UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of a doctoral thesis by

Haihua Su

and have found that it is complete and satisfactory in all aspects,
and that any and all revisions required by the final
examining committee have been made.

Professor Sachin S. Sapatnekar
____
Name of Faculty Advisor

____
Signature of Faculty Advisor

____
Date

GRADUATE SCHOOL

Design and Optimization of Global Interconnect in High Speed VLSI Circuits

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

HAIHUA SU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Sachin S. Sapatnekar, Advisor

JANUARY 2002

Abstract

This thesis consists of three parts, corresponding to various aspects of global interconnect design issues. The first part deals with the design and optimization of power distribution networks, with special consideration to power grid topology design, wire sizing and decoupling capacitor placement. The second part presents a hybrid mesh/tree clock network construction technique with wire sizing and buffer insertion for performance optimization. Finally, the last part is devoted to a global interconnect congestion reduction technique that simultaneously considers the performance requirements for both signal lines and power grids.

For the design and optimization of power distribution networks, we propose a structured skeleton that is intermediate to the conventional method that uses full meshes which are hard to analyze efficiently, and tree-structured networks, which provide poor performance. As an example, we consider a power/ground (P/G) network structure modeled as an overlying mesh with underlying trees originating from the mesh, which eases the task of analysis with acceptable performance sacrifices. A fast and efficient event-driven P/G network simulator is proposed, which hierarchically simulates the P/G network with an adaptation of PRIMA to handle non-zero initial conditions. This network is then optimized by using wire widening and placing decoupling capacitors (decaps). An adjoint network that incorporates the variable topology of the original P/G network, as elements switch in and out of the network, is constructed to calculate the transient adjoint sensitivity over multiple intervals. The gradients of the most critical node with respect to each wire width and decap are used by a sensitivity-based heuristic optimizer that minimizes a weighted sum of the wire and the decap area.

For ASIC-like circuits, the addition of decoupling capacitances is arguably the most powerful degree of freedom that a designer has for power-grid noise abatement and is becoming more important as technology scales. We propose and demonstrate an algorithm for the automated placement and sizing of decaps and show that this allows power grid noise to be significantly reduced with little change in the total chip area.

Next, the idea of a hybrid mesh/tree structure is applied to the design of clock distribution networks. We hierarchically construct a hybrid mesh/tree clock network structure consisting of overlying zero-skew clock meshes, with underlying zero-skew clock trees originating from the mesh nodes. We propose a mesh construction procedure, which guarantees zero skew under the Elmore delay model, using a simple and efficient linear programming formulation. Buffers are inserted to reduce the transition time (or rise time). As a post-processing step, wire width optimization under an accurate higher-order delay metric is performed to further minimize the transition time and propagation delay/skew. The hybrid mesh/tree construction scheme provides a smaller propagation delay and transition time than a comparable clock tree. The tolerance to process variations is also shown to be much better than a clock tree. Results of comparing various hybrid structures have also been studied.

Finally, a global wire design methodology that simultaneously considers the performance needs for both signal lines and power grids under congestion considerations is presented. An iterative procedure is employed in which the global routing is performed according to a congestion map that includes the resource utilization of the power grid, followed by a step in which the power grid is adjusted to relax the congestion in crowded regions. This adjustment is in the form of wire removal in noncritical regions, followed by a wire sizing step that overcomes the effects of wire removal. The overall routability is demonstrated to be significantly improved while the power grid noise is maintained within a specified constraint.

Acknowledgment

First of all, I would like to thank my advisor, Professor Sachin Sapatnekar, for the invaluable guidance and patient help he has given to me throughout my Ph.D. study and research. I sincerely appreciate his persistent encouragement all these years, especially when I met difficulties. His broad knowledge and keen perception into various cutting-edge research directions in VLSI CAD have led me right into the door of this area, which is beneficial to my research and this dissertation, and of course, will be profitable to my future career. What also impressed me is his enthusiastic pursuit towards the truth, which has been and will be stimulating me throughout my life.

I would also like to thank my committee members, Professors Eugene Shragowitz, Professor Gerald Sobelman and Professor Kia Bazargan for their helpful advice.

I am grateful to Dr. Sani Nassif, the mentor of my internship at the IBM Austin Research Lab, who has not only provided the precious help and guidance to this research, but has also brought me into the most fascinating world of industry CAD tool implementation as well. I would like to acknowledge Dr. Jiang Hu at IBM Microelectronics and Kaushik Gala at Motorola, both previous graduate students from our group, for their collaborative work with me. I also thank Dr. Charles Alpert and Dr. Frank Liu at IBM Austin Research Lab, Dr. Chandu Visweswariah at IBM T. J. Watson Research Center and Professor Duncan M. H. Walker of Texas A&M University for their helpful discussions.

I owe many thanks to fellow graduate students for helping me at different times and making my life in Minneapolis pleasant and fruitful: Min Zhao, Jiang Hu, Kaushik Gala, Jatuchai Pangjun, Mahesh Ketkar, Suresh Raman, Shrirang Karandikar, Raza ul Mustafa, Haitian Hu, Rupesh Shelar, Cheng Wan, Venkatesan Rajappan, Tianpei Zhang, Yong Zhan and Brent Goplen.

I am grateful to National Science Foundation, Semiconductor Research Cooperation for funding parts of my research, and IEEE, ACM and the University of Minnesota for providing financial support to attend conferences, and to the IBM Austin Research Lab for providing me with the opportunity to work as a co-op.

Finally I would like to thank my husband Xiaolue Lai for his encouragement and support throughout these years, which makes my life much more meaningful and enjoyable. And lastly, I am very grateful to my parents for their illuminating guidance ever since I was a child. I would like to give my special thanks, from my deepest heart, to my mother for all her sacrifice and hardship of raising me up and supporting me toward higher education and new challenges ever.

Contents

1 Introduction **1**

1.1 Technology Trends and Research Goal 1

1.2 Contributions 5

2 Preliminaries **8**

2.1 Supply and Clock Network Modeling 8

2.1.1 Supply Network Modeling 8

2.1.2 Clock Network Modeling 9

2.2 Circuit Analysis Basics 10

2.3 Interconnect Performance Metrics 11

2.3.1 Power Grid Noise Metric 11

2.3.2 Delay modeling 12

2.4 Transient Adjoint Sensitivity Analysis 13

2.5 Global Wire Congestion Measurement 15

3 Fast Analysis and Optimization of Structured Power/Ground Networks 16

3.1 Introduction 16

3.1.1 Previous Work 17

3.1.2 Motivation for Hybrid Topology 17

3.1.3 Contributions 18

3.2 Hierarchical Analysis of the P/G Network Incorporating Non-zero Initial Conditions 22

3.2.1 Reduction of the Trees 23

3.2.2 Solving the Mesh 23

3.2.3 Propagating Waveforms Down the Trees 24

3.3 Adjoint Sensitivity Computation Over Multiple Intervals 26

3.3.1 Sensitivity with Respect to Capacitors 27

3.3.2 Sensitivity with Respect to Resistors and Inductors 29

3.3.3 Closed-form Transient Sensitivity Formula 30

3.4 Heuristic Optimization 32

3.5 Experimental Results 35

3.6 Conclusion 37

4 Optimal Decoupling Capacitor Sizing and Placement for Standard Cell Layouts 39

4.1 Introduction and Motivation 39

4.1.1 Motivation 39

4.1.2 Previous Work 42

4.1.3 Contributions 43

4.2 Power Supply Noise and Sensitivity Analysis 44

4.2.1 Noise Analysis 44

4.2.2 Integral Sensitivity Computation 45

4.3 Optimization and Placement 48

4.3.1 Problem Formulation 48

4.3.2 Optimization and Placement Scheme 49

4.4 Experimental results 50

4.5 Conclusion 54

5 Hybrid Clock Distribution Network Construction 55

5.1 Introduction 55

5.1.1 Clock Design Objective 55

5.1.2 Previous Work 56

5.1.3 Contributions 56

5.2 Linear-programming Based Zero-skew Mesh Construction 59

5.2.1 Mathematical Derivation 59

5.2.2 Linear Programming Formulation 63

5.2.3 Wire Elongation Heuristic 63

5.2.4 Transition Time Improvement Heuristics 66

5.2.5 Overall Construction Algorithm 67

5.2.6 Extension to Multiple Level Mesh Structures 67

5.3 Fast Analysis and Heuristic Optimization 68

5.3.1 Fast Higher-order Delay Calculation 68

5.3.2 Calculation of Sensitivities 69

5.3.3 Heuristic Optimization Procedure 70

5.4 Experimental Results 72

5.5 Conclusion 79

6 Congestion-driven Codesign of Power and Signal Networks 80

6.1 Introduction 80

6.1.1 Motivation for the Codesign of Signal and Power Wires 80

6.1.2 Previous Work on Signal Wires and Power Wires Individually 82

6.1.3 Contributions 83

6.2 Power Grid-aware Signal Routing 84

6.2.1 Power Grid-aware Congestion Estimation 84

6.2.2 Congestion-driven Global Routing 86

6.3 Power Supply Noise Analysis 87

6.4 Power Grid Design Scheme 88

6.4.1 Power Grid Wire Removal Heuristic 89

6.4.2 Power Grid Sizing 92

6.5 Overall Flow of the Algorithm 94

6.5.1 Overall Flow 94

6.5.2 Complexity Discussions 96

6.6 Experimental Results 96

6.7 Conclusion 101

7 Conclusions **103**

List of Figures

2.1 RLC π-model of wire segment. 9

2.2 Linear buffer modeling. 10

2.3 Illustration of the voltage droop at a given node in the V_{dd} power grid. The area of the shaded region corresponds to the integral z at that node. 12

2.4 A Circuit N and its adjoint circuit η. 13

2.5 Wire congestion estimation based on tessellation on a chip. 15

3.1 A structured P/G bus topology. 19

3.2 Switching events at a node in a P/G network. 19

3.3 Voltage droop of node j in a power network during event $i{-}1$ and i. . 20

3.4 Simulation results on a 1000-node and a 2500-node supply network. The reduced orders for the two networks are 13 and 15, respectively. HPRIMA stands for our Hierarchical PRIMA simulator. $V_{dd} = 1.65V$. 36

4.1 A canonical and approximate circuit representation of a power network. 41

4.2 One row of cells in a standard cell layout showing decap locations. . . 43

4.3 Compressed piecewise-linear waveforms for the original and adjoint networks. 47

4.4 Illustration of the initial equal distribution of decaps. 49

4.5 The original voltage droop contour of the Vdd plane. 52

4.6 The original voltage droop contour of the ground plane. 52

4.7 Results of the decap placement algorithm on chip2. The dark regions represent the standard cells, and the light regions are the decaps. . . 53

4.8 Variation of the noise metric with the occupancy ratio. 53

5.1 A clock network as a mesh/tree structure. 57

5.2 One-level mesh with underlying trees. 60

5.3 Mesh “shrinking” heuristic. 66

5.4 Two-level mesh with underlying trees. For simplicity, only some of the sixteen zero skew trees are shown. 68

6.1 Congestion-driven power grid design and global routing. 83

6.2 Wire congestion estimation based on tessellation on a chip. 85

6.3 Illustration of the criticality of power wires. 90

6.4 Heuristic power wire removal. 92

6.5 Power grid sizing procedure. 94

6.6 Congestioin map after the initial routing. 101

6.7 Voltage droop contour on the ground plane. 101

6.8 Optimal M3 and M4 power grid of ac3. 102

List of Tables

1.1 Trends in IC technology parameters. 2

3.1 Runtime comparisons with HSPICE. 37

3.2 Optimization results. 38

3.3 Topology comparison 38

4.1 IC technology parameters. 40

4.2 Optimization results 51

5.1 Comparison between structures of clock nets. 73

5.2 Mesh wire elongation ratio. 74

5.3 Wire width optimization results. 75

5.4 Process variation results. 76

5.5 Hybrid clock networks after heuristic transition time improvement. . 77

5.6 Comparison with the IBM structure. 78

6.1 Test circuit parameters. 97

6.2 A comparison of the congestion improvement after global routing using the conventional approach and our new method. 98

6.3 Optimal power grid results. 99

6.4 CPU time statistics and number of iterations for the benchmarks. . . 100

Chapter 1

Introduction

1.1 Technology Trends and Research Goal

Moore's Law [1] defined the technology scaling trends for the past thirty years and it is expected that the exponential scaling will continue for at least another 10 to 14 years as projected in the International Technology Roadmap for Semiconductors (ITRS'01) [2], summarized in Table 1.1. As a result, over 2.8 billion transistors will be integrated on a single-chip with a clock frequency of 29GHz in the 22nm technology by Year 2016.

It can be seen from Table 1.1 that the trend for high performance integrated circuits is towards ever higher operating frequency, lower power supply voltages and higher power dissipation. The increased power consumption is driven by higher operating frequency and the higher overall capacitance and resistance of larger chips with more on-chip functions [2]. Global interconnect, which distributes the clock and other signals and providing power/ground to the various circuit functions on a chip, has become critical in determining system performance and reliability.

This work aims at the optimal design of various global wires. In specific, it

Year	Tech	Num	Num	f	V_{dd}	Size	Power
	Node	of	Wire	(MHz)	(V)	(mm^2)	(W)
	(nm)	Tran	Level				
2001	130	89M	7	1684	1.1	310	130
2002	115	112M	$7 \sim 8$	2317	1.0	310	140
2003	100	142M	8	3088	1.0	310	150
2004	90	178M	8	3990	1.0	310	160
2005	80	225M	$8 \sim 9$	5173	0.9	310	170
2006	70	283M	9	5631	0.9	310	180
2007	65	357M	9	6739	0.7	310	190
2010	45	714M	$9 \sim 10$	11511	0.6	310	218
2013	32	1427M	$9 \sim 10$	19348	0.5	310	251
2016	22	2854M	10	28751	0.4	310	288

Table 1.1: Trends in IC technology parameters.

addresses both the separate design of power and clock distribution networks, as well as the codesign of power and signal networks.

The power distribution network (also known as supply network) provides the V_{dd} and ground signals throughout a chip. These signals are among the most important signals to control reliably since supply voltage variations can lead not only to problems related to spurious transitions in some cases, particularly when dynamic logic is used, but also to delay variations [3] and timing unpredictability. In deep sub-micron designs, power supply noise is among the major reasons that affect the circuit functionality. The problems in power grid are: the voltage droop due to current flow in the network, ground bounce due to inductive effects, and possible electromigration effects due to excessive current densities. Even if a reliable supply is provided at an input pin of a chip, it can deteriorate significantly within the chip due to the fact that the conductors that transmit these signals throughout the chip are electrically imperfect.

Various techniques including topology optimization, wire sizing optimization and decoupling capacitor placement have been applied to solve the above problems in supply networks. In this thesis, we study power grid design problems in the following aspects: topology design, fast analysis methods, wire sizing and automatic decoupling capacitor optimization/placement schemes.

Clock distribution has become an increasingly challenging problem for VLSI designs, and careful design of clock networks is essential in high-performance circuits.

The timing metrics for a clock distribution network include:

- *Skew*, defined as the maximum difference of the delays from the clock driver or buffer to the clock pins or buffer input capacitances.
- *Propagation delay*, defined as the maximum 50% delay from the clock driver or buffer to the clock pins or buffer input capacitances.
- *Transition time (or slew rate or rise time)*, defined as the time it takes for the step/ramp response waveform to change from 10% to 90% of the V_{dd} level.

In addition, the tolerance of clock skew to process variations and the skew sensitivity to loading conditions are also very important factors to determine the performance of clock networks.

The most common objectives of clock network design are to ensure near-zero skew, sharp transition times, small propagation delays and the optimal use of routing resources. Various clock network design techniques include topology design, wire sizing optimization and buffer insertion. In [4], different clock network topologies are reviewed and they drew a conclusion that each of these topologies have both advantages and disadvantages that should be considered for a high speed design and none of these solutions is ideal in itself, which motivates designs of hybrid clock distribution structures that marry the positive attributes of the various techniques. In this thesis, we construct our own zero-skew hybrid clock network and provide fast and accurate analysis and performance optimization technique for such a structure.

In high-performance integrated circuits, signal wires and power wires compete for routing resources. As the number and criticality of global signal wires becomes more dominant, there is a strong need for a unified approach to the design of signal wires and power grids, with an integrated approach to routing resource management. In this thesis, we provide a solution to this problem and present our congestion-driven design flow to perform a concurrent optimization of the power grid along with signal wires under routing congestion constraints.

1.2 Contributions

The major contributions of this thesis are:

- We propose a hybrid structure that is intermediate to full meshes and pure trees to benefit from the advantages of both structures. We apply such a structure to both the design of supply networks and clock distribution networks. In specific, we study a power/ground (P/G) network structure modeled as an overlying mesh with underlying trees originating from the mesh and a clock network structure with overlying zero-skew mesh driving underlying zero-skew buffered trees. For P/G networks, our proposed structure is amenable to fast analysis, while also maintaining the performance by using mesh structures in the upper level. Experimental results for clock networks show that the hybrid mesh/tree construction scheme can provide smaller propagation delay and transition time than a comparable clock tree. The tolerance to process variations is much better than a clock tree.

- For the hybrid-structured supply network, we present an analysis/sensitivity calculation/heuristic optimization procedure to design reliable power distribution networks for ASIC designs, where fast turnaround time of the design is a major criterion, and ease of analysis of the P/G network with an acceptable performance hit can greatly ease the task of P/G network optimization. For such a structure, we first develop a fast and efficient event-driven P/G network simulator (HPRIMA) that hierarchically simulates the P/G network with an adaptation of PRIMA [5] to handle non-zero initial conditions. Then, we extend the traditional adjoint sensitivity analysis technique appropriatly over multiple switching intervals to deal with the variant topology of the P/G network. The sensitivity computation is coupled with our efficient PRIMA-based order reduction approach so that it can handle large-scale P/G circuits. A closed-form transient sensitivity expression is provided for a PRIMA approximation of a given order. The TILOS-like [6] heuristic wire width optimization is performed according to the gradients calculated through the adjoint sensitivity analysis. We have successfully extended this procedure to include the optimization of decoupling capacitors.

- We propose and demonstrate an algorithm for the automated placement and sizing of decaps for standard cell layouts. In this work, a compressed piecewise linear voltage waveform for every node in the power grid circuit is stored to reduce the amount of memory to be used. Thereafter, the complexity of the convolution computation in the adjoint sensitivity analysis has been significantly reduced. Experimental results show that power grid noise can be significantly reduced after a judicious optimization of decap placement, with little change of the total chip area.

- We develop a novel technique that describes how some classes of meshes can be constructed very easily for zero skew using a simple linear-programming based formulation. This method can be extended very easily to build clock trees with prespecified nonzero skews for cases when deliberate skews are used in the clock network for cycle borrowing [7]. While the first-order Elmore Delay is used in this technique to build the initial zero-skew mesh/tree structure, for accurate analysis of delays and skews and further optimization of the constructed clock network to some target transition time and skews, we extend our hierarchical HPRIMA simulator to analyze the step response of every clock pin. The reduced order modeling approach based on HPRIMA is also employed in calculating the adjoint sensitivities of delays, as well as of skew and transition time with respect to all the wire widths to guide our heuristic wire width optimization for meeting more stringent skew and transition time constraints.

- We perform a concurrent optimization of the power grid along with signal wires under routing congestion constraints. By adding a feedback loop into the conventional global routing flow, we can alter the pre-designed power grid according to the congestion information and re-route congested signal nets using updated routing resources. The power grid adjustments consist of wire removal from the grid in congested regions, and sizing of the power grid to compensate for this removal. Our approach incorporates a tight coupling between power grid adjustments and the routing of signal wires to exploit the altered congestions that result from these adjustments, and aims to solve problems with severe congestion constraints where conventional techniques are inadequate. Experimental results show that the overall routability can be significantly improved while the power grid noise is maintained within the voltage droop constraint.

Parts of this research have been published in [8–13].

Chapter 2

Preliminaries

2.1 Supply and Clock Network Modeling

2.1.1 Supply Network Modeling

The following supply network model has been widely used in industry [14, 15].

- The power distribution network (grid) is modeled as a resistive mesh.
- The cells are modeled as time-varying current sources connected between power and ground.
- The decoupling capacitors are modeled as single lumped capacitors connected between power and ground.
- The top-level metal is connected to a package modeled as an inductance connected to an ideal constant voltage source.

. We use the above model for the work of decoupling capacitor optimization and the work of congestion-driven power and signal networks because both are industrysupported projects aiming to solve problems in current technologies where the inductance is dominated by pin inductances.

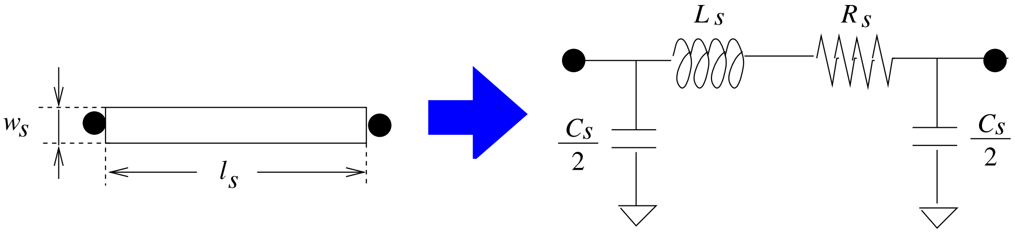

Figure 2.1: RLC π-model of wire segment.

For more accurate analysis, a slightly complicated interconnect model for the P/G network is used. In this model, each wire in the power grid is represented as a set of connected segments under the π-model (shown in Figure 2.1), with each segment modeled using lumped RLC parameters given by

$$\begin{aligned} R_s &= \rho l_s / w_s \\ C_s &= (\beta w_s + \alpha) l_s \\ L_s &= \gamma l_s / w_s \end{aligned} \tag{2.1}$$

where l_s and w_s are the length and the width of the segment, and the parameters ρ, β, α and γ are the sheet resistance per square, sheet capacitance per square, fringing capacitance per unit length and the inductance per square of the metal layer that is being used for routing the P/G network. Package pins are modeled as an RLC branch connected to pads on the mesh.

2.1.2 Clock Network Modeling

The clock distribution network is usually modeled as a lumped RC network. The RC parameters can be found out using corresponding equations in Eq. (2.1). Loads at the leaf nodes of the clock distribution network, which are clock pins to the memory elements (i.e., flip-flops, latches, etc), are modeled as sink capacitances. Clock buffers use the linear model which consists of buffer input capacitance (C_g), intrinsic delay (D_0) and effective resistance (R_d). An ideal clock input signal is assumed to be applied to the buffer effective resistance. The buffer model is illustrated in Figure 2.2.

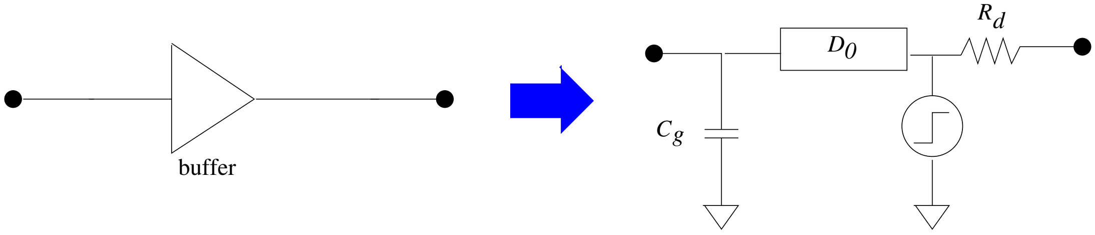

Figure 2.2: Linear buffer modeling.

2.2 Circuit Analysis Basics

The behavior of linear circuits can be described in the time domain by a first order differential equation formulated using modified nodal analysis (MNA) [16]:

$$Gx(t) + C\dot{x}(t) = u(t) \tag{2.2}$$

where x is a vector of node voltages and source and inductor currents; G is the conductance matrix; C includes both capacitance inductance terms, and $u(t)$ includes the current and voltage sources.

The MNA equation can be solved using the Backward Euler integration formula [16]:

$$(G + C/h)x(t+h) = u(t+h) + x(t)C/h \tag{2.3}$$

Where h is the time step for the transient analysis.

The linear circuit system can be transformed in the frequency domain. Applying the Laplace transform, Eq. (2.2) becomes

$$(G + sC)V(s) = J(s) \tag{2.4}$$

Power distribution network is huge in terms of total number of nodes. In both Eq. (2.2) and (2.4), matrices G and C are large such that solving these equations directly will be extremely slow and significant amount of memory will be used. Various research (i.e., [8, 17, 18]), either in the time domain or in the frequency domain, has been performed on seeking techniques to solve these equations fast and efficiently.

2.3 Interconnect Performance Metrics

2.3.1 Power Grid Noise Metric

Maximum voltage droop of a node is defined as the largest voltage droop value along the period of time for simulation. The maximum voltage droop among all nodes in the power grid circuit can give a rough idea about the performance of the power grid. However, such a measurement is very sensitive to the accuracy of circuit analysis and does not take the timings of the voltage violations into account. Hence it would be inefficient to be used as the objective function for optimization purpose.

An efficient noise metric for the performance of every node in a power/ground network is the integral of voltage droop beyond the noise margin:

$$\begin{aligned} z_j(p) &= \int_0^T \max\{NM_H - v_j(t,p), 0\}dt \\ &= \int_{t_s}^{t_e} \{NM_H - v_j(t,p)\}dt, \end{aligned} \tag{2.5}$$

where p represents the tunable circuit parameters. It is represented by the shaded area in Figure 2.3. This idea was first introduced in [19] and it proves to be an efficient measure for circuit optimization. In our work, this metric has been applied successfully to efficient optimization of power distribution networks.

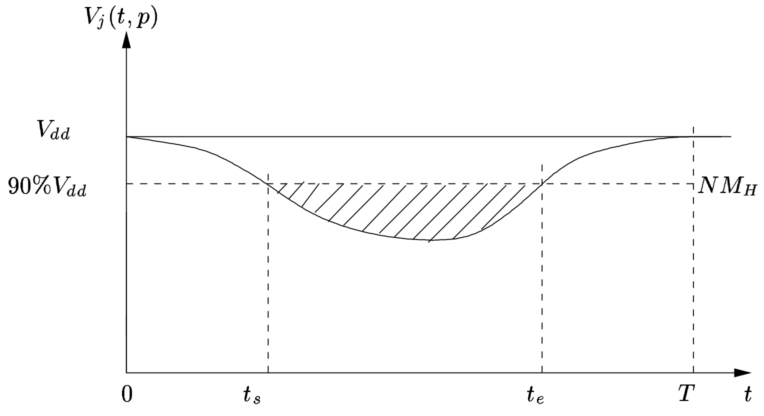

Figure 2.3: Illustration of the voltage droop at a given node in the V_{dd} power grid. The area of the shaded region corresponds to the integral z at that node.

2.3.2 Delay modeling

For an RC circuit, Elmore delay [20] has been widely used to estimate interconnect delays because of its simplicity and high fidelity [21]. The definition of Elmore delay is described by Eq.(2.6):

$$\vec{\tau_k} = G^{-1}C \tag{2.6}$$

For an RC tree structure with driver resistance R_d and total downstream capacitance C_i seen from tree node v_i, the Elmore delay from driver to a sink v_k can be recursively calculated as

$$\vec{\tau_k} = R_d C_0 + \sum_{e_{ij} \epsilon path(v_0, v_k)} R_{ij}(\frac{C_{ij}}{2} + C_j) \tag{2.7}$$

Elmore delay tends to overestimate the interconnect delay because it does not correctly take the resistive shielding effect [22] into consideration. The degree of fidelity becomes lower as interconnect wires become thinner and longer. Hence we use Elmore delay model during the initial construction of the zero-skew clock structures and use an accurate higher-order delay metric to post-optimize the performance of of

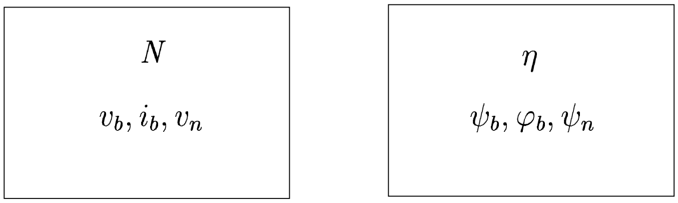

Figure 2.4: A Circuit N and its adjoint circuit η.

the network. In our work, the higher-order delays are analytically obtained from the higher-order step response waveforms simulated by our HPRIMA-based simulator [9].

2.4 Transient Adjoint Sensitivity Analysis

Incremental sensitivity is defined as the partial derivative of a circuit response of interest with respect to a parameter of interest. It is valid in a small range around the nominal value of the parameter of interest. *Direct sensitivity* and *adjoint sensitivity* are two methods of the incremental sensitivity analysis, where the direct method solves the sensitivity of many circuit responses with respect to a single parameter while the adjoint methods calculates the sensitivity of one performance function with respect to many circuit parameter values [16, 23, 24]. In this thesis, the use of the adjoint method is a natural choice since we are interested in the sensitivity of the noise metric in power grids and the sensitivity of skew and transition time in clock networks with respect to all parameters in the interconnect circuit.

Adjoint sensitivity analysis is based on Tellegen's theorem (Eq. (2.8)), which states that the instantaneous power in any circuit is zero.

$$\sum v_b i_b = 0 \tag{2.8}$$

Given a pair of circuits with identical topologies (Figure 2.4), from Tellegen's theorem and the KCL and KVL relations of each circuit, the following basic sensitivity relation over a time period of interest from t_0 to t_f can be derived:

$$\sum_{all\ branches} \int_{t_0}^{t_f} [\varphi \delta v - \psi \delta i] dt = 0 \tag{2.9}$$

A detailed derivation process can be found in [16] and will be skipped here. We summarize the analysis process as follows: applying Eq. (2.9) to circuit elements such as capacitors, inductors and resistors, choosing corresponding adjoint elements of the same value as those in the original circuit, defining a time axis variable τ (for the purpose of analyzing the adjoint network) to be backward time t of the original network (i.e., $\tau = t_0 + t_f - t$), and setting initial conditions in the adjoint circuit zero and removing original voltage and current sources, we have the transient adjoint sensitivity formula as follows:

$$\int_{t_0}^{t_f} \frac{\partial f(t)}{\partial C} dt = -\int_{t_0}^{t_f} [\psi_C(\tau) \dot{v}_C(t)] dt \tag{2.10}$$

$$\int_{t_0}^{t_f} \frac{\partial f(t)}{\partial R} dt = \int_{t_0}^{t_f} [\varphi_R(\tau) i_R(t)] dt \tag{2.11}$$

$$\int_{t_0}^{t_f} \frac{\partial f(t)}{\partial L} dt = -\int_{t_0}^{t_f} [\varphi_L(\tau) \dot{i}_L(t)] dt \tag{2.12}$$

where $f(t)$ is the performance function of interest.

The same notations will be used throughout the thesis, i.e. the voltages [currents] in the original network are denoted by v [i], while the voltages [currents] in the adjoint network are ψ [φ], and the symbols t and τ denote the temporal variables in the original and adjoint network, respectively.

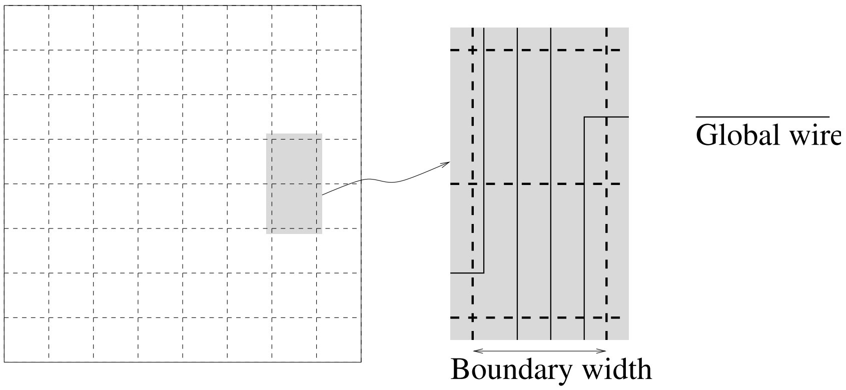

Figure 2.5: Wire congestion estimation based on tessellation on a chip.

2.5 Global Wire Congestion Measurement

In global routing, the entire chip is tessellated into an array of grid cells (shown in Figure 2.5) and the wiring information across the boundaries between neighboring grid cells is used to estimate the signal wire congestion distributions. If the capacity of a boundary b is $C(b)$ and there are $S(b)$ signal wires that cross the boundary, then the overflow on the boundary b is $S(b) - C(b)$. All tile boundaries with negative overflow values form a congestion map for a chip. The wire density at b is represented as $S(b)/C(b)$, measuring the congestion of b.

Chapter 3

Fast Analysis and Optimization of Structured Power/Ground Networks

3.1 Introduction

The design of power/ground (P/G) networks is critical to the correct functioning of a chip. With the rapid increases in the clock frequency and reduction of feature sizes of high-speed electronic circuits, it is becoming more and more important to design and optimize P/G networks fast and efficiently. The major reasons that affect the circuit functionality are the voltage droop due to current flow in the network, ground bounce due to inductive effects, and possible electromigration effects due to excessive current densities. The first two can lead to unacceptable circuit switching speeds and/or glitches, while the latter places a limit on the useful lifetime of a chip [14].

3.1.1 Previous Work

Various algorithms and simplified device models for P/G networks that offer faster but less accurate results have been explored in the past. Early work on P/G networks focuses on tree-like structures so as to allow the use of path tracing algorithms for efficiency [25, 26] and assumes resistance-only models for the network.

A 3-stage IR-drop analysis methodology during the whole design process is presented in [27]. The authors of [17] propose a hierarchical analysis technique and a novel sparsification method based on 0-1 integer linear programming. A PDE-like multigrid method is proposed in [18] to perform both DC and transient simulation of power grids efficiently. Each of the above methods aims at speeding up the analysis and predicting the power grid performance properly.

Other related work on optimizing P/G networks includes [28–30], which use techniques ranging from simulated annealing to the solution of a sequence of linear programs for wire widening, or [31], which optimizes the topology of the P/G network. A frequency domain sensitivity-based decoupling capacitor optimization method is proposed in [32]. [33] formulates the P/G network optimization problem as a nonlinear convex optimization problem.

3.1.2 Motivation for Hybrid Topology

Most of the existing techniques have focused on methods that optimize a specific topology that is typically specified by the user to be a large and complex mesh. There is an inherent conflict between P/G networks that are easy to analyze, and those that provide reliable power levels and evenly distributed current densities. While tree structures provide all of the former benefits, they result in poor quality in P/G signal delivery. On the other hand, dense meshes are excellent in satisfying the latter requirement but are very computationally difficult to analyze. For example, the work in [14] shows that it requires several hours to analyze a P/G network using SPICE. The key idea used in this work is that an approach that meets both of these requirements would be something between a pure tree and a full mesh. In this work, we use one such topology skeleton with a global mesh feeding local trees, as described later in this section; a similar method has been used in [34]. However, we emphasize that this approach can be modified to other topologies that are intermediate to the two extremes of full trees and full meshes: one such example is a global mesh that feeds smaller unconnected local meshes.

We point out that such an approach may not be optimal for a high-performance full-custom microprocessor, where a dense mesh may be essential for reliable P/G levels[1]. However, we believe that it will be of great utility in ASIC design, where fast turnaround time of the design is a major criterion, and ease of analysis of the P/G network with an acceptable performance hit can greatly ease the task of P/G network optimization. For this scenario, we present an analysis/sensitivity calculation/optimization procedure in this chapter.

3.1.3 Contributions

3.1.3.1 Hybrid Mesh/Tree Structure

The P/G network model proposed here, characterized by the mesh/tree topology of [34], is illustrated in Figure 3.1; for simplicity, only one tree is shown in the figure. An overlying coarse mesh structure of a user-specified topology provides global distribution of the P/G signals across the chip. From various nodes of this mesh, tree structures of user-specified topologies originate and distribute the supply voltage to the utilization points, each of which is modeled as an equivalent RC branch. The

[1] Even for processor designs, the use of more general hierarchical structures is not uncommon [17].

advantage of using an RC branch instead of an equivalent current source is that such a model also captures the loading effects of the utilization points on the P/G network. Practically, the designer often specifies a set of worst-case switching patterns and a list of the potential critical nodes in the network. Each specified switching event in a switching pattern provides information on which RC elements at the utilization points load the network at a given time. An example of a switching pattern for a given utilization point is shown in Figure 3.2, where each arrow indicates which RC element has entered or left the network due to a switching event at a gate.

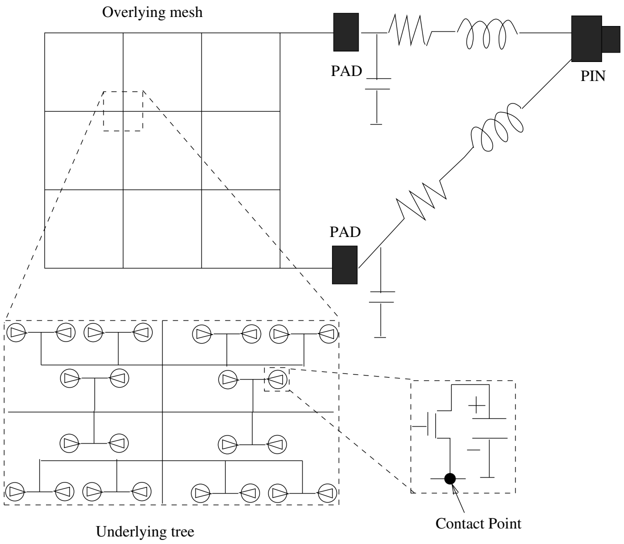

Figure 3.1: A structured P/G bus topology.

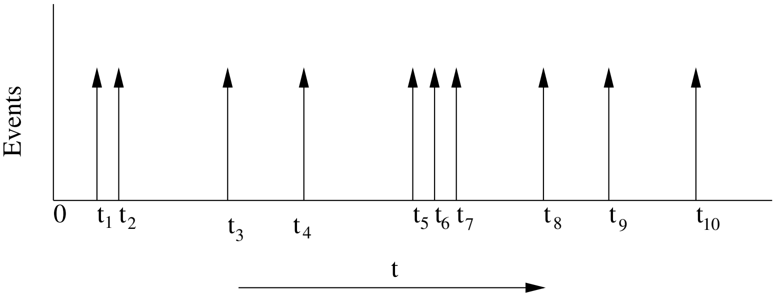

Figure 3.2: Switching events at a node in a P/G network.

3.1.3.2 Fast Noise and Sensitivity Analysis and Optimization

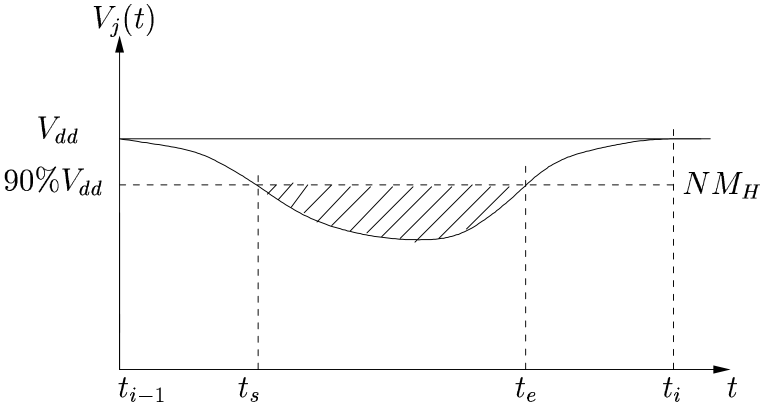

Figure 3.3: Voltage droop of node j in a power network during event $i{-}1$ and i.

As described in Chapter 2, the noise metric of a node between event $i - 1$ and i is the integral of voltage droop beyond the noise margin, which is represented by the shaded area in Figure 3.3. The most critical node in the power/ground network is defined as the node with the maximum sum of voltage droop integrals over all switching events. Our optimization problem is to minimize the total P/G bus area, subject to the constraints that the voltage droop integral of the most critical node is less than or equal to zero. The process of optimizing the P/G network requires an iterative loop within which it is necessary to analyze the network to determine whether it satisfies specifications or not, and to determine gradient information to guide the optimization.

Several techniques may be used for this optimization: varying the topology, varying wire widths, and adding decoupling capacitors. In this work, we focus on optimization by varying wire widths and extend this technique to add decoupling capacitors.

The structure in Figure 3.1 is amenable to fast analysis, while also maintaining the performance by using mesh structures that reduce the voltage drops and current densities in the highest current regions. In [34], an AWE-based technique is proposed for simulating P/G networks. While AWE tends to be unstable at higher orders of approximation, for such a structure, the task of analysis is rapidly performed using PRIMA [5], a reduced order modeling technique that produces provably passive macromodels.

Transient sensitivity has been particularly useful in circuit optimization and tuning and has been used to provide gradient information, for example, in [35] and [19]. In the case of P/G network optimization, the optimization of the objective function requires the computation of transient sensitivity for the specified critical nodes with respect to all the elements in the whole circuit, and we employ the adjoint method [16].

Traditionally, transient sensitivity computation for a circuit with a fixed topology is performed by a convolution between the forward-in-time voltage/current slope of the element (capacitor, resistor or inductor) in the original circuit and the backward-in-time voltage/current across the same element in the adjoint circuit, where the same fixed topology is chosen for the pair of circuits.

In our work, the topology of the original P/G network changes at the beginning of each switching event as new RC elements are added to the network or removed from it. This chapter presents an appropriate extension of the adjoint network technique over multiple-intervals so that the sensitivities can be efficiently computed. Our sensitivity computation is coupled with an efficient PRIMA-based order reduction approach so that it can handle large-scale P/G circuits. A closed-form transient sensitivity expression is provided for a PRIMA approximation of a given order. Finally, a TILOS [6]-like heuristic optimization procedure is proposed.

3.2 Hierarchical Analysis of the P/G Network Incorporating Non-zero Initial Conditions

The procedure for analysis of the power and ground network is symmetric. In other words, it is enough to develop a solution technique for one of these, and it is easy to extend this to solve the other.

As described in Chapter 2, the interconnect in the P/G network is modeled as a large-scale RLC linear circuit. The simulation is event-driven, proceeding one interval at a time, starting from the first interval until the last, updating the corresponding switch states specified in the event list while moving from one interval to the next. The final state, i.e., capacitor voltages and inductor currents, at the end of each interval constitutes the initial state for the next interval. We assume that the initial conditions at all nodes in the network are at a user-specified level.

The entire system can be described by the Laplace equation

$$(G + sC)V(s) = J(s) \tag{3.1}$$

The size of vector V(s) is equal to the sum of total number of mesh nodes, number of nonzero mesh and package inductances, total number of trees and each entire tree stamps. The objective of fast analysis is to reduce this system to a smaller system that captures the response of the system to the given set of inputs and the initial conditions in the system.

The hierarchical model reduction and simulation proceeds in three stages: first, all trees are reduced to an equivalent passive model. Next, the mesh is solved, along with these passive models to find all mesh voltages. Finally, these mesh voltages provide the voltage source at the root of each tree and are used to solve each tree individually and independently. This hierarchical approach serves to reduce the amount of computation required during the analysis.

3.2.1 Reduction of the Trees

The MNA equation for each of the trees with initial conditions can be written as

$$(G_T + sC_T)V_T(s) = [J_1(s)\ J_2(s)\ J_3(s)] \tag{3.2}$$

where $J_1(s)$ is the input excitation to the tree, which is 1 since we are interested in the transfer function (impulse response) of the tree during the later waveform propagation step. An initial condition at time T on a capacitor C_i[inductor L_i] may be modeled as a voltage [current] source of value $V_{Ci}(T)$ $[I_{Li}(T)]$ in series [parallel] with a capacitor [inductor] with zero initial conditions. The vector $J_2(s)$ captures these initial conditions of the tree and has entries of the type $C_iV_{Ci}(T)$ and $L_iI_{Li}(T)$, where the multiplications by C_i and L_i correspond to conversions between Thevenin and Norton forms for ease of application for the formulation. The vector $J_3(s)$ is only considered when an impulse current source or a current source of square wave is applied to the most critical node in the adjoint network discussed in Section 3.3.

The PRIMA reduction procedure is applied to obtain a provably passive tree reduction. A RICE-like tree traversal [36] computes the orthonormal basis $\boldsymbol{X}$ of the Krylov space, so that the procedure is extremely fast. The three-column right hand side matrix in Eq. (3.2) tells us that three columns are added to $\boldsymbol{X}$ in each iteration and to obtain a reduction of order q_T, $\lceil \frac{q_T}{3} \rceil$ iterations are required. While for the analysis of the original network, the right hand side only includes two columns ($[J_1(s)\ J_2(s)]$) and consequently the number of iterations is $\lceil \frac{q_T}{2} \rceil$.

3.2.2 Solving the Mesh

Substituting the reduced order model of each tree can reduce the MNA equation (3.1) for the whole system to

$$(G_m + sC_m)V_m(s) = J_m(s) \tag{3.3}$$

Comparing to the size of vector $V(s)$, the size of vector $V_m(s)$ is reduced to the sum of total number of mesh nodes, number of nonzero mesh and package inductances, total number of trees and each reduced tree stamps, which is still large. PRIMA is applied to Eq. (3.3) again to this reduced system to further reduce the system to a smaller order

$$(\tilde{G}_m + s\tilde{C}_m)\tilde{V}_m = \tilde{J}_m(s) \tag{3.4}$$

Since G_m is sparse, sparse matrix technique can be used to compute the orthonormal basis $\boldsymbol{X}$ of the Krylov space, where the inverse of the matrix $\tilde{G}_m$ is required. Since the overlying mesh is typically small in terms of the number of nodes and the order of the final system q_m is small, the computational cost of this is also reasonably small.

The transient response of each mesh node in the P/G net is found to be:

$$V_{Mesh}(s) = \sum_{i=1}^{q_m} \frac{r_i}{s - \lambda_i} \tag{3.5}$$

where λ_i and r_i are the ith pole and residue of a mesh node. q_m is the number of dominant poles for the mesh, which is determined by the reduced order of the mesh matrix. After taking an inverse Laplace transform, we have

$$V_{Mesh}(t) = \sum_{i=1}^{q_m} r_i e^{\lambda_i t}, \quad 0 \leq t \leq T_f, \tag{3.6}$$

where T_f is the period of time between two continuous events.

3.2.3 Propagating Waveforms Down the Trees

The solution to Eq. (3.4) is used to set the mesh node voltages to the computed value. These values are used to recursively compute the voltage at each of the internal node in the local trees. This process is continued until all the voltages and currents in the trees have been computed. The voltage at each node is computed as the sum of the zero input response and the zero initial condition response; note that the input for the tree is the voltage at the mesh node (root of the tree), which is typically nonzero.

The propagation formula of a tree node in Laplace domain is

$$V_{Tree}(s) = \sum_{j=1}^{q_T} \frac{r_{Tz_j}}{s - \lambda_{Tz_j}} + \sum_{i=1}^{q_m} \sum_{j=1}^{q_T} \frac{r_i}{s - \lambda_i} \times \frac{r_{Timp_j}}{s - \lambda_{Timp_j}}, \tag{3.7}$$

where λ_{Tz_j} and$r_{Tz_j}$are the jth pole and residue of the local tree's zero input response, λ_{Timp_j} and r_{Timp_j} are the jth pole and residue of its impulse response. q_T is the number of dominant poles for each local tree. Eq. (3.7) can further be derived into the following form using the partial fractional method:

$$V_{Tree}(s) = \sum_{i=1}^{Q_T} \frac{r_{T_i}}{s - \lambda_{T_i}} \tag{3.8}$$

where $Q_T = q_m + q_T$ and the poles of the tree are

$$\lambda_T = \begin{cases} \lambda_i, & i = 1 \cdots q_m \\ \lambda_{Timp_j} = \lambda_{Tz_j}, & j = 1 \cdots q_T \end{cases} \tag{3.9}$$

and the corresponding residues are

$$r_T = \begin{cases} \sum_{j=1}^{q_T} \frac{r_i r_{Timp_j}}{\lambda_i - \lambda_{Timp_j}}, & i = 1 \cdots q_m \\ r_{Tz_j} + \sum_{i=1}^{q_m} \frac{r_{Timp_j} r_i}{\lambda_i - \lambda_{Timp_j}}, & j = 1 \cdots q_T \end{cases} \tag{3.10}$$

By taking inverse Laplace Transform we get:

$$V_{Tree}(t) = \sum_{i=1}^{Q_T} r_{T_k} e^{\lambda_{T_k} t}, \; 0 \leq t \leq T_f \tag{3.11}$$

Currents flowing through mesh branches are solved from the MNA equation of the mesh, while every tree branch current flowing through node i and j is found through $i(s) = (v_i(s) - v_j(s))/(R_{ij}+ sL_{ij})$ and by taking an inverse Laplace transform of $i(s)$ we can get $i(t)$.

3.3 Adjoint Sensitivity Computation Over Multiple Intervals

Adjoint sensitivity analysis is a standard technique for circuit optimization where the sensitivity of one output with respect to many parameter values is required [16]. In adjoint sensitivity analysis, Tellegen's theorem is applied to a pair of circuits with the same topology by combining the branch currents and voltages at any two instants of time.

For our problem, we simulate the P/G network over the specified event list. At the beginning of each event, a set of switching activities occurs, with some RC elements switching out of the network and others switching in. This complicates the task of adjoint sensitivity computation since the topology changes for each interval. One contribution of this work is to extend adjoint analysis to this variable topology.

The basis for adjoint analysis comes from Tellegen's theorem, which when integrated over a time period of interest from t_0 to t_f, gives

$$\sum_{all\ branches} \int_{t_0}^{t_f} [\varphi(\tau)\delta v(t) - \psi(\tau)\delta i(t)]dt = 0 \tag{3.12}$$

Suppose there are a total of f +1 events, each event is lasting from $t_{k-1}$to t_k, k=1 to f, where t_0=0. Then (13) becomes

$$\sum_{all\ branches} \sum_{k=1}^{f-1} \int_{t_{k-1}}^{t_k} [\varphi^{(k)}(\tau)\delta v^{(k)}(t) - \psi^{(k)}(\tau)\delta i^{(k)}(t)]dt = 0 \tag{3.13}$$

The superscript $^{(k)}$ denotes the voltage or current response corresponding to the topology between switching event $k-1$ and k.

If we are interested in the sensitivity of $v(t)$ at some moment t, we isolate $\delta v(t)$ by setting all voltage sources in the adjoint circuit to zero. The left-hand side of the sensitivity term becomes

$$\sum_{all\ current\ sources} \left[\sum_{k=1}^{f-1} \int_{t_{k-1}}^{t_k} -\varphi^{(k)}(\tau)\delta v^{(k)}(t)dt \right] \tag{3.14}$$

Suppose the most critical voltage droop occurs at $t = T_{peak}$. To obtain the term $\delta v(T_{peak})$, we set an impulse current source at the node of interest in the adjoint network, i.e.,

$$\varphi^{(p)}(\tau) = -\delta(t - T_{peak}) \tag{3.15}$$

where $t_{p-1} < T_{peak} < t_p$. Only for interval $t_{p-1} < T_{peak} < t_p$, this term is non-zero:

$$\int_{t_{p-1}}^{t_p} \delta(t - T_{peak})\delta v^{(p)}(t)dt = \delta v^{(p)}(T_{peak}) \tag{3.16}$$

which is exactly what is desired in the left-hand side of the sensitivity term.

As described in Section 3.1, in our work, the performance of power/ground network is measured using the shaded area beyond the threshold voltage (noise margin) shown in Figure 3.3. Suppose the most critical node has only one overshoot between t_{sk} and t_{ek} during event k, the sum of shaded area over all switching events can be represented as

$$Z = \sum_{k=1}^{f} \int_{t_{sk}}^{t_{ek}} \left[NM_H - v(t)\right] dt \tag{3.17}$$

As discussed in [19], instead of applying an impulse current source of $-\delta(t - T_{peak})$, a current pulse of $u(t - t_{sk}) - u(t - t_{ek})$, $k = 1 \ldots f$, is applied to the most critical node. Then Eq. (3.14) becomes:

$$\sum_{k=1}^{f} \int_{t_{k-1}}^{t_k} \left[-u(t - t_{sk}) + u(t - t_{ek})\right] \delta v(t)dt = \sum_{k=1}^{f} \int_{t_{sk}}^{t_{ek}} -\delta v(t)dt = \delta Z \tag{3.18}$$

which, similar to Eq. (3.16), is exactly what is desired in the left-hand side of the sensitivity term.

3.3.1 Sensitivity with Respect to Capacitors

For capacitors, we have the device equation

$$\begin{aligned} i_C &= C\dot{v}_C(t) \\ \delta i_C(t) &= C\delta\dot{v}_C(t) + \dot{v}_C(t)\delta C \end{aligned} \tag{3.19}$$

From Eq. (3.13), the right-hand side of the sensitivity term becomes

$$\sum_{k=1}^{f}\int_{t_{k-1}}^{t_k}\{\varphi_C^{(k)}(\tau)\delta v_C^{(k)}(t)-\psi_C^{(k)}(\tau)[C\delta\dot{v}_C^{(k)}(t)+\dot{v}_C^{(k)}(t)\delta C]\}dt \tag{3.20}$$

We can integrate by part the $\delta\dot{v}_C^{(k)}(t)$ term in Eq. (3.20) to obtain[2]:

$$\begin{aligned}&\textstyle\sum_{k=1}^{f}[-\psi_C^{(k)}(\tau)C\delta v_C^{(k)}(t)\Big|_{t_{k-1}}^{t_k}-\delta C\int_{t_{k-1}}^{t_k}\psi_C^{(k)}(\tau)\dot{v}_C^{(k)}(t)dt]\\&+\textstyle\sum_{k=1}^{f}\int_{t_{k-1}}^{t_k}[\varphi_C^{(k)}(\tau)\delta v_C^{(k)}(t)+C\dot{\psi}_C^{(k)}(\tau)\delta v_C^{(k)}(t)]dt\end{aligned} \tag{3.21}$$

As in normal adjoint calculation, to avoid negative energy storage elements, we choose τ to be the backward time in each interval k. Thus we have

$$\tau=t_{k-1}+t_k-t \tag{3.22}$$

The capacitor in the adjoint circuit can be chosen as

$$\varphi_C(\tau)=-C\dot{\psi}_C(\tau)=-C\frac{d\psi(\tau)}{dt}=C\frac{d\psi(\tau)}{d\tau} \tag{3.23}$$

which is an ordinary capacitor. So Eq. (3.21) becomes

$$\begin{aligned}&\textstyle\sum_{k=1}^{f}\left[-\psi_C^{(k)}(\tau)C\delta v_C^{(k)}(t)|_{t_{k-1}}^{t_k}-\delta C\int_{t_{k-1}}^{t_k}\psi_C^{(k)}(\tau)\dot{v}_C^{(k)}(t)dt\right]=\\&-\psi_C^{(1)}(t_0)C\delta v_C^{(1)}(t_1)+\psi_C^{(1)}(t_1)C\delta v_C^{(1)}(t_0)-\delta C\textstyle\int_{t_0}^{t_1}\psi_C^{(1)}(\tau)\dot{v}_C^{(1)}(t)dt\\&-\psi_C^{(2)}(t_1)C\delta v_C^{(2)}(t_2)+\psi_C^{(2)}(t_2)C\delta v_C^{(2)}(t_1)-\delta C\textstyle\int_{t_1}^{t_2}\psi_C^{(2)}(\tau)\dot{v}_C^{(2)}(t)dt\\&\cdots\cdots\\&-\psi_C^{(f)}(t_{f-1})C\delta v_C^{(f)}(t_f)+\psi_C^{(f)}(t_f)C\delta v_C^{(f)}(t_{f-1})-\delta C\textstyle\int_{t_{f-1}}^{t_f}\psi_C^{(f)}(\tau)\dot{v}_C^{(f)}(t)dt\end{aligned} \tag{3.24}$$

To remove the integration-by-parts term in Eq. (3.24), we use the following procedure:

- $\psi_C^f(t_{f-1})=0$(initial conditions for the adjoint circuit is set to zero)

[2] Note that $\dot{\psi}_C$ and $\dot{v}_C$ are not defined everywhere. In particular, at switching time points, the derivatives can be discontinuous although the function is continuous. However, removing a finite number of points constitutes the removal of a set of zero measure and does not alter the evaluated value of the integral, where $\dot{\psi}_C$ and $\dot{v}_C$ are continuous [37].

- $\psi_C^k(t_k) = \psi_C^{k-1}(t_{k-2}), k = 2, \ldots, f$

- $\delta v_C^1(t_0) = 0$(by definition, since the initial conditions of the original circuit are known)

This will set all the non-integral terms in Eq. (3.24) to zero. In case there is only one interval, this result reduces to the conventional adjoint sensitivity calculation procedure [16] that sets $\psi_C(t_0) = 0$. It can be inferred from this that the adjoint circuit simulation proceeds in the backward order of event (time).

The transient sensitivity formula with respect to capacitor C at the moment of T_{peak} is as follows:

$$\frac{\delta v^{(p)}(T_{peak})}{\delta C} = -\sum_{k=1}^{p} \int_{t_{k-1}}^{t_k} \psi_C^{(k)}(t_{k-1} + t_k - t)\dot{v}_C^{(k)}(t)dt \tag{3.25}$$

where $t_{p-1} < T_{peak} < t_p$. Notice that since zero-initial conditions are set in the adjoint circuit and there is no excitation until $t = T_{peak}$ (assuming $t_0 = 0$), i.e., until $\tau = t_f - T_{peak}$, $\psi_C(\tau) = 0$ when $\tau < t_f - T_{peak}$. $\psi_C(\tau)$ is continuous over the period $t_f \geq \tau \geq t_f - T_{peak}$. In other words, $\psi_C(t)$ is continuous over the period $0 \leq t \leq T_{peak}$.

Similarly, the sensitivity of Z with respect to capacitor C is:

$$\frac{\delta Z}{\delta C} = -\sum_{k=1}^{f} \int_{t_{k-1}}^{t_k} \psi_C^{(k)}(t_{k-1} + t_k - t)\dot{v}_C^{(k)}(t)dt \tag{3.26}$$

where $\psi_C(\tau)$ is the voltage droop across capacitor C under the current excitation of $u(\tau - \tau_{sk}) - u(\tau - \tau_{ek})$, $\tau_{sk} = t_{k-1} + t_k - t_{ek}$ and $\tau_{ek} = t_{k-1} + t_k - t_{sk}$, applied to the most critical node.

3.3.2 Sensitivity with Respect to Resistors and Inductors

The sensitivity with respect to resistors and inductors can be derived similarly:

- Apply a current source of -$\delta(t - T_{peak})$ (referring to Eq. (3.16)) or $u(t - t_{sk}) - u(t - t_{ek})$ (referring to Eq. (3.18)) at the most critical node.

- Set $\tau = t_{k-1} + t_k - t$ for each interval k to maintain $d\tau/dt = -1$.

- Choose $\psi_R(\tau) = R\varphi_R(\tau)$ and $\psi_L(\tau) = L\dot{\varphi}_L(\tau)$.

- Set initial conditions in the adjoint circuit to zero.

- Simulate the circuit in the backward order of event

Specifically, for RLC circuits, the transient sensitivity formula with respect to R and L at T_{peak} are

$$\begin{array}{ll} \frac{\delta v^{(p)}(T_{peak})}{\delta R} &= \sum\limits_{k=1}^{p} \int_{t_{k-1}}^{t_k} \varphi_R^{(k)}(t_{k-1} + t_k - t) i_R^{(k)}(t)dt \\ \frac{\delta v^{(p)}(T_{peak})}{\delta L} &= -\sum\limits_{k=1}^{p} \int_{t_{k-1}}^{t_k} \varphi_L^{(k)}(t_{k-1} + t_k - t) \dot{i}_L^{(k)}(t)dt \end{array} \tag{3.27}$$

where$t_{p-1} < T_{peak} < t_p$. Similarly, φ_R (t) and $\varphi_L(t)$ are continuous over the period 0 $\leq t \leq T_{peak}$.

Similarly, the sensitivity of Z with respect to R and L are

$$\begin{array}{l} \frac{\delta Z}{\delta R} = \sum\limits_{k=1}^{f} \int_{t_{k-1}}^{t_k} \varphi_R^{(k)}(t_{k-1} + t_k - t) i_R^{(k)}(t)dt \\ \frac{\delta Z}{\delta L} = -\sum\limits_{k=1}^{f} \int_{t_{k-1}}^{t_k} \varphi_L^{(k)}(t_{k-1} + t_k - t) \dot{i}_L^{(k)}(t)dt \end{array} \tag{3.28}$$

3.3.3 Closed-form Transient Sensitivity Formula

The simulation technique as discussed in Section 3.2 can be applied to analyze the adjoint P/G network in backward time. The only trouble happens with the current source of $u(\tau - \tau_{sk}) - u(\tau - \tau_{ek})$ is applied to the most critical node for every event k. The Laplace transform of such a current source will make the right-hand side matrix of (3) or (4) a non-constant. Such a problem can be avoided by computing the response to the current source of $\delta(\tau)$ applied to the most critical node instead. If such a response is $\omega(s)$, then the response to $u(\tau-\tau_{sk})-u(\tau-\tau_{ek})$ is $v(s) = \omega(s)(e^{-s\tau_{sk}}-e^{-s\tau_{ek}})/s$.

The event-driven simulation of the adjoint P/G network is performed in a backward order of the specified events, so that the topology of the adjoint network is also changing in the reverse temporal order. The response of the tree with the most critical node is the superposition of $v(\tau) = L^{-1}(v(s))$, the voltage response propagated from the root (mesh node) and its zero-input response which is nonzero for all the backward events except for the very first one.

Given the nodal voltage $v(t)$ (for capacitors) and branch current $i(t)$ (for resistors or inductors) having the following form as indicated from Eq. (3.6) and (3.11):

$$f^k(t) = \sum_{i=1}^{P} r_i e^{\lambda_i t}, \quad 0 \le t \le T_k, \quad T_k = t_k - t_{k-1}, \tag{3.29}$$

the nodal voltage $\psi(\tau)$ (for capacitors) and branch current $\varphi(\tau)$ (for resistors or inductors) can be represented as

$$g^k(\tau) =$$

$$\begin{cases} \sum\limits_{j=1}^{Q} r_{z_j} e^{\Lambda_j \tau}, & 0 \le \tau \le T_k - t_{e_k} \\ \sum\limits_{j=1}^{Q} r_{z_j} e^{\Lambda_j \tau} + \sum\limits_{j=1}^{Q+1} R_j e^{\Lambda_j(\tau - T_k + t_{e_k})}, & T_k - t_{e_k} \le \tau \le T_k - t_{s_k} \\ \sum\limits_{j=1}^{Q} r_{z_j} e^{\Lambda_j \tau} + \sum\limits_{j=1}^{Q+1} R_j e^{\Lambda_j(\tau - T_k + t_{e_k})} - \sum\limits_{j=1}^{Q+1} R_j e^{\Lambda_j(\tau - T_k + t_{s_k})}, & T_k - t_{s_k} \le \tau \le T_k, \end{cases} \tag{3.30}$$

where $T_k = t_k - t_{k-1}$, Λ_j is the jth pole, r_{zj} is the jth residue of the zero-input response and R_j is the jth residue of the response to the current source of $u(\tau - \tau_{sk})$ - $u(\tau - \tau_{ek})$ at every switching event k.

Transient adjoint sensitivity calculation is performed using formula (27) or (29). As an example, the transient adjoint sensitivity of Z with respect to a capacitor C

can be computed as:

$$
\begin{aligned}
\frac{\partial Z}{\partial C} &= \sum_{k=1}^{f}\{-\int_0^{T_k} \sum_{i=1}^{P} r_i\lambda_i e^{\lambda_i t} \sum_{j=1}^{Q} r_{z_j} e^{\Lambda_j(T_k-t)}dt \\
&\quad - \int_0^{t_{ek}} \sum_{i=1}^{P} r_i\lambda_i e^{\lambda_i t} \sum_{j=1}^{Q+1} R_j e^{\Lambda_j(t_{ek}-t)}dt + \int_0^{t_{sk}} \sum_{i=1}^{P} r_i\lambda_i e^{\lambda_i t} \sum_{j=1}^{Q+1} R_j e^{\Lambda_j(t_{sk}-t)}dt\} \\
&= \begin{cases}
\begin{aligned} & -\sum_{i=1}^{P}\sum_{j=1}^{Q} r_i\lambda_i r_{z_j} e^{\Lambda_j T_k} T_k \\ & \quad - \sum_{i=1}^{P}\sum_{j=1}^{Q+1} r_i\lambda_i R_j e^{\Lambda_j t_{ek}} t_{ek} + \sum_{i=1}^{P}\sum_{j=1}^{Q+1} r_i\lambda_i R_j e^{\Lambda_j t_{sk}} t_{sk}, \end{aligned} & \lambda_i = \Lambda_j \\
\begin{aligned} & -\sum_{i=1}^{P}\sum_{j=1}^{Q} \frac{r_i\lambda_i r_{z_j}[e^{\lambda_i T_k} - e^{\Lambda_j T_k}]}{\lambda_i - \Lambda_j} \\ & \quad - \sum_{i=1}^{P}\sum_{j=1}^{Q+1} \frac{r_i\lambda_i R_j[e^{\lambda_i t_{ek}} - e^{\Lambda_j t_{ek}}]}{\lambda_i - \Lambda_j} + \sum_{i=1}^{P}\sum_{j=1}^{Q+1} \frac{r_i\lambda_i R_j[e^{\lambda_i t_{sk}} - e^{\Lambda_j t_{sk}}]}{\lambda_i - \Lambda_j}, \end{aligned} & \lambda_i \neq \Lambda_j
\end{cases}
\end{aligned}
\tag{3.31}
$$

3.4 Heuristic Optimization

The optimization technique used in this work is a sensitivity-based heuristic that is similar to the TILOS [6] algorithm, which is a greedy heuristic optimizer that changes the parameters that provide the "biggest bang for the buck." The basic philosophy of optimizing the P/G network using this technique is to try to reduce the maximum voltage droop violation in the network with the minimum increase in the area, by successively increasing parameter sizes by a small amount in each iteration.

The problem of optimizing a P/G network by varying wire widths can be formulated as

$$
\begin{aligned}
Minimize\ Area &= \sum_i l_i w_i \\
Subject\ to \quad Max(Z) &\leq 0
\end{aligned}
$$

In the objective function, l_i represents the total length of a set of the P/G wire segments with width w_i. In each iteration, we first analyze the network to identify the most critical node with the maximum voltage droop integral, then determine the parameter of the network that this critical node is most sensitive to, and finally bump up the width of this parameter by a certain small amount so that the most critical voltage droop integral is reduced.

In our method, we divide each wire in the mesh/trees into several π-segments, but model a set of adjacent wires as having the same width in order to reduce the network of optimization parameters. The gradients with respect to the area A_i of each set of N wires with width w_i is computed using the chain rule as follows:

$$\frac{\partial Z}{\partial A_i} = \frac{\partial Z}{\partial w_i} \times \frac{\partial w_i}{\partial A_i} = \frac{\partial Z}{\partial w_i} \times \frac{1}{\sum\limits_{j=1}^{N} l_j} \tag{3.32}$$

and

$$\frac{\partial Z}{\partial w_i} = \sum_{j=1}^{N} \left[\frac{\partial Z}{\partial C_{j1}} \frac{\partial C_{j1}}{\partial w_j} + \frac{\partial Z}{\partial C_{j2}} \frac{\partial C_{j2}}{\partial w_j} + \frac{\partial Z}{\partial R_j} \frac{\partial R_j}{\partial w_j} + \frac{\partial Z}{\partial L_j} \frac{\partial L_j}{\partial w_j} \right] \tag{3.33}$$

where the set of wires with width w_i consists of N wire segments; each of the segment j has resistance R_j, inductance L_j, and capacitance C_{j1}, C_{j2} at each terminal of the wire.

From Eq. (2.1), it is easy to see that

$$\frac{\partial R_j}{\partial w_j} = -\rho l_j / w_j^2 \tag{3.34}$$

$$\frac{\partial L_j}{\partial w_j} = -\gamma l_j / w_j^2 \tag{3.35}$$

$$\frac{\partial C_{j1}}{\partial w_j} = \frac{\partial C_{j2}}{\partial w_j} = \frac{\beta l_j}{2} \tag{3.36}$$

The overall optimization procedure is as follows:

- Simulate the original P/G network over the entire period using the hierarchical simulation method discussed in Section 3.2.

- Determine the most critical node with its Z_{max}.

- Save voltage approximants (poles and residues) for all C's and current approximants for all R's and L's in the network.

- Simulate the adjoint network backward in time (event) with zero initial conditions.

- Save voltage/current waveforms for the adjoint network.

- Compute the voltage sensitivities with respect to all R's, L's and C's.

- Compute the voltage sensitivities with respect to A_i using Eq. (3.32), (3.33), (3.34), (3.35) and (3.36).

- Bump up the width of the set of wires with maximum sensitivity by multiplying it with a small factor (< 1.1).

- Repeat the above procedure until the maximum voltage droop integral in the network is below zero.

The above procedure can be extended to include the optimization of decoupling capacitors. The objective of this optimization is to determine appropriate sizes of each wire and each decoupling capacitor for the minimum area overhead. Initially, decoupling capacitors with some small values are connected to some user-specified nodes in the P/G network. The gradients of the most critical node with respect to these decoupling capacitors are exactly the transient adjoint sensitivities calculated in each iteration. The cost function for the optimization is a weighted sum of the wire area and the areas of all the decoupling capacitors. In each iteration, either the wire width or the decoupling capacitor with the maximum sensitivity with respect to the objective function will be increased with a small factor until the constraints are met.

3.5 Experimental Results

The simulation and optimization procedure was implemented in C, and the results on several P/G networks were tested. The networks were constructed randomly for power delivery to a 2cm x 2cm chip in a $0.18\mu m$ technology with $V_{dd} = 1.65V$. The set of events is randomly generated and is different for each circuit. The results shown here can be considered to correspond to a top level P/G distribution network, since complete P/G networks may have several millions of nodes. The utilization points here would correspond to functional blocks, each of which is reduced to an equivalent RC representation.

We have used the commercial simulator, HSPICE, to analyze the speed and accuracy of our simulation results. All experiments are performed on Sun Ultra-60 Workstations.

The waveforms for two networks are shown in Figure 3.4, with the waveforms using HSPICE plotted concurrently on the same figures using dotted lines. In each case, our waveform and that of HSPICE are quite close. The order of approximation is chosen such as the integral of noisy area is within 10% to that of HSPICE.

The comparison of the run-time for the two cases and the speedup are shown in Table 3.1. It can be seen that our simulation runs significantly faster than HSPICE.

Table 3.2 lists the results of optimization and the run-time for five different P/G networks with and without decaps. Two rows are listed for each circuit, with the first row showing wire sizing results only and the second row showing both wire sizing and decap optimization. The total number of nodes ("total") and the number of user-specified critical nodes ("crt") are listed for each circuit. The result for the specific voltage constraint, listed in the "spec" column, is shown for each circuit, along with the total wire area. The "Z_{max}" and the "Init V_m" column refer to the worst-case noise integral and the worst-case voltage level when all wires are unsized. The "Opt

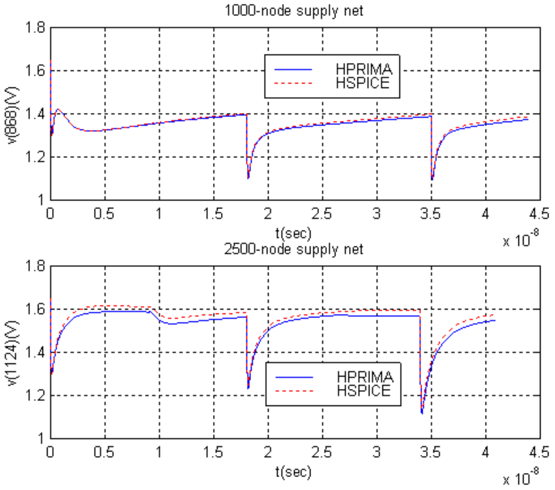

Figure 3.4: Simulation results on a 1000-node and a 2500-node supply network. The reduced orders for the two networks are 13 and 15, respectively. HPRIMA stands for our Hierarchical PRIMA simulator. $V_{dd} = 1.65V$.

V_m" column shows the voltage droop after the specifications are met or when Z_{max} is optimized to zero. The CPU times and the number of iterations of the heuristic optimizer are shown in the last two columns.

Table 3.3 shows the comparison between two networks with different topologies. Circuit 1 deviates from the one-level hierarchical scheme shown in Figure 3.1, and is a two-level hierarchy in which the top level is a 9-node mesh with a tree of 112 nodes originating from each mesh node; we will refer to such a structure as a "9x112 structure". Some of the tree nodes of this upper level are connected to separate 9x112 structures. Specifically, in the structure here, we have a total of nine such 9x112 bottom level structures. Pads are assigned to each of these bottom level networks. The optimization is performed hierarchically. The bottom-level net is first optimized to within 7% of Vdd with a voltage source of 97%Vdd applied to the connecting node to the top-level network. The top-level net is then optimized to within 3% of Vdd with the reduced order model connected to the top-level network. As a result, the

	# of nodes		T_{HPRIMA}	T_{HSPICE}	Speed
Ckt	Mesh/Tree	Crt	(s)	(s)	Up
1	9/1008	10	1.68	82.42	49.06
2	25/2500	25	4.48	232.09	51.81
3	25/3000	32	8.52	325.42	38.20
4	25/4000	38	10.02	499.53	49.85
5	25/5000	38	11.20	680.15	60.73
6	49/10800	78	21.49	1641.02	76.36

Table 3.1: Runtime comparisons with HSPICE.

two-level hierarchical network has a worst-case voltage droop of 10%Vdd. Optimizing a network to within 3% of Vdd normally takes more CPU time than optimizing it to 7%. Since we have only 1 top-level network and 9 bottom-level networks in the 2-level hierarchical structure, intuitively the selection of the constraint of 3% for the 1 top-level network and 7% for the 9 bottom-level networks should lead to some smaller total amount of CPU time. For comparison, a one-level 90x112 network (circuit 2) is constructed and optimized. The optimization results show that the two-level hierarchy can be performed far more quickly than the one-level network with similar wire areas and performance.

3.6 Conclusion

An efficient transient sensitivity computation method for P/G network design and optimization is presented. A fast and efficient event-driven P/G network simulator is developed. Experimental results show that the simulation is accurate and fast. The optimization procedure involves a procedure for fast calculation of adjoint sensitivities in a heuristic optimization loop. This procedure is illustrated on a specific family of

Ckt	# of nodes		Spec	Z_{max} $(V\times$ $ns)$	Init V_m (V)	Opt V_m (V)	Wire Area (cm^2)	Max Dcp (nF)	Num of Dcp	CPU time (hrs)	Num of Itr
	Total	Crt									
1	1017	10	0.165	7.50	0.698	0.161	0.083	-	-	0.35	109
1	1017	10	0.165	1.13	0.718	0.165	0.045	0.0379	9	0.41	142
2	2016	19	1.485	8.34	1.080	1.488	0.172	-	-	0.92	125
2	2016	19	1.485	2.39	1.174	1.485	0.125	0.3450	13	0.78	117
3	3025	32	0.165	6.60	0.760	0.165	0.347	-	-	3.64	276
3	3025	32	0.165	1.46	0.544	0.165	0.225	1.7700	20	2.42	186
4	5025	38	1.485	4.81	1.058	1.485	0.651	-	-	4.60	231
4	5025	38	1.485	0.59	1.058	1.485	0.476	0.1460	24	3.51	195
5	9849	78	1.485	0.81	1.224	1.485	0.119	-	-	8.60	256
5	9849	78	1.485	0.42	1.224	1.485	0.109	1.3310	30	7.58	227

Table 3.2: Optimization results.

topologies described in Figure 3.1, with an example of two-level hierarchy of such a topology. It can also be extended to other mesh topologies that have an overall tree-like structure, e.g., a tree-like macro structure in which each vertex is a mesh.

Ckt	# level	# of nodes		Spec	Z_{max} $(V\times$ $ns)$	Init V_m (V)	Opt V_m (V)	Wire Area (cm^2)	Max Dcp (nF)	Num of Dcp	CPU time (hrs)
		Mesh/Tree	Crt								
1	2	90/10080	100	1.485	12.07	1.260	1.486	1.323	1.1918	30	3.74
2	1	90/10080	100	1.485	16.72	1.120	1.485	1.308	1.2100	30	33.51

Table 3.3: Topology comparison

Chapter 4

Optimal Decoupling Capacitor Sizing and Placement for Standard Cell Layouts

4.1 Introduction and Motivation

4.1.1 Motivation

Noise margins have been greatly reduced in modern designs due to the lowering of supply voltages and the presence of a larger number of potential noise generators that eat significantly into the noise margins built into a design. A powerful technique for overcoming this problem is through the use of on-chip decoupling capacitors (decaps) that are intentionally attached to the power grid. To exemplify the role of decaps, let us consider the circuit shown in Figure 4.1, which can be thought of as a canonical model of a power grid and loading circuit. In the figure, G_g models the grid conductance, G_d and C_d model a decoupling capacitance, and I_{load} models the

Year	L_{eff} nm	f MHz	V_{dd} V	Size mm^2	Power W	Density W/mm^2
2001	65	1684	1.1	310	130	*0.42*
2002	53	2317	1.0	310	140	*0.45*
2003	45	3088	1.0	310	150	*0.48*
2004	37	3990	1.0	310	160	*0.52*
2005	32	5173	0.9	310	170	*0.55*
2006	28	5631	0.9	310	180	*0.58*
2007	25	6739	0.7	310	190	*0.61*

Table 4.1: IC technology parameters.

time-dependent current waveform of the load, which we model for simplicity as:

$$I_{load} = \begin{cases} 0 & : \quad t < 0 \\ \mu t & : \quad t < t_p \\ \mu(2t_p - t) & : \quad t < 2t_p \\ 0 & : \quad t > 2t_p \end{cases} \tag{4.1}$$

We will use data from the International Technology Roadmap for Semiconductors [2], summarized in Table 4.1, to predict the dependence of the load voltage V_{load} on the various circuit parameters in order to predict trends in power-grid-induced noise with technology scaling. The table shows the projected yearly trends for the effective length L_{eff}, of a transistor, the circuit frequency, f, the supply voltage level, V_{dd}, the chip size, the power dissipation and the density of power dissipation per unit area.

For the circuit shown in Figure 4.1, we observe that V_{load} normalized by the voltage supply V_{dd} over the time interval from $t = 0$ to $t = t_p$ can be expressed as:

$$V_{load} = 1 - \frac{\mu}{G_g}\left(t - \frac{C_d}{G_g}(1 - e^{-t/\tau})\right) \tag{4.2}$$

where

$$\tau = \frac{(G_g + G_d)C_d}{G_g G_d} \tag{4.3}$$

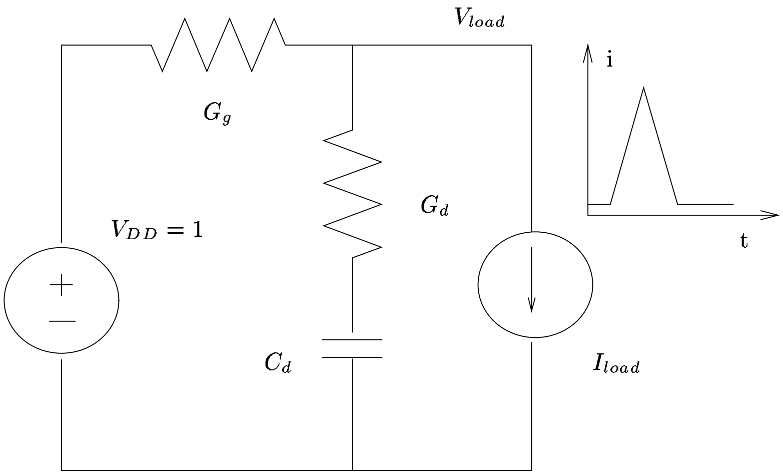

Figure 4.1: A canonical and approximate circuit representation of a power network.

The minimum V_{load}, or maximum normalized power-supply-induced noise occurs at $t = t_p$ and the magnitude of the noise is:

$$V_{max} = \frac{\mu}{G_g}\left(t_p - \frac{C_d}{G_g}(1 - e^{-t_p/\tau})\right) \tag{4.4}$$

We note that $t_p \propto f^{-1}$, and that power density $P_\Box \propto V_{dd}\mu t_p$, implying that $\mu \propto P_\Box f/V_{dd}$. Based on the trends in Table 4.1, f increases by 3.0X through the table, and μ increases at 9.13X. In order to keep V_{max} the same (i.e., keep the same amount of noise as a percentage of V_{dd}), we need to dramatically increase the last term in Eq. (4.4): $\frac{C_d}{G_g}(1 - e^{-t_p/\tau})$. This means:

- Increasing the decoupling capacitance C_d, which can be done at the cost of small additional area, because the area efficiency of decoupling capacitance is expected to increase as the gate oxide is scaled.

- Increasing the conductance associated with the decoupling capacitance G_d, which can be done by placing the capacitance *closer* to the load.

- Increasing the grid conductance G_g, which will be the most difficult to do because it goes somewhat against the prevailing scaling of interconnect, and the increased restrictions due to the consequent wire congestion emanating from this.

Unless we are able to do all of the above, it is likely that we will find the relative magnitude of power-grid-induced noise more than doubling by 2007.

The first two of these conclusions point a convincing finger towards the use of *appropriately placed* decaps for power grid noise reduction. While the use of decaps is certainly not new[1], the complexity of the problem requires shrewd optimal strategies driven by CAD tools, particularly in standard-cell environments in designs that require quick turn-around times in the face of strong time-to-market pressures.

4.1.2 Previous Work

Previous work [14, 32, 40] on decap allocation and optimization has focused on application in full custom design styles. A decap optimization procedure involving an iterative process of circuit simulation and floor planning is proposed in [14]. A linear programming technique is applied in [40] for allocation of white space for decap use and a heuristic is proposed to insert additional white space into an existing floorplan. Both [32] and our work presented in Chapter 3 ([8]) propose a sensitivity-based method of placing or optimizing decaps for reducing the voltage droop in the power distribution network; the former method handles the problem in the frequency domain, the latter in the time domain.

[1] For example, in a 300MHz CMOS RISC Microprocessor design [38], as much as 160nF of on-chip decoupling capacitance is added to control power-supply noise. In another example [39], the on-chip decoupling capacitance is sized at ten times that of the total active circuit switching capacitance.

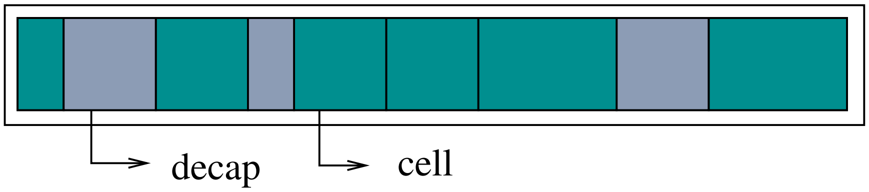

Figure 4.2: One row of cells in a standard cell layout showing decap locations.

4.1.3 Contributions

In this work, we investigate the decap optimization and placement issue in the context of row-based standard-cell design typical of Application Specific Integrated Circuits (ASIC) where each row has a fixed height. We consider a chip composed of N rows, with the ith row having M_i cells (blocks). Each of the N rows is filled by cells to some level of ratio $r_i(\leq 100\%)$. Decoupling capacitors can be placed in the empty space, which forms the $(1 - r_i)$ fraction of each row. One such row is illustrated in Figure 4.2.

Our approach is designed to be applicable subsequent to the placement phase for the design, where cells have already been assigned to rows. Since placement is designed to optimally place cells in order to achieve compactness for the layout and to control the wire length, timing and congestion, we use that result as the starting point for decap optimization, and perturb that solution in a minimal way in solving the *decap placement* problem.

Specifically, we propose to use the empty spaces that may be available within each row (when $r_i < 1$) to place decaps. In doing so, the exact position of each cell in that row is considered to be flexible although the order and the *relative* positions are fixed. Different placement of cells can lead to different widths and locations of decaps, and consequently different impacts on the power supply noise, and the problem that we wish to tackle is that of finding the optimal cell placement which results in the minimization of a metric for the power supply noise. Note that since typical values of r_i are close to 1, the major attributes of the original cell placement will be, for the most part, unaffected by our procedure.

4.2 Power Supply Noise and Sensitivity Analysis

4.2.1 Noise Analysis

For the ASIC row-based standard-cell design style outlined above, it is common to use a predefined mesh-like power distribution network. As described in Chapter 2, the linear circuit model [14, 15]is used in this work and the behavior of such a circuit is described by the MNA [16] equation:

$$Gx(t) + C\dot{x}(t) = u(t) \tag{4.5}$$

where x is a vector of node voltages and source and inductor currents; G is the conductance matrix; C includes both the decoupling capacitance and package inductance terms, and $u(t)$ includes the loads (piecewise linear current sources) and voltage sources.

Again, as shown in Chapter 2, by applying the Backward Euler integration formula [16] to Eq. (4.5), we have:

$$(G + C/h)x(t + h) = u(t + h) + x(t)C/h \tag{4.6}$$

Where h is the time step for the transient analysis. If h is kept constant, only a single initial factorization of the matrix $G + C/h$ is required (as is done in [18, 17]) leading to an efficient algorithm for transient analysis where each time step requires only a forward/backward solution step. After the transient analysis of the circuit, the voltage waveform at every node is known. Given that the treatment for nodes on the ground grid is completely symmetric, we restrict our discussion to the V_{dd} nodes for which we define the *droop* at node n to be simply $V_{dd} - V_n(t)$, where $V_n(.)$ signifies the voltage at node n.

The integral of the voltage droop below a user specified noise ceiling is again used as the noise metric and in our case, the tunable circuit parameters p in Eq. (2.5) are the *widths* of the decoupling capacitors[2].

We define the measure of goodness for the whole circuit as the sum of the individual node metrics:

$$Z = \sum_{j=1}^{K} z_j(p), \tag{4.7}$$

where K is the number of nodes. This metric penalizes more harshly transients that exceed the imposed noise ceiling by a large amount for a long time, and has empirically been seen to be more effective in practice than one that penalizes merely the maximum noise violation. Intuitively, this can be explained by the fact the the metric incorporates, in a sense, both the voltage and time axes together, as well as spatial considerations through the summation over all nodes in the circuit.

4.2.2 Integral Sensitivity Computation

4.2.2.1 General Adjoint Sensitivity Analysis

For this work we are interested in the sensitivity of the scalar objective function (Eq. (4.7)) with respect to the widths of all decaps in the network.

An adjoint network with the same topology as the original network is constructed, with all the voltage sources in the original network shorted and current sources open. For noise functions of the form given in Eq. (2.5), the adjoint network will include a current source of value $-u(t - t_s) + u(t - t_e)$ applied at node j if $z_j \neq 0$. We

[2] We choose the width since the height of the decoupling capacitors is constrained to be the same as the height of the functional cells in the same row, as illustrated in Figure 4.2.

set the initial conditions to the adjoint circuit to zero and analyze it backward in time. We use the same time step h as the original circuit, thus allowing us to reuse the previously computed LU factorization for $(G + C/h)^{-1}$. Consequently, the extra simulation cost is reduced to one forward/backward solve for each time step of the adjoint circuit. Obviously, a smaller time step results in a higher accuracy for both the original and adjoint waveforms, and consequently higher accuracy in the sensitivities at the expense of a longer runtime. We find that in order to insure the accuracy of adjoint sensitivities, using 500-1000 steps per clock cycle (i.e., $h = 0.002T_{period}$ or $0.001T_{period}$) is sufficient.

The sensitivity of the objective function with respect to all of the decoupling capacitors in the circuit can be computed from the following convolution [23, 24]:

$$\frac{\partial Z}{\partial C} = \int_0^T \psi_C(T-t)\dot{v}_C(t)dt, \tag{4.8}$$

where $\psi_C(\tau)$ is the waveform across the capacitor C in the adjoint circuit.

4.2.2.2 Waveform Compression and Fast Convolution

In our context, we cannot use this approach directly, and must tailor it to control the storage required by the direct application of this method. Specifically, a significant complication arises in the case of very large networks where the total amount of data to be stored is proportional to the number of nodes multiplied by the number of time steps, and can reach 10^9 bytes or more for networks with millions of nodes. In order to alleviate the problem, we store the waveforms of the original and adjoint network using a compressed piecewise linear form. This results in a situation of the type illustrated in Figure 4.3, where the time points on the original and adjoint waveforms are not aligned. However, since we know that waveforms are divided by linear segments, the convolution (Eq. (4.8)) of the waveforms $\psi_C(\tau) = g + k\tau$ and

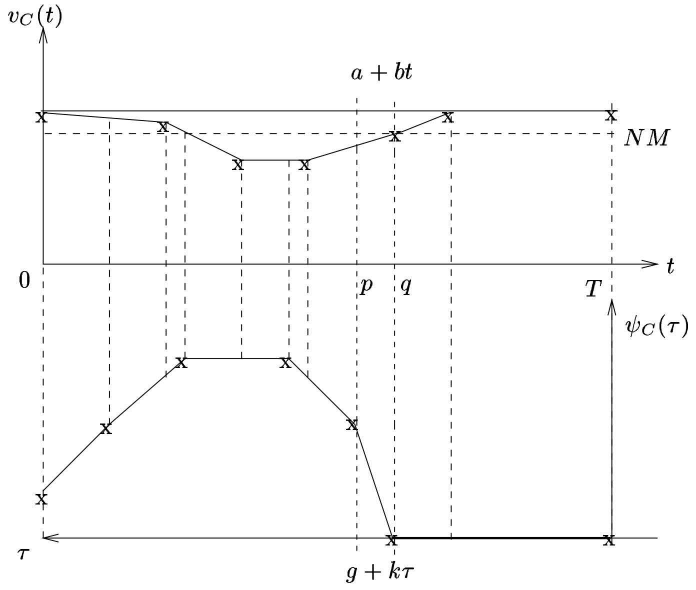

Figure 4.3: Compressed piecewise-linear waveforms for the original and adjoint networks.

$v_C(t) = a + bt$ over the time interval $[p, q]$ can be expressed as:

$$\begin{aligned}
&\int_p^q (g + k(T - t)) \; \frac{d(a + bt)}{dt} \; dt \\
= \; &\int_p^q (g + k(T - t)) \; b \; dt \\
= \; &b(q - p) \left(g - kT - k \left[\frac{q - p}{2} \right] \right) \qquad (4.9)
\end{aligned}$$

The complexity of the convolution calculation over $[0, T]$ is $O(N + M)$, where N and M are the number of linear segments on the original and adjoint waveforms.

4.2.2.3 Sensitivity with Respect to Decap Width

Once the sensitivities of Z with respect to all of the decoupling capacitor values are computed, the sensitivities to the *width* of each capacitor can be calculated using the chain rule, as in [8]:

$$\frac{\partial Z}{\partial w} = \frac{\partial Z}{\partial C} \times \frac{\partial C}{\partial w} \tag{4.10}$$

Given that we calculate the decoupling capacitance from:

$$C = \frac{\varepsilon_{ox}}{T_{ox}} \times w \times h, \tag{4.11}$$

where T_{ox} and ε_{ox} are the thickness and permittivity of the gate oxide, and h is the fixed height of the decap, it is easily verified that Eq. (4.10) becomes:

$$\frac{\partial Z}{\partial w} = \frac{\partial Z}{\partial C} \times \frac{\varepsilon_{ox}}{T_{ox}} \times h \tag{4.12}$$

4.3 Optimization and Placement

4.3.1 Problem Formulation

The problem of decoupling capacitor optimization is now formulated as:

$$\begin{array}{rcl} \textbf{Minimize} & Z(w_j) & j = 1 \cdots N_{decap} \\ \textbf{Subject to} & \sum_{k \in row_i} w_k \leq (1 - r_i)W_{chip} & i = 1 \cdots N_{row} \\ \textbf{and} & 0 \leq w_j \leq w_{max} & j = 1 \cdots N_{decap} \end{array}$$

The scalar objective Z, defined in Eq. (4.7), is a function of all the decap widths and N_{decap} is the total number of decaps in the chip. The first constraint states that the total decap width in a row cannot exceed the total amount of empty space in that row, and W_{chip} and N_{row} denote, respectively, the width of the chip and the number of rows in the chip. The second constraint restricts the decap widths within a realistic range. An upper bound w_{max} for a cell in row i is easily seen to be $(1 - r_i)W_{chip}$, which is the largest empty space in row i; while the lower bound of each decap width is zero.

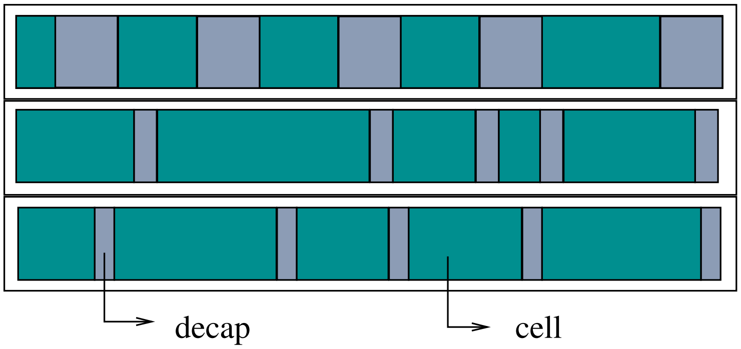

Figure 4.4: Illustration of the initial equal distribution of decaps.

Eq. (4.13) represents a linearly constrained nonlinear optimization problem. The objective function Z can be obtained after the transient analysis of the power grid circuit, and its sensitivity with respect to all the variables w_j can be calculated using the adjoint method discussed in Section 4.2.2. We choose to use a standard quadratic programming (QP) solver [41] for solving large nonlinear optimization problems. We start the optimization with an initial guess that uniformly distributes the vacant space in each row to each decoupling capacitor in each row, as illustrated in Figure 4.4. It can be seen that initially there is one decap next to each cell. The initial chip width is chosen to be the maximum width occupied by cells and decaps among all rows.

4.3.2 Optimization and Placement Scheme

The optimization procedure invokes the QP optimizer, and the set of steps that are repeated during each iteration of the optimizer can be summarized as follows:

- Perform the transient simulation of the original power grid circuit and store piecewise linear waveforms of all decaps.

- Check all nodal voltages for those that fall below the noise margin, identify hot spots and compute the objective function Z.

- Set up the sources corresponding to these failure nodes for the adjoint circuit.

- Perform the transient simulation of the adjoint circuit and store piecewise linear waveforms of all decaps.

- Compute the sensitivities $\frac{\partial Z}{\partial C_j}$ by convolution and use the chain rule to obtain $\frac{\partial Z}{\partial w_j}$.

- Compute the constraint function and its Jacobian.

- Feed all the information into a QP solver and update the vector of widths, $\vec{w}$, according to the values returned by the solver.

- According to the updated $\vec{w}$, reposition all of the cells and decaps in the row from left to right.

4.4 Experimental results

The proposed decap optimization and placement scheme has been integrated into a linear circuit simulator written in C++ and the QP solver is applied. All experimental results are performed on a 1.8GHz Pentium IV machine under the Redhat Linux operating system.

Table 4.2 lists the decap optimization results for three industrial ASIC designs, which are referred to as Chip1, Chip2 and Chip3. Each of them is a $0.18\mu m$ CMOS design operating under a supply voltage of $1.8V$. The occupancy ratio r_i for each row of these chips is around 80%. In Table 4.2, the second column shows the number of nodes with noise violations (i.e., nodes j with a nonzero value of z_j) before and after optimization; the total number of nodes in the power grid are shown in the third column. The next two columns compare the worst-case voltage droop and the sum of integral area Z (i.e., the objective function) before and after optimization. Column 6

		Num bad nds	Num of nds	V_m (V)	Z $(V\times ns)$	Num of rows	Num of dcps	CPU time (min)
Chip1	B	105	974	0.193	0.121	53	1964	0.9
	A	0		0.176	0.000			
Chip2	B	80	861	0.230	0.366	85	3288	15.2
	A	63		0.196	0.063			
Chip3	B	100	828	0.222	0.649	132	3664	12.5
	A	70		0.201	0.200			

B = Before optimization; A = After optimization

Table 4.2: Optimization results

shows the total number of rows in the chip. The total number of decoupling capacitors placed in the whole chip is listed in column 7. Finally, the last column lists the total amount of CPU time to run each example. For each of these three chips, the worst case voltage droops and sums of the integral area are both reduced successfully.

It should be noted that decoupling capacitor is not the only method for noise reduction, and that other techniques such as wire widening, or increasing the density of the power grid, can be applied to further improve the power grid performance. Therefore, these results that holistically reduce the degree of noise violation by decap placement can be complemented with other techniques to obtain a solution that satisfies the noise constraints imposed on the design.

The V_{dd} and ground contour of chip2 is shown in Figure 4.5 and Figure 4.6. In both figures, each gray-scaled color corresponds a voltage droop range and the number written in each color sample shows the lowest voltage droop in that range. Darker colors mean larger voltage droops. It can be seen that the voltage range in the V_{dd} plane is $1.610-1.8V$ and the hot spot is located on the right side of the chip. Similarly,

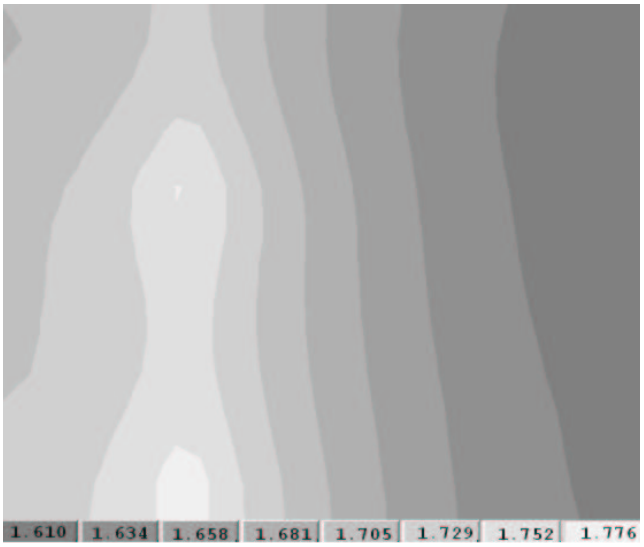

Figure 4.5: The original voltage droop contour of the Vdd plane.

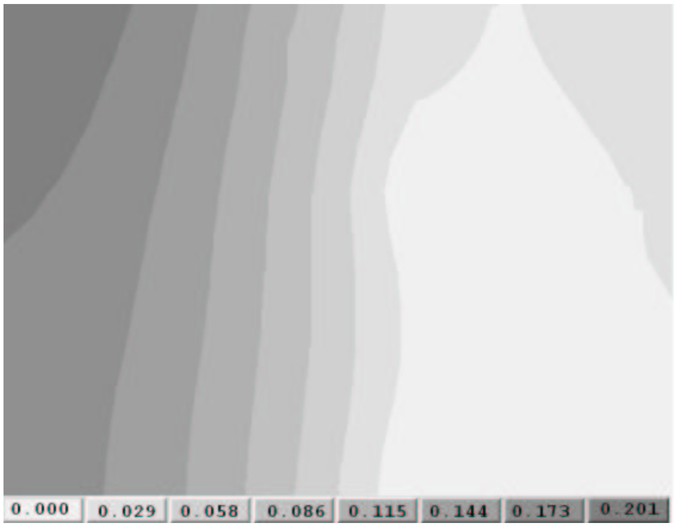

Figure 4.6: The original voltage droop contour of the ground plane.

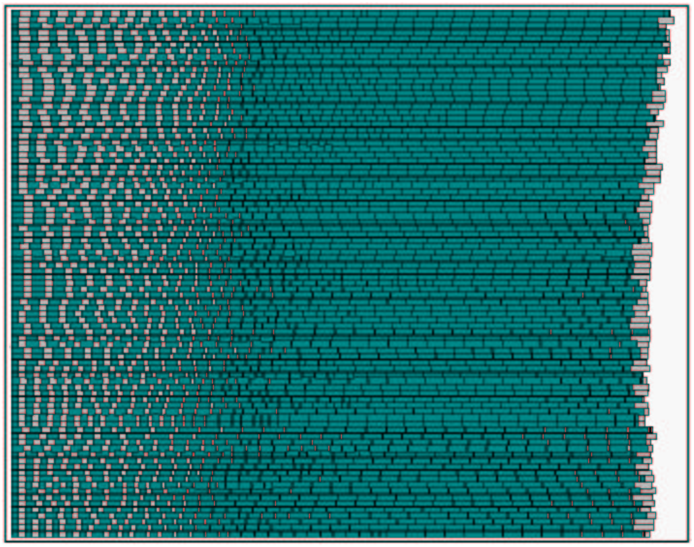

Figure 4.7: Results of the decap placement algorithm on chip2. The dark regions represent the standard cells, and the light regions are the decaps.

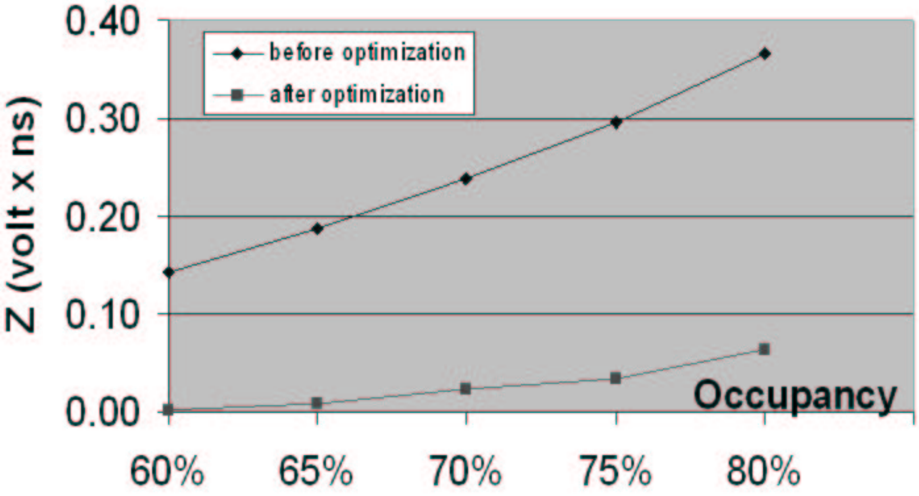

Figure 4.8: Variation of the noise metric with the occupancy ratio.

the voltage range in the ground plane is $0 - 0.230V$, and the hot spot is located on the left side of the chip. The result of the optimal cell and decap placement for chip2 is shown in Figure 4.7. We observe that this placement is consistent with the hot spots of the chip, i.e., larger decaps are allocated closer to the two sides of the chip. After optimization, the voltage droop in the V_{dd} plane is in the range of $0 - 0.196V$ and that of the ground plane is in the range of $0 - 0.191V$. The optimization process has judiciously balanced the power grid voltage droop on the whole chip.

The noise reduction trend with respect to the cell occupancy ratio r_i for chip2 is shown graphically in Figure 4.8. This experiment is performed by removing some cells from each row of the chip to achieve the desired occupancy ratio. For each case, around 10 percent of the total grid nodes are beyond the noise margin. A chip with lower occupancy ratios provides more empty space for decoupling capacitors and consequently is easier to optimize. Therefore, in Figure 4.8, the noise reduction is more efficient for cases with lower occupancy ratios than for those with higher ones.

4.5 Conclusion

This chapter has presented an on-chip decoupling capacitor sizing and placement scheme aimed at making the best use of empty spaces in the row-based standard-cell design of ASICs. The problem of decap insertion and placement has been motivated for current and future technologies, and the problem has been formulated as a constrained non-linear optimization problem that is successfully solved using the gradient-based QP solver. For a pre-designed power distribution network, the location and size of each decap is updated iteratively such that the total transient noise in the power grid is minimized, and the technique is demonstrated on several industrial designs.

Chapter 5

Hybrid Clock Distribution Network Construction

5.1 Introduction

5.1.1 Clock Design Objective

The most common objectives of clock network design are to ensure near-zero skew, sharp edges and the optimal use of routing resources. In addition, it is also important to reduce the total propagation delay through the network. This is because any modeling error or uncertainty in the clock signal arrival times between clock sinks (pins), which can cause performance or functional problems, is likely to be larger if the delay from the clock source to the clock pins is larger. A reduction in the overall delay through the clock network serves to reduce clock distribution uncertainties induced by modeling errors, process variations, and noise.

5.1.2 Previous Work

Traditionally, clock networks have been designed as tree structures since these are considered to be easier to optimize, and various algorithms [42–45] for constructing zero-skew clock trees under the Elmore delay metric have been explored in the past. Most of these methods recursively construct binary tree-like structures with the clock pins at the leaf nodes. For such structures, several delay/skew minimization techniques [46,47] and the wire width optimization techniques [48,49] have been proposed. As compared to tree structures, clock meshes are known to potentially yield better performance, but are seldom used at upper levels of the clock hierarchy since the reliable optimization of meshes is considered to be a difficult problem. Clock meshes have been used before, for example, in [50] and [51]. While in [50] the sizing of a non-tree topology is formulated as a sequence of network flow problems, and in [51] meshes are used near the leaf nodes of the network.

The benefits of mesh-like structures are illustrated in an interesting comparison among three commercial microprocessor clock network topologies, namely, grids (i.e., meshes), trees and serpentines, in [4] and [51], and we summarize their observations here. For uniformly distributed loads, the tree topology provides low skew and much lower capacitance than the other topologies. However, one major advantage of the grid topology is that changes in clock loads, locations, or electrical models only cause little change in the clock timing and rarely require re-tuning of the grid wires or drivers.

5.1.3 Contributions

In this chapter, we develop a novel technique that describes how some classes of mesh can be optimized very easily for zero skew using a simple linear-programming based formulation. An important property of our algorithm is that, like Tsay's method

[45], it works in a bottom-up fashion in constructing the clock network, solving a computationally inexpensive subproblem at each intermediate step. An additional feature of this work is that it can be extended very easily to build clock trees with prespecified nonzero skews for cases when deliberate skews are used in the clock network for cycle borrowing [7].

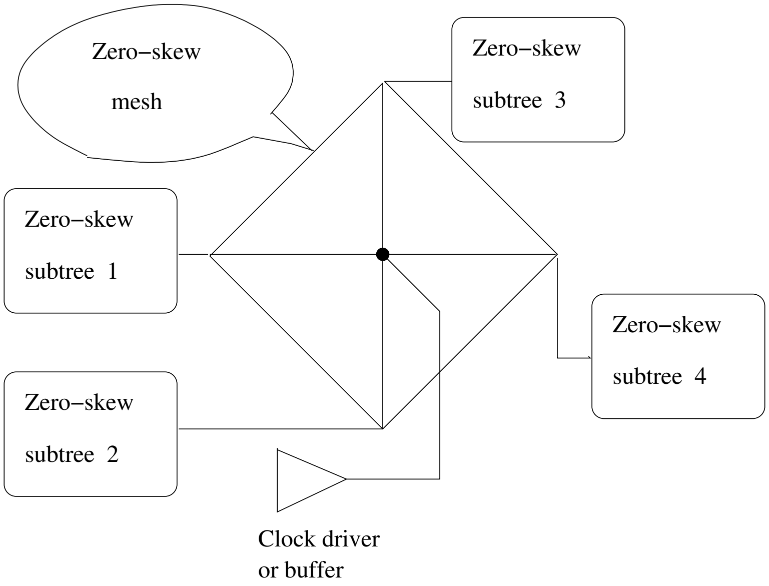

Figure 5.1: A clock network as a mesh/tree structure.

To blend the advantages of meshes and trees, we propose a hybrid mesh/tree structure that is guaranteed to have zero skew under the Elmore delay metric, and further optimize this structure to a target transition time and small non-zero skew under a higher order delay model. One example of our structured topology is a "one-level mesh" consisting of a global mesh feeding local trees, as shown in Figure 5.1; the same idea can be extended to multiple levels of hierarchy, as explained later in this chapter. The rationale for this hybrid approach is that our experiments have indicated that the use of full meshes can use excessive routing resources without delivering significant gains, and a judicious blend of tree and mesh topologies can provide the best of both worlds[1]. This work has been inspired by a similar idea of using a topology that is intermediate to the two extremes of full trees and full meshes in our work described in Chapter 3 [34, 8]

In designing clock trees, "zero skew" is something of a misnomer [49], since routing constraints and process variations make non-zero skew inevitable, so that a more realistic objective is to construct reliable low-skew clock networks. While the first-order Elmore Delay is used here to build the initial zero-skew mesh/tree structure, for accurate analysis of delays and skews and further optimization of the constructed clock network to some target transition time and skews, a higher order delay simulator is necessary. In [49], wire width optimization has been applied to the skew reduction of clock trees, but their procedure focuses mainly on the first-order Elmore delay. For our constructed mesh/tree structure, a hierarchical simulator (HPRIMA) similar to [8] is used to find the propagation delays and transition times at the clock pins. A reduced order modeling approach based on HPRIMA is also employed in calculating the adjoint sensitivities of delays, as well as of skew and transition time with respect to all the wire widths to guide our TILOS-like [6] heuristic wire width optimization for meeting more stringent skew and transition time constraints.

[1] The intuition behind the specific hybrid mesh/tree structure that we employ, with meshes near the top of the hierarchy and trees at the bottom, is explained as follows. It is well known that when wire sizing is used for delay reduction, the optimal solution sizes wires near the source to a greater degree than wires close to the sinks. Using a similar philosophy, we use mesh structures close to the root of the clock tree (as a way of reducing the effective resistance between the root and points on the mesh), and use tree structures near the sinks. As compared to sized tree structures, meshes are more efficient in reducing the effective resistance to a larger number of nodes, with less routing bottlenecks.

5.2 Linear-programming Based Zero-skew Mesh Construction

5.2.1 Mathematical Derivation

Our proposed clock network has a structure of overlying mesh with local trees rooted on the mesh, as shown in Figure 5.1. Given a set of sinks, the construction is in a two-step bottom-up manner:

(i) Construct the *underlying zero-skew trees.*

(ii) Construct the *overlying zero-skew mesh.*

We will not focus on step (i), since the zero-skew trees can be constructed using any of the conventional clock routing algorithms. Once these trees have been built, step (ii) requires the total delay and downstream capacitance for each such tree.

For a one-level mesh/tree structure, we divide the sinks into four groups and build four trees, one for each group. Multi-level meshes can be built in a similar bottom-up way as the construction of the trees: at each level, each node in the i^{th}level mesh is connected to a tree whose sink nodes are $(i\text{-}1)^{th}$ or lower level meshes

We will now illustrate the construction of a one-level zero-skew mesh shown in Figure 5.2, consisting of the set of segments $\mathbf{S} = \{ab,\ bc,\ cd,\ ad,\ as,\ bs,\ cs,\ ds\}$. We assume that the four underlying trees have been built and that they are rooted at nodes a, b, c and d.Tree k has a delay of Δk and a downstream capacitor C_{Dk}, where $k = a$, b, c and d. In our implementation, the $(x$,$y)$ coordinates of the tapping point s of the mesh are chosen as the mean of corresponding coordinates of the four roots, but the succeeding derivation does not depend on this assumption.

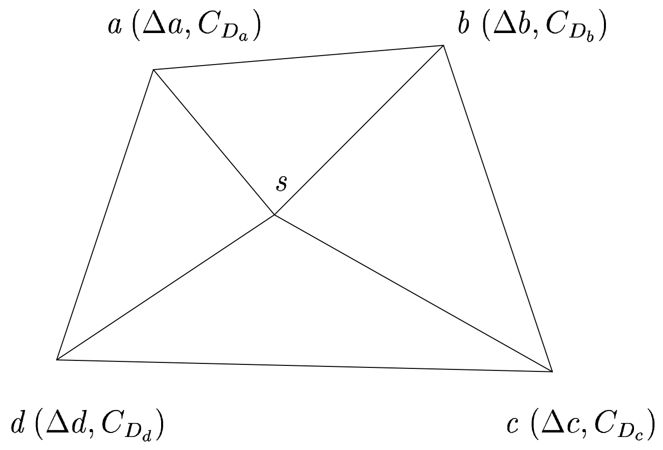

Figure 5.2: One-level mesh with underlying trees.

As described in Chapter 2, each wire on the mesh is modeled as a segment under the π-model, with each segment modeled using lumped RC parameters[2]. The admittance matrix [52] of the circuit is

$$G = \begin{bmatrix} g_{ab} + g_{ad} + g_{as} & -g_{ab} & 0 & -g_{ad} \\ -g_{ab} & g_{ab} + g_{bc} + g_{bs} & -g_{bc} & 0 \\ 0 & -g_{bc} & g_{bc} + g_{cd} + g_{cs} & -g_{cd} \\ -g_{ad} & 0 & -g_{cd} & g_{ad} + g_{cd} + g_{ds} \end{bmatrix} \tag{5.1}$$

and the capacitance matrix is

$$C = \begin{bmatrix} C_a + C_{Da} \\ C_b + C_{Db} \\ C_c + C_{Dc} \\ C_d + C_{Dd} \end{bmatrix} \tag{5.2}$$

[2] While the use of lumped elements is an approximation, our experimental results will show that it does not seriously affect the skew of the final network. Moreover, as a post-processing step, we can further optimize the network using a larger number of wire segments to model long wires, and a higher order model for delay calculation. Such a procedure will be described in Section 5.3.

where C_k is the total wire capacitance at node k, $k = a$, b, c and d.

The first-order (Elmore) delays from node s down to all the four roots are calculated by

$$\vec{\tau} = G^{-1}C \tag{5.3}$$

To achieve zero-skew, we require

$$\vec{\tau} = \begin{bmatrix} D - \Delta a \\ D - \Delta b \\ D - \Delta c \\ D - \Delta d \end{bmatrix} \tag{5.4}$$

where D is the target delay from the tapping point s to all the sinks. Rewriting Eq. (5.3) as $G^{\vec{\tau}} = C$, and substituting (2.1), (5.1), (5.2) and (5.4) into this new form, we have the following relations:

$$\begin{aligned} &\tfrac{1}{\rho}[\tfrac{w_{as}}{l_{as}}(D - \Delta a) - \tfrac{w_{ab}}{l_{ab}}(\Delta a - \Delta b) - \tfrac{w_{ad}}{l_{ad}}(\Delta a - \Delta d)] \\ &\quad = \tfrac{\beta}{2}\left[l_{as}w_{as} + l_{ab}w_{ab} + l_{ad}w_{ad}\right] + C_{Da} \end{aligned} \tag{5.5}$$

$$\begin{aligned} &\tfrac{1}{\rho}[\tfrac{w_{bs}}{l_{bs}}(D - \Delta b) - \tfrac{w_{ab}}{l_{ab}}(\Delta b - \Delta a) - \tfrac{w_{bc}}{l_{bc}}(\Delta b - \Delta c)] \\ &\quad = \tfrac{\beta}{2}\left[l_{bs}w_{bs} + l_{ab}w_{ab} + l_{bc}w_{bc}\right] + C_{Db} \end{aligned} \tag{5.6}$$

$$\begin{aligned} &\tfrac{1}{\rho}[\tfrac{w_{cs}}{l_{cs}}(D - \Delta c) - \tfrac{w_{bc}}{l_{bc}}(\Delta c - \Delta b) - \tfrac{w_{cd}}{l_{cd}}(\Delta c - \Delta d)] \\ &\quad = \tfrac{\beta}{2}\left[l_{cs}w_{cs} + l_{bc}w_{bc} + l_{cd}w_{cd}\right] + C_{Dc} \end{aligned} \tag{5.7}$$

$$\begin{aligned} &\tfrac{1}{\rho}[\tfrac{w_{ds}}{l_{ds}}(D - \Delta d) - \tfrac{w_{ad}}{l_{ad}}(\Delta d - \Delta a) - \tfrac{w_{cd}}{l_{cd}}(\Delta d - \Delta c)] \\ &\quad = \tfrac{\beta}{2}\left[l_{ds}w_{ds} + l_{cd}w_{cd} + l_{ad}w_{ad}\right] + C_{Dd} \end{aligned} \tag{5.8}$$

Let $\vec{w} = \begin{bmatrix} w_{ab} & w_{bc} & w_{cd} & w_{ad} & w_{as} & w_{bs} & w_{cs} & w_{ds} \end{bmatrix}^T$. We rewrite Equations (5.5) through (5.8) as

$$[A\ B]\vec{w} = \vec{C} \tag{5.9}$$

where submatix

$$A = \begin{bmatrix} -\frac{\Delta_{ab}}{\rho l_{ab}} - \beta\frac{l_{ab}}{2} & 0 & 0 & -\frac{\Delta_{ad}}{\rho l_{ad}} - \beta\frac{l_{ad}}{2} \\ -\frac{\Delta_{ba}}{\rho l_{ab}} - \beta\frac{l_{ab}}{2} & -\frac{\Delta_{bc}}{\rho l_{bc}} - \beta\frac{l_{bc}}{2} & 0 & 0 \\ 0 & -\frac{\Delta_{cb}}{\rho l_{bc}} - \beta\frac{l_{bc}}{2} & -\frac{\Delta_{cd}}{\rho l_{cd}} - \beta\frac{l_{cd}}{2} & 0 \\ 0 & 0 & -\frac{\Delta_{dc}}{\rho l_{cd}} - \beta\frac{l_{cd}}{2} & -\frac{\Delta_{da}}{\rho l_{ad}} - \beta\frac{l_{ad}}{2} \end{bmatrix} \tag{5.10}$$

and the diagonal submatrix

$$B = \begin{bmatrix} L_1(D) & 0 & 0 & 0 \\ 0 & L_2(D) & 0 & 0 \\ 0 & 0 & L_3(D) & 0 \\ 0 & 0 & 0 & L_4(D) \end{bmatrix} \tag{5.11}$$

and the right-hand-side (RHS) of Eq. (5.9) is

$$\vec{C} = \begin{bmatrix} C_{Da} \\ C_{Db} \\ C_{Dc} \\ C_{Dd} \end{bmatrix} \tag{5.12}$$

and in the above equations, $\Delta_{ij} = \Delta_i - \Delta_j$ and $L_i(D) = (D - \Delta_i)/(\rho\ l_{is}) - \beta\ l_{is}/2$, i, j = a, b, c and d. Correspondingly, we split $\vec{w}$ into the upper half vector $[w_{ab}\ w_{bc}\ w_{cd}\ w_{ad}]^T$and the lower half vector $[w_{as}\ w_{bs}\ w_{cs}\ w_{ds}]^T$. We can then rewrite Eq. (5.9) as

$$\begin{bmatrix} w_{as} \\ w_{bs} \\ w_{cs} \\ w_{ds} \end{bmatrix} = B^{-1}(\vec{C} - A \begin{bmatrix} w_{ab} \\ w_{bc} \\ w_{cd} \\ w_{ad} \end{bmatrix}) \tag{5.13}$$

To achieve zero-skew, equation (5.13) must hold, and this implies that $w_{as} \sim w_{ds}$ can be expressed as a linear combination of $w_{ab} \sim w_{ad}$.

5.2.2 Linear Programming Formulation

The linear programming model that creates a zero skew mesh can be formulated as follows:

Minimize Total wire area =

$$\begin{bmatrix} l_{ab} \\ l_{bc} \\ l_{cd} \\ l_{ad} \end{bmatrix} \bullet \begin{bmatrix} w_{ab} \\ w_{bc} \\ w_{cd} \\ w_{ad} \end{bmatrix} + \begin{bmatrix} l_{as} \\ l_{bs} \\ l_{cs} \\ l_{ds} \end{bmatrix} \bullet [B^{-1}(\vec{C} - A \begin{bmatrix} w_{ab} \\ w_{bc} \\ w_{cd} \\ w_{ad} \end{bmatrix})] \tag{5.14}$$

$$\textbf{Subject to } \vec{w}_{\max} \geq B^{-1}(\vec{C} - A \begin{bmatrix} w_{ab} \\ w_{bc} \\ w_{cd} \\ w_{ad} \end{bmatrix}) \geq \vec{w}_{\min} \tag{5.15}$$

$$\textbf{and } \vec{w}_{\max} \geq \begin{bmatrix} w_{ab} \\ w_{bc} \\ w_{cd} \\ w_{ad} \end{bmatrix} \geq \vec{w}_{\min} \tag{5.16}$$

where w_{ab}, w_{bc}, w_{cd} and w_{ad} are the decision variables for this problem, "•" refers to the dot product of two vectors, and $\vec{w}_{\min}$ and $\vec{w}_{\max}$ are the lower-bounds and upper-bounds for wire widths which are positive technology dependent parameters. We note that the size of the linear program, in terms of both the number of variables and the number of constraints, is very small, so that its solution can be found very easily.

5.2.3 Wire Elongation Heuristic

Although the above derivation has been made for a simple mesh of the type shown in Figure 5.2, the principles behind this derivation can be extended to any arbitrary mesh containing any number of nodes. There are two reasons why we do not pursue such an approach in this paper: firstly, the use of excessively large meshes consumes a large amount of routing resources, and secondly (and more importantly), the linear program described above may not have a feasible solution for all mesh structures with downstream trees with arbitrary delays and capacitances. In the following paragraphs, we will focus on developing a procedure that ensures a feasible solution for the mesh structure under consideration. We believe that this procedure can be extended for arbitrary meshes, but that perhaps the additional resources required to build these meshes and to ensure a solution for the linear program may not always be worthwhile. A more detailed study of this issue is a topic for further research.

In order for the linear program to have a feasible solution, we now examine equation (5.13). We can always choose the target delay D such that each entry ($L_i(D)$, $i = a$, b, c and d) in matrix B is positive. A direct observation is that to make $w_{as} \sim w_{ds}$ greater than w_{min}, we can make the diagonal entries in B smaller by either decreasing D or elongating l_{is}, $i = a$, b, c and d. We still need to ensure that w_{ab}, w_{bc}, w_{cd} and w_{ad} lie above w_{min}. Since the minimum wire width restriction is a hard limit imposed by the technology, we use a necessary condition to enforce this limit.

Theorem 1 *A necessary condition to ensure that the wire width does not violate the minimum width requirements is*

$$A \begin{bmatrix} w_{ab} \\ w_{bc} \\ w_{cd} \\ w_{ad} \end{bmatrix} \leq -B\vec{w}_{\min} \tag{5.17}$$

Proof 1 *Since B is a diagonal positive definite matrix, multiplying B to (13) to get*

$$\vec{C} - A \begin{bmatrix} w_{ab} \\ w_{bc} \\ w_{cd} \\ w_{ad} \end{bmatrix} \geq B\vec{w}_{\min}$$

where every entry in $\vec{C}$ is greater than zero. Therefore:

$$A \begin{bmatrix} w_{ab} \\ w_{bc} \\ w_{cd} \\ w_{ad} \end{bmatrix} \leq A \begin{bmatrix} w_{ab} \\ w_{bc} \\ w_{cd} \\ w_{ad} \end{bmatrix} - \vec{C} \leq -B\vec{w}_{\min}\square$$

When any entry in matrix A is positive, it is possible to violate condition (5.15) and send the solution outside the feasible region. In such a situation, if we manage to make the LHS of (5.15) smaller and the RHS of (5.15) larger, we will be able to draw the optimal solution back to the feasible region. This may be achieved through a wire elongation heuristic, which is stated as follows. Whenever there is a negative entry a_{ij} (i, j = valid combinations of a, b, c or d) in matrix A, elongate the wire length l_{ij} to make the LHS of (5.15) smaller and elongate the wire length l_{is} to the corresponding entry in matrix B larger. Notice here that when we elongate l_{ij}, another entry in matrix A is also decreased, which makes this entry more negative, and therefore does not affect the feasibility of the corresponding constraint.

After wire elongation, if the linear programming solver still cannot find an optimal solution, it means that our target delay D has been chosen to be too strict. We must then relax the target delay to a reasonable value.

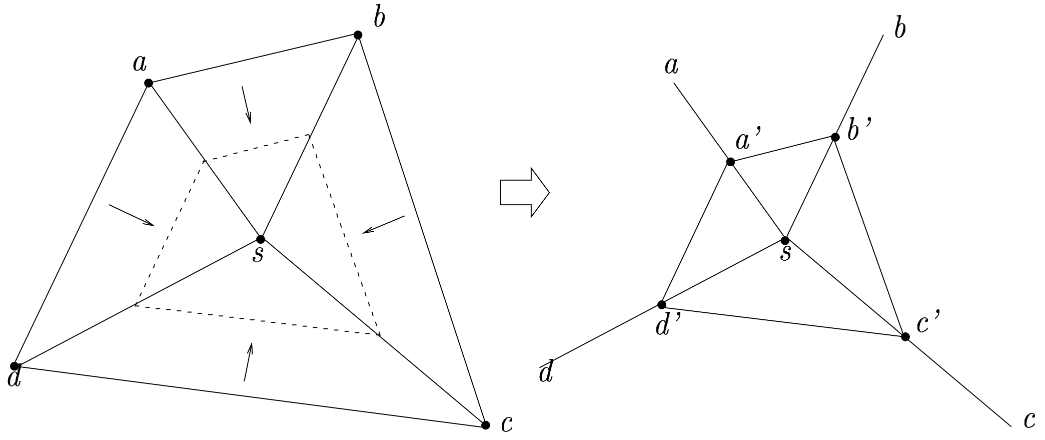

Figure 5.3: Mesh "shrinking" heuristic.

5.2.4 Transition Time Improvement Heuristics

Even though asserting a harsher transition time threshold during buffer insertion will improve the worst-case transition time, when the transition time of the four downstream trees are slightly less than the required transition time threshold, the overlying mesh built based on these will still have a large transition time. A straightforward method is to insert four buffers to the roots of the four downstream trees and the overlying mesh is build based on the buffer input capacitances and downstream delays.

When the root of the four downstream trees are very much away from each other, even though when the transition time of each tree is small, the above construction will end up with very long wires in the overlying mesh. The transition time in the upmost hybrid structure can still become undesirably large. We propose the following method called mesh "shrinking" heuristic to tackle this problem. It performs a heuristic move for each root toward the direction of the tapping point s. The step of each move is chosen to be a fixed shrinking ratio ($r_{sh} < 1$) of the corresponding distance from the tree root to the tapping point s. This heuristic move is illustrated in Figure 5.3, where the original mesh nodes $a - d$ has "shrunk" to nodes $a' - d'$ respectively and the shrinking ratio is 0.5 in the figure.

The mesh "shrinking" and buffer insertion can be applied together with each other for larger structures in order to meet the transition time constraint.

5.2.5 Overall Construction Algorithm

Our linear-programming based zero-skew mesh construction algorithm is as follows:

1) Choose a target delay D. Starting with the initial $D = \max\{\Delta a, \Delta b\ , \Delta c\ , \Delta d\}$, update the value of D by multiplying it with a small factor until all $L_i(D)$'s are positive, $i = a$, b, c and d.

2) Check each entry in matrix A: if any entry is positive, heuristically elongate the wire length(s) by multiplying it with a small factor. Update matrices A and B and the objective function. If any entry in matrix B becomes negative, go to step 4.

3) Solve the linear programming problem stated by (5.14), (5.15) and (5.16). If the solver finds an optimal solution for w_{ab}, w_{bc}, w_{cd} and w_{ad}, calculate w_{as}, w_{bs}, w_{cs} and w_{ds} using (5.13), connect the tapping point to the clock driver and go to step 5; else go back to step 2.

4) Relax the target delay D and go back to step 2.

5) If the transition time of the hybrid structure is too large, insert buffers to the four mesh nodes and apply the mesh shrinking heuristic.

5.2.6 Extension to Multiple Level Mesh Structures

This algorithm can be extended to build multi-level mesh structures, as shown in Figure 5.4. We can add clock buffers to the four tapping points of the lower-level

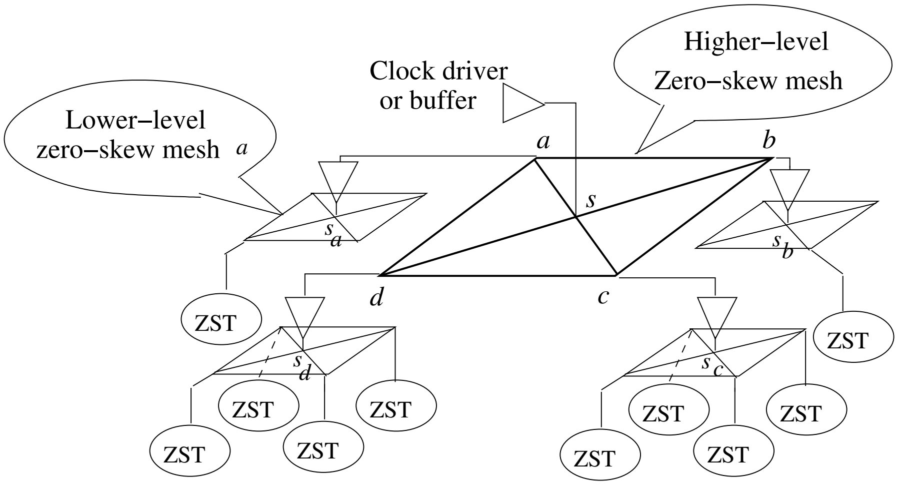

Figure 5.4: Two-level mesh with underlying trees. For simplicity, only some of the sixteen zero skew trees are shown.

mesh at nodes a, b, c and d and construct the upper-level mesh based on the delays and buffer capacitances of the lower-level four meshes. We point out that in such case the number of trees becomes sixteen (for convenience, only a few of these are shown in Figure 5.4).

5.3 Fast Analysis and Heuristic Optimization

5.3.1 Fast Higher-order Delay Calculation

The mesh/tree clock network constructed using the above algorithm assumes minimum wire width for the underlying zero-skew trees. We realize that the transition time, defined by the delays between 10% of Vdd and 90% of Vdd, is another important factor that affects the performance of the clock network. It determines the maximum clock frequency of the chip. In our work, this is achieved by simultaneous buffer insertion and wire widening optimization. A heuristic optimization procedure that increases the wire width with the largest transition time reduction while maintaining the worst-case skew within a small constraint is developed.

Due to the well-known limitations of the Elmore delay model, we use a core timing calculator based on a higher order model. The transition time and skew optimization is especially crucial for deep submicron networks to work reliably. For the mesh/tree model, the analysis can be performed using a hierarchical model reduction and simulation method similar to the procedure described in Chapter 3. In brief, this simulator proceeds in three stages: First, all trees are reduced to an equivalent passive model using PRIMA [5]. Next, the mesh is solved using PRIMA, along with the reduced passive tree models, to find out all the step responses of the mesh nodes. Finally, these mesh responses provide the voltages at the root of each tree and they are used to solve the step responses $V(t)$ of all the sinks at the bottom of the trees. This hierarchical approach serves to reduce the amount of computation required during the analysis. This procedure is faster and more efficient while applying to a pure tree structure due to the use of the tree traversal technique. The propagation delay at each clock sink is calculated as the time of 50% of the final steady value. The sink i with the maximum propagation delay D_i and the sink j with the minimum propagation delay D_j can be identified, and the skew of the whole circuit is $D_i - D_j$. Similarly, the transition time of each sink is calculated as the difference of time between 90% and 10% of the final steady value, and the sink D_k with the worst-case transition time can be identified.

5.3.2 Calculation of Sensitivities

To calculate the delay sensitivities and the skew sensitivities, the conventional adjoint sensitivity analysis is first applied to calculate the partial derivatives of voltages with respect to all the wire widths in the network. The adjoint network has the same topology as the original network. Applying a current source of -δ $(t - D_i)$ at sink i, we can simulate the adjoint network backward of time to obtain the waveforms $\psi_k(\tau)$, where k is all the nodes in the circuit, τ = -t. The value of $\frac{\partial V_i}{\partial C_k}$can be computed by the convolution between $V_k(t)$ and $\psi_k(\tau)$. Similarly, $\frac{\partial V_i}{\partial R_{pq}}$can be computed by the convolution between the current responses $I_{pq}(t)$ and $\varphi_{pq}(\tau)$, where R_{pq} is the resistor between node p and q. $\frac{\partial V_i}{\partial w_{pq}}$ can be found out using the chain rule. Using the formula described in [53], the delay sensitivity of sink i with respect to wire width w_{pq} can be computed as

$$\frac{\partial D_i}{\partial w_{pq}} = -\frac{\partial V_i/\partial w_{pq}}{\partial V_i/\partial t}|_{t=D_i} \tag{5.18}$$

In (5.16), a positive value of $\frac{\partial D_i}{\partial w_{pq}}$ means increasing wire width can only increase the delay and vice versa. The same method can be used to compute $\frac{\partial D_j}{\partial w_{pq}}$. From these, we obtain the skew sensitivity as:

$$\frac{\partial Skew}{\partial w_{pq}} = \frac{\partial D_i}{\partial w_{pq}} - \frac{\partial D_j}{\partial w_{pq}} \tag{5.19}$$

A positive value of $\frac{\partial Skew}{\partial w_{pq}}$means that increasing the wire width of pq will increase the skew of the whole circuit and vice versa. We point out here that to calculate the skew sensitivity, two adjoint networks are to be analyzed.

The transition time sensitivity with respect to wire widths can be calculated in a similar way. By applying the current source to the sink with the worst-case transition time, one adjoint network analysis is performed and the convolution is integrated from zero to $D_{90\%}$ and to $D_{10\%}$ respectively.

5.3.3 Heuristic Optimization Procedure

The heuristic optimization procedure works as follows:

- Set a target transition time and the maximum allowable skew constraints.

- Simulate the original clock network to find out the sink i with the maximum delay, the sink j with the minimum delay and the sink k with the largest transition time.

- Find out $\frac{\partial D_i}{\partial w}$, $\frac{\partial D_j}{\partial w}$ and $\frac{\partial Skew}{\partial w}$for all the wire widths in the network using the adjoint sensitivity analysis described above.

- Find out $\frac{\partial TranTime}{\partial w}$ for all the wire widths in the network using the adjoint sensitivity analysis described above.

- If the worst-case skew is greater than the constraint, for all the wire widths with negative skew sensitivities, bump up the wire width with the most negative $\frac{\partial D_i}{\partial w}$, positive $\frac{\partial D_j}{\partial w}$or negative $\frac{\partial D_j}{\partial w}$ with its absolute value smaller than $\frac{\partial D_i}{\partial w}$, by multiplying it with a small factor (< 1.1).

- Otherwise, if the worst-case transition time is greater than the constraint, bump up the wire width with the most negative $\frac{\partial TranTime}{\partial w}$, by multiplying it with a small factor (< 1.1).

- Repeat the above procedure until both the skew and transition time constraints are met or until there is no negative skew or transition time sensitivity.

If no negative skew or transition time sensitivity exists at certain iteration, increasing any of the wire widths will increase the skew or the transition time, which means this network cannot be optimized any further by wire widening only.

The above procedure assumes the priority of skew than that of the transition time, i.e., only when the skew is under the pre-specified constraint do we optimize the transition time. The reason of this heuristic is that, by widening wire widths according to the transition time sensitivity, the skew is likely to increase. By tracking the change of skew, we try to prevent increasing the skew too far. This heuristic is efficient due to another fact that reducing skew usually doesn't increase the transition time.

It may be noted that this sensitivity-based procedure is similar to the TILOS heuristic [6] for transistor sizing, where the convexity property is used to justify the use of the greedy heuristic. In our case, the complexity of the delay model and the double-sided timing constraints make it unlikely that convex properties hold. However, the only impact of this is that when we do obtain a zero-skew network, there is no guarantee that the cost function, given by the sum of wire areas, will be minimized. We believe this to be acceptable in the scenario of clock network construction, particularly since there are many other steps in the tree construction procedure that are chosen in heuristic ways, such as the decision on which pins to combine while building the tree, and in which order. Our experimental results in the next section will show that the cost of building the clock network is always reasonable.

5.4 Experimental Results

We have applied our clock network design scheme to the five benchmark examples in [45] and an industry chip (1.3cm×1.3cm, named as "Industry Ckt") under the 0.18micron technology. For the experiments of benchmark circuit, we scaled the wire widths while maintaining the same sink load capacitances under the assumption that under scaling theory, for a fixed area, the reduction in the feature size cancels the increase of the number of capacitive loads. Both the mesh/tree construction algorithm and the buffer insertion and heuristic wire width optimization procedure have been implemented using C++. All of the experiments are performed on Sun Ultra-60 workstations.

We use Tsay's [45] zero-skew algorithm to construct the underlying buffered trees. A simple heuristic for pairing up clock pins is used. We recursively partition the

	# of	Prop	Skew	Tran	Area	Num	CPU
	Pin	Delay	(ps)	Time	(cm^2)	of	Time
		(ns)		(ns)		Buf	(sec)
		Hybrid structure					
R1	267	0.090	3.052	0.169	3.18e-4	36	5.05
R2	598	0.128	3.332	0.229	5.35e-4	66	11.85
R3	862	0.122	2.234	0.200	4.61e-4	124	18.23
R4	1903	0.174	2.616	0.333	4.27e-4	220	39.50
R5	3101	0.209	1.741	0.389	4.67e-4	274	71.57
Ind Ckt	3371	0.215	3.967	0.456	1.01e-3	198	79.37
		Tree structure					
R1	267	0.115	6.676	0.261	1.44e-4	36	5.18
R2	598	0.145	5.355	0.305	1.75e-4	66	13.51
R3	862	0.125	2.583	0.229	1.43e-4	124	18.31
R4	1903	0.215	2.772	0.475	2.34e-4	220	39.98
R5	3101	0.230	3.098	0.469	2.46e-4	274	72.20
Ind Ckt	3371	0.296	6.038	0.728	5.23e-3	198	79.50

Table 5.1: Comparison between structures of clock nets.

chip into quadrants and group the pins in the same quadrant. Buffers are inserted while merging the two sub-trees. This creates a buffered binary tree for each case. A one-level mesh constructed using our linear programming based algorithm then connects the four underlying trees. We also constructed the full tree structure using Tsay's method for comparison. By asserting the same transition time constraint and the same total number of buffers to both structures, exactly the same buffered downstream trees are constructed being driven by the two structures. The same clock driver is applied to both structures. The wire width of trees in R1$\sim$R5 is

$0.54\mu m$ while that in the industry circuit is $0.8\mu m$. Table 5.4 shows the performance of our zero-skew mesh/tree structures and the comparison with the full tree results. The propagation delay and transition time have been reduced by an average of 10% and 15%. Under the Elmore delay metric, the skew of each example is zero, by construction. We further observed the actual delays and skews using the HPRIMA simulator and these are listed in the "Prop Delay" and "Skew" columns in Table 5.4. The transition time is listed in the "Tran Time" column. The CPU times are shown in the column of "CPU Time" and it is seen to be fast for all these examples. The maximum wire width obtained by the linear program is under 2 microns. In the buffer library, we have four different sized buffers. The total wire area for the topmost structure is listed in the "Area" column. It should be noted that since the industry chip has larger sink capacitances and wider minimum wire width than R5, larger sized buffers are applied to it and the total number of buffers are less than that in R5. Wire elongation ratio for the mesh in each circuit is listed in Table 5.4. It can be seen that the worst-case wire elongation ratio is below 7%.

	Initial total wire length (μm)	Optimal total wire length (μm)	Wire elongation ratio
R1	22720	22720	0.00%
R2	31446	31446	0.00%
R3	31838	33567	0.26%
R4	39366	39504	0.35%
R5	47189	47681	1.04%
Ind ckt	46868	49853	6.37%

Table 5.2: Mesh wire elongation ratio.

The performance comparison listed in Table 5.4 is only of relative significance since an experimental setup for an accurately fair comparison between the two structures is difficult to compose. Even though the hybrid structure has larger wire area than pure tree, it potentially needs a larger clock driver. When a pure tree is sized such that it has the same total wire area as the hybrid structure, it has much larger maximum wire widths, which induces wire congestion problems. So the balance of fairness for experiments in Table 5.4 is roughly achieved by using larger area to favor the hybrid structure while using the same clock driver to favor the pure tree. At the same time, both structures are constructed to drive exactly the same downstream buffered trees. Nevertheless, the argument that the performance of one structure is better over another is still hard to announce and there are yet other factors, for example, tolerance to process variations, sensitivity to loading conditions, etc, to consider.

	Prop Delay (ns)	Skew (ps)	Tran Time (ns)	Area (cm^2)	w_{max} (μm)	Num of Itr	CPU Time (sec)
R1	0.090	1.283	0.169	3.37e-4	2.6	9	0.55
R2	0.127	1.346	0.229	5.68e-4	4.5	19	1.09
R3	0.122	1.270	0.200	4.89e-4	3.3	18	1.27
R4	0.167	2.890	0.296	6.66e-4	3.4	82	8.72
R5	0.196	1.935	0.338	5.79e-4	1.6	41	5.37
Ind Ckt	0.198	4.297	0.387	1.35e-3	5.0	68	6.61

Table 5.3: Wire width optimization results.

Next, the circuits from Table 5.4 are further optimized using the technique described in Section 5.3, and Table 5.3 shows the wire width optimization results. The skews shown in Table 5.3 are analyzed by the HPRIMA simulator. The total number of iterations of the heuristic optimizer is listed in the "Num of Itr" column. The maximum wire width after optimization is lised in the "w_{max}" column. It can be seen that after optimization, the propagation delay and the transition time of all the networks are both efficiently reduced while the maximum skew is within $5ps$.

	Nom Skew (hybrid) (ps)	Nom Skew (tree) (ps)	Skew Var Mean (hybrid) (ps)	Skew Var Mean (tree) (ps)	Std (hybrid) (ps)	Std (tree) (ps)
R1	3.052	6.676	0.503	1.404	0.062	0.161
R2	3.332	5.355	0.677	1.365	0.056	0.192
R3	2.234	2.583	0.372	1.378	0.036	0.121
R4	2.616	2.772	0.391	0.492	0.084	0.114
R5	1.741	3.098	0.680	1.037	0.064	0.354
Ind Ckt	3.967	6.038	0.605	1.642	0.106	0.176

Table 5.4: Process variation results.

Table 5.4 shows the statistical results for process variations in the top-most level of structures (hybrid structure before wire sizing and pure tree). Both spatial and random wire width variations are applied to each wire. Specifically, every wire width is adjusted by

$$w = w_0 + p_1 \times x + p_2 \times y + N(0, \sigma) \tag{5.20}$$

where w_0 is the wire width with no variation, and $(x,\ y)$ is the location of a wire within the chip, which is taken to be the central point of the wire. Spatial variations are represented by the parameters p_1 and p_2, and $p_1$is set to be equal to p_2 in the sense that the same variation occurs in both orientations. In addition, a random variation is modelled by a normal distribution of $N(0,\ \sigma)$ with a mean of zero and a standard deviation of σ . We assume that the spatial variation and the random variation contribute equally to the wire width variations and the maximum variation at the minimum wire width is 15%. A Monte Carlo simulation similar to [42] is performed and the experiments with 1% accuracy and 99% confidence is listed in the table.

The nominal skew, which is the skew without any wire width variation is listed in the "Nom Skew" columns. The mean of variations away from the nominal skew is shown in the "Skew Var Mean" columns. In all these examples, both the skew variation and the standard deviation of skews of the hybrid structure are less than the pure tree structure. It can be seen that the hybrid structure is less sensitive to process variation than the pure tree structure and it works more reliably.

	Prop Delay (ns)	Skew (ps)	Tran Time (ns)	Shrink Ratio	Elongation Ratio	Num of Buf
R1	0.081	3.825	0.088	0.05	0.7%	40
R2	0.107	3.185	0.124	0.05	0	70
R3	0.119	3.582	0.157	0.05	3.0%	128
R4	0.124	3.190	0.116	0.35	0	224
R5	0.151	1.200	0.153	0.30	0	278
Ind Ckt	0.155	6.09	0.185	0.32	6.29%	202

Table 5.5: Hybrid clock networks after heuristic transition time improvement.

The experiments performed above primarily emphasize the comparison between trees and meshes and tries to make fair comparison with the pure tree structure. Consequently, the overlying mesh of some circuits are actually very large and their transition time is still large. After applying our buffer insertion heuristic to the four tree roots and the mesh shrinking heuristic, the updated performance for our hybrid structure is listed in Table 5.5.

Table 5.6 shows the comparison of our hybrid structure with the IBM structure (named as "R6") discussed in [51], which also uses mesh structures, but at the bottom

	# of Pin	Structure	Prop Delay (ns)	Skew (ps)	Tran Time (ns)	Area (cm^2)	Drv Res (Ω)
R6	64	Hybrid	0.156	0.120	0.428	9.46e-4	18.6
R6	64	Tree	0.149	2.880	0.428	7.24e-4	18.6
R6	64	IBM	0.148	2.300	0.428	1.16e-3	5.6
R6	64	Hybrid+IBM	0.155	3.160	0.428	1.38e-3	8.3

Table 5.6: Comparison with the IBM structure.

of the hierarchy. A total of 64(8x8) uniformly distributed sinks is generated, with the load capacitance ranging from $1.3fF$ to $83fF$. Connecting all the leaf nodes of the pure tree structure into a full 8x8 mesh generates the IBM structure. The minimum wire width for each case is $0.54\mu m$. All the networks are optimized to have the same transition time. A 0.1ns ramp input is applied to each circuit and HSPICE simulation is performed. Compared to the pure tree, the IBM structure reduces skew by 20%. Among all structures, the hybrid one has a much smaller skew than others. It can be seen that the IBM structure requires a much larger clock driver with smaller driver resistance (listed in the "Drv Res" column) to achieve the same transition time as others, which leads to more power consumption.

An interesting structure is constructed by combining our hybrid structure with the IBM structure and the performance of such a structure is also listed in Table 5.6. Compared to the pure IBM structure, even though the skew is enlarged, it requires a smaller clock driver. Another benefit of such structures (IBM and Hybrid+IBM) with meshes at the lower level is that changes in clock loads or locations cause little change in the clock skew, but this is not the case for either the hybrid or the pure tree structure.

5.5 Conclusion

This chapter has proposed an efficient algorithm to construct a zero-skew clock network with a hybrid mesh/tree structure. Tree structures are created using Tsay's approach, and each level of the mesh is designed by solving a small linear program, guaranteeing a zero-skew construction under the Elmore model. A heuristic wire width optimization procedure that can reduce both the clock transition time and skew has been proposed.

The procedure shown in this chapter is very versatile and can be used for the hierarchical construction of hybrid tree/mesh structures of other types. For example, one such topology could be a top-level mesh that distributes trees that are each connected to meshes at the next level, and so on. We expect these topics to be explored in future research, based on the theoretical foundation laid by this research that illustrates the feasibility of building clock meshes with guaranteed zero skews under Elmore delays.

Chapter 6

Congestion-driven Codesign of Power and Signal Networks

6.1 Introduction

6.1.1 Motivation for the Codesign of Signal and Power Wires

The role of interconnect has become increasingly critical in nanometer design and the need to meet stringent performance constraints has resulted in strong contention for scant routing resources. A major consumer of these resources is the power distribution network, which must be designed to ensure reliable V_{dd} and ground levels, and therefore requires the use of dense grids. On the other hand, global wires also compete for the same routing resources, as they often require shortest-path routes to meet their own performance requirements. Traditionally, these two have been designed independently, with the routing needs for a regular power grid being determined first, after which the remaining resources are calculated to provide routing resource budgets for the signal nets.

As the number and criticality of global signal wires becomes more dominant, such a methodology becomes unsustainable as the initial budgets may often be entirely unreasonable, so that ad hoc changes at the end are inadequate to meet the performance and/or routing completion targets. Moreover, the use of a completely regular supply grid cannot be defended in the face of large variations in the voltage droops on a regular grid at different points of the chip. Therefore, in nanometer design, there is a strong need for a unified approach to the design of signal wires and power grids, with an integrated approach to routing resource management.

The considerations involved in the design of the power grid and the signal wires are quite different. Since the former is intended to provide a reliable V_{dd} or ground voltage throughout the circuit, a typical grid contains a dense and regular distributed mesh-like structure that is designed to meet constraints on parameters such as the IR drop noise. Since this entails the use of numerous wide wires, the routing resources used by the power grid are considerable. On the other hand, signal nets are typically tree structures that are designed to connect a source to a number of sinks under constraints related to the timing criticality of the net. The complexity of routing signal nets is exacerbated by their sheer number, so that the routes for any individual net cannot be determined without considering the contributions of the others to the congestion. Most often, global signal wires seek out shortest-path connections, although detours are often essential to navigate around regions of high traffic.

While it is convenient to build a regular power grid with a constant pitch (defined as the distance between adjacent wires in the grid), some degrees of freedom exist and it is desirable that they be exploited. For instance, the density of the grid in different parts of the chip need not necessarily be the same, since the use of a denser grid near hot spots with strong current sinking requirements would help in providing reliable voltages, while a sparser grid is adequate in less critical regions. This may further be combined with considerations for routing signal nets, so that in regions where the demand for routing resources from signal nets is high, a sparser power grid may be used as long as the performance constraints on the supply and ground lines can be met; likewise, signal nets are well advised to avoid the hot spots of the chip if possible, since these may need a locally dense power grid.

Strictly speaking, the results of global routing have an effect on the demands on the power grid, since the locations of buffers are determined by the routes that are chosen by signal wires. Since the number of buffers can be extraordinarily large in nanometer technologies, buffers can collectively be nontrivial contributors to power grid current. However, these effects can be controlled at the methodology level. For example, using the model of [54], where buffers are predistributed and sprinkled into various functional blocks, we may estimate the current drawn by the buffers and include it in the current drawn by the functional block. As long as the routing is buffer-aware and deliberately seeks out regions with high buffer availability, the use of signal routing-independent current waveforms for the functional blocks can be justified in generating an initial power grid design. Later in the design process, wire widening and decap assignment may be used to overcome second-order effects.

6.1.2 Previous Work on Signal Wires and Power Wires Individually

The idea of managing wire congestion in signal routing has long been a significant objective in global routing. Various congestion-driven techniques include sequential routing (e.g., [55]), rip-up-and-reroute (e.g., [56]), and multicommodity flow based methods (e.g., [57]) have been proposed. Most of these techniques aim at solving the problem solely at the routing stage, assuming that the total routing resources are fixed. Recent publications [54,58] have presented techniques for simultaneous global routing and resource allocation under performance constraints.

Power grid optimization techniques have also been studied in [8, 30, 59]. All of these aim at minimizing total power wire area subject to the voltage droop and/or electromigration constraints, formulating the problem as a nonlinear program. While [30] solves the problem by relaxing it to a sequence of linear programs, both [8] and [59] have employed gradient-based nonlinear optimization techniques. The formulation in [12] handles the transient voltage droop noise under worst-case switching events, while [59] considers the worst-case voltage droop and current density. The work in [60] presents a technique for shield insertion in a predesigned power grid to control inductive effects.

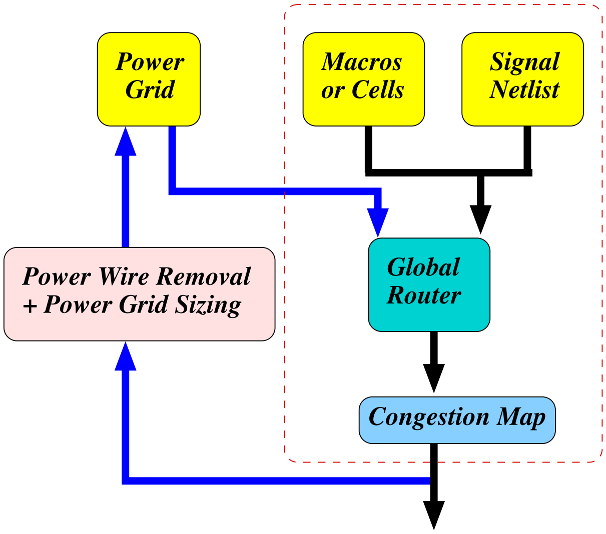

Figure 6.1: Congestion-driven power grid design and global routing.

6.1.3 Contributions

To the best of our knowledge, no published work performs a concurrent optimization of the power grid along with signal wires under routing congestion constraints, and this is the subject of the work presented in this chapter.

Our proposed congestion-driven flow is illustrated in Figure 6.1. The dashed rectangle corresponds to a more conventional global routing flow, where the routing budgets for power grid lines are used to allocate routing resources for the signal lines, and these budgets are frozen throughout the design. In essence, our approach presents a new flow that adds a feedback loop that permits the readjustment of the signal routing budgets by altering the power grid appropriately.

Our procedure begins by constructing the initial Steiner trees for the global signal wires without considering the congestion contributions of the power grid. The initial power grid is provided as an input to the algorithm (and may, perhaps, be a regular grid or a manually designed grid) and is assumed to be dense enough to sustain the switching activities of the functional blocks in the chip. Our proposed iterative scheme alternately (i) reroutes the global signal wires, and (ii) adjusts the supply network to decongest regions of high contention where the voltage droop constraints are easily satisfied. The adjustments made to the power grid consist of wire removal from the grid in congested regions, and sizing of the power grid to compensate for this removal. Our approach incorporates a tight coupling between power grid adjustments and the routing of signal wires to exploit the altered congestions that result from these adjustments, and aims to solve problems with severe congestion constraints where conventional techniques are inadequate.

6.2 Power Grid-aware Signal Routing

6.2.1 Power Grid-aware Congestion Estimation

As in global routing, we tessellate the entire chip into an array of grid cells, as shown in Figure 6.2(a), and use the wiring information across the boundaries between neighboring grid cells to estimate the signal wire congestion distributions. We denote the width, in μm, of a boundary b between two neighboring grid cells as $W(b)$. This width represents the limited resources that must be shared on each layer by the supply lines and the signal lines that traverse the boundary, as shown in Figure 6.2(b). In other words, the number of wires crossing b is inherently limited by the width $W(b)$, and $W(b)$ may partly or wholly be occupied by the crossing wires.

We represent the total width occupied by power grid wires on boundary b as $P(b)$. If a power grid wire p_i has a track width of $w(p_i)$, which includes its wire width and the required spacing from an adjacent wire, and there are a set of such wires, $p_1, p_2, ...p_m$, that cross b, then $P(b) = \sum_{i=1}^{m} w(p_i)$ represents the space that is unavailable for signal wires to cross the boundary. Therefore, we subtract this quantity from the boundary width to obtain the space available for signal wires as $W(b) - P(b)$. Typically, a uniform track width $\bar{w}$ is applied to all the signal wires in the congestion estimation or global routing stage. Hence, signal wire congestion is often expressed in terms of the number of wiring tracks and the number of tracks available for signal wires is $T(b) = \lfloor \frac{W(b)-P(b)}{\bar{w}} \rfloor$. If there are $S(b)$ signal wires that cross a boundary b, then the overflow on b is $S(b) - T(b)$. All tile boundaries with positive overflow values form a congestion map for a chip. The wire density at b is represented as $S(b)/T(b)$, measuring the congestion of b. A common objective for global routing is to ensure that there is no boundary with the wire density greater than one, i.e., $S(b) \leq T(b)$.

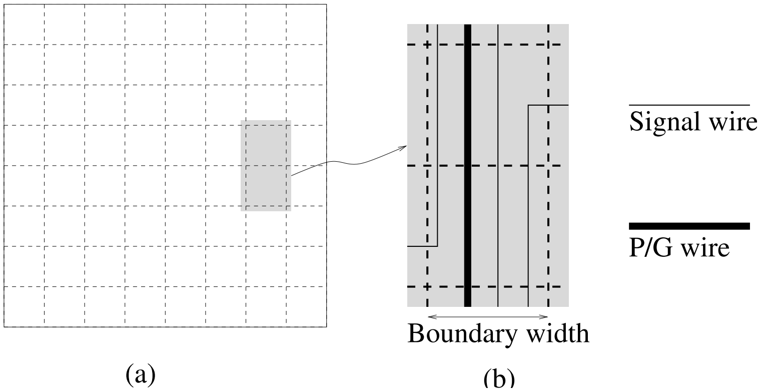

Figure 6.2: Wire congestion estimation based on tessellation on a chip.

6.2.2 Congestion-driven Global Routing

In this work, the topology selection for the power grid is tightly coupled with the requirements of global routing for signal wires, and it is important to estimate the congestion contribution of the signal wires by implementing a fast global routing procedure. The technique used for this purpose is described in this section, but it may be noted that this procedure may be replaced by any other computationally efficient method that shares a similar goal.

At the beginning of the algorithm, we perform a coarse global routing of all signal nets to obtain an estimate of the distribution of signal wire congestion at various boundaries, and feed this information to power/ground optimizer. The global routing technique used here is similar to [54], where a Steiner tree is initially constructed for each signal net using the AHHK algorithm [61] without considering congestion. The AHHK algorithm itself uses a two-step procedure for each net: it first constructs a minimum cost spanning tree for the net; this is then converted to a Steiner tree via a greedy overlap removal process. After the initial Steiner trees have been constructed, an iterative rip-up-and-reroute procedure is applied to further reduce the wire congestion. The outer loop, consisting of signal routing followed by power grid optimization, is repeated iteratively until the constraints are satisfied or no further improvement is possible. In each iteration, the global routing solution from the previous iteration is used as a starting point for the rip-up-and-reroute procedure, with updated congestion values being used to direct the routing.

The rip-up-and-reroute procedure processes each net sequentially using the fixed net ordering heuristic proposed in [62], continuing until the maximum wire density is no greater than one, or after two complete iterations, since the congestion reduction after the second iteration is limited. Each net undergoing rip-up-and-reroute is entirely deleted and then rerouted using an algorithm similar to the min-max tree algorithm in [55].

A min-max tree is a Steiner tree such that its maximum edge cost among all the edges is minimized. This tree is built on a tile graph that is the dual of the tessellation, where vertices correspond to tile rectangles and edges are used to connect the vertices corresponding to adjacent tiles in the graph. The weight on an edge is based on the wire density, so that for an edge that crosses over a boundary(edge) b, this is defined as:

$$weight(b) = \begin{cases} \frac{S(b)+1}{T(b)-S(b)}, & S(b) < T(b) \\ L\frac{S(b)}{T(b)}, & \text{otherwise} \end{cases} \tag{6.1}$$

where L is a large number. Therefore, if the capacity is not exceeded, the weight is the number of wires crossing b, divided by the number of available tracks that remain. This is found to be particularly effective since it increases the penalty of using the boundary as the resource usage approaches the full boundary capacity, and beyond that, presents successively larger penalties for capacity violations.

6.3 Power Supply Noise Analysis

As usual, the same power grid model as discussed in Chapter 2 is used in this work. This model leads to a large-scale linear circuit for power grid analysis, which is efficiently analyzed using the technique proposed in Section 4.2.1 of Chapter 4.

The wire width optimization procedure for the supply network, described in Section 6.4.2, also requires the computation of gradients, and this is performed using the simulation framework. Specifically, the transient adjoint sensitivity analysis technique to calculate the sensitivity of the noise metric with respect to every tuning parameter, which, in our case, is the width of every power wire.

The noise and sensitivity analysis techniques used here are substantially similar to our work on decoupling capacitor (decap) placement (discussed in Chapter 4) and have therefore been described only very briefly here. The only difference is that the tuning parameters in decap placement are the decap values, while in this work, the tuning parameters correspond to the width of each power grid wire. Consequently, current waveforms instead of voltage waveforms must be calculated for adjoint sensitivity computations in this work. The sensitivity of noise Z with respect to all of the resistors in the circuit can be computed from the following convolution [23, 24]:

$$\frac{\partial Z}{\partial R} = \int_0^T \varphi_R(T-t) i_R(t) dt, \tag{6.2}$$

where $\varphi_R(\tau)$ is the current waveform flowing through the resistor R in the adjoint circuit.

The same waveform compression and fast convolution technique proposed in Section 4.2.2.2 of Chapter 4 can be applied to the current waveforms here. Consequently, the complexity of the convolution calculation over $[0, T]$ is $O(N + M)$, where N and M are the number of linear segments on the original and adjoint current waveforms.

Similar to Eq. (3.33), the sensitivities to the *width* of each wire width can then be calculated using the chain rule:

$$\frac{\partial Z}{\partial w} = \sum_{i=1}^{P} \frac{\partial Z}{\partial R_i} \times \frac{\partial R_i}{\partial w}, \tag{6.3}$$

where P is the total number of segments in one grid wire with the same width w. Since the resistance $R_i = \rho \frac{l_i}{w\ h}$, where l_i the length of the wire segment, w is the width of the wire, h and ρ are the thickness and resistivity of the metal layer that the power wire lies in, it is easily verified that Eq. (6.3) becomes:

$$\frac{\partial Z}{\partial w} = - \sum_{i=1}^{P} \frac{\partial Z}{\partial R_i} \times \frac{\rho\ l_i}{h\ w^2} \tag{6.4}$$

6.4 Power Grid Design Scheme

Starting with a dense grid that is guaranteed to meet the constraints on the supply network, our scheme iteratively sparsifies the grid to ease the wire congestion. In each iteration of the loop in Figure 6.1, the power grid is adjusted using a two-step technique:

- In the first step, a wire removal heuristic is used to make the grid less dense in some regions, with due consideration paid to both the congestion information and the power grid noise.
- This is followed by a wire sizing step that readjusts the sizes of wires in the grid to compensate for the loss of this wire.

Therefore, our procedure ensures satisfactory performance of the power grid while utilizing just enough routing resources, so that the resources available to signal routing are maximized.

6.4.1 Power Grid Wire Removal Heuristic

As stated in Section 6.2, the congestion is measured in proportion to the overflow value on each tile edge. All power wires that lie in congested regions are potential candidates for removal, except those that lie within hot spots where the voltage droop is significant. The rationale behind this is that whenever a power wire is removed, the performance of the overall grid is compromised, and this is all the more noticeable if this wire lies within a hot spot. Therefore, we define a power wire as *critical* if the worst-case voltage droop on it is beyond some specified threshold VDP_{th}. Critical wires are not candidates for removal even if they lie in congested regions.

In addition, we define the *criticality* (Crt) of each non-critical power wire as the reciprocal of the average distance of that wire to critical nodes with nonzero noise within its closest region. Since the *criticality* of a power wire is only defined for non-critical wires, all these distances have non-zero values. For example, when a vertical

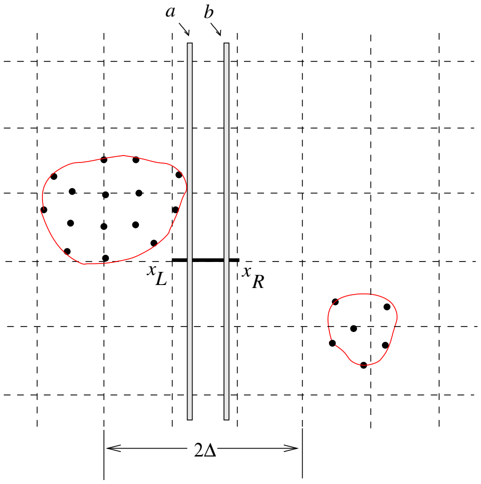

Figure 6.3: Illustration of the criticality of power wires.

non-critical power wire crosses a horizontal tile boundary located between $[x_L,\ x_R]$, the criticality of this wire to its closest noisy region is

$$Crt_p = \frac{K_c}{\sqrt{\sum\limits_{i\epsilon|x_b - x_i| \leq \Delta} (x_p - x_i)^2}} \tag{6.5}$$

where x_p is the x-coordinate of the vertical power wire, x_b is the average of the two boundary terminals x_L and x_R, K_c is the total number of critical nodes within some distance Δ to the tile center and x_i is the x-coordinate of the critical node i. In our experiments, we take Δ as 1.5 to 2 tile edge lengths. It is clear to see that the larger the criticality of a power wire is, the closer this wire is to one of the hot spot region. Given several candidate power wires across one tile boundary with its overflow value larger than OV_{th}, the criticality of each wire determines an optimal order of removal for these wires.

Fig. 6.3 illustrates the importance of the criticality of two non-critical power wires a and b. The criticality rule results in the order of removal for this example being first b and then a. Suppose the removal of any one of the two power wires can update the overflow value of the dark boundary shown in the figure below OV_{th}, then only power wire b is removed and a will be kept, which makes the resulting power grid perform better than the other option of removing a while keeping b and eases the post-processing task of power grid sizing discussed in the following section.

The power wire removal heuristic proceeds as follows. First, the tile boundaries are sorted in decreasing order of their overflow values and all non-critical power wires are sorted according to their criticality values. Next, non-critical power wires are removed in the order of the sorted tile boundaries, provided the overflow value at the boundary is larger than OV_{th}. Several practical considerations must be taken into account. Firstly, the power wire removal process should be accompanied by dynamically updating the overflow value on every tile edge, so that subsequent wire removal is based upon the updated congestion information. Secondly, a reasonable number of power wires must be removed in each iteration, and this depends on the values of OV_{th} and VDP_{th}. Since the choice of these numbers is necessarily empirical and cannot be entirely relied on, we assert an upper bound for the number of power wires to be removed in any iteration to conserve the amount of computation required in the wire sizing step discussed in Section 6.4.2. In our experiments, this upper bound is chosen to be around 6% of the total number of power wires. If a set of non-critical power wires are across one tile boundary, remove them in their increasing order of criticality. Through this rule, non-critical power wires with less criticality are removed first.

Fig. 6.4 illustrates a simple example of a power grid before and after a wire removal step. The three dark lines denote congested tile boundaries whose overflow values exceed OV_{th}. The overflow values on the three boundaries are proportional to the line widths shown in the figure. The two shaded areas on the power grid represent regions where the worst-case voltage droop is larger than VDP_{th}. There are a total of six wires, marked 1 through 6, that cross the congested boundaries. Since four of them (wires 1, 2, 4 and 6) are critical wires, they are not candidates for removal. The

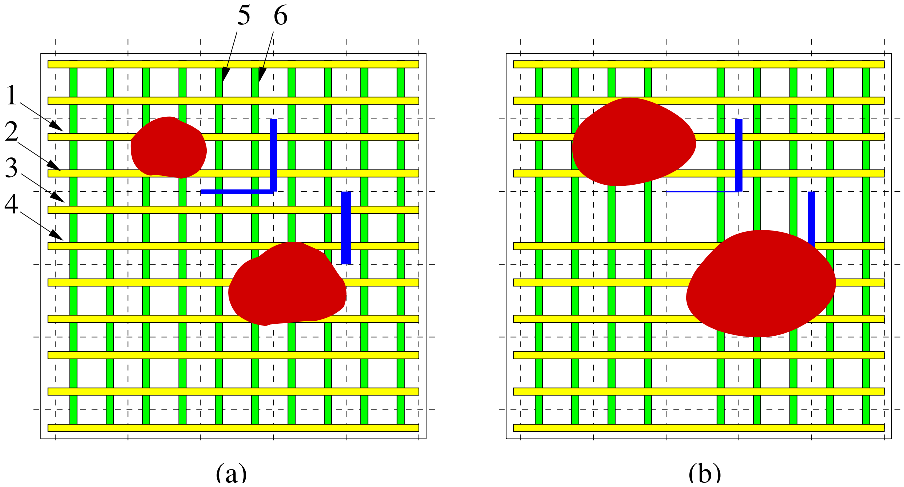

Figure 6.4: Heuristic power wire removal.

boundaries are processed in the order of their congestion, so that first wire 3, and then wire 5, is removed from the grid. After wire removal, the area of the shaded region is increased due to the increased voltage droop, and the overflow values on every tile edge traversed by wires 3 and 5 are accordingly updated, and this translates into dark edges of reduced thickness in Figure 3(b).

6.4.2 Power Grid Sizing

6.4.2.1 Problem Formulation

The second step in the adjustment of the power grid is related to sizing the wires in the grid to compensate for the increased voltage droop after the wire removal step. The problem is directly formulated as a nonlinearly constrained nonlinear programming problem as follows:

$$\begin{aligned} \text{minimize} \quad & Area(w_j) = \sum_{j=1}^{N_{wire}} l_j \times w_j \\ \text{subject to} \quad & w_{min} \leq w_j \leq w_{max},\ j = 1 \cdots N_{wire} \\ \text{and} \quad & Z(w_j) < \epsilon \end{aligned} \tag{6.6}$$

where ϵ is a very small number, and N_{wire} is the total number of wires in the grid.

The objective function that minimizes the total power wire area is consistent with the goal of congestion reduction. The first constraint restricts every power wire width to lie within a realistic range that is technology dependent. The second constraint requires total power grid noise Z (the same as is used in the decap placement work discussed in Chapter 4) to be near-zero.

In the context of power grid analysis the tunable circuit parameters p are the widths of the power grid wires. The noise Z for the entire circuit is defined as the weighted sum of all of the individual node metrics:

$$Z = \sum_{j=1}^{K} a_j z_j(p), \quad a_j = \frac{z_j}{\sum_{j=1}^{K} z_j}, \tag{6.7}$$

where K is the number of nodes, and the weight a_j magnifies node j with larger voltage droop noise. This metric finds the integral of the noise violation and is zero if all constraints are satisfied. It punishes larger violations more severely than minor violations, and, as pointed out earlier in this thesis, since it incorporates both the magnitude and time axes together, it is observed to be more practical than one that considers only the worst-case noise violation.

The nonlinear constraint function Z can be obtained by transient analysis of the power grid circuit, and its sensitivity with respect to all the variables w_j can be calculated using the adjoint method discussed in Section 6.3.

6.4.2.2 Optimization Scheme

We use a standard Sequential Quadratic Programming (SQP) solver [63, 64] to solve the optimization problem. This solver requires users to provide subroutines to evaluate the objective and constraint functions and their derivatives with respect to each

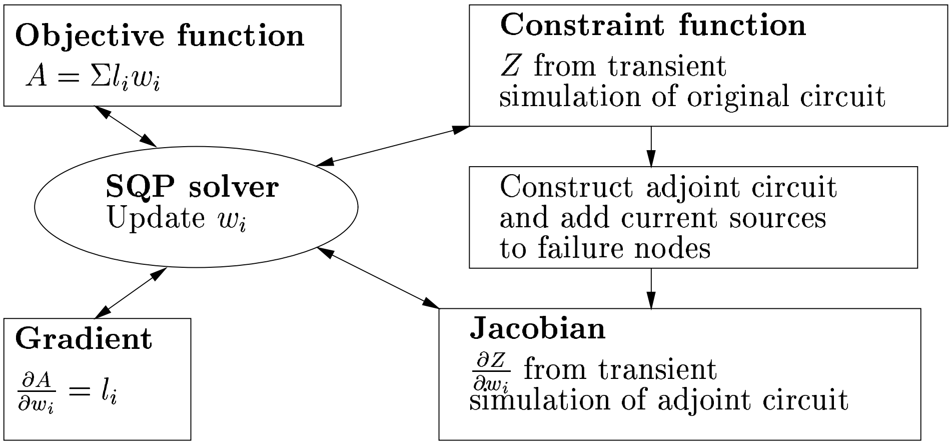
Objective function
$A = \Sigma l_i w_i$
Constraint function
Z from transient simulation of original circuit
SQP solver
Update w_i
Construct adjoint circuit and add current sources to failure nodes
Gradient
$\frac{\partial A}{\partial w_i} = l_i$
Jacobian
$\frac{\partial Z}{\partial w_i}$ from transient simulation of adjoint circuit

Figure 6.5: Power grid sizing procedure.

decision variable. The evaluations that are required for the SQP solver are illustrated in Figure 6.5.

Practically, it is observed that the SQP solver converges slowly near the optimum. However, an approximate convergence is sufficiently good for our purposes, so that we require Z to be near-zero rather than exactly zero by choosing a sufficiently small ϵ.

6.5 Overall Flow of the Algorithm

6.5.1 Overall Flow

Having described all of the individual pieces of the algorithm, we now outline the overall flow of the procedure. The starting point is a given tessellation of a chip and a given initial power grid construction, and the sequence of steps can be summarized as follows:

1) An initial power-grid aware global routing step is carried out to route the signal nets. Each net is first routed without considering congestion, followed by an iterative rip-up-and-reroute, described in Section 6.2, using the updated congestion information.

2) Based on the routes used by the power grid and the signal routes, a congestion map is generated and the overflow on each tile boundary is calculated. All tile boundaries whose overflow value exceeds the threshold OV_{th} are identified and are sorted in decreasing order of this value.

3) A transient simulation of the power grid is performed to identify critical power wires on which the worst-case voltage droop is above the threshold, VDP_{th}.

4) Sort the non-critical power wires in increasing order of the criticality.

5) Based on the congestion map generated in Step 2, non-critical power wires are removed according to the heuristic described in Section 6.4.1.

6) To compensate for the removal of these wires, the remaining power grid wires are resized using the nonlinear optimization procedure described in Section 6.4.2.

7) The congestion maps are updated, and the global routing is updated by performing rip-up-and-reroute based on the new congestions. At this point, the iterative loop is invoked so that the updated power grid and congestion map are fed back to step 3. The stopping criterion for the iteration is that the maximum wire density should be no greater than 1, or that no further improvement is possible. The latter is easily identified if it is detected that the changes in the congestion map after rip-up-and-reroute are insignificant, and that no further deletion of the power wires is possible.

6.5.2 Complexity Discussions

The initial routing takes $O(MN^3)$ time, where N is the number of pins for each net and M is the total number of nets. The rip-up-and-reroute has a complexity of $E \log \log E$, where E in our case is the total number of tile edges. The complexity of power wire removal is $O(KlogK + PlogP + KPE)$, where K is the total number of tile edges with negative overflow values, P is total number of power wires and E is total number of tile edges. The first term stands for the complexity of sorting these tile edges, the second term is the worst-case complexity of sorting the criticality of non-critical power wires, and the third term comes from the power wire removal and dynamic tile edge overflow update process. The power grid analysis and sensitivity analysis has the complexity of one LU decomposition and two forward/backward substitutions. In practice, the cost of these computations is just over $O(n)$ for a sparse positive definite matrix, where n is the total number of nodes and inductance branches. For a nonlinear optimization problem with w decision variables, advanced implementations of the SQP solver have $O(w^2)$ cost. Therefore the worst-case complexity for our wire sizing procedure ends up to be $O(I(n+w^2))$, where I is the total number of iterations. The efficiency of the gradient-based SQP solver relies largely on the initial solution and its distance away from an optimum. Practically, it is seen that the optimal solution can be reached in a limited number of iterations.

6.6 Experimental Results

The power grid analyzer and the global router have both been implemented in C++. The power grid removal and sizing scheme, and the overall congestion-driven power grid design and global routing flow has been written using Tcl. The wire size optimization is performed using an off-the-shelf SQP solver [64]. All experiments are performed on an Intel Pentium-IV 1.8GHz machine with 256M memory running Redhat

Linux 7.0. The entire procedure is encapsulated in a flow called PSiCo (**P**ower-**Si**gnal **Co**design).

We have tested our flow on seven benchmark circuits obtained from the authors of [54]. All designs correspond to a $0.18\mu m$ technology and a supply voltage of 1.8V. The time-varying current sources modeling the behavior of each functional block was not originally available in these benchmark circuits. These waveforms are constructed by modifying current waveforms from several industrial circuits by adjusting their magnitude according to the area of each block. However, this is not a critical assumption since our method is applicable to the exact waveforms where available.

Circuit	B	N	Tile size
apte	9	77	31x36
ami33	33	l12	30x28
ami49	49	368	33x34
playout	62	1294	30x26
ac3	27	200	38x36
hc7	77	430	34x39
a9c3	147	1148	33x31

Table 6.1: Test circuit parameters.

Initially, six layers of regularly distributed power grids with fixed wire widths are generated for all of these examples, with each layer containing only horizontal or vertical wires. Since a majority of the global routes are typically seen on the third and fourth metal layers, M3 and M4, we perform the global routing and power grid sizing on these two layers, assuming that the other layers are processed separately. The wire widths on these layers are assumed to be constrained within the range of $0.8\mu m$

and $4\mu m$ in our experiments. The initial power grid is constructed with a constant pitch in M3 and M4 such that under the given set of time-varying current sources that represent each block, the voltage droop on the entire power grid is within a threshold, which in our experiments, is chosen to be 0.18V. Table 6.1 lists the characteristics of each circuit, in terms of the total number of blocks B and nets N and the tile sizes.

Circuit	Method	D_{max}	Overflow
apte	Traditional	2.50	27
	PSiCo	1.00	1
ami33	Traditional	3.00	122
	PSiCo	1.00	0
ami49	Traditional	2.14	192
	PSiCo	1.00	0
playout	Traditional	1.94	158
	PSiCo	1.05	2
ac3	Traditional	2.50	316
	PSiCo	1.00	0
hc7	Traditional	2.13	518
	PSiCo	1.00	0
a9c3	Traditional	1.33	229
	PSiCo	1.00	0

Table 6.2: A comparison of the congestion improvement after global routing using the conventional approach and our new method.

The performance of PSiCo can be described in terms of two components: the global routing solution, and the performance of the final power grid, and these results are shown in Tables 6.2 and 6.3, respectively. The wire congestion results of PSiCo are compared in Table 6.2 against the results of the traditional rip-up-and-reroute method, which corresponds to the result at the end of Step 1 in Section 6.5. For each circuit, the maximum density D_{max}, calculated as the maximum ratio of the utilization to the capacity across any tile boundary. Also shown is the overflow, i.e., the total amount by which the tile boundary capacities are violated, summed over all boundaries. It can be seen that for all the cases PSiCo gives much better congestion results than the conventional method.

Circuit	Before Optimization				After Optimizatioin				
	Area	Nodes	W	Droop	Area	Nodes	W	Droop	Z (V × ns)
apte	16.8%	3411	133	0.178V	10.7%	3383	123	0.176V	2.84e-3
ami33	16.1%	11226	171	0.178V	8.90%	10245	156	0.179V	0.40e-3
ami49	19.3%	14986	198	0.178V	14.1%	13921	183	0.179V	3.65e-3
playout	19.3%	18981	223	0.170V	8.44%	16341	193	0.174V	0.10e-3
ac3	21.0%	4189	147	0.178V	9.38%	3985	128	0.180V	0.76e-3
hc7	22.4%	8882	215	0.180V	16.0%	8632	201	0.178V	7.05e-3
a9c3	19.3%	24168	251	0.175V	11.4%	22539	233	0.180V	0.85e-3

Table 6.3: Optimal power grid results.

Performance metrics for the initial power grid and the power grid obtained by PSiCo are listed in Table 6.3, in specific, the power wire area as a percentage of the area available on the two layers, the total number of nodes, the number of wires (W) in the power grid and the worst-case voltage droop. Theses numbers after optimization are also shown in the table and may be compared with the corresponding numbers before optimization. It can be seen that a well-designed power grid can save up to 12% of the total chip area while providing the optimal amount of resources needed for signal routing.

In each of the test cases, all grids whose worst-case voltage droop exceeds 0.14V (stricter than $10\%V_{dd}$ because layers M3 and M4 instead of M1 and M2 are considered) are marked as critical wires not to be removed. The voltage droop constraint (noise margin) for noise computation is chosen to be 0.17V, which is slightly stricter than $10\% V_{dd}$. To aid the rate of convergence, we heuristically terminate the SQP solver when we detect that the worst-case voltage droop in the circuit is below 0.18V, and it can be seen that the worst-case voltage droop always satisfies this requirement. The optimized noise integral, Z, for each circuit can be seen to be very small, and is nonzero due to the fact that voltage droops between 0.17V and 0.18V are tagged as "violations" by the SQP solver.

	apte	ami33	ami49	playout	ac3	hc7	a9c3
CPU (min)	3.6	7.6	9.8	18.1	6.0	7.3	28.8
# Iter.	6	2	1	5	2	3	2

Table 6.4: CPU time statistics and number of iterations for the benchmarks.

Lastly, Table 6.4 lists the total number of iterations and the CPU time, in minutes, required for each circuit. It can be seen that the number of iterations is typically small, and that the CPU times for these circuits are very reasonable.

Figure 6.6 shows the congestion map (38x36) of circuit ac3 after the initial routing. The overflow value on each tile boundary corresponds to the width of the dark line, so that congested regions are clearly identified. Through a transient simulation of the power grid circuit, the worst-case voltage droop for nodes on layers M3 and M4 are analyzed and shown as a contour plot in Figure 6.7. In Figure 6.7, only the ground plane is shown because our experiments show voltage droops in the ground plane are dominant for this circuit. Hot spots are plotted as darkest regions in the figure. The chip cell placement and the final optimal power grid are shown in Figure 6.8. By examining Figures 6.6 and 6.7 and the locations and widths of power wires in Figure 6.8 it can be seen that power wires away from hot spots and across congested tile boundaries have been removed and wider and/or denser power wires lie around hot spots. However, the widths of the wires cannot be seen easily in Figure 6.8 due to the limited resolution.

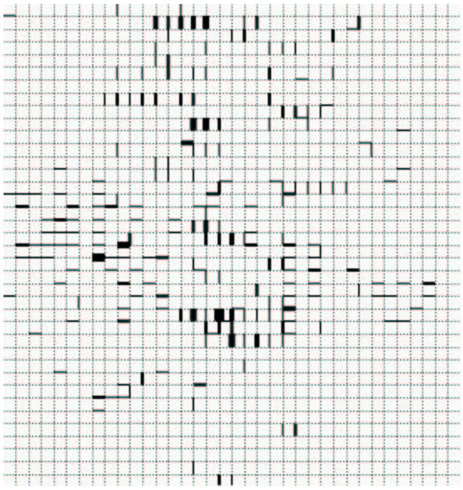

Figure 6.6: Congestioin map after the initial routing.

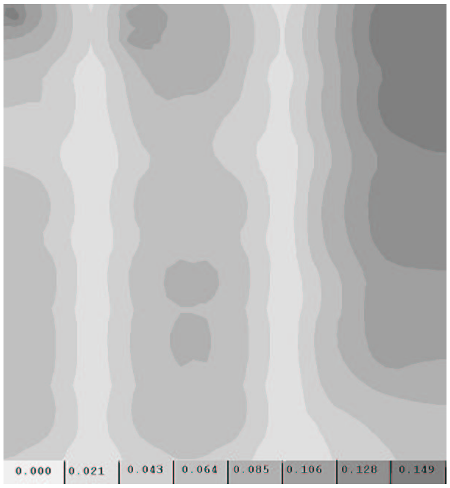

Figure 6.7: Voltage droop contour on the ground plane.

6.7 Conclusion

We have proposed a new design flow for the codesign of signal routes and power grids. The technique is guided by congestion maps, and proceeds by removing non-critical power wires greedily in congested areas, and rerouting the signal wires according to the updated congestions. The effects of removing power wires are compensated for by a gradient-based wire sizing scheme. Experimental results for several benchmark circuits are presented in this chapter. Future work includes applying this method to designs from industry to show the effectiveness of this method on larger industrial circuits.

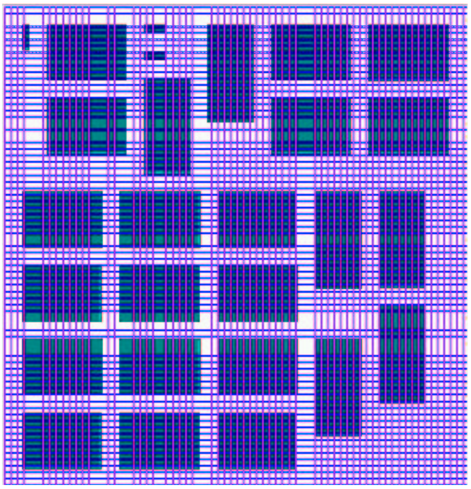

Figure 6.8: Optimal M3 and M4 power grid of ac3.

Chapter 7

Conclusions

This thesis emphasizes on various aspects of global interconnect performance optimization problems: in specific, analysis and optimization of power distribution networks (including decoupling capacitor sizing and placement), clock distribution network construction and optimization, and congestion-driven simultaneous optimization of power and signal networks. For power distribution networks, we use the sum of the voltage droop integrals below a user specified noise ceiling as the noise metric, which is very efficient for the optimization. The Elmore delay model is used for the initial construction of clock networks and analytical delay analysis technique on higher-order voltage waveforms is used in the accurate post-processing step.

We have explored the idea of designing hybrid mesh/tree structures in both power and clock networks and blended the advantages of full trees (easy for construction and analysis), and full meshes (reliable and robust in performance) structures.

For power distribution networks with an overlying mesh driving underlying trees, we have proposed a fast and efficient analyzer (HPRIMA) for transient analysis of voltage and current waveforms over multiple switching events. It hierarchically applies PRIMA to perform passive model-order reduction, incorporating nonzero initial conditions at the beginning of each event. We have extended the conventional transient adjoint sensitivity analysis for the variant circuit topologies over multiple switching events. Our sensitivity-based heuristic optimizer automatically sizes every wire width and decoupling capacitors and successfully suppresses voltage droop noise with the least amount of increase for the total wire and decap area. We also extended this structure to a two-level case where an overlying hybrid structure drives several underlying hybrid ones. Comparing to a one-level structure with the same total number of nodes and loads, the optimization for the two-level structure works faster and more robustly.

Next, we addressed a special issue in power grid design - decoupling capacitor sizing and placement in the context of standard-cell layouts. Our scheme makes a judicious use of empty spaces available in every row after the placement phase for the design. We have proposed a piece-wise linear waveform compression technique which achieves linear complexity for the waveform convolution computation during the adjoint sensitivity analysis. We applied the gradient-based sequential quadratic programming (SQP) solver to minimize the noise subject to the space constraints in each row. This scheme was demonstrated on several industrial designs.

The study of the hybrid structure in our clock network design began with the construction of a zero-skew mesh structure. We developed a linear-programming based zero-skew mesh construction algorithm. We also proposed wire elongation heuristics for the linear program to find guaranteed feasible zero-skew solutions, and buffer insertion and mesh shrinking heuristics to achieve required transition time. Our hybrid clock network consists of an overlying zero-skew mesh driving underlying buffer clock trees. A heuristic skew and transition time optimizer based on the higher-order delay model has been proposed and demonstrated. We have made several interesting performance comparisons for various clock structures.

Finally, a new design flow (PSiCo) for the codesign of power and signal networks has been developed. Tesselation is used to divide the entire chip into smaller regions. We use overflow values on every tile boundary to form a congestion map and use this to measure the performance of the wiring design. The congestion map in our work incorporates both the effects of power and signal wires. The whole scheme is congestion-driven, i.e., global signal routing is iteratively performed according to the congestion map and our proposed power wire removal heuristics are also based on these congestion numbers. A postprocessing power wire sizing scheme has been proposed to compensate the performance loss after wire removal. Our new flow proves to be able to make the best use of limited wiring resources.

Bibliography

[1] G. E. Moore, “Cramming more components onto intergrated circuits,” *Electronics Magazine*, vol. 38, pp. 114–117, April 1965.

[2] *The International Technology Roadmap for Semiconductors.* *http : //public.itrs.net/Files/*2001*ITRS/Home.html*: Semiconductor Industry Association, 2001.

[3] G. Bai, S. Bobba, and T. N. Hajj, “Static Timing Analysis Including Power Supply Noise Effect on Propagation Delay in VLSI Circuits,” in *Proc. Design Automation Conference*, (Las Vegas, NV), pp. 295–300, June 2001.

[4] K. Bernstein, K. M. Carrig, C. M. Durham, P. R. Hansen, D. Hogenmiller, E. J. Nowak, and N. J. Rohrer, *High Speed CMOS Design Styles.* Norwell, MA: Kluwer Academic Publishers, 1998.

[5] A. Odabasioglu, M. Celik, and L. T. Pilleggi, “PRIMA: Passive Reduced-order Interconnect Macromodeling Algorithm,” in *Proc. International Conference on Computer-Aided Design*, (San Jose, CA), pp. 645–654, November 1998.

[6] J. P. Fishburn and A. E. Dunlop, “TILOS: A Posynomial Programming Approach to Transistor Sizing,” in *Proc. International Conference on Computer-Aided Design*, (San Jose, CA), pp. 326–328, November 1985.

[7] J. P. Fishburn, "Clock Skew Optimization," *IEEE Transactions on Computers*, vol. 39, pp. 945–951, July 1990.

[8] H. Su, K. H. Gala, and S. S. Sapatnekar, "Fast Analysis and Optimization of Power/Ground Networks," in *Proc. International Conference on Computer-Aided Design*, (San Jose, CA), pp. 477–480, November 2000.

[9] H. Su and S. S. Sapatnekar, "Hybrid Structured and Clock Network Construction," in *Proc. International Conference on Computer-Aided Design*, (San Jose, CA), pp. 333–337, November 2001.

[10] H. Su, S. S. Sapatnekar, and S. R. Nassif, "An Algorithm for Optimal Decoupling Capacitor Sizing and Placement for Standard Cell Layouts," *to appear in International Symposium on Physical Design*, 2002.

[11] H. Su, J. Hu, S. S. Sapatnekar, and S. R. Nassif, "Congestion-driven Codesign of Power and Signal Networks," *Submitted to Design Automation Conference*, 2002.

[12] H. Su, K. H. Gala, and S. S. Sapatnekar, "Analysis and Optimizatioin of Structured Power/Ground Networks," *Submitted to IEEE Transactions on Computer-Aided Design.*

[13] H. Su and S. S. Sapatnekar, "Clock Network Construction using Hybrid Mesh-Tree Structures," *Submitted to IEEE Transactions on Computer-Aided Design.*

[14] H. H. Chen and D. D. Ling, "Power Supply Noise Analysis Methodology for Deep-Submicron VLSI Chip Design," in *Proc. Design Automation Conference*, (Anaheim, CA), pp. 638–643, June 1997.

[15] H. H. Chen and J. S. Neely, "Interconnect and Circuit Modeling Techniques for Full-Chip Power Supply Noise Analysis," *IEEE Transactions on Components,*

Packaging, and Manufacturing Technology, Part B, vol. 21, pp. 209–215, August 1998.

[16] L. T. Pillage, R. A. Rohrer, and C. Visweswariah, *Electronic and System Simulation Methods.* New York, NY: McGraw-Hill, 1995.

[17] M. Zhao, R. V. Panda, S. S. Sapatnekar, T. Edwards, R. Chaudhry, and D. Blaauw, “Hierarchical Analysis of Power Distribution Networks,” in *Proc. Design Automation Conference*, (Los Angeles, CA), pp. 481–486, June 2000.

[18] S. R. Nassif and J. N. Kozhaya, “Fast Power Grid Simulation,” in *Proc. Design Automation Conference*, (Los Angeles, CA), pp. 156–161, June 2000.

[19] A. R. Conn, R. A. Haring, and C. Visweswariah, “Noise Considerations in Circuit Optimization,” in *Proc. International Conference on Computer-Aided Design*, (San Jose, CA), pp. 220–227, November 1998.

[20] W. C. Elmore, “The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers,” *Journal of Applied Physics*, vol. 19, pp. 55–63, January 1948.

[21] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins, “Near-optimal Critical Sink Routing Tree Construction,” *IEEE Transactions on Computer-Aided Design*, vol. 14, pp. 1417–1436, December 1995.

[22] J. Qian, S. Pullela, and L. T. Pillage, “Modeling the Effective Capacitance for the RC Interconnect of CMOS Gates,” *IEEE Transactions on Computer-Aided Design*, vol. 13, pp. 1526–1535, December 1994.

[23] S. W. Director and R. A. Rohrer, “The Generalized Adjoint Network and Network Sensitivities,” *IEEE Transactions on Circuit Theory*, vol. 16, pp. 318–323, August 1969.

[24] P. Feldmann, T. V. Nguyen, S. W. Director, and R. A. Rohrer, "Sensitivity Computation in Piecewise Approximation Circuit Simulation," *IEEE Transactions on Computer-Aided Design of ICs and Systems*, vol. 10, pp. 171–183, February 1991.

[25] D. Stark and M. Horowitz, "Techniques for Calculating Currents and Voltages in VLSI Power Supply Networks," *IEEE Transactions on Computer-Aided Design of ICs and Systems*, vol. 9, pp. 126–132, February 1979.

[26] S. Chowdhry and J. S. Barkatullah, "Estimation of Maximum Currents in MOS IC Logic Circuits," *IEEE Transactions on Computer-Aided Design*, vol. 9, pp. 642–654, June 1990.

[27] A. Dharchoudhury, R. Panda, D. Blaauw, and R. Vaidyanathan, "Design and Analysis of Power Distribution Networks in PowerPC^{TM} Microprocessors," in *Proc. Design Automation Conference*, (San Francisco, CA), pp. 738–743, June 1998.

[28] B. R. Stanisic, N. K. Verghese, R. A. Rutenbar, L. R. Carley, and D. J. Allstot, "Addressing Substrate Coupling in Mixed-mode IC's: Simulation and Power Distribution Synthesis," *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 226–238, March 1994.

[29] B. R. Stanisic, R. A. Rutenbar, and L. R. Carley, "Addressing Noise Decoupling in Mixed-signal IC's: Power Distribution design and Cell Customization," *IEEE Journal of Solid-State Circuits*, vol. 30, pp. 321–326, March 1995.

[30] X. Tan, C. J. R. Shi, D. Lungeanu, J. Lee, and L. Yuan, "Reliability-Constrained Area Optimization of VLSI Power/Ground Networks Via Sequence of Linear Programmings," in *Proc. Design Automation Conference*, (New Orleans, LA), pp. 156–161, June 1999.

[31] T. Mitsuhashi and E. S. Kuh, "Power and Ground Network Topology Optimization for Cell-based VLSIs," in *Proc. Design Automation Conference*, (Anaheim, CA), pp. 524–529, June 1992.

[32] G. Bai, S. Bobba, and I. N. Hajj, "Simulation and Optimization of the Power Distribution Network in VLSI Circuits," in *Proc. International Conference on Computer-Aided Design*, (San Jose, CA), pp. 481–486, November 2000.

[33] S. Boyd, L. Vandenberghe, A. E. Gamal, and S. Yun, "Design of Robust Global Power and Ground Networks," in *Proc. International Symposium on Physical Design*, (Napa, CA), pp. 60–65, April 2001.

[34] J. C. Shah, A. A. Younis, S. S. Sapatnekar, and M. M. Hassoun, "An Algorithm for Simulating Power/Ground Networks Using Padé Approximants and its Symbolic Implementation," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 45, pp. 1372–1382, October 1998.

[35] P. Feldmann, T. V. Nguyen, S. W. Director, and R. A. Rohrer, "Sensitivity Computation in Piecewise Approximate Circuit Simulation," *IEEE Transactions on Computer-Aided Design*, vol. 10, pp. 171–183, February 1991.

[36] C. L. Ratzlaff, N. Gopal, and L. T. Pillage, "RICE: Rapid Interconnect Circuit Evaluator," in *Proc. Design Automation Conference*, (San Francisco, CA), pp. 555–561, June 1991.

[37] W. Rudin, *Real and Complex Analysis.* New York, NY: McGraw-Hill, 1987.

[38] W. J. Bowhill, R. L. Allmon, and S. L. Bell, "A 300MHz 64b Quad-Issue CMOS RISC Microprocessor," in *Proc. International Solid-State Circuits Conference*, (Piscataway, NJ), pp. 182–183, February 1995.

[39] D. W. Dobberpuhl, R. T. Witek, and R. Allmon, "A 200-MHz 64-b Dual-Issue CMOS Microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 1555–1567, November 1992.

[40] S. Zhao, K. Roy, and C.-K. Koh, "Decoupling Capacitance Allocation for Power Supply Noise Suppression," in *Proc. International Symposium on Physical Design*, (Napa, CA), pp. 66–71, April 2001.

[41] C. Zhu, R. H. Byrd, and J. Nocedal, *LBFGS-B: Fortran subroutines for large-scale bound constrained optimization.* EECS Department, Northwestern University, 1994.

[42] R. Burch, F. Najam, P. Yang, and T. Trick, "McPOWER: A Monte Carlo Approach to Power Estimation," in *Proc. International Conference on Computer-Aided Design*, (San Jose, CA), pp. 90–96, November 1992.

[43] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, "Clock Routing for High-Performance IC's," in *Proc. Design Automation Conference*, (Orlando, FL), pp. 573–579, June 1990.

[44] A. B. Kahng, J. Cong, and G. Robins, "High-Performance Clock Routing Based on Recursive Geometric Matching," in *Proc. International Conference on Computer-Aided Design*, (San Jose, CA), pp. 322–327, November 1991.

[45] R. Tsay, "An Exact Zero-Skew Clock Routing Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, pp. 242–249, February 1993.

[46] S. Lin and C. K. Wong, "Process-Variation-Tolerant Clock Skew Minimization," in *Proc. International Conference on Computer-Aided Design*, (San Jose, CA), pp. 284–288, November 1994.

[47] N. Menezes, A. Balivada, S. Pullela, and L. T. Pillage, “Skew Reduction in Clock Trees Using Wire Width Optimization,” in *Proc. of the IEEE Custom Integrated Circuits Conference*, (San Diego, CA), pp. 9.6.1–9.6.4, May 1993.

[48] S. Pullela, N. Menezes, J. Omar, and L. T. Pillage, “Skew and Delay Optimization for Reliable Buffered Clock Trees,” in *Proc. International Conference on Computer-Aided Design*, (San Jose, CA), pp. 556–562, November 1993.

[49] S. Pullela, N. Menezes, and L. T. Pillage, “Reliable Non-zero Skew Clock Trees Using Wire Width Optimization,” in *Proc. Design Automation Conference*, (Dallas, TX), pp. 165–170, June 1993.

[50] M. P. Desai, R. Cvijetic, and J. Jensen, “Sizing of Clock Distribution Networks for High Performance CPU Chips,” in *Proc. Design Automation Conference*, (Las Vegas, NV), pp. 389–394, June 1996.

[51] P. J. Restle, A. E. Ruehli, and S. G. Walker, “Multi-GHz Interconnect Effects in Microprocessors,” in *Proc. International Symposium on Physical Design*, (Napa, CA), pp. 93–97, April 2001.

[52] L. O. Chua and P.-M. Lin, *Computer-Aided Analysis of Electronic Circuits.* Englewod Cliffs, NJ: Prenticle-Hall, 1975.

[53] A. R. Conn, R. A. Haring, C. Visweswariah, and C. W. Wu, “Circuit Optimization via Adjoint Lagrangians,” in *Proc. International Conference on Computer-Aided Design*, (San Jose, CA), pp. 281–288, November 1997.

[54] C. J. Alpert, J. Hu, S. S. Sapatneker, and P. G. Villarrubia, “A Practical Methodology for Early Buffer and Wire Resource Allocation,” in *Proc. Design Automation Conference*, (Las Vegas, NV), pp. 189–194, June 2001.

[55] C. Chiang and M. Sarrafzadeh, “Global Routing Based on Steiner Min-max Tree,” *IEEE Transactions on Computer-Aided Design*, vol. 9, pp. 1318–1325, December 1990.

[56] B. S. Ting and B. N. Tien, “Routing and Techniques for Gate Array,” *IEEE Transactions on Computer-Aided Design*, vol. 2, pp. 301–312, October 1983.

[57] E. Shragowitz and S. Keel, “A global router based on a multicommodity flow model,” *Intergration: the VLSI Journal*, vol. 5, pp. 3–16, March 1987.

[58] J. Hu and S. S. Sapatnekar, “A timing-constrained algorithm for simultaneous routing of multiple nets,” in *Proc. International Conference on Computer-Aided Design*, (San Jose), pp. 99–103, November 2000.

[59] X. Wu, X. Hong, Y. Cai, C. K. Cheng, J. Gu, and W. Dai, “Area Minimization of Power Distribution Network using Efficient Nonlinear Programming Techniques,” in *Proc. International Conference on Computer-Aided Design*, (San Jose), pp. 153–157, CA 2001.

[60] J. D. Z. Ma and L. He, “Simultaneous Signal and Power Routing under Keff Model,” in *Proc. International Workshop on System-Level Interconnect Prediction*, (Rohnert Park, CA), pp. 175–182, April 2001.

[61] C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng, and D. Karger, “Prim-Dijkstra Tradeoffs for Improved Performance-Driven Routing Tree Design,” *IEEE Transactions on Computer-Aided Design of ICs and Systems*, vol. 14, pp. 890–896, July 1995.

[62] R. Nair, “A Simple Yet Effective Technique for Global Wiring,” *IEEE Transactions on Computer-Aided Design*, vol. 6, pp. 165–172, March 1987.

[63] P. T. Boggs and J. W. Tolle, *Sequential Quadratic Programming.* Cambridge: Cambridge Univ. Press, 1995.

[64] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for SOL/QPSOL: A Fortran Package for Quadratic Programming.* Department of Operations Research, Stanford University, July, 1983.