Power Scheduling with Active Power Grids

Zahi Moudallal
ECE Department, University of Toronto
Toronto, ON, Canada
zahi.moudallal@mail.utoronto.ca

Farid N. Najm
ECE Department, University of Toronto
Toronto, ON, Canada
f.najm@utoronto.ca

***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. 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 trade-off 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*—Power scheduling, power-gating, integrated circuits, power grid**

I. Introduction

Power gating 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 (PDN). 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 cut off 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 specifications, 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 for example [1], [2], [3]. 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 multi-power states and is not a limitation to our work. 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 work, we focus on analyzing the PDN under different working modes, but we are working to extend this to transition modes.

Several computer-aided design (CAD) 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 [4], [5]. An alternative power grid verification scheme, such as in [6], [7], 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 2 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. The authors in [8] propose an efficient transient analysis approach of the power delivery network exploiting localized voltage variations near the active blocks. Such an approach requires 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. In [9], the authors propose 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 work 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

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

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

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. 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 trade-off between how many blocks are ON simultaneously and how big the power budgets of the individual blocks are. We focus on active 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 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 run-time schedule can then use the BDD as a query engine to check whether a candidate working mode is safe or not.

II. Overview

In a power-gated design, functional blocks have their own local grids that are connected to the global grid via wide multi-fingered transistors, referred to as sleep transistors or power-gating switches. Typically, a power-gating transistor may be modeled as an ideal switch in series with a resistor, as in Fig. 1(a). We will refer to the PDN model in Fig. 1(a) as the *original grid*.

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 work, we verify a slightly simplified model of the grid, which we call the *equivalent passive grid*, shown in Fig. 1(b). 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.

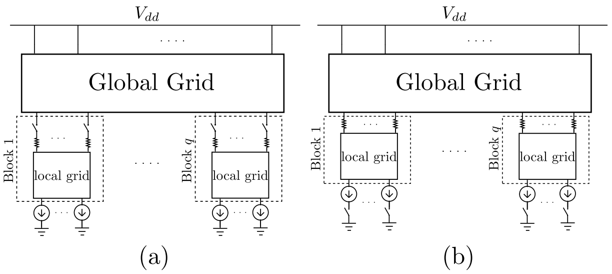


Fig. 1: (a) Schematic diagram of a power-gated PDN using resistive switches, referred to as the original grid, and (b) schematic diagram of the equivalent passive grid.

These benefits of using the equivalent passive grid come with a very small accuracy cost. Based on HSPICE simulations of a 400k nodes grid with 49 blocks, the relative error is 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 work, we will use the notion of a *current container*, introduced in [10], to capture the block-level power budgets. A container is usually expressed as a set of constraints on the currents drawn by the underlying logic circuitry. Taken in isolation, a block (local grid) can be analyzed separately using the inverse problem (constraints generation) approach for passive grids [10] 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 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 $V_{dd} - (1 - \alpha)\gamma_k$. 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 this α can be viewed as a parameter that becomes a "knob" of sorts by which we can have the local containers 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 α. 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 corresponding to any α. 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 the paper and we review some key theoretical results that were established for the constraints generation approach for passive power grids [10]. 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 work, we assume that a C4 pad is modeled with resistive and capacitive components only, because we focus on RC power grids. We are working to extend this to the RLC case in the future. 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 $1, 2, \ldots, n$ are the nodes not connected to a voltage source, while the remaining nodes $(n{+}1), (n{+}2), \ldots, (n{+}s)$ are the nodes where the s voltage sources are connected. Let $i(t)$ be the non-negative 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 $i_s(t) = Hi(t)$.

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 [6]:

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

where C is a $n \times n$ *diagonal* non-negative capacitance matrix, which is non-singular 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 non-positive 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 non-negative, $G^{-1} \geq 0$, i.e., its every entry is non-negative.

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:

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

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 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 [10] that is crucial to our work. 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 $v^*(\mathcal{F}) = \text{emax}_{i(\cdot)\subset\mathcal{F}}(v(\tau))$ where the "$\text{emax}(\cdot)$" notation denotes *element-wise* maximization, as in [10]. In [6], the authors provide 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:

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

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

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 [6], it has been shown that $\overline{v}(\mathcal{F})$ is an upper-bound on $v^*(\mathcal{F})$:

$$v^*(\mathcal{F}) \leq \overline{v}(\mathcal{F}), \quad \forall\mathcal{F} \subset \mathbb{R}^m \tag{4}$$

Furthermore, in [11], the authors 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 $P\overline{v}(\mathcal{F}) \leq V_{th}$.

Thus, a safe container $\mathcal{F}$ is useful because, due to (4), it guarantees that $Pv^*(\mathcal{F}) \leq V_{th}$, 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 [6], current containers were *specified* and the corresponding worst-case voltage drop was found by a process of optimization. In later work [10], 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 work, we build on and extend the work of [10] to the case of active grids. Some of the major results in [10] are restated below as they are necessary to understand the flow of the paper.

C. Maximal Containers

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

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

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

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

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*.

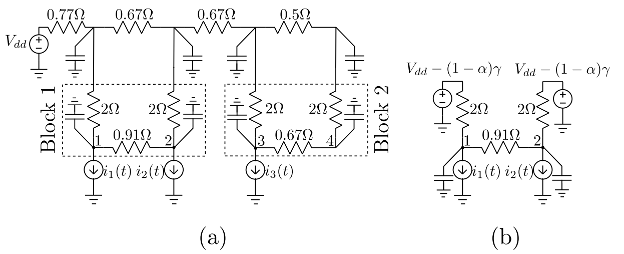
Lemma 1: [10] A container $\mathcal{J} \subset \mathbb{R}^m_+$ is safe if and only if it is a member of $\mathcal{S}$ or a subset of a member of $\mathcal{S}$.

The importance of this lemma is two-fold: 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 $\mathcal{F}(u)$. The authors in [10] show that if $V_{th,k} = 0$, for some k, then the only non-empty container in $\mathcal{S}$ is the trivial one $\mathcal{F}(0) = \{0\}$. Therefore, throughout this paper we will assume that $V_{th} > 0$.

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. 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. 1(b), 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 [10] – 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



(a) (b)

Fig. 2: (a) Simple example of a power grid with 2 blocks, and (b) block 1 in isolation with a supply value of $V_{dd}-(1-\alpha)\gamma$.

that are safe based on the voltage analysis of the full grid, which we capture as an *implicit binary decision diagram* (BDD) – this will be discussed in Section IV-B. Our approach uses an internal parameter α that becomes a "knob" 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 Lemma 6 that establishes the principle of superposition for the equivalent passive grid. In addition, we show that the block-level current containers (in Lemma 3) as well as the upper-bound on the worst-case voltage drop (in Lemma 5) have certain scalability properties in terms of the internal parameter α. These results allow us to easily manage the trade-off 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. 2(a) 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.

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 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. 2(b). 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 trade-off 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 re-derive 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 $\gamma\mathbb{1}_d$, where $\mathbb{1}_d$ 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 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:

$$Pv(t)\leq\alpha\gamma\mathbb{1}_d \tag{8}$$

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

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

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

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

The following lemma shows that, for any $\alpha>0$, $\mathcal{S}(\alpha)$ always has a current container that allows a non-zero current. This will be useful later on. *The proofs for this and all other theoretical results below are not shown, due to lack of space.*

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

2) Scalability of Current Containers: In [10], the authors 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 [10], once extended and parameterized by α as above, then applied to the grid in Fig. 2(b), for different values of α, generates the current containers shown in Fig. 3(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 [10], can be expressed in the following general form:

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

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 [10] 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:

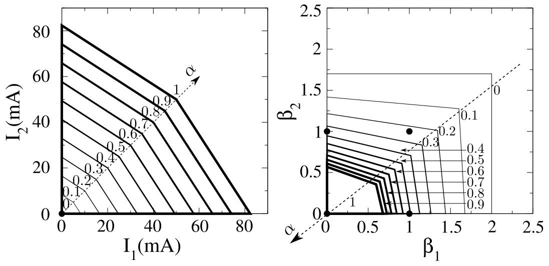
$$g(u)=\max_{I\in\mathcal{L}(u)}f(I,u) \tag{13}$$

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.

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

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 non-empty.

The lemma below 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.



(a) (b)

Fig. 3: (a) A current container $\mathcal{F}_1(\alpha)$ for the left block in Fig. 2(a) for different values of α, (b) the set of safe working modes $\mathcal{W}(\alpha)$ for different values of α under the containers generated for each block in isolation, i.e. $\mathcal{F}_1(\alpha)$ and $\mathcal{F}_2(\alpha)$.

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

$$\mathcal{L}(u^*(\alpha)) = \left\{ I \geq 0 : M'I \leq \alpha M G u^*(1) \right\} \tag{15}$$

It can be shown that the design objectives used in [10] 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. 2(b), the peak power algorithm in [10] for $\alpha = 1$ gives $u^*(1) = [100\ 100]^T$ mV. Thus, for $\alpha = 0.5$, we immediately have $u^*(\alpha) = \alpha u^*(1) = [50\ 50]^T$ mV. This gives us a scaled container $\mathcal{L}(u^*(\alpha))$.

B. Full Grid Analysis

In this section, we apply the results of Section IV-A to every block of the grid. Every block has its own current container that has the above scalability property in terms of the parameter α. The importance of this section is two-fold: 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 also has a scalability property in terms of α, culminating in the result of 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 α can be simply found by adding the scaled contribution of every block in that set for $\alpha = 1$.

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 our work 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 $(V_{dd} - (1 - \alpha)\gamma_k)$) at all nodes in block k. For every isolated block k and for any $\alpha \in [0, 1]$, define the sets $\mathcal{U}_k(\alpha)$, $\mathcal{L}_k(u_k)$, and $\mathcal{S}_k(\alpha)$, based on the analysis in Section IV-A, as follows:

$$\mathcal{U}_k(\alpha) \triangleq \{u_k \in \mathbb{R}^{n_k} : 0 \leq P_k u_k \leq \alpha V_{th,k}\} \tag{16}$$

$$\mathcal{L}_k(u_k) \triangleq \{I_k \in \mathbb{R}^{m_k} : I_k \geq 0,\ M'_k I_k \leq M_k G_k u_k\} \tag{17}$$

$$\mathcal{S}_k(\alpha) \triangleq \{\mathcal{L}_k(u_k) : u_k \in \mathcal{U}_k(\alpha)\} \tag{18}$$

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

$$g_k^*(\alpha) = \max_{u_k \in \mathcal{U}_k(\alpha)} g_k(u_k) \tag{19}$$

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

$$u_k^*(\alpha) = \alpha u_k^*(1), \qquad \forall \alpha \in [0, 1] \tag{20}$$

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

During chip design, we can set the internal parameter α 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 $\mathcal{F}_k(\alpha)$, i.e. $I_k \in \mathcal{F}_k(\alpha)$, 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 $\beta = [\beta_1 \cdots \beta_q] \in \mathbb{B}^q$ denote a working mode for the chip.

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

$$\mathcal{F}(\alpha, \beta) = \left\{ \begin{bmatrix} I_1 \\ \vdots \\ I_q \end{bmatrix} \in \mathbb{R}^m : I_k \in \left\{ \begin{array}{ll} \mathcal{F}_k(\alpha), & \text{if } \beta_k = 1 \\ \{0\}, & \text{if } \beta_k = 0 \end{array} \right. \right\} \tag{21}$$

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 $\upsilon(\alpha, \beta)$ 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:

$$\upsilon(\alpha, \beta) = P\overline{v}(\mathcal{F}(\alpha, \beta)) = PG^{-1}A\,\mathrm{emax}_{I \in \mathcal{F}(\alpha,\beta)}(M'I) \tag{22}$$

Notice that the current vector I that is used to carry out the maximization in (22) has the vector form defined in (21), i.e. its components $I_1, I_2, \ldots, I_q$ correspond to the current sources attached to block 1, block 2, $\ldots$, and block q. The columns of M' in (22) correspond to the different components of I, so that we can partition M' as follows:

$$M' = \begin{bmatrix} Z_1 & Z_2 & \cdots & Z_q \end{bmatrix} \tag{23}$$

where Z_k is an $n \times m_k$ matrix that is multiplied by I_k in (22).

For any $\alpha \in [0, 1]$, let

$$\upsilon_k(\alpha) \triangleq PG^{-1}A\,\mathrm{emax}_{I_k \in \mathcal{F}_k(\alpha)}(Z_k I_k) \tag{24}$$

and

$$V(\alpha) = [\upsilon_1(\alpha) \cdots \upsilon_q(\alpha)] \tag{25}$$

Notice that for any $\alpha \in [0, 1]$, we have $I_k \geq 0$, $\forall I_k \in \mathcal{F}_k(\alpha)$, 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 $\upsilon_k(\alpha) \geq 0$. Therefore, $V(\alpha) \geq 0, \forall \alpha \in [0, 1]$. Furthermore, the following lemma shows that if $\alpha > 0$, then $V(\alpha) > 0$. This will be useful in Section V.

Lemma 4: For any $\alpha \in (0, 1]$, 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 $\mathrm{emax}(\cdot)$ operations as in (24), i.e. $q \times n$ linear programs. The lemma below shows that, under a certain mild condition on $g_k(\cdot)$, $V(\alpha)$ has a scalability property in terms of α.

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

Based on the above lemma, for any $\alpha \in [0, 1]$, we have

$$V(\alpha) = V(1)\alpha \qquad (26)$$

which is clearly much faster to compute than solving q instances of (24) for every required value of α.

3) Superposition: It is practically impossible to solve (22) for every required β, as this could lead to combinatorial explosion in the required values of β. The lemma below establishes the principle of superposition for the equivalent passive grid.

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

$$v(\alpha, \beta) = \textstyle\sum_{k=1}^{q} \beta_k v_k(\alpha) = V(\alpha)\beta \qquad (27)$$

The importance of the above lemma is that, for a given value of $\alpha \in [0, 1]$, $v(\alpha, \beta)$ can be found for different working modes β by a simple matrix-vector multiplication between $V(\alpha)$ and β, which is faster than solving (22) for every required β.

This leads to our main theoretical result and the main reason behind the computational efficiency of our work, as follows. Combining the results of Lemmas 5 and 6, for any $\alpha \in [0, 1]$ and $\beta \in \mathbb{B}^q$, under a certain mild condition on $g_k(\cdot)$, we have

$$v(\alpha, \beta) = V(1)\alpha\beta \qquad (28)$$

Thus, we can 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 $\alpha \in [0, 1]$ and $\beta \in \mathbb{B}^q$ by solving $v_k(1)$, defined in (24), $\forall k \in \{1, \ldots, q\}$, constructing $V(1)$, defined in (25), and performing a single matrix-vector multiplication, as in (28). This is clearly much faster than solving (22) for every required value of α and β.

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 $v_{th}(\beta)$, which will then be used to check if the grid is safe by comparing $v(\alpha, \beta)$ to $v_{th}(\beta)$, as we will prove in Lemma 7. Based on the working mode β, the entries of $v_{th}(\beta)$ 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 $v_{th}(\beta)$ 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 ith node of interest belongs to the jth block, otherwise $T_{ij} = 0$. Also, let $v_{th}(\beta) = V_{th} + \rho T(\mathbb{1}_q - \beta)$, where $\rho > 0$ is a large number. It is enough for ρ to be larger than $\|V(1)\mathbb{1}_q\|_\infty$. 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:

$$v_{th}(\beta) = V_{th} + \rho T(\mathbb{1}_q - \beta) \geq V_{th} > 0 \qquad (29)$$

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

For any $\beta \in \mathbb{B}^q$ such that $V(\alpha)\beta \leq v_{th}(\beta)$, 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 $\mathcal{F}_k(\alpha)$, i.e.

$$\mathcal{W}(\alpha) \triangleq \{\beta \in \mathbb{B}^q : V(\alpha)\beta \leq v_{th}(\beta)\} \qquad (30)$$

which can be captured by a BDD.

To better visualize this, consider again the example of Fig. 2(a). In Fig. 3(b), we show the set $\mathcal{W}'(\alpha) = \{x \in \mathbb{R}^q : V(\alpha)x \leq v_{th}(x)\}$ for different values of α. Notice that, for any $\alpha \in [0, 1]$, $\mathcal{W}(\alpha)$ consists of the Boolean vectors $\beta \in \mathbb{B}^q$ that lie inside the space $\mathcal{W}'(\alpha)$. So, $\mathcal{W}(0.3) = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$ and $\mathcal{W}(0.2) = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$, as shown in Fig. 3(b).

So far, any $\alpha \in [0, 1]$ will give us the required block-level current containers $\mathcal{F}_k(\alpha) = \mathcal{L}_k(u_k^*(\alpha))$ and the corresponding set of safe working modes $\mathcal{W}(\alpha)$, as defined in (17) and (30).

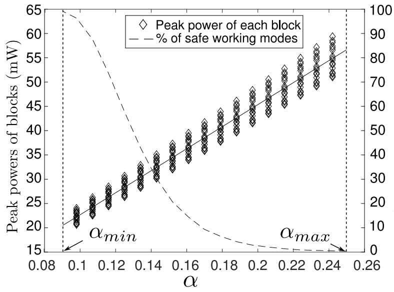


Fig. 4: Trade-off for different values of α.

V. Application

Referring again to the example of Fig. 2(a), notice that $\mathcal{F}_1(0.3) \supset \mathcal{F}_1(0.2)$, as shown in Fig. 3(a). The same case holds for the right block of Fig. 2(a) but is omitted due to lack of space. Furthermore, $\mathcal{W}(0.3) \subset \mathcal{W}(0.2)$, as shown in Fig. 3(b). Therefore, $\alpha = 0.3$ allows more flexibility to the underlying circuitry but allows less flexibility in terms of the number of safe working modes, as compared to $\alpha = 0.2$. There is a clear trade-off for different values of α. In Fig. 4, we show the trade-off achieved for different values of α on a 5k node grid with 16 blocks. 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 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, i.e. $\alpha_{min} \leq \alpha \leq \alpha_{max}$. Based on the design objectives, we will either choose α_{max} to allow large local power budgets at the cost of small number of safe working modes, or α_{min} to allow 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 range of α based on any specifications that may be known about the design at an early stage, thus achieving different trade-offs for chip operation. In a sense, these specifications will help reduce the range of α to a range that reflects design knowledge.

The user can enforce some working modes to be allowed during chip operation, which we can incorporate as in (33). Also, the user can enforce any local current/power budgets to satisfy some constraints, which we can incorporate as in (37). Assuming that the working modes constraints and the current/power constraints are consistent and feasible, so that there exists an $\alpha \in [0,1]$ that satisfies (33) and (37), then we can define the allowable range of α as follows:

$$\alpha_{min} \le \alpha \le \alpha_{max} \tag{31}$$

where $\alpha_{min} = \max(0, \Delta_{min})$ and $\alpha_{max} = \min(\Delta_{max}, \overline{\alpha}, 1)$.

1) Working Modes Constraints: Suppose we have some knowledge about the working modes of the circuit, for example, if there exists some dependencies 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 linear constraint on α, i.e.

$$\alpha \le \overline{\alpha} \tag{32}$$

We will assume that for any $\beta \in \mathcal{W}_0$, we have $\beta \neq 0$, as we already have $0 \in \mathcal{W}(\alpha), \forall \alpha \in [0,1]$, because $V(\alpha)0 = 0 \le v_{th}(0)$. For any $\beta \in \mathcal{W}_0$, let $w(\beta) = V(1)\beta$. Recall that $V(1) > 0$, due to Lemma 4, which, combined with $\beta \ge 0$ and $\beta \neq 0, \forall \beta \in \mathcal{W}_0$, gives $w(\beta) > 0, \forall \beta \in \mathcal{W}_0$. The following lemma transforms the constraint $\mathcal{W}_0 \subseteq \mathcal{W}(\alpha)$ into a linear inequality on α.

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

$$\alpha \le \overline{\alpha}, \qquad \text{where} \quad \overline{\alpha} \triangleq \min_{\forall \beta \in \mathcal{W}_0, \forall i} \left(v_{th,i}(\beta)/w_i(\beta) \right) \tag{33}$$

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 linear inequality on α, i.e.

$$\Delta_{min} \le \alpha \le \Delta_{max} \tag{34}$$

Define $\psi_k(\alpha)$ to be the largest instantaneous peak power dissipation achievable under $\mathcal{F}_k(\alpha)$, which is conservatively approximated by

$$\psi_k(\alpha) = V_{dd} \max_{I_k \in \mathcal{F}_k(\alpha)} \left(\mathbb{1}_{m_k}^T I_k \right) \tag{35}$$

Recall that for any $\alpha \in [0,1]$, $\mathcal{F}_k(\alpha)$ is non-empty, so that $\psi_k(\alpha) \ge 0$ is well-defined. Also, recall that $\mathcal{F}_k(1)$ allows a non-zero current $\hat{I}_k \ge 0$, so that $\psi_k(1) \ge V_{dd} \left(\mathbb{1}_{m_k}^T \hat{I}_k \right) > 0$.

The simplest bounds are on the minimum and peak-power, referred to as *local constraints*, such as $\psi_{lb} \le \psi(\alpha) \le \psi_{ub}$ where $\psi(\alpha) = [\psi_1(\alpha) \cdots \psi_q(\alpha)]^T$ is a $q \times 1$ vector of the peak power dissipation that each block can safely support and ψ_{lb} and ψ_{ub} are 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 $s_{lb} \le F\psi(\alpha) \le s_{ub}$ 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 $F \ge 0$ has no row with all zeros, s_{lb} and s_{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:

$$p_{lb} \le U\psi(\alpha) \le p_{ub} \tag{36}$$

where $p_{lb} = \begin{bmatrix} \psi_{lb} \\ s_{lb} \end{bmatrix}$, $p_{ub} = \begin{bmatrix} \psi_{ub} \\ s_{ub} \end{bmatrix}$, and $U = \begin{bmatrix} \mathbb{I}_q \\ F \end{bmatrix}$.

The following lemma 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 $c > 0$, $u \in \mathbb{R}^{n_k}$ and $\forall k \in \{1, \ldots, q\}$, then $\psi(\alpha) = \alpha\psi(1), \forall \alpha \in [0,1]$.

The following lemma transforms the user-specified power constraints into a linear inequality on α. Notice that $U \ge 0$ and U has no row with all zeros, and $\psi(1) > 0$. Therefore, $U\psi(1) > 0$.

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

$$\Delta_{min} \le \alpha \le \Delta_{max} \tag{37}$$

where $\Delta_{min} = \max_{\forall i} \left(\frac{p_{lb,i}}{U\psi(1)|_i} \right)$ and $\Delta_{max} = \min_{\forall i} \left(\frac{p_{ub,i}}{U\psi(1)|_i} \right)$.

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 jth 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. (31). We can formulate this as the following linear program:

$$\begin{aligned} \sigma^* = \text{Maximize} \quad & \mathbb{1}_q^T \psi(\alpha) \\ \text{subject to} \quad & \alpha_{min} \le \alpha \le \alpha_{max} \end{aligned} \tag{38}$$

Let α_p be a scalar at which the above maximization attains its maximum. In other words, $\alpha_{min} \le \alpha_p \le \alpha_{max}$ such that $\mathbb{1}_q^T \psi(\alpha_p) = \sigma^*$. Because the feasible region of (38) is non-empty, it follows that α_p is well-defined. Therefore, the resulting block-level containers are $\mathcal{F}_k(\alpha_p)$ which describe the following current constraints:

$$i_k(t) \ge 0 \tag{39}$$

$$M_k' i_k(t) \le \alpha_p M_k G_k u_k^*(1) \tag{40}$$

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

$$\mathcal{W}(\alpha_p) = \{\beta \in \mathbb{B}^q : \alpha_p V(1)\beta \le v_{th}(\beta)\} \tag{41}$$

Recall that, from Lemma 9, we have $\psi(\alpha) = \alpha\psi(1)$. Notice that for any $\alpha_{min} \le \alpha \le \alpha_{max}$, we have $\alpha_{max}\psi(1) \ge \alpha\psi(1)$, because $\psi(1) \ge 0$, or equivalently, we have $\psi(\alpha_{max}) \ge \psi(\alpha)$, for any $\alpha_{min} \le \alpha \le \alpha_{max}$. It follows that $\mathbb{1}_q^T \psi(\alpha_{max}) \ge \mathbb{1}_q^T \psi(\alpha)$, for any $\alpha_{min} \le \alpha \le \alpha_{max}$, so that $\mathbb{1}_q^T \psi(\alpha_{max}) = \sigma^*$. Therefore, there is no need to solve (38) and we can simply set $\alpha_p = \alpha_{max}$.

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:

$$\begin{aligned} \mu^* = \text{Maximize} \quad & |\mathcal{W}(\alpha)| \\ \text{subject to} \quad & \alpha_{min} \le \alpha \le \alpha_{max} \end{aligned} \tag{42}$$

TABLE I: Comparison of the two design objectives

Power Grid		Power Constraints	BDD Constraints		Maximum Local Power			Maximum Working Modes			Runtime	
Nodes	Blocks	Min. Avg. Power (mW)	# of working modes	Max. ON[a]	α_p $(\times 10^{-3})$	$P(\alpha_p)$ (mW)	$\omega(\alpha_p)$	α_w $(\times 10^{-3})$	$P(\alpha_w)$ (mW)	$\omega(\alpha_w)$	Isolated Block Analysis	Full Grid Analysis
5k	4	102	4	1	150	144	25.00%	108	102	56.25%	51 msec	14 sec
5k	9	38	18	2	174	71	9.76%	94	38	92.96%	103 msec	2 sec
5k	16	20	16	1	290	66	0.10%	90	20	99.40%	171 msec	1 sec
53k	16	164	29	2	101	239	48.29%	69	164	71.31%	279 msec	3.6 min
53k	25	102	43	4	113	172	1.04%	68	102	84.75%	338 msec	1.9 min
50k	36	66	53	3	154	146	-	70	66	-	447 msec	45 sec
600k	64	413	84	7	65	446	-	60	413	-	1 sec	2.9 hr
1.3M	100	568	120	11	68	655	-	59	568	-	3 sec	11.2 hr

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

Let α_w be a scalar at which the above maximization attains its maximum. In other words, $\alpha_{min} \leq \alpha_w \leq \alpha_{max}$ such that $\mathcal{W}(\alpha_w) = \mu^*$. Because the feasible region of (42) is non-empty, it follows that α_w is well-defined. Therefore, the resulting block-level containers are $\mathcal{F}_k(\alpha_w)$ which describe the following current constraints:

$$i_k(t) \geq 0 \tag{43}$$

$$M'_k i_k(t) \leq \alpha_w M_k G_k u^*_k(1) \tag{44}$$

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

$$\mathcal{W}(\alpha_w) = \{\beta \in \mathbb{B}^q : \alpha_w V(1)\beta \leq v_{th}(\beta)\} \tag{45}$$

The following lemma shows that α_{min} attains the maximum of (42), so that we can let $\alpha_w = \alpha_{min}$ without solving (42).

Lemma 11: If $g_k(cu) = cg_k(u)$, for any real number $c > 0$, $u \in \mathbb{R}^{n_k}$, and $\forall k \in \{1, \ldots, q\}$, then $|\mathcal{W}(\alpha_{min})| = \mu^*$.

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, number of blocks, 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 1V 45 nm CMOS technology. Table I shows the characteristics of a number of test grids. The grids generated have 8 metal layers in total with 6 metal layers in the global grid. Also, the number of nodes attached to current sources range from 800 to 214k. All results were obtained using a hyperthreaded 12-core 3GHz Linux machine with 128GB of RAM. The optimizations were performed using MOSEK optimization package [12]. All the linear systems are solved using Cholmod [13] from SuiteSparse [14]. In our implementation, we use *Pthread* to parallelize the computation and take advantage of the 12-cores machine. The runtime breakdown of our approach, i.e. the isolated block analysis and the full grid analysis, is shown in columns 12-13 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 [10] as the design objective for all the blocks.

Table I compares the results of using α_p (Section V-B) and α_w (Section V-C) based on user-specified constraints. In column 3, 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 3. Furthermore, in columns 4-5, 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 $\mathcal{F}_k(\alpha)$. Also, denote by $\omega(\alpha)$ the percentage of the working modes that are safe under block containers $\mathcal{F}_k(\alpha)$. To study the difference between the the generated block containers and the $\mathcal{W}(\cdot)$ using α_p and α_w, we found the average of the peak powers of all the blocks under $\mathcal{F}_k(\alpha_p)$ and $\mathcal{F}_k(\alpha_w)$, 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 53k node grid with 25 blocks, the average of the peak powers for all blocks under $\mathcal{F}_k(\alpha_p)$ and $\mathcal{F}_k(\alpha_w)$ are 172 mW and 102 mW, respectively and the percentage of safe working modes under $\mathcal{W}(\alpha_p)$ and $\mathcal{W}(\alpha_w)$ are 1.04% and 84.75%, 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 trade-off for the design team.

VII. Conclusion

Analysis and verification of active PDNs is crucial to ensure voltage integrity. With active devices, most traditional techniques are ill-equipped to verify the PDN. We propose a framework to generate block-level circuit current constraints as well as an implicit binary decision diagram that helps identify the safe working modes. Subject to user-guidance, we then propose two design objectives that exploit the trade-off between how many blocks are ON simultaneously and how big the power budgets of individual blocks are.

References

[1] J. Li and et al., “Thermal-aware task scheduling in 3D chip multiprocessor with real-time constrained workloads,” *ACM Trans. on ECS*, vol. 12, no. 2, pp. 24:1–24:22, 2013.

[2] A. K. Coskun and et al., “Static and dynamic temperature-aware scheduling for multiprocessors SoCs,” *IEEE Trans. on VLSI*, vol. 16, no. 9, pp. 1127–1140, 2008.

[3] H. Khdr and et al., “mDTM: Multi-objective dynamic thermal management for on-chip systems,” in *Design, Automation & Test in Europe*, 2014, pp. 330:1–330:6.

[4] A. Krstic and K.-T. Cheng, “Vector generation for maximum instantaneous current through supply lines for CMOS circuits,” in *IEEE DAC*, 1997, pp. 383–388.

[5] S. Pant and et al., “A stochastic approach to power grid analysis,” in *IEEE DAC*, 2004, pp. 171–176.

[6] F. N. Najm, “Overview of vectorless/early power grid verification,” *IEEE ICCAD*, pp. 670–677, 2012.

[7] Y. Wang and et al., “A realistic early-stage power grid verification algorithm based on hierarchical constraints,” *IEEE Trans. on CAD*, vol. 31, no. 1, pp. 109–120, 2012.

[8] H. Zhu and et al., “Efficient transient analysis of power delivery network with clock/power gating by sparse approximation,” *IEEE Trans. on CAD*, vol. 34, no. 3, pp. 409–421, 2015.

[9] Z. Zeng, Z. Feng, and P. Li, “Efficient checking of power delivery integrity for power gating,” in *IEEE ISQED*, 2011, pp. 1–8.

[10] Z. Moudallal and F. N. Najm, “Generating current budgets to guarantee power grid safety,” *IEEE Trans. on CAD*, vol. 35, no. 11, pp. 1914–1927, 2016.

[11] M. Fawaz and F. N. Najm, “Accurate verification of RC power grids,” in *Design, Automation, & Test in Europe*, 2016, pp. 814–817.

[12] MOSEK ApS, *The MOSEK C Optimizer API Manual. Version 7.1 (Revision 28).*, 2015. [Online]. Available: http://docs.mosek.com/7.1/toolbox/index.html

[13] Y. Chen and et al., “Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/downdate,” *ACM Trans. on Mathematical Software*, vol. 35, no. 3, pp. 22:1–22:14.

[14] T. Davis, *Suitesparse 4.4.6*, [Online]. [Online]. Available: http://faculty.cse.tamu.edu/davis/suitesparse.html