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

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 decompaction 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

List of Figures

List of Tables

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

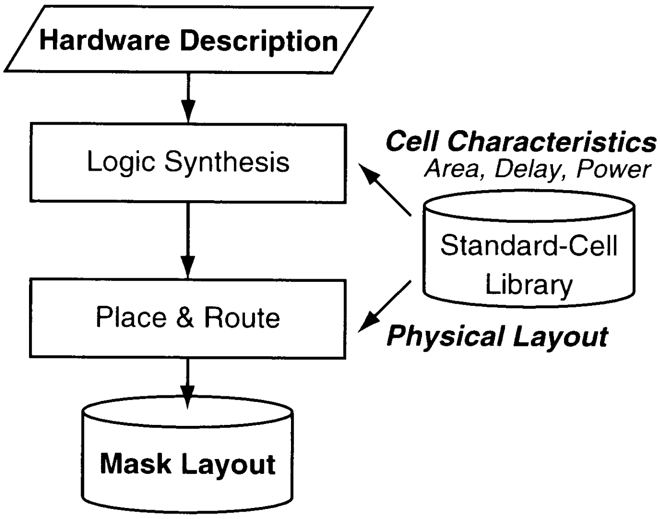

Figure 1.1 A simplified flow diagram of the cell-based LSI design.

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

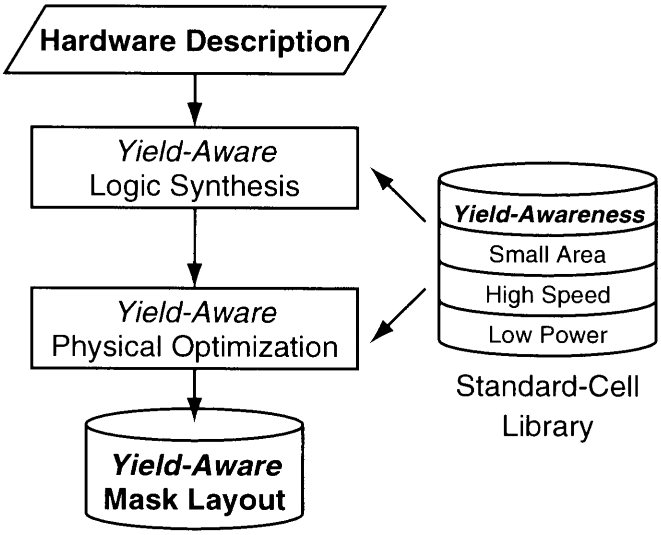

Figure 1.2 A flow diagram of the yield-aware cell-based LSI design.

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.

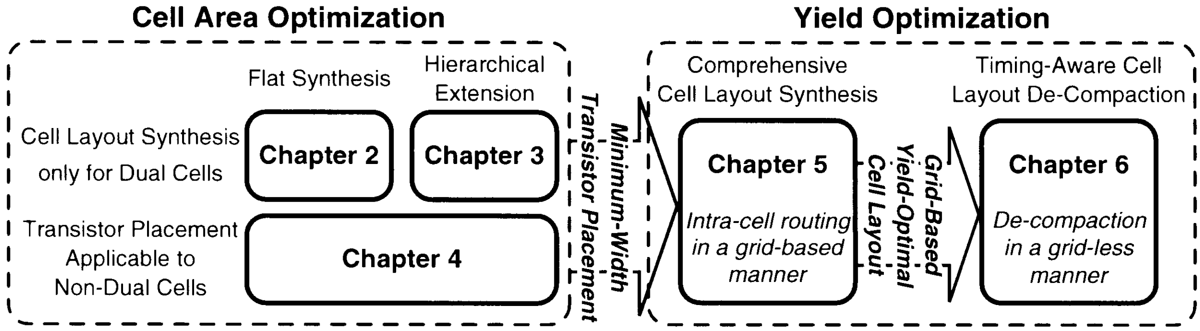
Cell Area Optimization
Yield Optimization
Flat Synthesis
Hierarchical Extension
Cell Layout Synthesis only for Dual Cells
Chapter 2
Chapter 3
Transistor Placement Applicable to Non-Dual Cells
Chapter 4
Minimum-Width Transistor Placement
Comprehensive Cell Layout Synthesis
Chapter 5
Intra-cell routing in a grid-based manner
Grid-Based Yield-Optimal Cell Layout
Timing-Aware Cell Layout De-Compaction
Chapter 6
De-compaction in a grid-less manner

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 conventional 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 problems 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.

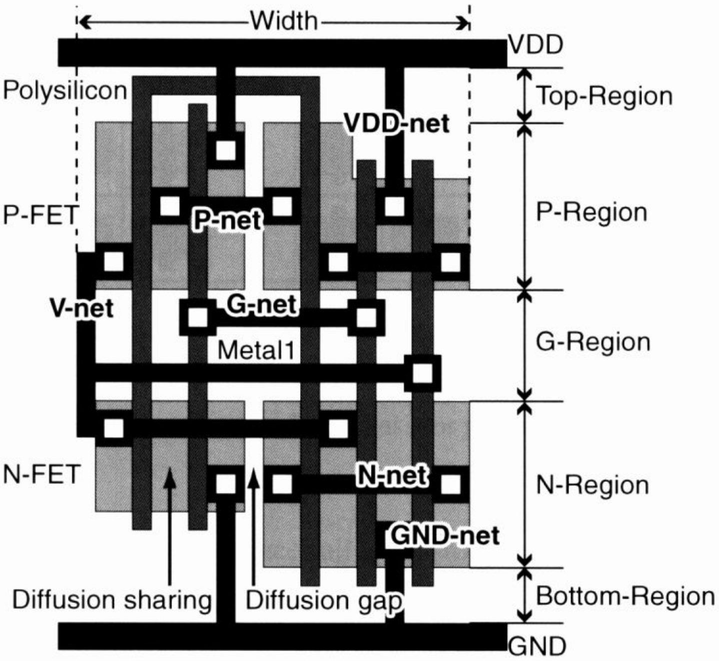

Figure 2.1 An illustration of the layout styles of the proposed SAT-based cell layout synthesis for dual CMOS cells.

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 2*N* transistors in the minimum area. This problem can

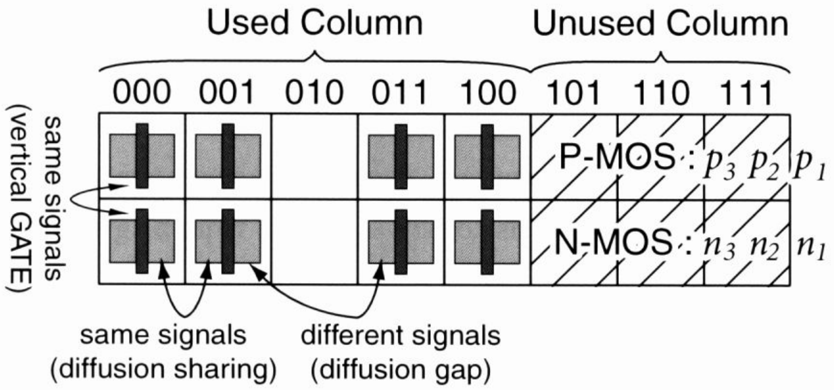
Used Column
Unused Column
000 001 010 011 100 101 110 111
P-MOS : p_3 p_2 p_1
N-MOS : n_3 n_2 n_1
same signals (vertical GATE)
same signals (diffusion sharing)
different signals (diffusion gap)

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.

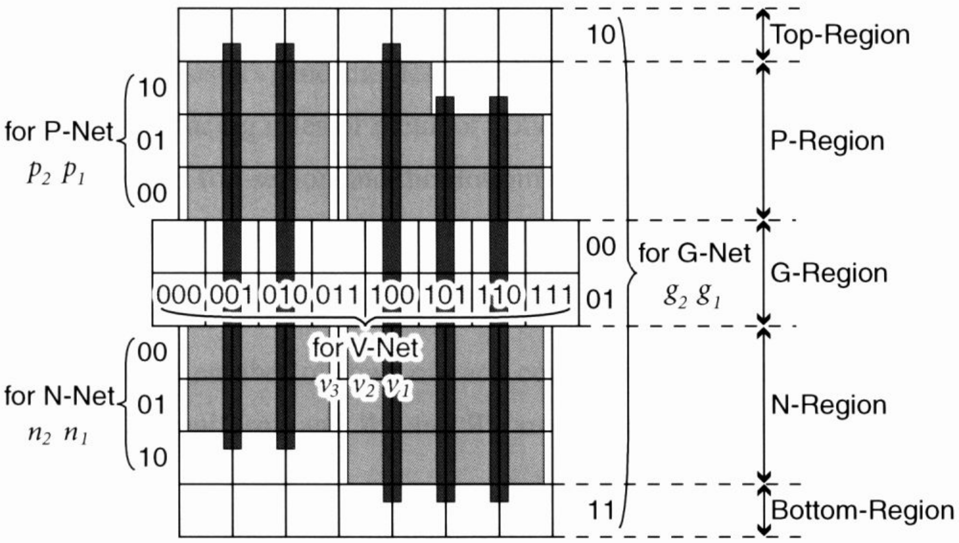

Figure 2.3 A SAT formulation of the intra-cell routing.

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

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

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

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

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.

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

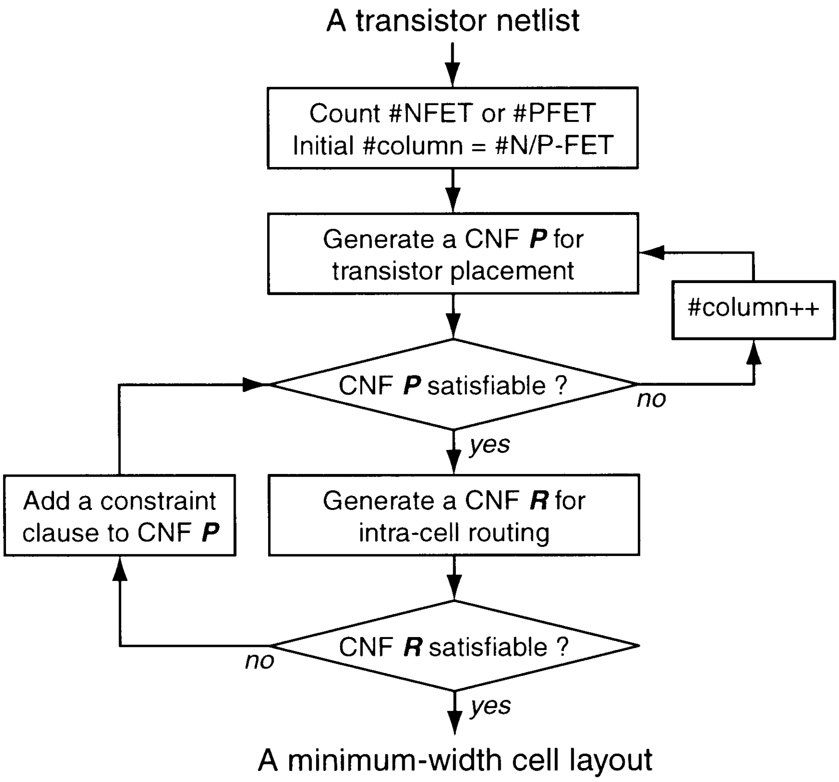
A transistor netlist
Count #NFET or #PFET
Initial #column = #N/P-FET
Generate a CNF P for transistor placement
#column++
CNF P satisfiable ?
no
yes
Add a constraint clause to CNF P
Generate a CNF R for intra-cell routing
CNF R satisfiable ?
no
yes
A minimum-width cell layout

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*		*Runtime (sec.)*	
		Problem Size		*ProGenesis*	*Proposed*
name	*#tr.*	*#var.*	*#cla.*		
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)

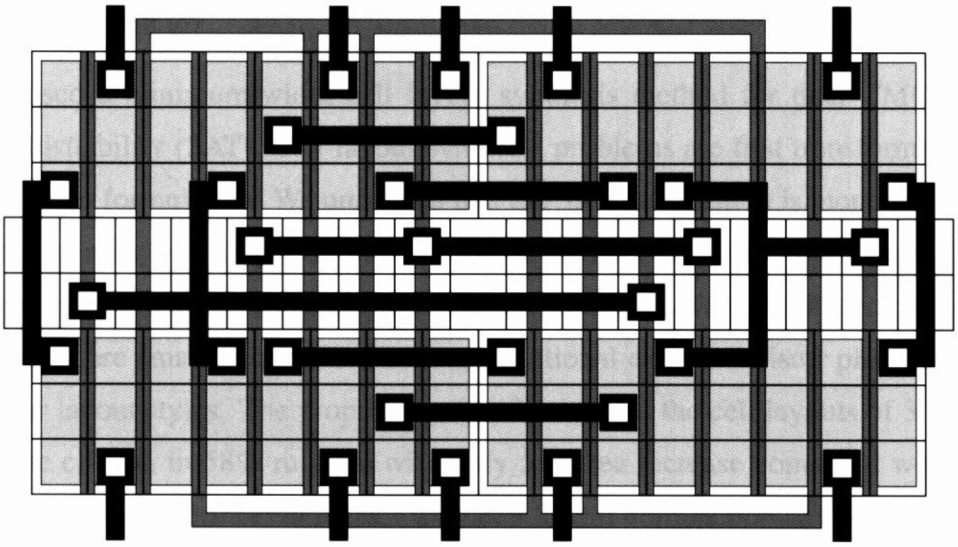

Figure 2.5 The example result of fad1 cell layout generated by the proposed SAT-based cell layout synthesis method for dual CMOS cells.

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.