Optimal Layout Synthesis of Standard Cells in Large Scale Integration

(大規模集積回路におけるスタンダードセルレイアウトの最適自動合成手法)

A Dissertation
Submitted to the Department of
Electronic Engineering,
the University of Tokyo
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Supervisor: Professor Kunihiro Asada

Tetsuya Iizuka

DEPARTMENT OF ELECTRONIC ENGINEERING,
THE UNIVERSITY OF TOKYO

December 2006

Copyright © 2006 by Tetsuya Iizuka, All Rights Reserved.

Abstract

This thesis focuses on the optimization methods for standard cell layouts. Standard cells are the most fundamental components of VLSI, and provide the building blocks for creating large complex functions in both application-specific and semi-custom domains. Therefore, their performance has significant effects on the final performance of the synthesized VLSI. We propose a minimum-width transistor placement and an intra-cell routing via Boolean satisfiability to optimize the area of the cell layouts. We also propose a comprehensive cell layout synthesis method and a cell layout de-compaction method for yield optimization.

Chapter 2 proposes a minimum-width layout synthesis method for dual CMOS cells via Boolean Satisfiability (SAT). Cell layout synthesis problems, *i.e.*, the transistor placement and the intra-cell routing problems are first transformed into SAT problems by this formulation. The proposed method guarantees to generate minimum-width cell layouts with routability under our layout styles. This method places complementary P and N type transistors individually during transistor placement, and can generate smaller width layout compared with the case of pairing the complementary P and N type transistors. The experimental results show that the proposed method generates the cell layouts of 30 dual CMOS logic circuits in 58% runtime with only 5% area increase compared with the commercial cell generation tool with cell layout compaction. This result indicates that our cell layout styles defined for the SAT formulation is practical enough to generate the layout of dual CMOS cells quickly with a little area overhead. Since this method still has a restriction in gate connection style between P and N type transistors, it is applicable only to dual CMOS cells. The extension of the transistor placement method for non-dual cells is explained in Chapter 4.

Chapter 3 describes a hierarchical extension of the cell layout synthesis method proposed in Chapter 2 for the cell layout synthesis of large dual CMOS cells. This method partitions a given transistor-level netlist into blocks considering the transistor connections by diffusions. Intra-block placement uses the exact transistor placement method proposed in Chapter 2, and hierarchically generates the transistor placement with routability. The comparison results with the flat cell layout synthesis method show that the proposed hierarchical method reduces the runtime for cell layout synthesis drastically with little width increase. The comparison

results with the commercial cell generation tool without cell layout compaction show that the total cell width is increased about 4% by the proposed method due to the layout style restriction, whereas the runtime is only about 3% of that of the commercial tool. These results show the effectiveness of the proposed method as a quick layout generator in the area of transistor-level circuit optimization such as on-demand cell layout synthesis.

Chapter 4 shows flat and hierarchical approaches for generating a minimum-width transistor placement of CMOS cells in presence of non-dual P and N type transistors, whereas the cell layout synthesis methods proposed in the previous chapters are only for dual cells. This chapter targets the minimum-width transistor placement, and does not take the intra-cell routings into consideration. Our approaches are the first exact transistor placement methods which can be applied to CMOS cells with any types of structure, whereas almost all of the conventional exact transistor placement method is applicable only to dual CMOS cells. Experimental results show that the proposed method is not only applicable to CMOS cells with any types of structure, but also more effective even for dual CMOS cells compared with the transistor placement method proposed in Chapter 2. The hierarchical single-row approach is shown to be effective to reduce the runtime drastically. This chapter also shows the generalization results of the single-row transistor placement method into the multi-row placement. The proposed exact minimum-width multi-row transistor placement method generates more area-efficient placement than the conventional method only for dual cells by using the gate connection style which is more suitable for multi-row transistor placement than the conventional style and can solve the cells with up to 26 transistors in reasonable runtime.

Chapter 5 introduces a cell layout synthesis technique to optimize the yield. The yield cost metric used in this chapter is the sensitivity to wiring faults due to spot defects. The sensitivity to faults on intra-cell routings is modeled with consideration to the spot defects size distribution and the end effect of critical areas. The impact of the sensitivity reduction on the yield improvement is also discussed in this chapter. The minimum-width cell layout of CMOS logic cells are comprehensively generated using the transistor placement method proposed in Chapter 2 and the comprehensive intra-cell routing method proposed in this chapter. The yield optimal layouts are selected from the exhaustively-generated layouts by using the proposed sensitivity to wiring faults as a cost function. The experimental results on 8 CMOS logic circuits which have up to 14 transistors show that the fault sensitivity is reduced about 15% on an average by selecting the minimum-sensitivity layouts rather than selecting the minimum-wire-length layouts.

Chapter 6 proposes a timing-aware cell layout de-compaction method for yield optimization using Linear Programming (LP). The proposed method performs a de-compaction of the original layout in order to improve the yield by minimizing the Critical Area (CA) inside the cell. This yield improvement procedure is executed under given timing constraints. To formulate the timing constraints into LP, a new accurate linear delay model which approximates the difference from the original delay is proposed. Using the proposed timing-aware yield enhancement method, we can explore the trade-off between yield and performance, and can pick up the yield/performance variants from the trade-off curve. The effectiveness of the proposed method for OPC mask data volume reduction is also shown in this chapter. The experimental results on 90nm cell layouts demonstrate an average of 4.28% reduction in the fractured mask data size in the case that 10% delay increase is allowed. This timing-aware de-compaction framework is extended to the redundant contact insertion adjacent to the original single contacts to minimize the yield loss due to contact failure. To take the parametric yield into account, the proposed method is also extended to the gate layout pattern regularity enhancement to reduce the systematic variation of the gate critical dimensions (CD). The edge placement error (EPE) estimation results show that the standard deviation of the gate CD EPE distribution is reduced by about 28% compared to that of the original layouts.

We are sure that these results in this thesis such as the exact minimum-width cell layout synthesis techniques, the comprehensive cell layout synthesis method, and the cell layout de-compaction method for yield optimization will be used for standard-cell layout optimization in terms of area, delay, and yield, and contribute to the VLSI performance and reliability improvements.

Acknowledgments

During my years in the University of Tokyo, I have been very fortunate to be surrounded by teachers, family, friends, and colleagues who have continually offered me support and encouragement. I would like to thank, from the bottom of my heart, for their contributions to my professional and personal growth.

With the utmost gratitude, I would like to thank the dissertation supervisor, Prof. Kunihiro Asada for his keen insight, enduring guidance, and encouragement throughout my undergraduate and graduate studies. His wisdom and knowledge not only in the field of my research have greatly expanded my interest in and enriched my knowledge of VLSI and philosophy about research, and his constant support, fruitful discussion, and many great opportunities he gave me since my undergraduate years led me to become a full-fledged member of society. I feel very fortunate to have taken him as my supervisor, and I will forever cherish this experience as my lifetime guidance.

I would also like to deeply thank Prof. Makoto Ikeda for his meaningful advices and discussions on my research. I greatly appreciate his constructive support and the time he was willing to spend for providing a comfortable environment to promote my research progress.

I am deeply grateful to Mr. Hiroaki Yoshida, who is a research colleague in Asada-Ikeda laboratory, for his professional experience and extensive knowledge. His diligent support in the start-up phase of my undergraduate research made me take a significant step toward my life as a researcher. He also provided a friendly atmosphere and a plenty of relaxed time in the laboratory. The time spent with him talking about both professional and private topics is an essential part of my laboratory life. I could never bring my research to be successful without his direct support and kind advices.

I would like to acknowledge Dr. Yusuke Oike, who is currently in Sony Corp., for his valuable advices originated in his exceptional research talent, and indispensable encouragement during my years in the laboratory. I decided to proceed to Ph.D. course and worked hard on trying to follow him. The inherited mental attitude and enthusiasm toward research activities were fundamental for the success of my research and will be invaluable assets in my professional career.

I would like to thank Dr. Toru Nakura, who is currently in NEC Corp., for his encouragement and thoughtful advices originated in his professional experience and self-confidence in being mature engineer. He gave me many technical suggestions on my research as well as frank advices on both professional and private topics. He has been a senior adviser of my life.

I am grateful to all the current and past members in Asada-Ikeda laboratory for their helpful advices, heartful encouragement, comfortable research circumstances and pleasant time: in particular, Prof. Toru Ishihara, who is currently in Kyushu University, for his great deal of support in the start-up phase of my research; Dr. Hiroaki Yamaoka, who is currently in Toshiba Corp., for his friendly talk and many essential advices on my laboratory life; Dr. Satoshi Komatsu, for his generous support and advices on my research and many other activities in the laboratory; Mr. Routong Zheng, for his assistance for establishing CAD environment which is essential to my research; Dr. Masahiro Sasaki, for his advices on my research as well as other chip design activities; Mr. Yusuke Yachide and Mr. Taisuke Kazama, for having many discussions as well as a plenty of pleasant time with me; Ms. Noriko Yokochi, Ms. Naomi Yoshida, and Ms. Yukako Maruyama, for their helpful assistance for my research activities in the laboratory.

I would like to acknowledge my dissertation committee, Prof. Tadashi Shibata, Prof. Masahiro Fujita, Prof. Minoru Fujishima, and Prof. Makoto Takamiya, for their extremely valuable suggestions and comments.

I wish to thank all the members of VLSI Design and Education Center (VDEC), the University of Tokyo, for their support in CAD environment and VLSI process technologies. The CAD tools and process technologies used in this study have been provided by Synopsys, Inc., Cadence Design Systems, Inc., Mentor Graphics, Corp., Rohm Corp., and Semiconductor Technology Academic Research Center (STARC) through VDEC.

I also wish to thank the Grant-in-Aid for Scientific Research of the Japan Society for the Promotion of Science (JSPS) for the financial support.

I would like to give my thanks to all of my friends, especially in Wimbledon Tennis Team of the University of Tokyo who have been giving me a plenty of pleasant and relaxed time, and encouragement for a long time of my life in the University of Tokyo.

Finally, I would like to express my greatest appreciation to my parents, Katsumi and Makiko, and to my grandparents, Keiichi and Michi, for their constant support and encouragement throughout my life.

Contents

Abstract i

Acknowledgments iv

List of Figures ix

List of Tables xiii

Chapter 1 Introduction 1

1.1 Background 1

1.2 Research Objectives and Thesis Organization 3

Chapter 2 Exact Minimum-Width Cell Layout Synthesis for Dual CMOS Cells 6

2.1 Introduction 6

2.2 Layout Styles 7

2.3 Transistor Placement Formulation 9

2.4 Intra-Cell Routing Formulation 12

2.5 Experimental Results 16

2.5.1 Comparison with the 0-1 ILP Formulation 16

2.5.2 Comparison with the Commercial Cell Generation Tool 18

2.6 Summary 22

Chapter 3 Hierarchical Extension for Large Cell Layout Synthesis 23

3.1 Introduction 23

3.2 Layout Styles 23

3.3 Hierarchical Transistor Placement 25

3.3.1 Partitioning into Logic Blocks 26

3.3.2 Intra-Block Transistor Placement Formulation 26

3.3.3 Inter-Block Placement Formulation 30

3.4 Intra-Cell Routing 31

3.5 Overall Flow 31
3.6 Experimental Results 32
3.6.1 Transistor Placement 33
3.6.2 Cell Layout Synthesis 33
3.7 Summary 35

Chapter 4 Exact Minimum-Width Transistor Placement for General CMOS Cells 36
4.1 Introduction 36
4.2 Problem Definition 39
4.3 Flat Single-Row Transistor Placement 40
4.4 Hierarchical Single-Row Transistor Placement 42
4.5 Generalization to Multi-Row Transistor Placement 46
4.5.1 Problem Definition 46
4.5.2 Placement Formulation 46
4.6 Experimental Results 50
4.6.1 Single-Row Flat Approach 50
4.6.2 Single-Row Hierarchical Approach 54
4.6.3 Multi-Row Placement 56
4.7 Summary 58

Chapter 5 Yield-Optimal Cell Layout Synthesis for CMOS Logic Cells 60
5.1 Introduction 60
5.2 Wiring Fault Model for Yield Cost Function 61
5.3 Layout Styles 64
5.4 Comprehensive Cell Layout Synthesis 65
5.4.1 Transistor Placement 65
5.4.2 Intra-Cell Routing 65
5.5 Overall System 69
5.6 Experimental Results 72
5.7 Discussion of Yield Cost Metrics 75
5.7.1 Lithography Impact on the Critical Area 75
5.7.2 Relation Between Sensitivity and Performance 77
5.7.3 Relation Between Sensitivity and Yield 77
5.8 Summary 79

Chapter 6 Yield Optimization by Timing-Aware Cell Layout De-Compaction 80
6.1 Introduction 80
6.2 Problem Definition and Design Rule Constraints 82
6.3 Yield Cost Metrics 83
6.3.1 Critical Area Minimization 83
6.3.2 OPC Relaxation 86
6.3.3 Redundant Contact Insertion 88
6.3.4 Gate Layout Pattern Regularity Enhancement 90
6.4 Delay Model 93
6.5 Overall Flow 99
6.6 Experimental Results 100
6.6.1 Critical Area Minimization 100
6.6.2 OPC Relaxation 106
6.6.3 Redundant Contact Insertion 110
6.6.4 Gate Layout Pattern Regularity Enhancement 112
6.7 Summary 119

Chapter 7 Conclusions 120

Bibliography 123

List of Publications 129

List of Figures

1.1 A simplified flow diagram of the cell-based LSI design. 2
1.2 A flow diagram of the yield-aware cell-based LSI design. 3
1.3 The proposed cell layout optimization methods in perspective. 4

2.1 An illustration of the layout styles of the proposed SAT-based cell layout synthesis for dual CMOS cells. 9
2.2 A SAT formulation of the transistor placement for dual CMOS cells. 10
2.3 A SAT formulation of the intra-cell routing. 12
2.4 Our SAT-based cell layout synthesis flow. 15
2.5 The example result of fad1 cell layout generated by the proposed SAT-based cell layout synthesis method for dual CMOS cells. 21

3.1 An illustration of the layout styles of the hierarchical SAT-based cell layout synthesis for dual CMOS cells. 25
3.2 Schematic of 2-input multiplexer and its logic block partitioning. 26
3.3 The problem definition of the intra-block transistor placement. 27
3.4 Additional variables introduced to maximize the number of the connections by diffusion sharing between logic blocks. 28
3.5 The problem definition of the inter-block placement. 31
3.6 Our hierarchical SAT-based cell layout synthesis flow. 32
3.7 A layout of buffered full adder generated by the proposed hierarchical SAT-based cell layout synthesis method. 35

4.1 An example of a dual CMOS circuit which has a non-dual structure after transistor folding. 38
4.2 The problem definition of the single row transistor placement for non-dual CMOS cells. 39
4.3 Schematic of 3-input multiplexer and its logic block partitioning. 43

4.4 Additional variables introduced to maximize the number of the connections by diffusion sharing between logic blocks. 44
4.5 The problem definition of the inter-block placement. 45
4.6 The problem definition of the multi-row transistor placement for dual and non-dual CMOS cells. 47
4.7 Examples of the layout of the cell number 7 in Table 4.9 created by (a)the conventional and (b)the proposed multi-row placement method. The conventional style assumes the pair of P/N transistors with the same gate input signals. 58

5.1 Critical areas between two wire segments spaced by d with the defect size x. . 62
5.2 Critical areas between two wire segments spaced by d when the defect size x is larger than $2d + w$. 63
5.3 Grid models used in our comprehensive intra-cell routing system. 66
5.4 Wire branching constraint of the proposed comprehensive intra-cell routing method. 67
5.5 The flow diagram of our comprehensive intra-cell router. 68
5.6 A sample problem for explaining our comprehensive intra-cell router. 70
5.7 An example of all possible connection patterns inside a column. 70
5.8 An example of all possible patterns to be extended to the subsequent column generated from the pattern (d) in Figure 5.7. 71
5.9 The flow diagram of our yield-optimal cell layout synthesis system. 72
5.10 Changes in layout sensitivity to spot defects by selecting the sensitivity-minimum layouts. 74
5.11 The optimal layouts of "mux2" in Table 5.2 generated by our method and selected by (a)wire length and (b)sensitivity to spot defects. 75
5.12 Lithography impact on the critical area before and after OPC for 340 cell layouts in a 90nm technology. 76
5.13 The relation between two cost metrics in the case of the exhaustively-generated 27414 cell layouts of xnor2. 77
5.14 The impact of the sensitivity reduction on the yield of the first metal layer. . . 78

6.1 Overview of the proposed timing-aware cell layout yield enhancement framework. 81

6.2 An example of a constraint graph constructed for polygons inside a given layout. 83
6.3 The conceptual layouts of 2-input NAND created by the conventional and the timing-aware de-compaction methods. 84
6.4 Schematic diagram of a short type critical area. 84
6.5 Variation of (a)vertical and (b)horizontal critical areas after horizontal de-compaction. 85
6.6 Change of the vertical critical area by the width or space. 86
6.7 Calculation of the vertical critical area. 86
6.8 The expected mask cost and mask data size in the future technologies. 87
6.9 The conceptual example of OPC data volume reduction by expanding the spacing between the polygons. 88
6.10 The change of the OPC pattern from (a)hammer head style to (b)serif style. . 88
6.11 Formulation of the redundant contact insertion. 89
6.12 Required gate CD control in the future technologies. 91
6.13 The conceptual example of gate layout pattern regularity enhancement for the regular pitch *P* by the proposed regularity enhancement method. 92
6.14 The regularity cost function used in the proposed regularity enhancement method. 93
6.15 An example of 2-input NAND and its RC network for calculating Elmore delay. 94
6.16 Preliminary results of delay variation by changing the transistor width and the input slew. 95
6.17 Preliminary results of output slew variation by changing the transistor width and the input slew. 96
6.18 The linear approximation of the intra-layer cross coupling capacitance between two vertically parallel wire segments. 97
6.19 Approximation of the cross coupling capacitance between two signals A and B. 98
6.20 The overall flow diagram of the proposed timing-aware yield enhancement method. 99
6.21 Accuracy of the proposed delay model in the case of the single-stage cell NOR4_1. 104

6.22 Accuracy of the proposed delay model in the case of the multi-stage cell ADDH_1. 104
6.23 Trade-off curves of (a)target delay versus CA and (b)cell area versus CA in the case of the single-stage cell NOR4_1. 105
6.24 Trade-off curves of (a)target delay versus CA and (b)cell area versus CA in the case of the multi-stage cell OR3_2. 106
6.25 The fractured mask data size of 20 cells in (a)90nm and (b)65nm technology in the case that 10% delay increase is allowed. 108
6.26 The reduction ratio of the fractured mask data size after de-compaction depending on the technology nodes. 109
6.27 An example of OPC results in 65nm technology (a)before and (b)after decompaction. 109
6.28 Accuracy of the proposed delay model in the case of NOR4_1. 113
6.29 Trade-off curves of target delay versus number of additional contacts in the case of NOR4_2. 113
6.30 The gate CD EPE histogram of all the cells used in this experiment without OPC to highlight the effectiveness. 116
6.31 Accuracy of the cell delay constraint in the case of the largest example in Table 6.9. 118
6.32 Trade-off curve between regularity enhancement and target cell delay in the case of the largest example in Table 6.9. 118

List of Tables

2.1 Our layout styles of the proposed SAT-based cell layout synthesis for dual CMOS cells. 8
2.2 The problem size comparison results of the transistor placement for dual cells between ILP and SAT formulations. 16
2.3 The runtime comparison results of the transistor placement for dual cells between ILP and SAT formulations. 17
2.4 The width comparison results of the cell layout synthesis between the commercial cell generation tool and the proposed method. 19
2.5 The runtime comparison results of the cell layout synthesis between the commercial cell generation tool and the proposed method. 20

3.1 Our layout styles of the hierarchical SAT-based cell layout synthesis for dual CMOS cells. 24
3.2 The variables used for the intra-block transistor placement formulation. . . . 28
3.3 Comparison results between the proposed hierarchical transistor placement and the original flat method proposed in Chapter 2. 34
3.4 Comparison results with the commercial cell generation tool and the original flat method proposed in Chapter 2. 34

4.1 The numbers of non-dual cells in commercial standard-cell libraries. 37
4.2 The variables used for the formulation of the single-row transistor placement for non-dual CMOS cells. 40
4.3 The variables used for the formulation of the multi-row placement. 48
4.4 Cells used for the experiment of the single-row transistor placement. 51
4.5 Problem size comparison results between the exact single-row transistor placement method for dual cells and the proposed flat approach. 52
4.6 Width and runtime comparison results between the exact single-row transistor placement method for dual cells and the proposed flat approach. 52

4.7 Comparison results between our flat and hierarchical approaches of the single-row transistor placement. 53
4.8 Comparison of the number of cells that can be applied to the proposed exact single-row transistor placement method and the method for dual cells in the case of a cell library with 340 cells. 55
4.9 Cells used for the experiment of the multi-row transistor placement. 55
4.10 Problem size of the proposed multi-row transistor placement formulation. . . 57
4.11 Width and runtime comparison results of the proposed multi-row transistor placement method. The conventional style assumes the pair of P/N transistors with the same gate input signals. 57

5.1 Our layout styles of the proposed comprehensive cell layout synthesis method. 64
5.2 The results of the comprehensive cell layout synthesis. 73
5.3 The cell selection results of the comprehensive cell layout synthesis. 73

6.1 The topological characteristics of the benchmark circuits used for the experiment of the critical area minimization. 100
6.2 The performance characteristics of the benchmark circuits used for the experiment of the critical area minimization. 101
6.3 The delay accuracy of the proposed timing-aware critical area minimization method. 102
6.4 Results of the critical area minimization by the proposed method. 103
6.5 The topological characteristics of the benchmark circuits used for the experiment of redundant contact insertion. 110
6.6 The performance characteristics of the benchmark circuits used for the experiment of redundant contact insertion. 110
6.7 The delay accuracy of the proposed timing-aware redundant contact insertion method. 111
6.8 Results of the redundant contact insertion by the proposed method. 112
6.9 The delay accuracy, area increase, and runtime of the proposed regularity enhancement method with 10% allowable delay increase. 114
6.10 The regularity cost reduction of the proposed regularity enhancement method with 10% allowable delay increase. 115

Chapter 1

Introduction

1.1 Background

The recent progress in VLSI process technologies enables us to integrate a large number of transistors on one chip, and significantly improves the circuit performance. On the other hand, due to the ever-increasing design complexity of the VLSI, we could never design any competitive SoCs within practical time-to-market without automated design techniques.

One of the major automated design methodologies for VLSIs is the cell-based design. Figure 1.1 shows the simplified flow diagram of the cell-based LSI design. In this design flow, a circuit mask layout is automatically generated through logic synthesis and place&route processes from a hardware description written in Hardware Description Language (HDL). This design flow has been automated by a lot of EDA vendors, and a lot of commercial tools are also available[1, 2, 3, 4, 5]. In these logic synthesis and place&route stage, we use a standard-cell library. The characteristics of cells including cell delay, area, and power are used in the logic synthesis stage, and the physical layout of each cell is used in the place&route stage. As is clear from this design flow, standard cells are the most fundamental components of VLSI, and provide the building blocks for creating large complex functions in both application-specific and semi-custom domains. Therefore, their performance has significant effects on the final performance of the synthesized VLSI.

Until recently, a lot of papers have been published in the area of the automatic cell layout synthesis[6, 7, 8, 9, 10]. Conventionally, these standard cells are usually designed and thoroughly optimized by hand. However, the progress in the automatic transistor-level layout generation now enables us to automatically generate the cell layouts which have comparable quality to those designed by hand and drastically reduces the required time for creating new standard-cell libraries. Moreover, some automatic cell synthesis methods are also used for

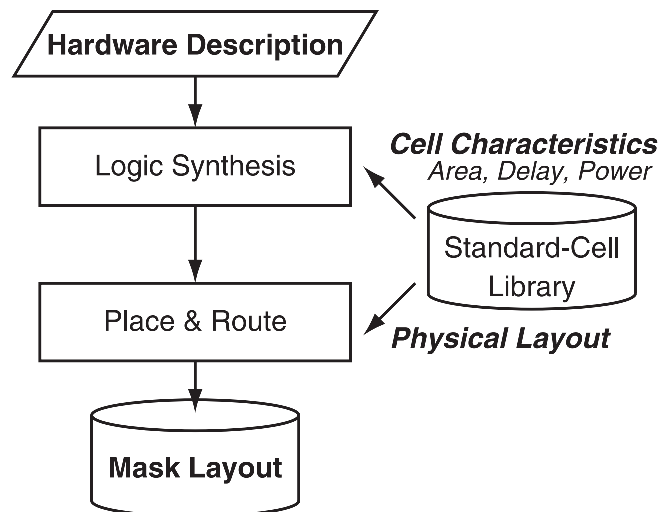


Figure 1.1 A simplified flow diagram of the cell-based LSI design.

on-demand cell synthesis. Especially in the area of the on-demand cell synthesis, not only the quality of cells but also the runtime reduction is extremely important. The coupling between on-demand cell synthesis and the technology mapping phase of logic synthesis can replace the concept of a cell library, and it is possible to reduce silicon area by a tighter coupling between cell generation and the automatic place&route system[11]. Based on these reasons, several commercial tools in automated standard-cell layout synthesis are widely used now[12, 13].

These standard cells are usually optimized for area, delay, and power and these factors are their basic characteristics used during logic synthesis and physical optimization. Therefore, almost all of the conventional methods and commercial tools target to optimize these factors. On the other hand, the recent progress in VLSI manufacturing technology and shrinking feature size lead to some new issues such as Design For Manufacturability (DFM). DFM is generally defined as a set of techniques to modify the design of circuits in order to make them more manufacturable, *i.e.*, to improve their yield. Due to very high costs associated with the manufacturability of deep sub-micron integrated circuits, even a small yield improvement can be extremely significant. Recently, a lot of papers related to VLSI yield improvement have been published[14, 15, 16, 17, 18]. Figure 1.2 illustrates the conceptual flow diagram of the yield-aware cell-based LSI design. The yield-aware logic synthesis[17] introduces the manufacturability cost into logic synthesis and replaces the traditional area-driven technology mapping with a new manufacturability-driven one. The yield-aware physical optimization[18] integrates manufacturability information into the timing-driven synthesis

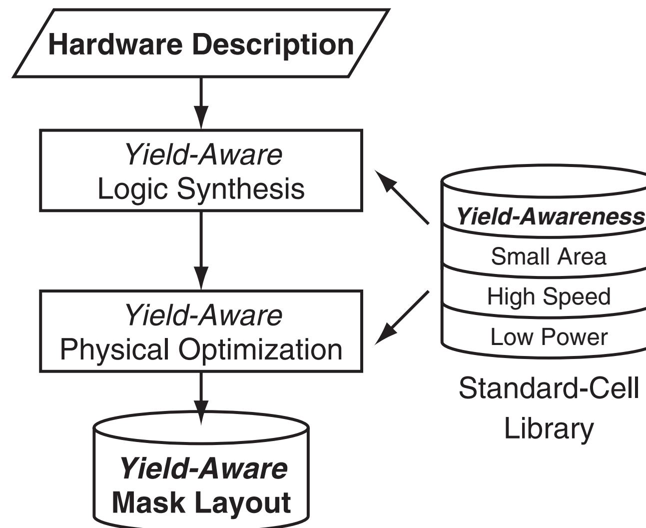


Figure 1.2 A flow diagram of the yield-aware cell-based LSI design.

and place&route cost function. In these yield-aware design flows, the yield-awareness is also a fundamental characteristic of the standard-cell layouts. Therefore, now the standard cells have to be optimized not only for area, delay, and power, but also for yield.

1.2 Research Objectives and Thesis Organization

This thesis focuses on the optimization methods for standard-cell layouts. Figure 1.3 illustrates the proposed optimization methods in perspective. There are two main optimization targets in this thesis, cell area optimization and yield optimization. Transistor placement and intra-cell routing methods for minimum-width cell layout synthesis via Boolean satisfiability are presented in Chapter 2 through Chapter 4. Chapters 2 and 3 propose flat and hierarchical cell layout synthesis methods for dual CMOS cells, respectively. The extension of these methods to the minimum-width transistor placement which is applicable to non-dual cells is explained in Chapter 4. Then, Chapter 5 shows a comprehensive cell layout synthesis method for yield optimization. This method utilizes the above-mentioned minimum-width transistor placement methods proposed for the cell area optimization, and routes the placements using a comprehensive intra-cell router in a grid-based manner. Yield-optimal cell layouts are selected from the exhaustively-generated cell layouts. Chapter 6 describes a timing-aware cell layout de-compaction method for yield optimization. This method performs a de-compaction of a given cell layout in a grid-less manner under given timing constraints for further yield optimization. A detailed explanation of each chapter is described as follows.

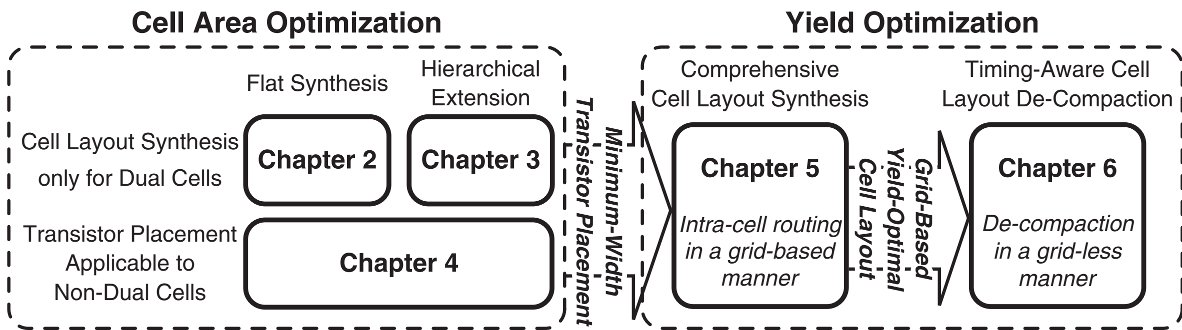


Figure 1.3 The proposed cell layout optimization methods in perspective.

Chapter 2 proposes a minimum-width layout synthesis method for dual CMOS cells via Boolean Satisfiability (SAT). Cell layout synthesis problems, *i.e.*, the transistor placement and the intra-cell routing problems are first transformed into SAT problems by this formulation. The proposed method guarantees to generate minimum-width cells with routability under our layout styles. This method places complementary P and N type transistors individually during transistor placement, and can generate smaller width layout compared with the case of pairing the complementary P and N type transistors. Since this method still has a restriction in gate connection style between P and N type transistors, it is applicable only to dual CMOS cells. The extension of the transistor placement method for non-dual cells is explained in Chapter 4.

Chapter 3 describes a hierarchical extension of the cell layout synthesis method proposed in Chapter 2 for cell layout synthesis of large dual CMOS cells. This method partitions a given transistor-level netlist into blocks considering the transistor connections by diffusions. Intra-block placement uses an exact transistor placement method which is proposed in Chapter 2, and hierarchically generates the transistor placement with routability. Experimental results show that the proposed method reduces the runtime for cell layout synthesis drastically with little width increase. The proposed method can be used as a quick layout generator in the area of transistor-level circuit optimization such as on-demand cell layout synthesis.

Chapter 4 shows flat and hierarchical approaches for generating a minimum-width transistor placement of CMOS cells in presence of non-dual P and N type transistors, whereas the cell layout synthesis methods proposed in the previous chapters are only for dual cells. This chapter targets the minimum-width transistor placement, and does not take the intra-cell routings into consideration. Our approaches are the first exact transistor placement method which can be applied to CMOS cells with any types of structure, whereas almost all of the conven-

tional exact transistor placement method is applicable only to dual CMOS cells. This chapter also shows the generalization results of the single-row transistor placement method into the multi-row placement. The proposed exact minimum-width multi-row transistor placement method uses more efficient gate connection style and generates more area-efficient transistor placements than the conventional multi-row transistor placement method only for dual cells.

Chapter 5 introduces a cell layout synthesis technique to optimize the yield. The yield cost metric used in the proposed method is the sensitivity to wiring faults due to spot defects. The sensitivity to faults on intra-cell routings is modeled with consideration to the spot defects size distribution and the end effect of critical areas. Although the critical area used for the sensitivity calculation is extracted from the original layout patterns without lithographic simulation, the feasibility of the proposed sensitivity model to the practical lithography system is discussed. The impact of the sensitivity reduction on the yield improvement is also discussed in this chapter. The minimum-width cell layout of CMOS logic cells are comprehensively generated using the transistor placement method proposed in Chapter 2 and the comprehensive intra-cell routing method proposed in this chapter. The yield optimal layouts are selected from the exhaustively-generated layouts by using the proposed sensitivity to wiring faults as a cost function.

Chapter 6 proposes a timing-aware cell layout de-compaction method for yield optimization using Linear Programming (LP). The proposed method performs a de-compaction of the original layout in order to improve the yield by minimizing the critical area inside the cell. This yield improvement procedure is executed under given timing constraints. To formulate the timing constraints into LP, a new accurate linear delay model which approximates the difference from the original delay is proposed. The effectiveness of the proposed method for OPC mask data volume reduction is also shown in this chapter. This timing-aware de-compaction framework is extended to the redundant contact insertion adjacent to the original single contacts to minimize the yield loss due to contact failure. To take the parametric yield into account, the proposed method is also extended to the gate layout pattern regularity enhancement to reduce the systematic variation of the gate critical dimensions. Using the proposed timing-aware yield enhancement method, we can explore the trade-off between yield and performance, and can pick up the yield/performance variants from the trade-off curve.

Finally, Chapter 7 gives conclusions of this thesis.

Chapter 2

Exact Minimum-Width Cell Layout Synthesis for Dual CMOS Cells

2.1 Introduction

This chapter targets to minimize the area of the dual CMOS cells and proposes a minimum-width layout synthesis method which is applicable only to dual CMOS cells. The proposed method generates minimum-width routable cell layouts via Boolean Satisfiability (SAT). In the area of the automated cell layout synthesis, several exact cell layout synthesis methods have been proposed. Gupta and Hayes proposed the CMOS cell width minimization via Integer Linear Programming (ILP)[19, 20, 21]. This method solves the width minimization of two dimensional transistor placement exactly. However, this method treats complementary P and N type transistors as a pair and aligns only these transistors vertically. There are some cases that the cell width becomes smaller when complementary transistors are not aligned vertically. Maziasz and Hayes proposed the exact algorithm for width and height minimization of CMOS cells[22]. This method minimizes both width and height considering intra-cell routability. However, this method also treats complementary transistors as a pair and the layout styles of this method have some difference from the practical cell layout styles. For example, it does not use horizontal polysilicon for routing.

We propose a minimum-width cell layout synthesis method for dual CMOS cells via SAT. Devadas has transformed various layout problems such as channel routing and partitioning into SAT problems[23]. However, the transistor placement and intra-cell routing problems could not be solved. The formulation of cell layout synthesis problems, *i.e.*, transistor placement and intra-cell routing problems, need some specific constraints such as diffusion sharing and complex routing patterns. We define the practical styles for cell layout to formulate these constraints into SAT, and cell layout synthesis problems are first transformed into SAT prob-

lems by our formulation.

The proposed method generates the minimum-width transistor placement with routability under our layout styles via SAT. The proposed method places complementary P and N type transistors individually during transistor placement, and can generate smaller width layout than the conventional exact methods explained above which treats the complementary P and N type transistors as a pair. Furthermore, we define more practical layout styles than the previous exact cell layout synthesis method, so that we can handle the multiple-sized transistors and the horizontal polysilicon. Our method does not minimize the cell height since we focus on the standard-cell layout synthesis whose height is usually fixed. Our SAT-based cell layout synthesis method generates the minimum-width placements in much shorter runtime than the 0-1 ILP-based transistor placement method, and also generates cell layouts in shorter runtime than the commercial cell generation tool. Therefore, it can significantly shorten the time-to-market of a cell library, and can also be used for many other applications such as on-demand cell synthesis, since it generates a layout quickly from a netlist, not from a pre-defined symbolic layout.

In Section 2.2, our layout styles are defined. The formulations of transistor placement and intra-cell routing are explained in Section 2.3 and Section 2.4, respectively. Experimental results are shown in Section 2.5, and finally Section 2.6 summarizes this chapter.

2.2 Layout Styles

Our layout styles are described in Table 2.1 and illustrated in Figure 2.1. By using these styles, the cell layout synthesis problems are efficiently transformed into SAT problems and a SAT solver can search for a solution quickly. The general styles for one dimensional width minimization of static dual CMOS logic cells were proposed by Uehara and vanCleemput[24]. Our layout style No. 1 through 5 are for the transistor placement essentially based on Uehara's styles. However, our layout styles have some differences from theirs. The first difference is the style No. 3. The complementary transistors have to be aligned vertically in Uehara's style. In contrast, transistors which have the same gate input signals can be aligned vertically in our layout style. As explained in Section 2.1, there are some cases that the generated layout has smaller width using our layout style. Moreover, our method can handle the cell circuits with non-complementary topologies in P/N type transistor networks if the circuit has no P and N type transistors which can not be paired by the common gate input signal. The second difference is that our method can take multiple-sized transistors as

Table 2.1 Our layout styles of the proposed SAT-based cell layout synthesis for dual CMOS cells.

1.	Static dual CMOS logic circuits.
2.	Transistors are drawn up in two horizontal rows. The upper row is for P type transistors and the bottom row is for N type transistors.
3.	Two transistors which have the same gate input signals are vertically aligned.
4.	Two transistors which have the same diffusion terminals are placed in the neighboring columns to share their diffusions.
5.	The bottom of the P diffusions and the top of the N diffusions are aligned to G-Region.
6.	The intra-cell routing uses polysilicon and first metal layers.
7.	All nets which connect diffusion terminals of P type transistors are in P-region.
8.	All nets which connect diffusion terminals of N type transistors are in N-region.
9.	All nets which connect gate terminals are in G-, Top- or Bottom-regions.
10.	Gate terminals are connected by the polysilicon layer in Top- or Bottom-region, and by the first metal layer in G-region.
11.	The same signals in P-region and N-region are connected by the vertical first metal through G-region at the top of N-region and the bottom of P-region.
12.	Vertical first metals and the gate terminals are connected by the horizontal first metals in G-region.
13.	VDD are connected from the top of P-diffusion through Top-region by the vertical first metal.
14.	GND are connected from the bottom of N-diffusion through Bottom-region by the vertical first metal.
15.	Single contact is assumed to be enough to connect between metal and diffusion or polysilicon.
16.	No dogleg is used.

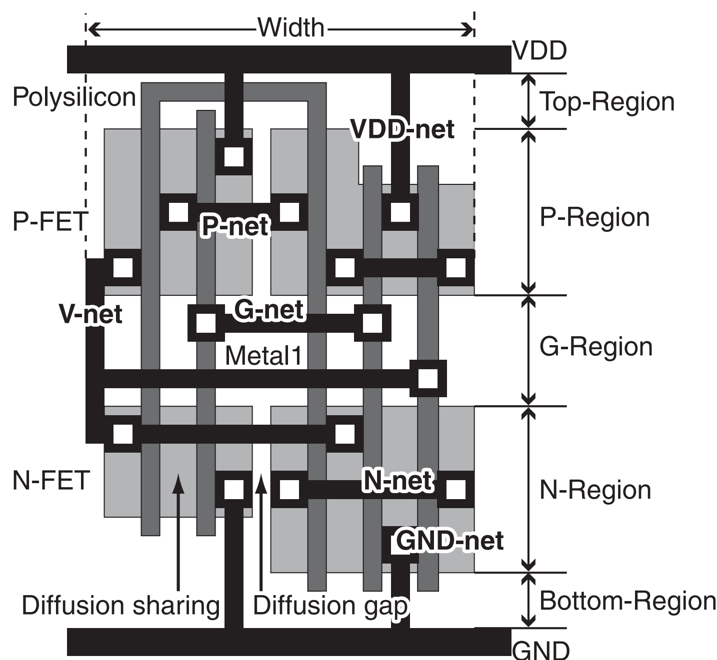


Figure 2.1 An illustration of the layout styles of the proposed SAT-based cell layout synthesis for dual CMOS cells.

an input, so that it can deal with more practical problems, while almost all previous optimal cell layout synthesis methods assumed the uniform-sized transistors. To treat multiple-sized transistors, the bottom of the P-diffusions and the top of the N-diffusions are aligned to the G-region, as described by No. 5 in Table 2.1 and illustrated in Figure 2.1.

The layout styles No. 6 through 16 in Table 2.1 are for intra-cell routing. These styles are based on Maziasz's styles[22]. Five routing regions are defined in the cell area as defined in Maziasz's styles. P- and N-regions are over the each diffusion, G-region is between the P- and N-regions, Top-region is above the P-region and Bottom-region is below the N-region, as shown in Figure 2.1. Our method can deal with outer channel polysilicon routing, *i.e.*, the connection which runs above P diffusions and below N diffusions as described by No. 9 and 10 in Table 2.1. By using outer channel routing, we can avoid the second metal layers for intra-cell routing in many cases. Therefore, our routing styles will be widely accepted in practical applications.

2.3 Transistor Placement Formulation

In this section, we explain the SAT constraints for transistor placement. Given N P and N N type transistors, we have to place these $2N$ transistors in the minimum area. This problem can

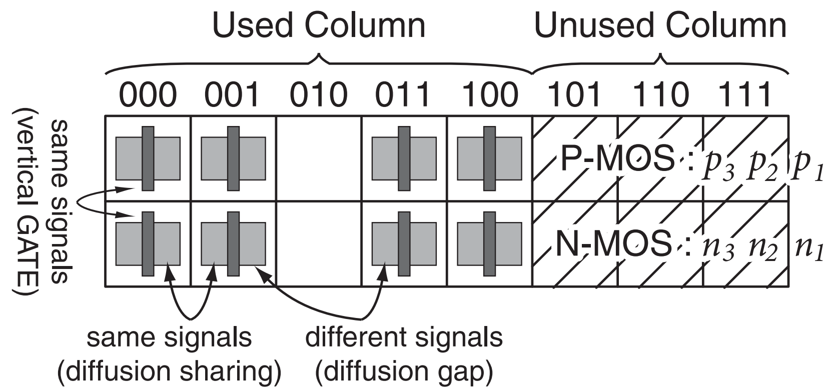


Figure 2.2 A SAT formulation of the transistor placement for dual CMOS cells.

be transformed into the problem that places all transistors using minimum number of columns as illustrated in Figure 2.2. The P type transistors are aligned in the upper row and the N type transistors are in the bottom (No. 2 in Table 2.1). The P and N type transistors which are placed in the same column must have the same gate input signals (No. 3 in Table 2.1). The neighboring transistors must face the diffusions which belong to the same signal each other to share their diffusions (No. 4 in Table 2.1). The empty columns result in the diffusion gaps in the final layouts. We transform these constraints into Boolean constraints.

Each transistor has P variables to identify its placement where $P = \lceil \log_2 W \rceil$ and W is the number of the columns, and one variable to identify whether the source/drain terminals are flipped or not. $\lceil X \rceil$ indicates a minimum integer which is equal to or larger than X. Thus, the total number of variables needed for this formulation is $2N \times (\lceil \log_2 W \rceil + 1)$. Here, we describe the Boolean constraints.

Transistor overlap constraints: N type transistors must not overlap in the same column, which is expressed by the following logic equation

$$n_{i1} \oplus n_{j1} \vee n_{i2} \oplus n_{j2} \vee \cdots \vee n_{iP} \oplus n_{jP} = 1 \tag{2.1}$$

where $n_{i1}, n_{i2}, \ldots, n_{iP}$ are the variables for placement of an N type transistor i, and $\oplus$ is the exclusive-or Boolean operator. The same logic equation must hold for P type transistors.

Unused column constraints: If $W < 2^P$, the columns with the top $2^P - W$ distinct bit vectors of length P, correspond to unused columns as illustrated in Figure 2.2. No N type transistors can be placed in these unused columns. The following logic equation expresses this constraint for N type transistors.

$$n_{i1} \oplus u_1 \vee n_{i2} \oplus u_2 \vee \cdots \vee n_{iP} \oplus u_P = 1 \tag{2.2}$$

where $u_{k1}, u_{k2}, \ldots, u_{kP}$ are constant bit vectors which indicate the unused column k. The same logic equation must hold for P type transistors.

Vertical gate constraints: The P and N type transistors which are placed in the same column must have the same gate input signals. Assume G_p^i is the group of P type transistors which have the same gate input signals as that of an N type transistor i, this constraint is expressed as follows.

$$\bigvee_{j \in G_p^i} \overline{(n_{i1} \oplus p_{j1} \vee n_{i2} \oplus p_{j2} \vee \cdots \vee n_{iP} \oplus p_{jP})} = 1 \tag{2.3}$$

Neighboring Transistor constraints: The transistors which face the diffusions which belong to the different signals each other can not be in the neighboring columns. Assume f_{ni} is the variable which determines the flip of an N type transistor i, and $C_n(i, k)$ takes the value of 1 if an N type transistor i is in the column k, the following logic equation expresses this constraint for N type transistors.

$$GAP_n(i, j) \wedge (\bigvee_{k=0}^{W-2} C_n(i, k) \wedge C_n(j, k+1)) = 0 \tag{2.4}$$

Here, $GAP_n(i, j)$ takes the value of 1 if an N type transistor i can not share its diffusion with an N type transistor j placed to its immediate right, otherwise 0. The same logic equation must holds for P type transistors.

These Boolean constraints express all the possible placement in W columns under our layout styles. These constraints are expressed in Conjunctive Normal Form (CNF) to be solved by the CNF-SAT solver. If there is no satisfiable assignment using W columns, it is guaranteed that there is no possible placement of width W. Therefore, we can find the minimum-width placement using a procedure described below.

1. For a given transistor netlist, enumerate the number of transistors. The initial number of column W is set to the number of N type transistors (=#P-FETs).

2. Search for a satisfiable assignment for the Boolean constraints constructed for W columns. If a satisfiable assignment is found, these transistors can be placed in W columns and this procedure terminates. Otherwise, go to step 3.

3. $W = W + 1$. Go to step 2 again.

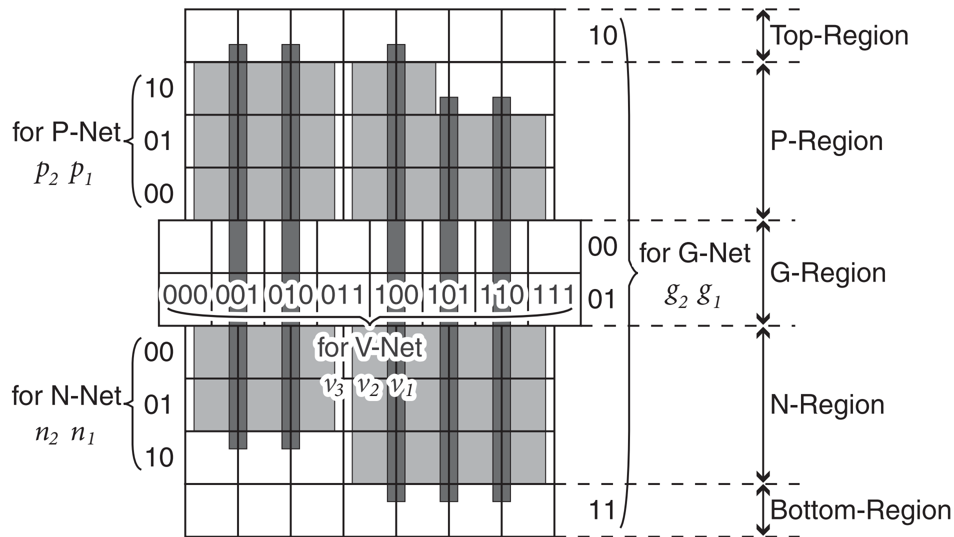


Figure 2.3 A SAT formulation of the intra-cell routing.

2.4 Intra-Cell Routing Formulation

We next explain the SAT constraints of the intra-cell routing in this section. After a transistor placement is generated using the formulation explained in Section 2.3, its intra-cell routability is checked using the constraints explained in this section. Our styles of the intra-cell routing are listed by No. 6 through 16 in Table 2.1. We defined five types of net listed below as illustrated in Figure 2.1.

N-net The net which connects the diffusion terminals of N type transistors.

P-net The net which connects the diffusion terminals of P type transistors.

G-net The net which connects the gate terminals.

V-net The net which connects the diffusion terminals of P and N type transistors, and the gate terminals.

VDD/GND-net The net which connects the VDD/GND signals to VDD/GND rail which runs top/bottom of cell area.

We also defined the region where each type of net can be placed as described by No. 7 through 14 in Table 2.1. Bit vectors which correspond to the row or column numbers are assigned to each region as illustrated in Figure 2.3. The routing grids are also defined as illustrated in Figure 2.3. We use the columns which are shifted by a half column in the G-region. The

number of rows of the N-region (P-region) is determined by the maximum width of N type transistors (P type transistors) and the design rules since the grid size is determined by the minimum width and spacing rules of metal or polysilicon and some other design rules. The numbers of rows of the Top-region and the Bottom-region are both fixed to 1. Therefore, the number of rows of the G-region W_g is given by the following equation

$$W_g = H_{cell} - W_p - W_n - 2 \tag{2.5}$$

where H_{cell} is the total number of rows of each cell, which is fixed for all cells so that the height of all cells are uniform, and W_p and W_n are the number of rows of P and N region, respectively.

Assume that P_n, P_p, P_g, and P_v are the number of the Boolean variables for each type of net, they are expressed as $P_n = \lceil \log_2 W_n \rceil$, $P_p = \lceil \log_2 W_p \rceil$, $P_g = \lceil \log_2 (W_g + 2) \rceil$, $P_v = \lceil \log_2 W_v \rceil$ where W_n, W_p, and W_g are the number of the rows of each region, and W_v is the number of the columns of the G-region. G-nets can be placed in $W_g + 2$ rows since they can be placed in the Top- and Bottom-region besides the G-region. The variables of the nets which belong to more than two groups consist of the combination of each group's variables. Therefore, the total number of variables P_{total}, used for the SAT formulation of intra-cell routing is expressed as

$$P_{total} = \sum_{i \in net} (a_i P_n + b_i P_p + c_i P_g + d_i P_v) \tag{2.6}$$

where a_i, b_i, c_i, and d_i are the variables which take the value of 1 if the net i belongs to each group, otherwise 0. Here, we construct the Boolean constraints:

Net overlap constraints: We define the interval of an N-net i as $I_n(i) = [l_{ni}, r_{ni}]$, where l_{ni} is the position of the leftmost terminal in the N-region which the net has to be connected to, and r_{ni} is the position of the rightmost one. Two nets with intersecting intervals can not be placed in the same row. For each net pair i and j, the logic equation describing this constraint is as follows

$$n_{i1} \oplus n_{j1} \vee n_{i2} \oplus n_{j2} \vee \ldots \vee n_{iP_n} \oplus n_{jP_n} = 1 \qquad i \neq j, \;\; I_n(i) \cap I_n(j) \neq \phi \tag{2.7}$$

where $n_{i1}, n_{i2}, \cdots, n_{iP_n}$ are the variables which correspond to the placement of an N-net i. The same logic equation must hold for P-nets and G-nets. For V-nets, all pairs of nets must satisfy the above logic equation. Therefore, the logic equation for V-nets is expressed as follows

$$v_{i1} \oplus v_{j1} \vee v_{i2} \oplus v_{j2} \vee \ldots \vee v_{iP_v} \oplus v_{jP_v} = 1, \;\; i \neq j \tag{2.8}$$

where, $v_{i1}, v_{i2}, \cdots, v_{iP_v}$ are the variables which correspond to the placement of a V-net i.

Unused row/column constraints: If $W_v < 2^{P_v}$ for V-nets, the columns with $2^{P_v} - W_v$ distinct bit vectors correspond to unused columns. No net can be placed in these unused columns. The same case emerges when $W_g + 2 < 2^{P_g}$ for G-nets. Whereas for N(P)-nets, if $min_{ni} < 2^{P_n}$ ($min_{pi} < 2^{P_p}$) where min_{ni} (min_{pi}) is the minimum grid number of N(P) diffusions whose signal has to be connected to an N(P)-net i, the N(P)-net i can not be placed in the columns denoted by $2^{P_n} - min_{ni}$ ($2^{P_p} - min_{pi}$) distinct bit vectors. This constraint enables us to route the multiple-sized transistors. This constraint is expressed by the following logic equation.

$$n_{i1} \oplus u_{k1} \vee n_{i2} \oplus u_{k2} \vee \cdots \vee n_{iP_n} \oplus u_{kP_n} = 1 \tag{2.9}$$

The constant bit vector $u_{k1}, u_{k2}, \ldots, u_{kP_n}$ indicates the unused row/column. The same logic equation must hold for P-nets, G-nets, and V-nets.

VDD/GND-net constraints: P-nets must not overlap the VDD-nets. We assume V is the group of the column numbers which has to be connected to VDD. If $x \in V$ and $x \in I_p(i)$, a P-net i can not be placed in the top row of the minimum-width diffusion whose column belongs to $I_p(i)$ and V. For N-nets, they must not overlap the GND-nets. We define the group G whose members are the column numbers which has to be connected to GND. If $x \in G$ and $x \in I_n(i)$, an N-net i can not be placed in the bottom row of the minimum-width diffusion whose column belongs to $I_n(i)$ and G. These constraint is expressed as equation (2.9) where $u_{k1}, u_{k2}, \ldots, u_{kP_n}(u_{kP_p})$ indicate the row where N(P)-nets can not be placed in.

V-net connection through the G-region constraints: If a V-net i is placed in the column c_i, there must be no horizontal net over the column c in the G-region so that the V-net can go through the G-region. Assume H_g is the group of horizontal nets in the G-region and $I_{hg}(i) = [l_i, r_i]$ is the interval of the horizontal net where l_i (r_i) is the leftmost (rightmost) terminal or net in the G-region, this constraint is described as follows.

$$|X| = 0, \ X = \{x \mid x \in H_g, \ c_i \in I_{hg}(x)\} \tag{2.10}$$

V-net to gate connection constraints: If a V-net i must be connected to the gate terminals, this V-net has to be connected to these terminals in the G-region by horizontal first metal layer. Therefore, there has to be at least one empty row between this V-net and terminals in the G-region. This constraint is described as follows using H_g and I_{hg} defined before.

$$|X| < W_g, \ X = \{x \mid x \in H_g, \ I_{hg}(x) \cap I_{hg}(i) \neq \phi\} \tag{2.11}$$

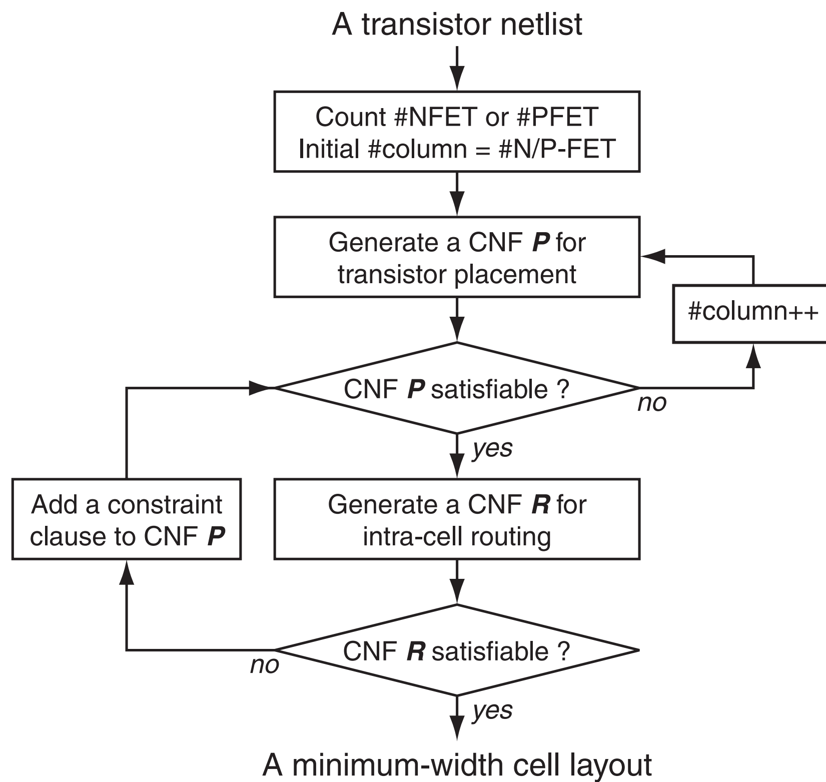


Figure 2.4 Our SAT-based cell layout synthesis flow.

V-net to diffusion connection constraints: If a V-net i must be connected to the diffusion terminals of N type transistors, this V-net is connected to these diffusion terminals in the top row of the N-region by horizontal first metal layer. Therefore, there has to be no other net placed between such V-nets and terminals in the top row of the N-region. Assume H_n is the group of horizontal nets which are placed in the top row of the N-region and $I_{hn}(j) = [l_j, r_j]$ is the interval of the horizontal net where l_j (r_j) is the leftmost (rightmost) terminal or net in the N-region, this constraint is described as follows.

$$|X| = 0,\ X = \{x \mid x \in H_n,\ I_{hn}(x) \cap I_{hn}(i) \neq \phi\} \tag{2.12}$$

In the case of P type transistors, this V-net and terminals must be connected to in the bottom row of the P-region and the same logic equation must hold for P type transistors.

The SAT formulation which consists of these constraints allows us to determine whether the placement is routable or not. To find a routable placement, we iterate the generation of placements and satisfiability checks. If a placement is proved to be unroutable, we add a new clause which suppresses the previous placement to CNF for transistor placement to generate a new placement. These clauses are called constraint clause. The flow of our SAT-based cell layout synthesis is illustrated in Figure 2.4.

Table 2.2 The problem size comparison results of the transistor placement for dual cells between ILP and SAT formulations.

Circuit			*Problem Size*			
			ILP		*SAT*	
name	*#tr.*	*#col.*	*#var.*	*#ineq.*	*#var.*	*#cla.*
ao222	14	8	1054	1926	56	1624
ao44	20	11	2767	5450	100	5360
aoi211	8	4	241	394	24	244
fad1	28	15	7011	14082	140	47404
gen3	16	10	1509	2836	80	3342
maj3	12	6	699	1214	48	1100
mux2	12	8	699	1240	48	1260
nand2	4	2	39	56	8	18
nand3	6	3	114	178	18	94
xnor2	10	6	432	716	40	790

2.5 Experimental Results

The formulations explained in Section 2.3 and Section 2.4 and the flow illustrated in Figure 2.4 enable us to generate a minimum-width routable cell layout from a netlist via Boolean satisfiability. To test the effectiveness of our SAT-based cell layout synthesis method, we compare it with a 0-1 ILP-based transistor placement and a commercial cell generation tool.

2.5.1 Comparison with the 0-1 ILP Formulation

We conducted the runtime comparison with 0-1 ILP formulation in transistor placement stage. To conduct the experiment, we also transformed the transistor placement problems into the 0-1 ILP problems. Our formulation of the 0-1 ILP is based on the Gupta and Hayes' formulation[20]. The differences from [20] are that we assume only one dimensional placement and P and N type transistors with the same gate input signals can be aligned vertically, while their formulation assumed two dimensional placement and only complementary P and N type transistors are aligned vertically.

We used the CNF-SAT solver *Chaff*[25], the 0-1 ILP solver *OPBDP*[26], and the generic ILP solver *CPLEX*[27] for the experiments. The *Chaff* and *OPBDP* experiments were con-

Table 2.3 The runtime comparison results of the transistor placement for dual cells between ILP and SAT formulations.

Circuit			*Runtime (sec.)*				
			ILP		*SAT*		
name	*#tr.*	*#col.*	*OPBDP*	*CPLEX*	*OPBDP*	*CPLEX*	*Chaff*
ao222	14	8	0.59	134.17	1.30	14.17	**0.15**
ao44	20	11	1551.45	>3600	20.36	>3600	**0.66**
aoi211	8	4	0.07	0.04	0.03	0.06	**0.01**
fad1	28	15	>3600	>3600	>3600	>3600	**201.05**
gen3	16	10	9.95	2872.81	11.6	2150.69	**1.67**
maj3	12	6	0.30	15.42	0.19	2.01	**0.06**
mux2	12	8	0.39	37.74	0.92	20.12	**0.23**
nand2	4	2	0.03	**~0**	0.05	0.02	0.03
nand3	6	3	0.04	0.02	0.03	0.03	**0.01**
xnor2	10	6	0.17	1.56	0.30	1.91	**0.08**
Total Ratio	—	—	>33.04	>71.18	>17.72	>68.21	1.00

ducted on a 750MHz UltraSPARC-III workstation with 2GB of RAM. The *CPLEX* experiments were done on a Pentium-III 1GHz with 2GB of RAM. We used default settings for all of them. A time-out limit of 3,600 seconds was used for each run. We experimented the ILP formulation based on the Gupta and Hayes' formulation and the ILP formulation simply transformed from CNF for the SAT solver, as an input file for the two ILP solvers. Tables 2.2 and 2.3 list the results of solving the transistor placement problems by SAT and ILP solvers. Although both tables contain only 10 circuits, we tested 30 static dual CMOS logic circuits in a standard-cell library. The data in the row of Total Ratio in Table 2.3 means the ratio of the total runtime for the 30 circuits. For each circuit, Table 2.2 indicates the number of transistors, the number of columns after placement, the problem size for the ILP as well as for the SAT formulations (number of ILP variables and inequalities are expressed as *#var.* and *#ineq.*; CNF variables and clauses are as *#var.* and *#cla.*). The problem size of SAT formulation is the size of the problem which has a satisfiable assignment first. Table 2.3 shows the *OPBDP* and *CPLEX* runtimes using the ILP formulation, and *OPBDP*, *CPLEX*, and *Chaff* runtimes using the SAT formulation as the input.

Compared with the two ILP solvers, the SAT solver *Chaff* has shorter runtimes in most

cases. Moreover, the ILP solvers can not solve some large circuits in one hour whereas *Chaff* can in minutes. The total runtime of the ILP solvers are about 17 to 70 times longer than the SAT solver. These results clearly show that the SAT formulation and the SAT solver are more suitable for solving the transistor placement problems.

The layouts generated by our method are equal to or smaller than the one dimensional layout generated by Gupta and Hayes' method[19] which treats complementary P and N type transistors as a pair. For example, the width of the full adder "fad1" in Table 2.3 is 15 using our method whereas the minimum width of the one dimensional layout of the full adder described in [19] is 16.

2.5.2 Comparison with the Commercial Cell Generation Tool

We also compared our cell generation method with the commercial cell generation tool *ProGenesis*[13]. The experiments were conducted on a 750MHz UltraSPARC-III workstation with 2GB of RAM. Tables 2.4 and 2.3 list the results of generating the 30 static dual CMOS logic circuits in a standard-cell library by our method and *ProGenesis*. The height of all circuits is fixed to 10 rows in this experiment. Table 2.4 shows the number of transistors, the SAT problem size of the intra-cell routing, the width of cells generated by the *ProGenesis* and our method. Table 2.5 shows the runtime spent on generating a routable cell layout from a netlist. In the column of *Resultant Width* in Table 2.4, we assumed the 0.35μm process technology. The layouts generated by *ProGenesis* and our method are both design rule correct. The column *width* for *ProGenesis* lists the width of the generated layout after compaction and the asterisk in this column means that it uses second metal layers to route these circuits. The column *#col.* for *ProGenesis* lists the number of columns of the generated symbolic layouts before compaction. The column *width* for the proposed method lists the width of the generated layout by the proposed method without compaction. The column *#col.* shows the number of columns needed for each circuit to complete the intra-cell routing using our layout style.

One column is added to the minimum-width placement of ao33, ao44, oa33, and oa44 to complete the intra-cell routing of each circuit, and the width of these cells in column are 1 column larger than the commercial tool as shown in tables 2.2 and 2.4. Because of the routing restrictions of out layout styles, the numbers of columns of these cells are larger than those generated by *ProGenesis*. Although the number of columns of cell mux2 generated by the proposed method is also larger, this reason is not the routing restriction but the transistor

Table 2.4 The width comparison results of the cell layout synthesis between the commercial cell generation tool and the proposed method.

Circuit		*SAT*		*Resultant Width*			
		Problem Size		*ProGenesis*		*Proposed*	
name	*#tr.*	*#var.*	*#cla.*	*width (μm)*	*#col.*	*width (μm)*	*#col.*
ao222	14	14	337	**11.85**	8	13.20	8
ao33	16	20	1102	**13.95**	9	15.90	*10*
ao44	20	20	1420	**15.50**	11	18.90	*12*
aoi21	6	5	14	5.40	3	5.40	3
aoi211	8	7	16	**6.45**	4	6.90	4
buf1	4	4	49	**3.90**	2	4.20	2
eno	10	8	144	8.40	5	8.40	5
eor	10	8	144	8.40	5	8.40	5
fad1	28	36	6169	26.05*	15	**24.00**	15
gen2	12	14	295	**11.00***	7	11.70	7
gen3	16	20	520	21.40*	10	**16.20**	10
maj3	12	14	215	10.80	6	**10.20**	6
mux2	12	22	900	**10.60**	6	13.20	*8*
nand2	4	4	4	**3.70**	2	3.90	2
nand22	6	6	81	5.80	3	**5.70**	3
nand3	6	4	7	5.40	3	5.40	3
nand4	8	4	7	6.90	4	6.90	4
nand44	10	8	150	**8.30**	5	8.70	5
nor2	4	4	4	3.90	2	3.90	2
nor22	6	6	81	**5.40**	3	5.70	3
nor3	6	5	8	**5.10**	3	5.40	3
nor4	8	5	8	8.10	4	**6.90**	4
nor44	10	8	150	**8.10**	5	8.70	5
oa222	14	14	335	**12.00**	8	13.20	8
oa33	16	17	1095	**13.25**	9	15.90	*10*
oa44	20	20	1506	**15.70**	11	18.90	*12*
oai21	6	7	18	5.40	3	5.40	3
oai211	8	7	18	**6.55**	4	6.90	4
xnor2	10	14	229	**8.80**	6	10.20	6
xor2	10	14	219	**8.65**	6	10.20	6
Total	—	—	—	285.2 (1.00)	172 (1.00)	298.5 (1.05)	178 (1.03)

Table 2.5 The runtime comparison results of the cell layout synthesis between the commercial cell generation tool and the proposed method.

Circuit		*SAT Problem Size*		*Runtime (sec.)*	
name	*#tr.*	*#var.*	*#cla.*	*ProGenesis*	*Proposed*
ao222	14	14	337	144.96	**1.06**
ao33	16	20	1102	**185.07**	392.26
ao44	20	20	1420	**291.79**	637.26
aoi21	6	5	14	28.51	**0.04**
aoi211	8	7	16	52.06	**0.02**
buf1	4	4	49	18.67	**0.06**
eno	10	8	144	77.67	**0.04**
eor	10	8	144	63.54	**0.06**
fad1	28	36	6169	509.27	**305.76**
gen2	12	14	295	141.17	**0.16**
gen3	16	20	520	386.69	**2.20**
maj3	12	14	215	87.93	**0.20**
mux2	12	22	900	77.28	**6.42**
nand2	4	4	4	16.56	**0.04**
nand22	6	6	81	27.02	**0.04**
nand3	6	4	7	25.91	**0.04**
nand4	8	4	7	33.84	**0.04**
nand44	10	8	150	76.76	**0.06**
nor2	4	4	4	16.73	**0.05**
nor22	6	6	81	30.77	**0.05**
nor3	6	5	8	26.53	**0.06**
nor4	8	5	8	40.98	**0.07**
nor44	10	8	150	62.03	**0.07**
oa222	14	14	335	167.45	**1.07**
oa33	16	17	1095	**204.5**	443.94
oa44	20	20	1506	295.87	**116.41**
oai21	6	7	18	30.03	**0.04**
oai211	8	7	18	54.60	**0.05**
xnor2	10	14	229	55.43	**0.09**
xor2	10	14	219	59.00	**0.12**
Total	—	—	—	3288.62 (1.00)	1907.78 (0.58)

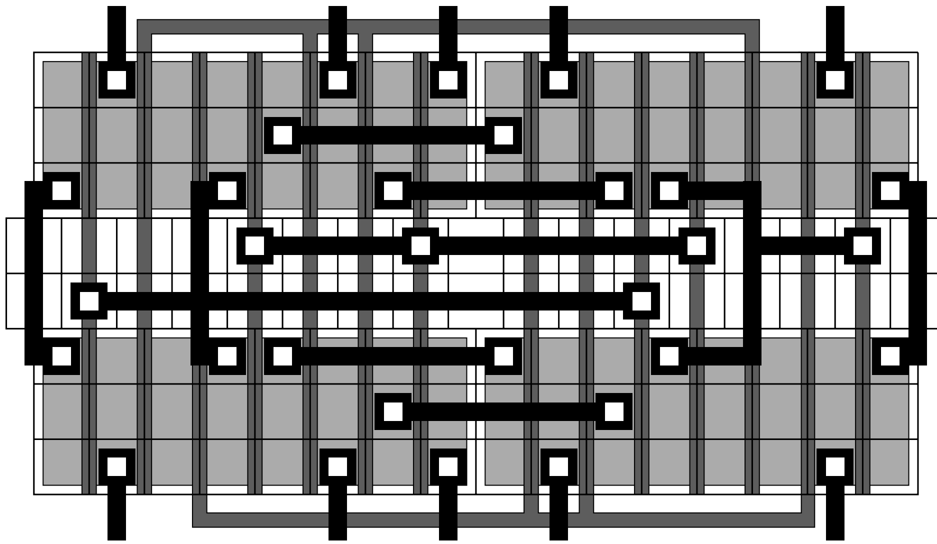


Figure 2.5 The example result of fad1 cell layout generated by the proposed SAT-based cell layout synthesis method for dual CMOS cells.

placement restriction. The P and N type transistors which are placed in the same column must have the same gate input signals in the proposed method, whereas *ProGenesis* does not have such restriction. Because of this restriction, the width of the cell mux2 which includes transmission gates becomes larger than *ProGenesis*. The width increase in terms of the number of columns is about 3% for total of 30 circuits used in this experiment.

ProGenesis generates smaller width layouts for 18 circuits after layout compaction, since it can use the bending gate in the G-region and the smaller width diffusion if it has no contact on it during compaction, whereas the proposed method does not include the compaction. However, it uses the second metal layers to complete the intra-cell routing for fad1, gen2, gen3 and the width of fad1, gen3, maj3, nand22, nor4 are larger than our method. The runtimes of our method are smaller than *ProGenesis* for almost all cases as shown in Table 2.5. The proposed method generates all these 30 circuits in total 58% runtime with only 5% area increase compared with *ProGenesis*. These results show that our cell layout styles defined for the SAT formulation is practical enough to generate the layout of dual CMOS cells quickly with a little area overhead. The snapshot of fad1 layout generated by the proposed method is illustrated in Figure 2.5 for example.

2.6 Summary

We proposed a minimum-width cell layout synthesis method for dual CMOS cells via Boolean Satisfiability (SAT). Cell layout synthesis problems are first transformed into SAT problems by our formulation. We presented that the SAT formulation is more suitable for the transistor placement by comparing the runtime of the SAT and the 0-1 ILP formulations of the transistor placement problems. We also presented that the width of placements generated by our method are smaller than that of the conventional exact transistor placement method by using our layout styles. The proposed method generates the cell layouts of 30 static dual CMOS logic circuits in 58% runtime with only 5% area increase compared with the commercial cell generation tool *ProGenesis* with cell layout compaction. These results showed that our cell layout styles defined for the SAT formulation is practical enough to generate the layout of dual CMOS cells quickly with a little area overhead. We can conclude from these results that our method can significantly shorten time-to-market of a cell library and can also be useful for many other applications such as on-demand cell layout synthesis. Since this method still has a restriction in gate connection style between P and N type transistors, it is applicable only to dual CMOS cells. The extension of the transistor placement method to non-dual cells is explained in Chapter 4.

Chapter 3

Hierarchical Extension for Large Cell Layout Synthesis

3.1 Introduction

This chapter describes a hierarchical extension of the layout synthesis method explained in Chapter 2 for layout synthesis of large dual CMOS cells. The proposed method partitions a given transistor-level netlist into blocks considering the transistor connections by diffusions and places all transistors hierarchically. Intra-block placement uses an exact transistor placement method which is proposed in Chapter 2. During intra-block transistor placement, a new optimization cost is introduced to maximize the number of the connections by diffusion sharing between logic blocks in the subsequent inter-block placement step. This new cost function is a key to realize a hierarchical transistor placement and the drastic runtime reduction with little increase in cell width. Intra-cell routability of the generated transistor placement is also checked in the same manner as in Chapter 2 in order to generate a routable cell layout.

The layout styles of the hierarchical cell layout synthesis method are defined in Section 3.2. Section 3.3 and Section 3.4 explain the formulation of the hierarchical transistor placement and intra-cell routings. In Section 3.6, experimental results of the proposed cell layout synthesis are presented. Finally, Section 3.7 summarizes this chapter.

3.2 Layout Styles

Our cell layout styles for the hierarchical cell layout synthesis are described in Table 3.1 and illustrated in Figure 3.1. These styles are basically same as the styles used in Chapter 2. Particularly the layout styles No. 5 and 10 in Table 3.1, the alignment style of the multiple-

Table 3.1 Our layout styles of the hierarchical SAT-based cell layout synthesis for dual CMOS cells.

1.	Static dual CMOS logic circuits.
2.	Transistors are drawn up in two horizontal rows. The upper row is for P type transistors and the bottom row is for N type transistors.
3.	Two transistors which have the same gate input signals are vertically aligned.
4.	Two transistors which have the same diffusion terminals are placed in the neighboring columns to share their diffusions.
5.	The top of the P diffusions are aligned to Top-Region and the bottom of the N diffusions are aligned to Bottom-Region.
6.	The intra-cell routing uses polysilicon and first metal layers.
7.	All nets which connect diffusion terminals of P type transistors are in P-region.
8.	All nets which connect diffusion terminals of N type transistors are in N-region.
9.	All nets which connect GATE terminals are in G-, Top- or Bottom-regions.
10.	Gate terminals are connected by the polysilicon layer in Top- or Bottom-region, and by the first metal or polysilicon layer in G-region.
11.	The same signals in P-region and N-region are connected by the vertical first metal through G-region at the top of N-region and the bottom of P-region.
12.	Vertical first metals and the GATE terminals are connected by the horizontal first metals in G-region.
13.	VDD are connected from the top of P-diffusion through Top-region by the vertical first metal.
14.	GND are connected from the bottom of N-diffusion through Bottom-region by the vertical first metal.
15.	Single contact is assumed to be enough to connect between metal and diffusion or polysilicon.
16.	No dogleg is used.

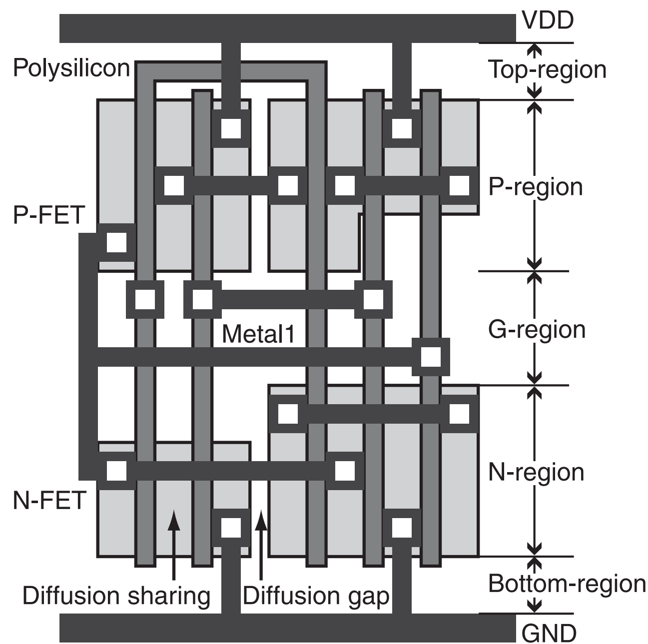


Figure 3.1 An illustration of the layout styles of the hierarchical SAT-based cell layout synthesis for dual CMOS cells.

sized P and N type transistors and the usage of polysilicon layer inside the G-region are different from the styles in Chapter 2. Although these styles are changed to improve the intra-cell routability, they do not have a significant effect on the results of the cell layout synthesis. Under these layout styles, the proposed method synthesizes the layout of the dual CMOS logic cells hierarchically.

3.3 Hierarchical Transistor Placement

In this section, we explain the formulation of the hierarchical transistor placement. The proposed hierarchical transistor placement method has three steps, partitioning, intra-block placement, and inter-block placement. The proposed method partitions a given transistor-level netlist into blocks considering the transistor connections by diffusions, then place all transistors inside each block in minimum width via Boolean satisfiability. After all the intra-block transistor placements are finished, all the blocks are placed in minimum-width in the same manner as the intra-block transistor placement. Detailed explanation of each step is described as follows.

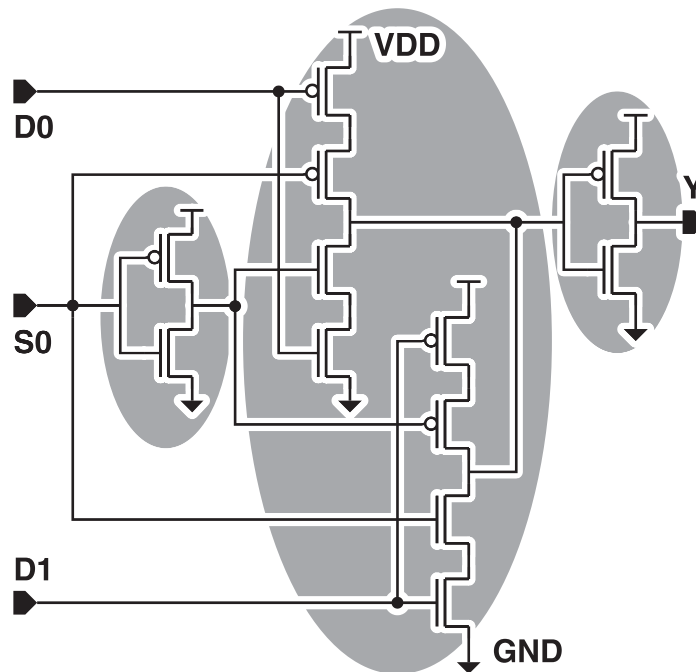


Figure 3.2 Schematic of 2-input multiplexer and its logic block partitioning.

3.3.1 Partitioning into Logic Blocks

Our approach first partitions the given circuit into logic blocks. Conventionally, several hierarchical layout synthesis methods for runtime reduction have been proposed[28, 29]. The proposed method partitions a given transistor network into logic blocks in the same manner as these methods[28, 29]. A logic block consists of transistors that are connected together by their diffusions except power and ground. An example of the partitioning result of 2-input multiplexer is shown in Figure 3.2. Two clocked inverters whose outputs are connected are identified as one logic block using our partitioning procedure. Since it is important to maximize the diffusion sharing between transistors in order to minimize the placement width, the partitioning procedure based on the diffusion connections is not expected to worse the results severely.

3.3.2 Intra-Block Transistor Placement Formulation

Next, we explain the intra-block transistor placement procedure. The proposed method uses Boolean satisfiability to generate the minimum-width intra-block transistor placement. In this chapter, we use Pseudo-Boolean formulation[30] to formulate the hierarchical cell layout synthesis. Pseudo-Boolean formulation can handle Conjunctive Normal Form (CNF),

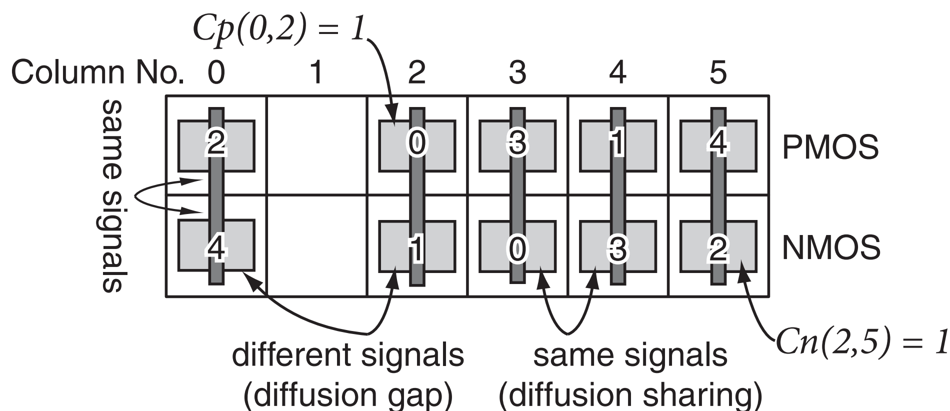


Figure 3.3 The problem definition of the intra-block transistor placement.

Pseudo-Boolean Form (PBF), and optimization constraints[30]. PBF constraints are expressed by linear inequalities of Boolean variables. Since this intra-block transistor placement formulation requires an optimization constraint, we select this formulation.

When N N type transistors and N P type transistors are given, this problem is defined as the problem to place these $2N$ transistors in the minimum area. This problem can be transformed into the problem that places all transistors using the minimum number of columns as illustrated in Figure 3.3. As explained in Section 3.2, the P type transistors are aligned in the upper row and the N type transistors are in the bottom. Two neighboring transistors must face the diffusions which belong to the same signals each other to connect their source or drain by diffusion sharing. The empty columns result in the diffusion gaps in the final layout. These transistor placement constraints are same as those explained in Chapter 2. Under these placement constraints, we search for a possible placement with minimum number of columns of the placement area.

When the intra-block transistor placement is formulated, a new optimization cost is introduced to maximize the number of the connections by diffusion sharing between logic blocks in the subsequent inter-block placement step. The diffusions which can be shared between logic blocks are only power and ground in our partitioning procedure. For maximizing the number of the connections by diffusion sharing in the inter-block placement step, it is better to place the diffusions which belong to VDD and GND on the edge of the placement of each block as illustrated in Figure 3.4. Therefore, we introduce new variables E_l and E_r, and a new optimization function into the formulation of transistor placement to maximize the number of the connections by diffusion sharing. E_l (E_r) takes the value of 1 unless the diffusions of

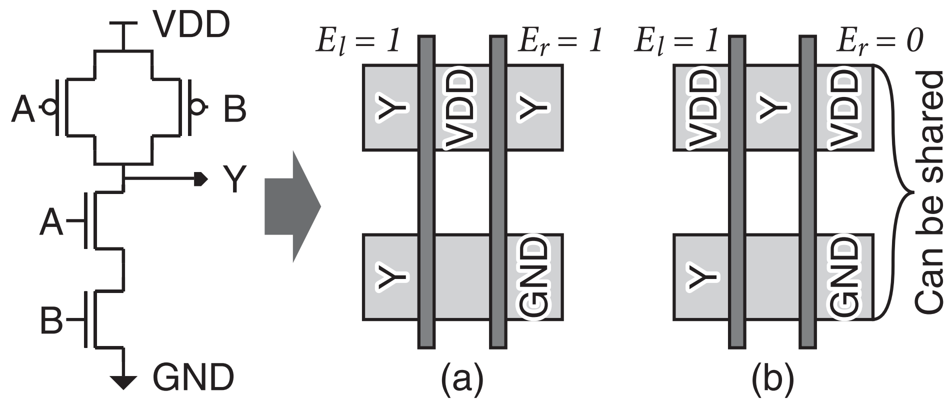


Figure 3.4 Additional variables introduced to maximize the number of the connections by diffusion sharing between logic blocks.

Table 3.2 The variables used for the intra-block transistor placement formulation.

name	*number*	*condition which makes the value 1*
$C_n(i,k)$	$N \times W$	N-FET i is placed in the column k.
$C_p(i,k)$	$N \times W$	P-FET i is placed in the column k.
$F_n(i)$	N	N-FET i is flipped.
$F_p(i)$	N	P-FET i is flipped.
E_r, E_l	2	The diffusions on the right/left edge of the intra-block placement do not belong to VDD and GND.

the left (right) edge are assigned to the pair of VDD and GND as illustrated in Figure 3.4. We explain the details of the placement formulation in the following paragraph.

In our formulation of intra-block transistor placement, $W + 1$ Boolean variables are introduced for each transistor to identify its location and whether it is flipped or not, where W is the number of the columns of the placement area. In addition, 2 Boolean variables are required to identify whether the diffusions of left and right edges of the intra-block placement is a pair of VDD and GND signals or not. All variables required for this formulation are listed in Table 3.2. Table 3.2 shows the names of the variable type, the numbers of each type of variables, and the conditions when each type of variables takes the value of 1. Some example cases of such conditions are also illustrated in Figure 3.3. These variables are used to formulate the transistor placement constraints as Conjunctive Normal Form (CNF) and Pseudo-Boolean Form (PBF) constraints. We formulate the following Boolean constraints which express all the possible transistor placements in W columns under our layout style.

Objective function: For maximizing the number of the connections by diffusion sharing in the inter-block placement step, it is better to place diffusions that belong to VDD and GND on the edge of the placement of each block. Therefore, we introduce the objective function as follows.

$$\text{Minimize :} \quad E_r + E_l \tag{3.1}$$

Transistor overlap constraint: N (P) type transistors must not overlap in the same column. This constraint is expressed by the following PBF constraints.

$$\sum_{i=0}^{N-1} C_n(i,k) \leq 1, \quad 0 \leq k < W \tag{3.2}$$

$$\sum_{i=0}^{N-1} C_p(i,k) \leq 1, \quad 0 \leq k < W \tag{3.3}$$

Transistor instantiation constraint: Each transistor must be instantiated once. The following PBF constraints express this constraint.

$$\sum_{k=0}^{W-1} C_n(i,k) = 1, \quad 0 \leq i < N \tag{3.4}$$

$$\sum_{k=0}^{W-1} C_p(i,k) = 1, \quad 0 \leq i < N \tag{3.5}$$

Different gate constraint: P and N type transistors which have different gate input signals must not be placed in the same column. This constraint is expressed as following PBF constraints.

$$C_n(i,k) + C_p(j,k) \leq 1, \quad 0 \leq k < W \tag{3.6}$$

where i and j meet following condition.

$$0 \leq i < N, \quad 0 \leq j < N, \quad GATE_n(i) \neq GATE_p(j) \tag{3.7}$$

where $GATE_n(i)$ and $GATE_p(j)$ means that the gate input signal of N type transistor i and P type transistor j, respectively.

Neighboring transistors constraint: Two transistors facing the diffusions which belong to the different signals each other can not be placed in the neighboring columns. For example, N type transistors 4 and 1 in Figure 3.3 can not be placed in the neighboring columns. This

constraint is expressed by the following logic equation for N type transistors.

$$GAP_n(i, j) \wedge \bigvee_{k=0}^{W-2} (C_n(i,k) \wedge C_n(j,k+1)) = 0 \qquad (3.8)$$
$$i \neq j, \quad 0 \leq i < N, \quad 0 \leq j < N$$

Here, $GAP_n(i, j)$ is the logic function of $F_n(i)$ and $F_n(j)$, and takes the value of 1 if N type transistor i can not share its diffusion with N type transistor j placed to its immediate right, otherwise 0. This logic equation is expressed as CNF. The same constraint is also expressed as follows for P type transistors.

$$GAP_p(i, j) \wedge \bigvee_{k=0}^{W-2} (C_p(i,k) \wedge C_p(j,k+1)) = 0 \qquad (3.9)$$
$$i \neq j, \quad 0 \leq i < Y, \quad 0 \leq j < Y$$

Finally, we can determine whether given transistors can be placed in W columns or not using these constraints. If no satisfiable assignment is found by the Boolean solver, it is guaranteed that there is no possible placement of W columns. Therefore, we can find an exact minimum-width transistor placement using the procedure described below.

1. For a given transistor netlist, count the number of N and P type transistors. The initial number of column W is set to the number of N type transistors(=#P-FET).

2. Search for a satisfiable assignment for the Boolean constraints constructed for W columns. If a satisfiable assignment is found, these transistors can be placed in W columns and this procedure terminates. Otherwise, go to step 3.

3. $W = W + 1$. Go to step 2 again.

3.3.3 Inter-Block Placement Formulation

After intra-block placement of all the blocks are finished, these blocks are placed in the minimum area. In the proposed method, inter-block placement is also based on Boolean Satisfiability. The inter-block placement problem is formulated as Pseudo-Boolean formulation in the same manner as the intra-block placement. The problem definition of the inter-block placement is shown in Figure 3.5. In the formulation of the inter-block placement, each logic block is placed in one column and two logic blocks must not overlap in the same column. Two logic blocks facing the diffusions that can not be shared each other must not be placed on the neighboring column, as illustrated in Figure 3.5.

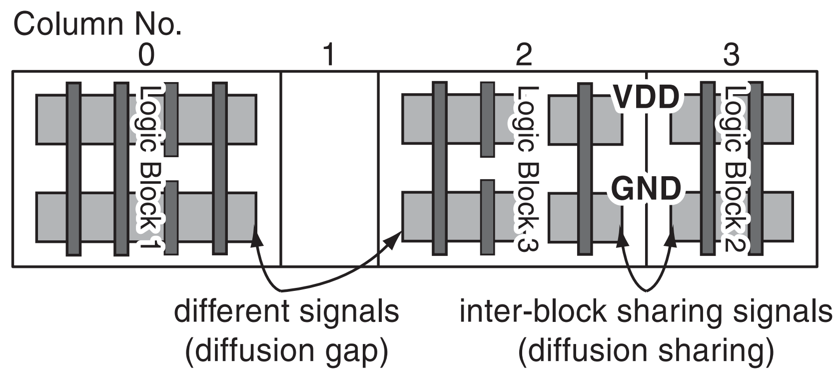


Figure 3.5 The problem definition of the inter-block placement.

These constraints are expressed as CNF and PBF constraints in the same manner as explained in Section 3.3.2. Therefore, we obtain an exact minimum-width block placement using the following procedure similar to that in Section 3.3.2.

1. For a given partitioned block list, count the number of logic blocks. The initial number of column W is set to the number of the blocks.

2. Search for a satisfiable assignment for the Boolean constraints constructed for W columns. If a satisfiable assignment is found, these blocks can be placed in W columns and this procedure terminates. Otherwise, go to step 3.

3. $W = W + 1$. Go to step 2 again.

3.4 Intra-Cell Routing

The routability check of the intra-cell wiring is formulated in the same manner as explained in Chapter 2. Since the routability check consumes extremely less time than the transistor placement procedure, the routability check is executed flatly to the generated transistor placement in the proposed method.

3.5 Overall Flow

Figure 3.6 shows the overall cell layout synthesis flow of the proposed method. In this layout synthesis flow, the routability check is executed once for each block after intra-block transistor placement. Therefore, all blocks are guaranteed to be routed individually before inter-block placement step. After inter-block placement, the routability is checked again

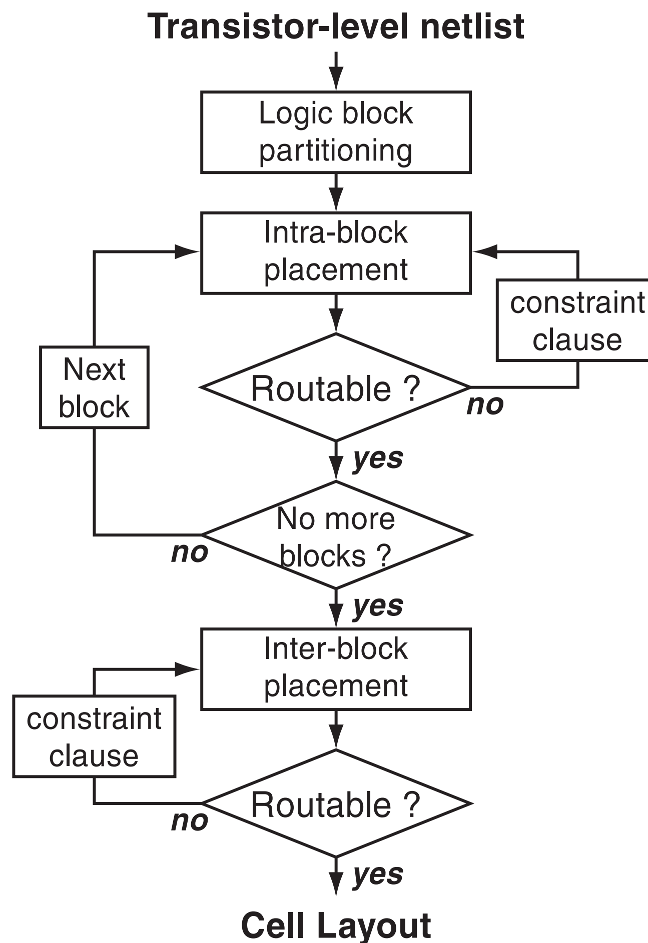


Figure 3.6 Our hierarchical SAT-based cell layout synthesis flow.

for whole transistor placement iteratively until a routable block placement is found. If a placement is found to be unroutable in the routability check point, a constraint clause is added to the set of placement constraints in order not to generate this unroutable placement again.

3.6 Experimental Results

The proposed hierarchical cell layout synthesis method is implemented to show its effectiveness. In this experiment, we used Pseudo-Boolean Solver, *PBS*[30] to solve formulated satisfiability problems. *PBS* is a complete Boolean solver which can handle CNF, PBF, and optimization constraints.

3.6.1 Transistor Placement

First, the comparison results between the proposed hierarchical transistor placement method and the original flat placement method explained in Chapter 2 are shown. In the case of only transistor placement generation, all the routability check in the flow shown in Figure 3.6 is not executed. Table 3.3 summarizes the comparison results. This table shows the circuit name, the number of transistors inside each circuit, the number of columns of the generated placement, and runtime. *Flat* in this table indicates the original flat method and *Hierarchical* indicates the proposed hierarchical method. Although this table shows the results of only 5 circuits, the experiment is conducted for 30 circuits. The row Total(30 circ.) shows the total of 30 circuits. The results show that all the placements generated by the proposed method have the same width as those generated by the flat method. The runtimes are drastically reduced by the hierarchical method particularly in the case of large cells such as fad1 (full adder).

3.6.2 Cell Layout Synthesis

Next, we compared the proposed cell layout synthesis method with a commercial cell layout synthesis tool *ProGenesis*[13] and the original flat synthesis method explained in Chapter 2. In this comparison, the commercial tool only generates the symbolic layout without layout compaction. The comparison results are shown in Table 3.4. This table shows the circuit name, the number of columns of the generated placement, and runtime comparison. *Commercial*, *Flat*, and *Hier.* indicate the commercial tool, the original flat synthesis method, and the proposed hierarchical method, respectively. Although this table also shows the results of only 5 circuits, the comparison is conducted for 30 circuits.

Compared with the original flat method, the proposed method generates 1 column larger cell layout only for mux2 circuit. On the other hand, the runtime of the proposed method is drastically reduced compared with the flat method. These results show that the proposed hierarchical cell layout synthesis method realizes a drastic runtime reduction with little increase in cell width. Figure 3.7 illustrates a layout of a buffered full adder cell generated by the proposed method. This cell has 40 transistors and can not be solved by the original flat method in practical runtime, whereas the proposed method can solve it less than 1 minute.

Cell width comparison with the commercial tool shows that the total cell width of the proposed method is increased about 4%. The reason of this cell width increase is the layout style restriction that only the P and N type transistors which have the same gate input signal

Table 3.3 Comparison results between the proposed hierarchical transistor placement and the original flat method proposed in Chapter 2.

Circuits		*Width (#column)*		*Runtime (sec.)*	
name	*#transistors*	*Flat*	*Hierarchical*	*Flat*	*Hierarchical*
aoi21	6	3	3	0.02	0.03
mux2	12	8	8	0.23	0.13
ao33	16	9	9	0.25	0.06
oa44	20	11	11	0.69	0.04
fad1	28	15	15	201.05	0.28
Total(30 circuits)	—	174	174	205.86	2.49
Ratio	—	1.000	1.000	1.000	0.012

Table 3.4 Comparison results with the commercial cell generation tool and the original flat method proposed in Chapter 2.

Circuits	*Width (#column)*			*Runtime (sec.)*		
name	*Commercial*	*Flat*	*Hier.*	*Commercial*	*Flat*	*Hier.*
aoi21	3	3	3	18.78	0.04	0.06
mux2	*6*	8	**9**	24.23	6.42	1.45
ao33	*9*	10	10	28.12	392.26	2.57
oa44	*11*	12	12	32.00	116.41	4.60
fad1	15	15	15	73.06	305.76	1.24
Total(30 circ.)	172	178	179	789.45	1907.78	19.93
Ratio	1.000	1.035	1.041	1.000	2.416	0.025

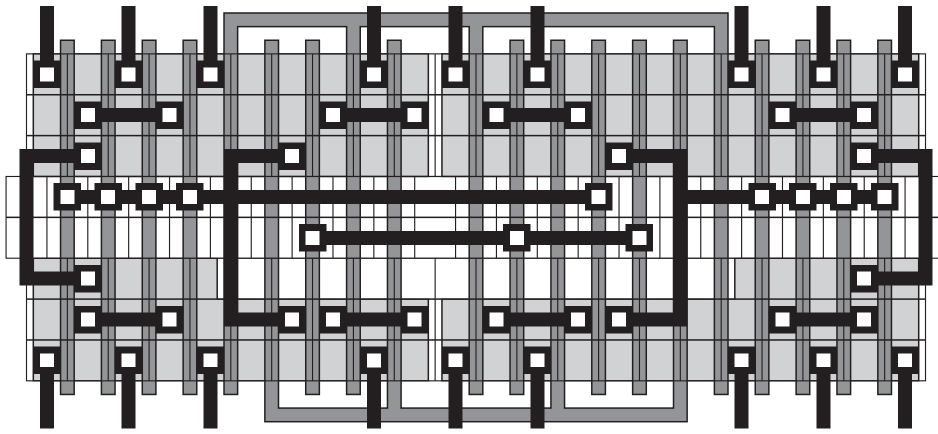


Figure 3.7 A layout of buffered full adder generated by the proposed hierarchical SAT-based cell layout synthesis method.

can be placed in the same column in the proposed method, whereas the commercial tool does not have such restriction. The runtime of the proposed method, however, is only about 3% of that of the commercial tool. These results show that the proposed method can generate the cell layout much more quickly with a little width increase compared with the commercial tool.

3.7 Summary

This chapter proposed a hierarchical layout synthesis method for large dual CMOS cells via Boolean Satisfiability. Experimental results showed that the proposed hierarchical method generates the same width placement as the exact flat method explained in Chapter 2 and drastically reduces the runtime for transistor placement. The comparison results of the cell layout synthesis for 30 benchmark circuits showed that the proposed method generates the same width layout as the flat method except one circuit. The comparison results with the commercial cell generation tool without cell layout compaction showed that the total cell width of the proposed method is increased about 4% due to the layout style restriction, whereas the runtime is only about 3% of that of the commercial tool. From these results, we can conclude that the proposed method can be used as a quick layout generator in the area of transistor-level circuit optimization such as on-demand cell layout synthesis.

Chapter 4

Exact Minimum-Width Transistor Placement for General CMOS Cells

4.1 Introduction

This chapter shows a minimum-width transistor placement method which is applicable to CMOS cells in presence of non-dual P and N type transistors, whereas the cell layout synthesis methods explained in the previous chapters are only for dual cells. This chapter only targets the minimum-width transistor placement, and does not take the intra-cell routings into consideration. The proposed method are the first exact method which can be applied to CMOS cells with any types of structure, whereas almost all of the conventional exact transistor placement method[19, 22, 24] are applicable only to dual CMOS cells. All of these methods make pairs of complementary P and N type transistors and align them in minimum width. Therefore, all these methods can not be applied to some cells that have non-dual P and N type transistors. Even for the dual circuits, these methods can not always generate the minimum-width layouts, since the width depends on pairing of P and N type transistors. Since non-dual CMOS cells occupy a major part of an industrial standard-cell library, the exact minimum-width transistor placement should be applied even to non-dual CMOS cells.

Zhang *et al.*[31] proposed the novel transistor pairing algorithm which is applicable to the cells including non-series-parallel networks. This algorithm decides pairs of P and N type transistors for general complex gates, in which there are more than two transistors with a common input signal in their gates, so that the resulting cell width is minimized. This method, however, is not an exact method and generates large placement when a given cell has a transistor which does not have a pair transistor with the common gate input signal.

We have proposed a cell layout synthesis method using Boolean Satisfiability (SAT) in Chapter 2. This method can generate an exact minimum-width transistor placement for non-

Table 4.1 The numbers of non-dual cells in commercial standard-cell libraries.

cell library	*total number of cells*	*the number of non-dual cells*	*ratio*
130nm A	527	274	52%
130nm B	578	294	51%
90nm A	462	90	19%
90nm B	340	176	52%

series-parallel networks if there is no P and N type transistors which can not be paired by the common gate input signal. However, this method has the pairing restriction same as [31] and is not applicable to the cells including the P and N type transistors which can not be paired by the common gate input signal. This chapter proposes a minimum-width transistor placement method for CMOS cells in presence of non-dual P and N type transistors using SAT.

There are three main contributions of this work. First, an exact transistor placement problem of non-dual CMOS cells is defined for the first time. Table 4.1 lists the numbers of cells including non-dual P and N type transistors in several industrial standard-cell libraries. In a standard-cell library, not only flip flops or three-state buffers but also intrinsically dual CMOS circuits possibly have non-dual P and N type transistors as a result of transistor folding to meet height requirement, as shown in Figure 4.1. As a practical matter, non-dual CMOS cells occupy about a half of an industrial standard-cell library. We propose a Pseudo-Boolean[30] formulation of an exact minimum-width transistor placement problem of non-dual CMOS cells. As explained in Chapter 3, Pseudo-Boolean formulation can handle Conjunctive Normal Form (CNF), Pseudo-Boolean Form (PBF), and optimization constraints. PBF constraints are expressed by linear inequalities of Boolean variables. The proposed method can solve all these non-dual cells that can not be solved by the conventional exact transistor placement methods only for dual cells. In addition, P and N type transistors with common gate input signal are aligned vertically as much as possible for ease of intra-cell routing which follows the transistor placement. Using this formulation, an exact minimum-width transistor placement is generated for the cells to which the conventional exact method only for dual cells is not applicable. Moreover, this method generates a smaller width placement even for a number of dual cells by introducing the pairs of P and N type transistors with different gate input signals.

Second, the hierarchical approach of the transistor placement is also presented for practical

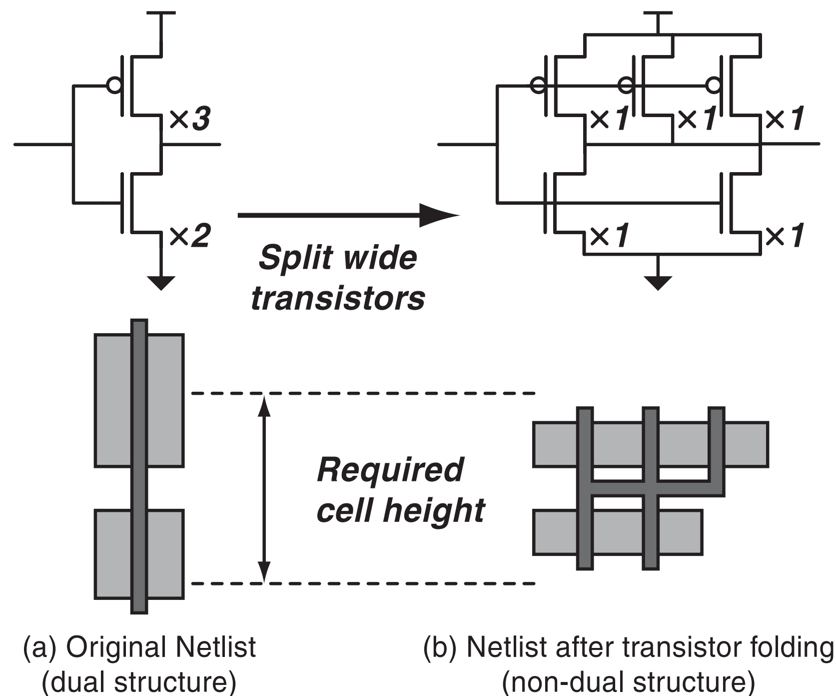


Figure 4.1 An example of a dual CMOS circuit which has a non-dual structure after transistor folding.

use. Although this approach has possibility to generate wider placement than that generated flatly, the experimental results show that it generates the transistor placements with little increase in cell width. It generates a minimum-width transistor placement hierarchically in about one second even for the cells with large number of transistors, whereas the runtimes of our flat approach is over one hour for larger cells with more than 24 transistors.

Third, we generalize the proposed single-row placement method to the multi-row placement and propose an exact minimum-width multi-row transistor placement method for dual and non-dual CMOS cells. The multi-row style layout is essential in several applications such as data paths in which the height of cells is variable while the width of cells is often predetermined. The multi-row style also has flexibility to place large P transistors and small N transistors together and vice versa. Moreover, the multi-row style has other advantages over the single-row style. It offers control over the cell shape and aspect ratio, and can also reduce the intra-cell routing and wire lengths. Therefore, the multi-row style cell layout attracts attention on recent deep sub-micron technologies and a lot of place&route and cell layout synthesis tools begin to support multi-height cells. The proposed method uses an efficient gate connection style and generates more area-efficient transistor placements than the conventional exact minimum-width multi-row style transistor placement method only for dual cells.

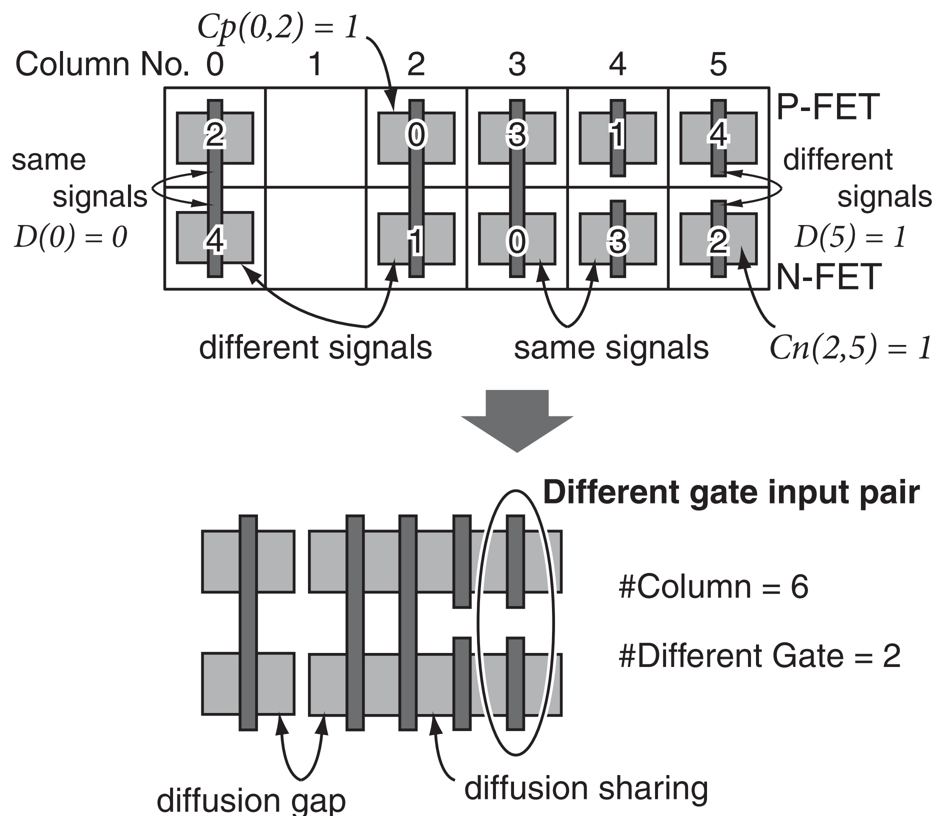


Figure 4.2 The problem definition of the single row transistor placement for non-dual CMOS cells.

Section 4.2 describes the problem definition of the single-row transistor placement method for non-dual CMOS cells. The Boolean formulation of our flat and hierarchical single-row transistor placements are explained in Section 4.3 and Section 4.4, respectively. Next, Section 4.5 generalizes the single-row placement method to multi-row placement. Then, Section 4.6 presents the experimental results and finally Section 4.7 summarizes this chapter.

4.2 Problem Definition

When X N type transistors and Y P type transistors are given, we have to place these $X + Y$ transistors in the minimum area. This problem can be transformed into the problem that places all transistors using the minimum number of columns as illustrated in Figure 4.2. The P type transistors are aligned in the upper row and the N type transistors are in the bottom. Two neighboring transistors face the diffusions that belong to the common signals each other to connect their source or drain by diffusion sharing. The empty columns result in the diffu-

Table 4.2 The variables used for the formulation of the single-row transistor placement for non-dual CMOS cells.

name	*number*	*condition which makes the value 1*
$C_n(i,k)$	$X \times W$	N-FET i is placed in the column k.
$C_p(i,k)$	$Y \times W$	P-FET i is placed in the column k.
$F_n(i)$	X	N-FET i is flipped.
$F_p(i)$	Y	P-FET i is flipped.
$D(k)$	W	Different gate input pair is placed in the column k.

sion gaps in the final layout as illustrated in Figure 4.2. The P and N type transistors placed in the same column form a transistor pair. When a pair of P and N type transistors have different gate input signals, this pair is called a *different gate input pair*. For example, P type transistor 4 and N type transistor 2 in Figure 4.2 form a *different gate input pair*. If a column has only a P or an N type transistor, this transistor is also called a *different gate input pair*. Under these layout styles, the transistor placement problem is transformed into the Conjunctive Normal Form (CNF) and Pseudo-Boolean Form (PBF) constraints. We will explain two approaches for generating a minimum-width single-row transistor placement in following sections. First, the flat approach which generates an exact minimum-width placement of any types of CMOS cells will be explained. Next, we will explain the hierarchical approach which reduces the runtime drastically with little increase in cell width.

4.3 Flat Single-Row Transistor Placement

In our formulation of the flat approach, $W + 1$ variables are introduced for each transistor to identify its location and whether it is flipped or not, where W is the number of the columns of the placement area. Additional W variables are needed to identify whether the *different gate input pair* is placed in each column or not. All variables needed for this formulation are listed in Table 4.2. Table 4.2 shows the names of the variable type, the numbers of each type of variables, and the conditions when each type of variables takes the value of 1. Some example cases of such conditions are also illustrated in Figure 4.2. We formulate the following Boolean constraints which express all the possible transistor placements in W columns under our layout style.

Transistor overlap constraint: N type transistors must not overlap in the same column. This constraint is expressed by the following PB constraints.

$$\sum_{i=0}^{X-1} C_n(i,k) \leq 1, \quad 0 \leq k < W \tag{4.1}$$

The same constraint is expressed as PB constraints for P type transistors.

$$\sum_{i=0}^{Y-1} C_p(i,k) \leq 1, \quad 0 \leq k < W \tag{4.2}$$

Transistor instantiation constraint: Each transistor must be instantiated once. The following PB constraints express this constraint.

$$\sum_{k=0}^{W-1} C_n(i,k) = 1, \quad 0 \leq i < X \tag{4.3}$$

$$\sum_{k=0}^{W-1} C_p(i,k) = 1, \quad 0 \leq i < Y \tag{4.4}$$

Neighboring transistors constraint: Two transistors facing the diffusions which belong to the different signals each other can not be placed in the neighboring columns. For example, N type transistors 4 and 1 in Figure 4.2 can not be placed in the neighboring columns. This constraint is expressed by the following logic equation.

$$GAP_n(i,j) \wedge \bigvee_{k=0}^{W-2} (C_n(i,k) \wedge C_n(j,k+1)) = 0 \tag{4.5}$$
$$i \neq j, \quad 0 \leq i < X, \quad 0 \leq j < X$$

Here, $GAP_n(i,j)$ is the logic function of $F_n(i)$ and $F_n(j)$, and takes the value of 1 if N type transistor i can not share its diffusion with N type transistor j placed to its immediate right, otherwise 0. This logic equation is expressed as CNF. The same constraint is also expressed as follows for P type transistors.

$$GAP_p(i,j) \wedge \bigvee_{k=0}^{W-2} (C_p(i,k) \wedge C_p(j,k+1)) = 0 \tag{4.6}$$
$$i \neq j, \quad 0 \leq i < Y, \quad 0 \leq j < Y$$

Objective function: The number of the *different gate input pairs* is minimized for ease of intra-cell routing which follows the transistor placement. The objective function is expressed as follows.

$$Minimize: \quad \sum_{k=0}^{W-1} D(k) \tag{4.7}$$

Definition of $D(k)$: When a *different gate input pair* is placed in the column k, the value of $D(k)$ must be 1. This definition is expressed by the following logic equation,

$$D(k) = \bigvee_{i,j} (C_n(i,k) \wedge C_p(j,k)), \quad (4.8)$$
$$0 \leq k < W$$
$$GATE_n(i) \neq GATE_p(j), \quad (4.9)$$
$$0 \leq i < X, \;\; 0 \leq j < Y$$

where $GATE_n(i)$ and $GATE_p(j)$ mean the gate input signals of N type transistor i and P type transistor j, respectively. This logic equation is transformed into the PB constraints as follows for all combinations of i and j expressed by (4.9)

$$C_n(i,k) + C_p(j,k) \leq D(k) + 1, \quad (4.10)$$
$$0 \leq k < W$$

under the condition that $D(k)$ is minimized. The minimization condition of each $D(k)$ value is already expressed by the objective function (4.7). Therefore, this definition is simply expressed by the PB constraints (4.10) for all combinations of i and j expressed by (4.9). When the column k has only a P or an N type transistor, $D(k)$ also takes a value of 1.

Finally, we can determine whether given transistors can be placed in W columns or not using these constraints. If no satisfiable assignment is found by the Boolean solver, it is guaranteed that there is no possible placement of W columns. Therefore, we can find an exact minimum-width transistor placement using the procedure described below.

1. For a given transistor netlist, count the number of N and P type transistors. The initial number of column W is set to max(#N-FET, #P-FET).

2. Search for a satisfiable assignment of the Boolean constraints constructed for W columns. If a satisfiable assignment is found, these transistors can be placed in W columns and this procedure terminates. Otherwise, go to step 3.

3. $W = W + 1$. Go to step 2 again.

4.4 Hierarchical Single-Row Transistor Placement

As explained in Section 4.1, the flat approach can not solve the cells with large number of transistors in reasonable time. Therefore, we also propose a hierarchical approach of our

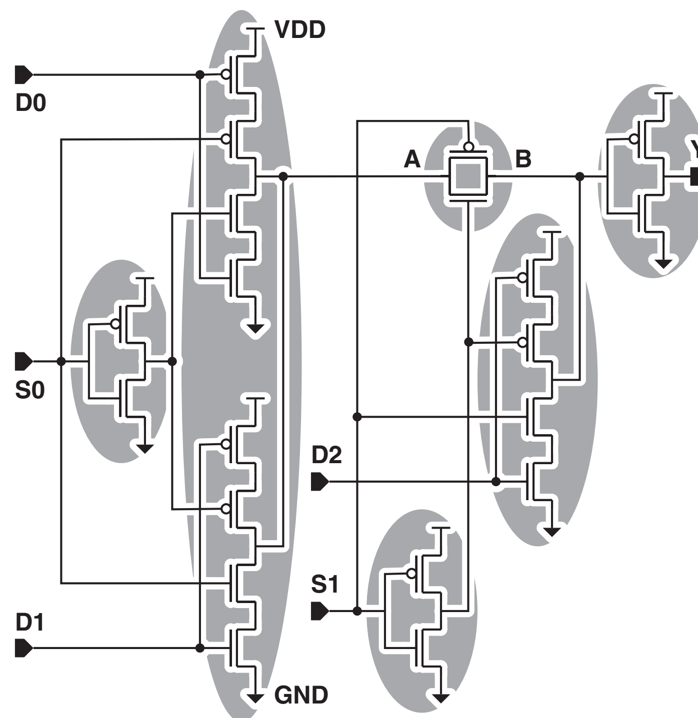


Figure 4.3 Schematic of 3-input multiplexer and its logic block partitioning.

transistor placement method based on circuit partitioning for practical use. The hierarchical transistor placement procedure used in this method is similar to that of Chapter 3 and also composed of partitioning, intra-block placement, and inter-block placement. However, the partitioning and intra-block placement procedures have some differences to handle non-dual CMOS cells effectively. The details of these procedures are explained in the following paragraphs.

Partitioning: Our approach first partitions the given cell into logic blocks. A logic block consists of transistors that are connected together by their diffusions except power and ground. We also recognize a transmission gate as a logic block in this partitioning procedure. Figure 4.3 shows an example of 3-input multiplexer circuit. A transmission gate is identified as an individual logic block. Two clocked inverters whose outputs are connected are identified as one logic block using our partitioning procedure.

Intra-block placement: Next, intra-block placement of each logic block is performed using the procedure explained in Section 4.3. In this stage, an exact minimum-width transistor placement of each block is generated. When Pseudo-Boolean constraints are generated, one new optimization cost is introduced to maximize the number of the connections by diffusion sharing between logic blocks in the inter-block placement step. The diffusions which can be

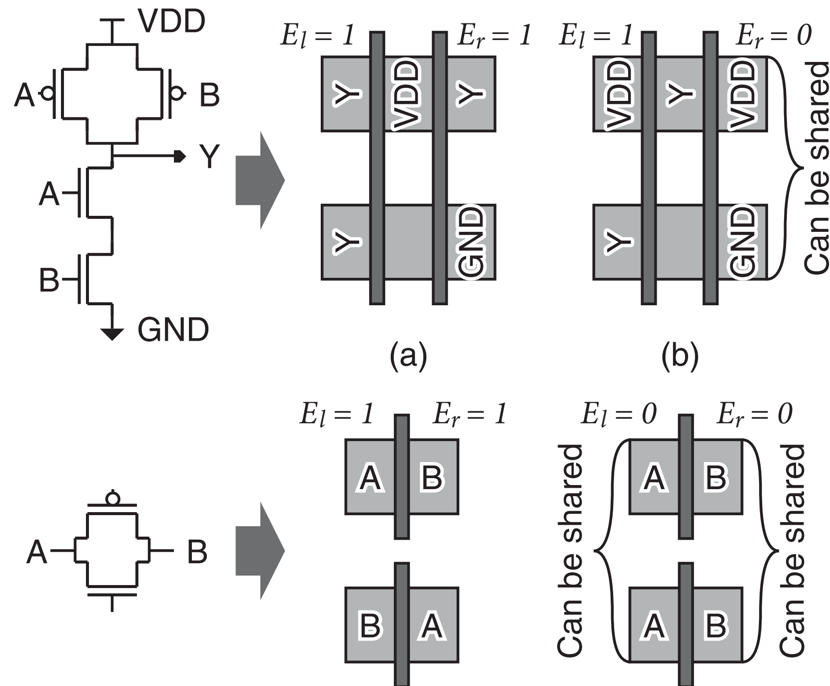


Figure 4.4 Additional variables introduced to maximize the number of the connections by diffusion sharing between logic blocks.

shared between logic blocks are power, ground, and the diffusions of the transmission gates. For example of the Figure 4.3, signal name "VDD", "GND", "A", and "B" can be shared by two blocks. These signals are called *inter-block sharing signals* hereafter. For maximizing the number of the connections by diffusion sharing in the inter-block placement step, it is better to place the diffusions that belong to the inter-block sharing signals on the edge of the placement of each block as illustrated in Figure 4.4. Therefore, we introduce new variables E_l and E_r, and a new objective function into the formulation of transistor placement to maximize the number of the connections by diffusion sharing. E_l (E_r) takes the value of 1 unless the diffusions of the left (right) edge are assigned to the pair of the inter-block sharing signals as illustrated in Figure 4.4. The new objective function is expressed as follows.

$$Minimize : (W + 1) \times (E_l + E_r) + \sum_{k=0}^{W-1} D(k) \tag{4.11}$$

In our formulation, the number of the connections by diffusion sharing between logic blocks has the higher priority than the number of the *different gate input pairs*. Therefore, E_l and E_r have the coefficient $W + 1$ to have larger cost than the *different gate input pairs*.

Inter-block placement: Inter-block placement is also based on Boolean Satisfiability. The constraints explained in Section 4.3 except "different gate constraints" and the "objective

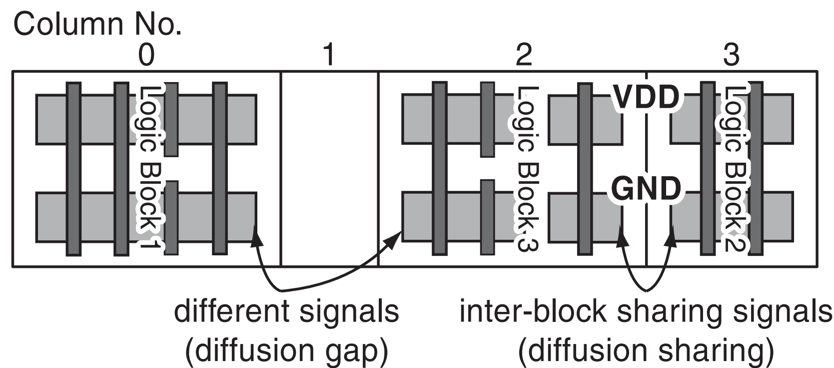


Figure 4.5 The problem definition of the inter-block placement.

function" are needed for the formulation of the inter-block placement. The problem definition of the inter-block placement is shown in Figure 4.5. In the formulation of the inter-block placement, each logic block is placed in one column and two logic blocks must not overlap in the same column. Two logic blocks facing the diffusions that can not be shared each other must not be placed in the neighboring column, as illustrated in Figure 4.5.

These constraints are expressed as CNF and PBF constraints in the same manner as explained in Section 4.3. Therefore, we obtain an exact minimum-width block placement using the same procedure as explained in Section 4.3. Finally, the flow of our hierarchical approach for transistor placement is described as follows.

1. A given transistor netlist is partitioned into logic blocks.
2. Exact minimum-width transistor placements are generated for each block using the new objective function and the same flow as explained in Section 4.3.
3. After placements of all blocks are generated, count the number of the logic blocks. The initial number of the column W is set to the number of logic blocks.
4. Search for a satisfiable assignment of the Boolean constraints for the inter-block placement of W columns. If a satisfiable assignment is found, these logic blocks can be placed in W columns and this procedure terminates. Otherwise, go to step 5.
5. $W = W + 1$. Go to step 4 again.

Although this approach has a possibility to generate wider placement than that of the flat approach, the experimental results show that it generates the transistor placements quickly with little increase in cell width.

4.5 Generalization to Multi-Row Transistor Placement

In this section, we generalize the single-row placement method for dual and non-dual CMOS cells explained previously to the multi-row placement and propose an exact minimum-width multi-row transistor placement method for dual and non-dual CMOS cells.

4.5.1 Problem Definition

When X N type transistors, Y P type transistors, and the number of the P/N row R are given, we have to place these $X + Y$ transistors in the minimum width. This problem can be transformed into the problem which places all transistors using the minimum number of columns. Figure 4.6 illustrates the problem definition of the proposed multi-row transistor placement. *Column*, *row*, and *boundary* are defined as shown in Figure 4.6 (a), respectively. Although we assume two P/N rows and these rows abut their P transistor side each other (so called NPPN style) in this figure, the number of P/N rows and their directions can be changed. Moreover, our formulation can be extended to handle other styles, *e.g.*, NPN or PNP styles, and so on. Two horizontally neighboring transistors face the diffusions that belong to the common signals each other to connect their source or drain by diffusion sharing. The empty columns result in the diffusion gaps in the final layout as illustrated in Figure 4.6 (b). When two vertically neighboring transistors have different gate input signals, these gate signals can not be connected. This point is called a *different gate* as shown in Figure 4.6 (b). *Different gate* is defined on the *boundary*. If one of the vertically neighboring grids has a transistor and the other does not, this point is also called a *different gate*. The conventional multi-row transistor placement method does not consider the gate connection between two P/N rows. Our gate connection style is a key to handling the multi-row placement of CMOS cells including non-dual P and N type transistors efficiently. Under these layout styles, the multi-row transistor placement problem is transformed into the Conjunctive Normal Form (CNF) and Pseudo-Boolean Form (PBF) constraints.

4.5.2 Placement Formulation

In our formulation of the multi-row transistor placement, $C \times R + 1$ variables are introduced for each transistor to identify its location and whether it is flipped or not, where C and R are the number of columns and rows of the placement area, respectively. The value of R is assumed to be given as an input. Additional $C \times (2 \times R - 1)$ variables are needed to identify

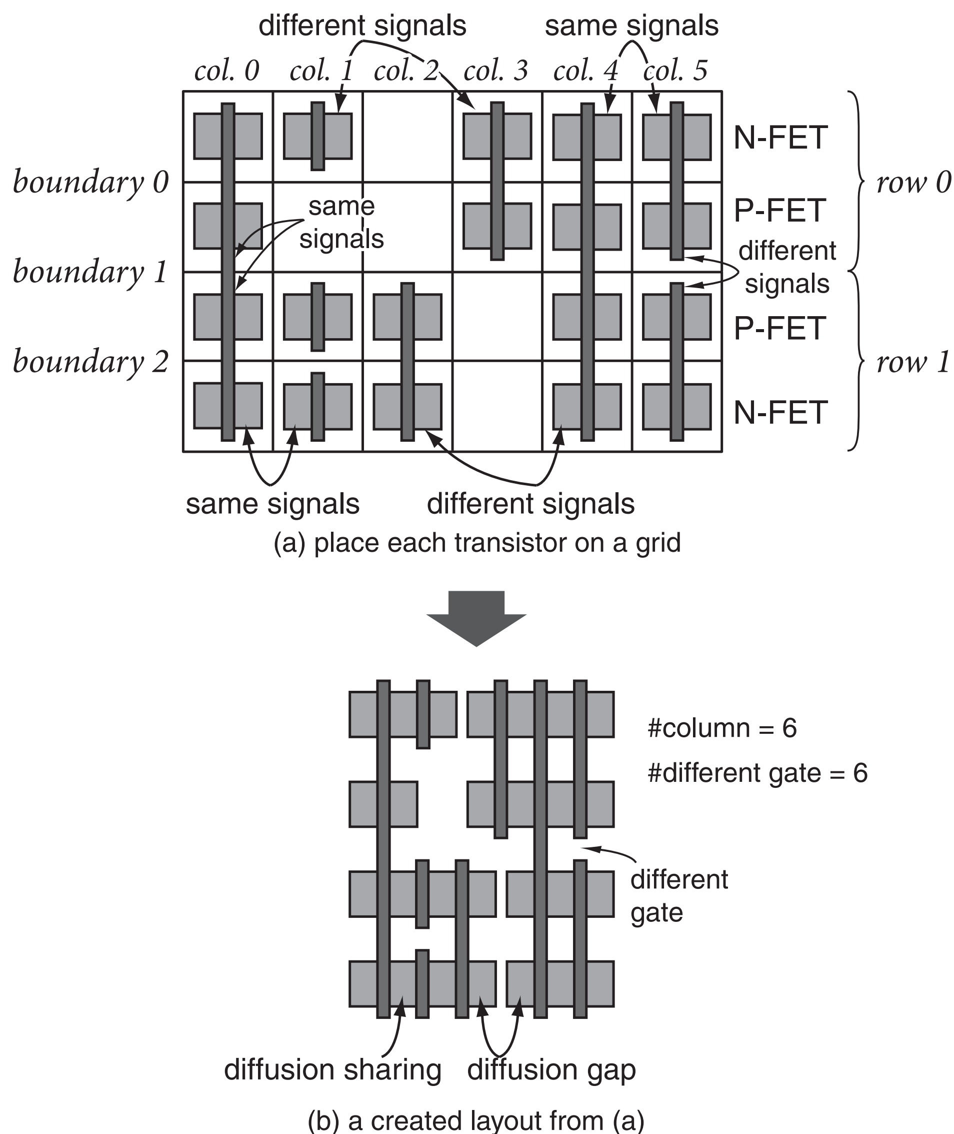


Figure 4.6 The problem definition of the multi-row transistor placement for dual and non-dual CMOS cells.

whether there is a *different gate* on each *boundary* of vertically neighboring grids or not. The number of the *different gate* is minimized in our formulation for ease of intra-cell routing which follows the transistor placement. All variables needed for this formulation are listed in Table 4.3. Table 4.3 shows the names of the variable type, the numbers of each type of variables, and the conditions when each type of variables takes the value of 1. We formulate the following Boolean constraints which express all the possible transistor placements in C columns and R rows under our layout style.

Table 4.3 The variables used for the formulation of the multi-row placement.

name	*number*	*condition which makes the value 1*
$L_n(i,k,l)$	$X \times C \times R$	N-FET i is placed in the column k, row l.
$L_p(j,k,l)$	$Y \times C \times R$	P-FET j is placed in the column k, row l.
$F_n(i)$	X	N-FET i is flipped.
$F_p(j)$	Y	P-FET j is flipped.
$D(k,l)$	$C \times (2 \times R - 1)$	Different gate is placed in the column k, boundary l.

Transistor overlap constraint: N type transistors must not overlap in the same *column* and the same *row*. This constraint is expressed by the following PBF constraints.

$$\sum_{i=0}^{X-1} L_n(i,k,l) \leq 1, \quad 0 \leq k < C, \quad 0 \leq l < R \tag{4.12}$$

The same constraint is expressed as PBF constraints for P type transistors.

$$\sum_{j=0}^{Y-1} L_p(j,k,l) \leq 1, \quad 0 \leq k < C, \quad 0 \leq l < R \tag{4.13}$$

Transistor instantiation constraint: Each transistor must be instantiated once. The following PBF constraints express this constraint.

$$\sum_{k=0}^{C-1} \sum_{l=0}^{R-1} L_n(i,k,l) = 1, \quad 0 \leq i < X \tag{4.14}$$

$$\sum_{k=0}^{C-1} \sum_{l=0}^{R-1} L_p(j,k,l) = 1, \quad 0 \leq j < Y \tag{4.15}$$

Neighboring transistors constraint: Two N type transistors facing the diffusions which belong to the different signals each other can not be placed in the horizontally neighboring grids. This constraint is expressed by the following logic equation.

$$GAP_n(i,j) \wedge \bigvee_{k=0}^{C-2} (L_n(i,k,l) \wedge L_n(j,k+1,l)) = 0 \tag{4.16}$$
$$i \neq j, \quad 0 \leq i < X, \quad 0 \leq j < X, \quad 0 \leq l < R$$

Here, $GAP_n(i,j)$ is the logic function of $F_n(i)$ and $F_n(j)$, and takes the value of 1 if N type transistor i can not share its diffusion with N type transistor j placed to its immediate right, otherwise 0. This logic equation is expressed as CNF. The same constraint is also expressed

as follows for P type transistors.

$$GAP_p(i, j) \wedge \bigvee_{k=0}^{C-2} (L_p(i,k,l) \wedge L_p(j,k+1,l)) = 0 \tag{4.17}$$
$$i \neq j,\ \ 0 \le i < Y,\ \ 0 \le j < Y,\ \ 0 \le l < R$$

Objective function: The number of the *different gate* is minimized for ease of intra-cell routing which follows the transistor placement. The objective function is expressed as follows.

$$Minimize: \sum_{k=0}^{C-1} \sum_{l=0}^{2R-2} D(k,l) \tag{4.18}$$

Definition of $D(k,l)$: When a *different gate* is placed in the *column k*, *boundary l*, the value of $D(k,l)$ must be 1. Inside *row* $r(= l/2)$, this definition is expressed by the following logic equation,

$$D(k,l) = \bigvee_{i,j} (L_n(i,k,r) \wedge L_p(j,k,r)), \tag{4.19}$$
$$0 \le k < C,\ \ 0 \le r < R$$
$$GATE_n(i) \neq GATE_p(j),\ \ 0 \le i < X,\ \ 0 \le j < Y \tag{4.20}$$

where $GATE_n(i)$ and $GATE_p(j)$ mean the gate input signals of N type transistor i and P type transistor j, respectively. This logic equation is transformed into the PBF constraints as follows for all combinations of i and j expressed by (4.20)

$$L_n(i,k,r) + L_p(j,k,r) \le D(k,l) + 1, \tag{4.21}$$
$$0 \le k < C,\ \ 0 \le r < R$$

under the condition that $D(k,l)$ is minimized. The minimization condition of each $D(k,l)$ value is already expressed by the objective function (4.18). Therefore, this definition is simply expressed by the PBF constraints (4.21) for all combinations of i and j expressed by (4.20). When the *column k*, *row r* has either a P or an N type transistor, $D(k,l)$ also takes a value of 1. $D(k,l)$ is defined not only inside each P/N row (*e.g.*, boundaries 0 and 2 in Figure 4.6 (a)), but also on the boundary between two P/N rows (*e.g.*, boundary 1 in Figure 4.6 (a)) in similar manner.

Finally, we can determine whether given transistors can be placed in C columns and R rows or not, using these constraints. If no satisfiable assignment is found by a Boolean

constraint solver, it is guaranteed that there is no possible placement of *C* columns and *R* rows. Therefore, we can find an exact minimum-width multi-row transistor placement using the procedure described below.

1. A netlist and the number of rows *R* are given.

2. For a given transistor netlist, count the number of N and P type transistors. The initial number of column *C* is set to $\lceil max(\#\text{N-FET}, \#\text{P-FET}) / R \rceil$[4.1].

3. Search for a satisfiable assignment of the Boolean constraints constructed for *C* columns and *R* rows. If a satisfiable assignment is found, these transistors can be placed in the *C* columns and *R* rows, and this procedure terminates. Otherwise, go to step 4.

4. $C = C + 1$. Go to step 3 again.

4.6 Experimental Results

The proposed flat and hierarchical single-row transistor placement methods and multi-row transistor placement method were implemented to show the effectiveness of the proposed methods. In this experiment, we used *PBS*[30] for a Boolean constraint solver. *PBS* can handle CNF constraints, PBF constraints, and optimizations. The results of each placement method are shown in the following sections.

4.6.1 Single-Row Flat Approach

The characteristics of the cells used for the experiment of single-row placement methods are shown in Table 4.4. Tables 4.5 and 4.6 show the comparison with the exact transistor placement method for dual cells explained in Chapter 2, which is applicable to the cells that can not be applied to other conventional exact methods such as [22] and [19]. Moreover, it theoretically generates the same or smaller width placement than that of other existing exact transistor placement methods for dual cells.

Table 4.5 compares the problem sizes of each formulation, and Table 4.6 compares the numbers of columns of generated placement, and the total runtimes of each method for generating the minimum-width transistor placement from a netlist. In Table 4.5, *#variable*, *#clause*, and *#inequality* mean the number of variables, clauses, and inequalities of each

[4.1] $\lceil X \rceil$ indicates a minimum integer which is equal to or larger than X.

Table 4.4 Cells used for the experiment of the single-row transistor placement.

Cell No.	*Circuit*	*#transistor*
1	Series-parallel circuit for $Y = (A \vee B) \wedge C$	10
2	Three-state buffer	12
3	2-input multiplexer	14
4	2-input AND	15
5	Series-parallel circuit for $Y = (A \wedge B) \oplus C$	18
6	3-input exclusive OR	20
7	3-input exclusive NOR	22
8	D Flip Flop	24
9	D Flip Flop (buffered)	26
10	3-input exclusive NOR (buffered)	28

formulation, respectively. Since both methods are based on the Boolean satisfiability, the problem size of each formulation is also compared in this table. In Table 4.6, *#column* and *#different* mean the number of columns and the number of the *different gate input pairs* of generated placement, respectively. The value of *#different* for the placement method only for dual cells is always 0, since it always makes pairs of P and N type transistors with a common gate input signal. The proposed method used *PBS* for a Boolean constraint solver since it creates CNF, PBF, and optimization constraints, whereas the placement method for dual cells used *Chaff*[25] for a CNF-SAT solver since it creates only CNF constraints. Since *PBS* is proposed to generalize the algorithms used in *Chaff* to solve 0-1 ILP problems that may include PBF and optimization constraints, *Chaff* finds solutions more quickly to the problems of pure CNF constraints. For some cells which have more than 20 transistors, we have found that *Chaff* can find solutions much more quickly to the problems created by the placement method for dual cells than PBS. Therefore, the placement method for dual cells used *Chaff* in this experiment.

The cell number 2 (three-state buffer) has transistors which can not be paired by the common gate input signals. Although the cell number 4 (2-input AND) is an intrinsically dual CMOS circuit, this cell also has non-dual P and N type transistors as a result of transistor folding. Therefore, the placement method only for dual cells can not be applied to these cells, whereas the proposed flat approach generates the exact minimum-width transistor placements of them as shown in Table 4.6. In the case of the cell number 10, the runtime was over 3600

Table 4.5 Problem size comparison results between the exact single-row transistor placement method for dual cells and the proposed flat approach.

Cell No.	*Problem Size*				
	Dual		*Proposed*		
	#variable	*#clause*	*#variable*	*#inequality*	*#clause*
1	40	760	65	121	488
2	N/A	N/A	103	251	1104
3	70	2522	103	251	1104
4	N/A	N/A	143	336	1960
5	90	4588	208	699	4212
6	100	6412	251	1011	6320
7	110	8478	321	1427	9120
8	144	17122	349	1163	11040
9	130	13070	404	2127	14170
10	140	15516	434	2465	15860

Table 4.6 Width and runtime comparison results between the exact single-row transistor placement method for dual cells and the proposed flat approach.

Cell No.	*Width*			*Runtime (sec.)*	
	Dual	*Proposed*		*Dual*	*Proposed*
	#column	*#column*	*#different*		
1	6	**5**	2	0.03	0.04
2	N/A	7	2	N/A	0.15
3	9	**7**	2	0.86	0.17
4	N/A	8	1	N/A	0.49
5	10	10	0	1.66	2.03
6	14	**11**	2	55.82	29.06
7	15	**13**	2	159.63	969.96
8	18	**13**	4	1382.98	1240.83
9	—	—	—	>3600	>3600
10	16	—	—	1730.13	>3600

— : the procedure does not terminate after 3600 seconds

Table 4.7 Comparison results between our flat and hierarchical approaches of the single-row transistor placement.

Cell No.	*Width*				*Runtime (sec.)*	
	Flat		*Hierarchical*		*Flat*	*Hierarchical*
	#column	*#different*	*#column*	*#different*		
1	5	2	**6**	0	0.04	0.02
2	7	2	7	2	0.15	0.02
3	7	2	7	2	0.17	0.14
4	8	1	8	1	0.49	0.03
5	10	0	10	**2**	2.03	0.08
6	11	2	11	**4**	29.06	0.29
7	13	2	13	**4**	969.96	3.39
8	13	4	13	4	1240.83	0.31
9	—	—	15	4	>3600	1.21
10	—	—	16	4	>3600	1.85

— : the procedure does not terminate after 3600 seconds

seconds for the flat approach, and that of both the flat approach and the placement method for dual cells were over 3600 seconds in the case of the cell number 9. In the other cases, the proposed method generated the same or smaller width placements by introducing *different gate input pairs*. Although the intra-cell routing may become difficult by introducing the *different gate input pairs*, it is easy to complete it if using the second metal layer. The runtimes of the proposed flat approach were comparable to the exact method for dual cells for cells with about 20 transistors, but the runtimes became over one hour for cells with about 26 transistors whereas the dual method could solve the cells with about 28 transistors in one hour under our experimental setup. The next experimental results show that these runtimes can be reduced drastically using the hierarchical approach.

We also compared the proposed transistor placement method for non-dual cells with the commercial cell generation tool *ProGenesis*[13] for the “mux2” circuit which the transistor placement method for dual cells proposed in Chapter 2 generates larger width placements than the commercial tool as shown in Table 2.4. The method in Chapter 2 generates the larger width placement because of the gate connection restriction between P and N type transistors, and generates the placement of 8 columns whereas the commercial tool generates the

6 column placement. The transistor placement proposed in this chapter has no restriction in the gate connection between P and N type transistors and the experimental result demonstrated that the proposed method generates the 6 column placement for this circuits. From this result, we can conclude that the proposed transistor placement method considers the gate connection between P and N type transistors more efficiently than the placement method for dual cells, and generates more area-efficient transistor placement.

4.6.2 Single-Row Hierarchical Approach

Table 4.7 shows the comparison result between our flat and hierarchical approaches. This table compares the numbers of the columns and the *different gate input pairs* of the generated placements, and the total runtimes. The width of the generated placement increased in the case of the cell number 1, and the numbers of different gate input pairs increased in the cases of the cells number 5, 6, and 7. This table clearly shows that the runtimes for transistor placement was drastically reduced by our hierarchical approach with little increase in cell width. Moreover, this hierarchical approach could solve the cells with larger number of transistors. The solutions for the cells number 9 and 10 were generated within two seconds by the hierarchical approach, whereas the flat approach could not generate the solutions within one hour.

Table 4.8 compares the numbers of cells that can be applied to the exact transistor placement method for dual cells explained in Chapter 2 and the proposed flat and hierarchical approaches in the case of an industrial standard-cell library of a 90nm technology including 340 cells. This table shows the numbers of the cells and coverage ratios of the cells that are applicable to each method and that can be solved in 3600 seconds. Since the proposed method is applicable to CMOS cells with any types of structures, this method was applied to all of the 340 cells, whereas the method for dual cells was about a half. The coverage ratios of the cells that can be solved in 3600 seconds by the flat approach and the method for dual cells are 43% and 32%, respectively. The number of the cells solved by the flat approach was larger than that of the dual one since the proposed method can be applied to non-dual circuits as well. Among the cells solved by both methods, the proposed flat approach generated smaller width placements for 29 out of 103 dual cells by introducing the *different gate input pairs*. Using the hierarchical approach, the coverage ratio increased to 81%, and only 3 out of 147 cells have one column wider placement. Among the 144 cells whose width are kept minimum, only 16 cells have larger number of *different gate input pairs* than the placement

Table 4.8 Comparison of the number of cells that can be applied to the proposed exact single-row transistor placement method and the method for dual cells in the case of a cell library with 340 cells.

placement method	*applicable*		*solvable*	
	#cells	*coverage*	*#cells*	*coverage*
Dual	164	48%	110	32%
Proposed (flat)	340	100%	147	43%
Proposed (hierarchical)	340	100%	274	81%

Table 4.9 Cells used for the experiment of the multi-row transistor placement.

Cell No.	*Circuit*	*#transistor*
1	Series-parallel circuit for $(A0 \vee A1) \oplus B$	12
2	Half-adder	14
3	Series-parallel circuit for $(A0 \vee A1) \oplus B$ (buffered)	18
4	D latch	19
5	3-input XNOR	20
6	2-input multiplexer	21
7	3-input XNOR (buffered)	22
8	D flip flop	24
9	4-input multiplexer	26
10	Full adder	28

generated by the flat approach. The rest of 19% cells can not be solved within one hour by the proposed hierarchical approach since these cells have logic blocks with large number of transistors even after the partitioning procedure. We expect the hierarchical approach can solve all of the cells by improving the partitioning algorithm to partition the large logic blocks into multiple blocks. Although the total time for generating 81% of 340 cells was about 7 hours, a large part of this time was consumed by several cells. Except these time-consuming cells, 76% of 340 cells were solved in 30 minutes. This result also shows that our hierarchical approach reduced the runtimes drastically with little increase in cell width.

4.6.3 Multi-Row Placement

We used another set of 10 CMOS cells from a standard-cell library of 90nm technology for the experiment of multi-row transistor placement. The characteristics of the cells used in this experiment are summarized in Table 4.9. Table 4.10 summarizes the problem size of the proposed multi-row transistor placement formulation and Table 4.11 shows the comparison results with the conventional style multi-row placement method[19]. In the conventional style, only P and N type transistors with the same gate input signal can be placed in the same column in each P/N row, and the gate connection between two adjacent P/N rows is not considered. This conventional style placement method was also formulated in the same manner as the proposed method for comparison in this experiment, although the original method[19] was solved using ILP. We assumed two P/N rows and NPPN placement style for both methods.

In Table 4.10, *#variable*, *#clause*, and *#inequality* mean the number of variables, clauses, and inequalities of the proposed formulation, respectively. Table 4.11 shows the numbers of columns of generated placement, and the total runtimes of each method for generating the minimum-width transistor placement from a netlist. In the column of width, *#column* and *#different* mean the number of columns and the number of the *different gate* of generated placement, respectively. Since the conventional style does not consider the gate connection between two adjacent rows, this table does not show the number of *different gate* for the conventional style. However, it is notable that the gates of the P/N transistor pair inside each P/N row are always connected in the conventional style.

The cells number 4 and 6 in Table 4.9 have non-dual structure and have transistors which can not be paired by the common gate input signals. Therefore, the conventional style can not be applied to these cells, whereas the proposed method generates the exact minimum-width transistor placements of them as shown in Table 4.11. In the case of the cell number 10, the runtime was over 3600 seconds for the proposed method. In the other cases, the proposed method generated the same or smaller width placements by using our gate connection style. The runtimes of the proposed method were comparable to the case of using conventional style for the cells with up to 22 transistors and much shorter for the cells number 8 and 9, but the runtime became over one hour for the cell with 28 transistors whereas the conventional style could solve this cell within one hour under our experimental setup. Although the runtime depends not only on the number of the transistors but also on the connections inside the cell, these results show that the proposed method can solve the cells with up to 26 transistors in

Table 4.10 Problem size of the proposed multi-row transistor placement formulation.

Cell No.	*#transistor*	*Problem Size*		
		#variable	*#inequality*	*#clause*
1	12	93	294	504
2	14	138	486	1200
3	18	213	1028	2496
4	19	265	1399	3460
5	20	278	1628	3560
6	21	291	1653	4700
7	22	304	1954	4480
8	24	330	2280	5360
9	26	411	3162	6816
10	28	441	7344	3796

Table 4.11 Width and runtime comparison results of the proposed multi-row transistor placement method. The conventional style assumes the pair of P/N transistors with the same gate input signals.

Cell No.	*#transistor*	*Width*			*Runtime (sec.)*	
		Conventional	*Proposed*		*Conventional*	*Proposed*
		#column	*#column*	*#different*		
1	12	4	**3**	4	0.13	0.04
2	14	4	4	2	0.21	0.21
3	18	5	5	2	1.47	1.81
4	19	N/A	6	8	N/A	90.30
5	20	7	**6**	5	37.67	87.99
6	21	N/A	6	3	N/A	42.09
7	22	7	**6**	6	34.91	157.09
8	24	—	6	8	>3600	47.00
9	26	—	7	8	>3600	631.27
10	28	8	—	—	735.05	>3600

— : the procedure does not terminate after 3600 seconds

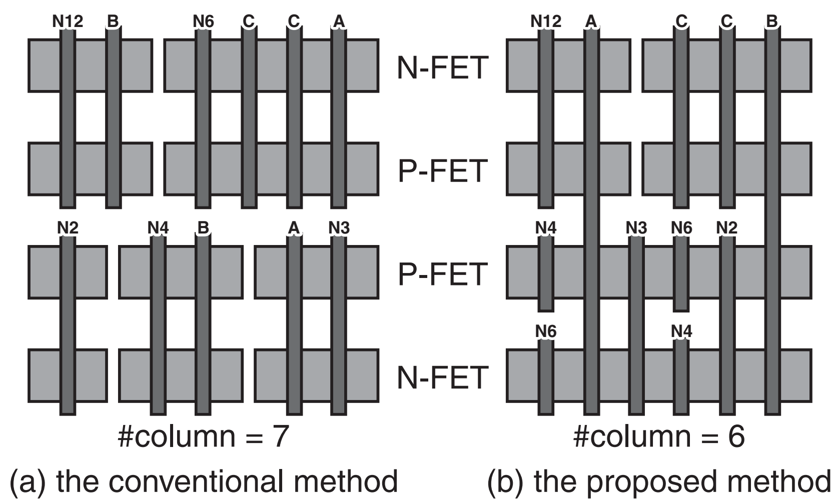


Figure 4.7 Examples of the layout of the cell number 7 in Table 4.9 created by (a)the conventional and (b)the proposed multi-row placement method. The conventional style assumes the pair of P/N transistors with the same gate input signals.

reasonable time. Figure 4.7 shows examples of the layout of the cell number 7 in Table 4.9 created by the conventional and the proposed method. This result clearly shows that the proposed method considers the gate connection more efficiently and creates more area-efficient multi-row transistor placement than the conventional one.

4.7 Summary

This chapter proposed flat and hierarchical approaches for generating a minimum-width single-row transistor placement of CMOS cells in presence of non-dual P and N type transistors, and generalized the single-row placement method to the multi-row placement method. Our approaches are the first exact minimum-width transistor placement method for non-dual CMOS cells. The flat single-row approach generated smaller width placement for 29 out of 103 dual cells than the placement method for dual cells explained in Chapter 2 which theoretically generates the smallest width placement among the existing exact methods for dual cells. The experimental results showed that it is not only applicable to CMOS cells with any types of structure, but also more effective even for dual CMOS cells compared with the transistor placement method only for dual cells.

The hierarchical single-row approach which is based on circuit partitioning reduced the runtime drastically and generated 81% of 340 cells in an industrial standard-cell library of a 90nm technology within one hour for each cell, whereas the flat approach and the exact

method only for dual cells generated 43% and 32%, respectively. Except several cells which consumed a large part of runtime, 76% of 340 cells were solved in 30 minutes. By improving the partitioning algorithm, we expect to implement a new hierarchical approach which covers 100% cells of industrial libraries.

The experimental results of the exact minimum-width multi-row transistor placement method showed that the proposed method generates more area-efficient multi-row placement than the conventional method only for dual cells by using the gate connection style which is more suitable for multi-row transistor placement than the conventional style and can solve the cells with up to 26 transistors in reasonable runtime.

Chapter 5

Yield-Optimal Cell Layout Synthesis for CMOS Logic Cells

5.1 Introduction

The recent improvement of VLSI process technologies enables us to integrate a large number of transistors on one chip, and significantly improves the circuit performance. On the other hand, the methodology of VLSI design becomes more and more complex and some new problems, such as Design For Manufacturability (DFM) have arisen. Due to the very high costs associated with the manufacturability of sub-micron integrated circuits, even a modest yield improvement can be extremely significant. In order to achieve the high yield, a standard-cell layout synthesis considering the DFM is required since standard-cells are the most basic elements of the cell-based design methodology as described in Chapter 1.

This chapter describes a comprehensive CMOS logic cell layout synthesis technique for yield optimization by minimizing the sensitivity to wiring faults due to spot defects. Spot defect is one of the main sources of electrical failure in VLSI integrated circuits. We modeled the sensitivity to wiring faults on intra-cell routings with consideration to the spot defects size distribution and the end effect of critical areas. Critical area is defined as the area in which the center of a spot defect must fall to cause a fault. We comprehensively generate the minimum-width layouts of CMOS logic cells and the exact optimal layouts are selected from all the possible minimum-width layouts by using our model of the sensitivity to wiring faults as a cost metric. Although the critical area used for the sensitivity calculation is extracted from the original layout patterns, the feasibility of the proposed sensitivity model to the practical lithography system is discussed. Moreover, the adequacy of the proposed sensitivity to the other cost metrics such as the cell delay and the total intra-cell wire length is demonstrated. The impact of the sensitivity reduction on the yield is also discussed in this chapter.

At first, Section 5.2 explains our yield cost metric for intra-cell routings and our layout styles are described in Section 5.3. Our comprehensive layout synthesis method and overall system are explained in Section 5.4 and Section 5.5 respectively, and experimental results are shown in Section 5.6. Section 5.7 discusses the relation of the critical area between before and after lithography, and the adequacy of the proposed sensitivity. In addition, the impact of the cost metric used in the proposed method on yield is also discussed in this section. Finally, Section 5.8 summarizes this chapter.

5.2 Wiring Fault Model for Yield Cost Function

Critical area is defined as the area in which the center of a spot defect must fall to cause a fault. Random defects caused by particulate contamination have been typically the dominant reason for yield losses[32]. Therefore, the correct estimation of the critical areas plays an important role in layout sensitivity to spot defects and yield prediction. The spot defect size distribution and the critical area are defined as $D(x)$ and $A(x)$ respectively, where x is the spot defect size. Assuming the area distribution of defect density is uniform and written as P_0, the fault probability P is expressed as:

$$P = P_0 \int_{min}^{max} D(x)A(x)dx \tag{5.1}$$

where *min* and *max* are the minimum and the maximum defect size[33]. The spot defects size distribution function can be assumed to be

$$D(x) = \frac{{X_0}^2}{x^3} \tag{5.2}$$

where X_0 is the peak defect size of the distribution[34]. We consider only a bridging fault of two wire segments on the same layer due to a spot defect, which is called fault type OE (One-layer Extra-material defect). Since it is impossible to define the shape of a real spot defect, the shape of defect is assumed to be rectangular for simplicity. The critical areas of two wire segments whose width are w and spaced by d with the defect size x are illustrated in Figure 5.1. The slashed area of Figure 5.1 (a) illustrates the critical area between two parallel wire segments. This type of critical area is defined as $L(x)$. Assume that the length of overlapped section is l, the critical area is expressed as $l \times (x - d)$ when the end effect is neglected[33]. We take the end effect into account so that the model can be applicable to more complex intra-cell routings, and the back-slashed areas are added to the critical area

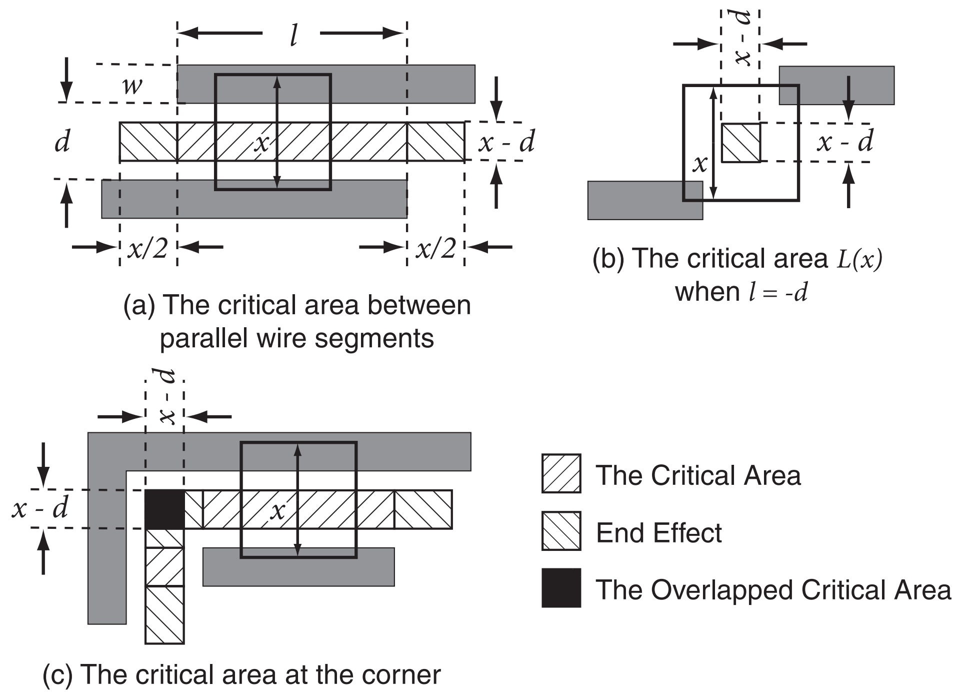


Figure 5.1 Critical areas between two wire segments spaced by d with the defect size x.

$L(x)$ which is newly expressed as

$$L(x) = (x + l) \times (x - d), \quad (l \geq -d). \tag{5.3}$$

$L(x)$ is calculated in the range of $l \geq -d$. When $l = -d$, the shape of the critical area $L(x)$ becomes square as illustrated in Figure 5.1 (b). Figure 5.1 (c) illustrates the critical area at the "L" type corner. Because the end sections of two critical areas are overlapped here, the area illustrated as a black square is counted twice. Therefore, this area is subtracted once from the critical area. If the corner type is "T" or "+", the area of the square is subtracted two or three times respectively. The black square area $R(x)$ is expressed as

$$R(x) = (x - d)^2. \tag{5.4}$$

The total critical area is calculated using these equations when the size of the spot defect is x. The probabilities of fault which results from each critical area is calculated using the equation (5.1) and the defect size distribution function (5.2) as follows:

$$P_L = P_0 {X_0}^2 \int_{min}^{max} \frac{1}{x^3} L(x) dx \tag{5.5}$$

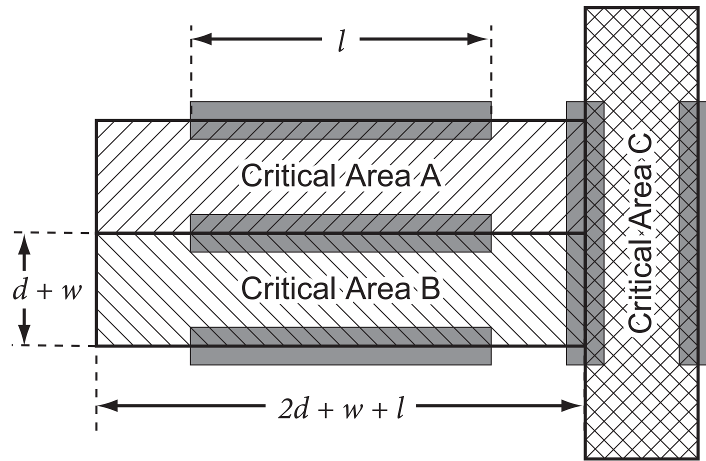


Figure 5.2 Critical areas between two wire segments spaced by d when the defect size x is larger than $2d + w$.

where P_L is the probability of fault which results from the critical area $L(x)$. Since P_0 and X_0 are process-dependent constant in this equation, these factors are excluded in the following discussion. The integral value without $P_0{X_0}^2$ is defined as a *Sensitivity* to spot defect and written as S_L. If the defect size x is smaller than d, the defects will not cause any fault because of the zero critical area. If $d \leq x < 2d + w$, the critical area $L(x)$ is expressed as the equation (5.3). The critical areas when the defect size x is equal to $2d + w$ are illustrated in Figure 5.2. For $2d + w < x$, the adjacent critical areas have overlapped sections and these sections are calculated redundantly. Since a standard-cell layout commonly has dense wiring, this causes overestimation of the sensitivity. Therefore, we assume the critical area is saturated to $(2d + w + l) \times (d + w)$ for $2d + w \leq x$. As a result, the sensitivity is expressed as

$$S_L = \int_d^{2d+w} \frac{(x + l)(x - d)}{x^3} dx + \int_{2d+w}^{max} \frac{(2d + w + l)(d + w)}{x^3} dx. \tag{5.6}$$

By setting *max* to ∞, we obtain

$$S_L = \ln \frac{2d + w}{d} + \frac{l - d}{2} \left(\frac{1}{d} - \frac{1}{2d + w}\right). \tag{5.7}$$

By calculating in the same manner, we also obtain the sensitivity S_R which results from the critical area $R(x)$ as follows:

$$S_R = \ln \frac{2d + w}{d} - \frac{d + w}{2d + w}. \tag{5.8}$$

These sensitivities are calculated between two different wire segments only when these are placed on the neighboring grids as a result of a grid-based intra-cell routing. The total sensitivity is calculated as follows:

Table 5.1 Our layout styles of the proposed comprehensive cell layout synthesis method.

1.	Static CMOS logic circuits.
2.	Transistors are drawn up in two horizontal rows, the upper row for P type transistors and the bottom row for N type transistors.
3.	Two transistors which have the same gate input signals are vertically aligned.
4.	All transistors are uniform-sized.
5.	The intra-cell routing uses only first metal layer.
6.	VDD are connected from the top of P-diffusion to the top boundary by the vertical first metal.
7.	GND are connected from the bottom of N-diffusion to the bottom boundary by the vertical first metal.
8.	A single contact hole is assumed to be enough to connect metal and diffusion or polysilicon.

1. Enumerate the adjacent parallel wire segments and calculate the sensitivity using the equation (5.7) for each wire segment.

2. Enumerate the overlapped critical areas at the corner and calculate the redundant sensitivity using the equation (5.8) for each overlapped critical area.

3. Subtract the sum of the redundant sensitivities from the sum of the sensitivities of the adjacent parallel wire segments.

The sensitivity calculated using our model is more accurate than that using the conventional model which does not take the end effect of the critical area into consideration. The conventional model neglects about 40% of the sensitivities compared with our model. Therefore, our model is more suitable for the cell layout which has dense and complex intra-cell wirings.

5.3 Layout Styles

Our layout styles are described in Table 5.1. The layout styles No. 1 through 3 for the transistor placement are based on the transistor placement styles defined in Chapter 2, since the cell layout synthesis method in this chapter uses the minimum-width transistor placement method proposed in Chapter 2. Although we assumed that all transistors are uniform-sized for simplicity in this chapter, it is easy to extend it to handle multiple-sized transistors. The

layout styles No. 5 through 8 in Table 5.1 are for intra-cell routing. We use only first metal layer for intra-cell routing since it is suitable for standard-cell layouts. A single contact hole is assumed to be enough to connect the metal and diffusion or polysilicon due to the commonly used silicidation.

5.4 Comprehensive Cell Layout Synthesis

5.4.1 Transistor Placement

Our method utilizes the procedure proposed in Chapter 2 to generate the minimum width transistor placements. In Chapter 2, the minimum width placements are generated one by one until a routable placement is found via Boolean satisfiability. To generate all possible minimum-width placements comprehensively, we repeatedly generate a minimum-width placement until no other placement is found for the width of the minimum-width placement. Although the layout style used in this chapter assumes that two transistors which have the same gate input signals are vertically aligned for simplicity, the comprehensive intra-cell routing method explained in the following section is applicable to the placement which has a vertically aligned P and N type transistor pair with different gate input signals. Therefore, we can also use the minimum-width transistor placement method for non-dual cells proposed in Chapter 4 to generate all possible minimum-width placements, though the proposed comprehensive cell layout synthesis method uses the transistor placement method only for dual cells in this chapter.

5.4.2 Intra-Cell Routing

The proposed cell layout synthesis method uses a comprehensive intra-cell router to realize an exhaustive pattern generation for intra-cell routing. Conventionally, several exact routing algorithms have been proposed[23, 35, 36]. Among them, [35] can generate all routing patterns exactly, but can not solve larger problems than 4×4 grids. On the other hand, [36] can solve larger problems, but this method has some restrictions and the search space of this method is too small for general cell generation. In this chapter, we propose a new exact routing algorithm that can solve problems which is large enough to generate standard cell routing patterns and its search space is much larger than that of [36] by using more practical restrictions for standard cells. In our method, a generated placement is transformed into a grid model problem illustrated in Figure 5.3 (a). When it is given to our router, the grids for

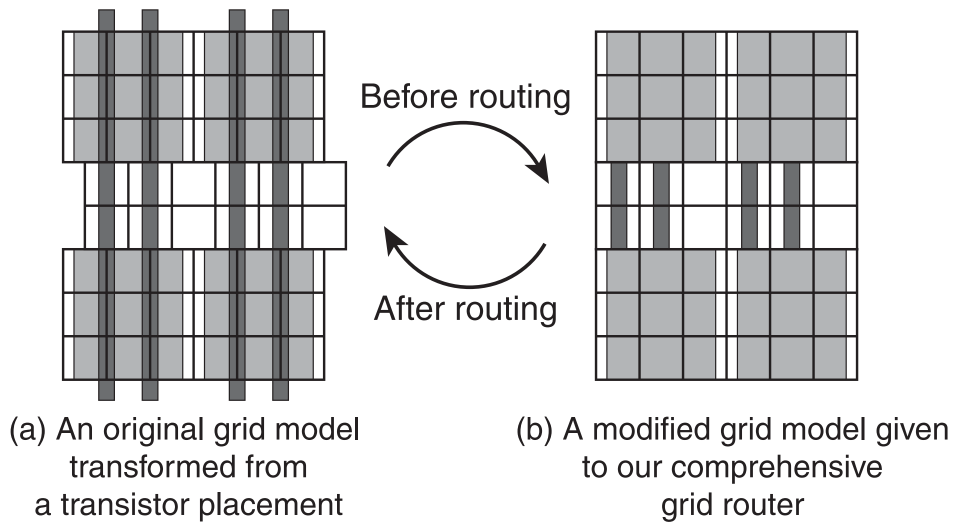
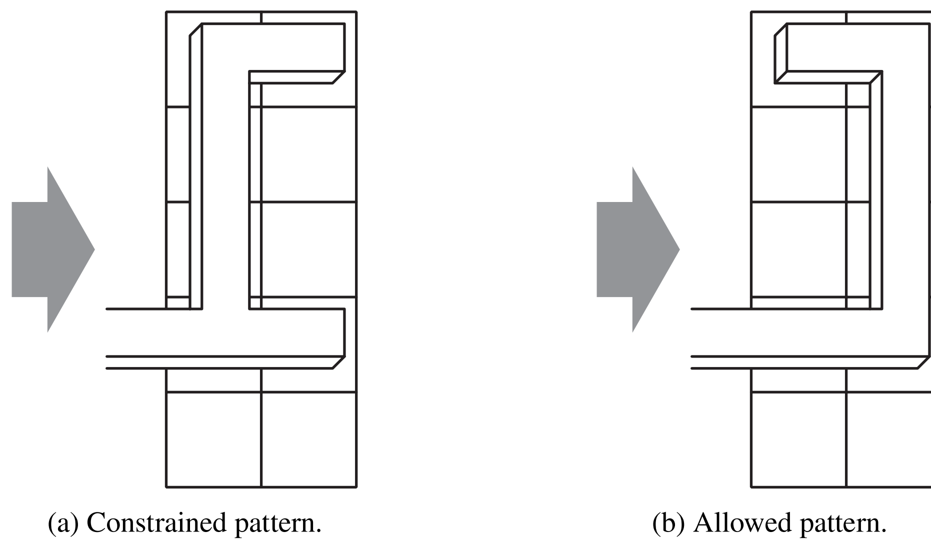


Figure 5.3 Grid models used in our comprehensive intra-cell routing system.

gate, between the P and N diffusions, are shifted half grid to the left, since our method routes the problems of complete grid model as illustrated in Figure 5.3 (b). Our method routes a given grid model problem from left to right, column by column. Inside each column, all possible patterns are generated for each pattern of the previous column considering VDD/GND terminals, diffusion silicides, gate, and I/O. VDD terminals must be connected from the top of P diffusion to the top boundary of routing area, and GND terminals from the bottom of N diffusion to the bottom. A single contact hole is assumed to be enough to connect the metal and the terminals on the diffusion or polysilicon because of the silicides. When redundant via holes are needed, it is easy to add them to all generated routing patterns after routing is finished and select the optimal one. If a gate terminal is an input port and is not connected to other terminals by metal layer, at least one grid of this terminal must be reserved so that an I/O contact can be placed on it. It also considers the design rules so as not to generate the routing patterns which violate metal design rules when the grids are transformed back to the original grid position as shown in Figure 5.3 (a). The proposed intro-cell router has a constraint for search space reduction as follows:

- A single net can not fork when it extends to the subsequent column.

Figure 5.4 illustrates examples about this restriction. The routing pattern of Figure 5.4 (a) is not generated by the proposed method since a single net forks to extend to the subsequent column and this pattern violates the above restriction. On the other hand, the pattern of Figure 5.4 (b) is allowed because two nets, which is not connected yet, are extended to the



(a) Constrained pattern. (b) Allowed pattern.

Figure 5.4 Wire branching constraint of the proposed comprehensive intra-cell routing method.

subsequent column individually and connected inside the subsequent column.

Figure 5.5 illustrates the flow of our intra-cell routing. The detail of each procedure is described as follows. The numbers below corresponds to the numbers in Figure 5.5. Each procedure is explained using a sample problem illustrated in Figure 5.6. This figure shows that the nets on the left are extended to the leftmost column. The thick rectangles in this figure indicate the terminals on source or drain, and the narrow rectangles indicate those on gate. The nets and terminals that have the same number have to be connected to each other on at least one grid.

1. At first, the pattern to be routed in the current column is determined by the terminal pattern of this column and the net pattern given from the previous column. If there are some terminals which are already connected, these terminals are removed from this pattern. Figure 5.7 (a) shows an example of this procedure. The terminal number 4 on the bottom row is removed because this terminal is already connected to the net by silicides.

2. Next, all the terminals which are not connected yet are found and all possible connection patterns are generated considering the silicides and space of I/O contact. In Figure 5.7 (a), the terminals number 5 and 6 can be connected inside this column. The patterns illustrated in Figure 5.7 (b), (c), (d), and (e) show all the possible connection patterns generated from Figure 5.7 (a).

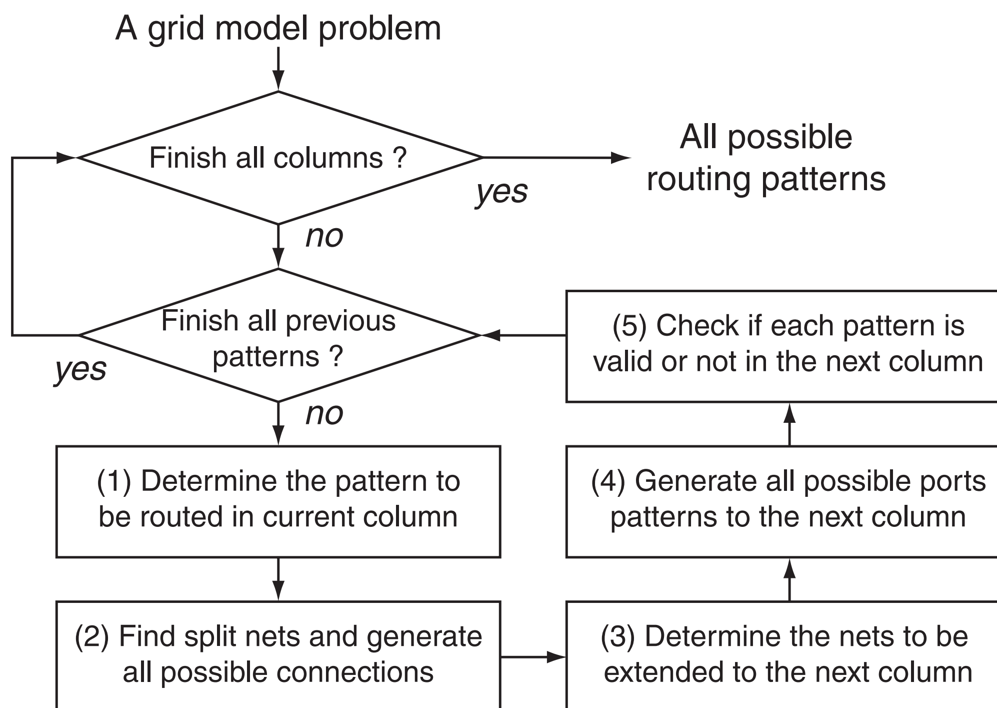


Figure 5.5 The flow diagram of our comprehensive intra-cell router.

3. Then, the nets which have to be extended to the subsequent column, *i.e.*, the nets with unconnected terminals, are determined for each pattern generated in the former step. For example of the pattern of Figure 5.7 (d), the nets number 3, 4, and 6 have other unconnected terminals in the following columns and the net number 5 is not finished connection yet. Therefore, these nets have to be extended to the next column.

4. All possible net patterns extended to the subsequent column are generated for each pattern generated in the step 3. Figure 5.8 illustrates all the possible patterns generated from the pattern of Figure 5.7 (d). Because of the constraint explained before, a single net can not fork when it extends to the subsequent column. Each net in this figure has at most one extension to the subsequent column.

5. Finally, check if all the generated patterns to the subsequent column are valid or not. If some nets of a generated pattern conflict to other connections of nets or I/Os, this pattern is impossible and removed from the group of generated patterns. For example of Figure 5.8, (d-1) and (d-2) have nets which conflict all the terminals of terminal number 4 in the subsequent column in Figure 5.6. Therefore, these patterns are removed and others are given to the subsequent column.

These procedures are applied for each pattern of each column and generate all possible routing patterns of a given grid-model problem under our layout styles.

5.5 Overall System

Using the procedures described in Section 5.4, we can generate all possible layouts comprehensively. Then calculate the sensitivity using the model described in Section 5.2 for each layout. To obtain the exact yield-optimal layouts, the proposed method does not use heuristic approaches. The proposed method selects the yield-optimal layout from all the possible minimum-width layouts. Although the sensitivity can be reduced by inserting redundant spaces, this chapter targets to generate the yield-optimal minimum-width cell layout. It is possible to optimize the yield further by adjusting the space between wires in a grid-less manner after the yield-optimal layouts are selected. Moreover, if there are spaces between cells after placing the generated cells, we can also improve the yield by increasing the width of a cell without any area overhead. The next chapter proposes a further yield optimization by inserting redundant spaces through a cell layout de-compaction in a grid-less manner. Overall flow of our yield-optimized cell layout synthesis is illustrated in Figure 5.9 and described as follows.

1. For a given transistor netlist, generate a minimum-width transistor placement using the procedure explained in Chapter 2. The number of diffusion gaps of this placement is defined as G.

2. Comprehensively route the generated placement using the procedure explained in Section 5.4.2. If this placement is routable, generate routed layouts. Otherwise no solution.

3. Find another placement with G gaps using the procedure explained in Chapter 2. If there is one, go to step 2 again. Otherwise go to step 4.

4. If there is no routed layout, increment G and generate a transistor placement with G gaps using the procedure explained in Chapter 2. Then go to step 2 again. Otherwise go to step 5.

5. Place the I/O contact holes for each generated layout if needed. If several possible I/O placement patterns can exist for one layout, generate all possible I/O patterns.

6. Calculate the sensitivity for each layout and find the optimal solution.

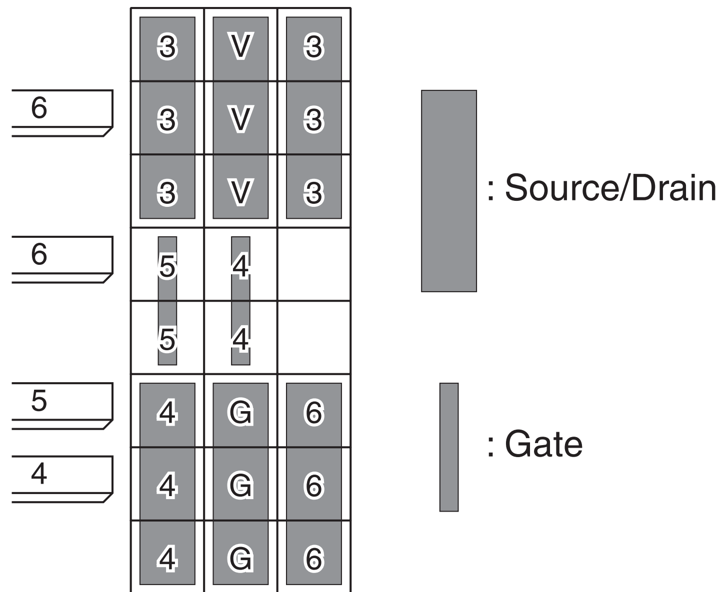


Figure 5.6 A sample problem for explaining our comprehensive intra-cell router.

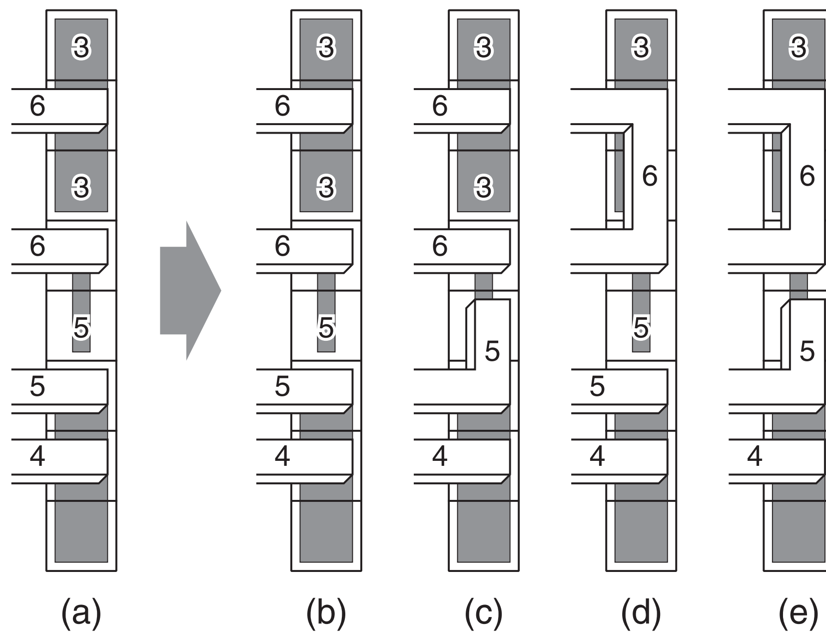


Figure 5.7 An example of all possible connection patterns inside a column.

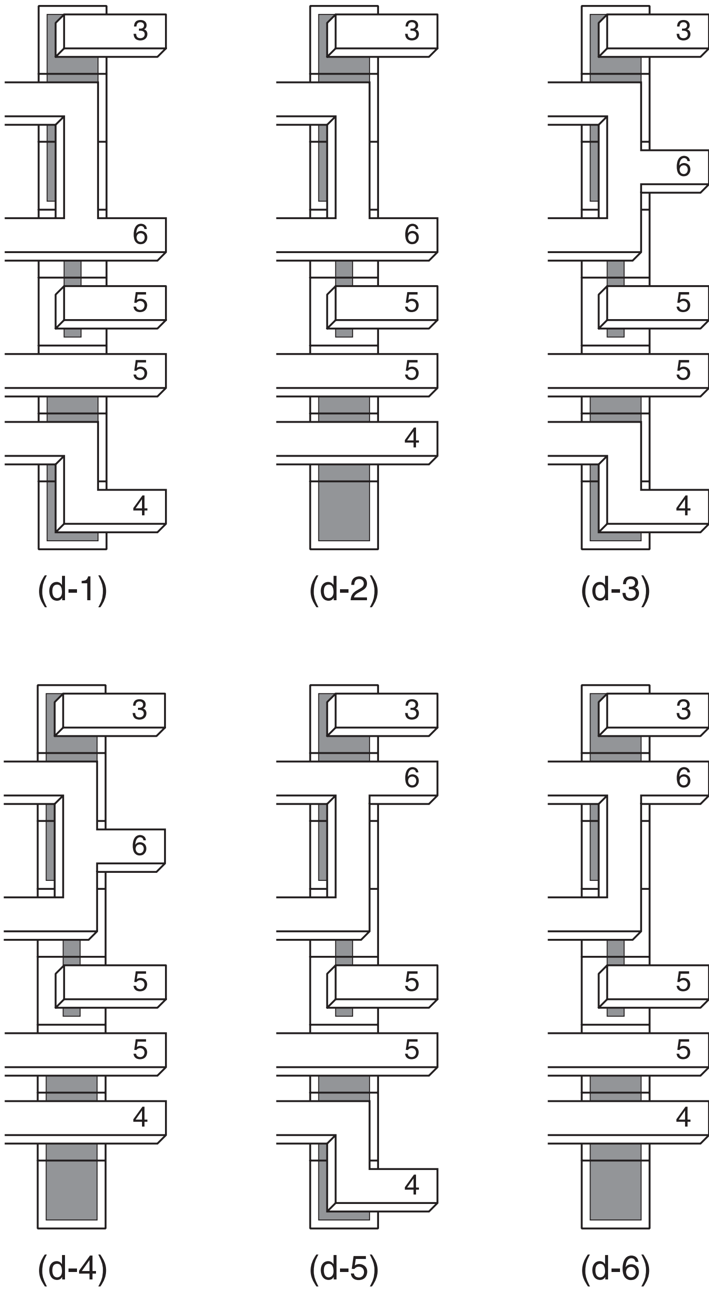


Figure 5.8 An example of all possible patterns to be extended to the subsequent column generated from the pattern (d) in Figure 5.7.

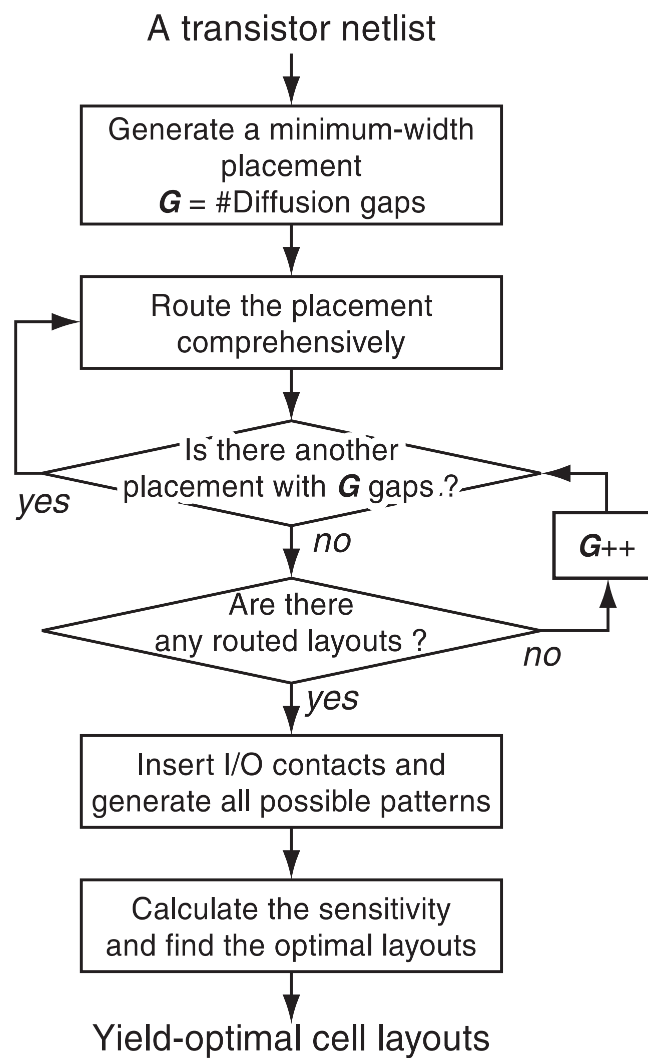


Figure 5.9 The flow diagram of our yield-optimal cell layout synthesis system.

5.6 Experimental Results

We applied our layout synthesis method to 8 CMOS logic circuits in a standard-cell library, which have up to 14 transistors. Tables 5.2 and 5.3 summarize the results. In this experiment, we assumed a 0.35μm process technology and used 8 rows for routing grids. For each circuit, these tables indicate the number of transistors, the number of columns of routing grids, the number of possible placements, the number of routable placements, the number of generated layouts, the CPU times for layout synthesis and for selecting the sensitivity-minimum layout, the minimum sensitivity value S_{min} and the average sensitivity value of all wire-length-minimum layouts S_{wire}. This table also shows the percentage of reduction in the sensitivity by selecting the sensitivity-minimum layout. This percentage is calculated by

Table 5.2 The results of the comprehensive cell layout synthesis.

Circuit name	*#tr.*	*#col.*	*#place*	*#route*	*#layout*	*Cell synth. CPU(sec.)*	*Selection CPU(sec.)*
ao222	14	9	32	8	948852	14685.04	323.70
aoi21	6	4	4	4	999	0.36	0.17
aoi211	8	5	4	4	8839	1.70	1.60
eno	10	6	2	1	19648	13.70	4.45
gen2	12	8	24	2	34120	419.98	10.38
mux2	12	9	144	4	392	6449.82	0.19
nand4	8	5	4	4	38366	3.01	6.92
xnor2	10	7	144	24	27414	2164.30	6.97

Table 5.3 The cell selection results of the comprehensive cell layout synthesis.

Circuit name	*Minimum sensitivity* S_{min}	*Average sens. wire-min.* S_{wire}	*Reduction ratio (%)*
ao222	18.04	20.55	12.19
aoi21	5.37	6.56	18.10
aoi211	6.08	7.13	14.77
eno	11.35	13.33	14.93
gen2	17.78	20.90	14.91
mux2	31.72	33.79	6.10
nand4	5.37	6.38	15.78
xnor2	11.18	15.27	26.81

$(S_{wire} - S_{min})/S_{wire} \times 100$. As shown in Table 5.2, our method generates the layout comprehensively in only a few seconds for circuits with up to 8 transistors, whereas the runtime is severely high for the circuit with 14 transistors. In the case of standard-cell layout synthesis, we are allowed to consume relatively long time to obtain high quality layouts. For larger circuits, however, we need to utilize some heuristic techniques such as netlist partitioning for transistor placement or branch-and-bound for intra-cell routing to solve them in reasonable time, even though these techniques do not guarantee the optimality.

In Figure 5.10, the minimum sensitivity value and the minimum, average, and maximum

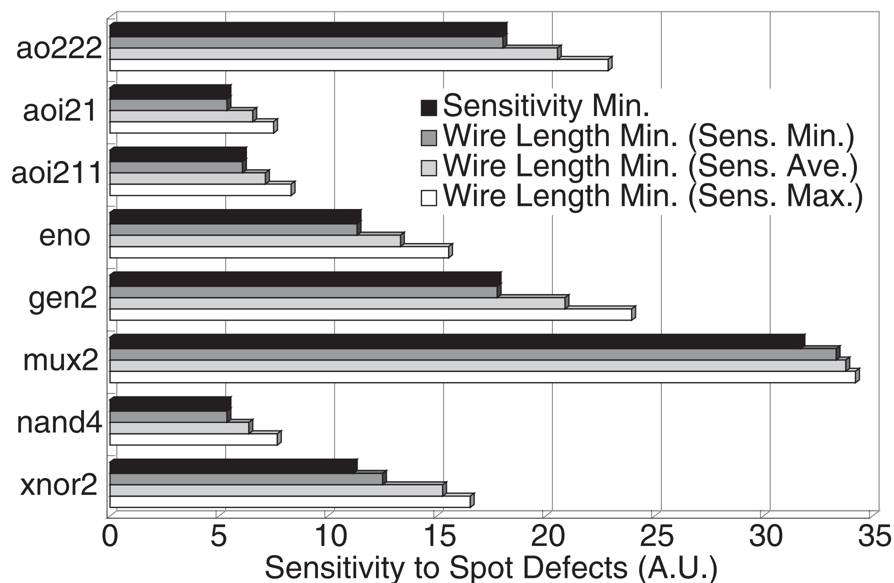
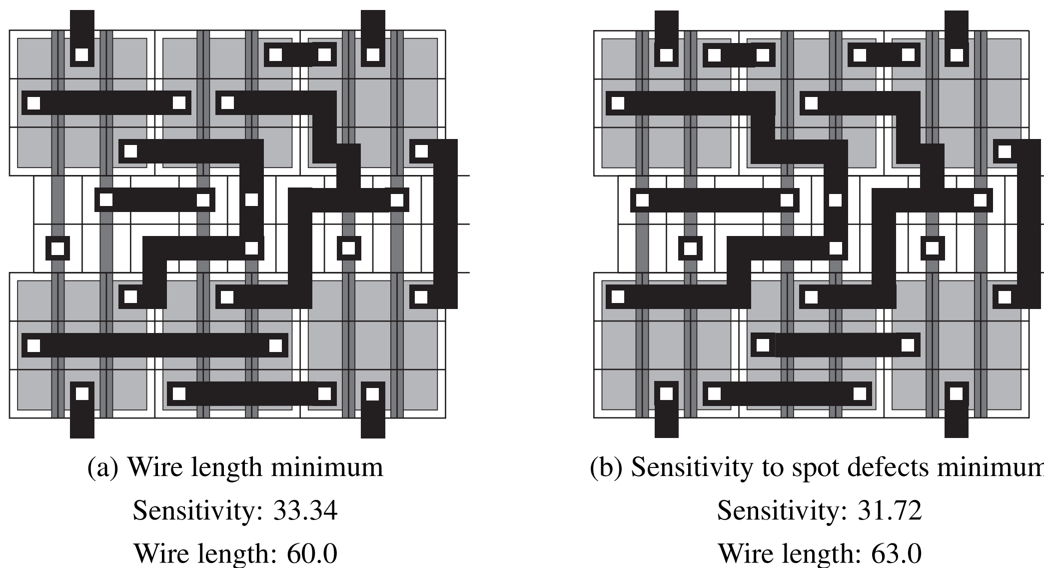


Figure 5.10 Changes in layout sensitivity to spot defects by selecting the sensitivity-minimum layouts.

sensitivity value of wire-length-minimum layouts are plotted for comparison. These results show that the wire-length-minimum layouts are not always the optimal layout for the sensitivity to spot defects. Compared with the average sensitivity value of all the wire-length-minimum layouts, our method can improve the sensitivity about 15% on an average of 8 circuits by picking up the optimal layout of sensitivity from all the generated layouts.

The snapshots of the wire-length-minimum layout and sensitivity-minimum layout of mux2 are illustrated in Figure 5.11 for example. The sensitivity optimal layout has smaller sensitivity whereas its wire length are longer than the wire length optimal layout. We also compared the cell delay of these two layouts. The sum of the delay of all the timing arc with no output capacitance was calculated in simulation. A timing arc is defined as a signal flow from an input to an output on a cell, *e.g.*, A rise → Y fall. The comparison result shows that the delay of the sensitivity-minimum layout is slower only 0.23% than that of wire-length-minimum layout. This result shows that there is little performance degradation between these two layouts and the layouts selected by the sensitivity are acceptable also in terms of cell delay.



(a) Wire length minimum
Sensitivity: 33.34
Wire length: 60.0

(b) Sensitivity to spot defects minimum
Sensitivity: 31.72
Wire length: 63.0

Figure 5.11 The optimal layouts of "mux2" in Table 5.2 generated by our method and selected by (a)wire length and (b)sensitivity to spot defects.

5.7 Discussion of Yield Cost Metrics

5.7.1 Lithography Impact on the Critical Area

The critical area used for modeling the sensitivity through this section is calculated from the original layout pattern of the first metal layer. However, the actual patterns printed on the silicon wafer usually changes due to lithography and/or Optical Proximity Correction (OPC). Therefore, it is necessary to clarify the impact of the lithography processes on the critical area. We used a cell library of a 90nm technology to show the effect of the lithography. Although we used cell layouts of a 0.35μm technology in the experimental results section, a 90nm cell library which includes the subwavelength feature size is used to highlight the optical proximity effects in this section. Figure 5.12 shows the relations between the critical area of the original layout patterns and the patterns after lithographic simulation before and after OPC, respectively. The reference line in this figure indicates $y = x$. The critical area plotted in this graph is calculated only for a bridging fault of two wire segments on the same layer due to a spot defect, which is called fault type OE (One-layer Extra-material defect). This graph plots the critical area of 340 cells in a 90nm cell library using the defect size of 1.5 times larger value than the minimum spacing of the first metal layer. Let D denote the defect size, the OE type critical area calculation procedure in this experiment is written as follows.

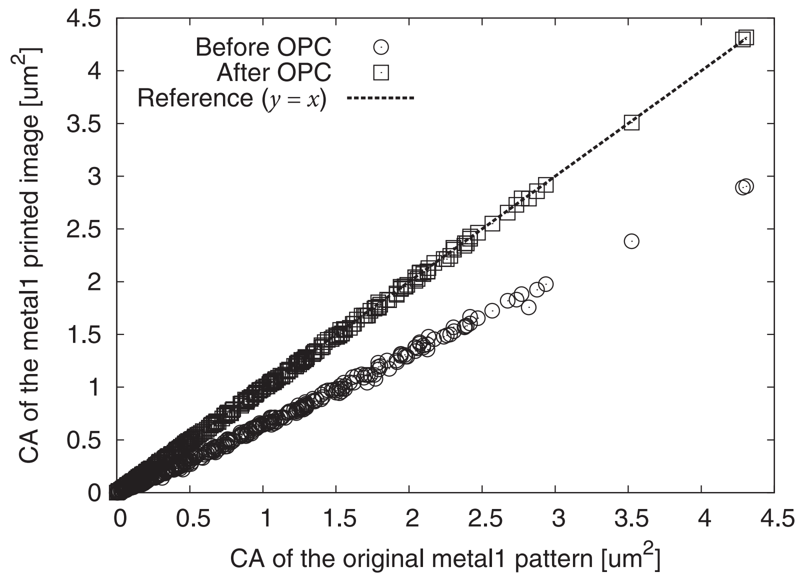


Figure 5.12 Lithography impact on the critical area before and after OPC for 340 cell layouts in a 90nm technology.

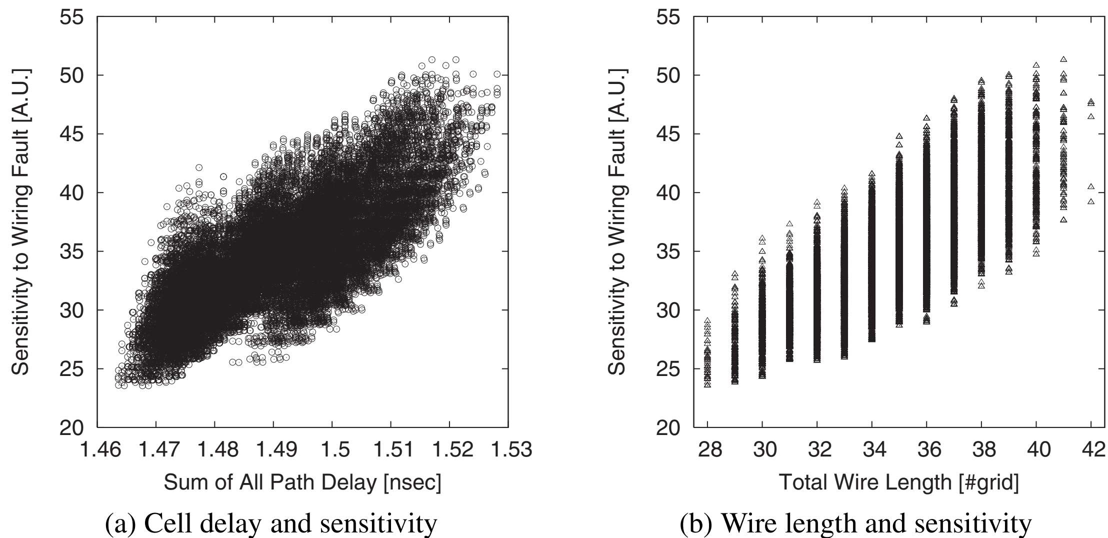
1. Expand the polygons on the first metal layer by $D/2$ moving the sides of the polygons perpendicularly.

2. Sum up the overlapped areas of the expanded polygons.

The printed images are obtained by the lithographic simulation using *Calibre OPC*[37]. The lithographic condition used in this experiment is summarized as follows.

- Lithography Wavelength: 193nm
- Numerical Aperture: 0.8
- Mask Reduction Factor: 4

As shown in this graph, the critical areas of the printed images after OPC are almost always equal to that of the original layouts under this condition. Both before and after OPC, the correlation coefficients between the critical area of the printed images and the original layouts are almost equal to 1. Therefore, the critical area of the printed images has strong correlation with the original critical area even without OPC. We can conclude from this result that the proposed model of sensitivity to wiring fault calculated from the critical area of the original layout pattern also has strong correlation with the sensitivity after lithography, and can be used to model the yield of the printed images on the silicon wafers.



(a) Cell delay and sensitivity (b) Wire length and sensitivity

Figure 5.13 The relation between two cost metrics in the case of the exhaustively-generated 27414 cell layouts of xnor2.

5.7.2 Relation Between Sensitivity and Performance

This section describes the relation between the proposed sensitivity to wiring faults and other cost metrics of cell layout, such as cell delay and total intra-cell wire length. Figure 5.13 (a) and (b) show the relation between cell delay and sensitivity, and wire length and sensitivity, respectively, in the case of the comprehensively-generated 27414 cell layouts of "xnor2" in Table 5.2. For each layout of xnor2, we simulated and summed up the delay of all the timing arcs, and plotted it in the graph (a). Both (a) and (b) show the positive correlation between two cost metrics, and the values of the correlation coefficient are 0.77 and 0.76, respectively. There are strong correlations both between cell delay and sensitivity, and wire length and sensitivity. Therefore, we can conclude that the selected layout by the sensitivity might have the acceptable quality in terms of the other cost metrics such as cell delay and total intra-cell wire length.

5.7.3 Relation Between Sensitivity and Yield

This section describes the relation between the sensitivity and yield. Given the critical area of the first metal layer, the yield of this layer can be modeled using a Poisson-based yield model[38]:

$$Y_{met1} = exp(-P_0 \int_{min}^{max} D(x)A(x)dx) \tag{5.9}$$

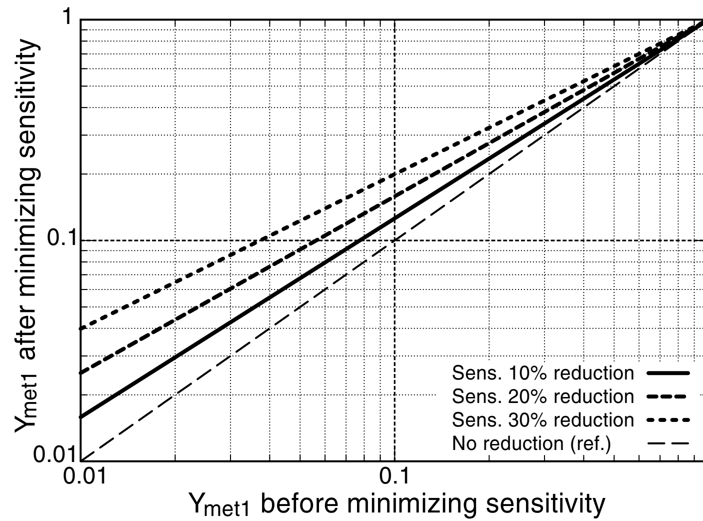


Figure 5.14 The impact of the sensitivity reduction on the yield of the first metal layer.

where Y_{met1} is the yield of the first metal layer. This equation can also be expressed as below:

$$Y_{met1} = exp(-P_0 X_0{}^2 \times S_{met1}) \tag{5.10}$$

where S_{met1} is the sensitivity of the first metal layer. We define the sensitivity reduction ratio σ and the yield improvement ratio ϕ as follows:

$$\sigma = \frac{S_{orig} - S_{met1}}{S_{orig}}, \tag{5.11}$$

$$\phi = \frac{Y_{met1} - Y_{orig}}{Y_{orig}} \tag{5.12}$$

where S_{orig} and Y_{orig} are the value of the sensitivity and yield before the sensitivity is minimized, respectively. Although $P_0 X_0{}^2$ is the process-dependent constant and is hard to be determined accurately, we can derive the relation between σ and ϕ using the equations (5.10), (5.11), and (5.12) as follows:

$$\phi = exp(-\sigma \log Y_{orig}) - 1. \tag{5.13}$$

Using this equation, the relation between the yield value before and after sensitivity minimization can be illustrated as Figure 5.14. This graph clearly shows that the impact of the sensitivity reduction on the yield improvement is greater when the original yield value Y_{orig} is small. For example of the sensitivity reduction is 20% case, the yield improvement ϕ is 0.20% when $Y_{orig} = 0.99$, while 4.56% when $Y_{orig} = 0.80$.

The total yield Y_{total} can be expressed as the product of all yield loss reasons like below[38]:

$$Y_{total} = \prod_{i=1}^{N} Y_i \tag{5.14}$$

where N is the number of yield loss reasons and Y_i is the yield value of each reason. Therefore, the yield improvement ratio ϕ is the same value for both the yield of the first metal layer and the total yield if the yield values of all the other reasons remain the same. From this point of view, we can conclude that our method is more effective when the total yield value of the manufacturing process is small, *i.e.*, the process is not mature yet.

5.8 Summary

An optimal cell layout synthesis technique to minimize the sensitivity to wiring faults has been presented in this chapter. The sensitivity to wiring fault due to spot defects for intra-cell routings was modeled considering the spot defects size distribution and the end effect of critical areas, and used as a cost function. The impact of the sensitivity reduction on the yield improvement is also described. Our cell layout synthesis technique generates the minimum width layouts of CMOS logic cells comprehensively, and selects the optimal layouts based on the cost functions. We applied our comprehensive layout synthesis method to 8 CMOS logic circuits which have up to 14 transistors and the results show that the fault sensitivities are reduced about 15% compared with the wire-length-minimum layouts. Our layout synthesis method will be applicable for deriving the optimal cell layouts by some other cost metrics, such as power, delay, and signal integrity, if reasonable cost functions are given.

Chapter 6

Yield Optimization by Timing-Aware Cell Layout De-Compaction

6.1 Introduction

This chapter targets a timing-aware cell layout de-compaction method for yield enhancement. Recently, a lot of papers related to VLSI yield improvement have been published[14, 15, 16, 17, 18]. Most of them are proactive methodologies, which are not a post process. In [17], logic synthesis for manufacturability is proposed. This methodology introduces the manufacturability cost into logic synthesis and replaces the traditional area-driven technology mapping with a new manufacturability-driven one. It realizes larger reduction of the manufacturability cost when yield-optimized cells are available in the cell library. A new design flow proposed in [18] integrates manufacturability information into the timing-driven synthesis and place&route cost function. Yield-aware logic synthesis, place&route, and timing optimization are executed incrementally in this design flow. This flow uses a DFM extension library, which has variants of the basic logic functions with different manufacturability costs. They demonstrated the advantages of their methodology by applying it to several commercial ICs.

As stated above, a yield-enhanced standard-cell library is essential to these yield-aware VLSI design methodologies. Yield-enhanced standard-cell libraries were, however, designed mainly by hand and there is no fully automated standard cell yield optimization method proposed for this purpose. This chapter proposes an automatic yield-optimization technique for standard cells. Several papers have proposed the de-compaction method for yield-optimization[39, 40]. However, these methods consider only yield and area as costs, and the circuit performances are not considered. The careless modification of the original layout may degrade its performances severely and the created layout is not always acceptable for the

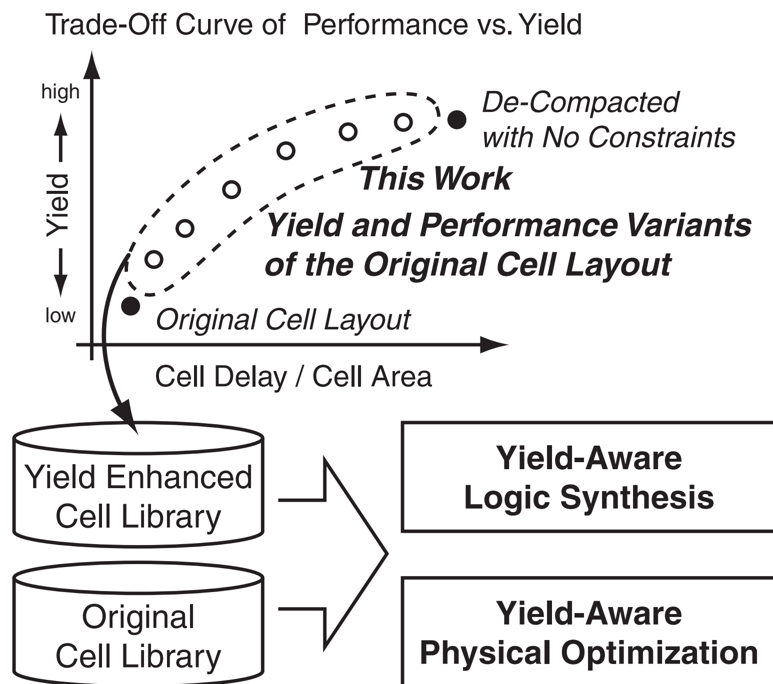


Figure 6.1 Overview of the proposed timing-aware cell layout yield enhancement framework.

target performances. Therefore, we propose a timing-aware cell layout de-compaction technique for yield optimization. The proposed method relaxes the width of a given cell layout under given timing constraints to optimize the yield. Since there is normally a high quality, hand-crafted original cell library, it is straightforward to optimize the yield by relaxing the width of the original layout with low computational effort, rather than creating it from scratch. Moreover, the layout after de-compaction preserves the integrity and the predictability of the original layout because it also preserves the relative geometry as the original layouts.

The proposed method optimizes the yield by minimizing the Critical Area (CA). Critical area is defined as the area in which the center of a spot defect must fall to cause a fault and its reduction plays an important role for yield enhancement. This de-compaction framework is also effective for OPC-relaxation. The OPC-relaxation results in terms of the fractured mask data size reduction is also shown in this chapter. Moreover, the proposed framework is applied to the redundant contact insertion adjacent to the single contacts. Recently, contact failure becomes one of the most dominant yield loss reasons and redundant contact insertion is highly recommended to improve the yield. To take the parametric yield into account, the proposed de-compaction framework is also extended to the gate layout pattern regularity enhancement to reduce the systematic variation of the gate critical dimensions.

The overview of the proposed method is illustrated in Figure 6.1. This method creates a yield-optimized cell layout under given various timing constraints and can pick up the yield variants of a cell layout from the cell delay versus yield trade off curve. These cells are prepared as a yield-enhanced cell library and used for the yield-aware logic synthesis and physical optimization. The proposed method performs a de-compaction of the original cell layout using Linear Programming (LP). We develop a new accurate linear delay model to formulate the timing constraints into LP. This model approximates the delay difference from the original delay induced by the differences of the parasitic capacitances after de-compaction.

Section 6.2 defines the problem solved in this chapter and explains the LP formulation of the design rule constraints. The formulation of the yield cost metrics, *i.e.*, critical area minimization, OPC relaxation, redundant contact insertion, and gate layout pattern regularity enhancement are explained in Section 6.3. Section 6.4 introduces the developed delay model, and the overall flow of the proposed method is described in Section 6.5. Then, the experimental results are shown in Section 6.6. Finally, Section 6.7 summarizes this chapter.

6.2 Problem Definition and Design Rule Constraints

The yield enhancement problem explained in this chapter is defined as follows:

- *Given*: Original cell layout
- *Minimize*: Yield cost function
- *Subject to*:
    1. Design rule constraints
    2. Timing constraints

Design rule constraints are formulated from a constraint graph constructed from a given layout, *i.e.*, a set of polygons, as shown in Figure 6.2. The target of the proposed de-compaction method is standard cells and their heights are usually fixed. Therefore, we explain the de-compaction of only horizontal direction. Each constraint exists between the vertical edges of polygons. Each vertex of the graph corresponds to each vertical edge of polygons and each edge of the graph has a weight value which corresponds to either the value of the minimum space or width. Once a constraint graph is constructed, it is straightforward to formulate the

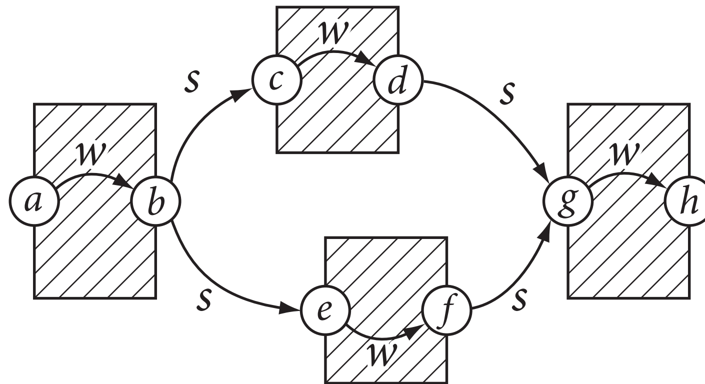


Figure 6.2 An example of a constraint graph constructed for polygons inside a given layout.

linear constraints. For example of Figure 6.2, a minimum-width constraint $b - a \geq w$, and a minimum-spacing constraint $g - d \geq s$, etc., are extracted.

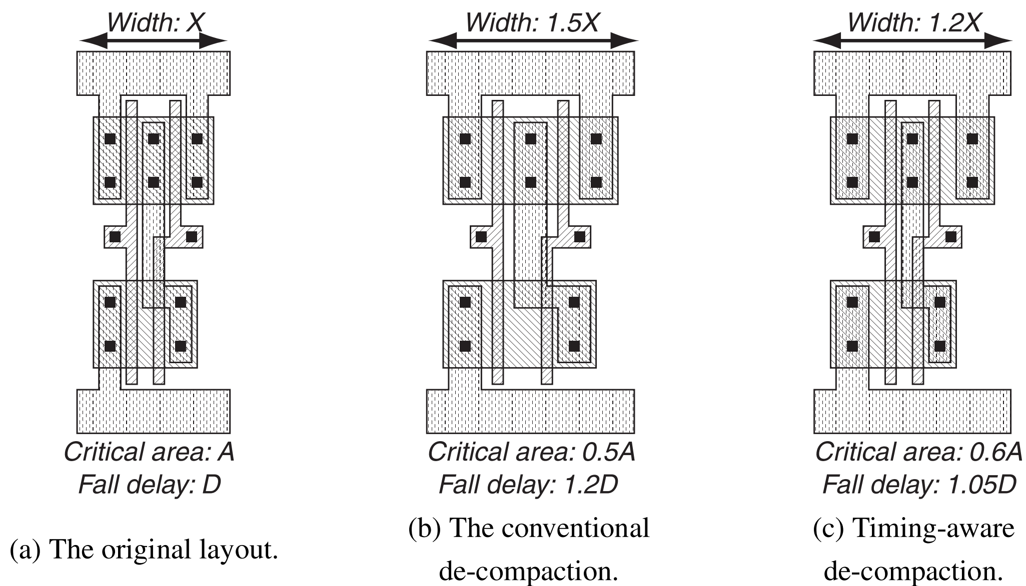
Formally speaking, a node V_i in a constraint graph represents a layout element E_i, usually an edge of a polygon or an instance of an object such as via. For example, let x_i represent the x coordinate of a layout element E_i, the minimum spacing constraint between two layout elements E_i and E_j is written as $x_j - x_i \geq d_{ij}$, where d_{ij} is the minimum distance required by a design rule. This constraint corresponds to a directed arc A_{ij}, from node V_i to V_j with weight d_{ij} in the constraint graph.

Of course, not only spacing and width design rules, but also other miscellaneous rules are formulated to create a layout without design rule violations after de-compaction. The LP formulation of the yield cost functions and the timing constraints will be explained in the following sections.

6.3 Yield Cost Metrics

6.3.1 Critical Area Minimization

In this section, we will explain how to minimize the total CA. Figure 6.3 shows the conceptual illustration of the cell layouts created by the conventional de-compaction method and the proposed timing-aware de-compaction method by minimizing the critical area. The conventional timing-unaware method increases the width and space in the original layout for CA minimization, whereas the timing-aware one does not increase the width and space of the nets which have an effect on the target delay during CA minimization to meet the given timing constraint. Therefore, the proposed method creates the yield and performance variants of



(a) The original layout. (b) The conventional de-compaction. (c) Timing-aware de-compaction.

Figure 6.3 The conceptual layouts of 2-input NAND created by the conventional and the timing-aware de-compaction methods.

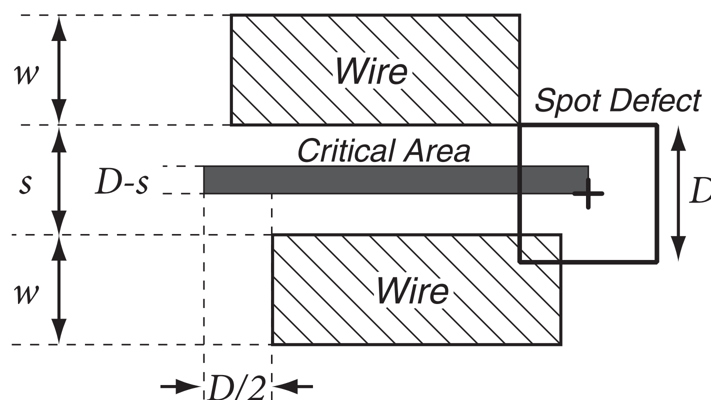
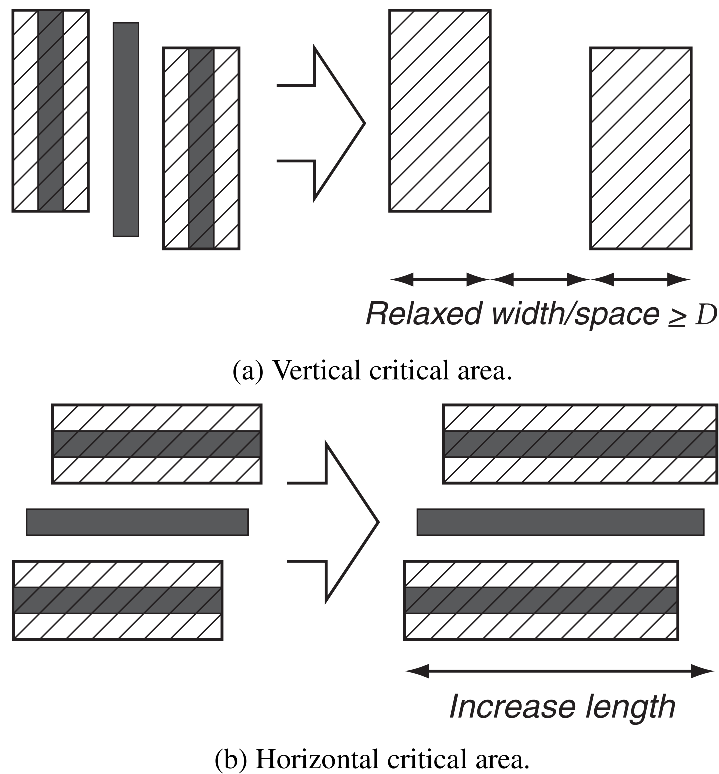


Figure 6.4 Schematic diagram of a short type critical area.

a cell layout depending on the given timing constraint. Figure 6.4 illustrates the schematic diagram of short type CA between two parallel wire segments whose width are w and spaced by s with the defect size D. We assume that the shape of the spot defect is square for simplicity of the CA calculation. If the center of the defect falls inside the CA, these two wires are connected and cause a fault. Open type CA is also defined for each single wire segment in the same manner. The total CA is calculated if the coordinates of all edges are known. In our formulation, all these coordinates are given as variables and the total CA should be minimized. Figure 6.5 shows variation of vertical and horizontal CA of both short and open type



(a) Vertical critical area.

(b) Horizontal critical area.

Figure 6.5 Variation of (a)vertical and (b)horizontal critical areas after horizontal de-compaction.

after horizontal de-compaction. The vertical CA are reduced by relaxing the width/space of polygons and finally become 0 when the width/space becomes the same value as the defect size D, whereas the horizontal CA are possibly increased since the length of horizontal wire segments are increased by horizontal de-compaction. The horizontal CA is easy to formulate as a linear function because it increases in proportion to the length. On the other hand, the calculation of the vertical CA is not so easy because it changes as shown in Figure 6.6. The critical area should be 0 if the width/space is larger than the defect size D. To realize this function, we use temporary variables r and l. These variables are defined as follows:

$$r \geq \frac{x_1 + x_2}{2}, \quad r \geq x_1 + \frac{D}{2} \tag{6.1}$$

$$l \leq \frac{x_1 + x_2}{2}, \quad l \leq x_2 - \frac{D}{2} \tag{6.2}$$

where x_1 and x_2 are the right and left edge of the polygons, respectively, as shown in Figure 6.7, and D is the defect size. Assume A is the vertical CA between these polygons, A can be written as $A \propto (r - l)$. To minimize the CA A, r should be minimized and l should be

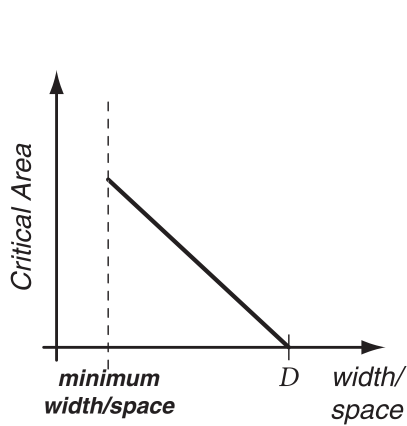


Figure 6.6 Change of the vertical critical area by the width or space.

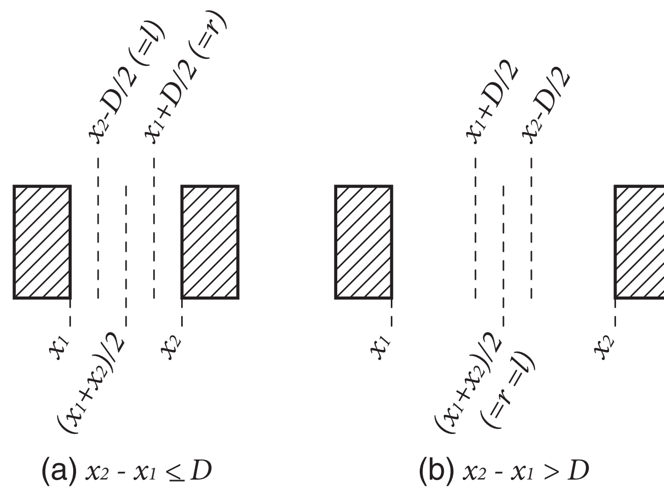


Figure 6.7 Calculation of the vertical critical area.

maximized. Under this condition, r and l are described as follows.

$$\begin{cases} r = x_1 + D/2, \;\; l = x_2 - D/2 & (x_2 - x_1 \leq D) \\ \qquad r = l = (x_1 + x_2)/2 & (x_2 - x_1 > D) \end{cases} \tag{6.3}$$

Figure 6.7 also illustrates these conditions. We can formulate the cost function as a sum of CAs, each of which is formulated as Figure 6.6 using these variables.

6.3.2 OPC Relaxation

As the VLSI feature sizes are substantially smaller than the lithography wavelength, the resolution enhancement techniques (RETs) such as Optical Proximity Correction (OPC) and Phase Shift Masks become essential to draw the patterns correctly. Due to these RETs, the VLSI lithography processes become more and more complex and the cost of the masks is expected to increase steeply along with the mask data size as shown in Figure 6.8[41]. Among several RETs, OPC is one of the main contributors to the mask cost, which includes the mask data preparation and the mask writing process costs. Both of the costs increase almost in proportion to the data size of the mask writer format, which is converted by fracturing the GDSII layout data into rectangles and trapezoids. Therefore, the OPC alleviation by modifying the layout has a significant effect on the mask data size reduction. Recently, a lot of papers have been published in the area of the lithography-aware design

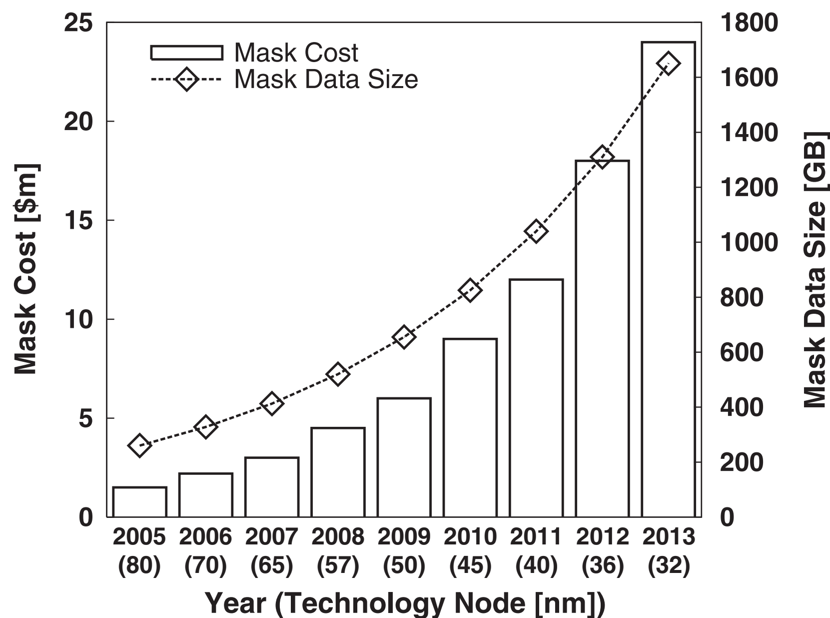


Figure 6.8 The expected mask cost and mask data size in the future technologies.

methodology[42, 43, 44, 45, 46, 16]. Among them, [42, 46] address the mask cost reduction problem considering the design intent. These OPC techniques are aware of the design informations, *e.g.*, timing slacks. Features in the layout which are not timing-critical might be expected to tolerate a larger degree of variation, and corrected by the less aggressive OPC. These papers showed significant reduction in OPC data volume with little or no increase in circuit delay. In this section, we will demonstrate the OPC relaxation in terms of the mask cost reduction using the proposed timing-aware cell layout de-compaction framework. The proposed de-compaction method expands the spacings between the polygons inside the layout and eases the optical proximity effects under given timing constraints under timing constraints. The layout after de-compaction can be printed correctly even by the modest OPC. Therefore, we can generate the OPC-friendly cell layouts and can reduce the OPC data size in terms of the fractured mask data size using the proposed method.

Figure 6.9 illustrates the conceptual example of OPC data volume reduction. By expanding the spacings between the polygons, the optical proximity effect between these polygons is reduced and the pattern modification by OPC is also reduced. The complex OPC results lead to complex mask patterning and large mask data volume. Therefore, the OPC relaxation through de-compaction is effective for OPC data volume reduction. This space expansion procedure is almost same as the wire spreading procedures to minimize the critical area between two wire segments explained in the previous section. Although the wire spreading

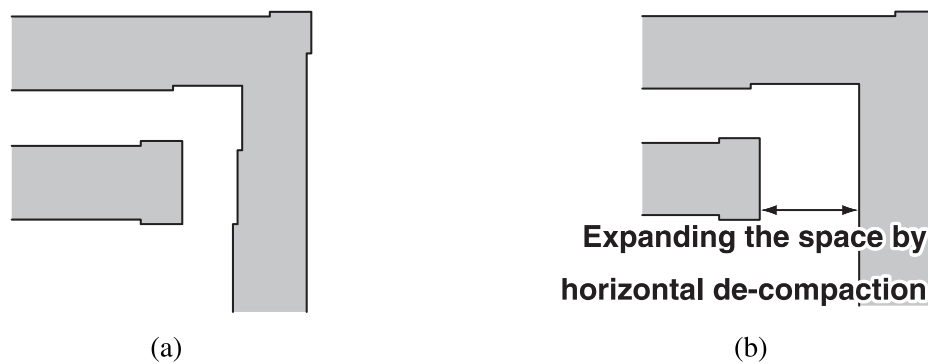


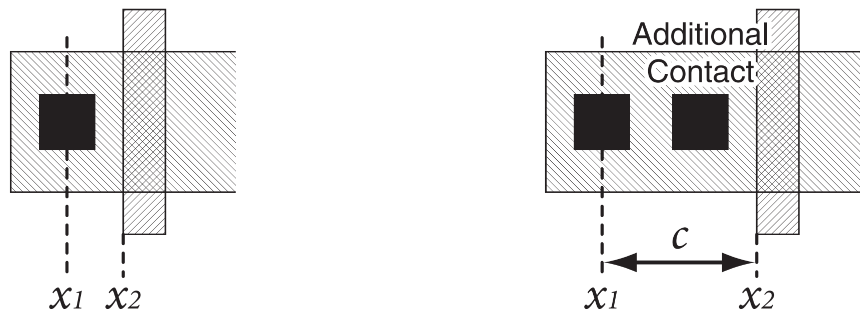
Figure 6.9 The conceptual example of OPC data volume reduction by expanding the spacing between the polygons.

Figure 6.10 The change of the OPC pattern from (a)hammer head style to (b)serif style.

procedure for critical area minimization also increases the wire width to avoid disconnection type failures, the wire width expansion is not effective for OPC data volume reduction. When the width of the wire increases wider than some threshold value, the OPCed pattern of this wire changes from so-called hammer-head style to serif style pattern as shown in Figure 6.10, and results in the data volume increase. Therefore, the proposed OPC-relaxation method executes only wire space expansion during de-compaction in the same manner as the previous section, whereas the width of the wire segments keep the same value as those of before the de-compaction.

6.3.3 Redundant Contact Insertion

In this section, we will explain how to maximize the number of the additional redundant contacts under timing and area constraints. In addition to the critical area based on the random defect related faults, there are a lot of other yield loss reasons in reality. Contact/Via open failure is one of the most important[47, 48, 49]. Due to various reasons such as cut misalignment in a manufacturing process, electro-migration and thermal stress, a contact may fail partially or completely. A well-known method to improve the contact yield is to



(a) No additional contact ($r_i = 1$). (b) An additional contact can be inserted ($r_i = 0$).

Figure 6.11 Formulation of the redundant contact insertion.

add a redundant contact adjacent to a single contact. Also for standard cells, the redundant contact insertion is commonly used to improve their yield[50]. In this section, we explain the formulation of the redundant contact insertion for standard cells under the timing-aware de-compaction framework.

The proposed method inserts redundant contacts adjacent to as many single contacts as possible under given timing and area constraints. Since two parallel contacts are assumed to be enough for yield enhancement, at most one redundant contact is inserted adjacent to each single contact. In our formulation, a Boolean variable is assigned to every single contact. Assume that the variable is written as r_i for a single contact i, this variable takes the value of 0 if a redundant contact is inserted adjacent to this single contact, otherwise 1. Therefore, to maximize the number of the redundant contacts, the cost function of the LP is formulated as follows.

$$Minimize : \sum_i r_i \tag{6.4}$$

We use Figure 6.11 to explain the constraint of the variable r_i. Figure 6.11 shows the simplified example of the contact insertion to the diffusion area. If the space between a contact and a gate becomes larger than the space for an additional contact, r_i takes the value of 0. This constraint is written as

$$\begin{cases} r_i = 1 \quad (x_2 - x_1 < c), \\ r_i = 0 \quad (x_2 - x_1 \geq c) \end{cases} \tag{6.5}$$

where c is the distance required for an additional contact. Under the condition of minimizing r_i, this constraint is expressed as follows.

$$x_2 - x_1 - c + c \times r_i \geq 0 \tag{6.6}$$

This constraint is formulated for every single contact inside the given layout. Using these constraints and the cost function, the proposed method inserts the redundant contact as many as possible under given timing constraints. Of course this optimization constraint can be used in linear combination with the critical area minimization which is explained in Section 6.3.1. Since the variables r_i are Boolean, while all the other variables are real numbers, the problem solved in the proposed method can be referred to as Mixed Integer Linear Programming (MILP).

6.3.4 Gate Layout Pattern Regularity Enhancement

Not only the functional yield considered in the previous sections, but also the parametric yield of the circuits become more and more important in the recent VLSI processes. As the VLSI technologies scale down to the subwavelength lithographic regime, the process parameter variations lead to severe variability of chip performances and both timing and power of integrated circuits are increasingly affected by process variations. Among many process variation sources, one of the most important parameter variations affecting circuit performance is the gate length variation of MOS transistors, since it directly affects both transistor switching speed and leakage power[51, 48, 52]. However, the gate Critical Dimension (CD) control in the future technologies is projected as one of the most difficult problems in International Technology Roadmap for Semiconductors(ITRS)[41] and indicated that the manufacturable solutions are not known after 2007 as shown in Figure 6.12.

Although the impacts from both systematic and random CD variations become greater and greater, it is reported that more than 50% of transistor gate length variations are due to systematic sources[52]. Empirical data show that the intra-die systematic CD variations resulting from the layout pattern non-uniformity are becoming comparable to, even dominant over lot to lot, wafer to wafer, and die to die variations[53]. Moreover, CD variation due to the depth of focus variation is known to show different behavior dependent on the pitches of CD patterns[15] and introduces the significant problems about the performance predictability. To reduce the systematic process variations, the regular layout has been shown to be effective[54]. Therefore, various regular design styles have been suggested[55, 56] and Restrictive Design Rules (RDR) [57, 14] have also been utilized to improve pattern printability and reduce variations. As for the standard cell layouts, the technology migration technique with on-pitch constraint has been proposed in [58]. Although the conventional technology migration techniques use linear shrink in conjunction with legalization technique to clean up

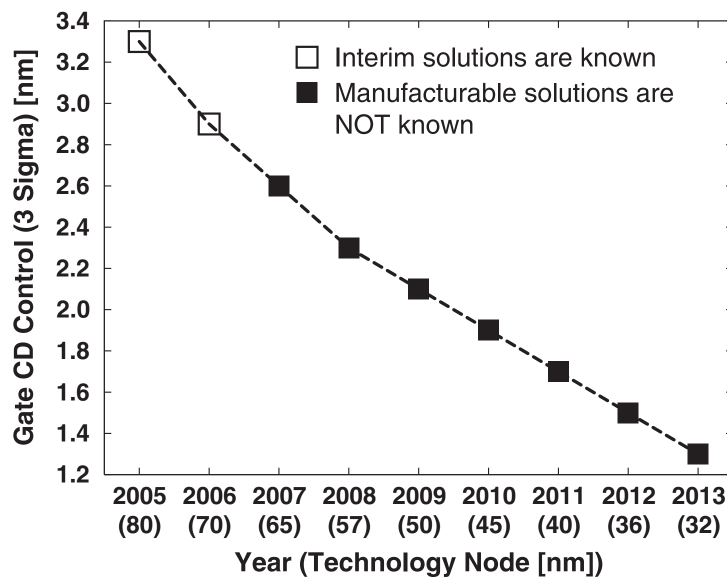


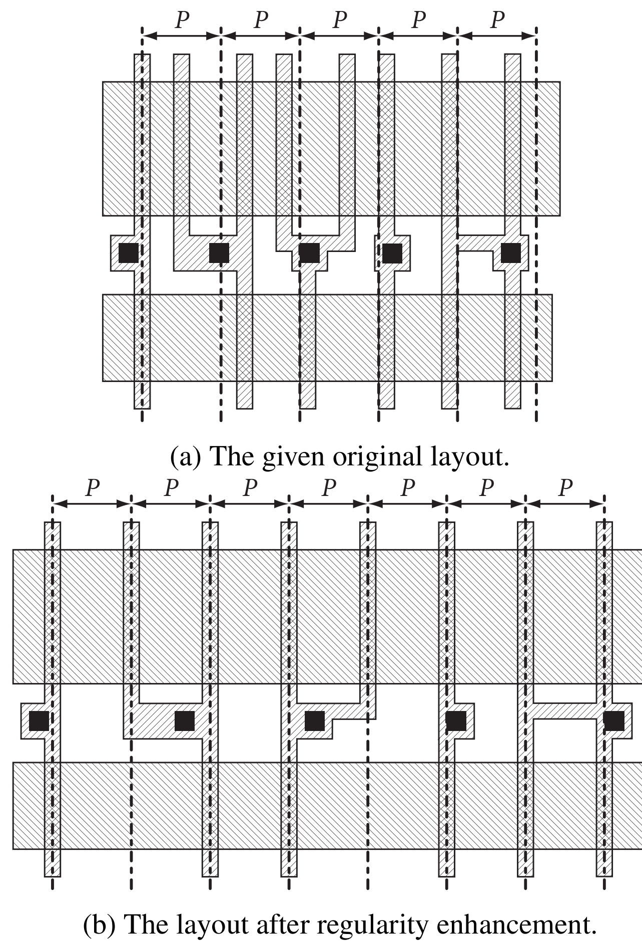
Figure 6.12 Required gate CD control in the future technologies.

the ground rule violations, this technique legalizes a given layout in order to meet on-pitch constraints with minimum layout perturbation.

In this section, we propose a gate layout pattern regularity enhancement method to reduce the systematic variation of the gate CD. The proposed method performs a de-compaction of a cell layout considering the on-pitch constraints and enhances the pattern regularity under given timing constraints using LP. In contrast to [58] which minimizes the layout perturbation to meet the design rules and on-pitch constraints during technology migration, the proposed method minimizes the regularity cost under the design rules and timing constraints during de-compaction. Using the proposed timing-aware regularity enhancement method, we can explore the trade-off between performance and regularity cost, and can pick up the regularity variants from the trade-off curve.

To enhance the regularity, gate patterns in a cell layout have to meet three restrictions: fixed gate length, single orientation, and uniform pitch. We assume that the gate patterns in a given cell layout always have first two factors: fixed gate length and single orientation. Figure 6.13 shows the conceptual example of the regularity enhancement. As shown in this figure, the gate patterns in a layout are aligned in regular pitch during de-compaction under design rule and timing constraints.

The cost of the regularity is defined as shown in Figure 6.14. A piecewise linear cost function is used in the proposed method. When the gate layout pattern is placed in the



(a) The given original layout.

(b) The layout after regularity enhancement.

Figure 6.13 The conceptual example of gate layout pattern regularity enhancement for the regular pitch P by the proposed regularity enhancement method.

regular pitch, the cost is 0. As the distance between the gate and the regular pitch increases, the regularity cost increases linearly. This cost function is equivalent to the distance from the nearest regular pitch. The LP formulation of this cost function is explained in the following paragraph.

Let E_{gate} denote the set of layout elements of transistor gate, the regularity cost used in this paper is written as

$$\sum_{E_i^g \in E_{gate}} |x_i^g - k_i P|, \tag{6.7}$$

where x_i^g is an x coordinate of a gate element E_i^g, P is the regular pitch and k_i is an integer variable. This objective function is linearized by introducing two variables L_i and R_i for each

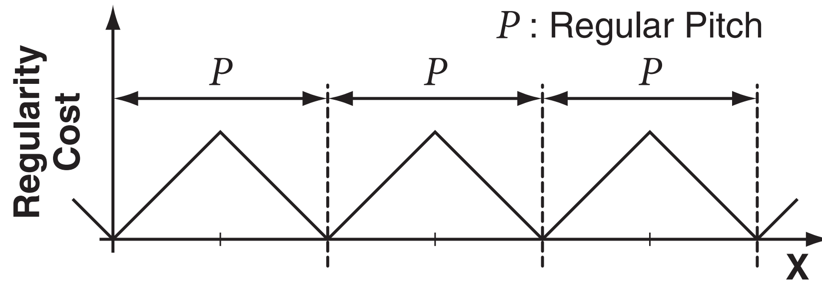


Figure 6.14 The regularity cost function used in the proposed regularity enhancement method.

element E_i^g. The linear formulation of this minimization constraint is written as follows.

$$Minimize \ : \ \sum_{E_i^g \in E_{gate}} (R_i - L_i) \tag{6.8}$$

$$Subject\ to \ : \ L_i \le x_i^g, \ \ L_i \le k_i P \tag{6.9}$$

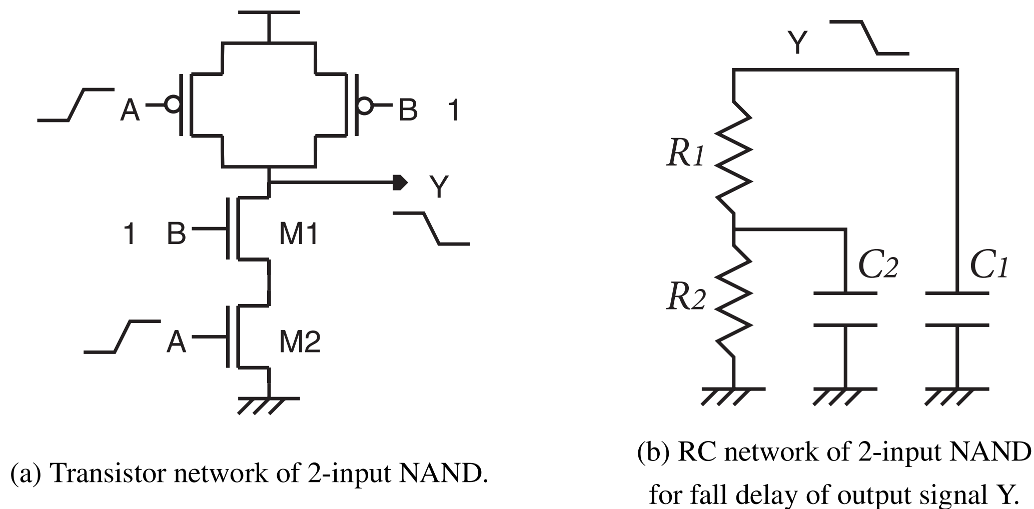
$$R_i \ge x_i^g, \ \ R_i \ge k_i P \tag{6.10}$$

In this formulation, L_i is smaller than both x_i^g and $k_i P$, whereas L_i is maximized by the minimization constraint (6.8). Therefore, L_i is always equal to the smaller one of x_i^g or $k_i P$. On the other hand, R_i is larger than both x_i^g and $k_i P$, whereas R_i is minimized. Therefore, R_i is always equal to the larger one of them. Since k_i is an integer variable, the distance between a gate and the nearest regular pitch is calculated. By minimizing this cost function, we can optimize the regularity of the given cell layout. Since the proposed formulation includes the integer variables k_i whereas all the other variables are real numbers, this formulation is also referred to as Mixed Integer Linear Programming (MILP).

6.4 Delay Model

We need a linear delay model to formulate the timing constraint as LP. The well-known linear timing approximation is Elmore delay model. Figure 6.15 (a) illustrates a simple example of 2-input NAND. When the input signal A rises from logic level 0 to 1, then the output signal Y falls to logic level 0. In this situation, this transistor network is replaced by an RC network shown in Figure 6.15 (b) which consists of ON resistors R_1 and R_2 of the transistors M1 and M2, respectively, and parasitic capacitors C_1 and C_2. Using Elmore delay model, we can calculate the fall delay of the output signal Y as follows.

$$Delay_{A \to Y} = (R_1 + R_2) \times C_1 + R_2 \times C_2 \tag{6.11}$$



(a) Transistor network of 2-input NAND.

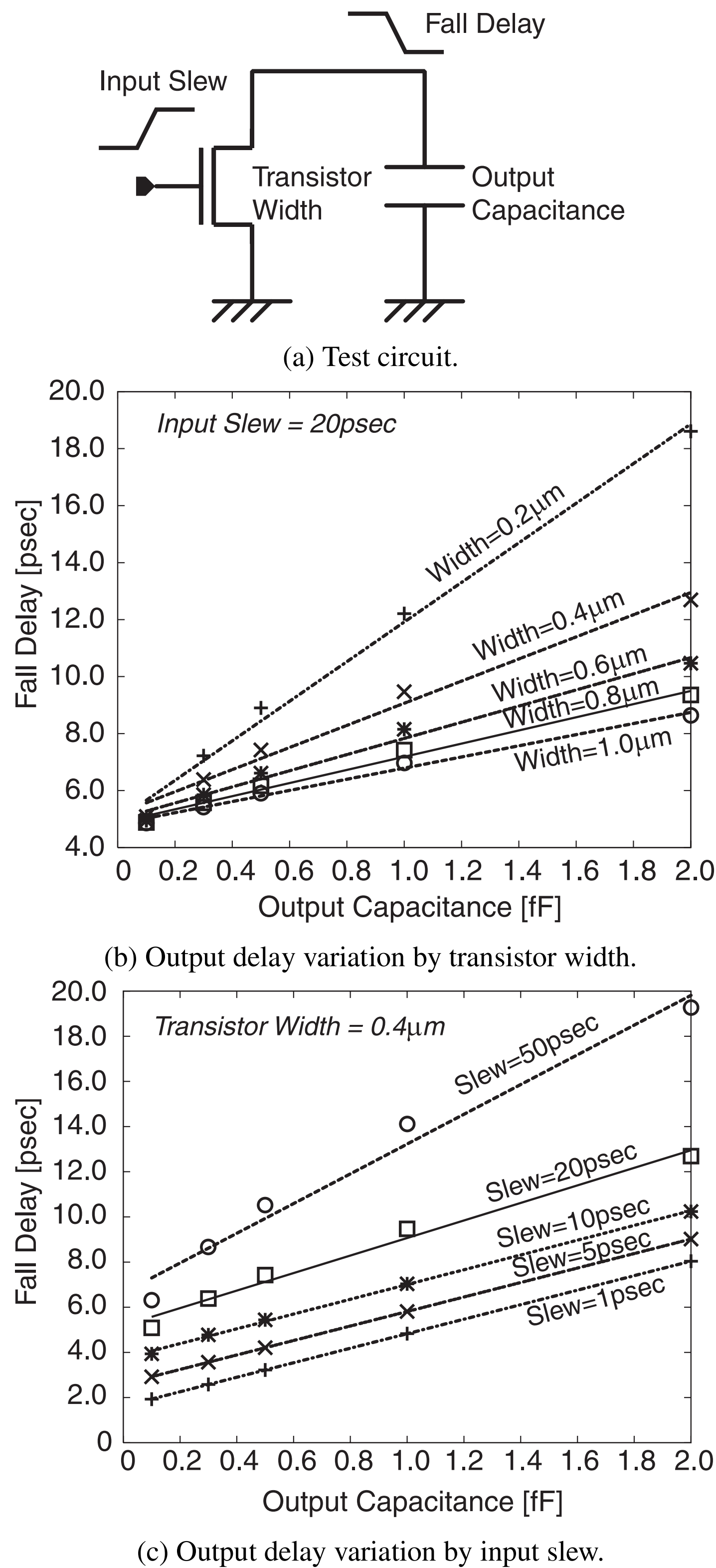
(b) RC network of 2-input NAND for fall delay of output signal Y.

Figure 6.15 An example of 2-input NAND and its RC network for calculating Elmore delay.

However, this model is not accurate enough to model the transistors of the recent deep submicron technologies. Therefore, we develop a new delay model which only calculates the delay difference induced by the difference of parasitic elements after de-compaction. Since the proposed method performs a de-compaction of a given original layout, we can extract the original parasitic elements and simulate the original delay values from this layout using a SPICE-like circuit simulator. Once the original delay value is simulated, a delay difference by the difference of the parasitic capacitances is approximated by a linear function. Figure 6.16 shows the graphs of fall delay of an N type transistor versus output capacitance. The schematic of the simulated circuit is shown in Figure 6.16 (a). Figure 6.16 (b) shows the delay variation by changing the width of the transistor and (c) shows the delay variation by changing the value of gate input slew. In both cases, the delay increases almost in proportion to the output capacitance, and the slope values are different from each other. In addition, the slew value of the output signal can also be approximated as a linear function to the output capacitance. The slope value of the slew is also dependent on the transistor width and the gate input slew. Figure 6.17 (a) shows the output slew variation by changing the width of the transistor and (b) shows the output slew variation by changing the value of gate input slew. Therefore, these slope values are calculated in advance and stored as a table of transistor width and input slew for P and N type transistors, respectively.

Using these slope values, the procedure of delay or slew increase calculation is written as follows.

1. Convert a given transistor network into an RC network and formulate the Elmore delay

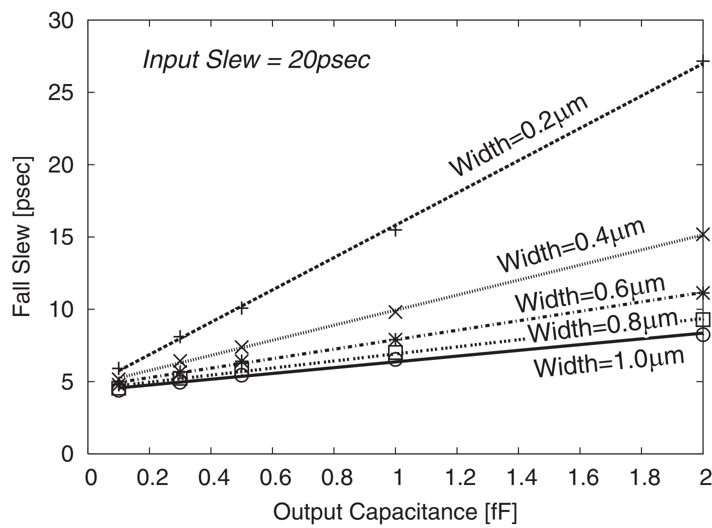


(a) Test circuit.

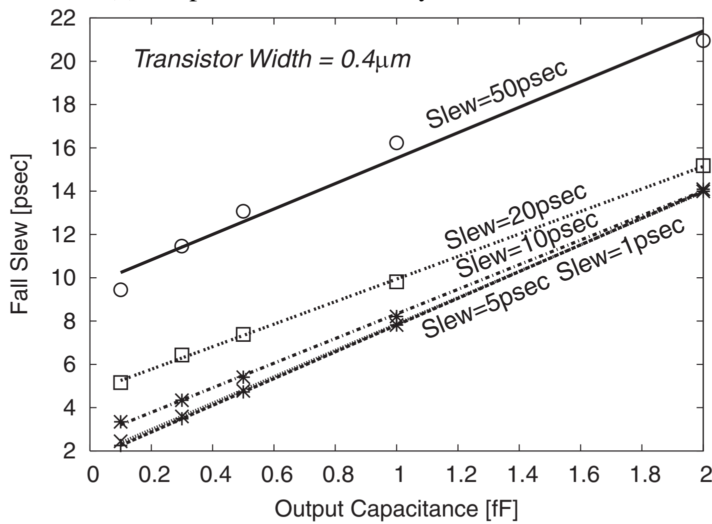
(b) Output delay variation by transistor width.

(c) Output delay variation by input slew.

Figure 6.16 Preliminary results of delay variation by changing the transistor width and the input slew.



(a) Output slew variation by transistor width.



(b) Output slew variation by input slew.

Figure 6.17 Preliminary results of output slew variation by changing the transistor width and the input slew.

function in the same manner as shown in Figure 6.15.

2. Replace the values of ON resistors by the slope values that are determined by the values of the width and the gate input slew of each transistor.

3. Replace the values of parasitic capacitors by the difference of each parasitic capacitor after de-compaction.

For example of Figure 6.15, the fall delay increase $\Delta Delay_{A\rightarrow Y}$ is described as a linear func-

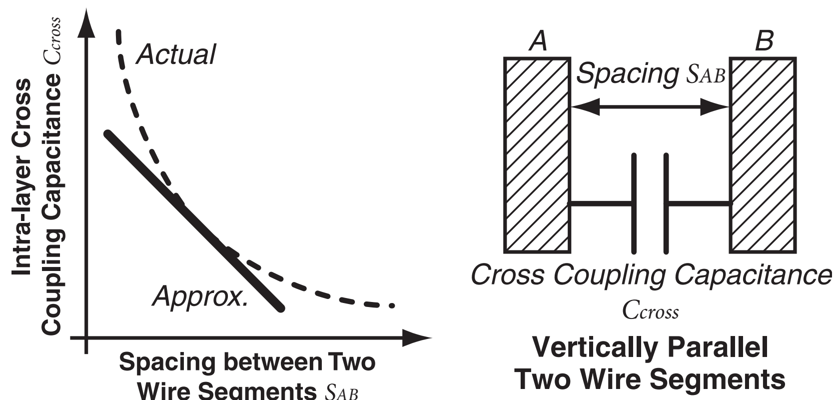


Figure 6.18 The linear approximation of the intra-layer cross coupling capacitance between two vertically parallel wire segments.

tion of parasitic capacitance differences as follows,

$$\Delta Delay_{A\to Y} = (k_1 + k_2) \times \Delta C_1 + k_2 \times \Delta C_2 \tag{6.12}$$

where k_1 and k_2 are the delay slope values of the transistors M1 and M2, respectively, and ΔC_1 and ΔC_2 are the differences of the parasitic capacitors C_1 and C_2 after de-compaction, respectively. Using this model, we can describe the timing constraints by a linear inequation as

$$Delay^{initial}_{A\to Y} + \Delta Delay_{A\to Y} \leq Delay^{target}_{A\to Y} \tag{6.13}$$

where $Delay^{initial}_{A\to Y}$ and $Delay^{target}_{A\to Y}$ are the original and the target cell delay, respectively. The increase of the output slew is also modeled in the same manner.

The difference of the parasitic capacitances also has to be linearly modeled by the coordinates of polygons in the original layout. The diffusion capacitance is calculated by a linear function of the area and the perimeter of the diffusion region. The parasitic capacitance of wires to ground or between overlapped layers is also calculated by a linear function of the area of the overlapped regions. For intra-layer cross coupling capacitances, it is easy to extract the capacitances between horizontally parallel wire segments because the de-compaction of the horizontal direction only increases the length of the parallel wires and this type of capacitance is mainly proportional to the length of the parallel wires. However, the coupling capacitances between vertically parallel wire segments are not so easy to calculate because the distance between these two wires are increased by the horizontal de-compaction and the capacitance value is not proportional but inversely proportional to the distance. Therefore, we approximate the value of this type of capacitance by linear function which is proportional to the

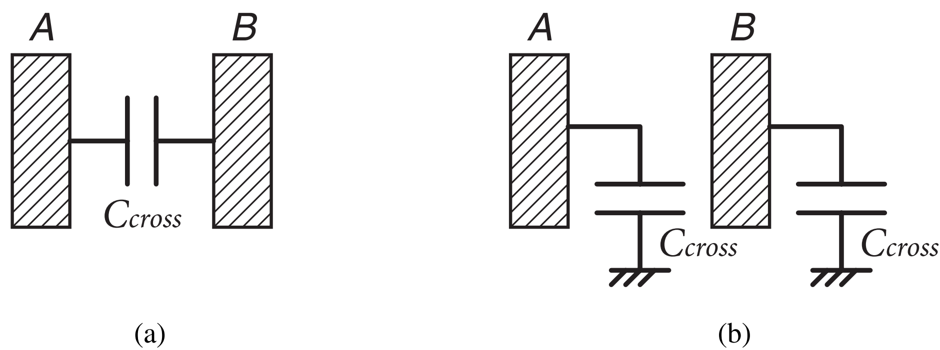


Figure 6.19 Approximation of the cross coupling capacitance between two signals A and B.

distance with negative slope value as shown in Figure 6.18. A capacitor between two signals is approximated by two capacitors from each signal to ground as shown in Figure 6.19. Both of them have the same capacitance as the original capacitor. Experimental results will show that these approximations are accurate enough to calculate the timing constraints under the proposed timing-aware de-compaction framework. At this stage, we can describe the timing constraints as linear functions of the coordinates of the polygons in the layout and can formulate them into the LP problem.

This model requires the slew of all the gate input signals in advance to determine the slope values of all the transistors. Therefore, this model describes the delay and slew value of single-stage transistor networks, since all the gate input waveforms are given for these cells. For multi-stage cells, on the other hand, the slew value of an inner node, which is an output of one stage and inputs to other stages, is not given in advance. Therefore, we have to approximate the change of the slew value of this inner node to apply the proposed model to multi-stage cells. The preliminary experiment, however, shows that the changes of the slew values of inner nodes caused by our de-compaction method are within several percents and these slew changes have little effects on the delay changes. For this reason, we use the slew values of the inner nodes of the original layout for the slew values of the inner nodes during de-compaction in the case of the multi-stage cells. These original slew values of the inner nodes are simulated and saved at the same time of the original delay simulation. Experimental results will show that even this simple approximation realizes accurate timing constraints for the multi-stage cells.

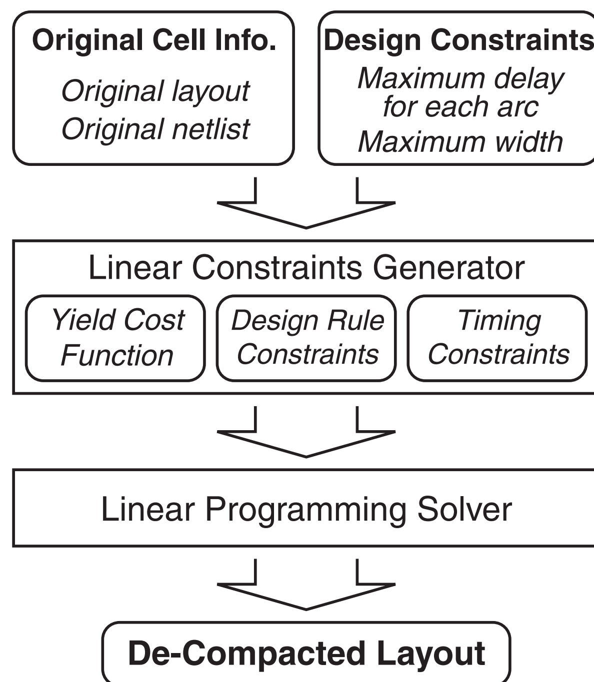


Figure 6.20 The overall flow diagram of the proposed timing-aware yield enhancement method.

6.5 Overall Flow

Figure 6.20 shows the overall flow diagram of the proposed method. The input to the proposed method is the original cell information and design constraints. The original cell information includes the original GDSII format cell layout for polygon information and the netlist of the cell in SPICE format for transistor connection information. Design constraints include the target cell delay for each timing arc and the maximum cell width. A timing arc is defined as a signal flow from an input to an output on a cell, *e.g.*, A rise → Y fall. These maximum delay and area values are given as plain text format in a specific syntax. Although the maximum width constraint was not explained in the previous sections, it can be formulated as a part of the design rule constraints. Using these informations, the linear constraints generator, which implements the proposed timing-aware cell layout de-compaction for yield optimization, formulates the constraints explained in the previous sections and generates an input LP file for the LP solver. The yield cost function is selected from CA minimization, OPC relaxation, redundant contact insertion, or gate layout pattern regularity enhancement. Of course, the linear combination of these cost functions can be used as the yield cost function. Then, the LP solver searches for the solution of the generated LP problem. Finally, a GDSII file of the cell layout after de-compaction is created from the solution of LP solver

Table 6.1 The topological characteristics of the benchmark circuits used for the experiment of the critical area minimization.

Circuit	*Explanation*	*#stage*	*#trans.*
NAND3_1	3-input NAND	1	6
NAND3_2	3-input NAND (buffered)	1	12
NAND4_3	4-input NAND (buffered)	1	36
NOR4_1	4-input NOR	1	8
NOR4_2	4-input NOR (buffered)	1	28
ON2222_3	A series-parallel circuit	1	56
ADDH_1	Half Adder	4	14
OR3_2	3-input OR (buffered)	2	23

which is equivalent to the coordinates of the polygons inside the layout. During final GDSII file generation, the width of the outside frame of the generated cell is enlarged to fit the multiple of the fixed pitch to be used as a standard cell layout.

6.6 Experimental Results

6.6.1 Critical Area Minimization

The proposed timing-aware de-compaction method for critical area minimization was implemented to show its effectiveness. In this experiment, we used ILOG *CPLEX 9.1*[27] for an LP solver and 8 cells from a standard-cell library of a 90nm technology were used as benchmarks. Among these 8 cells, 6 cells are single-stage and the others are multi-stage. Tables 6.1 and 6.2 summarize the characteristics of these cells. These tables show the circuit name, the explanation of each circuit, the number of the stages inside each circuit, the number of transistors, the original cell delay value, the original cell area, and the original critical area. $Delay_{orig}$ column shows the original delay of a timing arc. The delay values of these arcs were constrained in this experiment. To calculate the original cell delay, a netlist with parasitic capacitances is extracted using Mentor Graphics *Calibre xL*[59] and simulated using Synopsys *HSPICE*[60]. The tables of the delay slope value(Figure 6.16) for P and N type transistors are also calculated using *HSPICE* in advance. In this experiment, we did not connect an additional capacitor to the output net to clarify the effect of the intra-cell parasitic elements. The defect size used in this experiment is 1.5 times larger than the minimum

Table 6.2 The performance characteristics of the benchmark circuits used for the experiment of the critical area minimization.

Circuit	$Delay_{orig}$ *[psec]*	$Area_{orig}$ $[\mu m^2]$	CA_{orig} $[\mu m^2]$
NAND3_1	33.05	3.70	0.97
NAND3_2	34.56	6.53	1.97
NAND4_3	53.11	22.15	10.97
NOR4_1	73.65	4.76	1.30
NOR4_2	65.95	17.41	8.47
ON2222_3	64.83	28.75	9.92
ADDH_1	78.56	8.77	2.38
OR3_2	92.85	12.52	4.17

width/space of the first metal layer and the CAs are calculated only for the first metal layers. All the experiments were conducted on a Linux machine with Xeon 3.4GHz processor and 2GB of RAM.

Tables 6.3 and 6.4 summarize the results of the proposed timing-aware de-compaction method. These tables show the target and the actual delay value, the cell area, and the CA of the generated cell layouts. Delay error is calculated by $(t_{target} - t_{actual})/t_{target} \times 100$, where t_{target} and t_{actual} are the target and the actual delay, respectively. "No constraint" in the column of *Target* means that no timing constraints were set in this case. Because the vertical CA decreases but the horizontal CA possibly increases by horizontal de-compaction, there must be an optimal value of the total CA. The value of CA in the no constraint case is the minimum CA value for each cell. The runtime to create the de-compacted layout is about 0.1 second even for the largest example of ON2222_3 which consists of 56 transistors. The runtime for creating the tables of slope values and the first *HSPICE* simulation for each cell is excluded because they are conducted just once in advance. The delay values of the generated layouts are also simulated by *HSPICE* using a netlist extracted by *Calibre xL*. The errors of the target and actual delay values are less than 1% for most cases and the average absolute error is 0.49%. Figure 6.21 plots the target and the simulated delay values in the case of the single-stage cell NOR4_1 when the input signal to the P type transistor connected to VDD falls from logic level 1 to 0. The simulated delay values show good accordance with the target delay values. Figure 6.22 also plots the target and the simulated delay values in the case of the multi-stage cell ADDH_1. These results show that the developed delay model is accurate

Table 6.3 The delay accuracy of the proposed timing-aware critical area minimization method.

Circuit	*Target [psec]*	*Actual [psec]*	*error [%]*
NAND3_1	35	34.98	0.06
	37	36.77	0.63
	No constraint	37.92	—
NAND3_2	35	34.95	0.14
	36	35.76	0.67
	No constraint	36.54	—
NAND4_3	54	53.81	0.35
	55	54.72	0.51
	No constraint	55.83	—
NOR4_1	76	75.59	0.54
	80	79.62	0.48
	No constraint	81.33	—
NOR4_2	67	67.47	-0.70
	70	68.87	1.64
	No constraint	70.06	—
ON2222_3	66	65.63	0.56
	67	66.18	1.25
	No constraint	67.65	—
ADDH_1	79	79.08	-0.10
	80	80.03	-0.05
	No constraint	80.91	—
OR3_2	94	94.06	-0.06
	96	95.88	0.13
	No constraint	97.91	—
average	—	—	0.49

Table 6.4 Results of the critical area minimization by the proposed method.

Circuit	*Target [psec]*	*Area [μm²]*	*increase [%]*	*CA [μm²]*	*reduction [%]*
NAND3_1	35	5.70	54.05	0.47	51.55
	37	5.70	54.05	0.47	51.55
	No constraint	5.70	54.05	0.47	51.55
NAND3_2	35	8.11	24.20	1.56	20.81
	36	8.49	30.02	1.47	25.38
	No constraint	8.49	30.02	1.47	25.38
NAND4_3	54	23.87	7.77	10.29	6.20
	55	24.81	12.01	10.10	7.93
	No constraint	25.58	15.49	10.00	8.84
NOR4_1	76	5.71	19.96	0.92	29.23
	80	6.02	26.47	0.85	34.61
	No constraint	6.02	26.47	0.85	34.61
NOR4_2	67	19.33	11.03	7.59	10.39
	70	20.25	16.31	7.30	13.81
	No constraint	20.76	19.24	7.27	14.17
ON2222_3	66	31.20	8.52	9.11	8.17
	67	31.29	8.83	8.94	9.88
	No constraint	31.68	10.19	8.86	10.69
ADDH_1	79	10.94	24.74	1.87	21.43
	80	10.74	22.46	1.82	23.53
	No constraint	10.74	22.46	1.82	23.53
OR3_2	94	14.91	19.09	3.04	27.10
	96	15.62	24.76	2.94	29.50
	No constraint	15.78	26.04	2.93	29.74
average	—	—	25.50*	—	24.81*

∗ : An average of the no constraint cases

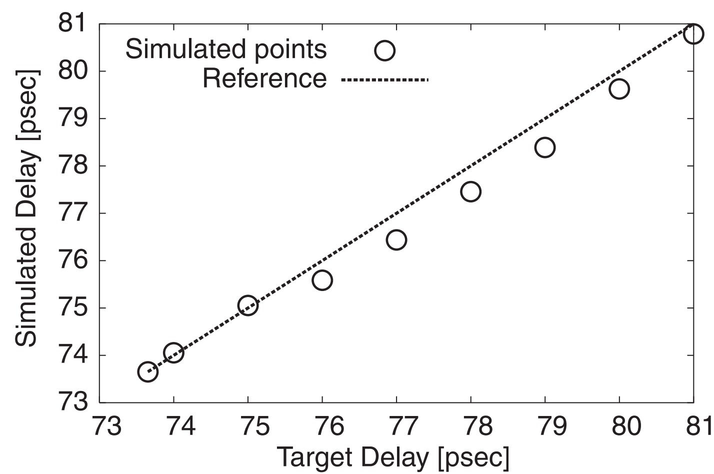


Figure 6.21 Accuracy of the proposed delay model in the case of the single-stage cell NOR4_1.

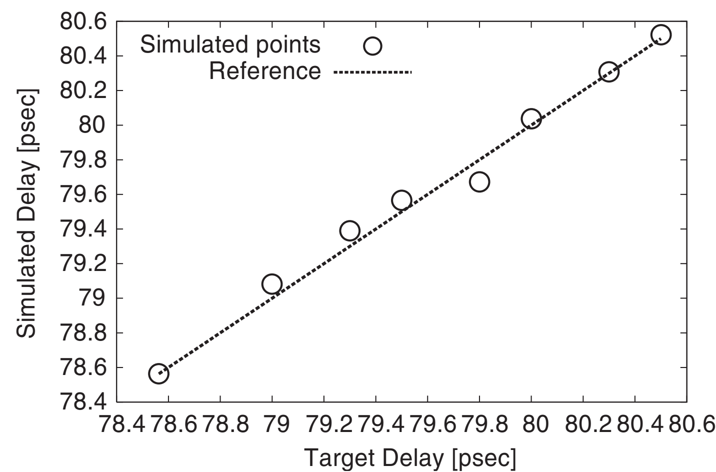
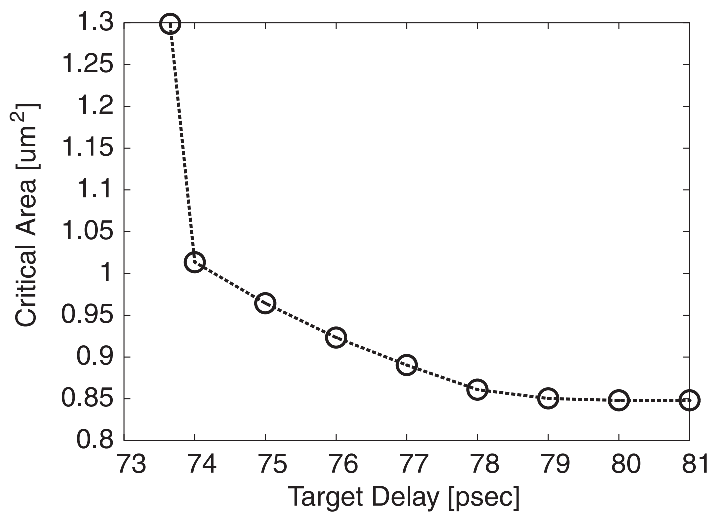


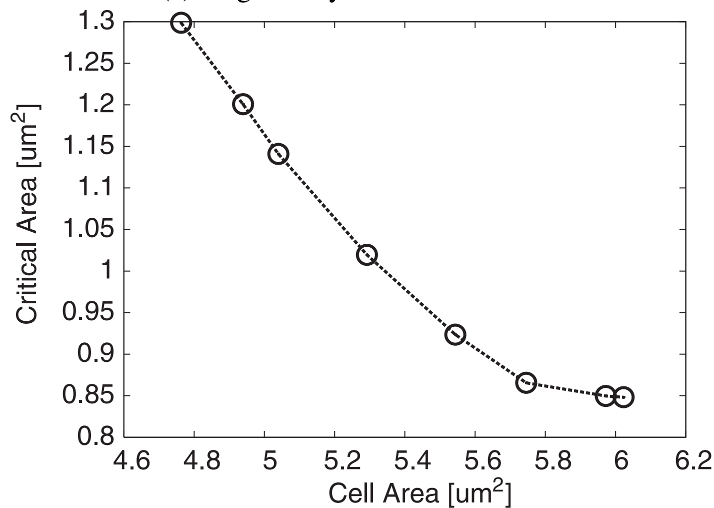
Figure 6.22 Accuracy of the proposed delay model in the case of the multi-stage cell ADDH_1.

enough for the proposed de-compaction method and can be applied to both single and multi-stage cells.

After de-compaction, the cell areas increase about 10 to 50% and the CAs decrease about 10 to 50% depending on the cells and the constraints. The average values of the cell area increase and the CA reduction without timing constraints are about 26% and 25%, respectively. As the timing constraint becomes tight, the cell area increase and CA reduction become small. In the cases of NAND3_1, NAND3_2, NOR4_1, and ADDH_1, the proposed method creates the cells with smaller delay than the no constraint case while their area and CA is equal to that of the no constraint case. As a special case, cell area of ADDH_1 with 79 psec delay constraint is larger than that of the 80 psec constraint. These results show the fact that the



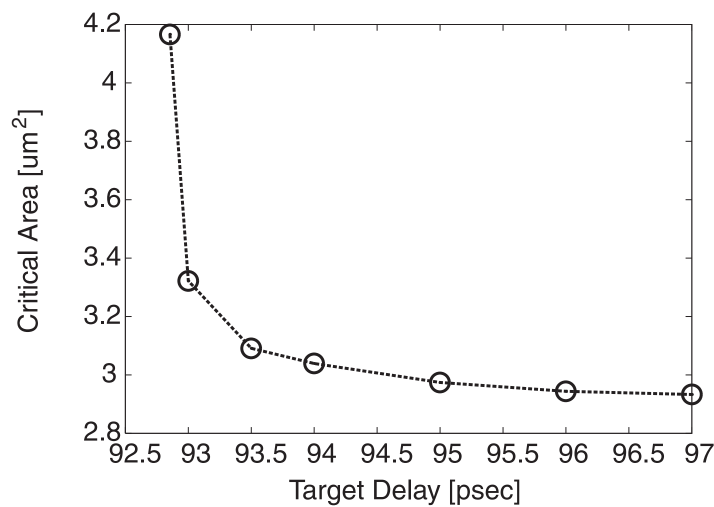
(a) Target delay versus critical area.



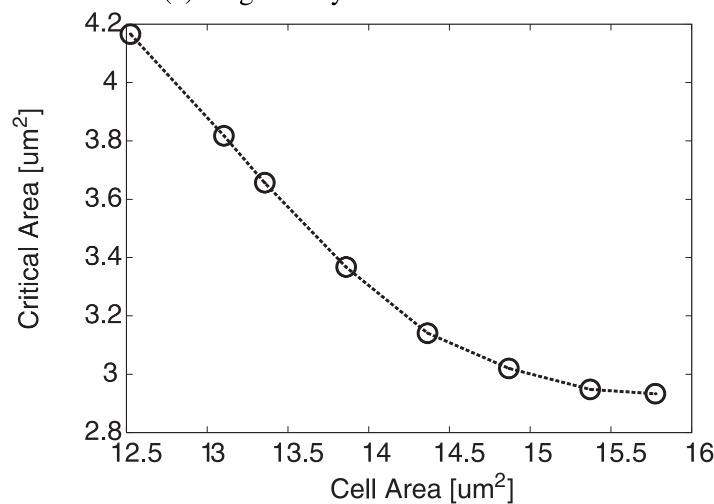
(b) Cell area versus critical area.

Figure 6.23 Trade-off curves of (a)target delay versus CA and (b)cell area versus CA in the case of the single-stage cell NOR4_1.

layout impacts on the timing are different by location, and the significance of the timing consideration during layout de-compaction. Figure 6.23 and Figure 6.24 show the trade-off curves of target delay versus CA and cell area versus CA in the cases of the single-stage cell NOR4_1 and the multi-stage cell OR3_2, respectively. As shown in these figures, the critical areas are minimized under given timing or area constraints for both cases. The conventional simple de-compaction method can not create these various yield-optimized cell layouts. On the other hand, the proposed method can pick up the yield and performance variants of a cell layout from these trade-off curves.



(a) Target delay versus critical area.



(b) Cell area versus critical area.

Figure 6.24 Trade-off curves of (a)target delay versus CA and (b)cell area versus CA in the case of the multi-stage cell OR3_2.

6.6.2 OPC Relaxation

This section shows the experimental results of the proposed timing-aware OPC relaxation method. In this experiment, we also used ILOG *CPLEX 9.1*[27] for an LP solver, and 20 cell layouts from a standard-cell library of a 90nm technology were used as benchmarks. The maximum wiring space allowed during de-compaction is defined as 1.5 times larger value than the minimum space of the first metal layer, whereas the width of the wire segments keep the same value as those of before the de-compaction as explained in Section 6.3.2. The OPC cost in terms of the fractured mask data size is calculated only for the first metal layer. All

the experiments were conducted on a Linux machine with Xeon 3.4GHz processor and 2GB of RAM. The runtime to perform a cell layout de-compaction was less than 0.1 seconds even for the largest example which consists of 32 transistors.

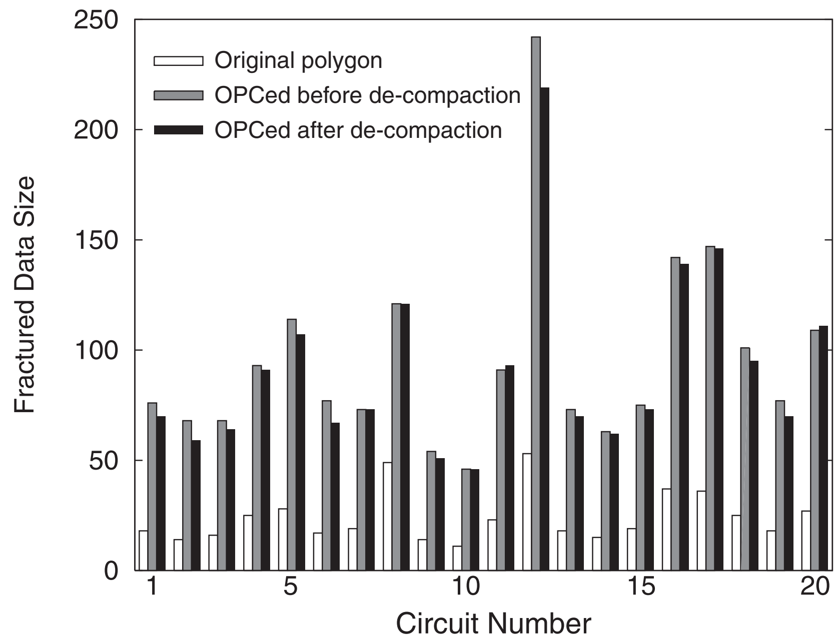
The fractured mask data size comparison between before and after de-compaction is shown in Figure 6.25. These two graphs show the fractured mask data size for 20 cell layouts in 90nm and 65nm technology, respectively. The 65nm cell layouts used in this experiment are prepared by shrinking the 90nm cell layouts. The OPC simulation and the fractured mask data size calculation is conducted using Mentor Graphics *Calibre*[61]. The lithographic condition used in this experiment is summarized as follows.

- Lithography Wavelength: 193nm
- Numerical Aperture: 0.8
- Mask Reduction Factor: 4

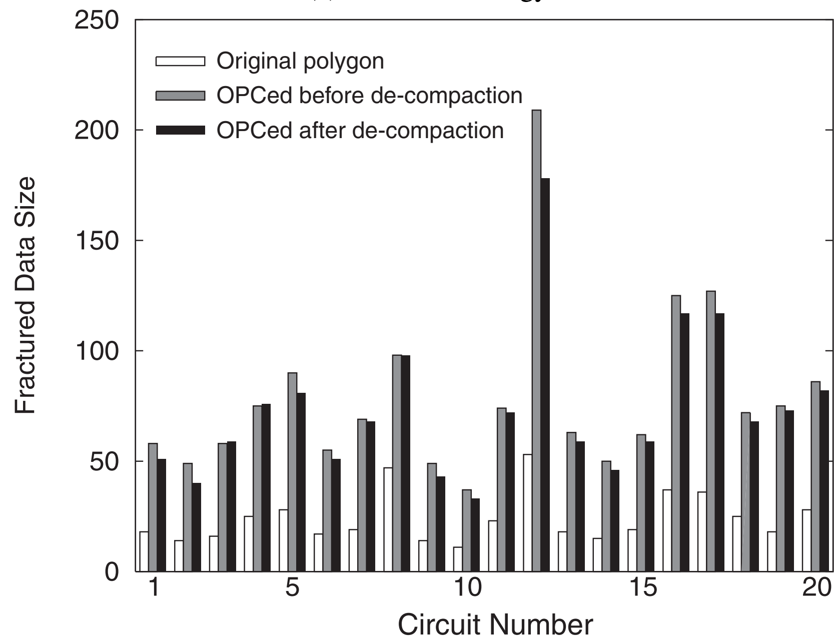
In this experiment, all cells are allowed 10% delay increase during de-compaction. However, 13 cells among 20 cells were enlarged to the maximum width through the de-compaction under this constraint. Therefore, the average delay increase of these 20 cells after de-compaction was about 6.73%. Under this condition, the average area overhead was 31.2%. As shown in these graphs, the fractured mask data size is reduced in the case of almost all cells for both 90nm and 65nm technologies. The data size was reduced 4.28% and 6.65% on an average in the case of 90nm and 65nm technology, respectively, and the maximum reduction ratios were 13.2% and 18.4%, respectively. The reduction of fractured mask data size directly translates to lower costs through shorter mask data preparation and mask writing time, and higher mask yield.

Figure 6.26 shows the expected maximum and average fractured mask data size reduction ratio depending on the technology nodes from 90nm to 65nm. The cells used in this experiment are also prepared by shrinking the 90nm cell layouts. All the simulations are conducted under the same lithographic conditions as explained before. As shown in this figure, the proposed de-compaction method for OPC mask data volume reduction is expected to be more effective in the small feature size technology.

Figure 6.27 illustrates an example of OPC results before and after de-compaction in the case of 65nm cell layout. From this figure, we can see that some pattern modifications by OPC is effectively removed by the de-compaction with small delay increase, particularly on the corner of wires.



(a) 90nm technology.



(b) 65nm technology.

Figure 6.25 The fractured mask data size of 20 cells in (a)90nm and (b)65nm technology in the case that 10% delay increase is allowed.

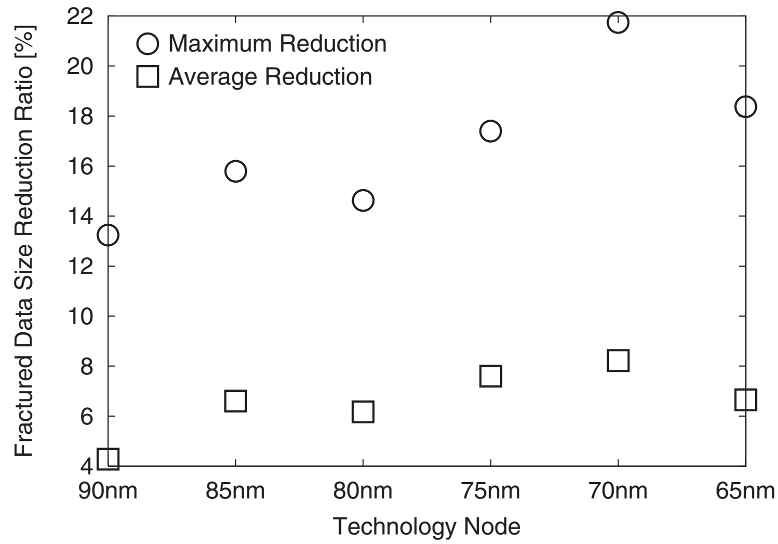
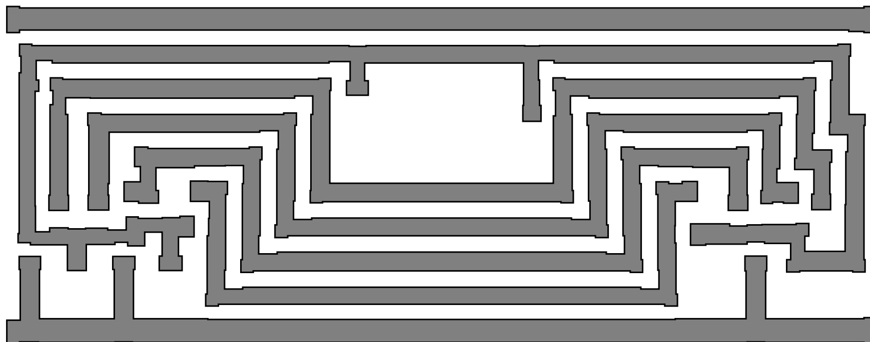
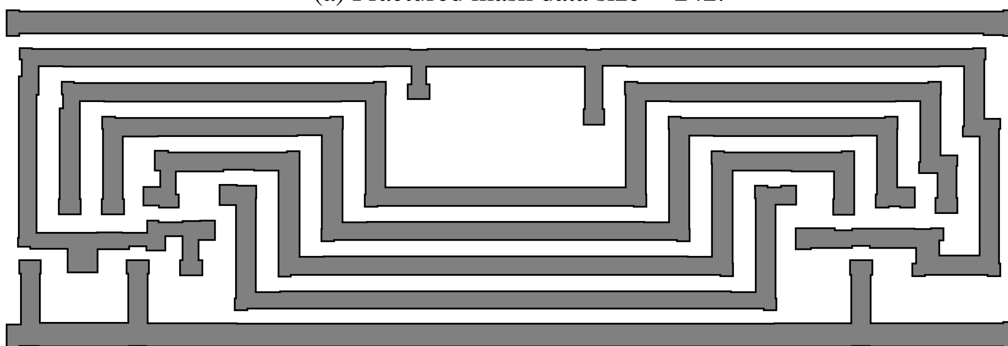


Figure 6.26 The reduction ratio of the fractured mask data size after de-compaction depending on the technology nodes.



(a) Fractured mask data size = 242.



(b) Fractured mask data size = 219, Delay increase = 3.4%, Area increase = 15.7%.

Figure 6.27 An example of OPC results in 65nm technology (a)before and (b)after de-compaction.

Table 6.5 The topological characteristics of the benchmark circuits used for the experiment of redundant contact insertion.

Circuit	*Explanation*	*#trans.*	*#stage*
NAND3_1	3-input NAND	6	1
NAND3_2	3-input NAND (buffered)	12	1
NAND4_3	4-input NAND (buffered)	36	1
NOR4_1	4-input NOR	8	1
NOR4_2	4-input NOR (buffered)	28	1
ON2222_3	A series-parallel circuit	56	1

Table 6.6 The performance characteristics of the benchmark circuits used for the experiment of redundant contact insertion.

Circuit	$Delay_{orig}$ *[psec]*	$Area_{orig}$ [μm^2]	*#Single Contact*
NAND3_1	33.05	3.70	7
NAND3_2	34.56	6.53	10
NAND4_3	53.11	22.15	20
NOR4_1	73.65	4.76	9
NOR4_2	65.95	17.41	28
ON2222_3	64.83	28.75	62

6.6.3 Redundant Contact Insertion

This section shows the experimental results of the proposed timing-aware redundant contact insertion method. In this experiment, we also used ILOG *CPLEX 9.1*[27] for an MILP solver and the 6 single-stage cells from a standard-cell library of a 90nm technology were used as benchmarks. Tables 6.5 and 6.6 summarize the characteristics of these cells. All the experiments were conducted on a Linux machine with Xeon 3.4GHz processor and 2GB of RAM. These tables show the circuit name, the explanation of each circuit, the number of transistors, the number of the stages inside each circuit, the original cell delay value, the original cell area, and the number of the single contacts inside the original cell layouts. $Delay_{orig}$ column shows the original delay of a timing arc. The delay values of these arcs were constrained in this experiment.

Tables 6.7 and 6.8 show the results of the proposed timing-aware redundant contact insertion method. These tables show the target and the actual delay value, the cell area, and the number of the additional contacts inside the generated cell layouts. Delay error is calculated

Table 6.7 The delay accuracy of the proposed timing-aware redundant contact insertion method.

Circuit	*Target [psec]*	*Actual [psec]*	*error [%]*
NAND3_1	36	36.18	-0.50
	39	39.65	-1.64
	No constraint	41.78	—
NAND3_2	36	35.47	1.49
	39	37.79	3.20
	No constraint	42.61	—
NAND4_3	55	54.75	0.46
	60	57.97	3.50
	No constraint	65.83	—
NOR4_1	78	78.31	-0.40
	86	85.59	0.48
	No constraint	93.88	—
NOR4_2	71	71.31	-0.43
	77	75.67	1.76
	No constraint	83.98	—
ON2222_3	71	69.18	2.63
	77	72.70	5.91
	No constraint	81.75	—
average	—	—	1.87

by $(t_{target} - t_{actual})/t_{target} \times 100$, where t_{target} and t_{actual} are the target and the actual delay, respectively. "No constraint" in the column of *Target* means that no timing constraints were set in this case. The runtime to insert the redundant contacts is about 2 second even for the largest example of ON2222_3 which consists of 56 transistors. The average absolute error of the target and actual delay value is less than 2%. Figure 6.28 plots the target and the simulated delay values in the case of NOR4_1 when the input signal to the P type transistor connected to VDD falls from logic level 1 to 0. The simulated delay values show good accordance with the target delay values. Figure 6.29 shows the trade-off curves of target delay and cell area versus number of the additional contacts in the case of NOR4_2. As the timing and area constraint become tight, the coverage of the redundant contact becomes small. These graphs show that the redundant contacts are correctly inserted under the given timing/area constraint.

Table 6.8 Results of the redundant contact insertion by the proposed method.

Circuit	*Target [psec]*	*Area [μm^2]*	*increase [%]*	*#Additional*	*Coverage [%]*
NAND3_1	36	5.12	38.38	3	42.9
	39	6.58	77.84	6	85.7
	No constraint	6.63	79.19	7	100
NAND3_2	36	8.34	27.72	4	40.0
	39	9.42	44.26	7	70.0
	No constraint	11.52	76.42	10	100
NAND4_3	55	24.27	9.57	5	25.0
	60	28.05	26.64	11	55.0
	No constraint	34.62	56.30	20	100
NOR4_1	78	6.33	32.98	4	44.4
	86	7.83	64.50	7	77.8
	No constraint	8.57	80.04	9	100
NOR4_2	71	23.38	34.29	13	46.4
	77	25.68	47.50	19	67.9
	No constraint	30.21	73.52	28	100
ON2222_3	71	35.66	24.03	24	38.7
	77	38.40	33.57	36	58.1
	No constraint	47.58	65.50	62	100
average	—	—	71.83*	—	—

∗ : An average of the no constraint cases

6.6.4 Gate Layout Pattern Regularity Enhancement

This section shows the experimental results of the proposed timing-aware gate layout pattern regularity enhancement method. In this experiment, we also used ILOG *CPLEX 9.1*[27] for an MILP solver, and 25 cell layouts from a standard-cell library of a 90nm technology were used as benchmarks. The regular pitch used in this experiment is defined as two times larger value than the minimum allowable spacing between gates without a contact between them. All the experiments were conducted on a Linux machine with Xeon 3.4GHz processor and 2GB of RAM.

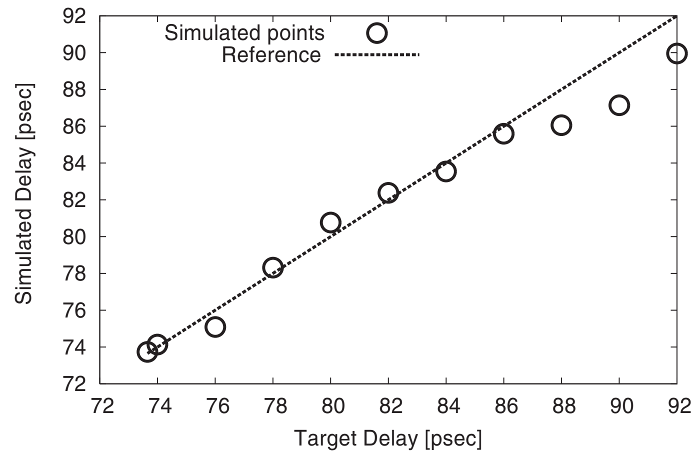
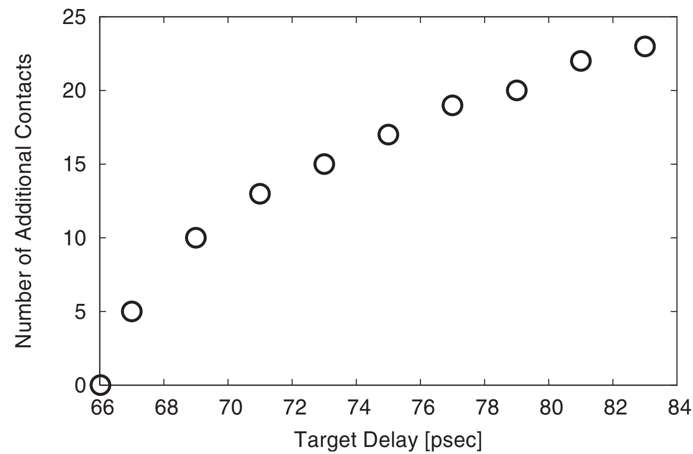
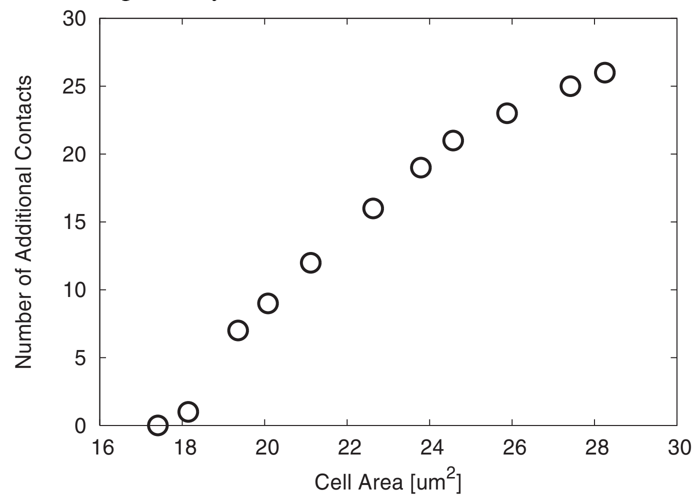


Figure 6.28 Accuracy of the proposed delay model in the case of NOR4_1.



(a) Target delay versus number of additional contacts



(b) Cell area versus number of additional contacts

Figure 6.29 Trade-off curves of target delay versus number of additional contacts in the case of NOR4_2.

Table 6.9 The delay accuracy, area increase, and runtime of the proposed regularity enhancement method with 10% allowable delay increase.

Cell Number	*Transistor Count*	*Delay Error [%]*	*Area Increase [%]*	*Runtime [sec]*
1	4	2.59(max.)	31.0	0.03
2	6	-0.01	26.5	0.03
3	6	0.73(max.)	23.4	0.03
4	6	1.41(max.)	31.3	0.03
5	7	0.45(max.)	22.6	0.04
6	8	0.92(max.)	25.1	0.05
7	8	-1.36	23.8	0.05
8	8	-3.26	44.3	0.02
9	8	-0.66	31.6	0.06
10	8	1.37(max.)	24.2	0.04
11	10	-3.00	38.4	0.05
12	10	2.27(max.)	31.3	0.08
13	10	0.24(max.)	25.6	0.07
14	12	0.02(max.)	31.9	0.05
15	12	-0.86	31.7	0.05
16	12	1.06(max.)	22.6	0.11
17	14	-1.53	46.8	0.08
18	16	-2.03	36.0	0.27
19	20	-0.71	48.3	0.30
20	20	-0.86	46.4	0.27
21	24	-2.81	48.3	0.64
22	28	0.59	36.5	14.62
23	32	-1.51	31.9	5.12
24	32	4.24(max.)	31.1	0.15
25	36	0.73	30.7	81.89
Average	—	1.42*	32.9	—

∗: An average absolute error of non-maximum cases.

Table 6.10 The regularity cost reduction of the proposed regularity enhancement method with 10% allowable delay increase.

Cell Number	*Transistor Count*	*Regularity Cost before RE [A.U.]*	*Regularity Cost after RE [A.U.]*	*Reduction [%]*	*#Perfectly On-Pitch Gates*
1	4	650	100	84.6	3
2	6	810	140	82.7	4
3	6	850	250	70.6	3
4	6	810	100	87.7	5
5	7	830	100	88.0	6
6	8	980	400	59.1	4
7	8	1010	240	76.2	6
8	8	1010	0	100	8
9	8	1000	160	84.0	5
10	8	1050	440	58.1	4
11	10	1310	80	93.9	8
12	10	1340	280	79.1	8
13	10	1530	480	68.6	5
14	12	1520	480	68.4	7
15	12	1560	400	74.4	7
16	12	1320	480	63.6	8
17	14	1920	100	94.8	12
18	16	2160	440	79.6	9
19	20	2520	500	80.2	15
20	20	2510	300	88.0	16
21	24	3210	540	83.2	19
22	28	3580	960	73.2	23
23	32	3960	820	79.3	23
24	32	2020	0	100	32
25	36	4310	1230	71.5	23
Average	—	—	—	79.6	—

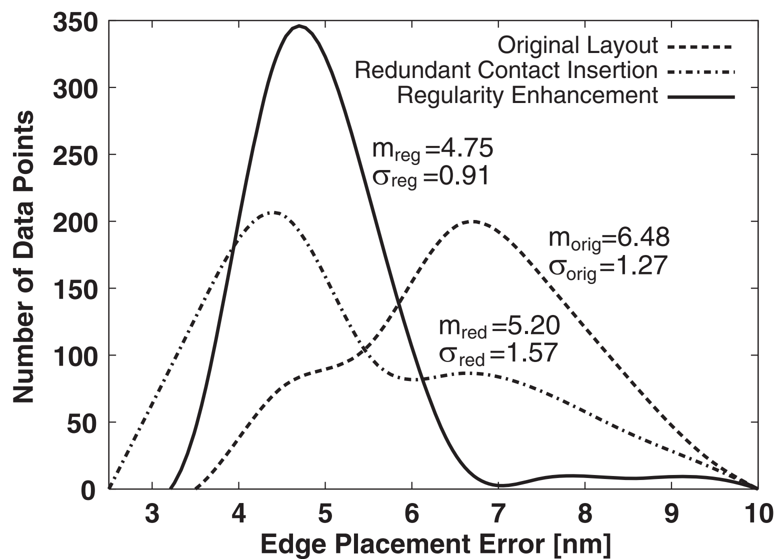


Figure 6.30 The gate CD EPE histogram of all the cells used in this experiment without OPC to highlight the effectiveness.

Tables 6.9 and 6.10 summarize the detailed experimental results of the proposed timing-aware regularity enhancement method when 10% delay increase is allowed during regularity enhancement (RE). These tables show the number of transistors, the delay error between the target and actual delay, the area increase, the runtime for each cell, the regularity cost before and after RE and its reduction ratio, and the number of gates placed perfectly on-pitch after RE. Delay error is calculated by $(t_{target} - t_{actual})/t_{target} \times 100$, where t_{target} and t_{actual} are the target and the actual delay, respectively. (max.) in the delay error column means that the maximum delay during regularity enhancement, which is equivalent to the delay of the regularity-enhanced cell with no timing constraint, is smaller than the target delay. Therefore, an average of absolute delay error is calculated without these cases. The average absolute error is about 1.4% and we can conclude that the delay increases are accurately constrained during regularity enhancement. The average regularity cost reduction of 79.6% is achieved by the proposed method. As a result, 73.7% gates in the regularity-enhanced layouts are placed perfectly on-pitch when 10% delay increase is allowed. Since the regular pitch used in this experiment is a little larger than the usual cases, this result is rather optimistic. However, this result clearly shows that the proposed method effectively enhances the regularity by de-compaction of the cell layout.

Figure 6.30 plots the gate CD Edge Placement Error (EPE) distribution before and after de-compaction for all of the cells used in this experiment with 10% allowable delay increase. This figure shows the EPE distributions of the original and the regularity-enhanced cell lay-

outs generated by the proposed regularity enhancement method. In addition, the EPE distribution of the cell layouts through timing-aware redundant contact insertion explained in the previous section is also plotted for comparison. Note that the redundant contact insertion is executed under the same timing constraints of the previous experiment without considering regularity constraint. Moreover, each generated cell has almost equal cell area to the corresponding regularity-enhanced cells. In this experiment, the print images used for calculating the EPE values are simulated using *Calibre*[37] through mask layout without OPC to highlight the effectiveness of the proposed method. The lithographic condition used in this experiment is summarized as follows.

- Lithography Wavelength: 193nm
- Numerical Aperture: 0.8
- Mask Reduction Factor: 4

As shown in this figure, the average value m of EPE decreases both through redundant contact insertion and regularity enhancement since the average pitch of the gate layouts increases. However, the standard deviation σ of EPE distribution is increased through redundant contact insertion due to the extra non-uniformity. On the other hand, the proposed regularity enhancement method effectively reduces the EPE distribution by introducing the gate layout pattern uniformity. The standard deviation σ of the EPE distribution is reduced by about 28% compared to that of the original layouts. This result shows that the proposed regularity enhancement method can reduce the systematic variation of the gate CD.

The timing accuracy of the proposed timing-aware regularity enhancement in the case of the largest example of 4-input NAND which has 36 transistors is shown in Figure 6.31. This figure plots the target and the simulated delay increase in the case that the input signal to the N type transistor connected to GND rises from logic level 0 to 1. The simulated delay values show good accordance with the target delay values. This result clearly shows the timing accuracy of the proposed timing-aware regularity enhancement method.

Figure 6.32 shows the trade-off curve of target delay versus regularity cost reduction ratio in the case of the same example. The proposed regularity enhancement method can pick up the regularity and performance variants of the original cell layout from this curve and these cells are prepared as a yield-enhanced library which is essential to realize yield-aware VLSI design flows.

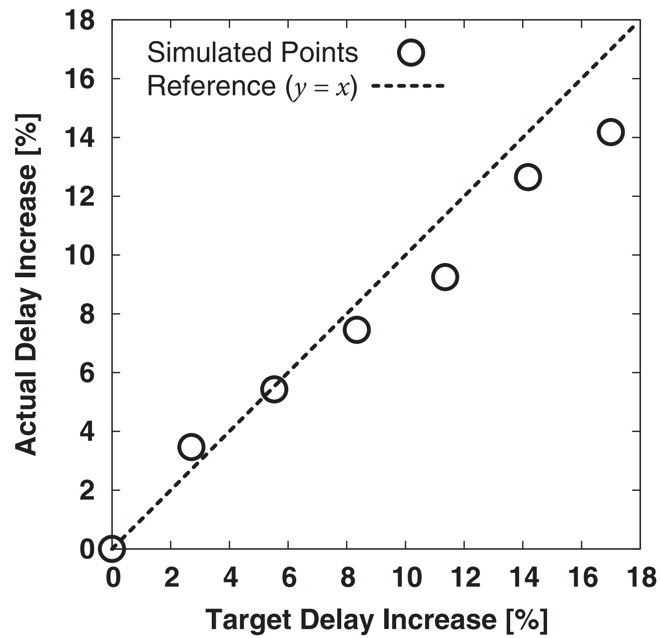


Figure 6.31 Accuracy of the cell delay constraint in the case of the largest example in Table 6.9.

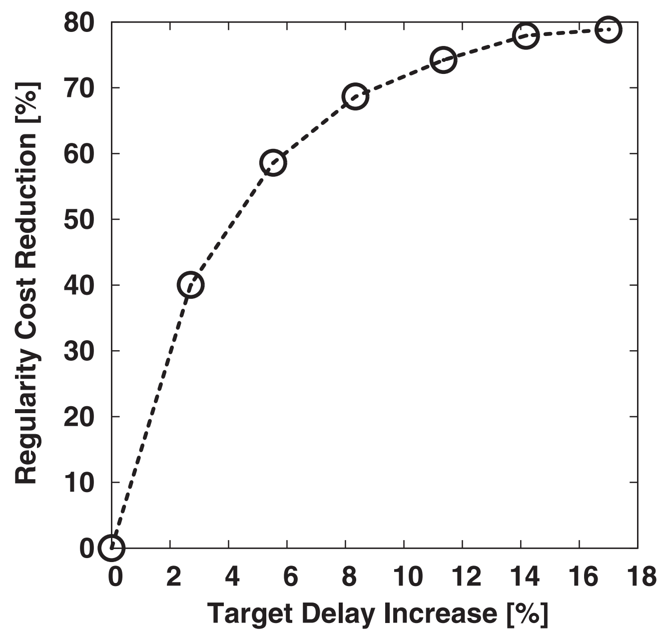


Figure 6.32 Trade-off curve between regularity enhancement and target cell delay in the case of the largest example in Table 6.9.

6.7 Summary

This chapter proposed a yield optimization method for standard cells by cell layout de-compaction under timing constraints. The proposed method performs a de-compaction of the original layout under given timing constraints using LP. We developed a new linear delay model which approximates the difference from the original cell delay and used this model to formulate the timing constraints as LP. Experimental results showed that the developed delay model is accurate enough to constrain the delay during de-compaction. The maximum CA reduction was about 25% on an average of 8 cells.

The proposed method was also shown to be effective for OPC mask data volume reduction. The proposed de-compaction method expands the spacings of the polygons inside the layout and eases the optical proximity effects under given timing constraints using LP. We use the fractured mask data size during mask creation to evaluate the OPC cost. Experimental results on a 90nm cell layouts showed that the proposed method reduces the fractured mask data size 4.28% on an average in the case that 10% delay increase is allowed. The effectiveness of the proposed method in the future technology was also demonstrated.

The redundant contact insertion was realized by the proposed timing-aware de-compaction framework. The proposed method inserts the redundant contacts as many as possible under given timing and area constraints using LP.

This chapter also showed the extension of the de-compaction method to a gate layout pattern regularity enhancement to reduce the systematic variation of the gate CD. With 10% allowable delay increase, 73.7% gates of 25 cells in a 90nm technology are placed perfectly on-pitch by the proposed method. Experiment on the EPE estimation showed that the standard deviation of the gate CD EPE distribution is reduced by about 28% compared to that of the original layouts and showed that the proposed regularity enhancement method is effective for reducing the systematic CD variation.

The proposed timing-aware yield enhancement method enables us to explore the trade-off between yield and performance. We can pick up the yield/performance variants from the trade-off curve and provide a yield-enhanced cell library. The proposed method is the essential technique to realize the yield-aware VLSI design methodologies.

Chapter 7

Conclusions

This thesis focused on the optimization methods for standard-cell layouts. We have proposed minimum-width transistor placement and intra-cell routing via Boolean satisfiability to optimize the area of the cell layouts, and also proposed a comprehensive cell layout synthesis method and a cell layout de-compaction method for yield optimization. The followings are conclusions through this thesis.

Chapter 2: We have proposed a minimum-width cell layout synthesis method for dual CMOS cells via Boolean Satisfiability. Cell layout synthesis problems *i.e.*, the transistor placement and the intra-cell routing problems are first transformed into SAT problems by our formulation. We have presented that the SAT formulation is more suitable for the transistor placement by comparing the runtime of the SAT and the 0-1 ILP formulations of it. We have also presented that the width of the placements generated by the proposed method are smaller than that of the conventional method by using our layout styles. Our method generates the cell layouts of 30 static dual CMOS logic circuits in 58% runtime with only 5% area increase compared with the commercial cell generation tool with cell layout compaction. These results showed that our cell layout styles defined for the SAT formulation is practical enough to generate the layout of dual CMOS cells quickly with a little area overhead.

Chapter 3: We have proposed a hierarchical layout synthesis method for large dual CMOS cells via Boolean Satisfiability. Experimental results showed that the proposed hierarchical transistor placement method generates the same width placement as the exact flat method proposed in Chapter 2 and drastically reduces the runtime. The comparison results of the cell layout synthesis for 30 benchmark circuits showed that the proposed method generates the same width layout as the flat method except one circuit. The comparison results with the commercial cell generation tool without cell layout compaction showed that the total cell width is increased about 4% by the proposed method due to the layout style restriction,

whereas the runtime is only about 3% of that of the commercial tool. From these results, we can conclude that the proposed method can be used as a quick layout generator in the area of transistor-level circuit optimization such as on-demand cell layout synthesis.

Chapter 4: We have proposed flat and hierarchical approaches for generating a minimum-width single-row transistor placement of CMOS cells in presence of non-dual P and N type transistors and generalized the single-row placement method to the multi-row placement method. Our approaches are the first exact minimum-width transistor placement method for non-dual CMOS cells. The experimental results showed that the flat single-row approach generates smaller width placement for 29 out of 103 dual cells than the transistor placement method for dual cells explained in Chapter 2 which theoretically generates the smallest width placement among the existing exact methods for dual cells. This result showed that the proposed method is not only applicable to CMOS cells with any types of structure, but also more effective even for dual CMOS cells compared with the transistor placement method only for dual cells. The hierarchical single-row approach which is based on circuit partitioning reduced the runtime drastically and generated 81% of 340 cells in an industrial standard-cell library of a 90nm technology within one hour for each cell, whereas the flat approach and the exact method only for dual cells generated 43% and 32%, respectively. The experimental results of the multi-row placement method showed that the proposed method generates more area-efficient placement than the conventional method only for dual cells by using the gate connection style which is more suitable for multi-row transistor placement than the conventional style and can solve the cells with up to 26 transistors in reasonable runtime.

Chapter 5: We have proposed an optimal cell layout synthesis technique to minimize the sensitivity to wiring faults. The sensitivity to wiring fault due to spot defects for intra-cell routings was modeled considering the spot defects size distribution and the end effect of critical areas, and used as a cost function. The impact of the sensitivity reduction on the yield improvement was also discussed in this chapter. Our cell layout synthesis technique generates the minimum width layouts of CMOS logic cells comprehensively, and selects the optimal layouts based on the cost functions. We applied our comprehensive layout synthesis method to 8 CMOS logic circuits which have up to 14 transistors and the results showed that the fault sensitivity is reduced about 15% on an average by selecting the minimum-sensitivity layouts rather than selecting the minimum-wire-length layouts. Our layout synthesis method is applicable for deriving the optimal cell layouts by some other cost metrics, such as power, delay, and signal integrity, if reasonable cost functions are given.

Chapter 6: We have proposed a timing-aware cell layout de-compaction method for yield optimization using Linear Programming (LP). The proposed method performs a de-compaction of the original layout in order to improve the yield by minimizing the Critical Area (CA) inside the cell. This yield improvement procedure is executed under given timing constraints. We developed a new linear delay model which approximates the difference from the original cell delay and used this model to formulate the timing constraints as LP. Experimental results showed that the developed delay model is accurate enough to constrain the delay during de-compaction. The CA is correctly minimized under the given timing constraint, and the maximum CA reduction was about 25% on an average of 8 cells. The proposed method was also shown to be effective for OPC mask data volume reduction. The proposed de-compaction method expands the spacings of the polygons inside the layout and relaxes the optical proximity effects under given timing constraints. Experimental results on a 90nm cell layouts showed that the proposed method reduces the fractured mask data size 4.28% on an average in the case that 10% delay increase is allowed. The redundant contact insertion was realized under the proposed timing-aware de-compaction framework. The proposed method inserts the redundant contacts as many as possible under given timing and area constraints using LP. This chapter also showed the extension of the de-compaction method to a gate layout pattern regularity enhancement to reduce the systematic variation of the gate critical dimension (CD). With 10% allowable delay increase, 73.7% gates of 25 cells in a 90nm technology are placed perfectly on-pitch by the proposed method. Experiment on the edge placement error (EPE) estimation showed that the standard deviation of the gate CD EPE distribution is reduced by about 28% compared with that of the original layouts and the proposed regularity enhancement method is effective for reducing the systematic CD variation. The proposed timing-aware yield enhancement method enables us to explore the trade-off between yield and performance. We can pick up the yield/performance variants from the trade-off curve and provide a yield-enhanced cell library. The proposed method is the essential technique to realize the yield-aware VLSI design methodologies.

Now we are sure that these results in this thesis such as the exact minimum-width cell layout synthesis techniques, the comprehensive cell layout synthesis method, and the cell layout de-compaction method for yield optimization will be used for standard-cell layout optimization in terms of area, delay, and yield, and contribute to the VLSI performance and reliability improvement.

Bibliography

[1] Synopsys, Inc., *Design Compiler User Guide*, 2005.

[2] Synopsys, Inc., *Astro User Guide*, 2005.

[3] Cadence Design Systems, Inc., *Quick Reference for Cadence Synthesis*, 2005.

[4] Cadence Design Systems, Inc., *Encounter User Guide*, 2005.

[5] Magma Design Automation, Inc., *Blast Fusion User Guide*, 2003.

[6] R. Bar-Yehuda, J. A. Feldman, R. Y. Pinter, and S. Wimer, "Depth-First-Search and Dynamic Programming Algorithms for Efficient CMOS Cell Generation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 7, pp. 737–743, Jul. 1989.

[7] C. J. Poirier, "Excellerator: Custom CMOS Leaf Cell Layout Generator," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 7, pp. 744–755, Jul. 1989.

[8] Y.-C. Hsieh, "LiB: A Cell Layout Generator," in *Proceedings of ACM/IEEE 27th Design Automation Conference*, 1990, pp. 474–479.

[9] C. Ong, J. Li, and C. Lo, "GENAC: An Automatic Cell Synthesis Tool," in *Proceedings of ACM/IEEE 26th Design Automation Conference*, 1989, pp. 239–243.

[10] M. Guruswamy, R. L. Maziasz, D. Dulitz, S. Raman, V. Chiluvuri, A. Fernandez, and L. G. Jones, "CELLERITY: A Fully Automatic Layout Synthesis System for Standard Cell Libraries," in *Proceedings of ACM/IEEE 34th Design Automation Conference*, 1997, pp. 327–332.

[11] M. Lefebvre, D. Marple, and C. Sechen, "The Future of Custom Cell Generation in Physical Synthesis," in *Proceedings of ACM/IEEE 34th Design Automation Conference*, 1997, pp. 446–451.

[12] Synopsys, Inc., *abraCAD Documentation*, 2003.

[13] Prolific, Inc., *ProGenesis Guide*, 2003.

[14] L. Capodieci, P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang, "Toward a Methodology for Manufacturability-Driven Design Rule Exploration," in *Proceedings of ACM/IEEE 41st Design Automation Conference*, 2004, pp. 311–316.

[15] P. Gupta and F.-L. Heng, "Toward a Systematic-Variation Aware Timing Methodology," in *Proceedings of ACM/IEEE 41st Design Automation Conference*, 2004, pp. 321–326.

[16] J. Mitra, P. Yu, and D. Z. Pan, "RADAR: RET-Aware Detailed Routing Using Fast Lithography Simulations," in *Proceedings of ACM/IEEE 42nd Design Automation Conference*, 2005, pp. 369–372.

[17] A. Nardi and A. L. Sangiovanni-Vincentelli, "Synthesis for Manufacturability: a Sanity Check," in *Proceedings of IEEE/ACM Design, Automation and Test in Europe*, 2004, pp. 796–801.

[18] C. Guardiani, N. Dragone, and P. McNamara, "Proactive Design For Manufacturability (DFM) for Nanometer SoC Designs," in *Proceedings of IEEE Custom Integrated Circuits Conference*, 2004, pp. 309–316.

[19] A. Gupta and J. P. Hayes, "Width Minimization of Two-Dimensional CMOS Cells Using Integer Programming," in *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, 1996, pp. 660–667.

[20] A. Gupta and J. P. Hayes, "CLIP: An Optimizing Layout Generator for Two-Dimensional CMOS Cells," in *Proceedings of ACM/IEEE 34th Design Automation Conference*, 1997, pp. 452–455.

[21] A. Gupta and J. P. Hayes, "Optimal 2-D Cell Layout with Integrated Transistor Folding," in *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, 1998, pp. 128–135.

[22] R. L. Maziasz and J. P. Hayes, "Exact Width and Height Minimization of CMOS Cells," in *Proceedings of ACM/IEEE 28th Design Automation Conference*, 1991, pp. 487–493.

[23] S. Devadas, "Optimal Layout via Boolean Satisfiability," in *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, 1989, pp. 294–297.

[24] T. Uehara and W. M. vanCleemput, "Optimal Layout of CMOS Functional Arrays," *IEEE Transactions on Computers*, vol. C-30, no. 5, pp. 305–312, May 1981.

[25] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik, "Chaff: Engineering an Efficient SAT Solver," in *Proceedings of ACM/IEEE 38th Design Automation Conference*, 2001, pp. 530–535.

[26] P. Barth, "A Davis-Putnam Based Enumeration Algorithm for Linear Pseudo-Boolean Optimization," *Research Report MPI-I-95-2-003, Max-Planck-Institut für Informatik*, Jan. 1989.

[27] ILOG, Inc., *ILOG CPLEX User's Manual*, 2005.

[28] T. Sadakane, H. Nakao, and M. Terai, "A New Hierarchical Algorithm for Transistor Placement in CMOS Macro Cell Design," in *Proceedings of IEEE Custom Integrated Circuits Conference*, 1995, pp. 461–464.

[29] A. Gupta, S. C. The, and J. P. Hayes, "XPRESS: A Cell Layout Generator with Integrated Transistor Folding," in *Proceedings of IEEE European Design and Test Conference*, 1996, pp. 393–400.

[30] F. A. Aloul, A. Ramani, I. L. Markov, and K. A. Sakallah, "Generic ILP versus Specialized 0-1 ILP: An Update," in *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, 2002, pp. 450–457.

[31] H. Zhang and K. Asada, "An Improved Algorithm of Transistor Pairing for Compact Layout of Non-Series-Parallel CMOS Networks," in *Proceedings of IEEE Custom Integrated Circuits Conference*, 1993, pp. 17.2.1–17.2.4.

[32] R. W. Dutton and A. J. Strojwas, "Perspectives on Technology and Technology-Driven CAD," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1544–1560, Dec. 2000.

[33] E. P. Huijbregtz, H. Xue, and J. A. G. Jess, "Routing for Reliable Manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 8, no. 2, pp. 188–194, May 1995.

[34] C. H. Stapper, “Modeling of Defects in Integrated Circuit Photolithographic Patterns,” *IBM Journal of Research and Development*, vol. 28, no. 4, pp. 461–475, Jul. 1984.

[35] F. Schmiedle, R. Drechsler, and B. Becker, “Exact Channel Routing Using Symbolic Representation,” in *Proceedings of IEEE International Symposium on Circuit and Systems*, 1999, pp. 394–397.

[36] K. Sulimma and W. Kunz, “An Exact Algorithm for Solving Difficult Detailed Routing Problems,” in *Proceedings of ACM International Symposium on Physical Design*, 2001, pp. 198–203.

[37] Mentor Graphics, Corp., *Calibre Model-Based OPC User’s Manual*, 2005.

[38] H. T. Heineken, J. Khare, and W. Maly, “Yield Loss Forecasting in the Early Phases of the VLSI Design Process,” in *Proceedings of IEEE Custom Integrated Circuits Conference*, 1996, pp. 27–30.

[39] C. Bamji and E. Malavasi, “Enhanced Network Flow Algorithm for Yield Optimization,” in *Proceedings of ACM/IEEE 33rd Design Automation Conference*, 1996, pp. 746–751.

[40] Y. Bourai and C.-J. R. Shi, “Layout Compaction for Yield Optimization via Critical Area Minimization,” in *Proceedings of IEEE/ACM Design, Automation and Test in Europe*, 2000, pp. 122–125.

[41] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors*, 2005.

[42] P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang, “A Cost-Driven Lithographic Correction Methodology Based on Off-the-Shelf Sizing Tools,” in *Proceedings of ACM/IEEE 40th Design Automation Conference*, 2003, pp. 16–21.

[43] Q. D. Qian and S. X.-D. Tan, “Advanced Physical Models for Mask Data Verification and Impacts on Physical Layout Synthesis,” in *Proceedings of IEEE International Symposium on Quality Electronic Design*, 2003, pp. 125–130.

[44] L. D. Huang and M. D. F. Wong, “Optical Proximity Correction (OPC)-Friendly Maze Routing,” in *Proceedings of ACM/IEEE 41st Design Automation Conference*, 2004, pp. 186–191.

[45] Y. C. Ban, S. H. Choi, K. H. Lee, D. H. Kim, J.-S. Hong, Y.-H. Kim, M.-H. Yoo, and J.-T. Kong, “A Fast Lithography Verification Framework for Litho-Friendly Layout Design,” in *Proceedings of IEEE International Symposium on Quality Electronic Design*, 2005, pp. 169–174.

[46] P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang, “Performance-Driven OPC for Mask Cost Reduction,” in *Proceedings of IEEE International Symposium on Quality Electronic Design*, 2005, pp. 270–257.

[47] K.-Y. Lee and T.-C. Wang, “Post-Routing Redundant Via Insertion for Yield/Reliability Improvement,” in *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference*, 2006, pp. 303–308.

[48] J. Luo, S. Sinha, Q. Su, J. Kawa, and C. Chiang, “An IC Manufacturing Yield Model Considering Intra-Die Variations,” in *Proceedings of ACM/IEEE 43rd Design Automation Conference*, 2006, pp. 749–754.

[49] H.-Y. Chen, M.-F. Chiang, Y.-W. Chang, L. Chen, and B. Han, “Novel Full-Chip Gridless Routing Considering Double-Via Insertion,” in *Proceedings of ACM/IEEE 43rd Design Automation Conference*, 2006, pp. 755–760.

[50] A. B. Kahng, L. K. Scheffer, M. Orshansky, and A. Strojwas, “DFM Tools and Methodologies for 65nm and Below,” in *Tutorial of IEEE/ACM Asia and South Pacific Design Automation Conference*, 2006.

[51] M. Orshansky, L. Milor, P. Chen, K. Keutzer, and C. Hu, “Impact of Spatial Intrachip Gate Length Variability on the Performance of High-Speed Digital Circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 5, pp. 544–553, May 2002.

[52] K. Cao, S. Dobre, and J. Hu, “Standard Cell Characterization Considering Lithography Induced Variations,” in *Proceedings of ACM/IEEE 43rd Design Automation Conference*, 2006, pp. 801–804.

[53] B. E. Stine, D. S. Boning, and J. E. Chung, “Analysis and Decomposition of Spatial Variation in Integrated Circuit Process and Devices,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 10, no. 1, pp. 24–41, Feb. 1997.

[54] L.-T. Pang and B. Nikolic, "Impact of Layout on 90nm CMOS Process Parameter Fluctuations," in *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, 2006, pp. 69–70.

[55] L. Pileggi, H. Schmit, A. J. Strojwas, P. Gopalakrishnan, V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner, and K. Y. Tong, "Exploring Regular Fabrics to Optimize the Performance-Cost Trade-Off," in *Proceedings of ACM/IEEE 40th Design Automation Conference*, 2003, pp. 782–787.

[56] V. Kheterpal, V. Rovner, T. G. Hersan, D. Motiani, Y. Takegawa, A. J. Strojwas, and L. Pileggi, "Design Methodology for IC Manufacturability Based on Regular Logic Bricks," in *Proceedings of ACM/IEEE 42nd Design Automation Conference*, 2005, pp. 353–358.

[57] L. W. Liebmann, "Layout Impact of Resolution Enhancement Techniques: Impediment or Opportunity?" in *Proceedings of ACM International Symposium on Physical Design*, 2003, pp. 110–117.

[58] X. Yuan, K. W. McCullen, F.-L. Heng, R. F. Walker, J. Hibbeler, R. J. Allen, and R. R. Narayan, "Technology Migration Technique for Designs with Strong RET-driven Layout Restrictions," in *Proceedings of ACM International Symposium on Physical Design*, 2005, pp. 175–182.

[59] Mentor Graphics, Corp., *Calibre xL User's Manual*, 2005.

[60] Synopsys, Inc., *HSPICE Simulation and Analysis User Guide*, 2005.

[61] Mentor Graphics, Corp., *Calibre Mask Data Preparation User's Manual*, 2005.

List of Publications

Journal Papers

1. T. Iizuka, M. Ikeda, and K. Asada, "High Speed Layout Synthesis for Minimum-Width CMOS Logic Cells via Boolean Satisfiability," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E87-A, no. 12, pp. 3293–3300, Dec. 2004.

2. T. Iizuka, M. Ikeda, and K. Asada, "Yield-Optimal Layout Synthesis of CMOS Logic Cells by Wiring Fault Minimization," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, no. 7, pp. 1957–1963, Jul. 2005.

3. T. Iizuka, M. Ikeda, and K. Asada, "Exact Minimum-Width Transistor Placement for Dual and Non-Dual CMOS Cells," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, no. 12, pp. 3485–3491, Dec. 2005.

4. T. Iizuka, M. Ikeda, and K. Asada, "Timing-Aware Cell Layout De-Compaction for Yield Optimization by Critical Area Minimization," *IEEE Transactions on Very Large Scale Integration Systems*. (*submitted*)

International Conference Papers

1. T. Iizuka and K. Asada, "An Exact Algorithm for Practical Routing Problems," in *Proceedings of the Third IEEE Asia-Pacific Conference on ASICs*, pp. 343–346, Aug. 2002.

2. T. Iizuka, M. Ikeda, and K. Asada, "High Speed Layout Synthesis for Minimum-Width CMOS Logic Cells via Boolean Satisfiability," in *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference*, pp. 149–154, Jan. 2004.

3. T. Iizuka, M. Ikeda, and K. Asada, "Exact Wiring Fault Minimization via Comprehensive Layout Synthesis for CMOS Logic Cells," in *Proceedings of IEEE International Symposium on Quality Electronic Design*, pp. 377–380, Mar. 2004.

4. T. Iizuka, M. Ikeda, and K. Asada, "Minimum-Width Transistor Placement Without Dual Constraint for CMOS Cells," in *Proceedings of ACM Great Lakes Symposium on VLSI*, pp. 74–77, Apr. 2005.

5. T. Iizuka, M. Ikeda, and K. Asada, "Timing-Driven Cell Layout De-Compaction for Yield Optimization by Critical Area Minimization," in *Proceedings of IEEE/ACM Design, Automation and Test in Europe*, pp. 884–889, Mar. 2006.

6. T. Iizuka, M. Ikeda, and K. Asada, "Exact Minimum-Width Multi-Row Transistor Placement for Dual and Non-Dual CMOS Cells," in *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 5431–5434, May 2006.

7. T. Iizuka, M. Ikeda, and K. Asada, "Timing-Driven Redundant Contact Insertion for Standard Cell Yield Enhancement," in *Proceedings of IEEE International Conference on Electronics, Circuits and Systems*, pp. 704–707, Dec. 2006.

8. T. Iizuka, M. Ikeda, and K. Asada, "OPC-Friendly De-Compaction with Timing Constraints for Standard Cell Layouts," *IEEE International Symposium on Quality Electronic Design*, Mar. 2007. (*to appear*)

9. T. Iizuka, M. Ikeda, and K. Asada, "Timing-Aware Cell Layout Regularity Enhancement for Reduction of Systematic CD Variation," *ACM/IEEE 44th Design Automation Conference*, Jun. 2007. (*submitted*)

Domestic Conference Papers

1. T. Iizuka, M. Ikeda, and K. Asada, "An Exact Algorithm for Practical Routing Problems," in *Proceedings of IEICE Society Conference*, A-3-6, p. 61, Sep. 2002. (*in Japanese*)

2. T. Iizuka, M. Ikeda, and K. Asada, "Cell Layout Synthesis via Boolean Satisfiability," in *Proceedings of IPSJ DA Symposium*, pp. 139–144, Jul. 2003. (*in Japanese*)

3. T. Iizuka, M. Ikeda, and K. Asada, "Exact Wiring Fault Minimization via Comprehensive Layout Synthesis for CMOS Logic Cells," *IEICE Technical Report*, vol. 103, no. 476, pp. 157–161, Nov. 2003. (*in Japanese*)

4. T. Iizuka, M. Ikeda, and K. Asada, "Minimum-Width Transistor Placement Method via Boolean Constraints for Non-Complementary Transistors," in *Proceedings of IPSJ DA Symposium*, pp. 121–126, Jul. 2004. (*in Japanese*)

5. T. Iizuka, H. Yoshida, M. Ikeda, and K. Asada, "Hierarchical Layout Synthesis for CMOS Logic Cells via Boolean Satisfiability," *IEICE Technical Report*, vol. 104, no. 478, pp. 1–6, Dec. 2004. (*in Japanese*)

6. T. Iizuka, M. Ikeda, and K. Asada, "Computational Cost Reduction for Minimum-Width Transistor Placement of Arbitrary Circuit Structures," in *Proceedings of IPSJ DA Symposium*, pp. 121–126, Aug. 2005. (*in Japanese*)

7. T. Iizuka, M. Ikeda, and K. Asada, "Timing-Driven Cell Layout De-Compaction for Yield Optimization by Critical Area Minimization," *IEICE Technical Report*, vol. 105, no. 442, pp. 79–84, Dec. 2005. (*in Japanese*)

8. T. Iizuka, M. Ikeda, and K. Asada, "Exact Minimum-Width Multi-Row Transistor Placement for Dual and Non-Dual CMOS Cells," in *Proceedings of IEICE Society Conference*, A-3-20, p. 64, Sep. 2006. (*in Japanese*)

Awards

1. *IEICE Young Researcher's Award*
   T. Iizuka, M. Ikeda, and K. Asada, "An Exact Algorithm for Practical Routing Problems," in *Proceedings of IEICE Society Conference*, A-3-6, p. 61, Sep. 2002. (*in Japanese*)

2. *Inose Science Award from Association for Promotion of Electrical, Electronic and Information Engineering*, Jul. 2003.

3. *Outstanding Paper Award from IPSJ SIGSLDM*
   T. Iizuka, M. Ikeda, and K. Asada, "Minimum-Width Transistor Placement Method via Boolean Constraints for Non-Complementary Transistors," in *Proceedings of IPSJ DA Symposium*, pp. 121–126, Jul. 2004. (*in Japanese*)

4. *Nikkei-BP LSI IP Design Award (Development Promotion)*
   T. Iizuka, M. Ikeda, and K. Asada, "Yield-Optimized CMOS Logic Cell Layout IP Synthesis System", May 2005. (*in Japanese*)

5. *IPSJ Yamashita SIG Research Award*
   T. Iizuka, M. Ikeda, and K. Asada, "Computational Cost Reduction for Minimum-Width Transistor Placement of Arbitrary Circuit Structures," in *Proceedings of IPSJ DA Symposium*, pp. 121–126, Aug. 2005. (*in Japanese*)

6. *Nikkei-BP LSI IP Design Award (IP Award)*
   T. Iizuka, M. Ikeda, and K. Asada, "Yield-Optimized Standard Cell Layout IP Synthesis System", May 2006. (*in Japanese*)

7. *Best Student Paper Award from IEEE International Conference on Electronics, Circuits and Systems*
   T. Iizuka, M. Ikeda, and K. Asada, "Timing-Driven Redundant Contact Insertion for Standard Cell Yield Enhancement," in *Proceedings of IEEE International Conference on Electronics, Circuits and Systems*, pp. 704–707, Dec. 2006.