

Copyright  
by  
Hung-Ming Chen  
2003

The Dissertation Committee for Hung-Ming Chen  
certifies that this is the approved version of the following dissertation:

## **Algorithms for VLSI Design Planning**

Committee:

---

Aloysius K. Mok, Supervisor

---

Martin D.F. Wong, Supervisor

---

Donald S. Fussell

---

Mohamed G. Gouda

---

Xiaoping Tang

---

Hai Zhou

**Algorithms for VLSI Design Planning**

**by**

**Hung-Ming Chen, B.E., M.S.**

**DISSERTATION**

Presented to the Faculty of the Graduate School of  
The University of Texas at Austin  
in Partial Fulfillment  
of the Requirements  
for the Degree of

**DOCTOR OF PHILOSOPHY**

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2003

Dedicated to my family.

## Acknowledgments

I am greatly indebted to my dissertation advisor Professor Martin D.F. Wong for bringing me to the area of VLSI design automation. His guidance and encouragement have been a tremendous assistance throughout my study. The fruitful discussions with him have inspired me to many stimulating thoughts for challenging problems. The criticism from him has led to a higher quality of my research work as well.

I would like to thank my co-advisor Professor Aloysius K. Mok and other members of my dissertation committee, Professors Mohamed G. Gouda, Donald S. Fussell, Hai Zhou, and Dr. Xiaoping Tang for their interests and constructive comments to my work. Special thanks go to Professor Mohamed G. Gouda, who provided me some useful thoughts both in research and living.

I thank all the members of the CAD group in UTCS, specifically Xia Chen, Yongseok Cheon, Li-Da Huang, Seokjin Lee, I-Chung Liu, Muzhou Shao, Yu Sun, and Gang Xu for helpful communications. Also I would like to thank Gloria Ramirez and Katherine Utz at the department's Graduate Office for their great help in administrative issues. In addition, I would like to thank other people who are outside of UTCS for their professional suggestions for my research work, including Professor Yaowen Chang from National Taiwan University, Dr. I-Min Liu and Dr. Minghorng Lai from Cadence Design Systems, and Hua Xiang from University of

Illinois at Urbana-Champaign.

I thank all my friends who have given me a pleasant time staying in Austin, including Chin-Tser Huang family, Hongming Yeh family, Li-San Wang family, Chloe Chang, Eric Lin family, Kiki Wang family, Joyce Wu, and Heng-Ru Lin.

Finally, I would like to give my wholehearted thanks to my parents for their love and care that I have consistently received throughout my life. Especially I would like to express my love and appreciation to my dearest wife, Yi-Chien Yang, for her understanding, encouragement and comfort in all these years. I couldn't reach such achievement without her love.

# Algorithms for VLSI Design Planning

Publication No. \_\_\_\_\_

Hung-Ming Chen, Ph.D.

The University of Texas at Austin, 2003

Supervisors: Aloysius K. Mok  
Martin D.F. Wong

With shrinking feature sizes, much more transistors can be integrated on a single chip. Moore's Law has been followed closely in the past decades, resulting in larger and faster chips every year. In order to design larger and faster chips in deep submicron (DSM) technology, it is necessary to perform early design planning. In this dissertation, we present several algorithms for a number of VLSI design planning problems.

First, we propose a method to integrate interconnect planning with floorplanning. Our approach is based on the Wong-Liu floorplanning algorithm. We perform pin assignment and fast global routing during every iteration of floorplanning. We use a multi-stage simulated annealing approach in which different interconnect planning methods are used in different ranges of temperatures to reduce running time. A temperature adjustment scheme is designed to give smooth transitions between different stages of simulated annealing.

Second, floorplanning problems typically have relatively small number of blocks (e.g., 50-100) but have a large number of nets (e.g. 20K). Since existing

floorplanning algorithms use simulated annealing which needs to examine a large number of floorplans, this has made interconnect-centric floorplanning computationally very expensive. We present approaches that can dramatically improve the run time of problems with large number of nets and at the same time improve solution quality.

Third, we propose a method for simultaneous power supply planning and noise avoidance in floorplan design. Without careful power supply planning in layout design, the design of chips will suffer from mostly signal integrity problems including IR-drop,  $\Delta I$  noise, and IC reliability. Post-route methodologies in solving signal integrity problem have been applied but they will cause a long turn-around time, which adds costly delays to time-to-market. We show that the noise avoidance in power supply planning problem can be formulated as a constrained maximum flow problem.

Fourth, I/O placement has been a concern in modern IC design. Due to flip-chip and multi-chip module technologies, I/O can be placed throughout the whole chip without long wires from the periphery of the chip. However, because of I/O placement constraints and I/O buffer site building cost, the decision of positions for placing I/O buffers has become critical. Our objective is to reduce the number of I/O buffer sites and to decide their positions in an existing standard cell placement. We formulate it as a minimum cost flow problem.

# Table of Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>Chapter 1. Introduction</b>	<b>1</b>
1.1 Dissertation Overview . . . . .	4
<b>Chapter 2. Integrated Floorplanning and Interconnect Planning</b>	<b>7</b>
2.1 Introduction . . . . .	7
2.2 Efficient Interconnect Planning . . . . .	10
2.2.1 Pin Assignment . . . . .	11
2.2.2 Simple-Geometry Routing . . . . .	11
2.2.3 Incremental Routing Cost Computation . . . . .	15
2.3 Multi-Stage Simulated Annealing . . . . .	18
2.3.1 Cost Function Transitions . . . . .	19
2.3.2 Temperature Adjustment . . . . .	21
2.4 Experimental Results . . . . .	24
2.5 Concluding Remarks . . . . .	25
<b>Chapter 3. Faster and More Accurate Wiring Evaluation in Interconnect-Centric Floorplanning</b>	<b>28</b>
3.1 Introduction . . . . .	29
3.2 Wiring Evaluation in Interconnect Centric Floorplanning . . . . .	31
3.3 Faster Wiring Evaluation Based on Net Reduction . . . . .	33
3.3.1 Problem Formulation . . . . .	33

3.3.2	Algorithm for Net Reduction . . . . .	35
3.3.2.1	Bounded-Degree Hypergraph-to-Graph Transformation . . . . .	36
3.3.2.2	Net Reduction by Multi-terminal Net Decomposition . . . . .	38
3.4	More Accurate Wiring Evaluation Based on Lagrangian Relaxation . . . . .	41
3.4.1	Problem Formulation . . . . .	41
3.4.2	Lagrangian Relaxation . . . . .	41
3.4.2.1	Simplification of Lagrangian Relaxation Subproblem . . . . .	42
3.4.2.2	Solving Lagrangian Relaxation Subproblem . . . . .	43
3.4.2.3	Solving Lagrangian Dual Problem . . . . .	44
3.5	Experimental Results . . . . .	44
3.6	Concluding Remarks . . . . .	45

**Chapter 4. Simultaneous Power Supply Planning and Noise Avoidance in Floorplan Design 47**

4.1	Introduction . . . . .	48
4.2	Floorplans with Power Supply Noise Considerations . . . . .	51
4.2.1	IR-Drop Requirement . . . . .	55
4.2.2	$\Delta I$ Noise Constraint . . . . .	55
4.2.3	Problem Formulation . . . . .	57
4.3	Power Supply Planning with Noise Avoidance . . . . .	58
4.3.1	Feasible Power Supply Region . . . . .	59
4.3.2	Constrained Network Formulation . . . . .	60
4.3.3	Priority-Augmenting-Path Algorithm . . . . .	63
4.3.3.1	Complexity Analysis . . . . .	64
4.3.4	Graph Reduction by Power Zones . . . . .	65
4.4	Floorplanning with Power Supply Planning and Noise Avoidance Design . . . . .	67
4.5	Experimental Results . . . . .	68
4.6	Concluding Remarks . . . . .	70

<b>Chapter 5. I/O Buffer Site Placement in Area-Array IC Design</b>	<b>73</b>
5.1 Introduction . . . . .	74
5.2 I/O Buffer Site Placement in Area-Array IC Design . . . . .	76
5.3 The Algorithm . . . . .	81
5.4 Cell Re-Placement from I/O Buffer Site Planning . . . . .	84
5.5 Experimental Results . . . . .	85
5.6 Concluding Remarks . . . . .	87
<b>Chapter 6. Conclusion and Future Directions</b>	<b>89</b>
<b>Bibliography</b>	<b>94</b>
<b>Vita</b>	<b>104</b>

## List of Tables

2.1	Experimental results of our approach on MCNC examples, compared with the method in [65] . . . . .	25
2.2	Performance improvement of our approach on MCNC examples, compared with the method in [65] . . . . .	25
3.1	Speedup comparisons between floorplanner in [2] and our approaches using net reduction and Lagrangian relaxation global router. . . . .	45
4.1	Comparison of our approach with [65] and [35] on MCNC benchmarks. The wirelength data are described in Section 4.5. . . . .	69
5.1	Number of cells, nets, and IO/terminals in some MCNC standard cell placement benchmarks. . . . .	86
5.2	Experimental results of our approach on MCNC benchmarks summarized in Table 5.1, compared with a greedy approach. With slight increase percentage in voltage drop threshold violation, much less number of I/O buffer sites can be obtained in minimizing wirelength. . . . .	87
5.3	Performance improvement of our approach on MCNC benchmarks, compared with a greedy approach [61]. . . . .	88

## List of Figures

2.1	Floorplanning greatly influences interconnect structure . . . . .	8
2.2	Floorplanning and interconnect planning . . . . .	9
2.3	Pin assignment illustration . . . . .	12
2.4	L-shaped routing . . . . .	13
2.5	Z-shaped routing . . . . .	15
2.6	$v(i, j)$ is the accumulated cost of shaded bin boundaries . . . . .	16
2.7	$h(i, j)$ is the accumulated cost of shaded bin boundaries . . . . .	17
2.8	The routing cost in terms of $h(i, j)$ 's and $v(i, j)$ 's . . . . .	17
2.9	Normal curve of cost versus number of iterations . . . . .	20
2.10	Abnormal curve (solid line) of cost versus number of iterations: quickly converges to suboptimal solution . . . . .	21
2.11	Abnormal curve (solid line) of cost versus number of iterations: takes a much longer time to converge . . . . .	22
2.12	Temperature adjustment . . . . .	23
2.13	Result of packing ami49 using interconnect planning approach . . . . .	26
2.14	Result of packing ami49 using original approach . . . . .	26
3.1	Net reduction (a) Original netlist. (b) Reduced netlist. . . . .	34
3.2	A netlist with multi-terminal nets. . . . .	34
3.3	Two net decomposition solutions for the example in Figure 3.2. (a) An illegal net decomposition since block $b_4$ exceeds its pin-limit. (b) A feasible net decomposition. . . . .	35
3.4	Hypergraph-to-graph transformation. (a)A hypergraph with four hyperedges: $e_1$ , $e_2$ , $e_3$ , and $e_4$ . (b)A graph formed by combining four spanning trees corresponding to the four hyperedges in (a). . . . .	37
3.5	(a)Flow network built upon the hypergraph model. (b) An integral maximum flow. The number besides each arc is the (a)capacity/(b)flow in the arc. . . . .	40
3.6	Spanning trees for hyperedges $e_2$ and $e_3$ . . . . .	40

4.1	C4 chip patterns from IBM online document. . . . .	49
4.2	$\Delta I$ noise constraint illustration when sharing power sources. Two examples of two blocks with a power supply bump. The V-t curves are voltage fluctuations for blocks and the superposition seen in power supply bump. (a) The power supply bump is clean since the voltage fluctuations of two blocks are within the upper bound on $\Delta V$ for both blocks. (b) The power supply bump is noisy since the voltage fluctuations of block C and D exceeds the upper bound on $\Delta V$ for block D. . . . .	53
4.3	Floorplanning affects power supply planning and noise avoidance. Block $b_3$ can get four power supply bumps to deliver power on the left floorplan while it can possibly only get three power supply bumps on the right one. This block may also suffer from $\Delta I$ noise constraint violation. . . . .	54
4.4	The relationship between power supply bump and circuit block. The circuit block can obtain power from several power supply bumps, as long as the noise constraint holds. . . . .	57
4.5	A floorplan and the available power supply bumps. A circuit block can use the power supply bumps within its feasible power supply region (FPSR). . . . .	59
4.6	The network graph captures the power demand of the circuit blocks and the power that the amount of power supply bumps can provide in Figure 4.5. . . . .	61
4.7	Numeric examples include two max-flow solutions of the network graph from Figure 4.6. Those number are calculated from technology and given IP parameters. (a) The solution with randomly choosing augmenting path. The darker numbers and the edge show that there is a $\Delta I$ noise constraint violation. The number on the edge is the amount of flow on that edge. The number inside the parentheses on the edges between power supply bumps and blocks is the amount of inductive induced voltage drop on that edge. The number inside the parentheses above the block node is the upper bound on $\Delta V$ for the block. For example, $e(p1, b1)$ has 0.15mV for inductive induced voltage drop, which does not exceed $\Delta V_1=0.23mV$ . But for $e(p3, b2)$ , it has 0.3mV, which exceeds $\Delta V_2=0.23mV$ , indicating a violation. (b) The solution using the algorithm in Section 4.3.3. There is no $\Delta I$ noise constraint violation. . . . .	71
4.8	The power zones in a floorplan. . . . .	72
5.1	Area-array footprint ASIC. The Vdd and Gnd bumps are uniformly distributed across the die with signal bumps in fixed interspersed locations. I/O buffers are associated with some specified signals bump and connected by pad transfer metal. . . . .	75

5.2	Power supply network in area-array design for efficient analysis. Power grids are modeled as linear RC networks, power sources are modeled as simple constant voltage sources, and power drains are modeled as independent time-varying currents. . . . .	77
5.3	The relationship between signal bump, power bump, power bump bin, I/O buffer possible positions, and possible current drawn region.	80
5.4	Network construction for IBSP. Some signal bump (corresponding I/O buffer) vertex $io_i$ only connects to power bump bin vertices which are inside the possible current drawn region for $io_i$ . . . . .	81
5.5	Vertex splitting for capacitated vertices. The new edge has capacity $U(r)$ and cost 0 [66]. . . . .	82
5.6	Force-directed based cell re-placement. With fixed I/O buffer locations, use force attraction to refine existing placement. . . . .	85

# Chapter 1

## Introduction

With the rapid development of Very Large Scale Integration (VLSI) fabrication technologies, we have reached an era where the minimum feature sizes of the leading processes is well below  $.25 \mu\text{m}$ . Such processes are called deep sub-micron (DSM) processes. With shrinking feature sizes, much more transistors can be integrated on a single chip performing different tasks pre-designed in intellectual properties (IPs). Moore's Law has been followed closely and a huge value in the electronic market has been introduced. However, at the same time, many new problems arise. Those problems will force the change of traditional design flow in electronic design automation (EDA). EDA is one of the key enablers of the semiconductor industry and no chip is designed without EDA [38, 50]. In fact, semiconductors (or technologies) drive EDA technology, especially in physical design. Synthesis, placement and routing are enabled by multiple technology constraints, for example, a logic library, usually in the form of standard cells of the same height and similar size which simplifies placement and routing, decouples technology from logic and enables synthesis. Furthermore, not long ago interconnect was assumed to have negligible delay and digital circuits were modeled with "lumped" components so that secondary effects like crosstalk could not be analyzed. Power consumption issues were limited to the power grid. All that has changed.

With rapid feature size scaling, the circuit performance is increasingly determined by the interconnects instead of devices [19]. Although the use of new interconnect materials (e.g., copper) is helpful in reducing interconnect delay, they do not provide the ultimate solution to the increasing performance mismatch between devices and interconnects. Even with the projected improvements in interconnect performance, the global interconnects remain the performance bottleneck. In conventional VLSI designs, much emphasis has been to design and optimization of logic and devices. The interconnection was done by either layout designers or automatic place-and-route tools as an afterthought. Now we need an interconnect-centric design flow for further interconnect planning throughout the physical design process.

In physical design phase, the typical netlist usually contains large number of nets. Also since for many designs under  $0.25 \mu\text{m}$  interconnect delay actually becomes larger than circuit delay [19], wire delays cannot be ignored in logic design any more. Moreover, as the technology node approaches  $.10 \mu\text{m}$ , inductance becomes noticeable for long wires at high speed such as busses of several  $mm$  of length operating at more than 1-2 GHz [24]. This is already a problem for microprocessor design. Current solutions do placement and synthesis (or re-synthesis) together, or start with wire planning, so that wire behavior can be more accurately estimated from early on in the design process. Therefore, a good quality of floorplan with large number of global interconnects is absolutely desired.

Generating satisfactory power supply to meet the requirements of different components in a single chip is becoming difficult in DSM regime due to reduced

power supply voltage, tighter noise margin, and DC voltage drop [7]. Among the approaches of handling/estimating power delivery [44], the planning of mesh power rail followed by hierarchical power/ground (P/G) networks designs is still a major method to design high performance integrated circuits (ICs) [3, 62, 68, 69]. Nevertheless, post-floorplanning power supply synthesis alone cannot guarantee high-quality power supply under limited routing resources. In many cases, when the circuit block locations and sizes are fixed, the constraints such as voltage drop and current density are so tight that there is no feasible power network design capable of keeping power supply noise within a specified margin. In addition, during manufacturing, the number of silicon failures are caused by signal integrity problems, such as IR-drop,  $\Delta I$  noise, and electromigration. These problems are on the rise due to the lack of existing design tools and methodologies to address these issues effectively.

In order to keep up the performance in technology advances, flip-chip and multi-chip module (MCM) technologies now allows high-performance ICs and microprocessors to be built with many more power and I/O connections than in the past, among which area array bonding is considered a rather better one. Besides helping solve the power delivery engineering problems, to effectively alleviate voltage drop problem we need to focus on the placement of highly power hungry buffers, I/O buffers. Since area-array style allows I/O buffers to be placed anywhere on the die, we need to be aware of I/O buffer placement constraints to better the design.

## 1.1 Dissertation Overview

We investigate and develop several algorithms in VLSI design planning. We first describe several results that we have already obtained [14–17, 35], and then present our conclusion and future directions.

In chapter 2, we propose a method to combine interconnect planning with floorplanning. Our approach is based on the Wong-Liu floorplanning algorithm. We perform pin assignment and fast global routing during every iteration of floorplanning. We use a multi-stage simulated annealing approach in which different interconnect planning methods are used in different ranges of temperatures to reduce running time. A temperature adjustment scheme is designed to give smooth transitions between different stages of simulated annealing. Experimental results show that our approach performs well.

Deeper in the integration of interconnect planning and floorplanning, we found that the previous approach is not enough for large netlist. Since existing floorplanning algorithms use simulated annealing which needs to examine a large number of floorplans, the increasing number of nets has made interconnect-centric floorplanning computationally very expensive. Moreover, there is almost no systematic way to resolve the congestion problem in such magnitude of number of nets in a given floorplan. In chapter 3, we present a simple yet effective approach to significantly reduce the runtime of interconnect-centric floorplanning algorithms. Our idea is to group common nets between two blocks into a single net. We also present a more accurate global router for wiring evaluation based on Lagrangian Relaxation. The new router helps further congestion reduction while doing interconnect plan-

ning in floorplanning. We have incorporated our algorithms into [17] and observed dramatic improvement in runtime. For a 33-block 15K-net problem, we reduced runtime from over 23 hours to less than 50 minutes while getting comparable solution quality.

Meanwhile, without careful power supply planning in layout, the design of chips will suffer from mostly signal integrity problems including IR-drop,  $\Delta I$  noise, and IC reliability. Post-route methodologies in solving signal integrity problem have been applied but they will cause a long turn-around time, which adds costly delays to time-to-market. In chapter 4, we study the problem of simultaneous power supply planning and noise avoidance as early as in the floorplanning stage. We show that the noise avoidance in power supply planning problem can be formulated as a constrained maximum flow problem and present an efficient yet effective heuristic to handle the problem. Experimental results are encouraging.

Furthermore, along with careful power planning in layout design, I/O placement is becoming critical in modern IC design. The design will suffer from mainly hot-spot problem and long interconnect length if there is no planning on placing I/O buffers. There is a certain amount of cost to generate an I/O buffer site, which can be treated as a cluster of I/O buffers. We cannot just place I/O buffers greedily to minimize IR drop and wirelength since this will end up generating more I/O buffer sites and increase the design cost. In chapter 5, we study the problem of I/O buffer site placement in area-array ASIC designs and propose an algorithm to solve the problem with respect to design cost reduction. With slight increase in the percentage of voltage drop threshold violation, we can obtain much smaller design cost in

I/O buffer site placement.

Finally we conclude the dissertation with a summary of our results and a discussion of future directions in chapter 6.

## Chapter 2

# Integrated Floorplanning and Interconnect Planning

In this chapter, we propose a method to combine interconnect planning with floorplanning. Our approach is based on the Wong-Liu floorplanning algorithm. When the positions, orientations, and shapes of the cells are decided, the pin positions and routing of the interconnects are decided as well. We use a multi-stage simulated annealing approach in which different interconnect planning methods are used in different ranges of temperatures to reduce running time. A temperature adjustment scheme is designed to give smooth transitions between different stages of simulated annealing. Experimental results show that our approach performs well.

### 2.1 Introduction

With VLSI technology entering the DSM era, devices are scaled down to smaller sizes and placed at an ever increasing proximity. At the same time, with the increase of die dimensions, more functions are integrated into one chip. All these significantly increase the communication between different components, thus increasing the amount of interconnect on a chip. Moreover, the scaling down of fabrication geometry also makes interconnect delay a dominant factor in total circuit delay [20]. These trends make interconnect planning a necessary step in DSM

design [48].

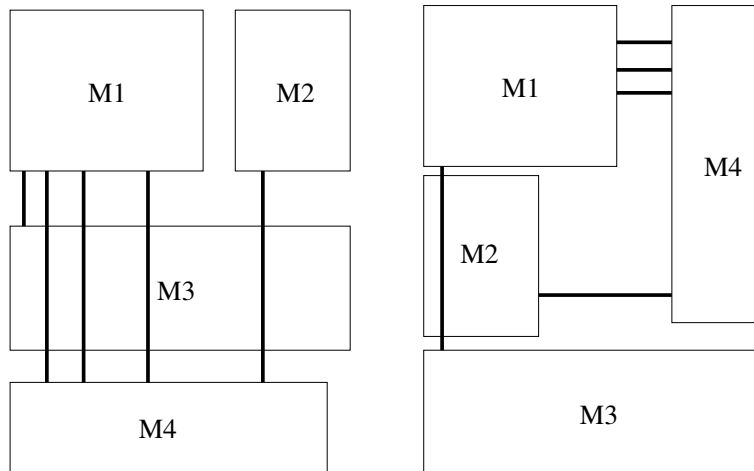


Figure 2.1: Floorplanning greatly influences interconnect structure

Global interconnects have significant influence on system performance in DSM technologies. Floorplanning, the process of placing functional blocks on the chip, can significantly affect the global interconnect structure. (Figure 2.1 shows two floorplans and their corresponding interconnect structures.) Many floorplanning algorithms have been proposed in the past 20 years [29, 43, 45, 49, 52, 60, 65]. All these algorithms focus on placing the circuit blocks using simple interconnect cost (e.g., total wire length) to guide the optimization. Without accurate interconnect planning during the floorplanning process, it is difficult for these algorithms to meet performance constraints due to unexpected “long” global interconnects resulted in the later routing stage.

In this chapter we propose a method to combine interconnect planning with floorplanning. Our approach is based on the Wong-Liu floorplanning algorithm [65].

Recall that the Wong-Liu algorithm uses Polish expressions to represent floorplans and searches for an optimal floorplan using simulated annealing by iteratively generating Polish expressions. Every time a Polish expression (i.e., a floorplan) is examined, the shape of the blocks are optimized and the total wire length is used as the interconnect cost. Instead of using the total wire length, we propose to perform careful interconnect planning with respect to the current floorplan being considered and obtain a much more accurate interconnect cost. The comparison of the original approach and our new approach is shown in Figure 2.2.

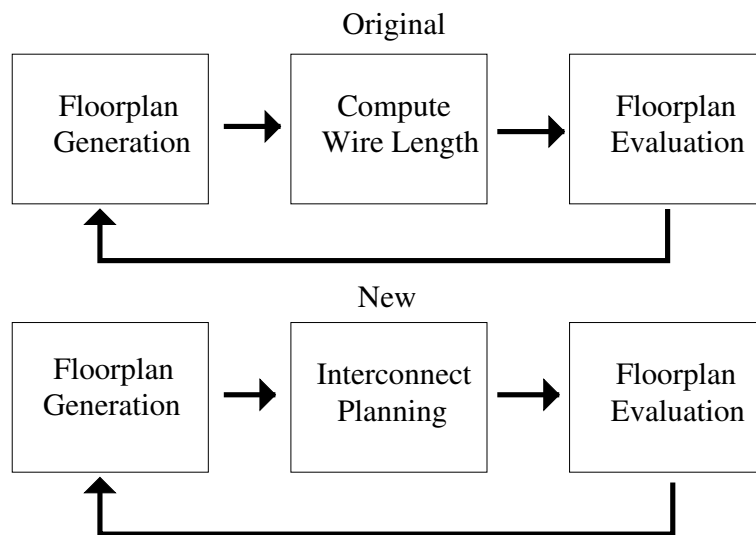


Figure 2.2: Floorplanning and interconnect planning

The interconnect planning step performs pin assignment and simple-geometry routing based on L-shaped and Z-shaped wires. Taking advantage of the nature of simulated annealing, we use different interconnect planning methods in different ranges of temperatures to reduce the running time. In particular, we use the conventional wire length estimation by half-perimeter of net bounding box when tem-

perature is high, use a more accurate interconnect cost based on L-shaped routing when temperature is in the medium range, and finally use Z-shaped routing when temperature is low. In order to implement our multiple cost function scheme, we found that it was necessary to introduce a temperature adjustment method to cope with the intrinsic discontinuities resulted in the process of switching cost functions.

The rest of the chapter is organized as follows. We introduce the algorithms for interconnect planning in Section 2.2. Section 2.3 discusses the multi-stage simulated annealing approach. Section 2.4 reports the experimental results for MCNC benchmarks and Section 2.5 concludes the chapter.

## **2.2 Efficient Interconnect Planning**

To simplify our discussion, we assume there are two layers for the routing of global interconnects – one layer for vertical wires and the other layer for horizontal wires. (However, our approach is applicable to designs with more than two layers.) We allow different layers to have different design rules, i.e., the minimum wire width and the minimum spacing in each layer are different. In order to estimate congestion/routability, we divide the floorplan into a number of bins by a grid the same way that it is typically done in global routing [57]. For each bin boundary, we define its *capacity* as the maximum number of nets that can cross it. Clearly, the capacity of each boundary can be easily computed based on its length (or width) and the design rules (i.e., minimum wire width and minimum wire spacing) for that layer. If the number of nets crossing a bin boundary exceeds the capacity of the bin boundary, we say there is *overflow*. Each global routing solution gives us the num-

ber of nets crossing each bin boundary, thus giving us detailed congestion/overflow information. Our goal is to plan the interconnects to avoid congestion/overflow as much as possible.

### **2.2.1 Pin Assignment**

The first step of interconnect planning is pin assignment. After module sizes and positions are fixed in a given floorplan, we determine the pin positions on each module. A simple strategy is used for efficiency. For each net, we connect the centers of the modules in this net and get the intersection points on the module boundaries as pin positions, as shown in Figure 2.3. This simple heuristic makes sense since it tries to minimize total wire length. Note that each module boundary is partitioned into a number of boundary segments by the grid. Since each boundary segment can only accommodate a limited number of pins, we should make sure that the number of pins we assign to each boundary segment does not exceed its capacity. If segment overflow occurs, we redistribute some of the pins to neighboring segments. Another guideline for pin assignment is to evenly distribute the pins so that no boundary segments are too crowded.

### **2.2.2 Simple-Geometry Routing**

After pin assignment, pin positions are known. We then perform simple-geometry based global routing to connect the pins. For a net with  $n$  pins where  $n > 2$ , we first construct a minimum spanning tree connecting the pins using the Manhattan distance metric. The net is then decomposed into a set of two-terminal

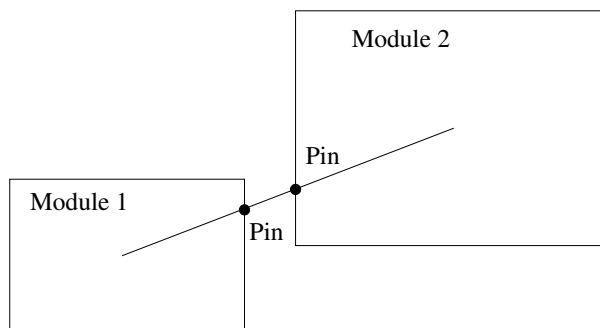


Figure 2.3: Pin assignment illustration

nets which correspond to the edges of the minimum spanning tree. After that, we have a set of nets with only two pins. For each of them, we connect the two pins using simple-geometry routing based on L-shaped or Z-shaped wires. Since the algorithms for L-shaped routing and Z-shaped routing are similar, they will be described together. Before we do simple-geometry routing, we map the pin positions of the nets to the corresponding bins. We use a sequential routing approach, that is, we route one net at a time. There are two steps in our simple-geometry routing algorithms. The first step is to use a stochastic approach to obtain the initial global congestion information. The second step is to utilize the information from the first step to route nets one by one.

In the first step, we estimate the congestion on each bin boundary by the expected number of nets crossing that boundary. Consider a two-pin net with pins  $p_1 = (x_1, y_1)$  and  $p_2 = (x_2, y_2)$ . If only L-shaped routes are allowed, there are at most two routes to connect the two pins, as shown in Figure 2.4. Assume that each possible route is equally likely, we can add  $1/2$  to each bin boundary on the two routes as the net's contribution to the expected number of nets crossing that

boundary. For Z-shaped routing, we compute the expected number of nets crossing each bin boundary as follows. Let  $m$  denote the total number of Z-shaped routes connecting  $p_1$  and  $p_2$ . As we can see, if  $x_1 = x_2$  or  $y_1 = y_2$ , then  $m = 1$ . Otherwise,  $m$  can be computed as follows.

$$m = |x_1 - x_2| + |y_1 - y_2|$$

For each bin boundary  $e$ , let  $m_e$  be the number of possible Z-shaped routes for the net to cross  $e$ . We again assume all routes are equally likely. Clearly, the net's contribution to the expected number of nets crossing  $e$  is  $m_e/m$ . For the example shown in Figure 2.5 for Z-shaped routing,  $m = 6$  and  $m_e = 1$  where  $e$  is the right boundary of  $bin(2,3)$ . Thus the net's contribution to the expected number of nets crossing  $e$  is  $m_e/m = 1/6$ . Putting contributions from different nets together, we can get the expected number of crossing nets on each boundary.

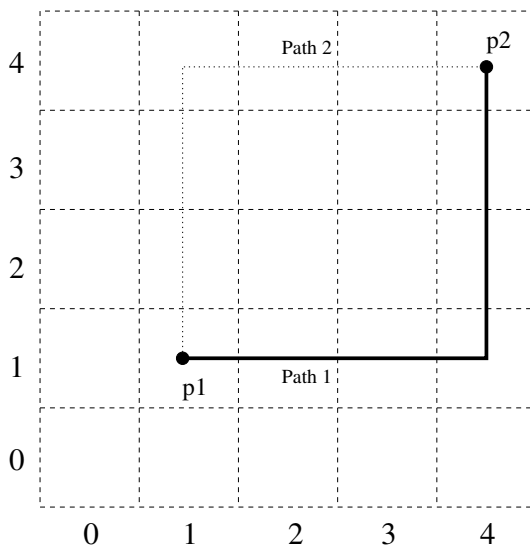


Figure 2.4: L-shaped routing

In the second step, we route one net at a time. When routing a net, we first remove its contribution from the expected number of crossing nets at each bin boundary. Then we determine a routing path with minimum crossing cost. The cost of crossing a bin boundary depends on a few factors. We use  $X_e$  to represent the overflow amount on bin boundary  $e$ . If there is no overflow on bin boundary  $e$ , let  $Y_e$  to be the difference between the current crossing and the capacity of  $e$ , and use  $Z$  to represent the overlapping length with previously routed wires belonging to the same (multi-pin) net. We determine a routing path which minimizes the following quantity:  $\alpha \sum X_e^2 + \beta \sum 1/Y_e^2 - \gamma Z^2$ . The first part is a *penalty* term, meaning that the global router is penalized because of going through the congested bin boundary. The second term is a *prevention* term, that is, the global router prevents from taking the path that is reaching saturation of the capacity. The third term is a *reward* that the router follows previous routes for those two-terminal nets within a multi-terminal net. After routing a net, if the route crosses a bin boundary  $e$ , its contribution to the expected number of nets crossing  $e$  will become 1 to reflect the real route. If the current crossing of the bin boundary exceeds the capacity, mark this net to be ripped-up and re-routed.

For all nets that are needed to be ripped-up and re-routed, we process them in the order from the most congested net, which is crossing the maximum number of congested bin boundaries, to the less congested ones trying to remove overflow as much as possible. Then we examine the results by getting the total square overflow terms of all bin boundaries. If the current overflow status exceeds the former one, recover the net to its original route.

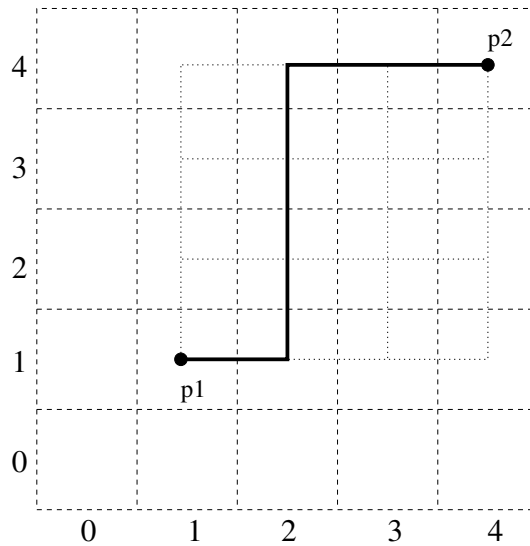


Figure 2.5: Z-shaped routing

### 2.2.3 Incremental Routing Cost Computation

A direct method to determine the path with minimum crossing cost connecting two points is as follows. For each possible path, L-shape or Z-shape, we need to sum up the crossing costs for all bin boundaries along the path to get the routing cost of this path. In this way, the time complexity of examining all L-shaped/Z-shaped paths joining two points is  $O(n^2)$  in the worst case, where the grid size is  $n \times n$ , since the total number of bin boundaries crossed by all L-shaped and Z-shaped paths between two points can be  $O(n^2)$ . It then follows that the total time to route  $N$  nets is  $O(Nn^2)$ .

In the following we present the idea of incremental routing cost computation which significantly speed up the cost function computation. For each  $(i, j)$ , we define  $v(i, j)$  as the accumulated crossing cost starting from the top bin boundary

of  $bin(i, 0)$  to that of  $bin(i, j - 1)$ . Similarly, we define  $h(i, j)$  as the accumulated crossing cost starting from the right bin boundary of  $bin(0, j)$  to that of  $bin(i - 1, j)$ . (See Figure 2.6, 2.7.) Note that all  $h(i, j)$ 's for a row can be computed in  $O(n)$  time, and all  $v(i, j)$ 's for a column can be computed in  $O(n)$  time. Thus all  $h(i, j)$ 's and  $v(i, j)$ 's can be precomputed in  $O(n^2)$  time.

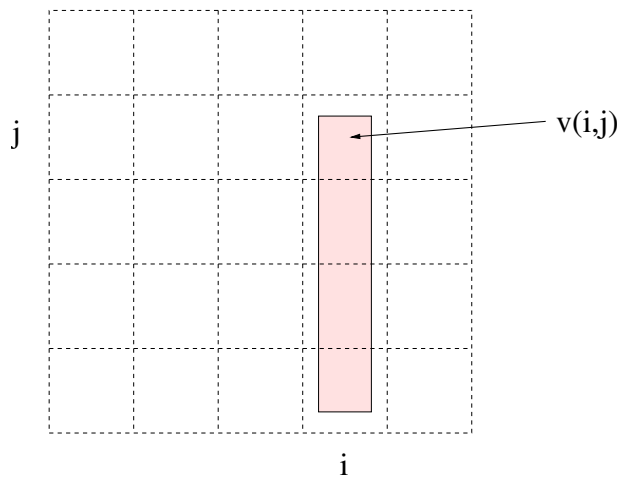


Figure 2.6:  $v(i, j)$  is the accumulated cost of shaded bin boundaries

Note that the routing cost for each L-shaped/Z-shaped path can be expressed in terms of  $h(i, j)$ 's and  $v(i, j)$ 's. (The number of  $h(i, j)$  or  $v(i, j)$  terms in a L-shaped path is at most 4 and that for a Z-shaped path is at most 6.) For example, the routing cost of the Z-shaped path in Figure 2.8 can be computed by  $v(3, 3) - v(3, 2) + h(3, 2) - h(1, 2) + v(1, 2)$ . So if all the  $h(i, j)$ 's and  $v(i, j)$ 's are precomputed, the time for evaluating all L-shaped/Z-shaped paths between two points is  $O(n)$  since there are  $O(n)$  such paths and the routing cost of each path can be computed in  $O(1)$  time. After we route a path, we need to update the  $h(i, j)$ 's and  $v(i, j)$ 's on at most three columns/rows, and therefore can be done in  $O(n)$  time. If

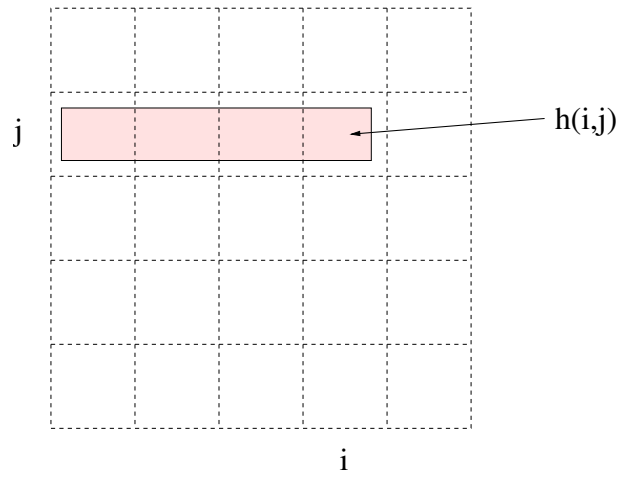


Figure 2.7:  $h(i, j)$  is the accumulated cost of shaded bin boundaries

there are  $N$  nets, the total time for updating  $h(i, j)$ 's and  $v(i, j)$ 's is  $O(Nn)$ . As a result, the total time for routing  $N$  nets is  $O(n^2 + Nn)$ , which compares well with the  $O(Nn^2)$  time direct method. The speed-up is roughly from cubic to quadratic in runtime.

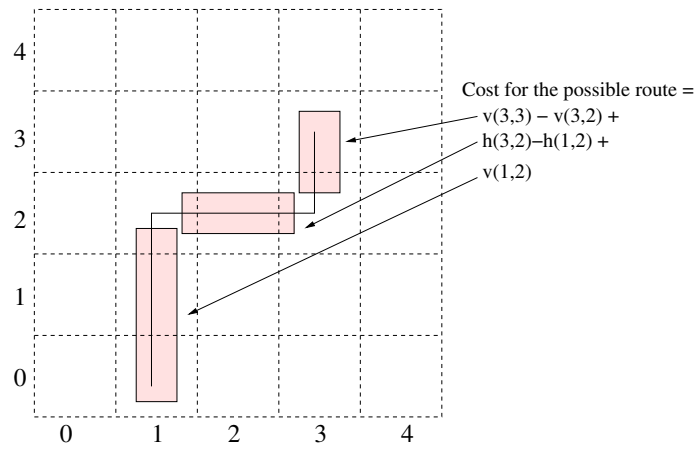


Figure 2.8: The routing cost in terms of  $h(i, j)$ 's and  $v(i, j)$ 's

## 2.3 Multi-Stage Simulated Annealing

Among our two interconnect planning approaches, Z-shaped routing is more accurate than L-shaped routing. But Z-shaped routing is also more expensive than L-shaped routing. Using Z-shaped routing all along will give the most accurate estimation. However, based on the characteristics of simulated annealing, we can speed up the procedure without sacrificing the quality of solutions.

The Wong-Liu floorplanning algorithm [65] is based on simulated annealing which is a technique for solving general optimization problems. The algorithm moves from one solution to another, trying to find the optimum solution. It accepts a move with the probability  $e^{-\Delta C/T}$ , where  $\Delta C$  is the increase of cost by that move and  $T$  is the current temperature. When the temperature is very high, different estimation methods for the cost will not show much difference on  $-\Delta C/T$ . That means it does not affect much in performance if we use rough cost function at the beginning of annealing. When temperature gradually decreases, we use more accurate cost estimation. The L-shaped routing estimation is more accurate than the simple center-to-center or half-perimeter estimation. Similarly, the Z-shaped routing is more accurate than the L-shaped routing estimation. Therefore, we will start with the the center-to-center or half-perimeter estimation, gradually transfer to L-shaped routing, and finally switch to Z-shaped routing. This multi-stage approach is very effective in reducing total running time.

In fact, multi-stage simulated annealing is just a method to combine different approaches together in one process. It should be reasonable if those different approaches used in multi-stage simulated annealing are not totally different, which

means they have a certain degree of correlation. In this chapter we use a three-stage simulated annealing approach. The first stage is to get a good initial solution by using only the half-perimeter wire length estimation. The second stage is to estimate interconnect cost by using L-shaped global routing. The third stage is to estimate interconnect cost by using Z-shaped global routing. The transitions between stages are not very abrupt since they evolve from simple to complex, from rough to accurate. However, even for very similar estimations, we still need to find a way to take care of any possible discontinuity in switching cost functions.

### 2.3.1 Cost Function Transitions

The cost function used in [65] is  $A + \lambda W$ , where  $A$  is the total area of the packing,  $W$  is the half-perimeter estimation of the interconnect cost, and  $\lambda$  is a constant which controls the relative importance of these two terms and is usually set such that the area term and the interconnect term are approximately balanced. The normal curve of cost versus the number of iterations of simulated annealing process is shown in Figure 2.9.

In our approach, we use the cost function  $\Psi = \alpha A + \beta W + \gamma O$ , where  $A$  and  $W$  are the same as in [65] and  $O$  is the sum of the square of overflow in routings. Although the format of cost function is identical for three stages of the process, the content of each term is different. The term  $W$  in stage 1 is obtained by applying half-perimeter method of net bounding boxes; the term  $W$  in stages 2 and 3 are obtained by applying pin assignment and summing the net length from pin positions. The term  $O$  in stage 1 is zero; the term  $O$  in stages 2 and 3 is obtained



Figure 2.9: Normal curve of cost versus number of iterations

by applying simple-geometry routing and computing the congestion/routability estimation of bin boundaries. Because of the difference of cost functions used in different stages during simulated annealing process, discontinuities may occur when switching stages. One possible scenario is that the annealing process will suddenly converge to suboptimal solution when cost function transition occurs, as shown in Figure 2.10. The other possible scenario is that the annealing process will take much longer time to converge to optimal solution when cost function transition occurs, as shown in Figure 2.11. The discontinuities happen because the temperature is too low for the former scenario and is too high for the latter one when switching cost functions. In order to cope with the discontinuities resulted in the process of switching cost functions, we introduce a temperature adjustment method, which is described in the next subsection.

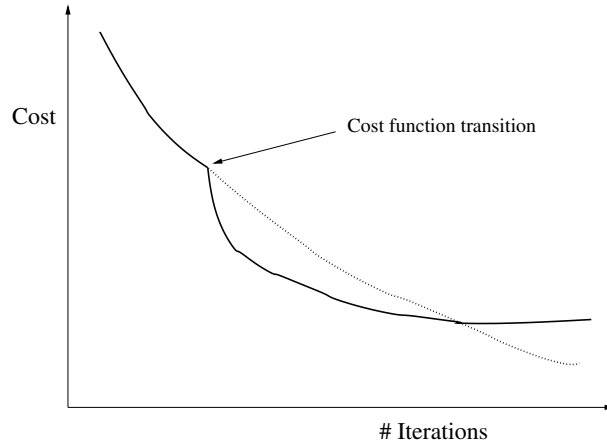


Figure 2.10: Abnormal curve (solid line) of cost versus number of iterations: quickly converges to suboptimal solution

### 2.3.2 Temperature Adjustment

Simulated annealing uses temperature to control the probability in accepting uphill moves. We use a temperature schedule of the form  $T_k = r * T_{k-1}$ ,  $k = 1, 2, 3, \dots$ . The initial temperature  $T_0$  is determined by performing a sequence of random moves and computing the quantity  $\Delta_{avg}$ , the average value of the magnitude of change in cost per move. We should have  $e^{-\Delta_{avg}/T_0} = P \cong 1$  so that there will be a high probability of acceptance at high temperatures. This suggests that  $T = -\Delta_{avg}/\ln(P)$  is a good choice for  $T_0$ .

In [65], a single cost function is used to evaluate the quality of a solution. However, in our approach, we use different cost functions in different stages. We know that one major term to decide the acceptance of a solution in simulated annealing is  $e^{-\Delta C/T}$ . Take the transition between the first stage and the second stage as an example, the difference of cost in the second stage is typically larger than that

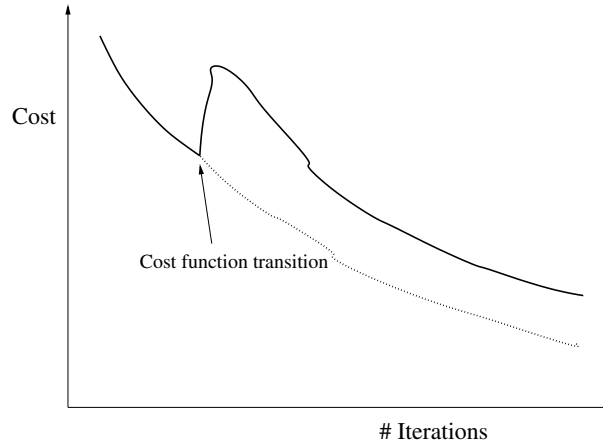


Figure 2.11: Abnormal curve (solid line) of cost versus number of iterations: takes a much longer time to converge

in the first stage. That is,  $-\Delta C_{old} \gg -\Delta C_{new}$ . Therefore, the probability of accepting uphill moves in the iterative-based process will decrease suddenly and the simulated annealing process will end prematurely. For example, when we encounter the stage transfer from half-perimeter estimation to L-shaped routing, suppose the current temperature is 100, the average  $\Delta C_{old}$  is 20 and the average  $\Delta C_{new}$  is 100. The probability of accepting uphill moves is  $e^{-\Delta C_{old}/T} = 0.8$  before switching cost function but it is  $e^{-\Delta C_{new}/T} = 0.36$  after cost function transition. This abrupt decrease in acceptance probability would result in quick convergence to suboptimal solution because the current temperature is too low to sustain the annealing process. Similarly, it is possible that after cost function transition, the acceptance probability will substantially increase. In this case, that the current temperature is too high results in slow convergence of the annealing process.

In our approach, in addition to calculating the starting temperature of the

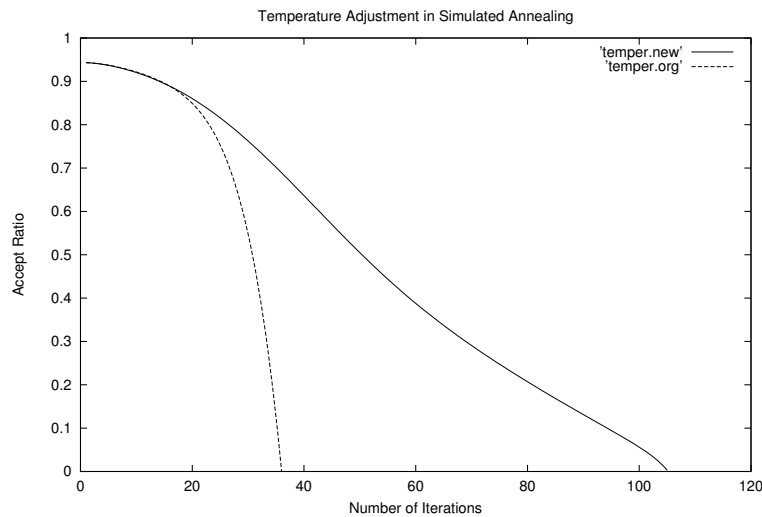


Figure 2.12: Temperature adjustment

first stage, we also determine the starting temperature of the second and the third stages by calculating random move cost with the same approach. When we reach the transition between first and second stages or between second and third stages, we compute the starting temperature of the second or the third stage  $T'$  by getting the new average value of the magnitude of change in cost per move, and using the current acceptance ratio,  $P_{curr\_acpt}$ , as a reference probability:  $T' = -\Delta'_{avg} / \ln(P_{curr\_acpt})$ .

Although we use the current acceptance ratio to compute the new initial temperature during transition, the acceptance ratio will rise. The reason is that for the very first initial temperature estimation, we measure the term by random walks, but there exists very few random walk when transition occurs. We handle this by reducing the temperature much faster than the usual cooling ratio, until the accep-

tance ratio goes back on track. Experimental results show that this approach is really helping the continuity of the simulated annealing process and the quality of performance. (Figure 2.12 shows the effectiveness of applying temperature adjustment approach. The curve would have been the one in dotted line: it suddenly drops because of the abnormal end of the process.)

## 2.4 Experimental Results

We have tested our approach on some MCNC building blocks examples. All experiments were carried out on a 300MHz Pentium II Intel Processor. In order to compare the performance of the interconnect planning approach with that of the original approach [65] in terms of routability, we perform pin assignment and use Z-shaped routing to route the nets in the final floorplans produced by the conventional approach. Figure 2.13 shows the floorplan obtained by our pin assignment and interconnect planning approach. Figure 2.14 shows the floorplan obtained by the original approach. The dashed lines are the grid lines, and the thickness of line in the boundaries denotes the degree of overflow. We can see significant difference in Figure 2.13 and Figure 2.14 for *ami49* benchmark in terms of wire overflow, while the packing areas are about the same (Table 2.1 and 2.2). For the five MCNC benchmarks shown in these two tables, we observe that the new approach produces floorplans which are much more routable than the ones produced by the original floorplanner. Note that the maximum violation in Table 2.1 indicates the maximum amount of overflow occurred in any bin boundary after interconnect planning, while the total violations indicate the total amount of overflow occurred in a floorplan. In

fact, the new method achieves a significant percentage of improvement in maximum violation and total violations without any area overhead.

Table 2.1: Experimental results of our approach on MCNC examples, compared with the method in [65]

Data	$n$	Our Floorplanner			Floorplanner in [65]			
		Time (sec)	Dead Space(%)	Total Vios( $\mu\text{m}$ )	Max Vio( $\mu\text{m}$ )	Dead Space(%)	Total Vios( $\mu\text{m}$ )	Max Vio( $\mu\text{m}$ )
apte	9	277.6	0.99	0.51	0.27	0.86	10.31	3.45
xerox	10	589.7	0.14	0.0	0.0	0.07	23.92	8.88
hp	11	141.2	0.30	0.76	0.68	0.61	15.16	3.34
ami33	33	2220	3.66	1.55	0.64	5.68	15.96	2.64
ami49	49	4041	2.93	7.68	2.75	3.21	38.62	6.75

Table 2.2: Performance improvement of our approach on MCNC examples, compared with the method in [65]

Data	$n$	#net	Improvement	
			Total Vios(%)	Max Vio(%)
apte	9	97	95	92
xerox	10	203	100	100
hp	11	83	95	80
ami33	33	123	90	76
ami49	49	408	80	59

## 2.5 Concluding Remarks

This chapter presents a method to integrate floorplanning with interconnect planning. Simple-geometry routing is used to efficiently plan wires during module packing. A congestion cost is combined into the Wong-Liu simulated annealing

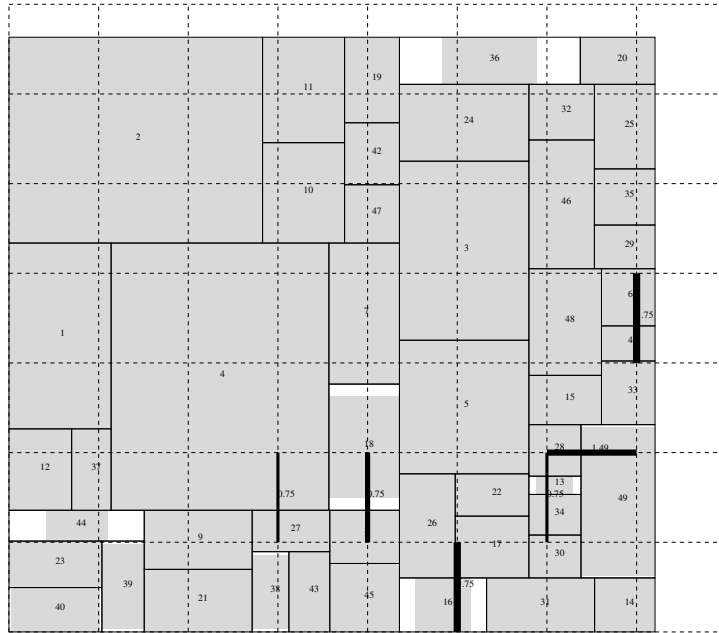


Figure 2.13: Result of packing ami49 using interconnect planning approach

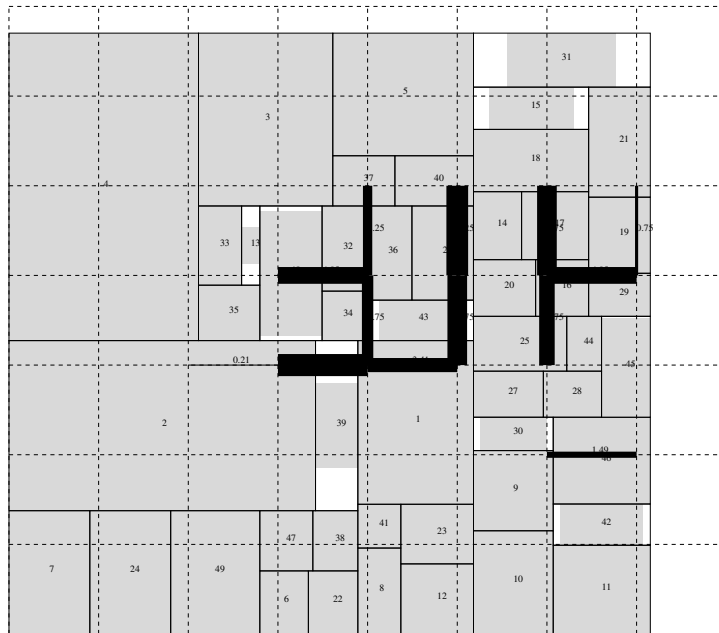


Figure 2.14: Result of packing ami49 using original approach

based floorplanner, and a multi-stage simulated annealing strategy is used to effectively reduce the running time. We further develop a temperature adjustment approach to cope with the discontinuities resulting from switching cost functions. Experimental results show that our approach works well.

## Chapter 3

### **Faster and More Accurate Wiring Evaluation in Interconnect-Centric Floorplanning**

Floorplanning problems typically have relatively small number of blocks (e.g., 50-100) but have a large number of nets (e.g. 20K). Since existing floorplanning algorithms use simulated annealing which needs to examine a large number of floorplans, the increasing number of nets has made interconnect-centric floorplanning computationally very expensive. Moreover, there is almost no systematic way to resolve the congestion problem in such magnitude of number of nets in a given floorplan. In this chapter, we present a simple yet effective idea to significantly reduce the runtime of interconnect-centric floorplanning algorithms. Our idea is to group common nets between two blocks into a single net. This faster wiring evaluation technique is very effective. We also present a more accurate global router for wiring evaluation based on Lagrangian Relaxation. The new router helps further congestion reduction while doing interconnect planning in floorplanning. We have incorporated our algorithms into [17] and observed dramatic improvement in runtime. For a 33-block 15K-net problem, we reduced runtime from over 23 hours to less than 50 minutes.

### 3.1 Introduction

With VLSI technology entering the DSM era, devices are scaled down to smaller sizes and placed at an ever increasing proximity. Meanwhile, with the increase of die sizes, more functions are integrated into one chip. All these significantly increase the communication between different elements. In physical design, the netlist is generated after the partitioning step while the circuits are restructured and functional blocks are obtained. The subsequent floorplanning, placement, and routing steps use the netlist to perform various kinds of optimizations to meet certain constraints. Due to the interconnect-dominant factor in modern VLSI design, communication between components is increasing dramatically. Most of logic hierarchy are flattened during placement and route, so it is very possible that there are hundreds of thousands of nets, while there are only tens of functional blocks in contrast. It goes without saying that the complexity of evaluating such netlists with enormous size of nets is extremely high. How to model and evaluate those high-density interconnection is becoming one of the most challenging issues in modern high-performance VLSI design.

Very recently, interconnect planning (especially global interconnects) is the key issue in modern VLSI physical design [47, 48]. However, with tens or hundreds of thousands of interconnect in standard-cell or full-custom design after circuits partitioning, interconnect planning is time-consuming. Many floorplanning algorithms, including slicing and non-slicing, have been proposed in past decades [17, 29, 43, 45, 49, 65], but they did not take interconnect planning into account except [17]. In [17], we integrate floorplanning and efficient interconnect planning.

Since every wiring evaluation step uses a simple-geometry global router, [17] can only process moderate-sized netlists (e.g. 1000 nets) but will not be efficient to solve problems with very large netlists (e.g. 10K nets). Furthermore, [17] used a heuristic global router to sequentially route global interconnects, the performance of which depends on the ordering of nets to be routed.

Since existing floorplanning algorithms use simulated annealing which needs to examine a large number of floorplans, the increasing number of nets has made interconnect-centric floorplanning computationally very expensive. In this chapter, we present a simple yet effective idea to significantly reduce the runtime of interconnect-centric floorplanning algorithms. Our approach is to group common nets between two blocks into a single net. Before grouping the nets, multi-terminal nets are first decomposed into two-terminal nets. Bounded-degree hypergraph-to-graph transformation is used to preserve the constraint for pin-limit in a block. This net reduction technique is very effective. Suppose we are given a problem with 50 blocks and 20K nets. After net reduction, we can have at most 1250 nets (which is the case when we have connections between all pairs of blocks). This is a significant reduction from 20K nets. If we need linear time to process the nets each time we examine a candidate floorplan, then we would have more than 15X speedup in runtime after net reduction. In our experiments, for a problem with 33 blocks and 15K nets, the floorplanning algorithm in [17] took more than 23 hours to run on the original netlist in 300MHz machines, but took less than 50 minutes after net reduction. Meanwhile, we present a more accurate global router for wiring evaluation in this chapter. We use Lagrangian relaxation technique to systematically route

global interconnects, trying to minimize the maximum violation against the routing resources. This Lagrangian relaxation router is performing well. It successfully minimizes the maximum violation and total violations compared with router in [17] when given a candidate floorplan.

The rest of the chapter is organized as follows. Section 3.2 describes the wiring evaluation. Faster wiring evaluation and the algorithm for net reduction are presented in Section 3.3; more accurate wiring evaluation and Lagrangian relaxation technique application are presented in Section 3.4. Experimental results are shown in Section 3.5 and the concluding remarks are presented in Section 3.6.

## **3.2 Wiring Evaluation in Interconnect Centric Floorplanning**

In physical design, the netlist is generated after partitioning step while the circuits are restructured and functional blocks are obtained. The subsequent floorplanning, placement, and routing steps use the netlist to perform various kinds of optimizations to meet certain constraints. In the literature of floorplanning research, the focus of optimization was to minimize the total packing area as well as the wiring cost. In modern VLSI design, the size of design becomes much larger, thus the complexity of netlist is extremely higher than ever. Since the interconnect-centric floorplanning tool plays more important role in physical design, how to model and evaluate large number of interconnections has become one of the most challenging issues in modern high-performance VLSI design.

Our approach to speedup the wiring evaluation for such a huge netlist in floorplanning is net reduction. The idea is to group common nets between two

blocks into a single net. Before grouping the nets, it is necessary for multi-terminal nets to be decomposed into two-terminal nets first. We use bounded-degree hypergraph-to-graph transformation to accomplish the decomposition. The main objective of this approach is to meet the constraint that the connection of each block is within its pin-limit. It may look like the general connectivity approaches in [59, 64], which are different from this net reduction in terms of the objectives. In [59, 64], they use general connectivity model to estimate the wirelength and timing evaluation during placement. In our approach, however, in addition to get “connectivity matrix”, we perform global routing based on the net regrouping result.

After net reduction, we can use another approach to further reduce the congestion occurred in floorplanning. This approach is based on Lagrangian relaxation and the objective is to minimize the maximum violation against routing resources in a given floorplan. The update of Lagrangian multipliers will help reduce the local congestion in a systematical way, trying to balance the routing among routing regions. Because of the nature of Lagrangian relaxation, that is, it takes longer runtime to converge to optimal solutions, we perform it at the end of the process. Combining the net reduction and Lagrangian relaxation based global router makes the interconnect-centric floorplanning more efficient and effective.

To effectively evaluate wiring results, we give some general definitions to some subjective terms in this chapter. In order to measure routability, we divide the floorplan into a number of bins by a grid the same way in [17]. For each bin boundary, we define its *capacity* as the maximum number of nets that can cross it. If the number of nets crossing a bin boundary exceeds the capacity of the bin boundary,

there is *overflow* in that bin boundary. We estimate the *maximum overflow* among all bin boundaries and *total overflow* for all overflow occurred as important part of our objectives to evaluate the quality of a floorplan. In the following sections, we describe the details of efficient wiring evaluation in Section 3.3 and those of effective wiring evaluation in Section 3.4.

### 3.3 Faster Wiring Evaluation Based on Net Reduction

In this section, we present an approach, *net reduction*, to significantly reduce the runtime of interconnect-centric floorplanning algorithms. Net reduction is a technique to help interconnect cost evaluation for huge netlists. For two-terminal nets, net reduction is easy to be accomplished by grouping them into wider nets (See Figure 3.1). For multi-terminal nets, however, net decomposition is needed to perform the net reduction. Arbitrary net decomposition will need to pay the price of crowding the connections into a block. As an example, Figure 3.2 shows a netlist with one two-terminal net and three multi-terminal nets in a floorplan containing six blocks. Without loss of generality, assume that the pin-limit for block  $b_4$  is 3. In Figure 3.3, with the same number of wider nets, (a) shows an illegal net decomposition since the connection of block  $b_4$  is 6, which exceeds the pin-limit, while (b) shows a feasible net decomposition.

#### 3.3.1 Problem Formulation

Based on the objective illustrated above, we can state the net reduction problem as follows.

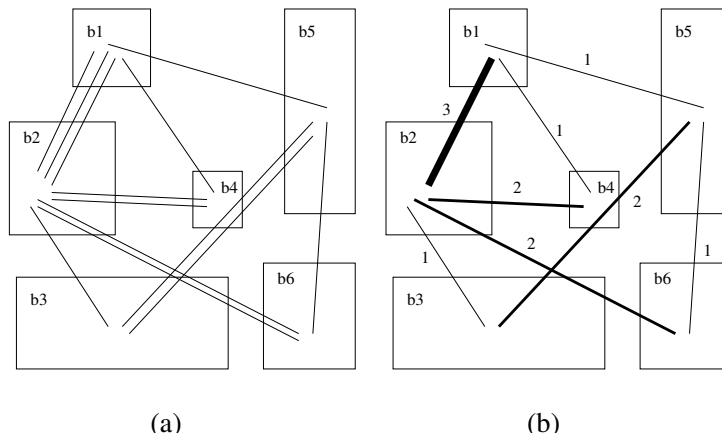


Figure 3.1: Net reduction (a) Original netlist. (b) Reduced netlist.

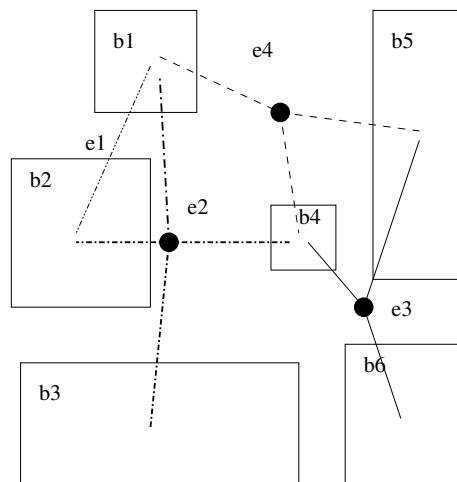


Figure 3.2: A netlist with multi-terminal nets.

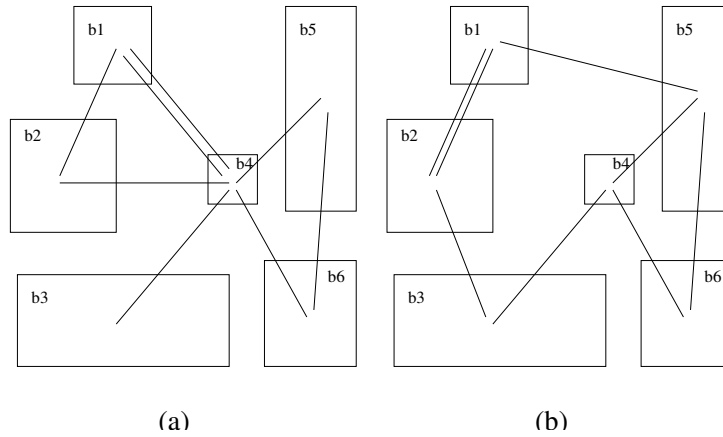


Figure 3.3: Two net decomposition solutions for the example in Figure 3.2. (a) An illegal net decomposition since block  $b_4$  exceeds its pin-limit. (b) A feasible net decomposition.

**Problem 3.3.1.** *Given a floorplan of blocks  $b_1, \dots, b_n$ , and their pin-limit  $d_1, \dots, d_n$ , respectively, and given a netlist of nets (multi-terminal and two-terminal)  $e_1, \dots, e_m$ , find a net reduction solution such that the resultant nets are two-terminal nets (re-grouping to wider nets) and that the solution meets the constraint that the connection for each block is within its pin-limit.*

### 3.3.2 Algorithm for Net Reduction

Our algorithm for net reduction is similar to the work in [41] for board-level routing for FPGA-based logic emulation. According to the objective in the problem formulation, we observe that if we can model the netlist as a hypergraph by representing the functional blocks in a floorplan as *vertices* and the nets as *hyperedges*, the problem of net decomposition can be seen as a bounded-degree hypergraph-to-graph transformation, where the bounded degree is the pin-limit for a block.

We introduce the bounded-degree hypergraph-to-graph transformation and use it to decompose those nets and create the corresponding spanning trees. Then we present the net reduction algorithm based on the transformation and resultant spanning trees. The last step of the algorithm is to group all decomposed two-terminal nets into wider interconnect.

### 3.3.2.1 Bounded-Degree Hypergraph-to-Graph Transformation

We want to solve the problem of transforming a hypergraph to a graph by modelling each hyperedge as a spanning tree so that the degree of each vertex  $v$  in the resultant graph does not exceed some given bound  $\sigma_v$ . This problem is studied as the *bounded-degree hypergraph-to-graph transformation problem*. Figure 3.4 shows a transformation of a hypergraph to a graph where the degrees of all vertices are bounded by 3. Each hyperedge is transformed to a spanning tree that connects all the vertices in the hyperedge. In general, the degree bound  $\sigma_v$  can be different for different vertex  $v$ .

To model a hyperedge of  $p(\geq 2)$  vertices as a spanning tree that connects the  $p$  vertices, clearly the sum of the degrees of the vertices in the spanning tree must be  $2(p-1)$  and the degree of each vertex must be at least one. On the other hand, it can be shown that given any vector  $d = (d_1, \dots, d_p) \in \mathbf{N}^p$  such that  $\sum_{i=1}^p d_i = 2(p-1)$  and  $d_1, \dots, d_p \geq 1$ , we can always construct a spanning tree of  $p$  vertices whose degrees are equal to the  $p$  elements of vector  $d$ . We can easily construct an efficient algorithm for generating a spanning tree given any valid degree specification vector. Furthermore, we can guarantee to get a *minimum-height* spanning tree, which is

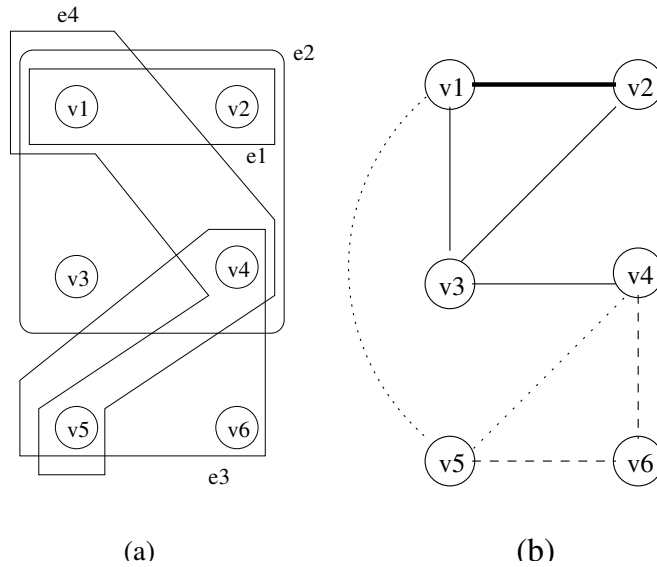


Figure 3.4: Hypergraph-to-graph transformation. (a) A hypergraph with four hyperedges:  $e_1$ ,  $e_2$ ,  $e_3$ , and  $e_4$ . (b) A graph formed by combining four spanning trees corresponding to the four hyperedges in (a).

good for performance.

Now we describe the algorithm for the bounded-degree hyper-graph-to-graph transformation problem. Suppose we are to transform a hypergraph  $H = (V, E)$  to a graph  $G$  given the degree bound  $\sigma_v$  of each vertex  $v$  in  $V$ . We construct a flow network  $W = (\mathcal{N}, \mathcal{A})$  as follows. The node set  $\mathcal{N}$  is  $\{e_1, \dots, e_{|E|}, v_1, \dots, v_{|V|}, s, t\}$  where node  $e_i$  corresponds to hyperedge  $e_i$  in  $E$  ( $i = 1, \dots, |E|$ ), node  $v_j$  corresponds to vertex  $v_j$  in  $V$  ( $j = 1, \dots, |V|$ ), node  $s$  is the *source*, and node  $t$  is the *sink*. For every hyperedge  $e_i$ , if it connects  $p$  vertices, then there is an arc from node  $s$  to node  $e_i$  with capacity  $c(s, e_i) = p - 2$ , and for every vertex  $v_j$  connected by hyperedge  $e_i$ , there is an arc from node  $e_i$  to node  $v_j$  with capacity  $c(e_i, v_j) = p - 2$ . For every vertex  $v_j$  in  $V$ , there is an arc from node  $v_j$  to node

$t$  with capacity  $c(v_j, t) = \sigma_{v_j} - \text{deg}_H(v_j)$ , where  $\text{deg}_H(v_j)$  is the degree of vertex  $v_j$  in  $H$ . For example, to transform the hypergraph in Figure 3.4(a) to a graph where the degree of each vertex is bounded by 3, we construct the network shown in Figure 3.5(a).

To model each hyperedge as a spanning tree so that the total degree of each vertex  $v$  in the resultant graph is bounded by  $\sigma_v$ , we have to find an integral maximum flow from  $s$  to  $t$  in the constructed network. It is well-known that if the capacities of all arcs in a network are integers, then there exists an integral maximum flow (i.e., the flow in each arc is an integer). Furthermore, in this case, maximum flow algorithms such as the Ford-Fulkerson method [27] always produce an integral maximum flow. In the following theorem, we show that how a feasible transformation can be derived from an integral maximum flow solution.

**Theorem 3.3.2.** *A bounded-degree hypergraph-to-graph transformation problem is feasible if and only if in a maximum flow of the constructed network  $W$ , the flow in arcs  $(s, e_1), (s, e_2), \dots, (s, e_{|E|})$  are all at their capacities.*<sup>1</sup>

### 3.3.2.2 Net Reduction by Multi-terminal Net Decomposition

We use the transformation presented in Section 3.3.2.1 to construct our net reduction algorithm. An illustrated example is followed by the algorithm.

---

<sup>1</sup>By the construction of  $W$ , if one maximum flow saturates arcs  $(s, e_1), \dots, (s, e_{|E|})$ , then any other maximum flow also saturates arcs  $(s, e_1), \dots, (s, e_{|E|})$ .

### Algorithm Net Reduction

1. Model the netlist as a hypergraph  $H$  by representing the functional blocks as vertices and the nets as hyperedges.
2. Transform hypergraph  $H$  to a graph where the degree of each vertex does not exceed the pin-limit per block:
  - (a) Construct network  $W$ .
  - (b) Find a maximum flow in  $W$ .
  - (c) Get a valid degree specification vector for each hyperedge from the maximum flow solution.
  - (d) Transform each hyperedge into a spanning tree according to its degree specification vector.
3. Decompose each multi-terminal net into subnets according to the corresponding hyperedge to spanning tree transformation.
4. Group all subnets according to the spanning trees to generate a set of wider interconnects.

For example, we are to decompose the multi-terminal nets in a floorplan shown in Figure 3.2. We first model it as the hypergraph (like Figure 3.4(a)) where hyperedge  $e_i$  represents net  $i$  and vertex  $v_j$  represents block  $j$ . Then we construct the network in Figure 3.5(a). A maximum flow  $f$  of the network is found in Figure 3.5(b). Using flow  $f$ , the degree specification vector for hyperedge  $e_2$  is  $(f(e_2, v_1) + 1, f(e_2, v_2) + 1, f(e_2, v_3) + 1, f(e_2, v_4) + 1) = (1, 2, 2, 1)$  and the degree specification vector for hyperedge  $e_3$  is  $(f(e_3, v_4) + 1, f(e_3, v_5) + 1, f(e_3, v_6) + 1) = (1, 1, 2)$ . So we model hyperedge  $e_2$  as a spanning tree  $T_2$  where the degrees of vertices  $v_1, v_2, v_3$ , and  $v_4$  in  $T_2$  are 1, 2, 2, and 1, respectively. And model hyperedge  $e_3$  as a spanning tree  $T_3$  where the degrees of vertices  $v_4, v_5$ , and  $v_6$  in  $T_3$  are 1, 1, and 2, respectively (See Figure 3.6). For hyperedges  $e_1$  and  $e_4$ , it is similar to

model them as spanning trees. Figure 3.3(b) shows the final, and also feasible, net reduction for the example shown in Figure 3.2.

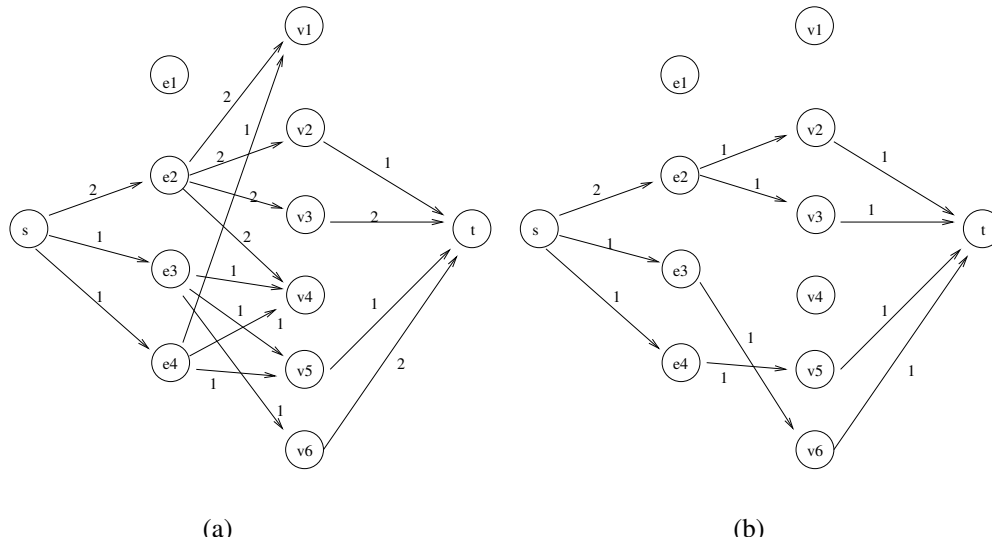


Figure 3.5: (a)Flow network built upon the hypergraph model. (b) An integral maximum flow. The number besides each arc is the (a)capacity/(b)flow in the arc.

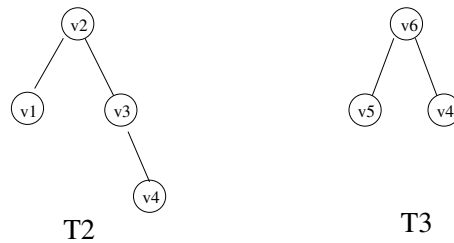


Figure 3.6: Spanning trees for hyperedges  $e_2$  and  $e_3$ .

### 3.4 More Accurate Wiring Evaluation Based on Lagrangian Relaxation

In this section, a new approach for more accurate wiring evaluation based on Lagrangian relaxation is presented. We use the Lagrangian relaxation technique to construct a systematic global router. Based on this technique, updating the Lagrangian multipliers is the guidance to help the router to alleviate local congestion in routing regions.

#### 3.4.1 Problem Formulation

We are given  $m$  nets and  $n$  bin boundaries in routing grid. The decision vector  $\mathbf{x}$  is defined as follows. Let  $x_{ik}$  be a decision variable of each bin boundary  $i$  such that

$$x_{ik} = \begin{cases} 1 & \text{if net } k \text{ cross bin boundary } i \\ 0 & \text{otherwise} \end{cases}$$

Then the global routing problem can be formulated as the following *primal problem*:

$$\begin{aligned} & \text{Minimize} && O_{max} \\ & \text{Subject to} && \sum_k x_{ik} \leq C_i + O_{max} \quad i = 1, 2, \dots, n \end{aligned}$$

where  $O_{max}$  is the maximum overflow/violation among all bin boundaries, and  $C_i$  is the capacity for the  $i_{th}$  bin boundary. The objective is to minimize the maximum overflow in routing grid so that local congestion is minimized.

#### 3.4.2 Lagrangian Relaxation

Lagrangian relaxation is a general technique for solving optimization problems with difficult constraints [2]. According to the Lagrangian relaxation pro-

cedure, we introduce non-negative multipliers, called *Lagrangian multipliers*, to the constraints in order to get rid of those difficult constraints and incorporate them into the objective function. Let  $\lambda_i$  denote the multiplier for the constraint  $\sum_k x_{ik} \leq C_i + O_{max}$ ,  $\vec{\lambda}$  be the vector of all Lagrangian multipliers introduced to the constraints. Then the Lagrangian relaxation subproblem (LRS) associated with the multiplier  $\vec{\lambda}$  becomes:

$$\text{Minimize } O_{max} + \sum_i \lambda_i (\sum_k x_{ik} - C_i - O_{max})$$

Let  $L(\vec{\lambda})$  be the amount above that we are trying to minimized. For any value of the Lagrangian multiplier  $\vec{\lambda}$ ,  $L(\vec{\lambda})$  is a lower bound of the optimal objective function value of the original problem. To obtain the sharper possible lower bound, we need to solve the following problem

$$\begin{aligned} &\text{Maximize } L(\vec{\lambda}) \\ &\text{Subject to } \vec{\lambda} \geq 0 \end{aligned}$$

which we refer to as the *Lagrangian multiplier problem* or *Lagrangian dual problem (LDP)* of primal problem. In the following subsections, we present the approaches to solve the primal problem by solving LRS and LDP.

### 3.4.2.1 Simplification of Lagrangian Relaxation Subproblem

LRS can be simplified by applying Kuhn-Tucker conditions.

$$\begin{aligned} L(\vec{\lambda}) &= O_{max} + \sum_i \lambda_i (\sum_k x_{ik} - C_i - O_{max}) \\ &= \sum_i \lambda_i (\sum_k x_{ik} - C_i) + O_{max} (1 - \sum_i \lambda_i) \end{aligned}$$

The Kuhn-Tucker conditions imply that  $\partial L(\vec{\lambda})/\partial O_{max} = 0$  for all  $1 \leq i \leq n$  at the optimal solution of the primal problem. Therefore, in searching for  $\vec{\lambda}$  to optimize LDP, we only need to consider the situation such that the conditions are satisfied. Thus, we get the following equation by applying Kuhn-Tucker condition:

$$\partial L(\vec{\lambda})/\partial O_{max} = 1 - \sum_i \lambda_i = 0$$

That is,

$$\sum_i \lambda_i = 1$$

We use  $\Phi$  to denote the  $\vec{\lambda}$  satisfying the above relationship for a given routing solution. If  $\vec{\lambda} \in \Phi$ , the objective function  $L^*(\vec{\lambda})$  becomes:

$$L^*(\vec{\lambda}) = \sum_i \lambda_i \sum_k x_{ik} - \sum_i \lambda_i C_i$$

where  $\sum_i \lambda_i C_i$  is a constant for given  $\vec{\lambda}$  and capacities in all bin boundaries.

### 3.4.2.2 Solving Lagrangian Relaxation Subproblem

As mentioned above, we only need to consider solving the Lagrangian relaxation subproblem when  $\vec{\lambda} \in \Phi$ . We use a reasonable heuristic to find the solution for *LRS*. From the equation in previous subsection, the cost of the objective function depends on the term  $\sum_i \lambda_i \sum_k x_{ik}$ . A simple idea to route the nets would be trying to minimize this amount: for each two-terminal net, pick the minimum cost path to route, where the cost  $\phi = \sum_i \lambda_i \sum_k x_{ik}$ . As in [17], we use L-shaped and Z-shaped routing instead of general maze routing.

### 3.4.2.3 Solving Lagrangian Dual Problem

In order to maximize  $L(\vec{\lambda})$  in LDP, we only need to consider those  $\vec{\lambda} \in \Phi$ . First, the approach is to use the solutions of the Lagrangian relaxation subproblem to update the multipliers until the result converges. Here we use *subgradient optimization technique* to search for “optimal”  $\vec{\lambda}$ .

$$\lambda'_i = [\lambda_i + \theta_k (\sum_k (x_{ik} - C_i - O_{max}))]^+$$

where

$$[x]^+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

and  $\theta_k$  is a step length at the  $k$ th iteration such that

$$\lim_{k \rightarrow \infty} \theta_k = 0 \text{ and } \sum_{k=1}^{\infty} \theta_k = \infty.$$

Second, in order to make sure the new updated multiplier  $\vec{\lambda}' \in \Phi$ , we need to *project* the updated multiplier  $\vec{\lambda}'$  back to nearest point  $\vec{\lambda}^*$  in  $\Phi$ .

## 3.5 Experimental Results

We have tested our approaches on 33 blocks with 5K, 10K, and 15K nets, which are generated artificially from *ami33* in MCNC benchmarks to test on huge netlists. All the experiments were carried out on a 300MHz Pentium II Intel Processor. Table 3.1 shows the runtime speedup by using the new improved approach in wiring evaluation. We use this approach on the work in [17] with all test sets in the following way. As in [17], we perform accurate interconnect planning as follows: Apply bounding-box wirelength estimation during high temperature; apply

Table 3.1: Speedup comparisons between floorplanner in [2] and our approaches using net reduction and Lagrangian relaxation global router.

	5K nets	10K nets	15K nets
Approach	Time (sec)	Time (sec)	Time (sec)
Floorplanner in [17]	19874	60124	85026
New Approach	1864	2478	2938

pin assignment and simple geometry routing during medium and low temperature. In order to justify the effectiveness of our approaches, we perform the more accurate global router on the original netlist in both results, which are from [17] and ours. The new improved approach spent less than 50 minutes to finish while [17] took more than 23 hours to finish in a 15K-net test set, for example. The qualities of the floorplans obtained are comparable in area and routability. On the other hand, we get improvement by using more accurate global router in congestion issue for 41.9% average among 5 MCNC benchmarks in terms of maximum overflow. Note that the network flow based algorithm takes some CPU time, which is a much smaller part compared with longer duration in interconnect-centric floorplanning.

### 3.6 Concluding Remarks

We present a simple yet effective idea to significantly reduce the runtime of interconnect-centric floorplanning algorithms in this chapter. Network flow based net decomposition algorithm is used to decompose multi-terminal nets without violating the pin-limit constraint in each block. We also use the Lagrangian relaxation

paradigm to obtain an effective global router in minimizing the maximum overflow in routing region. Combining these two approaches makes the interconnect-centric floorplanning more efficient and effective.

## Chapter 4

# Simultaneous Power Supply Planning and Noise Avoidance in Floorplan Design

With today's advanced ICs manufacturing technology in DSM environment, we can integrate entire electronic systems on a single chip (SoC). However, without careful power supply planning in layout, the design of chips will suffer from local hot spots, insufficient power supply, and signal integrity problems. Post-floorplanning or post-route methodologies in solving power delivery and signal integrity problem have been applied but they will cause a long turn-around time, which adds costly delays to time-to-market. In this chapter, we study the problem of simultaneous power supply planning and noise avoidance as early as in floorplanning stage. We show that the problem of simultaneous power supply planning and noise avoidance can be formulated as a constrained maximum flow problem and present an efficient yet effective heuristic to handle the problem. Experimental results are encouraging. With slight increase in total wirelength, we achieve almost no static IR-drop requirement violation in meeting the power demand requirement imposed by the circuit blocks compared with a traditional floorplanner and 46.6% of improvement on  $\Delta I$  noise constraint violation compared with the approach that only considers power supply planning.

## 4.1 Introduction

With the advent of further technology scaling, circuits which contain more functionality are operating at higher frequencies, currents, and power. The lower supply voltage helps to consume less power dissipation in general, but at the same time narrows down the noise margin of the devices as well. As a result, many effects that were less important in the previous technology of designs have become major factors in correct functionality and performance of these dense chips. In today's new interconnect-centric paradigm [19], power delivery and dissipation, timing, signal integrity and reliability have become as important, or more important, as die area, which was a prime concern for previous technologies.

During manufacturing, the number of silicon failures are caused by signal integrity problems, such as IR-drop,  $\Delta I$  noise, and electromigration. IR-drop and  $\Delta I$  noise may cause circuits' incorrect functioning and timing requirement mismatch, while electromigration may cause the damage of circuits' lifetime. These problems are on the rise due to the lack of existing design tools and methodologies to address these issues effectively. Under these circumstances, as [42] pointed out, the ability to design the chip, the package, and the surrounding system concurrently becomes a primary advantage.

A packaging technique utilizing flip-chip bonding (or Controlled Collapsible Chip Connection, C4) has been developed by IBM for decades to manufacture VLSI chips quickly and cost effectively [18, 33]. Nowadays C4/flip-chip technology is more widely used in microprocessor and high-performance IC manufacturing than wire-bonded technology is. The general C4 chip patterns are shown in

Figure 4.1. The use of area interconnect packaging in high frequency microprocessors is motivated by its high bandwidth and good power distribution capability[26]. However, even though the technology minimizes on-chip voltage fluctuations, difficulty still lies in the interaction of two independent functional blocks that share a power source [23].

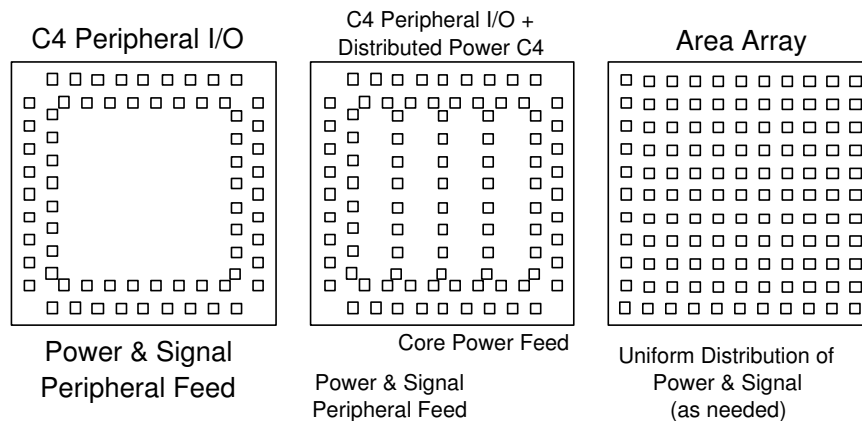


Figure 4.1: C4 chip patterns from IBM online document.

Post-floorplanning or post-route power supply synthesis have been applied to generate satisfactory power supply, trying to meet the requirements of different components in SoC design. Due to reduced power supply voltage, tighter noise margin and DC voltage drop, the task has become difficult [7]. Among the approaches of handling/estimating power delivery [44], the planning of mesh power rail followed by hierarchical power/ground (P/G) networks designs is still a major method to design high-performance IC and microprocessors [3, 62, 68, 69]. Nevertheless, power supply synthesis after floorplanning, even after routing, stage cannot guarantee high-quality power supply under either obviously limited routing resources or infeasible routing constraint being generated. In many cases, when the circuit

block locations and sizes are fixed, the constraints such as voltage drop and current density are so tight that there is no feasible power network design capable of keeping power supply noise within a specified margin. Hence it is important to consider power supply planning during early design stage where the circuit block locations and shapes can be flexibly changed.

There has been a lot of research in floorplanning, e.g., [12, 29, 43, 65]. Parallel to those works, interconnect-driven floorplanning related works [17, 21, 56] have been proposed to extend the capability of floorplanner. However, all these approaches ignored power supply planning. The resultant floorplans may suffer from serious local hot spots and insufficient power supply in some regions. In [13], the authors use two levels of packing for different kinds of P/G network requirements to take power planning into consideration. This model cannot handle the power delivery problem with power supply noise constraint.

High-performance ICs require a robust power delivery network with nominal supply voltage fluctuations. We formulate the problem of simultaneous power supply planning and noise avoidance as a supply-demand problem for power delivery with side constraint for power supply noise requirement. We use a constrained network flow model to represent this problem and handle it with a modified max-flow algorithm. We have incorporated our algorithm into a floorplanning algorithm for integrated floorplanning and power supply planning. (Note that our approach can be applied to any floorplanner as well.) Experimental results are encouraging. Comparing with a traditional floorplanner with no power supply planning at all [65] and a floorplanner with power delivery planning for avoiding hot spots [35],

we obtain floorplans in a fixed die area but significantly better in terms of meeting the IR-drop requirements and minimizing the violations of  $\Delta I$  noise constraint imposed by the circuit blocks. This design can augment the P/G distribution network design and can be an alternative solution other than decoupling capacitance (decap) allocation [70] in power supply noise avoidance.

The rest of the chapter is organized as follows. Section 4.2 describes the floorplan design with power supply noise considerations and problem formulation. The network model and the algorithm for power supply noise avoidance are presented in Section 4.3. Section 4.4 shows our approach for simultaneous power supply planning and noise avoidance in floorplanning. Experimental results are shown in Section 4.5 and Section 4.6 concludes this chapter.

## **4.2 Floorplans with Power Supply Noise Considerations**

Because of DSM technology, chips now contain more functionality and are being driven to higher performance levels than ever before. Furthermore, reduced supply voltage in low power design nowadays tightens the noise margin. Without careful layout planning, the design will suffer from local hot spots, insufficient power supply, and signal integrity problems, among which we focus primarily on IR-drop and  $\Delta I$  noise.

In traditional VLSI design, as [4, 18, 54] pointed out for power supply noise analysis, the resistive IR-drop occurs mostly on the chip and the inductive  $\Delta I$  noise only occurs on the package. IR-drop is voltage drop of the power and ground due to current flowing in the P/G resistive network. However, as we move into DSM

design, the inductive component of wire impedance  $j\omega L$  becomes comparable to  $R$ . The  $\Delta I$  noise, also referred to as simultaneous switching noise (SSN) or ground bounce, is caused by changes in current through various parasitic inductors.

Scientists and engineers have been doing research on accurate transient power analysis, trying to minimize power supply noise across the entire chip. According to [31, 34], SSN has always been a concern in sampled-analog and mixed-mode circuits and a 10% supply voltage fluctuation may translate to more than a 10% timing uncertainty. C4 has been developed to manufacture VLSI chips quickly and cost effectively [18, 33]. C4/flip-chip technology provide high I/O density, uniform chip power distribution, leading-edge cooling capability, and high reliability. An array of PbSn solder balls bumps are arranged around the surface of a chip, either in a peripheral or area array configuration (Figure 4.1 and 4.3). The major advantage of the technology is, after packaging, that the uniform- and low- inductive/resistive power is fed across the face of the die, minimizing on-chip voltage fluctuations that lead to improved voltage tolerances, resulting in improved on-chip frequencies. Also this technology help to replace global on-chip power rails by local power pads/bumps and smaller local rails, which saves chip area [26]. Nevertheless, the technology still suffers the problem of power delivery in mainly two constraints mentioned above.

The primary difficulties occurred in static and transient voltage drop during the planning of power supply in SoC design [23]. Firstly, components in an IC share a common power source that supplies voltage and current to transistors. The transistors draw current when they turn on and off. The power network must be designed

such that voltage and current supplies from power sources are uniformly available to all transistors. IR-drop can have a significant impact on a design. Secondly, the interaction of two independent functional blocks that share a power source causes voltage drop problem. Illustrated in Figure 4.2(b), the voltage fluctuation of block C and its subsequent load on the power bus affect the power supply voltage seen by block D and vice versa. If block D experiences a reduced supply voltage, it exhibits a higher than normal delay and might not function properly.

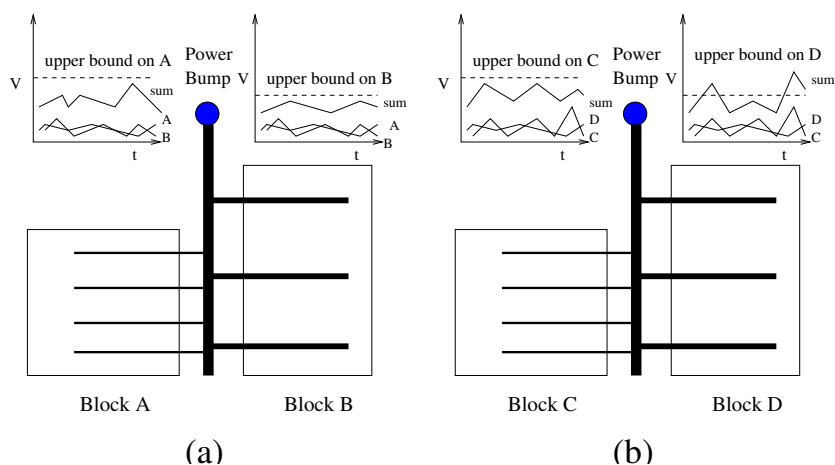


Figure 4.2:  $\Delta I$  noise constraint illustration when sharing power sources. Two examples of two blocks with a power supply bump. The V-t curves are voltage fluctuations for blocks and the superposition seen in power supply bump. (a) The power supply bump is clean since the voltage fluctuations of two blocks are within the upper bound on  $\Delta V$  for both blocks. (b) The power supply bump is noisy since the voltage fluctuations of block C and D exceeds the upper bound on  $\Delta V$  for block D.

From above observation, obviously the positions of blocks are important variables in power supply planning and noise avoidance, meaning floorplanning will affect the quality of power delivery. In Figure 4.3, there are two floorplans

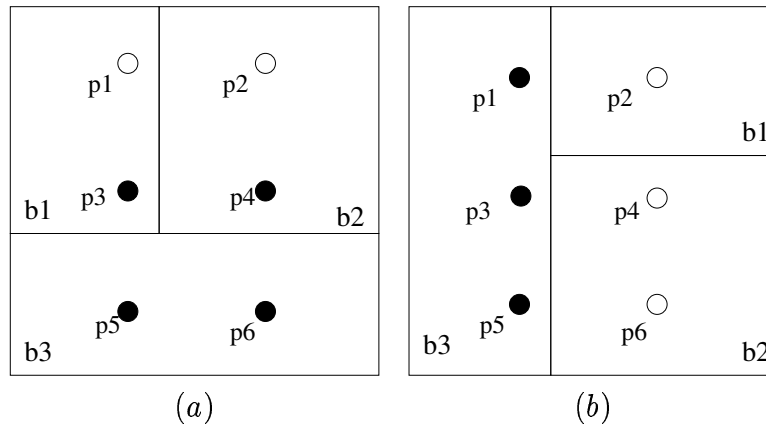


Figure 4.3: Floorplanning affects power supply planning and noise avoidance. Block  $b_3$  can get four power supply bumps to deliver power on the left floorplan while it can possibly only get three power supply bumps on the right one. This block may also suffer from  $\Delta I$  noise constraint violation.

with the same area but different relative position of blocks in area-array design<sup>1</sup>. Block  $b_3$  can get four power supply bumps to deliver power on the left floorplan while it can possibly only get three power supply bumps on the right one<sup>2</sup>. That block may suffer from insufficient power and noise constraint violation, thus fail to function normally. Next we introduce the models of the constraints in our problem formulation.

<sup>1</sup>The area-array design figures shown in this chapter only present power supply bumps and functional blocks. In fact, there are signal bumps in the design as well.

<sup>2</sup>The reason for block  $b_3$  obtaining more power supply bumps in Figure 4.3(a) can be justified by the possible range for a block to draw currents from power source. Next subsection and Section 4.3.1 have more explanation.

### 4.2.1 IR-Drop Requirement

IR-drop is caused by current drawn off of the P/G resistive network. If the wire resistance is too high or the cell current is larger than predicted, an unacceptable voltage drop may occur, which causes the supply voltage to be lower than required. Similar to [8], the following equation gives the effective resistance ( $R$ ) for pad transfer metal from a block to the power supply bump with  $C1$  and  $C2$  being constants derived from simulation, where  $dist(b, p)$  is the distance between the center of the block and power supply bump.

$$R = C1 + C2 * dist(b, p)$$

Using  $I_{avg}$  as the average DC current for the block, we can obtain voltage drop as  $I_{avg} * R$ . Here we try to bound the resistance between a block and its power sources so that blocks can obtain sufficient power to alleviate IR-drop effect. If some blocks can not obtain enough power to meet power requirement, the violation of IR-drop requirement has occurred.

### 4.2.2 $\Delta I$ Noise Constraint

In general, the SSN voltage should be less than some peak voltage for a circuit to operate properly [63]. Here we define this peak voltage as the upper bound on  $\Delta V$  for a block, which is given from intellectual property(IP) block manufacturer. Let  $x_{ij}$  be the amount of power units delivered from power supply bump  $p_i$  to block  $b_j$ ,  $\delta(x_{ij}) = 1$  if  $x_{ij} > 0$ ,  $\delta(x_{ij}) = 0$  otherwise, which means this delivering path will be used,  $(\frac{dI}{dt})_j$  be the maximum rate of current change during transition

at block  $b_j$ , which is assumed to be given from IP provider. Let  $L_i$  be the parasitic inductance for power supply bump  $p_i$ , which is described in technology file,  $L_{ij}$  be the effective wire inductance from power supply bump  $p_i$  to the center of block  $b_j$ <sup>3</sup>. Also let  $S_j$  be the set of all power supply bumps that connect to block  $b_j$  and  $\Delta V_j$  be the upper bound on  $\Delta V$  for block  $b_j$ . We define the parasitic voltage drop from package to power supply bump  $p_i$  as

$$\Delta V_{package} = \sum_h \delta(x_{ih}) L_i \frac{(\frac{dI}{dt})_h}{\sum_{k \in S_h} \delta(x_{kh})}$$

We sum up all the  $L \frac{dI}{dt}$  values from  $p_i$  to all the blocks which the power supply bump delivers power to. This is to reflect the sharing of power sources by adding all the inductive induced voltage drop associated with the power sources. Also each  $L \frac{dI}{dt}$  value is divided by the number of power supply bumps which deliver power to each block in the set  $S_h$  ( $\sum_{k \in S_h} \delta(x_{kh})$ ) due to the sharing of power demand, here we simplify the sharing to be equally divided by the power supply bumps, not very realistic though. We also define the inductive induced voltage drop from  $p_i$  to  $b_j$  as

$$\Delta V_{wire} = L_{ij} \frac{(\frac{dI}{dt})_j}{\sum_{k \in S_j} \delta(x_{kj})}$$

The summation of these two parts for each  $p_i$  and  $b_j$ , which is the total  $\Delta I$  noise induced voltage drop, should be less than or equal to the upper bound on  $\Delta V$

---

<sup>3</sup>Before a design is approved for manufacturing, it is critical that inductance is accurately extracted and modeled for on-chip interconnects [28]. During the floorplanning stage, however, much of the detailed information needed to do this is unknown. So at this stage we use a simple model where inductance is proportional to the distance between the center of the block and power supply bump. This part is for self-inductance only. From [5, 58], the mutual inductance could be on the order of 1/3 or less of the self-inductance and since we do not have routing information here to calculate the exact value, we consider it as lumped inductance.

for block  $b_j$ .

$$\Delta V_{package} + \Delta V_{wire} \leq \Delta V_j$$

Block  $b_j$  can work properly only when the inductive induced voltage drop from package to  $p_i$  and from  $p_i$  to  $b_j$  do not exceed this bound. Figure 4.4 shows the relationship between power supply bump and circuit block.

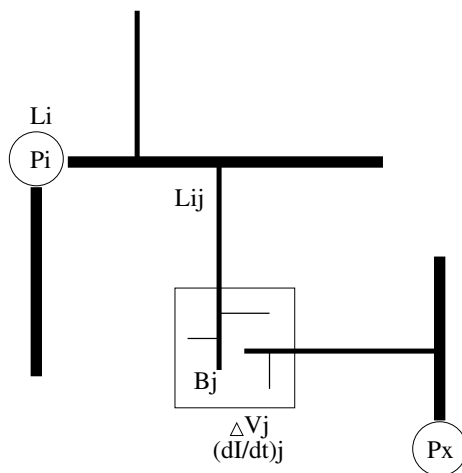


Figure 4.4: The relationship between power supply bump and circuit block. The circuit block can obtain power from several power supply bumps, as long as the noise constraint holds.

### 4.2.3 Problem Formulation

The goal is to search for feasible power delivery distributions with the reduction of the distance between power supply bumps and circuit supply connections and power supply noise minimization, i.e., we try to satisfy the demand of power for every block with delivery path resistance bounded and avoid possible power supply noise during floorplanning. In area-array designs, I/O pads/cells can be placed in-

side the die and they need to be driven directly by power sources. In that case, we can simply treat I/O pads/cells as small circuit macros in our problem formulation.

**Problem 4.2.1.** *Given a floorplan of  $n$  blocks  $b_1, \dots, b_n$  and their minimum power requirements  $d_1, \dots, d_n$ , respectively, and given a set of  $m$  power supply bumps  $p_1, \dots, p_m$  and the maximum power they can deliver  $s_1, \dots, s_m$ , respectively, find a feasible solution such that each circuit block  $b_j$  obtains  $d_j$  units of power from power supply bumps, and each power supply bump  $p_i$  delivers  $s_i$  units of power or less. In addition, the resistance of delivering path from power supply bumps to blocks should be bounded. Meanwhile, the power delivery assignment needs to meet the  $\Delta I$  noise constraint:*

$$\left[ \sum_h \delta(x_{ih}) L_i \frac{(\frac{dI}{dt})_h}{\sum_{k \in S_h} \delta(x_{kh})} \right] + \left[ L_{ij} \frac{(\frac{dI}{dt})_j}{\sum_{k \in S_j} \delta(x_{kj})} \right] \leq \Delta V_j,$$

for each  $p_i, b_j$  s.t.  $\delta(x_{ij}) = 1$

where those symbols are defined in Section 4.2.2.

### 4.3 Power Supply Planning with Noise Avoidance

In order to handle the power supply planning problem along with static IR-drop and noise constraints to be met, we need to develop reasonable and efficient strategies to deal with the constraints. In this section, we define feasible power supply region (FPSR) to consider IR-drop requirement, then introduce the construction of special network for power supply planning based on FPSR with noise avoidance. In preserving the advantage of polynomial time max-flow algorithm, we

also develop an effective algorithm to deal with  $\Delta I$  noise constraint. In addition, we introduce power zones to further reduce the size of the network graph.

### 4.3.1 Feasible Power Supply Region

We try to bound the resistance between a block and its power sources to reflect IR-drop effect. Given the current and the upper bound on  $\Delta V$  for a block, we can derive a region which is an expansion of the block in all four directions by a distance  $r$ . Such a region is referred to as the *feasible power supply region (FPSR)* for the block. Only the power supply bumps within the FPSR of a block can deliver power to the block. In Figure 4.5, the FPSR of block  $b_2$  is within the dashed lines, meaning four bumps  $p_1$ - $p_4$  can supply power to block  $b_2$ .

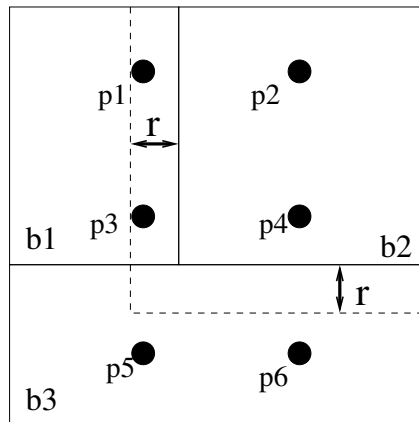


Figure 4.5: A floorplan and the available power supply bumps. A circuit block can use the power supply bumps within its feasible power supply region (FPSR).

### 4.3.2 Constrained Network Formulation

In this subsection, we then construct a special network graph and run a modified max-flow algorithm [2] based on FPSR to solve the problem. The graph consists of two kinds of vertices besides the source  $s$  and the sink  $t$ : the circuit block vertices  $B = \{b_1, b_2, \dots, b_n\}$  and the power supply bump vertices  $P = \{p_1, p_2, \dots, p_m\}$ . To simplify the presentation, we use the same name for a vertex and for the corresponding circuit block or power supply bump interchangeably.

The network graph  $G = (V, E)$  is constructed as follows. There is an edge from the source  $s$  to every power supply bump vertex and there is an edge from every circuit block vertex to the sink  $t$ . The edge capacity from the source  $s$  to a power supply bump vertex  $p_i$  is  $s_i$ , which is the maximum power that can be delivered by  $p_i$ . The edge capacity from a circuit block vertex  $b_j$  to the sink  $t$  is  $d_j$ , which is the minimum power that is required by  $b_j$ . There is an edge from  $p_i$  to  $b_j$  if  $p_i$  is inside the FPSR of  $b_j$ . If such an edge exists, the edge capacity is set to  $\infty$ . We wish to find the maximum flow from the source  $s$  to the sink  $t$  that satisfies the edge capacities and mass balance constraints at all nodes. We can state the problem formally as follows.

Maximize  $v$

subject to

$$\sum_{j:e_{ij} \in E} x_{ij} - \sum_{j:e_{ji} \in E} x_{ji} = \begin{cases} v & \text{for } i = s \\ 0 & \text{for all } i \in V - \{s, t\} \\ -v & \text{for } i = t \end{cases} \quad (4.1)$$

$$\begin{aligned}
& \left[ \sum_h \delta(x_{ih}) L_i \frac{(\frac{dI}{dt})_h}{\sum_{k \in S_h} \delta(x_{kh})} \right] + \left[ L_{ij} \frac{(\frac{dI}{dt})_j}{\sum_{k \in S_j} \delta(x_{kj})} \right] \leq \Delta V_j, \\
& \text{for each } p_i, b_j \text{ s.t. } \delta(x_{ij}) = 1
\end{aligned} \tag{4.2}$$

We refer to  $x = \{x_{ij}\}$  satisfying (4.1) as a flow and the corresponding value of the scalar variable  $v$  as the value of the flow.  $x_{ij}$  is the amount of power units delivered from  $p_i$  to  $b_j$ ,  $\delta(x_{ij}) = 1$  if  $x_{ij} > 0$ ,  $\delta(x_{ij}) = 0$  otherwise.  $(\frac{dI}{dt})_j$  is the maximum rate of current change during transition at  $b_j$ .  $L_i$  is the parasitic inductance for  $p_i$  and  $L_{ij}$  is the effective wire inductance from  $p_i$  to the center of  $b_j$ .  $S_j$  is the set of all power supply bumps that connect to  $b_j$  and  $\Delta V_j$  is the upper bound on  $\Delta V$  for  $b_j$ .

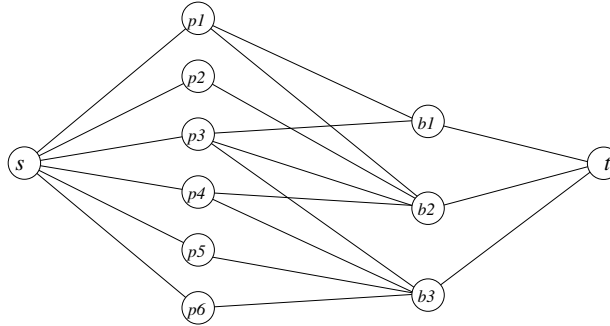


Figure 4.6: The network graph captures the power demand of the circuit blocks and the power that the amount of power supply bumps can provide in Figure 4.5.

Figure 4.6 illustrates the construction of the network graph for the floorplan example in Figure 4.5. Block  $b_2$  can obtain power from power supply bumps  $p_1$ - $p_4$ , as shown in Figure 4.5. The feasible power supply regions of block  $b_1$  and  $b_3$  are not shown in Figure 4.5, where block  $b_1$  can obtain power from power supply bumps

$p_1$  and  $p_2$ , and block  $b_3$  can obtain power from power supply bumps  $p_3$ - $p_6$ . Note that in Figure 4.6 some power supply bump vertices can be connected to two circuit block vertices or more because a power supply bump can supply power to several circuit blocks at the same time, as long as the demanded power never exceeds the maximum power that can be delivered by the power supply bump.

Any flow from the source to the sink in the network assigns power delivering from a power supply bump to a circuit block. If there is a feasible power supply planning solution satisfying all power requirement of the circuit blocks, the total flow on every edge from the source to a circuit block should equal to the edge capacity. It can be shown that our network flow algorithm optimally solves the power supply planning problem if we do not consider the other constraint we have introduced. We have the following theorem.

**Theorem 4.3.1.** *A maximum flow in the network graph corresponds to a power supply planning solution which maximizes the amount of power delivered from the power supply bumps to the circuit blocks. A feasible solution with respect to FPSRs for all blocks exists if and only if all edges from the circuit block vertices to the sink are saturated.*

As can be seen in the problem definition, the side constraints are non-linear, so it may be treated as NP-hard or approximately NP-hard problem. We cannot use min-cost max-flow/min-cut or maximum bipartite matching algorithms to optimally solve this problem. In the following section, we introduce an efficient yet effective algorithm to minimize the violations of  $\Delta I$  noise constraint and still obtain maximum flow.

### 4.3.3 Priority-Augmenting-Path Algorithm

In this subsection, we describe a priority-based heuristic to deal with power supply noise constraint in max-flow algorithm. In Ford-Fulkerson method [22], we try to find any augmenting path to increase the flow. However, randomly pick feasible augmenting path may cause serious violations for noise constraint in power delivery planning. Figure 4.7 shows the constraint violation example when not carefully augmenting the flow. Due to this observation, we implement an efficient algorithm to decide the order of finding augmenting paths based on the priority assigned on the edges between power supply bump vertices and block vertices in our network.

The main point is that we want to choose a path or edge with either low inductive induced voltage drop or large  $\Delta V$  for the block to augment the flow. The reason for low inductive induced voltage drop is obvious: we want to deliver power via low voltage drop to blocks; the reason for large voltage tolerance of blocks on inductive induced voltage is that delivering power to small  $\Delta V$  blocks is harder due to cleaner power supply requirement. We use the following implementation to realize these two reasonings.

We assign the cost first to reflect the rough inductive induced voltage drop without the effect of sharing power demand of the block. The cost for edge  $e_{ij}$  from  $p_i$  to  $b_j$  is  $c_{ij} = (\frac{dl}{dt})_j * (L_i + L_{ij})$ . We then assign priority values for the edges between  $p_i$  and  $b_j$  as follows. Note that for forward and backward direction of edges, we should assign different priority values so that the preferred augmenting path can be found. For forward direction, the priority value  $P_{ij} = \frac{c_{ij}}{N_j} + \frac{1}{\Delta V_j}$ ; for

backward direction, the priority value  $Q_{ji} = \frac{N_j}{c_{ij}} + \Delta V_j$ , where  $N_j$  is the current number of power supply bumps which deliver power to block  $b_j$ .  $N_j$  needs to be updated whenever we obtain an augmenting path and augment the flow since the intermediate flow solution has been modified.

During the process of finding an augmenting path, we can use the priority values to select a preferred path. In this way, finding augmenting path which minimizes the violations of noise constraint can be accomplished. We have the following algorithm.

**Algorithm** Priority\_Augmenting\_Path  
**begin**  
 $x := 0$ ;  
**while**  $G(x)$  contains a directed path from  $s$  to  $t$  **do**  
    Identify an augmenting path  $U$  from  $s$  to  $t$   
    based on priority of the edge from some  
    power supply bump to some block;  
     $\nu := \min\{r_{ij} : e_{ij} \in U, i, j \in V\}$ ;  
    Augment  $\nu$  units of flow along  $U$ ;  
    Update  $G(x)$  and  $N_k, k \in B$ ;  
**end**  
**end**

In the algorithm,  $x$  is the flow vector,  $G(x)$  is the residual network,  $r_{ij}$  is the residual capacity for edge  $e_{ij}$ , and  $\nu$  is the residual capacity of the augmenting path  $U$  [2].

#### 4.3.3.1 Complexity Analysis

We use Edmonds-Karp algorithm to implement pure max-flow problem, the runtime is  $O(|V||E|^2)$  [22]. To be more specific, the number of iterations is at

most  $O(|V||E|)$  and each iteration of Ford-Fulkerson method can be implemented in  $O(|E|)$  time using breadth-first search (BFS). We can use Fibonacci heap to implement priority queue and obtain logarithmic runtime in operations. In addition, the dequeue and enqueue operations in BFS both take  $O(1)$  time originally, but in our proposed algorithm, they take  $O(\lg |V|)$  time. The update of the number of power supply bumps which deliver power to blocks can be done in  $O(|V||E|)$  since they are only updated when we obtain augmenting paths. The runtime of the priority augmenting path algorithm hence is  $O(|V||E|(|V| \lg |V| + |E|))$  time. We have the following corollary.

**Corollary 4.3.2.** *The priority augmenting path algorithm with prioritized breadth-first search solves max-flow problem and heuristically minimizes side constraint violations in  $O(|V| \lg |V| + |E|)$  time. Hence the modified Edmonds-Karp algorithm runs in  $O(|V||E|(|V| \lg |V| + |E|))$  time.*

#### 4.3.4 Graph Reduction by Power Zones

Although the proposed approach runs correctly, there can be numerous power bump vertices in the network graph when the number of the power bumps is large, making the graph size large. In this subsection, we introduce “power zones” to reduce the network graph size. A power zone represents the collaborative efforts of the power bumps in the proximity. By introducing power zones, the power bump vertices are dropped from the graph. This simplification gives a power supply planning as good as the method previously described, but at a reduced CPU time for running the network flow algorithm.

Precisely, power zones are the disjoint regions bounded by the boundaries of the FPSRs (See Figure 4.8). The network graph is then constructed as follows. Similarly, there is an edge from the source  $s$  to every power zone vertex and there is an edge from every circuit block vertex to the sink  $t$ . Again, the edge capacity from the a circuit block vertex  $b_i$  to sink  $t$  is still  $d_i$ . However, the edge capacity from source  $s$  to power zone vertex  $z_j$  is set to be the sum of the maximum power the power bumps inside the corresponding power zone can deliver, i.e.,  $\sum_{p_k \in z_j} s_k$ . There is an edge from zone  $z_i$  to block  $b_j$  if  $z_i$  is inside the FPSR of  $b_j$ . If such an edge exists, the edge capacity is again set to be  $\infty$ .

Here we show a way to obtain power zones out of a floorplan. First, we allocate a “zone index” (which is a binary number) for each power bump. Depending on which FPSRs a power supply bump is contained, a proper zone index is set. This can be done by setting the bits (in the binary number) corresponding the FPSRs containing the power supply bump to be one, and keep remaining bits to be zero. The detail of this is best explained by an example.

Consider a floorplan in Figure 4.8. The power supply bump  $p_7$  is contained by the FPSRs of blocks  $b_2$ ,  $b_3$  and  $b_4$ . We set first, second and fourth bits in the binary number to be one and keep other bits to be zero. Therefore, the zone index of  $p_7$  is set to  $b'01110$ . Similarly, the zone index of  $p_{11}$  will be  $b'01110$ . Therefore they are in the same zone. On the other hand, the zone index of  $p_{10}$  will be  $b'01011$  since  $p_{10}$  is inside the FPSRs of blocks  $b_1$ ,  $b_2$  and  $b_4$ . Because the zone index of  $p_{10}$  is different from those of  $p_7$  and  $p_{11}$ ,  $p_{10}$  is in different power zone. By sorting power supply bumps according to the zone index, power bumps in the same power

zone can be easily grouped into clusters in  $O(m \lg m)$  time.

#### **4.4 Floorplanning with Power Supply Planning and Noise Avoidance Design**

Our floorplanning algorithm with simultaneous power supply planning and noise avoidance is based on the Wong-Liu floorplanning algorithm [65]. Recall that the Wong-Liu algorithm uses Polish expressions to represent floorplans and searches for an optimal floorplan using simulated annealing by iteratively generating Polish expressions. Once a Polish expression is examined, the shapes of the blocks are optimized and the total wirelength is used as the interconnect cost. In addition to optimizing total wirelength and chip area, we propose to perform simultaneous power delivery planning and power supply noise avoidance design with respect to the current floorplan being considered and in result to obtain a much better floorplan with less power supply noise constraint violations.

We choose to optimize the floorplan in fixed die context, but our approach can also comply with the objective of minimizing chip area in floorplanning. In fact, it is intuitive to implement this feature in shape curve computation. During shape curve computation of the floorplan generation in minimal area and wirelength, we can search for the nearest point on the final shape curve to match the given aspect ratio of fixed die. In this way, we will not obtain the solution which has too small value of x or y dimensions in shape curve, and eventually we can obtain the floorplans which are inside the fixed die.

The cost function used to evaluate a floorplan in [65] is  $A + \lambda W$ , where  $A$  is

the total area of the packing,  $W$  is the half-perimeter estimation of the interconnect cost, and  $\lambda$  is a constant which controls the relative importance of these two terms. In this chapter, we use the cost function  $\alpha A + \beta W + \gamma P$  for floorplanning with simultaneous power supply planning and noise avoidance, where  $A$  can be either total area of the packing or fixed die penalty if using fixed die implementation, which is zero if the area of floorplan is within the fixed die and is the difference between the area of current floorplan and fixed die area otherwise,  $W$  is total wirelength estimation<sup>4</sup>, and  $P$  is the power supply cost penalty, which is positive if the current floorplan cannot find max-flow solution and/or obtain the violations of power supply noise constraint. The coefficients  $\alpha$ ,  $\beta$ , and  $\gamma$  are weighting parameters and can be changed due to the importance of the terms.

## 4.5 Experimental Results

We have tested our approach on some MCNC building blocks benchmarks. All experiments were carried out on 650MHz+ Pentium-III processor. The minimum power required by a circuit block and the max rate of current change during transition at a circuit block are roughly proportional to its area. The power supply bumps are in a regular array structure and the maximum amount of power they can deliver are all the same since C4 provides uniform power distribution. (In fact, our approach can be applied to other equivalent structures.) The values of parasitic and wire inductance and other technology parameters are from ITRS'97 roadmap [1],

---

<sup>4</sup>Here we use simplified model for timing constraint since we consider floorplans which need immediate attention to power supply planning and signal integrity issues.

.18 $\mu$ m. As for bump pitch, we use the scaling number from [8]. In order to show the effectiveness of our approach, we implement three algorithms: the traditional approach without any power supply planning consideration [65], the approach with rough IR-drop requirement consideration in power supply planning (in [35]), and feature approach in simultaneous power supply planning and noise avoidance.

Table 4.1: Comparison of our approach with [65] and [35] on MCNC benchmarks. The wirelength data are described in Section 4.5.

Data	Block#	Traditional Floorplanner [65]		Floorplanner with Power Supply Planning [35]			Simultaneous PSP-NA		
		IR-drop Vio(%)	Noise Vio(%)	IR-drop Vio(%)	Noise Vio(%)	Time (hr)	IR-drop Vio(%)	Noise Vio(%)	Time (hr)
apte	9	0	54	0	54.6	0.2	0	3.9	0.24
xerox	10	0	61	0	63.2	0.6	0	9.1	0.44
hp	11	27.3	66	0	61.4	0.11	0	7.3	0.1
ami33	33	31.3	48	3.1	47.4	1.3	3.1	10.1	1.7
ami49	49	4.1	61	0	45.5	3.6	0	7.6	3.74
Average		12.54	58	0.62	54.2		0.62	7.6	

Table 4.1 shows the comparison between the floorplans obtained by our approach, those obtained by a traditional floorplanner in [65] without any power supply planning consideration, and those obtained by the approach (in [35]) with only supply-demand power supply planning consideration during the annealing process. All the floorplans obtained are within a fixed die area with 7% dead space. We use IR-drop violation and noise violation (in percentage) to reflect the effectiveness. Since we use FPSR to bound the power delivering path’s resistance to prevent static IR-drop violation, we thus use a percentage, which is the number of blocks which obtain insufficient power due to IR-drop effect divided by total number of blocks, to show IR-drop requirement violation.  $\Delta I$  noise constraint violation percentage

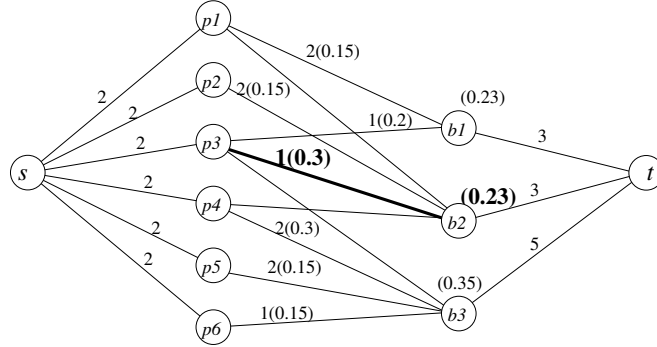
is the number of power supply bump-block edge constraint violation normalized by the number of total power supply bump-block edge in the network. From Figure 4.7, we can see that if there is no violating edge in the network graph, the  $\Delta I$  noise constraint violation percentage is 0%. The floorplans obtained by applying our approach have much less IR-drop violation, over 50% improvement on  $\Delta I$  noise constraint violations and less than 5% of total wirelength increase in average compared with the floorplan obtained in [65]. We have listed the CPU time for those benchmarks, in which we spent less than 4 hours for ami49.<sup>5</sup>

## 4.6 Concluding Remarks

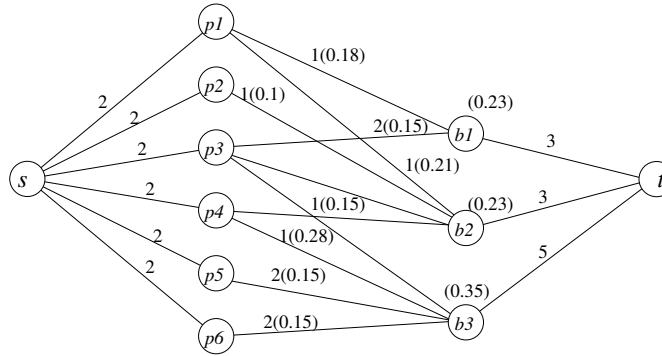
We have presented an approach to simultaneously solving power supply planning and noise avoidance in floorplan design. The efficient yet effective priority-based heuristic we have introduced ensures the polynomial time max-flow algorithm for this difficult problem and experimental results are encouraging. With slight increase of total wirelength, we can obtain big improvement on IR-drop and  $\Delta I$  noise constraint violations in the floorplanning stage.

---

<sup>5</sup>Note that the running time for both power supply planning algorithms are around the same scale, meaning the complexities are about the same.



(a)



(b)

Figure 4.7: Numeric examples include two max-flow solutions of the network graph from Figure 4.6. Those number are calculated from technology and given IP parameters. (a) The solution with randomly choosing augmenting path. The darker numbers and the edge show that there is a  $\Delta I$  noise constraint violation. The number on the edge is the amount of flow on that edge. The number inside the parentheses on the edges between power supply bumps and blocks is the amount of inductive induced voltage drop on that edge. The number inside the parentheses above the block node is the upper bound on  $\Delta V$  for the block. For example,  $e(p1, b1)$  has 0.15mV for inductive induced voltage drop, which does not exceed  $\Delta V_1=0.23$ mV. But for  $e(p3, b2)$ , it has 0.3mV, which exceeds  $\Delta V_2=0.23$ mV, indicating a violation. (b) The solution using the algorithm in Section 4.3.3. There is no  $\Delta I$  noise constraint violation.

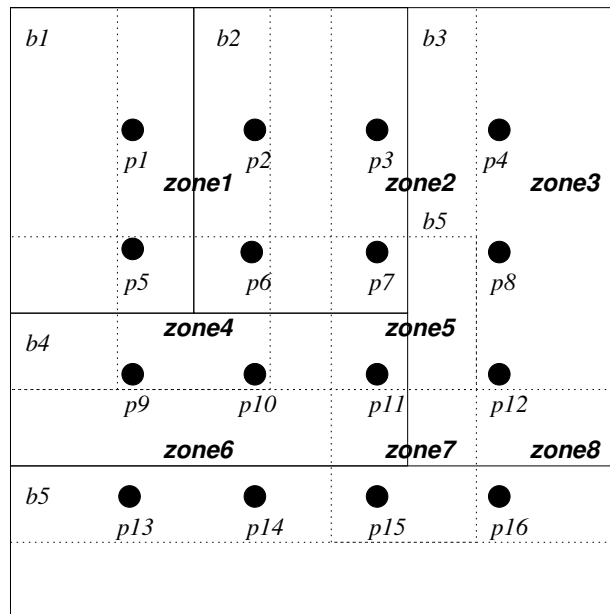


Figure 4.8: The power zones in a floorplan.

## Chapter 5

### I/O Buffer Site Placement in Area-Array IC Design

I/O placement has been a concern in modern IC design. Due to flip-chip and multi-chip module technologies, I/O can be placed throughout the whole chip without long wires from the periphery of the chip. However, because of I/O placement constraints, which includes excessive power demand and voltage drop, and the decision of positions for placing I/O buffers in an existing design being critical, I/O buffer placement becomes a pressing problem. Meanwhile, it is costly to build an I/O buffer site in a design, which can accommodate a certain amount of I/O buffers, according to modern design methodology. In addition, the wirelength cost minimization also needs to be considered while placing I/O buffers. In this chapter, our objective is trying to minimize the number of I/O buffer sites and to decide their positions in an existing standard cell placement. We formulate it as a minimum cost flow problem minimizing  $\alpha W + \beta D$ , where  $W$  is the total wirelength of the placement and  $D$  is the total voltage drop in the power network. The experimental results are encouraging.

## 5.1 Introduction

With today's advanced ICs manufacturing technology in DSM environment, we can integrate entire electronic systems on a single chip. Since more I/Os are needed in current designs, I/O placement has been a major concern in designing high-performance ICs. Flip-chip and MCM technologies now allow high-performance ICs and microprocessors to be built with many more I/O connections than in the past [10, 11], among which area-array bonded connection (Figure 5.1) is considered a better choice [40, 55]. Since area-array style allows I/O buffers to be placed anywhere on the die, we need to be aware of I/O buffer placement constraints, which includes power delivery problem since I/O buffers are power-demanding, to better the design. Another constraint in modern methodology is the cost for building I/O buffer sites in an existing cell placement.

There were some approaches/methodologies for this problem. In [8, 26, 67, 71], similar methodologies for I/O cell placement and electrical checking using flip-chip technology have been presented. They also have graphic or interactive I/O placement tool to provide some constraints checking, trying to avoid hot-spot problem. Recently, [30] further developed a greedy algorithm to place I/O buffers in an ILP formulation of voltage drop constraint. In [36], they utilized area I/O flip-chip packaging to minimize interconnect length, which is a major metric for cell and I/O placement optimization. However, those approaches failed to consider the building cost of I/O buffer sites. There is a certain amount of cost to generate an I/O buffer site, which can be treated as a cluster of I/O buffers. If we just place I/O buffer in greedy ways [26, 61], which is to greedily minimize IR drop and wirelength by

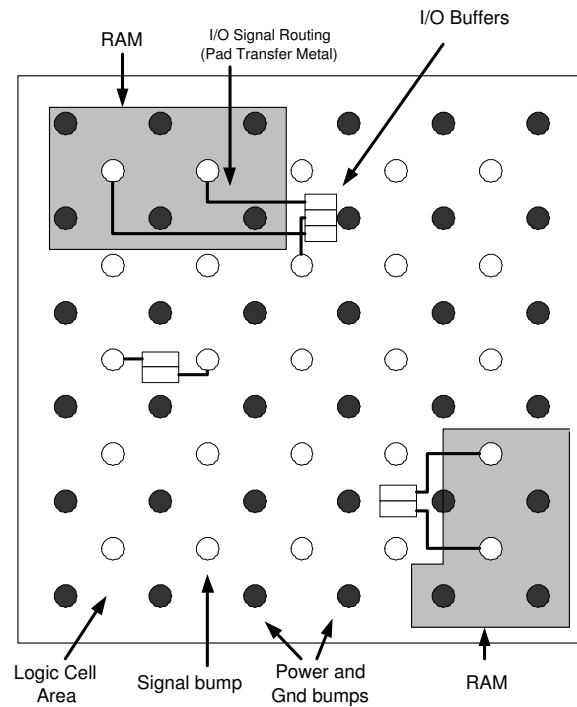


Figure 5.1: Area-array footprint ASIC. The Vdd and Gnd bumps are uniformly distributed across the die with signal bumps in fixed interspersed locations. I/O buffers are associated with some specified signals bump and connected by pad transfer metal.

placing them near the signal bumps and power bumps, such approaches will end up generating more I/O buffer sites and increasing the design cost.

In this chapter, we study the problem of I/O buffer site placement in area-array application specific IC (ASIC) designs and propose an algorithm to solve the problem with respect to design cost reduction. Our objective is to reduce the number of I/O buffer sites and to decide their positions in an existing standard cell placement. We formulate it as a min-cost maximum flow problem minimizing  $\alpha W + \beta D$ , where  $W$  is the total wirelength of the placement and  $D$  is the total voltage drop in

the power network. We also describe an approach to utilizing the result of I/O buffer site planning, in order to further re-place some cells for wirelength minimization. It is based on the force-directed approach in [25, 53, 57].

The rest of the chapter is organized as follows. Section 5.2 describes the I/O buffer site placement considerations and problem definition. The algorithm for I/O buffer site planning is presented in Section 5.3. Section 5.4 shows an approach to refining existing cell placement by using the result of I/O buffer site planning. Experimental results are shown in Section 5.5 and concluding remarks are presented in Section 5.6.

## **5.2 I/O Buffer Site Placement in Area-Array IC Design**

In order to keep up the performance in technology advances, flip-chip and MCM technologies now allow high-performance ICs and microprocessors to be built with many more power and I/O connections than in the past, among which area array bonding is considered a rather better one. Besides helping solve the power delivery engineering problems, to effectively alleviate voltage drop problem we need to focus on the placement of highly power hungry buffers, I/O buffers. Since area-array style allows I/O buffers to be placed anywhere on the die, we need to be aware of I/O buffer placement constraints to better the design.

The design will suffer mainly hot-spot problem [30] and long interconnect length [36] if not carefully planning I/O buffers. From the footprint of ASIC in area-array design (Figure 5.1 in [8]), I/O buffers are placed near signal bumps, one I/O buffer is connected to one signal bump. Those buffers also need to be placed

near power bump to consume power, avoiding large IR drop and long interconnections. Furthermore, some areas can not be used for placing I/O buffers, such as RAMs. [30] lists the placement constraints, mainly keeping voltage drop below the threshold in power sources when placing I/O buffers.<sup>1</sup>

The analysis of the effect of I/O buffer placement on the performance of power grids requires modeling the grids as well as the power sources and drains [30, 46]. For efficient analysis of power supply network, power grids are modeled as linear RC networks, power sources are modeled as simple constant voltage sources, and power drains are modeled as independent time-varying currents (Figure 5.2). On-chip inductance is ignored for now since it is too small to affect the analysis results in today's technology.

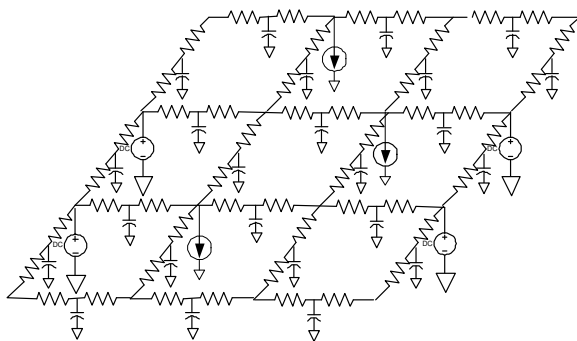


Figure 5.2: Power supply network in area-array design for efficient analysis. Power grids are modeled as linear RC networks, power sources are modeled as simple constant voltage sources, and power drains are modeled as independent time-varying currents.

We adopt part of the analysis and argument in [30] for solving I/O buffer

---

<sup>1</sup>However, we have ignored some other I/O placement constraints - many of which have yet to be precisely characterized - that arise from decoupling and ESD requirements.[9]

placement problem. The behavior of the system can be expressed in the modified nodal analysis (MNA) [51] formulation as the following ODE:

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

where  $x$  is a vector of node voltages and source currents,  $G$  is the conductance matrix,  $C$  includes the capacitance terms, and  $u(t)$  includes the contributions from the sources and the drains. Applying backward euler (BE) numerical integration, we can express the resultant linear equations as:

$$Ax(t+h) = u(t+h) + x(t)C/h$$

where  $A = G + C/h$ . The system matrix  $A$  can be shown to be symmetric, and further reformulated to be nonsingular  $\mathcal{M}$ -matrix [6]. Since the DC solution is a prudent, conservative, and practical approach to the problem, the system equation becomes  $Ax = u$ , where  $A = G$ . What we care is the voltage drop in power grids, so we reformulate the equation as  $A\delta = b$ , where  $\delta = V - x$  is the vector of voltage drops and  $b$  is the vector of current sources. In other words,  $b$  can be seen as the following:

$$b_i = \sum_{k=1}^n d_{ik} I_k, \forall i$$

where  $I_k$  is the current associated with buffer  $io_k$  and  $d_{ik} = 1$  if  $io_k$  consumes the power from node  $p_i$ ,  $d_{ik} = 0$  otherwise,  $n$  is the number of I/O buffers. Therefore, the relationship between the voltage drop at node  $p_j$  and all the entries of the vector  $b$  is:

$$\delta_j = \sum_{i=1}^m a_{ji}^{-1} b_i, \forall j$$

where  $a_{ji}^{-1}$  is the element on the row  $j$  and column  $i$  of the inverse of system matrix  $A^{-1}$ ,  $m$  is the number of nodes. The problem is to place a given set of I/O buffers while suppressing the voltage drop to below the user-specified voltage drop thresholds, denoted by  $\delta_{max}$ .

On the other hand, generating minimal number of I/O buffer sites is another major objective during cell placement. There is a certain amount of cost to generate an I/O buffer site, which includes the setup of extra space to accommodate cells and power delivery path. If we just place I/O buffer in greedy ways, which is to greedily minimize IR drop and wirelength by placing them near the signal bumps and power bumps, such approach will end up generating more I/O buffer sites and increase the design cost.

The problem **IBSP**(I/O Buffer Site Planning) is described as follows.

**Problem 5.2.1. IBSP:** *Given an existing standard cell placement with cell dimensions and positions, a set of I/O buffers (which also includes the set of corresponding signal bumps)  $IO = \{io_1, \dots, io_n\}$  and the current  $I_i$  associated with I/O buffer  $io_i$ , a set of power bumps  $P = \{p_1, p_2, \dots, p_m\}$ , a user-specified voltage drop threshold vector  $\delta_{max}$ , the system matrix  $A$  for power network, a certain cost of building one I/O buffer site, and a set of nets  $N = N_1 \cup N_2 \cup \dots \cup N_k$ , find a solution to simultaneously reduce the number of I/O buffer sites, the total wirelength for the placement, and voltage drop threshold violation for power network.*

We divide the whole die into bins based on signal bumps. Each bin has a certain amount of area for accommodating I/O buffers, based on the dead space

or other pre-planned free space in existing placement. For some bins which are occupied fully or partially by memory blocks, the area of corresponding bins will be zero or less than a certain amount. We define  $H = \{h_1, \dots, h_n\}$  to be the set of regions that the buffer  $io_i$  can possibly draw current from, which is shown in Figure 5.3, similar to [35]. Each region contains a set of power bumps that the I/O buffer can use. In next section, we introduce a cost function to minimize the wirelength and total voltage drop in power network and present an algorithm to solve the proposed problem with clustering the I/O buffers into buffer sites.

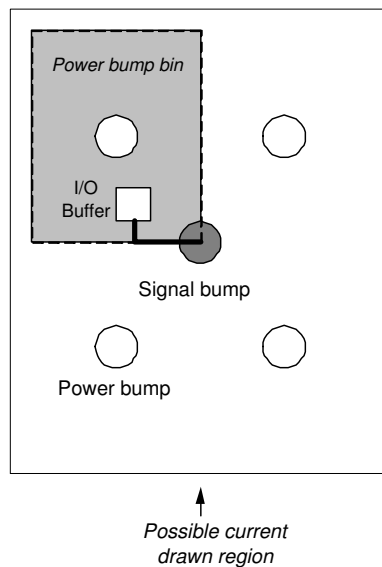


Figure 5.3: The relationship between signal bump, power bump, power bump bin, I/O buffer possible positions, and possible current drawn region.

### 5.3 The Algorithm

We first construct a network graph with embedded cost function and run a min-cost flow algorithm [2] to obtain the solution. The network graph  $G = (V, E)$  is constructed as follows, also see Figure 5.4 for illustration.

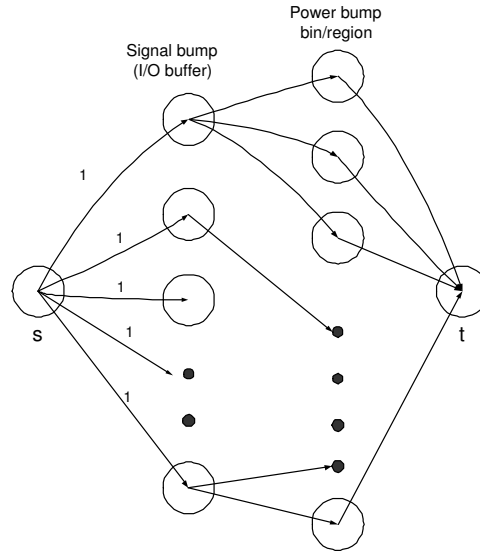


Figure 5.4: Network construction for IBSP. Some signal bump (corresponding I/O buffer) vertex  $io_i$  only connects to power bump bin vertices which are inside the possible current drawn region for  $io_i$ .

1.  $V = \{s, t\} \cup IO \cup P$ , where  $s$  is the source vertex,  $t$  is the sink vertex,  $IO = \{io_1, \dots, io_n\}$  is the set of I/O buffers (each one has corresponding signal bump), and  $P = \{p_1, p_2, \dots, p_m\}$  is the set of power bumps.
2.  $E = \{(s, io_i) | io_i \in IO\} \cup \{(io_i, p_j) | io_i \in IO, p_j \in P \cap h_i\} \cup \{(p_j, t) | p_j \in P\}$ , where  $h_i$  is the corresponding possible current drawn region for  $io_i$ .
3. Edge capacity:  $U(s, io_i) = 1, U(io_i, p_j) = 1, U(p_j, t) = \infty$ .

4. Vertex Capacity:  $U(p_i)$  = upper bound of the number of I/O buffers that bin  $p_i$  can accommodate. Other vertices are incapacitated.
5. Cost function:  $C(i_{o_i}, p_j) = \alpha W_{ij} + \beta I_i \sum_{k=1}^m a_{kj}^{-1}$ , where  $W_{ij}$  is the wire-length estimation if I/O buffer  $i_{o_i}$  is placed at bin  $p_j$ , along with the computation with other internal logic modules or cells,  $a_{kj}^{-1}$  is the element on the row  $k$  and column  $j$  of the inverse of system matrix  $A^{-1}$ . For other edge  $e \in E, C(e) = 0$ .

Note that we need to capacitate power bump bin vertices and classical network flow problem only capacitates edges. This can be resolved by splitting the capacitated vertex  $r$  into two vertices  $r'$  and  $r''$ , adding an edge  $(r', r'')$  with capacity  $U(r)$  and cost 0, and turning the original edges  $(u, r)$  and  $(r, v)$  into edges  $(u, r')$  and  $(r'', v)$  respectively (Figure 5.5).

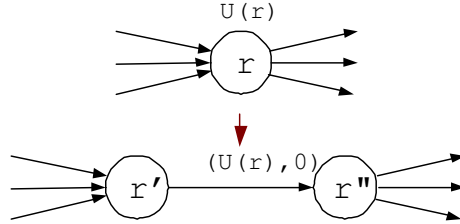


Figure 5.5: Vertex splitting for capacitated vertices. The new edge has capacity  $U(r)$  and cost 0 [66].

Any flow in the network can be mapped into an I/O buffer planning site solution for a subset of given I/O buffers. If a flow  $f$  exists and  $|f| = n$ , we can assign all I/O buffers to buffer sites in given power bump bins. And since the cost of the flow is the cost for the solution of I/O buffer placement, minimum cost flow

guarantees a solution with minimum total cost  $\alpha W + \beta D$ , where  $W$  is the total wirelength and  $D$  is the total voltage drop in power network. The total capacities of edges going from source vertex  $s$  is  $n$ , so the maximum flow  $|f_{max}| = n$ . We have the following theorem.

**Theorem 5.3.1.** *A min-cost flow  $f$  in  $G$  corresponds to an I/O buffer site planning solution to IBSP problem with minimum total cost:  $\alpha W + \beta D$ . A min-cost maximum flow assigns all I/O buffers in IO with minimum total cost.*

Here is the algorithm IBSP-McMaxflow (IBSP problem solved by Minimum cost Maximum flow algorithm).

#### **Algorithm IBSP-McMaxflow**

1. Construct the network graph  $G$ .
2. Assign capacities  $U$  and cost  $C$ .
3. Apply min-cost maximum flow algorithm on  $G$ .
4. Derive the I/O buffer site planning solution.

Finding a min-cost maximum flow in a network is a classical problem for which several polynomial-time optimal algorithms are available [2, 22]. We use capacity scaling algorithm to solve the network in  $O((m \lg U)(m + n \lg n))$  time [2], where  $n = |V|$ ,  $m = |E|$ , and  $U$  is the upper bound of the edge capacity.

Here we have presented an approach to clustering I/O buffers to buffer sites in order to reduce design cost. Since we estimate utilizable space for buffer sites in power bump bins, we need to move part of the existing cells around to accommodate

the sites. In next section, we propose a method to further refine the whole placement by using the result of I/O buffer site planning.

## 5.4 Cell Re-Placement from I/O Buffer Site Planning

We propose an approach to better the existing placement, based on force directed placement described in [25, 53, 57]. Force directed placement explores the similarity between placement problem and classical mechanics problem of a system of bodies attached to springs. In this method, the blocks connected to each other by nets that are supposed to exert attractive forces on each other. The magnitude of this force is directly proportional to the distance between the blocks.

The cells are usually categorized as movable or fixed and the I/O buffers are considered fixed. Let  $B = \{b_1, \dots, b_l\}$  be the cells. We also have a set of I/O buffer sites  $IO = \{io_1, \dots, io_n\}$ . Let  $(x_i, y_i)$  be the Cartesian coordinates for  $b_i$  and  $io_i$ ,  $\Delta x_{ij} = |x_i - x_j|$ ,  $\Delta y_{ij} = |y_i - y_j|$ ,  $\Delta d_{ij} = \sqrt{(\Delta x_{ij})^2 + (\Delta y_{ij})^2}$ . Let  $F_x^i$  be the total force enacted upon  $b_i$  and  $io_i$  by all other cells or I/O buffers in  $x$ -direction. Then the force equations can be expressed as:

$$F_x^i = \sum_{j=1}^{l+n} [-k_{ij}\Delta x_{ij} + \delta k_{ij}R\Delta x_{ij}/\Delta d_{ij}], i = 1, \dots, l+n$$

where  $k_{ij}$  is the attractive constant between cells and/or I/O buffers and  $k_{ij} = 0$  if  $i = j$  or they are not connected;  $\delta k_{ij} = 1$  when  $k_{ij} = 0$ , and  $\delta k_{ij} = 0$  when  $k_{ij} = 1$ ;  $R$  is the repulsion constant. The formulation of the force equation in  $y$ -direction is the same as in  $x$ -direction.

The placement problem becomes a problem in classical mechanics and the

variety of methods can be applied. One method to solve for the set of force equations is to set the potential energy equal to  $\sum_{i=1}^{l+n} [F_x^{i^2} + F_y^{i^2}]$ , and apply the unconstrained minimization method, such as Fletcher-Reeves method [37], since the solution correspond to the state of zero potential energy of the system.<sup>2</sup> The illustration of the approach is in Figure 5.6.

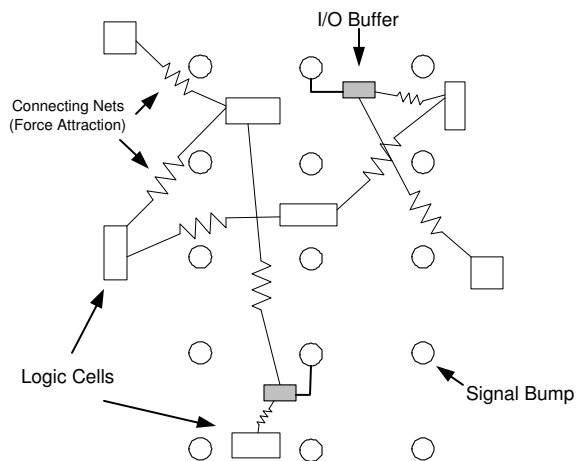


Figure 5.6: Force-directed based cell re-placement. With fixed I/O buffer locations, use force attraction to refine existing placement.

## 5.5 Experimental Results

We have implemented our algorithm and run on 650MHz Pentium III machine. The existing cell placements based on some MCNC benchmarks (in Table 5.1) are obtained from the placer *FENG SHUI* in [39], with aspect ratio 1.0. Note that the number of cells includes the number of terminals/I/Os in Table 5.1.

<sup>2</sup>Note that this approach should be applied along with effective strategies for mixing standard cells with macro cells (I/O buffer sites) and overlap-free placement.

Table 5.1: Number of cells, nets, and IO/terminals in some MCNC standard cell placement benchmarks.

Benchmark	Cells	Nets	IOs
fract	149	163	24
struct	1952	1920	64
biomed	6514	7052	97
industry1	3085	2594	814
industry2	12637	13419	495
industry3	15433	21967	374

We have adopted the following abstract model of I/O regimes from [9] for our experiments:

- I/O buffers must be placed exactly at pad locations, and any I/O buffer can be placed at any pad location.
- No two I/O buffers can occupy the same location.
- For a design with I/O buffers and a rectangular core layout region, we fix pad locations with an array of locations spaced uniformly within the core layout region.

The number of power bumps and signal bumps are scaled from IBM SA-27E area-array copper technology [8].

Table 5.2 shows the experimental results of our approach on MCNC benchmarks summarized in Table 5.1. Those results have been compared with a greedy approach to minimizing wirelength and IR-drop. This approach [61] is that the area array pads are placed at fixed sites on the top layer and each of the I/O ports is

routed to the closest pad. The voltage drop threshold violation percentage shown in the table is obtained by the number of nodes whose voltage drop exceeds the threshold normalized by total number of nodes. From the performance comparison shown in Table 5.3, we can obtain much less number of I/O buffer sites (up to 42.8% reduction) with slight increase percentage of voltage drop threshold violation in power nodes. Those violation increases, possibly due to the trade-off between wirelength and total voltage drop, can be eliminated by careful refinement in local cell and buffer locations. This table also shows the I/O wirelength comparison results, meaning they are comparable in both methods.

Table 5.2: Experimental results of our approach on MCNC benchmarks summarized in Table 5.1, compared with a greedy approach. With slight increase percentage in voltage drop threshold violation, much less number of I/O buffer sites can be obtained in minimizing wirelength.

Benchmark	# power bump	# signal bump	w/o Planning			Planning		
			Time (sec)	# i/o buf site	vol vio (%)	Time (sec)	# i/o buf site	vol vio (%)
fract	156	132	<1	22	3.8	<1	21	1.9
struct	306	272	<1	60	3.3	<1	53	5.5
biomed	342	306	<1	88	0.5	1	74	2.3
industry1	1116	1050	1	682	0.1	32	481	1.3
industry2	676	625	<1	417	0.2	11	292	2.5
industry3	576	529	<1	325	0.1	7	241	1.04

## 5.6 Concluding Remarks

We have presented an approach to simultaneously reducing the number of I/O buffer sites, the total wirelength, and voltage drop threshold violation in existing cell placement. We formulate the problem as a min-cost maximum flow problem

Table 5.3: Performance improvement of our approach on MCNC benchmarks, compared with a greedy approach [61].

Benchmark	# power bump	# signal bump	Comparison	
			imp (%)	I/O WL (x)
fract	156	132	14.3	2.2
struct	306	272	13.2	0.62
biomed	342	306	18.9	1.16
industry1	1116	1050	41.8	0.002
industry2	676	625	42.8	0.98
industry3	576	529	34.9	1.35

minimizing  $\alpha W + \beta D$ , where  $W$  is the total wirelength and  $D$  is the total voltage drop in the power network. The experimental results are encouraging. With slight increase percentage of voltage drop threshold violation, we can obtain much less design cost in I/O buffer site placement.

## **Chapter 6**

### **Conclusion and Future Directions**

In this chapter, we briefly summarize the overall results which are presented in the preceding chapters. We also discuss several perspectives and future research directions which are put aside from the dissertation due to time limitation on the work.

We have proposed a method to integrate interconnect planning with floorplanning. We perform pin assignment and fast global routing during every iteration of floorplanning. We use a multi-stage simulated annealing approach in which different interconnect planning methods are used in different ranges of temperatures to reduce running time. A temperature adjustment scheme is designed to give smooth transitions between different stages of simulated annealing.

The floorplanning problems typically have relatively small number of blocks but have a large number of nets. Since existing floorplanning algorithms use simulated annealing which needs to examine a large number of floorplans, this has made interconnect-centric floorplanning computationally very expensive. We have presented some approaches that can dramatically improve the run time of problems with large number of nets and at the same time improve solution quality.

Without careful power supply planning in layout design, the design of chips

will suffer from mostly signal integrity problems including IR-drop,  $\Delta I$  noise, and IC reliability. We have also proposed a method for simultaneous power supply planning and noise avoidance in floorplan design. We show that the noise avoidance in power supply planning problem can be formulated as a constrained maximum flow problem.

I/O placement has been a concern in modern IC design. Due to flip-chip and multi-chip module technologies, I/O can be placed throughout the whole chip without long wires from the periphery of the chip. However, because of I/O placement constraints and I/O buffer site building cost, the decision of positions for placing I/O buffers has become critical. We have presented an approach to reducing the number of I/O buffer sites and to deciding their positions in an existing standard cell placement. We formulate it as a minimum cost network flow problem and it can be further refined by force-directed approach.

Here we discuss some perspectives in solving big netlist floorplanning problem. The advantage of bounded-degree hypergraph-to-graph transformation approach is to optimally decompose multi-terminal nets into two-terminal nets due to the constraint of pin-limit in logic blocks. However, this must associate with fairly good pin assignment in order to perform good quality global routing. Besides, if we can compensate some area to provide space, we can redistribute block pins and generate slightly different version in decomposition of multi-terminal nets. We could also consider some other meaningful objectives to decompose multi-terminal nets. One is that we can find a way to get a decomposition so that we can overlap most two-terminal nets. This is due to the consideration of relieving the congestion later

in routing stage. Another one is to consider some pin placement constraint, such as position constraint and signal direction, etc. Another possible objective is planning for timing optimization. If some nets are timing-critical, the decomposition should pay attention to them, trying to meet the timing requirements.

We also have some remarks for Lagrangian relaxation global router. We use Lagrangian relaxation as a heuristic to solve routing problem. The reason for Lagrangian relaxation being a heuristic is that routing itself is NP-hard problem, so we can not use this technique to optimally solve routing problem unless some assumptions are made. In fact, the global router we use to solve Lagrangian relaxation subproblem is heuristically simple geometry router. In this way, we obtain one upper bound for max overflow by sequentially routing the nets. In [32], global routing is formulated as an integer program. By relaxing the constraints into objective function, good lower bound for max overflow is generated. Therefore, using Lagrangian relaxation in planning wires is a fairly good choice.

The scaling of process technology and device and interconnect size has led timing optimization techniques for VLSI circuits to become increasingly critical. Floorplanning is a very important step in designing timing-efficient circuits. If a floorplan is bad in terms of timing, even a very powerful router cannot help. One of the critical problems in physical design, interconnect-centric floorplanning, as we discussed in preceding chapters, is attracting more attention as technology moves deeper into DSM and the timing convergence problem is getting more difficult. Since some performance measures, like timing, power, congestion and routability needed to be evaluated repeatedly, could be very time consuming in estima-

tion, we need efficient and effective algorithms to take care of them. On the other hand, it has been shown that buffer insertion techniques are successfully applied in interconnect-centric timing optimization. The buffer insertion ability has been incorporated into floorplan evaluation: buffer block planning methodology, improved buffer insertion technique, and early resource planning. We can follow this paradigm to achieve timing closure in floorplanning stage.

In designs nowadays, the importance of lowering power dissipation is given comparable weight to reduced interconnect delay and area minimization. Two factors mainly contribute to this trend. First, the growth of portable applications which demand high-speed computation and complex functionality with low power consumption. Without low-power design techniques, current and future portable devices will suffer from either a very short battery life or a very heavy battery pack. Second, power dissipation in high-end ICs rises with increasing integration density and circuit speed. The resultant heat will increase the packaging and cooling costs and shorten the life of ICs as well. The low power design methodology in floorplanning should be applied.

Identically instantiated modules (or repeated modules) implement the same mapping of logic functions from input signals to output signals and are identical instantiations of the same module prototype. For repeated modules, replicated design costs such as compilation, debugging, and simulation are reduced. Repeated modules have become popular in VLSI circuits for smaller problem size, lower cost, and more feasible hierarchical design. They are making up a bigger and bigger portion of new communications and signal processing chips. For ex-

ample, POWER4 microprocessor from IBM used total 4341 macros/blocks while only 1015 macros/blocks are unique. Therefore, new design planning strategies are clearly needed to optimize circuit performance in the presence of repeated modules. We plan to study the optimization problem in floorplan design with repeated modules, trying to minimize the area and interconnect planning costs.

## Bibliography

- [1] “International Technology Roadmap for Semiconductor”. 1997.
- [2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. “*Network Flows - Theory, Algorithms, and Applications*”. Prentice-Hall, 1993.
- [3] G. Bai, S. Bobba, and I.N. Hajj. “Simulation and Optimization of the Power Distribution Network in VLSI Circuits”. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 481–486, 2000.
- [4] H. B. Bakoglu. “*Circuits, Interconnections, and Packaging for VLSI*”. Addison Wesley, 1990.
- [5] M.W. Beattie and L.T. Pileggi. “Inductance 101: Modeling and Extraction”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 323–328, 2001.
- [6] A. Berman and R.J. Plemmons. “*Nonnegative Matrices in the Mathematical Sciences*”. Academic Press, NY, 1996.
- [7] S. Bobba, T. Thorp, K. Aingaran, and D. Liu. “IC Power Distribution Challenges”. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 643–650, 2001.

- [8] P.H. Buffet, J. Natonio, R.A. Proctor, Y.H. Sun, and G. Yasar. “Methodology for I/O cell Placement and Checking in ASIC Designs Using Area-Array Power Grid”. In *IEEE Custom Integrated Circuits Conference*, pages 125–128, 2000.
- [9] A.E. Caldwell, A.B. Kahng, S. Mantik, and I.L. Markov. “Implications of Area-Array I/O for Row-Based Placement Methodology”. In *IEEE Symposium on IC/Package Design Integration*, pages 93–98, 1998.
- [10] L. Cao and J.P. Krusius. “A New Power Distribution Strategy for Area Array Bonded ICs and Packages of Future Deep Sub-Micron ULSI”. In *Electronic Components and Technology Conference*, pages 1138–1145, 1997.
- [11] A. Chandrakasan, W.J. Bowhill, and F. Fox, editors. “*Design of High-Performance Microprocessor Circuits*”. IEEE Press, 2001.
- [12] Yun-Chih Chang, Yao-Wen Chang, Guang-Ming Wu, and Shu-Wei Wu. “B\*-Trees: A New Representation for Non-Slicing Floorplans”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 458–463, 2000.
- [13] K.-Y. Chao and D.F. Wong. “Floorplan Design with Low Power Considerations”. In *Low Power VLSI Design and Technology*, pages 83–100. World Scientific, 1996.
- [14] H.-M. Chen, L.-D. Huang, I.-M. Liu, M. Lai, and D.F. Wong. “Floorplanning with Power Supply Noise Avoidance”. In *Proceedings IEEE Asia and South Pacific Design Automation Conference*, pages 427–430, 2003.

- [15] H.-M. Chen, D.F. Wong, W.-K. Mak, and H.H. Yang. “Faster and More Accurate Wiring Evaluation in Interconnect-Centric Floorplanning”. In *Great Lakes Symposium on VLSI*, pages 62–67, 2001.
- [16] H.-M. Chen, D.F. Wong, H. Zhou, F.Y. Young, H.H. Yang, and N. Sherwani. “Integrated Floorplanning and Interconnect Planning”. In *Layout Optimization in VLSI Designs*, pages 1–18. Kluwer Academic Publishers, 2001.
- [17] H.-M. Chen, H. Zhou, F.Y. Young, D.F. Wong, H.H. Yang, and N. Sherwani. “Integrated Floorplanning and Interconnect Planning”. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 354–357, 1999.
- [18] H.H. Chen and D.D. Ling. “Power Supply Noise Analysis Methodology for Deep-Submicron VLSI Chip Design”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 638–643, 1997.
- [19] J. Cong. “An Interconnect-Centric Design Flow for Nanometer Technologies”. *Proceedings of the IEEE*, 89(4):505–528, April 2001.
- [20] J. Cong, L. He, K. Y. Khoo, C. K. Koh, and Z. Pan. “Interconnect Design for Deep Submicron ICs”. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 478–485, 1997.
- [21] J. Cong, T. Kong, and D.Z. Pan. “Buffer Block Planning for Interconnect-Driven Floorplanning”. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 358–363, 1999.

- [22] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. “*Introduction to Algorithms*”. The MIT Press, 1990.
- [23] John F. Croix. “The Need for Accurate Power Models for Deep Submicron IP reuse”. *Electronic Systems*, 1999.
- [24] A. Deutsch, P. Coteus, G. Kopcsay, H. Smith, C. Surovic, B. Krauter, D. Edelstein, and P. Restle. “On-chip Wiring Design Challenges for Gigahertz Operation”. *Proceedings of the IEEE*, 89(4):529–555, April 2001.
- [25] H. Eisenmann and F.M. Johannes. “Generic Global Placement and Floorplanning”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 269–274, 1998.
- [26] R. Farbarik, X. Liu, M. Rossman, P. Parakh, T. Basso, and R. Brown. “CAD Tools for Area-Distributed I/O Pad Packaging”. In *IEEE Multi-Chip Module Conference*, pages 125–129, 1997.
- [27] L.R. Ford, Jr., and D.R. Fulkerson. “*Flows in Networks*”. Princeton University Press, 1962.
- [28] K. Gala, David Blaauw, V. Zolotov, P.M. Vaidya, and A. Joshi. “Inductance Model and Analysis Methodology for High-Speed On-Chip Interconnect”. *IEEE Transactions on Very Large Scale Integration Systems*, 10(6):730–745, December 2002.

- [29] Pei-Ning Guo, Chung-Kuan Cheng, and Takeshi Yoshimura. “An O-Tree Representation of Non-Slicing Floorplan and its Applications”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 268–273, 1999.
- [30] J.N. Kozhaya, S.R. Nassif, and F.N. Najm. “I/O Buffer Placement Methodology for ASICs”. In *IEEE International Conference on Electronics, Circuits and Systems*, pages 245–248, 2001.
- [31] P. Larsson. “Power Supply Noise in Future IC’s: A Crystal Ball Reading”. In *IEEE Custom Integrated Circuits Conference*, pages 467–474, 1999.
- [32] T. Lengauer and M. Luger. “Provably Good Global Routing of Integrated Circuits”. *SIAM J. OPTIM.*, 11(1):1–30, 2000.
- [33] L. Liang, J.D. Wilson, N. Brathwaite, L.E. Mosley, and D. Love. “High-Performance VLSI Through Package-Level Interconnects”. In *Proceedings of the 39th Electron. Components Conf.*, pages 518–523, 1989.
- [34] S. Lin and N. Chang. “Challenges in Power-Ground Integrity”. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 651–654, 2001.
- [35] I-Min Liu, H.-M. Chen, T.-L. Chou, A. Aziz, and D.F. Wong. “Integrated Power Supply Planning and Floorplanning”. In *Proceedings IEEE Asia and South Pacific Design Automation Conference*, 2001.

- [36] R.J. Lomax, R.B. Brown, M. Nanua, and T.D. Strong. “Area I/O Flip-Chip Packaging to Minimize Interconnect Length”. In *IEEE Multi-Chip Module Conference*, pages 2–7, 1997.
- [37] D.G. Luenberger. “*Linear and Nonlinear Programming*”. Addison-Wesley, 2nd edition, 1989.
- [38] D. MacMillen, M. Butts, R. Camposano, D. Hill, and T.W. Williams. “An Industrial View of Electronic Design Automation”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(12):1428–1448, December 2000.
- [39] P.H. Madden. “Reporting of Standard Cell Placement Results”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(2):240–247, February 2002.
- [40] V. Maheshwari, J. Darnauer, J. Ramirez, and W.W.-M. Dai. “Design of FPGAs with Area I/O for Field Programmable MCM”. In *Proceedings ACM Symposium on Field Programmable Gate Arrays*, pages 17–23, 1995.
- [41] Wai-Kei Mak and D.F. Wong. “Board-Level Multi-Terminal Net Routing for FPGA-based Logic Emulation”. *ACM Transaction on Design Automation of Electronic Systems*, 2:151–167, 1997.
- [42] Joel Mcgrath. “Chip/Package co-design: The bridge between chips and systems”. In *Advanced Packaging*, June 2001.

- [43] H. Murata, K. Fujiiyoshi, S. Nakatake, and Y. Kajitani. “Rectangle-Packing-Based Module Placement”. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 472–479, 1995.
- [44] Farid N. Najm. “Low-Power Design Methodology: Power Estimation and Optimization”. In *Proceedings of the 40th Midwest Symposium on Circuits and Systems*, pages 1124–1129, 1997.
- [45] S. Nakatake, K. Fujiiyoshi, H. Murata, and Y. Kajitani. “Module Placement on BSG-Structure and IC Layout Applications”. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 484–491, 1996.
- [46] S.R. Nassif and J.N. Kozhaya. “Fast Power Grid Simulation”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 156–161, 2000.
- [47] Ralph H.J.M. Otten. “Global Wires Harmful?”. In *Proceedings International Symposium on Physical Design*, pages 104–108, 1998.
- [48] Ralph H.J.M. Otten and Robert K. Brayton. “Planning For Performance”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 122–127, 1998.
- [49] R.H.J.M. Otten. “Automatic Floorplan Design”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 261–267, 1982.
- [50] R.H.J.M. Otten, R. Camposano, and P.R. Groeneveld. “Design Automation for Deepsubmicron: present and future”. In *Proceedings Design, Automation and Test in Europe*, pages 650–657, 2002.

- [51] L.T. Pillage, R.A. Rohrer, and C. Visweswariah. “*Electronic and System Simulation Methods*”. McGraw-Hill, 1995.
- [52] B. Preas and W. VanCleemput. “Placement Algorithms for Arbitrary Shaped Blocks”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 474–480, 1979.
- [53] N.R. Quinn. “The Placement Problem as Viewed from the Physics of Classical Mechanics”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 173–178, 1975.
- [54] Resve Saleh, Michael Benoit, and Pete McCrorie. “Power Distribution Planning”. Technical report, Simplex Solutions Inc., 1997.
- [55] P.A. Sandborn, M.S. Abadir, and C.F. Murphy. “The Tradeoff Between Peripheral and Area Array Bonding of Components in Multichip Modules”. *IEEE Transactions on Components, Packaging, and Manufacturing Technology - Part A*, 17(2):249–256, June 1994.
- [56] P. Sarkar, V. Sundararaman, and C.-K. Koh. “Routability-Driven Repeater Block Planning for Interconnect-Centric Floorplanning”. In *Proceedings International Symposium on Physical Design*, pages 186–191, 2000.
- [57] Naveed Sherwani. “*Algorithms for VLSI Physical Design Automation*”. Kluwer Academic Publishers, 3rd edition, 1999.
- [58] D.C. Smith. “A Method for Troubleshooting Noise Internal to an IC”. In *IEEE EMC Symposium Proceedings*, pages 223–225, 1997.

- [59] J. Song, H.K. Choo, and W. Zhuang. “A New Model for General Connectivity and its Application to Placement”. In *Proc. 6th Great Lake Symposium on VLSI*, pages 60–63, 1996.
- [60] T. Tamanouchi, K. Tamakashi, and T. Kambe. “Hybrid Floorplanning Based on Partial Clustering and Module Restructuring”. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 478–483, 1996.
- [61] C. Tan, D. Bouldin, and P. Dehkordi. “Design Implementation of Intrinsic Area Array ICs”. In *Proceedings 17th Conference on Advanced Research in VLSI*, pages 82–93, 1997.
- [62] X.-D. Tan, C.-J.R. Shi, D. Lungeanu, J.-C. Lee, and L.-P. Yuan. “Reliability-Constrained Area Optimization of VLSI Power/Ground Networks Via Sequence of Linear Programmings”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 78–83, 1999.
- [63] K.T. Tang and E.G. Friedman. “On-Chip Delta-I Noise in the Power Distribution Networks of High Speed CMOS Integrated Circuits”. In *IEEE ASIC/SOC Conference*, pages 53–57, 2000.
- [64] D. Wang and E.S. Kuh. “A New General Connectivity Model and Its Applications to Timing-Driven Steiner Tree Routing”. In *Proceedings International Symposium on Circuits and Systems*, pages 71–74, 1998.
- [65] D.F. Wong and C.L. Liu. “A New Algorithm for Floorplan Design”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 101–107, 1986.

- [66] H. Xiang, X. Tang, and D.F. Wong. “An Algorithm for Simultaneous Pin Assignment and Routing”. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 232–238, 2001.
- [67] G. Yasar, C. Chiu, R.A. Proctor, and J.P. Libous. “I/O Cell Placement and Electrical Checking Methodology for ASICs with Peripheral I/Os”. In *IEEE International Symposium on Quality Electronic Design*, pages 71–75, 2001.
- [68] J.-S. Yim, S.-O. Bae, and C.-M. Kyung. “A Floorplan-based Planning Methodology for Power and Clock Distribution in ASICs”. In *Proceedings ACM/IEEE Design Automation Conference*, pages 766–771, 1999.
- [69] M. Zhao, R.V. Panda, S.S. Sapatnekar, and D. Blaauw. “Hierarchical Analysis of Power Distribution Networks”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(2):159–168, February 2002.
- [70] S. Zhao, K. Roy, and C.-K. Koh. “Decoupling Capacitance Allocation for Power Supply Noise Suppression”. In *Proceedings International Symposium on Physical Design*, pages 66–71, 2001.
- [71] P.S. Zuchowski, J.H. Panner, D.W. Stout, J.M. Adams, F. Chan, P.E. Dunn, A.D. Huber, and J.J. Oler. “I/O Impedance Matching Algorithm for High-Performance ASICs”. In *IEEE International ASIC Conference and Exhibit*, pages 270–273, 1997.

## Vita

Hung-Ming Chen was born in Taipei, Taiwan, Republic of China on January 24, 1971. He received the degree of Bachelor of Engineering in Computer Science and Information Engineering from National Chiao Tung University, Hsinchu in 1993. After two years of military service, he joined Institute of Information Science, Academia Sinica, Taiwan as a research assistant from August 1995 to July 1996. He then entered the University of Texas at Austin and received the degree of Master of Science in Computer Sciences in 1998. He was a Teaching Assistant between 1997 and 2000 in UTCS and has been an Assistant Instructor since summer 2000. He was employed as a summer intern in Intel Corporation, Hillsboro, Oregon in 1999. He has published one book chapter and five conference papers in the area of VLSI design automation during his study towards the doctoral degree.

Permanent address: 12F-2, No. 32, Lane 31, Mintzu Rd.  
Danshuei Jen, Taipei County  
Taiwan 251, R.O.C.

This dissertation was typeset with  $\text{\LaTeX}^\dagger$  by the author.

---

<sup>†</sup> $\text{\LaTeX}$  is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's  $\text{\TeX}$  Program.