

Power Scheduling With Active *RC* Power Grids

Zahi Moudallal^{ID}, *Student Member, IEEE*, and Farid N. Najm^{ID}, *Fellow, IEEE*

Abstract—Power gating is widely used in large chip design as a way to manage the total power dissipation and avoid overheating. It works by turning OFF the power supply to circuit blocks that are not required to operate in certain operational modes. Many authors have studied the scheduling of chip workload to manage total power and temperature. But power gating also has an impact on the supply voltage levels across the die, because voltage drop is generated in the grid depending on the combination of blocks that are ON. We consider the question of how to manage the chip workload so that supply voltage variations remain within specs. The worst case voltage drop is the result of two things: the power budgets that were allocated to the various circuit blocks during the design process and the combination of blocks that are turned ON in a given operational mode. In this paper, we propose a framework to manage this tradeoff between how many blocks are ON simultaneously and how big the power budgets of the individual blocks are, assuming resistive and capacitive (*RC*) elements in the power grid model. Subject to user guidance, we generate block-level circuit current constraints as well as an implicit binary decision diagram (BDD) that helps identify the safe working modes. If the blocks are designed to respect these constraints, then the BDD can be used during normal operation to check whether a candidate working mode is safe or not.

Index Terms—Binary decision diagram (BDD), current constraints, dark silicon, design objectives, integrated circuits, optimization, power budgets, power distribution network, power scheduling, power-gated design, verification.

I. INTRODUCTION

POWER gating [1]–[3] refers to design techniques that partition the logic circuitry of a chip into functional blocks that may be selectively powered ON or OFF. Modern high-performance chips include very large power delivery networks (PDNs). While the PDN is mostly a passive *RLC* structure, PDNs often also include active devices (e.g., MOSFETs) that implement power gating to allow the supply currents (including leakage) of major circuit blocks to be turned OFF by disconnecting them from the rest of the PDN. Thus, such a circuit block has its own *local grid* (as we call it) that may be cutoff from the rest of the PDN (which we call the *global grid*). We refer to a PDN with active devices as an *active* PDN; otherwise, it is a *passive* PDN.

Depending on what blocks that are ON/OFF, the total power dissipation and temperature may exceed specifica-

tions, so that there is a need to schedule the chip workload (which blocks are ON/OFF) in order to remain within the allowed power/temperature specs. Several authors have looked at this question, including [4]–[6]. But the chip workload also impacts the voltage drop on the grid. Depending on the combination of blocks that are in operation, large amounts of current may flow through the PDN causing excessive voltage variations that put both circuit performance and reliability at risk. Proper design and operation of an active PDN is crucial to ensure supply integrity to the circuit blocks, and so avoid timing and signal integrity problems.

Typically, every block may have multiple *power states*, which may be as simple as high performance, low power, standby, or OFF. We assume that each block can either be turned ON or OFF—this can be easily extended to multipower states and is not a limitation to this paper. If every circuit block is in a certain power state, we say that the chip overall is in a certain *working mode*. If some circuit blocks are transitioning from one power state to another, we say that the chip is in a *transition mode*. A power-gated PDN should be verified under both working and transition modes. In this paper, we focus on analyzing the PDN under different working modes, but we are working to extend this to transition modes.

Several computer-aided design algorithms have been developed over the past decade to efficiently analyze and verify a *passive* PDN. Typically, verification methods require simulating the PDN to determine the voltage drop at every node, given detailed information on the current sources tied to the grid, which represent currents drawn by the underlying circuitry. These simulation-based techniques include [7]–[10]. An alternative power grid verification scheme, such as in [11]–[15], relies on information that may be available at an early stage of the design in the form of current budgets or current constraints. These methods are referred to as vectorless verification and consist of finding the worst case voltage fluctuations at all nodes of the grid under all possible transient current waveforms that satisfy user-specified current constraints. The grid is said to be *safe* if these fluctuations are below user-specified thresholds at all grid nodes.

With active PDNs, this verification becomes very difficult because of the many working modes that the chip can have. For example, a chip with 20 blocks, with two power states (i.e., ON and OFF) each, has over a million working modes. A brute-force approach would require exhaustive transient simulation under all possible working modes, each covering a very large number of clock cycles to capture the dynamics of the circuit. Zhu *et al.* [16] proposed an efficient transient analysis approach of the PDN exploiting localized voltage variations near the active blocks. Such an approach requires

Manuscript received May 28, 2018; revised August 28, 2018; accepted September 28, 2018. Date of publication November 15, 2018; date of current version January 23, 2019. This work was supported by the Natural Sciences and Engineering Council of Canada. (*Corresponding author: Zahi Moudallal.*)

The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: zahi.moudallal@mail.utoronto.ca; f.najm@utoronto.ca).

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.2018.2877107

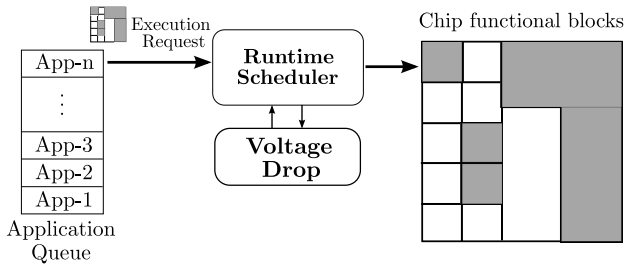


Fig. 1. Conceptual system-level representation of the proposed runtime workload scheduler in a power-gated chip. This figure was inspired from [18].

full knowledge of the current waveforms drawn by every logic block attached to the grid. Thus, it does not allow for *early* grid verification, when grid modifications can be most easily incorporated. Furthermore, the number of current traces needed to cover the space of voltage drops exhibited on the grid is intractable for modern designs. Zeng *et al.* [17] proposed a technique to drastically reduce the number of full simulations by modeling the local grids as switchable current sources. Assuming that the current waveforms representing the currents drawn by the underlying circuitry are available, the method determines an approximate set of working modes that generates the largest average current from the block's power taps. Then, the full grid is simulated under this set of working modes for hundreds of clock cycles. A major problem in this paper is that the worst case working modes are determined based on the currents rather than the voltage drop.

Typically, in a large die, one cannot have all the circuit blocks turned ON simultaneously, so that there will always be some circuit blocks that are turned OFF (so-called *dark silicon*). During normal chip operation, there is a need to manage the workload so that voltage variations remain within specs. The chip will, therefore, include a design component (a scheduler) to manage the workload of the active PDN, leading to a safe schedule of workload. In Fig. 1, we show a conceptual representation of the chip workload scheduler. This chip component monitors the on-chip hardware resources required to execute an incoming application issued by the application repository to ensure that the voltage variations remain within specs. Developing a scheduler requires, at the very least, up-front analysis to identify elements or patterns of workload that represent safe operation; this is a key problem that is addressed in this paper.

In active PDNs, the worst case voltage drop is the result of two things: the power budgets that were allocated to the various circuit blocks during the design process and the combination of blocks that are turned ON in a given working mode. Intuitively, more blocks can be turned ON simultaneously if the blocks are constrained to have low current levels, and vice versa. In this paper, we propose a framework to manage this tradeoff between how many blocks are ON simultaneously and how big the power budgets of the individual blocks are. We focus on *RC* power grids, but we are working to extend this to the *RLC* case. Subject to user guidance, we generate block-level circuit current constraints that identify the

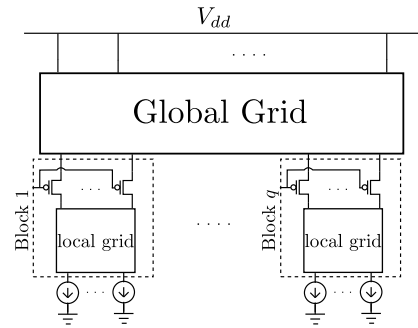


Fig. 2. Schematic of an active PDN with power-gating transistors.

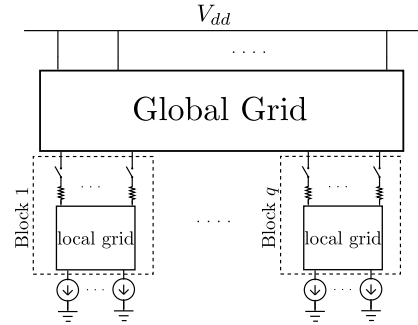


Fig. 3. Schematic of a power-gated PDN using resistive switches, referred to as the original grid.

allowable transient current waveforms for the underlying logic blocks as well as identify the *safe* working modes that the grid can safely support. These working modes are captured in a form of an implicit binary decision diagram (BDD). An on-chip runtime schedule can then use the BDD as a query engine to check whether a candidate working mode is safe or not.

A preliminary version of this paper has appeared in [19]. The proofs for all theoretical results in this paper, except for Lemma 3, are not shown due to lack of space. The rest of this paper is organized as follows. Section II provides an overview of our approach. In Section III, we describe the passive power grid model and give a brief review of the constraints' generation problem introduced in [20]. We then present a detailed description of our proposed method and the bulk of our theoretical contribution in Section IV. In Section V, we give two algorithms that generate block-level current constraints and the corresponding BDD. In Section VI, we present some test results, based on our implementation of these algorithms, and describe the various tradeoffs each algorithm provides. Finally, we give concluding remarks in Section VII.

II. OVERVIEW

In a power-gated design, functional blocks have their own local grids that are connected to the global grid via wide multifingered transistors, referred to as sleep transistors or power-gating switches. A schematic of a power-gated PDN is shown in Fig. 2. Typically, a power-gating transistor may be modeled as an ideal switch in series with a resistor, as in Fig. 3. We will refer to the PDN model in Fig. 3 as the *original grid*.

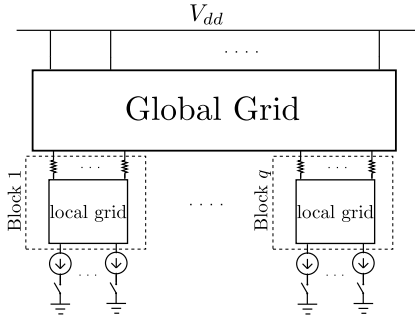


Fig. 4. Schematic of the equivalent passive grid.

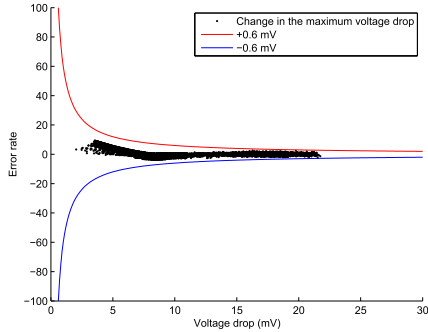


Fig. 5. Relative error of the maximum voltage drop using original grid versus the equivalent passive grid based on the HSPICE simulations of a 400k nodes' grid with 49 blocks.

Verifying the original grid for voltage drop is difficult because of the large number of working modes that the grid can have. A brute-force approach would be to verify the passive PDN corresponding to every possible working mode. Clearly, this method is prohibitively expensive as it requires the verification of an exponential number of passive PDNs, corresponding to the exponential number of possible working modes. Instead, in this paper, we verify a slightly simplified model of the grid, which we call the *equivalent passive grid*, as shown in Fig. 4. The simplification consists of simply moving the switches down to the bottom of the grid, as shown in the figure. The key benefit of this simplification is that as a result, as we will see in Section IV-B, the voltage integrity verification of the equivalent passive grid requires only one verification “run” for each local grid in isolation, combined by means of a type of *superposition* in order to identify the set of safe working modes for the full grid.

These benefits of using the equivalent passive grid come with a very small accuracy cost. Fig. 5 shows the relative error (below ± 0.6 mV) in the maximum voltage drop on the “nodes of interest” of the ON blocks, resulting from using the equivalent passive grid instead of the original grid. Here, and throughout this paper, the “nodes of interest” are the bottom-most nodes of the local grids that are tied directly to the underlying chip circuitry, i.e., to the current sources shown in the figures. Clearly, these are the only nodes whose voltage drop “matters” because they directly affect circuit operation.

In this paper, we will use the notion of a *current container*, introduced in [20], to capture the block-level power budgets.

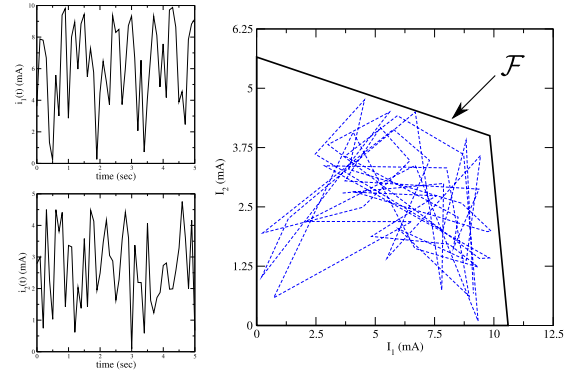
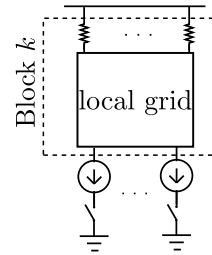
Fig. 6. Example of a current container \mathcal{F} for $i_1(t)$ and $i_2(t)$.

Fig. 7. Block in isolation.

A container is usually expressed as a set of constraints on the currents drawn by the underlying logic circuitry. Fig. 6 shows the idea of a container for a simple case of two current waveforms. Because the trace of these current waveforms belongs to the polygon \mathcal{F} , for all time instants, we say that \mathcal{F} contains $i(t) = [i_1(t) \ i_2(t)]^T$.

Taken in isolation, a block (local grid), as shown in Fig. 7, can be analyzed separately using the inverse problem (constraints’ generation) approach for passive grids [20] to give a container (or set of containers) that respects the maximum allowable voltage drop, referred to as a *voltage drop threshold*, at all the nodes of interest in the block; this approach will be reviewed in Section III. Because we expect lower levels of the grid to have less than ideal voltages, suppose that the supply value applied at every block’s power taps is parameterized by an artificial variable α . Specifically, for a block k with uniform voltage drop threshold at all its nodes of interest, i.e., the nodes of interest in that block have the same voltage drop threshold γ_k , suppose the supply value is $V_{dd} - (1 - \alpha_k)\gamma_k$. There is no need to actually relate this supply value to any actual supply value that the full chip may experience at certain layers. In fact, we will see that these variables $\alpha_1, \alpha_2, \dots$, and α_q (corresponding to block 1, ..., block q) can be viewed as parameters that become “knobs” of sorts by which we can have the local containers that expand when the supply voltage is increased or contract when it is decreased. The safety of these containers is not assumed based on the choice of $\alpha_1, \alpha_2, \dots$, and α_q . Rather, safety will be enforced as part of the subsequent analysis of the full grid, from which we will capture the set of safe working modes of the grid, represented by a set of safe assignments of a Boolean vector β

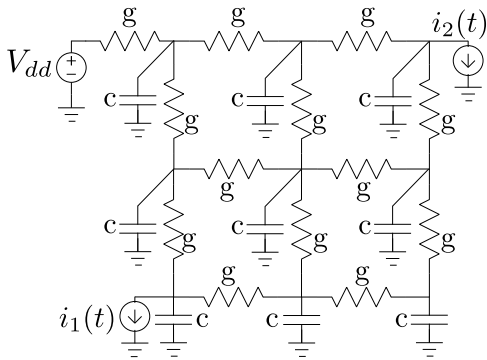


Fig. 8. Passive RC grid model.

corresponding to any $\alpha_1, \alpha_2, \dots$, and α_q . This safe space of β will be captured with a BDD.

III. BACKGROUND FOR PASSIVE GRIDS

In this section, we describe a passive power grid model that will be used throughout this paper and we review some key theoretical results that were established for the constraints' generation approach for passive power grids [20]. The results of this section apply to any passive grid and will be invoked to describe the power grid of each block in isolation as well as the full grid. Thus, for ease of extension and to avoid repetition, we will define a passive power grid “problem” $\mathcal{P}(\cdot)$ that includes the description of the grid model in Section III-A and the results presented in Sections III-B and III-C.

A. Passive Power Grids

Consider an RC model of a passive power grid. Some nodes of the top level layers of the grid may be connected to ideal voltage sources representing the connection to the external voltage supply V_{dd} . Assuming flip-chip technology, we will refer to an ideal supply voltage source as a C4 with the understanding that any parasitics that are part of a true C4 pad structure have already been modeled and included in the grid description. Note that, in this paper, we assume that a C4 pad is modeled with resistive and capacitive components only, because we focus on RC power grids. Some nodes of the bottom-most layers have ideal current sources (to ground) representing the currents drawn by the logic circuits tied to the grid. There exists also a capacitor from every grid node to ground. We assume that there are no node-to-node capacitors in the grid.

Excluding the ground node, let the power grid consist of $n + s$ nodes, where nodes $1, 2, \dots, n$ are the nodes not connected to a voltage source, while the remaining nodes $(n + 1), (n + 2), \dots, (n + s)$ are the nodes where the s voltage sources are connected. Let $i(t)$ be the nonnegative vector of all the m current sources connected to the grid, whose positive (reference) current direction is from node to ground. Let H be an $n \times m$ matrix of 0 and 1 entries that identifies (with a 1) which node is connected to which current source, and let $i_s(t) = Hi(t)$. Fig. 8 is an example of an RC grid model.

Let $v(t)$ be the $n \times 1$ vector of time-varying voltage drops (difference between V_{dd} and the true node voltages). We can

write the RC model for the power grid using nodal analysis, as [11]

$$Gv(t) + C\dot{v}(t) = i_s(t) \quad (1)$$

where C is an $n \times n$ diagonal nonnegative capacitance matrix, which is nonsingular because every node is attached to a capacitor; G is the $n \times n$ conductance matrix, which is known to be symmetric and diagonally dominant with positive diagonal entries and nonpositive off-diagonal entries. With this, it can be shown that G is a so-called \mathcal{M} -matrix, so that G^{-1} exists and is nonnegative, $G^{-1} \geq 0$, i.e., its every entry is nonnegative.

Using a finite-difference approximation for the derivative, such as a backward Euler scheme $\dot{v}(t) \approx (v(t) - v(t - \Delta t)) / \Delta t$, the grid system model (1) leads to

$$v(t) = A^{-1}Bv(t - \Delta t) + A^{-1}Hi(t) \quad (2)$$

where $B = C/\Delta t$ is an $n \times n$ diagonal matrix with $b_{ii} > 0$, $\forall i$, and $A = G + B$. It can also be shown that A , just like G , is an \mathcal{M} -matrix, so that $A^{-1} \geq 0$. Let $M = A^{-1} \geq 0$ and define the $n \times m$ matrix $M' = MH \geq 0$.

We assume that a certain number of grid nodes $d \leq n$ (the “nodes of interest”) are required to satisfy certain user-provided *voltage drop threshold specifications*, captured in the $d \times 1$ vector $V_{th} \geq 0$. These would typically be nodes at the lower metal layers, where the chip circuitry is connected. Thus, we assume that these nodes are internal to the blocks. Let P be a $d \times n$ matrix consisting of 0 and 1 elements only, specifying (with a 1 entry) the nodes that are subject to a voltage drop threshold specification. Note that $P \geq 0$ and has exactly one 1 entry in every row, otherwise 0s, and that no column of P has more than a single 1 entry.

With this, let $\mathcal{P}(n, m, d, G, C, H, P)$ denote a passive power grid problem as described above.

B. Safe Containers

For completeness of presentation, we review some terminology introduced in [20] that is crucial to this paper. The following definition introduces the notion of a container for a vector of current waveforms, which will help us express constraints that guarantee grid safety.

Definition 1 (Container): Let $t \in \mathbb{R}$, let $i(t) \in \mathbb{R}^m$ be a function of time, and let $\mathcal{F} \subset \mathbb{R}^m$ be a closed subset of \mathbb{R}^m . If $i(t) \in \mathcal{F}, \forall t \in \mathbb{R}$, then we say that \mathcal{F} contains $i(\cdot)$, represented by the shorthand $i(\cdot) \subset \mathcal{F}$, and we refer to \mathcal{F} as a container of $i(\cdot)$.

Definition 2 (Safe Grid): A grid is said to be *safe* for a given function $i(t)$, defined $\forall t \in \mathbb{R}$, if the corresponding $Pv(t) \leq V_{th}, \forall t \in \mathbb{R}$.

To check if a power grid is safe, one would typically be interested in the worst case voltage drop at some grid node k , at some time point $\tau \in \mathbb{R}$, over a wide range of possible current waveforms. Using the above notation, and given a container \mathcal{F} that contains a wide range of current waveforms of interest, we can express this as $\max_{i(\cdot) \subset \mathcal{F}} (v_k(\tau))$. Clearly, because \mathcal{F} is the same irrespective of time and applies at all time points $t \in \mathbb{R}$, then this worst case voltage drop must be

time-invariant, independent of the chosen time point τ . Therefore, one way to check grid safety is to compute the worst case voltage drop attained by each component of $v(t)$, denoted as $v^*(\mathcal{F}) = \text{emax}_{i(\cdot) \in \mathcal{F}}(v(\tau))$, where the “ $\text{emax}(\cdot)$ ” notation denotes *elementwise* maximization, as in [20]. Najm [11] provides an exact expression for the worst case voltage drop $v^*(\mathcal{F})$ that requires an infinite sum of $\text{emax}(\cdot)$ operations. Thus, requiring the exact $v^*(\mathcal{F})$ is prohibitively expensive and so we will instead use an upper bound on $v^*(\mathcal{F})$ based on the following.

Definition 3: For any $\mathcal{F} \subset \mathbb{R}^m$, define

$$\bar{v}(\mathcal{F}) \triangleq G^{-1} A \text{emax}_{I \in \mathcal{F}}(M'I) \quad (3)$$

with the convention that $\text{emax}_{I \in \mathcal{F}}(M'I) = 0$, if $\mathcal{F} = \emptyset$.

Note that, in (3), $I \in \mathbb{R}^m$ is a vector of artificial variables, with units of current, that is used to carry out the $\text{emax}(\cdot)$ operation.

In [11], it has been shown that $\bar{v}(\mathcal{F})$ is an upper bound on $v^*(\mathcal{F})$

$$v^*(\mathcal{F}) \leq \bar{v}(\mathcal{F}) \quad \forall \mathcal{F} \subset \mathbb{R}^m. \quad (4)$$

Furthermore, Fawaz and Najm [21] show that, for a certain range of the discretization time-step Δt , the accuracy of this upper bound relative to $v^*(\mathcal{F})$ is quite good.

Definition 4 (Safe Container): A container \mathcal{F} is said to be *safe* if $P\bar{v}(\mathcal{F}) \leq V_{\text{th}}$.

Thus, a safe container \mathcal{F} is useful because, due to (4), it guarantees that $Pv^*(\mathcal{F}) \leq V_{\text{th}}$, so that the grid is safe for that container. A safe container \mathcal{F} can be expressed as a set of constraints on the circuit currents that load the grid, thereby providing a set of linear current constraints that are sufficient to guarantee grid safety. In previous work [11], current containers were *specified* and the corresponding worst case voltage drop was found by a process of optimization. In later work [20], these containers were *generated* for passive grids so that, if the circuit is designed to respect these constraints, the grid becomes safe by design. In this paper, we build on and extend the work of [20] to the case of active grids. Some of the major results in [20] are restated below, as they are necessary to understand the flow of this paper.

C. Maximal Containers

Let $u \in \mathbb{R}^n$ and define the sets \mathcal{U} , $\mathcal{F}(u)$, and \mathcal{S} as follows:

$$\mathcal{U} \triangleq \{u \in \mathbb{R}^n : u \geq 0, Pu \leq V_{\text{th}}\} \quad (5)$$

$$\mathcal{F}(u) \triangleq \{I \in \mathbb{R}^m : I \geq 0, M'I \leq MG u\} \quad (6)$$

$$\mathcal{S} \triangleq \{\mathcal{F}(u) : u \in \mathcal{U}\} \quad (7)$$

where \mathcal{U} is effectively a set of safe voltage drop assignments u , $\mathcal{F}(u)$ is a special kind of container constructed based on $u \in \mathcal{U}$, and \mathcal{S} is the set of all containers $\mathcal{F}(u)$ corresponding to $u \in \mathcal{U}$. It turns out that it is enough to consider only containers of the form (6) due to the following *necessary and sufficient condition*.

Lemma 1 [20]: A container $\mathcal{J} \subset \mathbb{R}_+^m$ is safe if and only if it is a member of \mathcal{S} or a subset of a member of \mathcal{S} .

The importance of this lemma is twofold: 1) $\mathcal{F}(u)$ is safe for any $u \in \mathcal{U}$ and 2) all interesting safe containers \mathcal{J} may be found as either specific $\mathcal{F}(u)$ for some $u \in \mathcal{U}$ or as subsets of such $\mathcal{F}(u)$. Moudallal and Najm [20] show that if $V_{\text{th},k} = 0$, for some k , then the only nonempty container in \mathcal{S} is the trivial one $\mathcal{F}(0) = \{0\}$. Therefore, throughout this paper, we will assume that $V_{\text{th}} > 0$.

Note that, if $\mathcal{J} \subseteq \mathcal{F}(u)$, for some $u \in \mathcal{U}$, with $\mathcal{J} \neq \mathcal{F}(u)$, then clearly $\mathcal{F}(u)$ is a better choice than \mathcal{J} . Choosing \mathcal{J} would be unnecessarily limiting, while $\mathcal{F}(u)$ would allow more flexibility in the circuit loading currents. Therefore, it is enough to consider only containers of the form $\mathcal{F}(u)$ with $u \in \mathcal{U}$. Going further, if $\mathcal{F}(u_1) \subseteq \mathcal{F}(u_2)$ with $\mathcal{F}(u_1) \neq \mathcal{F}(u_2)$, then clearly $\mathcal{F}(u_2)$ is a better choice than $\mathcal{F}(u_1)$. Thus, in a sense, the “larger” the container, the better, because it allows flexibility to the underlying logic blocks. Therefore, we are interested in safe containers that are not fully contained in any other safe container. These containers are referred to as *maximal* containers.

IV. PROPOSED APPROACH—THEORY

Given the equivalent passive model in Fig. 4, our approach consists of two stages: 1) we perform isolated block analysis to generate block-level current containers by adapting the standard inverse problem (constraints generation) approach introduced in [20]—this will be discussed in Section IV-A; and 2) these block-level containers will then be used to identify the behavioral patterns of the whole chip that are safe based on the voltage analysis of the full grid, which we capture as an implicit BDD—this will be discussed in Section IV-B. Our approach uses an internal parameter α_k for every block k . These parameters become “knobs” of sorts by which we can have these block-level containers expand or contract, and in turn, the BDD will either allow for less or more blocks to operate simultaneously.

This section includes the bulk of our theoretical contribution, culminating in the result of Theorem 1 that leads to the computational efficiency of our approach. This theorem follows from the results of Lemma 5, which establishes a scalability property for the upper bound on the worst case voltage drop in terms of the internal parameters $\alpha_1, \alpha_2, \dots$, and α_q , and Lemma 6, which establishes the principle of superposition for the equivalent passive grid. In addition, we show that the block-level current containers (in Lemma 3) also have a scalability property in terms of these internal parameters. These results allow us to easily manage the tradeoff between the power budgets of the blocks and the number of blocks that are ON simultaneously. Throughout the rest of this section, we will refer to the example in Fig. 9 to help the reader better understand our approach.

A. Isolated Block Analysis

In this section, we prove some key results that are applicable to any passive grid, and thus will be used for every block in isolation. Every block k has a uniform voltage drop threshold γ_k and an internal parameter α_k . Throughout the remainder of this section, we will omit the subscript k for notational simplicity as we are considering a block in isolation.

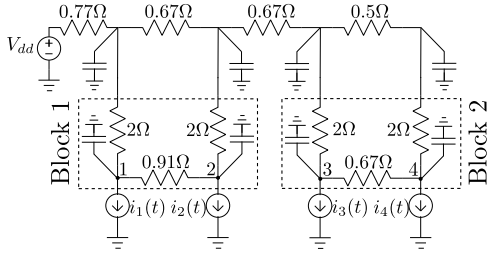
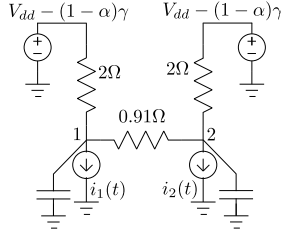
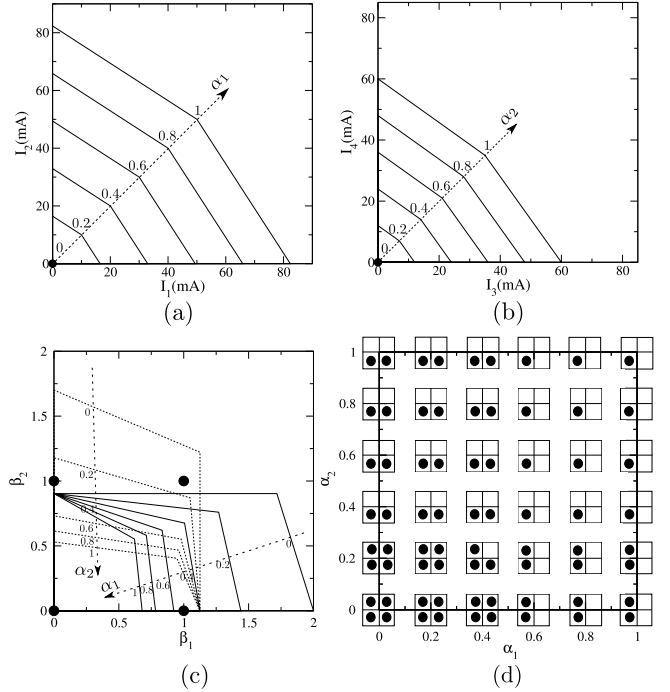


Fig. 9. Simple example of a power grid with two blocks.


 Fig. 10. Simple example of a power grid with a supply value of $V_{dd} - (1 - \alpha)\gamma$.

1) *Safety Condition:* Grid safety relates to the *voltage drop* at every node, i.e., the difference between the *ideal* supply voltage value V_{dd} and the true node voltage, denoted $\hat{v}_i(t)$ at every node i . Note that the voltage drop $V_{dd} - \hat{v}_i(t)$ is relative to the ideal V_{dd} , and that when we say that node i has a user-specified voltage drop threshold γ , we implicitly mean that γ is the threshold relative to V_{dd} , so that the node is safe if $V_{dd} - \hat{v}_i(t) \leq \gamma$. For a block in isolation, and because we expect lower levels of the grid to have less than ideal voltages, suppose that its power taps are connected to a parameterized ideal voltage supply of $V_{dd} - (1 - \alpha)\gamma$, with $0 \leq \alpha \leq 1$, as shown in the example in Fig. 10. When $\alpha = 1$, this supply value is V_{dd} and it decreases all the way to $V_{dd} - \gamma$ for $\alpha = 0$. For any node i in that block, $[V_{dd} - (1 - \alpha)\gamma] - \hat{v}_i(t)$ is the voltage drop relative to $V_{dd} - (1 - \alpha)\gamma$, and it is easy to see that the node safety condition $V_{dd} - \hat{v}_i(t) \leq \gamma$ is equivalent to $[V_{dd} - (1 - \alpha)\gamma] - \hat{v}_i(t) \leq \alpha\gamma$. Thus, the voltage drop threshold relative to the supply value $V_{dd} - (1 - \alpha)\gamma$ is simply $\alpha\gamma$. It is in this sense that the α parameter is simply a “knob” that, when reduced, exerts a more stringent safety conditions on grid nodes, which would naturally result in a smaller container for the local blocks, allowing more blocks to be turned ON simultaneously and vice versa. This α becomes an internal parameter that represents the tradeoff between the sizes of local grid containers and the number of full grid working modes that will be deemed to be safe.

We can then easily extend and rederive the theory of the passive grids from Section III so that it is parameterized by $0 \leq \alpha \leq 1$. Consider the generic passive power grid problem, denoted earlier as $\mathcal{P}(n, m, d, G, C, H, P)$, which we will apply to an isolated block. We assume that the voltage drop threshold specification is uniform within every block, i.e., all the “nodes of interest” in that block have the same voltage drop threshold $\gamma > 0$, relative to V_{dd} . We capture this by the $d \times 1$ vector $\gamma \mathbf{1}_d$, where $\mathbf{1}_d$ is a $d \times 1$ vector whose every entry is 1. Assuming that the power taps of the isolated passive grid are connected to an ideal voltage source


 Fig. 11. (a) Current container $\mathcal{F}_1(\alpha_1)$ for the left block in Fig. 9 for different values of α_1 . (b) Current container $\mathcal{F}_2(\alpha_2)$ for the right block in Fig. 9 for different values of α_2 . (c) Set of safe working modes $\mathcal{W}(\alpha)$ for different values of $\alpha = [\alpha_1 \alpha_2]^T$ under the containers generated for each block in isolation, i.e., $\mathcal{F}_1(\alpha_1)$ and $\mathcal{F}_2(\alpha_2)$. The dashed polygons correspond to $\alpha_1 = 0.4$ and for different values of α_2 and the solid polygons correspond to $\alpha_2 = 0.4$ and for different values of α_1 . (d) Set $\mathcal{W}(\alpha)$ for different values of α .

of $V_{dd} - (1 - \alpha)\gamma$, let $v(t)$ be the vector of voltage drops relative to $[V_{dd} - (1 - \alpha)\gamma]$ at all nodes in the block, then as we saw above, a safe voltage drop assignment for the block in isolation must satisfy

$$Pv(t) \leq \alpha\gamma \mathbf{1}_d. \quad (8)$$

For any $\alpha \in [0, 1]$, define the sets $\mathcal{U}(\alpha)$, $\mathcal{L}(u)$, and $\mathcal{S}(\alpha)$ as follows, motivated by (8):

$$\mathcal{U}(\alpha) \triangleq \{u \in \mathbb{R}^n : 0 \leq Pu \leq \alpha\gamma \mathbf{1}_d\} \quad (9)$$

$$\mathcal{L}(u) \triangleq \{I \in \mathbb{R}^m : I \geq 0, M'I \leq MG u\} \quad (10)$$

$$\mathcal{S}(\alpha) \triangleq \{\mathcal{L}(u) : u \in \mathcal{U}(\alpha)\}. \quad (11)$$

Lemma 2 shows that, for any $\alpha > 0$, $\mathcal{S}(\alpha)$ always has a current container that allows a nonzero current. This will be useful later on.

Lemma 2: For any $\alpha \in (0, 1]$, $\mathcal{S}(\alpha)$ always has a nonempty member $\mathcal{L}(u)$ with $\mathcal{L}(u) \neq \{0\}$.

2) *Scalability of Current Containers:* Moudallal and Najm [20] proposed several algorithms for passive grids that generate a container $\mathcal{L}(u) \subseteq \mathbb{R}_+^m$ that is both safe and maximal. These algorithms target specific design objectives, such as the total peak power that a grid can safely support, the uniformity of current distribution across the die area, or a combination of both objectives. The peak power algorithm in [20], once extended and parameterized by α as above, then applied to the grid in Fig. 10, for different values of α , generates the current containers shown in Fig. 11(a). Generating current containers

for different values of α requires solving an optimization problem for every required value of α , which is computationally expensive. In this section, we show that, under a certain mild condition on the design objective, the resulting containers can be found by “scaling” the container corresponding to $\alpha = 1$, as we will see in Lemma 3, which is clearly much faster than generating the containers for every required value of α .

Typically, these algorithms, such as in [20], can be expressed in the following general form:

$$\max_{u \in \mathcal{U}(\alpha)} \left(\max_{I \in \mathcal{L}(u)} f(I, u) \right) \quad (12)$$

where $f(I, u) : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}$ is some real-valued objective function. For example, the peak power algorithm in [20] can be expressed in the form of (12) where $f(I, u) = \sum_{\forall j} I_j$. Notice that, for any $u \in \mathbb{R}^n$, the inner maximization finds the maximum value of $f(I, u)$ over all possible current assignments $I \in \mathcal{L}(u)$. Thus, the result of the inner maximization is a function of u , denoted as

$$g(u) = \max_{I \in \mathcal{L}(u)} f(I, u) \quad (13)$$

and referred to as the design objective. The largest $g(u)$ achievable over all possible safe voltage drop assignments $u \in \mathcal{U}(\alpha)$ is found using the outer maximization, the result of which is a function of α , denoted as $g^*(\alpha)$, i.e.,

$$g^*(\alpha) = \max_{u \in \mathcal{U}(\alpha)} g(u) = \max_{\substack{I \in \mathcal{L}(u) \\ u \in \mathcal{U}(\alpha)}} f(I, u). \quad (14)$$

For any $\alpha \in [0, 1]$, let $u^*(\alpha)$ be a vector function that evaluates to a value of u for which the outer maximization attains its maximum, i.e., $g(u^*(\alpha)) = g^*(\alpha)$, $\forall \alpha \in [0, 1]$. In general, $u^*(\alpha)$ may not be unique. The vector $u^*(\alpha)$ produced in (14) can be used to construct the current container $\mathcal{L}(u^*(\alpha))$, where $\mathcal{L}(\cdot)$ is defined in (10). Note that the optimization problem (14) is always feasible, because $0 \in \mathcal{U}(\alpha)$ and $0 \in \mathcal{L}(0)$, so that $u^*(\alpha)$ is well defined and the resulting container $\mathcal{L}(u^*(\alpha))$ is nonempty.

Lemma 3 is a key theoretical result that gives a sufficient condition under which $\mathcal{L}(u^*(\alpha))$ for any supply value $V_{dd} - (1 - \alpha)\gamma$ can be found by simply scaling $u^*(1)$ to get $u^*(\alpha)$, which will then be used to construct $\mathcal{L}(u^*(\alpha))$ as in (10). This will be useful for the full grid analysis.

Lemma 3: If $g(cu) = cg(u)$, for any real number $c > 0$ and $u \in \mathbb{R}^n$, then $u^*(\alpha) = \alpha u^*(1)$, $\forall \alpha \in [0, 1]$.

Proof: Recall that for any $\alpha \in [0, 1]$, $g^*(\alpha)$ can be found using the following optimization problem:

$$\begin{aligned} g^*(\alpha) = \text{Max } & g(u) \\ \text{s.t. } & Pu \leq \alpha\gamma \mathbb{1}_d \\ & u \geq 0 \end{aligned} \quad (15)$$

where $u \in \mathbb{R}^n$ is a vector of artificial variables with the units of volts that is used to carry out the above maximization.

Notice that if $\alpha = 0$, then the constraints of the optimization problem (15) become $Pu \leq 0$ and $u \geq 0$; because $P \geq 0$ and has exactly one 1 in each row, it follows that $u = 0$ is the only vector satisfying those constraints. Also, recall that for any $\alpha \in [0, 1]$, $u^*(\alpha)$ is defined to be a vector function that

evaluates to a value of u for which (15) attains its maximum. It follows that $u^*(\alpha) = 0 = \alpha u^*(1)$, and the last step is due to $\alpha = 0$.

Consider the case where $\alpha > 0$. Using the following change of variable:

$$u = \alpha u' \quad (16)$$

we can rewrite (15) as

$$\begin{aligned} g^*(\alpha) = \text{Max } & g(\alpha u') \\ \text{s.t. } & P\alpha u' \leq \alpha\gamma \mathbb{1}_d \\ & \alpha u' \geq 0. \end{aligned} \quad (17)$$

Notice that $\alpha > 0$, so that $g(\alpha u') = \alpha g(u')$, because $g(cu') = cg(u')$, for any $c > 0$. Furthermore, $\alpha u' \geq 0$ is equivalent to $u' \geq 0$, and $P\alpha u' \leq \alpha\gamma \mathbb{1}_d$ is equivalent to $Pu' \leq \gamma \mathbb{1}_d$. With this, we can rewrite (17) as follows:

$$\begin{aligned} g^*(\alpha) = \text{Max } & \alpha g(u) \\ \text{s.t. } & Pu \leq \gamma \mathbb{1}_d \\ & u \geq 0. \end{aligned} \quad (18)$$

It follows that $g^*(\alpha) = \alpha g^*(1)$.

Let $\bar{u} = \alpha u^*(1) \geq 0$, because $\alpha > 0$ and $u^*(1) \geq 0$. Notice that

$$P\bar{u} = \alpha Pu^*(1) \leq \alpha\gamma \mathbb{1}_d \quad (19)$$

the second step due to $Pu^*(1) \leq \gamma \mathbb{1}_d$ and $\alpha > 0$, so that $\bar{u} \in \mathcal{U}(\alpha)$, and

$$g^*(\alpha) = \alpha g^*(1) = \alpha g(u^*(1)) = g(\alpha u^*(1)) \quad (20)$$

where in the last step we used the fact that $cg(u) = g(cu)$, for any $c > 0$ and $u \in \mathbb{R}^n$. Thus, $g^*(\alpha) = g(\bar{u})$. Therefore, we can let $u^*(\alpha) = \bar{u} = \alpha u^*(1)$ and the proof is complete. ■

Thus, for any $\alpha \in [0, 1]$, we have

$$\mathcal{L}(u^*(\alpha)) = \{I \geq 0 : M'I \leq \alpha MGu^*(1)\}. \quad (21)$$

It can be shown that the design objectives used in [20] satisfy the condition of the above lemma, so that the condition of the lemma is indeed mild and practical, leading to the above very useful scalability property. Referring to the grid in Fig. 10, the peak power algorithm in [20] for $\alpha = 1$ gives $u^*(1) = [100 \ 100]^T$ mV. Thus, for $\alpha = 0.6$, we immediately have $u^*(\alpha) = \alpha u^*(1) = [60 \ 60]^T$ mV. This gives us a scaled container $\mathcal{L}(u^*(\alpha))$, so that for any $\alpha \in [0, 1]$, we have

$$\mathcal{L}(u^*(\alpha)) = \left\{ \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \geq 0 : \begin{bmatrix} 1.01 & 0.65 \\ 0.65 & 1.01 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \leq \alpha \begin{bmatrix} 0.084 \\ 0.084 \end{bmatrix} \right\}.$$

B. Full Grid Analysis

In this section, we apply the results of Section IV-A to every block of the grid. Every block k has its own current container that has the above scalability property in terms of its parameter α_k . The importance of this section is twofold: 1) we show that the worst case voltage drop contribution at the nodes of interest in the full grid due to the activity of each individual block k also has a scalability property in terms of α_k , as presented in Lemma 5; and 2) we show that the upper

bound on the worst case voltage drop on the nodes of interest in the full grid due to the activity of a set of blocks is equal to the sum of the individual contributions of each block in that set, as presented in Lemma 6. Thus, an upper bound on the worst case voltage drop contribution on the nodes of interest in the full grid due to the activity of a set of blocks for some value of their internal parameters can be simply found by adding the scaled contribution of every block k in that set for $\alpha_k = 1$, culminating in the result of Theorem 1. For example, consider the example of Fig. 9 and suppose that an upper bound on the worst case voltage drop at node 1 due to the activity of block 1 for $\alpha_1 = 1$ is 189 mV. Also, an upper bound on the worst case voltage drop at node 1 due to the activity of block 2 for $\alpha_2 = 1$ is 71 mV. Then, an upper bound on the worst case voltage drop at node 1 due to the activity of both blocks, for $\alpha_1 = 0.5$ and $\alpha_2 = 0.25$, is simply $189 \times 0.5 + 71 \times 0.25 = 112.25$ mV.

1) *Definitions:* In isolation, each block is a separate passive power grid, and $\mathcal{P}(n_k, m_k, d_k, G_k, C_k, H_k, P_k)$ denotes its passive grid problem. Furthermore, let $B_k = C_k / \Delta t_k$ be the $n_k \times n_k$ capacitance matrix resulting from the backward Euler numerical integration scheme on block k , so that $A_k = G_k + B_k$. Also, let $M_k = A_k^{-1} \geq 0$ and $M'_k = M_k H_k$.

We assume that the voltage drop threshold specification is uniform within a block, so that all nodes of interest within the same block have the same threshold specification, i.e., $V_{th,k} = \gamma_k \mathbb{1}_{d_k}$, where $\gamma_k > 0$ and $\mathbb{1}_{d_k}$ is a $d_k \times 1$ vector of ones. This assumption does not limit this paper but allows for several scalability properties, as we will see below, that lead to the computational efficiency of our approach.

For every block k in isolation, let u_k be a voltage drop assignment [relative to $(V_{dd} - (1 - \alpha_k)\gamma_k)$] at all nodes in block k . For every isolated block k and for any $\alpha_k \in [0, 1]$, define the sets $\mathcal{U}_k(\alpha_k)$, $\mathcal{L}_k(u_k)$, and $\mathcal{S}_k(\alpha_k)$, based on the analysis in Section IV-A, as follows:

$$\mathcal{U}_k(\alpha_k) \triangleq \{u_k \in \mathbb{R}^{n_k} : 0 \leq P_k u_k \leq \alpha_k V_{th,k}\} \quad (22)$$

$$\mathcal{L}_k(u_k) \triangleq \{I_k \in \mathbb{R}^{m_k} : I_k \geq 0, M'_k I_k \leq M_k G_k u_k\} \quad (23)$$

$$\mathcal{S}_k(\alpha_k) \triangleq \{\mathcal{L}_k(u_k) : u_k \in \mathcal{U}_k(\alpha_k)\}. \quad (24)$$

For every $\alpha_k \in [0, 1]$ and for any $u_k \in \mathcal{U}_k(\alpha_k)$, let $g_k(u_k)$ be a design objective for block k satisfying the conditions of Lemma 3, and let $g_k^*(\alpha_k)$ be defined as follows:

$$g_k^*(\alpha_k) = \max_{u_k \in \mathcal{U}_k(\alpha_k)} g_k(u_k). \quad (25)$$

Let $u_k^*(\alpha_k)$ be a vector function that evaluates to a value of u_k for which the above maximization attains its maximum: $g_k(u_k^*(\alpha_k)) = g_k^*(\alpha_k)$, $\forall \alpha_k \in [0, 1]$. Then, using Lemma 3, $u_k^*(\alpha_k)$ can be expressed as

$$u_k^*(\alpha_k) = \alpha_k u_k^*(1), \quad \forall \alpha_k \in [0, 1]. \quad (26)$$

It is important to note that $u_k^*(\alpha_k)$ depends on the choice of $g_k(u_k)$ so that $\mathcal{L}_k(u_k^*(\alpha_k))$ depends on $g_k(u_k)$ as well. For ease of notation, let $\mathcal{F}_k(\alpha_k) \triangleq \mathcal{L}_k(u_k^*(\alpha_k))$, again keeping in mind that $\mathcal{F}_k(\alpha_k)$ depends on the choice of the design objective $g_k(u_k)$.

During chip design, we can set the internal parameters α_k , $k \in \{1, \dots, q\}$, to ensure the chip currents respect the desired power budgets for the individual blocks. Thus, in the

discussion below, we assume the chip is designed to respect these local containers, so that an ON block draws a current that is consistent with $\mathcal{F}_k(\alpha_k)$, i.e., $I_k \in \mathcal{F}_k(\alpha_k)$, and an OFF block does not draw any current, i.e., $I_k = 0$.

We will use the notation \mathbb{B} and \mathbb{B}^q to denote the Boolean spaces $\mathbb{B} = \{0, 1\}$ and $\mathbb{B}^q = \{0, 1\}^q$. Let $\beta_k \in \mathbb{B}$ denote the mode of operation of block k , i.e., $\beta_k = 1$ if block k is ON, otherwise $\beta_k = 0$. Also, let $\beta = [\beta_1 \dots \beta_q] \in \mathbb{B}^q$ denote a working mode for the chip, and $\alpha = [\alpha_1, \dots, \alpha_q] \in \mathbb{R}^q$ denote a vector, where the k th entry represents the internal parameter for block k . Note that $\alpha_k \in [0, 1]$, $\forall k \in \{1, \dots, q\}$, so that $0 \leq \alpha \leq \mathbb{1}_q$, which will be denoted using the shorthand $\alpha \in [0, \mathbb{1}_q]$. Furthermore, we will use the shorthand $\alpha \in (0, \mathbb{1}_q]$ to denote that $0 < \alpha \leq \mathbb{1}_q$, i.e., $\alpha_k > 0$, $\forall k \in \{1, \dots, q\}$.

Define $\mathcal{F}(\alpha, \beta) \subset \mathbb{R}^m$ as follows:

$$\mathcal{F}(\alpha, \beta) = \left\{ \begin{bmatrix} I_1 \\ \vdots \\ I_q \end{bmatrix} \in \mathbb{R}^m : I_k \in \begin{cases} \mathcal{F}_k(\alpha_k), & \text{if } \beta_k = 1 \\ \{0\}, & \text{if } \beta_k = 0 \end{cases} \right\}. \quad (27)$$

Notice that $\mathcal{F}(\alpha, \beta)$ denotes a current container for all the current sources attached to the grid under the working mode β and for the parameter α .

With this, we can define $v(\alpha, \beta)$ to be an upper bound on the worst case voltage drop experienced by the nodes of interest in the equivalent passive grid under the given α and β , based on the passive grid analysis in (3), as follows:

$$v(\alpha, \beta) = P\bar{v}(\mathcal{F}(\alpha, \beta)) = PG^{-1}A \operatorname{emax}_{I \in \mathcal{F}(\alpha, \beta)}(M'I). \quad (28)$$

Notice that the current vector I that is used to carry out the maximization in (28) has the vector form defined in (27), i.e., its components I_1, I_2, \dots, I_q correspond to the current sources attached to block 1, block 2, ..., and block q . The columns of M' in (28) correspond to the different components of I , so that we can partition M' as follows:

$$M' = [Z_1 \quad Z_2 \quad \dots \quad Z_q] \quad (29)$$

where Z_k is an $n \times m_k$ matrix that is multiplied by I_k in (28).

For any $\alpha \in [0, \mathbb{1}_q]$, let

$$v_k(\alpha_k) \triangleq PG^{-1}A \operatorname{emax}_{I_k \in \mathcal{F}_k(\alpha_k)}(Z_k I_k) \quad (30)$$

and

$$V(\alpha) = [v_1(\alpha_1) \dots v_q(\alpha_q)]. \quad (31)$$

Notice that for any $\alpha \in [0, \mathbb{1}_q]$, we have $I_k \geq 0$, $\forall I_k \in \mathcal{F}_k(\alpha_k)$, and $Z_k \geq 0$, because $M' \geq 0$. Furthermore, we have $G^{-1}A = \mathbb{I}_n + G^{-1}(A - G) = \mathbb{I}_n + G^{-1}B \geq 0$, where \mathbb{I}_n is the $n \times n$ identity matrix, and $P \geq 0$, so that $v_k(\alpha_k) \geq 0$. Therefore, $V(\alpha) \geq 0$, $\forall \alpha \in [0, \mathbb{1}_q]$. Furthermore, Lemma 4 shows that if $\alpha > 0$, then $V(\alpha) > 0$. This will be useful in Section V.

Lemma 4: For any $\alpha \in (0, \mathbb{1}_q]$, we have $V(\alpha) > 0$.

2) *Scalability:* It is expensive to compute $V(\alpha)$ for different values of α , as this would require solving q $\operatorname{emax}(\cdot)$ operations as in (30), i.e., $q \times n$ linear programs (LPs). Lemma 5 shows that, under a certain mild condition on $g_k(\cdot)$, $V(\alpha)$ has a scalability property in terms of α .

For any $q \times 1$ vector x , let $D(x)$ denote the $q \times q$ diagonal matrix with the vector x on the main diagonal, i.e.,

$$D(x) \triangleq \begin{bmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_q \end{bmatrix}. \quad (32)$$

Lemma 5: If $g_k(cu) = cg_k(u)$ for any real number $c > 0$, $u \in \mathbb{R}^{n_k}$, and $k \in \{1, \dots, q\}$, then $V(\alpha) = V(\mathbb{1}_q)D(\alpha)$, $\forall \alpha \in [0, \mathbb{1}_q]$.

Based on Lemma 5, for any $\alpha \in [0, \mathbb{1}_q]$, we have

$$V(\alpha) = V(\mathbb{1}_q)D(\alpha) \quad (33)$$

which is clearly much faster to compute than solving q instances of (30) for every required value of α .

For the example in Fig. 9, we have

$$V(\alpha) = \begin{bmatrix} 189 & 71 \\ 194 & 78 \\ 106 & 218 \\ 106 & 221 \end{bmatrix} \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} = \begin{bmatrix} 189\alpha_1 & 71\alpha_2 \\ 194\alpha_1 & 78\alpha_2 \\ 106\alpha_1 & 218\alpha_2 \\ 106\alpha_1 & 221\alpha_2 \end{bmatrix}$$

where the units are in mV.

3) *Superposition:* It is practically impossible to solve (28) for every required β , as this could lead to combinatorial explosion in the required values of β . Lemma 6 establishes the principle of superposition for the equivalent passive grid.

Lemma 6: For any $\alpha \in [0, \mathbb{1}_q]$ and $\beta \in \mathbb{B}^q$, we have

$$v(\alpha, \beta) = \sum_{k=1}^q \beta_k v_k(\alpha_k) = V(\alpha)\beta. \quad (34)$$

The importance of Lemma 6 is that, for a given value of $\alpha \in [0, \mathbb{1}_q]$, $v(\alpha, \beta)$ can be found for different working modes β by a simple matrix-vector multiplication between $V(\alpha)$ and β , which is significantly faster than solving (28) for every required β .

This leads to our main theoretical result and the main reason behind the computational efficiency of this paper, as stated in Theorem 1.

Theorem 1: If $g_k(cu) = cg_k(u)$ for any real number $c > 0$, $u \in \mathbb{R}^{n_k}$, and $k \in \{1, \dots, q\}$, then for any $\alpha \in [0, \mathbb{1}_q]$ and $\beta \in \mathbb{B}^q$, we have

$$v(\alpha, \beta) = V(\mathbb{1}_q)D(\alpha)\beta. \quad (35)$$

The importance of the above result is that it allows us to find an upper bound on the worst case voltage drop experienced by the nodes of interest in the full grid $v(\alpha, \beta)$ for any $\alpha \in [0, \mathbb{1}_q]$ and $\beta \in \mathbb{B}^q$ by solving $v_k(1)$, defined in (30), $\forall k \in \{1, \dots, q\}$, constructing $V(\mathbb{1}_q)$, defined in (31), and performing two matrix-vector multiplications, as in (35). Note that we only need to find $V(\mathbb{1}_q)$ once, which will then be used to find $v(\alpha, \beta)$, for any $\alpha \in [0, \mathbb{1}_q]$ and $\beta \in \mathbb{B}^q$. This is clearly much faster than solving (28) for every required value of α and β .

For the example in Fig. 9, we have

$$v(\alpha, \beta) = \begin{bmatrix} 189\alpha_1 & 71\alpha_2 \\ 194\alpha_1 & 78\alpha_2 \\ 106\alpha_1 & 218\alpha_2 \\ 106\alpha_1 & 221\alpha_2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \text{ mV}. \quad (36)$$

4) *Safe Working Modes:* In a power-gated PDN, the power-gating switches of a block are turned OFF when the logic circuitry underlying that block is in “idle” or “sleep” state. Clearly, the voltage levels inside an OFF block do not affect the voltage integrity of the PDN and the only nodes whose voltage drop “matters” are the nodes of interest inside the ON blocks as they are connected to switching logic circuitry. In this section, we provide a formal definition for the safety of the equivalent passive grid that is based on the voltage drops at the nodes of interest inside the ON blocks. Furthermore, we provide an equivalent mathematical condition that captures this safety criterion.

We start by defining the safety condition for the full grid.

Definition 5: The equivalent passive grid is said to be safe under $\mathcal{F}(\alpha, \beta)$, if for every node of interest i that belongs to an ON block j , we have $v_i(\alpha, \beta) \leq \gamma_j$.

In the following lemma, we will provide an equivalent mathematical condition that captures the safety of the equivalent passive grid. We will introduce a new voltage drop threshold vector that is a function of the working mode β , denoted as $v_{\text{th}}(\beta)$, which will then be used to check if the grid is safe by comparing $v(\alpha, \beta)$ to $v_{\text{th}}(\beta)$, as we will prove in Lemma 7. Based on the working mode β , the entries of $v_{\text{th}}(\beta)$ that correspond to the nodes of interest that belong to OFF blocks will become very large, so that the voltage drop at those nodes does not impact the safety of the grid, whereas the entries of $v_{\text{th}}(\beta)$ that correspond to the nodes of interest that belong to ON blocks will have the original voltage drop threshold specification.

Let T be a $d \times q$ matrix of 0 and 1 entries that identifies (with a 1) which node of interest belongs to which block, i.e., $T_{ij} = 1$ if the i th node of interest belongs to the j th block, otherwise $T_{ij} = 0$. Also, let $v_{\text{th}}(\beta) = V_{\text{th}} + \rho T(\mathbb{1}_q - \beta)$, where $\rho > 0$ is a large number. It is enough for ρ to be larger than $\|V(\mathbb{1}_q)\mathbb{1}_q\|_{\infty}$. Notice that for any $\beta \in \mathbb{B}^q$, we have $\beta \leq \mathbb{1}_q$, so that $\mathbb{1}_q - \beta \geq 0$ which, because $\rho \geq 0$ and $T \geq 0$, gives $\rho T(\mathbb{1}_q - \beta) \geq 0$. Thus, we have

$$v_{\text{th}}(\beta) = V_{\text{th}} + \rho T(\mathbb{1}_q - \beta) \geq V_{\text{th}} > 0. \quad (37)$$

Lemma 7: For any $\alpha \in [0, \mathbb{1}_q]$ and $\beta \in \mathbb{B}^q$, the equivalent passive grid is safe if and only if $V(\alpha)\beta \leq v_{\text{th}}(\beta)$.

For any $\beta \in \mathbb{B}^q$ such that $V(\alpha)\beta \leq v_{\text{th}}(\beta)$, β is said to be a *safe working mode*. Define the set $\mathcal{W}(\alpha)$ to be the set of all safe working modes under the blocks’ containers $\mathcal{F}_k(\alpha_k)$, i.e.,

$$\mathcal{W}(\alpha) \triangleq \{\beta \in \mathbb{B}^q : V(\alpha)\beta \leq v_{\text{th}}(\beta)\} \quad (38)$$

which is captured by a BDD.

For the example in Fig. 9, $\mathcal{W}(\alpha)$ is

$$\left\{ \beta \in \mathbb{B}^2 : V(\alpha)\beta \leq \begin{bmatrix} 100 \\ 100 \\ 70 \\ 70 \end{bmatrix} + 327 \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} (\mathbb{1}_q - \beta) \right\}. \quad (39)$$

To better visualize this, consider again the example of Fig. 9. In Fig. 11(c), we show the set $\mathcal{W}'(\alpha) = \{x \in \mathbb{R}^q : V(\alpha)x \leq v_{\text{th}}(x)\}$ for different values of α . Notice that, for any $\alpha \in$

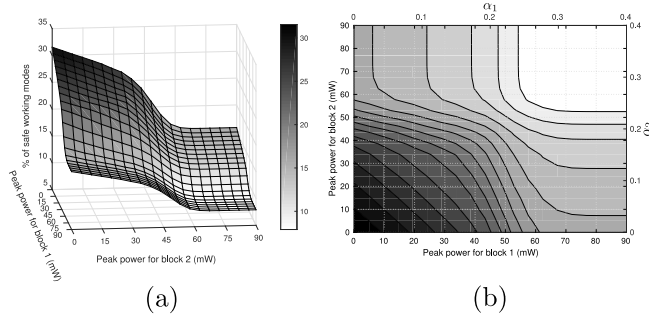


Fig. 12. (a) 3-D plot and (b) contour plot of the percentage of safe working modes for different values of α on a 5k node grid with 16 blocks. Color bar: percentage of safe working modes.

$[0, \mathbb{1}_q]$, $\mathcal{W}(\alpha)$ consists of the Boolean vectors $\beta \in \mathbb{B}^q$ that lie inside the space $\mathcal{W}'(\alpha)$. So, as shown in Fig. 11(c)

$$\mathcal{W}\left(\begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix}\right) = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\} \quad (40)$$

$$\mathcal{W}\left(\begin{bmatrix} 0.4 \\ 0.2 \end{bmatrix}\right) = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}. \quad (41)$$

So far, any $\alpha \in [0, \mathbb{1}_q]$ will give us the required block-level current containers $\mathcal{F}_k(\alpha_k) = \mathcal{L}_k(u_k^*(\alpha_k))$ and the corresponding set of safe working modes $\mathcal{W}(\alpha)$, as defined in (23) and (38).

V. APPLICATION

Referring again to the example of Fig. 9, notice that larger α_1 corresponds to a larger $\mathcal{F}_1(\alpha_1)$ [as shown in Fig. 11(a)], and hence, a larger power budget for block 1. Similarly, larger α_2 corresponds to a larger $\mathcal{F}_2(\alpha_2)$ [as shown in Fig. 11(b)], and hence, a larger power budget for block 2. On the other hand, larger local power budgets result in larger voltage drops at the grid nodes, and hence, a smaller number of safe working modes [as shown in Fig. 11(d)]. To better illustrate this, consider two different values of α : $\alpha^{(1)} = [0.4 \ 0.2]^T$ and $\alpha^{(2)} = [0.2 \ 0.2]^T$. Notice that $\mathcal{F}_1(\alpha_1^{(1)}) \supset \mathcal{F}_1(\alpha_1^{(2)})$, as shown in Fig. 11(a), and $\mathcal{F}_2(\alpha_2^{(1)}) = \mathcal{F}_2(\alpha_2^{(2)})$, because $\alpha_1^{(2)} = \alpha_2^{(2)}$. Furthermore, $\mathcal{W}(\alpha^{(1)}) \subset \mathcal{W}(\alpha^{(2)})$, as shown in Fig. 11(d). Therefore, $\alpha^{(1)}$ allows larger power budget for block 1 but allows less flexibility in terms of the number of safe working modes, as compared to $\alpha^{(2)}$. There is a clear tradeoff for different values of α . The tradeoff is between the local power budgets allocated to individual blocks (based on the generated local containers) and the number of safe working modes. In fact, as we will see below, the local power budget of block k is directly proportional to α_k , and hence, we can think of α as the allocated power budgets for the individual blocks which, in turn, determine the safe working modes. In Fig. 12, we show the tradeoff achieved for different values of α on a 5k node grid with 16 blocks. Fig. 12(a) and (b) corresponds to different values of α_1 and α_2 between 0 and 0.4, while $\alpha_3, \alpha_4, \dots$, and α_{16} are fixed to 0.85. Again, because the power budget of block k is directly proportional to its parameter α_k , we present the percentage of safe working modes as a function of the power budgets for blocks 1 and 2 [the corresponding values

of α_1 and α_2 are shown at the right and top axes of Fig. 12(b), respectively]. Some values of α allow for large local power budgets but a small number of safe working modes, whereas other values of α allow small local power budgets but a large number of safe working modes. Thus, the question becomes, which α should we choose?

In this section, we will describe two design objectives: 1) the maximum peak-power dissipation that each block can safely support; and 2) the largest number of safe working modes. In Section V-A, we will describe some types of user-specified constraints that our approach can handle, basically constraints on the peak power that each block can safely support and the allowable working modes, and we will see that these constraints can be represented as linear inequalities on α , resulting in a feasible space of α , denoted as \mathcal{A} . The proposed algorithms will each be formulated as a maximization of the corresponding design objective, overall $\alpha \in \mathcal{A}$, resulting in an α that allows large local power budgets at the cost of a small number of safe working modes, or an α that allows more blocks to turn ON simultaneously at the cost of smaller local power budgets. Or, as probably the most useful case, an intermediate value of α between the two limits will be chosen to achieve some objective on the size of the local containers or the percentage of safe working modes.

A. User-Specified Constraints

In this section, we will examine two approaches for users to influence the space of α based on any specifications that may be known about the design at an early stage, thus achieving different tradeoffs for chip operation. In a sense, these specifications will help reduce the space of α to a space that reflects design knowledge.

The user can enforce some working modes to be allowed during chip operation, which we can incorporate as in (47). Also, the user can enforce any local current/power budgets to satisfy some constraints, which we can incorporate as in (56). Assuming that the working mode constraints and the current/power constraints are consistent and feasible, so that there exists an $\alpha \in [0, \mathbb{1}_q]$ that satisfies (47) and (56), then we can define the feasible space of α as follows:

$$\mathcal{A} \triangleq \{ \alpha \in [0, \mathbb{1}_q] : W\alpha \leq w, p_{lb} \leq R\alpha \leq p_{ub} \}. \quad (42)$$

Fig. 13 shows an example of \mathcal{A} for the simple grid in Fig. 9 corresponding to the user-specified constraints in (44), (45), and (53)–(55).

1) Working Modes' Constraints: Suppose we have some knowledge about the working modes of the circuit, for example, if there exist some dependences among the blocks, i.e., a subset of the blocks are required to be ON at the same time. In general, let \mathcal{W}_0 denote the set of user-specified working modes that are required to be safe. This type of constraint can be easily embedded into our framework by searching for α that satisfies $\mathcal{W}_0 \subseteq \mathcal{W}(\alpha)$. We will see below that this constraint can be represented as a set of linear constraints on α , i.e.,

$$W\alpha \leq w. \quad (43)$$

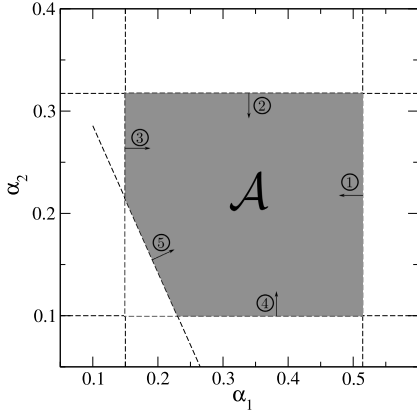


Fig. 13. Feasible space of α in Fig. 9 as a result of some user-specified constraints.

Referring to the example of Fig. 9, we can impose a constraint that each block is safe to turn ON separately. In other words, we are interested in the values of $\alpha \in [0, \mathbb{1}_2]$ such that $\beta^{(1)}, \beta^{(2)} \in \mathcal{W}(\alpha)$, where $\beta^{(1)} = [1 \ 0]^T$ and $\beta^{(2)} = [0 \ 1]^T$. Thus, based on (39), we have

$$V(\alpha)\beta^{(1)} \leq v_{\text{th}}(\beta^{(1)}) \iff \alpha_1 \leq 0.52 \quad (44)$$

$$V(\alpha)\beta^{(2)} \leq v_{\text{th}}(\beta^{(2)}) \iff \alpha_2 \leq 0.32. \quad (45)$$

In Fig. 13, we show the above two constraints; (44) is numbered 1 and (45) is numbered 2.

For any $\beta \in \mathcal{W}_0$, let $W(\beta) = V(1)D(\beta)$. Assuming that a total of ζ working modes are required to be safe, i.e., $\mathcal{W}_0 = \{\beta^{(1)}, \beta^{(2)}, \dots, \beta^{(\zeta)}\}$, let W and w be a $(\zeta d) \times q$ matrix and a $(\zeta d) \times 1$ vector, respectively, such that

$$W = \begin{bmatrix} W(\beta^{(1)}) \\ \vdots \\ W(\beta^{(\zeta)}) \end{bmatrix}, \quad w = \begin{bmatrix} v_{\text{th}}(\beta^{(1)}) \\ \vdots \\ v_{\text{th}}(\beta^{(\zeta)}) \end{bmatrix}. \quad (46)$$

Lemma 8 transforms the constraint $\mathcal{W}_0 \subseteq \mathcal{W}(\alpha)$ into a set of linear inequalities on α .

Lemma 8: If $g_k(cu) = cg_k(u)$, for any real number $c > 0$, $u \in \mathbb{R}^{nk}$, and $\forall k \in \{1, \dots, q\}$, then for any $\alpha \in [0, \mathbb{1}_q]$, $\mathcal{W}_0 \subseteq \mathcal{W}(\alpha)$ if and only if

$$W\alpha \leq w. \quad (47)$$

2) *Current/Power Constraints:* A broad range of power bounds can be imposed on the resulting containers, given specifications about the design at an early stage. In the following, we will discuss several examples of such constraints that could be embedded in our framework and we will show in Lemma 10 that these constraints can be represented as a set of linear inequalities on α , i.e.,

$$p_{lb} \leq R\alpha \leq p_{ub}. \quad (48)$$

Define $\psi_k(\alpha_k)$ to be the largest instantaneous peak power dissipation achievable under $\mathcal{F}_k(\alpha_k)$, i.e.,

$$\psi_k(\alpha_k) = V_{\text{dd}} \max_{I_k \in \mathcal{F}_k(\alpha_k)} (\mathbb{1}_{m_k}^T I_k). \quad (49)$$

Recall that for any $\alpha \in [0, \mathbb{1}_q]$, $\mathcal{F}_k(\alpha_k)$ is nonempty, so that $\psi_k(\alpha_k) \geq 0$ is well defined.

The simplest bounds are on the minimum peak power, referred to as *local constraints*, such as $\psi_{lb} \leq \psi(\alpha) \leq \psi_{ub}$, where $\psi(\alpha) = [\psi_1(\alpha_1) \ \dots \ \psi_q(\alpha_q)]^T$ is a $q \times 1$ vector of the peak-power dissipation that each block can safely support and ψ_{lb} and ψ_{ub} are the vectors of user-specified lower and upper bounds on the peak-power dissipation of the blocks. Another bound commonly available from design specification is the peak total power dissipation of a group of blocks, referred to as *global constraints*, that is available at an early stage of the design, then assuming we have a total of κ global constraints, we can incorporate these constraints as $c_{lb} \leq F\psi(\alpha) \leq c_{ub}$, where F is a $\kappa \times q$ matrix that consists only of 0s and 1s, which indicate which block is present in each constraint, so that $F \geq 0$ has no row with all zeros, and c_{lb} and c_{ub} are $\kappa \times 1$ vectors representing the lower and upper bounds on the peak-power dissipation. We can represent the local and global constraints compactly as

$$p_{lb} \leq U\psi(\alpha) \leq p_{ub} \quad (50)$$

where

$$p_{lb} = \begin{bmatrix} \psi_{lb} \\ c_{lb} \end{bmatrix}, \quad p_{ub} = \begin{bmatrix} \psi_{ub} \\ c_{ub} \end{bmatrix}, \quad \text{and } U = \begin{bmatrix} \mathbb{1}_q \\ F \end{bmatrix}.$$

Lemma 9 establishes the scalability of $\psi(\alpha)$, which will be useful to prove Lemma 10.

Lemma 9: If $g_k(cu) = cg_k(u)$, for any real number $c > 0$, $u \in \mathbb{R}^{nk}$, and $\forall k \in \{1, \dots, q\}$, then $\psi(\alpha) = D(\alpha)\psi(\mathbb{1}_q)$, $\forall \alpha \in [0, \mathbb{1}_q]$.

Based on Lemma 9, for any $\alpha \in [0, \mathbb{1}_q]$, we have

$$\psi(\alpha) = D(\alpha)\psi(\mathbb{1}_q) \quad (51)$$

which is clearly much faster to compute than solving q instances of (49) for every required value of α .

For the example in Fig. 9, we have

$$\psi(\alpha) = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} 100 \\ 70 \end{bmatrix} = \begin{bmatrix} 100\alpha_1 \\ 70\alpha_2 \end{bmatrix} \text{ mW} \quad (52)$$

which allows us to impose power constraints on the blocks. For example, the peak power of blocks 1 and block 2 is larger than 15 and 7 mW, respectively, and the total peak power that both blocks can dissipate simultaneously is larger than 30 mW. In other words, we are only interested in the values of $\alpha \in [0, \mathbb{1}_2]$ such that

$$\psi_1(\alpha_1) \geq 15 \text{ mW} \iff \alpha_1 \geq 0.15 \quad (53)$$

$$\psi_2(\alpha_2) \geq 7 \text{ mW} \iff \alpha_2 \geq 0.1 \quad (54)$$

$$\psi_1(\alpha_1) + \psi_2(\alpha_2) \geq 30 \text{ mW} \iff \alpha_1 + 0.7\alpha_2 \geq 0.3. \quad (55)$$

In Fig. 13, we show the above three constraints; (53) is numbered 3, (54) is numbered 4, and (55) is numbered 5.

Lemma 10 transforms the user-specified power constraints into a set of linear inequalities on α .

Lemma 10: If $g_k(cu) = cg_k(u)$, for any real number $c > 0$, $u \in \mathbb{R}^{nk}$, and $\forall k \in \{1, \dots, q\}$, then we have $p_{lb} \leq U\psi(\alpha) \leq p_{ub}$ if and only if

$$p_{lb} \leq R\alpha \leq p_{ub} \quad (56)$$

where $R = UD(\psi(\mathbb{1}_q))$.

B. Maximum Local Power

The design team may be interested in a workload scheduler that allows as much local power dissipation as possible to the underlying circuit. We refer here to the instantaneous power dissipation, which is conservatively approximated by $V_{dd} \sum_{j=1}^{m_k} i_{k,j}(t)$ for every block k , where $i_{k,j}(t)$ is the time-varying current waveform representing the current drawn by the j th current source in block k . Recall that $\psi_k(\alpha)$ defines the peak-power dissipation that block k can safely support in the underlying circuit. Thus, we are interested in an α that allows the highest possible $\sum_{\forall k} \psi_k(\alpha)$, while satisfying the user-specified requirements on the resulting local containers and working modes, i.e., (42). We can formulate this as the following optimization problem:

$$\begin{aligned} \sigma^* &= \text{Max } \mathbb{1}_q^T \psi(\alpha) \\ \text{s.t. } &\alpha \in \mathcal{A}. \end{aligned} \quad (57)$$

Let $\alpha^{(p)}$ be a vector at which the above maximization attains its maximum. In other words, $\alpha^{(p)} \in \mathcal{A}$ such that $\mathbb{1}_q^T \psi(\alpha^{(p)}) = \sigma^*$. Because \mathcal{A} is nonempty, it follows that $\alpha^{(p)}$ is well defined. Therefore, the resulting block-level containers are $\mathcal{F}_k(\alpha_k^{(p)})$, which describe the following current constraints:

$$i_k(t) \geq 0 \quad (58)$$

$$M'_k i_k(t) \leq \alpha_k^{(p)} M_k G_k u_k^*(1) \quad (59)$$

for every $k \in \{1, \dots, q\}$, where $i_k(t)$ is the time-varying current waveform representing the current drawn by the k th block. Furthermore, the resulting set of safe working modes is

$$\mathcal{W}(\alpha^{(p)}) = \{\beta \in \mathbb{B}^q : V(\mathbb{1}_q)D(\alpha^{(p)})\beta \leq v_{th}(\beta)\}. \quad (60)$$

In the following, we will show that the optimization problem in (57) is equivalent to the LP in (64), and hence, we will be solving (64) instead of (57). Notice that, due to Lemma 9, we have

$$\mathbb{1}_q^T \psi(\alpha) = \mathbb{1}_q^T D(\alpha) \psi(\mathbb{1}_q). \quad (61)$$

Also, notice that

$$D(\alpha) \psi(\mathbb{1}_q) = [\alpha_1 \psi_1(1) \cdots \alpha_q \psi_q(1)]^T \quad (62)$$

$$= D(\psi(\mathbb{1}_q)) \alpha. \quad (63)$$

Therefore, we have $\mathbb{1}_q^T \psi(\alpha) = \mathbb{1}_q^T D(\psi(\mathbb{1}_q)) \alpha = \psi^T(\mathbb{1}_q) \alpha$. Thus, we can rewrite (57) as follows:

$$\begin{aligned} \text{Max } &\psi^T(\mathbb{1}_q) \alpha \\ \text{s.t. } &\alpha \in \mathcal{A}. \end{aligned} \quad (64)$$

C. Maximum Working Modes

Another approach that the design team might be interested in is a workload scheduler that allows as much flexibility for the blocks to turn ON simultaneously as possible, while still satisfying the user-specified requirements. Let $|\mathcal{W}(\alpha)|$ denote the cardinality of the set $\mathcal{W}(\alpha)$. Thus, we are interested in α that maximizes $|\mathcal{W}(\alpha)|$ and satisfies the user-specified

requirements. We can find such an α by solving the following optimization problem:

$$\begin{aligned} \text{Max } &|\mathcal{W}(\alpha)| \\ \text{s.t. } &\alpha \in \mathcal{A}. \end{aligned} \quad (65)$$

Solving (65) is computationally expensive. For one thing, $|\mathcal{W}(\alpha)|$ is a nonconvex function of α . Alternatively, we propose a simpler optimization problem in (69), in fact an LP, motivated by Lemma 11. Lemma 11 establishes a sufficient condition that maximizes $|\mathcal{W}(\alpha)|$; to maximize $|\mathcal{W}(\alpha)|$, it is enough to minimize all the elements of α . In a sense, it is sufficient to “minimize” α . Typically, this can be achieved by minimizing some norm of α (e.g., Euclidean norm, sum norm, and infinity norm). In this paper, we will be minimizing the infinity norm of α , i.e., $\|\alpha\|_\infty$; this will be formulated as the LP in (69).

Lemma 11: If $g_k(cu) = cg_k(u)$ for any real number $c > 0$, $u \in \mathbb{R}^{n_k}$, and $k \in \{1, \dots, q\}$, then for any $\alpha, \alpha' \in [0, \mathbb{1}_q]$ with $\alpha \leq \alpha'$, we have $|\mathcal{W}(\alpha)| \geq |\mathcal{W}(\alpha')|$.

For any $\alpha \in \mathcal{A}$, let $\zeta(\alpha) \triangleq \|\alpha\|_\infty$ denote the infinity norm of α , i.e.,

$$\zeta(\alpha) = \max_{\forall i} |\alpha_i| = \max_{\forall i} \alpha_i \quad (66)$$

the last step due to $\alpha \geq 0$. Notice that $\zeta(\alpha)$ is the smallest real number greater than $\alpha_i, \forall i$, so that

$$\zeta(\alpha) = \min_{\zeta \mathbb{1}_q \geq \alpha} \zeta. \quad (67)$$

We define ζ^* to be the smallest $\zeta(\alpha)$ achievable over all possible $\alpha \in \mathcal{A}$, i.e.,

$$\zeta^* \triangleq \min_{\alpha \in \mathcal{A}} (\zeta(\alpha)). \quad (68)$$

Let $\alpha^{(w)}$ be a vector at which the above maximization attains its minimum. In other words, $\alpha \in \mathcal{A}$ such that $\zeta(\alpha^{(w)}) = \zeta^*$. Because \mathcal{A} is nonempty, it follows that $\alpha^{(w)}$ is well defined. We can express the combined (67) and (68) as the following LP:

$$\begin{aligned} \zeta^* &= \text{Min } \zeta \\ \text{s.t. } &\zeta \mathbb{1}_q \geq \alpha \\ &\alpha \in \mathcal{A}. \end{aligned} \quad (69)$$

Therefore, the resulting block-level containers are $\mathcal{F}_k(\alpha_k^{(w)})$, which describe the following current constraints:

$$i_k(t) \geq 0 \quad (70)$$

$$M'_k i_k(t) \leq \alpha_k^{(w)} M_k G_k u_k^*(1) \quad (71)$$

for every $k \in \{1, \dots, q\}$. Furthermore, the resulting set of safe working modes is

$$\mathcal{W}(\alpha^{(w)}) = \{\beta \in \mathbb{B}^q : V(\mathbb{1}_q)D(\alpha^{(w)})\beta \leq v_{th}(\beta)\}. \quad (72)$$

TABLE I
POWER GRID PROPERTIES AND THE RUNTIME BREAKDOWN

Power Grid					Runtime			
Name	Nodes	Current Sources	Blocks	Number of Layers ^a	Isolated Block Analysis	Full Grid Analysis	LP (64)	LP (69)
G1	5,654	828	4	8:6	59 msec	14 sec	30 msec	25 msec
G2	5,630	792	9	8:6	120 msec	2 sec	34 msec	26 msec
G3	5,582	800	16	8:6	185 msec	1 sec	41 msec	23 msec
G4	53,030	8,208	16	8:6	350 msec	3.7 min	224 msec	108 msec
G5	52,950	8,250	25	8:6	455 msec	2 min	301 msec	144 msec
G6	49,766	7,452	36	8:6	540 msec	48 sec	246 msec	127 msec
G7	595,380	96,960	64	8:6	1 sec	2.9 hr	1 sec	830 msec
G8	1,322,908	214,800	100	8:6	3 sec	11.2 hr	3 sec	1.6 sec

^a represented as $x : y$, where x is the number of metal layers in the full grid and y is the number of metal layers in the global grid

TABLE II
USER-SPECIFIED CONSTRAINTS' PARAMETERS AND COMPARISON OF THE TWO DESIGN OBJECTIVES

Power Grid	Power Constraints	BDD Constraints		Maximum Local Power		Maximum Working Modes	
Name	Min. Avg. Peak Power (mW)	# of working modes	Max. ON ^a	$P(\alpha^{(p)})$ (mW)	$\omega(\alpha^{(p)})$	$P(\alpha^{(w)})$ (mW)	$\omega(\alpha^{(w)})$
G1	43	8	2	85	37.50%	43	81.25%
G2	30	14	4	63	12.10%	30	95.70%
G3	15	21	7	35	1.80%	15	97.58%
G4	111	21	7	183	1.83%	111	80.62%
G5	77	30	8	155	0.20%	77	96.75%
G6	58	41	13	122	-	58	-
G7	344	69	8	586	-	344	-
G8	423	105	8	724	-	423	-

^a the maximum number of ON blocks in the user-specified working modes.

VI. EXPERIMENTAL RESULTS

The approach discussed in Section IV has been implemented in C++. We conducted tests on a set of power grids that were generated based on user specifications, including grid dimensions, metal layers, a number of blocks, a number of metal layers in the global grid, pitch and width per layer, and C4 and current source distributions. The technology specifications were consistent with 1-V 45-nm CMOS technology. Table I shows the characteristics of a number of test grids. All results were obtained using a hyperthreaded 12-core 3-GHz Linux machine with 128-GB RAM. The optimizations were performed using MOSEK optimization package [22]. All the linear systems are solved using Cholmod [23] from SuiteSparse [24]. In our implementation, we use *Pthread* to parallelize the computation and take advantage of the 12-core machine. The runtime breakdown of our approach, i.e., the isolated block analysis, the full grid analysis, LP (64), and LP (69), is shown in columns 6–9 of Table I, which represent the wall clock time for the parallel Pthread implementation. Recall that in the isolated block analysis, the block-level containers are generated based on a choice of the design objective $g_k(\cdot)$. In our tests, we used the peak power algorithm in [20] and the uniform current distribution algorithm in [25] as design objectives for all the blocks.

Table II compares the results of using $\alpha^{(p)}$ (see Section V-B) and $\alpha^{(w)}$ (see Section V-C) based on user-specified constraints. In column 2, we describe the user-specified constraints on the

local power. Specifically, we require the average of the peak powers of all the blocks to be larger than the specification in column 2. Furthermore, in columns 3 and 4, we describe the user-specified constraints on the working modes, i.e., the number of user-specified working modes as well as the maximum number of blocks that are ON in those working modes. Denote by $P(\alpha)$ the average of the peak powers of all the blocks under the block containers $\mathcal{F}_k(\alpha_k)$. Also, denote by $\omega(\alpha)$ the percentage of the working modes that are safe under block containers $\mathcal{F}_k(\alpha_k)$. To study the difference between the generated block containers and $\mathcal{W}(\cdot)$ using $\alpha^{(p)}$ and $\alpha^{(w)}$, we found the average of the peak powers of all the blocks under $\mathcal{F}_k(\alpha_k^{(p)})$ and $\mathcal{F}_k(\alpha_k^{(w)})$, which are $P(\alpha^{(p)})$ and $P(\alpha^{(w)})$, and the percentage of safe working modes in $\mathcal{W}(\alpha^{(p)})$ and $\mathcal{W}(\alpha^{(w)})$, which are $\omega(\alpha^{(p)})$ and $\omega(\alpha^{(w)})$. For instance, on a 52950 node grid with 25 blocks, the average of the peak powers for all blocks under $\mathcal{F}_k(\alpha_k^{(p)})$ and $\mathcal{F}_k(\alpha_k^{(w)})$ are 155 and 77 mW, respectively, and the percentage of safe working modes under $\mathcal{W}(\alpha^{(p)})$ and $\mathcal{W}(\alpha^{(w)})$ are 0.2% and 96.75%, respectively. The results show that $P(\alpha^{(p)}) \gg P(\alpha^{(w)})$ and $\omega(\alpha^{(p)}) \ll \omega(\alpha^{(w)})$. Therefore, each approach provides a distinct tradeoff for the design team.

VII. CONCLUSION

Power-gating design technique introduces active devices, such as MOSFETs, in the chip's power distribution network to disconnect circuit blocks that are not required to operate

from the rest of the PDN. Analysis and verification of active PDNs are crucial to ensure voltage integrity. In this paper, we focus on analyzing RC active grids under different working modes, but we are working to extend this to handle RLC grids under both working and transition modes. With active devices, most traditional techniques are ill-equipped to verify the PDN. The worst case voltage drop is the result of two things: the power budgets that were allocated to the various circuit blocks during the design process and the combination of blocks that are turned ON in a given operational mode. We propose a framework to generate block-level circuit current constraints as well as an implicit BDD that helps identify the safe working modes. Subject to user guidance, we then propose two design objectives that exploit the tradeoff between how many blocks are ON simultaneously and how big the power budgets of individual blocks are.

REFERENCES

- [1] H. Jiang, M. Marek-Sadowska, and S. R. Nassif, "Benefits and costs of power-gating technique," in *Proc. IEEE Int. Conf. Comput. Design*, Oct. 2005, pp. 559–566.
- [2] T. Xu, P. Li, and B. Yan, "Decoupling for power gating: Sources of power noise and design strategies," in *Proc. ACM/IEEE Design Autom. Conf.*, Jun. 2011, pp. 1002–1007.
- [3] T. Xu and P. Li, "Design and optimization of power gating for DVFS applications," in *Proc. 13th Int. Symp. Qual. Electron. Design (ISQED)*, Mar. 2012, pp. 391–397.
- [4] J. Li *et al.*, "Thermal-aware task scheduling in 3D chip multiprocessor with real-time constrained workloads," in *Proc. ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 2, Feb. 2013, Art. no. 24.
- [5] A. K. Coskun, T. S. Rosing, K. A. Whisnant, and K. C. Gross, "Static and dynamic temperature-aware scheduling for multiprocessor SoCs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 9, pp. 1127–1140, Sep. 2008.
- [6] H. Khdr, T. Ebi, M. Shafique, H. Amrouch, and J. H. Karlsruhe, "mDTM: Multi-objective dynamic thermal management for on-chip systems," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Dresden, Germany, Mar. 2014, pp. 1–6.
- [7] A. Krstic and K.-T. Cheng, "Vector generation for maximum instantaneous current through supply lines for CMOS circuits," in *Proc. 34th Design Automat. Conf.*, Anaheim, CA, USA, Jun. 1997, pp. 383–388.
- [8] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, and R. Panda, "A stochastic approach to power grid analysis," in *Proc. 41st Design Automat. Conf. (DAC)*, San Diego, CA, USA, Jun. 2004, pp. 171–176.
- [9] H. Zhuang, S.-H. Weng, J.-H. Lin, and C.-K. Cheng, "MATEX: A distributed framework for transient simulation of power distribution networks," in *Proc. 51st ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, Jun. 2014, pp. 1–6.
- [10] M. Fawaz and F. N. Najm, "Parallel simulation-based verification of RC power grids," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2017, pp. 445–452.
- [11] F. N. Najm, "Overview of vectorless/early power grid verification," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2012, pp. 670–677.
- [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] M. Fawaz and F. N. Najm, "Power grid verification under transient constraints," in *Proc. IEEE/ACM 36th Int. Conf. Comput.-Aided Design*, Nov. 2017, pp. 593–600.
- [14] M. Fawaz and F. N. Najm, "Fast vectorless RLC grid verification," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 3, pp. 489–502, Mar. 2017.
- [15] Z. Zhao and Z. Feng, "A spectral graph sparsification approach to scalable vectorless power grid integrity verification," in *Proc. 54th ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, Jun. 2017, pp. 1–6.
- [16] H. Zhu, Y. Wang, F. Liu, X. Li, X. Zeng, and P. Feldmann, "Efficient transient analysis of power delivery network with clock/power gating by sparse approximation," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 34, no. 3, pp. 409–421, Mar. 2015.
- [17] Z. Zeng, Z. Feng, and P. Li, "Efficient checking of power delivery integrity for power gating," in *Proc. 12th Int. Symp. Qual. Electron. Design (ISQED)*, Santa Clara, CA, USA, Mar. 2011, pp. 1–8.
- [18] A. M. Rahmani, P. Liljeberg, A. Hemani, A. Jantsch, and H. Tenhunen, *The Dark Side of Silicon*. Cham, Switzerland: Springer, 2016.
- [19] Z. Moudallal and F. N. Najm, "Power scheduling with active power grids," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2017, pp. 466–473.
- [20] Z. Moudallal and F. N. Najm, "Generating current budgets to guarantee power grid safety," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 11, pp. 1914–1927, Nov. 2016.
- [21] M. Fawaz and F. N. Najm, "Accurate verification of RC power grids," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Dresden, Germany, Mar. 2016, pp. 814–817.
- [22] MOSEK ApS. (2015). *The MOSEK C Optimizer API Manual. Version 7.1 (Revision 28)*. [Online]. Available: <http://docs.mosek.com/7.1/toolbox/index.html>
- [23] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, "Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/downdate," *ACM Trans. Math. Softw.*, vol. 35, no. 3, 2008, Art. no. 22.
- [24] T. Davis. *Suitesparse 4.4.6*. Accessed: 2015. [Online]. Available: <http://faculty.cse.tamu.edu/davis/suitesparse.html>
- [25] Z. Moudallal and F. N. Najm, "Generating voltage drop aware current budgets for RC power grids," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 2583–2586.



Zahi Moudallal (S'16) received the B.E. degree (honors) in electrical and computer engineering from the American University of Beirut, Beirut, Lebanon, in 2012, and the M.A.Sc. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2014, where he is currently working toward the Ph.D. degree at the Department of Electrical and Computer Engineering.

His current research interests include computer-aided design for integrated circuits with a focus on reliability, verification, and analysis of power grids.



Farid N. Najm (S'85–M'89–SM'96–F'03) received the B.E. degree in electrical engineering from the American University of Beirut, Beirut, Lebanon, in 1983, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC), Champaign, IL, USA, in 1989.

From 1989 to 1992, he was with Texas Instruments, Dallas, TX, USA. He then joined the ECE Department, UIUC, as an Assistant Professor and became an Associate Professor in 1997. In 1999, he joined the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, where he is currently a Professor and the Chair. In 2010, he has authored the book *Circuit Simulation* (New York, NY, USA: Wiley, 2010). His research is on CAD for VLSI, with an emphasis on circuit level issues related to power, timing, variability, and reliability.

Dr. Najm is a Fellow of the Canadian Academy of Engineering. He has received the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS Best Paper Award, the NSF Research Initiation Award, the NSF CAREER Award, the Design Automation Conference (DAC) Prolific Author Award, and the Best Paper Award from the International Conference on Computer-Aided Design. He served in the executive committee of the International Symposium on Low-Power Electronics and Design (ISLPED) from 1999 to 2013 and has served as the TPC Chair and the General Chair for ISLPED. He was an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS from 1997 to 2002 and the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS from 2001 to 2009.