Parallel Transient Simulation of Power Delivery Networks using Model Order Reduction

Marco T. Kassis, Yaswanth R. Akaveeti, Brett H. Meyer, and Roni Khazaka
Department of Electrical and Computer Engineering, McGill University, Montréal, Québec, Canada, H3A 0E9
Email: marco.kassis@mail.mcgill.ca, yaswanth.akaveeti@mail.mcgill.ca, brett.meyer@mcgill.ca, roni.khazaka@mcgill.ca

***Abstract*—On-chip power delivery networks have become an important design bottleneck while posing a significant challenge to design automation tools due to their large models. In this paper we propose a method that uses model order reduction methodologies in order to reformulate the simulation as a reduced parallel simulation problem that can take advantage of modern multi-core CPUs. Numerical examples are used to illustrate the accuracy and efficiency of the proposed method.**

***Index Terms*—Power Delivery Networks, Model Order Reduction, Parallel Transient Analysis.**

I. Introduction

As device sizes decrease and levels of integration increase on a single die, power consumption and on-chip power delivery networks (PDNs) have become an important design bottleneck [1]. Furthermore, PDNs pose a challenge to design automation tools due to the size of equations to be solved which can be in the order of hundreds of thousands of nodes [2]. One example of such tools is VoltSpot [2] which has been recently proposed and shown to be efficient and accurate through comparisons with the IBM PDN analysis benchmark suite described in [3]. However, the efficiency of VoltSpot is limited by the very large systems of equations that must be solved.

Model order reduction methods such as [4,5] have been proposed for improving the CPU cost of simulating large systems. However, these methods are ill-suited to PDN networks due to the large number of ports which result in an impractically large reduced order model [6]. In order to address this issue, instead of focusing on obtaining a reduced order macromodel, the proposed approach uses MOR techniques in order to efficiently recast the simulation as a parallel computation problem that can easily take advantage of modern multi-core CPUs. First the simulation problem is reformulated as a number of decoupled equations that can be solved in parallel. A parallel multi-level reduction algorithm is proposed to efficiently obtain the reduced order equations for each of the decoupled problems. These reduced equations can then be in turn solved in parallel. The final step is to map the solutions back to the original system space and use the superposition principle to obtain the final answer. This step can be formulated as multiplication of two dense matrices which is suitable for parallel computation. The efficiency and accuracy of the method is demonstrated in the numerical examples which are run in parallel on a computer with 16 cores.

The authors would like to thank Natural Sciences and Engineering Research Council of Canada (NSERC) and the Regroupement Stratégique en Microsystèmes du Québec (ReSMiQ) for supporting this project.

II. Problem Statement

The PDN architectural model generated by VoltSpot is described in full detail in [2]. The Modified Nodal Analysis (MNA) [7] equations of such a model can be expressed as,

$$\begin{aligned} \mathbf{G}x(t) + \mathbf{C}\dot{x}(t) &= \mathbf{B}i(t) + \mathbf{b}_{dc} \\ u(t) &= \mathbf{R}^T x(t) \end{aligned} \tag{1}$$

where $i(t) \in \mathbb{R}^p$ represents the current sources connecting the V_{dd} and V_{ss} meshes. These sources model the power consumption in each architectural block of the chip. $u(t) \in \mathbb{R}^m$ is the output voltage vector, containing the nodal voltages of the V_{dd} and V_{ss} meshes. $x(t) \in \mathbb{R}^n$ is the vector of unknowns. $\mathbf{G}, \mathbf{C} \in \mathbb{R}^{n\times n}$ contain the contributions of the memoryless and memory elements respectively. $\mathbf{B} \in \mathbb{R}^{n\times p}$ and $\mathbf{R} \in \mathbb{R}^{n\times m}$ are matrices mapping inputs and outputs to the circuit equations. $\mathbf{b}_{dc} \in \mathbb{R}^n$ contains the DC voltage sources. p is the number of input current sources, which is typically the same as the number of architectural blocks in the model we use. m is the number of output voltage nodes considered. While the system in (1) is sparse, its size n can be very large making the CPU cost of time-domain simulations high. It is also important to note that this system does not represent a macromodel, but is simply a set of circuit equations. Our goal in that paper is to use Model Order Reduction methodologies in order to improve the simulation time while maintaining accuracy. However, if model order reduction is applied directly on (1) using the block Arnoldi process, the reduced order model would be very large and the reduction cost very high due to the large number of input ports p. As a result we use the superposition principle to reformulate the problem into the following set of equations

$$\begin{aligned} \mathbf{G}x_{dc} &= \mathbf{b}_{dc} \\ \mathbf{G}x_j(t) + \mathbf{C}\dot{x}_j(t) &= \mathbf{b}_j i_j(t) \\ u(t) &= \mathbf{R}^T\left(x_{dc} + \sum_{j=1}^{p} x_j(t)\right) \end{aligned} \tag{2}$$

where $\mathbf{b}_j$ are the columns of $\mathbf{B}$ and $j = 1, 2, \ldots, p$. Note that while the above system of equations can be solved in parallel, it is much more efficient to simply solve the system in (1) directly. However our goal is not to solve (2) but to first reduce it before solving the reduced system.

III. Order Reduction of PDN Network

The proposed methodology is composed of a multi-level MOR followed by transient analysis of the reduced system. Note that the model order reduction time can be considered as

part of the setup cost and the reduced system, once computed, can be repeatedly solved for different input patterns.

A. Order Reduction

As discussed in Section II the model reduction is applied on the systems in (2). This results in p reduced systems (one for each input) which can be evaluated in parallel. The superposition principle is then used to compute the final result.

1) First Level of Reduction: The reduction subspace is obtained using a multi-point expansion on the frequency axis. The locations of the frequency expansions is found using a binary search approach similar to [8]. The details are outlined in Algorithm 1. Note that the binary search is only done on one input of the system *i.e.* one column vector $\mathbf{b}_j$ of the $\mathbf{B}$ matrix, and the same expansion points are then used for the remaining inputs. Once the moments are computed, QR decomposition is used to obtain the reduction subspace $\mathbf{Q}_{1j}$ for each input vector $\mathbf{b}_j$. Congruence transformation is then used to obtain p reduced systems as follows:

$$\begin{aligned} \tilde{\mathbf{G}}_j \tilde{x}_j(t) + \tilde{\mathbf{C}}_j \dot{\tilde{x}}_j(t) &= \tilde{\mathbf{b}}_j i_j(t) \\ \tilde{\mathbf{G}}_j &= Q_{1j}^T * \mathbf{G} * Q_{1j} \\ \tilde{\mathbf{C}}_j &= Q_{1j}^T * \mathbf{C} * Q_{1j} \\ \tilde{\mathbf{b}}_j &= Q_{1j}^T * \mathbf{b} \end{aligned} \tag{3}$$

Where $\tilde{\mathbf{G}}_j, \tilde{\mathbf{C}}_j \in \mathbb{R}^{q_1 \times q_1}$ and $\tilde{\mathbf{b}}_j \in \mathbb{R}^{q_1 \times 1}$ comprise the level 1 reduced order model, Q_{1j} are the set of orthonormal matrices, and the index $j = 1, 2, \ldots, p$ spans all the inputs. It is important to note that the p subspaces Q_{1j} and p reduced systems in (3) can be computed in parallel on a multicore CPU.

Algorithm 1 Binary search algorithm for optimal locations of frequency expansions

```
Set f_l = 0 and f_h = f_max
Compute 2 moments at f_l and f_h
repeat
  f_mid = (f_l + f_h)/2;
  Construct reduced system {G̃_j, C̃_j, b̃_j}
  Check accuracy of reduced system at f_mid
  if Accurate then
    Exit
  else
    Add 2 moments at f_mid
  end if
until All midpoints are accurate
```

2) Second Level of Reduction: SVD of the frequency response of the intermediate order is used to map to the final reduced subspace, similar to [5]. To save on frequency response CPU cost, diagonalization and real transformation of the intermediate order is done, as in [9], however the full details are not reported due to lack of space. A new set of orthonormal matrices Q_{2j} are used to map the intermediate model from an order of q_1 to q_2 as follows:

$$\begin{aligned} \hat{\mathbf{G}}_j \hat{x}_j(t) + \hat{\mathbf{C}}_j \dot{\hat{x}}_j(t) &= \hat{\mathbf{b}}_j i_j(t) \\ \hat{\mathbf{G}}_j &= Q_{2j}^T * \tilde{\mathbf{G}}_j * Q_{2j} \\ \hat{\mathbf{C}}_j &= Q_{2j}^T * \tilde{\mathbf{C}}_j * Q_{2j} \\ \hat{\mathbf{b}}_j &= Q_{2j}^T * \tilde{\mathbf{b}}_j \end{aligned} \tag{4}$$

Where $\hat{\mathbf{G}}_j, \hat{\mathbf{C}}_j \in \mathbb{R}^{q_2 \times q_2}$ and $\hat{\mathbf{b}}_j \in \mathbb{R}^{q_2 \times 1}$. The computations are also parallelized across the index j to reduce CPU time. The computations in (4) are also done in parallel. The final reduced model is passive by construction.

B. Time-Domain Simulation

Trapezoidal rule is used to perform transient analysis on the final reduced model, which uses parallel computation for every one of the independant inputs governed by index j. The time-domain responses are mapped back to the original space as well as superimposed to give the overall time-domain response using the transformation shown in equation (5).

$$\begin{aligned} \hat{\mathbf{R}}_j &= Q_j^T * \mathbf{R} \\ u(t) &= \mathbf{R}^T x_{dc} + \sum_{j=1}^{p} \hat{\mathbf{R}}_j^T \hat{x}_j(t) \end{aligned} \tag{5}$$

where $Q_j = Q_{1j} * Q_{2j}$. This can be formulated as a multiplication of 2 matrices, making the mapping computations more efficient.

IV. Simulation Results

To validate the proposed approach, 2 examples are presented. The VoltSpot model is used as the original large system, and the same setup as [2] is used. A constant step size of five time steps per cycle is used in the transient simulation following the same methodology as [2]. This leads to significant CPU saving for the original system because only one LU decomposition is needed and the CPU cost is dominated by forward/backward substitutions. Nonetheless, the proposed approach is shown to be much more efficient. Note that the simulations were performed over $10,000$ time steps, and the first half was considered warm-up and ignored. All computations are done on an Intel Xeon 16-core computer running at 2.4GHz with 64GB of memory.

A. Example 1: 4 Cores

The first example uses a PDN model for a $22\,\mathrm{nm}$ 4-core Intel Penryn-like processor running at 3.7GHz. Each core is $32-\mathrm{bit}$ $4-$way out-of-order, equipped with a $32\,\mathrm{kB}$ L1 instruction cache, a $32\,\mathrm{kB}$ L1 data cache and $3\,\mathrm{MB}$ unified L2 caches. The PDN has 111×111 nodes for each of the V_{dd} and V_{ss} meshes. The grid granularity, *i.e.* ratio of supply C4 pads to grid nodes is 1:5. This results in a total of 529 supply pads, 265 of which are V_{dd} and 264 are V_{ss}. The number of inputs of the system $p = 43$. The original order of the system is 122860, the intermediate order $q_1 = 42$ and the final reduced order $q_2 = 32$. The binary search algorithm converged after 5 iterations. Table I shows a summary of

the time profiling, speed-up and error results. Figure 1 shows the frequency magnitude plot of the original order versus the reduced order. The intermediate order's frequency expansions' locations are annotated on the plot. Figure 2 shows a section of the time-domain simulation for both the original and the final reduced one, demonstrating the precision of the analysis.

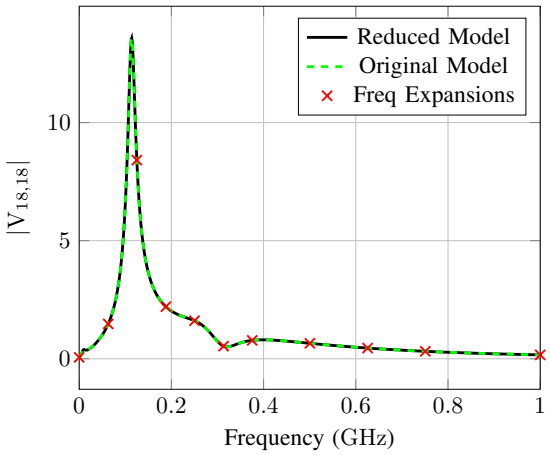

Fig. 1. Magnitude Plot for Example 1's Frequency Response

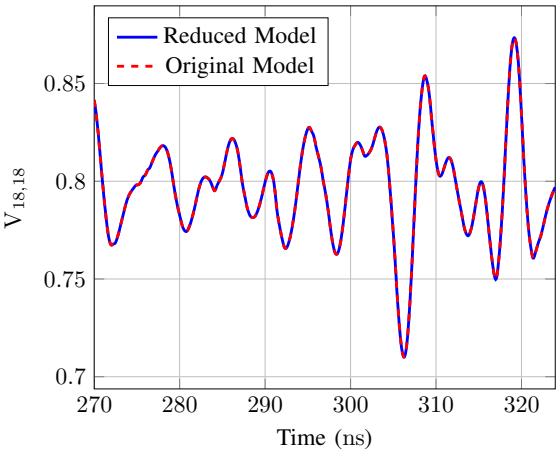

Fig. 2. Sample Transient Response for Example 1 at Node (18,18)

B. *Example 2:* 8 Cores

The second example uses a PDN model for a 22 nm 8-core Intel Penryn-like processor having the same specs as example 1, mentioned in subsection IV-A. The PDN has 161×161 nodes for each of the V_{dd} and V_{ss} meshes. The grid granularity is also 1:5. The total number of supply pads is 1089, 545 of which are V_{dd} and 544 are V_{ss}. The scenario where the second set of 4 cores have the same operations as the first 4 is assumed. This leads to having the same power traces as example 1, hence the number of inputs to the system is kept at $p = 43$. Note that this results in a better CPU cost performance of the MOR method while providing the worst case power supply noise. The original order of the system is 259020, the intermediate order $q_1 = 46$ and the final reduced order $q_2 = 32$. The binary search algorithm converged after 6 iterations. Table I shows that the speed-up is higher for example 2 than 1, illustrating the increase in efficiency of the proposed algorithm as the size of the PDN used increases.

TABLE I
TIME PROFILING FOR EXAMPLES 1 & 2

	Ex. 1	**Ex. 2**
Original Order	122860	259020
Intermediate Order	42	46
Reduced Order	32	32
MOR (s)	22.95	62.36
Original Transient (s)	76.36	182.45
Reduced Transient (s)	1.48	2.60
Speed-up (s)	51.7×	70.1×
Max. Transient Error	2.69×10^{-5}	4.58×10^{-5}
Max. Frequency Error	2.02×10^{-4}	5.59×10^{-4}

V. CONCLUSION

In this paper a model order reduction based approach for the simulation of power delivery networks is proposed. The proposed method uses the MOR methodology to a set of decoupled problems that can be solved in parallel and thus takes advantage of modern multi-core CPUs. The reduced models themselves are also generated in parallel using a multi-level reduction scheme to ensure a compact and accurate system. Numerical examples were presented to illustrate the accuracy and efficiency of the proposed approach running on a system with 16 computational cores.

REFERENCES

[1] K. Wang, B. H. Meyer, R. Zhang, M. Stan, and K. Skadron, "Walking pads: Managing c4 placement for transient voltage noise minimization," in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2014, pp. 1–6.

[2] R. Zhang, K. Wang, B. H. Meyer, M. R. Stan, and K. Skadron, "Architecture implications of pads as a scarce resource," in *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, June 2014, pp. 373–384.

[3] S. R. Nassif, "Power grid analysis benchmarks," in *2008 Asia and South Pacific Design Automation Conference*, March 2008, pp. 376–381.

[4] A. Odabasioglu, M. Celik, and L. T. Pileggi, "Prima: passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 8, pp. 645–654, Aug 1998.

[5] M. Ma and R. Khazaka, "Multi-level order reduction with nonlinear port constraints," in *2007 IEEE International Symposium on Circuits and Systems*, May 2007, pp. 1485–1488.

[6] ——, "Model order reduction with parametric port formulation," *IEEE Trans. Adv. Packag.*, vol. 30, no. 4, pp. 763–775, Nov 2007.

[7] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, Jun 1975.

[8] R. Sanaie, E. Chiprout, M. S. Nakhla, and Q. J. Zhang, "A fast method for frequency and time domain simulation of high-speed vlsi interconnects," *IEEE Trans. Microw. Theory Techn.*, vol. 42, no. 12, pp. 2562–2571, Dec 1994.

[9] A. Odabasioglu, M. Celik, and L. T. Pileggi, "Practical considerations for passive reduction of rlc circuits," in *Computer-Aided Design, 1999. Digest of Technical Papers. 1999 IEEE/ACM International Conference on*, Nov 1999, pp. 214–219.