Power Grid Verification Under Transient Constraints

Mohammad Fawaz and Farid N. Najm
ECE Department
University of Toronto
Toronto, Canada
mohammad.fawaz@mail.utoronto.ca, f.najm@utoronto.ca

***Abstract*—Checking the power grid must begin early in the design. One way of doing this is using vectorless verification which, unlike standard simulation, only requires limited information about the currents drawn from the grid, in the form of DC local and global upper-bounds, or current constraints. We extend the standard vectorless verification to allow transient constraints, where circuit currents may be bounded by different values at different times. This is useful to check the validity of candidate sequences of chip operations, each having different current requirements. We show that this framework leads to a less pessimistic estimation of voltage drops.**

***Index Terms*—Power grid, vectorless, verification, transient constraints**

I. Introduction

The robust operation of an integrated circuit (IC) is highly dependent on its power grid. Technology scaling has led to a situation where the IR drops and the Ldi/dt noise across the chip have become more substantial than ever. This requires attention because large voltage fluctuations can cause timing violations and logic hazards. Therefore, there is a clear need for power grid verification to make sure that the voltage variations at all grid nodes are within certain user-defined thresholds.

Standard verification flows are based on *simulation*. Simulation tools evaluate the nodal voltages in response to certain current traces provided by the user, using direct circuit solving techniques. One must repeat the process enough times until the grid is declared *safe* with a certain level of confidence. State of the art simulation tools are quite efficient and can handle grids with billions of nodes. Many tools have been proposed for efficient transient simulation such as [1]–[10]. Unfortunately, simulation tools suffer from two crucial problems. First, predicting the exact current waveforms that are going to be drawn from the grid when the chip is in operation is very difficult. In addition, modern ICs have an astronomical number of different possible behavioral patterns, and so, a brute force approach that covers all possible current traces is not practical. Second, checking the grid must begin early in the design process, to allow for quick changes to the grid, and simulation tools are not very useful then because they require exact current traces which cannot be known without detailed information about the position and the power dissipation of every logic block.

To overcome these problems, the constraints-based framework was first introduced in [11]. This framework allows for a *vectorless* verification of the power grid which can be done early in the design process, and requires little information about the behavior of the chip, in the form of DC current constraints. Given some description of the *space* of possible currents, using current constraints (effectively, power budgets), the user can compute the *worst-case* voltage drop at every node to check its safety. Several approaches were developed to efficiently compute the worst-case voltage drops under DC constraints, such as [12]–[18]. One drawback of vectorless techniques is that they assume that the circuit currents satisfy the current constraints *for all* time, which means that the current space has to be large enough to accommodate for all current traces resulting from all possible circuit behaviors. This leads to an overly-conservative estimation of the worst-case voltage drops. In this paper, we introduce a new framework for vectorless verification under *transient* constraints where the currents are allowed to be bounded by different values at different times. We expect that this approach would be useful to check the validity of candidate sequences or patterns of chip operations over time, each having different current requirements. A sequence of chip operations would be described by a sequence of current constraints in the form of a *schedule*. In fact, power scheduling has become an important component of any reliable IC design. For thermal and supply integrity reasons, one cannot have all logic blocks turned on simultaneously, so that there will always be some blocks that are turned off (*dark Silicon*). Thus, there is a need to manage the workload so that the temperature and the voltage drop variations remain within specs. Power budgeting at the application level for thermal and voltage safety has been proposed in [19] and [20]. Here, we consider the power budgeting problem for voltage safety, where the power schedules are in the form of current or power constraints derived from the valid power modes of the underlying logic blocks. We believe that the method proposed for checking the safety of a certain schedule can lead, in the future, to a complete *scheduler* that can keep the grid safe at all time.

Verification under transient constraints can be computationally very expensive because the worst-case voltage drops need to be computed for every time point. An attempt to solve the problem under one form of transient constraints is presented in [21] and [22]. However, the algorithms described are inefficient as they require tens of weeks to verify a grid containing 1 million nodes. Our framework, as will be demonstrated, is much more efficient as we will be able to verify a 1 million nodes grid in less than 2 days. Our formulation will also lead to a better estimation of the voltage drops on the grid, which is much less pessimistic compared to the standard vectorless approach.

The rest of the paper is organized as follows. In Section II, we present the power grid model. In Section III, we discuss the constraints-based framework for verification and we introduce the transient constraints. We present some background material in Sec-

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

978-1-5386-3093-8/17/$31.00 ©2017 IEEE

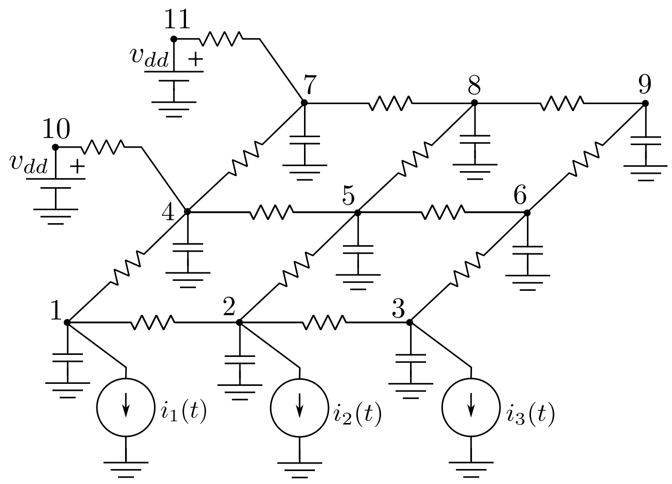


Fig. 1: The RC grid model

tion IV and show the exact computation of the worst-case voltage drops in Section V. In Section VI, we detail our proposed approach. Finally, we present the experimental results in Section VII before concluding the paper in Section VIII.

II. Power Grid Model

The power grid is a full-chip structure of connected metal lines that transfer voltage from the external supply to the on-die logic circuitry. Modern power grids consist of multiple layers interconnected through vias and connected by C4 bumps to wiring in the package and on the board. Typically, a power grid is modelled as a linear circuit composed of a large number of lumped linear (RLC) elements. In many cases, the inductance on the grid is ignored leading to a simpler RC model.

Consider an RC model of the power grid, similar to the one shown in Fig. 1, where each branch is represented by a resistors and where there exists a capacitor from every node (that is not a v_{dd} node) to ground. The work presented in this paper is not applicable to RLC grids because it relies on the special properties of the circuit matrices of an RC grid.

The nodes of an RC grid can be classified into three different types:

1) **Type 1:** nodes that are connected to ideal current sources to ground, in parallel with capacitors to ground.
2) **Type 2:** nodes that are connected to resistors to other grid nodes and capacitors to ground.
3) **Type 3:** nodes that are connected to resistors to other grid nodes and ideal voltage sources to ground.

Any connected interconnection of R and C elements is allowed, provided every node is one of the above three types. In the power grid of Fig. 1:

- Nodes 1, 2, and 3 are of type 1.
- Nodes 4, 5, 6, 7, 8, and 9 are of type 2.
- Nodes 10 and 11 are of type 3.

The current sources model the currents drawn by the underlying logic circuits while the ideal voltage sources model the external voltage supply v_{dd}. Excluding the ground node, assume that the power grid consists of $n + \kappa$ nodes where nodes $1, 2, \ldots, n$ are the nodes not connected to a voltage source, while the remaining nodes $(n + 1), (n + 2), \ldots, (n + \kappa)$ are the nodes where the κ voltage sources are connected. Also, let m be the number of current sources connected to the grid and $\boldsymbol{i}(t)$ be the vector of all the m current sources, whose positive (reference) direction is from node-to-ground. Finally, let $\boldsymbol{H}$ be the $n \times m$ matrix of 0 and 1 entries that identifies (with a 1) which node is connected to which current source. That is, if H_{jk} is the (j, k)th entry of $\boldsymbol{H}$, then

$$H_{jk} = \begin{cases} 1, & \text{if node } j \text{ is connected to current source } k; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

III. Current Constraints

A. DC Current Constraints

Generally, it is not practical to expect the users to specify the exact values of the currents drawn from the power grid. Determining these values requires specific information about the power dissipation of every logic block in the chip, as well as details about their activity patterns. In fact, during early stages of the power grid design, the user may not even know how the final underlying circuitry will look like, and so, performing standard simulation may be impossible. That being said, it is rarely the case that the users know *nothing* about the circuits loading the grid. It is reasonable to assume that there is always *some* engineering judgement or expertise from previous design activities, in the same or similar technologies, that users can bring to bear. Based on this assumption, and to overcome the uncertainty problem, the *current constraints* framework was introduced in [11]. These constraints are DC in the sense that they are fixed over time. They are an abstraction which aims to capture design knowledge, and they fall somewhere between knowing *everything* and knowing *nothing* about the currents and circuits that load the grid.

DC current constraints are upper-bound limits, or *envelopes*, on the current source waveforms that load the grid in the form $\mathbf{0} \leq \boldsymbol{i}(t) \leq \boldsymbol{i}_L, \forall t$. Even though they are DC constraints, they allow for all possible transient current waveforms that fit below them. Constraints on individual current sources are referred to as *local constraints*. If *only* local constraints are provided, the results would be overly pessimistic because it is never the case that all chip components simultaneously draw their maximum currents; hence the need for *global constraints*. A global constraint is an upper-bound on the *sum* of a group of current sources, such as

$$0 \leq i_{j_1}(t) + i_{j_2}(t) + \ldots + i_{j_k}(t) \leq i_{\max}, \quad \forall t, \tag{2}$$

for some $j_1, \ldots, j_k \in \{1, \ldots, m\}$ and some upper-bound $i_{\max} \geq 0$. Local and global constraints can be combined into the single matrix inequality

$$\mathbf{0} \leq \boldsymbol{W}\boldsymbol{i}(t) \leq \boldsymbol{i}_C, \quad \forall t, \tag{3}$$

where $\boldsymbol{W}$ is a matrix of 0s and 1s and $\boldsymbol{i}_C$ is a constant vector. This matrix inequality defines a multidimensional polytope in the current space [12], also referred to more generally as a *current container* [23] denoted, for example, by

$$\mathcal{Q} = \{\boldsymbol{i} : \mathbf{0} \leq \boldsymbol{W}\boldsymbol{i} \leq \boldsymbol{i}_C\}. \tag{4}$$

Given a current container, one can formulate the verification problem as: under all possible current waveforms that satisfy the constraints (i.e. $\boldsymbol{i}(t) \in \mathcal{Q}, \forall t$), check if the grid voltages satisfy certain user specifications, so that the grid is "safe". In this way, the verification problem becomes an optimization over a feasible space.

B. Transient Current Constraints

One drawback of the approach above is that it assumes that $\boldsymbol{i}(t) \in \mathcal{Q}$ *for all time*, which means that the feasible space $\mathcal{Q}$ must be large enough to accommodate for all current traces arising from all "reasonable" chip activity patterns. Thus, the *maximum* (or worst-case) voltage drop at every node over all possible $\boldsymbol{i}(t)$ vectors satisfying $\boldsymbol{i}(t) \in \mathcal{Q}, \forall t$, will be overly pessimistic. To overcome this problem, we propose a new framework for power grid verification which builds upon the DC constraints presented above, and which aims to reduce the pessimism in the worst-case voltage drop estimation.

Assume that the user is interested in the behavior of the power grid for the time instants $t \leq T$, for some $T \geq 0$, and let

$$0 = t_0 < t_1 < \ldots < t_{N-1} = T \tag{5}$$

be certain time instants inside the interval $[0, T]$ dividing the interval $(-\infty, T]$ into the N intervals

$$(-\infty, t_0], (t_0, t_1], (t_1, t_2], \ldots, (t_{N-2}, t_{N-1}]. \tag{6}$$

Define $T_0 \triangleq (-\infty, t_0]$ and

$$T_k \triangleq (t_{k-1}, t_k], \quad k = 1, \ldots, N-1. \tag{7}$$

For every interval $T_k, k = 0, 1, \ldots, N-1$, assume that the user can provide a set of local constraints, based on the same kind of information used to develop the local constraints of the previous section,

$$\mathbf{0} \leq \boldsymbol{i}(t) \leq \boldsymbol{i}_{L,k}, \quad \forall t \in T_k, \tag{8}$$

where $\boldsymbol{i}_{L,k}$ is an $m \times 1$ vector of the maximum values that the current sources can draw for $t \in T_k$ only. Similarly, we assume that the user can provide a set of global constraints which can be expressed in matrix form as

$$\mathbf{0} \leq \boldsymbol{U}_k \boldsymbol{i}(t) \leq \boldsymbol{i}_{G,k}, \quad \forall t \in T_k, \tag{9}$$

where $\boldsymbol{U}_k$ is a 0-1 matrix that indicates (with a 1) which current sources are present in each global constraint, and $\boldsymbol{i}_{G,k}$ is a vector of maximum sums of currents attained by each group of current sources considered in the global constraints. Together, the local and global constraints can be combined into 1 matrix inequality

$$\mathbf{0} \leq \boldsymbol{W}_k \boldsymbol{i}(t) \leq \boldsymbol{i}_{C,k}, \quad \forall t \in T_k. \tag{10}$$

For $k = 0, 1, \ldots, N-1$, let

$$\mathcal{F}_k \triangleq \{\boldsymbol{i} : \mathbf{0} \leq \boldsymbol{W}_k \boldsymbol{i} \leq \boldsymbol{i}_{C,k}\}. \tag{11}$$

Each set $\mathcal{F}_k$ is a convex polytope, and so, the user-defined space of currents is now a *time-varying* convex polytope in the time interval $(-\infty, T]$.

Re-defining the constraints-based framework in this way allows the user to have more freedom in describing the behavior of the chip using current constraints. Given a *sequence* of chip operations running for a certain period of time, the user can provide a unique feasible space tailored for each operation. This will reduce the pessimism in the worst-case voltage drop estimation that would arise if only *one* big feasible, accounting for all the operations, was to be considered.

We understand that this new way of formulating the vectorless problem is more demanding than the standard framework. However, we believe that the amount of information required to build the different containers is still reasonable, and the users can be expected to know this kind of information based on the power requirements of circuit blocks in their different modes of operation.

For ease of access, we define a map $\mathcal{F}$, over $(-\infty, T]$, such that, for every $t \leq T$,

$$\mathcal{F}(t) = \begin{cases} \mathcal{F}_k, & \text{if } t \in T_k, k = 0, 1, \ldots, N-1; \\ \phi, & \text{otherwise,} \end{cases} \tag{12}$$

where ϕ is the empty set in $\mathbb{R}^m$. Given the set of feasible spaces and their corresponding time intervals, one can now formulate the verification problem as follows. Under all current waveforms $\boldsymbol{i}(t)$ that satisfy $\boldsymbol{i}(t) \in \mathcal{F}(t), \forall t$, check if the grid voltages satisfy the user specification, i.e., check if the grid is "safe".

IV. Background

In this section, we derive a recurrence relationship that describes the evolution of an RC power grid over time. The recurrence is based on nodal analysis and backward Euler and was previously done in [12].

A. System Equations

Let $\boldsymbol{u}(t)$ be the $n \times 1$ vector of node voltages relative to ground. By superposition, $\boldsymbol{u}(t)$ can be found in three steps:

1) Open-circuit all the current sources and find the response, which would be $\boldsymbol{u}^{(1)}(t) = v_{dd}\mathbb{1}$, where $\mathbb{1}$ is an $n \times 1$ vector of ones.
2) Short-circuit all the voltage sources and find the response $\boldsymbol{u}^{(2)}(t)$, in this case clearly $\boldsymbol{u}^{(2)}(t) \leq \mathbf{0}$.
3) Find $\boldsymbol{u}(t) = \boldsymbol{u}^{(1)}(t) + \boldsymbol{u}^{(2)}(t)$.

To find $\boldsymbol{u}^{(2)}(t)$, KCL at every node provides, via Nodal Analysis (NA), that

$$\boldsymbol{G}\boldsymbol{u}^{(2)}(t) + \boldsymbol{C}\dot{\boldsymbol{u}}^{(2)}(t) = -\boldsymbol{H}\boldsymbol{i}(t), \tag{13}$$

where $\boldsymbol{C} \geq \mathbf{0}$ is the $n \times n$ diagonal *non-singular* capacitance matrix of the grid, and $\boldsymbol{G}$ is its $n \times n$ conductance matrix. The matrix $\boldsymbol{G}$ is known to be symmetric and diagonally dominant with positive diagonal entries and non-positive off-diagonal entries. Under the standard assumption that the resistive mesh is connected and has at least one voltage source, the matrix $\boldsymbol{G}$ becomes irreducibly diagonally dominant [13]. With this $\boldsymbol{G}$ can be shown to be an $\mathcal{M}$-matrix [24], so that $\boldsymbol{G}^{-1}$ exists and is non-negative, $\boldsymbol{G}^{-1} \geq \mathbf{0}$, and all the eigenvalues of $\boldsymbol{G}$ are real and positive. For verification purposes, we are mainly interested in the vector of *voltage drops* $\boldsymbol{v}(t) \triangleq v_{dd}\mathbb{1} - \boldsymbol{u}(t) = -\boldsymbol{u}^{(2)}(t) \geq \mathbf{0}$, so that

$$\boldsymbol{G}\boldsymbol{v}(t) + \boldsymbol{C}\dot{\boldsymbol{v}}(t) = \boldsymbol{H}\boldsymbol{i}(t). \tag{14}$$

Note therefore that $\boldsymbol{v}(t)$ can be found directly as the node voltages resulting from an analysis of the RC mesh modified such that all the voltage sources are short-circuited and the directions of all the current sources are reversed.

B. Time Discretization

Solving (14) typically starts by discretizing time using a finite difference approximation of the derivative. Using the backward Euler (BE) numerical scheme, the derivative $\dot{\boldsymbol{v}}(t)$ is approximated using $\frac{\boldsymbol{v}(t) - \boldsymbol{v}(t - \Delta t)}{\Delta t}$ for some discretization parameter (time step) Δt. Assuming that Δt is small enough, we obtain

$$\boldsymbol{A}\boldsymbol{v}(t) = \boldsymbol{B}\boldsymbol{v}(t - \Delta t) + \boldsymbol{H}\boldsymbol{i}(t), \tag{15}$$

where $\boldsymbol{B} \triangleq \frac{\boldsymbol{C}}{\Delta t}$ and $\boldsymbol{A} \triangleq \boldsymbol{G} + \boldsymbol{B}$. Recall that the BE scheme is motivated by the following Taylor expansion of a one dimensional time function $x(t)$, where $\xi \in [t - \Delta t, t]$,

$$x(t - \Delta t) = x(t) - (\Delta t)\dot{x}(t) + \frac{(\Delta t)^2}{2}\ddot{x}(\xi). \quad (16)$$

Thus, the choice of Δt relates only to the spectral properties of the node voltage signals. We assume that the time step Δt is chosen such that (15) is accurate enough irrespective of the current stimulus $\boldsymbol{i}(t)$. Accordingly, (15) leads to a recurrence relation that captures the evolution of the system over time, so that the voltage drop at any time t is given by

$$\boldsymbol{v}(t) = \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{v}(t - \Delta t) + \boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}(t). \quad (17)$$

The inverse $\boldsymbol{A}^{-1}$ exists and $\boldsymbol{A}^{-1} \geq \boldsymbol{0}$ because $\boldsymbol{A}$ is also an $\mathcal{M}$-matrix as it possesses the same properties as $\boldsymbol{G}$. This implies that $\boldsymbol{A}^{-1}\boldsymbol{B} \geq \boldsymbol{0}$, which is a property that will prove to be quite useful in developing our proposed approach.

V. Worst-Case Voltage Drops

In this section, we derive the exact solution to the RC verification problem under the transient constraints explained earlier.

Assume that T is a multiple of Δt. If T is not a multiple of Δt, then we can simply redefine T to be the largest multiple of Δt that is smaller than T. Because Δt is small, this will have a minimal effect on the quality of the verification process.

Using the discretized system (17), we will derive an expression for the vector of worst-case voltage drops at every time point t that is a multiple of Δt, i.e. for $t = p\Delta t$ where $p \in \mathbb{Z}$ and $p \leq \frac{T}{\Delta t}$. At $t = p\Delta t$, we can rewrite (17) as

$$\boldsymbol{v}(p\Delta t) = \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{v}((p-1)\Delta t) + \boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}(p\Delta t). \quad (18)$$

Similarly, at time $t = (p-1)\Delta t$, we can write

$$\boldsymbol{v}((p-1)\Delta t) = \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{v}((p-2)\Delta t) + \boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}((p-1)\Delta t). \quad (19)$$

Combining (18) and (19), we obtain

$$\begin{aligned}\boldsymbol{v}(p\Delta t) &= (\boldsymbol{A}^{-1}\boldsymbol{B})^2\boldsymbol{v}((p-2)\Delta t) \\ &\quad + \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}((p-1)\Delta t) + \boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}(p\Delta t). \quad (20)\end{aligned}$$

We can repeat this procedure to obtain

$$\begin{aligned}\boldsymbol{v}(p\Delta t) &= (\boldsymbol{A}^{-1}\boldsymbol{B})^r\boldsymbol{v}((p-r)\Delta t) \\ &\quad + \sum_{q=0}^{r-1}(\boldsymbol{A}^{-1}\boldsymbol{B})^q\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}((p-q)\Delta t). \quad (21)\end{aligned}$$

Recall that the *spectral radius* of a matrix $\boldsymbol{X}$, denoted by $\rho(\boldsymbol{X})$, is the maximum among the magnitudes of the eigenvalues of $\boldsymbol{X}$. From [12], we know that

$$\rho(\boldsymbol{A}^{-1}\boldsymbol{B}) < 1. \quad (22)$$

This implies [25]

$$\lim_{r\to\infty}(\boldsymbol{A}^{-1}\boldsymbol{B})^r = \boldsymbol{0}. \quad (23)$$

Therefore, because $\boldsymbol{v}((p-r)\Delta t)$ is bounded (the grid being a stable system with bounded inputs), we also have

$$\lim_{r\to\infty}(\boldsymbol{A}^{-1}\boldsymbol{B})^r\boldsymbol{v}((p-r)\Delta t) = \boldsymbol{0}. \quad (24)$$

Thus, we can let $r \to \infty$ in (21) to get

$$\boldsymbol{v}(p\Delta t) = \sum_{q=0}^{\infty}(\boldsymbol{A}^{-1}\boldsymbol{B})^q\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}((p-q)\Delta t). \quad (25)$$

For verification purposes, we are interested in the worst-case voltage drop at every node in the grid at every time point $t = p\Delta t$. We capture this using the vector $\boldsymbol{v}^*(p\Delta t)$ whose j^{th} entry is defined as

$$v_j^*(p\Delta t) \triangleq \max_{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}[v_j(p\Delta t)]. \quad (26)$$

Notice that this maximization of the voltage drop is over *all current waveforms* $\boldsymbol{i}(\tau)$ that are *feasible*. Feasibility is captured using the notation $\boldsymbol{i}(\tau) \in \mathcal{F}(\tau), \forall\tau$, where $\mathcal{F}$ is the map, defined in (12), that returns the feasible space corresponding to time point $t = \tau$.

To express the full vector of worst-case voltage drops $\boldsymbol{v}^*(p\Delta t)$, we use an extreme-value operator that we call $\mathrm{emax}[\cdot]$, which is an *element-wise maximization* operator. This is essentially an n-dimensional $\max[\cdot]$ operator which takes a vector of n objective functions and returns a vector containing the result of maximizing each function. Accordingly

$$\boldsymbol{v}^*(p\Delta t) = \begin{bmatrix} \max\limits_{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}[v_1(p\Delta t)] \\ \vdots \\ \max\limits_{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}[v_n(p\Delta t)] \end{bmatrix} = \operatorname*{emax}_{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}[\boldsymbol{v}(p\Delta t)].$$

Therefore

$$\boldsymbol{v}^*(p\Delta t) = \operatorname*{emax}_{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}\left[\sum_{q=0}^{\infty}(\boldsymbol{A}^{-1}\boldsymbol{B})^q\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}((p-q)\Delta t)\right].$$

Because the values of the source currents at any two time points are independent variables, we have

$$\boldsymbol{v}^*(p\Delta t) = \sum_{q=0}^{\infty}\operatorname*{emax}_{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}\left[(\boldsymbol{A}^{-1}\boldsymbol{B})^q\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}((p-q)\Delta t)\right].$$

This simplifies to

$$\boldsymbol{v}^*(p\Delta t) = \sum_{q=0}^{\infty}\operatorname*{emax}_{\boldsymbol{i}\in\mathcal{F}((p-q)\Delta t)}\left[(\boldsymbol{A}^{-1}\boldsymbol{B})^q\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}\right], \quad (27)$$

where $\boldsymbol{i}$ is a dummy variable. This expression is of theoretical interest only because it is prohibitively expensive to compute as it requires performing a large number of matrix-matrix multiplications and solving several optimization problems, for every p. For that reason, we propose a bound on $\boldsymbol{v}^*(p\Delta t)$ that is much easier to compute and that allows a *conservative* verification of the power grid.

VI. Proposed Approach

In this section, we derive a conservative upper-bound on the exact solution (27). The bound is much easier to compute than $\boldsymbol{v}^*(p\Delta t)$ as it requires solving a much smaller number of optimization problems and no matrix-matrix multiplications.

A. Upper-Bound Waveforms

Using (17), one can write, at $t = p\Delta t$

$$\begin{aligned} \boldsymbol{v}^*(p\Delta t) &= \underset{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}{\mathrm{emax}}[\boldsymbol{v}(p\Delta t)] \\ &= \underset{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{v}((p-1)\Delta t)+\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}(p\Delta t)] \\ &\le \underset{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{v}((p-1)\Delta t)] \\ &\qquad\qquad + \underset{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}(p\Delta t)]. \end{aligned} \tag{28}$$

We examine the terms of the right hand side of the inequality separately. Because $\boldsymbol{A}^{-1}\boldsymbol{B} \ge \mathbf{0}$, we have

$$\begin{aligned} &\underset{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{v}((p-1)\Delta t)] \\ &\qquad \le \boldsymbol{A}^{-1}\boldsymbol{B}\underset{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}{\mathrm{emax}}[\boldsymbol{v}((p-1)\Delta t)] \\ &\qquad = \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{v}^*((p-1)\Delta t). \end{aligned} \tag{29}$$

Also, we clearly have

$$\underset{\boldsymbol{i}(\tau)\in\mathcal{F}(\tau),\forall\tau}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}(p\Delta t)] = \underset{\boldsymbol{i}\in\mathcal{F}(p\Delta t)}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}], \tag{30}$$

for some dummy variable $\boldsymbol{i}$. Thus, using (29) and (30), the inequality in (28) becomes

$$\boldsymbol{v}^*(p\Delta t) \le \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{v}^*((p-1)\Delta t) + \underset{\boldsymbol{i}\in\mathcal{F}(p\Delta t)}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}]. \tag{31}$$

Based on the discussion above, we present the following theorem, which provides an upper-bound $\overline{\boldsymbol{v}}$ on the true worst-case voltage drop vector $\boldsymbol{v}^*$, defined at every point $p\Delta t \le T$.

Theorem 1. *For every $p \in \mathbb{Z}, p \le \frac{T}{\Delta t}$, let*

$$\overline{\boldsymbol{v}}(p\Delta t) = \begin{cases} \boldsymbol{G}^{-1}\boldsymbol{A}\underset{\boldsymbol{i}\in\mathcal{F}_0}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}], & p \le 0; \\ \boldsymbol{A}^{-1}\boldsymbol{B}\overline{\boldsymbol{v}}((p-1)\Delta t) + \underset{\boldsymbol{i}\in\mathcal{F}(p\Delta t)}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}], & 1 \le p \le \frac{T}{\Delta t}. \end{cases} \tag{32}$$

We have

$$\boldsymbol{v}^*(p\Delta t) \le \overline{\boldsymbol{v}}(p\Delta t), \quad \forall p \in \mathbb{Z}, p \le \frac{T}{\Delta t}. \tag{33}$$

Proof. If $p \le 0$, then we can rewrite (27) as

$$\boldsymbol{v}^*(p\Delta t) = \sum_{q=0}^{\infty} \underset{\boldsymbol{i}\in\mathcal{F}_0}{\mathrm{emax}}\left[(\boldsymbol{A}^{-1}\boldsymbol{B})^q\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}\right], \quad \forall p \le 0, \tag{34}$$

where we have replaced $\boldsymbol{i} \in \mathcal{F}((p-q)\Delta t)$ by $\boldsymbol{i} \in \mathcal{F}_0$ because $(p-q)\Delta t \le 0$ whenever $p \le 0$, and so, $\mathcal{F}((p-q)\Delta t) = \mathcal{F}_0$. Because $\boldsymbol{A}^{-1}\boldsymbol{B} \ge \mathbf{0}$, we can bound (34) as follows

$$\begin{aligned} \boldsymbol{v}^*(p\Delta t) &\le \sum_{q=0}^{\infty}(\boldsymbol{A}^{-1}\boldsymbol{B})^q \underset{\boldsymbol{i}\in\mathcal{F}_0}{\mathrm{emax}}\left[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}\right] \\ &= \left(\sum_{q=0}^{\infty}(\boldsymbol{A}^{-1}\boldsymbol{B})^q\right)\underset{\boldsymbol{i}\in\mathcal{F}_0}{\mathrm{emax}}\left[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}\right], \quad \forall p \le 0. \end{aligned}$$

The summation $\sum_{q=0}^{\infty}(\boldsymbol{A}^{-1}\boldsymbol{B})^q$ was shown to be convergent to $\boldsymbol{G}^{-1}\boldsymbol{A}$ in [12], so that

$$\boldsymbol{v}^*(p\Delta t) \le \boldsymbol{G}^{-1}\boldsymbol{A}\underset{\boldsymbol{i}\in\mathcal{F}_0}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}] = \overline{\boldsymbol{v}}(p\Delta t), \quad \forall p \le 0, \tag{35}$$

as required.

For $p \ge 1$, the proof is by induction, as follows.

Base case: for $p = 1$, $\boldsymbol{v}^*((p-1)\Delta t) = \boldsymbol{v}^*(0) \le \overline{\boldsymbol{v}}(0)$, by the first part of this proof. So, using (31) for $p = 1$, and because $\boldsymbol{A}^{-1}\boldsymbol{B} \ge \mathbf{0}$, we have

$$\boldsymbol{v}^*(\Delta t) \le \boldsymbol{A}^{-1}\boldsymbol{B}\overline{\boldsymbol{v}}(0) + \underset{\boldsymbol{i}\in\mathcal{F}(\Delta t)}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}]. \tag{36}$$

The right hand side of this inequality is exactly $\overline{\boldsymbol{v}}(\Delta t)$ as defined in (32). Therefore, (33) is true for $p = 1$.

Induction hypothesis: Assume that (33) holds for $p = s - 1$ with $1 \le s - 1 \le \frac{T}{\Delta t} - 1$, that is,

$$\boldsymbol{v}^*((s-1)\Delta t) \le \overline{\boldsymbol{v}}((s-1)\Delta t). \tag{37}$$

Induction step: By (31) and the induction hypothesis, and because $\boldsymbol{A}^{-1}\boldsymbol{B} \ge \mathbf{0}$, we have

$$\boldsymbol{v}^*(s\Delta t) \le \boldsymbol{A}^{-1}\boldsymbol{B}\overline{\boldsymbol{v}}((s-1)\Delta t) + \underset{\boldsymbol{i}\in\mathcal{F}(s\Delta t)}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}] = \overline{\boldsymbol{v}}(s\Delta t).$$

Therefore, (33) is true $\forall p \in \mathbb{Z}, 1 \le p \le \frac{T}{\Delta t}$, by induction. This completes the proof. □

Theorem 1 provides a simple bound on $\boldsymbol{v}^*(p\Delta t)$ for every $p \in \mathbb{Z}, p \le \frac{T}{\Delta t}$, that is much easier to compute than (27). The vector $\overline{\boldsymbol{v}}(p\Delta t)$ can now be used to *conservatively* check the safety of the grid: if $\overline{\boldsymbol{v}}(p\Delta t)$ is within spec for every p, then $\boldsymbol{v}^*(p\Delta t)$ is also within spec, and so, the grid is safe from voltage drop.

Notice that, unlike the work of [12] where the worst-case voltage drop at every node is bounded by a constant upper-bound for all t, our proposed algorithm generates different upper-bounds at different time points. This is necessary because, unlike the feasible space of [12], our feasible space *changes* with time. Effectively, our algorithm returns, for every node, a *waveform* that bounds the worst-case voltage drop at every $t = p\Delta t$.

An upper-bound waveform is much more useful than the simple DC bound generated by the standard vectorless approach for two reasons that will become more apparent in the experimental results section. First, the upper-bound waveforms we generate are much less conservative, and so, they lead to larger safety margins which allows for a more relaxed power grid design process. Second, an upper-bound waveform for every node is more useful than a DC upper-bound for debugging purposes, because it allows the user to know which time points and which circuit blocks are problematic.

B. Implementation

Computing (32) requires computing $\boldsymbol{A}^{-1}\boldsymbol{H}$ explicitly. We do this using a Cholesky factorization [25] of $\boldsymbol{A}$ followed by a sequence of forward/backward solves against the columns of $\boldsymbol{H}$. The Cholesky algorithm can be used because $\boldsymbol{A}$ is a symmetric $\mathcal{M}$-matrix, and so, it is *symmetric positive definite* (SPD) [24]. Every time a column of $\boldsymbol{A}^{-1}\boldsymbol{H}$ is generated, all "small" entries are dropped in order to reduce the memory consumption and to speed up the subsequent steps of the algorithm. The threshold that determines which entries to drop is found using a separate engine which takes, as input, a user-defined threshold (in mV) on the voltage drop at every node. We skip the details of this engine due to lack of space.

The bound also requires performing an $\mathrm{emax}[\cdot]$ operation over $\mathcal{F}_0$ and an $\mathrm{emax}[\cdot]$ operation over $\mathcal{F}(p\Delta t)$ for every $1 \le p \le \frac{T}{\Delta t}$. Because $\mathcal{F}(p\Delta t) \in \{\mathcal{F}_1, \ldots, \mathcal{F}_{N-1}\}$ for any $p \ge 1$, and because

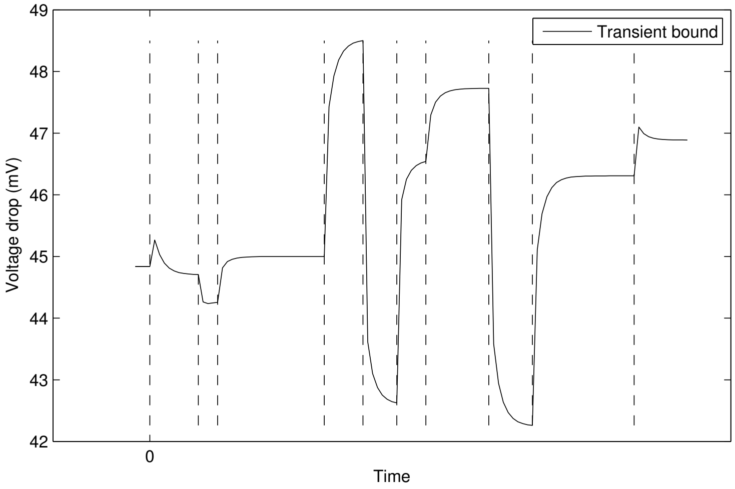


Fig. 2: Transient upper-bound for an arbitrary node in a 1596-nodes grid using 11 containers

Algorithm 1 COMPUTE_UPPER_BOUND_WAVEFORMS

```
Require: $\boldsymbol{G}, \boldsymbol{C}, \mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_{N-1}$
Ensure: $\overline{\boldsymbol{v}}(t)$
Compute $\boldsymbol{A}^{-1}\boldsymbol{H}$
for $k = 0, 1, \ldots, N-1$ do
    $\boldsymbol{e}_k^* = \underset{\boldsymbol{i} \in \mathcal{F}_k}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}]$
end for
Solve $\boldsymbol{G}\boldsymbol{r}_0 = \boldsymbol{A}\boldsymbol{e}_0^*$ for $\boldsymbol{r}_0$
Set $\overline{\boldsymbol{v}}(p\Delta t) = \boldsymbol{r}_0, \forall p \leq 0$
for $p = 1, \ldots, \frac{T}{\Delta t}$ do
    Solve $\boldsymbol{A}\boldsymbol{r}_p = \boldsymbol{B}\overline{\boldsymbol{v}}((p-1)\Delta t)$
    Find $k$ such that $p\Delta t \in T_k$
    Set $\overline{\boldsymbol{v}}(p\Delta t) = \boldsymbol{r}_p + \boldsymbol{e}_k^*$
end for
return $\overline{\boldsymbol{v}}(t)$
```

the coefficients of the objective functions are all the same (namely the rows of $\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}$), then it is enough to compute

$$\boldsymbol{e}_k^* \triangleq \underset{\boldsymbol{i} \in \mathcal{F}_k}{\mathrm{emax}}[\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{i}], \quad k = 0, 1, \ldots, N-1. \tag{38}$$

This would satisfy all the needs of (32) while performing the smallest number of $\mathrm{emax}[\cdot]$ operations possible. Computing each $\boldsymbol{e}_k^*$ can be done by solving n *linear programs* (LPs) because all the $\mathcal{F}_k$'s are *convex polytopes* and all the objective functions (which are obtained from the rows of $\boldsymbol{A}^{-1}\boldsymbol{H}$) are linear. Once all the $\boldsymbol{e}_k^*$'s are found, computing the actual bounds becomes an easy task. The full procedure for finding the upper-bounds is presented in Algorithm 1. The output of the algorithm is a *piece-wise linear* upper-bound waveform on the worst-case voltage drop waveform for every node.

VII. Experimental Results

We have implemented Algorithm 1 in C++. The simulation platform is a 64-bit Linux machine with a 12-core Intel Xeon CPU running at 3.0 GHz and 128 GB of RAM. All linear systems are solved using CHOLMOD [26] from SuiteSparse [27], and all linear programs (LP) are solved using MOSEK 8 [28]. We test the upper-bound algorithm on a set of 5 power grids generated based on user-defined parameters such as grid dimensions, number of layers, pitch and width per layer, number of sources, etc. The technology parameters chosen are consistent with 1 V 45 nm CMOS technology.

We choose $N = 11$ so that we have 11 containers $\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_{10}$, describing a certain sequence of chip operations. Container $\mathcal{F}_0$ describes the steady state behavior of the chip before the beginning of the set of operations of interest ($t \leq 0$). Each of the containers $\mathcal{F}_1, \ldots, \mathcal{F}_{N-1}$ describes the current requirements on the chip when certain blocks are being used while others are in stand-by mode. In other words, the containers are generated in a way to allow those current sources corresponding to the blocks that are ON to draw *high* currents, while blocks that are OFF can only draw *low* currents. In a more general setting, the containers can describe different power modes for each block such as high performance, lower performance, stand-by, etc.

To showcase the importance of our proposed framework, we first show, in Fig. 2, an example of how the transient upper-bound waveform $\overline{\boldsymbol{v}}(t)$ looks like for an arbitrary node in a 1596-nodes grid with $N = 11$. Each dotted line corresponds to a time point at which the feasible container changes. We observe that the upper-bound waveform transitions smoothly within each phase due to the capacitance on the grid. We also observe a *wide* variation in the worst-case voltage drop at different points in time, showing that a single upper-bound obtained from the standard vectorless framework may overestimate the voltage drops by significant amounts. Having an upper-bound *waveform* is also quite useful for debugging purposes as it tells the user where the critical points in time are and which blocks are more problematic than others for the sequence of operations in question.

For comparison purposes, we also perform experiments with standard DC constraints where the container we consider is the *smallest convex* container (polytope) $\overline{\mathcal{F}}$ that contains all the containers $\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_{N-1}$. Essentially, this represents the con-

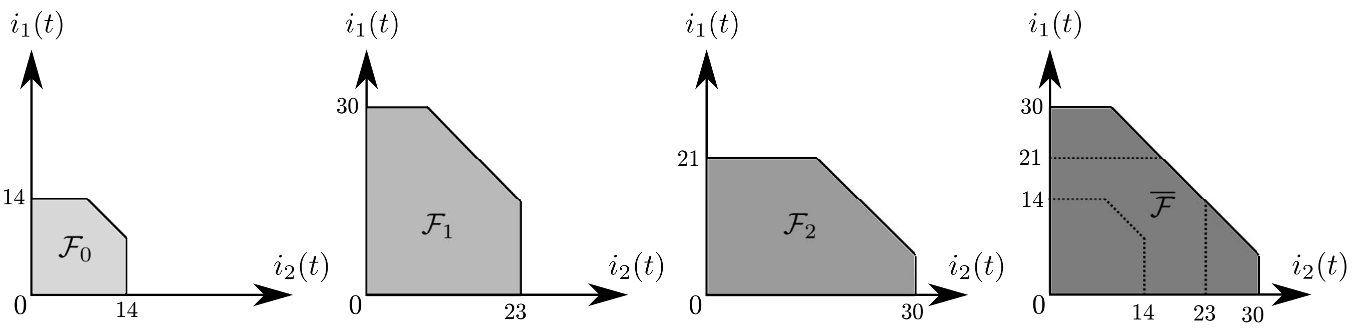
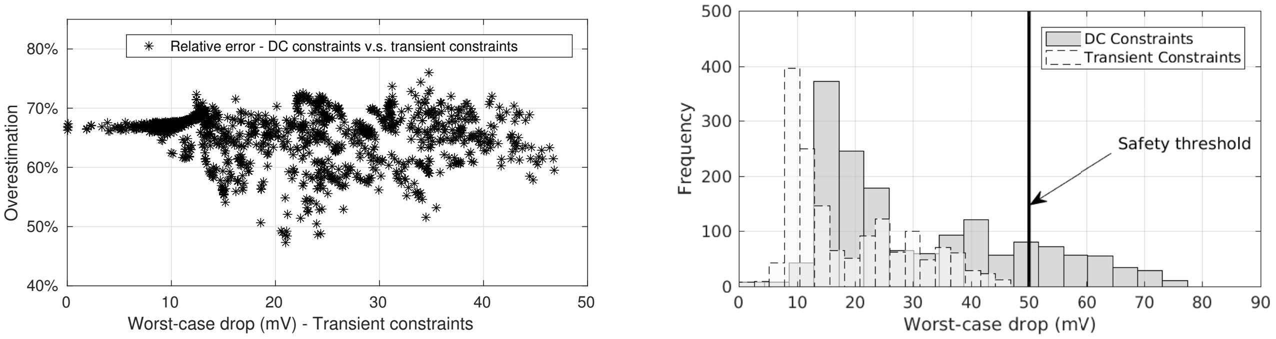


Fig. 3: Sample containers for a grid with 2 current sources

TABLE I: Performance of the proposed algorithm with 11 containers

Power Grid			Worst-Case Estimation for an Arbitrary Node			Runtime	
Grid	**Nodes**	**Sources**	**Transient Constraints**	**DC Constraints**	**Overestimation**	**Total**	**Average Per Container**
G1	51,838	11,234	45.91 mV	77.32 mV	**68.4%**	**4.47 min**	0.41 min
G2	100,596	21,774	36.52 mV	61.59 mV	**68.7%**	**19.28 min**	1.75 min
G3	251,112	54,360	23.96 mV	41.26 mV	**72.2%**	**2.39 h**	13.04 min
G4	506,488	109,952	19.62 mV	33.52 mV	**70.9%**	**11.54 h**	1.05 h
G5	1,007,064	218,340	17.51 mV	29.64 mV	**69.3%**	**43.46 h**	3.94 h



(a) Overestimation of the standard vectorless framework

(b) Worst-case voltage drop distribution across grid nodes

Fig. 4: The DC constraints framework versus the transient constraints framework for a 1596-nodes grid

tainer that describes the behavior of the chip undergoing the sequence of operations in question if only DC constraints were to be used. For example, Fig. 3 shows sample containers for a grid with two current sources and undergoing a sequence of operations that requires three consecutive feasible spaces $\mathcal{F}_0, \mathcal{F}_1$, and $\mathcal{F}_2$, for the periods $(-\infty, 0]$, $(0, t_1]$, and $(t_1, T]$ respectively. Notice that container $\mathcal{F}_1$ is built to be *biased* towards $i_1(t)$ as it allows the circuit block modelled by i_1 to draw *high current*. The same is true for $\mathcal{F}_2$ and $i_2(t)$. The last container $\overline{\mathcal{F}}$ shown represents the smallest container that contains $\mathcal{F}_0, \mathcal{F}_1$, and $\mathcal{F}_2$. The container $\overline{\mathcal{F}}$ is the container used for standard vectorless verification.

Table I shows, for each of the 5 grids tested, the worst-case voltage drop observed at an arbitrary node using both frameworks. For the case with transient constraints, the peak of the upper-bound waveform is reported. We observe that the standard approach overestimates the worst-case voltage drops by up to 72.2%. Moreover, the overestimations for all the nodes of a 1596-nodes grid are shown in Fig. 4a where the relative errors seem to vary between 47% and 76%, indicating that transient constraints generally lead to more realistic voltage noise estimation, thus increasing the safety margins and allowing cheaper power grid designs. To make this point clearer, Fig. 4b shows the distribution of the worst-case voltage drops estimated using each method for the same 1596-nodes grid. If the user-defined safety threshold is set to 5% of v_{dd}, that is 50 mV, then the grid would be deemed *unsafe* if the standard approach is used, with around 18.1% of the nodes failing to meet the safety spec. However, with a more accurate description of the chip operation using transient constraints, the grid can be actually found safe.

Table I also shows the total runtime of the proposed algorithm for $N = 11$. The runtime includes computing $\boldsymbol{A}^{-1}\boldsymbol{H}$ as well as solving n linear programs for each of the N container. Fig. 5 also shows the runtime versus the number of nodes, where the empirical complexity of the algorithm was found to be $\mathcal{O}(n^{1.92})$. Moreover Table I shows the average runtime per container, which turns out to be less than 4 hours for the largest grid tested (1M nodes). Knowing the average runtime per container allows predicting the total runtime of the algorithm for different values of N. Our

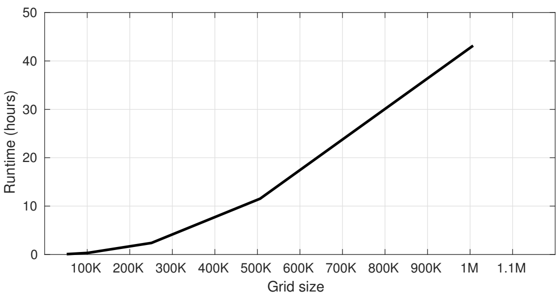


Fig. 5: Runtime analysis of the proposed algorithm

framework is roughly N times slower than the standard vectorless verification approach, which is quite reasonable considering that N is expected to be small since it is supposed to represent the number of different chip behaviors for a certain sequence of operations. In fact, there is a clear tradeoff between the accuracy of the estimated worst-case voltage drops and the time spent on checking the grid.

VIII. Conclusion

In this paper, we proposed a new framework for vectorless verification of the on-die power grid under an RC model and transient constraints. We showed that transient constraints are essential for checking the validity of candidate sequences of chip operations, each having different current requirements. We also showed that the new framework can potentially be useful for debugging. As compared to the standard vectorless technique, our framework leads to less pessimistic estimations of the voltage drops. Moreover, the runtime of the proposed algorithm was found to be reasonably low making the verification of large scale RC grids possible.

References

[1] T. H. Chen and C. C.-P. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods," in *ACM/IEEE Design Automation Conference (DAC-01)*, Las Vegas, NV, June 18-22 2001, pp. 559–562.

[2] J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "A multigrid-like technique for power grid analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 21, no. 10, pp. 1148–1160, October 2002.

[3] M. Zhao, R. V. Panda, S. S. Sapatnekar, and D. Blaauw, "Hierarchical analysis of power distribution networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 21, no. 2, pp. 159–168, February 2002.

[4] H. Qian, S. R. Nassif, and S. S. Sapatnekar, "Power grid analysis using random walks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 24, no. 8, pp. 1204–1224, August 2005.

[5] B. Boghrati and S. S. Sapatnekar, "Incremental analysis of power grids using backward random walks," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 19, no. 3, pp. 31:1–31:29, June 2014.

[6] H. Zhuang, S.-H. Weng, J.-H. Lin, and C.-K. Cheng, "MATEX: A distributed framework for transient simulation of power distribution networks," in *ACM/IEEE Design Automation Conference (DAC-2014)*, San Francisco, CA, June 1-5 2014, pp. 1–6.

[7] J. Yang, Z. Li, Y. Cai, and Q. Zhou, "PowerRush: efficient transient simulation for power grid analysis," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD-12)*, San Jose, CA, November 5-8 2012, pp. 653–659.

[8] T. Yu and M. D. F. Wong, "PGT_SOLVER: An efficient solver for power grid transient analysis," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD-12)*, San Jose, CA, November 5-8 2012, pp. 647–652.

[9] M. Fawaz and F. N. Najm, "Fast simulation-based verification of RC power grids," in *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE-16)*, Vancouver, Canada, May 15-18 2016, pp. 1182–1187.

[10] ——, "Parallel simulation-based verification of RC power grids," in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI-17)*, Bochum, Germany, July 3-5 2017, pp. 445–452.

[11] D. Kouroussis and F. N. Najm, "A static pattern-independent technique for power grid voltage integrity verification," in *ACM/IEEE Design Automation Conference (DAC-03)*, Anaheim, CA, June 2-6 2003, pp. 99–104.

[12] I. A. Ferzli, F. N. Najm, and L. Kruse, "A geometric approach for early power grid verification using current constraints," in *ACM/IEEE International Conference on Computer-Aided Design (ICCAD-07)*, San Jose, CA, Novermber 5-8 2007, pp. 40–47.

[13] N. H. Abdul Ghani and F. N. Najm, "Fast vectorless power grid verification under an RLC model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 30, no. 5, pp. 691–703, May 2011.

[14] X. Xiong and J. Wang, "Dual algorithms for vectorless power grid verification under linear current constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 30, no. 10, pp. 1469–1482, October 2011.

[15] 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 *International Symposium on Physical Design (ISPD-11)*, Santa Barbara, CA, March 27-30 2011, pp. 159–166.

[16] 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 Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 31, no. 1, pp. 109–120, January 2012.

[17] M. Fawaz and F. N. Najm, "Accurate verification of RC power grids," in *Design, Automation and Test in Europe (DATE-16)*, Dresden, Germany, March 14-18 2016, pp. 814–817.

[18] ——, "Fast vectorless RLC grid verification," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 36, no. 3, pp. 489–502, March 2017.

[19] K. Ma, Z. Li, M. Chen, and X. Wang, "Scalable power control for many-core architectures running multi-threaded applications," in *ACM/IEEE International Symposium on Computer Architecture (ISCA-11)*, San Jose, CA, June 4-8 2011, pp. 449–460.

[20] M. U. K. Khan, M. Shafique, and J. Henkel, "Hierarchical power budgeting for dark silicon chips," in *ACM/IEEE International Syposium on Low Power Electronics and Design (ISLPED-15)*, Rome, Italy, July 22-24 2015, pp. 213–218.

[21] X. Xiong and J. Wang, "Verifying RLC power grids with transient current constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 32, no. 7, pp. 1059–1071, July 2013.

[22] ——, "Vectorless transient power grid verification: a case study with IBM benchmarks," in *IEEE Symposium on Electromagnetic Compatibility & Signal integrity (EMC&SI)*, Santa Clara, CA, March 15-21 2015, pp. 271–276.

[23] Z. Moudallal and F. N. Najm, "Generating circuit current constraints to guarantee power grid safety," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC-15)*, Chiba/Tokyo, Japan, January 19-22 2015, pp. 358–365.

[24] R. Horn and C. Johnson, *Matrix Analysis*, 2nd ed. New York: Cambridge University Press, 2013.

[25] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. SIAM, 2003.

[26] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, "Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate," *ACM Transactions on Mathematical Software (TOMS)*, vol. 35, no. 3, pp. 22:1–22:14, 2008.

[27] T. A. Davis. Suitesparse 4.4.6. [Online]. Available: http://faculty.cse.tamu.edu/davis/suitesparse.html

[28] The MOSEK optimization software. [Online]. Available: http://www.mosek.com