

A Selected Inversion Approach for Locality Driven Vectorless Power Grid Verification

Jianlei Yang, *Student Member, IEEE*, Yici Cai, *Senior Member, IEEE*,
Qiang Zhou, *Senior Member, IEEE*, and Wei Zhao

Abstract—Vectorless power grid verification is a practical approach for early stage safety check without input current patterns. The power grid is usually formulated as a linear system and requires intensive matrix inversion and numerous linear programming (LP), which is extremely time-consuming for large-scale power grid verification. In this paper, the power grid is represented in the manner of domain-decomposition approach, and we propose a selected inversion technique to reduce the computation cost of matrix inversion for vectorless verification. The locality existence among power grids is exploited to decide which blocks of matrix inversion should be computed while remaining blocks are not necessary. The vectorless verification could be purposefully performed by this manner of selected inversion, while previous direct approaches are required to perform full matrix inversion and then discard small entries to reduce the complexity of LP. Meanwhile, constraint locality is proposed to improve the verification accuracy. In addition, a concept of quasi-Poisson block is introduced to exploit grid locality among realistic power grids and a scheme of pad-aware partitioning is proposed to enable the selected inversion approach available for practical use. Experimental results show that the proposed approach could achieve significant speedups compared with previous approaches while still guaranteeing the quality of solution accuracy.

Index Terms—IR drop, locality, selected inversion, signal integrity, vectorless power grid verification.

I. INTRODUCTION

THE power delivery network (PDN) of an integrated circuit delivers the power supply and ground voltages to the circuit, from voltage regulation module, through the chip package, and finally the on-die power grid. To guarantee normal circuit operation, a well-designed PDN must deliver well-regulated voltages to the circuit cells and meet the design tolerance. With the continuous scaling of process technology, a trend of decreasing supply voltages, increasing power density, and tighter noise margins has been observed to have a critical impact on voltage integrity, which is becoming a big concern for high-performance integrated circuits design. Excessive voltage fluctuation on the power grids can

Manuscript received March 1, 2014; revised September 7, 2014; accepted October 21, 2014. Date of publication November 20, 2014; date of current version October 21, 2015. This work was supported by the National Natural Science Foundation of China under Grant 61274031.

The authors are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: jerryyangs@gmail.com; caiyc@mail.tsinghua.edu.cn; zhouqiang@tsinghua.edu.cn; weakest3@gmail.com).

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

Digital Object Identifier 10.1109/TVLSI.2014.2365520

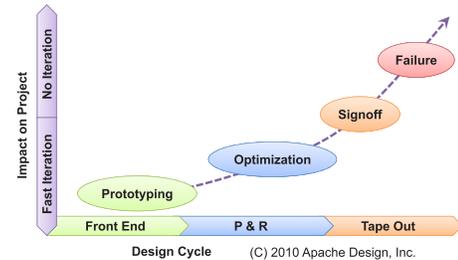


Fig. 1. Design challenge of billion-transistors VLSI design.

result in longer circuit delays and lead to functional failures. A conservative approach is to overdesign the grid while performing wire-sizing to make metal lines wide and reducing their pitch. However, many large chip designs at advanced nodes are becoming routing resources limited. Instead, there is a crucial need for EDA tools that can efficiently perform power grid verification to ensure that terminal voltages remain within design constraints, protecting the grid from the various performance and reliability concerns. Since the number of on-chip transistors is continuously increasing along with aggressive process technology scaling, the exploding complexity of corresponding on-chip power grids makes the analysis and verification very challenging in terms of excessive runtime and huge memory consumption.

Power grid verification is traditionally performed by simulation approach, which requires the information of the current excitations that represents the pattern of activity of the underlying circuit. Once the power grid is simulated and the voltage noise at every node is determined, it is easy to check the grid safety. Total safety check using this approach requires the simulation of a comprehensive set of stimulus patterns, which will be extremely time-consuming, and consequently impracticable to cover all working scenarios. To make things worse, it is extremely important to perform some type of grid verification during early design planning with uncertainty mode while the information on circuit workload/behaviors is often simply unknown early in the design flow. From point view of design cycles as shown in Fig. 1, it is easier to perform power grid prototyping and optimization in the early stage, but really difficult to perform design iterations when signoff. To overcome these issues, the vectorless verification is first proposed as a formal approach to verify power grid with uncertainty mode [1], and later many contributions are made to further improvements. It is based on the concept of current constraints to capture the uncertainty of the circuit behaviors.

The current constraints are a set of upper bounds on each current drawn and can be obtained from the knowledge of the overall power dissipation on each circuit block. Under these constraints, power grid verification becomes a problem for computing the worst case voltage fluctuations at each node subject to current constraints, which can cover all possible current waveforms. Such an approach is not computationally cheap, but it gives a good upper bound on the true worst case, and it holds the promise of leading to practical approaches of certain scenarios, especially for early design planning.

There are several further research topics among the recent works. On the aspect of power grid modeling method, dc model is first considered in [1], *RC* model is introduced to perform transient verification in [2], and *RLC* model is adopted to verify power grid circuits operating at high frequencies in [3]. On the aspect of current constraints, local constraints and global constraints are first proposed in [1], transient current constrains are introduced for practical transient noise predictions in [4], and hierarchical current and power constraints are adopted for more realistic verification in [5]. Due to the extremely high complexity of the verification problem, many contributions are made to reduce the problem size for efficient verification. They include several efficient sparse matrix inverse methods, such as sparse approximate inverse (SPAI) technique [6], AINV method [7], and \mathcal{H} Matrix approach [8]. Xiong and Wang [9] proposed a hierarchical matrix inversion algorithm to speed up the computation of each row of the inverse. Ghani and Najm [10] reduce the number of linear programming (LP) problems by examining dominance relations among node voltage drops and branch currents. In [11], a node elimination approach is utilized to systematically reduce the grid size and accurately compute the upper bounds while the node safety criteria are still guaranteed. In [12], model-order reduction was adopted to reduce the complexity of formulating LP problems. In [13] and [14], a convex dual algorithm was proposed to solve LP problems more efficiently. In [15], constraint abstraction was proposed for verification in a divide-and-conquer manner. Meanwhile, a PDE-constrained optimization method was proposed for scalable multilevel vectorless verification both on voltage integrity [16] and current integrity [17] issues. A selected inversion approach was proposed by exploiting locality in [18]. In [19], a maximum voltage drop location estimation approach was proposed for efficient vectorless verification.

Despite significant improvements among these approaches, it is still extremely expensive to verify a large-scale power grid by vectorless approach because the number of LP problems to be solved and the size of each LP problem are proportional to the size of the power grid. In this paper, by focusing on the power grids with flip-chip package, a novel selected inversion technique is proposed for efficient vectorless verification by exploiting grid locality and constraint locality. In this approach, the power grid is divided into several partitions by taking the advantage of locality effect across the power grid. To verify each node, it can capture the current sources that are most influential on this node, and subsequently it selectively computes the matrix inverse to formulate a reduced-size optimization problem while still

guaranteeing the quality of solution accuracy. Especially for verifying the practical designed grids, there usually exists complicated structures and irregular pads distribution, such as IBM power grid benchmarks [20]. Even though there have been many research works to take the advantages of grid locality characteristic [21]–[23], but all of them are based on the structured mesh or regular pads distribution. And yet there is still no developed methodology focused on how to measure the spatial locality and determine the grid shell for practical designed cases. In this paper, a concept of quasi-Poisson block is introduced to measure the locality effect and a scheme of pad-aware partitioning is proposed to ensure the selected inversion approach still available for practical use. Experimental results show that the proposed approach could achieve significant improvements in runtime compared with previous approaches.

The rest of this paper is organized as follows. A brief introduction to the problem formulation and the previous approaches are provided in Section II. The details of the proposed selected inversion technique are presented in Section III, and pad-aware partitioning scheme is provided in Section IV. Experimental results are illustrated in Section V, and concluding remarks are given in Section VI.

II. PRELIMINARIES

A. Vectorless Power Grid Verification

Power grid analysis is usually performed by Kirchoffs Law, which builds the *R/RC/RCL* network as linear systems. For clarity, only dc model is presented for static verification in this paper. In addition, without loss of generality, *RC* and *RLC* model can also be formulated for transient verification. Considering a n -node power grid with purely resistive model, static analysis can be formulated using traditional modified nodal analysis method as the following linear system of equations [24]:

$$\mathcal{G}\mathbf{v} = \mathbf{i} \quad (1)$$

where $\mathcal{G} \in \mathbb{R}^{n \times n}$ is the grid conductance matrix, $\mathbf{v} \in \mathbb{R}^{n \times 1}$ is the vector of node voltage, and $\mathbf{i} \in \mathbb{R}^{n \times 1}$ is the vector of current sources representing the underlying circuitry. By properly tackling the voltage sources, the above system can be reformulated as a revised system equation, which can be solved directly to obtain the voltage drop values [1]. From the revised system equation, the vector \mathbf{v} is to represent the voltage drops of the circuit nodes.

The vectorless verification approach is proposed for early verification on power grids when the details of the underlying circuitry may not be known. Current constraints provide a way to capture the uncertainty about circuit behavior, which can be obtained from the knowledge of the design specifications, such as chip area, power budget, and engineering judgments. Under this formulation of the grid, the verification problem becomes a linear optimization problem where the vector of voltage drops is maximized subject to the current constraints.

Following [1] and [9], we use two types of constraints: 1) local constraints and 2) global constraints. Local constraints are upper bounds on the individual current excitations and

represent the maximum current drawn by individual cells or blocks, which are connected to the grid nodes. They can be represented as

$$0 \leq \mathbf{i} \leq \mathbf{I}_L \quad (2)$$

where $\mathbf{I}_L \in \mathbb{R}^{n \times 1}$ is a vector of fixed current values. Global constraints are introduced to provide an upper bound on the sum of currents drawn by groups of current sources. They are typically chosen based on some knowledge of the peak power dissipation of a group of circuit blocks. Assuming there are m global constraints, and then can be expressed as matrix form

$$0 \leq \mathcal{U}\mathbf{i} \leq \mathbf{I}_G \quad (3)$$

where $\mathcal{U} \in \mathbb{R}^{m \times n}$ is a Boolean matrix that indicates which current sources are present in each global constraint, and $\mathbf{I}_G \in \mathbb{R}^{m \times 1}$ is a vector of upper bounds on the sum of currents included in each circuit block.

B. Problem Formulation

The vectorless verification is interested in finding the worst case voltage variations at all the nodes, that is, obtaining upper bounds on the maximum voltage drops of each node under all the possible current patterns that satisfy the local and global constraints. Thus, the verification problem has been formulated as an optimization problem of maximum voltage noise under linear current constraints in [13]. Consider a resistive power grid of n nodes with conductance matrix \mathcal{G} , and denote $A = \mathcal{G}$ for static analysis mode. Given local and global current constraints with parameters \mathbf{I}_L , \mathbf{I}_G , and \mathcal{U} , solve for each node $1 \leq j \leq n$

$$\begin{aligned} & \text{Maximize } v_j \\ & \text{s.t. } \mathbf{A}\mathbf{v} = \mathbf{i}, \quad 0 \leq \mathbf{i} \leq \mathbf{I}_L, \quad 0 \leq \mathcal{U}\mathbf{i} \leq \mathbf{I}_G \end{aligned} \quad (4)$$

where \mathbf{i} is the decision variables of current sources, \mathbf{v} is the vector of corresponding voltage noises, and v_j is the j th element of \mathbf{v} .

Since A is known to be an $n \times n$ symmetric positive definite \mathcal{M} matrix, it is invertible while A^{-1} is also symmetric and satisfies $A^{-1} \geq 0$. Subsequently, the optimization problem can be decomposed into two subproblems as follows [13]:

$$\begin{aligned} \text{I :} & \text{ Compute } \mathbf{c}_j \text{ by solving } \mathbf{A}\mathbf{x} = \mathbf{e}_j \\ \text{II :} & \text{ Maximize } v_j = \mathbf{c}_j^T \mathbf{i} \\ & \text{s.t. } 0 \leq \mathbf{i} \leq \mathbf{I}_L, \quad 0 \leq \mathcal{U}\mathbf{i} \leq \mathbf{I}_G \end{aligned} \quad (5)$$

where $\mathbf{c}_j \in \mathbb{R}^{n \times 1}$ is a vector of coefficients, and $\mathbf{e}_j \in \mathbb{R}^{n \times 1}$ is a Boolean vector with 0s except for its j th element being 1. The problem decomposition (5) provides a friendly verification framework [13]. All circuit nodes could be verified independently, and each node could be verified as two stages: 1) coefficients computation and 2) LP solving. For verifying each circuit node, it first computes the vector of coefficients (subproblem I) and then solves the involved LP problem, which adopts the above coefficients as its objective function (subproblem II). Computing \mathbf{c}_j is equivalent to picking up the j th column of A^{-1} , which can be obtained by solving linear equations shown in subproblem I. It can be

considered as the sensitivity of the j th node to the current sources attached to the circuit nodes across the grid. The subproblem I has been studied in other similar researches, such as equivalent resistance analysis between each two circuit nodes, and electric static discharge analysis [25]. For verifying each node, it requires to solve linear equations once and LP problem once while both of the problem size is n . The total complexity of full grid verification is proportional to the number of nodes in the grid, which is still very challenging or even prohibitive for practical use.

C. Previous Approaches

The primary goal is to reduce the problem size or improve the solving efficiency of subproblem I and subproblem II. SPAI technique is utilized to compute an approximate \mathbf{c}_j with a small number of nonzero entries using least square methods, and then the involved LP problem can be much simplified and solved efficiently [6]. In [13] and [14], convex dual algorithms are adopted to solve the LP problem more efficiently, and a preconditioned conjugate gradient (PCG) method is utilized to solve subproblem I. But as we have observed, the used SPAI technique and PCG method are still relatively inefficient in verifying large-scale power grids. Since it is obvious that direct linear solvers are more efficient than iterative solvers for transient simulation, such as Cholmod [26], we realize that direct solvers should also be more efficient when utilized to solve subproblem I. Consequently, it is of great necessity to investigate novel approaches for more potential improvements in large-scale verification.

III. SELECTED INVERSION

The idea of selected inversion originally arose from several scientific applications, which need to calculate a subset of the entries of the inverse for a given matrix. Examples include examining inverse covariance matrices in uncertainty quantification [27], finding a rational approximation for Fermi–Dirac functions in the electron density function theory [28], and so on. From a computational viewpoint, it is natural to develop algorithms for selected inversion that are faster than inverting the whole matrix. In particular, for some model problems, when A is obtained from a finite difference discretization of a Laplacian operator or from lattice models in statistical or quantum mechanics with a local Hamiltonian [29], the specified diagonal entries of the matrix inverse can be efficiently extracted by selected inversion. Similarly, it is of great interest to take the advantage of selected inversion for verifying power grids because it is also formulated by Laplacian discretization. In the aspect of circuit simulation, a selective inversion approach is proposed for inverting the inductance matrix of large-scale sparse *RLC* network [30]. In this section, a selected inversion implemented on block level is demonstrated for efficient vectorless power grid verification.

A. Methodology

The standard approach for computing A^{-1} is to first perform Cholesky factorization $A = \mathcal{L}\mathcal{L}^T$, where \mathcal{L} is a lower

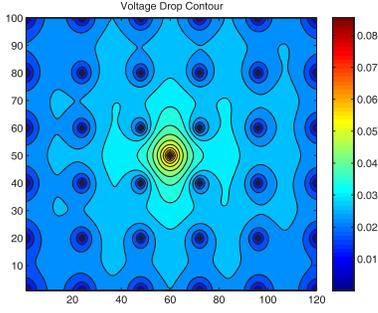


Fig. 2. Locality effect on flip-chip (a strong current source attached to the center node of a sample 100×120 power grid with 6×6 pads array).

triangular matrix. Subsequently, $A^{-1} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ can be obtained by solving a number of triangular systems

$$\mathcal{L}\mathbf{y} = \mathbf{e}_j, \quad \mathcal{L}^T \mathbf{x}_j = \mathbf{y} \quad (6)$$

for $j = 1, 2, \dots, n$, where \mathbf{e}_j is the j th column of the identity matrix. The computational cost of such direct inversion is generally $\mathcal{O}(n^3)$, with n being the dimension of A .

However, when A is sparse symmetric positive and definite, we can exploit the sparsity structure of A^{-1} to perform selected inversion on block level by exploiting grid locality. The inverse of a nonsingular sparse matrix is dense, especially for a matrix that results from a mesh structure; its inverse is almost full. However, the values of the entries among the inverse are observed to decay exponentially as one moves away from the diagonal when A is diagonally dominant and symmetric positive definite. This phenomenon can be confirmed from the sensitivity analysis among the grid. In addition, it can be utilized to construct a specific sparsity pattern for approximate representation. Especially for the power grid with flip-chip package (also known as controlled collapse chip connection or its acronym C4), the locality effect [21] can preserve this sparsity pattern to be more significant.

Locality means that the absorbing current sources can only trigger voltage drop within a local area around them. And beyond this area that is called grid shell in [21], the triggered voltage drop will attenuate to zero very fast. That is, for such a grid node, its voltage drop sensitivity is mainly influenced by the current sources attached to the nodes within the grid shell. Take a small grid with flip-chip package as an example, a strong current source is attached to the center node of a 100×120 grid with uniformly distributed 6×6 pads array, while small current sources are attached to other nodes. Fig. 2 shows the voltage drop distribution triggered by this strong current source. The area that contains dominant voltage drop is around the center part of the grid. Meanwhile, the voltage drop decreases dramatically with the increasing distance from the center node. Especially for the area outside the four nearest pads array, the voltage drop is affected slightly by the strong current source. And for the area outside the second nearest pads array, the voltage drop is even much smaller. Hence, the area bounded by the nearest pads array can be considered as a local area to exploit sparsity representation.

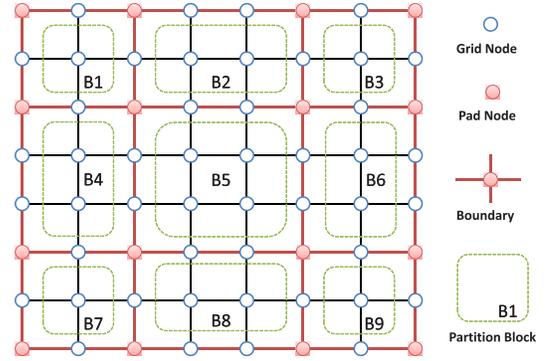


Fig. 3. Power grid partitioning.

Considering a structured power grid of flip-chip package with uniform pads distribution, its conductance matrix A can be reformulated as a domain-decomposition form. As shown in Fig. 3, the power grid is naturally partitioned into several subgrids according to the uniform pad distribution (for nonuniform distribution of realistic cases, considering pad-aware partitioning scheme, which will be demonstrated in Section IV). For each subgrid, the nodes inside the subgrid are denoted as internal nodes, and the nodes on the subgrid boundaries are referred as interface nodes or global nodes. Assuming through possible row and column permutations that the full grid is partitioned into p subgrids $\{\Omega_k\}$ for $k = 1, 2, \dots, p$, and then A has the following form:

$$A = \begin{pmatrix} B_1 & & & F_1 \\ & B_2 & & F_2 \\ & & \ddots & \vdots \\ & & & B_p & F_p \\ F_1^T & F_2^T & \dots & F_p^T & G \end{pmatrix} := \begin{pmatrix} B & F \\ F^T & G \end{pmatrix}$$

where $B_k \in \mathbb{R}^{n_k \times n_k}$ is the conductance matrix of k th subgrid Ω_k , $F_k \in \mathbb{R}^{n_k \times n_G}$ is a negative matrix representing the conductance links between internal nodes and global interface nodes, $G \in \mathbb{R}^{n_G \times n_G}$ is the conductance matrix representing the relationship between all global interface nodes, for $k = 1, 2, \dots, p$, and $n_G + n_B = n$ with $n_B := \sum_{k=1}^p n_k$, while B_k and F_k are denoted as

$$B := \text{diag}(B_1, B_2, \dots, B_p) \\ F^T := (F_1^T, F_2^T, \dots, F_p^T)$$

for intuitive representation. Each F_k represents a set of connections from the internal nodes of subgrid k to its global interface nodes. Define n_{F_k} as the number of nonzero columns of F_k , consequently it is determined by the number of these global interface nodes that are adjacent to subgrid Ω_k , and it satisfies $\sum_{k=1}^p n_{F_k} \ll n_G$ for a relatively small p .

Before we present the selected inversion approach, it will be helpful to first review the major operations involved in the block level \mathcal{LDL}^T factorization

$$A = \begin{pmatrix} I & \\ H^T & I \end{pmatrix} \begin{pmatrix} B & \\ & G - F^T B^{-1} F \end{pmatrix} \begin{pmatrix} I & H \\ & I \end{pmatrix}$$

where $H := B^{-1}F$, and $S := G - F^T B^{-1}F = G - F^T H$ is known as the Schur complement. Taking the inverse of the above matrix yields

$$A^{-1} = \begin{pmatrix} B^{-1} + B^{-1}FS^{-1}F^TB^{-1} & -B^{-1}FS^{-1} \\ -S^{-1}F^TB^{-1} & S^{-1} \end{pmatrix}$$

$$A^{-1} := \begin{pmatrix} B^{-1} + HS^{-1}H^T & -HS^{-1} \\ -S^{-1}H^T & S^{-1} \end{pmatrix}. \quad (7)$$

Since B is a block diagonal matrix, $H \in \mathbb{R}^{n_B \times n_G}$ can be obtained by solving p sequences of n_G linear systems

$$B_k H_k = F_k, \quad k = 1, 2, \dots, p \quad (8)$$

where $H^T := (H_1^T, H_2^T, \dots, H_p^T)$ and $H_k \in \mathbb{R}^{n_k \times n_G}$. Notice that each H_k can be interpreted as a transformed influence of the local inverse B_k^{-1} on the global interface nodes, and has at least n_{F_k} nonzero columns. Accordingly, the number of linear equations to be solved in (8) can be reduced from n_G to n_{F_k} for each k .

The global Schur complement S can be computed by

$$S = G - \sum_{k=1}^p F_k^T H_k \quad (9)$$

where each term of the sum can be regarded as a local contribution, and the sparsity of F_k can be further exploited. If the dimension n_G of Schur complement is not relatively large, the directly computing S^{-1} can be efficient because it is known that S can be generally well-approximated by a sparse matrix.

After that, the term $HS^{-1}H^T$ in the upper-diagonal block of (7) can be obtained by representing it as

$$A^{-1} := \begin{pmatrix} B^{-1} + HL & -U \\ -L & S^{-1} \end{pmatrix} \quad (10)$$

where $U = L^T$, and thus it requires to solve a sequence of n_B linear equations of the form $SL = H^T$ with $L \in \mathbb{R}^{n_G \times n_B}$, which can be carried out locally by distributing S to each subgrid and solving

$$SL_k = H_k^T, \quad k = 1, 2, \dots, p \quad (11)$$

by a direct method when n_G is not relatively large, where $L^T := (L_1^T, L_2^T, \dots, L_p^T)$ and $L_k \in \mathbb{R}^{n_G \times n_k}$. Subsequently, the resulted HL can be similarly obtained by locally computing $H_k L_k$ for $k = 1, 2, \dots, p$.

B. Grid Locality and Constraint Locality

If the number of global nodes n_G is relatively small, S^{-1} can be formed explicitly and the sparsity can be considered to perform a well-approximate inverse. Notice that the term $B^{-1} + HS^{-1}H^T$ in the upper-diagonal block in (7) dominates the most dimensions of A^{-1} , while B^{-1} can be obtained by inverting the conductance of each subgrid. The internal details of $HS^{-1}H^T$ can be represented as

$$HS^{-1}H^T = \begin{pmatrix} H_1 S^{-1} H_1^T & H_1 S^{-1} H_2^T & \dots & H_1 S^{-1} H_p^T \\ H_2 S^{-1} H_1^T & H_2 S^{-1} H_2^T & \dots & H_2 S^{-1} H_p^T \\ \vdots & \vdots & \ddots & \vdots \\ H_p S^{-1} H_1^T & H_p S^{-1} H_2^T & \dots & H_p S^{-1} H_p^T \end{pmatrix}$$

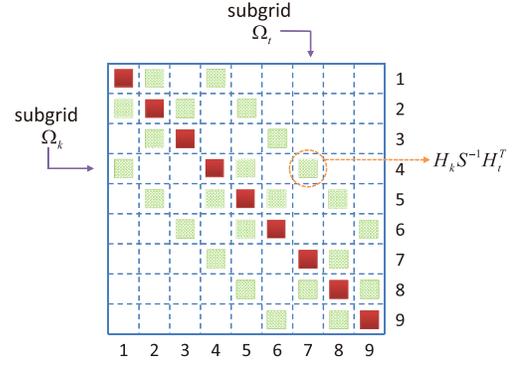


Fig. 4. Entries in $HS^{-1}H^T$.

where each block $H_k S^{-1} H_t^T$ can indicate the relationship between subgrid Ω_k and Ω_t for $1 \leq k, t \leq p$. In addition, the entries in this term dominate the total number of the entries of the full inverse. But as we have observed, the entries of most of the blocks in this term are significantly small. This phenomenon has confirmed the existence of the locality effect among the grid. Taking a power grid with nine partitions as an example, the entries of $HS^{-1}H^T$ can be shown in Fig. 4. The corresponding entries representing subgrid Ω_k and subgrid Ω_t is known as $H_k S^{-1} H_t^T$. If subgrid Ω_k is far enough away from subgrid Ω_t , the influence between them is significantly small, which is called grid locality and can be exploited for selected inversion. As shown in Fig. 4, if only the influence between the neighbored subgrids is considered, the entries $H_k S^{-1} H_t^T$ of corresponding blocks are kept, otherwise there is no need to figure them out.

Even though the values of ignored entries are significantly small, the total number of ignored entries may be very large. In general, their cumulative effects cannot be simply ignored and subsequently more considerations should be taken to improve the verification accuracy. From the viewpoint of verification, the problem can be expressed as

$$\begin{pmatrix} B & F \\ F^T & G \end{pmatrix} \begin{pmatrix} \mathbf{v}_{\text{inner}} \\ \mathbf{v}_{\text{inter}} \end{pmatrix} = \begin{pmatrix} \mathbf{i}_{\text{inner}} \\ \mathbf{i}_{\text{inter}} \end{pmatrix} \quad (12)$$

where $\mathbf{v}_{\text{inner}}$ and $\mathbf{v}_{\text{inter}}$ are the grid nodes to be verified for internal nodes of each subgrid and global interface nodes, respectively; and $\mathbf{i}_{\text{inner}}$ and $\mathbf{i}_{\text{inter}}$ are the current sources, which are attached to internal nodes and interface nodes, respectively. By employing the represented inverse in (7), the verification problem can be reformulated as

$$\begin{pmatrix} \mathbf{v}_{\text{inner}} \\ \mathbf{v}_{\text{inter}} \end{pmatrix} = \begin{pmatrix} B^{-1} + HS^{-1}H^T & -HS^{-1} \\ -S^{-1}H^T & S^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{i}_{\text{inner}} \\ \mathbf{i}_{\text{inter}} \end{pmatrix}$$

that is, $\mathbf{v}_{\text{inner}}$ and $\mathbf{v}_{\text{inter}}$ can be verified independently by

$$\begin{aligned} \mathbf{v}_{\text{inner}} &= B^{-1} \mathbf{i}_{\text{inner}} + HS^{-1}H^T \mathbf{i}_{\text{inner}} - HS^{-1} \mathbf{i}_{\text{inter}} \\ \mathbf{v}_{\text{inter}} &= -S^{-1}H^T \mathbf{i}_{\text{inner}} + S^{-1} \mathbf{i}_{\text{inter}}. \end{aligned} \quad (13)$$

It can be seen from (13) that, the interface nodes can be verified accurately if the sparsity patterns of S^{-1} and H^T are not exploited.

Since there are many entries of blocks in $HS^{-1}H^T$ that are not kept due to the grid locality, it will result in significant errors when verifying the voltage noise. In this paper, a concept of constraint locality is proposed to take this phenomenon

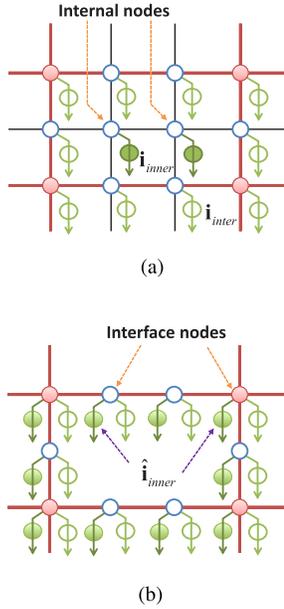


Fig. 5. Constraint locality demonstration. (a) Internal nodes and interface nodes under the global constraints. (b) Current sinks attached into the internal nodes are moved to the interface nodes around them.

into account for efficient verification. That is, the sensitivity of voltage drop has a locality characteristic for global constraint. To verify the internal nodes in subgrid Ω_k , consider if the subgrid Ω_t is far away enough from it, the corresponding entries $H_k S^{-1} H_t^T$ will be ignored due to the grid locality. Aiming to improve the verification accuracy, the current sources attached to the internal nodes of subgrid Ω_t are considered to be removed and attached on the interface nodes around them. The constraint locality is that, regardless of whether the current sources are attached to the internal nodes in subgrid Ω_t or attached to the interface nodes around them, the influence of them to verify subgrid Ω_k is almost the same. As shown in Fig. 5, the current sinks \mathbf{i}_{inner} attached to the internal nodes are properly moved to the interface nodes around them. The most exact way to mapping them to the interface nodes is to compute $H_k S^{-1} H_t^T$, but as we have observed, it should be sufficient to roughly assign them as $\hat{\mathbf{i}}_{inner}$ on average since the constraint locality can preserve the verification accuracy.

Considering the expression for voltages \mathbf{v}_k of inner nodes in partition Ω_k , we have

$$\mathbf{v}_k = B_k^{-1} \mathbf{i}_k + \sum_{j=1}^p H_k S^{-1} H_j^T \mathbf{i}_j - H_k S^{-1} \mathbf{i}_g \quad (14)$$

where \mathbf{i}_g is the currents at interface nodes, \mathbf{i}_k and \mathbf{i}_j are the inner currents in block Ω_k and Ω_j , respectively. It can be seen from (14), the second term for $j = t$ can be skipped and we can still compute exactly the same \mathbf{v}_k by adjusting \mathbf{i}_g with $-H_t^T \mathbf{i}_t$. This is equivalent to moving every internal current of Ω_t to its interface nodes in proper ratios determined by $-H_t^T$. When Ω_t is far off from Ω_k , the internal currents of Ω_t can be distributed in some approximate fashion (e.g., equally among all its interface nodes), without incurring much error.

Algorithm 1: Selected Inversion

```

1 Partition blocks and determine  $\{B_k\}$ ,  $\{F_k\}$  and  $G$ ;
2 Set  $S := G$ ;
3 for  $k = 1, 2, \dots, p$  do
4   Compute  $D_k := B_k^{-1}$ ;
5   Compute  $H_k := B_k^{-1} F_k$ ;
6   Update  $S := S - F_k^T H_k$ ;
7 end
8 Compute  $S^{-1}$ ;
9 for  $k = 1, 2, \dots, p$  and  $t = 1, 2, \dots, p$  do
10  if  $\langle k | t \rangle$  then
11    Compute  $D_k := B_k^{-1}$ ;
12    Compute  $U_k := H_k S^{-1}$  and  $L_k := U_k^T$ ;
13    Compute  $X_{kt} := U_k H_t^T$ ;
14    Update  $D_k := D_k + X_{kt}$ ;
15  end
16 end
17 Set  $A^{-1} := \begin{pmatrix} B^{-1} + X & -U \\ -L & S^{-1} \end{pmatrix}$ 

```

C. Implementation

Based on the grid locality and constraint locality, the resulted selected inversion technique is implemented, as shown in Algorithm 1. The major computation cost of selected inversion is first to perform inversion for each subgrid, and subsequently to compute the resulted H , U , and X by matrix multiplication. However, their sparsity pattern should be carefully further exploited with a proper threshold for efficient implementation. The term $\langle k | t \rangle$ in line 10 is to decide whether the corresponding entries of $H_k S^{-1} H_t^T$ between subgrid Ω_k and subgrid Ω_t are computed or ignored. Define a parameter *sense_level* to control the depth number of neighbored subgrids to be considered, that is, if $abs(k - t)$ is no more than *sense_level*, then $H_k S^{-1} H_t^T$ cannot be ignored, otherwise ignored.

The divide and conquer strategy efficiently reduces the problem size by each subgrid. The resulted inversion tasks are much easier to be finished when compared with inverting the whole matrix once. It is worth mentioning that the required peak memory is usually very high, if the matrices inverse for all subgrids is explicitly computed at the same time. Hence, there is actually no need to explicitly figure out all of the entries for each subgrid together and then solve the involved LP problems. A preferable approach is to first verify the global interface nodes, and subsequently to verify each subgrid one-by-one, that is, to compute the entries of the corresponding inverse for each subgrid and then solve the involved LP problems. Such ordering approach could significantly reduce the peak memory footprint in Algorithm 1.

D. Complexity Analysis

Similar to *SellInv* in [28], we also present the computation advantage of the proposed selected inversion technique compared with the direct inversion method. Without loss of generality, suppose for a power grid with $n \times n$ mesh, the total

number of grid nodes is $N = n \times n$. In addition, suppose that an $m \times m$ pads array is uniformly distributed throughout the power grid, where $m \ll n$ and denote $M = m \times m$. It is easy to know that the power grid can be partitioned into $p = (m - 1)^2$ subgrids. The number of global interface nodes is $(2mn - m^2)$, that is, the matrix dimension of the Schur complement satisfies $S \in \mathbb{R}^{(2mn-m^2) \times (2mn-m^2)}$. In addition, the matrix dimension of each subgrid is $(n^2 - (2mn - m^2))/p = (n - m)^2/p = ((n - m/m - 1))^2$.

Suppose that the complexity of computing the Cholesky decomposition is $\mathcal{O}(N^3)$, and substitution of the triangular matrix is $\mathcal{O}(N^2)$, the total complexity of directly computing the inverse is about $\mathcal{O}(N^3)$. For inversion of all subgrids, the total complexity of Cholesky factorization is about $(n - m)^6/p^2 \propto N^3/p^2$, and the total complexity of substitution is about $(n - m)^4/p \propto N^2/p$. For computing the inversion of Schur complement, the total complexity of Cholesky factorization is about $(2mn - m^2)^3 \propto (MN)^{1.5}$, and the total complexity of substitution is about $(2mn - m^2)^2 \propto MN$. The total computation cost of matrix multiplication in step 5, step 6, step 12, and step 13 can be presented as

$$\left[\frac{(n - m)^4(2mn - m^2)}{p} + (n - m)^2(2mn - m^2)^2 \right] \propto \frac{N^{2.5}}{p}.$$

It can be observed that, the major computation cost of the selected inversion is to perform matrix inversion of each subgrid. Consequently, the total complexity of selected inversion can be roughly estimated as N^3/p^2 . In practice, the sparsity pattern of the matrix B_k , F_k , H_k , and U_k has been further utilized for efficient implementation to make the actual computation costs to be less than such theoretical bounds. Accordingly, the complexity of direct inversion still remains greater than selected approach. Consequently, the proposed selected inversion will be much more efficient than the direct approach when the grid locality and constraint locality can be properly exploited.

IV. LOCALITY-DRIVEN GRID PARTITIONING

For structured power grids with flip-chip package, the demonstrated selected inversion methodology can be easily utilized by a natural partitioning strategy. However, it is difficult to exploit the grid locality for practical designed cases where exist irregular pads distribution, such as IBM power grid benchmarks in [20]. There have been many research works to take the advantages of grid locality characteristic [21]–[23], [31]–[34], but all of them are based on the structured mesh or regular pads distribution. And yet there is still no reported works focused on how to measure the spatial locality and determine the grid shell for practical designed cases. In this section, we will introduce a concept of quasi-Poisson block and also demonstrate a novel pad-aware partitioning approach for exploiting the grid locality on practical designed cases.

A. Quasi-Poisson Block

Chiprout [21] first proposes the concept of grid shell for large-scale power grid with flip-chip package. The whole

grid is partitioned as many subgrids according to grid shell, which is not accurate but possible to be exploited for parallel power grid analysis. However, no obvious rule is proposed to determine the grid shell in [21].

A concept of Poisson block was originally proposed in [23], to explore the locality characteristic according to the particularity of structured power grid. Locality means that the absorbing current sources only can trigger voltage drop within a local area around it. And beyond this area, the triggered voltage drop will attenuate to zero very fast. The Poisson block is based on several definitions and assumptions, which can be easily satisfied by modern industrial chip designs. A Poisson block stands for a regular grid area, which satisfies three conditions: 1) this area does not contain any pads; 2) this area contains only current sources as active elements; and 3) the current flowing through the boundary is small enough.

For flip-chip package model with locality property, pads are distributed evenly in the grid and they act as strong current drain points, which prevent the current from flowing far away from its source point. Thus, the local area among nearest pads in power grid with flip-chip package can be treated as Poisson block. Consequently, the partitioned subgrids in Section III could be regarded as a kind of Poisson block. But in general, the supply pads are not distributed uniformly while there are often some missing pads somewhere or additional pads placed elsewhere. Nevertheless, we still believe that some characteristic advantages could be taken for grid partitioning according to locality. Similar to the concept of Poisson block, an idea of quasi-Poisson block is introduced in this paper to measure the locality effect under the following assumptions.

- 1) Unlike the definition of a Poisson block, the area of a quasi-Poisson block could be rectangle or nonrectangle, that is, a polygon area (no matter whether of convex or not) could be still regarded as a quasi-Poisson block.
- 2) The boundary of quasi-Poisson blocks should be capable of preventing the current flow between neighboring quasi-Poisson blocks, that is, the neighboring quasi-Poisson blocks have the minimal coupling with each other.
- 3) The supply pads are expected to appear on the boundary of each quasi-Poisson blocks so that the coupling between neighboring quasi-Poisson blocks can be minimized by supply pads between them.

Under these definitions, it is supposed to perform the power grid partitioning according to the supply pads distribution. This partitioning strategy is named as pad-aware partitioning, which will be detailedly discussed in Section IV-B.

B. Pad-Aware Partitioning

Pad-aware partitioning approach is to partition the power grid into quasi-Poisson blocks based on the supply pads distribution. It first determines the boundaries of all quasi-Poisson blocks, and then extracts the grid nodes corresponding to each quasi-Poisson block as one subgrid.

From the perspective of minimizing the electrical coupling between neighboring quasi-Poisson blocks, we have observed they have a kind of important relationship with the relative neighborhood graph (RNG) in computational geometry.

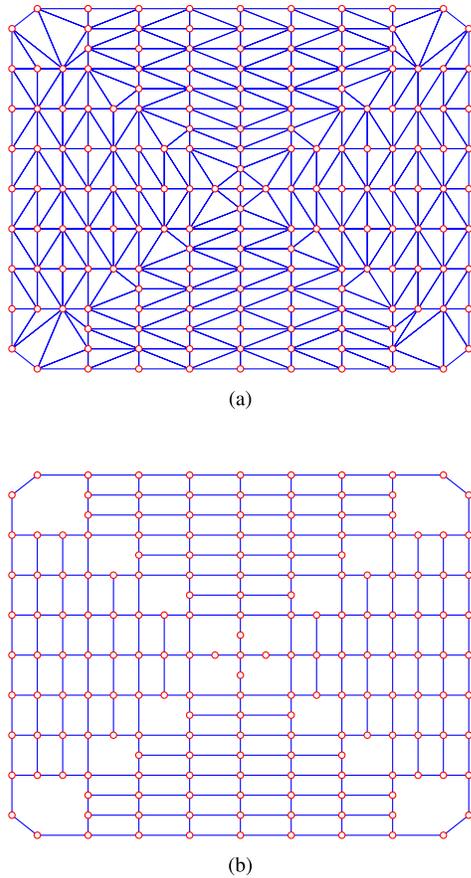


Fig. 6. Example of Delaunay triangulation and Urquhart graph. (a) Delaunay triangulation results of the pad distribution (177 pads in GND net of *ibmpg1*, denoted as red circles). (b) Urquhart graph (removing the longest edge from each triangle in the Delaunay triangulation, and regarding the enclosed rectangles or polygons as quasi-Poisson blocks).

RNG is an undirected graph defined on a set of points in the Euclidean plane by connecting two points p and q by an edge whenever there does not exist a third point r that is closer to both p and q than they are to each other. This graph was proposed as a way of defining a structure from a set of points that would match human perceptions of the shape of the set [35]. The RNG could be computed in linear time from the Delaunay triangulation of the point set since it is a subgraph of the Delaunay triangulation. In addition, actually the Urquhart graph, the graph formed by removing the longest edge from every triangle in the Delaunay triangulation, was originally proposed as a fast method to compute the RNG [36]. Although the Urquhart graph sometimes differs from the RNG, it can be used as an approximation to the RNG [37].

Taking the GND net of *ibmpg1* as an example and regarding its pads distribution as a set of points in the Euclidean plane, the triangulation results are shown in Fig. 6(a), where the red circles denote the pads location and blue segments are the edges of triangles. Its Urquhart graph is shown in Fig. 6(b), where the longest edge of each triangle has been removed from the Delaunay triangulation. From the Urquhart graph view, there exist many enclosed rectangles or polygons, which can be viewed as natural quasi-Poisson blocks and especially there are four larger quasi-Poisson blocks around the four corners.

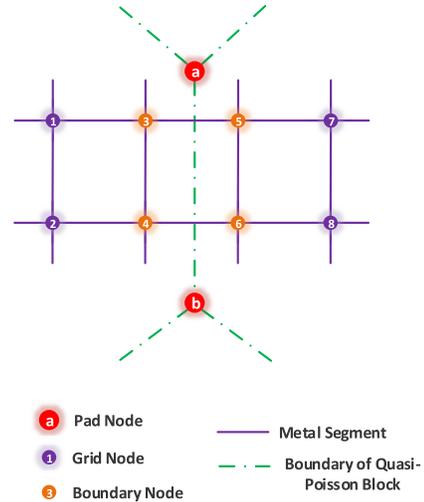


Fig. 7. Example of boundary edges between quasi-Poisson blocks.

Noticed that numerous quasi-Poisson blocks maybe exist but many of them are very small. In particular, when the required number of partitions is less than the number of obtained quasi-Poisson blocks we need to form several or some blocks as one partition. Although it can be accomplished by a greedy strategy with the consideration of balancing the partition size, we still prefer to perform the partitioning on the power grid nodes according to a general approach, such as the graph partitioning software METIS [38]. Consequently, the primary target is to partition the power grid using METIS while taking the boundaries of quasi-Poisson blocks into consideration. Therefore, the partitioning result should satisfy the assumption (3) from the definition of quasi-Poisson block, that is, each partition is expected to be surrounded by supply pads (some supply pads may exist inside the partitions). As demonstrated by the concept of quasi-Poisson block, the metal segments intersected with boundary edges are observed to be as weak coupling between neighboring quasi-Poisson blocks, and should be easier to be cut when performing METIS partition. This can be easily implemented by assigning a relatively small edge weight incident on each metal segment, which intersects with the boundary edges of quasi-Poisson blocks. As the conductance values reflect the coupling degree, small conductance values are assigned for these metal segments in edge weighting.

As shown in Fig. 7, the dashed lines between node a and b is the boundary edge of quasi-Poisson blocks from left and right. There are usually several metal segments intersected with the boundary edge, such as the metal segment from node 3 to node 5 and the metal segment from node 4 to node 6. Accordingly, the segments 3–5 and 4–6 are assigned as a relatively small edge weight. Since the partition algorithm METIS attempts to minimize the total cut weights on the boundary, those weighted metal segments have a very high possibility to be cut. Similar idea has been exploited in noise-aware partitioning for decap budgeting and minimization in [39].

The partitioning of quasi-Poisson blocks is aiming at determining the boundary nodes for forming the Schur complement,

and collecting the internal nodes for forming the matrices of subgrids, as shown in Fig. 3. Since the boundary nodes are usually not exactly on the obtained boundary edges of quasi-Poisson blocks (such as the grid node 3, 4, 5, and 6 are not on dashed lines $a-b$), these nodes adjacent to the cut edges are regarded as the boundary nodes to forming the Schur complement.

Another issue is that we should identify the adjacency relationship among the obtained partitions since the selected inversion implementation requires the parameter *sense_level* in Section III-C. This can be accomplished by construct a dual graph corresponding to the obtained partitions. Consequently, the neighborhood partitions to be considered in Algorithm 1 could be determined by searching the dual graph according to the parameter *sense_level*.

In summary, the pad-aware partitioning scheme is combined with selected inversion algorithm for comprehensive verification flow as listed below.

- 1) Setup phase:
 - a) Parse the netlist and extract the circuit information into built-in data structure.
 - b) Construct the circuit topology graph.
 - c) Pad-aware partitioning:
 - i) Delaunay triangulation on the pads distribution map.
 - ii) Build the Urquhart graph by removing the longest edge from each triangle.
 - iii) Determine the quasi-Poisson blocks and their boundary edges.
 - iv) Modify the weight of metal segments intersect with the boundary edges and perform grid partitioning with METIS.
 - v) Collect the grid nodes for each partitions and identify the boundary nodes.
- 2) Verification phase:
 - a) Selected inversion.
 - b) LP optimization.

In addition, the pad-aware partitioning scheme can be further employed for other locality-driven methodologies, such as partitioning-based parallel simulation, decap budgeting and optimization, and placement optimization of power supply pads.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

Various experiments are carried out to validate the proposing performance of the proposed selected inversion technique. Hardware platform is a 64-bit Linux server with Intel Xeon E5-2650 CPU @ 2 GHz and 128-GB RAM. Note that although the processor has multiple cores, only single core is utilized for evaluation. Both synthetic power grids and IBM power grid benchmarks [20] are evaluated for performance comparison. Synthetic power grids are generated according to the typical electrical parameters from industrial designs [20]. The voltage suppliers/pads array is uniformly distributed throughout the power grids (flip-chip package), while the entire grid nodes are connected to current sources. Local current constraints are

assigned according to the typical value from industrial power grids, and global current constraints are generated by scaling down the total amount of current drawn by groups of current sources. For each power grid, we specify nine global constraints for synthetic benchmarks and four global constraints for IBM power grid benchmarks in our experiments. Since IBM power grids have multiple networks, we prefer to verify a single network of each benchmark while their GND net are naturally separated with other networks and can be verified separately.

The proposed selected inversion technique has been implemented by MATLAB scripts for evaluation. The direct inversion approach is to compute an accurate inverse to obtain the exact upper bounds of voltage noises. The popular direct solver Choldmod [26] is employed to solve all of the involved linear equations through Cholesky factorization. And for selected inversion approach, a threshold of 10^{-3} is chosen to drop the extremely small entries of the resulted solution for exploiting its sparsity pattern. Gurobi optimizer [40] is invoked to solve all of the involved LP problems for verifying the voltage noises. It should be noticed that a significant reduction in computation cost can be achieved by keeping the returned *vbasis* and *cbasis* after each LP optimization and then pass them to the next LP optimization as an advanced warm start. Since the constraints information of each LP model is the same as others, the variable basis and constraint basis could be reused for verifying different nodes while it is possible to simply update the objective coefficients.

For evaluation on IBM power grid benchmarks, additional preparation works are conducted due to their unstructured characteristics and irregular pads distribution. Each SPICE netlist is parsed by the approach in [41]. For pad-aware partitioning of each power grid, Delaunay triangulation is first performed on the pad distribution by Qhull [42]. In addition, we simply construct the Urquhart graph to identify the quasi-Poisson blocks because we observed that the results of Urquhart graph are same as the RNG, which is obtained by NGL Library [43]. After modifying the weights of metal segments intersect with boundary edges, the resulted power grid is partitioned with METIS package [38].

B. Performance Results

For evaluation on synthetic benchmarks, the comprehensive performance results of the proposed selected inversion are listed in Table I when compared with the direct inversion approach. N is the number of synthetic grid nodes. M is the number of voltage suppliers/pads. The runtime is separately reported by inversion time and LP time for comprehensive comparison. T_{inv}^d and T_{LP}^d are the runtime of direct inversion and involved LP optimization, respectively. T_{inv}^s and T_{LP}^s are the runtime of selected inversion and involved LP optimization, respectively. Since the verification accuracy is decided by *sense_level*, more considerations should be taken for the implementations of selected inversion. As the power supply pads behave as the strong current sources, which could be regarded as good decoupling measurement, *sense_level* is observed to depend on the subgrid size and the number

TABLE I

PERFORMANCE RESULTS OF THE PROPOSED SELECTED INVERSION COMPARED WITH THE DIRECT INVERSION APPROACH FOR VERIFICATION ON SYNTHETIC BENCHMARKS. THE RUNTIME IS ILLUSTRATED IN SECOND. THE VOLTAGE DROP IS ILLUSTRATED IN MILLIVOLT

Benchmark			Direct Inversion			Selected Inversion				Speedup	
N	M	P	T_{inv}^d	T_{LP}^d	v_{max}	T_{inv}^s	T_{LP}^s	E_{max}	E_{avg}	T_{inv}^d/T_{inv}^s	T_{LP}^d/T_{LP}^s
7200	12	6	34.49	9.73	50.41	1.64	10.55	0.67	0.08	21	1
27000	25	16	688.94	142.78	49.71	26.23	79.65	3.15	1.16	26	2
34200	42	30	898.42	260.75	46.71	42.15	78.64	3.07	1.36	21	3
50600	56	42	2157.10	721.33	48.46	94.70	104.38	4.61	2.49	23	7
90000	100	81	6040.58	4003.01	47.18	329.73	115.18	2.19	1.50	18	35
140000	144	121	17954.62	14903.61	46.12	881.09	175.31	5.47	2.73	20	85
560000	484	441	307204.93	3963075.20	45.10	17162.19	858.86	4.43	1.07	18	4614

TABLE II

PERFORMANCE RESULTS OF THE PROPOSED SELECTED INVERSION COMPARED WITH THE HIERARCHICAL INVERSION APPROACH [9]. THE RUNTIME IS ILLUSTRATED IN SECOND

Selected Inversion			Hierarchical Inversion [9]					Speedup	
N	$T_{inv,perK}^s$	$T_{LP,perK}^s$	\hat{N}	T_{inv}^h	T_{LP}^h	$T_{inv,perK}^h$	$T_{LP,perK}^h$	$T_{inv,perK}^h/T_{inv,perK}^s$	$T_{LP,perK}^h/T_{LP,perK}^s$
7200	0.23	1.47	5875	10.31	9.91	1.75	1.69	8	1
27000	0.97	2.95	22939	123.60	106.20	5.39	4.63	6	2
34200	1.23	2.30	35668	304.20	233.40	8.53	6.54	7	3
50600	1.87	2.06	51195	636.60	495.00	12.43	9.67	7	5
90000	3.66	1.28	90643	2062.80	1573.20	22.76	17.36	6	14
140000	6.29	1.25	141283	5184.00	3816.00	36.69	27.01	6	22
560000	30.65	1.53	562363	103680.00	69120.00	184.36	122.91	6	80

of power supply pads of each subgrid. However, there are still not universal values for all power grid types. Users could choose a proper *sense_level* according to the previous experience. As we have observed, it is accurate enough to set *sense_level* as 2 for the synthetic benchmarks. Meanwhile, the drop tolerance is chosen as 10^{-3} for sparsity pattern exploiting which is also accurate enough. The subproblem II in (5) is to compute the upper bounds and lower bounds by solving LP problems. However, only the upper bounds (maximum voltage drop) will be listed and discussed since we usually pay more attention to the so-called upper bounds in worst case analysis. v_{max} is the exact solution of maximum voltage drop across the grid. This golden solution is computed by the direct inversion approach and listed in column 6. E_{max} and E_{avg} are the maximum solution error and the average solution error, respectively, between the above two approaches. As shown in Table I, the solution errors of selected inversion approach are about several millivolt, which can be adopted for practical verification. The proposed selected inversion approach can improve the runtime efficiency both on matrix inversion and LP problem solving. For computing the matrix inversion, it can achieve about $20\times$ speedups compared with the direct inversion approach, while both of them use Cholmod [26] as internal linear solver. And for LP problem solving, it can achieve about tens or hundreds of speedups since the computed coefficients are much sparser while the solution accuracy is still guaranteed by constraint locality.

In addition, the proposed selected inversion is compared with the hierarchical inversion approach [9] and their performance results are listed in Table II. N is the number

of synthetic grid nodes. \hat{N} is the number of grid nodes used in [9]. T_{inv}^h and T_{LP}^h are the runtime of hierarchical inversion and involved LP optimization, respectively, which are reported in [9]. Due to different benchmarks used for them, we compare the runtime of verifying each thousand of nodes while the number of nodes is almost same for each pair of benchmarks. For verifying each thousand of nodes, $T_{inv,perK}^s$ and $T_{LP,perK}^s$ are the runtime of selected inversion and involved LP optimization, respectively. Accordingly, $T_{inv,perK}^h$ and $T_{LP,perK}^h$ are the runtime of hierarchical inversion and involved LP optimization, respectively, for verifying each thousand of nodes. Even though disadvantage in hardware platforms and programming language (kernel solvers also implemented in C/C++), the proposed selected approach still obtains significant improvements. For matrix inverse computation, the selected approach can achieve several speedups than hierarchical approach which are listed in column 9. And for LP problem solving, the selected approach still can achieve about scores of speedups for the largest case. Consequently, it is worth noting that the proposed approach should be more efficient in verifying larger power grids, which make it scalable for large-scale power grid verification.

The proposed selected inversion approach is also evaluated on IBM power grid benchmarks and their performance results are listed in Table III when comparing with direct inversion and DualVN approach [14]. Only the GND network is verified for each benchmark. The notation N is the number of grid nodes after merging the short paths as equivalent nodes (not including the supply pads). M is the number of supply pads, and P is the number of partitions. The maximum voltage

TABLE III

PERFORMANCE RESULTS OF THE PROPOSED SELECTED INVERSION COMPARED WITH DIRECT INVERSION AND DUALVN APPROACH [14] FOR IBM POWER GRID BENCHMARKS. THE RUNTIME IS ILLUSTRATED IN SECOND. THE VOLTAGE DROP IS ILLUSTRATED IN MILLIVOLT

Benchmark				Direct Inversion		DualVN [14]	Selected Inversion				Speedup	
Netlist	N	M	P	T_{tot}^d	v_{max}	T_{tot}^{dual}	T_{inv}^s	T_{LP}^s	T_{tot}^s	E_{max}	T_{tot}^d/T_{tot}^s	T_{tot}^{dual}/T_{tot}^s
ibmpg1/GND	10242	177	40	285.92	632.52	123.60	15.32	45.85	61.17	9.71	5	2
ibmpg2/GND	65228	210	56	4210.50	313.21	5940.00	124.56	148.03	272.59	5.83	15	22
ibmpg3/GND	150687	461	80	35366.49	172.80	33228.00	1079.50	220.58	1300.08	4.58	27	26
ibmpg4/GND	478094	650	144	3645705.91	3.06	237600.00	18524.84	958.28	19483.12	0.42	187	12
ibmpg5/GND	291382	177	40	1052581.76	40.58	152064.00	14283.40	1362.70	15646.10	3.80	67	10
ibmpg6/GND	430337	249	100	3261534.89	103.21	391392.00	19544.65	1237.23	20781.88	5.25	157	19

drop v_{max} is still computed by direct inversion as a golden solution. Since the runtime of setup phase (including SPICE parser, matrix builder, and grid partitioning) is much less than verification phase, only the runtime of verification phase is listed in Table III. T_{tot}^d is the total runtime when using direct inversion method, including the runtime of direct inversion and involved LP optimization. T_{tot}^{dual} is the total runtime of DualVN approach that is obtained from [14]. T_{tot}^s is the total runtime when using selected inversion approach, including the runtime of direct inversion T_{inv}^s and the runtime of involved LP optimization T_{LP}^s . The maximum errors of selected inversion are list in column 11 while the number of partitions is carefully chosen for each benchmark so that the verification accuracy can be guaranteed. As shown in Table III, the selected inversion approach can achieve more than one hundred speedups compared with direct inversion method, and more than $20\times$ speedups compared with DualVN method. These results have confirmed that the introduced pad-aware partitioning scheme enables the selected inversion approach available for industrial power grids and the significant improvements of performance evaluation make vectorless verification of large-scale power grids practical.

The major advantage of the proposed selected approach is that it can significantly reduce the problem size by selectively computing the matrix inverse while the verification accuracy can still be ensured by constraint locality. For early stage power grid verification, the proposed approach can be very efficient for roughly estimating the hotspot regions as a prediction and subsequently other exact approaches can be further utilized on these regions for more accurate verification.

VI. CONCLUSION

In this paper, we presented a selected inversion approach for efficient vectorless verification of power grids. The original power grid is formulated in the manner of domain-decomposition approach, and grid locality is exploited to selectively compute the matrix inverse while the constraint locality is utilized to guarantee the verification accuracy. In addition, a scheme of pad-aware partitioning is employed for verifying realistic power grids and expected to make a contribution to other locality-driven power grid problems. Experimental results have confirmed the efficiency of the proposed approaches. The selected inversion technique has further improved the runtime efficiency, which will ensure that vectorless verification can be practical for large-scale

power grids. In addition, since both the proposed selected inversion algorithm and the involved LP problems can be performed in parallel at different levels, the verification problem will be conducted by exploiting parallelism in the future.

REFERENCES

- [1] D. Kouroussis and F. N. Najm, "A static pattern-independent technique for power grid voltage integrity verification," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2003, pp. 99–104.
- [2] M. Nizam, F. N. Najm, and A. Devgan, "Power grid voltage integrity verification," in *Proc. IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Aug. 2005, pp. 239–244.
- [3] N. H. A. Ghani and F. N. Najm, "Handling inductance in early power grid verification," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2006, pp. 127–134.
- [4] X. Xiong and J. Wang, "Vectorless verification of RLC power grids with transient current constraints," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2011, pp. 548–554.
- [5] C.-K. Cheng, P. Du, A. B. Kahng, G. K. H. Pang, Y. Wang, and N. Wong, "More realistic power grid verification based on hierarchical current and power constraints," in *Proc. ACM Int. Symp. Phys. Design (ISPD)*, 2011, pp. 159–166.
- [6] N. H. A. Ghani and F. N. Najm, "Fast vectorless power grid verification using an approximate inverse technique," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2009, pp. 184–189.
- [7] M. Avci and F. N. Najm, "Early P/G grid voltage integrity verification," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2010, pp. 816–823.
- [8] W. Zhao, Y. Cai, and J. Yang, "A multilevel \mathcal{H} -matrix-based approximate matrix inversion algorithm for vectorless power grid verification," in *Proc. 18th Asia South Pacific Design Autom. Conf. (DAC)*, Jan. 2013, pp. 163–168.
- [9] X. Xiong and J. Wang, "A hierarchical matrix inversion algorithm for vectorless power grid verification," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2010, pp. 543–550.
- [10] N. A. Ghani and F. N. Najm, "Power grid verification using node and branch dominance," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2011, pp. 682–687.
- [11] A. Goyal and F. N. Najm, "Efficient RC power grid verification using node elimination," in *Proc. Design Autom. Test Eur. (DATE)*, Mar. 2011, pp. 1–4.
- [12] Y. Wang, X. Hu, C.-K. Cheng, G. K. H. Pang, and N. Wong, "A realistic early-stage power grid verification algorithm based on hierarchical constraints," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 1, pp. 109–120, Jan. 2012.
- [13] X. Xiong and J. Wang, "An efficient dual algorithm for vectorless power grid verification under linear current constraints," in *Proc. 47th ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2010, pp. 837–842.
- [14] X. Xiong and J. Wang, "Dual algorithms for vectorless power grid verification under linear current constraints," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 10, pp. 1469–1482, Oct. 2011.
- [15] X. Xiong and J. Wang, "Constraint abstraction for vectorless power grid verification," in *Proc. 50th ACM/IEEE Design Autom. Conf. (DAC)*, May/Jun. 2013, pp. 1–6.
- [16] Z. Feng, "Scalable multilevel vectorless power grid voltage integrity verification," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 8, pp. 1388–1397, Aug. 2013.

- [17] Z. Feng, "Scalable vectorless power grid current integrity verification," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2013, pp. 1–8.
- [18] J. Yang, Y. Cai, Q. Zhou, and W. Zhao, "Selected inversion for vectorless power grid verification by exploiting locality," in *Proc. IEEE 31st Int. Conf. Comput. Design (ICCD)*, Oct. 2013, pp. 257–263.
- [19] W. Zhao, Y. Cai, and J. Yang, "Fast vectorless power grid verification using maximum voltage drop location estimation," in *Proc. 19th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2014, pp. 861–866.
- [20] S. R. Nassif. (Aug. 2011). *IBM Power Grid Benchmarks*. [Online]. Available: <http://dropzone.tamu.edu/~pli/PGBench>
- [21] E. Chiprout, "Fast flip-chip power grid analysis via locality and grid shells," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2004, pp. 485–488.
- [22] Z. Zeng and P. Li, "Locality-driven parallel power grid optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 8, pp. 1190–1200, Aug. 2009.
- [23] J. Shi *et al.*, "GPU friendly fast Poisson solver for structured power grid network analysis," in *Proc. 46th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2009, pp. 178–183.
- [24] T.-H. Chen and C. C. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2011, pp. 559–562.
- [25] J. Rommes and W. H. A. Schilders, "Efficient methods for large resistor networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 1, pp. 28–39, Jan. 2010.
- [26] T. A. Davis. *Cholmod in Suitesparse 4.1.2*. [Online]. Available: <http://www.cise.ufl.edu/research/sparse/SuiteSparse>, accessed Feb. 2014.
- [27] C. Bekas, A. Curioni, and I. Fedulova, "Low cost high performance uncertainty quantification," in *Proc. 2nd Workshop High Perform. Comput. Finance*, 2009, pp. 1–8.
- [28] L. Lin, C. Yang, J. C. Meza, J. Lu, L. Ying, and E. Weinan, "SelInv—An algorithm for selected inversion of a sparse symmetric matrix," *ACM Trans. Math. Softw.*, vol. 37, no. 4, 2011, Art. ID 40.
- [29] J. M. Tand and Y. Saad, "Domain-decomposition-type methods for computing the diagonal of a matrix inverse," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2813–2847, 2011.
- [30] I. Apostolopoulou, K. Daloukas, N. Evmorfopoulos, and G. Stamoulis, "Selective inversion of inductance matrix for large-scale sparse RLC simulation," in *Proc. 51st ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.
- [31] Z. Zeng, Z. Feng, P. Li, and V. Sarin, "Locality-driven parallel static analysis for power delivery networks," *ACM Trans. Design Autom. Electron. Syst.*, vol. 16, no. 31, pp. 28:1–28:17, Jun. 2011.
- [32] R. Achar, M. S. Nakhla, H. S. Dhindsa, A. R. Sridhar, D. Paul, and N. M. Nakhla, "Parallel and scalable transient simulator for power grids via waveform relaxation (PTS-PWR)," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 2, pp. 319–332, Feb. 2011.
- [33] S. Köse and E. G. Friedman, "Fast algorithms for IR voltage drop analysis exploiting locality," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2011, pp. 996–1001.
- [34] J. Yang, Y. Cai, Q. Zhou, and J. Shi, "Fast Poisson solver preconditioned method for robust power grid analysis," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2011, pp. 531–536.
- [35] G. T. Toussaint, "The relative neighbourhood graph of a finite planar set," *Pattern Recognit.*, vol. 12, no. 4, pp. 261–268, 1980.
- [36] R. B. Urquhart, "Algorithms for computation of relative neighbourhood graph," *Electron. Lett.*, vol. 16, no. 14, pp. 556–557, Jul. 1980.
- [37] D. V. Andrade and L. H. de Figueiredo, "Good approximations for the relative neighbourhood graph," in *Proc. 13th Can. Conf. Comput. Geometry*, 2001, pp. 25–28.
- [38] G. Karypis. *METIS—Graph Partitioning Software 4.0*. [Online]. Available: <http://glaros.dtc.umn.edu/gkhome/metis/metis/overview>, accessed Feb. 2014.
- [39] H. Li *et al.*, "Partitioning-based approach to fast on-chip decoupling capacitor budgeting and minimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 11, pp. 2402–2412, Nov. 2006.
- [40] Gurobi Optimization Inc. *Gurobi Optimizer 5.0.2*. [Online]. Available: <http://www.gurobi.com/download/gurobi-optimizer>, accessed Feb. 2014.
- [41] J. Yang, Z. Li, Y. Cai, and Q. Zhou, "PowerRush: A linear simulator for power grid," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2011, pp. 482–487.
- [42] *Qhull 2012.1*. [Online]. Available: <http://www.qhull.org>, accessed Feb. 2014.
- [43] C. D. Correa and P. Lindstorm. *NGL (Neighborhood Graph Library) 0.1*. [Online]. Available: <http://www.ngraph.org>, accessed Feb. 2014.



Jianlei Yang (S'12) received the B.S. degree in microelectronics from Xidian University, Xi'an, China, in 2009, and the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2014.

He was a Research Intern with Intel Labs China, Intel Corporation, Beijing, from 2013 to 2014, where he was involved in power estimation techniques for embedded systems. His current research interests include numerical algorithms for VLSI power grid analysis and verification, spintronics, and neuromorphic computing.

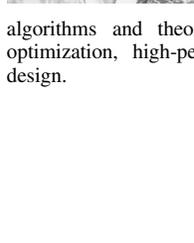
phic computing.

Dr. Yang's team was the recipient of the first place on TAU Power Grid Simulation Contest in 2011, and the second place on TAU Power Grid Transient Simulation Contest in 2012. He was a recipient of the IEEE ICCD Best Paper Award in 2013.



Yici Cai (M'04–SM'10) received the B.S. degree in electronic engineering and the M.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 1983 and 1986, respectively, and the Ph.D. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2007.

She has been a Professor with the Department of Computer Science and Technology, Tsinghua University. Her current research interests include design automation for VLSI integrated circuits algorithms and theory, power/ground distribution network analysis and optimization, high-performance clock synthesis, and low-power physical design.



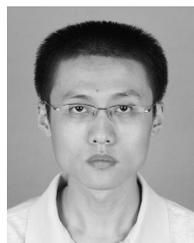
Qiang Zhou (M'04–SM'10) received the B.S. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 1983, the M.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 1986, and the Ph.D. degree in control theory and control engineering from the Chinese University of Mining and Technology, Beijing, in 2002.

He is a Professor with the Department of Computer Science and Technology, Tsinghua University. His current research interests include VLSI layout theory and algorithms.



University. His current

research interests include VLSI layout theory and algorithms.



Wei Zhao received the B.S. and M.S. degrees in computer science and technology from Tsinghua University, Beijing, China, in 2010 and 2013, respectively.

He has been with Huada Empyrean Software Corporation, Beijing, since 2013. His current research interests include power grid simulation and verification.

Mr. Zhao was a recipient of the IEEE ICCD Best Paper Award in 2013.