SCALABLE ANALYSIS, VERIFICATION AND DESIGN OF IC POWER DELIVERY

A Dissertation

by

ZHIYU ZENG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2011

Major Subject: Computer Engineering

SCALABLE ANALYSIS, VERIFICATION AND DESIGN OF IC POWER DELIVERY

A Dissertation

by

ZHIYU ZENG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Peng Li
Committee Members,	Weiping Shi
	Duncan M. Walker
	Byung-Jun Yoon
	Sani R. Nassif
Head of Department,	Costas N. Georghiades

December 2011

Major Subject: Computer Engineering

ABSTRACT

Scalable Analysis, Verification and Design of IC Power Delivery. (December 2011)

Zhiyu Zeng, B.S., Zhejiang University

Chair of Advisory Committee: Dr. Peng Li

Due to recent aggressive process scaling into the nanometer regime, power delivery network design faces many challenges that set more stringent and specific requirements to the EDA tools. For example, from the perspective of analysis, simulation efficiency for large grids must be improved and the entire network with off-chip models and nonlinear devices should be able to be analyzed. Gated power delivery networks have multiple on/off operating conditions that need to be fully verified against the design requirements. Good power delivery network designs not only have to save the wiring resources for signal routing, but also need to have the optimal parameters assigned to various system components such as decaps, voltage regulators and converters. This dissertation presents new methodologies to address these challenging problems.

At first, a novel parallel partitioning-based approach which provides a flexible network partitioning scheme using locality is proposed for power grid static analysis. In addition, a fast CPU-GPU combined analysis engine that adopts a boundary-relaxation method to encompass several simulation strategies is developed to simulate power delivery networks with off-chip models and active circuits. These two proposed analysis approaches can achieve scalable simulation runtime.

Then, for gated power delivery networks, the challenge brought by the large verification space is addressed by developing a strategy that efficiently identifies a number of candidates for the worst-case operating condition. The computation complexity is reduced from $O(2^N)$ to $O(N)$.

At last, motivated by a proposed two-level hierarchical optimization, this dissertation presents a novel locality-driven partitioning scheme to facilitate *divide-and-conquer*-based scalable wire sizing for large power delivery networks. Simultaneous sizing of multiple partitions is allowed which leads to substantial runtime improvement. Moreover, the electric interactions between active regulators/converters and passive networks and their influences on key system design specifications are analyzed comprehensively. With the derived design insights, the system-level co-design of a complete power delivery network is facilitated by an automatic optimization flow. Results show significant performance enhancement brought by the co-design.

To my wife Jing and my parents.

ACKNOWLEDGMENTS

Looking back, I am very grateful for all I have received throughout these years.

My sincere gratitude goes to my advisor, Prof. Peng Li. Without his guidance and support, most of the work presented in this dissertation would not have been accomplished. He has enlightened me through his wide knowledge in Electronic Design Automation area and his deep intuitions about where my research should go and what is necessary to get there. During more than four years of knowing him, he has taught me the ways of how to analyze the problems and how to build concrete steps towards solving them. Prof. Li's valuable feedbacks from numerous one-on-one meetings, informal discussions and group meeting presentations have been the continuous source of my motivation and inspiration. He has always encouraged me to bear a big dream and to think for the long term which has helped to shape my personality.

I would like to thank Dr. Sani Nassif from IBM ARL and Prof. Vivek Sarin from the CSE Department. Discussing problems with Dr. Nassif has always been a pleasure to me. It was him who encourage me to look at the problems of on-chip voltage regulation. He also invited me to give a presentation about my research at IBM ARL and introduced me to Dr. Frank Liu, Dr. Zhuo Li, Dr. Howard Smith and Mr. Dave Thomas for helpful technical interactions. It was my pleasure to work with Prof. Sarin on building performance model for the parallel power grid static analysis approach. He taught me a lot on parallel computing and helped me migrate my codes to the supercomputer, Hydra. I also owe a lot to Prof. Weiping Shi, Prof. Duncan Walker, Prof. Byung-Jun Yoon and Prof. Jiang Hu for serving on my PhD committee and providing insightful feedbacks on my preliminary exam.

I would also like to thank many other people in the Computer Engineering Group who contributed a lot on my dissertation by providing reference codes and technical discussions. Dr. Zhuo Feng has been a big brother to me whose passion and diligence keep driving me forward. His GPU-based multigrid power grid solver is essential to the successful implementation of my CPU-GPU combined simulation engine, GSim. Dr. Wei Dong taught me a lot on general circuit simulation. Dr. Guo Yu guided me to design and draw layout for the incrementer residing in a digital PLL. Dr. Xiaoji Ye provided me with the modified SPICE code for GSim and his experience on using APPS. Suming Lai's voltage regulator designs and solid analog circuit background were critical in my research for on-chip voltage regulation. Tong Xu's fast implementation skill guaranteed our success in 2011 TAU Power Grid Simulation Contest. Dr. Boyuan Yan was always there when I needed help. I also thank Leyi Yin, Yong Zhang, Amandeep Singh, and Haokai Lu for helpful discussions.

I thank Prof. Jiming Song and Prof. Kangsheng Chen who served as my advisors when I was in Iowa State University and Zhejiang University respectively. They were the first two people who led me into the research field and made me enjoy doing research. Dr. Rajesh Gupta, Mr. Mandeep Singh, Mr. Zach Coombes and Mr. Lam Nguyen from Samsung Austin R&D Center helped me understand the importance of power delivery in real circuit designs and shorten the gap between industrial designs and research at the school.

Thank to my family, especially to my grandparents, my parents, my uncles and aunts, and my cousins for their help, support, comfort and encouragement through all these years. The last but not the least, I appreciate so much to the spiritual and emotional support and help from my wife Jing. Without her I would be a very different person today, and it would have been certainly much harder to finish my PhD degree. Still today, her unselfish love makes me a better person.

I finish with a silence of gratitude for my life.

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1

A. Preliminaries 1
1. Design Perspectives 3
2. Modeling 4
3. Design Trends and Challenges 6
B. Survey of Previous Work 8
1. Survey of PDN Analysis 8
2. Survey of PDN Verification 9
3. Survey of PDN Design 10
C. Proposed Solutions 12
1. Proposed Solutions on PDN Analysis 12
2. Proposed Solutions on Power-Gated PDN Verification 13
3. Proposed Solutions on PDN Wire Sizing and On-Chip Voltage Regulation Design 14

II POWER DELIVERY NETWORK ANALYSIS AND VERIFICATION 16

A. Locality-Driven Parallel Static Analysis for Power Delivery Networks 16
1. Background 17
2. Overview of the Proposed Approach 18
3. Parallel Locality-Driven Static Analysis 21
a. Parallel Boundary Current Approximation Using Locality 21
b. Parallel Partition Simulation Using Approximated Boundary Currents 24
c. Block-Based Iterative Error Reduction 28
d. Algorithm Flow and Implementation 29
e. Computational Cost Analysis 32
f. Performance Modeling for Parallel Implementation 35
4. Experimental Results 38
5. Summary 44

CHAPTER Page

B. GSim: A Fast CPU-GPU Combined Parallel Simulator for Power Delivery Networks with On-Chip Voltage Regulation 44
1. Background and Overview 44
2. CPU-GPU Combined Transient Simulation 46
a. GPU-Based Multigrid Method 46
b. Boundary Relaxation 48
c. Convergence 49
d. Simulation Flow 49
3. Experimental Results 49
4. Summary 52
C. Transient Verification of Power-Gated Power Delivery Networks 52
1. Background 52
a. Modeling for Power-Gated PDNs 53
b. Verification Metrics 53
2. Overview of the Transient Verification 55
a. Verification Tasks 55
b. Overview of the Proposed Transient Verification Methodology 58
3. Stable-Mode EM Verification for Global Grids 59
a. Challenges for Transient EM Verification 59
b. Superposition Approximation 62
c. Worst Case Validation 67
4. Other Verifications 70
a. Stable-Mode EM Verification for Local Grids . . . 70
b. Stable-Mode Peak Dynamic Voltage Drop Verification 72
c. Power-On Peak Dynamic Voltage Drop Verification 75
5. Experimental Results 78
a. Stable-Mode Verification 79
b. Power-On Verification 83
6. Summary 83

III POWER DELIVERY NETWORK DESIGN 84

A. Locality-Driven Parallel Power Grid Wire Sizing 84
1. Background 85
a. Problem Formulation 85
b. Constrained Nonlinear Optimization 86

CHAPTER Page

2. Overview of the Proposed Parallel Optimization . . . 87
a. Key Issues in Parallelizable Power Grid Optimization 87
b. Two-Level Hierarchical Optimization 89
c. Proposed Parallel Two-Step Optimization Formulation 92
3. Parallel Solution of Optimal Boundary Conditions . . 94
4. Parallel Optimization of Partitioned Sub Power Grids 96
a. Optimization of Partitioned Grids Using Relaxed Boundary Constraints 98
b. Maintenance of IR Drop Constraints 100
c. Fixing EM Violation 103
d. Algorithm Flow for the Proposed Locality Driven Parallel Optimization 104
5. Experimental Results 104
a. Partition Optimization 105
b. Overall Optimization 106
6. Summary 110
B. System-Level Co-Design of Power Delivery Networks with On-Chip Voltage Regulation 112
1. Background 112
a. System Modeling 112
b. Benefits of On-Chip Voltage Regulation 113
c. Low-Dropout Regulator Design 115
d. Decoupling Capacitor Sizing and Placement . . . 117
e. Buck Converter Design 117
2. System-Level Co-Design 120
a. LDO-Decap System Co-Design 120
b. LDO-Decap-BC System Co-Design 126
3. Co-Optimization Formulation and Methodology 127
a. Co-Optimization for LDO-Decap System 127
b. Co-Optimization for LDO-Decap-BC System . . . 133
4. Experiment Results 134
a. LDO-Decap System Co-Optimization 135
b. LDO-Decap-BC System Co-Optimization 141
5. Summary 143

IV CONCLUSION 144

REFERENCES 146

Page

VITA 153

LIST OF TABLES

TABLE Page

I Comparison of boundary IR drops and currents obtained in boundary window simulations using different window sizes with the exact solutions. Voltage drop is in mV. Current is in mA. 25

II Simulation runtime for flat simulation and the proposed parallel simulation. All the grids are divided into 16 partitions with 24 windows. 16 processors are used. Chol: Cholesky decomposition time; Sol: triangular solve time; Tot: total runtime; Mem: memory; Iter: number of iterations; Sp: speedup; E_m: maximum node voltage error; E_a: average node voltage error. Runtime is in seconds; error is in mV; and memory is in GB. 39

III Speedups of the parallel implementation of the algorithm over the sequential implementation of the algorithm. 16 processors are used for the parallel simulation. Runtime is in seconds. 40

IV Runtime analysis with various numbers of partitions for 9M, 12.96M, and 16M grids. Chol: Cholesky decomposition time; Sol: triangular solve time; Iter: number of iterations; Tot: total runtime. Runtime is in seconds. 42

V Number of iterations required by using various window sizes. Iter: number of iterations; Tot: total runtime. Runtime is in seconds. . . . 43

VI Transient simulation runtime and numbers of iterations of GSim for PNDs with on-chip LDOs. 1200 time steps are simulated. CPU%: the percentage of runtime spent on CPU. The runtime is in seconds. 51

TABLE Page

VII EM stable mode verification for gated PDN. The global VDD grid is examined. # Con.: number of configurations; T: total runtime; $|EM|_{max}$: maximum absolute average current; P: number of worst case candidates. Runtime is in hrs. EM is in mA. Runtime of the enumeration methods for larger circuits are the estimated time ($\sim$ time value). The transient verifications are run for 200 clock cycles (2000 time steps). 80

VIII Peak DVD stable mode verification for gated PDN. For grids with 8 and 16 grids, two different local grids are examined. Gcor: corner grid; Gcen: center grid; # Con.: number of configurations; T: total runtime; $DVD_{p,max}$: maximum peak DVD; P: number of worst case candidates. Runtime is in hrs. DVD is in mV. Runtime of the enumeration methods for larger circuits are the estimated time ($\sim$ time value). The transient verifications are run for at least 200 clock cycles (2000 time steps). 81

IX Peak DVD transition verification for gated PDN. The transition grid is at the center. For grids with 8 and 16 grids, two nodes in two different grids are examined. Gclo: the grid close to the transition grid; Gfar: the grid far away from the transition grid; # Con.: number of configurations; T: total runtime; $DVD_{p,max}$: maximum peak DVD; P: number of worst case candidates. Runtime is in hrs. DVD is in mV. Runtime of the enumeration methods for larger circuits are the estimated time ($\sim$ time value). The transient verifications are run for at least 200 clock cycles (2000 time steps). 82

X Optimized boundary voltage and current as functions of window size in the window-based optimization. WS: window size. C4S: C4 ring size. NV: voltage for a boundary node in V. NC: current for a boundary branch in mA. AV: average voltage for the boundary in V. AC: average current for the boundary in mA. OPT: optimal value. 96

TABLE Page

XI Optimization runtime for flat optimization, serial locality-driven optimization, and parallel locality-driven optimization. N: number of nodes. P: number of partitions. SIM: flat simulation runtime in sec. OPT: optimization runtime in sec. BWO: boundary window optimization runtime in sec. PO: partition optimization runtime in sec. TOT: total runtime in sec. AR: area reduction in %. $\mathrm{WS_b}$: beginning window size. $\mathrm{WS_t}$: maximum terminating window size. IT: number of iterations for window size determination. NVio: number of EM violation. $\mathrm{Vio_{max}}$: maximum EM violation in mA/um. 108

XII Multi-level optimization vs. straight optimization. 141

LIST OF FIGURES

FIGURE Page

1 Diagram of a complete power delivery network. 2

2 Diagram of a on-chip power delivery network. 2

3 A complete model of the power delivery network. 5

4 VDD grid model. 5

5 Design trends and challenges for power delivery networks. 7

6 Work towards scalable analysis, verification and design for IC power delivery. 12

7 Power grid partitions and partition boundaries. 18

8 Spatial locality in power grids. Red dots represent C4 connections. . 22

9 Finding the near-exact boundary currents via locality. Red dots represent C4 connections. Black dash lines represent partition boundaries. Blue dash lines represent window boundaries. 23

10 Impact of window size on boundary current approximation. Red dots represent C4 connections. Black dash lines represent partition boundaries. Blue dash lines represent window boundaries. 24

11 Power grid partition simulation by setting boundary currents. 27

12 Algorithm flow of the proposed parallel locality-driven method. . . . 31

13 Accuracy comparison of boundary current approximations using different numbers of partitions. The power grid in (a) is divided into 16 partitions, and the power grid in (b) is divided into 32 partitions. The values are node voltage errors after first iteration. The unit is Volt. 36

FIGURE Page

14 Accuracy comparison of boundary current approximations using different window sizes. The power grid in (a) uses window size 40 (C4 ring size 1), and the power grid in (b) uses window size 80 (C4 ring size 2). The values are node voltage errors after first iteration. The unit is Volt. 36

15 Runtime comparison between the flat simulation and the proposed parallel simulation. 41

16 The model for a power delivery network with on-chip voltage regulators. 45

17 GSim simulation diagram. 46

18 Scalability of runtime and memory consumption for the GPU solver. 47

19 Boundary relaxation for a single LDO. 48

20 GSim simulation flow. 50

21 A power-gated power delivery network model. 53

22 EM and DVD metrics. 54

23 Stable and transition configurations. 56

24 Proposed transient verification tasks. 57

25 The proposed transient verification methodology. 59

26 A simple power-gated PDN in [26]. For simplicity, only four local grids are shown. Each local grid has a single sleep transistor and two DC current loads. 60

27 The PDN model diagram. For simplicity, only four local grids are shown. Each local grid has two sleep transistors, one current load and one decoupling capacitor. 61

28 Switchable current source model for local grids. For simplicity, the package is not shown and there are only three local grids. 62

FIGURE Page

29 A simple RLC model for the power delivery network. 64

30 Switchable current source values for the global basic configuration (a) and the full-decap basic configuration (b). For simplicity, the package is not shown and there are only three local grids. 65

31 A simple example for switchable current source approximation. The package is not shown and there are only two local grids. 65

32 Superposition for global grid verification. 66

33 Flow for worst case validation. 68

34 Superposition for local grid verification. 71

35 Dynamic voltage drop for circuit block T_n. For simplicity, the package is not shown. 73

36 Drain-source conductance of a PMOS sleep transistor during power-on time. 76

37 Verification of power-on peak dynamic voltage drop. 77

38 A naive divide-and-conquer optimization approach. 88

39 Power grid partitioning by setting boundary voltages and currents. 89

40 Optimization of the partitioned grid under relaxed boundary conditions. 99

41 Merging process analysis 102

42 Impact of boundary conditions on the quality of final optimized power grid. (a): node voltage distribution before optimization. (b): EM distribution of horizontal wires before optimization. (c): node voltage distribution after optimization using window size 40. (d): EM distribution of horizontal wires after optimization using window size 40. (e): node voltage distribution after optimization using window size 65. (f): EM distribution of horizontal wires after optimization using window size 65. 107

FIGURE Page

43 Speedup of parallel over serial locality-driven optimization. 111

44 Threads runtime (in sec) of boundary window optimizations and partition optimizations for 640K-node power grid. 111

45 System structure and model of a power delivery network with on-chip voltage regulation. 113

46 Voltage drops for a power domain with LDOs and without LDOs. . 114

47 LDO topology. 116

48 Buck converter topology. 118

49 Power consumption of the PDN with on-chip LDOs. 120

50 Stability check reasoning procedure. 125

51 Phase margin vs. R_s. 125

52 LDO-decap system co-optimization flow. 130

53 Multigrid-based optimization. 132

54 Optimization results of decap uni-optimization, LDO uni-optimization, and co-optimization. (a) LDO and decap area in mm^2. (b) System power efficiency. (c) Ground power in mW. (d) Number of LDO blocks. 136

55 Optimization results of favoring different design aspects. (a) LDO and decap area in mm^2. (b) System power efficiency. (c) Ground power in mW. (d) Number of LDO blocks. 138

56 Optimization results for using planar gate-oxide decoupling capacitance vs. deep trench decoupling capacitance (a) LDO and decap area in mm^2. (b) System power efficiency. (c) Ground power in mW. (d) Number of LDO blocks. 140

57 Optimization results of LDO-decap-BC system co-optimization vs. LDO-decap system co-optimization. (a) LDO and decap area in mm^2. (b) System power efficiency. (c) Ground power in mW. (d) Number of LDO blocks. 142

CHAPTER I

INTRODUCTION

A. Preliminaries

Very Large-Scale-Integration (VLSI) Power Delivery Networks (PDNs), also known as power grids, play the critical role of reliably powering on all on-chip devices [1]. The main functions of a power delivery network include [1] [2] [3]:

- Maintaining stable voltage supply levels to all on-chip devices under all possible chip activities.
- Providing average and peak power demands for the entire chip.
- Providing current return path for signals.

The entire power delivery network mainly consists of a on-board voltage regulation/conversion module, off-chip PDNs (power and ground networks) and on-chip PDNs (power and ground networks). The diagram showing a complete power delivery network is presented in Figure 1 [3]. The current flows from the on-board Voltage Regulator Module (VRM), through the Printed Circuit Board (PCB) PDN, the socket, the package and the C4 bumps to the chip. On the chip, as shown in Figure 2 [1], the current goes from the top metal layers all the way down to the transistors. The current returns to the ground in the path from the on-chip PDN to the PCB. There are off-chip decoupling capacitors (decap) and on-chip decoupling capacitors in the network to reduce the noise caused by fast on-chip switching circuits.

The journal model is *IEEE Transactions on Very Large Scale Integration (VLSI) Systems.*

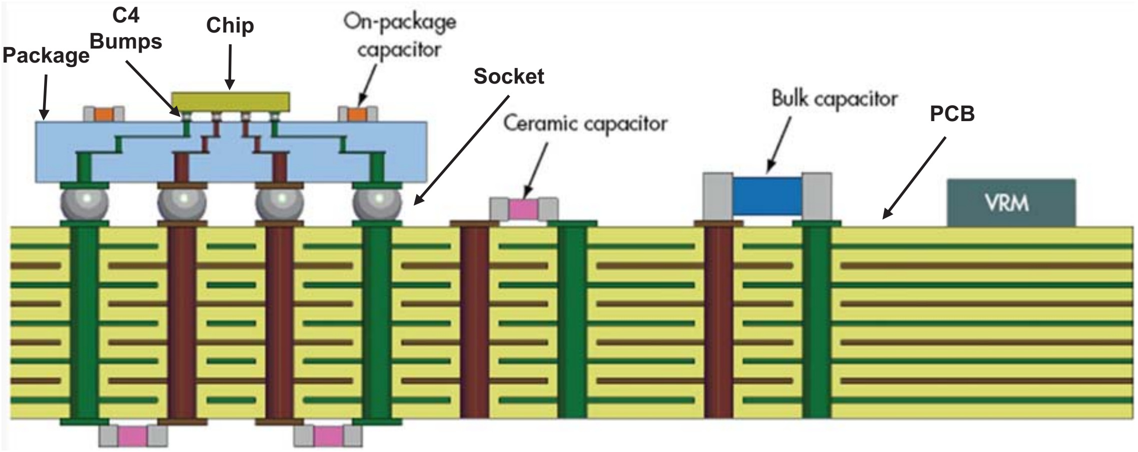

Fig. 1. Diagram of a complete power delivery network.

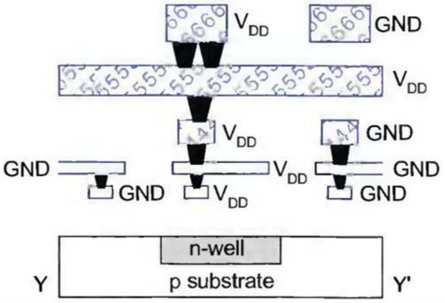

Fig. 2. Diagram of a on-chip power delivery network.

1. Design Perspectives

To design a power delivery network, the following five perspectives should be considered:

- Power: While delivering power to on-chip devices, the power delivery network itself dissipates power. The dissipated power mainly consists of the power loss of the voltage regulation/conversion module (caused by the regulator/converter circuit dynamics and the quiescent current) and the leakage power of the on-chip decoupling capacitance (caused by gate leakage for MOS-based decaps) [2]. Therefore, the power efficiency and the quiescent power of the voltage regulation/conversion module as well as the total amount of on-chip decaps and the on-chip decap technology have to be considered.

- Noise: The power supply noise in the PDN has two components: IR drop and Ldi/dt noise [4]. The IR drop is caused by the resistance between the on-board voltage supply and the on-chip nodes. The Ldi/dt noise is introduced by the inductive parasitics of the network. Since all the on-chip devices reside between the power network and the ground network, both the voltage drop of the power network and voltage overshoot of the ground network have to be considered.

- Reliability: Electromigration (EM) is an effect of having metal ions transported by a direct current flowing through a metal wire in a substantial time period [5]. If this effect is accumulated over a long time period, eventually, it causes the wire to break or to short-circuit to another wire. The rate of the electromigration highly depends on the average current density. Thus, the average current density on each wire segment is required to be checked during design.

- Routing: Enough number of metal wires for the on-chip PDN should be allocated in all metal layers in order to reduce current densities and IR drops [2]. However, on-chip PDNs may use up a lot of signal routing resources. Therefore, the total number and area of the metal wires that are used by the PDN on each layer have to be considered.

- Area: The area cost of the PDN mainly consists of the area used by the voltage regulation/conversion module and the die area taken by on-chip decoupling capacitance. Therefore, the total area for the PDN is constrained by the on-board and on-chip available area.

All the design specifications associated with the above design perspectives must be satisfied in the PDN design. The exact numbers of these design specifications are determined by the circuit functions, CMOS technology, system budgets, etc..

2. Modeling

In order to efficiently and accurately check the performance of PDN designs and analyze the network electric characteristics, a complete model of the power delivery network is built, as presented in Figure 3. The PDN model consists of an off-chip model and an on-chip model [2] [6] [7].

The off-chip model captures the decoupling capacitors and the parasitics of the package and the PCB that reside between the on-chip PDN and the on-board voltage regulator. A variety of distributed models, lumped models and macromodels can be used for the PCB and the package. In this dissertation, a ladder RLC model [6], as shown in Figure 3, is used. The on-board voltage regulator is modeled as an ideal DC voltage source.

The on-chip power delivery network has the following major components: C4 bumps, a VDD grid, a GND grid and on-chip decoupling capacitors. The C4 bumps

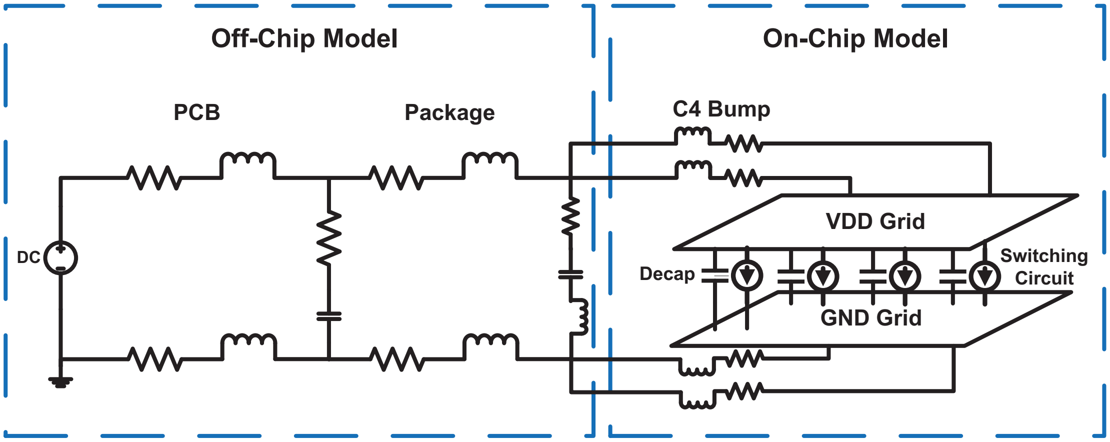

Fig. 3. A complete model of the power delivery network.

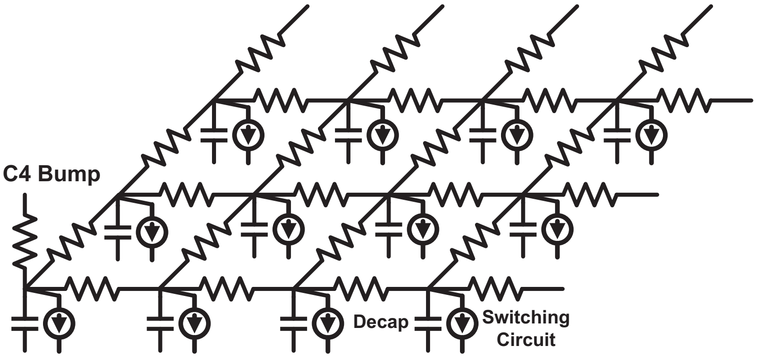

Fig. 4. VDD grid model.

that connect the power/ground grid with off-chip network are modeled as RL pairs [6]. The VDD and GND grids are purely resistive meshes as shown in Figure 4. The decoupling capacitors reside between the VDD grid and the GND grid. The switching circuits are replaced by linear current sources, also called the current loadings, that mimic the current consumption of the circuits [7].

3. Design Trends and Challenges

Due to the recent aggressive process scaling into the nanometer regime and the design trend of pushing the high-performance vs. low-power envelope, the power delivery network has been impacted in a number of perspectives, as presented in Figure 5:

- Firstly, the circuit clock frequency has been increased but the supply voltage has been significantly scaled down, which leads to larger power supply noise and a higher voltage drop percentage with respect to the supply voltage.

- Secondly, with the increased gate density and the reduced chip area, the network complexity becomes much larger and less metal wire resources are available for the power delivery network.

- At last, due to the significant increase of the leakage power and power density in the nanometer regime, many fine-grain power management techniques have been proposed, such as adding sleep transistors to cut the leakage power of unused circuit blocks and employing multiple power supplies for the circuit blocks having different timing and power requirements.

All these impacts have raised more stringent and specific requirements on Electronic-Design-Automation (EDA) tools for power delivery networks, as shown in Figure 5.

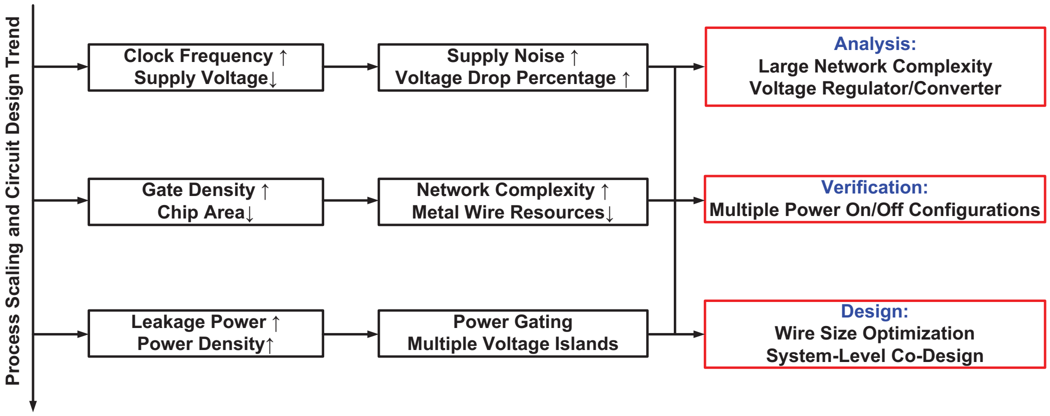

Fig. 5. Design trends and challenges for power delivery networks.

- From the perspective of PDN analysis, simulation techniques must be developed to be able to analyze a on-chip power grid with multi-million nodes or more efficiently and accurately. In addition, techniques that have the capability of tackling the entire power delivery network with an off-chip model and integrated sophisticated/nonlinear devices such as on-chip voltage regulators are also needed.

- For a gated PDN, turning on/off gated grids would create many power gating configurations. Therefore, in terms of performance verification, a scheme needs to be developed to verify whether the gated power delivery network works properly under all possible on/off configurations.

- In terms of PDN design, on one hand, power grid wire sizes must be optimized to save wiring resource for signal routing. On the other hand, due to the sizes of traditional on-board voltage regulators with large inductors or capacitors, there are significant interests in developing fully integrated on-chip voltage regulators to facilitate fine-grain multiple power domains. Hence, systematic analysis on the electric interactions between passive network and active voltage regulators/converters, detailed tradeoff analysis over different design specifications and a system-level co-design scheme that automatically optimizes key design parameters for the entire power delivery network must be provided for PDNs employing on-chip voltage regulation.

B. Survey of Previous Work

In the past several years, there have been a lot of efforts and progresses made on power delivery network analysis, verification and design, however with different tradeoffs.

1. Survey of PDN Analysis

Many approaches have been proposed to address the on-chip power grid analysis problem [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21]. Among these, ideas of employing Cholesky decomposition [9] [15], preconditioned conjugate gradient method [8], multigrid techniques [9] [10] [11] [12] [13] [14], random walks [18], locality [19], relaxation iterative method [20], and Poisson solver optimized for GPU platforms [21] have emerged. However, all these methods, called flat methods, can only be applied to analyze the power grid as an entirety (non-partitioned). As a result, when tackling modern power grid designs with many-million nodes, these methods may suffer from memory overflow and unbearable runtime. At the same time, some of the algorithms are not parallelizable so that they may not fully utilize the increasingly available parallel computing resources to improve efficiency. To overcome these limitations, macromodeling method [15] [16] and non-overlapping domain decomposition method [17], called partitioning methods, are proposed. These methods employ the strategy of *divide-and-conquer* that is realized through grid partitioning. In [15], [16] and [17], the power grids are divided into several partitions or subdomains whose electric properties are represented by the circuit responses on the ports or interface nodes by applying matrix transformation and substitution to each partition or subdomain. However, for the macromodeling method, the final system matrix for the global grid (including the ports) is dense; and for the non-overlapping domain decomposition method, the dominant time is spent on forming the Schur complements. Therefore, although these two methods are naturally parallelizable, when the number of boundary nodes increases, they may suffer from runtime inefficiency.

Moreover, for power delivery networks with active components, such as voltage converters and regulators, existing simulation tools are not capable to handle them since the existing tools can only solve the passive networks. On the other hand, if traditional SPICE-like simulators are used, due to the multi-million-node complexity of the passive network, the simulation could easily run out of memory or take days to get the results.

2. Survey of PDN Verification

PDN verification is a very important but challenging task to chip designers. It can be defined as: under all possible conditions, verify if the PDN can satisfy given electromigration and voltage drop specifications. Traditionally, by considering the current loadings variations (due to carrying out different instructions over the time), several works have been done to find a current loading distribution (current profile) that produces the largest voltage drop [22] [23] [24] [25]. In these works, the loading currents are considered as unknown variables, and the worst-case voltage drops are obtained by formulating a formal optimization problem which is then solved with existing optimization techniques. While some times limited by the capacity of underlying optimization packages, particularly for large power grid designs, these approaches provide a valuable methodology to addressing current loading variation.

Currently, due to the wide adoption of the power gating technique, it is equally important to examine that for the PDN with a given set of current loadings, whether the EM and voltage drop constraints are satisfied under all possible on/off configurations and on/off transitions. In this kind of verification, called the power gating verification, the current waveforms are treated as known. In [26], a useful DC EM analysis approach for the global grid is proposed which can efficiently compute exact DC currents for all possible power gating configurations. Whereas, in the more complicated transient verification of gated power grid networks, a number of complications arise, such as new transient noise behaviors, the handling of transient superposition under multiple sleep transistors per local grid, the need to verify both the global and local grids, and the handling of decoupling sharing effect between local grids. On the other hand, since there is a very large verification space that consists of all possible power gating configurations and transitions, a brute-force exhaustive enumeration over all possible conditions is impractical. For example, a multi-core design with 16 local power grids has 2^{16} possible power gating configurations.

3. Survey of PDN Design

Despite the progresses made on analysis and verification, it is equally important to address the design and optimization issues of such large networks that are even more challenging. In [27], sequential linear programming based approach is proposed to improve the efficiency of flat power grid wire size optimization. Multigrid-like heuristic is proposed to reduce the complexity of large power optimization in [28]. In [29], the macromdeling idea of [15] and [16] is adopted to facilitate partitioning-based optimization. In principle, flat optimization is only limited to small or medium sized power grids. It may be impractical for a complete large grid due to extensive runtime and memory requirement. Multigrid-like heuristic improves the optimization efficiency by operating on significantly simplified coarse grids. Whereas, design constraints may not be exactly satisfied during the approximation step [28]. The macromodeling based approach provides a nice feature of incremental optimization that allows for individual optimization of one partition at a time [29]. However, partitioning in the mesh-like power grid structure creates a large number of interface nodes at the partitioning boundaries. It produces large and dense macromodels expensive to compute. Furthermore, the optimization of the entire grid requires a sequential optimization of all partitions. It is important to note, the above approaches are not ready for simultaneous optimization of multiple partitions, hence, cannot be immediately parallelized.

Moreover, a great amount of effort has been geared towards developing fully integrated on-chip voltage regulators/converters. DC-DC converters are considered to be power efficient even when the input-to-out voltage difference is large [30] [31]. Therefore they are widely used for Dynamic Voltage Scaling (DVS). However, to fully integrate DC-DC converters on the chip, designing an area-efficient inductor at the converter output becomes the major obstacle. On the other hand, Low-Dropout regulators (LDOs) are more amenable for on-chip integration due to their small area overhead, low standby current, low dropout voltage, improved power efficiency and superior transient response to fast load current variation [32] [33] [34] [35]. Fully integrated LDOs are very attractive for regulating large and fast local voltage fluctuations and for providing multiple levels of supply voltage [32] [33] [34] [35]. They can be used as post-regulators following switching converters (with high power conversion efficiency) to provide low-noise supply voltage while maintaining good overall power efficiency [36] [37]. On the other hand, passive power delivery network design, for example through the means of decoupling capacitance insertion, has also been the subject of many researchers [28] [38]. However, so far these two threads of work are

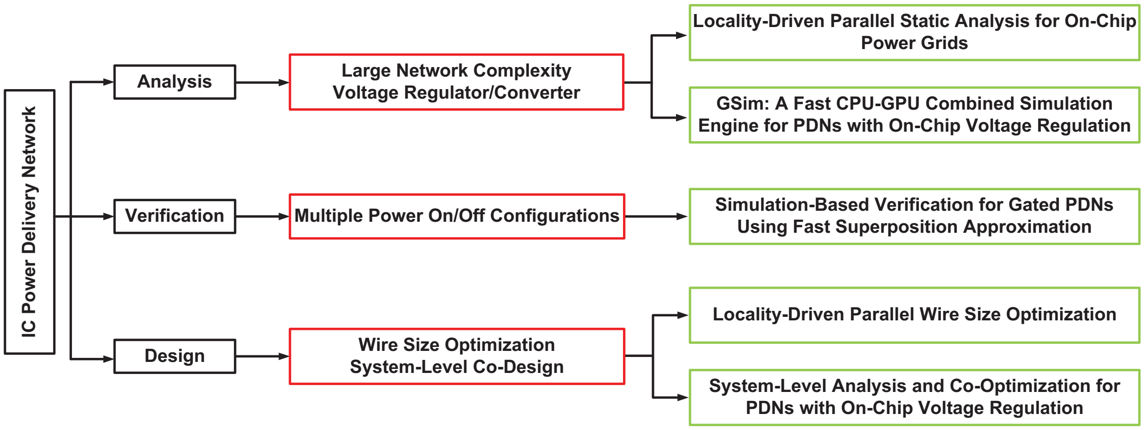
IC Power Delivery Network
Analysis
Verification
Design
Large Network Complexity
Voltage Regulator/Converter
Multiple Power On/Off Configurations
Wire Size Optimization
System-Level Co-Design
Locality-Driven Parallel Static Analysis for On-Chip Power Grids
GSim: A Fast CPU-GPU Combined Simulation Engine for PDNs with On-Chip Voltage Regulation
Simulation-Based Verification for Gated PDNs Using Fast Superposition Approximation
Locality-Driven Parallel Wire Size Optimization
System-Level Analysis and Co-Optimization for PDNs with On-Chip Voltage Regulation

Fig. 6. Work towards scalable analysis, verification and design for IC power delivery.

disjointed. Voltage regulator/converter design is typically done in isolation with an assumed simple capacitive load; existing passive PDN optimization work does not consider active regulator/converter circuits. Little work has been geared towards understanding the detailed electric characteristics of having multiple on-chip voltage regulators operate inside a large power delivery network.

C. Proposed Solutions

In the dissertation, to overcome the limitations of existing work on power delivery network analysis, verification and design, new methodologies and approaches are proposed as presented in Figure 6.

1. Proposed Solutions on PDN Analysis

For the on-chip power grid static analysis, by focusing on the C4 flip-chip type power grids, a novel parallel partitioning based power grid simulation method is proposed. In this approach, the power grid is divided into several partitions. For each partition, the impact of the rest of the grid is modeled as the currents flowing into that partition. Using these currents, called the boundary currents, each partition is solved independently. Thus the entire circuit responses can be obtained in parallel. An efficient and effective boundary current approximation scheme using spatial locality of the flip-chip power grids is introduced to provide near-exact current values on the boundary. This scheme only requires to solve several small-size power grids to get the approximations. Thus would not jeopardize the performance when the number of boundary nodes increases. Errors can be reduced quickly by using a block-based iterative process which employs the same boundary current approximation scheme. As a result, the proposed approach not only has the feature of natural parallelization and the ability to tackle large power grids, but also can achieve excellent runtime efficiency and partitioning flexibility. In addition, by looking into the main factors that affect the parallel performance and conducting extensive experimental studies, detailed computational cost analysis and performance modeling are provided. In addition, we propose a strategy that helps users determine the optimal (or near-optimal) values of some key parameters to achieve the lowest parallel runtime.

On the other hand, we address the significant challenge in simulating complex PDNs with a large number of integrated LDOs at SPICE-level accuracy by developing an integrated CPU-GPU analysis engine: GSim. Our engine achieves its efficiency through circuit partitioning and the integration of linear iterative, linear direct and nonlinear solvers running on Graphics Processing Unit (GPU) and Central Processing Unit (CPU) respectively. These solvers are optimized for large on-chip power grids, off-chip models and transistor-level LDO models, respectively.

2. Proposed Solutions on Power-Gated PDN Verification

In the dissertation, by focusing on power gating verification, we propose a practical simulation-based approach that verifies the complete power delivery hierarchy (local and global grids), under all possible stable power gating configurations and core power-on noise injection in terms of EM and voltage drops. The proposed approach achieves efficient verification by fast equivalent circuit modeling and superposition methods for approximate and conservative identification of worst-case violations in the large verification space. A few selective full simulations are carried out for validation.

3. Proposed Solutions on PDN Wire Sizing and On-Chip Voltage Regulation Design

In terms of wire size optimization, by focusing on the C4 flip-chip type power grids, we take the same basic partitioning philosophy to achieve scalability for large power grid optimization, but via a rather different avenue. Although applying partitioning seems to be rather natural for attacking large mesh-like circuits such as power grids, its proper employment under the context of constrained optimization is nontrivial. Simply neglecting the coupling along the partitioning boundaries can easily lead to a large number of IR drop and EM violations, preventing any effective partitioning based optimization. We address such a challenge by taking a different route to the partitioning based optimization. We first re-formulate the original (flat) constrained power grid optimization problem into a two-level optimization problem. This new two-level hierarchical formulation is built upon the essential idea of partitioning based optimization and has an appealing form that seemingly enables *divide-and-conquer*-based scalable optimization. Motivated by this hierarchical view, we develop practical solutions to address the fundamental limitations of the hierarchical approach in terms of convergence and efficiency which lead to a locality-driven two-step optimization. One key feature of the proposed approach is that it is fully parallelizable since our algorithm construction permits simultaneous sizing of an arbitrary selection of partitions, including those that are adjacent to each other. This offers the important ability of utilizing the increasing parallel computing hardware to address the challenges in power grid design. The performance of the proposed approach is largely independent of the choice of partitioning boundaries, hence is not constrained by design hierarchy. As a result, the partitioning boundaries, size and number of partitions can be flexibly chosen to tradeoff between runtime and memory requirements as well as to facilitate load balancing in parallelization. It shall also be noted that our power grid optimization approach is general and it does not depend on a specific choice of underlying numerical optimization packages used.

Moreover, efforts are spent towards understanding the benefits and detailed electric characteristics of on-chip LDOs under the large power delivery network context. An attempt is also made to link the regulator/converter design together with passive decoupling capacitance insertion, which targets the critical joint co-optimization of active and passive components so as to achieve the optimal performance for the entire PDN design. To achieve our goal, we first conduct systematic design analysis to describe the analog characteristics of voltage regulators under a network context and use it as the basis to understand the interactions between the active voltage regulation and the passive decoupling. The derived design insights are employed to facilitate a system-level co-design in which key regulator/converter parameters, the number of on-chip regulators, and the amount of decap inserted are considered as design variables. To feasibly optimize large PDNs, we leverage a custom fast simulation environment, a multi-level based optimization strategy as well as design knowledge to develop an automatic optimization flow. Using our optimization approach, we demonstrate huge benefits of system-level co-optimization that involves both active voltage regulation/conversion optimization and passive power grid optimization. The tradeoffs between different design specifications, such as area, power, placement and routing, and noise, are presented. We also analyze the impact of decoupling technologies (such as deep-trench decaps) on the design of power delivery.

CHAPTER II

POWER DELIVERY NETWORK ANALYSIS AND VERIFICATION [*]

As stated in Chapter I, the PDN analysis faces the challenges of large network complexity (large runtime and memory consumption) and the integration of nonlinear devices such as on-chip voltage regulators with the passive network (existing power grid solvers are not applicable and the general SPICE is too slow), while the verification for gated PDNs has a large verification space (exhaustive simulation-based verification is impractical). To address these challenges, in this chapter, a novel parallel partitioning-based static analysis approach for large on-chip power grids is first presented. Then, a fast CPU-GPU combined simulation engine that can efficiently analyze complex power delivery networks with sophisticated on-chip voltage regulators is introduced. At last, an efficient simulation-based verification methodology using an effective circuit modeling and a fast superposition approximation is illustrated.

A. Locality-Driven Parallel Static Analysis for Power Delivery Networks

In this section, by employing the *divide-and-conquer* methodology and the locality property of the flip-chip type power grids, a partitioning-based parallel power grid static analysis approach is proposed to reduce the excessive runtime and heavy workload caused by the large power delivery network complexity.

[*] Part of this chapter is reprinted with permission from "Locality-driven parallel static analysis for power delivery networks" by Z. Zeng, Z. Feng, P. Li and V. Sarin, 2011. *ACM Trans. on Design Automation of Electronic Systems*, Vol. 16, pp. 28:1-28:17, Copyright [2011] by ACM.

1. Background

The static analysis is the most fundamental analysis for power delivery networks. It is widely used to detect potential EM failures (wires with large average currents) and hot spots (nodes with large voltage drops). For the static analysis, the PDN is simply a resistive network which can be further divided into two completely separated grids by splitting the current sources [39]. For simplicity, only the VDD grid static analysis is discussed in this dissertation. The GND grid can be solved in the same way. Assume the power grid has N nodes, using the Modified Nodal Analysis (MNA), the system equation can be represented as [9] [15]

$$\mathbf{G}\mathbf{V} = \mathbf{I}, \tag{2.1}$$

where $\mathbf{G} \in \mathbb{R}^{N \times N}$ is the conductance matrix, $\mathbf{V} \in \mathbb{R}^{N \times 1}$ is the vector for node voltages, and $\mathbf{I} \in \mathbb{R}^{N \times 1}$ is the vector for current sources and voltage supplies. For modern power grid designs, N can be multimillion. Such a system can be solved by direct methods i.e. LU or Cholesky factorization [9] [15]. Other methods, such as preconditioned conjugate gradient method [8], multigrid techniques [9] [10] [11] [12] [13] [14], macromodeling method [15] [16], non-overlapping domain decomposition method [17], random walks [18], locality [19], relaxation iterative method [20], and a Poisson solver optimized for GPU platforms [21], have also been proposed to solve (2.1).

2. Overview of the Proposed Approach

To apply the *divide-and-conquer* methodology to the power grid static analysis as shown in (2.1), the power grid $\mathbf{\Omega}$, represented by the system matrix $\mathbf{G}$, is divided into K partitions $\Omega_1, \cdots, \Omega_K$ (as shown in Figure 7, $K = 16$ in this case), and the system equation can be expressed as

$$\mathbf{GV} = \begin{bmatrix} G_1 & G_{12}^T & \dots & G_{1K}^T \\ G_{12} & G_2 & \dots & G_{2K}^T \\ \vdots & & & \vdots \\ G_{1K} & G_{2K} & \dots & G_K \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_K \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_K \end{bmatrix}, \tag{2.2}$$

where $G_1, \cdots, G_K$ are the conductance matrices for partitions $\Omega_1, \cdots, \Omega_K$. G_{ij} represents the connections between partition Ω_i and partition Ω_j ($i, j = 1, \ldots, K$; $i \neq j$). V_i and I_i are the node voltage vector and current loading vector for partition Ω_i ($i = 1, \ldots, K$).

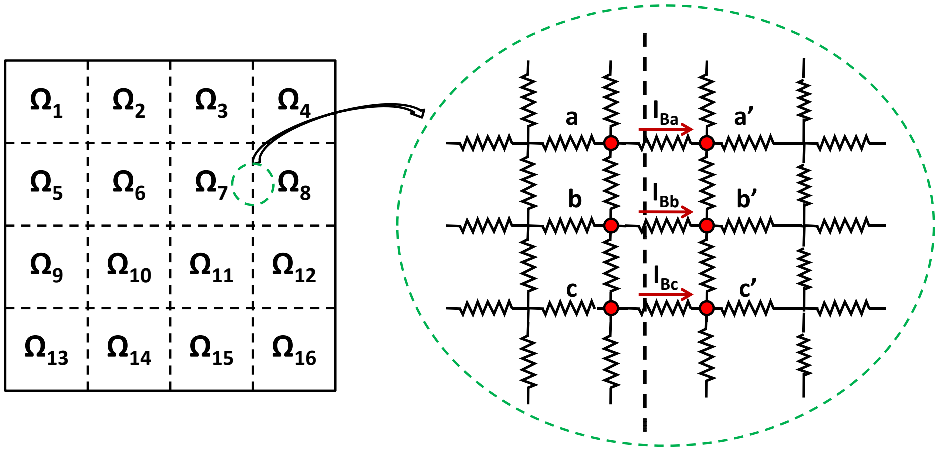

Fig. 7. Power grid partitions and partition boundaries.

By moving the off-diagonal terms into the right-hand side, (2.2) becomes

$$\begin{aligned}\begin{bmatrix} G_1 & & & \\ & G_2 & & \\ & & \ddots & \\ & & & G_K \end{bmatrix}\begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_K \end{bmatrix} &= \begin{bmatrix} I_1 - G_{12}^T V_2 \ldots - G_{1K}^T V_K \\ I_2 - G_{12} V_1 \ldots - G_{2K}^T V_K \\ \vdots \\ I_K - G_{1K} V_1 \ldots - G_{(K-1)K} V_{K-1} \end{bmatrix} \\ &= \begin{bmatrix} I_1 - I_{B1} \\ I_2 - I_{B2} \\ \vdots \\ I_K - I_{BK} \end{bmatrix}, \end{aligned} \tag{2.3}$$

where $I_{Bi} = \sum_{k=1}^{i-1} G_{ki} V_k + \sum_{k=i+1}^{K} G_{ik}^T V_k$, $i = 1, \ldots, K$. I_{Bi} is the vector of currents flowing from Ω_i to other partitions, called boundary current. As shown in Figure 7, the currents I_{Ba}, I_{Bb}, and I_{Bc} are the boundary currents for Ω_7 (-I_{Ba}, -I_{Bb}, and -I_{Bc} are the boundary currents for Ω_8). Without explicit boundary currents, the partition simulations are messy and highly coupled with each other, which are difficult to be parallelized. However, once the boundary currents are obtained, each V_i can be calculated by solving $G_i V_i = I_i - I_{Bi}$ independently which leads to straightforward parallelization. The methods in [15], [16] and [17] spend a significant amount of computation to obtain the exact boundary currents through dense matrix factorization and Schur complement formation. Thus their runtime efficiencies are limited by expensive boundary current calculations.

We address the limitations of the existing partitioning-based simulation methods, runtime efficiency and partitioning flexibility, by adopting a novel approach to provide near-exact approximations to the boundary currents. For clarity, assuming the power grid is partitioned in the way illustrated in Figure 7, the flow of the proposed approach is shown as follows.

- Step 1: Obtain near-exact approximated boundary currents $I_{B1}^*, \ldots, I_{BK}^*$ in parallel.
- Step 2: Use $I_{B1}^*, \ldots, I_{BK}^*$ to solve (2.3) in parallel.
- Step 3: Residues are computed to form the new right-hand side of the matrix equation for the full grid. Repeat step 1, step 2 and step 3 until the convergence is reached.

Although, in step 1, called the boundary current approximation step, it is ideal to find the exact boundary currents at the first place, they may only be available after the full system or at least the system consisting of all the boundary nodes is solved. However, by exploiting the strong locality behavior in C4 flip-chip-type power grids [19], it can be shown that near-exact approximations could be efficiently obtained without solving large complex systems. According to the same locality property, I_{Bi}^*s can be computed by solving a set of uncoupled local grid simulation problems, leading to an immediate parallel implementation, as detailed in Section II.A.3.a. In step 2 (called the partition simulation step), a set of partition grids are analyzed in parallel, since they are shielded to the rest of the system by the boundary settings of using the approximate boundary currents I_{Bi}^*s, as illustrated in Section II.A.3.b. Errors would occur due to the inaccuracy of I_{Bi}^*s. In step 3 (called the error reduction step), those errors are computed and fed back to the original grid to solve for the correction components. Instead of using the block Jacobi iteration process in [40], the analysis schemes in step 1 and step 2 are employed to correct the circuit responses. It can be shown that by including step 1 into this error reduction step to update the boundary information, analysis convergence can be significantly improved.

In summary, the convergence and runtime efficiency of the proposed approach are achieved by finding near-exact boundary currents efficiently. In this dissertation, we restrict our discussion on using the Cholesky factorization method for each grid solve. However, it shall be noted that in principal any power grid analysis method can be applied to carry out the boundary currents approximations and partition simulations. Hence, the proposed approach is generic and formulated purely based upon the nature of the application.

3. Parallel Locality-Driven Static Analysis

In this section, we first describe the parallel boundary current approximation, the parallel partition simulation, and the block-based iterative error reduction scheme using spatial locality in detail. Then, the overall flow of the proposed method is presented. At last, computational cost analysis is provided to identify the key factors that affect the solution process and parallel performance modeling is carried out to determine the optimal (or near-optimal) values of the key parameters for the most efficient runtime.

a. Parallel Boundary Current Approximation Using Locality

In modern chip designs, C4 flip-chip packaging technology is commonly used which provides a large number of V_{DD}/V_{GND} connections evenly spreading out in the on-chip power delivery networks. In a local region, due to the existence of many C4 bump-s, the majority of currents are supplied through the low impedance paths from/to the nearby V_{DD}/V_{GND} pads. Hence, the local voltage responses are largely dependent on the V_{DD}/V_{GND} connections, wire resistances, and current loadings in the neighborhood, exhibiting strong spatial locality. For example, in Figure 8, a power grid is divided into nine partitions, and node A is at the center of partition 5. Then, the voltage response at A is primarily determined by C4 bumps (red dots), current loadings, and wire resistances in the same partition, and is less influenced by the elements

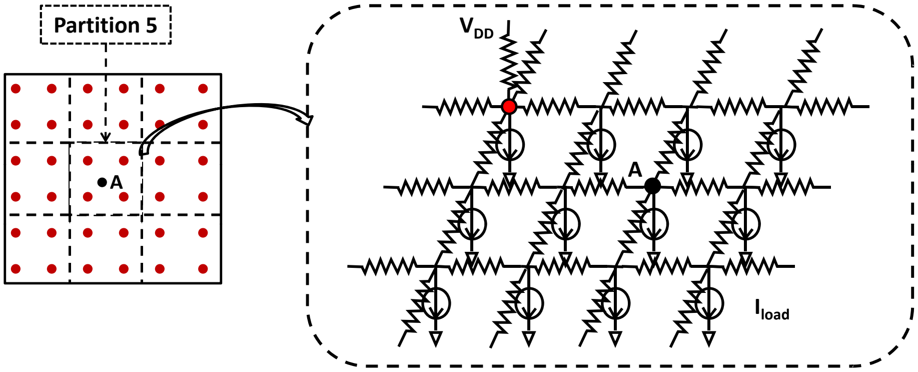

Fig. 8. Spatial locality in power grids. Red dots represent C4 connections.

in other partitions. When the size of partition 5 is large enough so that the impacts of other partitions are negligible, the circuit response for node A obtained by only simulating partition 5 (the circuit elements in other partitions are neglected) would be close to the exact circuit response obtained by analyzing the entire power grid. An overlapping power grid shell-based partitioning method has employed the spatial locality to accelerate the power grid analysis and has shown favorable effectiveness for solving industrial power grid designs [19].

In this work, the spatial locality is employed for finding near-exact boundary currents for individual partition simulations. As shown in Figure 9, the basic idea is to introduce a window to enclose each partition boundary (the black dash line) at the window center. The size of the window is made large enough to include a ring of C4 bumps around the partition boundary. Then, we are only focusing on the circuit elements in the window and neglecting all other circuit elements outside of the window. After solving the truncated circuit in that window, the currents on the partition boundary are retained as the approximate boundary currents. Usually, near-exact boundary currents could be obtained when the size of the window is made large enough to include a sufficient number of C4 bumps such that the grid outside

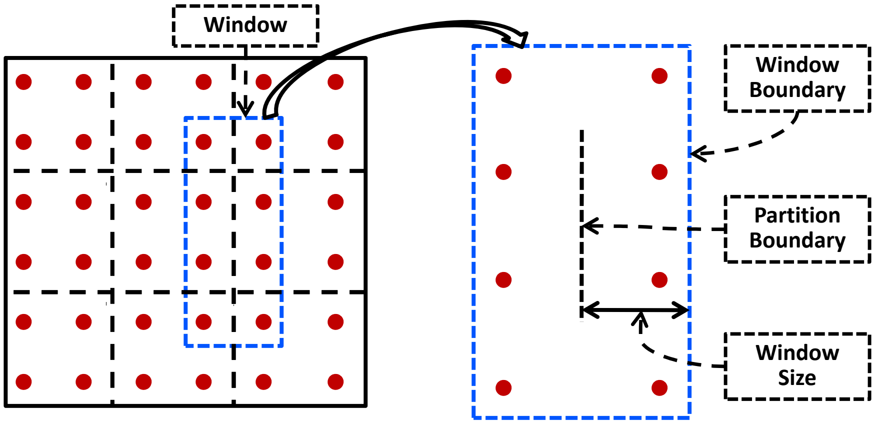

Fig. 9. Finding the near-exact boundary currents via locality. Red dots represent C4 connections. Black dash lines represent partition boundaries. Blue dash lines represent window boundaries.

of the window has negligible influence to the circuit responses of the boundary nodes. As expected, by using these near-exact boundary currents, the errors introduced in the partition simulations are small, which only requires a few iterative correction steps to be reduced to an acceptable level. Since the approximation for each set of boundary currents is determined by solving an independent partial grid problem, the entire procedure can easily be parallelized.

The window size is defined as the number of nodes the partition boundary away from the window boundary, as shown in Figure 9. Assume the partition boundary has l node and the window size is s, then, the window incloses $2s \times (l + 2s)$ nodes. Since the approximate boundary current values are largely dependent on the number of C4 connections in the window, we introduce another term: C4 ring size, to describe the size of the window. The C4 ring size is the number of C4 connections existing between the partition boundary and window boundary. For example, in Figure 10, the C4 ring sizes for the three windows are 1, 2, 3, respectively. A 1-million-node power grid with C4 flip-chip packaging is used as an example to illustrate the spatial

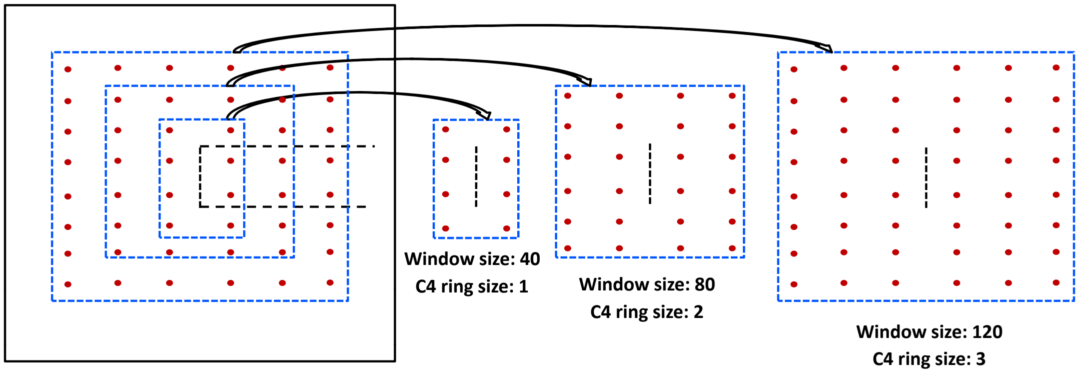

Fig. 10. Impact of window size on boundary current approximation. Red dots represent C4 connections. Black dash lines represent partition boundaries. Blue dash lines represent window boundaries.

locality. The C4 bumps are evenly distributed in the grid with 40-node distance away from each other as shown in Figure 10. The power grid is divided into four equally sized partitions. A boundary is examined as an example. The IR drops and branch currents of a boundary node obtained via the boundary window simulation using different window sizes are shown in Table I, and they are compared with the exact IR drop and current solutions. The average IR drops and branch currents for the same boundary are also examined. Note the quick convergence of the result. The convergence is reached at the window size of 120, corresponding to the C4 ring size of 3.

b. Parallel Partition Simulation Using Approximated Boundary Currents

As presented in Section II.A.3.a, the approximated boundary currents I^*_{Bi}s are computed in step 1 of the proposed approach. In step 2, each power grid partition is solved using I^*_{Bi}, which is added to the boundary nodes as extra current loadings. Using the same notations presented in Section II.A.3.a, the system equation for the partition simulation can be expressed as

Table I. Comparison of boundary IR drops and currents obtained in boundary window simulations using different window sizes with the exact solutions. Voltage drop is in mV. Current is in mA.

Window Size	C4 Ring Size	Node Drop	Node Current	Avg. Drop	Avg. Current
40	1	55.693	0.9712	60.850	0.470
80	2	51.366	1.195	54.287	0.513
120	3	50.474	1.245	53.195	0.490
Exact	4	50.469	1.278	53.280	0.490

$$\begin{bmatrix} G_1 & & & \\ & G_2 & & \\ & & \ddots & \\ & & & G_K \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_K \end{bmatrix} = \begin{bmatrix} I_1 - I_{B1}^* \\ I_2 - I_{B2}^* \\ \vdots \\ I_K - I_{BK}^* \end{bmatrix}, \tag{2.4}$$

where $G_1, \ldots, G_K$ are the conductance matrices of the partitions of $\Omega_1, \ldots, \Omega_K$; $V_1, \ldots, V_K$ are the internal partition node voltage vectors; $I_1, \ldots, I_K$ are the internal partition current loading vectors; and $I_{B1}^*, \ldots, I_{BK}^*$ are the approximate partition boundary currents.

In Figure 11, the partitioning of a grid along a vertical line is illustrated as an example. Starting from the initial grid, the approximated boundary currents (I_{Bi}^*s) are computed, as shown on the left of the figure. The grid is then partitioned into two. For each partition, the boundary currents are set to I_{Bi}^*s by attaching ideal current sources to the corresponding boundary nodes, as shown in the middle of the figure. Each partition is simulated by including these additional current sources on the boundary. At last, the node voltages for different partitions are retained to form the complete solution, as shown on the right of the figure. Since the approximate boundary currents may not be exact, after assembling partition simulation results (node voltages), the KCL equations for the boundary nodes might not be satisfied. The residues on the boundary nodes would lead to the errors of internal nodes of each partition, thus cause global errors throughout the entire grid. Since the boundary currents shield each partition from the rest of system, the simulation for each partition has no coupling with others (there are no off-diagonal matrix blocks in (2.4)), thus can be easily parallelized.

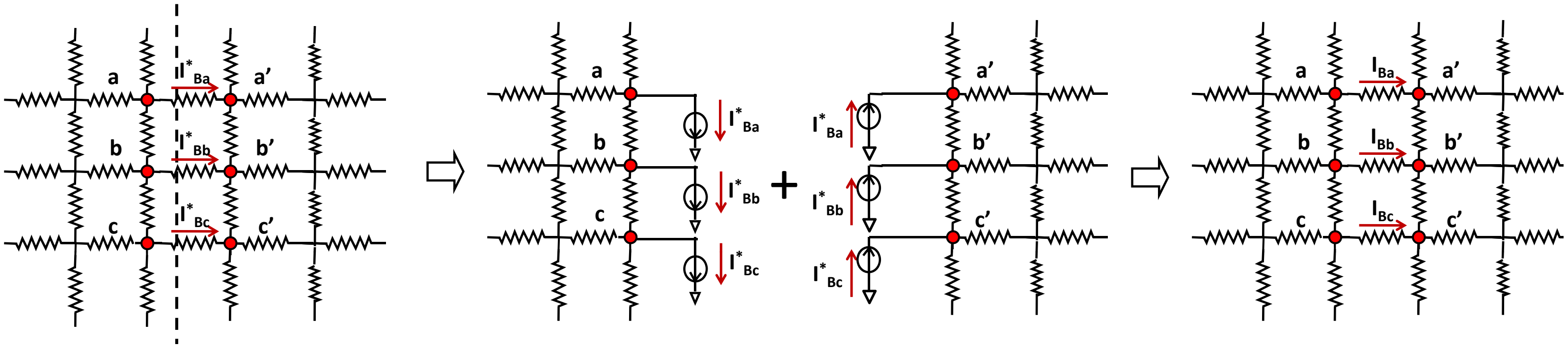

Fig. 11. Power grid partition simulation by setting boundary currents.

c. Block-Based Iterative Error Reduction

As stated in previous sections, inaccurate boundary currents always cause errors in the solution for the entire system. Although larger window size could always be chosen to provide more accurate boundary currents, it would cost longer simulation time and use more computing resources, like memory. Therefore, an effective and efficient error reduction scheme is indispensable to control the errors in a reasonable level while maintaining the runtime efficiency.

A block-based iterative scheme using locality is proposed to reduce errors. After obtaining the full system solution, residues are computed and set as the new current loadings to the original grid to form a correction grid. Unlike the traditional block Jacobi iteration scheme [40], which always sets boundary currents to zero in the error correction grid, the proposed process employs the boundary currents approximation scheme in step 1 to provide a large coupling region to obtain more accurate boundary currents, thus reduce the errors more rapidly. Then the correction components of node voltages are computed in step 2. These newly obtained correction components are added to previous solutions, and the residues are computed again for the next error reduction iteration, until the convergence is reached.

The system matrix equation for the kth iteration, the residue expression after $k-1$th iteration, and the updated node voltage after kth iteration are expressed as

$$\mathbf{G}\mathbf{V}_{res}^{k} = \mathbf{I}_{res}^{k} \Longrightarrow \begin{bmatrix} G_1 & G_{12}^T & \dots & G_{1K}^T \\ G_{12} & G_2 & \dots & G_{2K}^T \\ \vdots & & & \vdots \\ G_{1K} & G_{2K} & \dots & G_K \end{bmatrix} \begin{bmatrix} V_{1,res}^k \\ V_{2,res}^k \\ \vdots \\ V_{K,res}^k \end{bmatrix} = \begin{bmatrix} I_{1,res}^k \\ I_{2,res}^k \\ \vdots \\ I_{K,res}^k \end{bmatrix}, \qquad (2.5)$$

$$\mathbf{I}_{res}^{k} = \begin{bmatrix} I_{1,res}^{k} \\ I_{2,res}^{k} \\ \vdots \\ I_{K,res}^{k} \end{bmatrix} = \begin{bmatrix} I_1 - G_1 V_1^{k-1} - \ldots - G_{1K}^T V_K^{k-1} \\ I_2 - G_{12} V_1^{k-1} - \ldots - G_{2K}^T V_K^{k-1} \\ \vdots \\ I_K - G_{1K} V_1^{k-1} - \ldots - G_K V_K^{k-1} \end{bmatrix}, \tag{2.6}$$

$$\mathbf{V}^{k} = \begin{bmatrix} V_1^{k} \\ V_2^{k} \\ \vdots \\ V_K^{k} \end{bmatrix} = \begin{bmatrix} V_1^{k-1} + V_{1,res}^{k} \\ V_2^{k-1} + V_{2,res}^{k} \\ \vdots \\ V_K^{k-1} + V_{K,res}^{k} \end{bmatrix}. \tag{2.7}$$

Noted that $\mathbf{V}^0 = 0$ and $\mathbf{I}_{res}^0 = \mathbf{I}$. Obviously, the circuit conductance matrices for the boundary windows and partitions remain the same throughout the iterations, and only the right-hand sides of the system equations are updated. Therefore, those matrices can be pre-formulated and stored locally for fast resolving at each iteration. From the classic iterative method point of view, step 1 and step 2 form a block Gauss-Siedel iteration. Step 1 updates the solutions for boundary currents, and Step 2 updates the solutions for internal nodes. Therefore, the convergence of this iterative error reduction scheme can be guaranteed.

d. Algorithm Flow and Implementation

Finally, we summarize the entire flow of the proposed approach in Algorithm 1. Assuming the original power grid $\mathbf{\Omega}$ is divided into K partitions $\Omega_1, \ldots, \Omega_K$ with L partition boundaries $B_1, \ldots, B_L$. L boundary windows $\Omega_{W1}, \ldots, \Omega_{WL}$ are created with the window size s. The conductance matrices for the partitions are $G_1, \ldots, G_K$, and the conductance matrices for those boundary windows are $G_{W1}, \ldots, G_{WL}$. The error tolerance ε is set to check the convergence. The algorithm flow is presented in Figure 12.

Algorithm 1 Parallel Locality-Driven Method

Input: Partition conductance matrices $G_1, \ldots, G_K$, window conductance matrices $G_{W1}, \ldots, G_{WL}$, partition current loadings $I_1, \ldots, I_K$, window current loadings $I_{W1}, \ldots, I_{WL}$, error tolerance ε.

Output: Node voltage $\mathbf{V}$.

```
$k = 0$;
$\mathbf{V}^0 \leftarrow 0$, $\mathbf{I}^0_{res} \leftarrow \mathbf{I}$;
while $||I^k_{res}||_2 > \varepsilon$ do
    for $i \leftarrow 1$ to $L$ par do
        Solve $G_{Wi} V^k_{Wi,res} = I^k_{Wi,res}$;
    end for
    Store $I^{*k}_{B1,res}, \ldots, I^{*k}_{BK,res}$;
    for $i \leftarrow 1$ to $K$ par do
        Solve $G_i V^k_{i,res} = I^k_{i,res} - I^{*k}_{Bi,res}$;
        $V^k_i \leftarrow V^{k-1}_i + V^k_{i,res}$;
        $I^{k+1}_{i,res} \leftarrow I_i - \left(\sum_{j=1}^{i-1} G_{ji} V^k_j + G_i V^k_i + \sum_{j=i+1}^{K} G^T_{ij} V^k_j\right)$;
    end for
    $k \leftarrow k + 1$;
end while
$\mathbf{V} \leftarrow \{V^{k+1}_1, \ldots, V^{k+1}_K\}$;
```

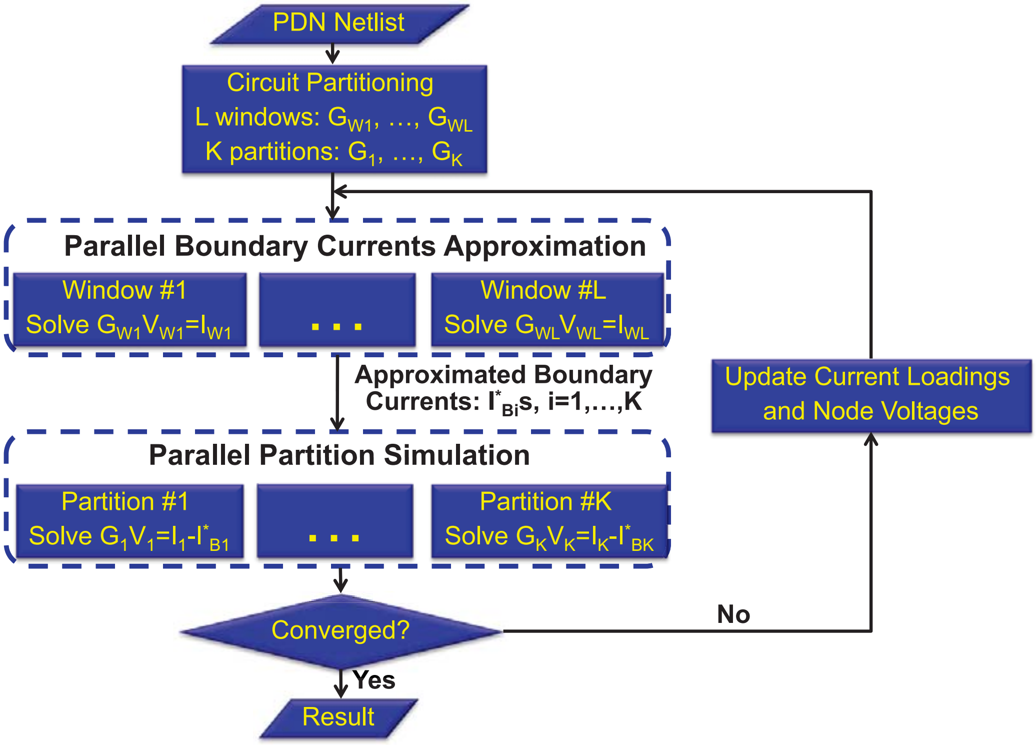
PDN Netlist
Circuit Partitioning
L windows: G_{W1}, …, G_{WL}
K partitions: G_1, …, G_K
Parallel Boundary Currents Approximation
Window #1
Solve $G_{W1}V_{W1}=I_{W1}$
. . .
Window #L
Solve $G_{WL}V_{WL}=I_{WL}$
Approximated Boundary
Currents: I^*_{Bi}s, i=1,…,K
Parallel Partition Simulation
Partition #1
Solve $G_1V_1=I_1-I^*_{B1}$
. . .
Partition #K
Solve $G_KV_K=I_K-I^*_{BK}$
Converged?
No
Yes
Result
Update Current Loadings
and Node Voltages

Fig. 12. Algorithm flow of the proposed parallel locality-driven method.

The *divide-and-conquer* strategy of this approach makes it able to take advantage of the increasingly available parallel computing resources such as multicore machines and distributed computing network systems. For multimillion (less than 10 million) node grids, which can be stored in the memory a single multicore machine (shared memory), a significant amount of runtime improvement can be obtained by running the proposed approach in parallel. However, for many-million (over 10 million) node grids, which cannot fit into the memory of a single machine, it is favorable to use distributed computing networks. The original grid can be divided into several window grids and partition grids at first, so that each grid is small enough for an efficient solve on a single machine. Then those grids are stored and analyzed at different machines of the distributed network locally. The only communication through the network is to feed the approximate boundary currents to partition simulations and to update adjacent boundary node voltages of partition boundaries for residue computation. Although we have not implemented the proposed approach on the distributed computing networks, its performance is expected to be promising when handling many-million node grids.

e. Computational Cost Analysis

In this section, we present the computational cost analysis for the proposed approach and the trade-offs between total runtime, the number of partitions, and the window size.

Assume the power grid $\mathbf{\Omega}$ (with N nodes) is divided into K partitions $\Omega_1, \ldots, \Omega_K$ and L boundary windows $\Omega_{W1}, \ldots, \Omega_{WL}$. The numbers of nodes in the partitions and windows are $n_1, \ldots, n_K$ and $m_1, \ldots, m_L$, respectively. Suppose the cost of Cholesky factorization for a sparse matrix with dimension $n \times n$ is $C_1(n)$, the cost of solving the system with the Cholesky factor is $C_2(n)$, and the cost of matrix formulation is $C_3(n)$. Assume there are d iterations being carried out, the overall cost of the sequential implementation of the algorithm would be

$$\begin{aligned}\mathscr{C}_s = &\sum_{i=1}^{K} C_1(n_i) + \sum_{i=1}^{L} C_1(m_i) + d\left(\sum_{i=1}^{K} C_2(n_i) + \sum_{i=1}^{L} C_2(m_i)\right)\\ &+ \sum_{i=1}^{K} C_3(n_i) + \sum_{i=1}^{L} C_3(m_i),\end{aligned} \tag{2.8}$$

where $\mathscr{C}_s$ is the overall sequential cost; $N = \sum_{i=1}^{K} n_i$; $\sum_{i=1}^{K} C_1(n_i)$ is the Cholesky factorization time for the partitions; $\sum_{i=1}^{L} C_1(m_i)$ is Cholesky factorization time for the windows; $\sum_{i=1}^{K} C_2(n_i)$ is the triangular solve time for partitions; $\sum_{i=1}^{L} C_2(m_i)$ is the triangular solve time for windows; $\sum_{i=1}^{K} C_3(n_i)$ is the matrix formulation time for partitions; and $\sum_{i=1}^{L} C_3(m_i)$ is the matrix formulation time for windows. If no partitioning is applied to the power grid, the cost for solving the entire grid is $C_1(N)+C_2(N)+C_3(N)$. For a sparse matrix arising from a grid with n nodes, it is known that $C_1(n) = O(n^{1.5})$, $C_2(n) = O(n \log n)$, and $C_3(n) = O(n)$, which is superlinear in n [41]. For extreme large power grids, by careful selection of the window size, the number of partitions, and the number of iterations, the sequential implementation of the proposed approach could be much faster than the flat simulation.

One of the most promising features of this method is the parallel simulation for the power grid, as illustrated in Algorithm 1. For parallel implementation, assume there are enough number of processors available for concurrent executions and the parallelization overhead (such as communication cost and system overhead) is negligible, the computational cost can be estimated by

$$\begin{aligned}\mathscr{C}_p = &\max_{i=1}^{K}\{C_1(n_i)\} + \max_{i=1}^{L}\{C_1(m_i)\} + d\left(\max_{i=1}^{K}\{C_2(n_i)\} + \max_{i=1}^{L}\{C_2(m_i)\}\right)\\ &+ \max_{i=1}^{K}\{C_3(n_i)\} + \max_{i=1}^{L}\{C_3(m_i)\},\end{aligned} \tag{2.9}$$

where $\mathscr{C}_p$ is the overall cost for parallel simulation. When the partitions or windows have different sizes, the workloads for different processors would vary, which causes the load imbalanced problem. In this case, the partition or window with the largest size determines the overall runtime. Therefore, in order to improve the parallel efficiency, it is desired to have similar sizes for partitions or windows. Note that parallel time can be lowered by allowing processors to proceed to the next step when their work is finished, as long as the computation remains error-free.

Assume the power grid is divided into K partitions with the equal numbers of nodes n, all the L windows have m nodes as well as size s, and there are p processors available for concurrent execution. Then, the computational cost for the parallel implementation can be approximated as

$$\begin{aligned} \mathscr{C}_{pe} = & \left\lceil \frac{K}{p} \right\rceil C_1\left(n\right) + \left\lceil \frac{L}{p} \right\rceil C_1\left(m\right) + d\left(\left\lceil \frac{K}{p} \right\rceil C_2\left(n\right) + \left\lceil \frac{L}{p} \right\rceil C_2\left(m\right) \right) \\ & + \left\lceil \frac{K}{p} \right\rceil C_3\left(n\right) + \left\lceil \frac{L}{p} \right\rceil C_3\left(m\right), \end{aligned} \tag{2.10}$$

where $\mathscr{C}_{pe}$ is overall parallel simulation cost for power grids with equal-size partitions; $n = \frac{N}{K}$. Since the window Ω_{Wi} on the boundary of partition Ω_j is a subgrid of $2s \times (2s + \sqrt{n})$ nodes, it can be shown that $m \propto \frac{s}{\sqrt{K}}$. Moreover, the number of windows is in proportion to the number of partitions: $L \propto K$.

The number of partitions K and window size s are two most important factors affecting the computational cost, and we have following important observations.

- Due to the superlinear complexity of Cholesky decomposition, an increase in the number of partitions K would decrease the cost of decomposition for partitions. However, since $\frac{L}{p}C_1\left(m\right) \propto KC_1\left(\frac{s}{\sqrt{K}}\right)$, the cost for window decomposition increases along with K for a fixed s.

- Increase in the number of partitions K may increase the number of iterations d, since more regions are affected by the errors of boundary current approximations. As shown in Figure 13, when K increases from 16 (Figure 13(a)) to 32 (Figure 13(b)), after the first iteration, there are more nodes with large errors. Therefore, the case with 32 partitions may require larger number of iterations to reduce the errors.

- Increase in the window size s may decrease the number of iterations d, since the boundary current approximations are more accurate. As shown in Figure 14, when s increases from 40 (C4 ring size 1, Figure 14(a)) to 80 (C4 ring size 2, Figure 14(b)), a significant reduction of errors after first iteration is observed. However, since $m \propto s$, increase in s increases the time for window matrix decomposition and triangular solves.

Therefore, in order to maximize the runtime efficiency, the number of partitions and the window size must be carefully chosen. The coefficients of the dominant terms in (2.10) could be fitted and used to guide the parallel implementation as demonstrated in the next section.

f. Performance Modeling for Parallel Implementation

From the observations illustrated in Section II.A.3.e, there is trade-off between the number of partitions and overall simulation cost. Hence, there exists an optimal number of partitions for which the simulation cost can be minimized. For extremely large power grids, a single run of simulation may cost several hours, therefore, it is required to find the optimal (or near-optimal) number of partitions in order to save time cost. A simple strategy to determine the optimal number of partitions uses an approximation to the parallel execution time. Using the notation introduced in

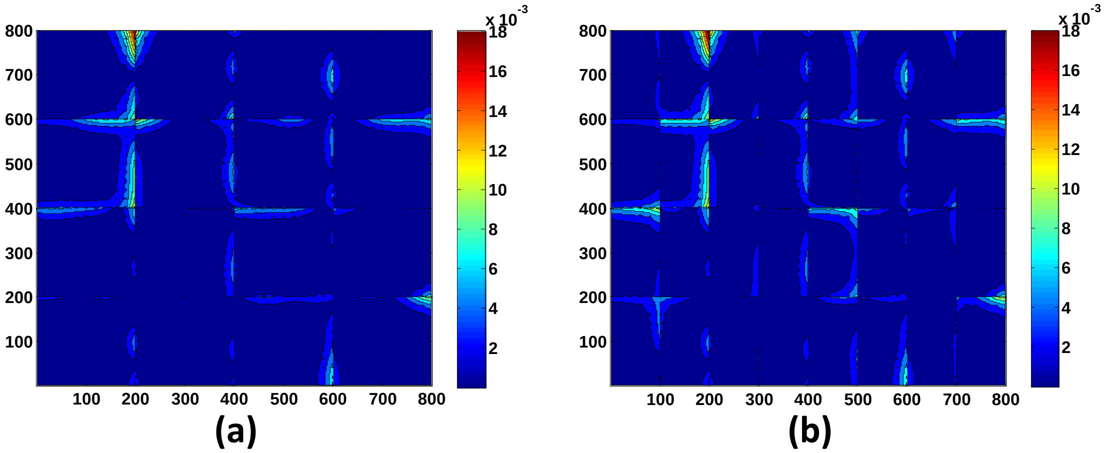

Fig. 13. Accuracy comparison of boundary current approximations using different numbers of partitions. The power grid in (a) is divided into 16 partitions, and the power grid in (b) is divided into 32 partitions. The values are node voltage errors after first iteration. The unit is Volt.

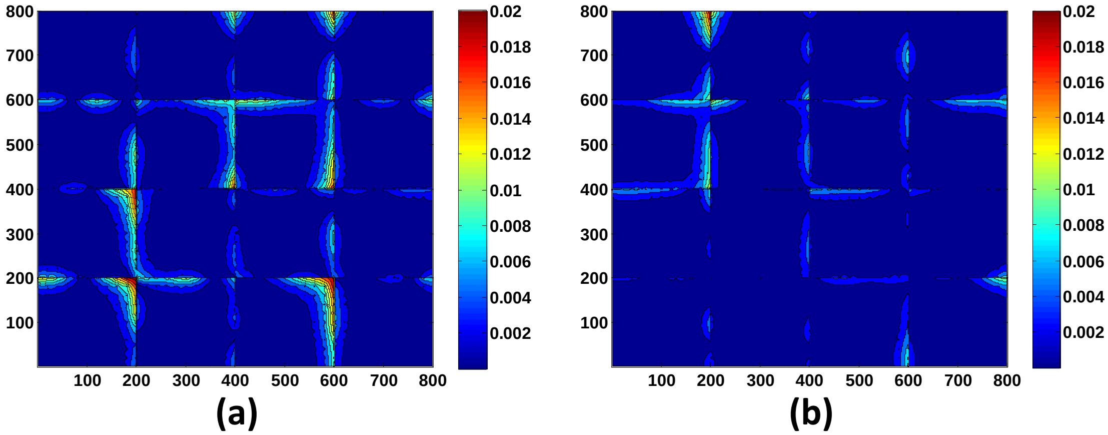

Fig. 14. Accuracy comparison of boundary current approximations using different window sizes. The power grid in (a) uses window size 40 (C4 ring size 1), and the power grid in (b) uses window size 80 (C4 ring size 2). The values are node voltage errors after first iteration. The unit is Volt.

Section II.A.3.e, we approximate C_1, C_2, and C_3 as

$$C_1(n) \approx \alpha_1 n^{1.5} + \beta_1, \tag{2.11}$$

$$C_2(n) \approx \alpha_2 n \log n + \beta_2, \tag{2.12}$$

$$C_3(n) \approx \alpha_3 n + \beta_3, \tag{2.13}$$

where α_i and β_i $(i = 1, 2, 3)$ are constants used to obtain a fit with the experimental observations.

Based on our experimental setups and results shown in Section II.A.4, we have the following observations (using different direct solvers may lead to different observations).

- s is usually chosen as C4 ring size 2 so that the number of iterations is small.
- $C_2(n)$ is a very small term compared with $C_1(n)$ and $C_3(n)$, since the decomposition time and matrix formulation time are dominant, as shown in Section II.A.4.

Notice that $m \approx 2s(\sqrt{n} + 2s) \approx 2s\sqrt{n} = 2s\sqrt{\frac{N}{K}}$ and $L \approx 2K$. Therefore, the parallel computational cost in (2.10) can be approximated as

$$\begin{aligned} \mathscr{C}_{pe} &\approx \frac{K}{p} C_1(n) + \frac{L}{p} C_1(m) + \frac{K}{p} C_3(n) + \frac{L}{p} C_3(m) && (2.14) \\ &= aK + b(NK)^{\frac{1}{2}} + c\left(N^3 K\right)^{\frac{1}{4}} + dN + e\left(\frac{N^3}{K}\right)^{\frac{1}{2}}, && (2.15) \end{aligned}$$

where $a = \frac{3\beta_1 + 3\beta_3}{p}$, $b = \frac{4s\alpha_3}{p}$, $c = \frac{4\sqrt{2}s\sqrt{s}\alpha_1}{p}$, $d = \frac{\alpha_3}{p}$, $e = \frac{\alpha_1}{p}$. Using the experimental runtime data and doing linear regression, we can get the values for a, b, c, d, e. Then, by solving

$$\frac{d\mathscr{C}_{pe}}{dK} = f(K) = a + \frac{b}{2}\left(\frac{N}{K}\right)^{\frac{1}{2}} + \frac{c}{4}\left(\frac{N}{K}\right)^{\frac{3}{4}} - \frac{e}{2}\left(\frac{N}{K}\right)^{\frac{3}{2}} = 0 \tag{2.16}$$

we can obtain the optimal (or near-optimal) number of partitions K_o.

4. Experimental Results

The proposed parallel partitioning-based simulation method has been implemented in C. Parallelization is implemented by creating multithreads using Pthreads. Experimental results for the flat simulation, and the sequential as well as parallel simulations using the proposed approach are obtained on an IBM p5-575 processing node with 16 Power5+ processors at 1.9GHz and 32G RAM (25G available for computing) running 64-bit AIX 5L (5.3).

The proposed approach has been tested under seven large-scale power grids with varying sizes: 2.56M, 4M, 5.76M, 7.82M, 9M, 12.96M, and 16M. All the power grids use C4 bump power supply pads that are evenly distributed (40 nodes away from each other). The current loadings of the grid differ from blocks to blocks but are regular inside each block. A direct solver using Cholesky decomposition [42] is chosen to carry out circuit simulations. **L** factors for boundary window circuits and partition circuits are stored for reusing in the iterative error reduction process. The matrix formulation time, data transferring time, Cholesky decomposition time, and triangular solve time are included in the total runtime. The Cholesky decomposition time for parallel simulation includes the time for window decompositions and the time for partition decompositions. Moreover, the triangular solve time for parallel simulation consists of the time for window matrices solves and partition matrices solves.

At first, all the grids are divided into 16 partitions (equal-sized) with 24 windows, and the window size is 80 (C4 ring size of 2). The runtime of the flat simulation and the proposed parallel simulation is presented in Table II. As shown, the significant advantage of the proposed approach is its scalability. The standard flat simulation does not scale well with circuit complexity (especially the Cholesky decomposition).

Table II. Simulation runtime for flat simulation and the proposed parallel simulation. All the grids are divided into 16 partitions with 24 windows. 16 processors are used. Chol: Cholesky decomposition time; Sol: triangular solve time; Tot: total runtime; Mem: memory; Iter: number of iterations; Sp: speedup; E_m: maximum node voltage error; E_a: average node voltage error. Runtime is in seconds; error is in mV; and memory is in GB.

Num.	Flat Simulation				Parallel Simulation							
Nodes	Chol	Sol	Tot	Mem	Chol	Sol	Iter	Tot	Sp	Mem	E_m	E_a
2.56M	21.0	1.5	33.2	1.8	1.6	0.15	3	3.9	8.5X	2.6	5.8e-2	2.7e-3
4M	40.7	2.4	59.1	2.9	2.2	0.22	3	5.3	11.2X	3.5	4.8e-2	1.0e-3
5.76M	69.4	3.5	97.2	4.4	3.0	0.32	3	7.0	13.9X	4.8	6.5e-2	1.1e-3
7.84M	101.0	4.8	138.1	6.0	4.1	0.43	3	9.4	14.7X	6.6	6.6e-2	9.4e-4
9M	121.5	5.6	164.3	6.9	4.4	0.49	3	10.6	15.5X	7.4	5.2e-2	1.1e-3
12.96M	203.3	8.1	266.1	10.3	6.9	0.70	3	14.9	17.8X	10.5	4.3e-2	1.3e-3
16M	267.9	9.9	346.6	12.8	9.5	0.97	3	18.4	18.8X	12.8	5.0e-2	1.1e-3

Table III. Speedups of the parallel implementation of the algorithm over the sequential implementation of the algorithm. 16 processors are used for the parallel simulation. Runtime is in seconds.

Num. Nodes	Window Size	Num. Partition	Sequential	Parallel	Speedup
2.56M	80	16	36.4	3.9	9.3X
4M	80	16	53.6	5.3	10.1X
5.76M	80	16	76.7	7.0	10.9X
7.84M	80	16	104.0	9.4	11.1X
9M	80	16	118.1	10.6	11.1X
12.96M	80	16	173.7	14.9	11.6X
16M	80	16	216.9	18.4	11.8X

In contrast, the *divide-and-conquer* nature of the proposed method makes itself highly scalable (the Cholesky decomposition time is almost linear). As shown in Figure 15, the runtime for the proposed method is almost linear with very small slope, therefore the runtime speedups over the flat simulation keep increasing throughout all the cases. For the largest 16M-node grid, the speedup is already 18.8X. More speedups are expected for even larger grids. The superlinear speedups come from the proposed method itself and the parallel implementation on the 16-core machine. Moreover, Table II also shows that the proposed block-based iterative process is very effective and efficient. The accuracy of less than 0.07mV for the maximum node voltage error can be reached after only three iterations. For small power grids, our method consumes more memory than the flat method for the reason that the numbers of nodes in the windows are comparable to the numbers of nodes in the partitions. However, along with the increasing size of the power grid, the memory consumptions growth rate of the proposed approach is less than that of the flat method. We can expect

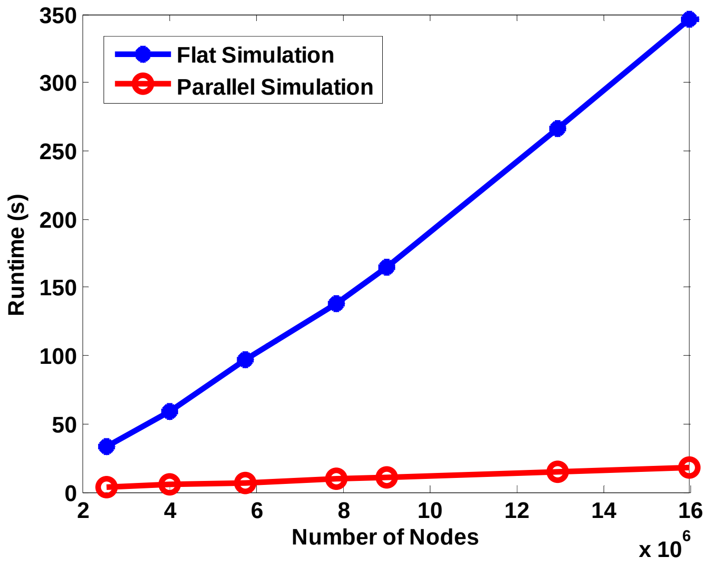

Fig. 15. Runtime comparison between the flat simulation and the proposed parallel simulation.

favorable memory consumptions of our method for extremely large power grids.

Next, we analyze the parallel efficiency of the proposed algorithm. Table III compares the execution time of the proposed method on a single processor with that on 16 processors. As presented, the parallel implementation can bring about 11X speedups. The ideal 16X speedup is not achieved due to system overheads in the present implementation.

As discussed in Section II.A.3.e, the number of partitions and the window size determines the number of iterations, thus have impacts on the total runtime. As shown in Table IV, when we increase the number of partitions (from 8 to 16) so that all the available cores can be fully utilized for parallel simulation, the total runtime decreases. However, when we further divide the grid into more partitions, the runtime efficiency degrades. There are three reasons for this behavior. One is the hard limit of the number of cores available, 16 in our experiment. Another reason is that increasing the number of partitions increases the simulation cost for windows.

Table IV. Runtime analysis with various numbers of partitions for 9M, 12.96M, and 16M grids. Chol: Cholesky decomposition time; Sol: triangular solve time; Iter: number of iterations; Tot: total runtime. Runtime is in seconds.

Num.	8 Partitions				16 Partitions				32 Partitions			
Nodes	Chol	Sol	Iter	Tot	Chol	Sol	Iter	Tot	Chol	Sol	Iter	Tot
9M	10.7	0.94	3	19.6	4.4	0.49	3	10.6	4.1	0.54	4	12.0
12.96M	15.7	1.3	3	28.3	6.9	0.70	3	14.9	6.3	0.80	4	16.3
16M	18.5	1.6	3	33.5	9.5	0.97	3	18.4	8.9	1.3	3	19.6

Table V. Number of iterations required by using various window sizes. Iter: number of iterations; Tot: total runtime. Runtime is in seconds.

Window Size	C4 Ring Size	4M Grid		5.76M Grid		7.84M Grid		9M Grid	
		Iter	Tot	Iter	Tot	Iter	Tot	Iter	Tot
40	1	5	5.4	5	7.1	5	9.7	6	11.3
80	2	3	5.3	3	7.0	3	9.4	3	10.6
120	3	3	6.6	3	8.6	3	10.9	3	12.2

The last is that using small-size partitions may lead to more iterations because small partitions tend to have more coupling to the rest of the grid. In addition, as shown in Table V, more accurate boundary currents obtained from the large boundary window simulation would reduce the number of iterations needed for the error correction, but may cost more time in parallel boundary current approximation. Our experiments indicate that a window size of 80 (C4 ring size of 2) is sufficient for the test power grids, which is consistent with [19]. The block Jacobi method is corresponding to the case with window size of 0, whose convergence can be expected to be much slower than our method.

By using the data in Table II and doing linear regression, we can get that in our experimental setup,

$$[a,b,c,d,e]^T = [1.10, -1.35\times 10^{-2}, 9.18\times 10^{-4}, 1.86\times 10^{-5}, 3.18\times 10^{-9}]^T. \quad (2.17)$$

For the 16M power grid, by solving (2.16), the optimal (or near-optimal) number of partitions K_o is found to be 16 which is consistent with the result shown in Table IV. It can be expected that K_o increases with N.

5. Summary

In this section, we have presented a parallel partitioning-based power grid analysis approach using the spatial locality. We have identified the main factors that effect the solving process: the number of partitions and the window size. The interdependence of these parameters and their influence on the execution time have been analyzed. We have also suggested a strategy that helps users in determining the optimal (or near optimal) values of these parameters to achieve lowest parallel runtime. The proposed approach is shown to have excellent parallel efficiency, fast convergence, flexible partitioning, and favorable scalability. By using distributed computing networks, it is believed to be able to handle extremely large power grids (with many-million nodes) in an efficient way.

B. GSim: A Fast CPU-GPU Combined Parallel Simulator for Power Delivery Networks with On-Chip Voltage Regulation

Detailed and accurate analysis to the PDNs with on-chip voltage regulators is hindered by the lack of efficient simulation techniques for such PDNs. In this section, the simulation challenges are addressed by the proposed partitioning relaxation method. Using this method, an existing fast GPU multigrid solver for on-chip power grids and a general SPICE simulator for nonlinear regulators are integrated together to achieve excellent efficiency and accuracy.

1. Background and Overview

The detailed model of a multiple-domain power delivery network with on-chip low-dropout voltage regulators is presented in Figure 16. The on-chip PDN has a global VDD grid, several on-chip LDOs, a number of local grids and a global GND grid.

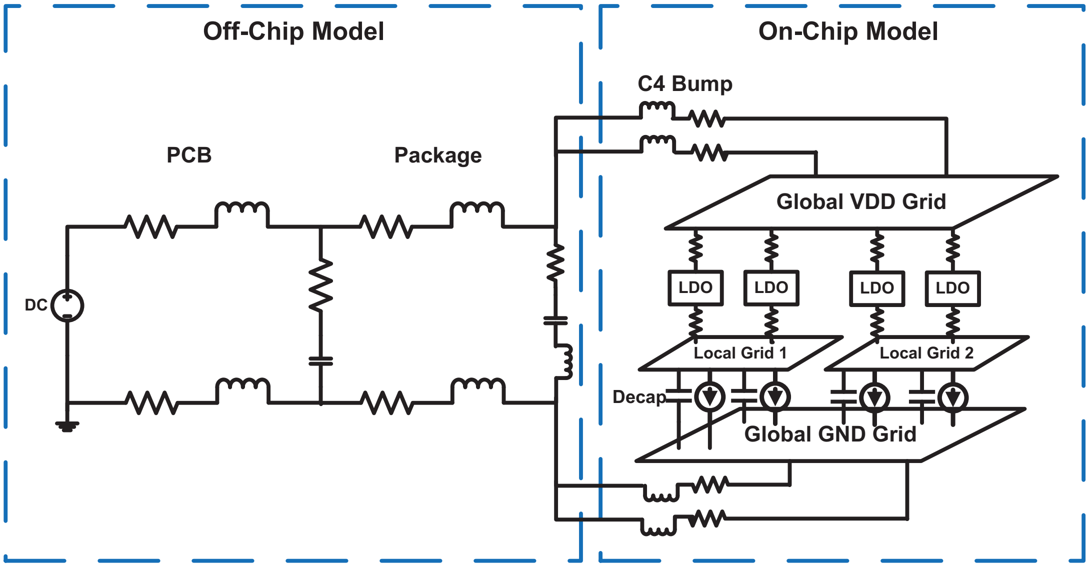

Fig. 16. The model for a power delivery network with on-chip voltage regulators.

The global VDD grid distributes input voltage to on-chip LDOs through metal wires. Each local grid corresponds to a power domain, and its voltage is provided by LDOs.

By identifying the need to accelerate the simulation for large linear power grids, the difficulty of simulating multiple non-linear voltage regulators and the hierarchical nature of the network structure, we adopt a black-box Gauss-Seidel relaxation algorithm [43] to develop the GSim, a GPU accelerated simulation engine that can solve extremely large PDNs with good runtime and memory efficiency. GSim utilizes an efficient iterative partitioning relaxation method to analyzes LDOs, the off-chip circuit as well as extremely large on-chip power grids. As presented in the left of Figure 17, the entire PDN can be partitioned into five major circuit blocks: the off-chip circuit, the LDOs, the global VDD grid, the local grids and the global GND grid. For the transient simulation, at each time point, GSim solves each circuit block individually and updates the solutions through the partition boundary until the convergence is reached. To solve each block in the most efficient way, the off-chip circuit is solved

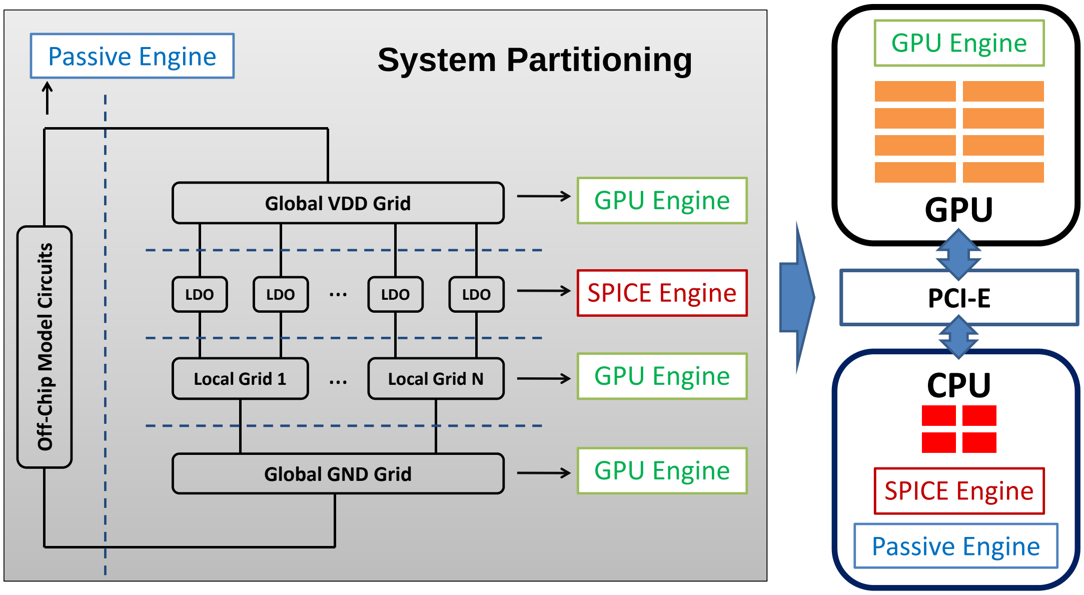

Fig. 17. GSim simulation diagram.

by a passive network solver on CPU, the transistor-level LDOs are analyzed by a SPICE solver on CPU, and all the power grids (the global VDD grid, local grids and the global GND grid) are tackled by a GPU multigrid solver [13] which is over 50X faster than the state-of-the-art direct solver CHOLMOD [42]. The updated results for partition boundaries are exchanged through PCI-E between CPU and GPU. It will be shown in the experimental results, most of the simulation time is spent on solving power grids. Thus our partitioning-based simulation scheme which puts power grid simulations on a fast GPU engine is very efficient in terms of runtime.

2. CPU-GPU Combined Transient Simulation

a. GPU-Based Multigrid Method

In this work, we solve the on-chip power grids on GPU by adopting the hybrid multigrid (HMD) method [13]. The idea of the HMD method is to set the original 3D irregular power grid as the finest grid in the multigrid hierarchy, and define a set of

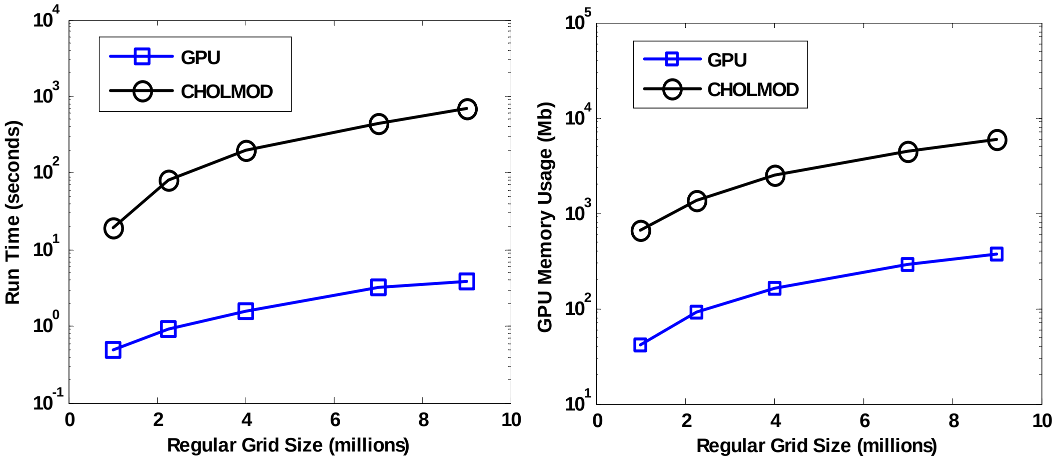

Fig. 18. Scalability of runtime and memory consumption for the GPU solver.

topologically regularized 2D grids as the coarser to coarsest level grids that can be obtained from the original 3D irregular PNDs. The most time-consuming smoothing steps, as well as other multigrid operations are accelerated on GPU efficiently. More specifically, PDN simulation can be efficiently performed on GPU with no explicit sparse matrix-vector operations. The convergence of the HMD algorithm shares the common properties of multigrid methods which is linear in time and memory consumption. Our experiments on a variety of industrial power grid designs show that after only a few HMD iterations, power grid error components can be damped out quickly (with maximum errors smaller than 1mV and average errors smaller than 0.1mv).

The GPU based HMD solver [13] has been shown to be very efficient for solving the on-chip power grids. For instance, for synthetic power grids with one million to nine million nodes, as shown in Figure 18 [13], the GPU simulation engine can solve them at a rate of three million nodes per second, which is 50X to 180X faster than the state of art direct solver CHOLMOD [42] executed on a quad-core 2.6GHz computer. Additionally, the GPU solver only consumes 20X less memory than the

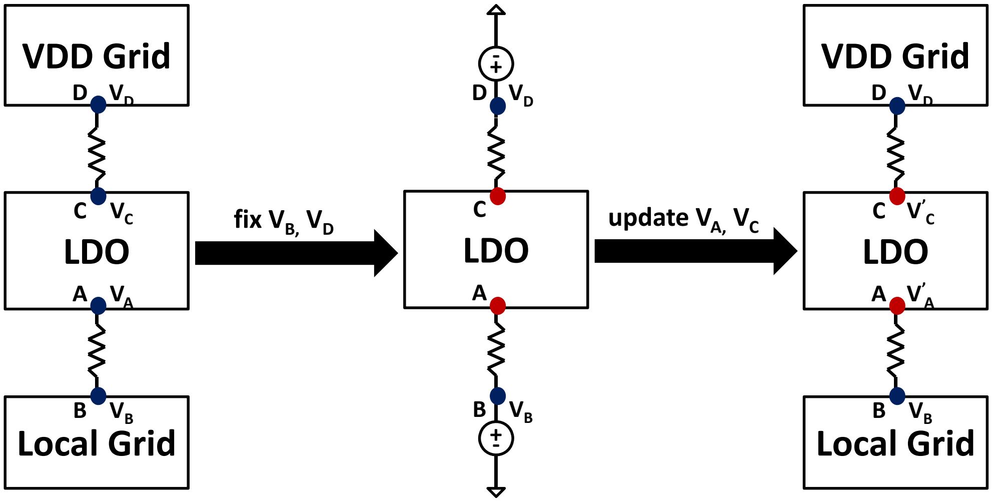

Fig. 19. Boundary relaxation for a single LDO.

CHOLMOD solver. For instance, less than 500Mb memory is required for solving the nine-million-node grid.

b. Boundary Relaxation

Detail analysis for individual circuit block simulation and the boundary relaxation scheme is given to a single LDO. All other circuit blocks and partition boundaries can be handled in the same way. As shown in Figure 19, an LDO is connected with a local grid (through a resistor between nodes A and B) and the VDD grid (through a resistor between nodes C and D). A and C are contained in the LDO block, B is on the local grid, and D is on the VDD grid. LDO is analyzed along with those two resistors and two ideal voltage sources at B and D whose values are V_B and V_D respectively. Then, after the simulation finishes, new node voltages for A (V'_A) and C (V'_C) are obtained to update their values. It should be noted that the decoupling capacitors between local grids and GND grid can be converted to a resistor and a current source using the Norton companion model.

c. Convergence

The convergence is examined by checking the average and maximum voltage changes at partition boundaries. Although LDO is a nonlinear device, due to its property that automatically maintains the output voltage, the voltage change at the boundary between LDO and local grids is small for consecutive time steps. Other partition boundaries have small voltage change from time step to step as well. Therefore, the convergence can be quickly reached, which is consistent with our experimental results. Further convergence improvement can be introduced by employing multi-level Newton method.

d. Simulation Flow

The simulation flow for GSim at the time step k is summarized in Figure 20. Circuit blocks are solved separately and the solutions of the interfacing nodes are updated using the boundary relaxation scheme presented in Figure 19. However, considering the strong interactions between the on-chip global and local PDNs (caused by the decoupling capacitors), a naive implementation of the block circuit iteration scheme may lead to slow convergence. In this work, we propose to first solve the local grids and the global GND grid through a number of inner iterations, and then solve the rest of the circuit blocks in an outer iteration loop. As shown in our experimental results, only a few (two to four) outer iterations are needed at each time step.

3. Experimental Results

The PDN simulator GSim has been implemented in CUDA [44] and C++, respectively. The GPU program is executed on a single GPU of the NIVIDIA Geforce 9800 GX2 card (including two GPUs), with a total on board memory of 512Mb. All the

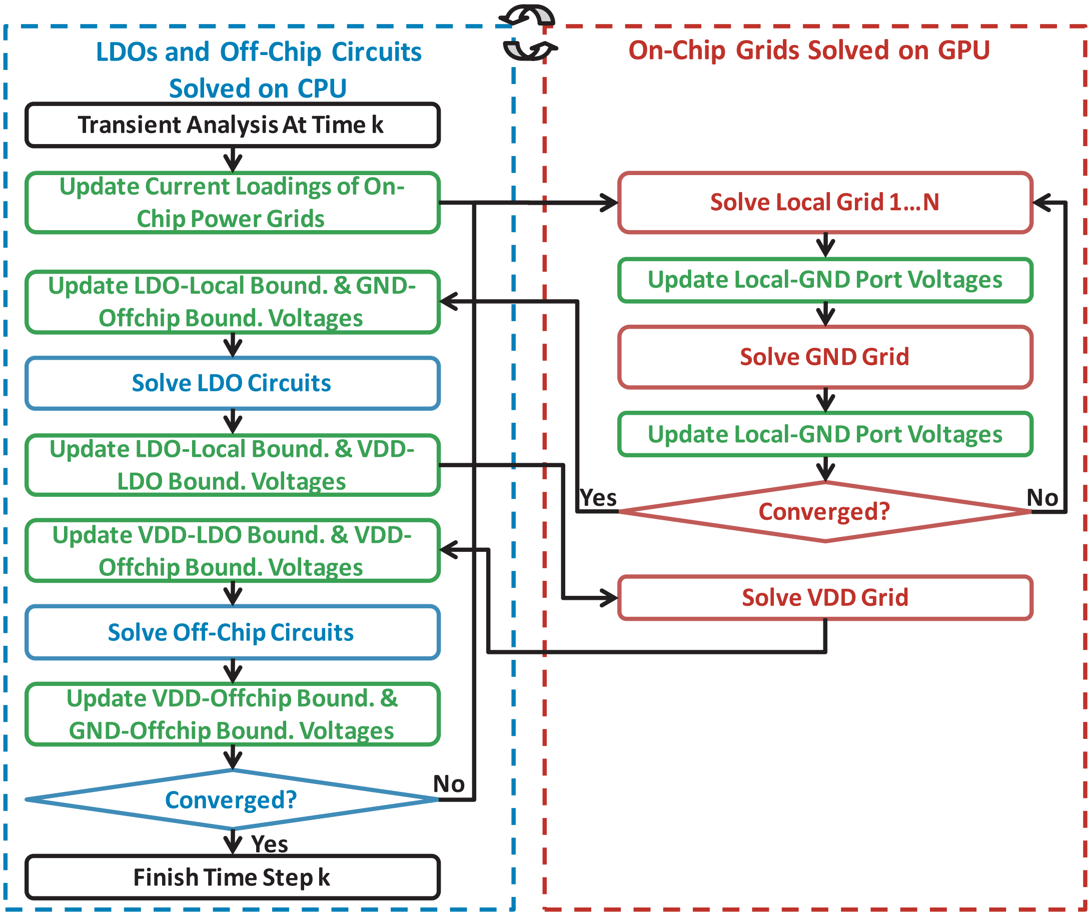
LDOs and Off-Chip Circuits Solved on CPU
Transient Analysis At Time k
Update Current Loadings of On-Chip Power Grids
Update LDO-Local Bound. & GND-Offchip Bound. Voltages
Solve LDO Circuits
Update LDO-Local Bound. & VDD-LDO Bound. Voltages
Update VDD-LDO Bound. & VDD-Offchip Bound. Voltages
Solve Off-Chip Circuits
Update VDD-Offchip Bound. & GND-Offchip Bound. Voltages
Converged?
No
Yes
Finish Time Step k
On-Chip Grids Solved on GPU
Solve Local Grid 1…N
Update Local-GND Port Voltages
Solve GND Grid
Update Local-GND Port Voltages
Yes
Converged?
No
Solve VDD Grid

Fig. 20. GSim simulation flow.

Table VI. Transient simulation runtime and numbers of iterations of GSim for PNDs with on-chip LDOs. 1200 time steps are simulated. CPU%: the percentage of runtime spent on CPU. The runtime is in seconds.

Num. Nodes	Num. LDOs	Runtime			Num. Iteration	
		Total	/Step	CPU%	Total	/Step
2.25M	36	1810	1.6	22	2274	1.9
2.25M	144	1768	1.5	23	2000	1.7
9M	64	7398	6.2	24	2864	2.4
9M	256	4500	3.7	27	1900	1.4

C++ programs are executed on a workstation with Intel Xeon CPU at 2.33GHz and 4G RAM running 64-bit Linux OS.

The transient simulation runtime is examined by analyzing PDNs with 2.25M or 9M on-chip nodes. Each PDN has a different number of LDOs. The results are shown in Table VI. For PDNs with less LDOs, since the voltage changes are smoother than the PDNs with less LDOs, they require more iterations to converge. But all four cases can converge in less than an average of three outer iterations per time step. Notice that two inner iterations are forced for each outer iteration. As can be seen, the cost of analyzing on-chip power grids is dominant, and the overhead introduced by simulating LDOs is not significant because of the proposed partitioning scheme. Therefore, putting on-chip power grid simulation to the fast GPU engine is extremely effective and speeds up the entire simulation visibly. In summary, GSim is an efficient solver to tackle multi-million-node PDNs with on-chip LDO regulators.

4. Summary

A fast CPU-GPU combined simulation engine, GSim, has been developed to provide SPICE-level accuracy for simulating complex PDNs employing on-chip voltage regulation techniques. GSim identifies the simulation difficulties for different circuit blocks and achieves its efficiency by using a block-based Gauss-Seidel relaxation scheme to integrate several fast simulation strategies together. These simulation strategies are specifically designed for three types of circuit blocks in the PDN. Most importantly, GSim provides a foundation to comprehensively analyze electric characteristics and various design tradeoffs for PDNs with on-chip voltage regulation.

C. Transient Verification of Power-Gated Power Delivery Networks

The on/off states and transitions of gated power grids can generate a large number of power gating configurations which makes the verification of gated PDNs very difficult. By using an effective circuit modeling method and a fast superposition approximation flow, the verification methodology presented in this section makes the verification for gated PDNs feasible.

1. Background

In our work, we focus on the transient power gating verification problem, which can be defined as: for a given loading current distribution for each core, verify if the required EM and voltage drop specifications are satisfied under all possible on/off configurations and on/off transitions of the power-gated PDN network.

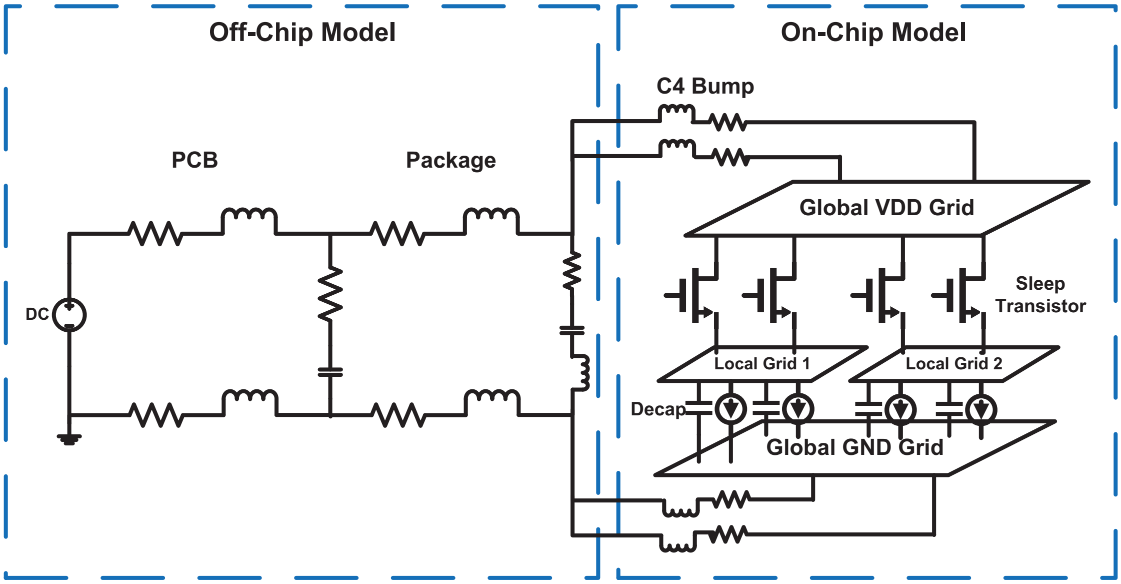

Fig. 21. A power-gated power delivery network model.

a. Modeling for Power-Gated PDNs

The model for a power-gated PDN is shown in Figure 21. On the chip, the global VDD grid delivers power to gated local grids through multiple sleep transistors. The decoupling capacitors and switch circuits reside between the local grids and the common global GND grid. When a sleep transistor is completely turned on, it is modeled as a resistor. Otherwise, it is treated as an open circuit. However, during the power-on process, due to the charging effect of the decoupling capacitors, sleep transistors work in the saturation region and the linear region in different phases. Hence, sleep transistors are modeled as time-varying resistors during power-on.

b. Verification Metrics

We define important electromigration and Dynamic Voltage Drop (DVD) metrics for interconnects and switch circuits, respectively, in our transient verification tasks. Since the EM effect is proportional to the average current flowing through the wire

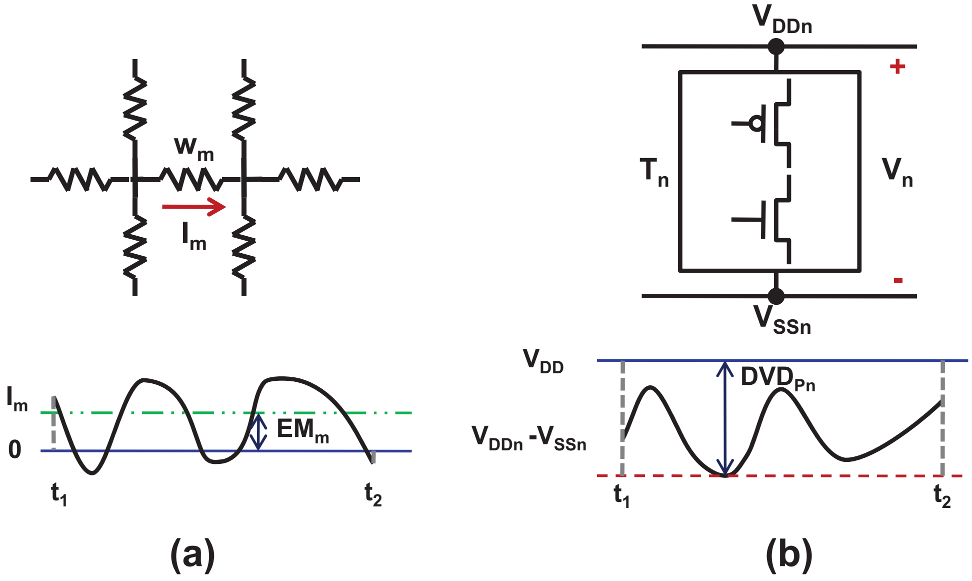

Fig. 22. EM and DVD metrics.

[5], we use the average current to define the EM metric. As shown in Figure 22(a), for a wire w_m with transient current $I_m(t)$, if the time period for verification starts at t_1 and ends at t_2, the EM metric is given as

$$|EM_m| = \left| \frac{\int_{t_1}^{t_2} I_m(t)dt}{t_2 - t_1} \right|. \tag{2.18}$$

On the other hand, the peak DVD is important to the operation and timing performance of circuit designs [4]. Correspondingly, it is adopted as a verification metric. As shown in Figure 22(b), for a circuit block T_n connected to V_{DDn} and V_{SSn}, during the verification time period t_1 to t_2, the peak DVD is defined as,

$$DVD_{Pn} = \max_{t_1 \le t \le t_2} \{V_{DD} - V_{DDn}(t) + V_{SSn}(t)\}. \tag{2.19}$$

2. Overview of the Transient Verification

a. Verification Tasks

In a power-gated power delivery network, each local grid has four possible states: sleep, active, transition, and idle. In the sleep state, the sleep transistors are completely turned off. In the active state, the sleep transistors are completely turned on and the switch circuits work in steady state. In the transition state, the sleep transistors are gradually turned on while the decoupling capacitors are under charging, but the switch circuits are idle. Finally, in the idle state, the sleep transistors are completely turned on while the switch circuits are still idle. In this work, the idle state is not explicitly considered, since it can be considered active by modeling leakage currents. For a local grid G_k, its state is represented by b_k,

$$b_k = \begin{cases} 0 & \text{sleep state,} \\ 1 & \text{active state,} \\ 2 & \text{transition state,} \\ 3 & \text{idle state.} \end{cases} \tag{2.20}$$

A power gating configuration is a combination of the above possible states of gated local grids. Two kinds of power gating configurations are examined in this work:

- Stable configuration: as shown in Figure 23, each gated local grid is either in the active state or sleep state. In this configuration, the voltage drops and current distributions across the entire PDN are only caused by switch circuits in the steady state (modeled as current sources). Assume there are N independent

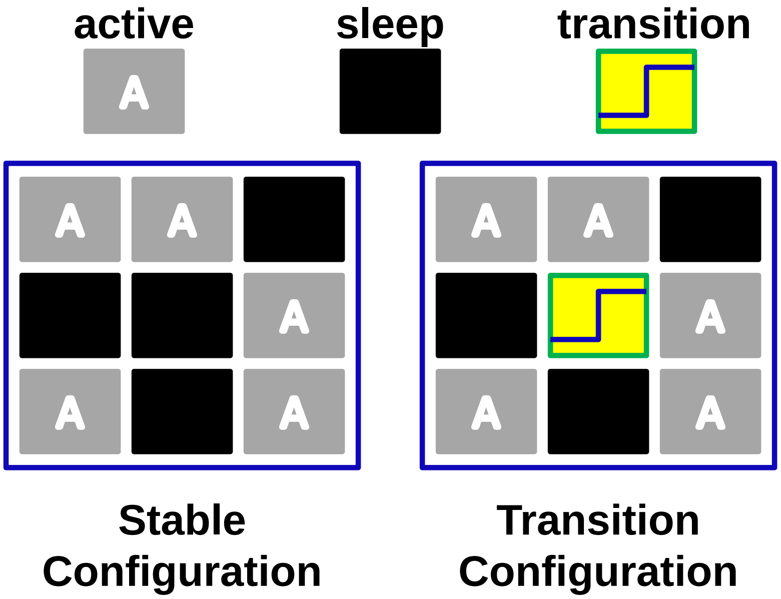

Fig. 23. Stable and transition configurations.

gated local grids $G_1, \ldots, G_N$. A stable configuration $\mathscr{C}_i$ can be represented as

$$\mathscr{C}_i = \{b_{1i}, b_{2i}, \ldots, b_{Ni}\},\ b_{ki} = 0, 1;\ k = 1, \ldots, N, \tag{2.21}$$

where b_{ki} is the state of grid G_k. The corresponding state of b_{ki} is defined in (2.20).

- Transition configuration: some gated local grids are in the transition state while others are either in the active or sleep state. In this configuration, the voltage drops and current distributions are influenced by sleep transistors' power-on processes (charging effect). Similar to the stable configuration, if there are N independent gated local grids $G_1, \ldots, G_N$, a transition configuration $\mathscr{D}_j$ can be represented by

$$\mathscr{D}_j = \{b_{1j}, b_{2j}, \ldots, b_{Nj}\},\ b_{kj} = 0, 1, 2;\ k = 1, \ldots, N, \tag{2.22}$$

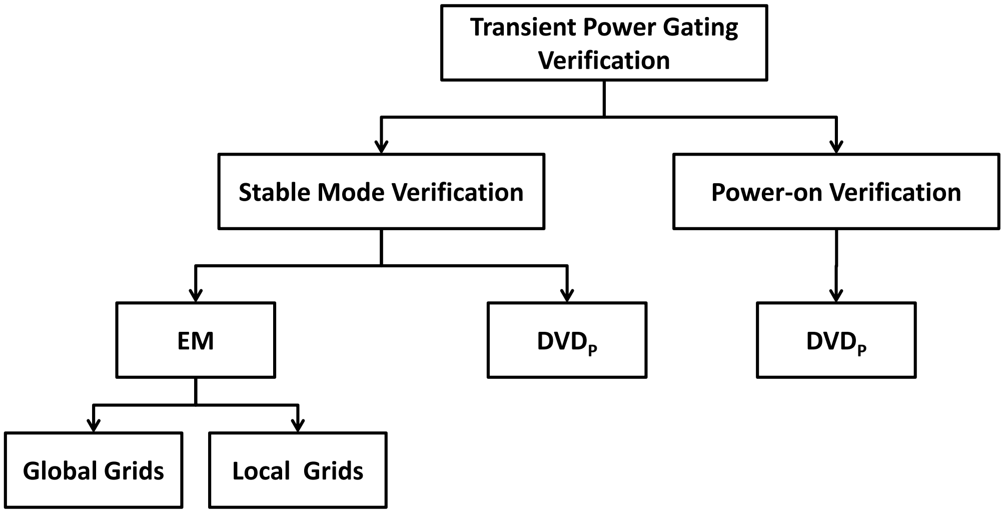

Fig. 24. Proposed transient verification tasks.

where b_{kj} is the state of grid G_k.

As summarized in Figure 24, the proposed verifications tasks are categorized into stable mode verification and power-on verification.

- In the stable mode verification, the mission is to find the worst or near-worst dynamic performance (the largest EM and peak DVD values) of the PDN under all possible stable configurations. This helps designers ensure that when their chip work in steady state under any possible stable gating configuration, the gated PDN satisfies the power integrity requirements. We capture both time-average and temporal effects by checking the values of EM and peak DVD metrics. Moreover, interconnect EM values in both the global and the local grids are checked. However, it is only necessary to perform peak DVD checks for local grids where switch circuits reside.

- In the power-on verification, the worst or near-worst dynamic performance under all possible transition configurations is identified. The goal is to determine when coupled with stable workloads from other cores, whether or not the power-on current transients would jeopardize power delivery integrity.

b. Overview of the Proposed Transient Verification Methodology

With current loadings given for each local grid, according to (2.21) and (2.22), a PDN with N gated local grids, or cores, has 2^N stable configurations and $3^N - 2^N$ transition configurations. A brute-force exhaustive verification would require at least 2^N lengthy die-package PDN transient simulations, each covering at least hundreds of clock cycles to capture the dynamics of the network. Hence, the brute-force approach is simply infeasible.

To achieve the feasibility and efficiency, in this work, we propose a novel technique to drastically reduce the number of full simulations from 2^N or even more to $O(N)$. Our method develops an equivalent circuit modeling scheme, which makes superposition able to be used by efficient approximations for all power gating configurations. These approximations are obtained by fully simulating $O(N)$ number of configurations, whose contributions are superimposed to find out several configurations (also in the order of $O(N)$) that can potentially cause the worst-case circuit performance. These configurations are called worst-case candidates in the paper. Then, this set of worst-case configuration candidates are fully simulated and the worst or near-worst performance can be found. The high-level flow of the proposed methodology is summarized in Figure 25. As can be seen, the proposed process can significantly reduce the computation complexity from $O(2^N)$ to $O(N)$. Overall, the proposed approach makes the transient verification for a power-gated PDN with several-million nodes feasible.

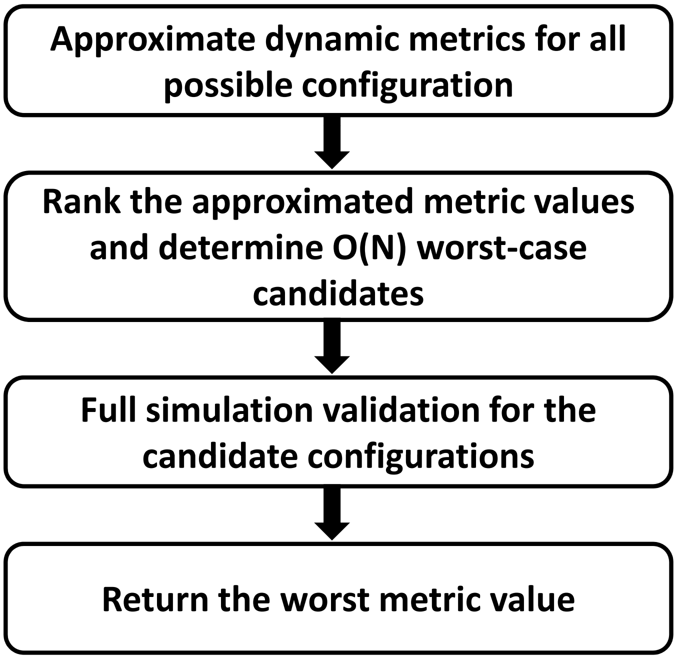
Approximate dynamic metrics for all possible configuration
Rank the approximated metric values and determine O(N) worst-case candidates
Full simulation validation for the candidate configurations
Return the worst metric value

Fig. 25. The proposed transient verification methodology.

3. Stable-Mode EM Verification for Global Grids

The stable-mode EM verification is to find the largest average current, for a wire in the on-chip PDN, under all possible stable configurations.

a. Challenges for Transient EM Verification

Firstly, consider the DC EM analysis for a simple gated power grid used in [26], as shown in Figure 26. For illustration purpose, assume there are only two current loads for each gated local grid. Each gated local grid is connected to the global grid through a single sleep transistor. Since only the DC condition is examined, the current flowing through the sleep transistor is equal to the total current loads of its connected local grid. For example, for local grid 1 in Figure 26, we have,

$$I_T = I_{s1} + I_{s2} \tag{2.23}$$

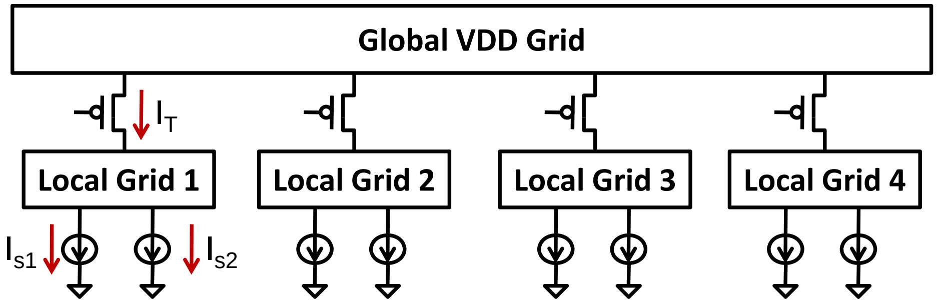

Fig. 26. A simple power-gated PDN in [26]. For simplicity, only four local grids are shown. Each local grid has a single sleep transistor and two DC current loads.

where I_T is the DC current through the sleep transistor and I_{s1} and I_{s2} are the DC current loads for local grid 1.

Therefore, turning on/off a gated local grid is equivalent to connecting or disconnecting a current source to the global grid. The value for this current source can be obtained by summing up all the current loads under its corresponding local grid. Note that each local grid can be treated as an independent current *input* to the global grid. Thus superposition theory can be easily applied in this case to find the exact branch current for any configuration.

In this work, a more general and complex PDN model is used (as shown in Figure 27). Meanwhile, we consider the more meaningful transient dynamics for gated PDNs. However, the above leads to three key difficulties for employing superposition theory.

- The existence of decoupling capacitance as well as the package makes the entire PDN a strongly coupled system. Therefore, finding a clear cut for the *inputs* to the system is difficult.

- Since multiple sleep transistors are used for each local grid, the change of gating configuration would cause redistribution of the currents through these transistors. Thus, the value of the current source model for each local grid is not fixed.

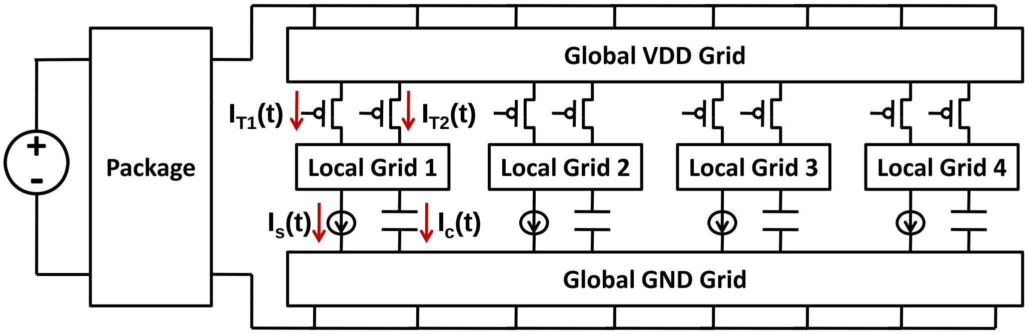

Fig. 27. The PDN model diagram. For simplicity, only four local grids are shown. Each local grid has two sleep transistors, one current load and one decoupling capacitor.

For example, for local grid 1 in Figure 27, even though the current load I_s is given, the currents through two sleep transistors I_{T1} and I_{T2} may change when other local grids' states change (e.g. local grid 2 changes from the active mode to sleep mode).

- The current flowing through the decoupling capacitance is subject to change when the power gating configuration changes. For example, for local grid 1 in Figure 27, the current through the decoupling capacitance $I_c(t)$ would change when the configuration changes from $\{1, 1, 1, 1\}$ (all grids are active) to $\{1, 1, 1, 0\}$ (grid 4 is in sleep, others are active), because the change of total capacitance in the system leads to the variation of frequency response.

Although the transient EM analysis for PDNs shown in Figure 27 is challenged by the above difficulties, we develop the following concepts of equivalent circuit modeling and superposition approximation to speedup the verification in the large space of possible gating configurations.

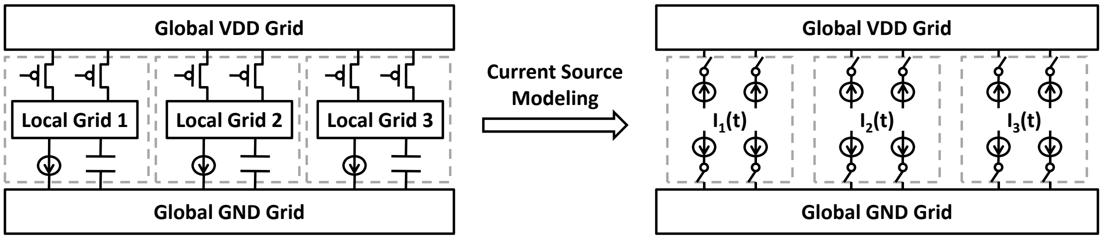

Fig. 28. Switchable current source model for local grids. For simplicity, the package is not shown and there are only three local grids.

b. Superposition Approximation

Identifying the *circuit* to which superposition is applied and the corresponding *inputs* is crucial in applying superposition technique. If we treat the current loads in each local grid as the *inputs* and the remaining PDN as the *circuit*, then, turning on or off the local grids not only changes the *inputs*, but also changes the *circuit*. Therefore, in order to maintain the *circuit* while turning on/off gated local grids, the local grids should be treated as the *inputs*, and the global grids, C4 bumps and the off-chip circuit are included in the *circuit*. When a local grid is active, it draws currents from or provides currents (through decaps) to the global grids. Hence, each local grid can be modeled as a set of switchable time-varying current sources attached to the global grids. As a simple example, in Figure 28 local grid 1, 2 and 3 are modeled as switchable time-varying current sources $\mathbf{I_1(t)}$, $\mathbf{I_2(t)}$ and $\mathbf{I_3(t)}$ respectively.

Before determining the value for each switchable current source, we define a *global basic stable configuration* $\mathscr{C}_i^g$ and a *full-decap basic stable configuration* $\mathscr{C}_i^f$ as below (assume there are N independent gated local grids),

$$\mathscr{C}_i^g = \{b_{1i}, b_{2i}, \ldots, b_{Ni}\}, \quad b_{ii} = 1, \text{others} = 0, \tag{2.24}$$

$$\mathscr{C}_i^f = \{b_{1i}, b_{2i}, \ldots, b_{Ni}\}, \quad b_{ii} = 1, \text{others} = 3, \tag{2.25}$$

where in $\mathscr{C}_i^g$, only grid G_i is active and others are in sleep; in $\mathscr{C}_i^f$, only grid G_i is active and others are idle (acting as decoupling capacitance). Moreover, we assume the decoupling capacitances for local grids $G_1, \ldots, G_N$ are $C_1, \ldots, C_N$ respectively. For the *global basic stable configurations*, the decoupling capacitance of all other grids are neglected. In $\mathscr{C}_i^g$, there is only C_i decap in the PDN. Whereas the full amount of decoupling capacitance is considered in the *full-decap basic configurations.* In $\mathscr{C}_i^f$, the total amount of decoupling capacitance is $\sum_{i=1}^{N} C_i$.

As we know, the value of each switchable current source $\mathbf{I_i(t)}$ changes with configuration alternation, since the amount of decoupling capacitance which is dependent on the configuration directly impacts the dynamic currents and voltages. Let us look at a simple RLC circuit in Figure 29 which models the power delivery network in the simplest way and assume that only G_i is active and all other local grids are in sleep. L_g, R_p, R_g, R_i, C_i and I_{si} represent the off-chip inductance, the resistance of the global VDD grid, the resistance of the global GND grid, the resistance of the local grid G_i, the decap of local grid G_i and the current loading of G_i respectively. By solving this circuit in AC, the current flowing through G_i (from X to Y) is

$$I_i(s) = I_{si}(s)\left(1 - \frac{2sL_g + R_p + R_g}{2sL_g + R_p + R_g + R_i + \frac{1}{sC_i}}\right). \tag{2.26}$$

We can obtain

$$I_i(s) \approx I_{si}(s), \text{ at low frequency,} \tag{2.27}$$

$$I_i(s) \approx 0, \text{ at high frequency.} \tag{2.28}$$

Having some local grids in idle is equivalent to adding some resistor-capacitor branches between X and Y. Therefore, for a general power gating configuration $\mathscr{C}_k$,

$$I_i(s) \approx I_{si}(s), \text{ at low frequency,} \tag{2.29}$$

$$I_i(s) \approx I_{si}(s)\left(1 - {}^{C_i}\!/\!\sum_{n=1}^{N} b_{nk}C_n\right), \text{ at high frequency,} \tag{2.30}$$

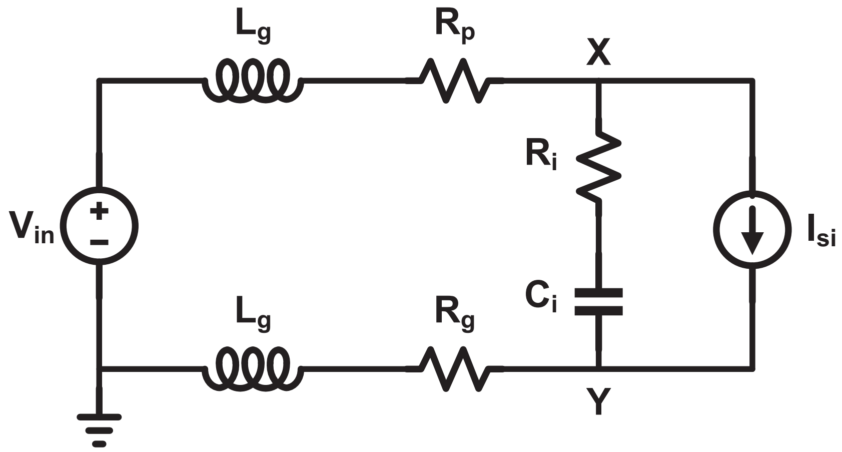

Fig. 29. A simple RLC model for the power delivery network.

This means that $\mathbf{I_i(t)}$ can be approximated as a linear function of the loading current of G_i, $\mathbf{I_{si}(t)}$. The coefficient for the high-frequency component is determined by the total capacitance in the system. Therefore, we approximate $\mathbf{I_i(t)}$ using the currents flowing in or out of the global grids under its *global basic configuration* $\mathscr{C}_i^g$ ($\mathbf{I_i^g(t)}$) and *full-decap basic configuration* $\mathscr{C}_i^f$ ($\mathbf{I_i^f(t)}$). For the power delivery network in Figure 30, current sources $\mathbf{I_1^g(t)}$ and $\mathbf{I_1^f(t)}$ are

$$\mathbf{I_1^g(t)} = \{I_{T1}^g(t), I_{T2}^g(t), I_s(t), I_c^g(t)\}, \tag{2.31}$$

$$\mathbf{I_1^f(t)} = \{I_{T1}^f(t), I_{T2}^f(t), I_s(t), I_c^f(t)\}, \tag{2.32}$$

where $I_{T1}^g(t)$, $I_{T2}^g(t)$ are the currents through the sleep transistors and $I_c^g(t)$ is the current through the decoupling capacitor, all under the *global basic configuration* $\mathscr{C}_1^g = \{1, 0, 0\}$; $I_{T1}^f(t)$, $I_{T2}^f(t)$ are the currents through the sleep transistors and $I_c^f(t)$ is the current through the decoupling capacitor, all under *full-decap basic configuration* $\mathscr{C}_1^f = \{1, 3, 3\}$; and $I_s(t)$ is the current load. $I_{T1}^g(t)$, $I_{T2}^g(t)$ and $I_c^g(t)$ can be obtained by simulating the basic configuration $\mathscr{C}_1^g$ as shown in Figure 30(a), and $I_{T1}^f(t)$, $I_{T2}^f(t)$ and $I_c^f(t)$ can be obtained by simulating the basic configuration $\mathscr{C}_1^f$, as shown in

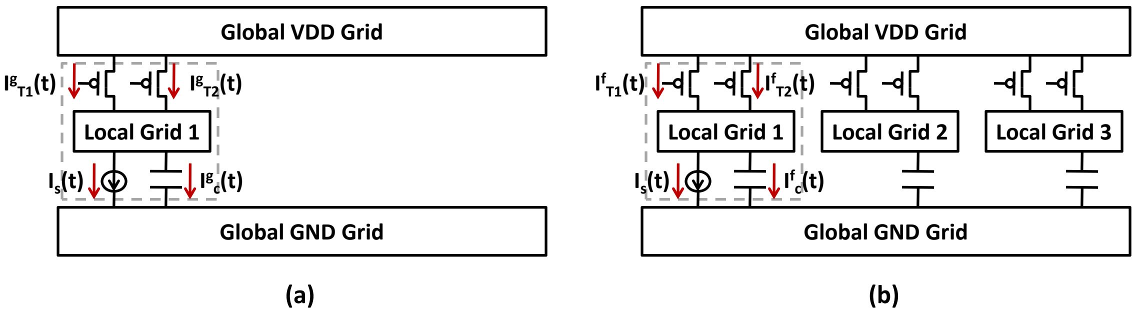

Fig. 30. Switchable current source values for the global basic configuration (a) and the full-decap basic configuration (b). For simplicity, the package is not shown and there are only three local grids.

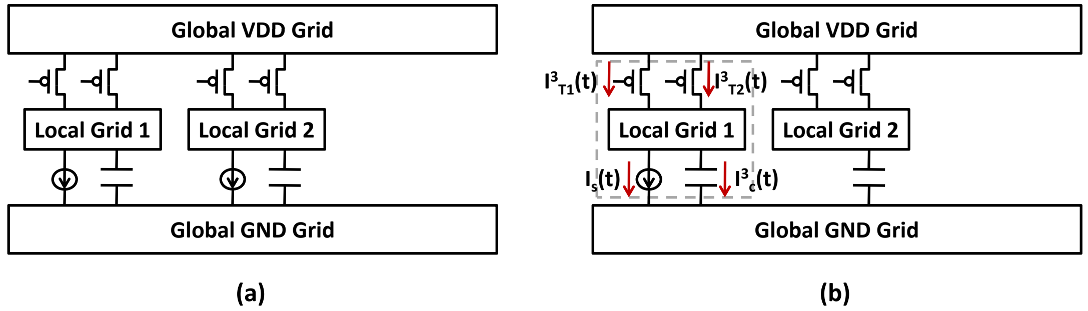

Fig. 31. A simple example for switchable current source approximation. The package is not shown and there are only two local grids.

Figure 30(b).

For a simple case of configuration $\mathscr{C}_3 = \{1, 1, 0\}$ shown in Figure 31(a), the value for the switchable current source $\mathbf{I_1^3(t)}$ (shown in Figure 31(b), the upper index 3 is the power gating configuration index) is approximated as a linear function of $\mathbf{I_1^g(t)}$ and $\mathbf{I_1^f(t)}$ in the form of

$$\mathbf{I_1^3(t)} \approx \mathbf{I_1^{3,apx}(t)} = \alpha \left(\mathbf{I_1^f(t)} - \mathbf{I_1^g(t)}\right) + \mathbf{I_1^g(t)} \quad (2.33)$$

$$\alpha = \left(C_1 + C_2 + C_3\right) C_2 / \left(C_2 + C_3\right) \cdot \left(C_1 + C_2\right). \quad (2.34)$$

(2.34) is obtained based on (2.29-2.30).

The immediate benefit of this approximation is that with each local grid modeled

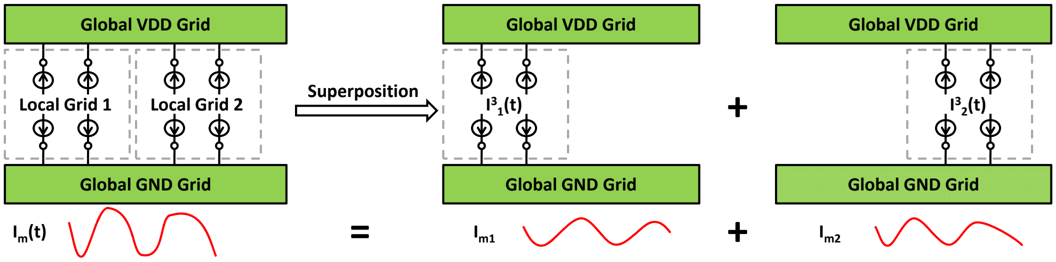

Fig. 32. Superposition for global grid verification.

as a set of switchable currents whose values can be approximated in the way as (2.33), the circuit responses on the global grids in any stable configuration $\mathscr{C}_i$ can be efficiently estimated using the principle of superposition [45]. As shown in Figure 32, for a wire w_m on the global grid under the stable configuration $\mathscr{C}=\{1,1,0\}$, its current $I_m^3(t)$ can be approximated as

$$I_m^3(t) = I_{m1}^3(t) + I_{m2}^3(t) \approx I_{m1}^{3,apx}(t) + I_{m2}^{3,apx}(t), \tag{2.35}$$

where $I_{m1}^{3,apx}(t)$ and $I_{m2}^{3,apx}(t)$ are the contributions from two current sources $\mathbf{I_1^{3,apx}(t)}$ and $\mathbf{I_2^{3,apx}(t)}$. Assume with current source $\mathbf{I_1^g(t)}$ and $\mathbf{I_1^f(t)}$, the currents on W_m are $I_{m1}^g(t)$ and $I_{m1}^f(t)$ respectively, then

$$I_{m1}^{3,apx}(t) = \alpha\left(I_{m1}^f(t) - I_{m1}^g(t)\right) + I_{m1}^g(t), \tag{2.36}$$

where α is the same as (2.34). $I_{m1}^g(t)$ and $I_{m1}^f(t)$ are obtained by simulations of the *global basic configuration* $C_1^g = \{1,0,0\}$ and the *full-decap basic configuration* $C_1^f = \{1,3,3\}$ respectively. The contribution $I_{m2}^{3,apx}(t)$ from current source $\mathbf{I_2^{3,apx}(t)}$ can be obtained in the similar way but with a different value for α ($\alpha = (C_1 + C_2 + C_3)\, C_1 / (C_1 + C_3) \cdot (C_1 + C_2)$ in this case).

In a more general PDN, assume there are N gated local grids (with indices from 1 to N). Under the configuration $\mathscr{C}_k = \{b_{1k}, \ldots, b_{Nk}\}$, for wire w_m in the global grid, the current contribution from local grid G_i can be represented as a time varying variable $I_{mi}^{k,apx}(t)$. Applying superposition, the average current (from time t_1 to t_2) flowing through the wire w_m can be approximated as

$$|EM_{mk}| \approx |EM_{mk,apx}| = \left| \sum_{i=1}^{N} b_{ik} \frac{\int_{t_1}^{t_2} I_{mi}^{k,apx}(t)dt}{t_2 - t_1} \right|, \tag{2.37}$$

$$I_{mi}^{k,apx}(t) = \alpha_{ki} \left(I_{mi}^{f}(t) - I_{mi}^{g}(t) \right) + I_{mi}^{g}(t), \tag{2.38}$$

$$\alpha_{ki} = \frac{\sum_{n=1}^{N} C_n \cdot \sum_{n=1}^{n=N:n\neq i} b_{nk} C_n}{\sum_{n=1}^{n=N:n\neq i} C_n \cdot \sum_{n=1}^{N} b_{nk} C_n}, \tag{2.39}$$

where $I_{mi}^{g}(t)$ and $I_{mi}^{f}(t)$ can be obtained by simulating the *global basic configuration* $\mathscr{C}_i^g$ and the *full-decap basic configuration* $\mathscr{C}_i^f$ respectively.

c. Worst Case Validation

As stated above, errors are introduced by approximating the local grids using independent current sources. The configuration that has the $|EM_{m,apx}|_{max}$ may not be the worst-case configuration. Therefore, a validation scheme, as shown in Figure 33, is needed. Instead of only selecting the approximate worst-case configuration, a set of configurations, $C_{t1}, \ldots, C_{tP}$, that correspond to the P largest approximate average currents, $|EM_{mt1,apx}|, \ldots, |EM_{mtP,apx}|$, are selected as the top-P worst-case EM configuration candidates. The top-P cases can be found by going through all 2^N possible configurations (rank them according to their approximate average currents) and picking up the P configurations that have the P largest approximate average currents. According to the experimental results, the runtime for identifying top-P worst cases is insignificant compared to lengthy die-package transient simulations, thus does not affect the overall complexity of our approach. Next, full simulations are applied to all these P candidate configurations. Finally, the real $|EM_m|_{max}$ is obtained by choosing the largest validated $|EM_{mti}|$ $(i = 1 \ldots P)$ ($|EM_{mv1}|$ in Figure

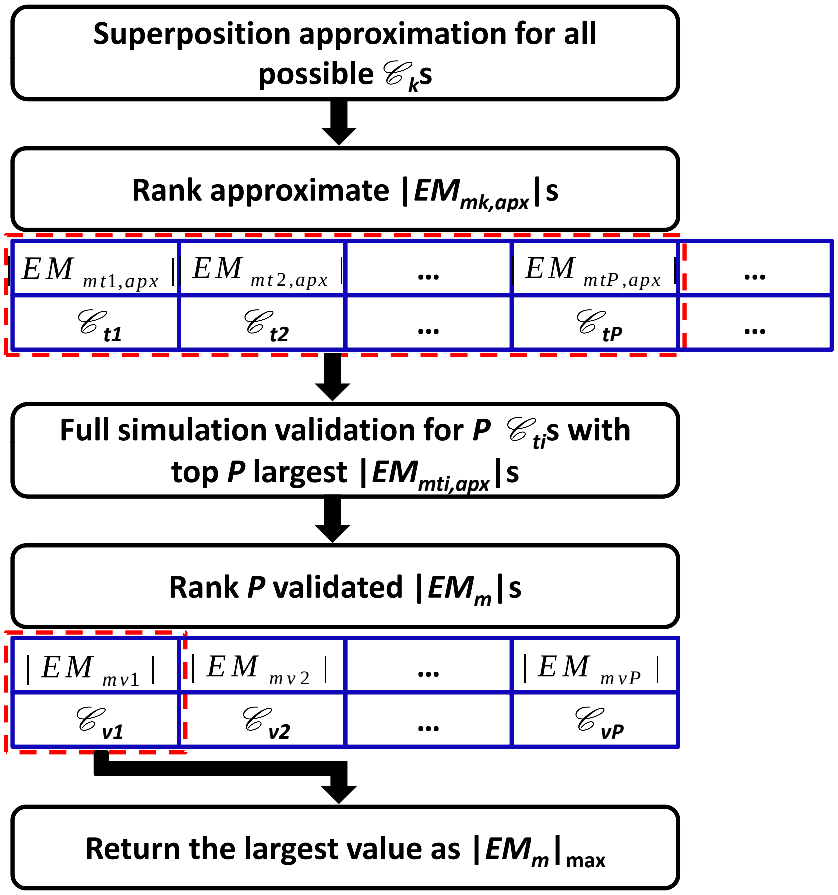
Superposition approximation for all possible $\mathscr{C}_k$s
Rank approximate $|EM_{mk,apx}|$s
$|EM_{mt1,apx}|$
$|EM_{mt2,apx}|$
...
$|EM_{mtP,apx}|$
...
$\mathscr{C}_{t1}$
$\mathscr{C}_{t2}$
...
$\mathscr{C}_{tP}$
...
Full simulation validation for P $\mathscr{C}_{ti}$s with top P largest $|EM_{mti,apx}|$s
Rank P validated $|EM_m|$s
$|EM_{mv1}|$
$|EM_{mv2}|$
...
$|EM_{mvP}|$
$\mathscr{C}_{v1}$
$\mathscr{C}_{v2}$
...
$\mathscr{C}_{vP}$
Return the largest value as $|EM_m|_{max}$

Fig. 33. Flow for worst case validation.

33), and its corresponding stable configuration can also be found. The number P can be increased to cover more possible configurations to assure that the actual $|EM_m|_{max}$ can be identified in the final validation simulations. Usually P is in the order of $O(N)$ and much smaller than 2^N. In contrast to exhaustive enumeration, the presented approach reduces the complexity from $O(2^N)$ to $O(N)$, leading to significant efficiency improvement. The entire algorithm for stable-mode EM verification is summarized in Algorithm 2.

Algorithm 2 Stable-mode EM verification for wire W_m in global grids

Input: Global basic stable configuration $\mathscr{C}_1^g \cdots \mathscr{C}_N^g$, full-decap basic stable configuration $\mathscr{C}_1^f \cdots \mathscr{C}_N^f$ and the number of worst-case candidates P.
Output: $|EM_m|_{max}$ and its corresponding stable configuration $\mathscr{C}_{max}$.

```
for $i \leftarrow 1$ to $N$ do
    Full simulation for the global basic stable configuration $\mathscr{C}_i^g$.
    Full simulation for the full-decap basic stable configuration $\mathscr{C}_i^f$.
end for
for $k \leftarrow 1$ to $2^N$ do
    Obtain $|EM_{mk,apx}|$ by (2.37).
end for
Rank $|EM_{mk,apx}|$s and obtain $P$ worst-case candidates $\mathscr{C}_{t1}, \ldots, \mathscr{C}_{tP}$.
for $i \leftarrow 1$ to $P$ do
    Full simulation for worst-case candidate $\mathscr{C}_{ti}$.
    $|EM_{mti}| = \left|\frac{\int_{t_1}^{t_2} I_m^{ti}(t)dt}{t_2-t_1}\right|$.
end for
$|EM_m|_{max} \leftarrow \max_{i=1}^{P} |EM_{mti}|$.
return $|EM_m|_{max}$ and its configuration as $\mathscr{C}_{max}$.
```

We have discussed how to identify the worst-case EM condition for a single wire. When it is required to identify the worst-case EM condition among all the wires in the global grid, for each configuration, the maximum approximate average current among all the wires in the global grid can be obtained from fast superposition approximation. Then, all the configurations are ranked according to their maximum approximate average current values, and the top-P worst case candidates can be found for validation. The extra handling required is just to find the maximum EM metric among all wires for each configuration, which does not require any additional full circuit simulations. This applies to other types of verification presented in the following subsections.

4. Other Verifications

a. Stable-Mode EM Verification for Local Grids

A modified approach is taken to perform the EM verification of a wire on a local grid. Under the context of power gating, this also implies that this local grid is always powered on. Similar to the stable-mode EM verification for global grids, in order to apply superposition approach, the *circuit* and the *inputs* should be identified. Different from the setup in Figure 32, since the target of verification is a particular local grid, this local grid is always included in the 'global' PDN circuit as shown in Figure 34, where the wire of verification is assumed to be in local grid 1. Therefore, the *circuit* includes the global grids, the off-chip circuit, the targeted local grid and its decoupling capacitors and resistor models for the turned-on sleep transistors. As stated in the previous verification, all other local grids are modeled as switchable current sources to the *circuit*. These current source models as well as the intrinsic current loads at the targeted local grid are treated as *inputs*. In Figure 34, when analyze the impact of $\mathbf{I_2(t)}$ on local grid 1, grid G_1 should be considered in the idle state. The contribution from its own current loads $\mathbf{I_{s1}(t)}$ is computed separately by only keeping grid G_1 active.

Similar to the *global basic stable configuration*, for a PDN with N independent gated local grids, assume the targeted wire is in local grid G_j, a *local basic stable*

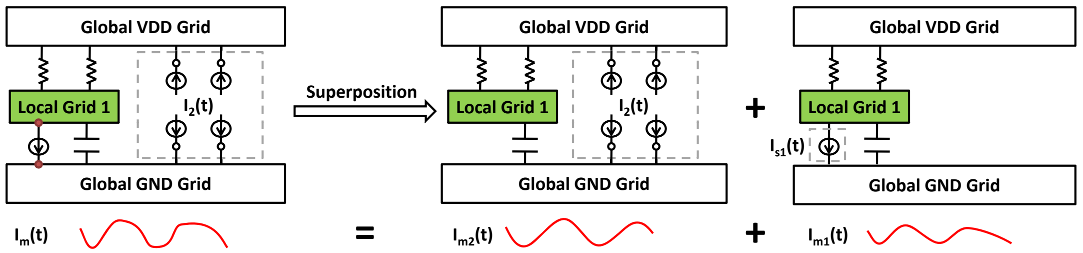

Fig. 34. Superposition for local grid verification.

configuration $\mathscr{C}_i^{lj}$ can be defined as,

$$\mathscr{C}_i^{lj} = \{b_{1i}, b_{2i}, \dots, b_{Ni}\},\ b_{ji} = 3, b_{ii} = 1, \text{others} = 0, \tag{2.40}$$

where grid G_j is idle, grid G_i is active and others are in sleep.

As shown in Figure 34, for a wire w_m on the local grid 1 under stable configuration $\{1, 1, 0\}$, its current $I_m^3(t)$ can be approximated by using superposition approach,

$$I_m^3(t) = I_{m1}^3(t) + I_{m2}^3(t) \approx I_{m1}^{3,apx}(t) + I_{m2}^{3,apx}(t), \tag{2.41}$$

where $I_{m1}^{3,apx}(t)$ is obtained from the *global basic configuration* $\mathscr{C}_1^g = \{1, 0, 0\}$ and the *full-decap basic configuration* $\mathscr{C}_1^f = \{1, 3, 3\}$; $I_{m2}^{3,apx}(t)$ is obtained by by linear combination of the contributions from the *local basic configuration* $C_2^{l1} = \{3, 1, 0\}$ and the *full-decap basic configuration* $\mathscr{C}_2^f = \{3, 1, 3\}$.

More generally, assuming the targeted wire w_m is on local grid G_j, the average current (from t_1 to t_2) under the configuration $\mathscr{C}_k = \{b_{1k}, \dots, b_{Nk}; b_{jk} = 1\}$ is approximated as follows,

$$|EM_{mk,apx}| = \left| \frac{\sum_{i=1}^{N;i\neq j} b_{ik} \int_{t_1}^{t_2} I_{mi}^{k,apx}(t)dt + \int_{t_1}^{t_2} I_{mj}^{k,apx}(t)dt}{t_2 - t_1} \right|, \tag{2.42}$$

$$I_{mi}^{k,apx}(t) = \alpha_{ki} \left(I_{mi}^{f}(t) - I_{mi}^{lj}(t) \right) + I_{mi}^{lj}(t), \tag{2.43}$$

$$\alpha_{ki} = \frac{(C_i + C_j) \cdot \sum_{n=1}^{N} C_n \cdot \sum_{n=1}^{n=N:n\neq i} b_{nk}C_n}{C_i \cdot \sum_{n=1}^{n=N:n\neq i,j} C_n \cdot \sum_{n=1}^{N} b_{nk}C_n}, \tag{2.44}$$

$$I_{mj}^{k,apx}(t) = \alpha_{kj} \left(I_{mj}^{f}(t) - I_{mj}^{g}(t) \right) + I_{mj}^{g}(t), \tag{2.45}$$

$$\alpha_{kj} = \frac{\sum_{n=1}^{N} C_n \cdot \sum_{n=1}^{n=N:n\neq j} b_{nk}C_n}{\sum_{n=1}^{n=N:n\neq j} C_n \cdot \sum_{n=1}^{N} b_{nk}C_n}, \tag{2.46}$$

where $i = 1 \dots N$ and $i \neq j$; $I_{mi}^{lj}(t)$ is obtained by simulating the *local basic configuration* $\mathscr{C}_i^{lj}$ and $I_{mi}^{f}(t)$ is obtained by simulating the *full-decap basic configuration* $\mathscr{C}_i^{f}$; whereas $I_{mj}^{g}(t)$ is obtained by simulating the *global basic configuration* $\mathscr{C}_j^{g}$ and $I_{mj}^{f}(t)$ is obtained by simulating the *full-decap basic configuration* $\mathscr{C}_j^{f}$.

The procedures of identifying top-P worst-case candidates and the following full simulation validation illustrated in Algorithm 2 can be applied here to find the maximum average current $|EM_m|_{max}$ for the wire w_m in a local grid.

b. Stable-Mode Peak Dynamic Voltage Drop Verification

Here, the task is to find the largest peak dynamic voltage drop for a circuit block in the on-chip PDN under all possible stable power gating configurations. It should be noted that the current loadings are assumed to be given. Therefore, the uncertainty of the current profile is not considered under this scope.

For a simple PDN circuit shown in Figure 35, a circuit block T_n is connected to local grid 1 at node a and to the global GND at node b. Assume the voltages at a and b are $V_a(t)$ and $V_b(t)$ respectively. The dynamic voltage drops of node a and b

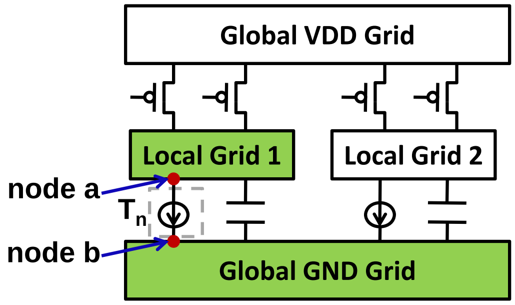

Fig. 35. Dynamic voltage drop for circuit block T_n. For simplicity, the package is not shown.

can be expressed as,

$$DVD_a(t) = V_{DD} - V_a(t), \text{ for VDD grid nodes,} \tag{2.47}$$

$$DVD_b(t) = -V_b(t), \text{ for GND grid nodes.} \tag{2.48}$$

Therefore, the dynamic voltage drop for circuit T_n is,

$$DVD_n(t) = V_{DD} - (V_a(t) - V_b(t)) = DVD_a(t) - DVD_b(t), \tag{2.49}$$

According to the PDN model in Figure 21, all the circuit blocks reside between local grids and the global GND grid. Therefore, the voltage of the nodes in local grids must be considered in the DVD verification, which implies that the targeted local grids should be always on. Since in this case the *circuit* and *inputs* categorization is the same as in stable-mode EM verification for the local grids, the current source modeling, approximation method, worst-case identification scheme and validation procedure presented in Section II.C.4.a can be applied here. It should be noted that in order to use the superposition theorem, voltage drop (defined in (2.47) and (2.48)) is used here instead of the actual node voltage.

In a general PDN with N independent gated local grids, assuming the peak DVD (from t_1 to t_2) of a circuit block T_n which is connected to the node a on local grid

G_j and the node b in the global GND grid, is examined. Under the configuration $\mathscr{C}_k = \{b_{1k}, \ldots, b_{Nk}; b_{jk} = 1\}$, the voltage drop at node a and b can be approximated by,

$$DVD^k_{a,apx}(t) = \textstyle\sum_{i=1}^{N;i\neq j} b_{ik} DVD^k_{ai,apx}(t) + DVD^k_{aj,apx}(t), \tag{2.50}$$

$$DVD^k_{b,apx}(t) = \textstyle\sum_{i=1}^{N;i\neq j} b_{ik} DVD^k_{bi,apx}(t) + DVD^k_{bj,apx}(t), \tag{2.51}$$

where $DVD_{aj}(t)$ and $DVD_{bj}(t)$ are defined in (2.47) and (2.48). Similar to the transient current on a wire in the local grids, we have

$$DVD^k_{ai,apx}(t) = \alpha_{ki}\left(DVD^f_{ai}(t) - DVD^{lj}_{ai}(t)\right) + DVD^{lj}_{ai}(t), \tag{2.52}$$

$$\alpha_{ki} = \frac{(C_i + C_j)\cdot \sum_{n=1}^{N} C_n \cdot \sum_{n=1}^{n=N:n\neq i} b_{nk}C_n}{C_i \sum_{n=1}^{n=N:n\neq i,j} C_n \cdot \sum_{n=1}^{N} b_{nk}C_n}, \tag{2.53}$$

$$DVD^k_{aj,apx}(t) = \alpha_{kj}\left(DVD^f_{aj}(t) - DVD^g_{aj}(t)\right) + DVD^g_{aj}(t), \tag{2.54}$$

$$\alpha_{kj} = \frac{\sum_{n=1}^{N} C_n \cdot \sum_{n=1}^{n=N:n\neq j} b_{nk}C_n}{\sum_{n=1}^{n=N:n\neq j} C_n \cdot \sum_{n=1}^{N} b_{nk}C_n}, \tag{2.55}$$

where $i = 1 \ldots N$, $i \neq j$; $DVD^{lj}_{ai}(t)$ is obtained by simulating the *local basic configuration* $\mathscr{C}^{lj}_i$ and $DVD^f_{ai}(t)$ is obtained by simulating the *full-decap basic configuration* $\mathscr{C}^f_i$; whereas $DVD^g_{aj}(t)$ is obtained by simulating the *global basic configuration* $\mathscr{C}^g_j$ and $DVD^f_{aj}(t)$ is obtained by simulating the *full-decap basic configuration* $\mathscr{C}^f_j$. $DVD^{k,apx}_{bi}(t)$ $(i = 1 \ldots N; i \neq j)$ and $DVD^{k,apx}_{bj}(t)$ can be obtained in the same way.

Therefore, the peak dynamic voltage drop for T_n under configuration $\mathscr{C}_k$ is approximated as

$$DVD^k_{Pn,apx} = \max_{t_1 \leq t \leq t_2} \left\{DVD^k_{a,apx}(t) - DVD^k_{b,apx}(t)\right\}. \tag{2.56}$$

Similar to finding the top-P worst-case configuration candidates for the stable-mode EM verification, the top-P worst-case configuration candidates for the peak

DVD verification can also be found. The full simulation validation will be carried out for these candidates to find out the maximum DVD_{Pn}.

Unlike the average current (a time-average effect), dynamic voltage drop (a transient phenomenon) is sensitive to the total decoupling capacitance in the configuration, so more errors (compared to the exact peak DVD) are expected from the linear approximation in (2.50-2.51). However, it should be noted that the main purpose of the superposition approximation is not to obtain the exact peak DVD values for all the configurations, but to explore the relative rankings among different configurations. The approximate peak DVD ($DVD^{k}_{Pn,apx}$s) are only used to rank their corresponding power gating configurations ($\mathscr{C}_k$s) and to select the top-P worst-case candidates. As long as the true worst-case (or near worst-case for small P) configuration is included in the top P of this ranking or the ranking trend is preserved, the proposed approach is effective, which is validated by our experimental results. Since for each node, under each basic configuration, the entire dynamic wave form has to be stored, we only choose a small amount of nodes to do the verification for the entire grid. These nodes have large dynamic voltage drops in the basic configurations and they are expected to have large DVD in other configurations as well.

c. Power-On Peak Dynamic Voltage Drop Verification

When a local grid is in transition, although no switching activity has been experienced from the devices powered by the grid yet, large rush currents may be drawn from the global grids to charge the local grid's decoupling capacitors which are discharged in the sleep state due to leakage. Such current disturbances may propagate through the global grids and cause drops on the power and ground lines of other local grids. The turn-on time, a critical design variable, can be properly chosen to control the amount of generated power-on noise. While a longer turn-on time is beneficial from

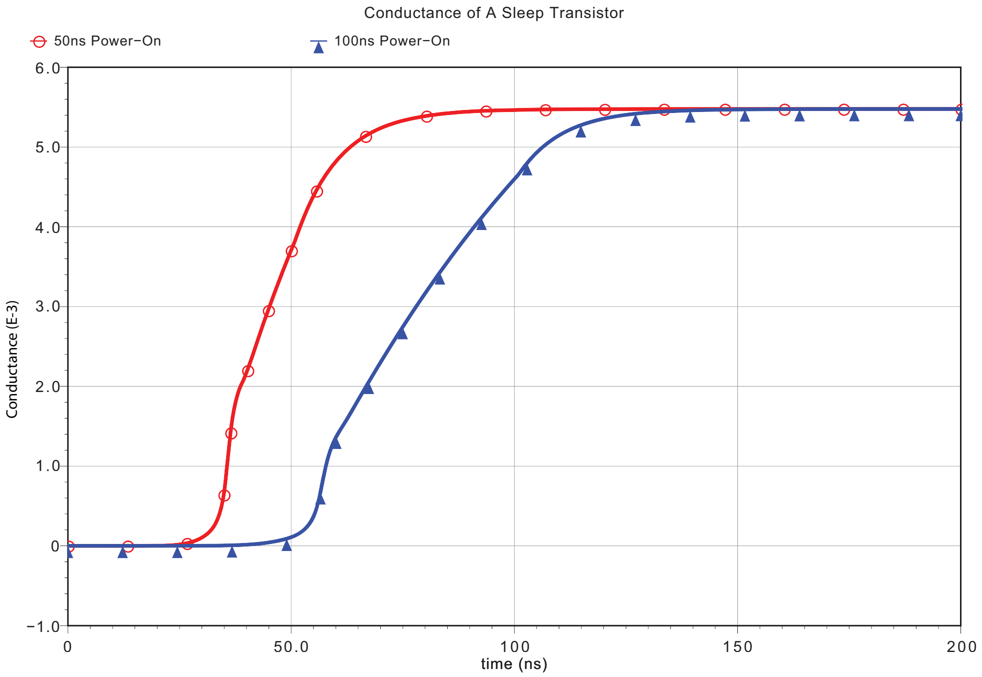

Fig. 36. Drain-source conductance of a PMOS sleep transistor during power-on time.

the noise point of view, it nevertheless increases the timing overhead and prevents more effective use of power gating. When a sleep transistor is turned on, at first it works in the saturation region and then goes to linear region which leads to the drain-source conductance variation along time (as shown in Figure 36). Therefore, during the turn-on procedure, the sleep transistor can be simply modeled as time-varying resistor.

For a given turn-on time, the task of power-on peak dynamic voltage drop verification is to identify the worst-case peak voltage drop caused by the turn-on noise in conjunction with the noise contributions from all possible stable workloads. For the purpose of illustration, the case in which only one local grid is powered on at a time is examined, as shown in Figure 37. The cases with multiple power-on local grids can be handled in a similar way. The grid in transition can be modeled as rush current

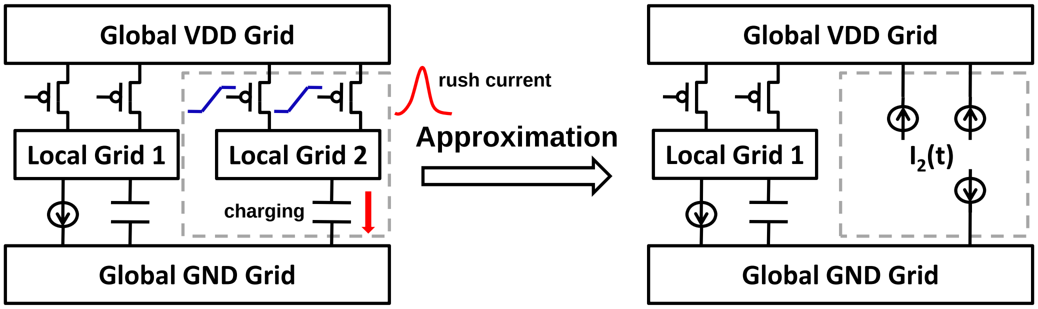

Fig. 37. Verification of power-on peak dynamic voltage drop.

sources which are a part of the *inputs* to the *circuit*, thus the superposition technique can be applied for peak DVD approximation.

For a general PDN with N independent gated local grids, assume the targeted grid is G_j and G_s is in transition, we introduce a *basic transition configuration* $\mathscr{D}_s^{tj}$ which is defined as

$$\mathscr{D}_s^{tj} = \{b_{1s}, b_{2s}, \ldots, b_{Ns}\},\ b_{js} = 3, b_{ss} = 2, \text{others} = 0, \tag{2.57}$$

where grid G_s is in transition, G_j is in idle, and others are in sleep.

Assuming the peak DVD (from t_1 to t_2) of a circuit block T_n connected to the local grid G_j as shown in Figure 35 is examined. Local grid G_s is in transition. Under the transition configuration $\mathscr{D}_k = \{b_{1k}, \ldots, b_{Nk}; b_{jk} = 1, b_{sk} = 2\}$, the peak voltage drop at node a and b can be approximated by,

$$DVD^k_{Pa,apx}(t) = \textstyle\sum_{i=1}^{N;i\neq j,s} b_{ik} DVD^k_{ai,apx}(t) + DVD^k_{aj,apx}(t) + DVD^k_{as,apx}(t), \tag{2.58}$$

$$DVD^k_{Pb,apx}(t) = \textstyle\sum_{i=1}^{N;i\neq j,s} b_{ik} DVD^k_{bi,apx}(t) + DVD^k_{bj,apx}(t) + DVD^k_{as,apx}(t), \tag{2.59}$$

where we have

$$DVD^{k}_{ai,apx}(t) = \alpha_{ki}\left(DVD^{f}_{ai}(t) - DVD^{lj}_{ai}(t)\right) + DVD^{lj}_{ai}(t), \tag{2.60}$$

$$\alpha_{ki} = \frac{(C_i + C_j)\cdot\sum_{n=1}^{N} C_n \cdot \left(\sum_{n=1}^{n=N:n\neq i,s} b_{nk}C_n + C_s\right)}{C_i \cdot \sum_{n=1}^{n=N:n\neq i,j} C_n \cdot \left(\sum_{n=1}^{n=N:n\neq s} b_{nk}C_n + C_s\right)}, \tag{2.61}$$

$$DVD^{k}_{aj,apx}(t) = \alpha_{kj}\left(DVD^{f}_{aj}(t) - DVD^{g}_{aj}(t)\right) + DVD^{g}_{aj}(t), \tag{2.62}$$

$$\alpha_{kj} = \frac{\sum_{n=1}^{N} C_n \cdot \left(\sum_{n=1}^{n=N:n\neq j,s} b_{nk}C_n + C_s\right)}{\sum_{n=1}^{n=N:n\neq j} C_n \cdot \left(\sum_{n=1}^{n=N:n\neq s} b_{nk}C_n + C_s\right)}, \tag{2.63}$$

$$DVD^{k}_{as,apx}(t) = DVD^{tj}_{as}(t), \tag{2.64}$$

where $i = 1 \ldots N$, $i \neq j, s$; $DVD^{lj}_{ai}(t)$ is obtained by simulating the *local basic configuration* $\mathscr{C}^{lj}_{i}$ and $DVD^{f}_{ai}(t)$ is obtained by simulating the *full-decap basic configuration* $\mathscr{C}^{f}_{i}$; $DVD^{g}_{aj}(t)$ is obtained by simulating the *global basic configuration* $\mathscr{C}^{g}_{j}$ and $DVD^{f}_{aj}(t)$ is obtained by simulating the *full-decap basic configuration* $\mathscr{C}^{f}_{j}$; $DVD^{tj}_{as}(t)$ is obtained by simulating the *basic transition configuration* $\mathscr{D}^{tj}_{s}$. $DVD^{k}_{bi,apx}(t)$ ($i = 1 \ldots N; i \neq j, s$), $DVD^{k}_{bj,apx}(t)$ and $DVD^{k}_{bs,apx}(t)$ can be obtained in the same way.

Therefore, the peak dynamic voltage drop for T_n under transition configuration $\mathscr{C}_k$ is approximated as

$$DVD^{k}_{Pn,apx} = \max_{t_1 \leq t \leq t_2}\left\{DVD^{k}_{Pa,apx}(t) - DVD^{k}_{Pb,apx}(t)\right\}. \tag{2.65}$$

Following the scheme shown in Algorithm 2, the $DVD_{Pn,max}$ can be found.

5. Experimental Results

The PDN simulator GSim and the transient power gating verification flow have been implemented in CUDA [44] and C++, respectively. The GPU program is executed on a single GPU of the NVIDIA Geforce 9800 GX2 card (including two GPUs), with a total on board memory of 512Mb. All the C++ programs are executed on a workstation with Intel Xeon CPU at 2.33GHz and 4G RAM running 64-bit Linux OS. The on-chip power grids of PDNs are generated according to the typical current loadings and wire conductance of the IBM power grid benchmarks [39], while the package level model parameters, such as inductance and capacitance values, as well as total on-chip capacitance are adopted from [6].

Three power gated million-node PDNs with N gated local grids, $N = 4, 8, 16$ are employed for the transient verification presented in the paper. Each PDN has millions of on-chip nodes and a few hundred chip-to-package pins. Each local grid has several blocks with different current loadings to represent different function modules. 200 clock cycles (2000 time steps) are simulated to capture the time averaging and temporal effects (waveforms are in steady state).

a. Stable-Mode Verification

The stable-mode EM verifications have been carried out for three multi-million-node PDNs. The run time and the largest $|EM|$ obtained by choosing P as 4, 8, 12 are shown in Table VII. The largest EM among all the wires in the global VDD grid is examined. As expected, even for small P, the EM worst-case configuration can be captured very effectively. Moreover, in terms of runtime, for the PDNs with a large number of gated grids, the number of possible stable configurations is very large, which leads to excessive runtime for brute-force enumeration (107,000 hours for the largest case!). However, by using the proposed verification scheme, only a small number of full simulations need to be carried out, therefore, the runtime has been greatly reduced (only a couple of hours for the largest grid).

The stable-mode peak DVD verification results shown in Table VIII demonstrate a similar behavior to the EM verifications. Still, the largest peak DVD can be well

Table VII. EM stable mode verification for gated PDN. The global VDD grid is examined. # Con.: number of configurations; T: total runtime; $|EM|_{max}$: maximum absolute average current; P: number of worst case candidates. Runtime is in hrs. EM is in mA. Runtime of the enumeration methods for larger circuits are the estimated time ($\sim$ time value). The transient verifications are run for 200 clock cycles (2000 time steps).

Gated	# Con.	# Nodes	Enumeration		P=4		P=8		P=12	
Grids			T	$\|EM\|_{max}$	T	$\|EM\|_{max}$	T	$\|EM\|_{max}$	T	$\|EM\|_{max}$
4	15	2.25M	6.69	4.75	4.72	4.75	6.41	4.75	9.20	4.75
8	255	4.25M	~ 214.8	NA	8.31	2.40	11.77	2.40	14.0	2.40
16	65535	8.25M	$\sim 1.07e5$	NA	16.17	2.33	17.75	2.33	21.31	2.33

Table VIII. Peak DVD stable mode verification for gated PDN. For grids with 8 and 16 grids, two different local grids are examined. Gcor: corner grid; Gcen: center grid; # Con.: number of configurations; T: total runtime; $DVD_{p,max}$: maximum peak DVD; P: number of worst case candidates. Runtime is in hrs. DVD is in mV. Runtime of the enumeration methods for larger circuits are the estimated time ($\sim$ time value). The transient verifications are run for at least 200 clock cycles (2000 time steps).

Gated Grids	# Con.	# Nodes	Node	Enumeration		P=4		P=8		P=12	
				T	$DVD_{p,max}$	T	$DVD_{p,max}$	T	$DVD_{p,max}$	T	$DVD_{p,max}$
4	8	2.25M		4.82	131.7	4.92	131.7	7.00	131.7	NA	NA
8	128	4.25M	Gcor	~ 141.9	NA	9.6	124.3	13.25	124.3	16.54	124.3
			Gcen	~ 141.9	NA	9.47	128.4	13.12	128.4	16.57	128.4
16	32768	8.25M	Gcor	$\sim 7.06e4$	NA	19.21	117.1	27.54	117.1	35.01	117.1
			Gcen	$\sim 7.06e4$	NA	19.23	121.9	26.45	121.9	34.22	121.9

Table IX. Peak DVD transition verification for gated PDN. The transition grid is at the center. For grids with 8 and 16 grids, two nodes in two different grids are examined. Gclo: the grid close to the transition grid; Gfar: the grid far away from the transition grid; # Con.: number of configurations; T: total runtime; $DVD_{p,max}$: maximum peak DVD; P: number of worst case candidates. Runtime is in hrs. DVD is in mV. Runtime of the enumeration methods for larger circuits are the estimated time ($\sim$ time value). The transient verifications are run for at least 200 clock cycles (2000 time steps).

Gated Grids	# Con.	# Nodes	Node	Enumeration		P=4		P=8		P=12	
				T	$DVD_{p,max}$	T	$DVD_{p,max}$	T	$DVD_{p,max}$	T	$DVD_{p,max}$
4	4	2.25M		2.62	125.4	5.01	125.4	NA	NA	NA	NA
8	64	4.25M	Gclo	~ 79.28	NA	9.54	122.7	13.03	122.7	17.0	122.7
			Gfar	~ 79.28	NA	9.69	122.3	13.10	122.3	17.32	122.3
16	16384	8.25M	Gclo	$\sim 3.94e4$	NA	19.40	128.6	27.0	128.7	35.9	128.7
			Gfar	$\sim 3.94e4$	NA	19.04	117.8	26.48	117.9	34.57	117.9

captured by the proposed approach. 20 nodes (the one that have large DVD in the basic configurations) are chosen to represent all the nodes in a local grid. The largest DVD among them is examined. The runtime saving over brute-force enumeration for PDNs with a large number of gated grids is huge, estimated as over 2000X for the largest case when $P = 12$.

b. Power-On Verification

In terms of power-on verification, without loss of generality, a gated grid at the center is chosen to be in transition state. A grid near the transition and a grid far away from the transition are chosen to be examined. Similar to stable-mode peak DVD verification, for each of the grid, we use 20 nodes (the ones that have large DVD in the basic configurations) to present all the nodes. As can be seen from the results shown in Table IX, similar to the stable mode verification, the proposed method can effectively capture the largest or near largest peak DVD by simulating only a small number of configurations. Note that for the PDN with 4 gated grids, the total number of configuration is only 4, therefore, the results for $P = 8$ and $P = 12$ are marked as NA.

6. Summary

In this section, we propose a simulation-based transient verification approach. Specific circuit modeling techniques have been developed to individually verify each of the on-chip global and local power grids against given electromigration and voltage drop constraints. The proposed approach allows the use of fast superposition approximation methods to identify the top worst-case conditions that are validated by a small number of full simulations to achieve feasibility.

CHAPTER III

POWER DELIVERY NETWORK DESIGN *

As stated in Chapter I, the PDN design faces the challenges of saving metal wires for signal routing and choosing the optimal parameters for various components of the network. To address these challenges, in this chapter, a novel partitioning-based two-step power grid wire sizing approach is proposed which has the capability of utilizing parallel computing resources to improve efficiency. Then, systematic analysis to investigate the important electric interactions between active regulators/converters and passive networks under the entire power delivery context is conducted. Based on the insights obtained from the analysis, a system-level co-design scheme that can automatically find the optimal parameters for important network components is illustrated.

A. Locality-Driven Parallel Power Grid Wire Sizing

The power grid wire sizing can hardly be applied to large grids due to its inefficiency. In this section, by novelly reformulating the wire sizing problem into a two-step optimization problem, an efficient parallel optimization methodology using the locality of the flip-chip type power grids is presented.

*Part of the chapter is reprinted with permission from "Locality-driven parallel power grid optimization" by Z. Zeng and P. Li, 2009. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 28, pp. 1190-1200, Copyright [2009] by IEEE.

1. Background

a. Problem Formulation

In a power grid $\mathbf{\Omega}$ consisting of nodes $N = \{n_1, ..., n_k\}$ and branches $R = \{r_1, ..., r_l\}$, the node voltage for n_i is V_i. Each branch r_i, whose width and length are w_i and l_i respectively, connects two nodes n_{i1} and n_{i2}. The branch current for r_i is I_i, and voltage of the two end nodes are V_{i1} and V_{i2}. There are m metal layers. R_k $(k = 1, \cdots, m)$ includes all the branches on metal layer k and ρ_k is the sheet resistance for metal layer k. The branch conductance can be expressed as: $g_{ki} = w_{ki}/(l_{ki}\rho_k) = I_{ki}/(V_{ki1} - V_{ki2})$. Therefore, the area of the power grid, which is the objective function that should be minimized, can be expressed in terms of the branch conductance [46].

$$f(\mathbf{G}) = \sum_{i \in B} w_i l_i = \sum_{k=1}^{m} \sum_{i \in R_k} \rho_k g_{ki} l_{ki}^2. \tag{3.1}$$

Since using branch conductance $\mathbf{G}$ to compute the node voltage $\mathbf{V}$ directly consumes too much time and computing resources, we introduce node voltages as variables to the formulation to help us define the property of the circuit. Similar to [27], the constraints are as follows.

1. IR drop constraint:

$$V_i \geq V_{min}. \tag{3.2}$$

2. Minimum width constraint:

$$w_{ki} = \rho_k g_{ki} l_{ki} \geq w_{k,min}. \tag{3.3}$$

3. Current density constraint (electromigration): For a layer k, the electromigration constraint is $I_{ki} \leq \sigma_k w_{ki}$, where σ_k is the Electromigration constant.

$$-\rho_k \sigma_k l_{ki} \leq V_{ki1} - V_{ki2} \leq \rho_k \sigma_k l_{ki}. \tag{3.4}$$

4. Kirchhoff's current law: Assume that the branches connecting node n_j form the set $\mathbf{R}_j$, and the current loading at n_j is I_{sj}.

$$\sum_{b_i \in \mathbf{R}_j} (V_{i1} - V_{i2}) g_i + I_{sj} = 0. \tag{3.5}$$

The area optimization is to minimize (3.1) subject to constraints defined in (3.2)-(3.5). This formulation leads to a constrained nonlinear optimization problem.

b. Constrained Nonlinear Optimization

There exists a large body of general nonlinear optimization algorithms that can be applied to solve our power grid sizing problem. In recent years, interior point type methods have been particularly popular for large-scale nonlinear programming and found their application in circuit optimization [47] [48]. In [48], a novel state-of-art interior point (or barrier) algorithm has shown better robustness and efficiency than the augmented Lagrangian active-set method.

By introducing slack variables, a general nonlinear optimization problem can be formulated as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) && (3.6a) \\ \text{s.t.} \quad & c(x) = 0 && (3.6b) \\ & x \geq 0, && (3.6c) \end{aligned}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$, and the equality constraint functions $c : \mathbb{R}^n \to \mathbb{R}^m$ with $m < n$, are all assumed twice continuously differentiable.

The optimum is found by solving a sequence of barrier problems (3.7a-b) with a set of decreasing barrier parameters μ_l whose limit is zero ($\lim_{l\to\infty}\mu_l = 0$) [47].

$$\min_{x\in\mathbb{R}^n} \quad \varphi_{\mu_l}(x) = f(x) - \mu_l \sum_{i=1}^{n} \ln(x_i) \tag{3.7a}$$

$$\text{s.t.} \quad c(x) = 0. \tag{3.7b}$$

In addition to constrained nonlinear optimization algorithms, the power grid optimization may also be achieved by solving a sequence of approximated linear problems, as demonstrated in [27] or a set of unconstrained Lagrangian penalty problems [49]. It is important to note that in practice any of these flat optimization methods is difficult to apply to solve real-life power grid problems.

2. Overview of the Proposed Parallel Optimization

Today's on-chip power distribution networks can reach a complexity of millions of nodes or even greater. The design of such large networks via flat optimization is simply infeasible even if state-of-the-art optimization methods are employed. Hence, a partitioning based strategy, in which a large network is divided into manageable pieces that are optimized individually, is desirable to provide a scalable solution as well as to utilize the increasing parallel computing resources to improve efficiency.

a. Key Issues in Parallelizable Power Grid Optimization

Nevertheless, it is nontrivial to devise a systematic divide-and-conquer methodology. To see this, consider a naive approach as shown in Figure 38. A large power grid is directly cut into smaller partitions. Each partition is then wire sized to minimize the wiring area. Finally, the optimized partitions are merged to form a complete solution. Although cutting the wires between partitions leads to multiple independent smaller optimization problems, the negligence of coupling between partitions may lead to two

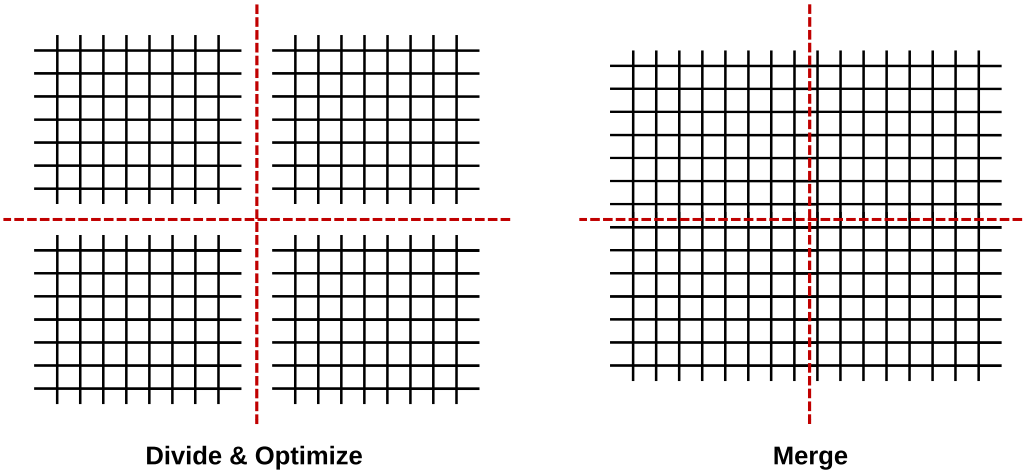

Fig. 38. A naive divide-and-conquer optimization approach.

key difficulties for the merged solution:

- Even if each partition is sized optimally, the merging of several *locally* optimized partitions does not necessarily correspond to an overall optimal solution. This is because that the sizing optimization is limited within each cut partition and lacks the global view.

- The merged solution is not guaranteed to be even feasible. Although the IR drop and EM constraints can be strictly enforced while independently optimizing each partition, the interconnections between the partitions can alter the voltage and current distributions when the partitions are merged to form a complete solution.

This creates a practically messy problem, which may create constraint violations possibly not only around the partitioning boundaries but also throughout the entire grid.

Consider another *divide-and-conquer* approach illustrated in [29]. In this approach, a partition of a large power grid is optimized and the rest of the grid is

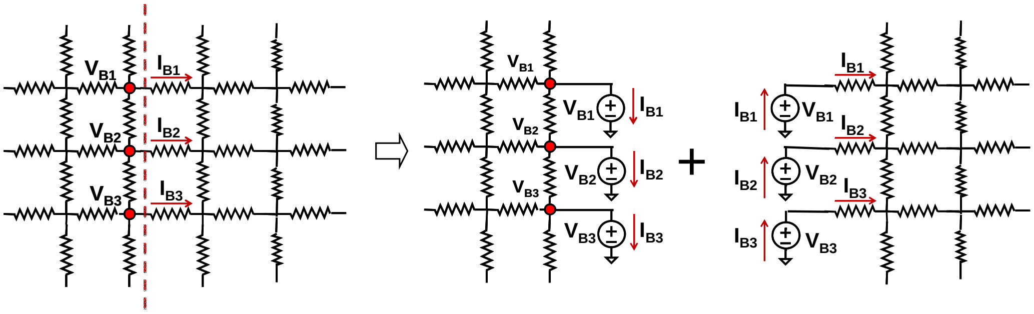

Fig. 39. Power grid partitioning by setting boundary voltages and currents.

represented by a compact macromodel [16]. This approach shows another potential difficulty: It does not immediately permits simultaneous sizing of multiple partitions while guaranteeing the feasibility and optimality of the complete solution.

It is clear that under the context of this work, a desirable partition-based optimization shall have three essential characteristics: it is parallelizable, and it maintains the feasibility and optimality of the final solution.

b. Two-Level Hierarchical Optimization

In the following, we present a two-level hierarchical optimization formulation which possesses key characteristics relevant to parallel optimization of large power grids. However, as will be discussed shortly after, this formulation has limitations that prevents its practical application. Nevertheless, its new hierarchical perspective underpins the proposed parallel locality-driven optimization approach that will be presented in the next subsection.

Instead of creating independent partitions by cutting wires, a different approach is taken, which is based upon setting voltages and currents along the partitioning boundaries as shown in Figure 39. Each partition is wire sized by considering these additional voltage and current constraints at the boundary. The usefulness of this partitioning scheme is two fold:

- First, attaching ideal voltage sources along the partitioning boundary electrically isolates each partition, making simultaneous independent optimization of partitions possible.
- Second, constraining the boundary voltages and currents for each partition helps the merging operation to keep the circuit responses in both partitions untouched, thus lead to a feasible solution for the entire grid.

The above procedure optimizes the grid under a given boundary condition (i. e., $V_{B,i}$'s and $I_{B,i}$'s). As a result, the reached solution may not be globally optimal. To address this problem, a two-level hierarchical optimization is introduced where the boundary conditions correspond to the first-level optimization variables. It is assumed that the power grid is divided into N partitions. The problem formulation for the first level is

Level 1:

$$\begin{aligned} \min_{V_B, I_B \in \mathbb{R}^n} \quad & A(V_B, I_B) && \text{(3.8a)} \\ \text{s.t.} \quad & A = \sum_{i=1}^{N} A_{i,min} && \text{(3.8b)} \\ & V_B \geq V_{min} && \text{(3.8c)} \\ & I_{kB} \leq \sigma_k w_{kB}, k = 1, \cdots, m, && \text{(3.8d)} \end{aligned}$$

where the boundary conditions between the N partitions are defined by n boundary voltages and currents: $V_B, I_B \in \mathbb{R}^n$; A is the total wire area of the power grid, which is an implicit function in V_B and I_B; and $A_{i,min}$ is the minimal wire area for partition i achievable under its boundary condition $V_{B,i}$ and $I_{B,i}$, which are subsets of V_B and I_B, respectively; the wire widths on the boundaries are denoted as w_B; (3.8c) and

(3.8d) enforce the IR drop and EM constraints at the partitioning boundaries. Each $A_{i,min}$ is obtained by solving one of N second level optimization problems

Level 2:

$$
\begin{aligned}
&\text{for each } i,\ i = 1, \cdots, N \\
&\min_{w_i \in \mathbb{R}^m} \quad A_i(w_i) && \text{(3.9a)} \\
&\text{s.t.} \quad (3.2), (3.3), (3.4), (3.5) && \text{(3.9b)} \\
&\qquad \mathbb{V}_{B,i} = V_{B,i},\ \mathbb{I}_{B,i} = I_{B,i} && \text{(3.9c)}
\end{aligned}
$$

where (3.9b) sets the standard constraints for the internal of partition i; $\mathbb{V}_{B,i}$ and $\mathbb{I}_{B,i}$ are partition i's boundary conditions, which are set by their counterparts passed from the first level problem.

As can be seen, this hierarchical optimization and the flat optimization share the same set of constraints and also the two objective functions are effectively identical. Therefore, the feasible solution for the hierarchical optimization is also a feasible solution to the flat optimization, and vice versa. If both formulations are solved to reach the global optimal solutions, it can be seen that they must reach the same optimal objective function value. Essentially, the first-level problems seeks the optimal V_B^* and I_B^* that lead to the overall area minimization while forcing the IR drop and EM constraints at the partitioning boundaries. When it comes to evaluate the total area achieved at given V_B and I_B, a set of N second level optimization problems are solved and the resulting areas of all the partitions are summed up. Mathematically, this two-level hierarchical problem formulation has appealing characteristics for scalable parallel optimization, since for each level, the problem dimension has been significantly reduced. Furthermore, the use of boundary voltages and currents based partitioning makes the N second level optimization problems completely independent of each other, allowing straightforward parallelization.

Unfortunately, there exist two limitations that prevent a practical application of this formulation:

- The hierarchical nature of the formulation implies that a large number of the second level optimization problems need to be solved. Therefore, a brute-force application of the two-level optimization can lead to excessive runtime.
- In each second-level optimization problem, setting the boundary voltages and currents as in (3.9c) may make the problem overly constrained, which is particularly troublesome as the unknown optimal boundary conditions are searched by the first-level problem.

c. Proposed Parallel Two-Step Optimization Formulation

We address the two issues with the two-level hierarchical formulation, runtime efficiency and convergence, by adopting a much more practical parallel two-step formulation. For clarity, the problem formulation is stated as

Step 1:

$$\text{Solve for the optimal } V_B^* \text{ and } I_B^* \text{ in parallel.} \tag{3.10}$$

Step 2:

$$\text{for each } i,\ i = 1, \cdots, N$$

$$\min_{w_i \in \mathbb{R}^m} \quad A_i(w_i) \tag{3.11a}$$

$$\text{s.t.} \quad (3.2), (3.3), (3.4), (3.5) \tag{3.11b}$$

$$C_R(\mathbb{V}_{B,i}, \mathbb{I}_{B,i}, V_{B,i}^*, I_{B,i}^*) \tag{3.11c}$$

$$A_{tot,min} = \sum_{i=1}^{N} A_{i,min}, \tag{3.11d}$$

where $C_R(\mathbb{V}_{B,i}, \mathbb{I}_{B,i}, V^*_{B,i}, I^*_{B,i})$ is the constraint for boundary condition $\mathbb{V}_{B,i}$ and $\mathbb{I}_{B,i}$ with respect to $V^*_{B,i}$ and $I^*_{B,i}$.

Compared to the two-level hierarchical formulation of (3.8) and (3.9), the two steps in the above are executed only once in sequence. In step 1, the optimal voltages and currents along the partitioning boundaries, e.g. V^*_B and I^*_B corresponding to the (unknown) optimal power grid design, are solved. Conceptually, this seemingly presents a chicken-and-egg dilemma: V^*_B and I^*_B may seem only to be obtained after finding the optimal power grid solution. However, by exploiting the strong locality behavior in C4-type power grids, it can be shown that V^*_B and I^*_B (or near optimal boundary values) can be rather efficiently obtained without solving the entire power grid optimization problem. The same locality property allows V^*_B and I^*_B be determined by solving a set of independent local optimization problems, leading to an immediate parallelization of step 1, as detailed in Section III.A.3.

In step 2, a set of N optimization problems are solved to optimize the N power grid partitions based on V^*_B and I^*_B computed in Step 1. In comparison with (3.9c), where the boundary condition for partition i is constrained by setting both the boundary voltages and currents, here in (3.11c) a relaxed boundary constraint is adopted. As detailed in Section III.A.4, the use of the relaxed boundary constraint makes the step-2 optimization problems significantly easier to solve, enhancing the convergence. Feeding step 2 with V^*_B and I^*_B (or near optimal boundary values) also significantly improves the convergence of these optimization problems. This is in contrast with the two-level hierarchical optimization, where arbitrary V_B and I_B values can make the level-2 optimization problems very difficult to solve or even infeasible.

In summary, the convergence, consequently also the runtime efficiency, of the proposed approach, are achieved by inputting V^*_B and I^*_B to the second step and adopting relaxed boundary constraints. The runtime efficiency is further enhanced by converting from the two-level hierarchical formulation to the sequential two-step formulation, where both steps are parallelizable. It shall be noted that in principle any robust nonlinear constrained optimization methods can be applied to solve the optimization problems in the two steps. Hence, the proposed approach is generic and formulated purely based upon the nature of the application.

3. Parallel Solution of Optimal Boundary Conditions

The locality property of the flip-chip type power delivery networks and its application in static analysis are presented in Section II.A. For notation simplicity, we refer to the locality exhibited under the context of the circuit analysis as *analysis locality.* In this work, the spatial locality is exploited for finding the optimal boundary conditions in power grid optimization. Similar to the idea shown in Figure 8, an optimization window is used to enclose a partition boundary at the window center. The size of each window is made large enough to include a ring of C4 bumps around the partition boundary such that the circuit responses along the boundary are influenced in a negligible way by the part of the power grid outside of the window.

In optimization, we further exploit *optimization locality* as described as follows. Each truncated window is treated as an independent grid and optimized using the standard optimization formulation (3.1)-(3.5). The optimized window is analyzed to compute the voltage and branch current responses on the partitioning boundary (located at the center of the window). *Optimization locality* exists if the voltage and current responses obtained in the above procedure are very close to those under the flat optimization for the entire power grid. Intuitively, based on *analysis locality*, it is well expected that *optimization locality* can also be achieved by choosing a proper window size. Again, in a large enough window, the nodes at the center will not be influenced much by the part of the grid outside the window. This spatial locality can propagate into the optimization stage. That is, towards to the center, the optimal wire widths obtained via the window-based optimization are increasingly closer to the true optimal values. Further considering *analysis locality*, the circuit responses of this optimized window shall match closely with those of the true optimal power grid solution at the center of the window.

We adopt window size and C4 ring size definition in Section II.A. A 160K-node power grid with C4 bumps is used as an example to illustrate the observed *optimization locality*. The C4 bumps are 25 nodes away from each other and are evenly distributed in the grid. The power grid is divided into four equally sized partitions. A boundary is examined as an example. The obtained node voltage and branch current for a single node on the boundary via the window-based optimization as functions of window size are shown in Table X. The average node voltage and the average branch current for the boundary are also examined. All the voltages and currents obtained from window optimizations are compared with the corresponding optimal values. Note the quick convergence of the results. The window size of 65, corresponding to the C4 ring size of 3, is already good enough for practical use.

In practice, the best choice of the window size is problem dependent and not known *a priori*. In this work, starting from a relatively small initial value, the window size is gradually increased till the convergence of the partition boundary responses is observed. The use of this procedure avoids spending unnecessary runtime due to an overly conservative choice of the window size. It is found that for most cases, the window size converges at a quite manageable value. Since each optimal boundary condition is determined by solving an independent window based optimization, this entire procedure can easily be parallelized, as summarized in Algorithm 3, where the original power grid $\mathbf{\Omega}$ is assumed to be divided into K partitions $\Omega_1, \cdots, \Omega_K$ with L partition boundaries $B_{1,opt}, \cdots, B_{L,opt}$. L boundary windows $W_1, \cdots, W_L$ are created

Table X. Optimized boundary voltage and current as functions of window size in the window-based optimization. WS: window size. C4S: C4 ring size. NV: voltage for a boundary node in V. NC: current for a boundary branch in mA. AV: average voltage for the boundary in V. AC: average current for the boundary in mA. OPT: optimal value.

WS	C4S	NV	NC	AV	AC
15	1	1.91007	0.281127	1.89857	0.109043
40	2	1.91208	0.292751	1.89802	0.368246
65	3	1.91218	0.293240	1.89798	0.341391
90	4	1.91218	0.293262	1.89798	0.338898
OPT	NA	1.91219	0.293271	1.89798	0.339042

with an initial window size S_{ini}. Each of these window-based optimization can be solved using one thread on a multi-core (or shared memory) machine in parallel.

4. Parallel Optimization of Partitioned Sub Power Grids

As presented in Section III.A.2.c, the optimal (or near optimal) boundary conditions V_B^* and I_B^* are computed in step 1 of the proposed approach. In step 2, each partitioned sub power grid is optimized in parallel using V_B^* and I_B^*. In practice, setting the boundary constraints exactly to V_B^* and I_B^* can still make the step-2 optimization problems very difficult to solve numerically. As such, relaxed boundary constraints as in (3.11c) are adopted. On the other hand, the relaxation of the boundary conditions must be handled with care since inconsistent boundary conditions may alter the circuit responses in each partition, leading to an infeasible solution after all the optimized partitions are merged to form the complete grid. We show a relaxed boundary constraint scheme, which is provable to guarantee the IR drop constraints for the merged grid. In practice, the scheme may produce a small amount of EM violations,

Algorithm 3 Parallel optimization for the optimal boundary conditions

Input: Original power grid $\mathbf{\Omega}$, partition boundaries $B_1, \cdots, B_L$, initial window size S_{ini}, the convergence tolerance C_{tol}.

Output: Optimized boundary conditions $B_{1,opt}, \cdots, B_{L,opt}$, maximum terminating window size $S_{max,t}$.

```
for $i \leftarrow 1$ to $L$ par do
    $B_{i,opt} \leftarrow B_i$
    $S_i \leftarrow S_{ini}$
    while NOT CONVERGED do
        Form the boundary optimization window $W_i$ with $B_{i,opt}$ and $S_i$.
        Create a sub optimization problems $O_i$ using $W_i$ subject to (3.2)-(3.5).
        Solve $O_i$. The optimized boundary $B_{i,temp}$ is extracted.
        if $\frac{B_{i,temp} - B_{i,opt}}{B_{i,temp}} < C_{tol}$ then
            CONVERGED
        else
            NOT CONVERGED
        end if
        $B_{i,opt} \leftarrow B_{i,temp}$
        if NOT CONVERGED then
            Increase window size $S_i$.
        end if
    end while
end for
$S_{max,t} \leftarrow \max_{i=1,\cdots,L}\{S_i\}$.
return $B_{1,opt}, \cdots, B_{L,opt}$, $S_{max,t}$
```

which can be reduced via fixing techniques. It should be noted that there is a tradeoff between the number of sub-grids to be optimized and the size of each sub-grid. Having smaller sub-grid size makes the optimization of each one faster; however, there might be a large number of them to be processed.

a. Optimization of Partitioned Grids Using Relaxed Boundary Constraints

The basic idea is to maintain the exact boundary voltage conditions while constraining (relaxing) boundary currents in a way such that each partition is optimized under a *potentially* worse current loading condition. As will be shown later, this ensures that the merged power grid sees a possible reduction in current loading if it ever changes, which increases or at least maintains the same voltage level for every node, thereby keeping the IR drop constrains satisfied after the merging. Although there is no theoretical guarantee for zero EM constraint violation, if optimal or near optimal boundary conditions are used, the non-degraded overall current loading in the merged grid tends not to alter the current distributions to jeopardize the EM constraints in a significant way, which is consistent with empirical observations.

Consider the illustrative example shown in Figure 40, where a power grid is partitioned into two pieces P and P' at node interface $B = \{B1, B2, B3\}$. The optimization procedure is illustrated in three steps:

- Step I: Before partitioning, the optimal (or near optimal) boundary voltages $V_B^* = \{V_{B1}^*, V_{B2}^*, V_{B3}^*\}$ as well as the optimal (or near optimal) boundary currents $I_B^* = \{I_{B1}^*, I_{B2}^*, I_{B3}^*\}$ are computed in the first step of the proposed two-step optimization approach. It is assumed the directions of I_B^* are as shown in the figure.

- Step II: The grid is partitioned into two parts: P and P'. The left part still

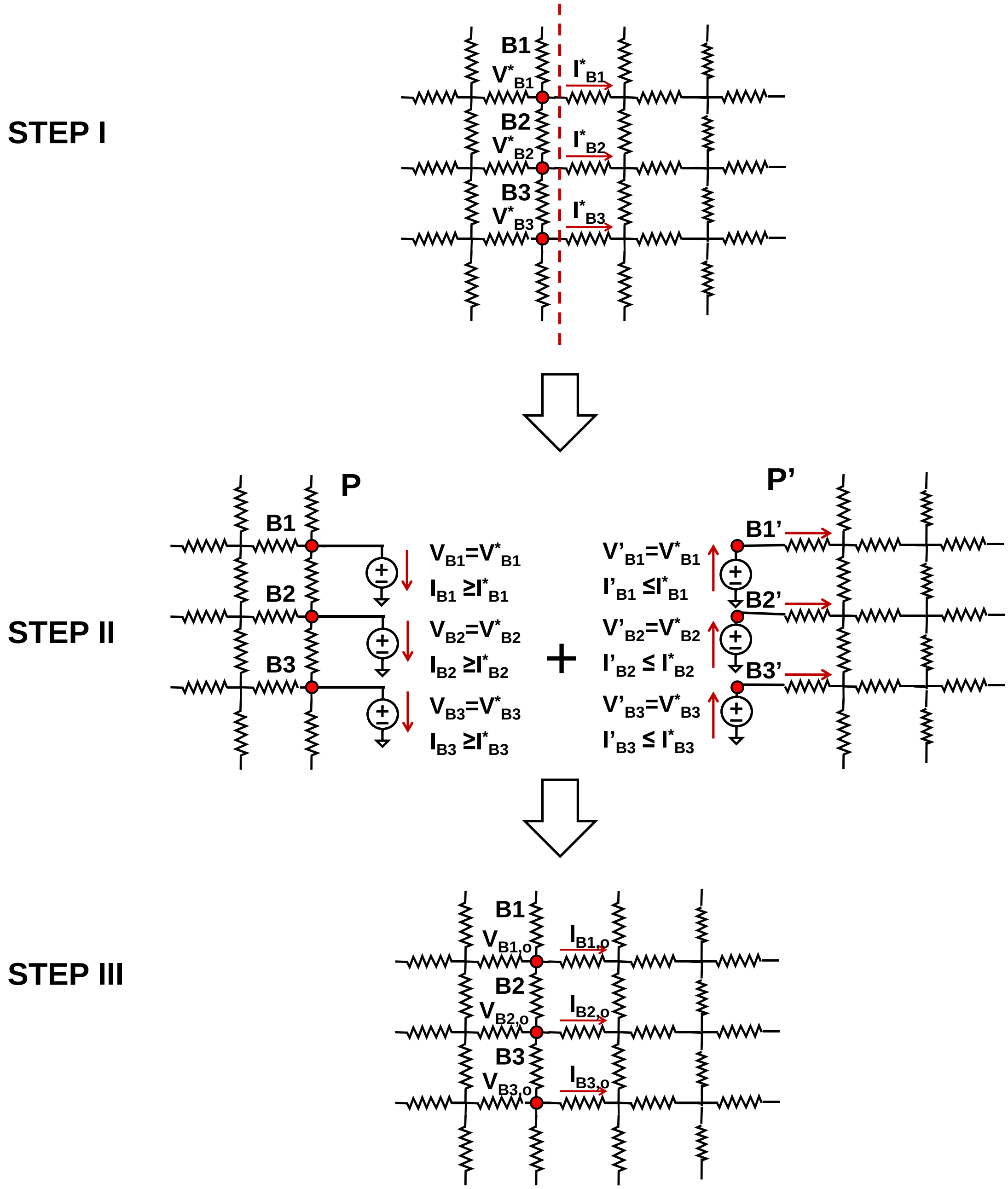

Fig. 40. Optimization of the partitioned grid under relaxed boundary conditions.

possesses the original nodes B, and new nodes $B' = \{B1', B2', B3'\}$ are created for the right partition. Ideal voltage sources $V_B = \{V_{B1}, V_{B2}, V_{B3}\}$ and $V'_B = \{V'_{B1}, V'_{B2}, V'_{B3}\}$ with values V^*_B are attached to the split nodes B and B' in each partition. Equality and inequality constraints are set for the voltages and branch currents of the voltage sources, respectively

$$\begin{cases} V_{Bi} = V^*_{Bi},\ V'_{Bi} = V^*_{Bi} \\ I_{Bi} \geq I^*_{Bi},\ I'_{Bi} \leq I^*_{Bi}. \end{cases} \quad i = 1, 2, 3. \tag{3.12}$$

Note that the reference directions of the two branch currents are consistent to that of the boundary branch current in step I. The current constraints are such that the currents drawn out from P by V_B are at least I^*_B while the currents provided by V'_B into P' are not greater than I^*_B. This leads to the potentially worse loading conditions on which the two partitions are independently optimized.

- Step III: After the optimization, P and P' are merged to form the complete grid, in which the ideal voltages used to set the boundary conditions are removed.

b. Maintenance of IR Drop Constraints

We prove the maintenance of IR drop constraints in the proposed optimization with relaxed boundary conditions. First, several theoretical results relevant to power grid analysis are presented.

Definition 1. *A nonnegative matrix is a matrix with all the elements being nonnegative.*

To perform a DC analysis for a power grid, the standard modified nodal analysis (MNA) can be used to generate a system matrix. If the voltages of the $\mathrm{V_{DD}}$ pads are substituted by the known supply level in the system of equations, the system matrix becomes a so-called $\mathcal{M}$-matrix [50], as shown in [10]. The following results exist for $\mathcal{M}$-matrices.

Lemma 1. *The inverse of a $\mathcal{M}$-matrix is a nonnegative matrix [50].*

The monotonicity of power grids is suggested in [23].

Lemma 2. ***(monotonicity)*** *None of the node voltages of a V_{DD} distribution network decreases if the current loading to the network decreases.*

Proof. The variations of node voltages due to any change in the current loading are determined by the following linear matrix problem

$$A\Delta V = \Delta I_{in}, \tag{3.13}$$

where ΔV is the vector of node voltage changes; ΔI_{in} correspond to that of the current loading; A is the system matrix, which is a $\mathcal{M}$-matrix. If the current loading decreases, all the entries in ΔI_{in} are positive. Further considering that A^{-1} is nonnegative ($A^{-1} \geq 0$) leads to

$$\Delta V = A^{-1}\Delta I_{in} \geq 0. \tag{3.14}$$

□

Now, we show the following theorem.

Theorem 1. *Merging individually optimized grid partitions under the relaxed boundary conditions (e.g. (3.12)) maintains the IR drop constraints.*

Proof. Since the IR drop constraints are enforced when each partition is individually optimized, the Theorem 1 amounts to show that after the merging none of the node voltages in the entire grid decreases. Without loss of generality, the example in Figure 40 is used to show the result, and for simplicity, only node $B1$ is analyzed. In Figure

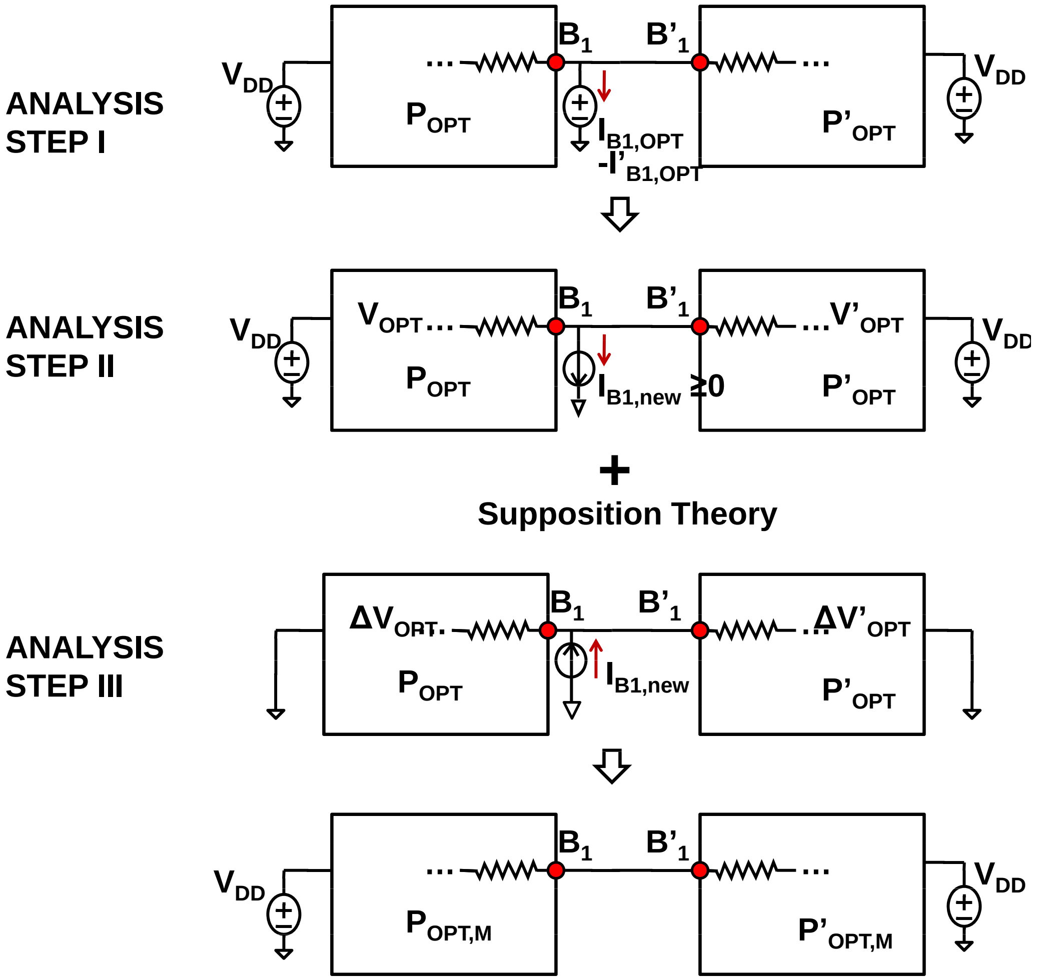

Fig. 41. Merging process analysis

41, a sequence of analysis steps are followed to analyze the voltage response change after the merging.

- Analysis step I: Assume that after the optimization, the actual voltage source branch currents are $I_{B1,OPT}$ and $I'_{B1,OPT}$ for partitions P and P', respectively. Now the two partitions are merged at the interface node $B1$ with a single ideal voltage source of V_{B1} attached. Since constraints (3.12) are enforced in the optimization, it is easy to see that $I_{B1,OPT} - I'_{B1,OPT} \geq 0$, which is equal to the branch current through the single ideal voltage source. Note that the reference direction of the branch current is going into the voltage source as shown in the figure. Furthermore, all the voltage and current responses in the grid remain unaltered in this step.

- Analysis step II: The ideal voltage source at the interface is replaced by an ideal current source with a value $I_{B1,NEW} = I_{B1,OPT} - I'_{B1,OPT} \geq 0$. According to the substitution theory, no change is made to any circuit response.
- Analysis step III: The voltage responses of the merged grid are now analyzed. Note that the merged grid is the result of removing the ideal current source in Step II from the network, which is equivalent to keeping a zero-valued current source. By the principle of superposition, it implies that the circuit responses of the merged network can be computed by summing up the responses of two networks. The first network is identical to what is in the previous analysis step. The second network is constructed by modifying the network in Analysis Step II: reversing the current direction of the ideal current source and zeroing all other independent sources, that is, grounding all the V_{DD} connections. Since the ideal voltage source provides current into the grid, by Lemma 2, the voltage responses of the second network are all positive.

Compared to analysis step II, the net changes of voltage responses in the merged grid are equal to the responses of the second network. Hence, the theorem is approved.

□

c. Fixing EM Violation

Due to the adopted relaxed boundary current constraints, the current boundary conditions between partitions may not be completely identical. Theoretically, the merging step can alter the current distributions of each partition, contributing to a small amount of EM violations. It has been observed in extensive experiments that if the boundary voltages and boundary currents computed in the first step of the two-step optimization method are close to the optimal solutions, no significant EM violation (less than 0.1%) will be generated after merging. Therefore, if the merged grid has large EM violations, the boundary conditions can be recomputed by tightening the convergence tolerance in Algorithm 3. An alternative fixing strategy is to guard band the small amount of merging induced EM violations by tightening up the EM constraints slightly when optimizing each partition.

d. Algorithm Flow for the Proposed Locality Driven Parallel Optimization

Finally, we summarize the overall flow of the proposed locality driven parallel optimization method in Algorithm 4. Assume we have obtained the optimal boundary conditions $B_{1,opt}, \cdots, B_{L,opt}$ from the boundary window optimization using Algorithm 3. If the maximum EM violation Vio_{max} found in the obtained optimal global grid exceeds the EM violation tolerance Vio_{tol} (e.g. 1%), then the convergence tolerance C_{tol} decreases by half. In practice, once the C_{tol} is well chosen, the maximum EM violation is always negligible. All other notations are the same as in Algorithm 3.

5. Experimental Results

The proposed parallel locality-driven optimization method has been implemented in C and integrated with IPOPT (Interior Point OPTimizer) [47]. Parallelization is implemented by creating multi-threads using Pthreads. Experimental results for the flat optimization (optimizing the power grid as a whole without partitioning), the serial locality-driven optimization (running the boundary window optimization and individual partition optimization in serial), and the parallel locality-driven optimiza-

Algorithm 4 Locality-driven parallel optimization algorithm

```
Input: Original power grid $\mathbf{\Omega}$, $K$ partitions $\Omega_1, \cdots, \Omega_K$, convergence tolerance $C_{tol}$, EM violation tolerance $Vio_{tol}$.
Output: Optimized power grid $\Omega_{opt}$, maximum EM violation $Vio_{max}$.

NOT OPTIMUM
while NOT OPTIMUM do
  Get $L$ optimal boundaries $B_{1,opt}, \cdots, B_{L,opt}$ from parallel boundary window optimization in Algorithm 3.
  Create $K$ partition optimization problems $O_{p1}, \cdots, O_{pK}$ using $B_{1,opt}, \cdots, B_{L,opt}$ subject to (3.2)-(3.5) and (3.12).
  Solve $O_{p1}, \cdots, O_{pK}$ simultaneously. The optimized partitions $\Omega_{1,opt}, \cdots, \Omega_{K,opt}$ are merged to form the optimal global grid $\Omega_{opt}$.
  Simulate the grid $\Omega_{opt}$, and find the maximum EM violation $Vio_{max}$.
  if $Vio_{max} > Vio_{tol}$ then
    $C_{tol} \leftarrow C_{tol}/2$
    NOT OPTIMUM
  else
    return $\Omega_{opt}$ and $Vio_{max}$
    OPTIMUM
  end if
end while
```

tion are obtained on a workstation with two quad-core Intel Xeon CPUs at 2.33GHz and 8G RAM running 64-bit Linux OS.

a. Partition Optimization

As stated in previous sections, the feasibility and convergence of each partition optimization greatly depends on the choice of boundary conditions V_B^* and I_B^*, thus the size of the window chosen for the boundary window optimization. Use the 160K-node power grid mentioned in Section III.A.3 as an example. The power supply is 2V. The IR drop constraints and EM constraints are set as that for each node voltage V_i: $V_i \geq$ 1.8V; and for each branch current I_b and width w_b: $I_b/w_b \leq$ 1mA/um.

Figure 42 shows how the quality of the final optimized power grid, in terms of IR drop violations and EM violations, depends on the optimality of the boundary conditions we employ for the partition optimizations. The original power grid exhibits IR drop violations, in Figure 42(a), and EM violations, in Figure 42(b), (only the EM distribution for horizontal grid wires is shown in the figure. The distribution for vertical wires has a similar pattern) in some regions. As shown in Table X, if the boundary window size is chosen to be 40 (C4 ring size of 2), the resulting boundary voltages and currents are not completely accurate. This creates convergence problems in the subsequent individual partition optimizations, leading to long runtime and low optimization quality. According to Theorem 1, in this case, there is no IR drop violation in the final optimized grid, as shown in Figure 42(c). However, EM violations do exist, as shown in Figure 42 (d) (the maximum I_b/w_b is over 1.2mA/um). After including more C4 bumps into the boundary window optimization and using 65 as the window size (C4 ring size of 3), the boundary conditions obtained for all partitioned sub grids are much closer to the optimal values. With those near optimal boundary conditions, all the partition optimizations are able to converge. Finally, as before, there exists no IR drop violations (the largest IR drop is less than 140mv), and also all the current densities are within the specified EM constraints as shown in Figure 42(e) and Figure 42(f). The results presented in the figure also confirms the previous claim that if the boundary conditions are near to the optimums sufficiently, no significant EM violations would be generated after merging partitions together (less than 0.03% in this case).

b. Overall Optimization

The proposed locality-driven parallel power grid optimization algorithm has been tested under six large-scale power grids with varying sizes: 40K-node, 90K-node, 160K-node, 360K-node, 640K-node, and 1M-node. All the power grids use C4 bump power supply pads. Among the six initial designs, the starting wire widths are chosen

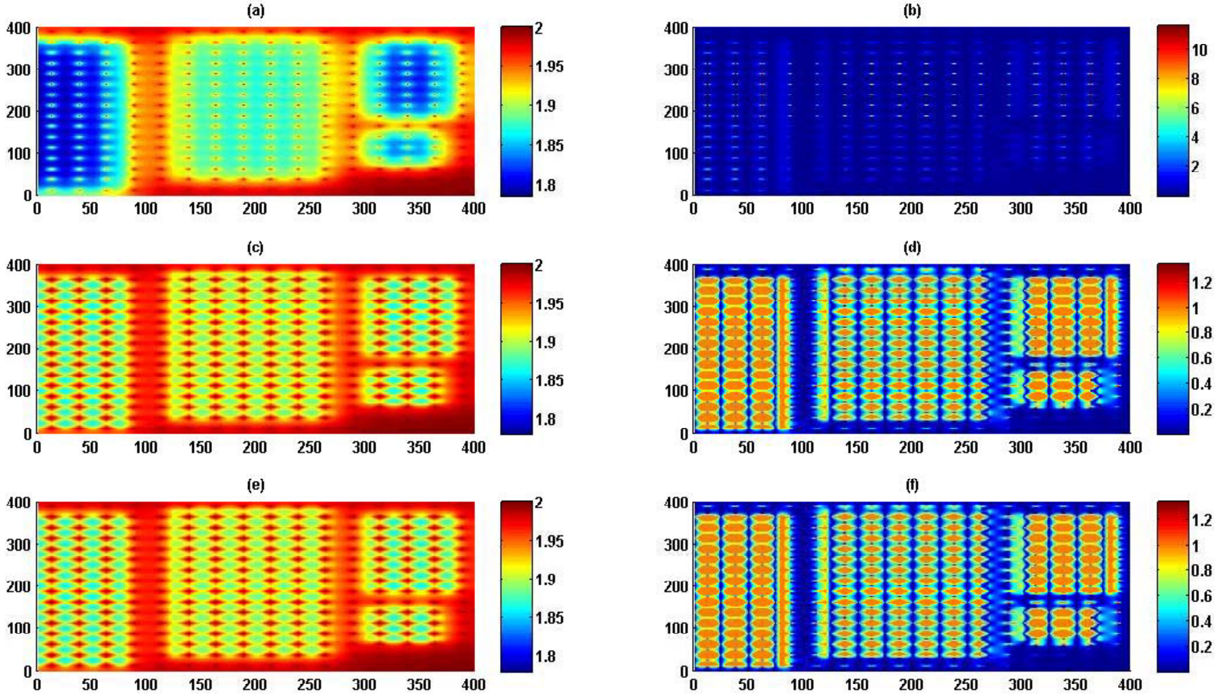

Fig. 42. Impact of boundary conditions on the quality of final optimized power grid. (a): node voltage distribution before optimization. (b): EM distribution of horizontal wires before optimization. (c): node voltage distribution after optimization using window size 40. (d): EM distribution of horizontal wires after optimization using window size 40. (e): node voltage distribution after optimization using window size 65. (f): EM distribution of horizontal wires after optimization using window size 65.

Table XI. Optimization runtime for flat optimization, serial locality-driven optimization, and parallel locality-driven optimization. N: number of nodes. P: number of partitions. SIM: flat simulation runtime in sec. OPT: optimization runtime in sec. BWO: boundary window optimization runtime in sec. PO: partition optimization runtime in sec. TOT: total runtime in sec. AR: area reduction in %. $\mathrm{WS_b}$: beginning window size. $\mathrm{WS_t}$: maximum terminating window size. IT: number of iterations for window size determination. NVio: number of EM violation. $\mathrm{Vio_{max}}$: maximum EM violation in mA/um.

N	P	SIM	Flat Optimization			Serial Optimization				Parallel Optimization								
			OPT	TOT	AR	BWO	PO	TOT	AR	BWO	PO	TOT	AR	$\mathrm{WS_b}$	$\mathrm{WS_t}$	IT	NVio	$\mathrm{Vio_{max}}$
40K	4	1	110	111	52.87	381	63	446	52.86	134	24	160	52.86	40	65	2	6156	$4e^{-4}$
90K	9	2	890	892	53.96	1055	195	1254	53.95	257	39	300	53.95	40	65	2	8306	$2e^{-4}$
160K	4	3	2122	2126	61.08	539	670	1218	61.06	366	223	598	61.06	40	90	3	9946	$3e^{-4}$
360K	9	10	8057	8068	75.50	1707	1267	3001	75.49	333	336	698	75.49	40	65	2	10478	$3e^{-4}$
640K	16	18	20243	20264	75.58	3500	2588	6143	75.57	686	652	1398	75.57	40	65	2	17503	$3e^{-4}$
1M	25	43	NA	NA	NA	6827	3283	10234	59.97	1264	802	2185	59.97	40	65	2	1094	$2e^{-4}$

such that some of them do not have any constraint violation (e.g. over designed), while others do. A fast multigrid-like power grid simulator is employed for the grid simulation. Since the simulation only provides the initial values for the optimization, accurate power grid simulation through expensive analysis (e.g. direct solve) is not needed. The flat simulation runtime, optimization runtime and the area reduction for all the cases with three optimization methods: flat, serial and parallel, are shown in Table XI. The simulation runtime is negligible compared to the optimization runtime. Since the serial and the parallel methods follow the same algorithm, only the beginning window size, the maximum terminating window size, and the number of iterations for window size determination scheme of the parallel method are shown in Table XI. Moreover, only the number of EM violations and the maximum EM violation for the parallel method are presented. It should be noted that the percentage of the reduced wiring area relative to the original wiring area is used to reflect the optimal area we have obtained by using one of the three optimization methods.

As can be seen from the table, the significant advantage of the proposed approach is its scalability. Due to excessive memory and runtime requirement, the standard flat optimization does not scale well with the circuit complexity. In our case, it takes the flat optimization 20,264 seconds to size the 640K node grid. The 1-million node grid cannot be optimized in flat due memory overflow. Given the large size of practical power grids, this presents a severe limitation. In contrast, the divide-and-conquer nature of the proposed method makes it highly scalable. The serial version of our locality driven approach can successfully size all the benchmarks. Even for the grids that can be optimized in flat, our serial method can achieve good runtime speedups under some cases. In practice, the amount of speedup may depend on the grid size, structure, current loading and initial design. Furthermore, our method is naturally parallelizable. This allows the use of parallel processing to gain further runtime efficiency. For example, the parallel locality-driven method can size the largest one-million node grid by using only 2,185 seconds on the machine with two quad-cores. We expect much larger grids can be also successively optimized by our approach. Moreover, our methods generate almost the same optimal results as the flat optimization. Although EM violations exist in the final merged power grid, their values are insignificant (in the order of e^{-4} compared to the constraint bound of the order of 1), thus can be neglected.

For our method, the parallel version can bring in a 2X speedup for grids with four partitions, and 4X speedup for ones that have more than eight partitions, respectively. The achieved parallel speedups are shown in Figure 43. As can be seen, the achieved parallel processing factors are less than one, which is primarily due to load imbalance caused by the partitioning we use, as shown in Figure 44. For example, for some boundary optimizations, due to the strong couplings to the rest of the grid, they may need large window size to get converged boundary conditions (may need to increase the window size three times). Whereas, for the boundary windows with weak coupling to other regions of the grid, the near optimal boundary conditions could easily be reached. Moreover, the optimizations for some partitions could easily converge while others may take longer time to reach the optimum. Therefore, the work of some threads (in terms of number of optimization iterations) may be significantly larger than the work of others. Those heavily loaded threads have a great impact on the parallel runtime. In our future work, loading balancing techniques will be developed to further improve the efficiency of the proposed parallel optimization method.

6. Summary

In this section, we proposed a novel partitioning-based locality-driven two-step optimization scheme for the power grid wire size optimization problem. This scheme

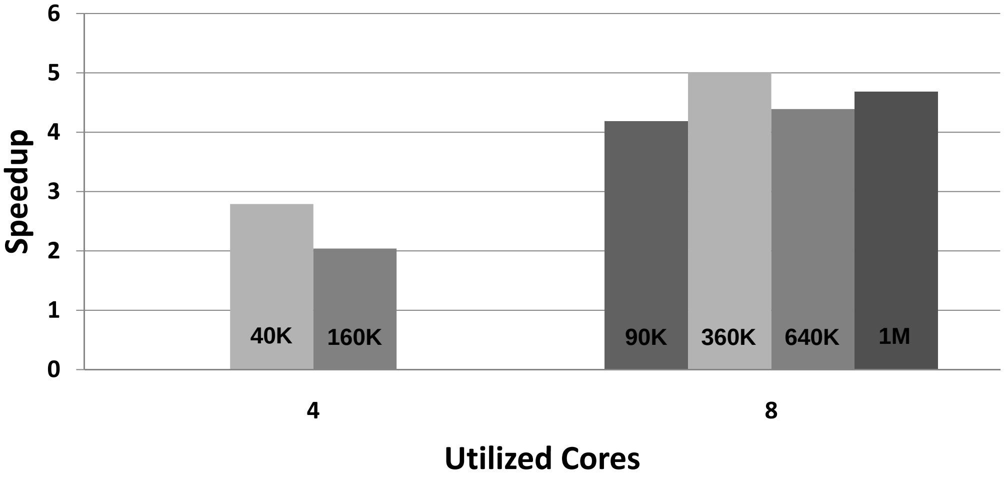

Fig. 43. Speedup of parallel over serial locality-driven optimization.

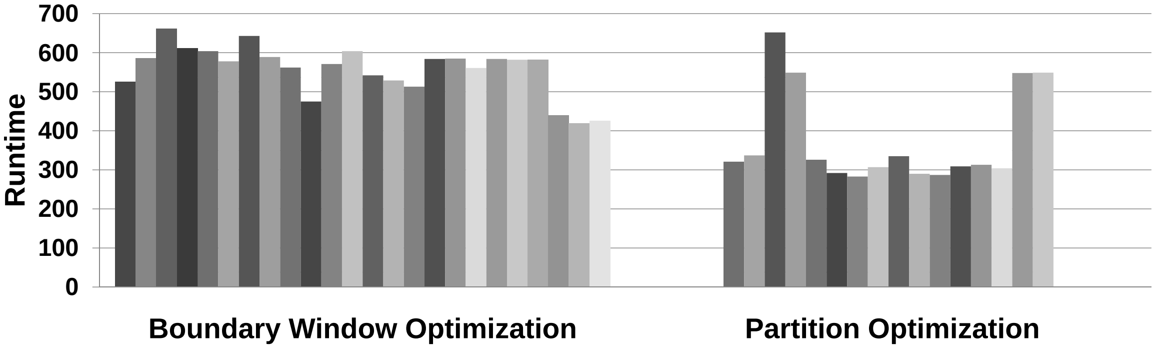

Fig. 44. Threads runtime (in sec) of boundary window optimizations and partition optimizations for 640K-node power grid.

exploits the locality of the power grids for scalable optimization of large grids. In the first step of the proposed approach, optimal (or near optimal) partitioning boundary voltages and currents are obtained via localized window-based optimization. In the following step, partitions are individually optimized under the obtained boundary conditions. The *divide-and-conquer* nature of the proposed method not only leads to its favorable scalability but also makes it possible to employ increasing parallel computing resources to facilitate the optimization of large power grids.

B. System-Level Co-Design of Power Delivery Networks with On-Chip Voltage Regulation

Integrating a large number of on-chip voltage regulators has the appeal of facilitating fine-grain multiple voltage domains on chip. However, how to choose the optimal parameters for the voltage regulators/converters designs and the passive network design so that the overall optimal performance can be achieved becomes a critical problem. In this section, using the fast GSim engine, detailed systematic analysis for the entire network with on-chip regulators is carried out. With the obtained design insights, a system-level co-design flow is proposed to automatically optimize the entire network.

1. Background

a. System Modeling

The system-level components as well as their detailed models for a power delivery network with on-chip voltage regulation is presented in Figure 45. Low-dropout regulators are integrated on the chip to provide local voltage supplies and regulations to different power domains (modeled as local grids). The input voltage to the LDO is

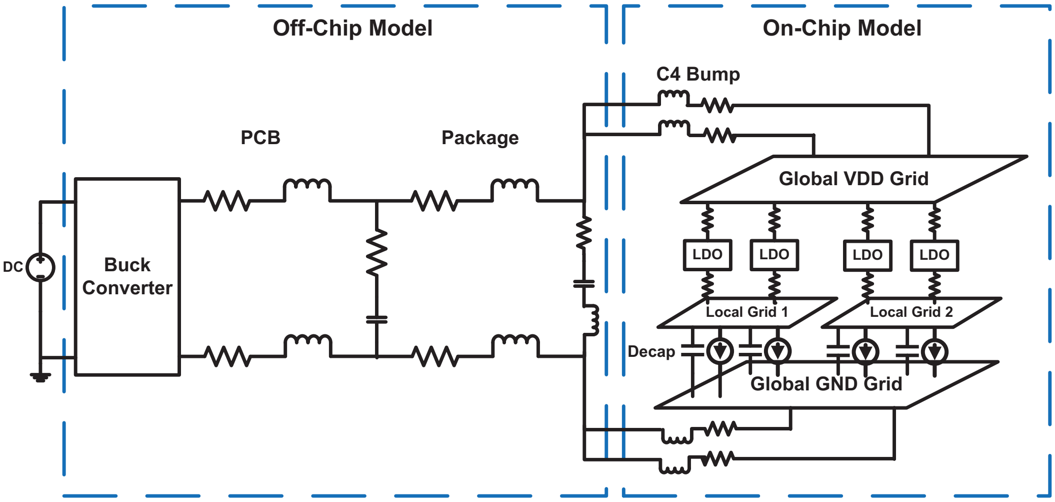

Fig. 45. System structure and model of a power delivery network with on-chip voltage regulation.

provided by an efficient on-board Buck Converter (BC) which converts the external power supply to the level that is close to the preset LDO output voltage so as to improve the overall power efficiency.

b. Benefits of On-Chip Voltage Regulation

High-frequency local voltage drops due to the fast switching circuits, lower-frequency global resonance caused by off-chip inductive parasitics and IR drop due to the resistance between the voltage supply and the on-chip nodes are three major contributors to the voltage fluctuation [6]. Suppressing or remedying these effects would significantly improve the performance of PDN.

With the accurate and powerful analysis engine GSim illustrated in Section II.B, we give a quantitative analysis to the effects of having on-chip voltage regulation. A random node's voltage drops before and after integrating on-chip LDOs are examined, and the voltage drop waveforms are shown in Figure 46. For the PDN that does not have LDOs, local grids are connected to the VDD grid through vias. Otherwise the

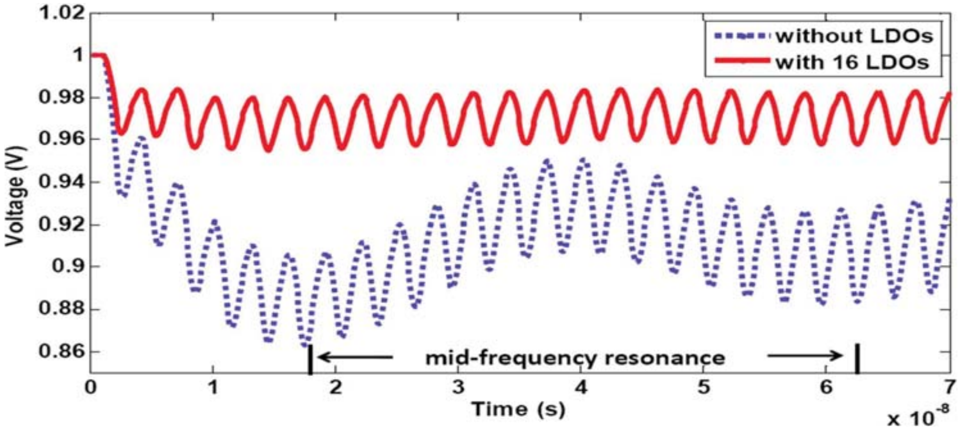

Fig. 46. Voltage drops for a power domain with LDOs and without LDOs.

local grids are connected to LDOs as shown in Figure 45.

Figure 46 shows that without the regulation of on-chip LDOs, on top of the high-frequency voltage drop, there is a large mid-frequency swing due to chip-package resonance. Even though the voltage drop caused by switching currents is about 40mv, the resonance introduces another 20mV drop, and makes the total voltage drop much larger. However, in contrast, by having on-chip voltage regulators, the benefits are threefold:

1. *Suppressing high-frequency local drop*: On-chip LDOs provide strong local regulation. Along with on-chip decoupling capacitors, the LDOs can respond quickly to local current fluctuations and automatically maintain the output voltage level. Hence large local voltage swings are suppressed significantly.

2. *Remedying mid-frequency global resonance*: On-chip LDOs do not suffer the global resonance since the resonance is blocked at the input of LDOs. LDOs have weak transfer functions so that as long as working in the regulation region they are not sensitive to the input changes. Therefore, large voltage fluctuations at the off-chip circuits and VDD grids can not propagate to local grids.

3. *Reducing IR drops*: On-chip LDOs shorten the distance between current loadings and voltage regulators, and thus reduce the IR drops due to the wire resistance. It should be noted that the DC shift of the waveform with voltage regulation is caused by line and load regulations of LDOs.

c. Low-Dropout Regulator Design

In this work we adopt a novel multi-loop topology for enhanced on-chip regulation performance. The analysis of this LDO is somewhat complex. Without loss of generality, a simplified topology (as shown in Figure 47) is employed to demonstrate major LDO design considerations, such as *dropout voltage*, *maximum load current*, *power efficiency*, *LDO output impedance*, *power supply rejection (PSR)* and *stability*. The same analysis approach can be applied to other LDOs.

In this topology, M1 is the pass transistor to deliver load current. M2 and M3 work as a current sensor to detect load current variation and generate signal (V_{ctrl2}) to drive M1. The Error Amplifier (EA) works as the output voltage (V_{out}) sensor that compares V_{out} with the reference voltage (V_{ref}) and generates V_{ctrl1} to adjust V_{out} through M2.

The *dropout voltage* is the input-to-output differential voltage at which the circuit ceases to regulate against further reductions in input voltage. If input-to-output voltage difference is less than the *dropout voltage*, the regulator is in the *dropout region* and the output voltage decreases in proportion to the decreasing input voltage. In contrast, if input-to-output voltage difference is larger than the *dropout voltage*, the regulator is in the *regulation region* and the output voltage maintains a stable level. The *maximum load current* as well as *dropout voltage* is determined by the dimension ratio (w/l) and the maximum allowable gate-to-source voltage (V_{gs}) of M1 in Figure 47.

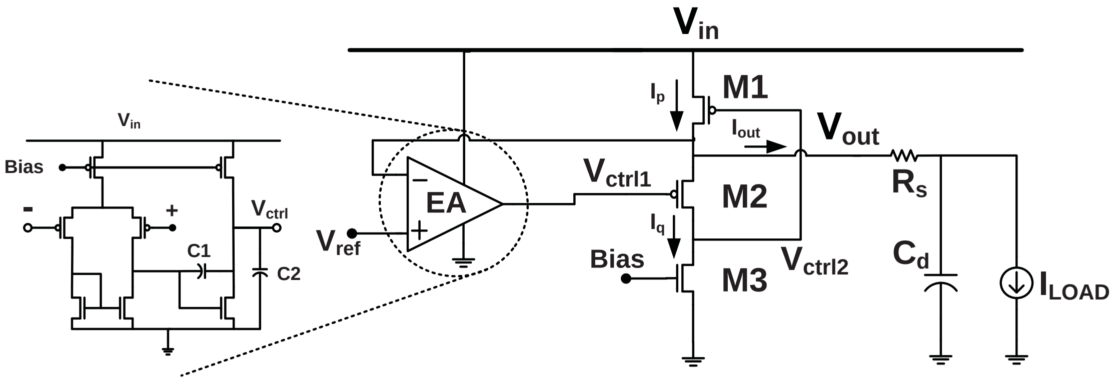

Fig. 47. LDO topology.

The *power efficiency* of an LDO is limited by the quiescent current and output-to-input voltage ratio, and is defined as,

$$\varepsilon_{LDO} = \frac{I_{out}V_{out}}{I_{in}V_{in}} = \frac{I_{out}V_{out}}{(I_{out}+I_q)V_{in}}, \tag{3.15}$$

where I_q is the quiescent current flowing to the ground. Obviously increasing input-to-output voltage difference would reduce power efficiency.

The LDO output noise caused by the load current variations is dependent on the small-signal *output impedance* (Z_{out}) which has the expression of

$$Z_{out} \approx \frac{(sC_dR_s+1)(sC_{gs1}+g_{ds2})}{(1+H_{ol})\cdot D}, \tag{3.16}$$

$$H_{ol} \approx \frac{g_{ma1}g_{m2}(g_{m1}+sC_{gs1})(g_{ma2}-sC_1)(1+sC_dR_s)}{[g_{oa1}g_{oa2}+(C_1g_{ma2}+C_2g_{oa1}s+C_1C_2s^2)]\cdot D}, \tag{3.17}$$

$$\begin{aligned} D =& C_dC_{gs1}s^2 + \big[C_dR_s(g_{m1}g_{m2}+g_{ds1}g_{ds2}+g_{ds2}g_{m1}) \\ &+ C_dg_{ds2}+Cgs1g_{m2}\big]s + g_{m1}g_{m2}, \end{aligned} \tag{3.18}$$

where H_{ol} is the loop gain; g_{mx}, g_{dsx}, and C_{gsx} represent transconductance, drain-source conductance and gate-source capacitance of the device M_x ($x = 1,2,3$); g_{ma1} and g_{ma2} are the equivalent transconductance of the first and second stage of the EA respectively; g_{oa1} and g_{oa2} are the equivalent DC output impedance of the two stages of EA respectively; C_1 is the frequency compensation capacitor and C_2 is capacitive load of EA; C_d and R_s are the load capacitance and parasitic resistance of power grids respectively.

The *power supply rejection* of the LDO measures how well the output is isolated from the supply noise. It can be expressed as

$$PSR \approx \frac{\left[sC_{gs1}\left(g_{ds1}+g_{ds2}\right)+\left(g_{m1}g_{ds1}\right)\left(g_{ds2}+g_{ds3}\right)\right]\left(1+sC_dR_s\right)}{\left(1+H_{ol}\right)\cdot D}. \tag{3.19}$$

For good noise performance, both Z_{out} and PSR are desired to be small.

As can be observed from (3.17) and (3.18), the system has four poles. Two of them are contributed by the EA, namely $p_1 \approx g_{oa1}/(A_2C_1)$ and $p_2 \approx g_{ma2}/C_2$. The other two are associated with the output stage, namely p_3 and p_4. The system also has three zeros, two of which are contributed by the output stage and the other by the Miller capacitor C_1. The relative positions of the poles and zeros determine the *stability* of the regulator.

d. Decoupling Capacitor Sizing and Placement

According to the analysis presented in [38], high-frequency noise introduced by fast transient loads can be effectively reduced by increasing decoupling capacitance or by placing capacitance closer to the load. However, increasing decoupling capacitance comes with the cost of occupying more precious chip area and aggravate the decap power leakage. Moreover, more efforts on circuit floorplanning, placement and routing may be required.

e. Buck Converter Design

A typical Pulse-Width Modulation (PWM) buck converter [51] is shown in Figure 48. The operation of this converter behaves in the following manner. The switches in

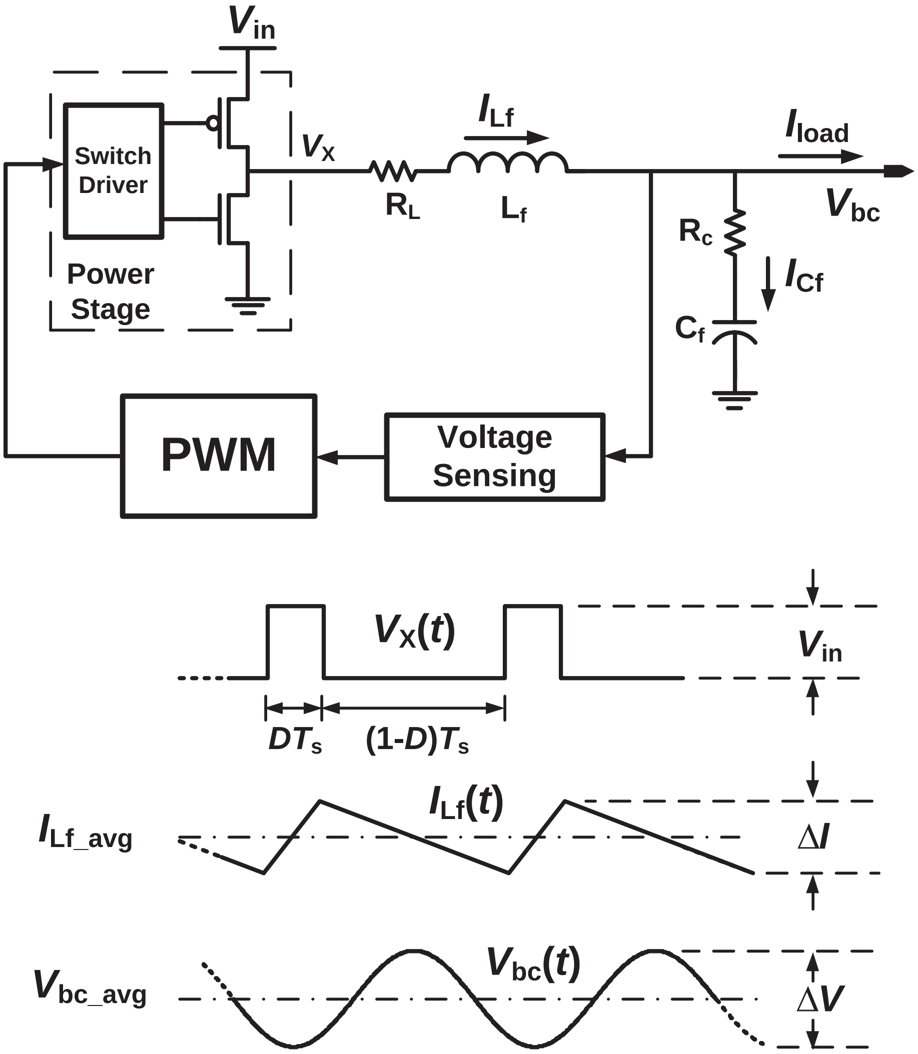

Fig. 48. Buck converter topology.

the power stage are controlled by the PWM block to be turned on and off, generating a square waveform of V_x. Then the DC component of V_x is passed to the output through a second-order low-pass LC filter (L_f and C_f).

In principle, the design of a transistor-level buck converter can be very complex. Without loss of generality, in this work, a behavioral model of the buck converter is used to capture the key design aspects that influence the performance power delivery systems [51]. In this model, the average output voltage (V_{bc_avg}), the voltage ripple (ΔV), the power loss (P_b) and power efficiency (ε_b) for the buck converter, are expressed as following.

$$V_{bc\_avg} = DV_{in} - R_L I_{load}, \tag{3.20}$$

$$\Delta V = \frac{DV_{in}(1-D)}{8L_f C_f f_s^2}, \tag{3.21}$$

$$\begin{aligned} P_b =& \frac{1}{2} C_{MOS} V_{in}^2 f_s + \left[D I_o^2 R_{PMOS} + (1-D) I_o^2 R_{NMOS} \right] \\ &+ \frac{1}{2} R_L I_o^2 + \frac{1}{3} \left(\frac{\Delta I}{2} \right)^2 R_C, \end{aligned} \tag{3.22}$$

$$\varepsilon_b = \frac{P_{load}}{P_{load} + P_b}, \tag{3.23}$$

where C_{MOS} is the total capacitance to be charged/discharged for turning on and off those switches in the power stage; R_{PMOS} and R_{NMOS} are the equivalent resistance for the PMOS and NMOS switches in the power stage; R_L and R_C are the equivalent serial resistance for L_f and C_f, respectively; the peak-to-peak variation of inductor current

$$\Delta I = \frac{V_{in} D(1-D)}{L_f f_s}, \tag{3.24}$$

$$I_o = \sqrt{I_{load}^2 + \frac{1}{3} \left(\frac{\Delta I}{2} \right)^2}, \tag{3.25}$$

$$P_{load} = I_{load} V_{bc\_avg}. \tag{3.26}$$

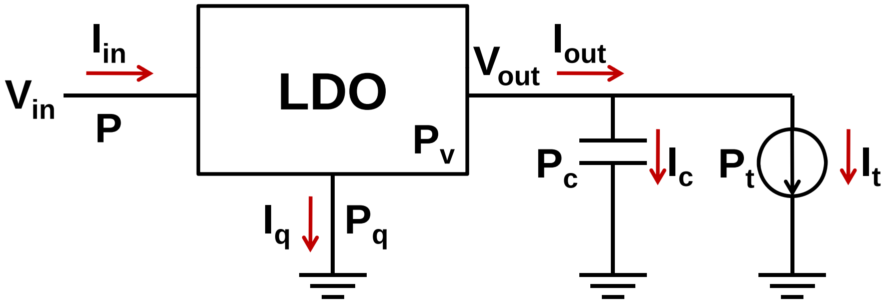

Fig. 49. Power consumption of the PDN with on-chip LDOs.

The zero-load power loss P_{b0} can be obtained from (3.22) by having $I_{load} = 0$.

2. System-Level Co-Design

Since the entire power delivery system is very complex and there are many aspects to be considered for the system-level co-design, for the illustration purpose, we first discuss the LDO-decap system co-design in which the buck converter is assumed to be an ideal power supply (100% power efficiency, zero power loss and a fixed output voltage). Then, we present how to introduce the buck converter design into the system co-design framework.

a. LDO-Decap System Co-Design

To simultaneously design low-dropout regulators and decoupling capacitors for a large power delivery network, specific design considerations as well as the electric interactions between LDOs and decaps in each major design aspect must be well understood. Note that the decaps are the capacitive loads to the LDOs in the network.

Power: The power consumed by the entire chip (P) consists of the power consumed by the transistors (P_t) (noted as useful power), the leakage power of the decoupling capacitors (P_c), the LDO voltage conversion power loss (P_v), and LDO quiescent power (P_q). As shown in Figure 49, these power terms can be expressed as

$$P = V_{in} \cdot I_{in} = V_{in} \cdot (I_q + I_{out}) \tag{3.27}$$

$$P_t = V_{out} \cdot I_t, \tag{3.28}$$

$$P_v = (V_{in} - V_{out}) \cdot (I_{out}) \tag{3.29}$$

$$P_q = V_{in} \cdot I_q, \tag{3.30}$$

$$P_c = V_{out} \cdot I_c, \tag{3.31}$$

where $I_{out} = I_c + I_t$.

Two system-level metrics, the system power efficiency ε_s and the ground power P_g, are introduced to provide power performance measurements of the power delivery network at the heavy-load situation (the activity of the load circuits is high) and the zero/low-load situation (the load circuits are idle), respectively. The system power efficiency is defined as the ratio between the useful power and the total input power, as expressed in (3.32); the ground power is the sum of the LDO quiescent power and the decoupling capacitor leakage power, as expressed in (3.33). Note that the LDO power efficiency is partially reflected in system power efficiency.

$$\varepsilon_s = \frac{P_t}{P} = \frac{P_t}{P_t + P_v + P_q + P_c} \tag{3.32}$$

$$P_g = P_q + P_c. \tag{3.33}$$

When the decap leakage current and the LDO quiescent current are small compared with the load current, the system power efficiency is bounded by the ratio between the output voltage and the input voltage $\frac{V_{out}}{V_{in}}$. Therefore, lowering the input voltage becomes the most effective way to enhance system power efficiency (i.e. reducing the dropout voltage of LDO). However, with a significantly low input voltage, LDOs are in danger of working in the dropout region and losing the regulation on output voltage, which will degrade the system noise performance. Decreasing decaps or the number and sizes of LDOs can cut down the ground power, but may increase noise as well.

Noise: The noise of the entire system consists of the noise caused by load current variation and the noise induced by supply voltage fluctuation. The former can be improved by reducing the LDO output impedance, while the latter can be suppressed by having good LDO PSR. The output node of the regulator is analyzed as an example of a node in PDN.

To achieve low voltage noise on the node, the impedance at that node should be low at the frequency range where the major of power spectrum of load current variations lies. In todays high-performance IC, the rise/fall time of load current can be as fast as less than 1ns. Hence node impedance from DC to very high frequency should be considered. The DC impedance of LDO output node can be derived from (3.16) as

$$Z_{out\_DC} \approx \frac{1/g_{m1}}{g_{m2}/\left(g_{ds2}+g_{ds3}\right)\cdot\left(1+A_1A_2\right)}, \tag{3.34}$$

where A_1 and A_2 represent the two stage DC gains of EA. The impedance can be approximated as $1/\left(g_{m1}A_{CS}H_{ol\_DC}\right)$ in which H_{ol_DC} is the DC loop gain and A_{CS} is the DC gain of the current sensor at the output stage. Therefore, increasing g_{m1} or H_{ol_DC} can improve DC output impedance and hence better suppression over slow variations of load current. g_{m1} can be increased by increasing either w_1 or I_{ds1} which will increases the size or quiescent power of LDO. A_{CS} and the DC loop gain H_{ol_DC} can be enhanced by enlarging w_2, l_2, A_1 or A_2 which can either increase the area or lay more stress on stability.

At very high frequencies (VHF), (3.16) can be approximated as

$$Z_{out\_VHF} \approx R_s + 1/sC_d, \tag{3.35}$$

where R_s is in the order of tens of milliohms. Hence increasing C_d (decap) can significantly reduce Z_{out_VHF} thus VHF noise, but at the cost of large area and leakage current.

In the mid-frequency range, (3.16) shows that Z_{out} has two zeros corresponding to the two poles of EA. Due to the stability consideration, one of poles must be the dominant pole of the loop gain and it can be approximated as $p1 \approx g_{oa1}/(A_2C_1)$. The loop gain and Z_{out} start to degrade noticeably after p_1. To push this turning point up to a higher frequency (i.e. to increase g_{ma1} or decrease C_1) while maintaining the LDO stability, the non-dominant poles have to be moved to higher frequencies at the same time. For the output stage shown in Fig. 47, this can only be done by increasing g_{m1} and g_{m2} without increasing C_{gs1} (suppose C_d cannot be decreased for suppressing VHF noise). By this means, quiescent current will grow.

Area: The total area overhead (A) includes decoupling capacitor area (A_c) and LDO area (A_l). In general, the area overhead is in direct proportion to the amount of on-chip decoupling capacitance ($\mathrm{C_d}$) and the number of LDOs (N, if the area of each LDO remains a constant).

Placement & Routing: LDOs are placed at selected locations, termed as LDO blocks. Each block can contain multiple LDOs. To have good noise performance, it is desired to have the LDO circuit blocks spread out on the chip. However, placing each LDO circuit block on chip not only has its own placement and routing overheads but may also reduce placement freedom of other circuit blocks and cause extra wire routing efforts. In a rough estimation, the LDO placement & routing overhead is proportional to the number of LDO circuit blocks on chip (M). It should be noted that due to the limited scope of the paper, the placement & routing overhead of decoupling capacitance is not considered.

Stability: A strict approach to check load regulation stability is to check the poles of the whole network's impedance under a wide range of load current conditions. However, the sheer complexity of the network is too huge for this approach to be done with a bearable cost. To get insights for developing a feasible stability checking approach that is empirical and safe, first imagine an extreme case where interconnect parasitic resistances are set to zero. Then all the nodes in the power grids are actually the same node. As a result, the output pins of LDOs are tied to one node, so are those decaps. The circuit evolves into multiple LDOs in parallel connection as a whole driving a huge capacitor whose value is the total amount of decaps (illustrated in Figure 50(b)). Then, since all the LDOs are identical, it is fair to say that each LDO drives a load capacitor equal to C_t/N, where N is the number of LDOs and C_t is the total decoupling capacitance. The stability check of this system is done by checking the stability of each LDO-capacitor pair (illustrated in Figure 50(c)). It is well-known in typical LDO design that if each LDO is well designed, adding back a resistor of up to several hundred mili-Ohm (R_s) representing interconnect parasitic resistance between the output of LDO and the capacitor (as shown in Figure 50(d)) will improve the stability of the LDO-capacitor pair. This improvement is proved by the circuit simulation results shown in Figure 51. As can be seen, when the resistor value increases from zero to a typical upper bound of interconnect resistance, the phase margin of the open-loop transfer function of the pair gets better. Therefore, the setup in Figure 50(c) is used as a conservative stability check for the entire network. If the circuit in Figure 50(c) has a right-hand-pole, then the entire system is treated as unstable for the safety. Our experiments have empirically demonstrated the good robustness of the approach.

A single LDO-decap pair as shown in Figure 50(c) is used to illustrate how the capacitive load (C_t/N) affects the stability of the system. The pole p_1 from EA is designed as the dominant pole. p_3 and p_4 are non-dominant poles and are at high

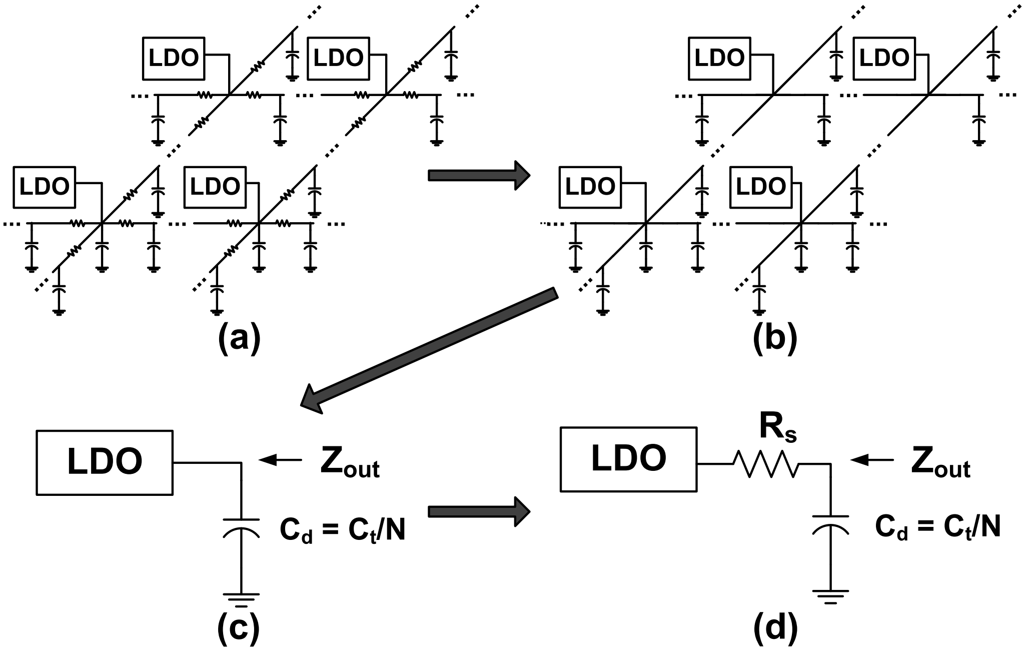

Fig. 50. Stability check reasoning procedure.

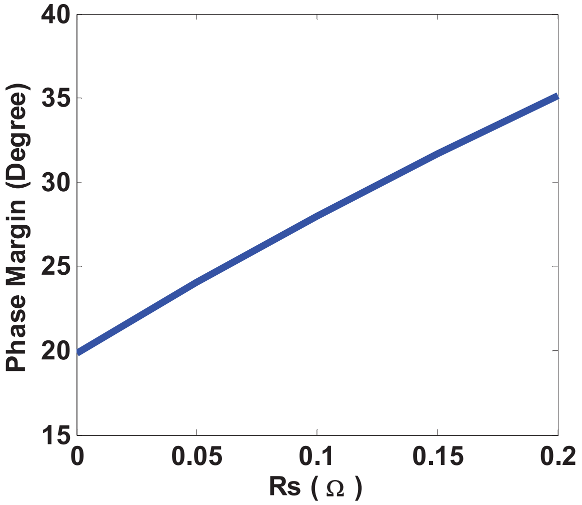

Fig. 51. Phase margin vs. R_s.

frequencies. Detailed reasoning is well-known in typical LDO design and left out here for conciseness. Then, there are two optional positions for p_2: one is between p_1 and p_3, p_4; the other is beyond p_3, p_4. If p_2 is at the first position, p_3, p_4 should be sufficiently higher than p_2 to prevent severe phase drop near p_2 that endangers stability. When C_t/N is increased, according to the root locus of p_3, p_4 with respect to C_t/N, p_3, p_4 will slide down to lower frequencies. Then, either increasing I_q to have higher p_3 and p_4 or enlarging C_2 or even C_1 to lower p_1 and p_2 will help to increase the phase margin. If p_2 is at the second optional position, high DC loop gain should be avoided to make the unity-gain frequency lower than p_3, p_4. Otherwise, if C_t/N is small enough, p_3, p_4 is possible to clustered with p_2, which degrades stability. In this case, enlarging C_1 or reducing DC loop gain H_{ol_DC} will help. However, this will degrade DC or low- and mid- frequency Z_{out} and PSR.

b. LDO-Decap-BC System Co-Design

After introducing the buck converter into the entire power delivery system co-design framework, the following aspects would be impacted:

- Power: In this case, the buck converter is no longer assumed to be ideal and its power efficiency is ε_b. Therefore, the entire system power efficiency ε_s is expressed as

  $$\varepsilon_s = \varepsilon_b \varepsilon_{ld}, \tag{3.36}$$

  where ε_{ld} is the power efficiency for the LDO-decap system which can be obtained by (3.32). Moreover, the ground power P_g has to include the power loss of the buck converter at zero-load state P_{b0}.

  $$P_g = P_q + P_c + P_{b0}. \tag{3.37}$$

- Noise: The output voltage of the buck converter (the input voltage to the LDO-decap system) has intrinsic voltage ripple due to the charge and discharge operations. This ripple would influence the output voltage of the LDO. The power supply rejection of LDO determines how well the network driven by the LDO is isolated from the buck converter output voltage ripple and is important to suppressing the mid-frequency resonance induced by the package parasitic inductance and on-chip decap. As can be seen from (3.19), PSR can be improved by increasing loop gain H_{ol} in which the decap C_d plays an important part.

- On-board component cost: The inductor (L_f) and capacitor (C_f) at the output stage of the buck converter mainly determine the accuracy and ripple of the BC output voltage. However, their values can not be too large since there are costs associated with them. Although the cost function is complex, in general, we assume the costs are in proportion to the L_f and C_f values.

3. Co-Optimization Formulation and Methodology

In order to design a PDN with the optimal performance, it is desired to use the optimal parameters for different network components, which naturally leads to a system-level co-optimization problem. We first introduce the co-optimization formulation and methodology for the LDO-decap system and then bring the buck converter into the picture.

a. Co-Optimization for LDO-Decap System

As the analysis presented in Section III.B.2.a, making one design aspect better can not guarantee to make other aspects better. For a power delivery network design, in order to achieve the most desired system performance (i.e., under a presumed weighing on noise performance, power, area, etc., the best overall system performance), optimal tradeoffs between LDO and decap designs must be reached. Such a task leads to the co-optimization of active on-chip regulator circuits and passive decoupling capacitors.

In this part, the optimization formulation is introduced first, which is followed by the illustration of the entire optimization flow. Finally, a multi-level optimization strategy is proposed to efficiently solve PDNs with large dimensions.

Objective Function: Assume there are K power domains $\{G_1, \ldots, G_K\}$ on chip. Domain G_i has M_i LDO blocks and the total number of N_i LDOs. The total decoupling capacitance in domain G_i is C_{ti}. The area overhead (A), power measurements (ε_s and P_g), placement & routing overhead (R) and noise (n) of the entire system can be expressed as:

$$A = \sum_{i=1}^{K} \left(\alpha C_{ti} + N_i A_{li}\right), \tag{3.38}$$

$$\varepsilon_s = \frac{\sum_{i=1}^{K} P_{ti}}{\sum_{i=1}^{K} P_i}, \tag{3.39}$$

$$P_g = \sum_{i=1}^{K} \left(P_{qi} + P_{ci}\right), \tag{3.40}$$

$$R = \sum_{i=1}^{K} \gamma M_i, \tag{3.41}$$

$$n = \sum_{i=1}^{K} \sum_{j=1}^{L_i} n_{ji}, \tag{3.42}$$

where α is the capacitance area ratio; A_{li} is the area for the LDO; γ is the placement & routing overhead coefficient; L_i is the number of nodes in D_i; n_{ji} is the noise violation integral for node j defined in [38] [52].

$$n_{ji} = \int_0^T \max\left\{v_l - v_{ji}(t), v_{ji}(t) - v_u, 0\right\} dt, \tag{3.43}$$

where v_u and v_l are the supply voltage upper and lower bounds respectively. A boolean variable S is introduced to represent the stability of the system (1 for stable and 0 for unstable).

Since all these terms have different unit and nominal values, normalization is applied. Then, the optimization is to minimize the objective function f which can be expressed as

$$f = a_1 \frac{A}{A'} + a_2 \frac{1 - \varepsilon_s}{1 - \varepsilon_s'} + a_3 \frac{P_g}{P_g'} + a_4 \frac{R}{R'} + a_5 \frac{n}{n'} + a_6 \frac{S}{S'}, \tag{3.44}$$

where $\{a_1, \ldots, a_6\}$ and $\{A', \varepsilon_s', P_g', R', n', S'\}$ are the weights and nominal values for area, power efficiency, ground power, placement & routing overhead, noise, and stability respectively. Obviously, by assigning a large weight to a term, that term would be optimized with preference. Since the stability must be assured and the noise violation must be zero, their weights a_5 and a_6 are assigned with large values.

Optimization Variables: In principle, any design parameters associated with LDOs and decaps can be considered as optimization variables, which would make the optimization space huge and the runtime cost unbearably high. Therefore, without the loss of generality, a few assumptions are made to reduce the optimization complexity:

- The LDO blocks are evenly distributed in the power delivery network, and each block contains the same number of LDOs. $M_i = X_i \times Y_i$ and $N_i = X_i \times Y_i \times Z_i$.
- The LDOs in the same domain are the same.
- The locations of decaps are determined and the amount of the decap at each location can be increased or decreased in proportion in a certain range.

In summary, for each power domain D_i, the optimization variables are the number of LDO circuit blocks (M_i, namely X_i and Y_i), the number of LDOs in each LDO

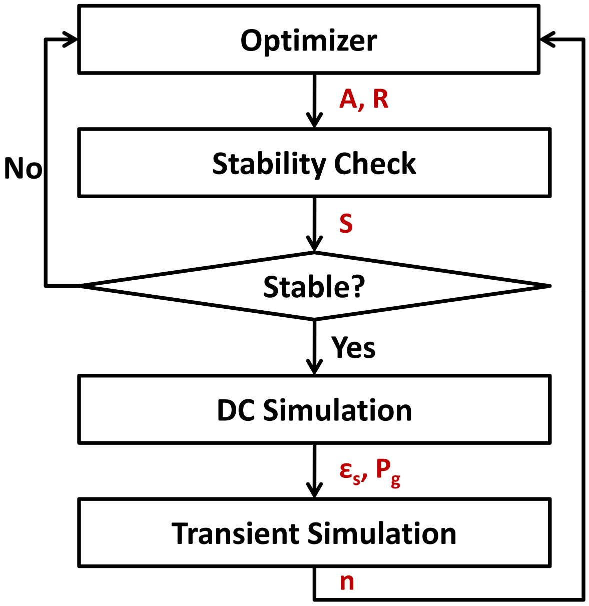

Fig. 52. LDO-decap system co-optimization flow.

block (Z_i), total decoupling capacitance (C_{ti}). On the LDO side, according to the design analysis presented in Section III.B.1, we choose the following subset of electrically important design parameters: the width of LDO pass transistor (w_{1i}) which determines the maximum load current and dropout voltage, the transistor width and length at the output stage of LDOs (w_{2i}, w_{3i}, l_{2i}) which decide the quiescent current and output impedance at light-load situation, and the compensation capacitors of LDOs (C_{1i}, C_{2i}) which ensure stability. Apparently, all the optimization variables are bounded. Therefore, the co-optimization problem is a bound-constrained optimization problem.

Optimization Flow: Asynchronous Parallel Pattern Search (APPS) [53] which solves unconstrained or bound-constrained optimization problems is employed to solve the above co-optimization problem. APPS is a search-based optimization approach requiring no derivative information. For our problem, each search (iteration) consists of three steps to evaluation the objective function as shown in Figure 52:

1. Stability check: Based on the observations in Section III.B.2.a, we propose an efficient stability check approach. First, the LDO-capacitor pair is generated from the new set of variable values chosen by the optimizer. Then, complete pole-zero analysis for the LDO-capacitor pair under various load conditions (from zero to maximum) are performed. If a right-half-plane pole is detected, the DC and transient simulations are skipped and optimizer moves to next search.

2. DC simulation: The DC simulation is carried out by an efficiency CPU-GPU combined simulator GSim. The system power efficiency ε_s and ground power P_g are obtained.

3. Transient simulation: The noise performance n is computed from the transient simulation which is also performed by GSim.

The area overhead A and the placement & routing overhead R can be determined right after the optimizer chooses a new set of values for the optimization variables. Then, based on the value of the objective function, a new set of optimization variable values is chosen for evaluation.

Multigrid-Based Optimization: For the proposed optimization scheme, hundreds of iterations may be needed to get the optimal result. In each optimization iteration, a costly transient simulation may be carried out. In addition, most of the simulation time is spent on solving large on-chip grids. Therefore, when optimizing for large power delivery network with multi-million on-chip nodes, the excessive runtime becomes a bottleneck. In order to address this problem, a multigrid-based optimization strategy is proposed to significantly reduce the optimization runtime for large PDNs.

Multigrid method is a very powerful and effective methods addressing power grid problems which have very large sheer complexity, for example power grid simulation

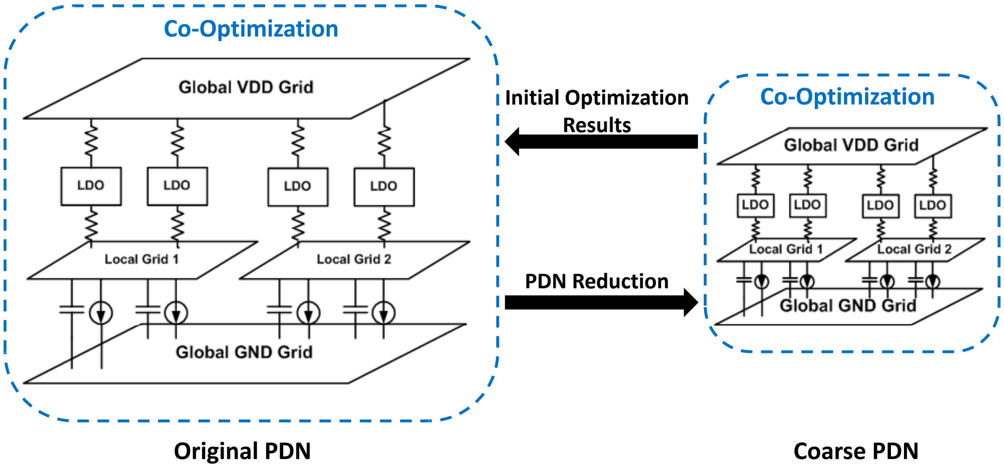

Fig. 53. Multigrid-based optimization.

[9] [10] [11] [12] [13] [14] and wire sizing optimization [28] problems. The basic idea of this method is to reduce the original large-scale problem into a much smaller problem using coarsening process. Then the small-scale problem is solved. Finally, the results of original problem is computed based on the solutions that are back-mapped from the small-scale problem results. The benefits brought by this method is the significant runtime and memory savings. This idea is adopted for the proposed active-passive co-optimization to reduce optimization runtime for large power delivery networks.

As illustrated in Figure 53, the proposed multi-grid method consists of three steps:

1. Generate a coarse PDN using the methods presented in [13].

2. Carry out the optimization for the coarse PDN.

3. Use the optimum variable values for the coarse PDN as the initial values for the optimization of the original PDN.

Since the coarse PDN has the same basic electric properties of the original PDN, such as load current, total decoupling capacitance and branch resistance, the optimization results for the coarse PDN are close to the results for the original PDN. Hence, using the results of the course PDN as the starting point, only a small number of optimization iterations may be required to reach the final optimum of the original PDN. It will be shown in the experimental results that this multigrid-based optimization strategy can significantly reduce the optimization runtime cost.

b. Co-Optimization for LDO-Decap-BC System

After introducing the buck converter into the co-optimization framework, the following modifications to the optimization must be carried out (the notations are the same as in the LDO-decap co-optimization):

- Objective function: According to (3.36) and (3.37), the system power efficiency and ground power should include the effect from the buck converter.

$$\varepsilon_s = \varepsilon_b \frac{\sum_{i=1}^{K} P_{ti}}{\sum_{i=1}^{K} P_i}, \tag{3.45}$$

$$P_g = P_{b0} + \sum_{i=1}^{K} \left(P_{qi} + P_{ci}\right). \tag{3.46}$$

  Moreover, the term $a_7 \frac{L_f}{L_f'} + a_8 \frac{C_f}{C_f'}$ which reflects the on-board component cost should be added to the objective function f. The a_7 and a_8 are the weight coefficients while L_f' and C_f' are the nominal values for the BC inductor and capacitor.

- Optimization variables: Besides the existing optimization variables shown in Section III.B.3.a, four new key parameters of the buck converter, switching frequency f_s, duty cycle D, output filter inductor L_f and capacitor C_f are introduced as optimization variables. Furthermore, four transistor size parameters of the LDO that impact the PSR are also included as optimization variables.

- Optimization flow: The basic optimization flow is almost the same as presented in Section III.B.3.a except that at the beginning of each optimization search (iteration), the input DC voltage V_{bc_avg} as well as the input voltage waveform (with ripple ΔV) to the LDO-decap system, buck converter power efficiency ε_b and zero-load power loss P_{b0} are computed based on the behavior model of the buck converter as shown in (3.20-3.23).

It should be noted that the multi-level optimization strategy can still be applied here.

4. Experiment Results

The LDO and decoupling capacitors are implemented using a commercial 90nm technology. The PDN simulator GSim has been implemented in CUDA [44] and C++. The optimization is carried out using APPSPACK 5.0.1 [53]. The GPU program is executed on the NIVIDIA Tesla C1060, with a total on board memory of 4GB. All the C++ programs are executed on a workstation with Intel Xeon CPU at 2.33GHz and 4G RAM running 64-bit Linux OS.

A test PDN with 1M on-chip nodes is used to illustrate the co-optimization benefits and various kinds of design tradeoffs. The test PDN is generated for a chip with 26mm^2 area, 2W power consumption, 1.2V supply voltage from on-board buck converter, 1V on-chip voltage, 2A average current and 4A peak current. The transition time for each current loading is 500ps. The noise bound is set as ±50mV. In the initial design, there are 40nF decoupling capacitance and 42 blocks of LDOs (1 LDO in each block). Gate-oxide capacitors are used for decaps and LDO compensation capacitors. Without the loss of generality, the entire chip is treated as a single power domain. It should be noted that for all the optimizations, stability and zero noise violation are strictly enforced.

a. LDO-Decap System Co-Optimization

The following experimental results are obtained only for the LDO-decap system co-optimization (in Section III.B.3.a) which assumes an ideal on-board buck converter.

Co-Optimization vs. Uni-Optimization: Starting from the same initial design, three optimizations are carried out: decap uni-optimization, LDO uni-optimization, and decap & LDO co-optimization. The results are presented in Figure 54.

- Decap uni-optimization: Blindly optimizing decaps without any stability checking can easily lead to instability. With the proposed stability check in place, since the LDO design can not be adjusted according to the capacitive load variation, the total decap can only be reduced to 30nF before losing stability.
- LDO uni-optimization: Although the number of LDO blocks is reduced to 36 and each LDO is sized to have the total quiescent power as low as 9.3mW, due to the initial large decap, the area is almost the same as the initial design and the leakage power remains at 76.6mW.
- LDO & decap co-optimization: On one hand, since the LDO design parameters can be adjusted to avoid stability problem, the decap is reduced to 18nF and the decap leakage power is cut down to 46mW. On the other hand, the number of LDO blocks is reduced to 36. Although compared with the results of LDO uni-optimization, the quiescent power is as large as 27.8mW, with the decap leakage power reduction, the total ground power is as low as 73.8mW.

As can be observed, compared with the uni-optimization, co-optimization has the advantage of mutually and simultaneously adjusting decap and LDO designs, and its optimum result is significantly better (or no worse) in every design aspects. Therefore, in order to achieve the system-level optimum, the co-optimization is indispensable.

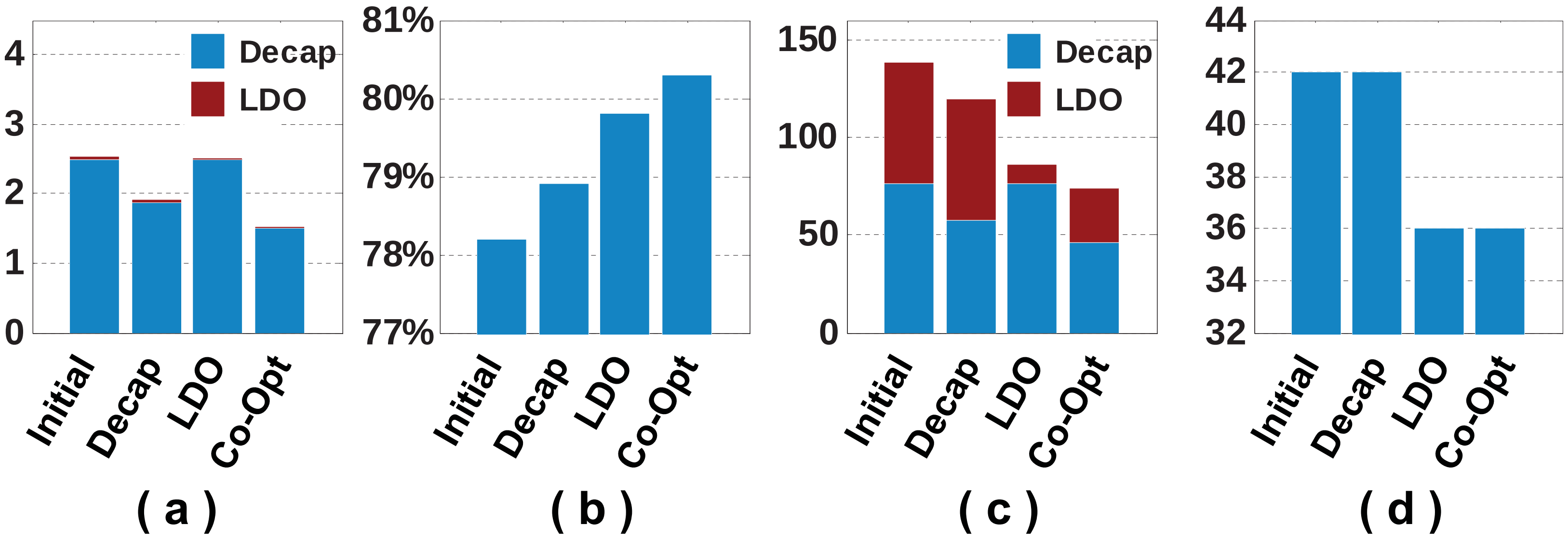

Fig. 54. Optimization results of decap uni-optimization, LDO uni-optimization, and co-optimization. (a) LDO and decap area in mm^2. (b) System power efficiency. (c) Ground power in mW. (d) Number of LDO blocks.

Design Tradeoffs: Different sets of weights are used to drive the optimization favoring different design aspects: area, power, placement & routing, and noise. These optimization results are compared with the result that is obtained using balanced weights. For the noise optimization, since the initial noise bound is very tight, the noise bound is extended to ±100mV to observe the trend. The results are presented in Figure 55.

- Area optimization: The total decap is reduced to 8nF which significantly cut down the area and leakage power (65.1% reduction). However, LDO block numbers are increased (16.7% increase) and the LDOs are sized up to suppress the noise which leads to more quiescent power (107% increase).

- Power optimization: The decap amount (12nF) and the LDO sizes are reduced, but the number of LDO blocks increases (16.7% increase). The ground power is as small as 58.3mV (21% reduction).

- Placement & routing optimization: The number of LDO blocks is cut down to 20 (40% reduction, 2 LDOs in each block). However, more decaps and larger LDOs are required to keep down the noise. Area overhead is increased by 65% and ground power is increased by 62%.

- Noise optimization: With relaxed noise constraint, noticeable area, power and placement & routing reductions (41.4%, 32.9%, 16.7% respectively) are observed. As can be inferred, when the noise bound becomes tighter, there will be increases in area, power and placement & routing.

It can be seen, reducing the overhead in one aspect may increase the overheads in other aspects. Hence, based on the design specifications, the weights for area, power, and placement & routing must be well set.

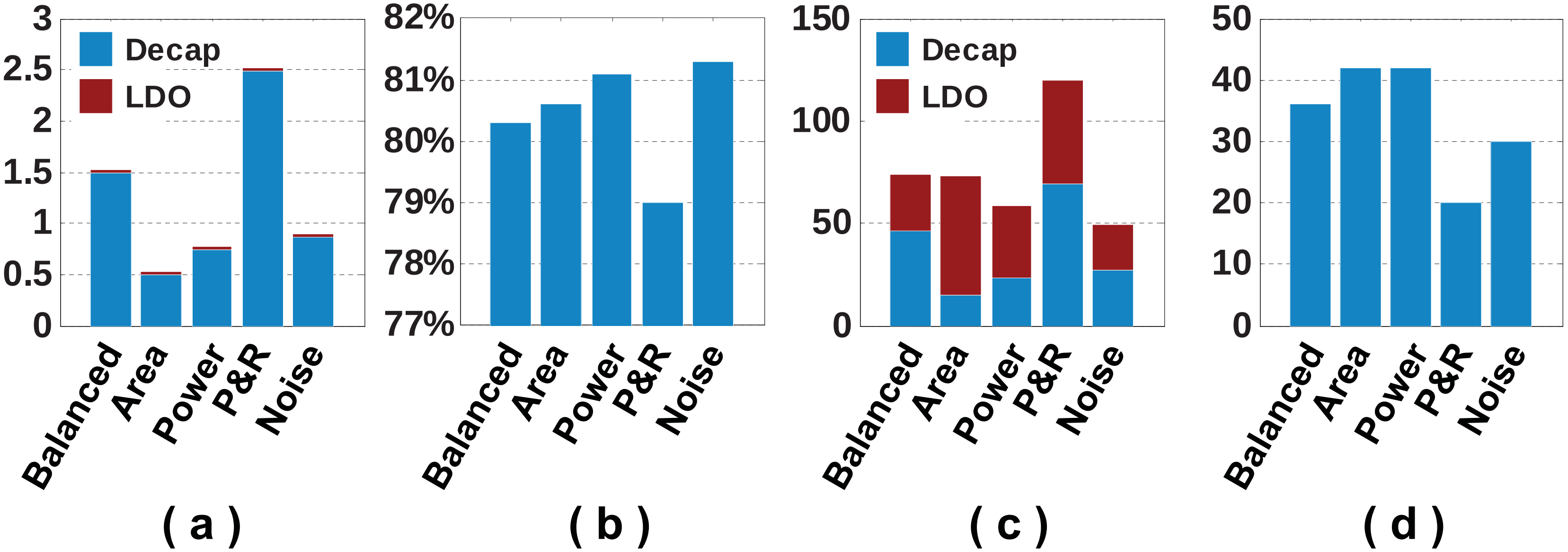

Fig. 55. Optimization results of favoring different design aspects. (a) LDO and decap area in mm^2. (b) System power efficiency. (c) Ground power in mW. (d) Number of LDO blocks.

Deep Trench Decoupling Capacitor: Deep trench decoupling capacitor is a novel decap that is fabricated in a similar manner to bulk eDRAM deep trench capacitor [54]. The advantages of this technology over the typical planar gate-oxide decoupling capacitor is its high capacitance per unit area as well as low leakage current. For example, in the commercial 90nm technology we use, for every one um^2, the planar gate-oxide capacitor only provides 16fF capacitance and draws 30nA leakage current. However, on the other hand the deep trench decoupling capacitor has 112fF capacitance and negligible leakage current (less than 0.05pA).

Comparison between the typical gate-oxide technology and the deep trench technology is done by using the same set of weights for the optimizations. The results are presented in Figure 56. Without considering the manufacturing costs, the electric benefits of using deep trench decoupling capacitor is obvious. Since the deep trench decap is not as area-costly as the gate-oxide decap, more decap can be placed on chip with even much less area consumption (36nF decap for only 0.32mm^2). Moreover, with the strong decap to suppress the noise, the number of LDOs and LDO sizes can be reduced, which results in less quiescent power and placement & routing overhead. There are only 30 LDO blocks (1 LDO per block), and the quiescent power is as low as 23.2mW. Notice that there is almost no leakage power consumption from the deep trench decap.

In summary, using deep trench decoupling capacitors would significantly enhance the power delivery system performance, but the hardware cost may be high.

Multi-Level Optimization Strategy: The multi-level optimization strategy is applied to a PDN with 5.3M on-chip nodes. The coarse PDN is generated from the original PDN and the total number of on-chip nodes is reduced to 270K.

The optimization iterations and runtime for the coarse PDN and the original PDN are shown in Table XII. It can be seen that by using the optimization results

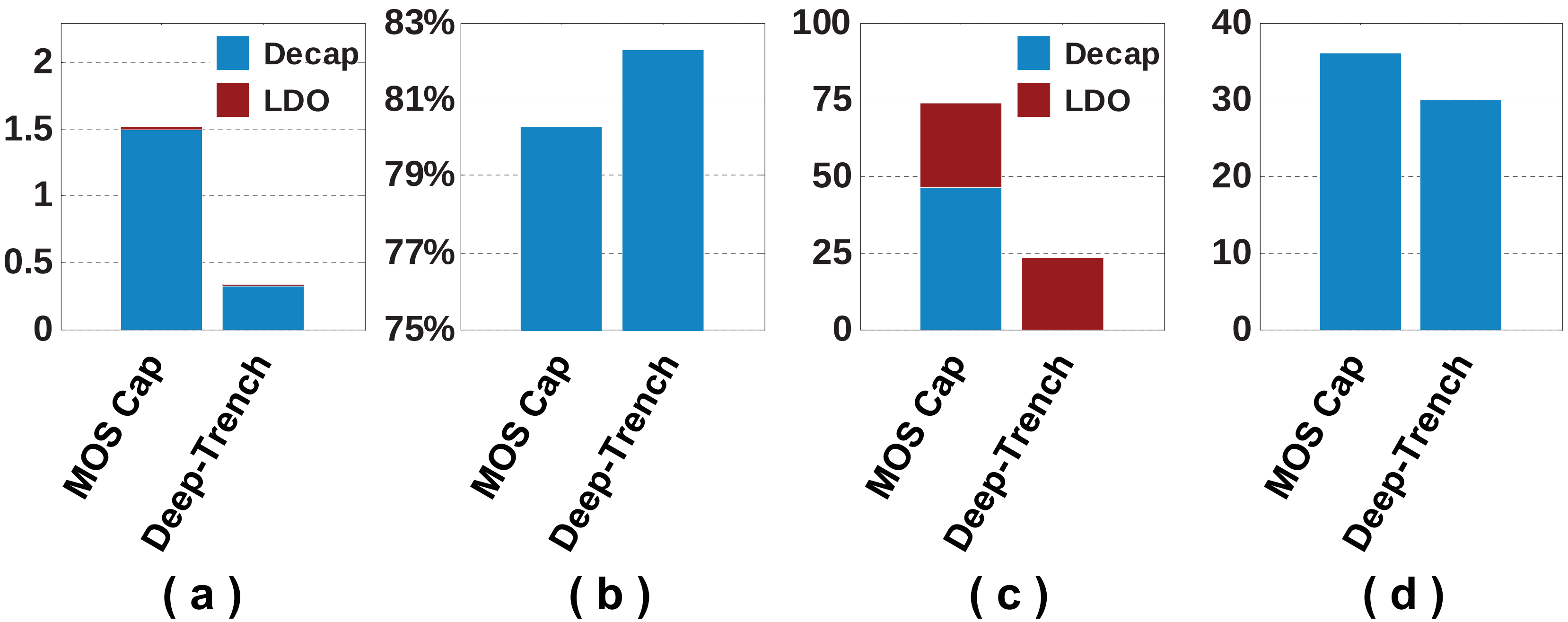

Fig. 56. Optimization results for using planar gate-oxide decoupling capacitance vs. deep trench decoupling capacitance (a) LDO and decap area in mm^2. (b) System power efficiency. (c) Ground power in mW. (d) Number of LDO blocks.

Table XII. Multi-level optimization vs. straight optimization.

Straight Optimization		Multigrid-Based Optimization				
# Iter.	Time	Coarse PDN		Original PDN		SP
		# Iter.	Time	# Iter.	Time	
290	64.7h	274	8.7h	34	6.4h	4.3X

of the coarse PDN as the initial values, the number of optimization iterations for the original PDN is significantly reduced. Compared with the optimization using the straight method (without using multigrid), over 4.3X speedup is reached. This shows the superior performance and effectiveness of the multi-level method for solving very large PDN co-optimization problems.

b. LDO-Decap-BC System Co-Optimization

From the previous optimization results, it is observed that the system power efficiency does not vary obviously. This is because the on-board buck converter, which provides voltage supply to the LDO-decap system, is not considered in the co-optimization framework, and only optimizing the LDO-decap system does not bring much benefits to improve the power efficiency. Therefore, we bring the buck converter into the system co-optimization as depicted in Section III.B.3.b.

Assume the initial buck converter design has $V_{in} = 3.6V$, $f_s = 5MHz$, $D = 0.389$, $L_f = 2uH$, $C_f = 2uF$, and its power efficiency $\varepsilon_b = 87\%$, average output voltage $V_{bc\_avg} = 1.2V$, and ripple $\Delta V = 1mV$. Optimization that only considers LDO-decap system and optimization that considers LDO-decap system as well as the buck converter are run with the same set of weights. As shown in Figure 57, after introducing the on-board buck converter into the co-optimization, the V_{bc_avg} is reduced to as low as 1.096V, then, the system power efficiency (79.7%) is significantly

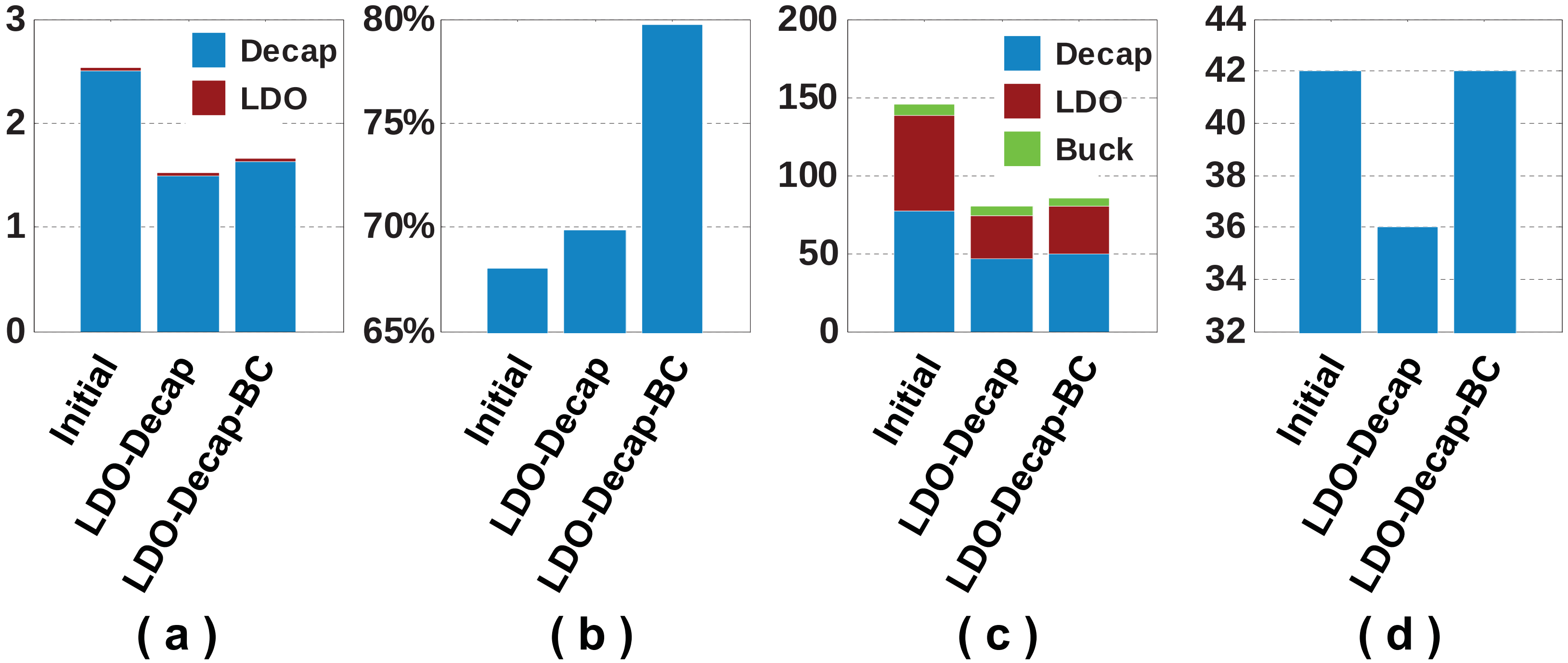

Fig. 57. Optimization results of LDO-decap-BC system co-optimization vs. LDO-decap system co-optimization. (a) LDO and decap area in mm^2. (b) System power efficiency. (c) Ground power in mW. (d) Number of LDO blocks.

better than the LDO-decap system co-optimization (69.8%). Moreover, LDO PSR are strengthened to tolerate a larger ripple $\Delta V = 6mV$ so that the L_f and C_f are reduced to $1uH$ and $0.94uF$ respectively. However, the gain of power efficiency does not come at no cost. By lowering the input voltage, the LDO is moving towards the dropout region so that it has weaker regulation capability on the load variance. Hence, with low input voltage, slightly more decap and LDO resources are needed to maintain well noise performance. In summary, including buck converter into the system co-optimization can significantly improve the overall power efficiency.

5. Summary

In this section, by giving a comprehensive analysis on the the electric interaction between voltage regulators and decoupling capacitance, their mutual influences and design tradeoffs on various system design aspects, such as power, noise, area, placement and routing, as well as stability are studied in detail. A thorough system-level co-optimization scheme which can simultaneously optimize active regulators and decaps is proposed. The results demonstrate the significant performance improvements brought by the holistic system co-optimization.

CHAPTER IV

CONCLUSION

This dissertation presents new methodologies and approaches to address a series of challenging issues in power delivery network analysis, verification and design.

On the analysis aspect, a parallel partitioning based power grid analysis approach using the spatial locality is presented. The main factors that effect the solution process have been identified, which are the number of partitions and the window size. The interdependence of these parameters and their influence on the execution time have been analyzed. A strategy that helps users in determining the optimal (or near optimal) values of these parameters to achieve lowest parallel runtime have also been suggested. The proposed approach is shown to have excellent parallel efficiency, fast convergence, flexible partitioning, and favorable scalability. By using distributed computing networks, it is believed to be able to handle extremely large power grids (with many-million nodes) in an efficient way. Moreover, a fast CPU-GPU combined simulation engine, GSim, has been developed to provide SPICE-level accuracy for simulating complex PDNs employing on-chip voltage regulation techniques. GSim identifies the simulation difficulties for different circuit blocks and achieves its efficiency by using a block-based Gauss-Seidel relaxation scheme to integrate three fast simulation strategies together. These simulation strategies are specifically designed for three types of circuit blocks in the PDN. Most importantly, GSim provides a foundation to comprehensively analyze various design tradeoffs for PDNs with on-chip voltage regulation.

In terms of verification, a simulation-based transient verification approach has been proposed. Specific circuit modeling techniques have been developed to individually verify each of the on-chip global and local power grids against given electromigration and voltage drop constraints. To achieve verification feasibility, the proposed approach allows the use of fast superposition approximation methods to identify the top worst-case candidates. These candidates are later validated by a small number of full simulations.

At last, for PDN design, a novel partition-based locality-driven two-step optimization scheme has been developed for power grid wire sizing optimization. This scheme exploits the locality of the power grids for scalable optimization of large grids. In the first step of the proposed approach, optimal (or near optimal) partitioning boundary voltages and currents are obtained via localized window-based optimization. In the following step, partitions are individually optimized under the obtained boundary conditions. The *divide-and-conquer* nature of the proposed method not only leads to its favorable scalability but also makes it possible to employ the increasing parallel computing resources to facilitate the optimization of large power grids. In addition, by giving a comprehensive analysis on the the electric interaction between voltage regulators/converters and on-chip decoupling capacitors. The mutual influences and design tradeoffs of the active regulators/converters and the passive decoupling capacitors on various system design aspects, such as power, noise, area, placement and routing, as well as stability are studied in detail. A thorough system-level simulation-based co-optimization scheme which can simultaneously optimize regulators and decaps has been proposed. The results demonstrate the significant performance improvements brought by the holistic system co-optimization.

REFERENCES

[1] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective.* Boston, MA: Addison Wesley, 2010.

[2] Q. K. Zhu, *Power Distribution Network Design for VLSI*, Hoboken, NJ: John Wiley & Sons, 2004.

[3] J. Shah, "Floorplanning a power delivery network with spice", 2008 [Online]. Available: http://electronicdesign.com/article/eda/floorplanning-a-power-delivery-network-with-spice1.aspx.

[4] K. Arabi, R. Saleh, and X. Meng, "Power supply noise in SoCs: metrics, management, and measurement", *IEEE Design & Test of Computers*, vol. 24, no. 3, pp. 236-244, 2007.

[5] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective.* Upper Saddle River, NJ: Prentice Hall, 2004.

[6] M. S. Gupta, J. L. Oatley, R. Joseph, G. Wei, and D. M. Brooks, "Understanding voltage variations in chip multiprocessors using a distributed power-delivery network", In *Proc. IEEE/ACM DATE*, 2007, pp. 783-790.

[7] H. H. Chen and D. D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design", In *Proc. IEEE/ACM DAC*, 1997, pp. 638-643.

[8] T.-H. Chen and C. C.-P. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods", In *Proc. IEEE/ACM DAC*, 2001, pp. 559-562.

[9] S. R. Nassif and J. N. Kozhaya, “Fast power grid simulation”, In *Proc. IEEE/ACM DAC*, 2000, pp. 156-161.

[10] J. Kozhaya, S. Nassif, and F. Najm, “A multigrid-like technique for power grid analysis”, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 10, pp. 1148-1160, 2002.

[11] H. Su, E. Acar, and S. Nassif, “Power grid reduction based on algebraic multigrid principles”, In *Proc. IEEE/ACM DAC*, 2003, pp. 109-112.

[12] C. Zhuo, J. Hu, M. Zhao, and K. Chen, “Power grid analysis and optimization using algebraic multigrid”, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 4, pp. 738-751, 2008.

[13] Z. Feng and P. Li, “Multigrid on GPU: tackling power grid analysis on parallel SIMT platforms”, In *Proc. IEEE/ACM ICCAD*, 2008, pp. 647-654.

[14] Z. Feng, Z. Zeng, and P. Li, “Parallel on-chip power distribution network analysis on multi-core-multi-GPU platforms”, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 10, pp. 1823-1836, 2011.

[15] M. Zhao, R. V. Panda, S. S. Sapatnekar, T. Edwards, R. Chaudhry and D. Blaauw, “Hierarchical analysis of power distribution networks”, In *Proc. IEEE/ACM DAC*, 2000, pp. 150-155.

[16] M. Zhao, R. Panda, S. Sapatnekar, and D. Blaauw, “Hierarchical analysis of power distribution networks”, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 2, pp. 159-168, 2002.

[17] K. Sun, Q. Zhou, K. Mohanram, and D. Sorensen, “Parallel domain decomposition for simulation of large-scale power grids”, In *Proc. IEEE/ACM ICCAD*,

2007, pp. 54-59.

[18] H. Qian, S. R. Nassif, and S. S. Sapatnekar, “Power grid analysis using random walks”, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 8, pp. 1204-1224, 2005.

[19] E. Chiprout, “Fast flip-chip power grid analysis via locality and grid shells”, In *Proc. IEEE/ACM ICCAD*, 2004, pp. 485-488.

[20] Y. Zhong and M. D. F. Wong, “Fast algorithms for IR drop analysis in large power grid”, In *Proc. IEEE/ACM ICCAD*, 2005, pp. 351-357.

[21] J. Shi, Y. Cai, W. Hou, L. Ma, S.-D. Tan, P.-H. Ho, and X. Wang, “GPU friendly fast Poisson solver for structured power grid network analysis”, In *Proc. IEEE/ACM DAC*, 2009, pp. 178-183.

[22] F. Najm, M. Nizam, and A. Devgan, “Power grid voltage integrity verification”, In *Proc. ACM/IEEE ISLPED*, 2005, pp. 239-244.

[23] I. A. Ferzli, F. N. Najm and L. Kruse, A geometric approach for early power grid verification using current constraints, In *Proc. IEEE/ACM ICCAD*, 2007, pp. 40-47.

[24] I. Ferzli, F. Najm, and L. Kruse, “Early power grid verification under circuit current uncertainties”, In *Proc. ACM/IEEE ISLPED*, 2007, pp. 116-121.

[25] I. Ferzli, E. Chiprout, and F. Najm, “Verification and co-design of the package and die power delivery system using wavelets”, In *Proc. IEEE EPEP*, 2008, pp. 7-10.

[26] A. Todri, M. Marek-Sadowska, and S. Chang, "Analysis and optimization of power-gated ICs with multiple power gating configurations", In *Proc. IEEE/ACM ICCAD*, 2007, pp. 783-790.

[27] X. D. S. Tan, C. J. R. Shi, and J. C. Lee, "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings", *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 12, pp. 1678-1684, 2003.

[28] K. Wang and M. Marek-Sadowska, "On-chip power supply network optimization ssing multigrid-based technique", *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 3, pp. 407-417, 2005.

[29] J. Singh and S. S. Sapatnekar, "Partition-based algorithm for power grid design using locality", *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 4, pp. 664-677, 2007.

[30] P. Hazucha, G. Schrom, J. Hahn, B. Bloechel, P. Hack, G. Dermer, S. Narendra, D. Gardner, T. Karnik, V. De, and S. Borkar, "A 233-MHz 80%-87% efficient four-phase DC-DC converter utilizing air-core inductors on package", *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 838-845, 2005.

[31] J. Wibben and R. Harjani, "A high efficiency DC-DC converter using 2nH on-chip inductors", In *Proc. IEEE VLSIC*, 2007, pp. 22-23.

[32] K. N. Leung and P. Mok, "A capacitor-free CMOS low-dropout regulator with damping-factor-control frequency compensation", *IEEE J. Solid-State Circuits*, vol. 38, no. 10, pp. 1691-1702, 2003.

[33] P. Hazucha, T. Karnik, B. Bloechel, C. Parsons, D. Finan, and S. Borkar, Area-Efficient Linear Regulator with Ultra-Fast Load Regulation, *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 933-940, 2005.

[34] R. Milliken, J. Silva-Martinez, and E. Sanchez-Sinencio, "Full on-chip CMOS low-dropout voltage regulator", *IEEE Trans. Circuits Syst. I*, vol. 54, no. 9, pp. 1879-1890, 2007.

[35] J. Guo and K. N. Leung, "A 6uW chip-area-efficient output-capacitorless LDO in 90nm CMOS technology", *IEEE J. Solid-State Circuits*, vol. 45, no. 9, pp. 1896-1905, 2010.

[36] C. Shi, B. Walker, E. Zeisel, B. Hu, and G. McAllister, "A Highly Integrated Power Management IC for Advanced Mobile Applications", *IEEE J. Solid-State Circuits*, vol. 42, no. 8, pp. 1723-1731, 2007.

[37] M. Hammes, C. Kranz, D. Seippel, J. Kissing, and A. Leyk, "Evolution on SoC integration: GSM baseband-radio in 0.13um CMOS extended by fully integrated power management unit", *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 236-245, 2008.

[38] H. Su, S. Sapatnekar, and S. Nassif, "Optimal decoupling capacitor sizing and placement for standard-cell layout designs", *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 4, pp. 428-436, 2003.

[39] S. R. Nassif, "Power grid analysis benchmarks", In *Proc. IEEE/ACM ASPDAC*, 2008, pp. 376-381.

[40] J. Shi, Y. Cai, S.-D. Tan, J. Fan, and X. Hong, "Pattern-based iterative method for extreme large power/ground analysis", *IEEE Trans. Comput.-Aided Design*

Integr. Circuits Syst., vol. 26, no. 4, pp. 680-692, 2007.

[41] A. George, “Nested dissection of a regular finite element mesh”, *SIAM J. on Numer. Anal.*, vol. 10, no. 2, pp. 345-363, 1973.

[42] 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.

[43] X. Cai, H. Yie, P. Osterberg, J. Gilbert, S. Senturia, and J. White, “A relaxation/multipole-accelerated scheme for self-consistent electromechanical analysis of complex 3-D microelectromechanical structures”, In *Proc. IEEE/ACM ICCAD*, 1993, pp. 283-286.

[44] NVIDIA CUDA programming guide, 2009 [Online]. Available: http://www.nvidia.com/object/cuda.html.

[45] C. Alexander and M. Sadiku, *Fundamentals of Electric Circuits.* NY: McGraw Hill, 2000.

[46] R. Dutta and M. Marek-Sadowska, “Automatic sizing of power/ground (P/G) networks in VLSI”, In *Proc. IEEE/ACM DAC*, 1989, pp.783-786.

[47] A. Wächter and L. T. Biegler, “On the implementation of an interior point filter line search algorithm for large-scale nonlinear programming”, *Math. Program.*, vol. 106, no. 1, pp. 25-57, 2006.

[48] A. Wächter, C. Visweswariah and A. R. Conn, “Large-scale nonlinear optimization in circuit tuning”, *Future Gener. Comput. Syst.*, vol. 21, no. 8, pp. 1251-1262, 2005.

[49] X. Wu, X. Hong, Y. Cai, Z. Luo, C. K. Cheng, J. Gu and W. W. M. Dai, "Area minimization of power distribution network using efficient nonlinear programming techniques", *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 7, pp. 1086-1094, 2004.

[50] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Philadelphia, PA: SIAM, 1994.

[51] V. Kursun, S. Narendra, V. De and E. Friedman, "Analysis of buck converters for on-chip integration with a dual supply voltage microprocessor", *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 3, pp. 514-522, 2003.

[52] A. Conn, R. Haring and C. Visweswariah, "Noise considerations in circuit optimization", In *Proc. IEEE/ACM ICCAD*, 1998, pp. 220-227.

[53] G. A. Gray and T. G. Kolda, "Algorithm 856: APPSPACK 4.0: asynchronous parallel pattern search for derivative-free optimization", *ACM Trans. Math. Softw.*, vol. 32, no. 3, pp. 485-507, 2006.

[54] C. Pei, R. Booth, H. Ho, N. Kusaba, X. Li, M.-J. Brodsky, P. Parries, H. Shang, R. Divakaruni and S. Iyer, "A novel, low-cost deep trench decoupling capacitor for high-performance, low-power bulk CMOS applications", In *Proc. IEEE ICSICT*, 2008, pp. 1146-1149.

VITA

Zhiyu Zeng received the B.S. degree in electronic and information engineering from Zhejiang University, Hangzhou, China, in 2006. In the fall 2007, he was enrolled in the Department of Electrical and Computer Engineering at Texas A&M University as a Ph.D. student. He graduated in December 2011. His research interests include general circuit simulation, parallel optimization of power delivery networks and system-level design of on-chip voltage regulation.

His permanent address is: Department of Electrical and Computer Engineering, Texas A&M University, 214 Zachry Engineering Center, TAMU 3128 College Station, Texas 77843-3128. His email address is albertzeng0308@gmail.com.

The typist for this thesis was Zhiyu Zeng.