

GENERATING CIRCUIT CURRENT CONSTRAINTS TO GUARANTEE
POWER GRID SAFETY

by

Zahi Moudallal

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Sciences
Graduate Department of Electrical & Computer Engineering
University of Toronto

© Copyright 2014 by Zahi Moudallal

Abstract

Generating Circuit Current Constraints to Guarantee Power Grid Safety

Zahi Moudallal

Master of Applied Sciences

Graduate Department of Electrical & Computer Engineering

University of Toronto

2014

Verification of the chip power distribution network is a critical step in modern IC design. Vectorless verification, developed as an alternative to simulation-based methods, requires user-specified current constraints and checks if the worst-case voltage drops at all grid nodes are below user-specified thresholds. However, obtaining/specifying the current constraints remains a burdensome task for design teams. In this thesis, we define and address the *inverse* problem: given a grid, we will *generate* circuit current constraints which, if satisfied by the underlying logic, will guarantee grid safety. This approach has many potential applications, including various grid quality metrics, and power grid-aware placement and floorplanning. We give a rigorous problem definition and develop some key theoretical results related to maximality of the current space. We then develop two constraints generation algorithms that target key quality metrics like the peak total power allowed by the grid and the uniformity of the temperature distribution.

Acknowledgements

“Why are people sad? That’s simple. They are the prisoners of their personal history. Everyone believes that the main aim in life is to follow a plan. They never ask if that plan is theirs or if it was created by another person. They accumulate experiences, memories, things, other people’s ideas, and it is more than they can possibly cope with. And that is why they forget their dreams.” - Paulo Coelho

First and foremost, I give thanks and praise to God, *Alhamdu lillah*, for all his blessings that guided me to finish this work.

I am deeply grateful to a number of people without whom this work would not have been finished. I am forever indebted to these people for providing me with their immense support and guidance, and with whom I have established a priceless friendship that I cherish.

I am truly thankful for my supervisor Professor Farid Najm for his steady support and guidance throughout the past two years. It has been an honor and a pleasure to work with Professor Najm, who is a role model for me, for his constant beacon of intellect, inspiration, and support both at the academic and personal level. Thank you professor; this research would not have been possible without your thought leadership, great efforts, and mentorship.

I would also like to thank Professors Mireille Broucke, Jianwen Zhu, and Wei Yu from the ECE department at the University of Toronto for their time and effort in reviewing this work.

During my internship at the University of Toronto in Summer 2011, I was very lucky to work with Dr. Nahi Abdul Ghani who I benefited from his deep insight and vast knowledge. I owe him a big thank you. I would also like to thank Abhishek for his assistance and guidance during that period. I wish them both a great success in their career. Many thanks go to my friend and my colleague Sandeep Chatterjee who I have enjoyed working with throughout my degree program. I appreciate the long discussions we had and his advice and assistance. I wish him the best in his future endeavor.

I am sincerely grateful for Mohammad B. Fawaz for being a great colleague and a true friend over the past two years. I will never forget the invaluable moments that we shared together inside and outside the office. I thank him for all the help, encouragement, and support that he provided me. I can not imagine how hard the last two years would have been without his constant assistance and inspiration. I wish him a great future.

A special acknowledgment go to my best friend Noha Sinno. I am very indebted to

her for her presence in my life during the past two years, the unwavering support during critical months of my thesis, and her ability to listen to me nagging about research. Thank you for the precious moments that we spent together, I would not be the same without you.

I am also thankful for all the office mates in Pratt building, room 392, for the friendly environment. I wish you all the best.

This journey has come to an end, with all its hard times and winding roads, which I could not have passed without the unconditional support of my family. My greatest gratitude goes to my wonderful parents Marwan and Amina, to whom I dedicate this work. Their continuous sacrifices and pride in my success have raised my ambition and provided me with unwavering inspiration. I owe everything to you, no way I can pay you back. I would also like to thank my brothers Shadi and Hani, my sister Nassma, my brother-in-law Walid Itani, my sister-in-law Sara Kurdi, and my one year old nephew Samir for sharing our life together which have always filled my heart with comfort and joy.

Thank you all and God bless you.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Contributions	3
1.3	Organization	4
2	Background	5
2.1	Introduction	5
2.2	The Class of \mathcal{M} -matrices	5
2.3	The Power Grid	6
2.3.1	Overview	6
2.3.2	Power Grid Model	8
2.3.3	Time Discretization	11
2.4	Power Grid Verification	11
2.4.1	Vector-Based Power Grid Verification	12
2.4.2	Vectorless Power Grid Verification	15
2.4.3	The Constraint-Based Vectorless Framework for Power Grid Verification	16
3	Maximal Safe Containers	22
3.1	Introduction	22
3.2	Problem Definition	23
3.3	Safe Containers	24
3.4	Maximizing Volume	26
3.4.1	Constraints Templates	26
3.4.2	Largest Volume Container	28
3.5	Maximal Containers	29
3.5.1	Feasible	29
3.5.2	Extremal	30

3.5.3	Irreducible	31
3.5.4	Maximality	34
3.6	Conclusion	35
4	Constraints Generation Algorithms	36
4.1	Introduction	36
4.2	Peak Power Dissipation	36
4.3	Uniform Current Distribution	39
4.4	Results	43
4.5	Conclusion	47
5	Conclusion and Future Work	49
	Bibliography	51

List of Tables

4.1	Power Grid Specifications	44
4.2	Results of both LPs	45
4.3	Results of each LP using the other container	45

List of Figures

1.1	Technology trends based on ITRS 2003-2005.	2
2.1	A simplified model of the PDN	7
2.2	An RC model of a power grid	9
2.3	Simple RC grid	10
2.4	Hierarchical power network analysis	14
2.5	Flow diagram of the Stochastic approach in [1]	15
4.1	An example of a power grid with 4 nodes, 2 current sources, and $V_{th} = [110 \ 100 \ 95 \ 105]^T$ (units of mV).	39
4.2	An example of $\mathcal{F}(u_p)$ and $\mathcal{F}(u_s)$	40
4.3	An illustration of perpendicular distances to hyperplanes.	42
4.4	CPU time of both LPs and SPAI versus the number of grid nodes.	46
4.5	Peak Power Dissipation problem	47
4.6	Contour plots for peak power density across the layout and the corresponding histograms. The color bar units are mA/cm^2	48

Chapter 1

Introduction

1.1 Motivation

A well-designed chip power/ground network should deliver well-regulated voltages at all grid nodes in order to guarantee correct logic functionality at the intended design speed. However, with the continued scaling of semiconductor technology, there's been a corresponding increase in chip operating frequency and power dissipation. Fig. 1.1 illustrates feature size, clock frequency, and maximum power dissipation of a high-performance chip based on the 2003-2005 International Technology Roadmap for Semiconductors (ITRS) data. As a result, modern high-performance integrated circuits often feature large switching currents that flow in the power and ground networks, causing excessive supply voltage variations that put both circuit performance and reliability at risk. Therefore, efficient verification of power grids is a necessity in modern chip design. We will use the term “power grid” to refer to either the power or ground distribution networks. In this work, we focus on RC power grids, but we are working to extend this to the RLC case in future.

Power grid verification problem has been extensively addressed in the recent past introducing novel methods and innovative ideas. A key concern for addressing this problem is that verification should be applied at an early stage of the design flow, when it is still possible to modify the grid. Generally, there are two approaches for power grid verification: simulation-based and vectorless methods. A major drawback of simulation-based methods is that they require detailed information on the circuit currents which are not available early in the design flow. A state-of-the-art verification framework was proposed in [2] that does not require full knowledge of the circuit currents. This method relies on information that may be available at an early stage of the design, in the form of *current constraints*. The idea behind this is to capture circuit uncertainty via design specs or

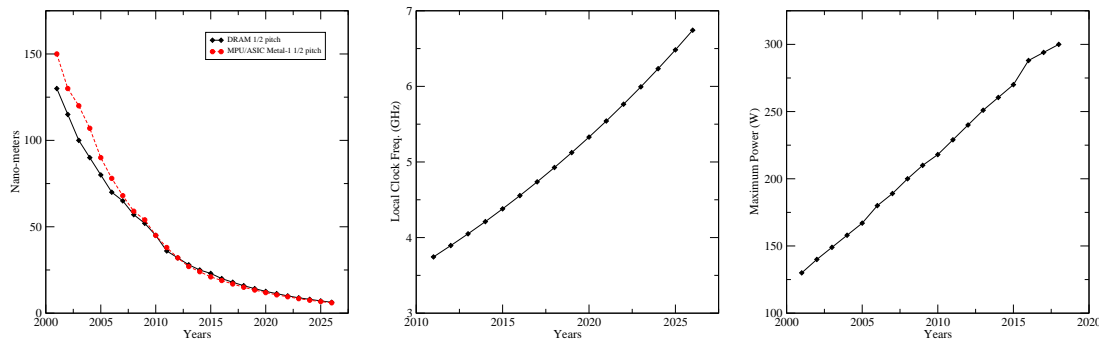


Figure 1.1: Technology trends based on ITRS 2003-2005.

power budgets known in the early design stages. This type of approach, referred to as *vectorless* verification, consists of finding the worst-case voltage fluctuations achievable 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.

This framework has been fully developed over the last decade [3], but a key question remains: how would one obtain/specify the current constraints? In our work with colleagues in the industry, this point always comes up and remains a hurdle to adoption of these methods! Providing the current constraints is a burdensome task for design teams, and this thesis is aimed at addressing this problem.

Instead of the traditional approach of expecting users to provide current constraints that would be used to check if the grid is safe (what one might call the *forward* problem), we propose to solve the *inverse* problem: given a grid and the allowed voltage drop thresholds at all grid nodes, we will *generate* circuit current constraints which, if satisfied by the underlying circuitry, would guarantee grid safety. This is a significant departure from previous work and represents the first time, in our experience, that this problem has been addressed.

We believe that these current constraints would encapsulate much useful information about the grid. First of all, the availability of current constraints at an early stage of the design process provides a way to drive the rest of the design process, because these are essentially power budgets for the logic blocks under the grid. If all design teams respect these budgets throughout the design flow, then the grid is *safe by construction* at the end of the design. If the constraints impose too severe a budget on a certain block in some corner of the die, for example, then one can address the problem early on by modifying

the grid, while it is still easy to do so, and generating a fresh set of constraints.

Alternatively, if the budgets are too severe for a candidate layout location of a high level block, then perhaps the floorplan needs to be reconsidered. Indeed, the constraints can be used to drive automated floorplanning as well as placement, so that grid-aware placement may become feasible, something that has never been done before.

Furthermore, as we will see below, our work provides a high-level and early way to *qualify* the candidate grid and assess how good it is relative to various *quality metrics*. For example, using our approach, one can check what maximum level of peak power dissipation for the whole die (or for a major part of the die) can be safely supported by the candidate power grid. If the design has a peak power budget of 100 Watts, for example, then the grid must be able to support the corresponding level of peak current in the underlying circuit, and we will see that we can verify that type of objective. Alternatively, one may be interested in spreading the power dissipation across the die in some uniform fashion, in order to avoid thermal hot-spots. We can target that objective by looking for constraints that spread the circuit current budget uniformly across the die area. Modifications of the grid may be required to allow for that, and our engine can identify whether these are needed, very early in the design process.

1.2 Contributions

The objective of this work is to define and address the inverse problem of an existing early power grid verification framework, namely the constraints-based framework: we provide circuit current constraints for power grids, knowing their allowed voltage fluctuation thresholds at each node. If satisfied, these constraints ensure the safety of the grid. Traditionally, the design team would rely on their expertise to specify current constraints. However, in this thesis we provide a systematic method to generate these constraints. This work builds on the constraints-based framework that was proposed in [2] and relies on the upper-bound to the worst-case voltage drop that was first derived in [4], but comprises a significant departure from previous work and represents the first time, in our experience, that this problem has been addressed.

This thesis develops some key theoretical results as well as algorithms for constraints generation. The contributions of this work are summarized below:

1. Establish a necessary and sufficient condition under which a current space would guarantee grid safety. There are infinitely many current spaces that satisfy this condition, which explains the second contribution.

2. Characterize the most desirable current spaces, which we refer to as “maximal”, in a sense that allows more flexibility in the circuit current loading.
3. Develop two algorithms for constraints generation that target key quality metrics such as peak power dissipation that the power grid can safely support and uniformity of temperature distribution. These algorithms allow a high level and an early assessment of the grid design.

1.3 Organization

The remainder of this thesis is outlined as follows: In chapter 2, we present a brief overview of the background material required for later chapters. Additionally, we describe the power grid model and review some major published works that focus on power grid integrity verification techniques. Chapter 3 contains the bulk of the theoretical contribution of this thesis. We define the problem of generating current constraints that guarantee grid safety and study a highly desirable property of current spaces, i.e., “maximality”. Chapter 4 proposes a high-level and early way to qualify the candidate grid design and assess how good it is relative to various quality metrics and develops two algorithms that generate current constraints targeting these quality metrics. We conclude giving future research directions in chapter 5.

Chapter 2

Background

2.1 Introduction

This chapter provides a brief overview of the background material necessary for the research presented in later chapters. We first review properties of a special class of matrices that will be crucial to the analysis to follow. We then switch gears to provide an overview of the power distribution network and the main parasitic effects in modern chip design, after which we describe the power grid model and derive the system equations. Subsequently, we introduce the power grid verification problem and discuss several research works that have contributed to the advancement of vector-based and vectorless methods. Finally, a vectorless technique that is of central importance to this work, namely the constraints-based framework, is reviewed in details along with several publications that suggest improvements thereon.

2.2 The Class of \mathcal{M} -matrices

We use standard definitions and results from [5, 6]. Throughout the rest of the thesis, we will use the notation $x \leq y$ (or $x < y$), for any two vectors x and y , to denote that $x_i \leq y_i$ (or $x_i < y_i$), $\forall i$, respectively. Similarly, we will use the notation $X \geq 0$ (or $X > 0$), for any matrix X , to denote that $X_{ij} \geq 0$ (or $X_{ij} > 0$), $\forall i, j$, respectively.

Definition 1. [5] *A square matrix G is called an \mathcal{M} -matrix if:*

$$g_{ij} \leq 0, \forall i \neq j \quad \text{and} \quad \Re(\lambda_i) > 0, \forall i \quad (2.1)$$

where g_{ij} is the (i, j) th element of G , λ_i is an eigenvalue of G , and $\Re(\lambda_i)$ is the real part of the eigenvalue λ_i .

Lemma 1. [5] *If G is an \mathcal{M} -matrix, then G^{-1} exists and its entries are non-negative, which we denote by $G^{-1} \geq 0$.*

An $n \times n$ matrix G can be used to construct a graph \mathcal{G} whose vertices are $\{1, 2, \dots, n\}$ and whose directed edges are (i, j) for every $g_{ij} \neq 0$. The graph \mathcal{G} is referred to as the *associated graph* of G .

Definition 2. *A directed graph \mathcal{G} is said to be strongly connected if it has a directed path from every vertex to every other vertex.*

Lemma 2. [5] *A square matrix G is said to be irreducible if and only if its associated graph \mathcal{G} is strongly connected.*

Definition 3. *A matrix G is said to be diagonally dominant if $|g_{ii}| \geq \sum_{j \neq i} |g_{ij}|$, $\forall i$. A matrix G is said to be strictly diagonally dominant if $|g_{ii}| > \sum_{j \neq i} |g_{ij}|$, $\forall i$.*

Definition 4. *A square matrix G is said to be irreducibly diagonally dominant if it is irreducible, it is diagonally dominant, and there is an $i \in \{1, 2, \dots, n\}$ for which $|g_{ii}| > \sum_{j \neq i} |g_{ij}|$, i.e., it is strictly diagonally dominant in at least one row.*

Lemma 3. [6] *If G is irreducibly diagonally dominant with $g_{ii} > 0$, $\forall i$, and $g_{ij} \leq 0$, $\forall i \neq j$, then G is an \mathcal{M} -matrix and its inverse has strictly positive entries, which we denote by $G^{-1} > 0$.*

2.3 The Power Grid

2.3.1 Overview

The power distribution network (PDN) of an integrated circuit is a distributed system that is responsible to deliver the appropriate supply voltage from pad locations to all on-chip logic cells. Typically, the PDN is modelled as three decoupling stages: starting at the voltage regulator module (VRM), through the motherboard, package and finally the on-die PDN. A simplified high level model of the PDN is depicted in Fig. 2.1. In this work, we will focus on the on-die part of the PDN, commonly referred to as the “power grid”. The power grid is a multiple-layer metallic mesh that connects the external power supply pins to the chip circuitry, providing the supply voltage connections to the underlying circuit components.

Ideally, the voltage levels on the grid are uniform and equal to the supply voltage (V_{dd}). However, due to the parasitics of the grid transmission lines, circuit activity,

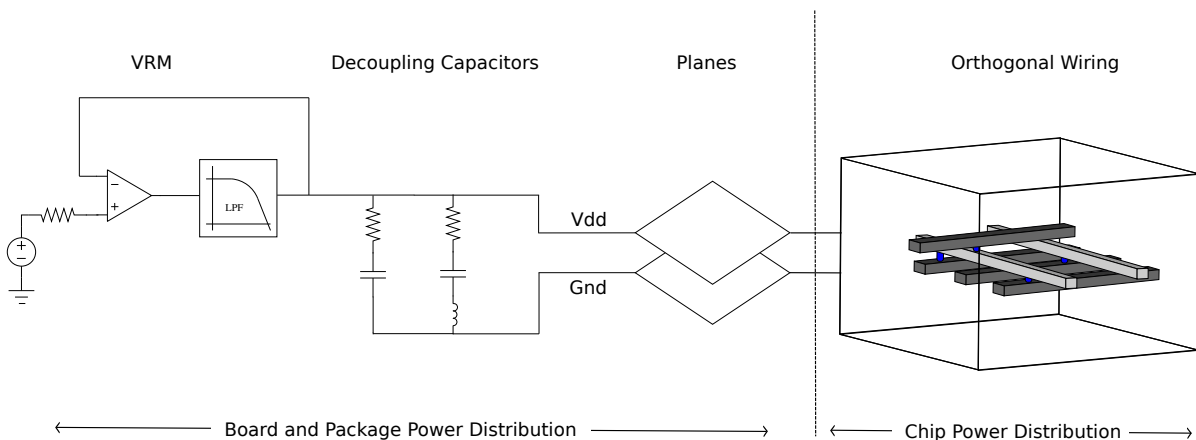


Figure 2.1: A simplified model of the PDN

coupling effects, and electromigration [7], the voltage levels on the grid might vary. This voltage drop variation on the power grid may have serious effects on the dynamic circuit behavior and timing analysis of the integrated circuit such as reducing the noise margins, slowing down the circuit, and causing soft errors [8, 9, 10, 11].

With the continued scaling of semiconductor technology, expected to arrive at 5nm in 2020, there's been a corresponding increase in chip operating frequency and power dissipation. As a result, modern high-performance integrated circuits often feature large switching currents that flow in the power and ground networks, causing excessive supply voltage variations that put both circuit performance and reliability at risk. Therefore, efficient analysis and verification of power grids is a necessity in modern chip design. In turn, this requires an accurate model of the power grid and its parasitic effects.

Several previous works have attempted to study the various parasitics of power grids and their effects on circuit behavior and consequently on the voltage drop. The main parasitic effects are: resistive, capacitive, and inductive. Due to the sheet resistance of metal lines, typically composed of copper, the power grid exhibits resistive effects. In modern microprocessors, the number of metal layers have substantially increased due to the large volume of interconnect required for grid routing. As a result, the voltage drop induced by the metal resistance becomes significant. This voltage drop, referred to as IR drop, is a major amount of the total voltage drop on the grid [12, 13], and has increased from one technology generation to the next, as metal lines widths have decreased. Capacitive effects refer to the parasitic capacitance between metal wires, MOSFET capacitance, and on-chip decoupling capacitance. The capacitance on power grids can be classified as either implicit or explicit. Implicit capacitance arise from the proximity of metal lines to one another, intrinsic capacitance of non-switching devices, and the capac-

itance between N-well and substrate [12]. Decoupling capacitance, commonly referred to as *decap*, behave as an on-chip low-pass filter that reduces noise and counters fast switching currents, keeping the supply voltage drop within a safe margin. Alone, implicit capacitance is not enough for dealing with supply voltage noise. A common practice is to insert explicit decap by filling on-die *white spaces* at strategic locations, i.e., empty areas in the chip floorplan, with devices whose gate oxide capacitance provides a decoupling effect. Therefore, any power grid model must account for effects arising from both types of capacitances. Other effects that need to be modeled are inductive effects. They are becoming increasingly significant on the power grid [14, 15], creating Ldi/dt noise. However, in this thesis we will ignore inductive effects in the power grid model for simplicity.

2.3.2 Power Grid Model

Consider an RC model of the power grid, where every grid metal branch is represented by a resistor, and where nodes are used to represent either a via or a connection of more than two branches on the same metal layer. We assume that there exists a capacitor from every node to ground and we ignore all line-to-line coupling capacitance. In a power grid, some nodes have ideal current sources (to ground) representing the currents drawn by the logic circuits tied to the grid at these nodes, while other nodes may be connected to ideal voltage sources representing the connection to the external voltage supply V_{dd} . Excluding the ground node, let the power grid consist of $n + p$ nodes, where nodes $1, 2, \dots, n$ are the nodes not connected to a voltage source, and the remaining nodes $(n + 1), (n + 2), \dots, (n + p)$ are the nodes where the p 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) direction of current is from node-to-ground. Without loss of generality, suppose that nodes attached to current sources are numbered $1, \dots, m$, where $m \leq n$. Let $H = \begin{bmatrix} I_m & 0 \end{bmatrix}^T$ be an $n \times m$ matrix where I_m is the m -dimensional identity matrix, and let $i_s(t) = Hi(t)$. Let $\nu_k(t)$ be the voltage waveform at every node k , and let $\nu(t)$ be the vector of all $\nu_k(t)$ signals. Applying Kirchhoff's Current Law (KCL) at every node, $k = 1, \dots, n$, leads to the following matrix formulation:

$$G\nu(t) + C\dot{\nu}(t) = -i_s(t) + G_0v_{dd} \quad (2.2)$$

where G , G_0 , and C are $n \times n$ matrices resulting from the application of the traditional modified nodal analysis formulation [16], and v_{dd} is a constant vector each entry of which is equal to V_{dd} . C is the $n \times n$ *diagonal* non-negative capacitance matrix, which is

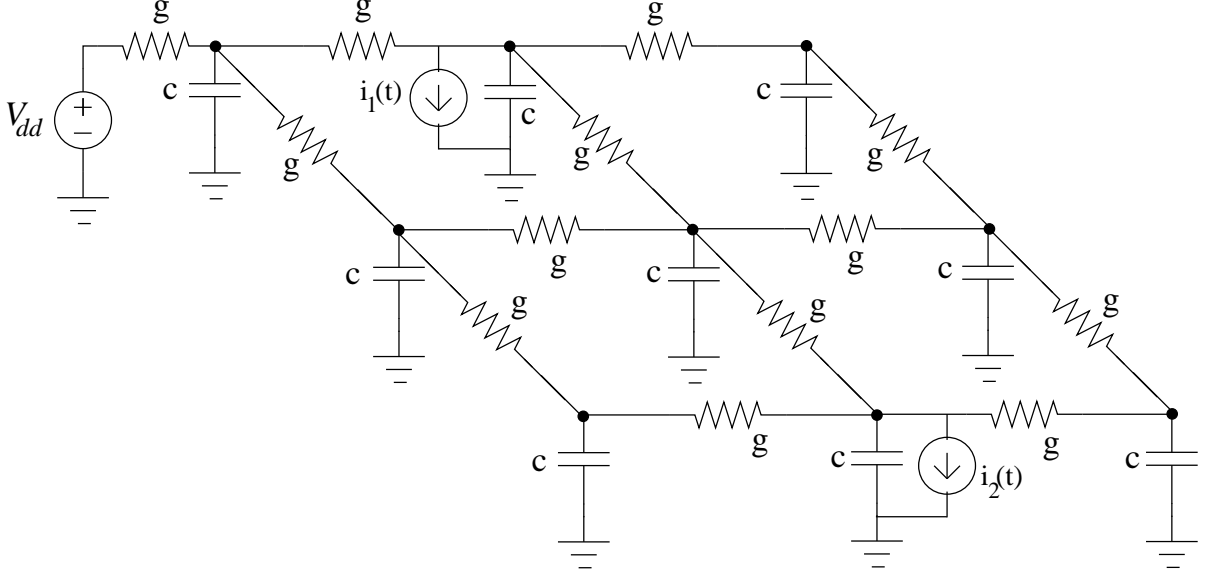


Figure 2.2: An RC model of a power grid

non-singular because every node is attached to a capacitor; G_0 is $n \times n$ matrix of the conductance elements connected to V_{dd} sources; 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. Assuming the grid is connected (so that G is irreducible) and has at least one voltage source (so G is strictly diagonally dominant in at least one row), then G is irreducibly diagonally dominant and, by Claim 3, we have that G is an \mathcal{M} -matrix with $G^{-1} > 0$. Notice that if we set all $i_{s,k}(t) = 0, \forall t$, in (2.2), and because the power grid is a stable RC system, then there is no transients and $\nu_k(t) = V_{dd}, \forall t$, so that the above system equation becomes:

$$Gv_{dd} = G_0v_{dd} \quad (2.3)$$

Because G , G_0 , and v_{dd} do not depend on $i_s(t)$, then the identity in (2.3) is valid in general. Benefiting from (2.3), we can rewrite (2.2) as:

$$G[v_{dd} - \nu(t)] - C\dot{\nu}(t) = i_s(t) \quad (2.4)$$

Define $v_k(t) = V_{dd} - \nu_k(t)$ to be the time-varying voltage drop at node k , note that $\dot{v}_k(t) = -\dot{\nu}_k(t)$, and let $v(t)$ be the $n \times 1$ vector of voltage drops, then the system equation can be written as:

$$Gv(t) + C\dot{v}(t) = i_s(t) \quad (2.5)$$

One could use the *revised* system equation described in (2.5) to directly solve for the

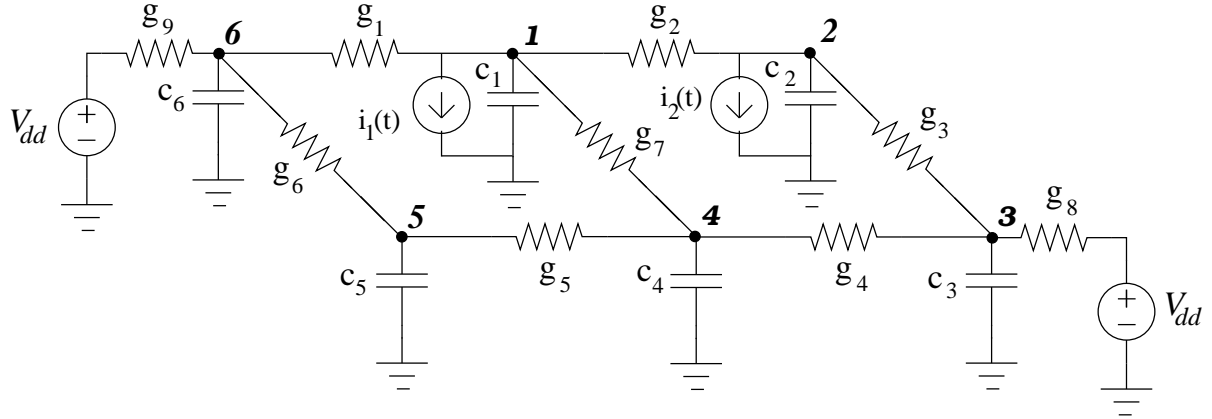


Figure 2.3: Simple RC grid

voltage drop values. It is easy to see that the system equation (2.5) is equivalent to (2.2), with all the voltage sources set to zero (short circuit) and all the current source directions reversed.

To better illustrate this, we will apply MNA to the circuit in Fig. 2.3. Notice that, the nodes attached to current sources are numbered first, so that:

$$i_s(t) = \begin{bmatrix} i_1(t) & i_2(t) & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.6)$$

$$G = \begin{bmatrix} g_1 + g_2 + g_7 & -g_2 & 0 & -g_7 & 0 & -g_1 \\ -g_2 & g_2 + g_3 & -g_3 & 0 & 0 & 0 \\ 0 & -g_3 & g_3 + g_4 + g_8 & -g_4 & 0 & 0 \\ -g_7 & 0 & -g_4 & g_4 + g_5 + g_7 & -g_5 & 0 \\ 0 & 0 & 0 & -g_5 & g_5 + g_6 & -g_6 \\ -g_1 & 0 & 0 & 0 & -g_6 & g_1 + g_6 + g_9 \end{bmatrix} \quad (2.7)$$

$$C = \begin{bmatrix} c_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_6 \end{bmatrix} \quad (2.8)$$

2.3.3 Time Discretization

We can write a discrete-time version of the system equation (2.5) using a finite-difference approximation for the derivative of a function $x(t)$, such as a Backward Euler numerical integration scheme, i.e.:

$$\dot{x}(t) \approx \frac{x(t) - x(t - \Delta t)}{\Delta t} \quad (2.9)$$

where $\Delta t > 0$ is the time step.

Applying this to (2.5) leads to:

$$v(t) = A^{-1}Bv(t - \Delta t) + A^{-1}i_s(t) \quad (2.10)$$

where $B = C/\Delta t$ is an $n \times n$ diagonal matrix with $b_{ii} > 0, \forall i$, and $A = G + B$. Because G satisfies the conditions of Claim 3, then it's clear that $A = G + B$ also satisfies the same conditions, so that A is an \mathcal{M} -matrix with $A^{-1} > 0$.

Notice that, this finite-difference approximation has modeled the voltage drop $v(t)$ as a linear function at a time interval Δt . Therefore, the choice of the time step Δt has a significant impact on the accuracy of the voltage drop values, which we will later discuss in more details.

2.4 Power Grid Verification

A well-designed chip power/ground network should deliver well-regulated voltages at all grid nodes in order to guarantee correct logic functionality at the intended design speed. As discussed earlier, the power grid exhibits a variation in the voltage levels due to parasitics. A large drop in supply voltage may lead to timing violations or logic failure. Therefore, to verify the grid safety one should check that the supply voltage is within an acceptable range under all possible current traces, i.e. an exhaustive simulation of the power grid under all possible input current waveforms. One would like, however, to verify grid safety without performing a brute-force approach. This is because an exhaustive simulation of the grid is infeasible. Large body of research work was devoted to develop efficient methods for finding the voltage drop on the power grid.

Generally, there are two approaches to verify grid safety: vector-based and vectorless approaches. Vector-based methods, also known as simulation-based methods or pattern dependant methods, involve simulating the grid under a set of current waveforms drawn by the underlying non-linear circuit. Vectorless methods, also known as pattern independent methods, attempt to find conservative bounds on the voltage drops without

knowledge of the input vectors.

In this section, we will review several state-of-the-art power grid voltage integrity verification methods. First, we will review few vector-based methods, then, we will review vectorless techniques. A particular vectorless framework that is of central importance to our work will be reviewed separately. Finally, we will review several publications that improved on this framework.

2.4.1 Vector-Based Power Grid Verification

Broadly defined, vector-based techniques consist of simulating the whole power grid under an exhaustive set of input vectors. A huge research work has been carried out in the recent past in this area, introducing novel pattern dependent methods. Some methods try to find a small set of input patterns that would maximize power supply noise, i.e., maximize voltage drop. Other methods attempt to reduce the grid to a coarser structure, analyze the reduced model, and map the solution back to the original grid. Such techniques are essential for modern industrial designs where grids can consist of over a billion nodes and exhaustive simulation of the whole grid is infeasible. In this section, we will provide an overview of some simulation-based verification methods that employ various innovations such as multigrid techniques, macromodeling techniques, and application of random walk algorithms.

A genetic algorithm (GA) based method was suggested in [17]. Basically, this technique generates iteratively a small set of input patterns that maximize power supply noise. The authors start with an initial set of patterns, which can be either generated randomly or specified by the user, and they assign a fitness value to each input pattern. The GA algorithm, then, iteratively generates successive set of patterns with likely higher fitness based on “evolution like” operations, such as selection, crossover, and mutation. A main issue with this class of search algorithms is that they require a suitable fitness function as the quality of the input patterns generated by GA highly relies on the fitness function used. In this approach, the authors assigns the fitness function to be the peak current the design draws in response to the pattern. The peak current can be efficiently estimated using a waveform simulator that employs event-driven logic simulation algorithm. Notice that higher peak currents means more current flows in the resistive network and, hence, larger voltage drops. If the peak voltage drop is itself the fitness function, then this would require circuit-level simulation which severely slows down the simulation. The algorithm terminates when no further improvement is achieved or the number of iterations exceeds a certain value. The final set of patterns is then fed to a

transistor-level simulator in order to derive a more accurate estimate of power supply noise. The authors in [18] use a similar approach where the fitness function is a combination of the peak current of each block and the peak current of the entire design. Results show that the method in [18] is capable of identifying a larger set of critical nodes than the method in [17]. Both methods, however, give a lower bound on the actual number of critical nodes. Such methods are therefore insufficient for the thorough verification of the power grid reliability.

For a given set of current waveforms, the authors in [19] propose a multigrid-like approach, inspired by the Algebraic Multigrid (AMG) and the Standard Multigrid (SMG) techniques. In short, the proposed method is an iterative grid reduction algorithm that produces a significantly reduced coarser grid. The objective of the grid reduction algorithm is to selectively remove as many nodes as possible while maintaining the ability to estimate voltages at the removed nodes. Generally speaking, grid reduction algorithm, such as those based on SMG techniques, skip every other wire of the grid. However, typical power grids may be irregular, i.e., different edges may have different lengths and different separation distances. Thus, the reduction algorithm should present a systematic mechanism for reducing any *general* grid. After coarsening the grid, the proposed technique solves the reduced system and maps the solution back to the original system. The voltages at the removed nodes on the finer grid are interpolated based on the voltages of the kept nodes that are strongly connected to it. Simulation results show that the proposed technique achieves up to 16X-20X speedups for static analysis and up to 600X for dynamic analysis.

A “divide and conquer” based method was proposed in [20] that exploits the hierarchical structure of the power grid. The grid is partitioned into small local grids that are connected to a global grid via port nodes at the interface. The hierarchical approach for power grid analysis is presented in Fig. 2.4 from [20]. Each local grid is modeled by the following linear system:

$$I = YV + S \tag{2.11}$$

where V and I are vectors of voltages at the ports and currents through the ports, respectively, Y is the port admittance matrix, and S is a current vector that captures the effect of the current sources internal to a local grid at the port nodes. The set (Y, S) in (2.11) is referred to as the macromodel of the respective local grid, and is derived from the modified nodal equations of the partition. Solving the original grid boils down to simulating the reduced global grid, and finally, computing the voltages at the local nodes of each local partition. Simulation results show that this approach achieves a 2X-5X speedup as well as a 10X-20X reduction in memory requirement as opposed to solving

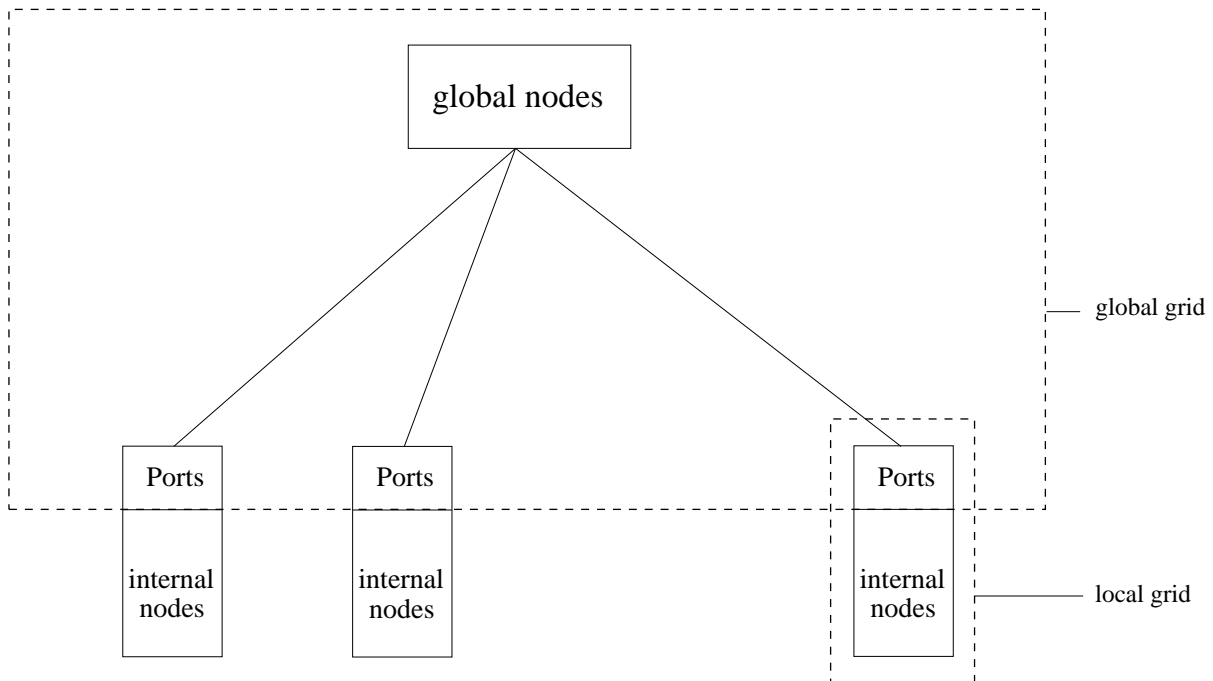


Figure 2.4: Hierarchical power network analysis

the traditional flat approach.

A random walk approach for power grid analysis was proposed in [21]. In brief, the authors pose power grid analysis as a probabilistic problem where the voltage at each node is expressed as the expected value of the voltages on neighboring nodes, weighted by conductance values. Several Monte Carlo simulations are performed in the form of random walks on the grid to estimate statistically the voltage levels on the power grid. A major feature of this approach is that it allows efficient estimation of the voltage drop at selected nodes without the need to solve the entire system.

To sum up, vector-based approaches to power grid verification require a circuit simulator to simulate the grid based on loading currents that are generated from a prior fast simulation of the underlying logic. Although simulation-based methods are widely adopted in industry, nowadays, these methods suffer two major limitations. On one hand, a simulation-based scheme cannot be applied at an early stage of the design flow, when detailed information on the circuit currents is not available. Typically, the grid is verified early in the design process before the underlying logic blocks have been finalized. On the other hand, these methods are optimistic in a sense that they provide a lower bound on the actual worst-case voltage drop. For one thing, it is impossible to generate a representative set of simulation vectors. Hence, finding the actual worst-case voltage drop requires an exhaustive set of current waveforms in order to cover all possible scenarios

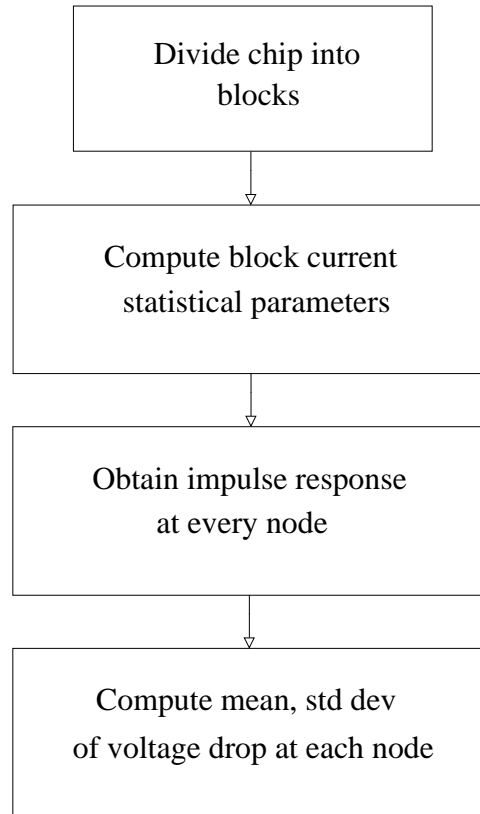


Figure 2.5: Flow diagram of the Stochastic approach in [1]

and guarantee power integrity, which is infeasible.

2.4.2 Vectorless Power Grid Verification

Vectorless verification refers to a class of techniques for verification that do not rely on simulating the power grid for specific input patterns. In this section, we will review a pattern independent verification method. Another method that relies on constraints-based framework is particularly relevant to this work and will thus be reviewed separately in section 2.4.3. After that, we will review several published works that suggest improvements to this framework.

In [1], the authors suggest a method that relies on statistical verification. The main objective of this method is to find the first-order statistics (mean and variance) of the voltage drop on the grid. In short, the authors model the currents drawn by the power grid as stochastic processes, then they propose a method to propagate the statistical parameters, which includes correlations between different blocks both in time and space, through the linear system of the power grid to obtain the distribution of voltage drops at any node in the grid. A flow diagram of the proposed methods is presented in Fig. 2.5. As

a first step, the grid is partitioned into large blocks keeping the correlation between the currents drawn by these blocks minimal. The blocks are then simulated in order to obtain the mean, auto-correlation, and cross-correlation for each block current. After that, the power supply network is modelled as a linear system in order to obtain the transfer function from every circuit block to every node voltage. These impulse responses, along with the statistical parameters of block currents are then used to generate statistical parameters for the voltage drop.

2.4.3 The Constraint-Based Vectorless Framework for Power Grid Verification

Overview

Constraint-based framework, which was first introduced in [2], is a vectorless power grid verification scheme that relies on information that may be available at an early stage of the design, in the form of *current constraints*. The intuition behind this framework is to capture circuit uncertainty via design specs or power budgets known in the early design stages, which are represented as current bounds. Essentially, vectorless verification consists of finding the worst-case voltage fluctuations achievable 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. These methods are often formulated as linear programs (LPs) to find the worst-case voltage drop under current constraints.

The authors in [2] distinguish two types of constraints: *local constraints* and *global constraints*. Local constraints represent bounds on the current drawn by individual current sources, whereas, global constraints express bounds on the sum of currents drawn by groups of current sources simultaneously, at any time t . Depending on the level of abstraction, a current source may represent a single logic gate, a cell, or a large block; the latter is more typical in practice.

Let $i_s(t)$ be an $n \times 1$ vector of current sources and I_L be the $n \times 1$ vector of peak values that the current sources can draw. Notice that local constraints are defined for *every* node on the grid. Thus, a node not connected to a current source, i.e., $i_{s,k}(t) = 0$, $\forall t$, can be represented by a zero-current upper bound, i.e., $I_{L,k} = 0$. Local constraints can be captured by a system of inequalities as:

$$0 \leq i_s(t) \leq I_L \tag{2.12}$$

Notice that, verifying the grid under local constraints alone would allow chip components to simultaneously draw their peak allowable currents, however, this is typically not the case. Although it would be much easier to verify the grid under local constraints alone, this will allow for pessimistic scenarios, i.e., a grid may be identified as failing when, in fact, it is safe. Different groups of current sources could belong to different functional blocks, where each functional block is characterized by a power budget. This notion is captured by means of global constraints. Global constraints are meant to represent circuit specification or functionality at an early design stage by imposing upper bounds on the sum of total currents drawn by groups of current sources. Said differently, global constraints can be thought of as power budgets on groups of logic blocks. If there are l global constraints, then these constraints may be expressed as:

$$0 \leq U i_s(t) \leq I_G, \quad \forall t \geq 0, \quad (2.13)$$

where I_G is an $l \times 1$ vector of upper bounds corresponding to global constraint values, and U is an $l \times n$ incidence matrix consisting of 1 and 0 elements only, where a 1 at the (i, j) th entry of U means that the current source at node j is included in the i th global constraint. Local and global constraints can be compactly captured using the following system of inequalities:

$$0 \leq R i_s(t) \leq I_M, \quad \forall t \geq 0 \quad (2.14)$$

where R is an $(n + l) \times n$ matrix, whose top $n \times n$ block is the n -dimensional identity matrix and the bottom $l \times n$ block is the matrix U , defined in (2.13), and where

$$I_M = \begin{bmatrix} I_L \\ I_G \end{bmatrix} \quad (2.15)$$

A current waveform vector $i_s(t)$ is said to be *feasible* if it satisfies the current constraints $0 \leq R i_s(t) \leq I_M$ at every time point. Let \mathcal{A} represent the set of all current vectors I_s that satisfy the constraints, which is referred to as the *feasible space* of currents:

$$\mathcal{A} = \{I_s : 0 \leq R I_s \leq I_M\} \quad (2.16)$$

For ease of expression, the notation $I_s \in \mathcal{A}$ will be used to signify that the transient current waveform $i_s(t)$ satisfies the current constraints $0 \leq R i_s(t) \leq I_M$ at all points in time.

Verifying the Grid

This framework assumes that the user has specified voltage drop thresholds that must not be exceeded. Finding the worst-case voltage fluctuations can be formulated as a voltage maximization problem, which would then be compared against the threshold. The current constraints defines a search space for currents that will be used to carry out the optimization. It is crucial that these constraints are *linear* so that the region \mathcal{A} is *convex*. If $v_i(t)$ is the *voltage drop* at node i and at time t , then the worst-case voltage drop can be expressed as follows:

$$v_{max,i} \triangleq \max_{I_s \in \mathcal{A}}(v_i(t)) \quad (2.17)$$

Note that, the feasibility region \mathcal{A} is time independent, so that the result of the above maximization is also time independent. Said differently, the voltage drop $v_i(t)$ is maximized over all possible transient current waveforms that satisfy $0 \leq Ri_s(t) \leq I_M, \forall t$. Therefore, the worst-case voltage drop at node i is the same at any time point, so that performing the optimization at only one time point is sufficient.

In this approach, power grid verification is done sequentially, maximizing the voltage drop at each node, one at a time. For ease of expression, we denote by v_{max} the vector of maximum voltage drops at all nodes, so that:

$$v_{max} \triangleq \text{emax}_{I_s \in \mathcal{A}}(v(t)) \triangleq \begin{bmatrix} \max_{I_s \in \mathcal{A}} v_1(t) \\ \max_{I_s \in \mathcal{A}} v_2(t) \\ \vdots \\ \max_{I_s \in \mathcal{A}} v_n(t) \end{bmatrix} \quad (2.18)$$

keeping in mind that the entries of v_{max} are computed sequentially.

The authors in [4] derived the exact worst-case voltage drop using (2.10). Assuming that we are only interested in steady state solution, i.e., when the solution is independent of the initial condition of the grid, and for current constraints in the form of (2.14), so that \mathcal{A} is time independent, the worst-case voltage drop can be expressed as:

$$v_{max} = \lim_{p \rightarrow \infty} \sum_{k=0}^{p-1} \text{emax}_{I_s \in \mathcal{A}} \left[\left(A^{-1} \frac{C}{\Delta t} \right)^k A^{-1} I_s \right] \quad (2.19)$$

Recall that evaluating an $\text{emax}(\cdot)$ operation requires n linear programs (LPs), where the runtime of each LP is proportional to the number of nodes in the grid. Also, notice

that solving (2.19) requires an infinite summation of $\text{emax}(\cdot)$ evaluations. Clearly, it is prohibitively expensive to compute (2.19).

In another approach, the authors in [4] derive an upper-bound to the exact worst-case voltage drop as:

$$v_{ub} \triangleq G^{-1} A \text{emax}_{I_s \in \mathcal{A}}(A^{-1} I_s) \quad (2.20)$$

The author in [3] provides a simpler proof for this upper-bound, which we will present below. Given that the values of the source currents $i_s(\cdot)$ at any two time points are independent variables, we can use (2.10) to write:

$$v_{max} = \text{emax}_{I_s \in \mathcal{A}}(v(t)) = \text{emax}_{I_s \in \mathcal{A}}[(A^{-1} B)v(t - \Delta t)] + \text{emax}_{I_s \in \mathcal{A}}(A^{-1} I_s) \quad (2.21)$$

where the variable vector I_s denotes the current vector at any chosen time point, again due to the independence of the currents across time. Recall that $A^{-1} B \geq 0$, because $A^{-1} \geq 0$ and $B \geq 0$, which leads to:

$$v_{max} \leq A^{-1} B \text{emax}_{I_s \in \mathcal{A}}[v(t - \Delta t)] + \text{emax}_{I_s \in \mathcal{A}}(A^{-1} I_s) \quad (2.22)$$

However, because v_{max} is time independent, so that $\text{emax}_{I_s \in \mathcal{A}}[v(t - \Delta t)] = \text{emax}_{I_s \in \mathcal{A}}[v(t)] = v_{max}$, then we can rewrite (2.22) as:

$$(I_n - A^{-1} B)v_{max} \leq \text{emax}_{I_s \in \mathcal{A}}(A^{-1} I_s) \quad (2.23)$$

where I_n is an $n \times n$ identity matrix. Notice that, because $B = A - G$, then $I_n - A^{-1} B = I_n - A^{-1}(A - G) = A^{-1} G$. Furthermore, we have $I_n + G^{-1} B = I_n + G^{-1}(A - G) = G^{-1} A$, where I_n is the $n \times n$ identity matrix. But $I_n \geq 0$, $G^{-1} > 0$, and $B \geq 0$ with $b_{ii} > 0, \forall i$, so that:

$$I_n + G^{-1} B = G^{-1} A > 0 \quad (2.24)$$

Multiplying (2.23) by $G^{-1} A$, due to (2.24), we get:

$$v_{max} \leq G^{-1} A \text{emax}_{I_s \in \mathcal{A}}(A^{-1} I_s) = v_{ub} \quad (2.25)$$

Notice that computing v_{ub} provides a major simplification of the problem as it requires a single $\text{emax}(\cdot)$ evaluation, A^{-1} computation, and LU factorization of G . Several improvements have been made to mitigate the runtime complexity of constraint-based framework. Because this method requires as many LPs as the number of nodes in a grid, [22] proposed a sparse approximate inverse technique to reduce the runtime of the LPs. Due

to locality and topology properties of a grid, voltages of grid nodes are not completely independent, which was exploited by *dominance* relations among voltage drops in [23]. Recently, in [24], a restriction of the problem to a hierarchically structured set of power constraints was considered to achieve a significant runtime improvement. Another interesting technique was proposed in [25] that verifies each subgrid of the original network independently, imposing boundary conditions on the neighboring nodes of that subgrid, i.e. without explicitly involving nodes that are not directly connected to the subgrid.

Choice of Time Step

Thus far, we have ignored the effect of the time step Δt on the accuracy of the voltage drop values. Generally speaking, for a discrete time system such as that in (2.10), the choice of the time step largely depends on the input waveform $i_s(t)$. For a fast switching $i_s(t)$, Δt should be small enough to trace the transient behavior of the grid voltages. As for the constraints-based framework, such as (2.19) and (2.25), the currents become variables that can take different values from one time-point to the next. In the above derivations, we have implicitly assumed that the current waveforms may change arbitrarily within the time step Δt . In other words, the currents can switch from zero to their maximum values in the feasibility region \mathcal{A} within a single time step. Although it greatly simplifies the analysis, it is unrealistic to assume that the currents at different time instances are independent variables, because the transient behavior of the currents is dictated by the speed of the underlying circuitry. It is worth mentioning, however, that our implicit assumption, when used with small Δt , is conservative as it covers all slower currents scenarios. Note that too small a Δt , close to machine epsilon, might introduce numerical instabilities. Therefore, the choice of the time step Δt must rely on design expertise, by accounting to both: the dynamics of the current loads and the grid transient behavior.

Obtaining Current Constraints

Constraints-based verification methods have been fully developed over the last decade [3], but a key question remains: how would one obtain/specify the current constraints? The constraints are meant to capture the peak power dissipation of circuit blocks. It's easy to see how to get the constraints for a logic block that is available (down to the cell level) and small enough to exhaustively simulate, by using an “offline” characterization process. Otherwise, if the block is unavailable or too large to simulate, one must rely on engineering judgment, and/or expertise from previous design activities, which places an undue burden on users. Thus, providing the current constraints is a burdensome task for

design teams, and our work is aimed at addressing this problem.

Chapter 3

Maximal Safe Containers

3.1 Introduction

The traditional approach of constraints-based framework is to check if the grid is safe under current constraints that are provided by the user. In this chapter, we address the inverse problem: given a grid and the allowed voltage drop thresholds at all grid nodes, we will generate circuit current constraints which, if satisfied by the underlying logic, would guarantee grid safety. The rest of the chapter is organized as follows. In section 3.2, we begin with a couple of definitions and notions that will help us provide a rigorous definition of the inverse problem. We term a set of circuit current constraints that, if adhered to by the underlying logic, would guarantee grid safety, as a *safe container*. In section 3.3, we present a major result of this thesis establishing a necessary and sufficient condition for a container to be safe. This condition, however, suggests that there is an infinity of possible safe containers. Recall that a container defines a set of circuit current constraints that can be used to drive automated floorplanning as well as placement. Therefore, we are interested in a container that would allow more flexibility in the circuit loading currents which, in turn, would allow more flexibility for the subsequent design stages. In section 3.4, we present two ideal, though impractical, approaches to find the largest volume container. Finally, in section 3.5, we introduce the notion of maximal containers and present the bulk of the theoretical contribution of this thesis culminating in the necessary and sufficient conditions given in Theorem 1.

3.2 Problem Definition

We have a power grid as defined in section 2.3.2, discretized in section 2.3.3, and we are modifying the constraints-based framework defined in 2.4.3. We will introduce the notion of a *container* for a current waveform, which will help us construct different alternative current constraints that guarantee grid safety.

Definition 5. (*Container*) Let $t \in \mathbb{R}$, let $i(t) \in \mathbb{R}^m$ be a bounded 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 $i(t)$, represented by the shorthand $i(t) \subset \mathcal{F}$, and we refer to \mathcal{F} as a container of $i(t)$.

Definition 6. (*Safe Grid*) Let $V_{th} \in \mathbb{R}^n, V_{th} \geq 0$, be a given vector of the voltage drop thresholds at all grid nodes. We say that the grid is safe for a given $i(t), \forall t \in \mathbb{R}$, if the corresponding $v(t) \leq V_{th}, \forall t \in \mathbb{R}$.

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 that are of interest, we can express this as $\max_{i(t) \subset \mathcal{F}} (v_k(\tau))$. 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 . At the risk of some abuse of notation, we will now redefine the $\text{emax}(\cdot)$ operator based on the ‘‘container’’ notation, which is used to capture in a single vector all the separate worst-case voltage drop maximizations, as follows.

Definition 7. (emax) For a given container \mathcal{F} , and for an arbitrary $\tau \in \mathbb{R}$, define:

$$v^*(\mathcal{F}) \triangleq \text{emax}_{i(t) \subset \mathcal{F}} [v(\tau)] \triangleq \begin{bmatrix} \max_{i(t) \subset \mathcal{F}} (v_1(\tau)) \\ \max_{i(t) \subset \mathcal{F}} (v_2(\tau)) \\ \vdots \\ \max_{i(t) \subset \mathcal{F}} (v_n(\tau)) \end{bmatrix} \quad (3.1)$$

to be the $n \times 1$ vector whose every entry is the worst-case voltage drop at the corresponding node under all possible current waveforms $i(t)$ contained in \mathcal{F} , with the convention that, if $\mathcal{F} = \phi$, then $\text{emax}_{i(t) \subset \mathcal{F}} [v(\tau)] = 0$.

Definition 8. (*Safe Container*) For a given container \mathcal{F} , we say that \mathcal{F} is safe if $v^*(\mathcal{F}) \leq V_{th}$.

Thus, if \mathcal{F} is safe, then the grid is guaranteed to be safe under all possible current waveforms that are contained in \mathcal{F} . We will see below that a safe container \mathcal{F} can be expressed as a set of constraints on the circuit currents that drive the grid, thereby providing a set of linear constraints that are sufficient to guarantee grid safety. We will find, however, that the choice of \mathcal{F} is not unique. Indeed, there is an infinity of possible safe containers. In the following sections, we will first characterize the most desirable safe containers, and then develop algorithms to generate specific types of containers for practical grids.

Let $M = A^{-1} > 0$ and define the $n \times m$ matrix $M' = MH > 0$.

Definition 9. For any $\mathcal{F} \subset \mathbb{R}^m$, define:

$$\bar{v}(\mathcal{F}) \triangleq G^{-1}A \operatorname{emax}_{I \in \mathcal{F}}(M'I) \quad (3.2)$$

where $I \in \mathbb{R}^m$ is a vector of artificial variables, with units of current, that is used to carry out the $\operatorname{emax}(\cdot)$ operation, and with the convention that $\operatorname{emax}_{I \in \mathcal{F}}(M'I) = 0$, if $\mathcal{F} = \phi$.

In [4] and [3], the authors have derived the following upper-bound on $v^*(\mathcal{F})$ ¹:

$$v^*(\mathcal{F}) \leq \bar{v}(\mathcal{F}) \quad (3.3)$$

In the *forward* version of the grid verification problem [3], where \mathcal{F} is given and one is interested to check if the grid is safe, the upper bound $\bar{v}(\mathcal{F})$ is useful as a *sufficient condition* for grid safety. If $\bar{v}(\mathcal{F}) \leq V_{th}$, then $v^*(\mathcal{F}) \leq V_{th}$ and the grid is safe. This has been found to be a tight upper bound [3]. For our *inverse* version of the problem, V_{th} is given and we are interested to discover a container \mathcal{F} for which $\bar{v}(\mathcal{F}) \leq V_{th}$, so that $v^*(\mathcal{F}) \leq V_{th}$ and the grid is safe. In the next section, this upper bound (3.2) will be used to characterize the class of safe containers, which will lead us to investigate the most desirable containers, which we will call *maximal*.

3.3 Safe Containers

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

$$\mathcal{U} \triangleq \{u \in \mathbb{R}^n : 0 \leq u \leq V_{th}\} \quad (3.4)$$

¹In [4], the upper bound on the worst case voltage drop is in terms of $i(t)$ and M' , as in (3.2). In [3], the proof is presented in terms of $i_s(t) = Hi(t)$ and M , but it can be easily shown that the upper bound in [3] also implies (3.2).

$$\mathcal{F}(u) \triangleq \{I \in \mathbb{R}^m : I \geq 0, M'I \leq MG u\} \quad (3.5)$$

and notice that:

$$MG u \leq MG u' \implies \mathcal{F}(u) \subseteq \mathcal{F}(u'), \quad \forall u, u' \in \mathbb{R}^n \quad (3.6)$$

We will see that it is enough to consider (as we will, in this thesis) only containers of the form (3.5). The restriction to $I \geq 0$ is obvious because $i(t) \geq 0$ is already assumed in our grid model, above, but the rest of (3.5) is due to the following *necessary and sufficient condition*. We will use \mathbb{R}_+^n to denote the set of n -dimensional vectors with non-negative components.

Lemma 4. *For any $u \in \mathbb{R}_+^n$, we have $0 \leq \bar{v}(\mathcal{F}(u)) \leq u$.*

Proof. For any $u \in \mathbb{R}_+^n$, if $\mathcal{F}(u) = \phi$ then, from Definition 9, $0 = \bar{v}(\mathcal{F}(u)) \leq u$. Otherwise, if $\mathcal{F}(u) \neq \phi$, then Definition 9 provides that $\bar{v}(\mathcal{F}(u)) \geq 0$, due to (2.24) and the fact that $I \geq 0$, for all $I \in \mathcal{F}(u)$, by definition. Furthermore, from (3.5), we have $M'I \leq MG u$, $\forall I \in \mathcal{F}(u)$, so that:

$$\operatorname{emax}_{I \in \mathcal{F}(u)}(M'I) \leq MG u \quad (3.7)$$

Multiplying both sides of (3.7) with $G^{-1}A \geq 0$, due to (2.24), we get $\bar{v}(\mathcal{F}(u)) \leq u$, which completes the proof. \square

Lemma 5. *For any $\mathcal{J} \subset \mathbb{R}_+^m$, $\bar{v}(\mathcal{J}) \leq V_{th}$ if and only if $\exists u \in \mathcal{U}$ such that $\mathcal{J} \subseteq \mathcal{F}(u)$.*

Proof. The proof is in two parts.

Proof of the “if direction:” Let $\mathcal{J} \subseteq \mathcal{F}(u)$ for some $u \in \mathcal{U}$, it follows that $\operatorname{emax}_{I \in \mathcal{J}}(M'I) \leq \operatorname{emax}_{I \in \mathcal{F}(u)}(M'I)$, from which $\bar{v}(\mathcal{J}) \leq \bar{v}(\mathcal{F}(u))$, due to $G^{-1}A \geq 0$. Using Lemma 4, we get $\bar{v}(\mathcal{J}) \leq \bar{v}(\mathcal{F}(u)) \leq u$ which, due to $u \in \mathcal{U}$, gives $\bar{v}(\mathcal{J}) \leq V_{th}$.

Proof of the “only if direction:” Let $\mathcal{J} \subset \mathbb{R}_+^m$ with $\bar{v}(\mathcal{J}) \leq V_{th}$, and let $u = \bar{v}(\mathcal{J}) \leq V_{th}$ so that:

$$G^{-1}A \operatorname{emax}_{I \in \mathcal{J}}(M'I) = u \quad (3.8)$$

Then, because $G^{-1}A \geq 0$ due to (2.24) and $I \geq 0$ for any $I \in \mathcal{J}$, then $u \geq 0$, so that $u \in \mathcal{U}$, and multiplying (3.8) with MG , we get:

$$\operatorname{emax}_{I \in \mathcal{J}}(M'I) = MG u \quad (3.9)$$

so that, $\forall I \in \mathcal{J}$, we have $M'I \leq MG u$ which, coupled with $I \geq 0$ gives $\mathcal{J} \subseteq \mathcal{F}(u)$. \square

Therefore, *i)* $\mathcal{F}(u)$ is safe for any $u \in \mathcal{U}$ (due to $\bar{v}(\mathcal{F}(u)) \leq V_{th}$), and *ii)* all possible safe containers \mathcal{J} (based on $\bar{v}(\mathcal{J}) \leq V_{th}$) may be found either as specific $\mathcal{F}(u)$ for some $u \in \mathcal{U}$, or as subsets of such $\mathcal{F}(u)$ containers.

3.4 Maximizing Volume

In this section, we present two ideal, but computationally infeasible, approaches for current constraints generation. Recall that, a container defines a closed subset of \mathbb{R}^m where every member of this set constitutes a possible assignment of the current waveforms. As pointed out earlier, a container can be used to drive subsequent steps of the design process, and hence, the “larger” a container is, the more flexibility is provided for the rest of the design stages. Therefore, it is desirable to provide the user with the “largest” container, or a container that has the largest volume. In the following, we show that these approaches are extremely expensive, but they are worth documenting nonetheless.

3.4.1 Constraints Templates

We digress briefly from the rest of the thesis to assume that the user has specified certain templates for the desired constraints, then we attempt to grow their constraint values as much as possible while still ensuring the safety of the power grid. These templates reflect the dependencies among different functional blocks in the circuit. For example, current sources I_1 , I_3 , and I_8 may be known to originate from blocks that are functionally dependent, and so the user may specify that a global constraint be found that includes only these three currents. Solving this problem is useful for two things:

- i) We will find a set of constraints, based on the user-specified templates, that forms a safe container.
- ii) Because these constraints represent dependencies among functional blocks, finding the constraints values would give us the largest allowable current drawn by a single, or a group, of functional blocks which represent power budgets for these functional blocks. The availability of these budgets at an early stage of the design process is crucially important for the rest of the design flow, as we have discussed earlier.

A drawback of this framework is that it requires the design team to identify functionally dependant blocks. At an early design stage, design teams usually have a tentative

placement of major functional blocks of the chip. Therefore, one would rely on engineering judgment and/or expertise, based on the available information of this chip or previous designs of similar chips, to capture these dependencies.

Definition 10. *With $\alpha \geq 0$ a $p \times 1$ vector of real variables, we define:*

$$\mathcal{G}(\alpha) \triangleq \{I \in \mathbb{R}^m : SI \leq \alpha\} \quad (3.10)$$

where S is a $p \times m$ incidence matrix consisting of 1 and 0 elements only, where a 1 at the (i, j) th entry of S means that the current source at node j is included in the i th global constraint.

We can formulate the problem as follows: given p -templates of global constraints provided by the user, which are represented by S , we are interested in finding the largest volume container $\mathcal{G}(\alpha)$, such that $\mathcal{G}(\alpha) \subseteq \mathcal{F}(u)$, for some $u \in \mathcal{U}$. This can be solved using an optimization problem in the variables α and u as follows:

$$\begin{aligned} & \text{Maximize} && V(\mathcal{G}(\alpha)) \\ & \text{Subject to} && \mathcal{G}(\alpha) \subseteq \mathcal{F}(u) \\ & && u \in \mathcal{U} \end{aligned} \quad (3.11)$$

where $V(\mathcal{G}(\alpha))$ denotes the volume of the polytope defined by $\mathcal{G}(\alpha)$.

There are two issues in the optimization problem that will be discussed hereafter: i) the objective function is the volume of the polytope $\mathcal{G}(\alpha)$, and ii) it has a subset-check, i.e. $\mathcal{G}(\alpha) \subseteq \mathcal{F}(u)$, as an optimization constraint.

Several research works have attempted to compute the volume of a convex polytope. Some works present efficient algorithms to compute the volume using recursive approaches or finite-element methods. Lasserre in [26] presents a recursive analytic expression for the volume of a convex polyhedron in \mathbb{R}^m which is:

$$V(m, S, \alpha) = \left(\frac{1}{m}\right) \sum_{i=1}^p \pi(x, \mathcal{H}_i) \times V(m-1, S, \alpha) \quad (3.12)$$

where $V(m, S, \alpha)$ is the volume of $\mathcal{G}(\alpha)$, $\pi(x, \mathcal{H}_i)$ is the Euclidean distance from any point x , e.g. the origin, to a hyperplane \mathcal{H}_i of $\mathcal{G}(\alpha)$, and $V(m-1, S, \alpha)$ is the volume of the polytope generated by projecting the original polytope on the hyperplane \mathcal{H}_i . Clearly, it is prohibitively expensive to compute this recurrence. Unfortunately, there is no closed form expression for the volume of a polytope.

One way to overcome this issue is to approximate the volume of the polytope by the volume of the ellipsoid inscribed in it. Thus, we are interested in finding the ellipsoid of maximum volume that lies inside $\mathcal{F}(u)$, for some $u \in \mathcal{U}$. Note that, we require the ellipsoid to be inscribed inside $\mathcal{F}(u)$, for some $u \in \mathcal{U}$, in order to guarantee that $\mathcal{G}(\alpha)$ is safe. The authors in [27] address the maximum volume inscribed ellipsoid problem where, for a given α , the maximum volume ellipsoid inscribed in $\mathcal{G}(\alpha)$ can be computed using the following convex optimization problem in the variables P and q :

$$\begin{aligned} & \text{Maximize} && \log \det P \\ & \text{Subject to} && \|P\psi_i\|_2 + \psi_i^T d \leq \alpha_i \quad i \in \{1, \dots, p\} \end{aligned} \tag{3.13}$$

where the ellipsoid is defined as $\varepsilon \triangleq \{Pu + q \mid \|u\|_2 \leq 1\}$, P is an $m \times m$ matrix, q is an $m \times 1$ vector, ψ_i is the i th row of S , α_i is the i th entry of α , and $\log \det P$ is the volume of the ellipsoid ε .

In the simple scenario where u is fixed in the maximization problem (3.11), one can employ an iterative random-search global optimization technique, such as Simulated Annealing (SA) [28, 29]. At each iteration, SA generates a candidate point and computes the *acceptance function* which depends on a variable called the *temperature*. The SA makes the transition to the candidate point if the value of the acceptance function is larger than certain probability. Nonetheless, the algorithm would still require a subset-check, i.e. to check whether $\mathcal{G}(\alpha) \subseteq \mathcal{F}(u)$, in every iteration. A subset-check problem, also referred to as polytope containment problem, is typically solved using as much LPs as the number of constraints in $\mathcal{F}(u)$ [30]. Therefore, this problem is prohibitively expensive and remains only of theoretical interest.

3.4.2 Largest Volume Container

In this section, we investigate a similar but simpler problem to the constraints templates problem. Instead of maximizing the volume of a subset of $\mathcal{F}(u)$, in this section we are interested in maximizing the volume of $\mathcal{F}(u)$ itself. This can be solved using a non-linear optimization problem in the variable u as:

$$\begin{aligned} & \text{Maximize} && V(\mathcal{F}(u)) \\ & \text{Subject to} && u \in \mathcal{U} \end{aligned} \tag{3.14}$$

Notice that, a major simplification of (3.14) over (3.11) is that we eliminated the subset-check from the constraints of the optimization problem. Although (3.14) still requires a closed-form expression of the volume of a polytope, one can use SA based

algorithm and approximate the volume using (3.13) as we have discussed in the previous section. However, this method is also computationally expensive and unscalable.

3.5 Maximal Containers

Recall that, due to Lemma 5, we have: *i)* $\mathcal{F}(u)$ is safe for any $u \in \mathcal{U}$ (due to $\bar{v}(\mathcal{F}(u)) \leq V_{th}$), and *ii)* all possible safe containers \mathcal{J} (based on $\bar{v}(\mathcal{J}) \leq V_{th}$) may be found either as specific $\mathcal{F}(u)$ for some $u \in \mathcal{U}$, or as subsets of such $\mathcal{F}(u)$ containers. 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 $\mathcal{F}(u)$.

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. We capture this with the notion of *maximality*, as follows.

Definition 11. *Let \mathcal{E} be a collection of subsets of \mathbb{R}^m and let $\mathcal{X} \in \mathcal{E}$. We say that \mathcal{X} is maximal in \mathcal{E} if there does not exist another $\mathcal{Y} \in \mathcal{E}$, $\mathcal{Y} \neq \mathcal{X}$, such that $\mathcal{X} \subseteq \mathcal{Y}$.*

Let $\mathcal{E} = \{\mathcal{F}(u) : u \in \mathcal{U}\}$ be a family of subsets of \mathbb{R}^m , induced by \mathcal{U} . Note that $0 \in \mathcal{U}$ for any $V_{th} \geq 0$, and $\mathcal{F}(0) = \{0\}$, due to $M' > 0$ combined with $I \geq 0$, $\forall I \in \mathcal{F}(0)$. It follows that \mathcal{E} always contains a non-empty set as a member.

Maximality is a highly desirable property and we would like to generate containers that are provably maximal. The purpose of the rest of this section is to give necessary and sufficient conditions for $\mathcal{F}(u)$ to be maximal in \mathcal{E} . The maximality of $\mathcal{F}(u)$ relies on crucial properties of u that we will discuss in the following subsections. Note that, because \mathcal{E} always contains a non-empty set as a member, then $\mathcal{F}(u) = \phi$ is never maximal in \mathcal{E} - this will be useful below.

3.5.1 Feasible

Definition 12. *For any $u \in \mathbb{R}^n$, u is said to be feasible if $\mathcal{F}(u)$ is not empty, otherwise it is infeasible.*

Because $\mathcal{F}(0) = \{0\}$, then $u = 0$ is always feasible. In general, we have the following lemma.

Lemma 6. *For any $u \in \mathbb{R}^n$, u is feasible if and only if $MG_u \geq 0$.*

Proof. To prove the “if direction,” let $u \in \mathbb{R}^n$ with $MGu \geq 0$, in which case clearly $0 \in \mathcal{F}(u)$, so that $\mathcal{F}(u)$ is not empty and u is feasible. To prove the “only if direction,” let $u \in \mathbb{R}^n$ be feasible so that $\mathcal{F}(u)$ is not empty, and there exists an $I \in \mathbb{R}^m$ such that $I \geq 0$ and $M'I \leq MGu$. Due to $M' \geq 0$ combined with $I \geq 0$, we have $0 \leq M'I \leq MGu$, so that $MGu \geq 0$. \square

Notice that if $u \in \mathbb{R}^n$ is feasible then multiplying both sides of $MGu \geq 0$ by $G^{-1}A \geq 0$ gives $u \geq 0$, so that, $u \in \mathbb{R}_+^n$. Notice also that, if $V_{th,k} = 0$, then for every $u \in \mathcal{U}$ we have $u_k = 0$. In this case, the only feasible $u \in \mathcal{U}$ is $u = 0$, because $MG = M(A - B) = I_n - MB$, so that $MGu|_k = u_k - MBu|_k = -MBu|_k < 0$.

3.5.2 Extremal

Definition 13. For any $u \in \mathcal{U}$, we say that u is extremal in \mathcal{U} if $\exists k \in \{1, \dots, n\}$ such that $u_k = V_{th,k}$.

Denote by m'_{ij} the (i, j) th element of M' and by c'_j its j th column.

Lemma 7. If $\mathcal{F}(u)$ is maximal in \mathcal{E} then u is feasible and extremal in \mathcal{U} .

Proof. We will prove the contrapositive. Let $u \in \mathcal{U}$ be either infeasible or not extremal in \mathcal{U} ; we will prove that $\mathcal{F}(u)$ is not maximal in \mathcal{E} . If u is infeasible then $\mathcal{F}(u) = \phi$, which we already know is not maximal in \mathcal{E} . Now consider the case when u is feasible but not extremal in \mathcal{U} . In other words, we have $MGu \geq 0$ and $0 \leq u < V_{th}$, so that $\epsilon \triangleq \min_{\forall i} (V_{th,i} - u_i) > 0$. Let $\mathbf{1}$ be the $n \times 1$ vector whose every entry is 1 and let $u' = u + \epsilon \mathbf{1}$. Because G is irreducibly diagonally dominant with positive diagonal and non-positive off-diagonal entries, then $G\mathbf{1} \geq 0$, with $G\mathbf{1} \neq 0$. Let $\gamma \triangleq G(u' - u) = \epsilon G\mathbf{1}$, so that $\gamma \geq 0$ with $\gamma \neq 0$, then $M\gamma > 0$ so that $MGu' > MGu$. Furthermore, considering $u' = u + \epsilon \mathbf{1}$, we have $0 \leq u' \leq V_{th}$ due to the definition of ϵ , so that $u' \in \mathcal{U}$. We have so far established that there exists $u' \in \mathcal{U}$ with $MGu < MGu'$, so that $\mathcal{F}(u) \subsetneq \mathcal{F}(u')$, due to (3.6). It only remains to prove that $\mathcal{F}(u) \neq \mathcal{F}(u')$. For some $i \in \{1, \dots, m\}$, let $j = \operatorname{argmin}_{\forall k} (MGu'|_k / m'_{ki})$, $\delta = (MGu'|_j / m'_{ji}) \geq 0$, $e_i \in \mathbb{R}^m$ be the vector whose i th entry is 1 and all other entries are 0, and $I^{(i)} = \delta e_i \geq 0$. Notice that, for any k , we have:

$$M'I^{(i)}|_k = \delta M'e_i|_k = \delta c'_i|_k = \delta m'_{ki} \quad (3.15)$$

Therefore, $M'I^{(i)}|_j = \delta m'_{ji} = MGu'|_j > MGu|_j$, so that $I^{(i)} \notin \mathcal{F}(u)$. By definition of δ , we have $\delta \leq (MGu'|_k / m'_{ki})$, for every k , meaning $\delta m'_{ki} \leq MGu'|_k$, for every k . Using (3.15), we then have $M'I^{(i)}|_k = \delta m'_{ki} \leq MGu'|_k$, for every k , which leads to $I^{(i)} \in \mathcal{F}(u')$, and the proof is complete. \square

3.5.3 Irreducible

Definition 14. We say that $u \in \mathbb{R}^n$ is reducible if there exists $u' \leq u$, $u' \neq u$, with $\mathcal{F}(u') = \mathcal{F}(u)$; otherwise, u is said to be irreducible.

We will see that irreducibility of u is a crucial property that is required for maximality of $\mathcal{F}(u)$.

Lemma 8. For any feasible $u \in \mathbb{R}^n$ and $z \in \mathbb{R}^n$ such that $0 \leq MGz \leq MG(u - \bar{v}(\mathcal{F}(u)))$, let $u' = u - z$, it follows that $\mathcal{F}(u') = \mathcal{F}(u)$.

Proof. For any $I \in \mathcal{F}(u')$, we have $I \geq 0$ and $M'I \leq MG u' = MG u - MGz \leq MG u$, because $MGz \geq 0$, so that $I \in \mathcal{F}(u)$. It follows that $\mathcal{F}(u') \subseteq \mathcal{F}(u)$. In addition, for any $I \in \mathcal{F}(u)$, we have $I \geq 0$ and:

$$M'I \leq \operatorname{emax}_{I \in \mathcal{F}(u)}(M'I) = MG\bar{v}(\mathcal{F}(u)) \quad (3.16)$$

Notice that for any z with $0 \leq MGz \leq MG(u - \bar{v}(\mathcal{F}(u)))$, we have $MG u' = MG u - MGz \geq MG u - MG(u - \bar{v}(\mathcal{F}(u))) = MG\bar{v}(\mathcal{F}(u))$. Combining this with (3.16), we get $M'I \leq MG u'$, so that $I \in \mathcal{F}(u')$. Therefore, $\mathcal{F}(u) \subseteq \mathcal{F}(u')$ from which $\mathcal{F}(u') = \mathcal{F}(u)$, and the proof is complete. \square

Corollary. For any feasible $u \in \mathbb{R}^n$, let $u' = \bar{v}(\mathcal{F}(u))$, it follows that $\mathcal{F}(u') = \mathcal{F}(u)$.

Proof. Let $z = u - \bar{v}(\mathcal{F}(u))$, so that z satisfies the conditions of Lemma 8, and let $u' = u - z = \bar{v}(\mathcal{F}(u))$. Then, by Lemma 8, $\mathcal{F}(u') = \mathcal{F}(u)$. \square

Lemma 9. For any $u \in \mathbb{R}_+^n$, u is irreducible if and only if it is feasible and $\bar{v}(\mathcal{F}(u)) = u$.

Proof. The proof is in two parts.

Proof of the “if direction:” The proof is by contradiction. Let u be feasible with $\bar{v}(\mathcal{F}(u)) = u$ and suppose that u is reducible so that there exists $u' \leq u$, $u' \neq u$, with $\mathcal{F}(u') = \mathcal{F}(u)$. Notice that $\mathcal{F}(u)$ is not empty, because u is feasible, so that $\mathcal{F}(u')$ is not empty and u' is feasible. Therefore, we get:

$$u' - \bar{v}(\mathcal{F}(u')) = u' - \bar{v}(\mathcal{F}(u)) = u' - u + u - \bar{v}(\mathcal{F}(u))$$

Notice that, because u' is feasible, we have $MG u' \geq 0$ and $u' \geq 0$, due to (2.24). Because $\bar{v}(\mathcal{F}(u')) \leq u'$, due to Lemma 4, it follows that $u' - u + u - \bar{v}(\mathcal{F}(u)) \geq 0$, meaning that $u - \bar{v}(\mathcal{F}(u)) \geq u - u' \geq 0$. But $u - u' \neq 0$, so that $\bar{v}(\mathcal{F}(u)) \neq u$ and we have a contradiction that completes the proof.

Proof of the “only if direction:” We will prove the contrapositive. Let u be either infeasible or $\bar{v}(\mathcal{F}(u)) \neq u$, and we will prove that u is reducible. If u is infeasible then $\mathcal{F}(u) = \phi$ and $u \neq 0$ (recall, $u = 0$ is always feasible), and it is easy to find another infeasible u' with $u' \leq u$ and $u' \neq u$, as follows. Let $u' = \frac{1}{2}u$, from which $MGu' = \frac{1}{2}MGu \not\geq 0$, because u is infeasible, so that u' is infeasible. Therefore, we have found $u' \leq u$, $u' \neq u$, with $\mathcal{F}(u') = \mathcal{F}(u) = \phi$ which means u is reducible. If u is feasible and $\bar{v}(\mathcal{F}(u)) \neq u$, let $u' = \bar{v}(\mathcal{F}(u)) \in \mathbb{R}_+^n$, due to Lemma 4, which also provides that $\bar{v}(\mathcal{F}(u)) \leq u$, so that $u' \leq u$, $u' \neq u$, with $\mathcal{F}(u') = \mathcal{F}(u)$ due to the Corollary to Lemma 8, and u is reducible. \square

Note, if u is irreducible and extremal in \mathcal{U} , then $u_k = V_{th,k}$ for some k , and $\bar{v}(\mathcal{F}(u))|_k = V_{th,k}$.

Lemma 10. *For any feasible $u \in \mathbb{R}_+^n$, let $u' = \bar{v}(\mathcal{F}(u))$, it follows that u' is irreducible.*

Proof. For any $u \in \mathbb{R}_+^n$, let $u' = \bar{v}(\mathcal{F}(u))$. Notice that $MGu' = MG\bar{v}(\mathcal{F}(u)) = \text{emax}_{I \in \mathcal{F}(u)}(M'I) \geq 0$ due to $M' \geq 0$ and $I \geq 0$ for any $I \in \mathcal{F}(u)$, so that u' is feasible, due to Lemma 6. Because $u' = \bar{v}(\mathcal{F}(u))$, it follows from the Corollary to Lemma 8 that $\mathcal{F}(u') = \mathcal{F}(u)$, from which $\bar{v}(\mathcal{F}(u')) = \bar{v}(\mathcal{F}(u))$. With this, notice that $u' - \bar{v}(\mathcal{F}(u')) = u' - \bar{v}(\mathcal{F}(u)) = 0$, from which $\bar{v}(\mathcal{F}(u')) = u'$. Using Lemma 9, it follows that u' is irreducible, and the proof is complete. \square

Lemma 11. *For any $u \in \mathbb{R}_+^n$, u is irreducible if and only if:*

$$MGu \leq MGu' \iff \mathcal{F}(u) \subseteq \mathcal{F}(u'), \quad \forall u' \in \mathbb{R}^n \quad (3.17)$$

Proof. The proof is in two parts.

Proof of the “if direction:” We give a proof by contradiction. Given (3.17) and suppose u is reducible, so that it is either infeasible or $\bar{v}(\mathcal{F}(u)) \neq u$. If u is infeasible, then $\mathcal{F}(u) = \phi \subseteq \mathcal{F}(u')$, for any $u' \in \mathbb{R}^n$, so that $MGu \leq MGu'$, for any $u' \in \mathbb{R}^n$, due to (3.17). But this is impossible, because we can always find a $u' \in \mathbb{R}^n$ that violates $MGu \leq MGu'$, as follows. Let $\mathbf{1}$ be the $n \times 1$ vector whose every entry is 1 and let $w = -G^{-1}A\mathbf{1}$ so that $MGw = -\mathbf{1} < 0$, and let $u' = u + w$ so that $MGu' - MGu = MGw < 0$. Therefore, it must be that u is feasible and $\bar{v}(\mathcal{F}(u)) \neq u$. Let $u' = \bar{v}(\mathcal{F}(u))$, so that $\mathcal{F}(u') = \mathcal{F}(u)$ due to the Corollary to Lemma 8, with $MGu' = MG\bar{v}(\mathcal{F}(u))$. Recall that $MG\bar{v}(\mathcal{F}(u)) = \text{emax}_{I \in \mathcal{F}(u)}(M'I) \leq MGu$, and $MG\bar{v}(\mathcal{F}(u)) \neq MGu$ due to $\bar{v}(\mathcal{F}(u)) \neq u$, so that $MGu' \leq MGu$, $MGu' \neq MGu$. This means that we have $\mathcal{F}(u) \subseteq \mathcal{F}(u')$ while $MGu \not\leq MGu'$, which contradicts (3.17), and the proof is complete.

Proof of the “only if direction:” Let u be irreducible, so that u is feasible with $\bar{v}(\mathcal{F}(u)) = u$. Due to (3.6), it only remains to prove that $\forall u' \in \mathbb{R}^n, \mathcal{F}(u) \subseteq \mathcal{F}(u') \implies MG u \leq MG u'$. Notice that $\mathcal{F}(u')$ is non-empty, because $\mathcal{F}(u) \neq \phi$ and $\mathcal{F}(u) \subseteq \mathcal{F}(u')$, from which u' is feasible. Because u and u' are feasible, and using $u = \bar{v}(\mathcal{F}(u))$, notice that:

$$\begin{aligned} MG u' - MG u &= MG u' - MG \bar{v}(\mathcal{F}(u)) \\ &= MG u' - \operatorname{emax}_{I \in \mathcal{F}(u)}(M'I) \\ &\geq MG u' - \operatorname{emax}_{I \in \mathcal{F}(u')} (M'I) \geq 0 \end{aligned}$$

where we used $\operatorname{emax}_{I \in \mathcal{F}(u')} (M'I) \geq \operatorname{emax}_{I \in \mathcal{F}(u)} (M'I)$, because $\mathcal{F}(u) \subseteq \mathcal{F}(u')$. Therefore, $MG u' - MG u \geq 0$, so $MG u \leq MG u'$ and the proof is complete. \square

The following lemma provides a necessary algebraic condition for u to be irreducible, which becomes a necessary and sufficient condition in the case $m = n$, i.e. when every grid node is connected to a current source. Denote by r'_i the i th row of M' and by m_{ij} the (i, j) th element of M .

Lemma 12. *For any $u \in \mathbb{R}^n$, define $w \triangleq MG u$, we have the following:*

1. *if u is irreducible then*

$$\frac{w_i}{m_{ii}} \leq \frac{w_j}{m_{ji}}, \quad \forall i, j \in \{1, \dots, n\} \quad (3.18)$$

2. *in case $m = n$, if (3.18) holds then u is irreducible.*

Proof. The proof is in two parts.

Proof of 1: Let $u \in \mathbb{R}_+^n$ be irreducible, so that $\bar{v}(\mathcal{F}(u)) = G^{-1} A \operatorname{emax}_{I \in \mathcal{F}(u)}(M'I) = u$. Equivalently, multiplying both sides by MG , we get $\operatorname{emax}_{I \in \mathcal{F}(u)}(M'I)|_i = w_i, \forall i \in \{1, \dots, n\}$. Then, for every i , there exists a $y^{(i)} \geq 0$, $M'y^{(i)} \leq w$, and $M'y^{(i)}|_i = w_i$, from which:

$$r'_j y^{(i)} \leq w_j, \quad \forall j \in \{1, \dots, n\} \quad (3.19)$$

$$r'_i y^{(i)} = w_i \quad (3.20)$$

which we can write, after expanding the dot product, as:

$$w_j \geq \sum_{k=1}^m m_{jk} y_k^{(i)}, \quad \forall j \in \{1, \dots, n\} \quad (3.21)$$

$$w_i = \sum_{k=1}^m m_{ik} y_k^{(i)} \quad (3.22)$$

For every j , multiply (3.21) by m_{ii} and (3.22) by m_{ji} , then subtract the second equation from the first, to get:

$$m_{ii}w_j - m_{ji}w_i \geq \sum_{k=1}^m (m_{ii}m_{jk} - m_{ji}m_{ik}) y_k^{(i)} \quad (3.23)$$

M being the inverse of an \mathcal{M} -matrix, the *path product condition* holds [31], so $m_{ii}m_{jk} \geq m_{ji}m_{ik}$, $\forall i, j, k$, and the right-hand side of (3.23) is non-negative. In turn, this means that the left-hand side is non-negative, which leads directly to (3.18) and completes the proof.

Proof of 2: For any $i \in \{1, 2, \dots, n\}$, let $e_i \in \mathbb{R}^n$ be the vector whose i^{th} entry is 1 and all other entries are 0, let $x^{(i)} = \frac{w_i}{m_{ii}} e_i \geq 0$. Then:

$$Mx^{(i)} = \frac{w_i}{m_{ii}} Me_i = \left[\frac{w_i}{m_{ii}} m_{1i} \quad \frac{w_i}{m_{ii}} m_{2i} \quad \cdots \quad \frac{w_i}{m_{ii}} m_{ni} \right]^T \leq w \quad (3.24)$$

where the last inequality is due to (3.18). Hence, $x^{(i)} \in \mathcal{F}(u)$. With $Mx^{(i)}|_i = w_i m_{ii} / m_{ii} = w_i$ due to (3.24), it follows that $\text{emax}_{I \in \mathcal{F}(u)}(MI) = w = MG_u$, so that $\bar{v}(\mathcal{F}(u)) = G^{-1}A \text{emax}_{I \in \mathcal{F}(u)}(MI) = u$ and u is irreducible. \square

3.5.4 Maximality

As pointed out earlier, we are interested in safe containers that are maximal in \mathcal{E} . We now present our main result that gives the necessary and sufficient conditions for maximality.

Theorem 1. $\mathcal{F}(u)$ is maximal in \mathcal{E} if and only if u is irreducible and extremal in \mathcal{U} .

Proof. The proof is in two parts.

Proof of the “if direction:” We give a proof by contradiction. Let $u \in \mathcal{U}$ be irreducible and extremal in \mathcal{U} , but suppose that $\mathcal{F}(u)$ is not maximal in \mathcal{E} , so that $\exists u' \in \mathcal{U}$ such that $\mathcal{F}(u) \subseteq \mathcal{F}(u')$, with $\mathcal{F}(u) \neq \mathcal{F}(u')$. Because $\mathcal{F}(u) \neq \mathcal{F}(u')$, then clearly $MG_u \neq MG_{u'}$, and using Lemma 11, we have $MG_u \leq MG_{u'}$. Let $\delta = MG_{u'} - MG_u$, so that $\delta \geq 0$ and

$\delta \neq 0$. Because $G^{-1}A > 0$ from (2.24), then $G^{-1}A\delta = u' - u > 0$. Then $u < u' \leq V_{th}$, due to $u' \in \mathcal{U}$, so that u is not extremal in \mathcal{U} , and we have a contradiction that completes the proof.

Proof of the “only if direction:” We give a proof by contradiction. Given that $\mathcal{F}(u)$ is maximal in \mathcal{E} , we know from Lemma 7 that u is feasible and extremal in \mathcal{U} . Suppose u is reducible, so that $\bar{v}(\mathcal{F}(u)) \neq u$, because we already have that u is feasible. Recall that $0 \leq \bar{v}(\mathcal{F}(u)) \leq u$. Let $u' \triangleq \bar{v}(\mathcal{F}(u)) \neq u$, so that $u' \in \mathcal{U}$. Let $\delta = MG_u - MG_{u'} = MG_u - \text{emax}_{I \in \mathcal{F}(u)}(M'I)$, then we have $\delta \geq 0$ and $\delta \neq 0$ (due to $u' \neq u$). Because $G^{-1}A > 0$, then $G^{-1}A\delta = u - u' > 0$. Consequently, we have $u' < u \leq V_{th}$, due to $u \in \mathcal{U}$, so that u' is not extremal in \mathcal{U} . Therefore, by Lemma 7, $\mathcal{F}(u')$ is not maximal in \mathcal{E} . However, $\mathcal{F}(u) = \mathcal{F}(u')$, due to the Corollary to Lemma 8, so that $\mathcal{F}(u)$ is not maximal in \mathcal{E} , a contradiction that completes the proof. \square

This important theoretical result forms the basis for our choice of practical constraints generation algorithms that are guaranteed to give maximal containers, as we will see in the next section. Recall that whenever u is irreducible and extremal in \mathcal{U} , then $\bar{v}(\mathcal{F}(u))|_k = V_{th,k}$, for some k , so that the k th grid node will experience its maximum allowable voltage drop. In other words, a maximal container always causes some node(s) on the grid to experience the maximum allowable voltage drop.

3.6 Conclusion

Early power grid verification is a key step in modern chip design. Traditionally, it has been performed either by simulation or by vectorless verification, both of which have serious shortcomings. We propose a novel method to solve the inverse problem of vectorless verification, by generating circuit current constraints that ensure power grid safety. We develop some key theoretical results to allow the generation of constraints that correspond to maximal current spaces.

Chapter 4

Constraints Generation Algorithms

4.1 Introduction

So far, we have shown that a container $\mathcal{F}(u)$ is safe and maximal in \mathcal{E} if and only if u satisfies the conditions of Lemma 5 and Theorem 1. The evident question becomes, among all safe containers that satisfy the maximality condition, which one is most convenient for the desired application-specific chip? Early in the design flow, design teams typically have some high level requirements for the intended design. A mobile processor, for example, might require a maximum power dissipation of 3 Watts whereas a desktop processor might require up to 200 Watts. One would like to check if the candidate grid can support the corresponding level of peak power in the underlying circuit. Additionally, one would like to verify the ability of the grid to spread the power dissipation across the die in a uniform fashion, which is intended for temperature-aware ICs. In this chapter, we will discuss some design objectives that will lead us to algorithms for finding specific maximal safe containers $\mathcal{F}(u)$.

4.2 Peak Power Dissipation

An interesting quality metric for a power grid is the peak total power dissipation that it can safely support in the underlying circuit. We refer here to the instantaneous power dissipation, which is conservatively approximated by $V_{dd} \sum_{j=1}^m i_j(t)$. Thus, we are interested in a safe container that is maximal in \mathcal{E} and that allows the highest possible $\sum_{v_j} I_j$. For any $u \in \mathcal{U}$, we define $\sigma(u)$ to be the largest sum of current source values allowed

under $\mathcal{F}(u)$:

$$\sigma(u) \triangleq \max_{I \in \mathcal{F}(u)} \left(\sum_{j=1}^m I_j \right) \quad (4.1)$$

and we define σ^* to be the largest $\sigma(u)$ achievable under all possible $u \in \mathcal{U}$, so that:

$$\sigma^* \triangleq \max_{u \in \mathcal{U}} (\sigma(u)) \quad (4.2)$$

Let $u_p \in \mathcal{U}$ be such that $\sigma(u_p) = \sigma^*$, and $I^* \in \mathcal{F}(u_p)$ be such that $\sum_{j=1}^m I_j^* = \sigma^*$. In general, u_p and I^* may not be unique. Based on (3.4) and (3.5), we can express the combined (4.1) and (4.2) as the following linear program (LP):

$$\begin{aligned} \sigma^* = \quad & \text{Max} && \sum_{j=1}^m I_j \\ & \text{subject to} && M'I \leq MG u \\ & && 0 \leq u \leq V_{th} \\ & && I \geq 0 \end{aligned} \quad (4.3)$$

Let \mathcal{D} be the feasible region of the LP (4.3):

$$\mathcal{D} \triangleq \{(I, u) : I \geq 0, M'I \leq MG u, 0 \leq u \leq V_{th}\} \quad (4.4)$$

so that, from the above, we have:

$$\sigma^* = \max_{(I, u) \in \mathcal{D}} \left(\sum_{j=1}^m I_j \right) \quad (4.5)$$

Notice that, $(0, 0) \in \mathcal{D}$ so that \mathcal{D} is not empty, and all of σ^* , u_p , and I^* are well-defined. Also, for every $(I, u) \in \mathcal{D}$, we have $M'I \leq MG u$ and $I \geq 0$ which, because $M' \geq 0$, gives $0 \leq M'I \leq MG u$ so that u is feasible, due to Lemma 6. Therefore, u_p is feasible and the container $\mathcal{F}(u_p) = \{I \in \mathbb{R}^m : I \geq 0, M'I \leq MG u_p\}$ provides the desired current constraints:

$$\begin{aligned} i(t) &\geq 0, \quad \forall t \in \mathbb{R} \\ M'i(t) &\leq MG u_p, \quad \forall t \in \mathbb{R} \end{aligned}$$

The following lemma establishes the maximality of $\mathcal{F}(u_p)$, based on Theorem 1. Denote by c_j the j th column of M , and notice that $c'_j = c_j$, for every $j \in \{1, 2, \dots, m\}$.

Lemma 13. $\mathcal{F}(u_p)$ is maximal in \mathcal{E} .

Proof. We give a proof by contradiction; the proof is in two parts. First, we will prove that if $\mathcal{F}(u_p)$ is not maximal in \mathcal{E} , then there exists a $u \in \mathcal{U}$ that is not extremal in \mathcal{U} , with $\sigma(u) = \sigma^*$. Second, we will prove that, for any $u \in \mathcal{U}$ such that $\sigma(u) = \sigma^*$, we have that u is extremal in \mathcal{U} , which will provide a contradiction that completes the proof.

Suppose that $\mathcal{F}(u_p)$ is not maximal in \mathcal{E} , so that either u_p is not extremal in \mathcal{U} or u_p is reducible, due to Theorem 1. If u_p is not extremal in \mathcal{U} , then we have found a $u = u_p \in \mathcal{U}$ that is not extremal in \mathcal{U} , with $\sigma(u) = \sigma^*$. If u_p is reducible, then by Lemma 9 and because u_p is feasible, we must have $\bar{v}(\mathcal{F}(u_p)) \neq u_p$. Let $u' = \bar{v}(\mathcal{F}(u_p))$, so that $\mathcal{F}(u') = \mathcal{F}(u_p)$, due to the Corollary to Lemma 8, and, by (4.1), we have $\sigma(u') = \sigma(u_p) = \sigma^*$. Let $\delta = MG u_p - MG u'$. Recall that $MG u_p \geq MG \bar{v}(\mathcal{F}(u_p))$ and $MG u_p \neq MG \bar{v}(\mathcal{F}(u_p))$, due to $\bar{v}(\mathcal{F}(u_p)) \neq u_p$, from which $\delta \geq 0$ and $\delta \neq 0$. Combining this with $G^{-1}A > 0$, from (2.24), we have $G^{-1}A\delta = u_p - u'$. Therefore, we have $0 \leq u' < u_p \leq V_{th}$, the final step due to $u_p \in \mathcal{U}$, so that $u' \in \mathcal{U}$, u' is not extremal in \mathcal{U} and $\sigma(u') = \sigma^*$. This completes the first part of the proof.

Next, we will prove that, for any $u \in \mathcal{U}$ with $\sigma(u) = \sigma^*$, we have that u is extremal in \mathcal{U} . For any such u , there must exist a vector $I \in \mathcal{F}(u)$ such that $\sum_{j=1}^m I_j = \sigma^*$. We will proceed by contradiction. Suppose that u is not extremal in \mathcal{U} , so that $u < V_{th}$. Let $\epsilon \triangleq \min_{\forall k} (V_{th,k} - u_k) > 0$, let $\mathbf{1}$ be the $n \times 1$ vector whose every entry is 1, and let $0 \leq \gamma = u + \epsilon \mathbf{1} \leq V_{th}$ due to the definition of ϵ , from which $\gamma \in \mathcal{U}$. Notice that:

$$MG\gamma = MG u + \epsilon MG \mathbf{1}$$

Because G is irreducibly diagonally dominant with positive diagonal and non-positive off-diagonal entries, then $G\mathbf{1} \geq 0$, with $G\mathbf{1} \neq 0$, so that $\epsilon MG \mathbf{1} > 0$, because $\epsilon M > 0$, and $MG\gamma > MG u$.

Now, let $\lambda = \min_{\forall i} (MG\gamma|_i - MG u|_i) / \max_{\forall i,j} (m_{ij})$. Because $MG\gamma > MG u$ and $M > 0$, it follows that $\lambda > 0$. Also, let $e_1 \in \mathbb{R}^n$ be the vector whose 1st entry is 1 and all other entries are 0 and let $I' = I + \lambda e_1$. Because $\lambda > 0$, we have $\lambda e_1 \geq 0$, $\lambda e_1 \neq 0$, $I' \geq I \geq 0$, and $I' \neq I$, so that $\sum_{j=1}^m I'_j > \sum_{j=1}^m I_j = \sigma^*$. Furthermore, we have $I' \in \mathcal{F}(\gamma)$, because:

$$M'I' = M'I + \lambda M'e_1 = M'I + \lambda c'_1 \tag{4.6}$$

$$= M'I + \frac{\min_{\forall i} (MG\gamma|_i - MG u|_i)}{\max_{\forall i,j} (m_{ij})} c_1 \tag{4.7}$$

$$\leq MG u + \min_{\forall i} (MG\gamma|_i - MG u|_i) \mathbf{1} \tag{4.8}$$

$$\leq MG\gamma \tag{4.9}$$

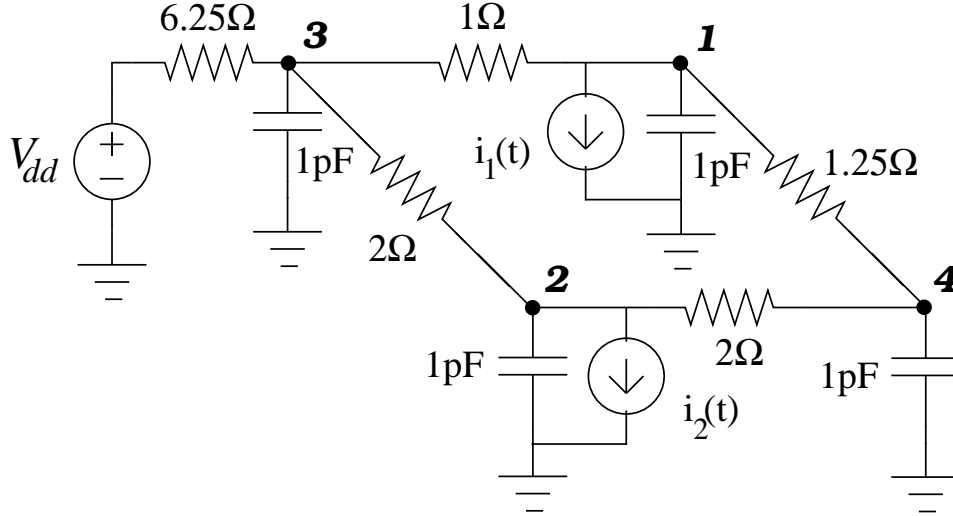


Figure 4.1: An example of a power grid with 4 nodes, 2 current sources, and $V_{th} = [110 \ 100 \ 95 \ 105]^T$ (units of mV).

where in (4.8) we used $I \in \mathcal{F}(u)$ and $c_1 / \max_{v_i, j} (m_{ij}) \leq \mathbf{1}$, and the last inequality is due to $MGu < MG\gamma$. Therefore, we have found $\gamma \in \mathcal{U}$ and $I' \in \mathcal{F}(\gamma)$ with $\sigma(\gamma) = \sum_{j=1}^m I'_j > \sigma^*$, which contradicts (4.5). It follows that u is extremal in \mathcal{U} . \square

As an example, the LP (4.3) is run on the small grid in Fig. 4.1 and the resulting container is shown in Fig. 4.2 where $u_p = [89 \ 100 \ 95 \ 98]^T$ (units of mV). Notice that this method, in order to allow the maximum peak power, may generate a container that is skewed in a way that imposes a tight constraint on current in certain locations of the die (such as at $i_2(t)$) while allowing larger current in other locations (such as at $i_1(t)$). Other approaches are possible to avoid this skew and even out the current allowances, as we will see next.

4.3 Uniform Current Distribution

The design team may be interested in a grid that safely supports a uniform current distribution across the die, so as to allow a placement that provides a uniform temperature distribution. We can generate constraints that allow that objective by searching for a safe maximal container $\mathcal{F}(u)$ that contains the hypersphere in current space that has the largest volume, or the largest radius θ . In other words, this method aims to “raise the minimum” and avoid the skew indicated above. We will develop a method (4.16) which, when applied to the simple grid in Fig. 4.1, generates the container shown in Fig. 4.2, where $u_s = [83 \ 91 \ 95 \ 92]^T$ (units of mV).

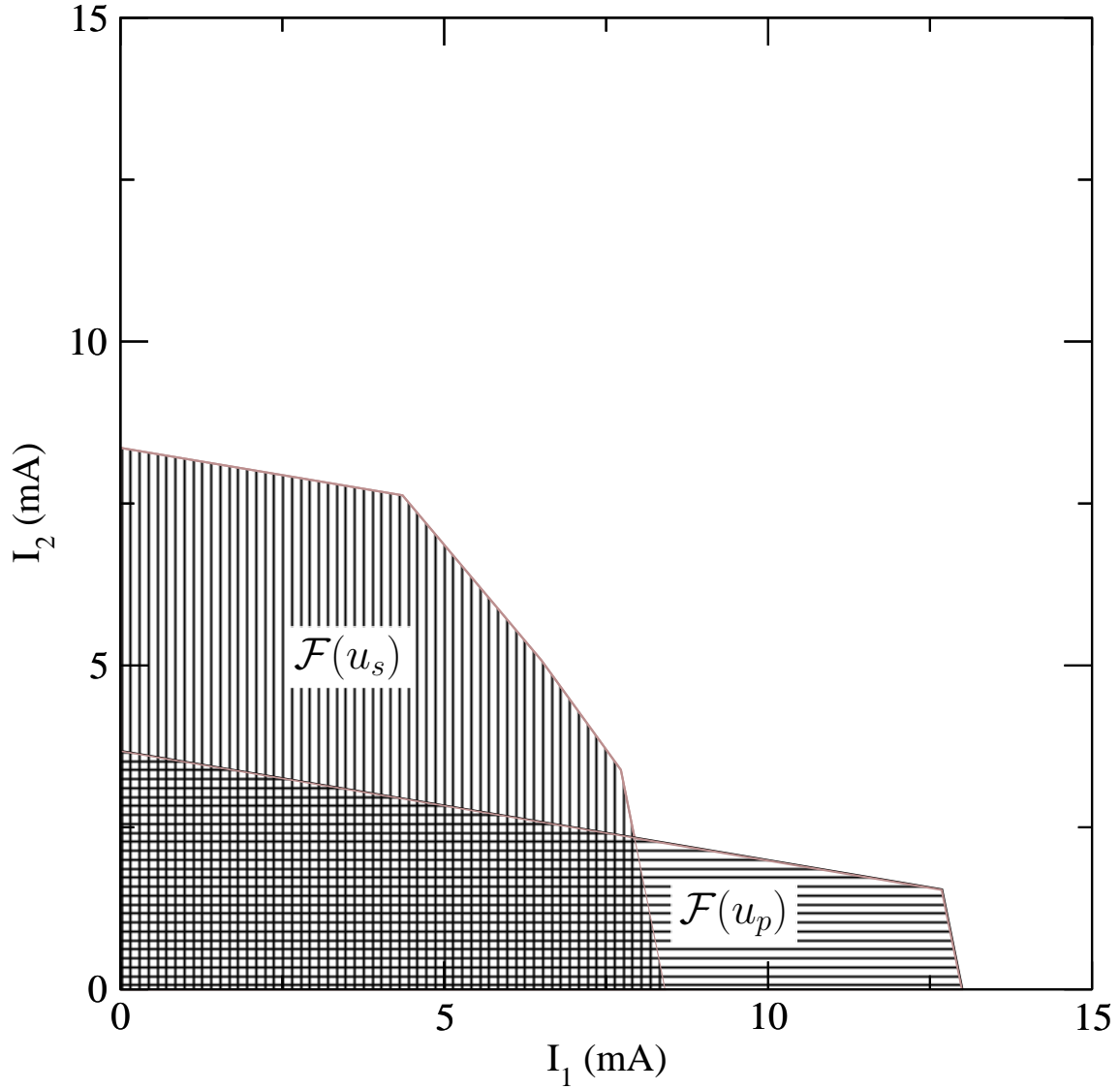


Figure 4.2: An example of $\mathcal{F}(u_p)$ and $\mathcal{F}(u_s)$.

Let $S(\theta) \subset \mathbb{R}^m$ denote the hypersphere with radius θ , centered at the origin and let $S^+(\theta) = S(\theta) \cap \mathbb{R}_+^m$ be the part of that hypersphere that is in the first quadrant of \mathbb{R}^m . Denote by r_i the i th row of M . For any $u \in \mathcal{U}$, define $\mathcal{H}_i = \{I \in \mathbb{R}^m : I \geq 0, r_i' I = r_i' Gu\}$ to be the hyperplane that constitutes the i th outer boundary of $\mathcal{F}(u)$, as in the 2-dimensional example in Fig. 4.3. Define D_i to be the distance from the origin to \mathcal{H}_i which, according to [32], can be expressed as:

$$D_i = \frac{|r_i' Gu|}{d_i} \quad (4.10)$$

where $d_i = \sqrt{\sum_{j=1}^m m_{ij}^2} > 0$. As we're interested in a non-empty $\mathcal{F}(u)$, we will enforce that $\theta \geq 0$ and u is feasible, i.e., $r_i G u \geq 0, \forall i$, so that:

$$D_i = \frac{r_i G u}{d_i} \quad (4.11)$$

In order to have $S^+(\theta) \subseteq \mathcal{F}(u)$, we will require that:

$$\theta \leq D_i, \quad \forall i \in \{1, \dots, n\} \quad (4.12)$$

which can be expressed compactly as:

$$\theta d \leq M G u \quad (4.13)$$

where $d = [d_1 \ \dots \ d_n]^T$. For any $u \in \mathcal{U}$, we define $\rho(u)$ to be the largest $\theta \geq 0$ for which (4.13) is satisfied, so that:

$$\rho(u) \triangleq \max_{0 \leq \theta d \leq M G u} (\theta) \quad (4.14)$$

and we define ρ^* to be the largest $\rho(u)$ achievable under all possible $u \in \mathcal{U}$, so that:

$$\rho^* \triangleq \max_{u \in \mathcal{U}} (\rho(u)) \quad (4.15)$$

Let $u_s \in \mathcal{U}$ be such that $\rho(u_s) = \rho^*$. In general, u_s may not be unique. We can express the combined (4.14) and (4.15) as the following linear program:

$$\begin{aligned} \rho^* = \text{Max} \quad & \theta \\ \text{subject to} \quad & \theta d \leq M G u \\ & 0 \leq u \leq V_{th} \\ & \theta \geq 0 \end{aligned} \quad (4.16)$$

Let \mathcal{S} be the feasible region of the LP (4.16):

$$\mathcal{S} \triangleq \{(\theta, u) : \theta \geq 0, \theta d \leq M G u, 0 \leq u \leq V_{th}\} \quad (4.17)$$

so that, from the above, we have:

$$\rho^* = \max_{(\theta, u) \in \mathcal{S}} (\theta) \quad (4.18)$$

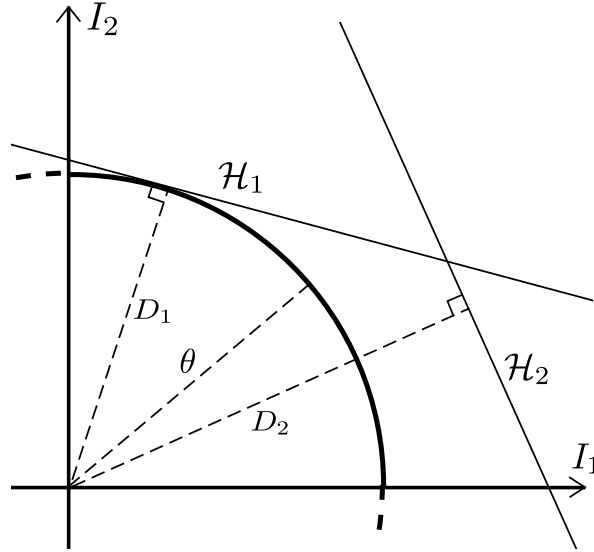


Figure 4.3: An illustration of perpendicular distances to hyperplanes.

Notice that, $(0,0) \in \mathcal{S}$ so that \mathcal{S} is not empty, and ρ^* and u_s are well-defined. Also, for every $(\theta, u) \in \mathcal{S}$, we have $\theta d \leq MG u$ and $\theta \geq 0$. Because $d \geq 0$, it follows that $0 \leq \theta d \leq MG u$ so that u is feasible, due to Lemma 6. Therefore, u_s is feasible and the container $\mathcal{F}(u_s) = \{I \in \mathbb{R}^m : I \geq 0, M'I \leq MG u_s\}$ provides the desired current constraints:

$$\begin{aligned} i(t) &\geq 0, \quad \forall t \in \mathbb{R} \\ M'i(t) &\leq MG u_s, \quad \forall t \in \mathbb{R} \end{aligned}$$

The following lemma, based on Theorem 1, establishes the maximality of $\mathcal{F}(u_s)$.

Lemma 14. $\mathcal{F}(u_s)$ is maximal in \mathcal{E} .

Proof. We give a proof by contradiction; the proof is in two parts. First, we will prove that if $\mathcal{F}(u_s)$ is not maximal in \mathcal{E} , then there exists a $u \in \mathcal{U}$ that is not extremal in \mathcal{U} , with $\rho(u) = \rho^*$. Second, we will prove that, for any $u \in \mathcal{U}$ such that $\rho(u) = \rho^*$, we have that u is extremal in \mathcal{U} , which will provide a contradiction that completes the proof.

Suppose that $\mathcal{F}(u_s)$ is not maximal in \mathcal{E} , so that either u_s is not extremal in \mathcal{U} or u_s is reducible, due to Theorem 1. If u_s is not extremal in \mathcal{U} , then we have found a $u = u_s \in \mathcal{U}$ that is not extremal in \mathcal{U} , with $\rho(u) = \rho^*$. If u_s is reducible, then by Lemma 9 and because u_s is feasible, we must have $\bar{v}(\mathcal{F}(u_s)) \neq u_s$. Let $u' = \bar{v}(\mathcal{F}(u_s))$, so that $\mathcal{F}(u') = \mathcal{F}(u_s)$ due to the Corollary to Lemma 8. Clearly, $S^+(\rho^*) \subseteq \mathcal{F}(u')$, because $S^+(\rho^*) \subseteq \mathcal{F}(u_s)$, so that $\rho(u') = \rho^*$. Let $\delta = MG u_s - MG u'$. Recall that $MG u_s \geq MG \bar{v}(\mathcal{F}(u_s))$ and $MG u_s \neq MG \bar{v}(\mathcal{F}(u_s))$, due to $\bar{v}(\mathcal{F}(u_s)) \neq u_s$, from which $\delta \geq 0$ and $\delta \neq 0$. Combining

this with $G^{-1}A > 0$, from (2.24), we have $G^{-1}A\delta = u_s - \bar{v}(\mathcal{F}(u_s)) > 0$. Therefore, we have $0 \leq u' < u_s \leq V_{th}$, the final step due to $u_s \in \mathcal{U}$, so that $u' \in \mathcal{U}$, u' is not extremal in \mathcal{U} and $\rho(u') = \rho^*$. This completes the first part of the proof.

Next, we will prove that, for any $u \in \mathcal{U}$ with $\rho(u) = \rho^*$, we have that u is extremal in \mathcal{U} . We will proceed by contradiction. Suppose that u is not extremal in \mathcal{U} , so that $u < V_{th}$. Let $\epsilon \triangleq \min_{\forall k} (V_{th,k} - u_k) > 0$, let $\mathbf{1}$ be the $n \times 1$ vector whose every entry is 1, and let $0 \leq \gamma = u + \epsilon \mathbf{1} \leq V_{th}$, due to the definition of ϵ , from which $\gamma \in \mathcal{U}$. Notice that:

$$MG\gamma = MG u + \epsilon MG \mathbf{1} \quad (4.19)$$

Because G is irreducibly diagonally dominant with positive diagonal and non-positive off-diagonal entries, then $G\mathbf{1} \geq 0$, with $G\mathbf{1} \neq 0$, so that $\epsilon MG\mathbf{1} > 0$, because $\epsilon M > 0$, and $MG\gamma > MG u$.

Now, let $\lambda = \min_{\forall i} (MG\gamma|_i - MG u|_i) / \max_{\forall i} (d_i)$ and let $\theta' = \rho^* + \lambda$. Because $MG\gamma > MG u$ and $d_i > 0, \forall i$, it follows that $\lambda > 0$ and $\theta' > \rho^* \geq 0$. Furthermore, we have $(\theta', \gamma) \in \mathcal{S}$, because:

$$\theta' d = \rho^* d + \frac{\min_{\forall i} (MG\gamma|_i - MG u|_i)}{\max_{\forall i} (d_i)} d \quad (4.20)$$

$$\leq MG u + \min_{\forall i} (MG\gamma|_i - MG u|_i) \mathbf{1} \quad (4.21)$$

$$\leq MG\gamma \quad (4.22)$$

where in (4.21) we used $(\rho^*, u) \in \mathcal{S}$ and $d / \max_{\forall i} (d_i) \leq 1$, and the last inequality is due to $MG u < MG\gamma$. Therefore, we have found $(\theta', \gamma) \in \mathcal{S}$ and $\theta' > \rho^*$, which contradicts (4.18). It follows that u is extremal in \mathcal{U} . \square

4.4 Results

The above two algorithms (4.3) and (4.16) have been implemented using **C++**. Both problems require the computation of the inverse of A , which was generated using the sparse approximate inverse technique (SPAI), as was done in [22]. The maximizations were performed using the Mosek optimization package [33]. We conducted tests on a set of power grids with a 1.1 V supply voltage that were generated based on user-specifications, including grid dimensions, metal layers, pitch and width per layer, and C4 and current source distributions, consistent with 65nm technology. All results were obtained using a 2.6 GHz Linux machine with 24 GB of RAM.

Table 4.1: Power Grid Specifications

Power Grid				
Name	Nodes	Current Sources	Voltage Sources	Interconnects
G1	8,413	552	364	12,512
G2	18,678	1,119	779	27,806
G3	32,554	2,070	1,272	48,515
G4	50,444	3,192	2,040	75,505
G5	113,304	7,140	4,400	169,160
G6	200,828	12,656	7,906	300,394
G7	312,232	19,460	12,191	466,773

The number of variables in (4.3) is $n+m$, and the number of variables in (4.16) is $n+1$, where n is the total number of nodes and m is the number of current sources attached to the grid. Denote by $P(u) \triangleq V_{dd} \times \sigma(u)$ the peak power dissipation allowed under $\mathcal{F}(u)$. In Table 4.3, we present the results of both LPs in columns 5 and 7, respectively. For instance, on a 312,232 node grid, the peak power dissipation is 59.77 mW and the largest current radius for which the part of the hypersphere in the first quadrant is contained in $\mathcal{F}(u_s)$ is 2.03 μA . The CPU times for solving (4.3) and (4.16) are given in columns 6 and 8, respectively. Note that these CPU times do not include the time for computing the approximate inverse, which is reported separately in column 4. For example, the total CPU time for solving (4.3) corresponds to the sum of the CPU times reported in columns 4 and 6. Fig. 4.4 shows a plot of the CPU time of both LPs and SPAI versus the number of nodes in a grid. Complexity analysis shows that both LPs have around $\mathcal{O}(n^{1.7})$ empirical complexity while SPAI has around $\mathcal{O}(n^{1.9})$ empirical complexity. Moreover, the only source of error is the sparse approximation (SPAI) of A^{-1} , which is controlled by enforcing an error tolerance of 10^{-4} between every entry of the exact inverse and the corresponding entry of the approximate inverse.

To study the difference between the containers generated using (4.3) and (4.16), we used two methods. First, we computed the peak power dissipation achievable under $\mathcal{F}(u_s)$, which is $P(u_s)$, and the largest current radius for which the part of the hypersphere in the first quadrant is contained in $\mathcal{F}(u_p)$, which is $\rho(u_p)$. The results obtained for $P(u_s)$ and $\rho(u_p)$ are reported in columns 9 and 10 of Table 4.3. The results show that $P(u_s) \ll P(u_p)$ and $\rho(u_p) \ll \rho(u_s)$ on all grids. In fact, the peak power dissipation achievable under $\mathcal{F}(u_s)$ is at most 59% of that achievable under $\mathcal{F}(u_p)$. Also, the largest current radius for which the part of the hypersphere in the first quadrant is contained

Table 4.2: Results of both LPs

Power Grid	SPAI	Peak Power		Uniform Current Distribution	
Name	CPU Time	$P(u_p)$ in mW	CPU Time	$\rho(u_s)$ in μA	CPU Time
G1	1.3 min	1.73	1.15 min	1.77	2.2 min
G2	4.6 min	3.71	5.51 min	2.11	12.86 min
G3	13.38 min	6.75	23.51 min	1.47	36.7 min
G4	27.7 min	9.83	29.28 min	1.8	57.2 min
G5	2.52 h	22.06	1.92 h	1.44	3.38 h
G6	6.8 h	40.69	6.37 h	2.11	11.9 h
G7	17.73 h	59.77	11.19 h	2.03	18.68 h

Table 4.3: Results of each LP using the other container

Power Grid	Peak Power using u_s	Uniform Current Distribution using u_p
Name	$P(u_s)$ in mW	$\rho(u_p)$ in μA
G1	0.6	0.84
G2	1.86	1.06
G3	3.02	0.462
G4	5.09	0.8
G5	11.07	0.45
G6	22.7	0.98
G7	35.23	0.98

in $\mathcal{F}(u_p)$ is at most 50% of that contained in $\mathcal{F}(u_s)$. Thus, each approach provides a unique trade-off for the chip design team. Furthermore, we present in Fig. 4.5 a plot of the peak power dissipation allowable under $\mathcal{F}(u_p)$ and $\mathcal{F}(u_s)$ for each grid with respect to the number of nodes in a grid. The power dissipation varies linearly with the number of nodes, so that one can estimate the peak power dissipation ($P(u_p)$) for a 1 billion node grid to be 191.5 Watts. This conforms with the maximum power dissipation of a modern IC as depicted in Fig. 1.1.

Another way to compare the two approaches (4.3) and (4.16), is to look at the *power density*, i.e., the power dissipation per unit area of the die, allowed by the two resulting containers. To assess this, we maximize the allowed power (current) within a small window of the die surface, and we do this for every position of that window across the die.

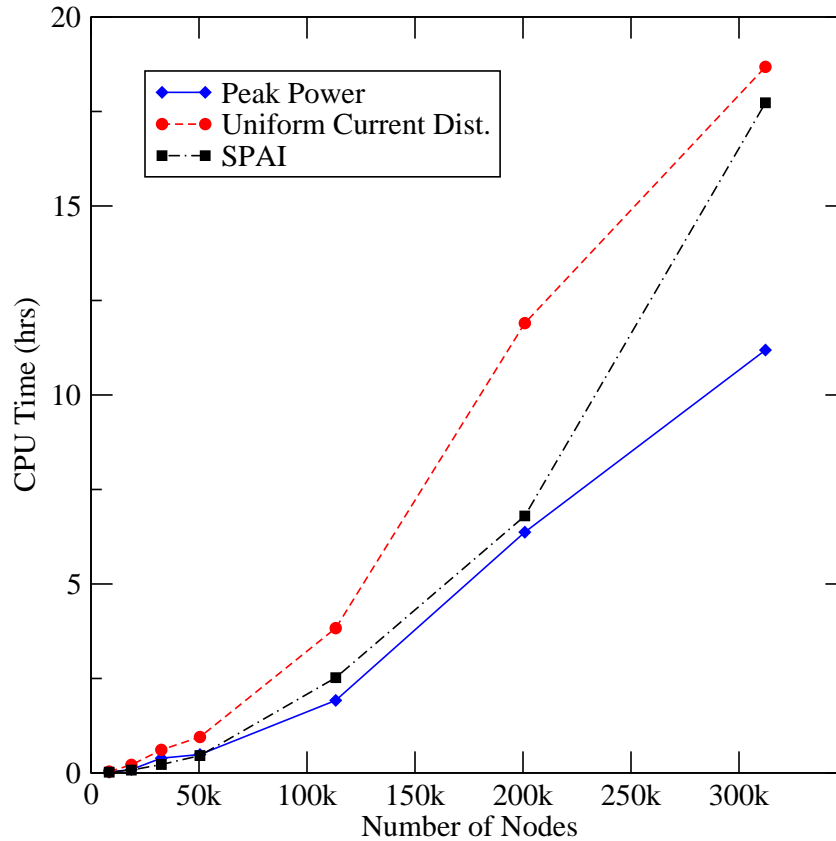


Figure 4.4: CPU time of both LPs and SPAI versus the number of grid nodes.

We divide the die area into $\kappa \times \kappa$ of these windows and compute the peak power density inside each, using a variation of (4.3). In Figs. 4.6a and 4.6b, we present contour plots for $\kappa = 35$ for the peak power densities under $\mathcal{F}(u_p)$ and $\mathcal{F}(u_s)$, respectively, on a 50k node grid. Note that the current constraints based on $\mathcal{F}(u_p)$ allow higher current densities at certain spots but also include some spots with very small and restricted current density budgets. This large spread in power densities can lead to thermal hotspots. This may be avoided by using $\mathcal{F}(u_s)$ which, as expected and as seen in the figure, provides a uniform distribution of power densities across the die area compared to $\mathcal{F}(u_p)$, which is reflected in a smaller standard deviation. Of course, $\mathcal{F}(u_p)$ supports a larger overall peak power dissipation than $\mathcal{F}(u_s)$, which is reflected in a larger mean. There is a clear trade-off between the two methods, making them both useful but also pointing the way to future

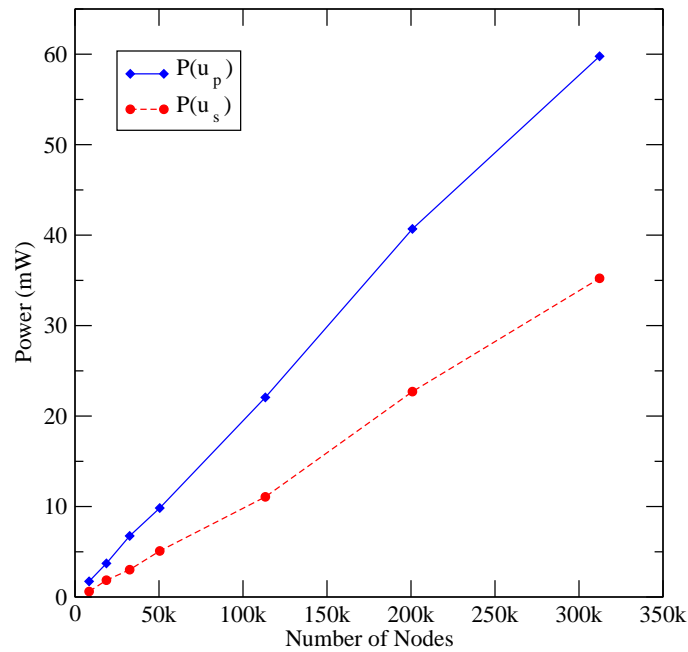


Figure 4.5: Peak Power Dissipation problem

work for managing this trade-off and investigating other possibilities.

4.5 Conclusion

In this chapter, we discuss some design objectives and develop two constraints generation algorithms that target key quality metrics of the grid: the maximum power dissipation the grid can safely support and the uniformity of the power spread across the die. These key quality metrics provide a high level and early assessment of the candidate grid design. We show that the generated constraints satisfy the safety and maximality conditions that were presented in chapter 3. We then study the difference between the generated containers and show that each approach provides a unique trade-off for the chip design team.

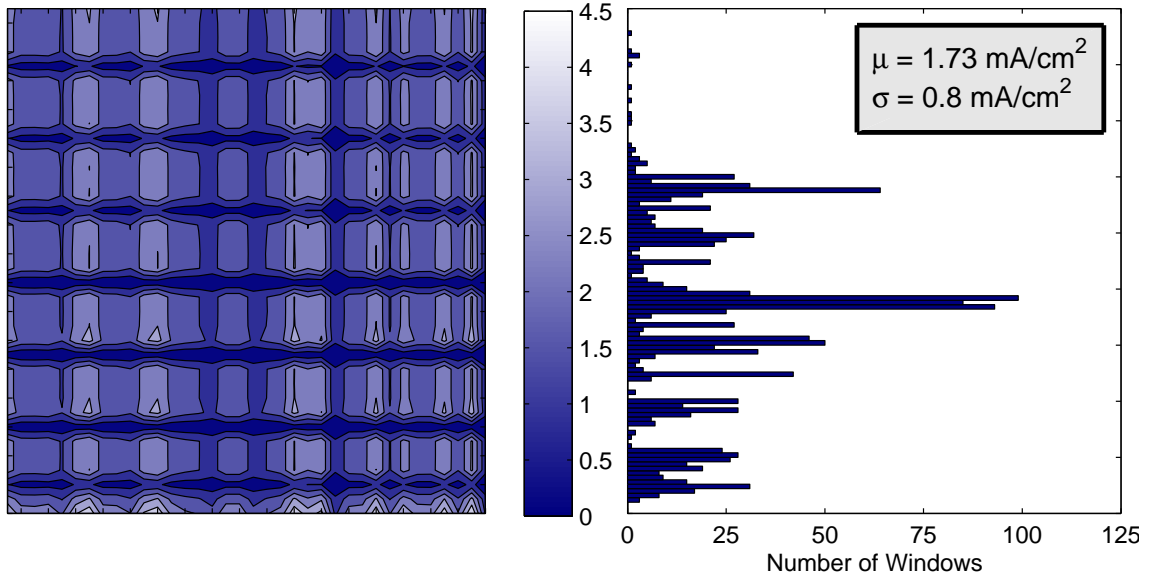
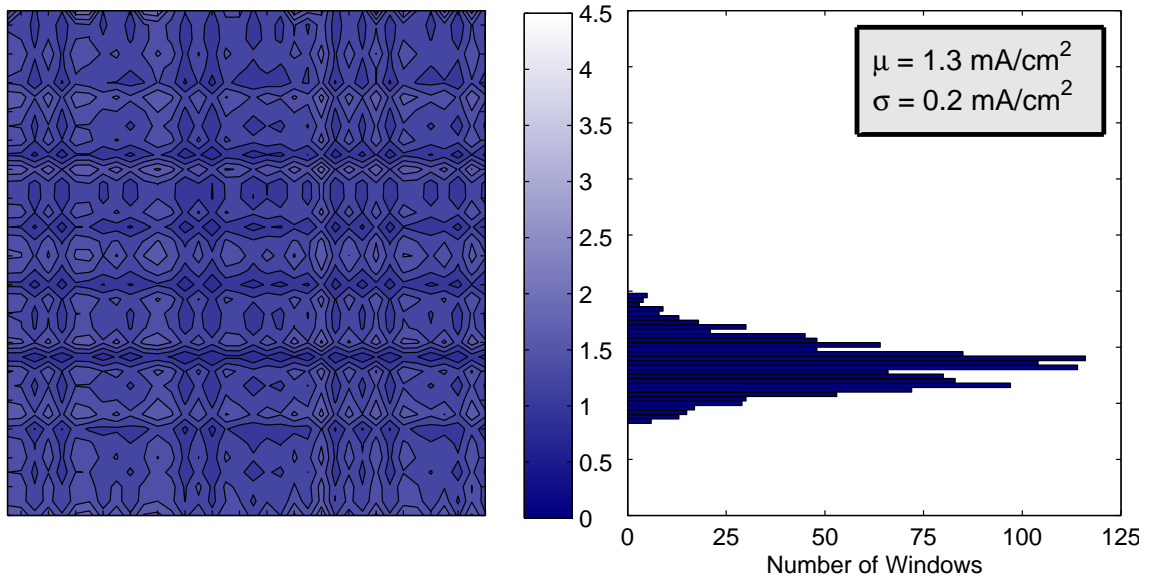
(a) Using $\mathcal{F}(u_p)$ (b) Using $\mathcal{F}(u_s)$

Figure 4.6: Contour plots for peak power density across the layout and the corresponding histograms. The color bar units are mA/cm².

Chapter 5

Conclusion and Future Work

With the continued scaling of semiconductor technology, there has been an emerging need for robust design of a chip's power distribution network. This distribution network should supply a well-regulated source of supply voltage at the intended design speed. Having excessive fluctuations in the voltage levels supplied to the underlying logic would put both circuit performance and reliability at risk.

The on-die part of the power distribution network, or simply the power grid, should be designed and examined prior to the completion of the circuit design itself. The reason behind this is that one should verify the grid when modifications to the design can be easily incorporated. There has been much research in the recent past on designing power grid verification techniques, however, the most commercially adopted tools assume a fully designed circuit and can only be used after placement and routing. As a result, design teams rely on previous design activities, without a way to factor in circuit uncertainty to reduce the extent of over-design.

To overcome these issues, there has been much focus on power grid verification techniques that can be performed early in the design process. One of the emerging tools is the constraints-based framework that relies on information that may be available at an early stage of the design, in the form of current constraints. This framework laid the foundation for several other methods employing various innovations. Essentially, grid verification is casted as computing the worst-case voltage fluctuation achievable at all nodes of the grid under all possible transient current waveforms that satisfy user-specified current constraints. A major contribution of this approach was in providing a systematic verification framework that can find the worst-case voltage drop on the power grid early in the design flow. However, this verification tool requires the user to provide current constraints. In practice, design teams rely on their expertise to specify current constraints which remains a burdensome task.

Instead of the traditional approach of requiring design teams to specify current constraints that can be used to verify the safety of the grid, the main focus of this thesis was to provide a systematic method to generate circuit current constraints, given the voltage drop thresholds at all grid nodes. If satisfied, these constraints would ensure power grid safety. Moreover, these constraints encapsulate much useful information about the grid and their availability at an early stage of the design process provides a way to drive the rest of the design process.

In chapter 3, we developed a rigorous problem definition and some key theoretical results related to maximality of the current space defined by the constraints. The subject of chapter 4 was to develop algorithms to generate sets of current constraints that are maximal targeting specific design objectives. For example, the maximum peak power dissipation a grid can safely support and the uniformity of the power spread across the die. Results showed that current constraints generated by both algorithms provide unique trade-offs for the chip design team.

Future work will mainly cover three headlines. First, the runtime complexity of the provided algorithms might become a hurdle to generate current constraints for larger power grids. Note that some parts of the algorithms can be parallalized. Therefore, it would be necessary to investigate various methods to optimize the runtime complexity. Second, we aim at developing an automated floorplanning as well as grid-aware placement. These algorithms can design possible floorplans and cell placements that factor in the safety of the grid. Finally, it might be viable to extend our work to an RLC grid model.

Bibliography

- [1] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, and R. Panda, “A stochastic approach to power grid analysis,” in *ACM/IEEE Design Automation Conference*, San Diego, CA, June 7-11 2004, pp. 171–176.
- [2] D. Kouroussis and F. N. Najm, “A static pattern-independent technique for power grid voltage integrity verification,” in *ACM/IEEE 40th Design Automation Conference (DAC-03)*, Anaheim, CA, June 2-6 2003, pp. 99–104.
- [3] F. N. Najm, “Overview of vectorless/early power grid verification,” *ACM/IEEE International Conference on Computer-Aided Design*, pp. 670–677, 2012.
- [4] I. A. Ferzli, F. N. Najm, and L. Kruse, “A geometric approach for early power grid verification using current constraints,” in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, November 5-8 2007, pp. 40–47.
- [5] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Science*. Society for Industrial and Applied Mathematics, 1994.
- [6] R. S. Varga, *Matrix Iterative Analysis*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1962.
- [7] E. Ajith and F. Najm, *Failure mechanisms in semiconductor devices*. J. Wiley, 1997.
- [8] K. Killpack, S. Natarajan, and A. Krishnamachary, “Case study on speed failure causes in a microprocessor,” in *IEEE Design and Test of Computers*, 2008, pp. 224–230.
- [9] R. Ahmadi and F. N. Najm, “Timing analysis in presence of power supply and ground voltage variations,” in *IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2003, pp. 176–183.

- [10] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, and R. Panda, "Vectorless analysis of supply noise induced delay variation," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, November 9-13 2003, pp. 184–191.
- [11] G. Bai, S. Bobba, and I. N. Hajj, "Static timing analysis including power supply noise effect on propagation delay in VLSI circuits," in *ACM/IEEE 38th Design Automation Conference (DAC-01)*, Las Vegas, NV, June 18-22 2001, pp. 295–300.
- [12] R. Panda, D. Blaauw, R. Chaudhry, V. Zolotov, B. Young, and R. Ramaraju, "Model and analysis for combined package and on-chip power grid simulation," in *ACM/IEEE International Symposium on Low Power Electronics and Design*, Italy, July 26-27 2000, pp. 179–184.
- [13] S. Sapatnekar and H. Su, "Analysis and optimization of power grids," *IEEE Design & Test of Computers*, pp. 7–15, May-June 2003.
- [14] W.-H. Lee, S. Pant, and D. Blaauw, "Analysis and reduction of on-chip inductance effects in power supply grids," in *IEEE International Symposium on Quality Electronic Design (ISQED)*, San Jose, CA, March 22-24 2004, pp. 131–136.
- [15] A. Muramatsu and M. Hashimoto, "Effects of on-chip inductance on power distribution grid," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 88, no. 12, pp. 3564–3572, 2005.
- [16] F. N. Najm, *Circuit Simulation*. Hoboken, NJ: John Wiley & Sons, Inc., 2010.
- [17] Y. Jiang, K. Cheng, and A. Deng, "Estimation of maximum power supply noise for deep sub-micron designs," in *Proceedings of the 1998 international symposium on Low power electronics and design*, Monterey, CA, 1998, pp. 233–238.
- [18] Y. M. Jiang, T. K. Young, and K. T. Cheng, "Vip - an input pattern generator for identifying critical voltage drop for deep sub-micron designs," in *ACM/IEEE International Symposium on Low Power Electronics and Design*, San Diego, CA, August 16-17 1999, pp. 156–161.
- [19] J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "Multigrid-like technique for power grid analysis," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, November 4-8 2001, pp. 480–487.

- [20] M. Zhao, R. Panda, S. Sapatnekar, and D. Blaauw, “Hierarchical analysis of power distribution networks,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 2, pp. 159–168, 2002.
- [21] H. Quan, R. Nassif, and S. S. Sapatnekar, “Random walks in a supply network,” in *ACM/IEEE Design Automation Conference*, Anaheim, CA, June 2-6 2003, pp. 93–98.
- [22] N. H. Abdul Ghani and F. N. Najm, “Fast vectorless power grid verification using an approximate inverse technique,” in *ACM/IEEE 46th Design Automation Conference (DAC-09)*, San Francisco, CA, July 26-31 2009, pp. 184–189.
- [23] —, “Power grid verification using node and branch dominance,” in *ACM/IEEE 47th Design Automation Conference*, San Diego, CA, June 5-9 2011, pp. 682–687.
- [24] Y. Wang, X. Hu, C.-K. Cheng, G.-K.-H. Pang, and N. Wong, “A realistic early-stage power grid verification algorithm based on hierarchical constraints,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 1, pp. 109–120, Jan. 2012.
- [25] X. Xiong and J. Wang, “Constraint abstraction for vectorless power grid verification,” *Proc. Design Automation Conference*, pp. 1–6, 2013.
- [26] J. Lasserre, “An analytical expression and an algorithm for the volume of a convex polyhedron in \mathbb{R}^n ,” *Journal of Optimization Theory and Applications*, vol. 39, no. 3, pp. 363–377, 1983.
- [27] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [28] S. Kirkpatrick, C. Gelatt, and M. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, pp. 671–680, 1983.
- [29] V. Cerny, “Thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm,” *Journal of Optimization Theory and Applications*, vol. 45, pp. 41–51, 1985.
- [30] V. Kaibel and M. Pfetsch, “Some algorithmic problems in polytope theory,” in *Algebra, Geometry and Software Systems*, pp. 23–47.
- [31] C. R. Johnson and R. L. Smith, “Path product matrices,” *Linear Multilinear Algebra*, vol. 46, no. 3, pp. 328–337, 1999.

- [32] K. Borsuk, *Multidimensional Analytic Geometry*. Polish Scientific Publishers, 1969.
- [33] The MOSEK optimization software. [Online]. Available: <http://www.mosek.com>