reducing ordering actions.

b) *Additional Fill-Monitoring Actions:* For circuits with large $\frac{p}{n}$ ratios though, finer NESDIS partitions are needed to achieve a small number of terminals per subnet (see the *Filter* net in Section IV-B). Also, additional fill-reducing orderings and minimum-fill tracking actions may further improve the sparsity attained from partitioning, at the cost of preserving more internal nodes. In Section IV examples illustrate the sparsity, dimensionality, and computational implications of the partitioning fineness and, where needed, of fill-monitoring actions.

c) *Clarifying Remarks:* The functionality of SparseRC is not tied strictly to the partitioner used or the choice of N. It is assumed that the original circuits (graphs) are sparse. The sparsity of the original circuit will dictate the partitioning performance and consequently the dimension and sparsity level of the reduced circuit. Precise judgements on the optimal N or the necessity of additional fill-monitoring operations cannot be made *a priori*. These could be resolved by the following multiterminal graph optimization problem [26]; for a multiterminal network $G = (V, E)$, of which $P \subset V$ are terminals, $|P| = p$, find an N-way partitioning with the objective of minimizing the number of separator nodes

subject to the following constraints: a) the separator nodes and terminals are balanced across the N parts, and b) eliminating the internal nodes from each subnet introduces minimum fill in the parts determined by terminals and separator nodes. Solving this problem is a separate research topic and subject to ongoing research. As concerns the approximation quality, the SparseRC model will always be at least as good as a PACT [14] reduced model. Due to Theorem 1, SparseRC guarantees not only local but also global moment matching for the recombined network, irrespective of N or the partitioner used.

2) *Computational Complexity:* The computational complexity of SparseRC is dominated by the cost of computing $\mathbf{W}$ inside EMMP (line 9 in Algorithm 2), for each of the $N_{\max} < N$ nonseparator components. With $n_{\max}$ denoting the maximum number of internal nodes for a component (i.e., the maximum size of block $\mathbf{G}_R$), and $m_{\max}$ the maximum size of $\mathbf{G}_S$, the cost of one EMMP operation is at most $\mathrm{O}(n_{\max}^{\alpha} m_{\max})$, with $1 < \alpha \leq 2$ [28]. When n and p are large and the circuit is partitioned, one aims at $n_{\max} \ll n$ and $m_{\max} \ll p$ [note that $m_{\max} = p_{\max} + s_{\max}$, with $p_{\max}$ denoting the maximum number of terminals per component (i.e., length of $\mathbf{i}_{ext}$) and $s_{\max}$ the maximum number of separator nodes connecting a component C_k to components $C_{k+1} \ldots C_N$ (i.e., length of $\mathbf{i}_{sep}$)]. Therefore, especially for netlists with many internal nodes n and many terminals p, the total cost $\mathrm{O}(N_{\max}(n_{\max}^{\alpha} m_{\max}))$ of SparseRC is much smaller than the $\mathrm{O}(n^{\alpha} p)$ cost of constructing (if at all feasible) a PACT reducing projection directly from the original, unpartitioned matrices. The results in Section IV-A, Table I confirm this through netlists 6.*RX* and 7.*PLL* which contain more than 300 000 internal nodes and 4000 terminals. For a graph $G = (V, E)$ with $|V| = n + p$ nodes and $|E|$ edges, the cost of NESDIS partitioning (being based on METIS [29]) is $O(|E|)$ hence cheap to perform for sparse graphs (here, $|V|$ is the number of circuit nodes and $|E|$ the number of resistor and capacitors). Should additional reorderings be employed per subnet, the cost of CAMD is $O(|V||E|)$ [19], [20], [30], and also a fast operation.

The cost for the advanced option of tracking fill-in is more expensive, especially for networks with nodes and elements exceeding thousands. The operation becomes a partial Gaussian elimination up to terminals, which may reach $O(|V|^3)$ in the worst case. Nonetheless, such fill-monitoring operations are only an optional, advanced feature of SparseRC which was rarely needed in our experiments.

With ingredients of SparseRC in hand, we summarize its properties in light of the reduction criteria defined in Section II-A. SparseRC meets the accuracy 1), passivity 2), and terminal reconnectivity 4) requirements while reducing multiterminal RC circuits via a block-wise EMMP reducing projection. The efficiency 5) of SparseRC is ensured via a partitioning-based implementation, which reduces much smaller subnets (also with fewer terminals) individually while maintaining the accuracy, passivity, and reconnectivity requirements of the entire circuit. The sparsity 3) of the SparseRC reduced model is enhanced by preserving a subset of internal nodes which are identified automatically from partitioning and where necessary, from additional fill-reducing orderings. The performance of SparseRC in practice is shown by the numerical and simulations results presented next.

IV. Numerical Results and Circuit Simulations

Several parasitic extracted RC circuits from industrial applications are reduced with SparseRC. For each circuit, the terminals are nodes to which nonlinear devices (such as diodes or transistors) are connected. During a parsing phase, the multiterminal RC circuit is stamped into the MNA form (11) and reduced with SparseRC. The reduced model (30), (31) is synthesized with RLCSYN [17] into its netlist description. As connectivity via the external nodes is preserved with SparseRC, no voltage/current controlled sources are generated during synthesis. The nonlinear devices are directly recoupled via the terminal nodes to the reduced parasitics. The reduced circuit is resimulated with Spectre and its performance is compared to the original simulation.

Most of the circuits are reduced with the default options of SparseRC; matching only the 0th and 1st order multiport admittance moments at $s = 0$ and also *without* employing additional fill-tracking options. For some examples, the functionality of advanced options within SparseRC is demonstrated, such as: additional moment or pole matching, or additional fill-monitoring actions per subnet.

SparseRC was implemented in MATLAB (version R2007a); all reduction experiments were performed on a Linux (Ubuntu) machine with 3.9 GB RAM main memory and two Intel(R) Core(TM)2 Duo, 2.4 GHz, CPUs. All Spectre simulations were run on a Linux (Redhat) machine with 16 GB RAM main memory and six Intel(R) Xeon(R) X5460, 3.1 GHz, CPUs.

A. General Results

Table I collects the main SparseRC reduction results for various multiterminal netlists obtained from real chip designs, and compares the results obtained with PACT.[5] Each block row consists of a netlist example with the corresponding number of terminals p (the same before and after reduction). For each netlist, the sparsity before and after reduction are recorded here as the number of circuit elements, i.e., $\#R$, $\#C$. The reduction rate (Red. Rate) shows the percentage reduction for the corresponding column quantity. For instance, the percentage reduction in internal nodes n_i is $P_n = \frac{100(n_{i_{\text{Orig.}}} - n_{i_{\text{SpRC}}})}{n_{i_{\text{Orig.}}}}$, similarly for $\#R$, $\#C$. The Red. Rate in simulation time is computed as a speed-up factor: $\frac{\text{Sim. Time}_{\text{Orig}}}{\text{Sim. Time}_{\text{SpRC}}}$ (similarly for PACT). The approximation error is measured as the root-mean-square (RMS) value of the difference between the signals of the original and the reduced circuit. Several simulations are performed, including: AC, noise, transient, (quasi) periodic steady state, and (quasi) s-parameter, usually a combination of these for each circuit depending on the underlying application. Due to space limitations, we can only present some of the simulation figures, however, the simulation timings recorded in the tables represent the sum of all analysis types performed for one circuit. The error RMS value is recorded for one analysis only, but is representative of all analysis types performed for that circuit.

[5]For simplicity, "PACT" is used here only as a short term to denote SparseRC reduction without partitioning; the full PACT methodology [14] also includes more advanced analysis such as pole matching.

TABLE I
REDUCTION FOR VARIOUS NETLISTS WITH SPARSERC VERSUS PACT (SPARSERC WITH NO PARTITIONING)

Net	Type	n_i	#R	#C	Sim. Time (s)	Total Red. Time (s)	Error (RMS)	Type	n_i	#R	#C	Sim. Time (s)	Total Red. Time (s)	Error (RMS)
1. Transmission line (TL) $p = 22$	Orig.	3231	5892	3065	0.51	–	–	Orig.	3231	5892	3065	0.51	–	–
	SparseRC	21	165	592	0.01	0.39	2.5*e*-4	PACT	0	40	231	0.01	0.1	6.1*e*-4
	Red. rate	99.35%	97.20%	80.69%	51 *X*	–		Red. rate	100%	99.32%	92.46%	51 *X*	–	
2. Low noise amplifier (*LNA*) $p = 79$	Orig.	29 806	53 285	12 025	1525	–		Orig.	29 806	53 285	12 025	1525	–	
	SpRC-dp	50	308	2608	13	53.4	4*e*-4	PACT	0	111	1080	9	0.92	1*e*-3
	Red. rate	99.8%	99.4%	78.3%	117 *X*	–		Red. rate	100%	99.79%	91.02%	169.4 *X*	–	
3. Mixer_3 (MX3) $p = 110$	Orig.	757	1393	2353	1900	–		Orig.	757	1393	2353	1900	–	
	SpRC-mf	40	137	810	754	1.09	1.6*e*-7	PACT	0	111	1117	831	0.03	9.4*e*-7
	Red. rate	94.72%	90.17%	65.58%	2.52 *X*	–		Red. rate	100%	92.03%	52.53%	2.28 *X*	–	
4. Interconnect structure (IS) $p = 646$	Orig.	16 216	26 413	173 277	1730	–		Orig.	16 216	26 413	173 277	1730	–	
	SpRC	208	10 228	72 748	18	3.49	3.5*e*-5	PACT	17	6844	66 758	17	3.56	5.6*e*-4
	Red. rate	98.72%	61.28%	58.02%	96.1 *X*	–		Red. rate	99.9%	74.09%	61.47%	101.7 *X*	–	
5. Mixer_7 (*MX7*) $p = 66$	Orig.	67	119	194	7.15	–		Orig.	67	119	194	7.15	–	
	SpRC-mf	11	59	187	5.51	0.32	1.2*e*-8	PACT	0	56	308	9.12	0.01	7.1*e*-8
	Red. rate	83.58%	50.42%	3.61%	1.3 *X*	–		Red. rate	100%	52.94%	–58.76%	0.78 *X*	–	
6. Receiver (*RX*) $p = 15171$	Orig.	788 081	1 416 454	1 961 224	NA	–		Orig.	788 081	1 416 454	1 961 224	NA	–	
	SpRC	6719	95 162	845 699	520	589.4	NA	PACT	NA	NA	NA	NA	NA	NA
	Red. rate	99.15%	93.28%	56.88%	∞	-		Red. rate	NA	NA	NA	NA	–	
7. Phase-locked loop (*PLL*) $p = 4041$	Orig.	377 433	593 786	555 553	NA	–		Orig.	377 433	593 786	555 553	NA	–	
	SpRC	3905	46 499	312 351	3710	151.54	NA	PACT	NA	NA	NA	NA	NA	NA
	Red. rate	98.97%	92.17%	43.78%	∞	–		Red. rate	NA	NA	NA	NA	–	
8. *Filter* $p = 5852$	Orig.	32 140	47 718	123 696	1140	–		Orig.	32 140	47 718	123 696	1140	–	
	SpRC	6882	31 995	155 011	700	185.06	1.4*e*-5	PACT	0	414 500	5 927 790	> 24 h	68.5	NA
	Red. rate	78.59%	32.95%	–25.32%	1.63 *X*	–		Red. rate	100%	–768.64%	–4692%	NA	–	

Sim. Time: Spectre [24] netlist simulation time; Total Red. Time: partitioning plus reduction time.

For most examples, excellent reduction rates (above 80%) in the number of internal nodes n_i were obtained. The number of internal nodes n_i in the reduced model represent the special internal nodes which, if otherwise eliminated, would have introduced too much fill-in. They are the separator nodes identified automatically from partitioning (plus, where suitable, some additional internal nodes identified from fill monitoring operations). With n_i thus preserved, very good reduction rates were obtained in the number of circuit elements: mostly above 60% reduction in resistors and above 50% for capacitors. The effect of reducing internal nodes as well as the number of circuit elements is revealed by significant speed-ups attained (mostly above $2X$) when simulating the reduced circuits instead of the original. Even more, for the largest examples (netlists *RX, PLL*) simulation was only possible after reduction, as the original simulations failed due to insufficient CPU and memory resources. In addition, the reduction times recorded in Table I show that these reduced netlists were obtained efficiently.

1) *Reduction Without Partitioning:* For comparison, results for SparseRC without partitioning (essentially, PACT) are also recorded in Table I. PACT amounts to running the SparseRC Algorithm 1 of Section III-D with $N = 1$ in line 2. The results reveal the strength of SparseRC *especially* when reducing challenging circuits with very large node and terminal numbers (e.g., nets 6–8). First, the computational advantages of partitioning for very large netlists are revealed through examples *PLL* and *RX*, for which an unpartitioned PACT projection could not even be computed. For the smaller examples, the PACT reduction times are smaller than the SparseRC ones, indicating that partitioning is not necessary. Second, partitioning and the preservation of separator nodes improves the sparsity of the reduced models. This is confirmed by the *Filter* and *MX7*, where the unpartitioned approach resulted in dense reduced netlists which were slower to simulate than the originals (the effect would be the same for *PLL* and *RX*).

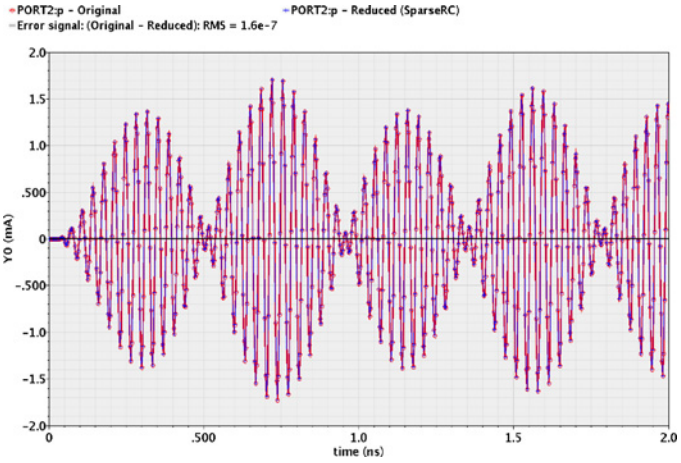

Fig. 4. *MX3*. Transient simulation of the original (red) and SparseRC (blue) overlap. The error signal (black) has an RMS value of 1.6*e*-7.

Next, selected simulation results are presented. The *MX3* net comes from a mixer circuit. Here, the SparseRC model was obtained by further reordering the partitioned circuit matrices (obtained via NESDIS) with CAMD and by keeping track of fill-in during the block-wise reduction process. The transient simulation in Fig. 4 shows that the original and SparseRC curves are indistinguishable.

2) *Improving Accuracy at Higher Frequencies:* Two examples in particular demonstrate possibilities for improving the approximation accuracy beyond matching the 0th and 1st moment at $s = 0$, as described in Section III-C1.

a) *LNA:* Two reduced models, SparseRC-dp and PACT, were computed for net *LNA* from Table I. SparseRC-dp

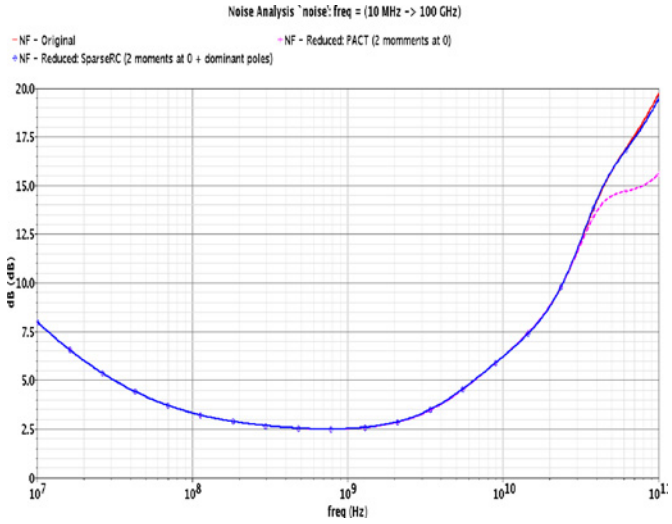

Fig. 5. *LNA*. Noise analysis comparison of original (red) versus two reduced models: in blue, SparseRC-dp (moment patching at $s = 0$ with additional dominant poles), and in pink, PACT with default moment matching at $s = 0$.

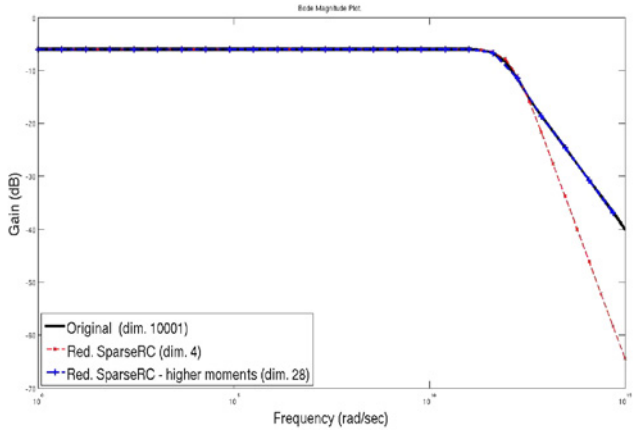

Fig. 6. RCtline. Bode plot for original and two reduced SparseRC models: by matching the first two admittance moments at $s = 0$ only (red) and by matching in addition higher moments from the $\mathbf{Y}'_R$ term per subnet (blue).

denotes a reduced model obtained from a partition into N=3 components where, for each component, the two default moments at s=0 are matched plus approximately eight dominant poles of the internal contribution $\mathbf{Y}'_R$. These dominant poles were computed with the subspace accelerated dominant pole method [3]. In Fig. 5, the noise simulations are shown for the original, SparseRC-dp, and PACT models. The effect of improving the SparseRC response with dominant poles is visible in Fig. 5. This was also confirmed in transient simulation, which gave an RMS error of $4{\cdot}10^{-4}$ for SparseRC-dp, smaller than $1{\cdot}10^{-3}$ for PACT (see Table I).

b) *Two Port RC TL:* A uniform RC TL with two terminals (one node at the beginning, and one node at the end of the line) is considered, with a cascade of 10 000 *R-C* sections. The circuit was partitioned into $N = 2$ subnets (and one separator block). Two reduced SparseRC models were computed: without and with additional matching at higher moments. The latter model was computed with the projection (40), which matched a combination of moments of the $\mathbf{Y}'_R$ term per subnet: two moments at 0, one at $s = 10^{10}$ and one at $s = 10^{14}$ were chosen. The Bode magnitude plot of

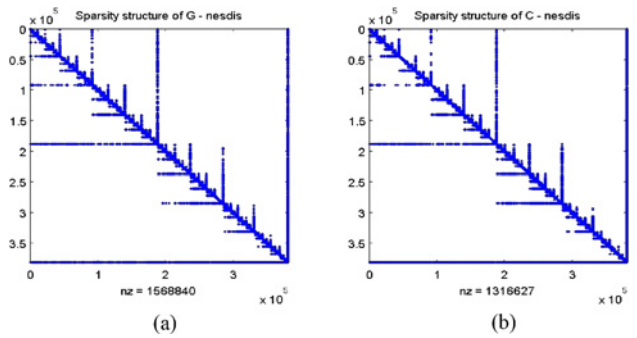

Fig. 7. *PLL*. Reordered (a) $\mathcal{G}$ and (b) $\mathcal{C}$ in BBD form after NESDIS partitioning (dimension $n + p = 381\,474$ nodes).

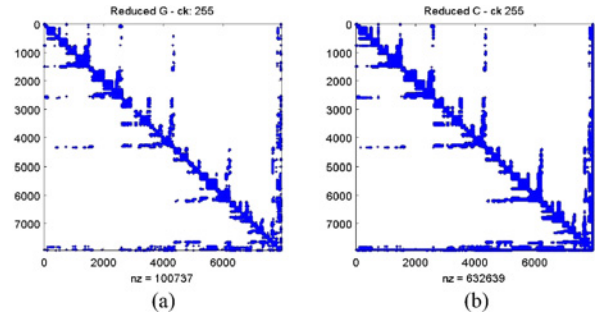

Fig. 8. *PLL*. Reduced (a) $\widehat{\mathcal{G}}$ and (b) $\widehat{\mathcal{C}}$ obtained with SparseRC (dimension $n + p = 7946$ nodes). The BBD structure is retained and the matrices remain sparse.

the frequency response for the original and the two reduced models is shown in Fig. 6. The behavior at higher frequencies is indeed approximated more accurately for the reduced model which preserves additional moments.

B. Advanced Comparisons

Nets *PLL* and *Filter*, two of the largest and most challenging netlists from Table I are analyzed in detail in Table II. The purpose of the analysis is threefold: 1) the advantages of SparseRC over existing methodologies are revealed; 2) the effects of various partitioning sizes and of additional reorderings are shown; and 3) possible limitations and improvement directions for SparseRC are identified.

1) *PLL:* The original simulation was for this circuit failed due to insufficient CPU and memory resources even on a larger machine, but reduction makes the simulation possible. The original $\mathcal{G}$ and $\mathcal{C}$ matrices, reordered and partitioned in BBD form are shown in Fig. 7. The borders are visible, collecting the separator nodes that will be preserved along with the terminals. The reduced matrices retain the BBD structure and are sparse, as seen in Fig. 8.

Two SparseRC reduced models were computed, SpRC$_c$ and SpRC$_f$, based on a coarse and fine NESDIS partitioning, respectively, with the relevant statistics shown in Table II. Both reduced models achieved excellent reduction rates in internal nodes and circuit elements, and were fast to simulate. After a coarse partitioning, the reduction time was smaller than after the fine partitioning due to less computational overhead in forming the reduced matrices per subnet. The SpRC$_f$ reduced model however was faster to simulate than SpRC$_c$, possibly due to the fact that, although larger in node numbers than SpRC$_c$, SpRC$_f$ has fewer circuit elements. Determining

TABLE II
ADVANCED COMPARISONS FOR VERY LARGE EXAMPLES

Net	Type	n_i	P_{n_i} (%)	#R	P_R (%)	#C	P_C (%)	Sim. Time (s)	Sim. Speed-Up (X)	Red. Time (s)	Partition Time (s)	N # Part.	AvgN Size	AvgS Size	Avg-p Size	Avg. Red. Time (s)
7. *PLL* $p = 4041$	Orig.	377 433	–	593 786	–	555 553	–	NA	–	–	–	–	–	–	–	–
	$SpRC_c$	3905	98.97	46 499	92.17	312 351	43.78	689	∞	116	36	255	2949	32	29	0.86
	$SpRC_f$	12 861	96.59	56 582	90.47	243 350	56.20	656	∞	1199	394	2423	304	11	3	0.93
	PartMOR	5392	98.57	138 628	76.6	209 835	62.22	836	∞	1410	559	229	–	–	–	–
	PACT	NA	NA	NA	NA	NA	NA	NA	NA	NA	–	–	–	–	–	–
	SIP-mf	NA	NA	NA	NA	NA	NA	NA	NA	> 24 h	–	–	–	–	–	–
8. *Filter* $p = 5852$	Orig.	32 140	–	47 718	–	123 696	–	1140	–	–	–	–	–	–	–	–
	SpRC	6882	78.59	31 995	32.95	155 011	−25.32	700	1.63 X	163	22.60	2065	29	8	4	0.15
	SpRC-mf	16 729	47.95	32 760	31.35	116 272	6.00	783	1.46 X	1237	19.44	2065	29	8	4	1.19
	PACT	0	100	414 500	−768.64	5 927 790	−4692	> 24 h	NA	69	–	–	–	–	–	–
	SIP-mf	30 935	3.75	46 399	2.76	123 135	0.45	892	1.28 X	11 648	–	–	–	–	–	–

N is the number of subnets, AvgN Size is the average size of a subnet, AvgS Size/Avg.–p Size are the average number of separator nodes/terminals per subnet, and Avg. Red. Time is the average reduction time per subnet.

the appropriate balance between preserved internal nodes n_i and sparsity, and its influence on simulation time remains to be further studied. A direct PACT reduction is immediately dismissed, due to the prohibitive computational and density considerations. A SIP-based reduced model was attempted, but the fill-in monitoring actions were too expensive (> 24 h).

a) *Comparison with PartMOR:* For the *PLL*, statistics of a reduced PartMOR model are also included, thanks to the authors of [16]. Compared to PartMOR, $SpRC_c$ has fewer internal nodes and circuit elements. The SparseRC models were faster to simulate, and also obtained in a shorter partitioning and reduction time. Although there is no original simulation to compare the reductions against, Fig. 9 shows the AC simulation waveform for the two SparseRC models, and PartMOR. $SpRC_c$ and $SpRC_f$ overlap, confirming that the accuracy of SparseRC is robust to changes in the partition strategy, due to guaranteed local *and* global moment matching. The reduced PartMOR model was determined by local matching of the 0th and 1st moments at DC, however with no guarantee of matched moments for the recombined network (personal communication with PartMOR [16] authors, March 14, 2011). Thus, since the accuracy of PartMOR is dependent on the number of partitions, finer partitioning would likely be needed to reach what we infer are the correct waveforms produced by SparseRC. While PartMOR offers advantages in other respects, such as guaranteed positive elements, these results motivate a more thorough investigation into combining the strengths of both methods in the future.

2) *Filter:* This netlist is more challenging due to its large ratio $\frac{p}{n_i} > 10^{-1}$. Two SparseRC reduced models were computed: SpRC, based on NESDIS partitioning alone, and SpRC-mf, where, after partitioning, each subnet was reordered with CAMD and additional internal nodes were preserved via fill-in monitoring operations. In both cases, a fine partitioning was needed to distribute the terminals across subnets. Although the SpRC-mf has much fewer circuit elements than SpRC, it takes longer to simulate, due to the presence of many internal nodes n_i. The fill-monitoring operations inside SpRC-mf also make the reduction time significantly longer than for SpRC. The PACT reduced model is the smallest in dimension (has no preserved internal nodes) but extremely dense and useless in simulation. An SIP reduced model was also computed:

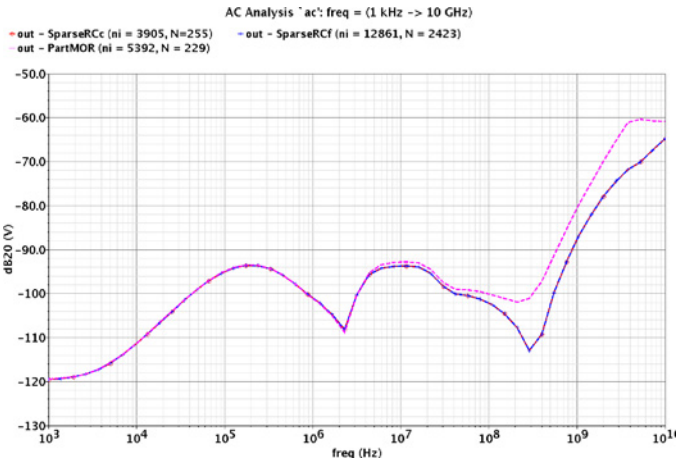

Fig. 9. *PLL*. AC analysis of reduced models: $SpRC_c$ (red) and $SpRC_f$ (blue) are overlapping as expected. PartMOR (magenta) deviates slightly.

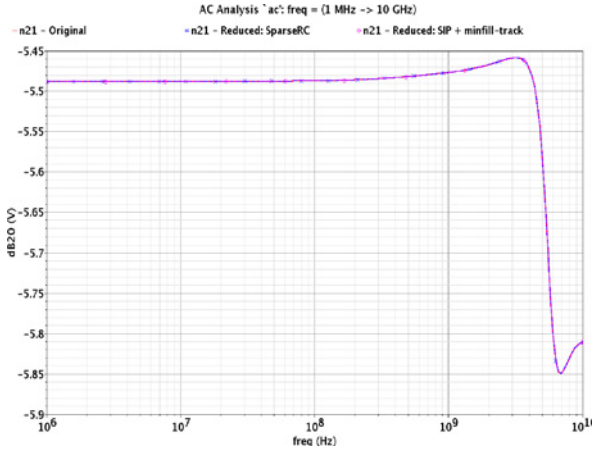

Fig. 10. *Filter*. AC analysis of original (red), reduced SparseRC model (blue), and reduced SIP model with minimum fill-track (magenta) match perfectly.

CAMD reordering and fill-monitoring actions were applied on the original netlist's graph to determine the reduced model with minimum fill. The result is shown in Fig. 11, where the minimum fill point is identified after eliminating only the first 1200 internal nodes. The SIP reduced circuit is larger, has more elements than SparseRC, and is slower to simulate. Also, the SIP reduction time was much larger than SparseRC. The AC analysis comparison of SparseRC and SIP match with

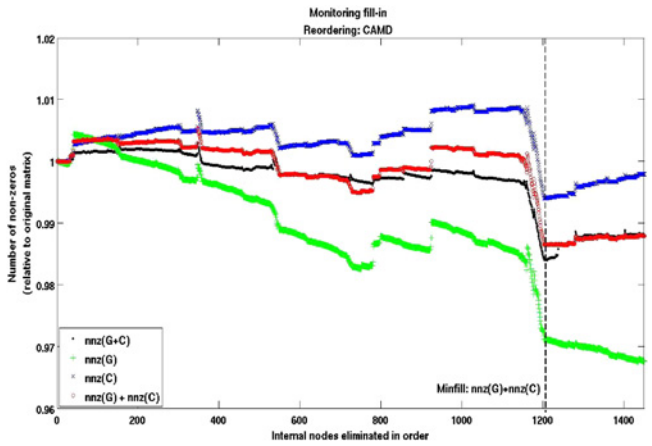

Fig. 11. *Filter*. Determining the dimension of the SIP reduced model from CAMD reordering and node-wise elimination. The minimum fill point is reached after eliminating the first 1200 of the 32 140 internal nodes (for clarity only the first 1400 internal nodes are shown).

the original, is shown in Fig. 10. This further strengthens the advantages of partitioning: aside from enhancing sparsity by preserving separator nodes, it also makes fill-monitoring actions cheaper to perform. In summary, SparseRC achieves the best tradeoff in terms of accuracy, dimension, sparsity, reduction time, and resimulation time. Finally, the *Filter* example reveals several directions for improving the SparseRC methodology. It indicates that other reduction transformations and/or further sparsification methods may be appropriate for circuits with many more capacitors than resistors. Also, circuits with large $\frac{p}{n}$ ratios are the most difficult to partition and reduce with a satisfactory sparsity level. This could be resolved with the help of partitioners which could directly control the distribution of terminals, and remains for further research [26].

V. Conclusion

In this paper, SparseRC was presented, which is a robust and efficient reduction strategy for RC circuits with many terminals. It reduced testcases where traditional model reduction fails, due to either the large dimension of the original problem or the density of the final reduced circuit. Developed on the divide and conquer principle, SparseRC used graph-partitioning to separate a network into minimally connected components. These were reduced individually with an appropriate update of the interconnections among them. This guaranteed that two multiport admittance moments were matched for the entire net irrespective of the partitioning size or strategy. SparseRC reduced circuits contained fewer nodes and circuit elements compared to conventional MOR results. As challenging industrial testcases showed, significant speed-ups were obtained when resimulating SparseRC reduced models in place of the original circuits.

Appendix A
Proofs of Theorems

Proof of Proposition 1: Expressing $\mathbf{x}_R$ in terms of $\mathbf{x}_S$ from the first block-row of (2), and replacing it in the second gives the circuit's multiport admittance: $\mathbf{Y}(s)\mathbf{x}_S = \mathbf{B}_S\mathbf{u}$, where $\mathbf{Y}(s) = (\mathbf{G}_S + s\mathbf{C}_S) - (\mathbf{G}_K + s\mathbf{C}_K)^T(\mathbf{G}_R + s\mathbf{C}_R)^{-1}(\mathbf{G}_K + s\mathbf{C}_K)$. The first moment of $\mathbf{Y}(s)$ at $s = 0$ is: $\mathbf{Y}(s)|_{s=0} = \mathbf{G}_S - \mathbf{G}_K^T\mathbf{G}_R^{-1}\mathbf{G}_K = \mathbf{G}_S'$. From the first derivative: $\frac{d\mathbf{Y}}{ds}(s) = \mathbf{C}_S - \mathbf{C}_K^T(\mathbf{G}_R + s\mathbf{C}_R)^{-1}(\mathbf{G}_K + s\mathbf{C}_K) - (\mathbf{G}_K + s\mathbf{C}_K)^T(\mathbf{G}_R + s\mathbf{C}_R)^{-1}\mathbf{C}_K + (\mathbf{G}_K + s\mathbf{C}_K)^T(\mathbf{G}_R + s\mathbf{C}_R)^{-1}\mathbf{C}_R(\mathbf{G}_R + s\mathbf{C}_R)^{-1}(\mathbf{G}_K + s\mathbf{C}_K)$, the second moment at $s = 0$ is obtained: $\frac{d\mathbf{Y}}{ds}(s)|_{s=0} = \mathbf{C}_S - \mathbf{C}_K^T\mathbf{G}_R^{-1}\mathbf{G}_K - \mathbf{G}_K^T\mathbf{G}_R^{-1}\mathbf{C}_K + \mathbf{G}_K^T\mathbf{G}_R^{-1}\mathbf{C}_R\mathbf{G}_R^{-1}\mathbf{G}_K = \mathbf{C}_S'$. ■

Proof of Theorem 1: By Proposition 1, the reduced subnet 1 defined by $\widehat{\mathcal{G}}_{11}$ (19) and $\widehat{\mathcal{C}}_{11}$ (22), preserves the first two multiport admittance moments at $s = 0$ of the original subnet 1 defined by $\mathcal{G}_{11}$, $\mathcal{C}_{11}$. The same holds for the reduced subnet 2, defined by $\widehat{\mathcal{G}}_{22}$ (32) and $\widehat{\mathcal{C}}_{22}$ (35). It remains to prove the admittance moment matching between the original (14) and reduced (30), (31) recombined circuits. This is shown by reconstructing from the individual EMMPs, an EMMP projection $\mathcal{V}$ associated with entire circuit (14), as follows. Recall $\mathcal{G}$ from (15), where in addition nodes $\mathbf{x}_2$ of the second subnet are split into $\mathbf{x}_{2_R}$ and $\mathbf{x}_{2_S}$ as in Section III-B2 as follows:

$$\mathcal{G} = \begin{bmatrix} \mathcal{G}_{11_R} & \mathcal{G}_{11_K} & \mathbf{0} & \mathbf{0} & \mathcal{G}_{13_R} \\ \mathcal{G}_{11_K}^T & \mathcal{G}_{11_S} & \mathbf{0} & \mathbf{0} & \mathcal{G}_{13_S} \\ \mathbf{0} & \mathbf{0} & \mathcal{G}_{22_R} & \mathcal{G}_{22_K} & \mathcal{G}_{23_R} \\ \mathbf{0} & \mathbf{0} & \mathcal{G}_{22_K}^T & \mathcal{G}_{22_S} & \mathcal{G}_{23_S} \\ \mathcal{G}_{13_R}^T & \mathcal{G}_{13_S}^T & \mathcal{G}_{23_R}^T & \mathcal{G}_{23_S}^T & \mathcal{G}_{33} \end{bmatrix}.$$

Recall $\mathcal{V} = \mathbf{V}_1\mathbf{V}_2 =$ with $\mathbf{V}_1$ from (16) and $\mathbf{V}_2$ from (29). Inside $\mathbf{V}_1$, let $\mathbf{I}_2 = \text{blockdiag}(\mathbf{I}_{R_2}, \mathbf{I}_{S_2})$ be partitioned according to the splitting of $\mathbf{x}_2$ into $\mathbf{x}_{2_R}$ and $\mathbf{x}_{2_S}$, respectively. Then, by straightforward matrix multiplication

$$\mathcal{V} = \mathbf{V}_1\mathbf{V}_2 = \begin{bmatrix} -\mathcal{G}_{11_R}^{-1}\mathcal{G}_{11_K} & \mathbf{0} & -\mathcal{G}_{11_R}^{-1}\mathcal{G}_{13_R} \\ \mathbf{I}_{S_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathcal{G}_{22_R}^{-1}\mathcal{G}_{22_K} & -\mathcal{G}_{22_R}^{-1}\mathcal{G}_{23_R} \\ \mathbf{0} & \mathbf{I}_{S_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{bmatrix}. \quad (41)$$

Let $\mathcal{P}$ be the permutation which interchanges the second with the third block-row of (41). Then, denoting $\mathcal{V}_\mathcal{P} = \mathcal{P}\mathcal{V}$

$$\mathcal{V}_\mathcal{P} = \begin{bmatrix} -\mathcal{G}_{11_R}^{-1}\mathcal{G}_{11_K} & \mathbf{0} & -\mathcal{G}_{11_R}^{-1}\mathcal{G}_{13_R} \\ \mathbf{0} & -\mathcal{G}_{22_R}^{-1}\mathcal{G}_{22_K} & -\mathcal{G}_{22_R}^{-1}\mathcal{G}_{23_R} \\ \mathbf{I}_{S_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{S_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{bmatrix}. \quad (42)$$

Define the permuted matrices $\mathcal{G}_\mathcal{P} = \mathcal{P}\mathcal{G}\mathcal{P}^T$, $\mathcal{C}_\mathcal{P} = \mathcal{P}\mathcal{C}\mathcal{P}^T$, $\mathcal{B}_\mathcal{P} = \mathcal{P}\mathcal{B}$, and notice their structure as follows:

$$\mathcal{G}_\mathcal{P} = \left[\begin{array}{cc|ccc} \mathcal{G}_{11_R} & \mathbf{0} & \mathcal{G}_{11_K} & \mathbf{0} & \mathcal{G}_{13_R} \\ \mathbf{0} & \mathcal{G}_{22_R} & \mathbf{0} & \mathcal{G}_{22_K} & \mathcal{G}_{23_R} \\ \hline \mathcal{G}_{11_K}^T & \mathbf{0} & \mathcal{G}_{11_S} & \mathbf{0} & \mathcal{G}_{13_S} \\ \mathbf{0} & \mathcal{G}_{22_K}^T & \mathbf{0} & \mathcal{G}_{22_S} & \mathcal{G}_{23_S} \\ \mathcal{G}_{13_R}^T & \mathcal{G}_{13_S}^T & \mathcal{G}_{23_R}^T & \mathcal{G}_{23_S}^T & \mathcal{G}_{33} \end{array}\right] := \left[\begin{array}{c|c} \mathbf{G}_R & \mathbf{G}_K \\ \hline \mathbf{G}_K^T & \mathbf{G}_S \end{array}\right] \quad (43)$$

$$\text{similarly } \mathcal{C}_\mathcal{P} := \left[\begin{array}{c|c} \mathbf{C}_R & \mathbf{C}_K \\ \hline \mathbf{C}_K^T & \mathbf{C}_S \end{array}\right], \quad \mathcal{B}_\mathcal{P} = \left[\begin{array}{c} \mathbf{0} \\ \mathbf{0} \\ \hline \mathcal{B}_{1_S} \\ \mathcal{B}_{2_S} \\ \mathcal{B}_{3_S} \end{array}\right] := \left[\begin{array}{c} \mathbf{0} \\ \hline \mathbf{B}_S \end{array}\right]. \quad (44)$$

From (43) and (44), and the analogy with Section II-B, one recognizes immediately in (42) the EMMP: $\mathcal{V}_{\mathcal{P}}=\left[\frac{-\mathbf{G}_R^{-1}\mathbf{G}_K}{\mathbf{I}_{S_1S_23}}\right]$, where $\mathbf{I}_{S_1S_23}$ = blockdiag($\mathbf{I}_{S_1}, \mathbf{I}_{S_2}, \mathbf{I}_3$). From Proposition 1, the reduced model obtained by projecting (43) and (44) with $\mathcal{V}_P$ matches the fist two DC multiport admittance moments (defined with respect to $\mathbf{B}_S$, the total number of terminals and separator nodes of the recombined circuit). Since $\mathcal{P}$ is a permutation, $\mathcal{P}^T\mathcal{P} = \mathbf{I}$, it follows that this reduced model is precisely: $\mathcal{V}_{\mathcal{P}}^T\mathcal{G}_{\mathcal{P}}\mathcal{V}_{\mathcal{P}} = \mathcal{V}^T\mathcal{G}\mathcal{V} = \widehat{\mathcal{G}}$, $\mathcal{V}_{\mathcal{P}}^T\mathcal{C}_{\mathcal{P}}\mathcal{V}_{\mathcal{P}} = \mathcal{V}^T\mathcal{C}\mathcal{V} = \widehat{\mathcal{C}}$, $\mathcal{V}_{\mathcal{P}}^T\mathcal{B}_{\mathcal{P}} = \mathcal{V}^T\mathcal{B} = \widehat{\mathcal{B}}$, where (30)–(39) hold. ■

ACKNOWLEDGMENT

The authors would like to thank P. Miettinen, M. Honkala, and J. Roos, the authors of PartMOR [16], for a fruitful knowledge exchange and for performing experiments on request. The authors would also like to thank G. de Jong, D. Jeurissen, M. Hanssen, J. Gobbels, H. van Walderveen, and T. Beelen from NXP Semiconductors, Eindhoven, The Netherlands, for providing the parasitic netlists, assessing the reduction results, and giving useful insights, Prof. T. Davis from the University of Florida, Gainesville, FL, and Prof. T. Antoulas from Rice University, Houston, TX, and Jacobs University Bremen, Bremen, Germany, for insightful discussions, and the anonymous referees and the Associate Editor, whose valuable comments helped to improve this paper.

REFERENCES

[1] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. Philadelphia, PA: SIAM, 2005.
[2] W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, vol. 13. New York: Springer, 2008.
[3] J. Rommes and N. Martins, "Efficient computation of multivariable transfer function dominant poles using subspace acceleration," *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1471–1483, Nov. 2006.
[4] T. Reis and T. Stykel, "PABTEC: Passivity-preserving balanced truncation for electrical circuits," *IEEE Trans. Comput.-Aided Des.*, vol. 29, no. 9, pp. 1354–1367, Sep. 2010.
[5] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 17, no. 8, pp. 645–654, Aug. 1998.
[6] R. Freund, "Sprim: Structure-preserving reduced-order interconnect macromodeling," in *Proc. IEEE/ACM ICCAD*, Nov. 2004, pp. 80–87.
[7] R. Ionutiu, J. Rommes, and A. Antoulas, "Passivity-preserving model reduction using dominant spectral-zero interpolation," *IEEE Trans. Comput.-Aided Des.*, vol. 27, no. 12, pp. 2250–2263, Dec. 2008.
[8] P. Feldmann and F. Liu, "Sparse and efficient reduced order modeling of linear subcircuits with large number of terminals," in *Proc. IEEE/ACM ICCAD*, Nov. 2004, pp. 88–92.
[9] H. Yu, L. He, and S. Tan, "Block structure preserving model order reduction," in *Proc. IEEE Int. BMAS*, Sep. 2005, pp. 1–6.
[10] J. Rommes and W. H. A. Schilders, "Efficient methods for large resistor networks," *IEEE Trans. Comput.-Aided Des.*, vol. 29, no. 1, pp. 28–39, Jan. 2010.
[11] Z. Ye, D. Vasilyev, Z. Zhu, and J. Phillips, "Sparse implicit projection (SIP) for reduction of general many-terminal networks," in *Proc. IEEE/ACM ICCAD*, Nov. 2008, pp. 736–743.
[12] B. Sheehan, "Realizable reduction of RC networks," *IEEE Trans. Comput.-Aided Des.*, vol. 26, no. 8, pp. 1393–1407, Aug. 2007.
[13] P. Elias and N. van der Meijs, "Extracting circuit models for large RC interconnections that are accurate up to a predefined signal frequency," in *Proc. 33rd Des. Automat. Conf.*, Jun. 1996, pp. 764–769.
[14] K. Kerns and A. Yang, "Stable and efficient reduction of large, multiport RC networks by pole analysis via congruence transformations," *IEEE Trans. Comput.-Aided Des.*, vol. 16, no. 7, pp. 734–744, Jul. 1997.
[15] K. Kerns and A. Yang, "Preservation of passivity during RLC network reduction via split congruence transformations," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 17, no. 7, pp. 582–591, Jul. 1998.
[16] P. Miettinen, M. Honkala, J. Roos, and M. Valtonen, "PartMOR: Partitioning-based realizable model-order reduction method for RLC circuits," *IEEE Trans. Comput.-Aided Des.*, vol. 30, no. 3, pp. 374–387, Mar. 2011.
[17] F. Yang, X. Zeng, Y. Su, and D. Zhou, "RLC equivalent circuit synthesis method for structure-preserved reduced-order model of interconnect in VLSI," *Commun. Comput. Phys.*, vol. 3, no. 2, pp. 376–396, 2008.
[18] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
[19] P. Amestoy, T. Davis, and I. Duff, "An approximate minimum degree ordering algorithm," *SIAM J. Matrix Anal. Applicat.*, vol. 17, no. 4, pp. 886–905, 1996.
[20] P. Amestoy, T. Davis, and I. Duff, "Algorithm 837: AMD, an approximate minimum degree ordering algorithm," *ACM Trans. Math. Softw.*, vol. 30, no. 3, pp. 381–388, 2004.
[21] A. Zecevic and D. Siljak, "Balanced decompositions of sparse systems for multilevel parallel processing," *IEEE Trans. Circuits Syst. I: Fundamental Theory Applicat.*, vol. 41, no. 3, pp. 220–233, Mar. 1994.
[22] Mathworks, Inc. *MATLAB, Version R2007a* [Online]. Available: http://www.mathworks.com
[23] M. Ugryumova, "Applications of model order reduction for IC modeling," Ph.D. dissertation, Dept. Math. Comput. Sci., Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2011.
[24] Cadence. *Spectre, Version 7.1.1.071* [Online]. Available: http://www.cadence.com
[25] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, "Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/downdate," *ACM Trans. Math. Softw.*, vol. 35, no. 3, pp. 1–14, 2008.
[26] R. Ionutiu, J. Rommes, and W. H. A. Schilders, "Graph partitioning with separation of terminals," in *Proc. 5th SIAM Workshop Combinatorial Sci. Comput.*, May 2011, pp. 39–41.
[27] A. N. Yzelman and R. H. Bisseling, "Cache-oblivious sparse matrix-vector multiplication by using sparse matrix partitioning methods," *SIAM J. Sci. Comput.*, vol. 31, no. 5, pp. 3128–3154, 2009.
[28] J. Phillips and L. Silveira, "Poor man's TBR: A simple model reduction scheme," *IEEE Trans. Comput.-Aided Des.*, vol. 24, no. 1, pp. 43–55, Jan. 2005.
[29] G. Karypis and V. Kumar, "Multilevel k-way partitioning scheme for irregular graphs," *J. Parallel Distrib. Comput.*, vol. 48, no. 1, pp. 96–129, 1998.
[30] S. Ingram. (2011). *Minimum Degree Reordering Algorithms: A Tutorial*. Univ. British Columbia, Vancouver, BC, Canada [Online]. Available: http://www.cs.ubc.ca/~sfingram/cs517_final.pdf

Roxana Ionuţiu (A'07) received the M.S. degree in electrical and computer engineering from William Marsh Rice University, Houston, TX, in 2008.

This paper is part of her Ph.D. research on model reduction for multiterminal systems, conducted jointly at Jacobs University Bremen, Bremen, Germany, Eindhoven University of Technology, Eindhoven, The Netherlands, and NXP Semiconductors, Eindhoven.

Joost Rommes received two M.S. degrees in computational science and computer science, and the Ph.D. degree in mathematics from Utrecht University, Utrecht, The Netherlands, in 2002, 2003, and 2007, respectively.

He is currently a Researcher with NXP Semiconductors, Eindhoven, The Netherlands, and works on model reduction, specialized eigenvalue methods, and algorithms for problems related to circuit design and simulation.

Wil H. A. Schilders received the M.S. degree in mathematics from Radboud University, Nijmegen, The Netherlands, in 1978, and the Ph.D. degree from Trinity College Dublin, Dublin, Ireland, in 1980.

Since then, he has been working in the electronics industry, first with Philips Research Laboratories, Eindhoven, The Netherlands, and since 2006 with NXP Semiconductors, Eindhoven. Here he developed algorithms for simulating semiconductor devices, electronic circuits, organic light emitting diodes, and electromagnetic problems (TV tubes, interconnects, magnetic resonance imaging). Since 1999, he has been a part-time Professor of numerical mathematics for industry with Eindhoven University of Technology, Eindhoven. Recently, he became the Managing Director of the Platform for Mathematics in The Netherlands. He is currently the President of the European Consortium for Mathematics in Industry.