Comparison of Circuit Reduction Techniques for Power Network Noise Analysis

Jinwook Kim and Young Hwan Kim

Abstract—**The endless scaling down of the semiconductor process made the impact of the power network noise on the performance of the state-of-the-art chip a serious design problem. This paper compares the performances of two popular circuit reduction approaches used to improve the efficiency of power network noise analysis: moment matching-based model order reduction (MOR) and node elimination-based MOR. As the benchmarks, we chose PRIMA and R2Power as the matching-based MOR and the node elimination-based MOR. Experimental results indicate that the accuracy, efficiency, and memory requirement of both methods very strongly depend on the structure of the given circuit, i.e., numbers of the nodes and sources, and the number of moments to preserve for PRIMA. PRIMA has higher accuracy in general, while the error of R2Power is also in the acceptable range. On the other hand, PRIMA has the higher efficiency than R2Power, only when the numbers of nodes and sources are small enough. Otherwise, R2Power clearly outperforms PRIMA in efficiency. In the memory requirement, the memory size of PRIMA increases very quickly as the numbers of nodes, sources, and preserved moments increase.**

Index Terms—**Power network noise, model order reduction (MOR), PRIMA, R2Power**

I. INTRODUCTION

According to the aggressive scaling down of the semiconductor process, an effect of power network noise has become a crucial problem. The continuous scaling down of the process technology increases the parasitic resistance and current density. Moreover, supply voltage has continuously decreased for low power design. These make the effect of power network noise more severe.

Power network noise increases gate delays and then it may cause the timing failure of a semiconductor chip. Reference [1] reports that gate delay increases by 4% when even the supply voltage decreases by 1% at the 90 nm process technology and 0.9 V supply voltage. Moreover, it decreases the noise margin and may cause the functional failure of a semiconductor chip. Therefore, power integrity should be considered during the design and verification processes.

The power network usually contains tremendous number of the parasitic elements and nodes, and hence, general-purpose circuit simulators such as SPICE are not adequate for analyzing the power network or interconnect due to CPU speed and memory limitation.

Several model order reduction (MOR) techniques have been proposed to solve these problems, which reduce a large circuit to a small reduced-order model. MOR techniques can be classified into two types: moment matching-based MOR techniques and node elimination-based MOR techniques. Moment matching-based MOR techniques reduce a large circuit into a small macro-model by matching some of the circuit moments. Several moment matching-based MOR techniques have been proposed; Asymptotic Waveform Evaluation [2], Passive Reduced-order Interconnect Macromodeling Algorithm (PRIMA) [3], Pade via Lanczos [4], and so on. These MOR techniques preserve important circuit moments, and show relatively accurate results. However, they have high computational complexity and the re-

Manuscript received Aug. 23, 2009; revised Nov. 1, 2009.
Dep. ECE., Pohang University of Science and Technology Pohang, Korea
E-mail : detest@postech.ac.kr

duced model cannot be directly converted to a general linear RLC circuit for SPICE simulation.

Node elimination-based MOR techniques such as TICER [5], and R^{2}Power [6], reduce a large circuit through the process of node elimination. These methods reduce a circuit to a general linear RC circuit directly. Although these methods usually show relatively lower accurate results than moment matching-based methods, these methods have relatively low computational complexity.

Since the above two MOR approaches generate different reduced results, the corresponding power network noise verification flows will also be different. Moreover, because the two approaches have their own merits and demerits, the proper selection of an MOR technique is very important.

This paper evaluates and compares the performances of PRIMA and R^{2}Power for power network noise analysis, which are moment matching based and node elimination based, respectively. PRIMA, which projects a circuit equation onto the Krylov subspace, has been a widely used method, because it preserves the circuit passivity. And R^{2}Power is the most recent node elimination method.

The rest of this paper is organized as follows. Section II describes the details of two approaches, including the background to explain the properties of the approaches. Section III presents the experimental results of the performance comparison. Finally, Section IV concludes this paper.

II. BACKGROUND

1. Circuit equation

For the power network noise analysis, we typically represent a power network as a linear RC network, where R and C are parasitic resistance and capacitance components. Sources are external DC voltage sources and the internal current sources that represent the operations of switching logic gates. A general linear RC power network, which contains N independent sources, can be expressed by the following modified nodal analysis (MNA).

$$\mathbf{G}\mathbf{x}(t)+\mathbf{C}\frac{d\mathbf{x}(t)}{dt}=\mathbf{B}\mathbf{u}(t) \tag{1}$$

where $\mathbf{G}$ is an $n\times n$ conductance matrix, $\mathbf{C}$ is an $n\times n$ susceptance matrix, and $\mathbf{B}$ is an $n\times N$ input connectivity matrix. $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are an n-dimensional internal variable vector and an N-dimension input current vector, respectively. The matrix dimension n is the sum of the number of nodes and the number of branch current variables for voltage sources or inductors.

G and C matrices and variable vector $\mathbf{x}(t)$ can be expressed as shown in (2).

$$\mathbf{G}=\begin{bmatrix}\mathbf{N} & \mathbf{E}\\ -\mathbf{E}^T & \mathbf{0}\end{bmatrix},\ \mathbf{C}=\begin{bmatrix}\mathbf{Q} & \mathbf{0}\\ \mathbf{0} & \mathbf{H}\end{bmatrix},\ \text{and } \mathbf{x}(t)=\begin{bmatrix}\mathbf{v}(t)\\ \mathbf{i}(t)\end{bmatrix}$$
$$\text{where } \mathbf{N},\mathbf{Q}\in\Re^{m\times m},\ \mathbf{E}\in\Re^{(n-m)\times m},\mathbf{H}\in\Re^{m\times m},$$
$$\mathbf{v}(t)\in\Re^{m},\ \text{and } \mathbf{i}(t)\in\Re^{(n-m)} \tag{2}$$

where m is the number of nodes. **N**, **Q**, and **H** represent stamps for resistors, capacitors, and inductors, respectively. **E** matrix, composed of 0, or ±1, represents the KCL equations for branch current variables. $\mathbf{v}(t)$ and $\mathbf{i}(t)$ are vectors of internal voltage and current variables, respectively. The values of the variable vector $\mathbf{x}(t)$ are usually calculated using a numerical integration method such as the backward Euler, the trapezoidal and Gear's method.

2. PRIMA

PRIMA projects a variable vector $\mathbf{x}(t)$, given in (1), onto the Krylov subspace using the block Arnoldi process [7]. The block Arnoldi process iteratively calculates k circuit moment matrices as follows.

$$\mathbf{m}_0=\mathbf{G}^{-1}\mathbf{B}$$
$$\mathbf{m}_k=\left(\mathbf{G}^{-1}\mathbf{C}\right)\mathbf{m}_{k-1} \tag{3}$$

where $\mathbf{m}_k$ is an $n\times N$ and k-th moment matrix. The Krylov subspace is a space that is spanned by the column vectors of moment matrices.

$$\mathrm{Kr}=colsp\left[\mathbf{m}_0\quad \mathbf{m}_1\quad \cdots\quad \mathbf{m}_{k-1}\right]$$
$$\mathbf{V}_k=\mathrm{QR}\left(\left[\mathbf{m}_0\quad \mathbf{m}_1\quad \cdots\quad \mathbf{m}_{k-1}\right]\right) \tag{4}$$

Ortho-normal projection matrix $\mathbf{V}_k$ of the Krylov subspace is used to project $\mathbf{x}(t)$ onto the Krylov subspace. However, this process is numerically unstable because there are great magnitude differences among the elements of moment matrices. For numerical stability, PRIMA performs an ortho-normalization process during the moment calculation processes. The ortho-normalized moment matrices are calculated as (5).

$$\begin{aligned} \mathbf{v}_0 &= \mathrm{QR}(\mathbf{m}_0) \\ [\mathbf{v}_0 \quad \mathbf{v}_1] &= \mathrm{QR}([\mathbf{v}_0 \quad \mathbf{m}_1']) \\ &\vdots \\ [\mathbf{v}_0 \quad \mathbf{v}_1 \quad \cdots \quad \mathbf{v}_{k-1}] &= \mathrm{QR}([\mathbf{v}_0 \quad \mathbf{v}_1 \quad \cdots \quad \mathbf{m}_{k-1}']) \\ \mathbf{m}_i' &= (\mathbf{G}^{-1}\mathbf{C})\mathbf{v}_{i-1} \text{ for } {}^{\forall} i > 1 \\ \mathbf{V}_k &= [\mathbf{v}_0 \quad \mathbf{v}_1 \quad \cdots \quad \mathbf{v}_{k-1}] \end{aligned} \tag{5}$$

The reduced order model is obtained by projecting (1) onto the Krylov subspace using the projection matrix $\mathbf{V}_k$ in (5).

$$\begin{aligned} \mathbf{G}_q &= \mathbf{V}_k^T \mathbf{G} \mathbf{V}_k \\ \mathbf{C}_q &= \mathbf{V}_k^T \mathbf{C} \mathbf{V}_k \\ \mathbf{B}_q &= \mathbf{V}_k^T \mathbf{B} \\ \mathbf{G}_q \mathbf{x}_q(t) + \mathbf{C}_q \frac{d\mathbf{x}_q(t)}{dt} &= \mathbf{B}_q \mathbf{u}(t) \end{aligned} \tag{6}$$

where q equals to k times N, $\mathbf{G}_q$, $\mathbf{C}_q$ are $q \times q$ matrices, $\mathbf{B}_q$ is a $q \times N$ matrix, and $\mathbf{x}_q(t)$ is a q-dimensional variable vector.

The original variable vector $\mathbf{x}(t)$ can be obtained by multiplying $\mathbf{V}_k$ to $\mathbf{x}_q(t)$.

$$\mathbf{x}(t) = \mathbf{V}_k \mathbf{x}_q(t) \tag{7}$$

During the reduction process, q times of sparse matrix solving processes are performed and k times of QR decomposition processes are performed. Total computational complexity is given as in (8).

$$\text{computational complexity} = O\left(f(n)kN + nk^3N^2\right) \tag{8}$$

where $f(n)$ is the computational complexity of the sparse matrix solving. In practical case, polynomial degree of $f(n)$ is less than 3, which is a degree of a computational complexity of dense matrix solving process. nk^3q^2 in (8) is the computational complexity of the QR decomposition. Originally, the computational complexity of the QR decomposition is $O\left(nk^2N^2\right)$. However, since the QR decomposition is performed repeatedly during the projection matrix calculation, the total complexity for the QR decomposition becomes $O\left(nk^3N^2\right)$.

In case of memory requirements, memory space for storing the projection matrix occupies almost all of total memory consumption, the projection matrix is a dense matrix and all elements are nonzero. Others are negligible. The memory requirements can be approximated to the memory size for storing the projection matrix. The memory complexity is expressed as (9).

$$\text{Memory complexity} \cong O(nkN) \ \textit{bytes}. \tag{9}$$

This memory requirement can be a serious problem, if an original circuit contains a large number of nodes and internal sources. For example, if an original circuit contains 10 million nodes and 1,000 internal sources, the PRIMA method requires at least 24 Gbytes of memory for storing the projection matrix.

3. R^{2}Power

R^{2}Power reduces a power network by iterative node eliminations. R^{2}Power can reduce an RC power network, which does not contain inductive components, that has the following properties.

(1) No grounded resistors; floating resistors only
(2) No floating capacitors; grounded capacitors only
(3) All supply voltage sources applied to the network have the same DC voltage value.

Above conditions can be satisfied for most practical cases. The **G** and **C** matrices and variable vector $\mathbf{x}(t)$ in (1) can be expressed as

$$\mathbf{G}=\mathbf{N},\mathbf{C}=\mathbf{Q},\ \text{and}\ \mathbf{x}(t)=\mathbf{v}(t). \tag{10}$$

In (10), **G** is a diagonally dominant matrix and **C** becomes a non-negative diagonal matrix.

In R^{2}Power, the backward Euler method is used for integration. Then, the circuit equation of (1) is expressed as follows.

$$\mathbf{A}\mathbf{x}(t)=\mathbf{b}(t) \tag{11}$$

where $\mathbf{A}=\mathbf{G}+\frac{\mathbf{C}}{h}$, and $\mathbf{b}(t)=\mathbf{B}\mathbf{u}(t)+\frac{\mathbf{C}}{h}\mathbf{x}(t-h)$.

R^{2}Power now removes nodes through matrix operations. Suppose R^{2}Power eliminates node n, which satisfies the elimination condition (12) where ε is a given error bound. Then, (11) can be re-written as (13).

$$\frac{g_{nn}c_{nn}}{h}\le\varepsilon \tag{12}$$

$$\begin{bmatrix}\mathbf{A}' & \mathbf{e}\\ \mathbf{e}^T & a_{nn}\end{bmatrix}\begin{bmatrix}\mathbf{x}'\\ x_n\end{bmatrix}=\begin{bmatrix}\mathbf{b}'\\ b_n\end{bmatrix} \tag{13}$$

In (13), a_{nn} is replaced by (14) to generate a realizable RC circuit.

$$\tilde{a}_{nn}=\frac{g_{nn}}{1-g_{nn}c_{nn}/h} \tag{14}$$

Then, (13) can be re-written as (15), and node n is eliminated as shown in (16).

$$\begin{bmatrix}\mathbf{A}' & \mathbf{e}\\ \mathbf{e}^T & \tilde{a}_{nn}\end{bmatrix}\begin{bmatrix}\mathbf{x}'\\ x_n\end{bmatrix}=\begin{bmatrix}\mathbf{b}'\\ b_n\end{bmatrix} \tag{15}$$

$$\mathbf{A}_{n-1}\mathbf{x}_{n-1}(t)=\mathbf{b}_{n-1}$$
$$\text{where}\ \mathbf{A}_{n-1}=\mathbf{A}'-\frac{\mathbf{e}\cdot\mathbf{e}^T}{\tilde{a}_{nn}},\ \text{and}\ \mathbf{b}_{n-1}=\mathbf{b}'-\frac{\tilde{a}_{nn}}{b_n}\mathbf{e} \tag{16}$$

Then, a circuit with n nodes is reduced to a circuit with n-1 nodes. The reduced circuit can be further reduced by iterating the above procedures. When the backward Euler method and fixed time step are used for the transient analysis, the error bound is given as (17).

$$\text{ERR}\le 2m\varepsilon \tag{17}$$

where m is the number of time points of transient analysis. Above relationship can be used to control analysis error of R^2 Power.

The upper bound for the computational complexity of R^{2}Power is given as

$$O(n^3) \tag{18}$$

where n is the matrix dimension. The actual computational complexity is $O(n)$~$O(n^3)$, depending on the number of non-zero elements of matrix **A**, which are generated during the elimination process of (16).

III. EXPERIMENTAL RESULTS

We evaluated the performances of PRIMA and R^{2}Power from the viewpoint of the accuracy, efficiency, and memory consumption. We used Synopsys HSPICE^{TM} [8] for the transient analysis of the networks reduced by two methods.

In case of PRIMA, a reduced order model is given as the circuit equation (6). Therefore, it is necessary to convert the reduced order model to a reduced RC network for SPICE simulation, and several methods have been proposed for the conversion [3, 9-11]. For our experiments, we used the Matsumoto's method [9] which is known as the best among them [12]. In addition, although R^{2}Power uses the Backward Euler for node elimination, it reduces the given circuit to an RC network. Therefore, both of the reduced networks can be analyzed using HSPICE.

All experiments were performed on a Linux machine with Intel Core2Quad^{TM} Q6600 2.4 GHz CPU and 4 Gbytes memory. PRIMA was implemented using the C programming language with the Intel Math Kernel Library (MKL) [13]. R^{2}Power was also implemented using the C programming language. Since R^{2}Power cannot handle inductive components, we generated several benchmark circuits that have a regular RC mesh structure, and we randomly assigned element values in reasonable ranges. The benchmark circuits have 10,000, 50,000, 120,000 and 200,000 nodes, and 10, 50, 100, and 500 internal current sources modeling switching logic

gates.

We ran PRIMA while increasing the number of preserved moments k from 1 to 5. In general, 2 or 3 moments are usually enough for RC networks, and 4 and 5 moments were used to show the high accuracy of PRIMA. We ran R^{2}Power while increasing the number of iterations from 50 to 100 by the interval of 10, and we set h and ε to 1ps and 1%.

1. CPU Time for Reduction

Fig. 1 shows the CPU times required for reduction process of two methods when the numbers of internal current sources are 10 and 100, the number of preserved moments for PRIMA is 3, and the number of iterations for R^{2}Power is 80. The CPU time of PRIMA rapidly increased as the number of sources increases. On the other hand, the CPU time of R^{2}Power remained almost constant although we increased the number of internal current sources. The CPU times of both methods increased super-linearly with the increasing number of nodes. Tables 1 and 2 present the CPU times of reduction process of both methods. In case of PRIMA, CPU times increase super-linearly with the increasing numbers of sources and moments (Table 1). In general, the reduction time of the moment matching-based methods increases as the number of sources increases, as they calculate all

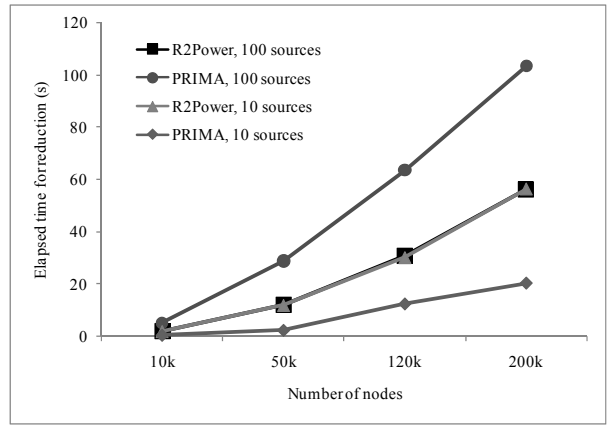

Fig. 1. CPU times of reduction processes of both methods when k = 3 and the number of iterations is 80.

circuit moments for each source. Also, it clearly increases as the number of preserved moments increases. On the other hands, the CPU time of R^{2}Power did not depend on the numbers of sources or the numbers of iterations; it depended only on the number of nodes (Table 2). It may be worthwhile mentioning that PRIMA failed to reduce some circuits that contain 500 internal current sources due to the high memory requirements.

2. CPU Time for SPICE Simulation

Fig. 2 shows the speed-up ratios of SPICE simulation of both methods for the cases that the number of pre-

Table 1. CPU Time of PRIMA Process (sec)

	$k=1$				$k=3$				$k=5$			
# of sources	10	50	100	500	10	50	100	500	10	50	100	500
10,000 nodes	0.25	0.56	1.23	11.83	0.5	1.93	5.35	80.71	0.83	4.26	13.39	242.51
50,000 nodes	1.19	2.51	8.01	59.51	2.36	8.51	29.02	397.64	3.82	23.9	69.91	Failed
120,000 nodes	4.81	6.54	17.45	142.02	12.42	25.66	63.68	Failed	27.46	47.66	151.11	Failed
200,000 nodes	7.77	12.33	28.9	Failed	20.37	43.46	103.5	Failed	45.49	86.23	251.98	Failed

Table 2. CPU Time of R2Power Process and # of Remained Nodes (sec (# of remained nodes))

	50 iterations				80 iterations				100 iterations			
# of sources	10	50	100	500	10	50	100	500	10	50	100	500
10,000 nodes	1.68 (265)	1.7 (269)	1.74 (298)	1.84 (540)	1.82 (67)	1.86 (87)	1.91 (131)	1.85 (540)	1.82 (32)	1.87 (65)	1.92 (131)	1.86 (540)
50,000 nodes	10.57 (1,386)	10.43 (1,427)	10.61 (1,450)	11.17 (1,543)	11.8 (365)	11.82 (371)	12.04 (396)	12.5 (582)	11.7 (148)	12.04 (158)	12.3 (185)	12.27 (539)
120,000 nodes	26.88 (3,455)	27.51 (3,439)	27.85 (3,523)	28.99 (3,587)	30.12 (905)	30.4 (908)	30.44 (950)	31.89 (1061)	30.73 (379)	30.78 (381)	31.94 (411)	32.83 (574)
200,000 nodes	49.08 (5,864)	50.25 (5,783)	51.39 (5,800)	48.79 (6,123)	56.42 (1,587)	55.42 (1,548)	56.08 (1,564)	55.52 (1758)	55.19 (666)	54.79 (646)	56.68 (669)	56.71 (844)

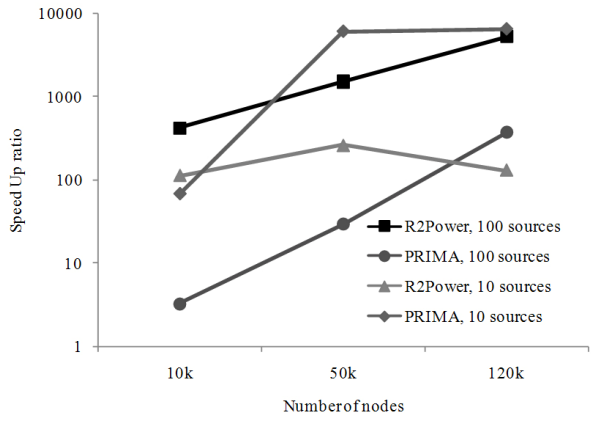

Fig. 2. Speed up of SPICE simulation of both methods, when k = 3, and the number of iteration is 80.

served moments, k, is 3 and the number of iteration is 80.

Tables 3 and 4 show the CPU times of SPICE simulation for the circuits reduced by both methods. In case of PRIMA, since the number of nodes of the reduced circuit depends only on the number of sources and the number of preserved moments, CPU time does not depend on the number of nodes of the original circuit (Table 3). Therefore, PRIMA showed superior results when the number of nodes of the original circuit is large and the number of sources is small. On the other hand, in case of R^2Power, CPU time does not seem to be strongly related with the number of sources. This is because, during the R^2Power reduction process, internal current sources are combined with other sources. Therefore, the number of remaining sources decreases as the number of remaining nodes decreases. In case of the moment matching-based methods, the number of nodes in a reduced circuit equals to the number of sources multiplied by the number of preserved moments. Therefore, the CPU time of SPICE simulation increases as the number of sources or preserved moments increases instead of the number of nodes in the original circuit.

3. Efficiency and Accuracy

We evaluated the efficiency and error of the two methods, compared to HSPICE. The efficiency is defined as the ratio of the SPICE CPU time on the original circuit to each method's total analysis time. Each method's total analysis time is the sum of the reduction time and SPICE simulation time on the reduced circuit. The speed up efficiency is given below.

$$\text{Speed up} = \frac{t_{\text{SPICE on original circuit}}}{t_{\text{reduction}} + t_{\text{SPICE on reduced circuit}}} \tag{19}$$

For accuracy evaluation, we observed the voltages at the nodes to which current sources are connected. These nodes are meaningful to observe as the current sources represent the switching logic gates in a circuit. Table 5 shows the efficiency and accuracy of both methods for three randomly generated circuits, all having 120,000

Table 3. CPU Time of Spice Simulation Using PRIMA Results (sec (speed up))

	$k=1$				$k=3$				$k=5$		
# of sources	10	50	100	500	10	50	100	500	10	50	100
10,000 nodes	0.14 (73.4)	0.64 (47.4)	5.35 (11.3)	348.99 (0.6)	0.15 (68.5)	2.52 (20.9)	18.28 (3.3)	1118.81 (0.2)	0.21 (48.9)	4.74 (6.4)	31.57 (1.9)
50,000 nodes	0.20 (3,031.1)	0.39 (1,212.9)	5.62 (103.8)	341.09 (1.5)	0.1 (6,062.2)	1.28 (369.5)	19.66 (29.7)	1137.44 (0.5)	0.13 (4,663.2)	2.3 (205.7)	29.5 (19.8)
120,000 nodes	0.31 (6,688.8)	0.46 (8,818.8)	5.64 (1,236.9)	344.04 (N/A)	0.32 (6,479.8)	1.15 (3,527.5)	18.58 (375.5)	N/A	0.54 (3,839.9)	2.23 (1,819.1)	30.66 (227.5)

Table 4. CPU Time of Spice Simulation Using R^2Power Results (sec (speed up))

	50 iterations				80 iterations				100 iterations			
# of sources	10	50	100	500	10	50	100	500	10	50	100	500
10,000 nodes	0.23 (44.7)	0.24 (126.4)	0.27 (223.3)	0.6 (354.1)	0.09 (114.1)	0.1 (303.3)	0.14 (430.6)	0.61 (348.3)	0.07 (146.7)	0.09 (337.0)	0.14 (430.6)	0.61 (348.3)
50,000 nodes	2.29 (264.7)	2.61 (181.2)	2.67 (218.5)	3.12 (165.3)	0.33 (1,837.0)	0.35 (1,351.5)	0.38 (1,535.3)	0.69 (747.3)	0.14 (4,330.1)	0.15 (3,153.5)	0.18 (3,241.2)	0.64 (805.7)
120,000 nodes	15.64 (132.6)	14.52 (279.4)	15.69 (444.6)	16.55 (N/A)	1.1 (1,885.0)	1.13 (3,590.0)	1.32 (5,284.8)	1.64 (N/A)	0.35 (5,924.4)	0.36 (11,268.5)	0.4 (17439.8)	0.68 (N/A)

Table 5. Total Analysis Times of Both Methods and Accuracy for Three Randomly Generated Circuits (120,000 nodes)

	PRIMA								R^2Power								
# sources	10			100			500		10			100			500		
Moments /Iterations	1	3	5	1	3	5	1	2	50	80	100	50	80	100	50	80	100
Total time (s)	5.1	12.7	28.0	23.1	82.3	181.8	486.1	1,149.5	42.5	31.2	31.1	43.5	31.8	32.3	45.5	33.5	33.5
Speed up	265	106	48	302	85	38	53	22	32	43	44	160	220	216	562	763	763
Max Err	2.2e-2	7.0e-4	3.7e-4	1.8e-2	1.9e-3	3.4e-4	1.4e-2	3.0e-3	5.4e-3	1.2e-2	1.6e-2	1.9e-2	2.6e-2	2.8e-2	8.7e-2	9.6e-2	10.2e-1
Avg Err	1.8e-3	1.7e-5	1.8e-5	1.7e-3	6.3e-6	6.5e-6	1.8e-3	2.4e-6	2.1e-4	4.2e-4	6.8e-4	4.0e-4	7.5e-4	1.1e-3	8.9e-3	9.6e-3	1.0e-3

nodes and different number of sources. In the table, 'Max Err' and 'Avg Err' represent the maximum and average error of all nodes at all time points, defined as follows; first, we define the node error for node i at time point t_j, Error_i^j, as in (20)

$$\text{Error}_i^j = \frac{\left| v_i^{\text{Original}}(t_j) - v_i^{\text{Reduced}}(t_j) \right|}{\left| v_i^{\text{Original}}(t_j) \right|} \tag{20}$$

where $v_i^{\text{Original}}(t_j)$ and $v_i^{\text{Reduced}}(t_j)$ are voltage values of node i at time point t_j, obtained through SPICE simulations for original and reduced circuits, respectively. Then, the average error of all nodes at all time points is defined as follows

$$\text{Avg Err} = \frac{1}{nN_t} \sum_i \sum_j \text{Error}_i^j \tag{21}$$

where n and N_t are the number of nodes and time points, respectively. The maximum error for all nodes at all time points is then given as follows.

$$\text{Max Err} = \max_i \left\{ \max_j \left(\text{Error}_i^j \right) \right\} \tag{22}$$

In the case of PRIMA, accuracy tended to increase while efficiency decreased, as we increased the number of sources increased or the number of preserved moments. On the other hand, R^2Power showed less accurate results as we increased the number of sources or the number of iterations. The efficiency was not affected by the number of sources. Fig. 3 and 4 illustrate the efficiency-error relationships of both methods for the

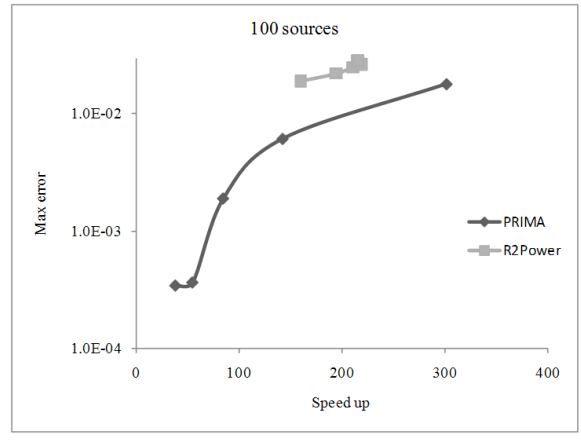

Fig. 3. Performance comparison of both methods (120,000 nodes and 100 sources).

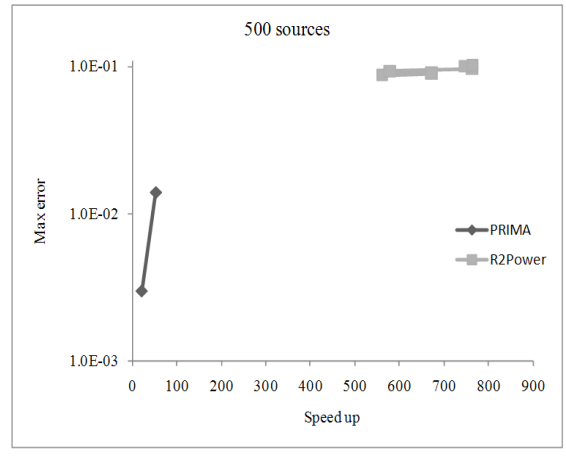

Fig. 4. Performance comparison of both methods, (120,000 nodes and 500 sources).

benchmark circuits with 100 current sources and 500 current sources, respectively. PRIMA showed more accurate results at the same efficiency. In case of the moment matching-based methods, if the number of sources

increases, both the number of nodes in a reduced circuit increases and the number of poles modeling the original circuit increases. Therefore, the accuracy increases, as the number of sources increases.

4. Memory Requirement

We compared the memory requirements of both methods for reduction using eight benchmark circuits that have 10,000, 50,000, 120,000, and 200,000 nodes, each with 100 and 500 internal current sources. Table 6 illustrates the comparison results. PRIMA required much larger memory than R^2Power. The memory requirement of PRIMA increases as the number of nodes, sources, or preserved moments increases. On the other hand, memory requirement of R^2Power depended only on the number of nodes, and they were almost independent of the number of current sources or iterations.

In the experiment, PRIMA consumed too much memory so that we could run it on the large circuits with lots of nodes and current sources. For example, PRIMA occupied about 4 Gbytes of memory when running on a circuit with 200,000 nodes and 100 current sources, and we could not run it for some circuits with 500 current sources. Since power network typically consists of 1M ~ 10M nodes, the large memory requirement of PRIMA can impose a big burden on users. With the memory requirement given in (9), most moment matching methods have a similar memory explosion problem.

IV. CONCLUSIONS

This paper investigated the basic concepts of PRIMA and R^2Power, and evaluated their performances using benchmark circuits. PRIMA is one of the widely used moment matching MOR techniques and R^2Power is the most recent node elimination MOR technique.

PRIMA can reduce a linear RLC circuit, and it provides very accurate results. However, it has high computational complexity and requires large memory space. The reduced circuit of PRIMA contains dependent sources and it may degrade the simulation efficiency. R^2Power has the advantages of lower computational complexity and lower memory requirements in comparison with PRIMA. The reduced model can directly be realizable as a circuit. The reduced circuit does not contain dependent sources. In order to use R^2Power, the given circuit must satisfy some assumptions, described in Section II.3. In most practical cases, these assumptions are satisfied.

In the experiments, PRIMA showed more accurate results than R^2Power at the same speed up. Unfortunately, PRIMA failed to reduce a large circuit due to the lack of memory space. The memory requirements of the moment matching based-methods are very high and it may be a big burden when we analyze the power networks of the state-of-art VLSI designs. The accuracy of the moment matching-based methods, including PRIMA, increases, as the number of sources or the number of preserved moments increases. On the other hand, the analysis efficiency decreases. The accuracy of R^2Power decreases as the number of sources or the circuit reduction rate increases. It is possible to control the analysis error can be controlled using (17). R^2Power requires smaller memory size, when compared to PRIMA.

In conclusion, if a power network contains inductive elements (L), we need to use PRIMA. If a power network does not contain L and if we need to set analysis error limit, we need to use R^2Power. Otherwise, either one can be used. In general, PRIMA is more accurate

Table 6. Memory Requirements (MBytes) of Both Methods for Eight Randomly Generated Circuits

	PRIMA						R^2Power					
# sources	100			500			100			500		
Moments /Iteration	1	3	5	1	3	5	50	80	100	50	80	100
10,000 nodes	69.6	147.5	270.3	241.7	745.5	1327.1	8.2	8.2	8.2	8.2	8.2	8.2
50,000 nodes	303.1	749.6	1204.2	1105.9	3248.1	N/A	28.7	28.7	28.7	28.7	28.7	28.7
120,000 nodes	634.9	1634.3	2613.2	2474	N/A	N/A	69.6	69.6	69.6	69.6	69.6	69.6
200,000 nodes	1040.4	2601	3833.9	N/A	N/A	N/A	114.7	114.7	114.7	114.7	114.7	114.7

and R^2Power is more efficient. However, the actual accuracy and efficiency highly depends on the circuit structure as well as parameters. Thus, it is difficult to draw conclusions on PRIMA and R^2Power in accuracy and efficiency. Users are recommended to compare two approaches in terms of accuracy and efficiency to select an appropriate one using their own benchmark circuits.

ACKNOWLEDGMENTS

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-(C1090-0902-0024)).

REFERENCES

[1] C. Tirumurti, S. Kundu, S. Sur-Kolay *et al.*, "A modeling approach for addressing power supply switching noise related failures of integrated circuits." pp.1078-1083 Vol.2.

[2] L. Pillage, and R. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol.9, No.4, pp.352-366, 1990.

[3] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol.17, No.8, pp. 645, 1998.

[4] P. Feldmann, and R. Freund, "Efficient Linear Circuit Analysis by Pad Approximation via the Lanczos Process," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol.14, No.5, 1995.

[5] B. N. Sheehan, "TICER: Realizable reduction of extracted RC circuits." pp.200-203.

[6] H. CHE, H. PARK, J. KIM *et al.*, "Realizable Reduction of RC Networks with Current Sources for Dynamic IR-Drop Analysis of Power Networks of SoCs," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences,* Vol.92, No.2, pp. 475-480, 2009.

[7] W. Arnoldi, "The principle of minimized iterations in the solution of the matrix eigenvalue problem," *Q. Appl. Math,* Vol.9, No.17, pp.17-29, 1951.

[8] Synopsys. "HSPICE," [Online]. Available: http://www.synopsys.com/Tools/Verification/AMSVerification/CircuitSimulation/HSPICE/Pages/default.aspx.

[9] Y. Matsumoto, Y. Tanji, and M. Tanaka, "Efficient SPICE-netlist representation of reduced-order interconnect model," pp.145-148.

[10] T. Palenius, J. Roos, and S. Aaltonen, "Development and comparison of reduced-order interconnect macromodels for time-domain simulation," pp.757-760, Vol.2.

[11] M. Celik, L. Pileggi, and A. Odabasioglu, *IC interconnect analysis*: Kluwer Academic Pub, 2002.

[12] T. Palenius, and J. Roos, "Comparison of reduced-order interconnect macromodels for time-domain simulation," *IEEE transactions on microwave theory and techniques,* Vol.52, No.9, pp.2240-2250, 2004.

[13] Intel. "Intel Math Kernel Library (Intel MKL) 10.2," [Online]. Available: http://software.intel.com/en-us/intel-mkl/.

Jinwook Kim received the B.E. degree in Electrical and Electronic Engineering in 2007 from the Pohang University of Science and Technology (POSTECH), Republic of Korea. Currently, he is working toward the Ph.D. degree in Electrical and Computer Engineering at the Pohang University of Science and Technology (POSTECH), Republic of Korea. His research interests include model-order reduction techniques and power noise analysis.

Young Hwan Kim received the B.E. degree in Electronics in 1977 from the Kyungpook National University, Rep. of Korea, and the M.S. and Ph.D. degrees in Electrical Engineering in 1985 and 1988 from the University of California, Berkeley, USA. From 1977 to 1982, he was with the Agency for Defense Development, Rep. of Korea. From 1983 to 1988, he worked as a Post Graduate Researcher at the Electronic Research Lab., University of California, Berkeley, USA. He also has served as editor of the Journal of the Institute of Electronics Engineers of Korea from 1991 to 1996. He is currently a Professor of Electronic and Electrical Engineering, Pohang University of Science and Technology (POSTECH), Rep. of Korea. His research interests include the circuit design for flat panel applications, timing verification, crosstalk analysis, power analysis, process-aware chip design.