

Abstract of thesis entitled

Lithography-friendly routing via forbidden pitch avoidance

submitted by

Shi Shi Chang

for the degree of Doctor of Philosophy

at the University of Hong Kong

in July 2004

Resolution enhancement techniques (RETs) have been widely adopted in subwavelength optical lithography for reliable manufacture of ever smaller device features. However, their adoption simultaneously introduces certain restrictions on layouts, and therefore challenges physical design. When using subresolution assist features (SRAFs) in conjunction with off-axis illumination (OAI) to enhance image quality of dense patterns, intermediate pitches have a lower lithographic process window. Although allowed by current design rules, these pitches should ideally be avoided in layouts due to their low manufacturability. Exclusion of pitches with a low process window (also called forbidden pitches) presents unique challenges for routing. In this thesis, lithography-friendly routing via forbidden pitch avoidance is investigated.

A two-stage lithography-friendly routing method is proposed. It accomplishes the routing task in the first stage by traditional routing techniques. In the second

stage, pitch adjustment is iteratively executed to move the wire segments away from forbidden pitches. Pitch adjustment concentrates on avoiding forbidden pitches in one affected area at a time, and is divided into two steps: pitch optimization and spacing. In the pitch optimization step, the process window of wire segments forming the forbidden pitch and their neighbors that may be affected during the pitch adjustment process is formulated into a quadratic objective function. Based on the linear geometric constraints in the layout, new optimal locations for wire segments in the affected area are obtained by solving the constrained quadratic optimization problem. In the spacing step, each wire segment is adjusted to its new optimal location by a spacing technique that keeps the original wire connections and design rules intact during the layout modification process.

A lithography-friendly router (FnRouter) is developed, which incorporates the two-stage lithography-friendly routing method. The router was applied to three test chips, and successfully avoided more than 80% of forbidden pitches. The image quality of patterns in most affected areas was also substantially improved.

(341 words)

Lithography-friendly routing via forbidden pitch avoidance

by

Shi Shi Chang

Department of Electrical and Electronic Engineering

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
at the University of Hong Kong

July 2004

Declarations

I declare that the thesis represents my own work, except where due acknowledgement is made, and that it has not been previously included in a thesis, dissertation or report submitted to this university or to any other institution for a degree, diploma or other qualification.

Signed *Shi Shi Cheny*

Acknowledgements

I would like to express my sincere thanks to all those who generously gave me a great help on my project. Without them, it is impossible for me to finish this thesis. First, I would like to thank my supervisor, Dr. Alfred, K.K Wong. It is he who brought me to this interesting research world. His patient direction is the greatest help in my four years of Ph.D. study. I would also like to thank another supervisor, Professor T.S. Ng. It is he who continues the supervision on my research after Dr. Wong left the University of Hong Kong for his new job in USA. His kindness, profound knowledge and experience help me finish my research. Then I would like to thank all the people in our VLSI group, Leo Li, Jun Wang, Xin Yang, Mak Yick Hong Giuseppe, and Pun Chiu Ho. They gave me a lot of help on the research. Next many thanks should be given to Professor E.S. Yang, Dr. G.X. Shen, Dr. Q.Y. Ma, and all the colleagues of MRI Engineering Center in the University of Hong Kong for providing me a good environment to do my research during my first two academy years. Last I would like to thank Stan Chow in AmmoCore Technology, Inc., Robert C. Pack and Hardy K.S. Leung in MAGMA Inc. for their specialistic advices from the industry, which greatly facilitate my research.

Table of Contents

Chapter 1	Introduction	1
1.1	Resolution of Optical Lithography.....	3
1.2	Goal of This Research.....	6
1.3	Organization.....	7
Chapter 2	Resolution Enhancement Techniques	10
2.1	Resolution Enhancement Techniques.....	10
2.1.1.	Optical Proximity Correction.....	11
2.1.2.	Off-axis Illumination & Subresolution Assist Features.....	13
2.2	Design for Manufacturability.....	17
2.3	Summary.....	20
Chapter 3	Lithography-friendly Routing	21
3.1	Review of Routing Techniques.....	22
3.2	Lithography-friendly Routing.....	25
3.2.1.	OPC Constraints.....	25
3.2.2.	SRAFs & OAI Constraints.....	26
3.3	FnRouter.....	29
3.3.1.	Flow Chart of FnRouter.....	30
3.3.2.	Data Preparation.....	31
3.4	Summary.....	34

Chapter 4	Traditional Routing Techniques	35
4.1	Corner Stitching Data Structure	37
4.2	Database of FnRouter	39
4.3	H-V Tile-expansion Routing Algorithm	42
4.3.1.	Handing Design Rules	43
4.3.2.	H-V Tile-expansion Process	44
4.3.3.	Cost Functions	46
4.4	Rip-up and Reroute	48
4.5	OPC-friendly Routing	49
4.6	Summary	53
Chapter 5	Pitch Optimization	54
5.1	Definition of Forbidden Pitches	54
5.2	Formulation of Pitch Optimization	57
5.2.1.	Quadratic Curve Fitting	58
5.2.2.	Objective Function	59
5.2.2.1.	Layout Example	59
5.2.2.2.	Graph Expression	62
5.2.2.3.	Matrices of Objective Function	63
5.2.3.	Layout Constraints	68
5.2.3.1.	Design Rule Constraints	68
5.2.3.1.1.	Boundary Constraints	69
5.2.3.1.2.	Via Constraints	71
5.2.3.1.3.	Permanent Fixed Pattern Constraints	73

5.2.3.2. Inherited Constraints.....	74
5.2.3.3. Compulsory Constraints.....	75
5.2.3.4. Optional Constraints.....	78
5.2.4. General Case.....	79
5.3 Pitch Optimization in FnRouter.....	82
5.4 Summary.....	83
Chapter 6 Spacing Technique	84
6.1 Plow Operation.....	84
6.2 Spacing Technique.....	86
6.2.1. Edge Definition.....	86
6.2.2. Plowing Rules.....	87
6.2.3. Self-instructions.....	89
6.2.4. Jog Control.....	91
6.3 Implementation in FnRouter.....	92
6.3.1. Plowing Edges.....	93
6.3.2. Layout Example.....	95
6.4 Summary.....	97
Chapter 7 Other Factors and Application	98
7.1. Measures of Image improvement.....	98
7.2. Wire Length Variation.....	99
7.3. Constrained Optimization.....	101
7.4. Algorithm Speed.....	102

7.5.	Application.....	103
7.5.1.	Test Chips.....	104
7.5.2.	Image Improvement for Forbidden Pitch.....	106
7.5.3.	Image Improvement for Affected Area.....	108
7.5.4.	Wire Length Variation.....	110
7.5.5.	Constrained Optimization.....	110
7.6	Summary.....	113
Chapter 8 Assistant Techniques		114
8.1	Rip-up and Reroute.....	114
8.2	Wire Width Modification.....	117
8.3	Summary.....	120
Chapter 9 Discussions & Conclusion		121
9.1	Conclusion.....	122
References		124

List of Figures

Fig. 1.1: Shift to subwavelength optical lithography since the 0.35 μm process generation.....	1
Fig. 1.2: Basic components of optical lithography.....	4
Fig. 1.3: Rayleigh factor k_1 in the past, present, and future.....	6
Fig. 2.1: Examples of OPC-friendly pattern and ambiguous patterns.....	12
Fig. 2.2: Off-axis illumination.....	13
Fig. 2.3: Comparison between off-axis illumination and conventional illumination.....	14
Fig. 2.4: Use of SRAFs across the pitch.....	16
Fig. 3.1: Space constraints.....	27
Fig. 3.2: The flow chart of FnRouter.....	30
Fig. 3.3: Procedure to obtain the design files.....	32
Fig. 3.4: Design file vs. DEF and LEF file.....	33
Fig. 4.1: Corner stitching data structure and its C++ code definition.....	38
Fig. 4.2: Examples of the corner stitching layout.....	38
Fig. 4.3: Main database, routing planes, and spacing planes in FnRouter.....	40
Fig. 4.4: (a) Layout in the main database before routing. (b) Layout in the main database after routing. (c)-(f) Layout in the routing plane during the routing process with the help of the record plane.....	41
Fig. 4.5: Handling design rules in blockage planes.....	43
Fig. 4.6: H-V tile expansion process.....	45

Fig. 4.7: Routing results under different <i>HCOST</i> and <i>VCOST</i> assignment.....	52
Fig. 5.1: (a) Experimental process window across the pitch with the use of SRAFs. (b) Two forbidden ranges are defined for the experimental process window. (c) Two quadratic curves fit the experimental process window.....	56
Fig. 5.2: (a) Layout of the example. (b) The patterns in metal 2 (two forbidden pitches exist). (c) Vertical patterns and their geometric size.....	60
Fig. 5.3: Graph expression for example.....	63
Fig. 5.4: Two vertical wire segments and its graph expression.....	64
Fig. 5.5: Submatrices and objective matrices.....	66
Fig. 5.6: An example of the boundary constraints.....	70
Fig. 5.7: (a) The wire segment in metal 2 connects to the pin by via A. (b) The wire segment in metal 2 connects to the wire segment in metal 3 by via B.....	72
Fig. 5.8: Permanent fixed pattern constraints.....	73
Fig. 5.9: An example of worse case.....	76
Fig. 5.10: (a) Jog example. (b) Straighten the jog.....	79
Fig. 5.11: The flow chart of the pitch optimization in FnRouter.....	82
Fig. 6.1: Plowing operation.....	85
Fig. 6.2: Six kinds of edges in spacing technique.....	87
Fig. 6.3: Achieve the maximum plowing distance by spacing technique.....	88
Fig. 6.4: Pitch adjustment by self-instructions. (a) Moving wire segment A. (b) Moving wire segment B. (c) Layout after the pitch adjustment.....	90
Fig. 6.5: Jog control. (a) Plowing edge a. (b) Plowing by jog insertion.	

(c) Plowing without jogs.....	91
Fig. 6.6: The flow chart of spacing technique in the spacing step.....	93
Fig. 6.7: Edge queue in the plowing edges phase.....	94
Fig. 6.8: An example of forbidden pitch avoidance by spacing technique.....	97
Fig. 7.1: Four types of wire length variation.....	100
Fig.7.2: Circuit schematics of three test chips.....	104
Fig. 7.3: Process window improvement for each forbidden pitch in (a) 8-bit ripple-carry adder, (b) 12-bit ripple-carry adder with one clock delay, (c) 8-bit 3-tap FIR filter.....	107
Fig. 7.4: Process window improvement for each affected area in (a) 8-bit ripple-carry adder, (b) 12-bit ripple-carry adder with one clock delay, (c) 8-bit 3-tap FIR filter.....	109
Fig. 7.5: Wire length variation in (a) 8-bit ripple-carry adder, (b) 12-bit ripple-carry adder with one clock delay, (c) 8-bit 3-tap FIR filter.....	111
Fig. 7. 6: Wire segment number of each affected area in (a) 8-bit ripple-carry adder, (b) 12-bit ripple-carry adder with one clock delay, (b) 8-bit 3-tap FIR filter.....	112
Fig. 8.1: Solid tile in the blockage planes.....	115
Fig. 8.2: An example by ripping-up and rerouting net to avoid forbidden pitch.....	116
Fig. 8.3: Forbidden pitch avoidance by wire width modification.....	117

List of Tables

Table 7.1: Routing result of the three test chips (based on TSMC 180nm library)..106

Table 7.2: Image improvement for forbidden pitches.....108

Chapter 1 Introduction

Since the emergence of the first integrated circuit (IC) in 1958, the integration density of ICs has gone through an astounding revolution. With the continual reduction in feature size of optical lithography, smaller transistors can be fabricated in chips so that Moore's Law, i.e. the exponential growth in the number of transistors per IC, has been maintained and still holds true today [Intel04]. Fig. 1.1 illustrates the decrease of feature size during the past 20 years. In 1980, feature size of optical lithography was $3.0 \mu\text{m}$, while it decreased to $0.11 \mu\text{m}$ in 2003 and is expected to reach 90 nm in 2004 and 65 nm in 2007 according to the International Technology Roadmap for Semiconductors [ITRS03].

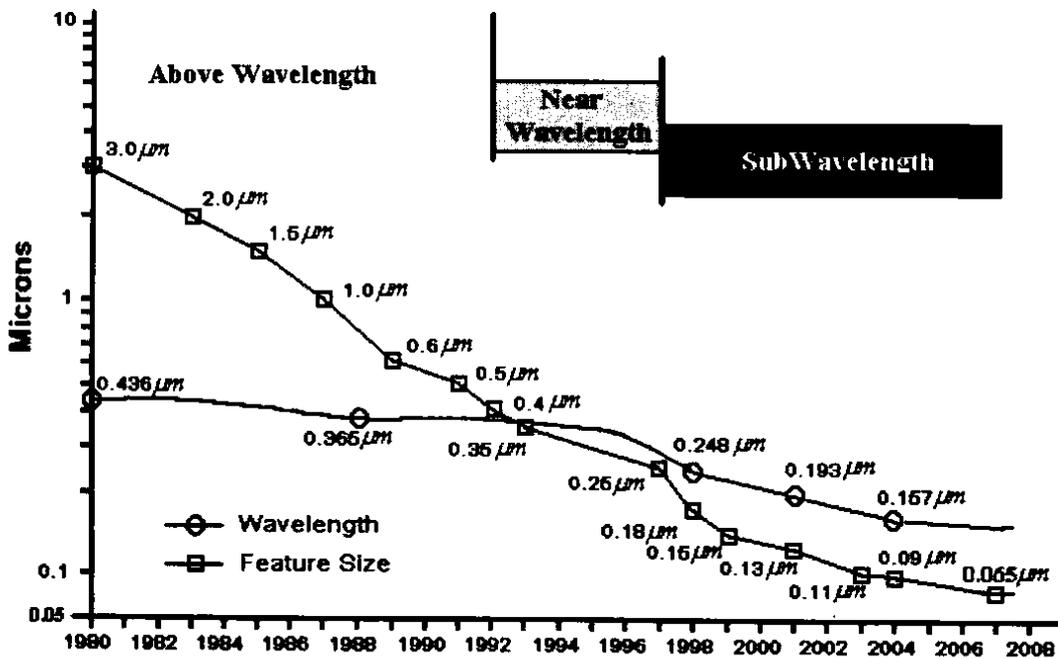


Fig. 1.1: Shift to subwavelength optical lithography since the $0.35 \mu\text{m}$ process generation [Kahng99].

However, the wavelength of light used in optical lithography does not keep pace with the rapid descent of feature size because of three limitations. First, few light sources can be chosen to expose wafers in optical lithography. Secondly, the cost of optical lithography technology dramatically increases after the wavelength of light used is below 193 *nm* since the operation environment is strictly required to be oxygen and water free. Thirdly, new photoresist and optical materials for making lens need to be developed with the change of wavelength used in optical lithography, which are not simple and straightforward [Wong03]. As illustrated in Fig. 1.1, since the mid 1990s when the feature size reduced to 0.35 μm , optical lithography has entered the subwavelength era, i.e. the minimum feature or critical dimension (CD) that needs to be resolved on the wafer is smaller than the wavelength of the light used to print them.

Based on the knowledge of optics, as images approach the dimensions of the wavelength of the light source, undesirable effects start to happen, which include image distortion, shortening, and blur due to physical limitations of the optical system. With the addition of interference from other processes in manufacturing such as etching and polishing, image quality control that directly affects the yield of chips becomes a difficult task in subwavelength optical lithography. In order to enhance the image quality of the photomask so that the printed patterns on the wafer are as close to the desired shapes in the layout as possible, resolution enhancement techniques (RETs) [Wong01] are widely adopted. However, as optical lithography is being pushed closer to its fundamental resolution limit, it is becoming increasingly difficult to implement RET without RET-enabling layout restrictions [Liebma03].

The requirement of lithography-friendly layout in IC manufacturing challenges the IC design especially in the physical design. New geometric constraints and RET-compliant design rules should be introduced in placement, routing, layout verification, and designing library cells. Moreover, closer communication between design team and manufacturing team should be developed since restrictions in layout design changes when applying different combination of RETs in subwavelength optical lithography. Therefore, the requirement of design for manufacturability (DFM) becomes more crucial in today's IC industry.

1.1 Resolution of Optical Lithography

When the IC design is completed, the final chip layout is taped-out (typically in GDSII file) and transferred to the IC foundry for manufacturing. Firstly, the layout data are converted into geometric patterns on the photomask by mask-writing tools. Then in optical lithography, optical instruments project light through the photomask onto the surface of silicon wafers that are coated with light-sensitive photoresist material [Lavin02]. After chemical development processes remove the resist material, geometric patterns are transcribed from the photomask to silicon wafers and ready to act as the template for ion implantation and etching. Therefore, function correctness and yield of chips are directly affected by how accurate the layout information is transcribed to the wafer, which is greatly determined by the resolution of optical lithography in addition to influences from the photoresist and chemical processes.

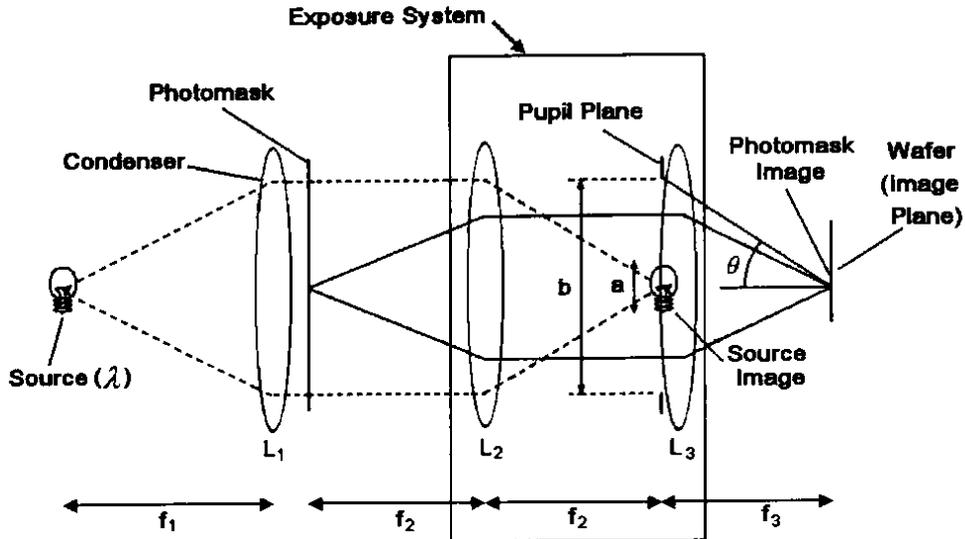


Fig. 1.2: Basic components of optical lithography [Wong01].

Optical lithography comprises four basic components: light source, photomask, exposure system, and wafer as illustrated in Fig. 1.2. Lights with wavelength λ illuminate the photomask, and then patterns on the photomask are replicated by the exposure system onto the wafer coated with photoresist. In this figure, θ is the semi-aperture angle that the pupil subtends at the wafer. Numerical aperture (NA) of the exposure system is defined as

$$NA = n \sin \theta, \quad (1.1)$$

where n is the refractive index of the image space medium. The CD of optical lithography, i.e. the minimum size that it can be resolved, can be defined as a function of three parameters [Wong01] as

$$CD = k_1 \times \frac{\lambda}{NA}, \quad (1.2)$$

where k_1 is the Rayleigh factor used as a measure of lithographic challenge. The smaller k_1 , the more difficult it is to resolve a certain dimension with a given optical lithography technology.

From (1.2), it is obvious that the resolution of optical lithography can be enhanced by reducing λ or k_1 , or increasing NA. However, λ can only be reduced to a certain extent due to the three limitations presented before. As to $\sin \theta$ in (1.1), its physical upper limit equals to 1. But increasing θ has adverse impacts on depth of focus (DOF), which is the maximum focus variation that the exposure process can tolerate. Therefore, the convenient approach for resolution enhancement is reducing k_1 . Although there is no theoretic low limit of k_1 , it is restricted by process control, material robustness, and the cost of optical lithography. In [Wong03], the practical low limit of k_1 is estimated to be 0.2.

Fig. 1.3 presents the Rayleigh factor k_1 achieved in optical lithography during the past decade and its estimation in the next few years [Liebma03a]. Four distinct k_1 zones can be identified in this figure. For the technology with feature size larger than 350 nm, $k_1 > 0.7$ can be easily achieved with conventional optical lithography. While for 250 nm and 180 nm technologies, k_1 can approach 0.5 by introducing optimization effort and RETs. As feature size reduces to 130 nm and 90 nm, k_1 is below 0.5 and the application of RETs is widespread. For the next two generations technologies — 65 nm and 45 nm, optical lithography is estimated to

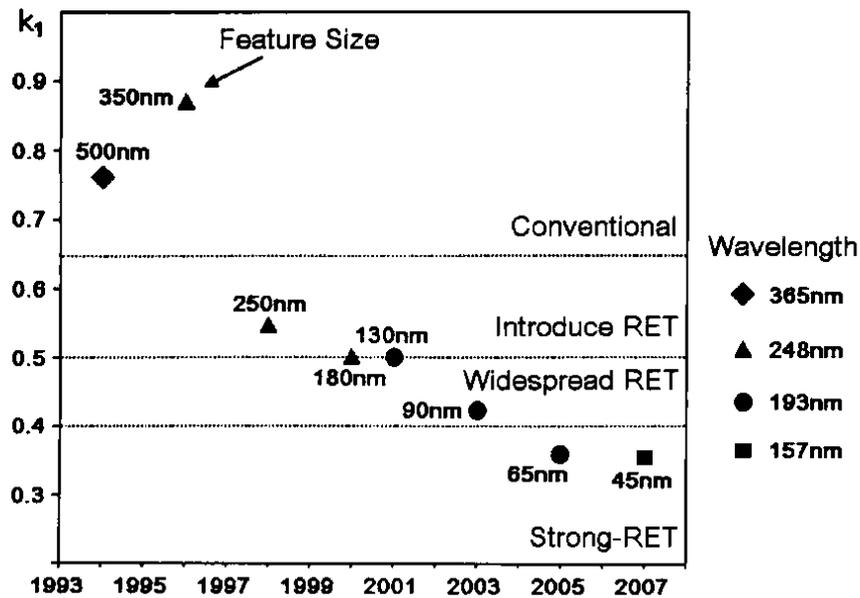


Fig. 1.3: Rayleigh factor k_1 in the past, present, and future [Liebma03a].

achieve $k_1 < 0.4$ with the help of strong-RETs, which belong to the second generation RETs and are more complex than those of the first generation.

1.2 Goal of This Research

The implementation of RETs in optical lithography contributes to the decrease of k_1 so that smaller features could be fabricated on the wafer. However, it simultaneously impacts and challenges the layout design. As RETs become inevitable in subwavelength optical lithography and more difficult to be implemented as k_1 is being pushed even closer to its low limit, designing lithography-friendly layout so as to facilitate the use of RETs is more urgent than before. Approaches such as introducing new geometric constraints and RET-compliant design rules in

physical design have been proposed by researchers. However, most of them concentrate on the layout design at library cell level. There are few approaches that consider the lithography-friendly requirement at higher level, such as the chip level.

Since physical locations of metal wires and their shapes in the whole chip are determined by the router, the goal of this research is to develop a new routing technique to achieve lithography-friendly layout at the chip level. Though some routing techniques to facilitate the use of specific RETs such as phase-shifting mask (PSM) have been presented [Liebma03][McCull04], this research is believed to be the first investigation on the lithography-friendly routing via forbidden pitch avoidance, which facilitates the use of another two RETs — subresolution assist features (SRAFs) in conjunction with off-axis illumination (OAI). A two-stage lithography-friendly routing method is proposed, which inherits the traditional routing techniques and integrates some RET-compliant restrictions.

1.3 Organization

Chapter 2 briefly reviews the principles of RETs including optical proximity correction (OPC), OAI, and SRAFs. Their challenges and restrictions on layout design, especially the forbidden pitch problem, are explored and suggestions to facilitate the use of RETs are presented.

Chapter 3 discusses the lithography-friendly routing after reviewing the traditional routing techniques. A two-stage lithography-friendly routing method is proposed, which avoids the forbidden pitches by adjusting the space between wire segments after traditional routing. The structure of FnRouter, which was developed in the research process to implement the two-stage lithography-friendly routing method, is introduced.

Chapter 4 briefly presents the traditional routing techniques implemented in FnRouter, which corresponds to the traditional routing stage, i.e. the first stage of the two-stage lithography-friendly routing method.

Chapter 5 introduces the pitch optimization step in the iterative pitch adjustment stage, i.e. the second stage of the two-stage lithography-friendly routing method. Lithography-friendly routing via forbidden pitch avoidance is formulated as a constrained quadratic optimization problem in this step.

Chapter 6 introduces the spacing technique implemented in the iterative pitch adjustment stage, which adjusts the space between wire segments in the layout while keeping the original wire connectivity and design rules intact.

Chapter 7 discusses other factors involved in the two-stage lithography-friendly routing method and presents the results when applying the method to three test chips.

Chapter 8 appends two assistant techniques to avoid the forbidden pitches that could not be solved by the two-stage lithography-friendly routing method.

Chapter 9 presents some discussions about the lithography-friendly routing via forbidden pitch avoidance and then draws the conclusion of the research.

Chapter 2 Resolution Enhancement Techniques

As Rayleigh factor k_1 falls below 0.75, RETs are widely implemented in optical lithography to allow even smaller features to be reliably manufactured. RETs are based on the prediction of the physical phenomena of the optical systems used for optical lithography, especially diffraction and phase interference. By predicting these effects and compensating for them, image contrast and pattern fidelity can be dramatically improved, and the minimum feature and pitch that can be resolved are significantly extended [Schellen01]. However, the adoption of RETs also places certain restrictions on layouts.

2.1 Resolution Enhancement Techniques

The main RETs include phase-shifting mask (PSM) [Levenson82] [Liu98], optical proximity correction (OPC), off-axis illumination (OAI), and subresolution assist features (SRAFs). In the following, all RETs except PSM are briefly reviewed and their restrictions on layouts are explored. Also, the rising challenges in RET-compliant layout design are discussed and some suggestions to facilitate the use of these RETs are provided.

2.1.1 Optical Proximity Correction

In low- k_1 optical lithography, the pupil of the exposure system acts as a low-pass filter [Wong01], which blocks the high-frequency components of the mask spectrum. Due to the loss of high-frequency components, images on the wafer plane are distorted from the original layout and thus results in proximity effects. Typical proximity effects, i.e. image distortions, include line shortening, corner rounding, and iso-dense bias (or called linewidth variation).

To compensate the image distortions, optical proximity correction (OPC) [Kahng99a] is applied in mask-making process, which predistorts mask patterns so that the printed features are as close to the desired shape as possible. OPC can be of several forms, which includes serifs, hammerheads, notches, and SRAFs. There are two approaches to implement OPC: rule-based [Otto96] and model-based [Cobb97]. Rule-based OPC corrects the layout according to a predetermined rule table. Model-based OPC is more complex, which modifies the layout iteratively according to the simulation using mathematical models of the fabrication process.

OPC is transparent to circuit design since the pattern predistortion process is not performed until the layout is taped out for mask fabrication. However, when optical lithography is driven to subwavelength era, the appeal of designing OPC-friendly layout for manufacturing is much louder than before due to the rising difficulties in applying OPC to irregular, ambiguous patterns in the layout. Take patterns from [Grobm01] as examples, the pattern in Fig. 2.1(a) is OPC-friendly

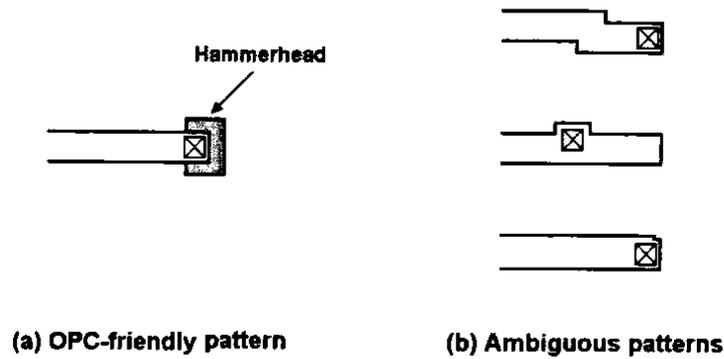


Fig. 2.1: Examples of OPC-friendly pattern and ambiguous patterns [Grobm01].

since it is easily recognized by rule-based OPC with little parameters to add a hammerhead at the line-end to ensure proper contact coverage. By comparison, rule-based OPC is not efficient for ambiguous patterns in Fig. 2.1(b) since more parameters are required to specify the patterns. In addition, rule generation and table look-up time increase exponentially with the rising number of parameters. Though model-based OPC may solve such problems, it is approximately two to three times slower than rule-based OPC [Liebma01] and extra polygons that are not needed may be introduced [Grobm01]. The rocketing expense on mask fabrication is another reason to appeal for OPC-friendly layout. Because the data volume of the corrected layout is explosively enlarged, the time of mask fabrication is increased and hence the cost is raised. If the layout is OPC-friendly, then OPC run time and output file size can be dramatically reduced so as to decrease the expense on mask fabrication.

OPC-friendly layout required by manufacturing makes OPC no longer independent of circuit design and is suggested to combine with physical layout verification [Schellen99]. To facilitate OPC operations, OPC-compliant restrictions

should be introduced in the layout design. Most of them require patterns in the layout to be well defined and easily recognizable.

2.1.2 Off-axis Illumination & Subresolution Assist Features

Off-axis illumination (OAI) [Reynolds86][Kamon91] enhances the resolution of optical lithography by using a shaped illuminator (annular, quadrupole, or dipole) to direct light only at certain angles onto the mask. The principle of OAI is demonstrated in Fig. 2.2. Due to the illumination has an oblique angle ϕ with the optical axis, the phase of the light front varies along the conventional mask. At a particular mask aperture pitch, which is determined by

$$pitch = \frac{\lambda}{2 \times \sin \phi}, \quad (2.1)$$

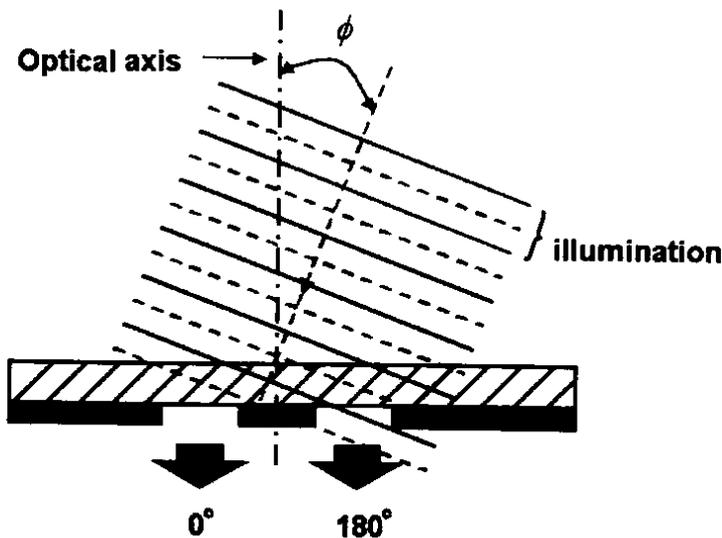


Fig. 2.2: Off-axis illumination [Rieger01].

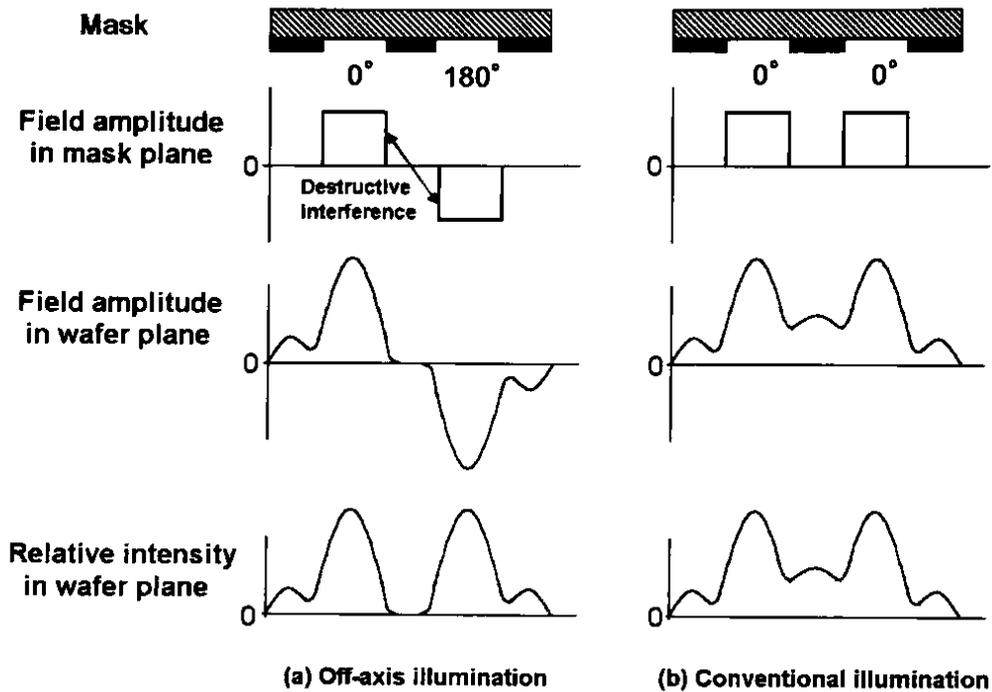


Fig. 2.3: Comparison between off-axis illumination and conventional illumination [Wong01].

the light phase appears to have shifted 180 degrees. As illustrated in Fig. 2.3, if lights project to the two clear regions on the mask with 180 degree shifted phase under OAI, lights diffracted into the dark region between two clear regions interfere destructively and yield a region with low field amplitude in the wafer plane. Since the light intensity is square of the field amplitude, image contrast of the printed feature is improved, which leads to better resolution and depth of focus (DOF) for optical lithography.

OAI can enhance the resolution by nearly a factor of two for the features with dense pitches optimized according to (2.1). However, it has less resolution improvement for features with large pitches, such as isolated lines. For other features

with intermediate pitches, it even deteriorates their image qualities [Rieger01]. Besides, linewidth variation due to iso-dense bias can be amplified with OAI due to the different diffraction performance of isolated line and dense lines [Schellen01a]. Therefore, subresolution assist features (SRAFs) [Prouty84] [Garofalo93] are always used in conjunction with OAI to overcome the limitation of pitch-specific resolution enhancement in OAI.

SRAFs, also called “scattering bars”, are small lines below the resolution of optical lithography so that they are not printed onto the wafer but can affect the light diffraction. The use of SRAFs is also one of OPC techniques. By adding SRAFs to both sides of the isolated or semi-isolated lines to create a dense environment, linewidth variation can be controlled in tolerance and the image process window (i.e. the measure of how well pattern fidelity can be controlled over a range of exposure dose and defocus [Liebma03a]) can be enhanced. However, due to the finite design grid of CAD tools and mask manufacturability constraints, SRAFs can only be used in discrete steps in size and spacing across the pitch [Liebma01]. Many researches on SRAFs design strategy under OAI have been reported, such as [Mansfield00] [Socha00]. Their experimental results show great similarity, which are illustrated in Fig. 2.4 with a discontinuous process window curve as a function of pitch.

Main features in Fig. 2.4 are expressed with rectangles in black color, while SRAFs are rectangles in white color. It is obvious that the process window for unassisted main features drops sharply with the increase of pitch. However, the process window can be recovered when the pitch between main features is large

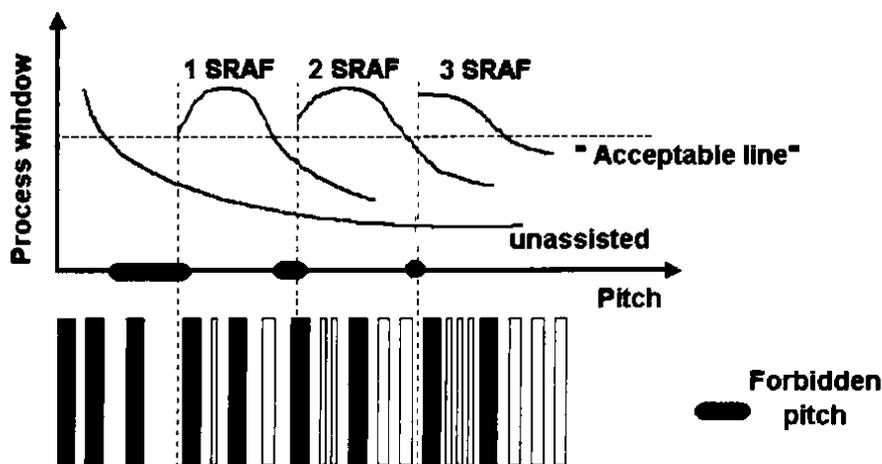


Fig. 2.4: Use of SRAFs across the pitch [Liebma01].

enough to insert a SRAF. As the pitch continues to grow, the optimum size of SRAF changes to maintain its benefit for high process window. When the process window cannot be held by single SRAF and the pitch is large enough, the second SRAF is inserted so as to restore the process window. Such rising and falling of process window continues with the increase of pitch. In the current technology, up to four SRAFs can be filled between main features.

It is clear that the use of SRAFs across various pitches is not straightforward. The number, size, and placement of SRAFs vary as a function of pitch [Wong01]. Furthermore, the process window across the pitch is not uniform. Some pitches have low process window, though the number, size, and placement of SRAFs have been optimized according to the design strategy achieved from the experimental result for specific optical lithography. These pitches always fall in the transition regions, where the process window keeps falling but there is no room to place a SRAF or an additional SRAF. From the optical lithography standpoint, these pitches should be

forbidden in the layout due to their low manufacturability. If only the process window above the “acceptable line” is accepted in manufacturing, then the range of “forbidden pitches” [Socha00] can be defined as in Fig. 2.4.

In order to facilitate the use of SRAFs in conjunction with OAI, forbidden pitches should be avoided in the layout due to their low manufacture robustness. Therefore, new space restrictions between features to avoid forbidden pitches should be introduced in the layout design, which greatly challenges the traditional physical design, especially the routing process that determines the physical locations for metal wires in the layout.

2.2 Design for Manufacturability

The yield of IC is important for IC companies since it is directly related to their profits. Several years ago, predicting IC yield was fairly straightforward: if each structure can be manufactured, then the entire chip can be manufactured since the layout drawn on the screen was just the pattern printed on the silicon wafer. However, when the optical lithography enters the subwavelength era and RETs are widely adopted, patterns printed on the silicon wafer are no longer the same as images on the mask and the layout. Factors that affect chip manufacturing increase dramatically, structures designed correctly may not be manufactured successfully on the wafer. Thus the work to achieve high yield becomes increasingly difficult and the time-to-market is greatly delayed due to the long time on debugging. Recently, design for

manufacturability (DFM) becomes an area of active research. Manufacturability should be considered in addition to timing, area, power, and signal integrity at the chip design level so as to maximize the IC yield and reliability [Wilcox98] [Camposa03].

The DFM research on optical lithography mainly concerns the use of RETs. In subwavelength optical lithography, the implementation of RETs can greatly improve the resolution and the image quality of masks so that chips' manufacturability can be enhanced. However, lithography-friendly restrictions need to be introduced in the layout design to facilitate the use of RETs. For example, patterns for OPC should be well defined and easily recognized; forbidden pitches should be avoided in the layout when using SRAFs in conjunction with OAI. Recent DFM researches suggest that these lithography-friendly restrictions can be realized by introducing RET-compliant design rules in layout design [Lavin02] [Grobm01a] or incorporate statistical information about the process into EDA tools [Camposa03], which can lead to great benefits on manufacturing:

1. **The yield and reliability of chips are greatly enhanced.** If the layout is lithography-friendly, RETs can improve the resolution and image quality during the layout post-processing, i.e. after the layout is taped out. Thus manufacturability can be greatly improved, which results in higher yield and reliability for chips.

2. **Expense on IC fabrication is saved.** The mask data volume is always explosively enlarged after OPC, as much as ten times [Camposa03], which increases both mask writing time and mask cost. However, it can be decreased by carefully designing the layout with the introduction of OPC-compliant design rules, such as making dense pitches for patterns to decrease the number of SRAFs. Therefore the expense on mask fabrication can be greatly saved.

3. **Time-to-market is shortened.** Lithography-friendly layout can speed up the implementation of RETs. For example, if patterns in the layout are OPC-friendly, i.e. well defined and easily recognized, rule-based OPC could be directly implemented to correct the image deformations, which is faster than model-based OPC. In addition to the time saved from mask fabrication due to the reduction of data volume, the time-to-market can be shortened.

Recently, lots of DFM researches on optical lithography have been published, such as using phase conflict graph to analyze the phase assignment in PSM application [Kahng00], designing lithography-friendly standard cells [Wang03], and layout optimization for optical lithography [Liebma03b]. All of them show novel ideas in introducing lithography-friendly restrictions in the design to facilitate the use of RETs in subwavelength optical lithography so as to enhance ICs' manufacturability. But most of these researches concentrate at the poly-silicon level, i.e. the library cell level. It is true that library cells contribute significantly to the manufacturability of ICs since they are used repeatedly throughout the design and

they blend logic design information (netlist), design trade-offs of performance, power, area, and yield (cell architecture), and manufacturing process requirements (design rules) [Prolific03]. However, lithography-friendly consideration should be moved up to higher level, such as the metal wiring level or chip level with the increase of IC integration [Liebma03]. Therefore, lithography-friendly routing is an obvious target since it determines the physical locations of metal wires and their shapes in the whole chip layout.

2.3 Summary

In this chapter, the principles of different RETs including OPC, OAI, and SRAFs have been reviewed. Their impacts and restrictions to layout design have also been presented. To facilitate the use of RETs, DFM appeals for lithography-friendly layout that is OPC-friendly and has no forbidden pitch, which leads to new challenges to physical design, especially in the routing process with the increase of IC integration.

Chapter 3 Lithography-friendly Routing

In the general IC design flow, after the front-end engineers finish the register-transfer-level (RTL) design, the abstract design files (usually in VHDL or Verilog-HDL format) are delivered to back-end engineers to finish the physical design that transfers the abstract netlist to the real layout. Routing, abbreviated as P&R with the placement in the industry, is an important stage in the physical design, whose task is to connect the placed library cells according to the netlist. During the past 40 years, routing techniques applied in EDA tools have been well developed and have contributed to the IC industry greatly. It will continue to develop to meet new challenges from the IC industry.

Since optical lithography enters the subwavelength era, RETs have been widely implemented in the industry to enhance the chip manufacturability. However, lithography-friendly restrictions should be introduced in the layout to facilitate the use of RETs. Recent researches on lithography-friendly layout design concentrate on poly-silicon level due to their relative ease. However, researches should be moved up to metal wiring level with the increase of IC integration. Because the physical locations of the metal wires and their shapes in the layout are mainly determined in the routing process, developing lithography-friendly routing technique becomes a new research topic. Until now, there are few formal publications on how to realize the lithography-friendly routing.

3.1 Review of Routing Techniques

Routing research started earlier than the emergence of IC since it was firstly implemented for wiring printed circuit boards (PCBs). In the following, the review of routing techniques is restricted only to their application to automatic design of IC layout. Generally, the routing problem can be defined as follows:

Given a set of placed cells with pins and a netlist that interconnects different pins on the cells, route all the nets with the minimum number of routing layers and satisfy some constraints. These constraints can be wire length minimization, crosstalk minimization, via minimization, and the manufacturability constraints, especially design rules, which guarantee the high yield and reliability [Sait99].

The routing problem has been proved that it is an NP-hard problem, i.e. it is unlikely that an algorithm exists for solving the routing problem in polynomial time [Sherwani99].

Early routing techniques were developed on the grid-based maze routing algorithm that was first described by Moore to connect the nets on the PCB [Moore59]. The famous Lee's algorithm [Lee61] that finds a path between the source and the target in a planar rectangular grid with obstacles, and its various improved version including [Hadlock75][Soukup78] all belong to maze routing algorithms.

Maze routing algorithm guarantees the shortest connection between two terminals if it exists. However, the limitation is that it requires more memory to record the grid information and is more time consuming for a large and dense layout. To overcome the drawback of maze routing algorithm, line-search algorithms that use line segments instead of grid cells in the search, were developed independently by Mikami and Tabuchi [Mikami68], and Hightower [High69]. As opposed to maze routing algorithm that proceeds in a breadth-first manner, line-search algorithms perform a depth-first search. Line-search algorithms are more memory and time efficient than maze routing algorithms, however, the drawback is that they cannot guarantee finding the shortest path, and sometimes cannot find a path even if such a path exists. In order to find an acceptable path and also ensure memory and time efficiency, many routers that combine maze routing with line-search algorithms were developed, such as the line-expansion routing algorithm to route nets on a grid-based plane [Heyns80], and the tile-expansion algorithm based on corner stitching data structure [Margari87].

With the increase of metal layers in IC manufacturing, routing algorithms were developed from single layer routing to multiple layers routing. Then gridless-based algorithms [Chen86][Chen86a] became more popular than the grid-based algorithms because it allowed arbitrary location of terminals, nets, and vias on the routing layers. Furthermore, it easily realized different wire width in different layers, which is required by design rules. In addition, to facilitate the routing in multiple layers, reserved layer model [Sherwani99] was widely adopted by most of routers, which restricts wires with certain directions to particular layers.

With the explosive increase of transistors integrated on the VLSI chip, millions of nets have to be routed in the layout, which makes the complexity of routing problem dramatically increase. Therefore, divide-and-conquer approach is applied in the routing techniques. The whole routing stage is divided into two phases: global routing and detail routing. Global routing partitions the routing area into a set of non-intersecting rectangular routing regions and generates a loose route for each net. Detail routing then finds the actual geometrical layout of each net within the assigned routing regions. Channel and switchbox are two types of routing regions partitioned by global routing. Correspondingly, detail routing is classified as channel routing and switchbox routing. Because channel region is not limited in one layer, and terminal locations are different with the styles of standard cells used, there are numerous channel routing algorithms, such as the Let-Edge algorithm (LEA) for two-layer channel routing [Hashi71], the VICTOR algorithm [Strunk93] for three-layer channel routing, specific for standard cells in BTM (Boundary Terminal Model) style [Sherwani95]. There are also lots of switchbox routing algorithms, such as the Greedy algorithm [Luk85], and the simulated evolution algorithm [Lin89].

In addition to the above general routing algorithms, specialized algorithms for routing clock, power, and ground are also developed. As the chip frequency moves into the GHz range, the clock skew budget, i.e. the maximum difference in the arrival time of a clock at two different components, becomes smaller and smaller. In order to ensure the clock arrive at all functional units at the same time, clock routing algorithms with clock buffering mechanisms were developed to minimize the clock skew and delay [Jackson90][Kahng93]. As to the power and ground routing, special

attention should be paid to line widths due to the larger amount of current required in power and ground nets. Different algorithms, such as the plane sweep algorithm [Xiong86] and top-down design approach [Lurs87], were developed to carefully size the wires to allow proper current flow.

3.2 Lithography-friendly Routing

Routing techniques have been well developed during the past 40 years and are still being improved to meet new challenges that come with the development of IC technology. As optical lithography enters the subwavelength era, DFM appeals for the lithography-friendly layout so as to facilitate the use of RETs. Because the layout of metal layers is determined during the routing process, routers should consider the lithography-friendly requirement when wiring nets. Currently, lithography-friendly routing is still a new research topic and only few results have been reported. In the following, constraints of RETs to the routing techniques are discussed, and approaches to realize lithograph-friendly routing in this thesis are proposed.

3.2.1 OPC Constraints

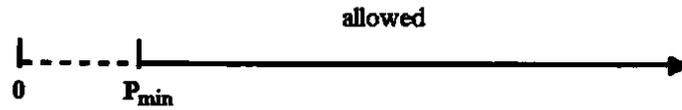
The layout that facilitates the use of OPC should be well defined and easily recognized. It is better to use rectangular line-ends, forbid circle or oval shapes,

avoid small zig-zag and jogs when designing layouts [Prolific03]. Since the locations of metal wires and their shapes in the layout are determined by the router, OPC-friendly layout design should introduce these OPC constraints in the routing process.

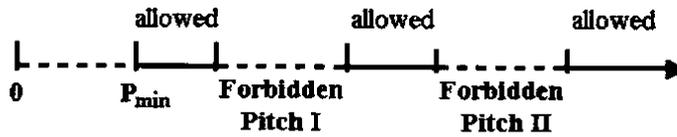
The lithography-friendly routing research in this thesis adopts two approaches to make the layout OPC-friendly. The first one is to introduce new OPC-compliant design rules that only allow Manhattan patterns (horizontal and vertical edges only) in the layout so as to forbid circle or oval shapes. The other one is to emphasize the OPC constraints in the cost function to control the routing style. Cost function gives high values to layout patterns with small zig-zag and jogs so that the final routed net with the least cost has the least number of zig-zag and jogs. Because these two approaches could be realized by improving traditional routing techniques, OPC-friendly routing is not the main topic of the lithography-friendly routing research in this thesis.

3.2.2 SRAFs & OAI Constraints

As presented in the last chapter, the use of SRAFs in conjunction with OAI results in the forbidden pitch problem. To avoid forbidden pitches, new space constraints between wire segments should be introduced when developing lithography-friendly routing techniques. It is a great challenge since the space constraint considered in traditional routing algorithms is only the minimum pitch required by the design rules. Any space beyond the minimum pitch is allowed during



(a) Space constraints in traditional routing algorithms



(b) Space constraints due to forbidden pitch avoidance

Fig. 3.1: Space constraints.

the routing process as illustrated in Fig. 3.1(a). However, when using SRAFs in conjunction with OAI, spaces in the ranges of forbidden pitches, which are beyond the minimum pitch, should be also forbidden due to their low manufacture robustness. Therefore, the allowed space range is no longer a contiguous one larger than the minimum pitch, but becomes several individual segments separated by forbidden pitches, as illustrated in Fig. 3.1(b).

The research in this thesis is the first investigation on lithography-friendly routing via forbidden pitch avoidance. Generally, introducing new space constraints as in Fig. 3.1(b) in the routing algorithms could prevent wire pitches falling into the forbidden ranges. However, routing problem is an NP-hard problem. The introduction of more constraints in the routing algorithm always greatly increases the algorithm complexity and tends to yield results that have larger routing area, more failed path in routing, etc. Furthermore, current OPC techniques including the use of

SRAFs is implemented judiciously to reduce the mask fabrication cost, i.e. the corrections are mainly applied for critical patterns that greatly influence the circuit performance rather than using the same degree of predistortion for all shapes [Wong03]. Therefore, it is not necessary to introduce new space constraints for each net during the routing process.

Another feasible approach to avoid forbidden pitches is post-processing the layout routed by traditional routing algorithms. The locations of wire segments forming forbidden pitches after traditional routing are adjusted so as to make their spaces out of forbidden pitches. The advantage of this approach is that traditional routing techniques can be directly implemented in the routing process and operations to avoid forbidden pitches are only concentrated in the post-routing process. Though this approach is easier to implement, there are two basic problems that need to be investigated. The first one is how to avoid producing new forbidden pitches when adjusting the wire segments from the old forbidden pitches. The second one is how to keep the design rules intact and the original wire connections during the layout modification.

In this thesis, a two-stage lithography-friendly routing method is proposed to avoid forbidden pitches in the layout design. In the first stage, traditional routing algorithm is implemented to finish the routing task. Then in the second stage, forbidden pitches are avoided sequentially by adjusting the wire segments forming them. To avoid producing new forbidden pitches or deterioration of the original image quality of the layout, the process to avoid forbidden pitches by layout

modification is formulated as a constrained quadratic optimization problem. To keep the design rules intact and the original wire connections during the layout modification, spacing technique developed from the plowing algorithm [Scott84] provided by the Magic layout editor [Ouster84] is implemented to deal with the pitch adjustment.

3.3 FnRouter

In the course of this research, a lithography-friendly router called FnRouter is developed, which adopts the proposed two-stage lithography-friendly routing method to deal with the forbidden pitch problem arisen from the use of SRAFs in combination with OAI in optical lithography. Moreover, FnRouter also considers the OPC constraints in the traditional routing process.

FnRouter is a gridless router based on corner stitching data structure [Ouster84a], which supports routing of a maximum of five metals. FnRouter is developed with C++ code under the Unix environment on Sun Ultra 10 workstations. It has more than 30,000 lines codes and 70% or so codes are related to traditional routing techniques. A graphical user interface (GUI) by Tcl/Tk toolkit [Ouster94] has also been developed to show the routing process and layout.

3.3.1 Flow Chart of FnRouter

Fig. 3.2 illustrates the flow chart of FnRouter, which can be divided into four phases: data preparation, traditional routing, iterative pitch adjustment, and assistant modification. In Fig. 3.2, data between two successive phases are expressed with rectangles, and programs in different phases are expressed with ovals. In the data preparation, a Perl [Lerner02] script program called splitter analyzes the input design files from other CAD tools, and outputs files including library, placement, design rules, and netlist information for routing. The next two phases, traditional routing and

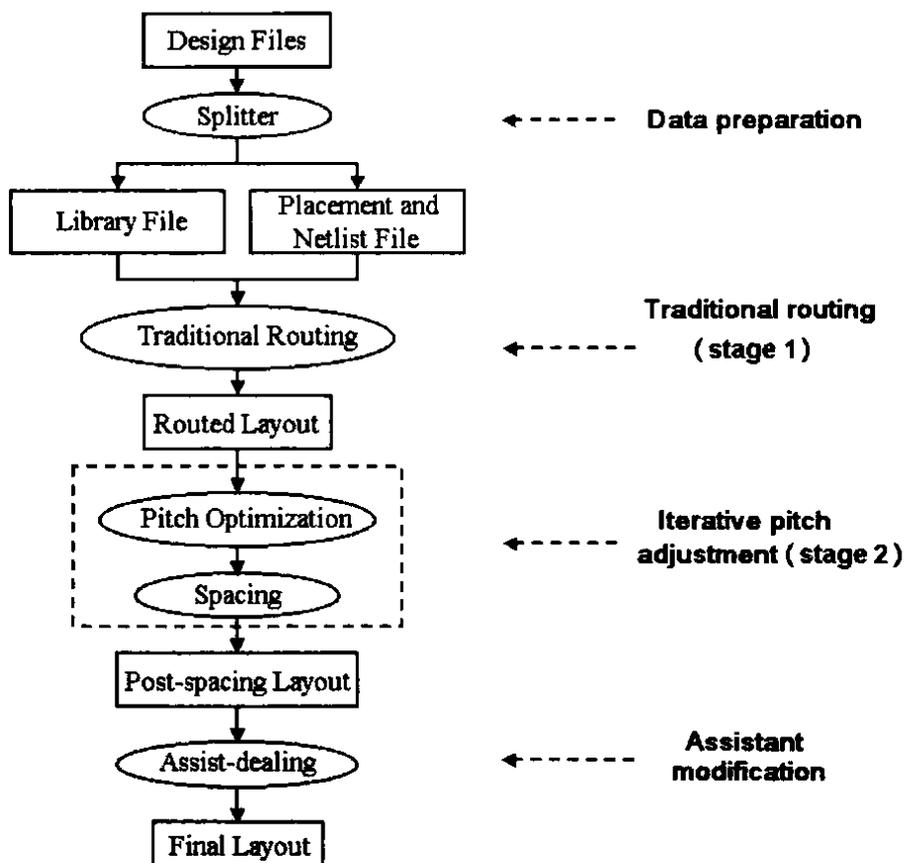


Fig. 3.2: The flow chart of FnRouter.

iterative pitch adjustment, correspond to the two stages of the adopted two-stage lithography-friendly routing method. H-V tile-expansion routing algorithm is adopted in the traditional routing. In the iterative pitch adjustment, forbidden pitches are avoided iteratively by pitch optimization and spacing. Pitch optimization calculates the new optimal locations for each wire segments that may be affected during the layout modification. In the spacing process, these wire segments are adjusted to their new optimal locations by spacing technique that keeps the design rules intact and original wire connections. The final assistant modification phase uses assistant techniques, such as ripping-up and rerouting nets or modifying the wire width, to deal with the remaining forbidden pitches that cannot be solved by the two-stage lithography-friendly routing method.

3.3.2 Data Preparation

In IC design flow, the input to the general routing problem includes at least netlist, placement, and design rules information. As to FnRouter, this information is supplied by the design files produced by other commercial EDA tools available in this research. Fig. 3.3 illustrates the procedure to obtain the design files for FnRouter. Firstly, the behavior and function of the chip are described in the module compiler file (*.mcl), which can be recognized by Synopsys Module Compiler™ [Synopsys04], a high-level datapath synthesis tool developed by Synopsys. Based on the TSMC standard cell library, RTL design expressed in the hardware description file (*.vhd or *.vrl) could be achieved after synthesis. Then two approaches can be chosen to

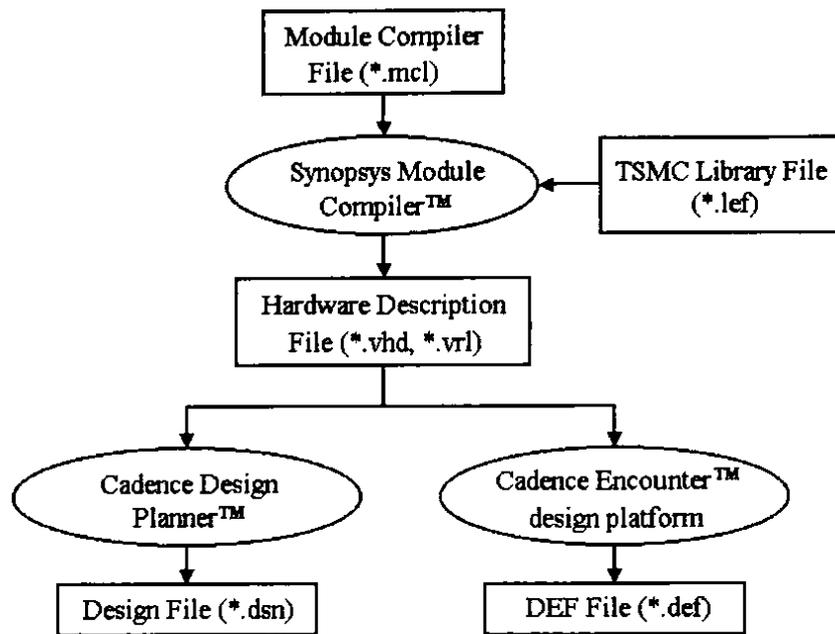


Fig. 3.3: Procedure to obtain the design files.

produce the design files for FnRouter. The first is to input the RTL design to Cadence Design Planner™, which is an IC floorplan and placement tool developed by Cadence. The second is to input the RTL design to Cadence Encounter™ design platform [Cadence04], which can finish the floorplan and placement by the tools integrated in the design platform. The outputs of the two approaches are design file (*.dsn) and DEF file (*.def) respectively. Either of them could be used as the input design file for FnRouter.

Design file (*.dsn) from Cadence Design Planner™ is a text file written in the Cadence IC CraftsMan™ design language [Cadence99]. This file contains all the data including placement, netlist, design rules and applied standard cells to define a

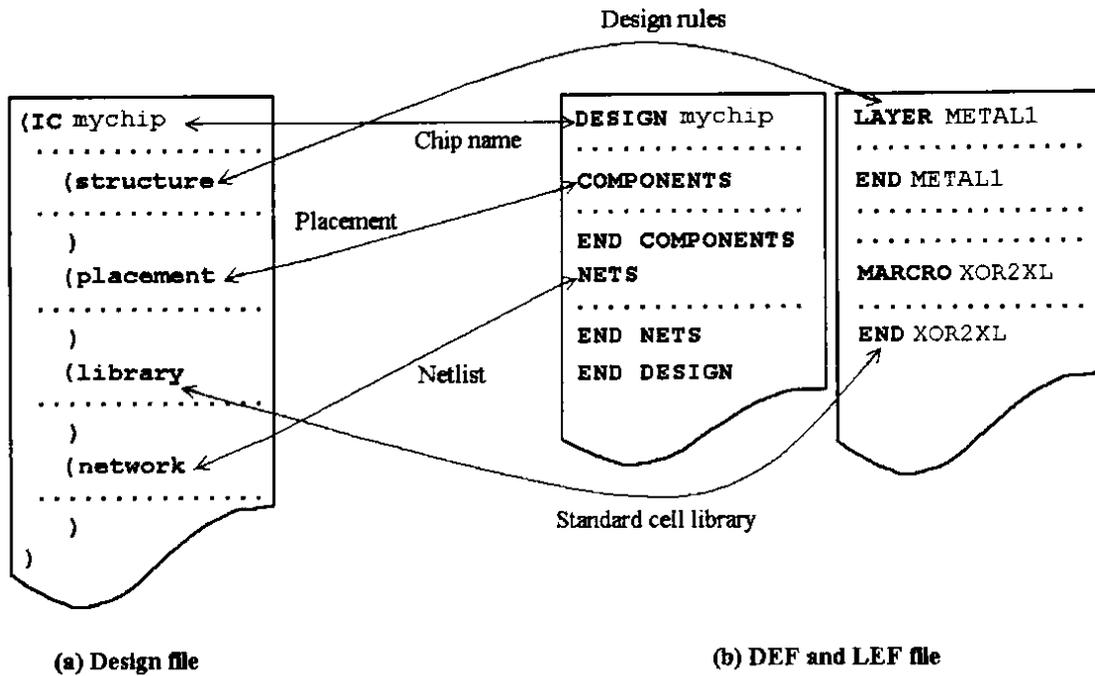


Fig. 3.4: Design file vs. DEF and LEF file.

design for IC CraftsMan™, which is a router developed by Cadence. The structure and syntax of design file are briefly illustrated in Fig. 3.4(a).

The design exchange format (DEF) file from Cadence Encounter™ design platform is also a text file, which only includes placement, netlist information. Therefore, it should be used with the library exchange format (LEF) file, which represents technology and standard cell library information. In Fig. 3.4(b), the structure and syntax of DEF and LEF are briefly illustrated. DEF and LEF were developed by Cadence in the early 1990's to provide a common input and output format for area-based place-and-route tools. Today, DEF and LEF have been the public and open source standards.

Design files, DEF and LEF files are all text files in standard formats. In the data preparation of FnRouter, a Perl script program called splitter is used to extract the necessary information by matching keywords in these files, and then outputs the placement, netlist, design rules, and standard cell library information with two simpler files called design.lib and design.in, which are used as the library file, placement and netlist file for the traditional routing in FnRouter.

3.4 Summary

In this chapter, constraints of different RETs to the routing techniques have been discussed after reviewing the traditional routing techniques. Methods to realize lithography-friendly routing, especially the two-stage lithography-friendly routing method to avoid forbidden pitches due to the use of SRAFs in conjunction with OAI in optical lithography, have been proposed. The flow chart and data preparation of FnRouter, which is developed in the research process to implement the proposed two-stage lithography-friendly routing method, have been briefly introduced.

Chapter 4 Traditional Routing Techniques

Traditional routing is the first stage of the two-stage lithography-friendly routing method. Though traditional routing algorithms have been well developed in the past 40 years, developing a router program based on these traditional algorithms is still a very time consuming work. This is because suitable data structures need to be constructed to apply these algorithms and a database should be developed to manage all the geometric shapes in the layout during the routing process. To speed up the lithography-friendly research, traditional routing in FnRouter is simplified based on the following three considerations:

1. **Forbidden pitch avoidance is a local operation.** As explained before, the two-stage lithography-friendly routing method iteratively adjusts the wire segments away from the forbidden pitch after the traditional routing. During each iteration, only part of the layout around the corresponding forbidden pitch is modified.
2. **Only the pitches of wire segments are adjusted.** In the iterative pitch adjustment stage, wire segments are moved away from the forbidden pitches. During this procedure, only the pitches of wire segments are changed, but their widths are kept intact.

3. Wire length is little changed after each forbidden pitch avoidance.

Though the lengths of adjusted wires are always changed after each pitch modification, results of test chips show that these wire length variations can be neglected when they are compared with their entire wire lengths.

Based on the first consideration, small chips are enough for lithography-friendly routing research since there is no difference in the forbidden pitches between large chips and small chips. Thus, global routing algorithms are not adopted in FnRouter, and detail routing algorithms are directly applied to small chips in the traditional routing. Based on the second and third considerations, iterative pitch adjustment has little influence on the timing and current of the nets. Thus, clock, power, and ground routing algorithms are not implemented in FnRouter, all the nets are connected by common routing algorithm with the assumption that their timing and current constraints are all satisfied.

Actually, any existing routing algorithm can be used in the traditional routing stage of the two-stage lithography-friendly routing method. FnRouter adopts H-V tile-expansion routing algorithm, which is a detail routing algorithm based on the gridless, reserved layer routing model, to finish the routing task. In the following, H-V tile-expansion routing algorithm and other traditional routing techniques adopted by FnRouter are discussed.

4.1 Corner Stitching Data Structure

In physical design, layout is generally represented by a collection of tiles, which are non-overlapping rectangular sections of the layout within a single layer [Sherwani99]. Wire segments and vias in the layout are referred to as solid tiles, while the empty areas in the layout are referred to as space tiles. In the lithography-friendly routing research, operations of traditional routing and pitch adjustment are both based on the tiles in the layout. Therefore, the choice of a suitable data structure to represent the tiles is an important task, which is directly related to the algorithm complexity, memory requirement, and running speed. In the past several decades, some data structures have been developed and implemented in the VLSI physical design and simple layout systems, such as linked list in the Caesar system [Ouster81], bins-based data structure [Bentley79], and neighbor pointers in Cabbage system [Hsueh79]. However, these data structures are inefficient to implement in the complex layout systems because they only represent solid tiles and cannot support adequate fast layout operations.

FnRouter adopts the corner stitching data structure [Ouster84a] that was developed for the Magic layout system [Ouster84]. As illustrated in Fig. 4.1, each tile represented by this data structure contains its lower and left edges, but not its upper or right edges, so every point in the plane is present in exactly one tile. And each tile locates in the layout with its coordinate of low left corner. In addition, each tile has four pointers called corner stitches at its corners, which are used to connect to its

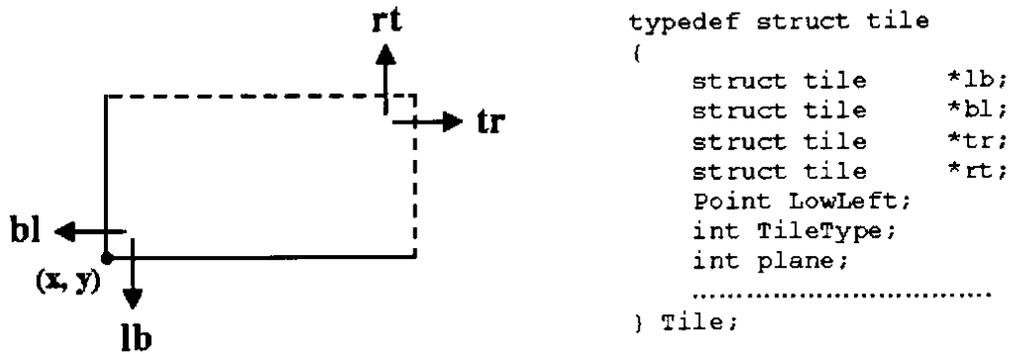


Fig. 4.1: Corner stitching data structure and its C++ code definition.

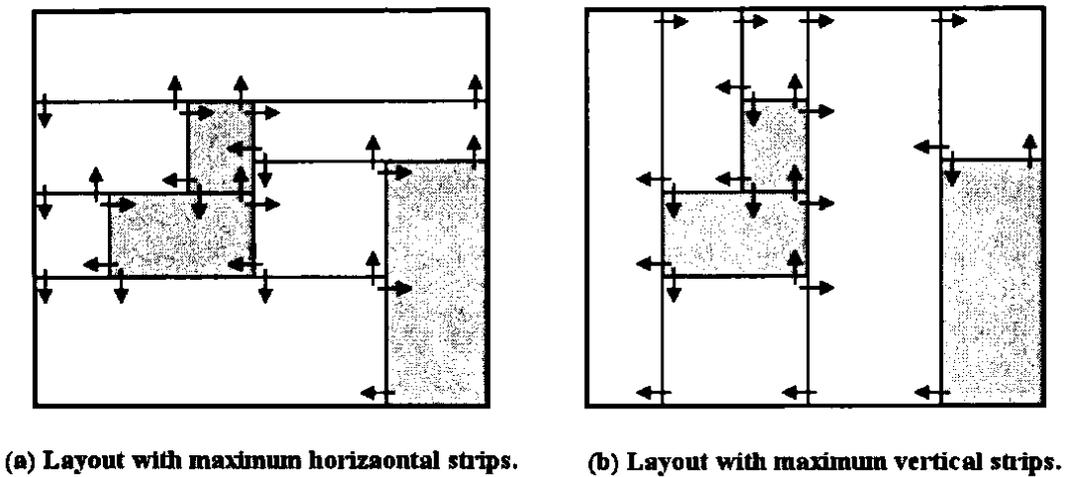


Fig. 4.2: Examples of the corner stitching layout.

neighbors. Therefore, corner stitching data structure has two important features: one is that both space and solid tiles are represented explicitly in the layout; the other is that tiles are stitched together at their corners like a patchwork quilt [Ouster84a] as illustrated in Fig. 4.2.

The disadvantage of corner stitching is that it requires much more memory space since space tiles are also represented in the layout. Theoretically, corner

stitching needs almost three times as much memory space as the simplest data structure (linked list) [Tsai92]. However, memory space can be greatly saved by organizing space tiles as maximal horizontal strips or maximal vertical strips as illustrated in Fig. 4.2. Take the organization of maximum horizontal strips as an example, space tiles should be as wide as possible in the horizontal direction, vertically adjacent space tiles should be merged together if they have the same horizontal span [Ouster84a]. Therefore, in actual circuit layouts, the number of space tiles is about equal to that of solid tiles.

Because each corner stitching data structure records the information of its neighbors and space tiles are explicitly represented, corner stitching provides a variety of powerful and fast layout operations that only depend on the local information, such as point finding, neighbor finding, area searches, tile creation and deletion, and so on. Their expected running times are generally linear with respect to the number of nearby tiles [Ouster84a]. Thus corner stitching data structure is attractive in developing physical design tools such as routing, stretching, compaction, and it is adopted by FnRouter to represent tiles in layout.

4.2 Database of FnRouter

In FnRouter, the primary elements of the database are tiles and planes. Tiles represent the geometric shapes in each plane, and a set of planes represents the whole layout of the chip. Since FnRouter supports routing of a maximum of five metals,

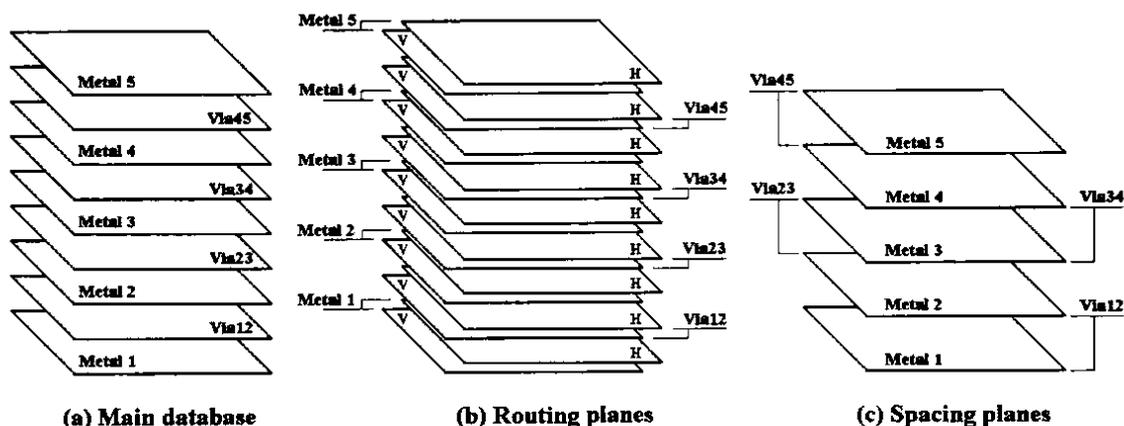


Fig. 4.3: Main database, routing planes, and spacing planes in FnRouter.

five metal planes (**Metal 1**, ..., **Metal 5**) and four via planes (**Via12**, ..., **Via45**) are adopted to represent the main database as illustrated in Fig. 4.3(a). Layouts of metal wire segments and vias are represented as solid tiles in their corresponding planes of the main database. For example, wire segments routed by metal 2 are expressed in the **Metal 2** plane of the main database; vias used to connect metal 1 and metal 2 are expressed in the **Via12** plane of the main database.

In order to facilitate the traditional routing and the pitch adjustment in the lithography-friendly routing research, two additional sets of planes called routing planes (or blockage planes) and spacing planes are implemented in the database of FnRouter. As illustrated in Fig. 4.3(b), the set of routing planes has twice the number as the main database since each layer in the main database corresponds to a dual H-V blockage planes, where the layouts are organized as maximal horizontal strips and maximal vertical strips according to the H-V tile-expansion routing algorithm in FnRouter. As to the set of spacing planes, it only has five planes since layouts in the

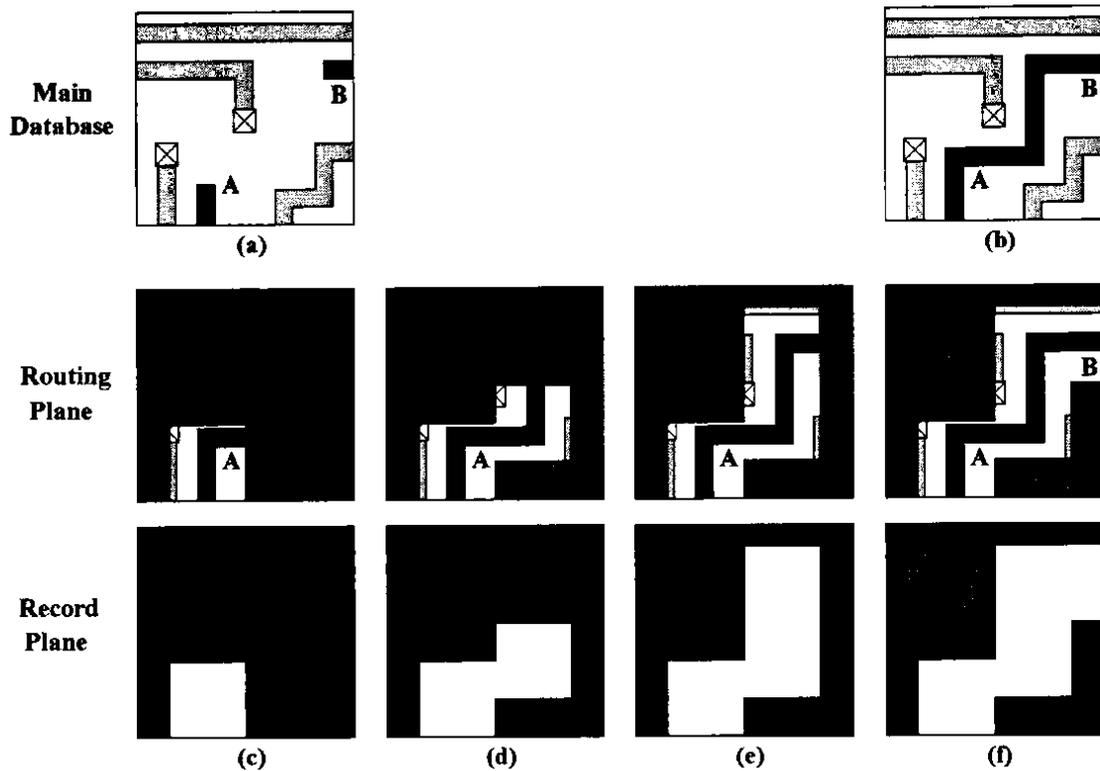


Fig. 4.4: (a) Layout in the main database before routing. (b) Layout in the main database after routing. (c)-(f) Layout in the routing plane during the routing process with the help of the record plane.

via planes of the main database are incorporated into their neighbor metal planes as illustrated in Fig. 4.3(c).

In the initial phase of traditional routing, FnRouter copies layouts from the main database to routing planes. After routing process finishes, the modified layouts are fed back to update the main database. The pitch adjustment to avoid forbidden pitches is executed in the same way by FnRouter. The difference is that layout operations are performed on spacing planes. For time and memory space efficiency, only parts of the layouts concerned by layout operations are copied from the main database to routing planes or spacing planes. A record plane in FnRouter remembers

the coverage areas where layouts are copied so that only these areas in the main database are updated after routing or pitch adjustment. A routing example is demonstrated in Fig. 4.4, (a) and (b) are the layouts in the main database before and after routing, (c)-(f) present the layouts in routing planes and the coverage areas in the record plane during the routing process.

4.3 H-V Tile-expansion Routing Algorithm

In this lithography-friendly routing research, gridless model is preferred since wire segments could be moved to other arbitrary locations without limitations as grid-based model, which gives much more freedom in the iterative pitch adjustment to avoid forbidden pitches. Therefore, the gridless H-V tile-expansion routing algorithm is adopted in FnRouter to accomplish the traditional routing. This algorithm, which was first implemented in the interactive router called Irouter [Arnold88] in Magic layout system, combines maze routing with line-search algorithm. The routing process is controlled by the cost function and executed on the dual H-V blockage planes where tiles are organized as maximal horizontal strips and maximal vertical strips respectively. In the following, some techniques involved in H-V tile-expansion routing algorithm are briefly introduced since they could be improved by FnRouter to realize lithography-friendly routing.

4.3.1 Handling Design Rules

In grid-based routing algorithms, design rule correctness is ensured by carefully determining the grid cell size. While in the H-V tile-expansion algorithm that is based on gridless model, design rule correctness is kept by extending solid tiles in blockage planes. For example, to route the net from the source to the target in the layout existing in the main database as Fig. 4.5(a), each solid tile in the existing layout is extended in all four directions when it is copied to blockage planes. As illustrated in Fig. 4.5(b), solid tile is extended by s to the top and right, and $s+w-1$ to the bottom and left, where s is the minimum pitch between solid tiles and w is the minimum width of the tiles in design rules. After finding the zero-width path from the source to the target that does not impinge on blocked areas in blockage planes, complete path that is design rule correct can be achieved when the zero-width path is filled out to full width wiring [Arnold88] to update the main database as illustrated in Fig. 4.5(c).

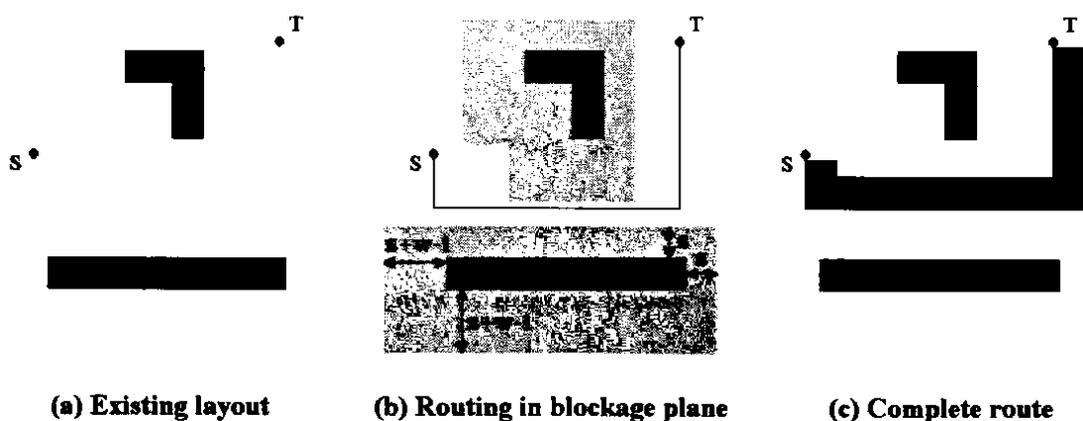


Fig. 4.5: Handling design rules in blockage planes.

Obviously, any zero-width path on space tiles in blockage planes is design rule correct when it is filled out to full width wiring in the main database. If forbidden ranges for each solid tile in the existing layout are also remarked as blocked areas in blockage planes, then the routed path is not only design rule correct but also out of forbidden ranges. One assistant technique in FnRouter to avoid forbidden pitches is just based on the above principle and presented in Chapter 8.

4.3.2 H-V Tile-expansion Process

As discussed above, the main routing process in H-V tile-expansion routing algorithm is just finding the zero-width path from the source to the target without impinging blocked areas in the dual H-V blockage planes (i.e. routing planes). Because H-V tile expansion routing algorithm adopts line-search concept, the zero-width path is composed of a set of line segments that connect “interesting points”. A point on the interior of a space tile in blockage planes is interesting if it aligns with the target, or the available routing space shrinks or expands there [Arnold88].

As illustrated in Fig. 4.6(a), a point of interest can expand in six directions: left, right, top, and bottom in the same routing plane; and up, down to the neighbor routing planes. If the current point of interest expands up or down to neighbor planes by a via, the new point of interest is just the point on neighbor planes with the same coordinates as the current point of interest. As to the expansion on the same routing plane, horizontal expansion, i.e. left and right, is processed on its V blockage plane

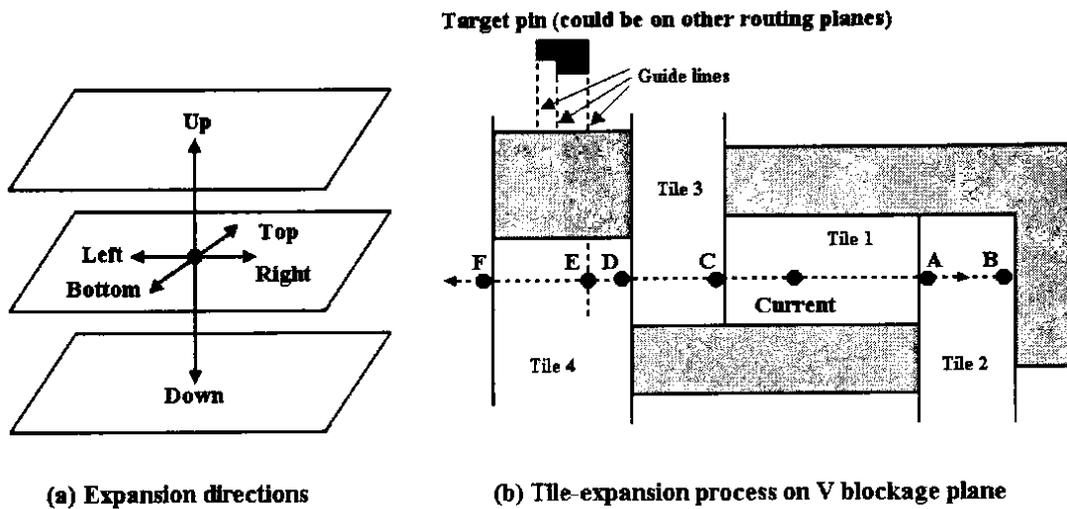


Fig. 4.6: H-V tile expansion process.

where the space tiles are organized as maximum vertical strips; and vertical expansion, i.e. top and bottom, is processed on its H blockage plane where the space tiles are organized as maximum horizontal strips. Fig. 4.6(b) demonstrates the horizontal expansion process on the V blockage plane and the vertical expansion process on the H blockage plane is similar. When the current point of interest expands right, **point A** is the first point of interest that locates near the border (just 1 unit to right from the border) of **Tile 1** and **Tile 2** where the routing space in the perpendicular direction expands. If **point A** continues to expand to the right, **point B** is its next point of interest that locates just before the solid tile which blocks the expansion and results in the shrinkage of the available routing space. Horizontal expansion to the left in Fig. 4.6 (b) is similar as to the right. However, it is different when **point D** expands to the left: **point E** that aligns with one guide line of the target pin is its next point of interest instead of **point F**.

It is obvious that H-V tile-expansion routing algorithm belongs to line-search methods since it skips over uninteresting points and directly finds the next point of interest until reaches the target pin with the help of guide lines. Actually, maze routing is also incorporated in H-V tile-expansion routing algorithm. When the point of interest cannot reach target pins due to the blockage of solid tiles, maze expansion is executed in the local area to bypass them. However, it is only implemented when the line-search processes of other points of interest all fail.

4.3.3 Cost Functions

The H-V tile-expansion routing algorithm is fast since it only searches the points of interest during the expansion process. However, it is still very time-consuming because each point of interest has six expansion directions and the amount of partial path, i.e. the path expands from the source to the points of interest but has not reached the target yet, increases exponentially after several iterative expansions. Therefore, two costs are incorporated in each partial path. One is the actual cost of the partial path, which is accumulated from the source to the current point of interest. The other is the estimated total cost of the partial path, which equals to the actual cost of the partial path plus the estimated cost from the current point of interest to the target. With the introduction of cost, partial paths that expand in the wrong directions, which always have high cost, can be avoided so as to speed up the routing process.

In the iterative expansion process, the partial path that has the least estimated total cost is chosen for the next H-V tile-expansion routing. When the partial path finds the next point of interest, the cost of the new line segment between the current point of interest and the new one is calculated by the cost function and the result is added to the actual cost of the partial path for update. The equation of the cost function is given as follows,

$$COST = HCOST \times \Delta X + VCOST \times \Delta Y + VIACOST + JOGCOST . \quad (4.1)$$

It is obvious that the cost function is a polynomial comprising four terms. The first term is the cost increase in the horizontal direction, where $HCOST$ is the horizontal weight in this routing plane and ΔX is the horizontal distance between the current point of interest and the next point of interest. The second term is the cost increase in vertical direction, where $VCOST$ is the vertical weight in this routing plane and ΔY is the vertical distance between the current point of interest and the next point of interest. The third term $VIACOST$ is the cost increase due to via insertion, which is a constant and only effective when the next point of interest is in neighbor routing planes. The fourth term $JOGCOST$ is also a constant, which represents the cost punishment if the new line segment is perpendicular to the current line segment so as to form the jog structure in the layout, which is not OPC-friendly.

Corresponding to the update of the actual cost of the partial path and the change of the current point of interest, the estimated total cost of the partial path also

should be renewed. In FnRouter, it is calculated by Dijkstra's shortest path algorithm [Dijkstra59] in an estimation plane, which can refer in [Arnold88].

4.4 Rip-up and Reroute

Besides the main routing techniques discussed above, there are still some other routing techniques implemented in FnRouter to facilitate the routing process. Such as the technique that determines the net order for routing since unsuitable routing order may prevent successful completion of routing even though all the nets are indeed routable. In the following, rip-up and reroute technique adopted in FnRouter is briefly introduced since one assistant technique to avoid forbidden pitches is performed in rip-up and reroute phase.

Although suitable net order for routing could improve the routing completion rate, it is generally not possible to complete the connections for all nets in the first running. Therefore, a clean-up phase following the main routing process in FnRouter attempts to connect the remaining nets with rip-up and reroute technique. The key of this technique is the rip-up procedure since the same routing technique can be applied for rerouting nets. In FnRouter, guaranteed path selection [Dees82] strategy is adopted as the rip-up method. The principle of guaranteed path selection is that before routing all nets, a trial routing is performed on each net to find a guaranteed path. Then any routed nets that block the guaranteed path of the unrouted net is ripped up in the clean-up phase. The disadvantages of guaranteed path selection are:

it cannot guarantee the reroutability of the nets ripped up; and the computation and memory requirement can be excessive [Dees82]. However, these drawbacks are not urgent in FnRouter since it is developed to route small chips and maze routing incorporated in H-V tile-expansion routing algorithm can find the path if it exists.

4.5 OPC-friendly Routing

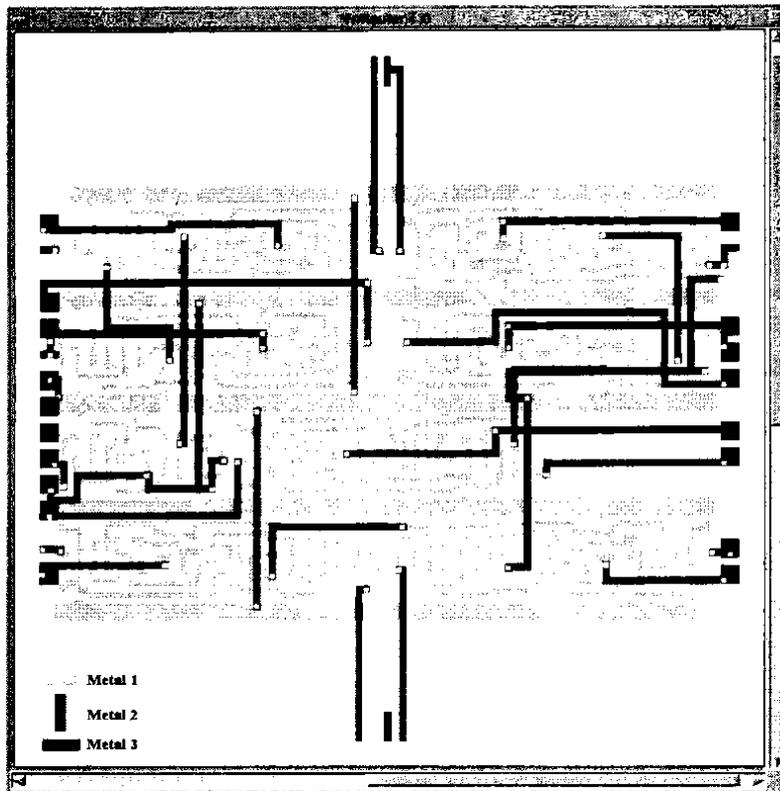
In the last chapter, two approaches adopted by the lithography-friendly routing research in this thesis to design OPC-friendly layout are proposed, both of them can be integrated in the traditional routing techniques. The first one, which uses rectangular line-ends and forbids circle or oval shapes in the layout, is realized by adopting corner stitching data structure in FnRouter so that only Manhattan patterns exist in the layout. The other one, which restricts zig-zag and jog structures in the layout, is realized by carefully designing each term of the polynomial cost function so as to control the routing style.

Design suitable cost function greatly contributes to the OPC-friendly routing in this research. In (4.1), the fourth term *JOGCOST* is specifically introduced to restrict zig-zag and jog structures in the layout. By giving high value to *JOGCOST*, partial paths with zig-zag and jog structures can be avoided during the H-V tile-expansion process since only the partial path with the least cost is selected for the next expansion. Moreover, zig-zag and jog structures can be greatly decreased by assigning large difference between the horizontal weight *HCOST* and the vertical

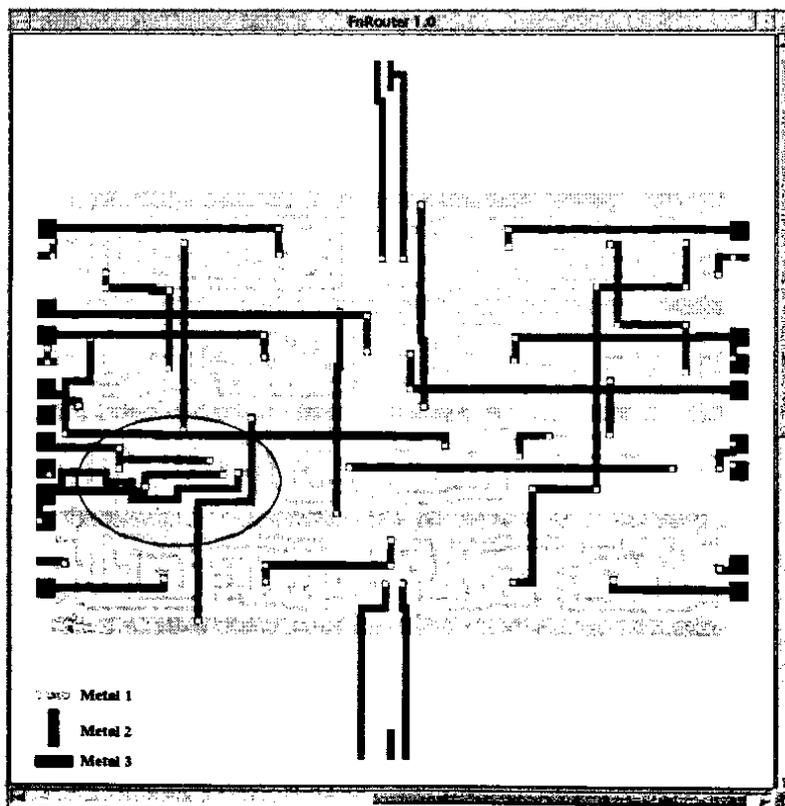
weight $VCOST$ in the cost function. For example, if FnRouter specifies that routing on **Metal 1** has a cost of 1 in the horizontal direction ($HCOST = 1$), and 5 in the vertical direction ($VCOST = 5$), and reverses the two weights for **Metal 2** (i.e. $HCOST = 5$, $VCOST = 1$), routing will have a very strong directionality. However, if the difference between $HCOST$ and $VCOST$ is small, zig-zag and jog structures will be preferred to layer changes for small distances, which leads to the layout not OPC-friendly. Fig. 4.7 demonstrates the influence of cost function on the routing style by controlling the assignment of $HCOST$ and $VCOST$ of each routing plane, which is based on the layout of an 8-bit ripple-carry adder test chip routed by FnRouter with three metals.

In Fig. 4.7(a), same weight is assigned to $HCOST$ and $VCOST$ in each routing plane. It is obvious that many horizontal wire segments are routed on **Metal 2** routing plane, which is reserved for routing vertical wire segments. This is because zig-zag and jog structures are preferred to layer changes for small distances when the weight difference between $HCOST$ and $VCOST$ in each routing plane is small. Therefore, the layout is not very OPC-friendly. In addition, the routing completion rate in Fig. 4.7(a) is only 94.1% since two nets cannot be routed because too many horizontal wire segments occupy the routing resource of **Metal 2** that is reserved for routing vertical wire segments.

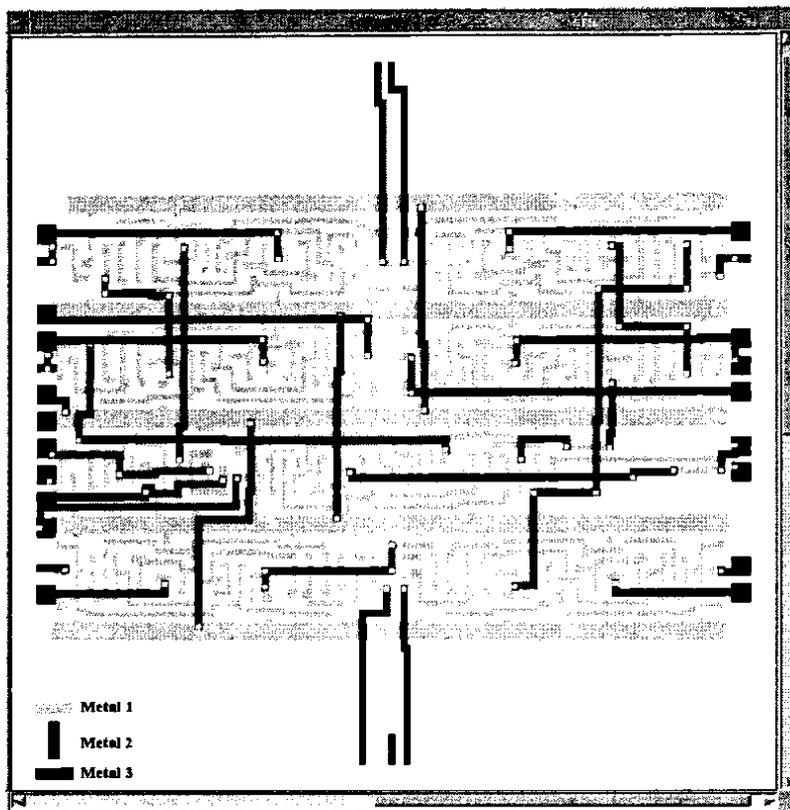
In Fig. 4.7(b), different weights are assigned to $HCOST$ and $VCOST$ in each routing plane ($HCOST = 1$, $VCOST = 2$ on **Metal 1** and **Metal 3**, while $HCOST = 2$, $VCOST = 1$ on **Metal 2**). It is obvious that less horizontal wire segments are generated



(a) Same weight for $HCOSt$ and $VCOSt$ in each routing plane.



(b) $HCOSt = 1$, $VCOSt = 2$ in Metal 1 and 3, and $HCOSt = 2$, $VCOSt = 1$ in Metal 2.



(c) $HCOST = 1$, $VCOST = 5$ in Metal 1 and 3, and $HCOST = 5$, $VCOST = 1$ in Metal 2.

Fig. 4.7: Routing results under different $HCOST$ and $VCOST$ assignment.

on **Metal 2** so that the routing result has a strong directionality. Though the routing completion rate in Fig. 4.7(b) is 100%, the existence of zig-zag and jog structures is still serious in one area of the layout, which is identified with a circle in the layout.

In Fig. 4.7(c), weights assigned to $HCOST$ and $VCOST$ in each routing plane are $HCOST = 1$, $VCOST = 5$ on **Metal 1** and **Metal 3**, while $HCOST = 5$, $VCOST = 1$ on **Metal 2**. The routing completion rate in Fig. 4.7(c) is 100%, and the number of zig-zag and jog structures is less than that in Fig. 4.7(b). Therefore, the layout of Fig. 4.7(c) is more OPC-friendly.

From the above routing examples, it can be concluded that the assignment of *HCOST* and *VCOST* in each routing plane is important to achieve the OPC-friendly layout. Generally, large difference between *HCOST* and *VCOST* is preferred in the cost function. However, too large difference is not necessary since the existence of zig-zag and jog structures is necessary to complete the routing. Take the 8-bit ripple-carry adder with the same chip area in Fig. 4.7 as an example, when the difference between *HCOST* and *VCOST* is more than ten times, the routing completion rate decreases.

4.6 Summary

In this chapter, routing techniques applied in the traditional routing stage of FnRouter that adopts the two-stage lithography-friendly routing method have been presented, which includes corner stitching data structure, database, H-V tile-expansion routing algorithm, ordering of nets, and rip-up and reroute. At the end of this chapter, approaches integrated in the traditional routing techniques to realize OPC-friendly layout have been introduced. Especially, the approach by assigning large difference between *HCOST* and *VCOST* in the cost function to restricts zig-zag and jog structures in the layout has been demonstrated with the 8-bit ripple-carry adder test chip.

Chapter 5 Pitch Optimization

In the proposed two-stage lithography-friendly routing method, pitch adjustment is iteratively executed after the traditional routing stage to move the wire segments away from the forbidden pitches. Pitch adjustment concentrates on avoiding forbidden pitches in one affected area at a time and is divided into two steps: pitch optimization and spacing. In the pitch optimization step, the process window of wire segments forming the forbidden pitch and their neighbors that may be affected during the pitch adjustment process is formulated into a quadratic objective function. Based on the linear geometric constraints in the layout, new optimal locations for wire segments in the affected area are obtained by solving the constrained quadratic optimization problem. Then in the spacing step, each wire segment is adjusted to its new optimal location by spacing technique that keeps the original wire connections and design rules intact during the layout modification process.

5.1 Definition of Forbidden Pitches

The definition of forbidden pitches has been roughly presented in Chapter 2 when introducing the use of SRAFs in conjunction with OAI. Actually, the definition depends on the optical lithography technology adopted and the acceptable process window determined by the designer. In the following, the forbidden pitches implemented in this lithography-friendly routing research are defined, which are

based on the $0.248 \mu\text{m}$ optical lithography technology and at most two SRAFs are used in conjunction with OAI.

Fig. 5.1(a) shows the experimental process window vs. pitch when applying different number of SRAFs [Wong01]. It is obvious that the curves in Fig. 5.1(a) agree with the general process window curve illustrated in Fig. 2.4. Without using SRAFs, the process window drops sharply when the pitch of the main pattern increases. If one SRAF is added, the image quality can be enhanced and then dropped when the pitch keeps increasing. When the second SRAF is added, the process window rises again. Based on the above characteristics, a reasonable curve to utilize SRAFs across the pitch could be achieved as presented in Fig. 5.1(b). If only the patterns with process window more than 0.75 are acceptable in the manufacture, then two pitch ranges (a_1, b_1) and (c_1, d_1) where the process window is below the “acceptable line” can be determined in Fig. 5.1(b). These two pitch ranges are defined as “forbidden range I” and “forbidden range II” respectively since any pitch in these two ranges is the forbidden pitch that should be avoided in the lithography-friendly routing. The actual value of these two forbidden ranges approximate $(1.1 \frac{\lambda}{NA}, 1.4 \frac{\lambda}{NA})$ and $(1.7 \frac{\lambda}{NA}, 1.8 \frac{\lambda}{NA})$. For the optical lithography technique $\lambda = 248 \text{nm}$ and $NA = 0.68$, forbidden pitches are within $401 \text{ nm} - 511 \text{ nm}$ and $620 \text{ nm} - 656 \text{ nm}$ respectively.

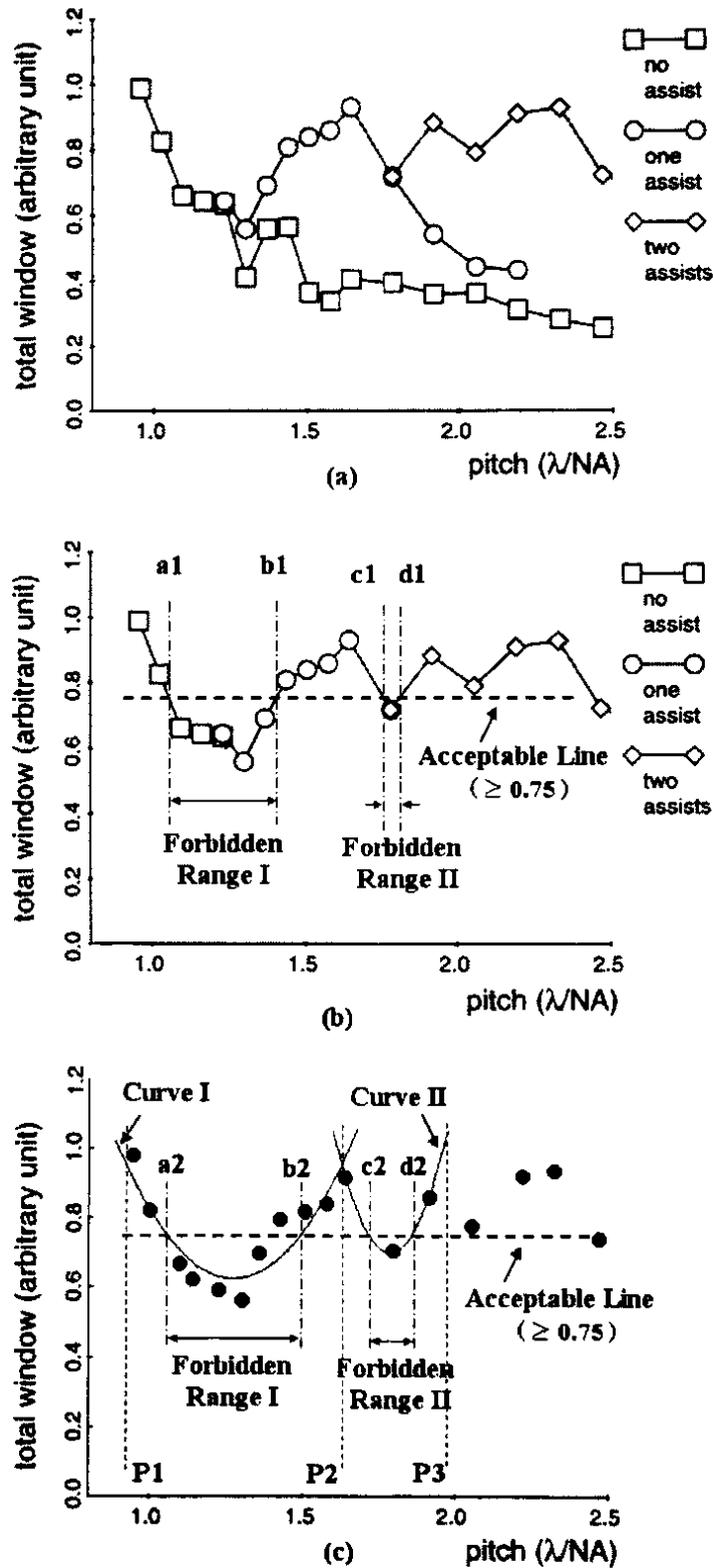


Fig. 5.1: (a) Experimental process window across the pitch with the use of SRAFs [Wong01]. (b) Two forbidden ranges are defined for the experimental process window. (c) Two quadratic curves fit the experimental process window.

The forbidden ranges change according to the adopted optical lithography technology and the choice of acceptable process window. In addition, there could be more than two forbidden ranges when more than two SRAFs are used in conjunction with OAI. However, the proposed two-stage lithography-friendly routing method is not sensitive to the change of forbidden ranges and their number. Therefore, all the discussion on the lithography-friendly routing via forbidden pitch avoidance in the thesis is based on the forbidden ranges defined above, i.e. $401\text{ nm} - 511\text{ nm}$ and $620\text{ nm} - 656\text{ nm}$.

5.2 Formulation of Pitch Optimization

In order to avoid forbidden pitches in the layout after traditional routing, wire segments forming forbidden pitches are adjusted to their new locations by spacing technique in this thesis. However, due to the fact that pitch adjustment for a specific wire segment always interacts with its nearby segments, the layout modification should be considered for all wire segments within the affected area, not just the wire segments forming forbidden pitch only. Besides, the pitch adjustment should not sacrifice the original image quality, i.e. new forbidden pitches should not be created during the pitch adjustment process. Therefore, as the first step of each pitch adjustment, pitch optimization determines the new locations of all wire segments that may be affected during the pitch adjustment with the aim to achieve the highest process window for the patterns in the affected area. The key is to solve a constrained quadratic optimization problem, which is formulated from the

experimental data of the applied optical lithography technology and the geometric constraints in the layout after traditional routing.

5.2.1 Quadratic Curve Fitting

To solve the pitch optimization problem in a mathematical way, experimental data in Fig. 5.1(b) are fitted by two quadratic curves with the least-squares approximation. The two curves are shown in Fig. 5.1(c), with pitch range (P_1, P_2) and (P_2, P_3) respectively. Therefore, the experimental process window across the pitch with the use of SRAFs in Fig. 5.1 can be approximated by the following equation:

$$f(p) = \begin{cases} A_1(p - B_1)^2 + C_1, & P_1 \leq p \leq P_2; \\ A_2(p - B_2)^2 + C_2, & P_2 \leq p \leq P_3. \end{cases} \quad (5.1)$$

In (5.1), p is the pitch size expressed in nm instead of the usual unit $\frac{\lambda}{NA}$; A_1, B_1, C_1 and A_2, B_2, C_2 are the coefficients of the two quadratic curves; P_1 normally equals to the minimum pitch required by design rules (it equals to 280 nm in the TSMC 180 nm technology adopted in this research); P_2 is the cross point of the two curves (it equals to 594 nm in this research); P_3 is the upper boundary of curve II, and it has some freedom for the designer to choose (it equals to 692 nm in this research). If the same “acceptable line” as in Fig. 5.1(b) is chosen, two different forbidden ranges (a_2, b_2) and (c_2, d_2) can be determined. The two new forbidden ranges are always different to

those in Fig. 5.1(b). However, the two old forbidden ranges (a_1, b_1) and (c_1, d_1) are retained in the formulation process since they approximate to the experimental data better than (a_2, b_2) and (c_2, d_2) do.

It should be pointed out that the reason why the experimental data are fitted by two quadratic curves instead of one high degree polynomial curve is based on two considerations. The first one is to simplify the final objective function so as to solve the pitch optimization problem quickly. The other one is due to the distribution of the experimental data that in one specific pitch range such as (P_1, P_2) , dense pitches and sparse pitches have higher process window while pitches with intermediate values have lower process window.

5.2.2 Objective Function

A layout example is first presented in this section to demonstrate the formulation process in obtaining the objective function of the pitch optimization problem, and the formulas for the general case is given in section 5.2.4.

5.2.2.1 Layout Example

Fig. 5.2(a) is a part layout of the 8-bit 3-tap FIR filter test chip after traditional routing. TSMC 180 nm standard cells are adopted for this test chip and

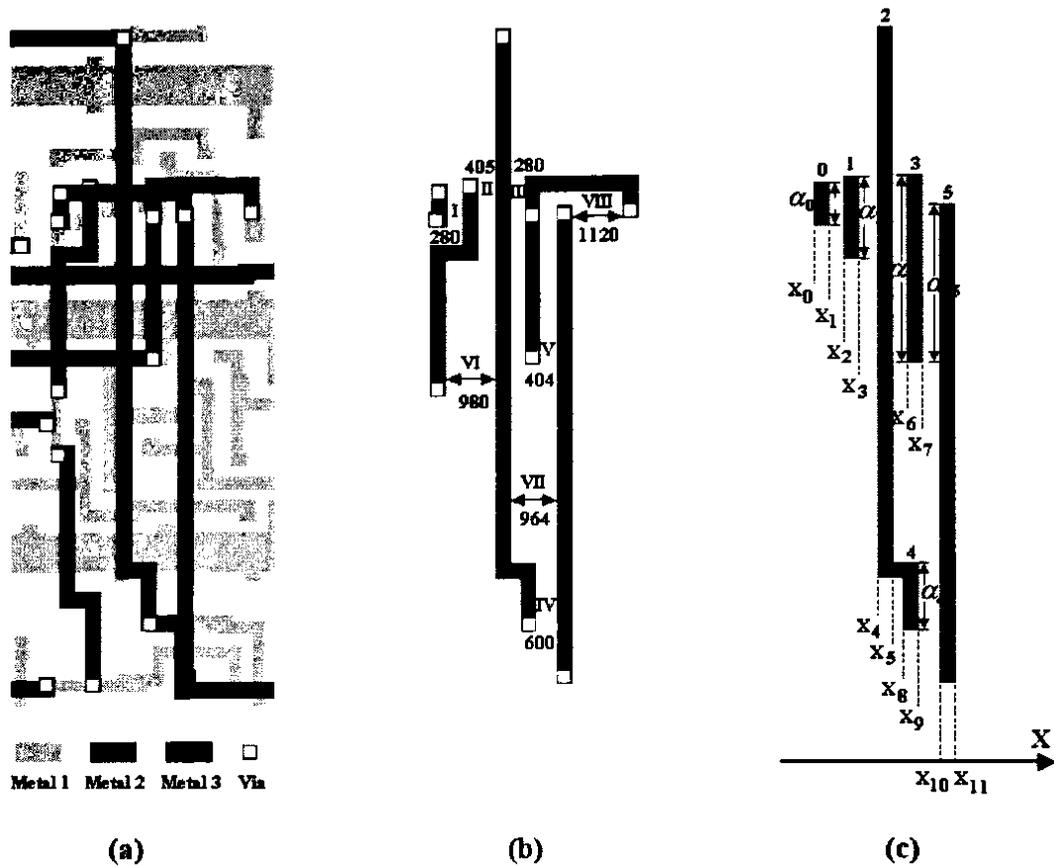


Fig. 5.2: (a) Layout of the example. (b) The patterns in metal 2 (two forbidden pitches exist). (c) Vertical patterns and their geometric size.

nets are connected with four routing planes. Patterns in metal 2 including wire segments and vias are given in Fig. 5.2(b), and the pitch values between wire segments are also presented. Since the forbidden pitch ranges adopted in the research are $401\text{ nm} - 511\text{ nm}$ and $620\text{ nm} - 656\text{ nm}$ respectively, pitch II and pitch V in Fig. 5.2(b) are both forbidden pitches. When FnRouter adjusts the locations of wire segments that form the forbidden pitches II and V, other pitch sizes are also changed. Therefore, the pitch optimization is executed for all wire segments within the affected area in Fig. 5.2(b). Without loss of generality, since wire segments in Fig. 5.2(b) are only adjusted in the horizontal direction, layout can be simplified as in Fig.

5.2(c) and the six vertical wire segments are labeled from 0 to 5. Some wire segments forming the pitch VI and VIII are not presented in Fig. 5.2(c) because the pitch optimization does not consider the pitches beyond P_3 defined in (5.1).

In order to achieve the highest process window for the patterns in the affected area of the layout, the objective function for the example layout in Fig. 5.2 is given as

$$\begin{aligned} \text{Max } F(p) &= \text{Max} [F(p_{01}) + F(p_{12}) + F(p_{23}) + F(p_{35}) + F(p_{45})] \\ &= \text{Max} [\alpha_{01}f(p_{01}) + \alpha_{12}f(p_{12}) + \alpha_{23}f(p_{23}) + \alpha_{35}f(p_{35}) + \alpha_{45}f(p_{45})], \end{aligned} \quad (5.2)$$

where

$$p = [p_{01}, p_{12}, p_{23}, p_{35}, p_{45}]^T. \quad (5.3)$$

In (5.2), $F(p)$ is the objective function; p_{01} , p_{12} , p_{23} , p_{35} , and p_{45} are sizes of pitch I, II, III, IV, and V; $f(p_{01})$, $f(p_{12})$, $f(p_{23})$, $f(p_{35})$, and $f(p_{45})$ are the process window of pitch I, II, III, IV, and V according to (5.1); α_{01} , α_{12} , α_{23} , α_{35} , and α_{45} are weights for these five pitches and are proportional to the length of the overlapping area between the two wire segments forming the pitch.

As demonstrated in (5.2), any wire segment pair that exists in the affected area and forms a pitch whose size is smaller than P_3 contributes to the objective function. Therefore, pitch VII in Fig. 5.2(b) is not included in the objective function since the pitch size is larger than P_3 . In addition, pitches with vias are also not

included in the objective function due to their small geometric sizes. However, they are taken care of in the layout constraints discussed in section 5.2.3.

5.2.2.2 Graph Expression

Objective function in (5.2) should be automatically produced by FnRouter when it avoids forbidden pitches in the layout after traditional routing. Therefore, graph expression, a more abstract level than the actual layout, is adopted in FnRouter to facilitate the process. In this expression, each wire segment that may be adjusted in the affected area is abstracted as a vertex in the graph and its geometric information such as its coordinates in the layout is recorded in its corresponding vertex data structure. In addition, for any two wire segments forming a pitch small than P_3 , an edge in solid line is used to connect their corresponding vertexes in the graph to represent their contribution to the objective function. For any two wire segments forming jog structure, an edge in dashed line is used to connect their corresponding vertexes in the graph to represent that the jog constraint, one of the layout constraints, may exist between them.

Fig. 5.3 demonstrates the process to build the graph for the affected area. Layout in Fig. 5.2(b) without vias is shown in Fig. 5.3(a). Since the vertical wire segments 1 and 2 forms a forbidden pitch, wire segment 2 could be selected as the seed for building the graph. Since wire segment 2 is a vertical pattern, the left and right zone not further than P_3 is defined as the search area. Any other vertical wire

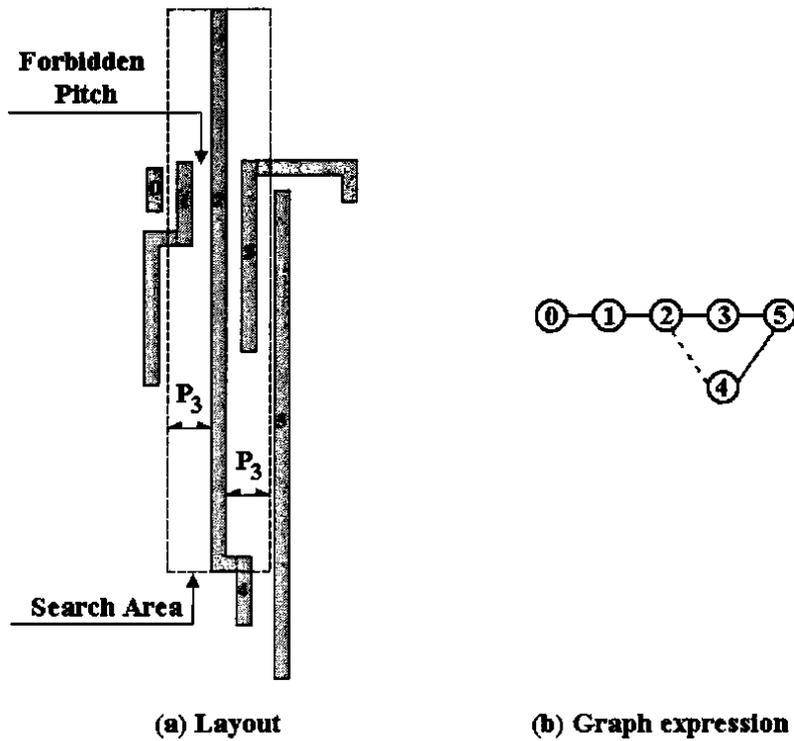


Fig. 5.3: Graph expression for example.

segment within the search area is converted to a vertex in the graph and connected to vertex 2 with solid or dashed edges according to the rules specified above. After all vertical patterns in the search area including 1, 3 and 4 are represented in the graph, the new vertexes 1, 3 and 4 will be taken as new seeds and the iteration process for building the graph will be repeated until there is no new vertex. Fig. 5.3(b) is just the final graph expression for the layout example, which has 6 vertexes.

5.2.2.3 Matrices of Objective Function

Any two vertexes connected by a solid edge in the graph contribute to the objective function of pitch optimization. If the two vertexes are with the label i and

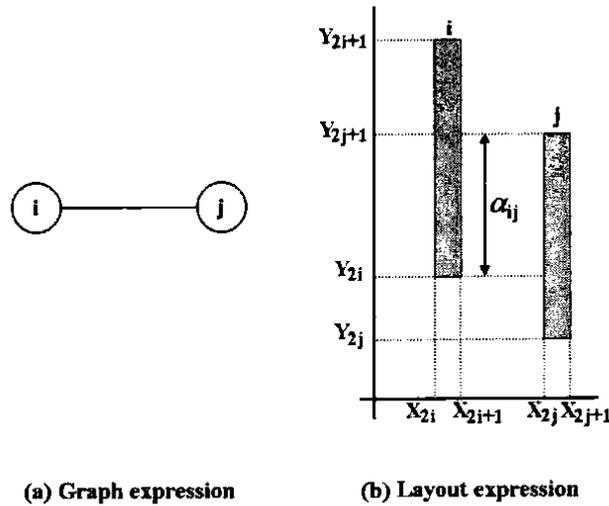


Fig. 5.4: Two vertical wire segments and its graph expression.

j as Fig. 5.4(a), their coordinates in the layout could be assumed as in Fig.5.4 (b). Then the pitch between the wire segment pair i and j is

$$p_{ij} = x_{2j} - x_{2i+1}. \quad (5.4)$$

If p_{ij} is assumed in the pitch range (P_1, P_2) defined in (5.1), the contribution of the wire segment pair i and j to the objective function using (5.1) is:

$$\begin{aligned} F(p_{ij}) &= F(x_{2j} - x_{2i+1}) = \alpha_{ij} [A_1 (x_{2j} - x_{2i+1} - B_1)^2 + C_1] \\ &= \alpha_{ij} [A_1 (x_{2j} - x_{2i+1})^2 - 2A_1 B_1 (x_{2j} - x_{2i+1}) + A_1 B_1^2 + C_1]. \end{aligned} \quad (5.5)$$

The constant terms in (5.5) can be neglected in the objective function because it does not affect the final optimal result. Then (5.5) can be written as

$$\begin{aligned}
F(p_{ij}) &= F(x_{2j} - x_{2i+1}) = \alpha_{ij} [A_1(x_{2j} - x_{2i+1})^2 - 2A_1B_1(x_{2j} - x_{2i+1})] \\
&= \alpha_{ij} A_1 \begin{bmatrix} x_{2i+1} \\ x_{2j} \end{bmatrix}^T \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_{2i+1} \\ x_{2j} \end{bmatrix} + 2\alpha_{ij} A_1 B_1 \begin{bmatrix} 1 \\ -1 \end{bmatrix}^T \begin{bmatrix} x_{2i+1} \\ x_{2j} \end{bmatrix}
\end{aligned} \tag{5.6}$$

Denote

$$H_{ij} = \begin{bmatrix} 2\alpha_{ij}A_1 & -2\alpha_{ij}A_1 \\ -2\alpha_{ij}A_1 & 2\alpha_{ij}A_1 \end{bmatrix} = \begin{bmatrix} h_{ij} & -h_{ij} \\ -h_{ij} & h_{ij} \end{bmatrix}, \tag{5.7}$$

and

$$f_{ij} = \begin{bmatrix} 2\alpha_{ij}A_1B_1 \\ -2\alpha_{ij}A_1B_1 \end{bmatrix} = \begin{bmatrix} h_{ij} \cdot B_1 \\ -h_{ij} \cdot B_1 \end{bmatrix}, \tag{5.8}$$

then (5.6) becomes

$$F(p_{ij}) = F(x_{2j} - x_{2i+1}) = \frac{1}{2} \begin{bmatrix} x_{2i+1} \\ x_{2j} \end{bmatrix}^T H_{ij} \begin{bmatrix} x_{2i+1} \\ x_{2j} \end{bmatrix} + f_{ij}^T \begin{bmatrix} x_{2i+1} \\ x_{2j} \end{bmatrix}. \tag{5.9}$$

For the weight α_{ij} in (5.7) and (5.8), if the ratio to the length of the overlapping area is set to 1, then

$$\alpha_{ij} = y_{2j+1} - y_{2i}. \tag{5.10}$$

From (5.9), the contribution of any wire segment pair i and j to the objective function is in quadratic form and can be expressed by two sub-matrices H_{ij} and f_{ij} which relate to the variable $[x_{2i+1}, x_{2j}]$. Obviously, all the elements in these two sub-matrices can be determined by the geometric size of the wire segment pair i and j .

The objective function is the sum of contributions from all the wire segment pairs whose corresponding vertexes in the graph are connected by solid edges.

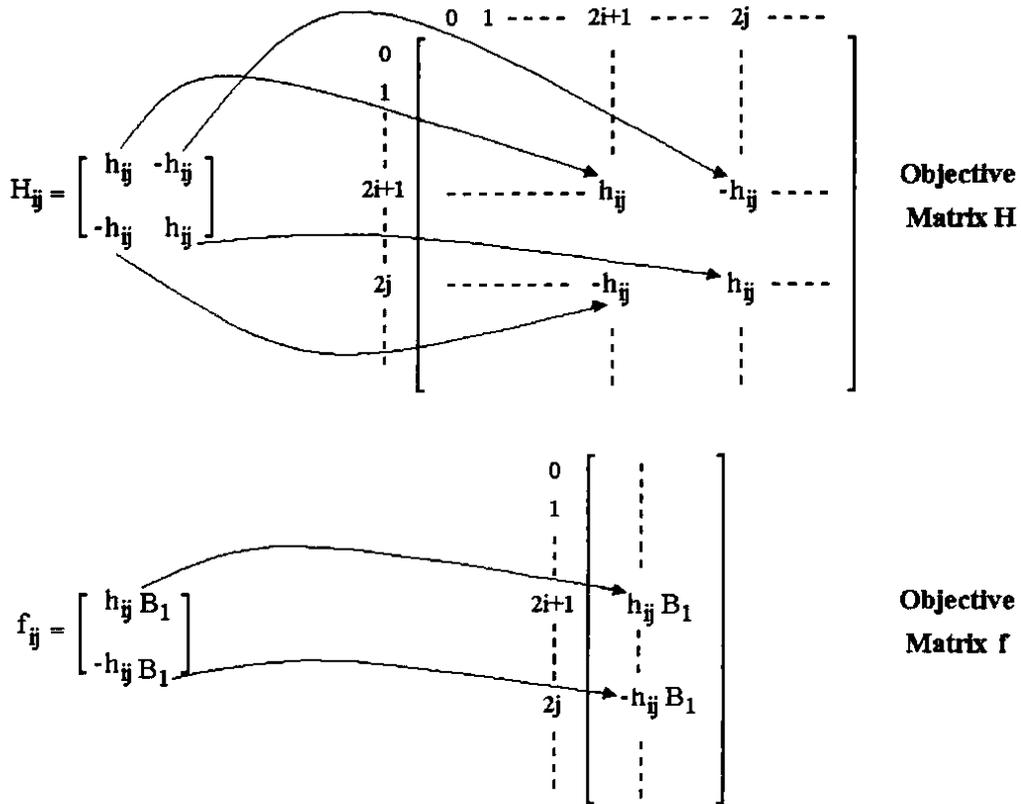


Fig. 5.5: Submatrices and objective matrices.

Therefore, the elements in the final objective matrices H and f that relate to all variables in the objective function are the accumulation of elements from all submatrices as in (5.7) and (5.8). Fig. 5.5 illustrates the relationship between the submatrices H_{ij} , f_{ij} and the objective matrices H and f .

With the help of graph expression, FnRouter can easily obtain the objective function of the layout example when it traverses the graph in Fig. 5.3(b) with the depth-first search method [Nyhoff99]. In Fig. 5.2(b), except pitch IV that is in the pitch range (P_2, P_3) , other pitches are all in the pitch range (P_1, P_2) . Therefore, the objective function of the layout example is given as

$$\text{Max } F(x) = \text{Max} \left(\frac{1}{2} x^T H x + f^T x \right), \quad (5.11)$$

where

$$x = [x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}]^T, \quad (5.12)$$

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & h_{01} & -h_{01} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -h_{01} & h_{01} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_{12} & -h_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -h_{12} & h_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & h_{23} & -h_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -h_{23} & h_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{35} & 0 & 0 & -h_{35} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{45} & -h_{45} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -h_{35} & 0 & -h_{45} & h_{35} + h_{45} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (5.13)$$

$$f = \begin{bmatrix} 0 \\ h_{01} \cdot B_1 \\ -h_{01} \cdot B_1 \\ h_{12} \cdot B_1 \\ -h_{12} \cdot B_1 \\ h_{23} \cdot B_1 \\ -h_{23} \cdot B_1 \\ h_{35} \cdot B_1 \\ 0 \\ h_{45} \cdot B_2 \\ -h_{35} \cdot B_1 - h_{45} \cdot B_2 \\ 0 \end{bmatrix}, \quad (5.14)$$

and

$$h_{01} = 2\alpha_{01} A_1, \quad (5.15)$$

$$h_{12} = 2\alpha_{12} A_1, \quad (5.16)$$

$$h_{23} = 2\alpha_{23} A_1, \quad (5.17)$$

$$h_{35} = 2\alpha_{35} A_1, \quad (5.18)$$

$$h_{45} = 2\alpha_{45} A_2. \quad (5.19)$$

Though the above formulation process in obtaining the quadratic objective function is based on the example layout in Fig. 5.2 with the vertical wire segments, the principle to generate the objective function for the general case and the cases with the horizontal wire segments is the same. In section 5.2.4, more precise description for building the objective function for the general case is presented.

5.2.3 Layout Constraints

In this section, geometric constraints in the layout for solving the quadratic optimization problem are discussed. They can be divided into design rule constraints, inherited constraints, compulsory constraints, and optional constraints.

5.2.3.1 Design Rule Constraints

Since the pitch adjustment should keep the layout design rules intact, the solution to the objective function is therefore constrained by the design rules especially the space constraints among wire segments. For example, if the minimum pitch required in the design rules for wire segment pair i and j in Fig. 5.4(b) is d_{\min} , then the following constraint

$$x_{2j} - x_{2i+1} \geq d_{\min} \quad (5.20)$$

should be imposed for the solution. In the following, three special design rule constraints including boundary constraints, via constraints, and permanent fixed pattern constraints are introduced. The first two constraints limit the size of the optimization problem and the third one is to keep the original placement information.

5.2.3.1.1 Boundary Constraints

Ideally, pitch optimization should be executed for all wire segments in the affected area of the layout. However, sometimes the affected area is irregular and too large. Take the pathological pattern in Fig. 5.6(a) as an example, where only pitch I between wire segments 0 and 1 is a forbidden pitch. If the wire segments 0 and 1 are adjusted to the left or right direction to avoid the forbidden pitch, all the other wire segments in Fig. 5.6(a) may be affected since they should at least keep the minimum space from their neighbors according to the design rules. The irregular and large affected area always leads to a great number of variables in the final objective function and more complex layout constraints, which increases the running time. Therefore, boundaries in parallel with the wire segments are supplied to limit the affected wire segments in the local area so as to simplify the pitch optimization problem.

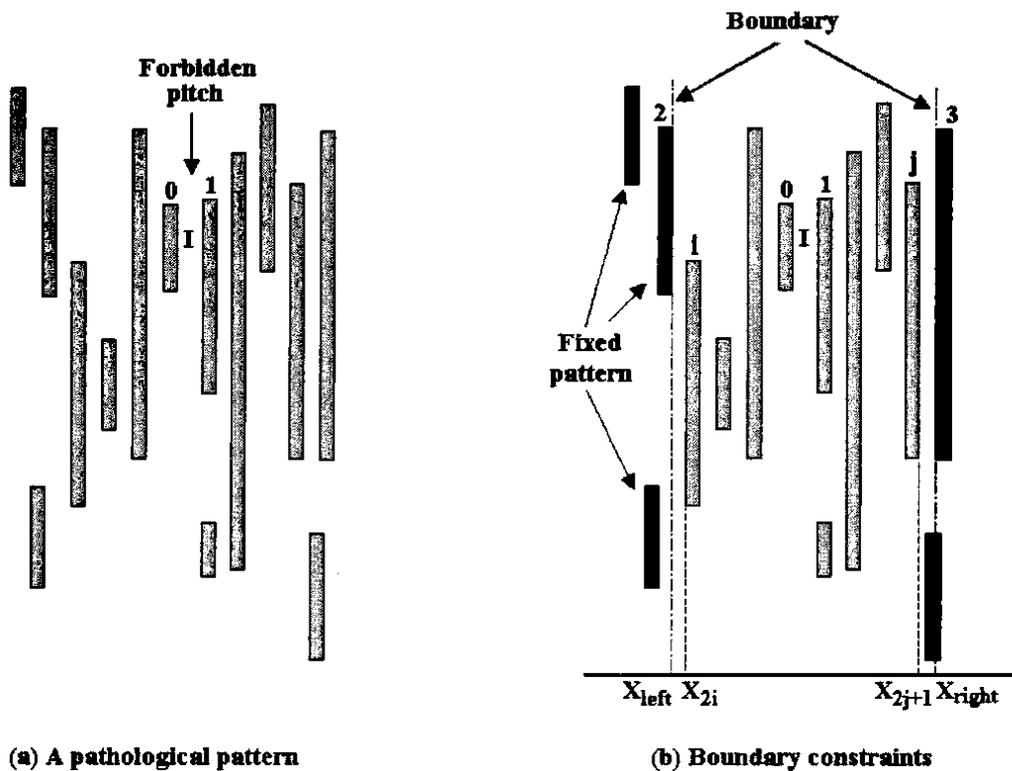


Fig. 5.6: An example of the boundary constraints.

The choice of boundaries depends on the complexity of the affected area. For example, the choice of left boundary X_{left} and right boundary X_{right} in Fig. 5.6(b) regulates that the pitch adjustment for avoiding forbidden pitch I at most affects the third wire segments far away from wire segments 0 and 1. In addition, boundaries could be reduced during the pitch optimization process when a solution is not found within the set maximum number of iteration.

Any wire segment outside or touching the boundaries is considered as the fixed pattern in the layout, i.e. it cannot be adjusted in the spacing step. Then design rules, especially the requirement of minimum pitch with fixed patterns, should be considered when the wire segments inside the boundaries are adjusted. These

constraints due to the introduction of boundaries are called boundary constraints. In Fig. 5.6(b), wire segments i and j have the boundary constraints as follows:

$$x_{2i} - x_{left} \geq d_{min}, \quad (5.21)$$

$$x_{right} - x_{2j+1} \geq d_{min}, \quad (5.22)$$

where d_{min} is the minimum pitch in the design rules.

5.2.3.1.2 Via Constraints

The adoption of via constraints is another measure to simplify the pitch optimization, which limits all pitch adjustments on one metal plane and isolates the affected area from other metal planes. Two situations are considered in the via constraints: the first one is that via is directly attached to the fixed elements, such as pins of the chip or terminals of standard cells, as via A in Fig. 5.7(a); the other is that via is attached to the metal wire in the other plane, as via B in Fig. 5.7(b).

In Fig. 5.7(a), the wire segment in metal 2 directly connects to a fixed pin in metal 1 by via A, and the available shift distance of via A to pin's left and right border are D_{left} and D_{right} respectively. If the original values of x_{2i} and x_{2i+1} are D_{2i} and D_{2i+1} , then via constraints for the wire segment in metal 2 are as follows:

$$x_{2i} \leq D_{2i} + D_{right} \quad (5.23)$$

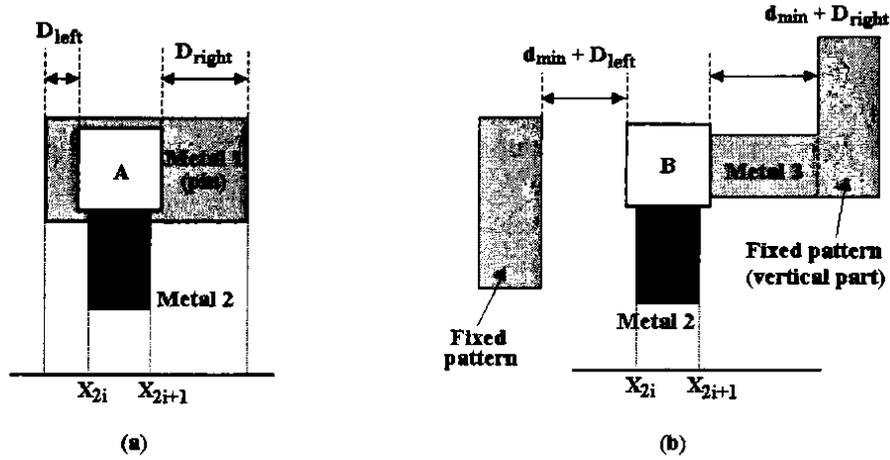


Fig. 5.7: (a) The wire segment in metal 2 connects to the pin by via A. (b) The wire segment in metal 2 connects to the wire segment in metal 3 by via B.

$$-x_{2i} \leq D_{left} - D_{2i} \quad (5.24)$$

$$x_{2i+1} \leq D_{2i+1} + D_{right} \quad (5.25)$$

$$-x_{2i+1} \leq D_{left} - D_{2i+1} \quad (5.26)$$

Via constraints (5.23)-(5.26) ensure the original wire connection and enough via coverage to pins. As to the situation in Fig. 5.7(b), similar techniques as in the boundary constraints are adopted: vertical wire segments in metal 3 except via B is considered as the fixed pattern. If the minimum pitch required by the design rules is d_{min} , then D_{left} and D_{right} in Fig. 5.7(b) are the largest distance that via B can be adjusted to the left and the right without violating the design rules. Via constraints in this situation are the same as (5.23) to (5.26).

5.2.3.1.3 Permanent Fixed Pattern Constraints

Pitch adjustment by spacing technique is only implemented to the wire segments routed by FnRouter. Other metal materials that exist before routing, such as the metal connections inside the standard cells and the placed metal pins of the chip, should be fixed on their original locations in the spacing step. Therefore, these metal materials are also looked as fixed patterns as the wire segments outside the boundaries. The difference is that these metal materials are permanently fixed even if they exist inside the boundaries.

To avoid moving the permanent fixed patterns during the pitch adjustment process, permanent fixed pattern constraint similar as the boundary constraint is considered, which forbids the wire segments in the affected area to be moved into the areas within the minimum pitch with the permanent fixed patterns. For example, if the vertical pattern k in Fig. 5.8 is permanently fixed, the wire segment i and j thus

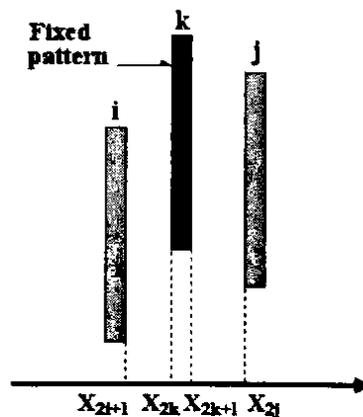


Fig. 5.8: Permanent fixed pattern constraints.

have the following constraints:

$$x_{2k} - x_{2i+1} \geq d_{\min}, \quad (5.27)$$

$$x_{2j} - x_{2k+1} \geq d_{\min}, \quad (5.28)$$

where d_{\min} is the minimum pitch in the design rules.

A variety of design rule constraints increase the difficulty to automatically produce layout constraints for each wire segment. However, the design rule constraints of each wire segment could be looked as its maximum shift distances along the adjustment direction without conflicting with the design rules. FnRouter adopts spacing technique to test the maximum shift distance for each wire segment in the affected area so as to achieve all design rule constraints, which is discussed in the next chapter.

5.2.3.2 Inherited Constraints

Inherited constraints exist in each wire segment pair that forms a pitch in the affected area. As in Fig. 5.1(c), more than one quadratic curves are utilized to approximate the experimental process window within the pitch range. Therefore, constraints should be imposed to ensure that the pitch still lies within the original pitch range after the layout modification. For example, if the pitch of the wire

segment pair i and j in Fig. 5.4(b) is in the range (P_1, P_2) , then their coordination should be kept as follows:

$$x_{2j} - x_{2i+1} \leq P_2, \quad (5.29)$$

$$x_{2i+1} - x_{2j} \leq -P_1. \quad (5.30)$$

Width constraint is another type of inherited constraints. Currently, only the space between wire segments is adjusted for avoiding the forbidden pitches, the width of wire segments should always be kept constant. Therefore, for each vertical wire segment i whose width equals to w , the two variables x_{2i} and x_{2i+1} which express its left and right borders as in Fig. 5.4(b) should agree with the following constraint:

$$x_{2i+1} - x_{2i} = w. \quad (5.31)$$

As to the horizontal wire segment i whose width equals to w , the two variables y_{2i} and y_{2i+1} which express its bottom and top borders should agree with the following constraint:

$$y_{2i+1} - y_{2i} = w. \quad (5.32)$$

5.2.3.3 Compulsory Constraints

Compulsory constraints, which are imposed by the designer to influence the solution of the optimization problem, are used to deal with cases that the outcome is worse off. Because forbidden ranges only occupy a small fraction of the whole pitch

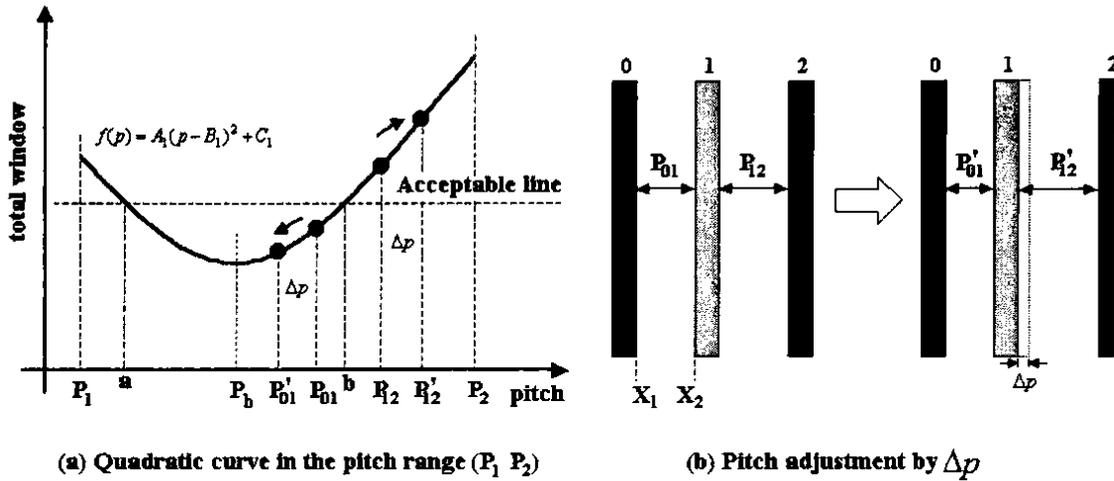


Fig. 5.9: An example of worse case.

range and the shift range for each wire segment in the layout is usually sufficiently wide, most forbidden pitches can be avoided by solving the optimization problem with the above layout constraints and applying spacing technique. However, in some “worse cases” where the shift range of wire segments is narrow due to the layout constraints, the process window of the individual patterns forming the forbidden pitch may be sacrificed to enhance the whole process window of all patterns within the affected area.

Fig. 5.9 presents a simple example of “worse case” that consists of three vertical wire segments with the same length. Wire segment 0 and 2 are both fixed patterns and have the same overlapping length with wire segment 1. Pitch p_{01} and p_{12} locate in the pitch range (P_b, P_2) and p_{01} is a forbidden pitch according to the definition in Fig 5.9(a). Assuming wire segment 1 can only be adjusted to the left or right with the distance Δp due to the layout constraints, and the adjusted pitch p'_{01}

and p'_{12} still locate in the original pitch range (P_b, P_2) no matter wire segment 1 is moved to the left or the right. Then the objective function for the example can be achieved according to the same deduction process in 5.2.2. Since the weight of the two pitches is same, the objective function could be presented as follows:

$$\text{Max } F(p) = \text{Max} [f(p_{01}) + f(p_{12})], \quad (5.33)$$

and $f(p_{01})$, $f(p_{12})$ can be calculated by (5.1). Based on the derivative form of (5.1) in the pitch range (P_1, P_2)

$$f'(p) = 2A_1(p - B_1), \quad (5.34)$$

it is obvious that in the pitch range (P_b, P_2) , enlarging the pitch p_{12} with Δp could achieve more process window increase than enlarging the pitch p_{01} with Δp . If wire segment 1 is adjusted to the left with Δp as in Fig. 5.9(b), the process window of the forbidden pitch p_{01} is worse off as in Fig. 5.9(a), though the objective function (5.33) gets the highest value.

Therefore, a check step is added in the pitch optimization step to compare the process window of the forbidden pitch after solving the optimization problem. If the process window of the forbidden pitch becomes worse off, compulsory constraints are imposed and the pitch optimization step is repeated. As to the example in Fig. 5.9 where pitch p_{01} is in the forbidden range (a, b) , one of the following compulsory constraints

$$x_2 - x_1 \geq b \quad (5.35)$$

$$x_1 - x_2 \geq -a \quad (5.36)$$

should be imposed to avoid the forbidden pitch in the “worse case”.

5.2.3.4 Optional Constraints

Optional constraints are not necessary in the pitch optimization step to avoid forbidden pitches. However, they are introduced to emphasize the layout optimization besides forbidden pitches. Currently, the jog constraint is the only optional constraint.

Though FnRouter is based on the reserved layer model that horizontal and vertical wire segments are restricted to particular layers, jog structures, i.e. short length wire segments with the direction not restricted in particular layers, are widely used to decrease the number of vias and make the layout more compact. However, the existence of jogs deteriorates the image quality and makes the layout not OPC-friendly. Therefore, for the cases that the jog size in the affected area is small, constraints can be optionally adopted by the designer to straighten the jog. Take the pattern in Fig. 5.10(a) as an example, if the length of the horizontal wire segment connecting the vertical wire segment i and j is short enough, the following constraints

$$x_{2i} = x_{2j} \quad (5.37)$$

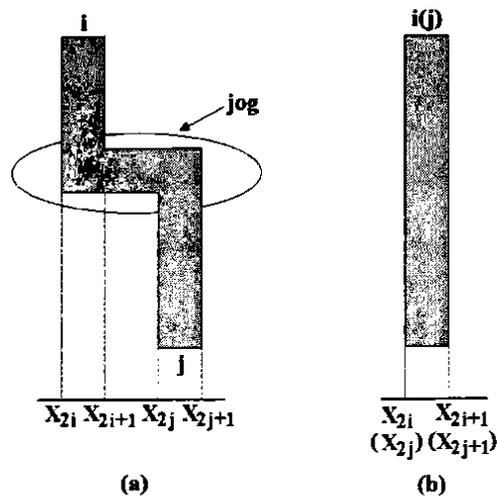


Fig. 5.10: (a) Jog example. (b) Straighten the jog.

$$x_{2i+1} = x_{2j+1} \quad (5.38)$$

can be optionally adopted to straighten the jog as in Fig. 5.10(b). FnRouter can automatically produce jog constraints by traversing the graph since any two vertexes whose corresponding wire segments form jog structures are connected with a dashed edge.

5.2.4 General Case

Without loss of generality, the formulation of the pitch optimization for the general case that forbidden pitch is formed by vertical wire segments is presented in the following. The general case that forbidden pitch is formed by horizontal wire is similar and will not be presented.

If a forbidden pitch p formed by two vertical wire segments i and j exists in the layout, and n vertical wire segments including i and j inside the boundaries are affected during the pitch adjustment process to avoid the forbidden pitch p , then the pitch optimization for the affected area can be formulated as a constrained quadratic optimization problem as follows,

$$\text{Max } F(x) = \frac{1}{2} x^T Hx + f^T x \quad (5.39)$$

subject to

$$A \cdot x \leq b, \quad (5.40)$$

$$A_{eq} \cdot x = b_{eq}, \quad (5.41)$$

$$x \geq 0, \quad (5.42)$$

where

$$x = [x_0, x_1, x_2, \dots, x_{2(n-1)}, x_{2(n-1)+1}]^T. \quad (5.43)$$

In (5.39), elements in objective matrices H and f are obtained by accumulating all the contributions of the wire segment pair forming a pitch whose size is smaller than P_3 in the affected area. All the layout inequality constraints are expressed with matrices A and b in (5.40), and all the layout equality constraints are expressed with matrices A_{eq} and b_{eq} in (5.41).

Four types of layout constraints discussed in section 5.2.3 are expressed altogether in (5.40) and (5.41), which leads to some layout constraints being redundant. For example, via constraints (5.25) and (5.26) are not necessary when the

inherited width constraint (5.31) is introduced. Therefore, redundant constraints are checked and thrown away from the final constraint matrices A , b , A_{eq} , and b_{eq} .

It is obvious that the objective function of the pitch optimization problem in (5.39) is quadratic, and the constraints (5.40)-(5.42) are linear. Such optimization problem is called quadratic programming (QP) problem in the numerical optimization research [Pardalos02]. Algorithms for solving QP problem have been thoroughly researched by mathematicians, which include classical active-set methods, gradient-projection methods, and interior-point method [Nocedal99]. And some mathematic packages with C codes are supplied by researchers to solve QP problem in the academy, such as COPL_QP [Zhang98], CFSQP [Lawrence97]. However, these mathematic packages have limitations such as the matrix H in the objective function (5.39) is required to be positive definite, which could not be always satisfied in the current problem. For example, the matrix H in (5.13) for the layout example is not positive definite. In FnRouter, Matlab API function **quadprog** in the optimization toolbox [Matlab98] is adopted to solve the QP problem, which contains several typical algorithms and could automatically select the suitable algorithm to solve QP problem based on the constraints supplied.

Though the formulas of the general case is deduced from the layout example with only two forbidden ranges, the formulation of the pitch optimization can be straightforwardly extended to situations with more than two forbidden ranges by introducing additional quadratic curves to approximate the experimental process window across the pitch with the use of more than two SRAFs.

5.3 Pitch Optimization in FnRouter

After traditional routing, FnRouter searches patterns in each metal layer and records locations of forbidden pitches. For each forbidden pitch, the pitch adjustment is executed and the flow chart of the pitch optimization in FnRouter is shown in Fig. 5.11. Firstly, the objective function and layout constraints in the affected area, which are expressed by matrices H , f , A , b , A_{eq} , and b_{eq} , are automatically produced by traversing the graph. Then, a C++ program that utilizes the Matlab API function

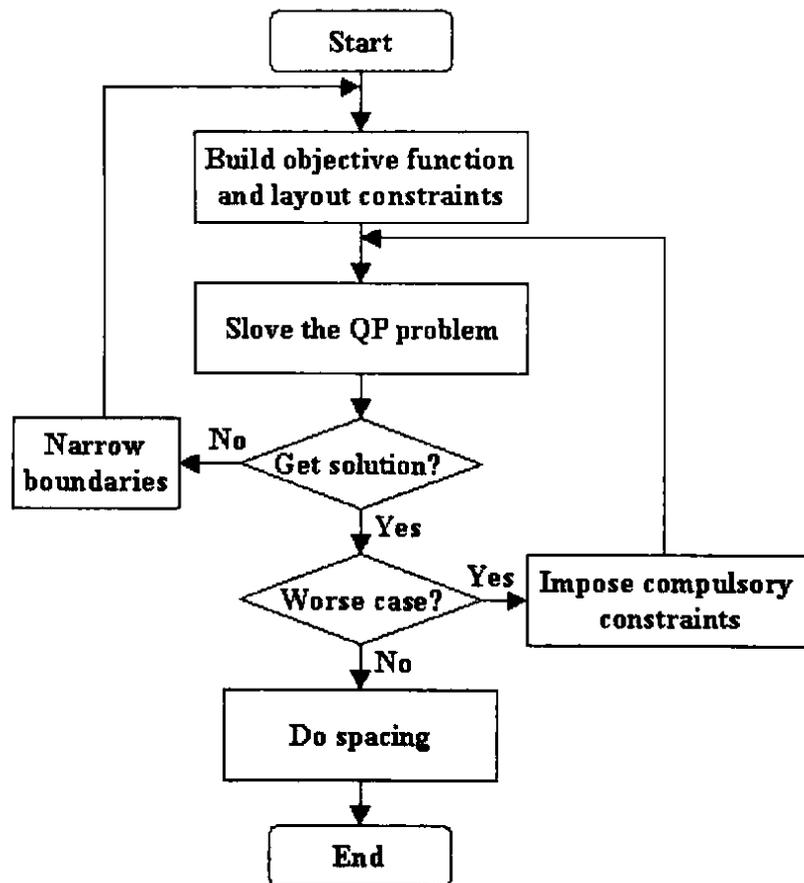


Fig. 5.11: The flow chart of the pitch optimization in FnRouter.

quadprog in the optimization toolbox is called by FnRouter to solve the QP problem. If the solution is found, it produces the new optimal locations for wire segments in the affected area. Otherwise, FnRouter narrows the boundaries and repeats the process. After each optimization, a check step is executed. If it is a worse case, compulsory constraints are imposed and optimization is repeated. Lastly, spacing technique is applied to adjust the pitches in the affected area. The above procedure is repeated until there is no forbidden pitch left or the preset maximum number of iterations is reached.

5.4 Summary

In this chapter, the formulation of the pitch optimization step in the iterative pitch adjustment stage has been introduced. Pitch optimization that calculates the new optimal locations for wire segments in the affected area for the pitch adjustment by spacing technique to avoid forbidden pitches has been formulated as a QP problem. The objective function is the accumulation of the process window of each pitch in the affected area inside the boundaries and the constraints are the geometric restrictions in the layout including design rule constraints, inherited constraints, compulsory constraints and optional constraints.

Chapter 6 Spacing Technique

In the iterative pitch adjustment stage, after solving the QP problem formulated from the pitch optimization step, wire segments in the affected area are adjusted to their new optimal locations by spacing technique in the spacing step so as to avoid the forbidden pitches. Spacing technique, similar to layout compaction, is a method to adjust the distances between wire segments so as to improve the layout performance while keeping the original wire connectivity and design rules intact. However, it not only compacts but also stretches the wire segments.

6.1 Plowing Operation

Spacing technique is developed from the “plowing” operation [Scott84] provided by the Magic layout editor to design symbolic layout. As an interactive operation, plowing can stretch and compact Manhattan layouts while keeping wire connectivity and design rules intact under the instructions of designer, which include plow location, plow direction, plow distance, and so on. The key of the plowing operation is finding edges (boundaries between materials) and moving them based on the rules predefined in its rule table.

Fig. 6.1 presents an example of plowing operation. The vertical line segment that parallels with the four wire segments in Fig. 6.1(a) is called plow and the plow

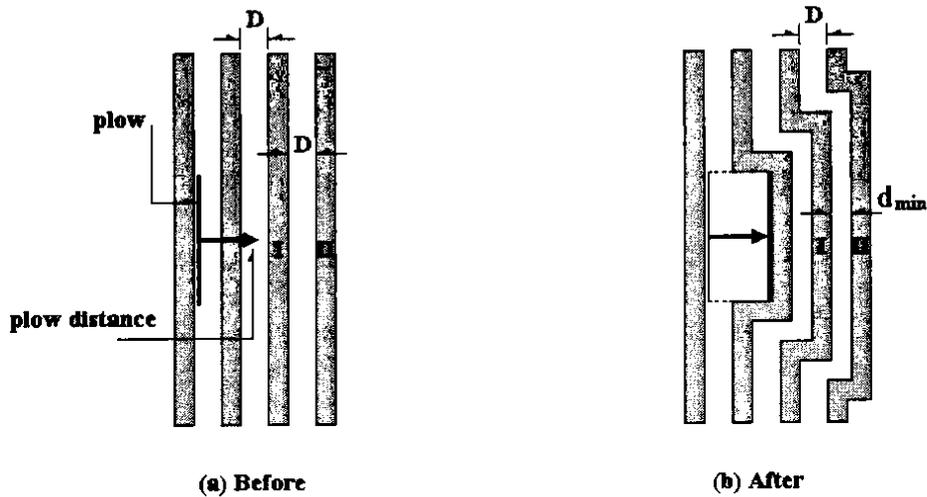


Fig. 6.1: Plowing operation [Scott84].

direction is to the right. During the plowing process, the plow catches vertical edges and carries them along with it as it moves through the layout by the distance specified. Each edge that is caught by the plow pushes other edges ahead of it in a way that preserves wire connectivity and keeps the design rule intact. This means that material in front of the plow is compacted down to the minimum pitch permitted by the design rules as illustrated in Fig. 6.1(b) that the pitch between wire segments I and II is decreased from D to the minimum pitch d_{\min} , and material behind the plow is stretched. Recursively, the pushed edges can cause others to be moved out of the way until no further edges need to be moved. A mound of edges thus builds up ahead of the plow in the manner as snow builds up on the blade of a snowplow [Scott84]. This is the reason why the operation is named as plowing.

6.2 Spacing Technique

Plowing operation in Magic layout editor modifies the layout with materials of diffusion, polysilicon, via, and metal. However, the pitch adjustment in FnRouter only concentrates on metal wires, i.e. only materials of metal and via need to be considered. Therefore, the algorithm of plowing operation is simplified when it is applied to FnRouter. Furthermore, some modifications are introduced to facilitate its usage in FnRouter. As a result, it is called spacing technique in FnRouter instead of plowing operation to identify their difference. In the following, their differences are presented and the explanation of spacing technique still uses the terminology of plowing operation.

6.2.1 Edge Definition

Edge is the basic element operated by plowing operation or spacing technique, which is a boundary between two types of materials. Since the pitch adjustment in FnRouter concentrates on materials of metal and via, only six kinds of edges need to be considered in spacing technique, which are illustrated in Fig. 6.2.

Because the plow direction, no matter to the top, bottom, or left, could be transformed to the right by rotating the layout when FnRouter copies it from the main database to spacing planes, the edge definition in Fig. 6.2 is given with the assumption that the plow direction is to the right. The minimum width w_{\min} and the

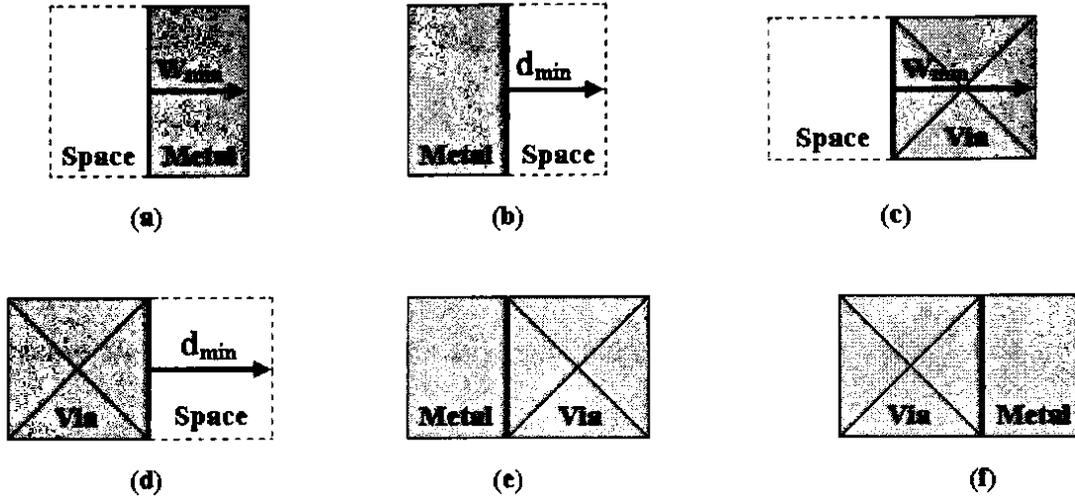


Fig. 6.2: Six kinds of edges in spacing technique.

minimum pitch d_{min} of metal and via in the design rules are defined as the minimum width of the right material of edges so that the design rule is kept intact when plowing the edges to the right.

6.2.2 Plowing Rules

For each edge caught by the plow, a set of rules predefined in the rule table is applied to determine which other edges must move as a consequence of this motion. These rules that can be referred to in [Scott84] ensure plowing operation keep wire connectivity and design rule intact during the layout modification. Therefore, most of them are inherited by spacing technique. In addition, an extra rule called fixed pattern rule is appended to the rule table of spacing technique to help FnRouter

achieve the complete design rule constraints for each wire segment in the affected area.

As discussed in the last chapter, boundary constraints, via constraints, and permanent fixed pattern constraints introduce fixed patterns in the layout. They increase the difficulty in automatically producing the layout constraints for each wire segment in the affected area. Since design rules are always kept intact during the plowing process, the design rule constraints of each wire segment could be considered as its maximum plowing distances along the directions perpendicular to itself without moving fixed patterns. Therefore, fixed pattern rule is developed to facilitate the process to achieve the design rule constraints by spacing technique.

Fig. 6.3 illustrates the principle to achieve the design rule constraints, or the maximum plowing distance in the horizontal direction without moving fixed patterns,

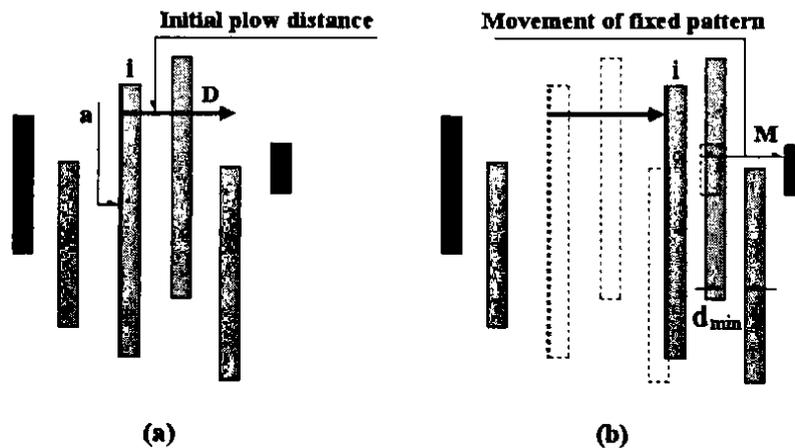


Fig. 6.3: Achieve the maximum plowing distance by spacing technique.

for wire segment i by spacing technique. To achieve the maximum plowing distance to the right, a plowing test is applied to wire segment i , which selects the left border of wire segment i as the plow and moves right by an initial plow distance D larger than P_3 defined in (5.1). The initial plow distance D is always so large that the fixed pattern has to be moved right by some distance M so as to maintain the design rule correctness as illustrated in Fig. 6.3(b). When the spacing technique moves the edge of the fixed pattern (it is expressed as the metal material with an extra flag in spacing planes), the fixed pattern rule is excited, which interrupts the plowing test and modifies the plow distance from D to $D - M$. Since the plowing process keeps the design rule intact, $D - M$ is just the maximum plowing distance to the right for wire segment i . Similarly, the maximum plowing distance to the left for wire segment i can be also achieved by the plowing test.

6.2.3 Self-instructions

As an interactive operation, plowing operation modifies the layout after the designer supplies the instructions of plow location, plow direction, plow distance, and so on. In contrast, pitch adjustment by spacing technique should be automatically implemented after the pitch optimization step. Therefore, spacing technique should create these instructions by itself so as to make the plowing operation without human interference, which are accomplished by comparing the current locations of wire segments in the affected area with their new optimal locations achieved by solving the QP problem.

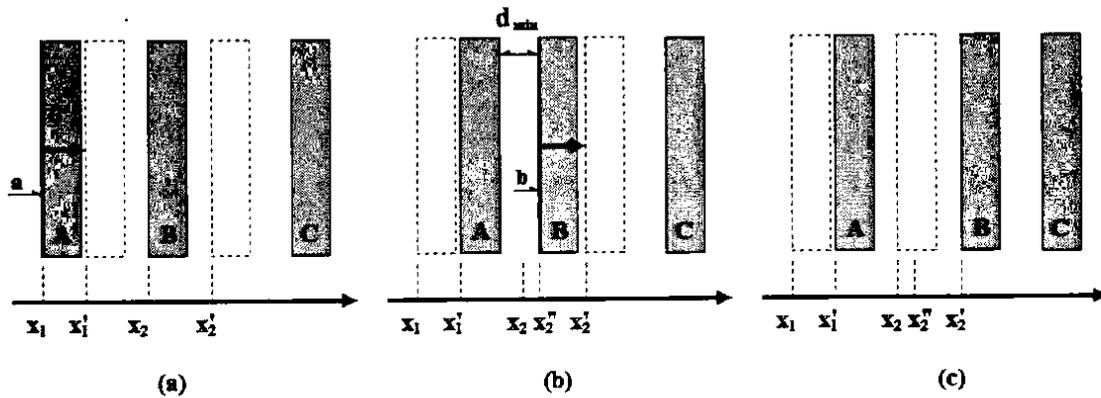


Fig. 6.4: Pitch adjustment by self-instructions. (a) Moving wire segment A. (b) Moving wire segment B. (c) Layout after the pitch adjustment.

Fig. 6.4 presents a layout example to demonstrate the principle of pitch adjustment by self-instructions. There are three vertical wire segments in Fig. 6.4(a) and their pitches need to be adjusted by moving the left border of wire segment A from x_1 to x'_1 , and the left border of wire segment B from x_2 to x'_2 respectively. The pitch adjustment in the spacing step is executed from the leftmost wire segment. By comparing the present location of wire segment A with its new locations, spacing technique determines that the plow direction is to the right and the plow distance is the difference between x_1 and x'_1 . By selecting the left border of wire segment A as the plow, spacing technique moves it from x_1 to x'_1 . During the plowing process, the left border of wire segment B is also pushed right from x_2 to x'_2 so as to keep the minimum pitch d_{min} with wire segment A. The process to adjust the location of wire segment B is same, the difference is that the current location of the left border of wire segment B is not x_2 but x'_2 .

6.2.4 Jog Control

In Magic layout editor, plowing creates jogs whenever it moves only part of the boundary between two different types of materials. As illustrated in Fig. 6.5(a), when plowing the edge a, jogs are automatically inserted in the final layout as in Fig. 6.5(b) since only part of the boundaries of the wire segments such as edge b, c, and d are moved. The introduction of a great number of jogs by plowing not only increases the data volume of the layout but also deteriorates the chip manufacturability since jog pattern is not OPC-friendly. To forbid new jogs being created due to plowing, spacing technique adopts jog control, which extends the edge to include the whole wire segment when plowing it, as illustrated in Fig. 6.5(c).

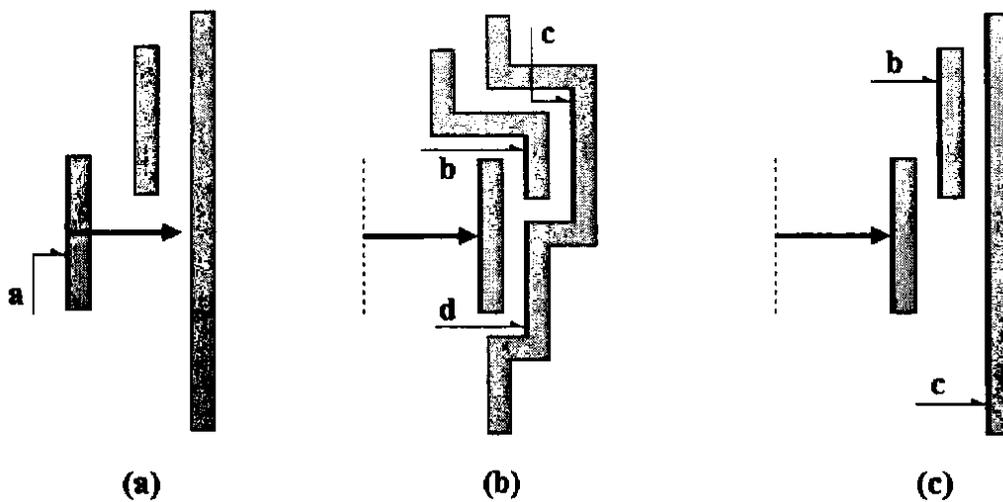


Fig. 6.5: Jog control. (a) Plowing edge a. (b) Plowing by jog insertion. (c) Plowing without jogs.

6.3 Implementation in FnRouter

In the iterative pitch adjustment stage of FnRouter, spacing technique does not directly deal with the layout data in the main database, but accomplishes all the operations in the spacing planes. Since FnRouter supports routing of a maximum of five metals, there are five spacing planes in FnRouter and materials that interact with each other should exist in the same spacing plane. As illustrated in Fig. 4.3(c), **Metal 1** and **Via12** are both in the first spacing plane since they could be connected together. Certainly, **Via12** are also in the second spacing plane with **Metal 2** since they also could be connected together.

Fig. 6.6 illustrates the flow chart of spacing technique when it is implemented in the spacing step, which consists of four phases including initialization, layout data preparation, plowing edges, and main database update. To avoid dependence on a particular technology, spacing technique is parameterized by a set of design rules contained in a technology file. Therefore, the tasks of initialization include defining the edges and building up the plowing rule table according to the design rules of the adopted technology, which are expressed in the input technology file. At the same time, FnRouter cleans the five spacing planes for later operations. Since the pitch adjustment by spacing technique only modifies the layout in the affected area limited by the boundaries, for time and memory space efficiency, only the layout inside the boundaries and some fixed patterns around are copied from the main database during the layout data preparation phase. In order to make the plow direction always to the

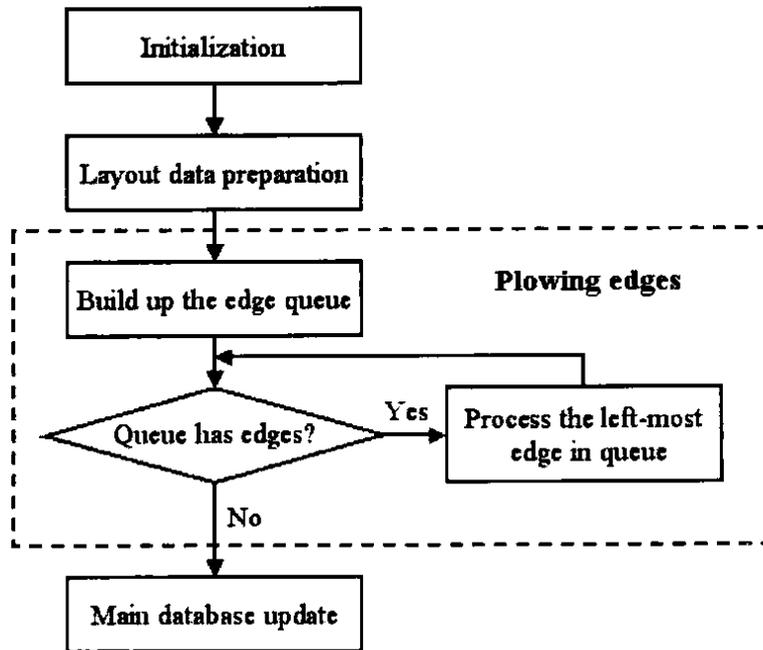


Fig. 6.6: The flow chart of spacing technique in the spacing step.

right, the copied layout is rotated by 90° , 180° , or 270° if the original plow direction is to the bottom, to the left, or to the top before it is rebuilt in the spacing planes. After the plowing edge phase, the modified layout by spacing technique is copied back to the main database to update the corresponding layout. However, when the spacing technique is implemented to achieve the design rule constraints for wire segments in the pitch optimization step, the main database need not be updated.

6.3.1 Plowing Edges

In the plowing edges phase, spacing technique finds edges and moves them based on the plowing rules predefined in its rule table. To facilitate the

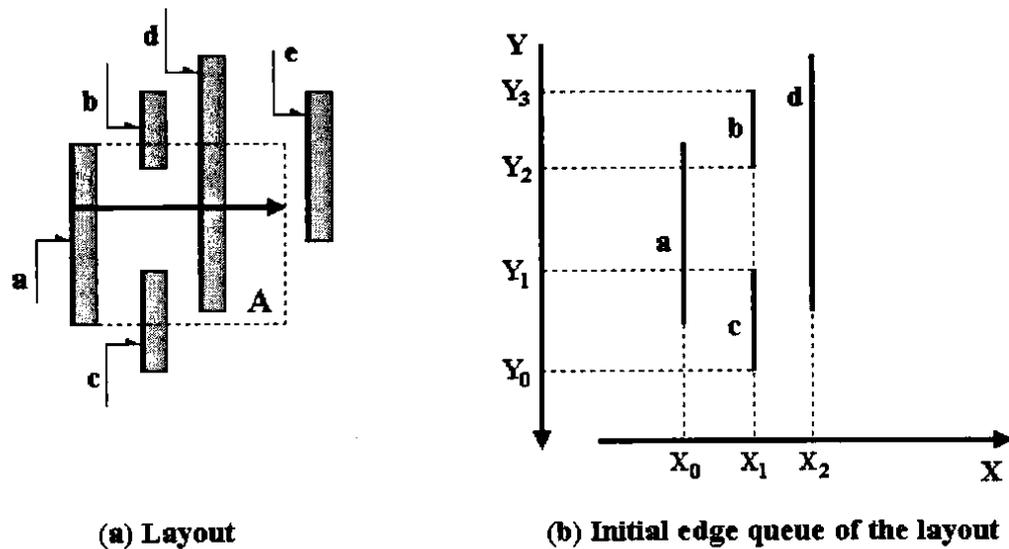
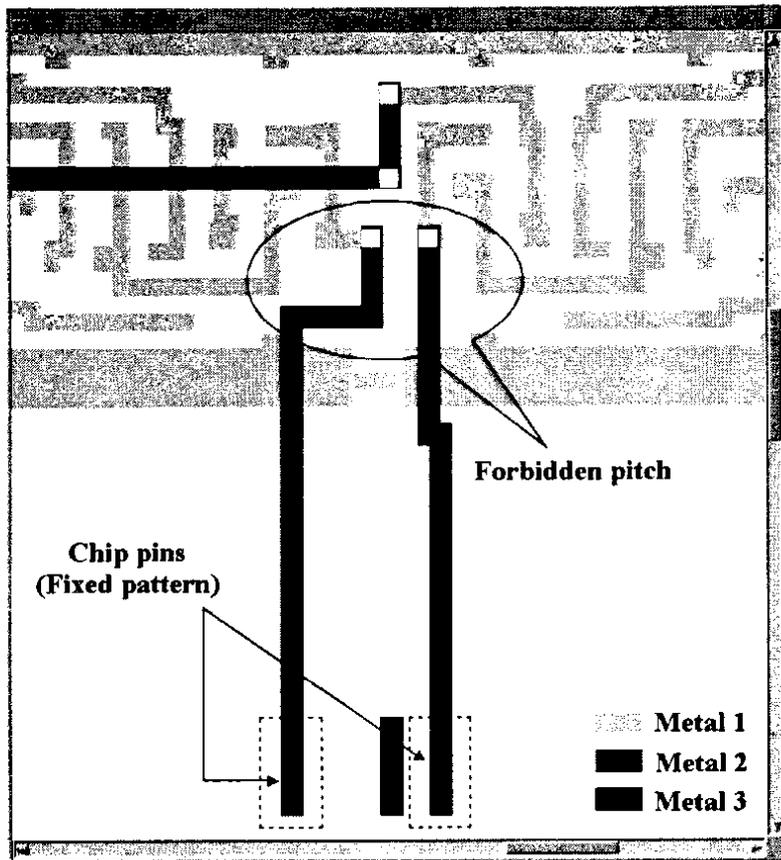


Fig. 6.7: Edge queue in the plowing edges phase.

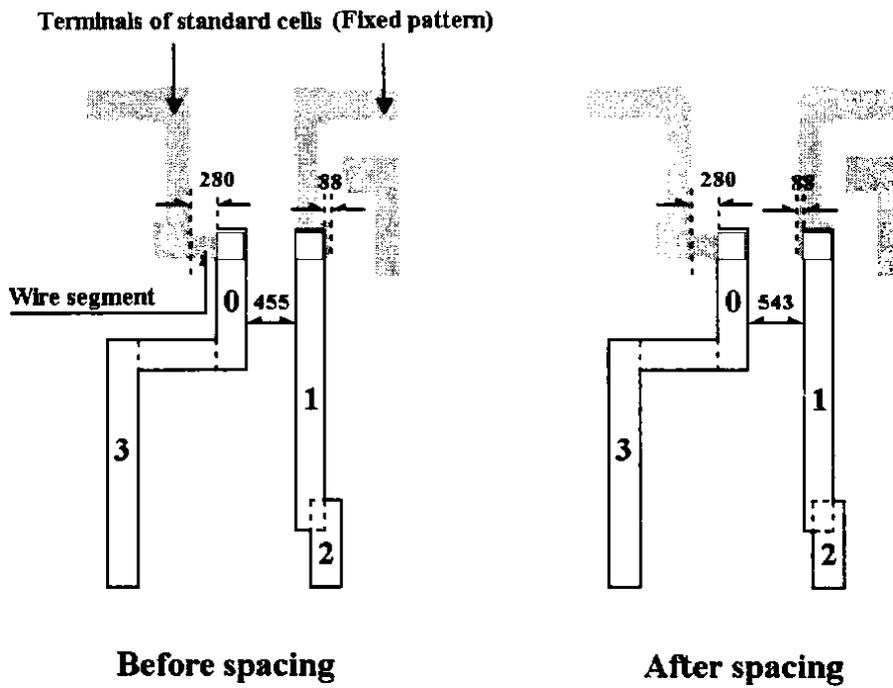
implementation of spacing technique, the edge queue data structure is adopted in FnRouter. Since the plow direction is always to the right, the edge queue arranges all the edges in one spacing plane according to their x coordinates, and for the edges that have the same x coordinates, they are further organized as a link list according to their y coordinates. The edge queue is not constructed in one step, but is always updated when applying plowing rules to edges. As illustrated in Fig. 6.7, the initial edge queue only records the edges in area A, which is specified by the location of the plow and the plow distance. Spacing technique processes the leftmost edge in the edge queue by applying the plowing rules according to the edge type. As a result, some new edges that should be moved according to the rules are inserted to the edge queue for update. As illustrated in the flow chart of Fig. 6.6, such process does not finish until there is no edge left in the edge queue.

6.3.2 Layout Example

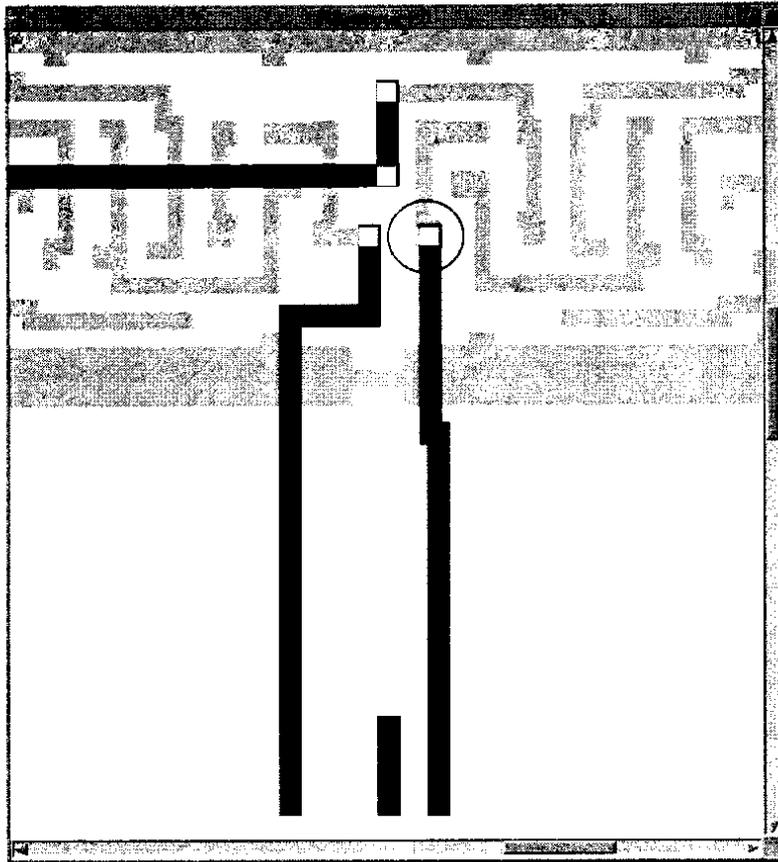
The implementation of spacing technique for the pitch adjustment is demonstrated by an example in Fig. 6.8. The layout in Fig. 6.8(a) is from the 8-bit ripple-carry adder test chip, where a forbidden pitch is formed by two vertical wire segments of metal 2. Fig. 6.8(b) shows all wire segments that may be affected by the pitch adjustment, where the vertical wire segments in metal 2 are labeled from 0 to 3. In the pitch optimization step, the design rule constraints of each vertical wire segment could be obtained by spacing technique with the plowing test. Because wire segment 2 and 3 both connect to the pins of the chip, which are permanent fixed patterns, they cannot be moved in the pitch adjustment. As to wire segment 0 that connects to the terminal of standard cell with a small horizontal wire segment in metal 1 through a via, the maximum plowing distance to the left is 0 since the space between the via and terminal is just 280 nm , the minimum pitch for TSMC 180 nm technology. However, its maximum plowing distance to the right is 263 nm . As to wire segment 1 that directly attaches the terminal of a standard cell, the maximum plowing distance to the left and right sides are 0 and 88 nm respectively. The new optimal locations for all vertical wire segments could be achieved after solving the QP problem. The right layout in Fig. 6.8(b) shows the wire segment in the affected area after the pitch adjustment by spacing technique, where only the wire segment 1 moves to the right by the distance 88 nm with the via together. Thus, the forbidden pitch is avoided since its size is changed from 455 nm to 543 nm , which is out of the “forbidden range I”, 401 nm - 511 nm . Fig. 6.8(c) is the full layout after the pitch adjustment.



(a) A forbidden pitch example



(b) Pitch adjustment by spacing technique



(c) Layout after avoiding forbidden pitch

Fig. 6.8: An example of forbidden pitch avoidance by spacing technique.

6.4 Summary

In this chapter, spacing technique implemented in the iterative pitch adjustment stage of FnRouter has been introduced. The difference between spacing technique and plowing operation has been presented. Finally, a layout example has been provided to illustrate the implementation of spacing technique in FnRouter.

Chapter 7 Other Factors and Application

The principles of the proposed two-stage lithography-friendly routing method have been introduced in the previous chapters. In the following, other factors related to its implementation in FnRouter and its application to three test chips are presented.

7.1 Measures of Image Improvement

The two-stage lithography-friendly routing method is applied to avoid forbidden pitches in the layout so as to enhance the process window of images when using SRAFs in conjunction with OAI in optical lithography. Therefore, two measures are provided in the research to show its ability to improve image quality. One is based on the comparison between the original process window and that after the pitch adjustment for each forbidden pitch. The other is based on the comparison between the original process window and that after the pitch adjustment for each affected area. The first measure can be easily executed since the process window of each pitch could be achieved according to (5.1). The second measure is complex since in addition to the wire segment pair forming the forbidden pitch, there are other wire segments in the affected area. To identify the influence on the process window of each affected area due to forbidden pitch avoidance, a parameter

$$\gamma = \frac{\sum_{i=0}^{n-1} \alpha_i f(p_i)' - \sum_{i=0}^{n-1} \alpha_i f(p_i)}{\sum_{i=0}^{n-1} \alpha_i f(p_i)} \quad (7.1)$$

is defined in this research to represent the image improvement for the affected area where n pitches exist. In (7.1), $f(p_i)$ is the original process window of pitch p_i and $f(p_i)'$ is that after the pitch adjustment; α_i is the weight of pitch p_i , which is proportional to the overlapping length of the two wire segments forming pitch p_i . Therefore, the larger γ , the more image improvement in affected area. In Fig. 5.1(b), the minimum process window of the curve is about 0.55, while the maximum process window is 1. Assuming all pitches in the affected area have the minimum process window before the pitch adjustment and all achieve the maximum process window after the pitch adjustment, then the maximum γ could be 81.8%.

7.2 Wire Length Variation

In traditional routing algorithm, each routed net should satisfy the timing constraint so as to ensure that the circuit operates correctly under the control of clock signals. As the IC technology enters the submicron era, delay time of the wire that provides communication between components becomes comparable with the delay time inside the components. Therefore, the wire length of each net that determines its delay time is important to the timing constraint. However, the pitch adjustment to

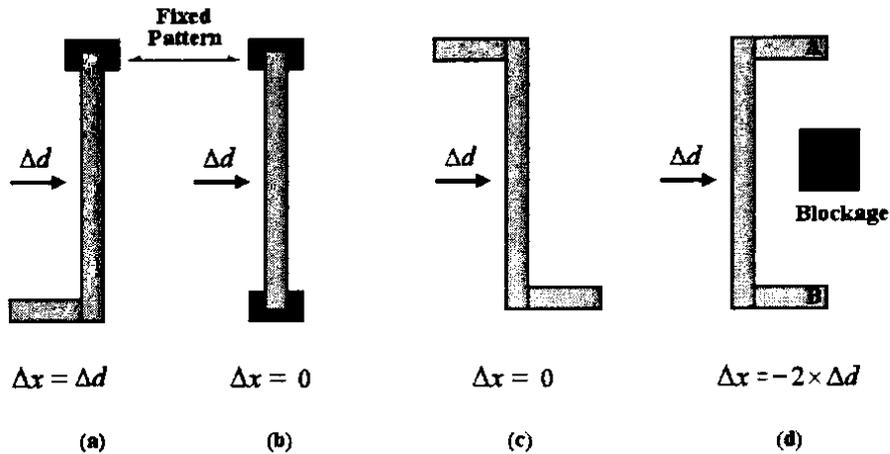


Fig. 7.1: Four types of wire length variation.

avoid forbidden pitches in the proposed two-stage lithography-friendly routing method may lead to wire length variation, therefore affects the timing of nets.

Fig. 7.1 illustrates the four types of wire length variation that correspond to the four types of wire connections in the layout. In Fig. 7.1(a), the wire connects to the fixed pattern that could be the terminal of a standard cell or the pin of the chip. When the vertical wire segment is adjusted to the right by the distance Δd , the horizontal wire segment is stretched so that the wire length variation Δx equals to Δd . In Fig. 7.1(b), the wire connects two fixed patterns. When it is adjusted to the right by the distance Δd , wire length does not change since the vertical wire segment cannot be moved out from the fixed patterns due to via constraints discussed in Chapter 5. In Fig. 7.1(c), the wire length variation Δx always equals to 0 when the vertical wire segment is adjusted since the wire length variations of the two horizontal wire segments counteract. In Fig. 7.1(d), when the vertical wire segment is adjusted by the distance Δd , the wire length variation Δx is twice of Δd . However,

such wire connection is only used to bypass blockages during routing process. It is not common in the layout since the two terminals A and B can directly connect if there is no blockage. Consequently, the wire length variations due to the pitch adjustment by the distance Δd for the vertical wire segments in Fig. 7.1 are Δd , 0, 0, and $2\Delta d$ respectively. Considering the scopes of forbidden ranges are small, the distance Δd for adjusting wire segments out of the forbidden pitches is therefore also small. For example, the scopes of forbidden range I (401 nm -511 nm) and forbidden range II (620 nm -656 nm) in this research are 110 nm and 36 nm respectively, which are only 39.3% and 12.8% of the minimum wire width (280 nm) in the design rules. Therefore, the wire length variation would be small after the forbidden pitch avoidance by pitch adjustment.

7.3 Constrained Optimization

In the two-stage lithography-friendly routing method, the pitch adjustment to avoid forbidden pitches after traditional routing is formulated as a QP problem. In FnRouter, an independent C++ program that utilizes the Matlab API function **quadprog** in the optimization toolbox is called to solve the QP problem, which communicates with the main program by Matlab files. The function **quadprog** contains several algorithms and it automatically selects the most suitable one to solve the QP problem according to the supplied constraints. In general, **quadprog** locates a solution for QP problem when supplying the constraints explicitly [Matlab98]. However, too many constraints would cause **quadprog** to fail to find a solution

within the set maximum number of iteration. In fact, a solution may not exist for the problem. For an affected area with n wire segments and m pitches that contribute to the objective function, there are $2n$ variables and $3n+2m$ constraints besides the compulsory constraints and optional constraints in the formulated QP problem. Therefore, the number of wire segments in each affected area indicates the complexity of the QP problem.

If the area where the forbidden pitch locates is routed with high-density wires, the number of wire segments affected during the pitch adjustment could be large. However, it is not necessary to consider all these affected wire segments when only adjusting the forbidden pitch formed by two wire segments. In FnRouter, boundaries are supplied to limit the number of wire segments in the affected area so as to limit the QP problem. In addition, FnRouter can narrow the boundaries so as to reduce the affected area and repeat the pitch optimization if **quadprog** does not find a solution within the set maximum number of iteration.

7.4 Algorithm Speed

In the two-stage lithography-friendly routing method, the speed of traditional routing depends on the routing algorithm adopted. In the iterative pitch adjustment stage, the most time consuming steps are building the graph to collect geometric information in the affected area, solving the QP problem, and pitch adjustment by spacing technique. Since the corner stitching data structure is adopted, the algorithm

of building the graph is simply a directed area enumeration algorithm [Ouster84a] and spacing technique is just the plowing operation. They are both fast local operations since corner stitching data structure records neighbors' information. Referred to [Ouster84a], they are both $O(N)$ operations for typical cases, where N is the number of tiles of direct interest to the algorithm (i.e., the number of tiles being enumerated in area enumeration, and the number of tiles being removed in plowing). As the core of this research is not on constrained optimization, an independent C++ program that utilizes the API function **quadprog** in Matlab optimization toolbox is called to solve the QP problem when needed. The speed of solving the QP problem is therefore not discussed here.

7.5 Application

In the research, the two-stage lithography-friendly routing method is applied to three test chips by FnRouter. In the following, the performance of this method is presented.

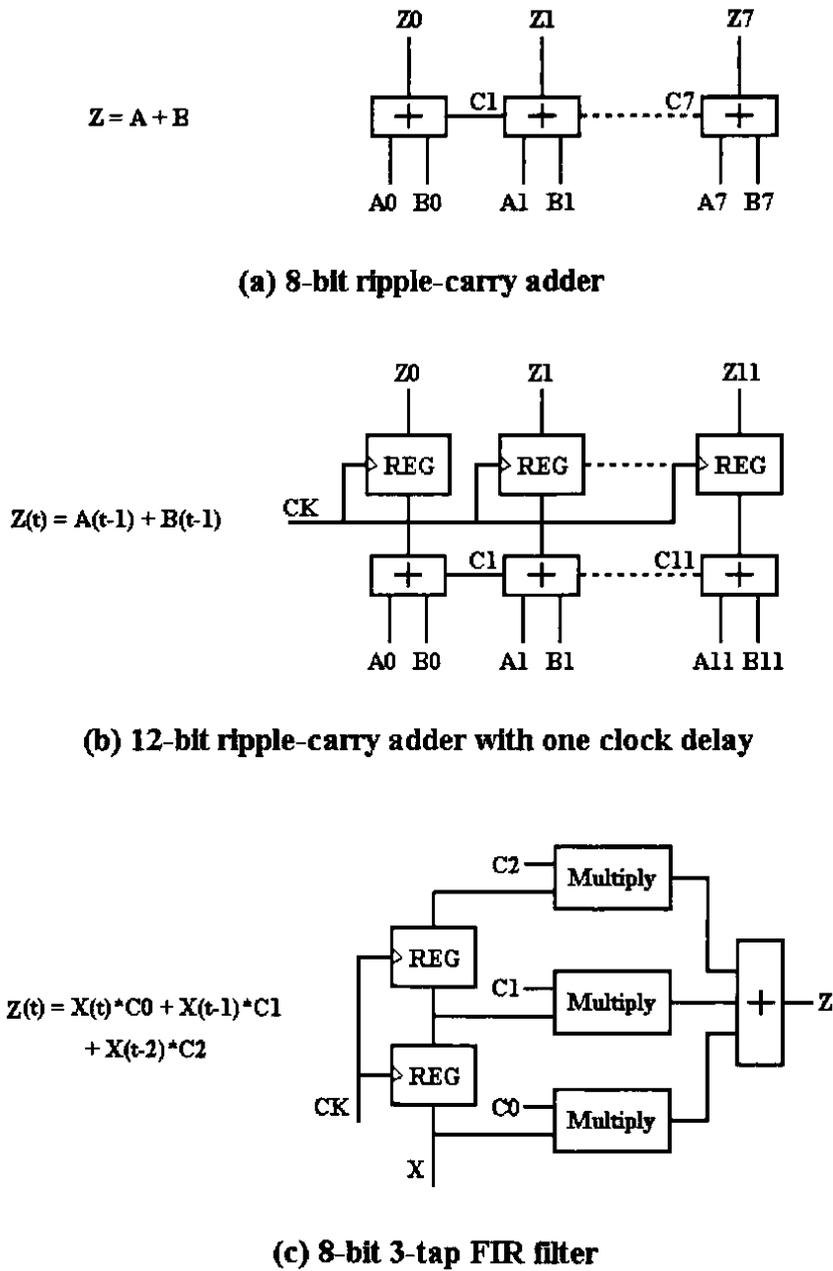


Fig.7.2: Circuit schematics of three test chips.

7.5.1 Test Chips

Fig. 7.2 presents the circuit schematics of the three test chips, and their functional specifications are as follows:

1. **8-bit ripple-carry adder:** the two input A, B, and the output Z are all 8-bit integer; and the adder is realized with the ripple-carry structure [Parhami00].
2. **12-bit ripple-carry adder with one clock delay:** the two input A, B, and the output Z are all 12-bit integer; and the adder is realized with the ripple-carry structure; the sum is stored in the 12-bit register for output in the next clock.
3. **8-bit 3-tap FIR filter:** the input X, the three coefficients C0, C1, C2, and the output Z are all 8-bit integer; and the multiplier is realized with non-Booth structure [Parhami00], the adder is realized with the carry-lookahead-select structure [Parhami00].

The behavior and function of the three test chips are described in the module compiler files, which are synthesized by Synopsys Module Compiler™ based on the TSMC 180 nm standard cell library. The floorplan and placement of the chips are accomplished by CAD tools integrated in the Cadence Encounter™ design platform. The traditional routing results of the three test chips by FnRouter are summarized in Table 7.1. Routing completion rate of the 12-bit ripple-carry adder with one clock delay and the 8-bit 3-tap FIR filter by FnRouter are only 89.9% and 89.4% respectively in Table 7.1. Actually, 100% completion rate for these two chips can be achieved by enlarging the chip size or increasing the routing planes. Limited routing resource is imposed to these two chips to achieve more dense routing result so as to

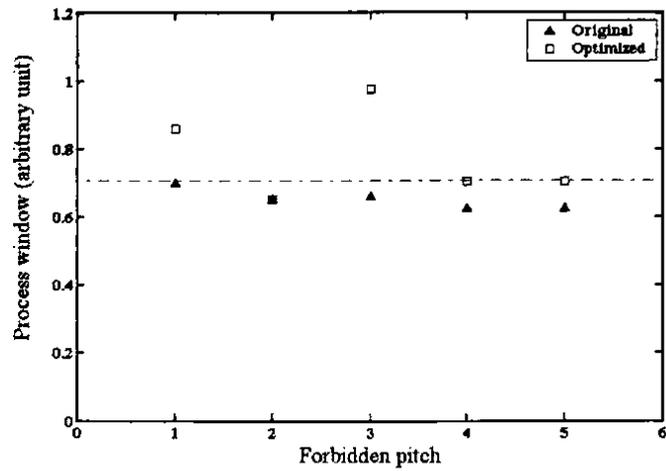
Table 7.1: Routing result of the three test chips (based on TSMC 180nm library).

Test chip	8-bit ripple-carry adder	12-bit ripple-carry adder with one clock delay	8-bit 3-tap FIR filter
Chip size (μm^2)	930	2348	2914
Standard cell number	36	68	117
Number of nets	34	79	188
Number of routing planes	3	4	4
Number of routed nets	34	71	168
Completion rate (%)	100	89.9	89.4
Number of forbidden pitches	5	16	35

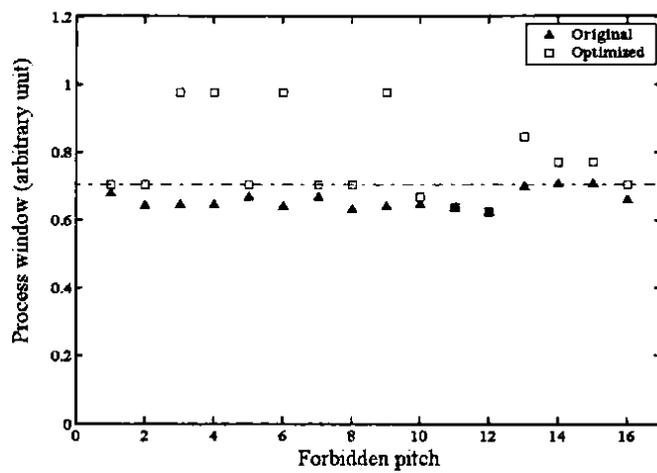
produce more forbidden pitches in the layout for lithography-friendly routing research via forbidden pitch avoidance.

7.5.2 Image Improvement for Forbidden Pitch

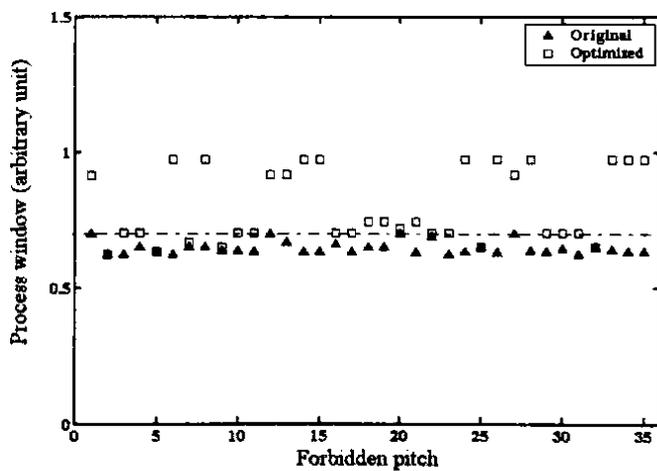
Fig. 7.3 presents the image improvement for each forbidden pitch in the three test chips. The original process window of each forbidden pitch is expressed by the triangle. Its corresponding optimized process window by pitch adjustment is expressed by the rectangle. The acceptable process window is expressed by the dash line. It is obvious that the process windows of most forbidden pitches are enhanced



(a) 8-bit ripple-carry adder



(b) 12-bit ripple-carry adder with one clock delay



(c) 8-bit 3-tap FIR filter

Fig. 7.3: Process window improvement for each forbidden pitch in (a) 8-bit ripple-carry adder, (b) 12-bit ripple-carry adder with one clock delay, (c) 8-bit 3-tap FIR filter.

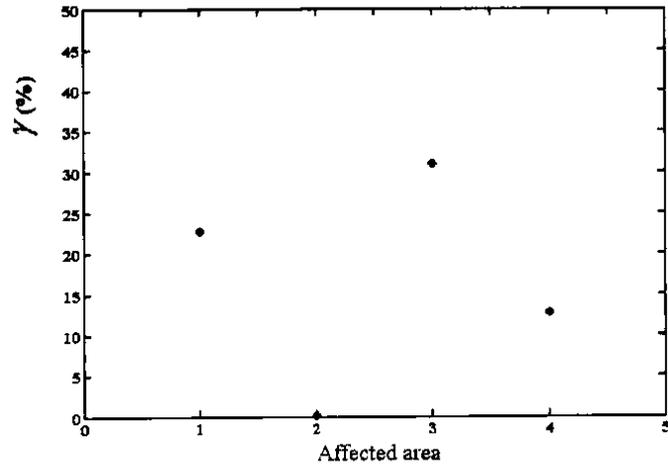
Table 7.2: Image improvement for forbidden pitches.

Test chip	Not improved	Improved but not acceptable	Improved and acceptable
8-bit ripple-carry adder	20%	0	80%
12-bit ripple-carry adder with one clock delay	12.5%	6.25%	81.25%
8-bit 3-tap FIR filter	11.43%	5.71%	82.86%

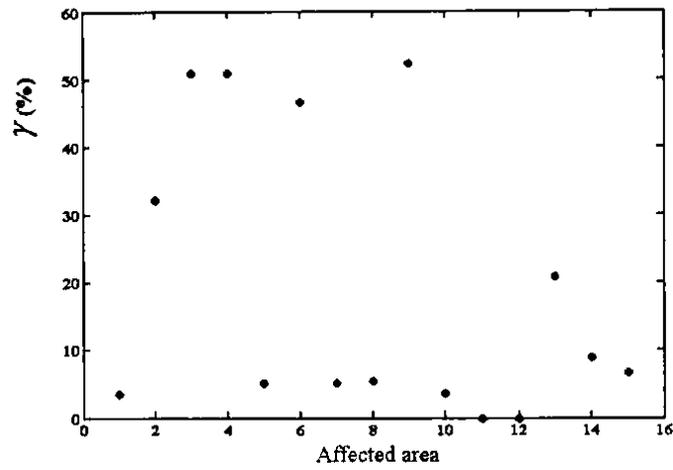
above the acceptable line, i.e. most forbidden pitches are avoided, after the pitch adjustment. However, the process windows of some forbidden pitches are not improved, or improved but still under the acceptable line. This is because these forbidden pitches have strict layout constraints so that they do not have sufficient ranges for pitch adjustment. For example, wire segments that connect to terminals of standard cells always have small ranges for pitch adjustment. From Table 7.2, it is obvious that more than 80% of forbidden pitches can be avoided by the two-stage lithography-friendly routing method.

7.5.3 Image Improvement for Affected Area

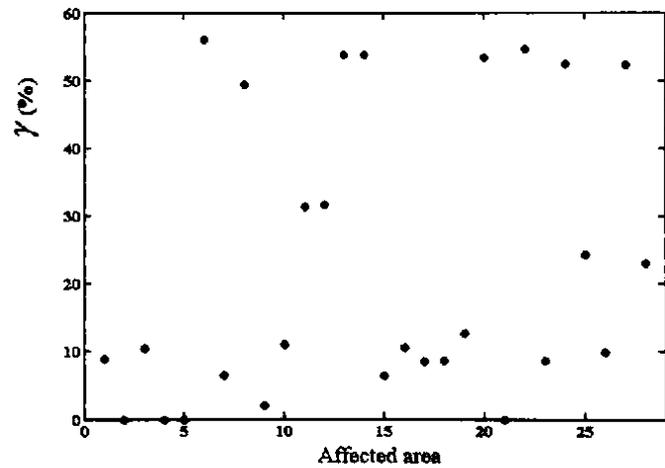
Fig. 7.4 presents the image improvement for each affected area in the three test chips. Since more than one forbidden pitches could exist in the same affected area, the number of data in Fig. 7.4 is smaller than those in Fig. 7.3. It is obvious that



(a) 8-bit ripple-carry adder



(b) 12-bit ripple-carry adder with one clock delay



(c) 8-bit 3-tap FIR filter

Fig. 7.4: Process window improvement for each affected area in (a) 8-bit ripple-carry adder, (b) 12-bit ripple-carry adder with one clock delay, (c) 8-bit 3-tap FIR filter.

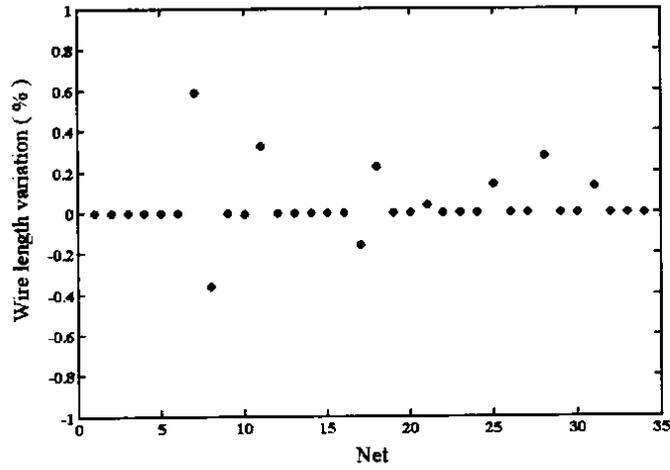
the pitch adjustment to avoid forbidden pitches does not deteriorate the image quality of each affected area. Furthermore, image quality of most affected areas are improved and γ can be more than 50%.

7.5.4 Wire Length Variation

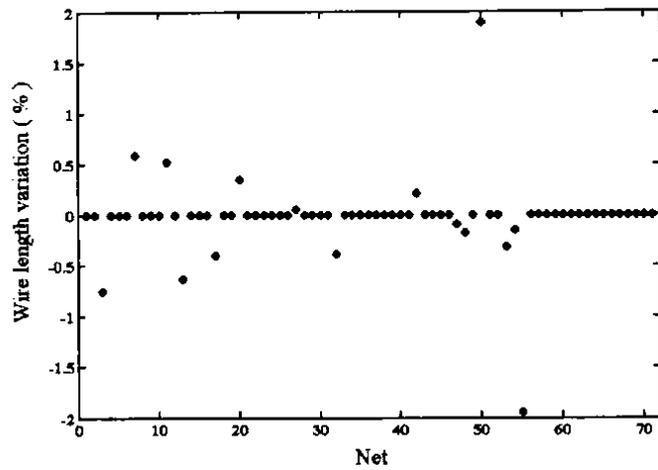
Fig. 7.5 illustrates the wire length variation of each net in the three test chips. The percentages of nets whose wire length is changed after the pitch adjustment are 26.5%, 18.3%, and 12.5% respectively. However, the wire length variation of each net is smaller than 1%, 2%, and 2.5% respectively when it compares to its original wire length. It is obvious that the results in Fig. 7.5 agree with the analysis in section 7.2. Therefore, the two-stage lithography-friendly routing method to avoid forbidden pitches by spacing technique does not lead to large wire length variation, which means that the pitch adjustment does not greatly affect the timing of nets. This is the reason why the clock routing algorithm is not implemented in FnRouter. As presented in Chapter 4, all the nets are connected by common routing algorithm with the assumption that their timing constraints are all satisfied.

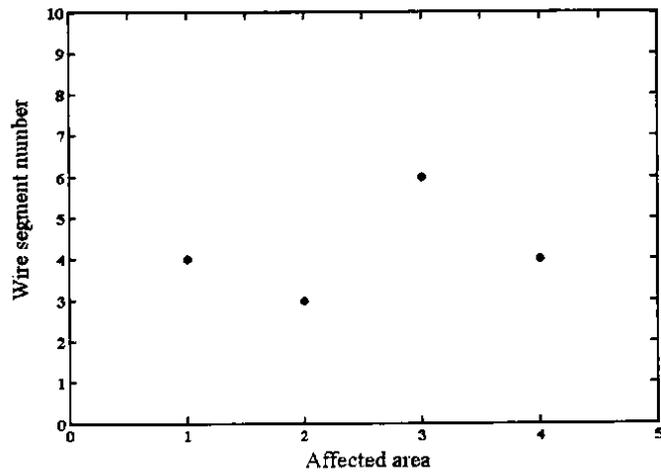
7.5.5 Constrained Optimization

Fig. 7.6 illustrates the wire segment number of each affected area in the three test chips. The values of the affected areas whose boundaries are reduced during the

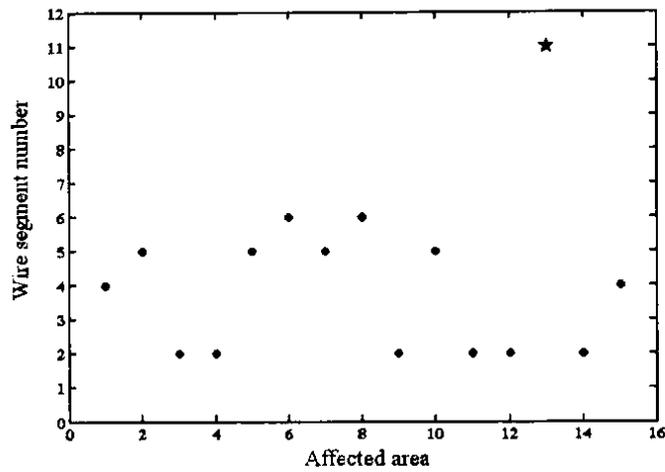


(a) 8-bit ripple-carry adder

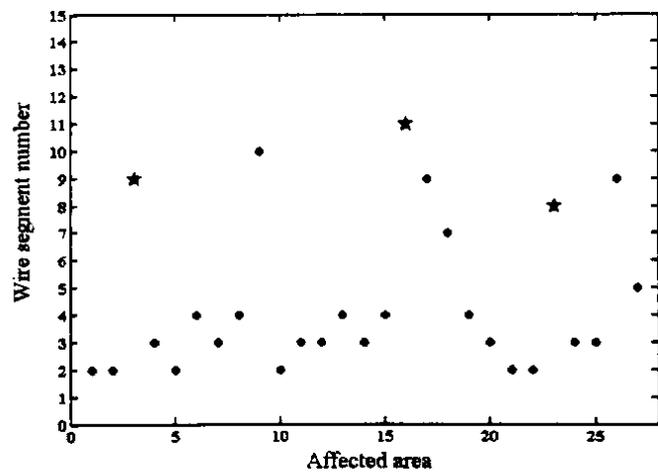




(a) 8-bit ripple-carry adder



(b) 12-bit ripple-carry adder with one clock delay



(c) 8-bit 3-tap FIR filter

Fig. 7. 6: Wire segment number of each affected area in (a) 8-bit ripple-carry adder, (b) 12-bit ripple-carry adder with one clock delay, (c) 8-bit 3-tap FIR filter.

pitch adjustment are specially marked with stars as shown in Fig. 7.6(b) and Fig. 7.6(c). It is obvious that in most of affected areas with wire segments less than 10, a solution can easily be found. The number of cases that requires boundary adjustment usually have wire segment number greater than 10. Therefore, it is recommended to control the wire segment number to less than 10 during the pitch adjustment in FnRouter.

7.6 Summary

In this chapter, other factors, including measures of image improvement, wire length variation, constrained optimization, and algorithm speed, related to the two-stage lithography-friendly routing method have been discussed. The application of the two-stage lithography-friendly routing method to three test chips including 8-bit ripple-carry adder, 12-bit ripple-carry adder with one clock delay, and 8-bit 3-tap FIR filter, have been presented. It has been shown that more than 80% forbidden pitches could be avoided with the pitch adjustment by spacing technique after the traditional routing. The image quality of most affected areas has been improved and γ can be more than 50%. It has demonstrated that the wire length variation due to the pitch adjustment is small and therefore does not greatly affect the original timing constraint of each net.

Chapter 8 Assistant Techniques

As presented in the last chapter, more than 80% forbidden pitches can be avoided by pitch adjustment after the traditional routing. However, there are still about 20% forbidden pitches that could not be avoided due to their strict layout constraints. Two assistant techniques are proposed in this chapter to tackle the unresolved forbidden pitches. One is ripping-up the nets that form the forbidden pitch and rerouting them with the forbidden pitch constraints. The other is modifying the widths of the wire segments that form the forbidden pitch so as to adjust the pitch out of the forbidden ranges.

8.1 Rip-up and Reroute

As discussed in section 3.2.2, it is not necessary to introduce new space constraints arisen from the forbidden pitch problem for each net during the routing process since it increases the algorithm complexity and tends to nonoptimal results. For example, the test chip, 8-bit ripple-carry adder, could be completely routed by H-V tile-expansion routing algorithm in the research if only the minimum pitch in design rules is considered as the space constraint. However, its completion rate decreases to 94.1% if the forbidden pitch constraints as in Fig. 3.1(b) are considered when routing each net. Therefore, two-stage lithography-friendly routing method routes the nets with the traditional routing algorithm first, then applies the pitch

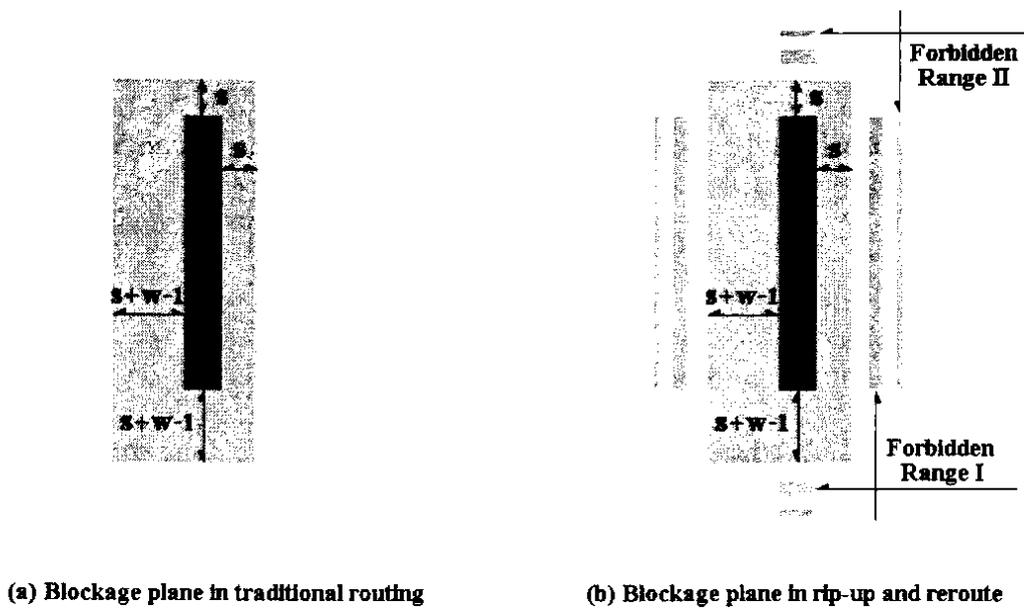


Fig. 8.1: Solid tile in the blockage planes.

adjustment to avoid forbidden pitches. However, after most of nets have been determined their locations in the layout after the traditional routing, ripping-up the nets forming the forbidden pitches and rerouting them with the forbidden pitch constraints does not greatly affect the final layout. Therefore, this approach is adopted in FnRouter as one of the assistant techniques to solve the remaining 20% forbidden pitches.

As introduced in Chapter 4, FnRouter routes the nets by H-V tile-expansion routing algorithm with the help of blockage planes to keep the design rule correct. During the routing process, each solid tile is extended in all four directions in the blockage planes as illustrated in Fig. 8.1(a) to satisfy the requirements of the minimum pitch s and the minimum width w in the design rules. However, if the forbidden pitch constraints are taken into account when constructing the blockage

planes during the rip-up and reroute phase, just as in Fig. 8.1(b) that the forbidden ranges to each solid tile are also marked as the blocked areas in blockage planes. Then the rerouted path is not only design rule correct but also avoided to form forbidden pitches with other existing patterns.

Fig. 8.2 presents an example to implement the rip-up and reroute approach to avoid the forbidden pitch, which is from the 8-bit ripple-carry adder test chip. In Fig. 8.2(a), the two wire segments form a forbidden pitch with size of 455 nm . If the left net is ripped-up and rerouted with the forbidden pitch constraint, the two net does not

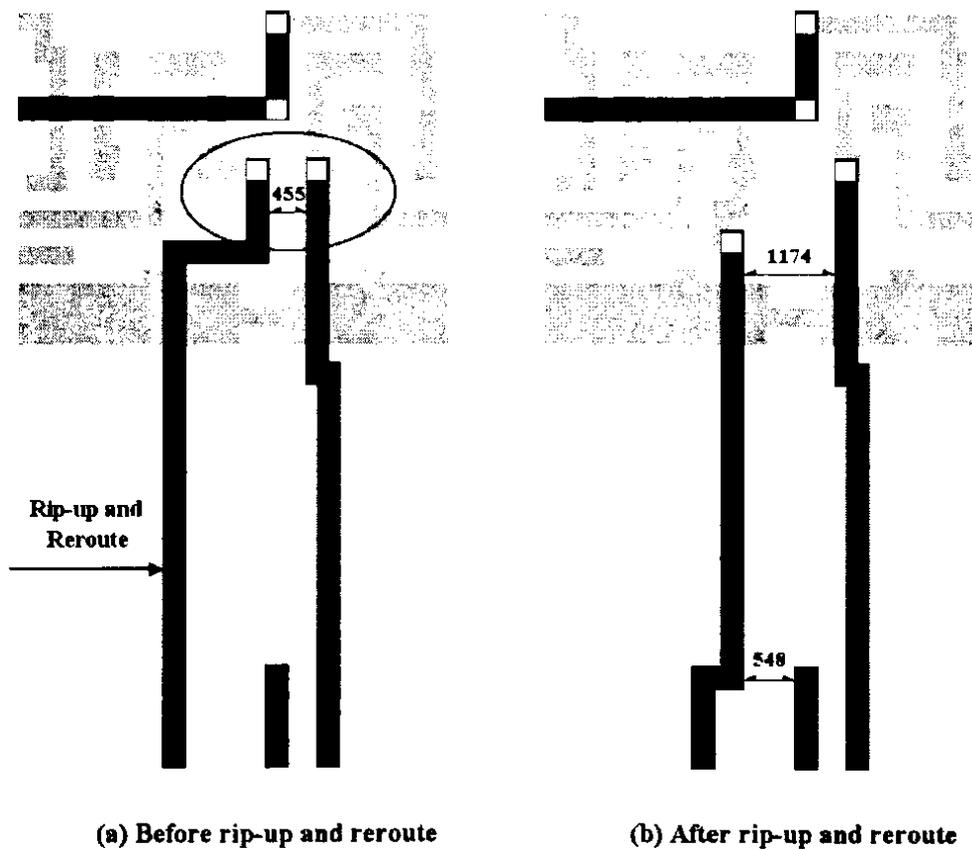


Fig. 8.2: An example by ripping-up and rerouting net to avoid forbidden pitch.

form a forbidden pitch any more, as illustrated in Fig. 8.2(b). When this assistant technique is applied to the three test chips, the percentage of the avoided forbidden pitches increases to 100%, 87.5%, and 88.6% respectively.

8.2 Wire Width Modification

Although forbidden pitches could be further avoided by the rip-up and reroute method, there may be still some forbidden pitches left since the nets ripped-up may fail to find a path with the forbidden pitch constraints during the rerouting process. Then another assistant technique, which adjusts the pitch out of the forbidden ranges by modifying the width of wire segments, could be applied as illustrated in Fig. 8.3.

Theoretically, wire width modification could avoid all the remaining forbidden pitches and is more convenient than other approaches including the

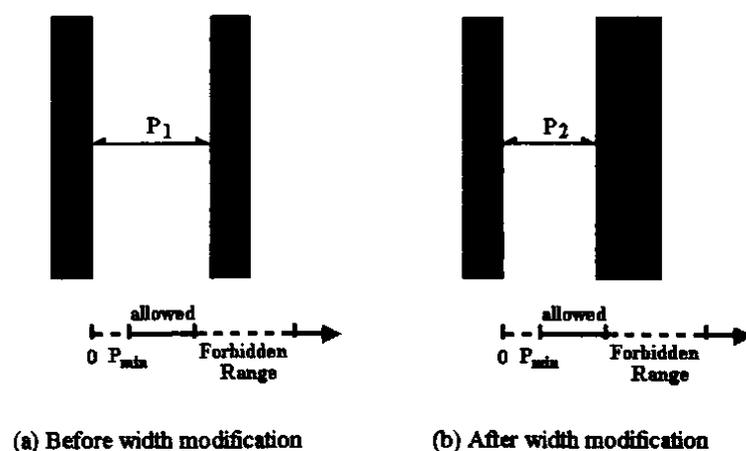


Fig. 8.3: Forbidden pitch avoidance by wire width modification.

iterative pitch adjustment. However, its drawback is that the modification of wire width leads to the variations of resistance and capacitance of the wire, which changes the original RC delay, cross talk, and IR voltage drop related to this wire.

The resistance of a wire segment can be expressed as

$$R = \frac{\rho L}{H W}, \quad (8.1)$$

where W , L , and H are the width, length, and thickness of the wire segment, ρ is the resistivity of the material. The capacitance analysis of a wire segment is complex in the submicron IC technology. In [Rabaey96], the empirical formula for the intrinsic capacitance of the wire segment is given as

$$C = \epsilon_{ox} L \left[\left(\frac{W}{t_{ox}} \right) + 0.77 + 1.06 \left(\frac{W}{t_{ox}} \right)^{0.25} + 1.06 \left(\frac{H}{t_{ox}} \right)^{0.5} \right], \quad (8.2)$$

where ϵ_{ox} and t_{ox} are the permittivity and thickness of the oxide. In addition, the nearby wire segments on the same layer and different layers contribute interwire capacitance to the wire segment, whose value is proportional to their overlapping areas and comparable with the intrinsic capacitance.

Generally, the wire width modification for avoiding forbidden pitches may be large. For example, the scopes of forbidden range I (401 nm -511 nm) and

forbidden range II (620 nm -656 nm) in this research are 110 nm and 36 nm respectively, which are 39.3% and 12.8% of the minimum wire width (280 nm) in the design rules. This means that in the worse cases, wire width may change by 39.3% and 12.8% to avoid forbidden pitches, which would lead to large variations of the resistance and the total capacitance of the wire segment. Thus the timing performance optimized on the original RC simulation model and IR drop voltage may change greatly so that the original timing constraint may be not satisfied any more. Furthermore, the wire segment after width modification may overlap more area with other wire segments in other layers. Therefore, the interference that is generally called cross talk is more serious, which deteriorates the signal integrity of the circuit.

Though wire width modification has the above shortcomings, it is still a convenient approach that could be used to avoid forbidden pitches completely. In this research, this assistant technique is only proposed but has not been implemented since the simplified routing algorithm adopted in FnRouter does not considered timing constraints and signal integrity. In the future, when developing more advance routers to realize the lithography-friendly routing in the industry, this assistant technique could be integrated in the routing algorithm and its shortcomings can be overcome by existing techniques. For example, inserting buffers in the net could control the variation of timing due to the wire width modification and inserting shielding wires could reduce the cross talk.

8.3 Summary

In this chapter, two assistant techniques to avoid the unresolved forbidden pitches have been described. One is ripping-up the nets that form the forbidden pitch and rerouting them with the forbidden pitch constraints. The other is adjusting the pitch by wire width modification.

Chapter 9 Discussions & Conclusion

Although the research in this thesis is based on the 0.248 μm optical lithography technology and the SRAFs design strategy with only two forbidden ranges, the proposed two-stage lithography-friendly routing method could be straightforwardly extended to other optical lithography technologies with different SRAFs design strategies. In practice, the two-stage lithography-friendly routing method could be parameterized by a set of rules contained in an input technology file so as to avoid dependence on a particular optical lithography with specific SRAFs design strategy.

The proposed approaches in this thesis demonstrate one of many possible approaches to realize lithography-friendly routing via forbidden pitch avoidance so as to facilitate the use of SRAFs in conjunction with OAI in optical lithography. Router designers can select their preferred data structure, database, traditional routing technique, and pitch adjustment technique to realize the two-stage lithography-friendly routing method. The designer could even directly adopt the assistant techniques discussed in Chapter 8 as the main techniques to avoid forbidden pitches if their shortcomings could be overcome. For example, if the timing variation and noise introduced by wire width modification could be controlled within acceptable tolerance, then wire width modification will be a more convenient approach since it could avoid all forbidden pitches theoretically.

9.1 Conclusion

In this thesis, the forbidden pitch problem arisen from the use of SRAFs in conjunction with OAI in optical lithography has been discussed and its challenge to the routing technique has been first investigated. A two-stage lithography-friendly routing method has been proposed, which accomplishes routing in the first stage by traditional routing techniques and post-processes the routed layout in the second stage by pitch adjustment to avoid forbidden pitches.

A two-step approach has been adopted in the proposed pitch adjustment mechanism. In the first step, the process window of wire segments forming the forbidden pitch and their neighbors that may be affected during the pitch adjustment process is formulated into a quadratic objective function with linear geometric constraints from the layout. New optimal locations for wire segments in the affected area are obtained by solving the constrained quadratic optimization problem. The spacing technique, which is an improved method of the plowing algorithm, has been developed. In the second step, each wire segment is adjusted to its new optimal location by spacing technique that keeps the original wire connectivity and design rules intact during the layout adjustment.

The two-stage lithography-friendly routing method has been adopted by **FnRouter**, a router developed in the course of this research, and applied to three test chips. It has been shown that more than 80% forbidden pitches could be avoided by

the proposed method and the process window of patterns in most affected areas has been substantially improved.

References

- [Arnold88] Michael H. Arnold, Walter S. Scott, "An interactive maze router with hints", in *Proc. of 25th ACM/IEEE conference on design automation*, 1988, pp. 672-676.
- [Bentley79] J. L. Bentley and J. H. Frieman, "A survey of algorithms and data structures for range searching", *ACM Computing Surveys*, Vol. 11, No. 4, 1979.
- [Cadence99] Cadence Inc., *IC CraftMan 5.0 Design Language Reference*, June 1999.
- [Cadence04] Cadence Inc., *Cadence SoC Encounter Datasheet*, 2004. Available at http://www.cadence.com/datasheets/socencounter_ds.pdf.
- [Camposa03] Raul Camposano, "Tighter link between manufacturing and electronic design automation becoming a necessity", *COMPILER*, Synopsys, April 2003. Available at http://www.synopsys.com/news/pubs/compiler/art1lead_dmf-apr03.html.
- [Chen86] H. H. Chen, "Trigger: a three layer gridless channel router", in *Proc. of IEEE international conference on computer-aided design, 1986*, pp. 196-199.
- [Chen86a] H. H. Chen, E. Kuh, "Glitter: a gridless variable-width channel router", *IEEE Trans. on Computer-Aided Design*, CAD-5(4), 1986, pp. 459-465.
- [Cobb97] N. Cobb and A. Zakhor, "Experimental results on optical proximity correction with variable threshold resist model", *Proc. SPIE*, Vol. 3051, 1997, pp. 458-468.
- [Dees82] William A. Dees, Jr. and Patrick G. Karger, "Automated rip-up and reroute techniques", in *Proc. of the 19th design automation conference*, 1982, pp. 432-439.
- [Dijkstra59] E. W. Dijkstra, "A note on two problems in connexion with graphs", *Numerische Mathematik*, Vol. 1, 1959, pp. 269-271.
- [Garofalo93] J. Garofalo, C. Biddick, R. Kostelak, and S. Vaidya, "Mask assisted off axis illumination technique for random logic", *J. Vac. Sci. Technol. B*, Vol. 11, No. 6, Nov. 1993, pp. 2651-2658.

- [Grobm01] W. Grobman, M. Thompson, R. Wang, C. Yuan, R. Tian, E. Demircan, "Reticle enhancement technology: implications and challenges for physical design", in *Proc. of the 38th conference on design automation*, June 2001, pp. 73-78.
- [Grobm01a] Warren Grobman, Robert Boone, Gece Philbin, Bob Jarvis, "Reticle enhancement technology trends: resource and manufacturability implications for the implementation of physical designs", in *Proc. of the 2001 symposium on physical design*, April 2001, pp. 45-51.
- [Hadlock75] F. Hadlock, "Finding a maximum cut of a planar graph in polynomial time", *SIAM Journal of Computing*, 4, No. 3, Sept. 1975, pp. 221-225.
- [Hashi71] A. Hashimoto and J. Stevens, "Wire routing by optimization channel assignment within large apertures", in *Proc. of the 8th design automation workshop*, 1971, pp. 155-163.
- [Heyns80] W. Heyns, W. Sansen, H. Beke, "A line-expansion algorithm for the general routing problem with a guaranteed solution", in *Proc. of the 17th ACM/IEEE conference on design automation*, 1980, pp. 243-249.
- [High69] D. W. Hightower, "A solution to the line routing problem on a continuous plane", in *Proc. 6th design automation workshop*, 1969.
- [Hsueh79] M. Y. Hsueh, "Symbolic layout and compaction of integration circuits", *University of California, Berkeley, Tech. Report UCM/ERL/M79/80*, Dec. 1979.
- [Intel04] Intel, Inc., see <http://www.intel.com/research/silicon/mooreslaw.htm>.
- [ITRS03] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors 2003 Edition*. Available at <http://public.itrs.net/Files/2003ITRS/Home2003.htm>.
- [Jackson90] M. A. B. Jackson, A. Sirinivasan, E. S. Kuh, "Clock routing for high-performances ICs", in *Proc. of 27th ACM/IEEE design automation conference*, June 1990, pp. 573-579.
- [Kahng93] A. Kahng, J. Cong, G. Robins, "Matching based models for high performance clock routing", *IEEE Trans. on CAD of Integrated Circuits and Systems*, August 1993, pp. 1157-1169.
- [Kahng99] Andrew B. Kahng, Y. C. Pati, "Subwavelength lithography and its potential impact on design and EDA", in *Proc. of the 36th ACM/IEEE conference on design automation*, June 1999, pp. 799-804.

- [Kahng99a] A. B. Kahng, Y. C. Pati, "Subwavelength optical lithography: challenges and impact on physical design", in *Proc. of the 1999 international symposium on physical design*, April 1999, pp. 112-119.
- [Kahng00] Andrew B. Kahng, "VLSI physical design algorithms for phase-shifting technology", *Final report 1999-2000 for MICRO project 99-058*, 2000, Available at http://www.ucop.edu/research/micro/99_00/99_058.pdf.
- [Kamon91] K. Kamon, T. Miyamoto, Y. Myoi, H. Nagata, M. Tanaka, and K. Horie, "Photolithography system using annular illumination", *Jpn. J. Appl. Phys.*, Vol. 30, No. 11B, Nov. 1991, pp. 3021-3029.
- [Lavin02] Mark Lavin, Lars Liebmann, "CAD computation for manufacturability: can we save VLSI technology from itself", in *Proc. of 2002 IEEE/ACM international conference on computer aided design*, Nov. 2002, pp. 424-428.
- [Lawrence97] Craig Lawrence, Jian L. Zhou, Andre L. Tits, *User's Guide for CFSQP Version 2.5: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints*, Electrical Engineering Department and Institute for Systems Research, University of Maryland, USA, April 1997.
- [Lee61] C. Y. Lee, "An algorithm for path connections and its applications", *IRE Transactions on Electronic Computers*, 1961.
- [Lerner02] Reuven M. Lerner, *Core Perl*, Prentice Hall PTR, 2002.
- [Levenson82] Marc Levenson, N. Viswanathan, and R. Simpson, "Improving resolution in photolithography with a phase-shifting mask", *IEEE Transactions on Electron Devices*, Vol. 29, No. 12, Dec. 1982, pp. 1812-1846.
- [Liebma01] L. W. Liebmann, S. M. Mansfield, Alfred K. Wong, M. A. Lavin, W. C. Leipold, T. G. Dunham, "TCAD development for lithography resolution enhancement", *IBM Journal of Research and Development*, Vol. 45, No. 5, Sept. 2001, pp. 651-665.
- [Liebma03] Lars W. Liebmann, Juan-Antonio Carballo, "Layout methodology impact of resolution enhancement techniques", *Electronic Design Processes Workshop*, April 2003.
- [Liebma03a] Lars W. Liebmann, "Layout impact of resolution enhancement techniques: impediment or opportunity? ", in *Proc. of the 2003 international symposium on physical design*, April 2003, pp. 110-117.

- [Liebma03b] L. Liebmann, et al., "Layout optimization at the pinnacle of optical lithography", in *Proc. of SPIE*, Vol. 5042, 2003.
- [Lin89] Y. L. Lin, Y. C. Hsu, F. S. Tsai, "Silk: a simulated evolution router", *IEEE Trans. on Computer-Aided Design*, October 1989, pp. 1108-1114.
- [Liu98] H. Y. Liu, L. Karklin, Y. T. Wang, and Y. C. Pati, "The application of alternating phase-shifting masks to 140nm gate patterning: Line width control improvements and design optimization", in *Proc. SPIE*, Vol. 3236, 1998, pp. 328-337.
- [Luk85] W. K. Luk, "A greedy switchbox router", *Integration, The VLSI Journal*, No. 3, 1985, pp. 129-149.
- [Lurs87] C. Lursinsap, D. Gajski, "An optimal power routing for top-down design architecture", in *Proc. of the international conference on computer design*, 1987, pp. 345-348.
- [Mansfield00] Scott M. Mansfield, Lars W. Liebmann, Antoinette F. Molless, Alfred K. Wong, "Lithographic comparison of assist feature design strategies", in *Proc. of SPIE*, Vol. 4000, 2000, pp. 63-76.
- [Margari87] A. Margarino, A. Romano, A. DE Gloria, F. Curatelli, P. Antognetti, "A tile-expansion router", *IEEE Trans. on Computer-Aided Design*, Vol. CAD-6, No. 4, July 1987, pp. 507-517.
- [Matlab98] MathWorks, Inc., *Standard Algorithms (Optimization Toolbox)*, 1998.
- [McCull04] Kevin McCullen, "Phase correct routing for alternating phase shift masks", in *Proc. of the 41st design automation conference*, 2004, pp. 317-320.
- [Mikami68] K. Mikami and K. Tabuchi, "A computer program for optimal routing of printed circuit connectors", *IFIPS Proc.*, H47, 1968, pp. 1475-1478.
- [Moore59] E. F. Moore, "Shortest path through a maze", *Annals of the Computation Laboratory of Harvard University*, Harvard Univ. Press, Cambridge, Mass., Vol. 30, 1959, pp. 285-292.
- [Nocedal99] Jorge Nocedal, Stephen J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.
- [Nyhoff99] Larry R. Nyhoff, *C++, An Introduction to Data Structures*, Prentice Hall, 1999.

- [Otto96] O. Otto, R. Henderson, "Advances in process matching for rules-based optical proximity correction", in *Proc. SPIE*, Vol. 2884, 1996, pp. 425-434.
- [Ouster81] John K. Ousterhout, "Caesar: an interactive editor for VLSI", *VLSI Design*, Vol. II, No. 4, pp. 34-38, fourth quarter 1981.
- [Ouster84] John K. Ousterhout, Gordon T. Hamachi, Robert N. Mayo, Walter S. Scott, and George S. Taylor, "Magic: a VLSI layout system", in *Proc. of 21st ACM/IEEE design automation conference*, 1984, pp. 152-159.
- [Ouster84a] John K. Ousterhout, "Corner stitching: a data-structuring technique for VLSI layout tools", *IEEE Trans. on Computer-Aided Design*, Vol. CAD-3, No. 1, Jan. 1984, pp. 87-100.
- [Ouster94] John K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley Publishing Company, 1994.
- [Pardalos02] Panos M. Pardalos, Mauricio G. G. Resende, *Handbook of Applied Optimization*, Oxford University Press, 2002.
- [Parhami00] Behrooz Parhami, *Computer Arithmetic Algorithms and Hardware Designs*, Oxford University Press, 2000.
- [Prolific03] Prolific Inc., "Prolific tackles design for manufacture", Feb. 2003. Available at http://www.prolificinc.com/pr/progenesis_dfm.html.
- [Prouty84] M. D. Prouty, A. R. Neureuther, "Optical imaging with phase shift masks", in *Proc. of SPIE*, Vol. 470, 1984, pp. 228-232.
- [Rabaey96] Jan M. Rabaey, *Digital Integrated Circuits, A Design Perspective*, Prentice Hall, 1996.
- [Reynolds86] G. O. Reynolds, "A concept for a high resolution optical lithographic system for producing one-half micron linewidths", in *Proc. of SPIE*, Vol. 633, 1986, pp. 228-238.
- [Rieger01] Michael L. Rieger, Jeffrey P. Mayhew, Sridhar Panchapakesan, "Layout design methodologies for sub-wavelength manufacturing", in *Proc. of the 38th ACM/IEEE conference on design automation*, June 2001, pp. 85-88.
- [Sait99] Sadiq M. Sait, Habib Youssef, *VLSI Physical Design Automation, Theory and Practice*, World Scientific, 1999.
- [Schellen99] F. M. Schellenberg, "Design for manufacturing in the semiconductor industry: the litho/design workshops", in *Proc. of the 12th*

- international conference on VLSI design*, IEEE Computer Society Press, Los Alamitos, CA, 1999, pp. 111-119.
- [Schellen01] Franklin M. Schellengerg, Luigi Capodieci, "Impact of RET on physical layouts", in *Proc. of the 2001 international symposium on physical design*, April 2001, pp. 52-55.
- [Schellen01a] F. M. Schellenberg, Olivier Toublan, Luigi Capodieci, Bob Socha, "Adoption of OPC and impact on design and layout", in *Proc. of the 38th conference on design automation*, June 2001, pp. 89-92.
- [Scott84] Walter S. Scott, John K. Ousterhout, "Plowing: interactive stretching and compaction in Magic", in *Proc. of 21st ACM/IEEE design automation conference*, 1984, pp. 166-172.
- [Sherwani95] Naveed A. Sherwani, Siddharth Bhingarde, Anand Panyam, *Routing in the Third Dimension*, IEEE PRESS, 1995.
- [Sherwani99] Naveed Sherwani, *Algorithms for VLSI Physical Design Automation*, Third Edition, Kluwer Academic Publishers Group, 1999.
- [Socha00] Robert Socha, Mircea Dusa, Luigi Capodieci, Jo Finders, Fung Chen, Donis Flagello, Kevin Cummings, "Forbidden pitches for 130nm lithography and below", in *Proc. of SPIE*, Vol. 4000, 2000, pp. 1140-1155.
- [Soukup78] J. Soukup, "Fast maze router", in *Proc. of 15th design automation conference*, 1978, pp. 100-102.
- [Strunk93] T. Strunk, N. holmes, "Victor: an almost channel-less three-layer over-the-cell router", in *Proc. of Third Great Lake symposium on VLSI*, 1993.
- [Synopsys04] Synopsys Inc., *Module Compiler Datasheet*, 2004. Available at http://www.synopsys.com/products/datapath/module_comp_ds.html.
- [Tsai92] Chia Chun Tsai, Sao Jie Chen, Wu Shiung Feng, "An H-V alternating router", *IEEE Trans. on Computer-Aided Design*, Vol. 11, No. 8, Aug. 1992, pp. 976-990.
- [Wang03] Jun Wang, Alfred K. Wong, "Effects of grid-placed contacts on circuit performance", in *Proc. of SPIE*, Vol. 5043, 2003, pp. 134-141.
- [Wilcox98] R. Wilcox, T. Forhan, G Starkey, D. Turner, "Design for manufacturability: a key to semiconductor manufacturing excellence", *IEEE/SEMI advanced semiconductor manufacturing conference and workshop*, Sept. 1998, pp. 308-313.

-
- [Wong01] Alfred K. Wong, *Resolution Enhancement Techniques in Optical Lithography*, SPIE press, 2001.
- [Wong03] Alfred K. Wong, "Microlithography: trends, challenges, solutions, and their impact on design", *IEEE Micro*, Vol. 23, No. 2, March 2003, pp. 12-21.
- [Xiong86] X. M. Xiong, E. S. Kuh, "The scan line approach to power and ground routing", in *Proc. of IEEE International conference on computer-aided design*, Nov. 1986, pp. 6-10.
- [Zhang98] Xiong Zhang, Yinyu Ye, *User's guide of COPL_QP, Computational Optimization Program Library: Convex Quadratic Programming*, Computational Optimization Laboratory, Department of Management Sciences, the University of Iowa, USA, July 1998.