DESIGN FOR MANUFACTURING (DFM) IN SUBMICRON VLSI DESIGN

A Dissertation

by

KE CAO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2007

Major Subject: Computer Engineering

DESIGN FOR MANUFACTURING (DFM) IN SUBMICRON VLSI DESIGN

A Dissertation

by

KE CAO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Jiang Hu
Committee Members,	Weiping Shi
	Duncan Walker
	Vivek Sarin
Head of Department,	Costas N. Georghiades

August 2007

Major Subject: Computer Engineering

ABSTRACT

Design for Manufacturing (DFM) in Submicron VLSI Design. (August 2007)
Ke Cao, B.S., University of Science and Technology of China;
M.S., University of Minnesota
Chair of Advisory Committee: Dr. Jiang Hu

As VLSI technology scales to 65nm and below, traditional communication between design and manufacturing becomes more and more inadequate. Gone are the days when designers simply pass the design GDSII file to the foundry and expect very good manufacturing and parametric yield. This is largely due to the enormous challenges in the manufacturing stage as the feature size continues to shrink. Thus, the idea of DFM (Design for Manufacturing) is getting very popular. Even though there is no universally accepted definition of DFM, in my opinion, one of the major parts of DFM is to bring manufacturing information into the design stage in a way that is understood by designers. Consequently, designers can act on the information to improve both manufacturing and parametric yield. In this dissertation, I will present several attempts to reduce the gap between design and manufacturing communities: Alt-PSM aware standard cell designs, printability improvement for detailed routing and the ASIC design flow with litho aware static timing analysis. Experiment results show that we can greatly improve the manufacturability of the designs and we can reduce design pessimism significantly for easier design closure.

To my parents and my beloved wife, Shasha Luo

ACKNOWLEDGMENTS

Pursing a doctoral degree certainly is a long and difficult process, and it would not be possible without the help of numerous people. To them I wish to express my sincere gratitude. I would especially like to thank my advisor, Dr. Jiang Hu, for his generous time and commitment. Throughout my doctoral work he encouraged me to develop independent thinking and research skills. He continually stimulated my analytical thinking and greatly assisted me with scientific writing.

I am also very grateful for having an exceptional doctoral committee and wish to thank Dr. Weiping Shi, Dr. Duncan Walker and Dr. Vivek Sarin for their continual support and encouragement.

My thanks go out to the many fellow students in the Computer Engineering program, Chin-Ngai Sze (Cliff), Ganesh Venkataraman, Di Wu, Mankang Mai, Shu Yan, Zhuo Li, Xiang Lu and Ying Zhou. They made my routine student life much more enjoyable.

My research for this dissertation was made more practical and more extensive through the communications with some of the brightest people in the ASIC design and EDA industry. Thus I would like to thank Mr. Sorin Dobre at Qualcomm for the many hour-long discussions on the trends of various design methodologies including DFM, Dr. Puneet Gupta at Blaze-DFM for debating lithography aware design methodology, Mr. Bill Graupp at Mentor Graphics for developing design tool usability, Dr. Andrew Kahng at UCSD/Blaze-DFM for his encouragement and support.

Finally, I'd like to thank my family. My parents extended their unconditioned support to my decision to continue with the graduate program. I'm especially grateful to my wife, Shasha Luo, for her patience, her sacrifice, and for helping me keep my life in proper perspective and balance. I can not imagine going through graduate school without her being by my side.

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1
- A. Alt-PSM Compliance and Composability for Standard Cell Layout 1
- B. Wire Sizing and Spacing for Lithographic Printability and Timing Optimization 3
- C. ASIC Design Flow Considering Lithography Induced Effects . . 6
- D. Organization of the Dissertation 8

II ALT-PSM COMPLIANCE AND COMPOSABILITY FOR STANDARD CELL LAYOUT 9
- A. Introduction 9
- B. Library Cell Construction 11
  - 1. Layout Style 11
  - 2. Transistor Placement 12
  - 3. Routing 15
  - 4. SAT Formulation Flow 21
- C. Cell Alt-PSM Composability 21
  - 1. Problem Formulation 21
  - 2. Algorithm 24
- D. Standard Cell Library Design Flow 26
- E. Experiments 28
  - 1. Standard Cell Construction 28
  - 2. Layout Modification 29
- F. Conclusion 30

III WIRE SIZING AND SPACING FOR LITHOGRAPHIC PRINTABILITY AND TIMING OPTIMIZATION 32
- A. Introduction 32
- B. Problem Formulation 35
- C. Printability Model 37
  - 1. Aerial Image 37
  - 2. Printability Function 41
- D. Timing Driven Wire Sizing Considering Coupling Capacitance . 42

CHAPTER Page

- E. Printability Optimization 48
- F. Experimental Results 51
- G. Conclusion 53

IV ASIC DESIGN FLOW CONSIDERING LITHOGRAPHY INDUCED EFFECTS 55

- A. Introduction 55
- B. Overview of Methodology 57
  - 1. FEOL Lithography Variations 57
    - a. Standard Cell Architecture 57
    - b. Calculation of Effective Transistor with Litho Effects 59
    - c. Netlist Back Annotation and Standard Cell Characterization 63
  - 2. BEOL Lithography Variations 64
    - a. Lithography Simulation On BEOL 65
    - b. STA Comparison 65
  - 3. Litho-aware STA Flow 65
- C. Experiment 66
  - 1. Standard Cells 66
  - 2. Design Implementation 69
  - 3. BEOL Comparison 70
- D. Conclusion 71

V CONCLUSION 74

REFERENCES 75

VITA 79

LIST OF TABLES

TABLE Page

I Experimental results for standard cell construction. 28

II Experimental results for standard cell modification. 31

III Benchmark circuit specification. 52

IV Experimental results. # vio is the number of delay violations. Θ is the printability. # vio remains 0 in phase 2. 53

V Lithographic simulation results on average EPE. 54

VI Experimental results for standard cell characterization. 72

VII Leakage variability analysis. 73

LIST OF FIGURES

FIGURE Page

1 Technology tread for transistor feature size. 2

2 (a) Alternating Phase Shifting Mask(Alt-PSM). (b) Phase conflict occurs for a T-shaped critical feature. (c) Phase conflict removal by splitting a phase region. 3

3 Layout without OPC and its silicon image [14]. 4

4 Layout with OPC and its silicon image [14]. 4

5 OPC results in fractures in mask layout [15]. 5

6 Standard cell Alt-PSM composability. 10

7 Library cell layout style. 12

8 Layout constraints examples. 19

9 PSM aware library cell synthesis algorithm. 22

10 Phase conflict graph. 23

11 Boundary nodes in phase conflict graph. 23

12 Phase conflict bipartite graph: empty dots represent boundary nodes, solid dots represent all other nodes. 25

13 Standard cell library design flow. 27

14 (a) XOR gate layout: T joints in Metal 1 layer that pose phase conflict problem. (b) PSM clean layout. 29

15 (a) 2 Input NOR gate layout with N-P pairs split in 3 fingers each: T joints in Poly. (b) PSM compliant and composable. layout 30

16 Wire segment parameters. 35

FIGURE Page

17 Infinite line model in dashed bounding box. Semi-infinite line model in dotted boxes. 39

18 Contour plot of complex field amplitude vs. feature linewidth and distance to feature for coherent illumination. 40

19 Contour plot of light intensity vs. feature linewidth and distance to feature for partially coherent illumination. 41

20 Comparisons of light intensity calculations. 41

21 Light intensity and EPE. 43

22 Illustration on relative positions among nodes and edges. 46

23 Timing driven wire sizing algorithm. 48

24 Printability optimization heuristic. 50

25 Test structures for CD variations through pitch. 58

26 Critical dimensions of the two test structures when *L* changes. 58

27 Dummy poly of two neighboring cells can be merged. 60

28 Systematic poly gate CD variations from lithography. 60

29 Calculation of effective gate length for timing and leakage. 61

30 Litho contours for routing metal. 64

31 Convert litho contour to Manhattan polygon. 66

32 Litho aware standard cell characterization flow. 67

33 Lithography induced Lgate deviation distribution: worst corner. 68

34 Lithography induced Lgate deviation distribution: best corner. 68

35 Timing variability reduction for critical paths. 70

36 Delay path timing changes. 73

CHAPTER I

INTRODUCTION

For an ASIC design, it used to be true that design and manufacturing are relatively independent of one another, with a set of design rules the only connection between the two. This configuration, however, is changing as VLSI technology scales towards deep submircon technology nodes. Manufacturing in sub-wavelength technology has become so challenging that we need to address some of the issues in design practice to improve manufacturing yield. At the same time, process variations in the advanced technologies cause either difficulty or unnecessary pessimism for design closure. Clearly, a mechanism is needed for modeling manufacturing processes accurately and bring the modeling into design space. Thus designer can effectively account for manufacturing realities to enhance manufacturing yield and at the same time reduce design pessimism. In this dissertation, I will attempt to present some of the new technologies to reduce the gap between design and manufacturing.

A. Alt-PSM Compliance and Composability for Standard Cell Layout

Current VLSI technology has the minimum transistor feature size of 90*nm* and 65*nm*, which is remarkably below the lithography wavelength of 192*nm*. The technology trend for future indicates that this gap will become even larger as shown in Figure 1. In the subwavelength lithography, Resolution Enhancement Techniques(RET) [1] have been deployed to combat strong diffractive effects that cause severe mismatch between mask shapes and printed shapes. Phase Shifting Mask(PSM) is one of the common techniques in RET. PSM uses the destructive interference between two 180 degree out of phase lights to print shape edges.

The journal model is *IEEE Transactions on Automatic Control.*

There are two types of PSM currently in use, Attenuated PSM(Att-PSM) and Alternating PSM(Alt-PSM). In Att-PSM, mask substrates are used to allow a small amount of the out of phase light to penetrate the normally opaque mask regions. In Alt-PSM, lights with opposite phases are shed on two sides of a thin critical feature. While Att-PSM poses less restriction in layout design than Alt-PSM does, Alt-PSM is easier to control in lithography process and the quality and the robustness of the printed image is better. Alt-PSM is currently being used in high performance VLSI designs in 90nm and 65nm technologies for its better critical dimension(CD) control of transistor gate poly. For routing metals of the standard cell, Att-PSM could be used instead of Alt-PSM as the requirement of CD control is not as rigorous. Alt-PSM may be the only option for Phase Shifting Mask of gate poly when VLSI technology further scales down 45nm and 32nm. The Alt-PSM technique is illustrated in Figure 2(a).

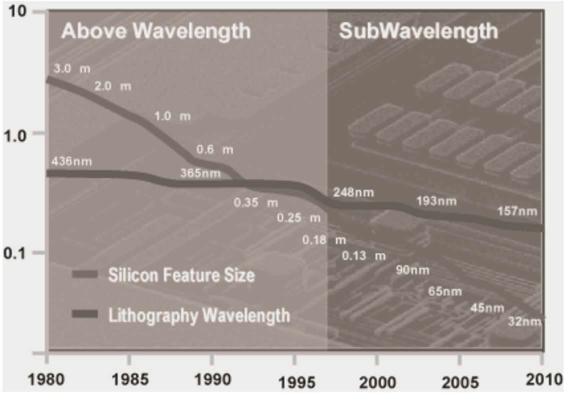


Fig. 1. Technology tread for transistor feature size.

Even though the Alt-PSM technique is carried out in the stage of mask design and lithography, it needs to be considered in circuit layout as well. For example, a T-shaped layout like in Figure 2(b) may make phase assignment infeasible. In [2], a region where a phase conflict occurs is split to enable Alt-PSM as shown in Figure 2(c). As the phases

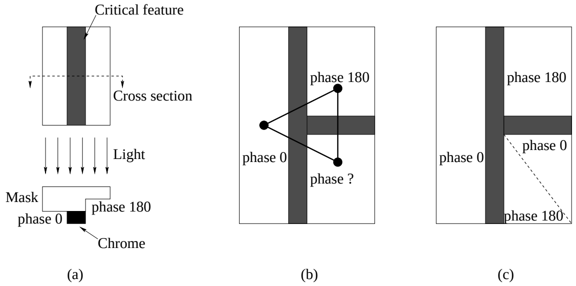
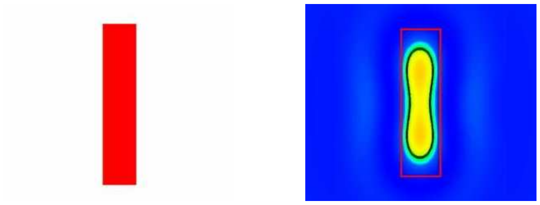


Fig. 2. (a) Alternating Phase Shifting Mask(Alt-PSM). (b) Phase conflict occurs for a T-shaped critical feature. (c) Phase conflict removal by splitting a phase region.

of lights are opposite along two sides of the splitting line (the dashed line in Figure 2(c)), unwanted features may be left there and need to be trimmed by another exposure. The second exposure will increase the already expensive mask cost and cause misalignment risk. In [3, 4], graph based algorithms are proposed to modify existing layout for Alt-PSM compliance. These algorithms can achieve global optimality, but demand large CPU time if they are applied on an entire chip layout. Furthermore, it is very complicated to modify layout at top level in standard cell based designs, considering both fixed IP designs and interactions between layout layers.

B. Wire Sizing and Spacing for Lithographic Printability and Timing Optimization

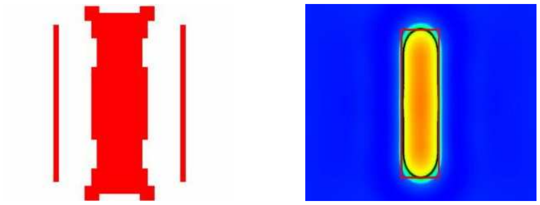
When lithography entered sub-wavelength regime, strong diffraction effect may cause significant discrepancy between photo-mask patterns and printed features. For example, a rectangle feature as in Figure 3(a) on photo-mask may result in printed feature with distortion as in Figure 3(b) on the silicon. A circuit layout with poor printability implies that it is difficult to make the printed features on wafers follow designed shapes without distortions. Currently, the printability of devices with sub-wavelength sizes is usually improved by using Resolution Enhancement Techniques (RET) [1, 11–13] such as Optical Proximity Correction (OPC), Phase Shift Mask (PSM), Off Axis Illumination (OAI), and Sub-Resolution Assist Feature (SRAF), so as to overcome diffraction limit and process imperfections. For example, mask layout for printing the rectangle feature in Figure 3(a) becomes Figure 4(a) with OPC and SRAF. As a result, the printed feature on silicon (Figure 4(b)) becomes closer to the desired rectangle shape.



(a) Traditional layout

(b) Silicon image

Fig. 3. Layout without OPC and its silicon image [14].



(a) Layout with OPC

(b) Silicon image

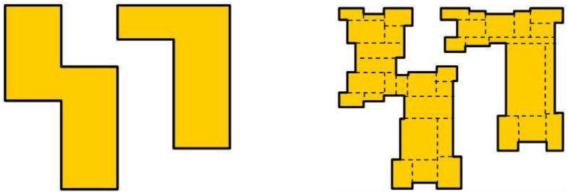
Fig. 4. Layout with OPC and its silicon image [14].

However, relying on RET alone is inadequate to harness the problem because of the following reasons.

- The increasingly large gap between feature size and wavelength forces several aggressive RETs to be jointly applied and thereby compounds the already complicated mask design and increases mask cost drastically. An example in [15] is shown in

Figure 5 to demonstrate this problem. Compared to mask layout without OPC like Figure 5(a), the mask layout with OPC like Figure 5(b) increases mask data volume, mask writing time and therefore mask cost dramatically.

- The circuit complexity keeps growing and makes RET formidably challenging.
- RET deployments may be obstructed or even prohibited by an RET-unfriendly layout. Therefore, the sub-wavelength printability problem has to be considered also in circuit layout design to ensure manufacturability.



(a) Without OPC (b) With OPC

Fig. 5. OPC results in fractures in mask layout [15].

In practice, a lithography friendly and/or RET compliant layout is often obtained through either rule based or model based methodology level approaches. In rule based approaches, lithography friendliness and/or RET compliance are expressed as a set of recommended/hard design rules which are applied in detailed layout design such as detailed routing and layout compaction. This approach is fast and relatively easy to use. However, lithography and RET procedures are so complicated that it is very difficult to convey their requests fully through simple rules. In model based approaches, lithography and RET simulations are performed on a circuit layout and any observed problems are fed back to circuit designers for layout modification. The model based solutions are generally more reliable than the rule based solutions, but the simulations are very time consuming and it usually takes several iterations to reach closure.

In order to overcome the weakness of the methodology level approaches, lithography friendliness and/or RET compliance need to be considered directly in circuit layout algorithms.

C. ASIC Design Flow Considering Lithography Induced Effects

The International Technology Roadmap for Semiconductors (ITRS) projects that process variations present a critical challenge for both manufacturing yield and parametric yield of integrated circuit products. The process variations consist of systematic components and random components. The systematic variations represent both Front End Of Line (FEOL) and Back End Of Line (BEOL) parameter variations caused by predictable design and process procedures, such as CD (Critical Dimension) variations from different poly gate pitches and metal thickness variations occurred during Chemical Mechanical Planarization (CMP). Therefore, systematic variations behave deterministically in general.

In many existing design methodologies, people usually treat systematic variations together with random variations without differentiation. By handling both kinds of variations together in a process corner based methodology, people can conveniently circumvent relatively complex systematic variation models. However, such simplification usually causes unnecessary pessimism in process corner estimations especially when the systematic components account for a large portion of the overall variations. Indeed, it is reported in [27] that more than 50% of transistor gate length variations are due to systematic sources. As VLSI technology aggressively scales to 65nm and beyond, the influences from both systematic and random variations become greater and greater. The consequently expanding process corners force designers to set aggressive timing targets which intensify both design productivity crisis and power crisis [26]. Therefore, significant pessimism in process corner estimations is no longer tolerable and systematic variations need to considered differently from random variations.

Among systematic variations, transistor gate length variation has perhaps the largest impact on circuit timing and power performance since it directly affects both transistor switching speed and leakage power [28]. Fortunately, gate length variation largely depends on lithography process and can be captured through lithography/OPC (Optical Proximity Correction) simulations. A pioneer work [29] tried to estimate gate length variations through computationally expensive aerial image process simulations. Recently, a post OPC extraction methodology was proposed [30] for timing analysis of critical paths in a design. In [30], it is found that timing critical paths are changed when post OPC extraction information is utilized. However, the overall timing performance of a circuit is not altered by this methodology. Another work [31] proposed a timing analysis methodology with awareness of lithography induced gate length variations according to different poly pitches. For a standard cell, three poly spacing ranges are considered for its four boundaries and thereby 81 variants are characterized for each cell. The timing characteristic of a cell instance in a layout is obtained by matching its surrounding layout pattern with one of the 81 variants. Others [13] proposed a Restricted Design Rule (RDR) concept. This approach imposes more stringent constriants on designs to enhance manufacturing yield. It basically trades off design area and possible performance loss for less design variability.

In this chapter, we present a new standard cell characterization methodology considering lithography effects, then we extend this discussion into BEOL to investigate the timing impact of lithography effects on routing metals. Based on these result, we propose a new litho-aware timing analysis flow which considers lithography induced effects on both gate length variations and interconnect wire width variations. In this methodology, the timing and power performance of a cell is based on layout shapes obtained from lithography and

OPC simulations and the interconnect wire width variations is considered with a lookup table.

D. Organization of the Dissertation

The rest of the dissertation is organized as follows. In Chapter II, we present a technique for Alt-PSM compliant and composable standard cell library design. Chapter III contains a wire size and spacing technology for timing optimization and printability improvement. Chapter IV discusses a lithography aware ASIC design flow for standard timing analysis. Chapter V concludes this dissertation.

CHAPTER II

ALT-PSM COMPLIANCE AND COMPOSABILITY FOR STANDARD CELL LAYOUT

Alternating Phase Shift Mask (Alt-PSM) has been identified as one of the important Resolution Enhancement Techniques (RET) solutions for manufacturing of high performance designs because of its superior Critical Dimension (CD) control of the printed features. To facilitate the use of Alt-PSM in VLSI deep submicron manufacturing, we developed a new methodology for Alt-PSM aware library cell generation. We proposed a two-way approach for library cell generation and a new library development flow that is very easy to incorporate into the current design flow. The methodology we proposed guarantees that the top level designs using our Alt-PSM aware library do not have poly layer phase errors, which is required for manufacturing with Alt-PSM.

A. Introduction

In standard cell designs, one speedup method is to exploit the repetition usage of library cells. The Alt-PSM compliance for each library cell does not need to be obtained repeatedly, it can be achieved once in library cell designs. However, due to the proximity effect, placing Alt-PSM compliant cells adjacent to each other may cause new phase conflict. For example, in Figure 6, when the two Alt-PSM compliant cells are placed next to each other, phase conflict will happen between the two regions indicated by the arrow.

We propose a two way approach for developing standard cell library with Alt-PSM compliance and composability. For smaller standard cells where placement of transistors has minimal impact on the electrical performance of the standard cell, we will construct the standard cell layout to be Alt-PSM compliant and Alt-PSM composable; for larger standard cells with fixed transistor placement and routing for performance optimization,

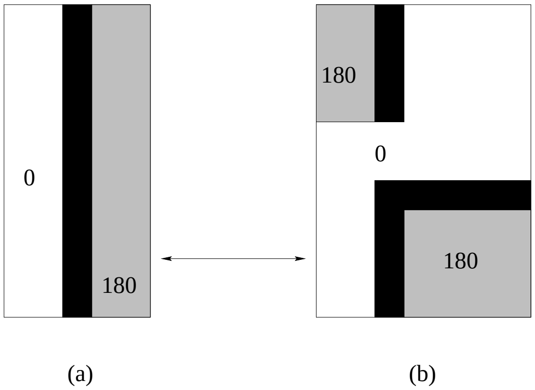


Fig. 6. Standard cell Alt-PSM composability.

we propose an optimal and efficient algorithm to modify the existing layout to achieve Alt-PSM compliance and composability.

For small standard cell construction, we consider the transistor placement and intra-cell routing at the same time so that the layout of poly layer and metal layer can match with each other. As an early classical work, Uehara and Cleemput [5] developed a graph based automatic cell layout system with a certain regular style. Recently, the minimum width version of the problem is tackled [6] with Boolean satisfiability(SAT) method [8]. We also propose to achieve Alt-PSM compliant and composable cell layout by using Boolean satisfiability. We transform the constraints for the standard cell synthesis to a SAT formulation and then use Siege SAT solver [7] to search for solution. Our formulation handles multiple sized transistors and considers transistor folding for large transistors.

For complicated standard cells, the placement of the transistors has been optimized for performance, we propose a network flow based algorithm to modify the existing layout for Alt-PSM composability with minimal cell area increase.

Our experiments show that Alt-PSM requirements can be satisfied efficiently at the IP development level. Block level and top level Alt-PSM compliance is guaranteed by our methodology.

The paper is organized as follows: section 2 presents the SAT formulation for standard cell construction, section 3 provides the algorithm for standard cell layout modification. The experiment results are presented in section 4. Section 5 is the conclusions of this work.

B. Library Cell Construction

For simple standard cells with small number of transistors, we use correct-by-construction approach for Alt-PSM compliant and composability. This section will present the SAT formulation for standard cell construction.

1. Layout Style

The layout style employed in this work follows the convention of [5, 6] and is illustrated by redrawing the example of [6] in Figure 7. The major characteristics of this style are summarized as follows.

- Transistors are placed in two rows, with PMOS at upper row and NMOS at bottom row.
- PMOS transistors are aligned with their bottom boundaries. NMOS transistors are aligned with their top boundaries.
- If a PMOS transistor and an NMOS transistor share the same GATE node, they are vertically aligned.
- The intra-cell routing uses only the poly layer and the Metal-1 layer.
- All nets connecting SOURCE/DRAIN nodes of PMOS are in P-Region.
- All nets connecting SOURCE/DRAIN nodes of NMOS are in N-Region.

- All nets connecting GATE nodes of are in G-Region, Top-Region or Bottem-Region.
- The VDD rail lies horizontally above the PMOS row and the GND rail lies horizontally below the NMOS row.
- No dogleg is employed.

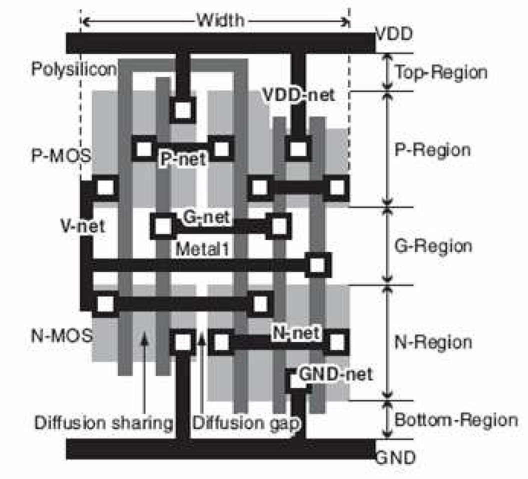

Width
VDD
Polysilicon
Top-Region
VDD-net
P-Region
P-MOS
P-net
V-net
G-net
Metal1
G-Region
N-Region
N-MOS
N-net
GND-net
Diffusion sharing
Diffusion gap
Bottom-Region
GND

Fig. 7. Library cell layout style.

It is not hard to see that this layout style is very regular and the constraints of regularity may sacrifice performance/area to certain degree.

2. Transistor Placement

This section describes the SAT formulation for transistor placement. Following the layout style mentioned in Figure 7 and placement formulation from [6], N PMOS transistors and N NMOS transistors need to be placed in minimum number of columns so that the resultant placement is PSM Clean and Composable. Each transistor's placement is defined by a set of boolean variables of length P, where $P = \lceil logW \rceil$, W being the number of columns. A variable of unit length is needed to define flip of each transistor. We call the flip variable as f_i. Total number of variables thus needed for placement is 2N $X(\lceil logW \rceil + 1)$. The placement constraints are:

MOS Overlap Constraints: Any two N/P transistors should not overlap in same column, i.e.

$$(x_n/x_{pi1} \oplus x_n/x_{pj1}) \vee (x_n/x_{pi2} \oplus x_n/x_{pj2}).... \vee (x_n/x_{pip} \oplus x_n/x_{pjp}) = 1; i \neq j; \qquad (2.1)$$

where $x_n/x_{pi1}, x_n/x_{pi2}..x_n/x_{pip}$ is the bit vector of length P defining placement of i^{th} N/P transistor. $x_{nip} \oplus x_{njp} = 1$ means that the column number of the ith NMOS transistor has the same value at the p^{th} digit as that of the column number of the j^{th} NMOS transistor.

Vertical Gate Constraint: N and P transistor placed in same column must share the same gate connection.

$$\vee_{j\varepsilon G}\overline{((x_{ni1} \oplus x_{pj1}) \vee (x_{ni2} \oplus x_{pj2}).... \vee (x_{nip} \oplus x_{pjp}))} = 1; \qquad (2.2)$$

where G is the set of P transistors having same gate as NMOS i. Basically, if there exists a PMOS transistor that shares the same gate connection with this NMOS transistor, this constraint aligns that PMOS transistor with the NMOS transistor. This configuration usually results in the most compact layout as the gate connection between P and N transistors can be made with the vertical poly. Similar expressions for NMOS transistor will provide the vertical gate constraint for PMOS transistors.

Neighboring MOS Constraint: transistors placed in neighboring columns must share diffusions: For N transistors

$$GAP_n(i,j) \bigvee [\bigvee_{k=0}^{W-2} (C_n(i,k) \wedge C_n(j,k+1)] = 1 \tag{2.3}$$

where $GAP_n(i,j) = 0$ if NMOS j is placed right of NMOS i to share diffusion.

$$GAP_n(i,j) = \overline{\overline{f_i f_j} same_{ds}(i,j) \vee \overline{f_i} f_j same_d(i,j) \vee f_i \overline{f_j} same_s(i,j) \vee f_i f_j same_{sd}(i,j)} \tag{2.4}$$

$same_{ds}(i,j)$ = 1 if i^{th} MOSs' drain is same as j^{th}'s source. $same_{sd}(i,j)$ = 1 if i^{th} MOSs' source is same as j^{th}'s drain. $same_s(i,j)$ = 1 if i^{th} MOSs' source is same as j^{th}'s source. $same_d(i,j)$ = 1 if i^{th} MOSs' drain is same as j^{th}'s drain.

f_i is the flip variable of MOS i. f_i is 0 if source is placed left of drain, 0 otherwise.

$C_n(i,k) = 1$ if N MOS i is placed at k^{th} column.

Similarly constraint is applied to all P transistors.

Transistor Folding Constraints: Smaller transistors resulting from splitting transistors larger than maximum allowable width should be placed together:

$$\bigvee_{k=0}^{W-B} [C_n(i,k) \wedge C_n(i+1,k+1) \wedge ....C_n(i+B-1,k+B-1)] = 1 \tag{2.5}$$

where B is the number of fingers a larger transistor is split into.

Flip of neighboring folded transistors should be opposite to ensure that the fingers share diffusion areas :

$$(f_n(i) \oplus f_n(i+1)) \wedge (f_n(i+1) \oplus f_n(i+2)) \wedge ...(f_n(i+B-2) \oplus f_n(i+B-1)) = 1 \tag{2.6}$$

Here, $f_n(i) \oplus f_n(i+1) = 1$ means that the diffusion area between two neighboring fingers of a folding transistor is either source or drain to both of the fingers, that in turn, indicates that the diffusion can be shared for these two fingers. Similar constraint is applies to PMOFETs as well.

PSM Constraint: For all transistors that are not folded, distance between the one's having the same gate connection should be even :

$$Q_{ij}(x_{nip} \oplus x_{njp}) = 1 \tag{2.7}$$

where Q_{ij} = 1 if gate $i - j$ is connected, 0 otherwise. x_{nip}/x_{njp} is the least significant bit of placement vector of NMOS i/j. Similar constraint is applied for PMOS transistors. This constraint is illustrated by point 1 in Figure 8. It is important to point out that this constraint is not necessary for placement stage of the standard cell since the gate connection can also be made by a metal 1 routing. But if it can be satisfied in placement stage without area cost, it will help increase the routing options of the standard cell.

Unused Column Constraint: If W $< 2^P$, no transistor can be placed in columns with top 2^P - W bit vectors.

$$(x_n/x_{pi1} \oplus u_{j1}) \vee (x_n/x_{pi2} \oplus u_{j2}) \vee ..(x_n/x_{piP} \oplus u_{jP}) = 1 \tag{2.8}$$

where $u_{j1}...u_{jP}$ is the unused column bit vector.

3. Routing

This section explains the intra-cell routing. Basically, we need to find available routing tracks for each net that has to be connected through routing. We have followed the routing style from [7]. Let P_n, P_p, P_g and P_v be the number of boolean variable for N, P, G and V nets respectively. If W_n, W_p, W_g be the number of rows in N, P and G region and W_v the number of columns in G region, then $P_n = \lceil logW_n \rceil$, $P_p = \lceil log_2 W_p \rceil$, $P_v = \lceil log_2 W_v \rceil$, and $P_g = \lceil logW_2(W_g + 2) \rceil$. A connection between two gate terminals is called a Poly connection if the net is placed in Over/Under G Region, a Gate Net if in G Region.

Poly Connection Constraints: Poly connection between two gate terminals is considered as a single instance. Let x_{ij} represents the variable that assumes value of 1 if a gate connection $i-j$ is necessary, otherwise 0. Let a_{ij} be 1 if gate connection $i-j$ uses the G region, otherwise 0.All instances in a G-Net with multiple gate terminals are treated as separate G-Nets. For example in a G-Net with gate terminals i, j and k that need to be connected together.
There should not be redundant connections:

$$x_{ij} \wedge x_{jk} \wedge x_{ik} = 0 \tag{2.9}$$

Terminals i, j and k should be connected together:

$$(x_{ij} \wedge x_{jk}) \vee (x_{ij} \wedge x_{ik}) \vee (x_{ik} \wedge x_{jk}) = 1 \tag{2.10}$$

Net Overlap Constraints: Any two nets i, j have overlapping intervals, can not be placed in same row.

$$(n_{i1} \oplus n_{j1}) \vee (n_{i2} \oplus n_{j2}) .... \vee (n_{ip} \oplus n_{jp}) = 1; i \neq j; I_n(i) \cap I_n(j) \neq \phi \tag{2.11}$$

where n_{ip} is the p^{th} significant bit for the track assignment for net i in N region, $I_n(i)$ is the range of column numbers that net i crosses. For example, a connection for net x has been made from two transistors in column 2 and column 5 respectively, then $I_n(x) = 2, 3, 4, 5$. Similar constraints apply for P and G regions.

VDD/GND Constraints: VDD connection will block the top metal track and GND connection will block the bottom metal routing track. If a N net has a GND in its interval

$I_n(i)$, it can not be placed in lowest row in N Region. If a P net has a VDD in its interval $I_P(i)$, it can not be placed in highest row in the P region.

$$(n/p_{i1} \oplus L_1) \vee (n/p_{i2} \oplus L_2) .... \vee (n/p_{ip} \oplus L_p) = 1; VDD/GND \in I_{n/p}(i/p) \tag{2.12}$$

where L_p is the p^{th} significant bit for the highest routing track in P region or the p^{th} significant bit for the lowest routing track in N region, depending on the n or p constraint used in the above equation.

V-Net Pass Constraints: We defined V-net in Figure 7 as a net that has PMOS and NMOS drain connection made vertically by a metal connection. Obviously, a V-net is going to block horizontal metal tracks at the location where the PMOS and NMOS drain connection is made. For this reason, special constraints have to be considered for the V-nets routing to avoid short on different nets. If we assign the vertical drain connection of a V-net to be at a specific column, other V-nets along with all its gate terminals should not take that column assignment to avoid short of the two V-nets. This is illustrated in Figure 8 at point 3 and point 4, two V-nets do not across each other. This constraint is represented by:

$$A_{i j_{k1}} \oplus A_{i j_{k2}} = 0; k1, k2 \in I_v(i) \tag{2.13}$$

where $A_{i j_{k1}} = 1$ means the source-drain connection of V-net i is at the left of the column k1 that are spanned by another V-net j. $I_v(i)$ is the range of column numbers that V-net i crosses.

For a G-net connection, it should not short with the vertical connection of the V-net drain connection for P and NMOS. A G-Net spans from column i to column j should be span the column k where a drain connection of a V-net is made. i.e.

$$x_{ij}a_{ij}(G_{ik} \oplus G_{jk}) = 0; \tag{2.14}$$

where $G_{ik} = 1$ means column i is at the left of column k. $a_{ij} = 1$ means that gate connection is done in G-region, $a_{ij} = 0$ means that gate connection is done with poly routing at the top or bottom of the cell.

Also for a G-net that spans from column i to column j, it should not take the same routing track assignment as the horizontal portion of the V-net l if both of the nets span any same column. i.e.

$$x_{ij}a_{ij}(G_{ik} \oplus G_{jk})\overline{[(g_{ij1} \oplus g_{k1}) \vee \ldots.(g_{ijPg} \oplus g_{kPg})]} = 0; k \in I_v(l) \tag{2.15}$$

Similar constraints apply for N nets and P nets.

Connections to gate terminals of two V-Nets if overlap, should not be placed in same row in G-Region. For any V-Net i with gate terminal j should not overlap another V-Net l with gate terminal k :

$$(A_{ik} \oplus A_{jk})\overline{[((g_{ij1} \oplus g_{kl1}) \vee ..(g_{iPg} \oplus g_{klPg}))]} = 0 \tag{2.16}$$

Similar constraints apply in N region and P region as well to avoid short of two V-nets.

Poly Routing Overlap Constraints: For two different poly nets, namely poly net m which spans from column i to column j and poly net n which spans from column k and column l can not overlap with the poly connection made at the top or bottom of the cell:

$$x_{ij} \wedge x_{kl} \wedge \overline{a_{ij}} \wedge \overline{a_{kl}} \wedge \overline{[(g_{m1} \oplus g_{n1}) \vee \ldots(g_{mPg} \oplus g_{nPg})]} = 0; I_{poly}(m) \cap I_{poly}(n) \neq \phi \tag{2.17}$$

where g_{mp} is the p^{th} significant bit of the row number in top/bottom regions assigned to poly gate connection net m. $I_{poly}(m)$ is the range of column numbers that poly net m crosses.

As we showed in Figure 2b, T-shaped poly connection creates a PSM phase error. The constraint to avoid this T-shape poly connection configuration is described as follows: two same poly nets with terminals $i-j$ and $k-l$, both can not be placed in either over or under region. This constraint can be expressed as:

$$x_{ij} \wedge x_{kl} \wedge \overline{a_{ij}} \wedge \overline{a_{kl}} \wedge \overline{[(g_{ij1} \oplus g_{kl1}) \vee \ldots (g_{ijPg} \oplus g_{klPg})]} = 0 \tag{2.18}$$

The example of this can be found in Figure 8. The gate connections at point 1 and point 4 in Figure 8 show that one of the gate connection is made with bottom poly routing track but the other connection is made with a metal 1 in G region.

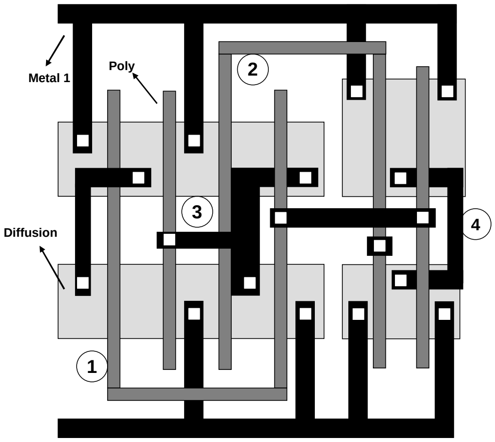


Fig. 8. Layout constraints examples.

Poly PSM Phase Error Constraint: If all columns between two poly nets are occupied by transistor gates, the number of poly gates in such interval should be even. This constraint makes sure that there is no phase error generated when poly routing exists at the top/bottom of the cell. It is further illustrated in Figure 8 as point 1.

$$x_{ij} \wedge x_{kl} \wedge \overline{a_{ij}} \wedge\ C_{ij} \wedge \overline{(l_{Pg} \oplus r_{Pg})} = 0 \tag{2.19}$$

C_{ij} = 1 if all columns between i and j are occupied, 0 otherwise. l_{Pg}/r_{Pg} are the least significant bits of left/right column of a poly connection.

Composibility Constraint: For any two poly connections $i - j$ ($i < j$) and $k - l$ ($k < l$), the column numbers of one poly connection should have same least significant bit as other connection if all the intervals are occupied. Satisfying this constraint guarantees that phase assignment at top and bottom of the cell is going to be same, thus achieve Alt-PSM composability. An example of this situation is shown in Figure 8 as point 1 and 2.

$$x_{ij} \wedge \overline{a_{ij}} \wedge x_{kl} \wedge \overline{a_{kl}} \wedge [C_{ik}(l_{pgi} \oplus l_{pgk}) \vee C_{il}\overline{(l_{pgi} \oplus l_{pgl})} \vee C_{jk}\overline{(l_{pgj} \oplus l_{pgk})} \vee C_{jl}(l_{pgj} \oplus l_{pgl}) = 0 \tag{2.20}$$

Metal 1 Layer PSM Constraint: To avoid T-type connections that introduce phase assignment problem in Metal 1 layer, the width of T's top can be increased to make it a non critical feature, thus preventing any phase error. In V-Nets with gate connections, such situations are very likely. To overcome this, if a V-Net has a T connection, the width of its vertical or source-drain connection is increased beyond critical size(see point 3 in Figure 8). To provide enough space and design rule compliance, both columns in vicinity of such V-Nets can not be occupied. At least one column should be empty.

$$\overline{V_{c-1} \wedge V_{c+1}} = 1 \tag{2.21}$$

where V_{c-1} = 1 if the column before V-Net source-drain column is occupied, 0 otherwise. V_{c+1} = 1 if the column after V-Net source-drain is occupied, 0 otherwise. However if V-Net with T connection is placed at column 0, column 1 should be unoccupied and similarly if in last column, last but one should be empty.

4. SAT Formulation Flow

The flow for generating standard cell with SAT formulation is shown in Figure 9.

C. Cell Alt-PSM Composability

For complicated standard cells, the placement of the transistor is determined primarily by performance and area. While our previous approach does yield the smallest area for the standard cell design, it does not account for timing tradeoffs of the transistor placement. Thus we will use the following methodology for Alt-PSM compliant and composability: we use the method proposed by [4] to achieve Alt-PSM compliant and then we use the algorithm presented in this section to achieve Alt-PSM composability.

1. Problem Formulation

We start by assuming that all standard cells are free of internal phase errors. It is then straightforward to derive the fact that if we can assign the boundary regions with the same phase for all standard cells, we create a standard cell library that is composable, i.e. putting any two standard cells together will not generate phase errors. Thus the Alt-PSM composability problem is described as follows:

Given standard cell design without phase errors, determine if the design is Alt-PSM composable. If not, find the minimum modification of the design in order to make it composable.

As in previous work of Alt-PSM [9], the relation of $b < B$ holds, where b is the minimum spacing between two features defined by design rules, B is the minimum spacing for features with different phases between them. It is also commonly true that $B < 2b$.

```
Procedure: PSMPlaceRoute
Input:  standard cell schematic netlist
Output: PSM compliant and composable standard
            cell layout.
1. $W \leftarrow$ # NMOS (# PMOS)
2. Generate and solve CNF for placement constraints
3. While placement is UNSAT{
  $W \leftarrow W + 1$
  Generate & solve new CNF for placement.
  }
4. Apply PSM constraint for Placement
5. If (solution is SAT) Retain placement
  else
    Revert to last satisfiable placement
6. Generate & solve CNF for routing constraints
7. While routing is UNSAT{
  If routing not possible at $W$
    $W \leftarrow W + 1$
    Go back to 2
  }
```

Fig. 9. PSM aware library cell synthesis algorithm.

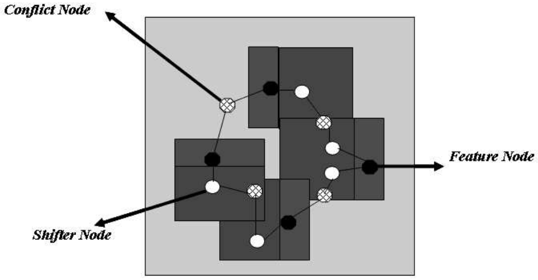


Fig. 10. Phase conflict graph.

We construct the phase conflict graph according to [4] in Figure 10. In the graph, each feature is represented by a feature node, the region with 180 degree light is designated by a shifter node, and if space between two features is less than B, i.e. the two features are in phase conflict, there is a conflict node representing this situation. There exists an edge between a feature node and its associated shifter node, the shifter node and the conflict node. Also an edge between a feature node and the conflict node is added if there is no shifter node in that side of the feature. In case one feature is in phase conflict with multiple features, each phase conflict will be represented by a different conflict node and a different shifter node if the conflict occurs at the shifter side of the feature. The work of [4] has proved that this graph is planar. The layout is phase error free if the phase conflict graph is a bi-partite graph [4], i.e., there is no odd cycle in the graph.

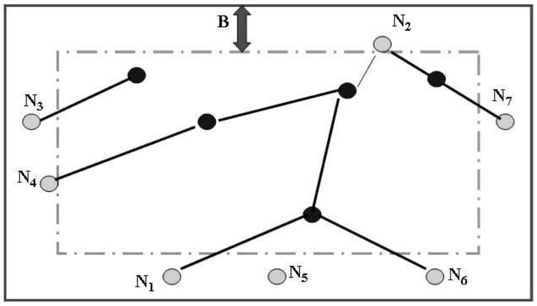


Fig. 11. Boundary nodes in phase conflict graph.

We define *critical boundaries* to be the boundaries of a cell that need the same phase for composability. Critical boundaries can be specified by designers, for a row based design, critical boundaries are the left and right side of the cell borders. In a conflict graph, for every feature node whose location is within a certain distance B to any critical boundary of the cell, we define another type of node - *boundary node*, it is either the feature node if the shifter of the feature node is pointing into the cell, or the shifter node of the feature otherwise. For example, assuming the phase conflict graph looks like Figure 11, if critical boundaries are the four boundaries of the cell, node N1-N7 are boundary nodes. We further define the *odd path* in a graph as a path composed by odd number of edges. For example, the path between N2 and N1 in Figure 11 is an odd path.

We want to point out an obvious observation: we can assign the same phase to all critical boundary *if and only if* there is no odd path existed between any two of the boundary nodes in the phase conflict graph. Another fact is that if there are multiple paths between two boundary nodes, either all of them are odd paths or none of them is an odd path, since there is no odd cycle in a conflict graph for a cell layout without phase error itself. Thus, the problem of cell composability is reduced to:

Given a phase conflict graph of a cell layout without phase errors, determine if odd path between any two boundary nodes exists. If so, determine a minimum change of layout that can remove such odd paths.

2. Algorithm

For every edge in phase conflict graph, we define the *edge weight* as follows: For an edge that has a conflict node incident on it, if we traverse this edge in both directions, we will reach two feature nodes in phase conflict, we define the weight of the edge to be the silicon area penalty if we have to increase the spacing between those two features to remove the phase conflict. For all other edges, the edge weights are infinite.

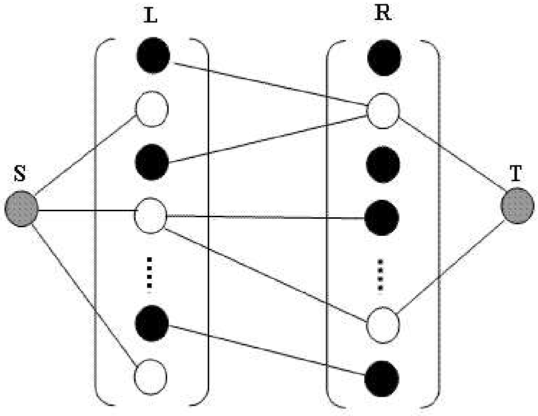


Fig. 12. Phase conflict bipartite graph: empty dots represent boundary nodes, solid dots represent all other nodes.

Since the original cell layout does not have any phase errors, the corresponding phase conflict graph is bi-partite, i.e. we can color the graph with two colors. Then, the nodes can be partitioned into two sets L and R, with all the edges between L and R. For any node x in L and node y in R, if they are connected in the conflict graph, the path from x to y is an odd path. For any two nodes x and y in the same set, the path from x to y can not be an odd path. Hence, the composability problem becomes to find the minimum edge weight cut to separate boundary nodes that belong to different sets. In Figure 12, the empty dots indicate boundary nodes and the solid nodes indicate all other nodes. We use the network flow model to separate the set of boundary nodes in subset L and those in subset R by introducing a source node S and a sink node T. We also add edges between S and all the boundary nodes in subset L and edges between T and all the boundary nodes in subset R, with all the added edges having infinite edge weights We then search for a minimum weight cut of S and T, the result will give us the minimum cut of the boundary nodes in L and R, which is the same as the minimum change of the layout for composability.

The optimality of this algorithm is obvious. If we cut all edges with finite weight, we can separate S from T, which implies that the minimum cut of S and T has to be finite, and each edge that has been cut has a conflict node incident on it since the weight of the edge is infinite otherwise. Because all the cut that we make are necessary and the weight of the cut is minimized, this algorithm is optimal for the design. Note that the cut could also include an edge that has no boundary node incident on it, but it has to have a conflict node incident on it.

We analyze the complexity of this algorithm as follows: the generation of the conflict graph has the complexity of O(nlogn) as demonstrated in [4], with n being the number of features. The partitioning of the nodes into L and R sets has the complexity of O(n) as each feature needs to be processed only once. We use the Minimum Capacity Cut algorithm [10] as the S-T minimum cut algorithm, the capacity of each edge is the weight of the edge. This minimum cut algorithm takes time $O(n^3)$. Therefore, the complexity of our algorithm is $O(n^3)$. Since the number of feature in a typical standard cell is fairly small, the running time for our algorithm is quite reasonable.

D. Standard Cell Library Design Flow

In Figure 13, we propose a standard cell design flow for Alt-PSM compliant and composability. We take the standard cell schematic or netlist and use the Alt-PSM aware cell construction flow to generate the layout of the cell, if the cell meets the performance specification, this is the final design. If not, we go through the normal design flow to generate the layout and try to modify the layout for Alt-PSM purposes, the final design also has to satisfy the performance specification.

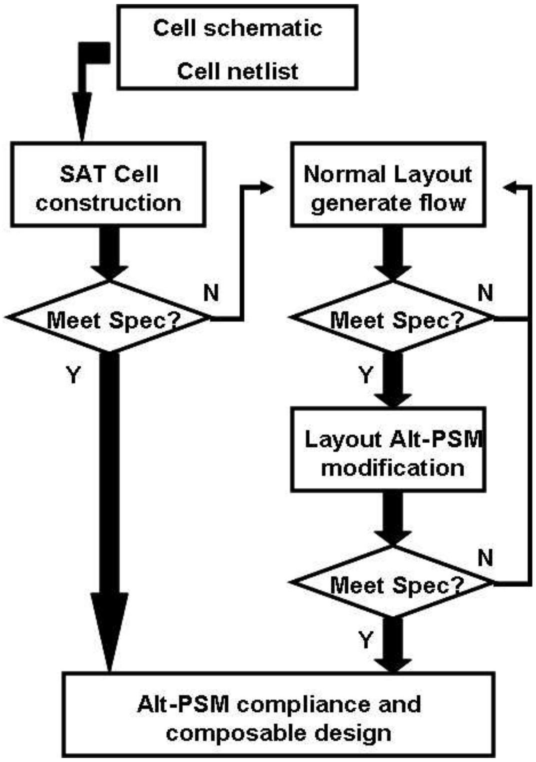

Cell schematic
Cell netlist
SAT Cell construction
Meet Spec?
N
Y
Normal Layout generate flow
Meet Spec?
N
Y
Layout Alt-PSM modification
Meet Spec?
N
Y
Alt-PSM compliance and composable design

Fig. 13. Standard cell library design flow.

E. Experiments

1. Standard Cell Construction

The constraints for Placement and Routing described above were transformed into SAT formulations and solved using Siege Variant 4 SAT Solver [7]. In additional to its fast run time, Siege SAT solver has the major advantage that its output depends on it's seed. This helps to avoid adding additional clauses during placement stage to suppress an existing placement result in order to get a new one.

Table I. Experimental results for standard cell construction.

Ckt	**# trs**	**Pvars**	**Rvars**	**Cols**	**CPU(s)**
and3	8	24	10	4	0.03
and8	16	90	12	9	0.73
aoi22	10	40	7	6	0.05
nd2ab	8	32	17	5	0.32
nor2	4	48	21	6	0.23
mux2	10	40	23	7	0.07
nand3	6	40	14	5	0.12
xor2	12	48	30	8	0.58
cgi2	10	40	18	6	0.04
nor5	10	40	3	5	0.08
nor8	18	64	18	8	0.53

Our formulation has been implemented in ’C++’ and experiments were conducted on Linux server with 2GB of RAM. A timeout of 3000s was used in our experiment. Table 1 lists some of the test cases used in our experimentation. The fist column gives the circuit name, followed by the number of transistors in each circuit, the number of variables for Placement, number of variables for Routing, the resultant width in columns, and the time taken to get a satisfiable result. The number of rows in N/P region are assumed to be 3, for G region 2 and one row each in Over and Under G region. Apart from being of minimum width, the layouts generated by our formulation are PSM Compliant and Composable.

Figure 14 shows a comparison of layouts with and without PSM considerations in placement and routing phases for a 2 input XOR gate that uses 12 transistors. Two T junctions (highlighted by circles) can be noticed in Metal 1 Layer that are not PSM compliant. To avoid any phase conflict, the width of source-drain portion of the two V-Nets needs to increased beyond critical feature size. This is shown in Figure 14(b). Without any increase in area, we generate a PSM clean layout. Figure 15 compare the layouts for a 2 input NOR gate with each transistor split into 3 fingers each. Again T junctions in POLY layer are removed to get a PSM compliant and composable layout.

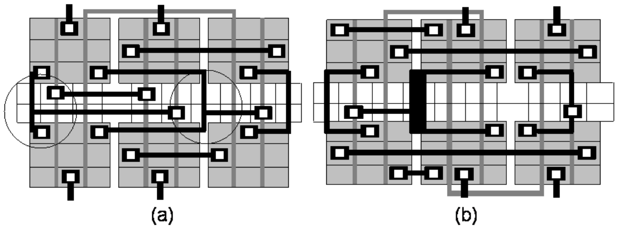


Fig. 14. (a) XOR gate layout: T joints in Metal 1 layer that pose phase conflict problem. (b) PSM clean layout.

2. Layout Modification

We ran the algorithm on a selected set of test cells. All test cells are pre-processed so that they are Alt-PSM compliant before applying our algorithm. This pre-process is done

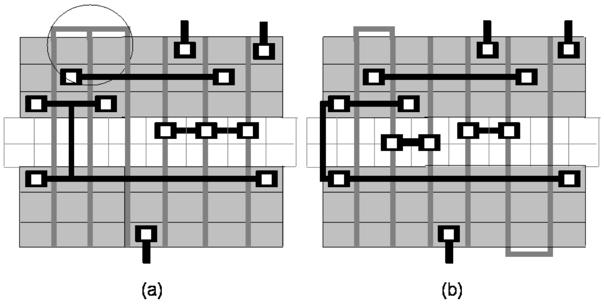


Fig. 15. (a) 2 Input NOR gate layout with N-P pairs split in 3 fingers each: T joints in Poly. (b) PSM compliant and composable. layout

manually as typical standard cells require only minor changes for Alt-PSM compliance. Table 2 shows the area increment comparison of our algorithm and the method of applying blank area around the cells. We used the minimum poly spacing b = 0.16 unit length and phase conflict spacing B = 0.32 unit length.

The comparison shows area savings of our approach. For the library cells that will be placed into the design for multiple times, the total area saving for Alt-PSM designs will be significant.

F. Conclusion

We have proposed a two-way approach to generate Alt-PSM aware standard cell library. Our SAT based methodology results in Alt-PSM clean standard cells with minimum area while our cell modification based methodology generates Alt-PSM clean standard cells with minimum impact on cell performance. We introduced a new Alt-PSM aware standard cell design flow. Using our approach, the generated standard cell library can be used directly in the place and route environment without any concerns of poly layer Alt-PSM phase errors.

Table II. Experimental results for standard cell modification.

Cell	**# of transistors**	**Initial area** (um^2)	**Area increase with composable processing** (um^2)	**Area increase with blank area insertion** (um^2)
Balanced AND	8	4.76	0.21	1.50
Divider	99	46.99	3.86	8.03
Scanable D-Latch	54	21.42	0.041	4.08
Clock Gating	56	29.04	0.041	5.26
2:1 Multiplexer	12	4.76	0.12	1.50
Boundary Scan Reg	68	34.28	0.21	6.06
Scanable DFF	64	39.04	0.22	6.80
Average	52	25.76	0.67	4.75

CHAPTER III

WIRE SIZING AND SPACING FOR LITHOGRAPHIC PRINTABILITY AND TIMING OPTIMIZATION

The printability problem due to strong diffraction effects poses a serious threat to the progress of VLSI technology. A circuit layout with poor printability implies that it is difficult to make the printed features on wafers follow designed shapes without distortions. The development of Resolution Enhancement Techniques (RET) can alleviate the printability problem but cannot reverse the trend of deterioration. Moreover, over-usage of RET may dramatically increase photo-mask cost and increase the cycle time for volume production. Thus, there is a strong demand to consider the sub-wavelength printability problem in circuit layout designs. However, layout printability optimization should not degrade circuit timing performance. In this chapter, we introduce a wire sizing and spacing method to improve wire printability with minimal adverse impact on interconnect timing performance. A new printability model is proposed to handle partially coherent illuminations. The complex printability and timing optimization problem is solved in a 2-phase approach. The difficulty of the printability optimization due to its multimodal nature is handled with a sensitivity based heuristic. A coupling aware timing driven continuous wire sizing algorithm is also provided. Lithographic simulation results show that our approach can improve the printability in term of EPE (Edge Placement Error) by 20% – 40% without violating timing, wire width and spacing constraints.

A. Introduction

There are previous works to address DFM issues within design environment. In [3, 4], the Alternating PSM compliance is modeled as a graph problem and layout modification algorithms are developed to achieve Alternating PSM compliance with minimum cost change where the cost can be defined in term of area or timing. Compared with Alternating PSM compliance, the other RET procedures are very difficult to be abstracted into concise models that can be easily embedded in automatic layout algorithms. Recently, the work of [14] circumvents the difficult RET abstraction issue by optimizing interference intensity instead. This is based on the observation that a reduced interference intensity can alleviate the workload of RET and the interference intensity model is relatively easier to be obtained. An OPC friendly maze routing algorithm is proposed in [14] using the interference intensity model. An RET aware routing algorithm based on fast litho simulation is proposed in [16].

In layout printability optimizations, conventional design objectives such as timing performance cannot be ignored since timing performance heavily depends on physical layout in today's interconnect dominated technology [17]. In this chapter, we focus on the wire sizing problem which plays an important role on affecting timing performance and we combine wire sizing and spacing to improve printability of the design. There are many previous works on wire sizing but mostly for timing optimization alone. The work of [17] attempts to minimize a weighted sum of sink delays for a Steiner tree. A sensitivity based wire sizing heuristic is reported in [18]. A dynamic programming based simultaneous buffer insertion and wire sizing algorithm is proposed in [19] to achieve timing-area tradeoff for a Steiner tree. A circuit-wise gate sizing and wire sizing method based on local refinement is developed in [20]. In [21], the circuit-wise simultaneous gate sizing and wire sizing problem is solved optimally using Lagrangian relaxation. A simultaneous wire sizing and spacing algorithm considering coupling capacitance is introduced in [22].

In this chapter, we address both timing optimization amd sub-wavelength printability issues in both wire sizing and spacing. We propose a new printability model and we also show that the printability model can be used in the aggressively optimized design to improve printability significantly. By doing timing and printability optimizations in series, we also imply this approach as the use model of the lithography optimization for interconnection metals. Our goal is to improve wire printability, i.e., make printed wires have sharper boundaries, so that the cost of RET and photo-mask can be reduced. Moreover, the printability driven wire sizing method should minimize any adverse impact on interconnect timing performance. The major contributions of this work are listed as follows.

- We propose a new approximated printability model for layout optimization. The lithography procedure is so complicated that a practical printability model becomes a bottleneck for printability optimization. Compared to the model in [14], our model has two advantages: (1) our model can handle partially coherent illuminations which is the mainstream illumination method in practical photolithography while the model in [14] is limited to coherent illuminations; (2) our model directly measures the feature sharpness and considers the overall light intensity effect instead of considering only interference light intensity as in [14].

- The complicated printability and timing optimization problem is solved in a 2-phase approach considering different problem natures of printability and timing. The difficulty of the printability optimization due to its multimodal nature is handled with a sensitivity based heuristic.

- A coupling aware timing driven continuous wire sizing algorithm is also provided. The closest works are [21] which does not consider coupling capacitance, and [22] which is only for discrete wire sizing.

Implementing litho friendly design techniques in design phase, such as our approach to adjust wire sizing and spacing, will alleviate the printability problem. It will help to reduce the effort level of OPC/RET that has to be performed at the manufacturing stage. This will have a direct advantage at reducing mask cost, but the more important cost saving is realized by reducing the number of iteration cycles of the mask correction, which contributes to shorten the time needed to achieve volume production. Lithographic simulation results show that our approach can improve the wire printability in term of EPE (Edge Placement Error) by 20%-40% without violating timing and wire width/spacing constraints.

B. Problem Formulation

The input to the wire sizing problem is a set of Steiner trees $\mathcal{T} = \{T_1, T_2, ...\}$ representing the layout of signal nets. In $\mathcal{T}$, there is a set of wire edges $E = \{e_1, e_2, ...\}$ and a set of sink nodes $S = \{s_1, s_2, ...\}$ such that each edge and each sink belong to a certain Steiner tree. Each edge $e_i \in E$ has a width of w_i which is bounded in a range of $[L_i, U_i]$. The edge width vector for all edges is $\mathbf{w} = (w_1, w_2, ...)^t$, location vector of all edges are $\mathbf{x} = (x_1, x_2, ...)^t$ and $\mathbf{y} = (y_1, y_2, ...)^t$. In Figure 16, an example for horizontal wires is illustrated. The space between the wires can be calculated as $s_{ik} = (y_k - y_i) - \frac{w_k}{2} - \frac{w_i}{2}$.

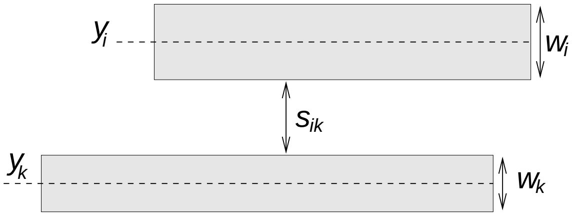


Fig. 16. Wire segment parameters.

Each sink $s_j \in S$ has a required arrival time (RAT) q_j and a delay t_j whose model is provided in Section D. The printability function $\Theta(\mathbf{w}, \mathbf{x}, \mathbf{y})$ will be defined in Section C. We will solve the following problem in this work:

$$\text{Maximize} \quad \Theta(\mathbf{w}, \mathbf{x}, \mathbf{y}) \tag{3.1}$$

$$\text{Subject to} \quad t_j(\mathbf{w}, \mathbf{x}, \mathbf{y}) \leq q_j \quad \forall s_j \in S \tag{3.2}$$

$$L_i \leq w_i \leq U_i \qquad \forall e_i \in E \tag{3.3}$$

$$s_{min} \leq s_{ij} \leq s_{max} \quad \forall \text{ adjacent wires } i \text{ and } j \tag{3.4}$$

In other words, we attempt to maximize the overall printability of the layout subject to timing and wire width/spacing constraints. This optimization framework can include other objectives such as area and power consumption. We consider continuous wire sizing as in [21] so that it is easier to handle the complicated printability function. If discrete wire sizing solutions are needed, they can be obtained through rounding the continuous solutions as in [18].

The above seemingly simple formulation is actually a rather difficult non-linear programming problem, since the expressions for objective (3.1) and constraint (3.2) are complicated. Especially, the objective function Θ is not unimodal in general. Therefore, we propose to solve this problem in the following two phases:

1. Obtain a wire sizing solution considering coupling that satisfies constraints (3.2), (3.3) and (3.4) regardless the printability. A Lagrangian relaxation based algorithm is described in Section D for solving this sub-problem.

2. Based on the solution of phase 1, maximize the printability Θ while the constraints (3.2), (3.3) and (3.4) are still satisfied. A sensitivity based local adjustment heuristic is introduced in Section E to solve this sub-problem by adjusting both wire width and wire spacing. The result of this phase is a much better design in terms of printability, yet still satisfies all the timing and wire width/spacing constraints.

The problem natures of the printability maximization and satisfying delay constraints are different. The delay optimizations are net based and have no clear geometrical boundary, especially when coupling capacitance is considered. In other words, the delays for sinks far apart are mingled with each other through the nets and coupling. In contrast, the lithographic effect from an edge or to an edge is localized. By solving the timing constraints first in phase 1, phase 2 can be focused on maximizing the printability function through geometrically local adjustments. The different problem natures also justify why our 2-phase approach is more practical than solving the entire problem using Lagrangian relaxation as in [21]. In contrast, the problem in [21] is unimodal as printability is not considered.

C. Printability Model

1. Aerial Image

Since the printability model is based on the light intensity distribution on the wafer plane, we first discuss the light intensity models for three basic types of illuminations: (1) coherent, (2) incoherent and (3) partially coherent. In the following discussions, we assume that the optical system is a 1× reduction system. Although practical steppers and scanners are usually 4× or 5× reduction systems, a 1× system with the same NA(numerical aperture) [1] gives essentially identical printing results under the assumption of thin mask approximation and aberration-free.

Coherent illumination: The complex field distribution $g_i(x_i, y_i)$ on the image plane (wafer) can be expressed as:

$$g_i(x_i, y_i) = \int\int h(x_i - x_o, y_i - y_o) g_o(x_o, y_o) dx_o dy_o \qquad (3.5)$$

where $g_o(x_o, y_o)$ is the complex field distribution on the object plane (mask) and $h(x, y)$ represents the impulse response function of the optical system. In the frequency domain, spatial frequency along x and y directions are denoted as f_x and f_y, respectively, and the above equation can be rewritten as:

$$G_i(f_x, f_y) = H(f_x, f_y)G_o(f_x, f_y) \tag{3.6}$$

where $G_i(f_x, f_y)$ and $G_o(f_x, f_y)$ are obtained through Fourier transform of $g_i(x_i, y_i)$ and $g_o(x_o, y_o)$, respectively. The coherent transfer function $H(f_x, f_y)$ is given by:

$$H(f_x, f_y) = \begin{cases} 1 & : \quad \sqrt{f_x^2 + f_y^2} \leq \frac{NA}{\lambda} \\ 0 & : \quad \text{otherwise} \end{cases}$$

where λ is the lithographic wavelength.

Incoherent illumination: The aerial image formation under incoherent illumination is based on linear superposition of light intensity.

$$I_i(x_i, y_i) = \int\int |h(x_i - x_o, y_i - y_o)|^2 I_o(x_o, y_o) dx_o dy_o \tag{3.7}$$

where $I_i(x_i, y_i) = |g_i(x_i, y_i)|^2$, $I_o(x_o, y_o) = |g_o(x_o, y_o)|^2$ are the light intensity distribution on the image and object planes, respectively.

Partially coherent illumination: In reality, almost all practical photolithography systems employ partially coherent illumination, although the coherent and incoherent illumination based models provide theoretic foundations for partially coherent illumination based models. For partially coherent illuminations, there are several existing methods of computing aerial image such as Hopkins formula and eigenfunction expansion [23]. Abbe's approach and eigenfunction expansion method decompose the illumination source into many coherent sources, then calculates the complex fields due to each source, and finally add together the light intensities due to each source to obtain the total light intensity distribution. Hopkins formula requires the computation of Transmission Cross-Coefficients. These existing methods are too computationally expensive to be adopted in the wire sizing and spacing optimization procedure.

We propose an optimization friendly approximated model for partially coherent illuminations based on a linear combination of coherent and incoherent imaging. In the proposed model, the wafer image is given by:

$$I_{pc}(x_i, y_i) = (1 - \sigma^2)|g_i(x_i, y_i)|^2 + \sigma^2 I_i(x_i, y_i) \tag{3.8}$$

where $\sigma \in [0, 1]$ is the partial coherence factor, $\sigma = 0$ corresponds to coherent illumination and $\sigma = 1$ approaches incoherent illumination.

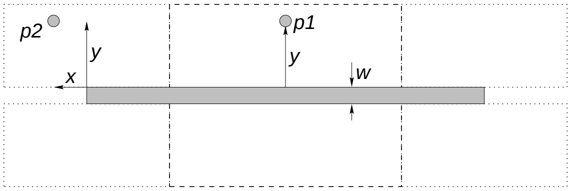
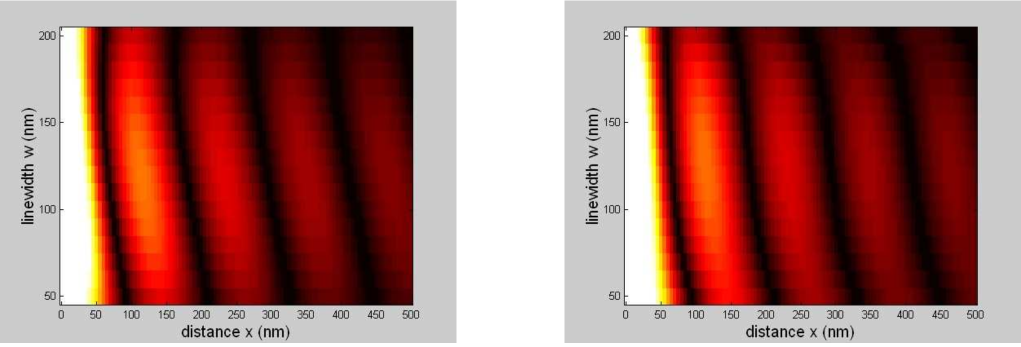


Fig. 17. Infinite line model in dashed bounding box. Semi-infinite line model in dotted boxes.

The complex field at one point p due to a line segment (wire segment) e under coherent illumination can be approximated by a quadratic function. If the width of the segment is w, the distance vector from the segment e to p is $\vec{d}$, then the complex field contributed by e at p can be approximated as $a_0(\vec{d}) + a_1(\vec{d})w + a_2(\vec{d})w^2$ which is a quadratic function of w. The coefficients of this quadratic function depend on the distance vector $\vec{d}$. Similarly, the light intensity from incoherent illumination can also be approximated as $b_0(\vec{d}) + b_1(\vec{d})w + b_2(\vec{d})w^2$. If there are m line segment with widths of $w_1, w_2, ...w_m$, and the distances from p to them are $\vec{d}_1, \vec{d}_2, ..., \vec{d}_m$, then the total light intensity at p is:

$$\begin{aligned} I_p &= (1 - \sigma^2)|\sum_{k=1}^{m}[a_0(\vec{d}_k) + a_1(\vec{d}_k)w_k + a_2(\vec{d}_k)w_k^2]|^2 \\ &+\sigma^2 \sum_{k=1}^{m} b_0(\vec{d}_k) + b_1(\vec{d}_k)w_k + b_2(\vec{d}_k)w_k^2 \end{aligned} \tag{3.9}$$

The distance vector $\vec{d}$ is determined differently in two cases. If the point is around the middle of segment e as $p1$ in Figure 17, the infinite line model is applied and $\vec{d}$ is equivalent to a distance scalar y in Figure 17. Since the coefficient functions such as $a_0(\vec{d})$ and $b_1(\vec{d})$ are multimodal and very complex, their values are saved in a lookup table. If the point is close to one end of the segment as $p2$ in Figure 17, the semi-infinite line model is employed. In this case, the distance vector $\vec{d}$ is decided by x and y component as shown in Figure 17. Consequently, a 2-D lookup table is needed for the semi-infinite line model. In contrast to the model in [14], our model does not depend on the segment length directly and therefore the number of lookup tables can be reduced.



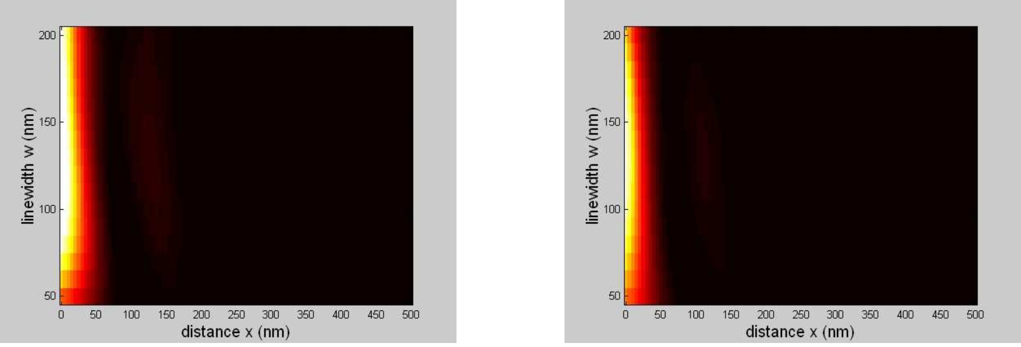
(a) From lithographic simulation

(b) From our approximated model

Fig. 18. Contour plot of complex field amplitude vs. feature linewidth and distance to feature for coherent illumination.

Comparisons between the approximated model and simulation [24] results are shown in Figure 18 and Figure 19. Figure 18 plots the contours of the amplitude of complex field for coherent illumination with respect to the linewidth of a feature and distance to the feature. In Figure 19, the contours of light intensity with respect to the linewidth and the distance are plotted. We also pick a few real circuit patterns and compare light intensity calculated with our approximated model and the simulation [24] results in Figure 20. In

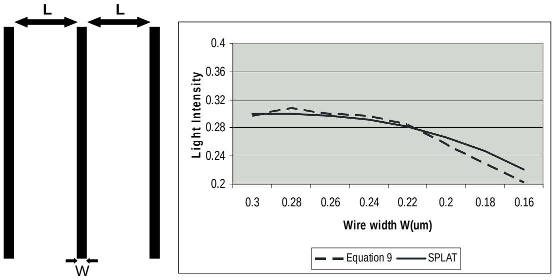
Figure 20(a), we show the test pattern we used, in Figure 20(b), we plot light intensity result from our models with those from SPLAT calculations. It can be seen that the results from the approximated models are very close to the simulation results.



(a) From lithographic simulation

(b) From our approximated model

Fig. 19. Contour plot of light intensity vs. feature linewidth and distance to feature for partially coherent illumination.



a. Test structure

b. Light intensity

Fig. 20. Comparisons of light intensity calculations.

2. Printability Function

In order to print out a sharp image, we wish the overall light intensity inside a feature (wire segment) to be full level represented by 1 while the light intensity outside the feature should be 0. Ideally, there is a sharp light intensity transition along boundary of the rectangle representing a wire segment. If the transition threshold is denoted as I_{th}, we wish $I_i(x_i, y_i) \geq I_{th}$ inside segment, $I_i(x_i, y_i) \leq I_{th}$ outside segment. Therefore, the ideal case is $I_i(x_i, y_i) = I_{th}$ when (x_i, y_i) is on the segment boundary. For a wire segment, we chop its boundary into multiple small pieces which are sufficiently small, then the light intensity on every point of a single **boundary piece** ξ can be regarded as the same I_ξ. Then the **printability function** is defined as

$$\Theta(\mathbf{w}, \mathbf{x}, \mathbf{y}) = -\sum_{\forall \xi} (I_\xi(\mathbf{w}, \mathbf{x}, \mathbf{y}) - I_{th})^2 \tag{3.10}$$

This printability model has two major differences from the model employed in [14]. First, the model in [14] is for coherent illumination while ours is for partially coherent illumination which is much closer to the practical reality. Second, the model in [14] emphasizes on interference while ours is focused on image sharpness. The work of [14] attempts to limit the interference to a target wire from its neighbor wires. However, the effect of light from the target wire itself is not considered. In practice, it is the overall effect of lights from the target wire and its neighbor wire that determines the printability of the target wire. Therefore, our model captures a more complete picture of the printability problem.

With the printability model, we are able to simplify the complicated lithography modeling and simulations and make it suitable for use in the design environment. In Figure 21, we reproduce a figure from [16] to show the relationships of light intensity and EPE (Edge Placement Error). We can see EPE is the error in x-axis caused by light intensity error in y-axis, they are highly correlated.

D. Timing Driven Wire Sizing Considering Coupling Capacitance

In phase 1 of our method, we need to find a wire sizing solution which satisfies the constraints (3.2), (3.3) and (3.4). Here we perform wire sizing without changing the center

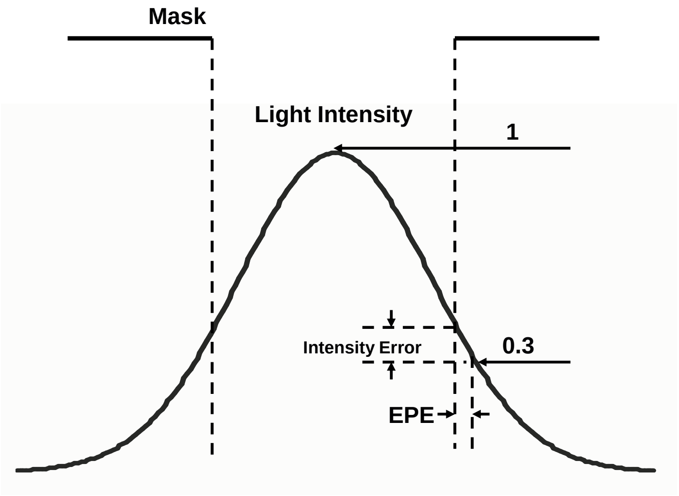


Fig. 21. Light intensity and EPE.

locations wires. Thus, only **w** are variables and the spacing constraint (3.4) can be implicitly satisfied by enforcing the width constraint (3.3). As in [21], this sub-problem can be solved through Lagrangian relaxation which is formulated as:

$$\begin{array}{lll} \text{Minimize} & \sum_{\forall s_j \in S} \mu_j (t_j(\mathbf{w}) - q_j) & \\ \text{Subject to} & L_i \leq w_i \leq U_i & \forall e_i \in E \end{array} \tag{3.11}$$

where $\{\mu_1, \mu_2, ...\}$ forms the set of Lagrangian multipliers. Alternatively, the problem in phase 1 can be formulated as

$$\begin{array}{lll} \text{Maximize} & \tau & \\ \text{Subject to} & t_j(\mathbf{w}) + \tau \leq q_j & \forall s_j \in S \\ & L_i \leq w_i \leq U_i & \forall e_i \in E \end{array}$$

where τ indicates the minimum timing slack. This minimum slack maximization problem can also be solved through Lagrangian relaxation as

$$\begin{array}{lll} \text{Minimize} & -\tau + \sum_{\forall s_j \in S} \mu_j (t_j(\mathbf{w}) + \tau - q_j) & \\ \text{Subject to} & L_i \leq w_i \leq U_i & \forall e_i \in E \end{array} \tag{3.12}$$

We can see that the Lagrangian problems (3.11) and (3.12) are very similar with each other and can be solved in a similar method. The value of the Lagrangian multipliers can be found using the sub-gradient method as in [21]. For each set of fixed Lagrangian multipliers, the problem (3.11) is equivalent to minimizing a weighted sum of sink delays [22]. However, the algorithm of [22] is for discrete wire sizing and the work of [21] does not consider coupling capacitance. Therefore, we introduce a continuous wire sizing algorithm considering coupling capacitance as follows.

First, we describe the expression of delay function $t_j(\mathbf{w})$ for a sink $s_j \in S$. For an edge e_i, its length is l_i and its width is w_i. Its edge capacitance is $C_{e,i}$ and its downstream capacitance is $C_{L,i}$. The wire resistance coefficient is denoted as r. Consider a sink node s_j in a Steiner tree T_k with source node at s^k, driver resistance R_k and total load capacitance $C_{total,k}$. Then, the delay t_j to sink s_j can be expressed as

$$t_j = R_k C_{total,k} + \sum_{\forall e_i \in path(s^k, s_j)} \frac{r l_i}{w_i} \left(\frac{C_{e,i}}{2} + C_{L,i}\right) \tag{3.13}$$

The area, fringing and coupling capacitance coefficient are represented as c_a, c_f and c_x, respectively. The set of edges adjacent with e_i is denoted as $Adj(e_i)$. The wire pitch between two edges e_h and e_i is P_{hi}. Then, the edge capacitance $C_{e,i}$ can be obtained as:

$$C_{e,i} = c_a l_i w_i + c_f l_i + \sum_{e_h \in Adj(e_i)} \frac{2 c_x l_i}{2P_{hi} - w_h - w_i}$$

The objective function (3.11) can be decomposed as:

$$F(\mathbf{w}) = \sum_{\forall s_j \in S} \mu_j(t_j(\mathbf{w}) - q_j) = J(\mathbf{w}) + U(\mathbf{w}) + E$$

where E is a constant. Function $J(\mathbf{w})$ is the coupling related part and $U(\mathbf{w})$ is similar as the delay function in [21] where coupling capacitance is not considered. They can be expressed as:

$$\begin{aligned} J(\mathbf{w}) &= \sum_{\forall e_i \in E} \sum_{e_h \in Adj(e_i)} \frac{2c_x l_i}{(2P_{hi} - w_h - w_i)} \\ & [\frac{rl_i}{w_i} \sum_{s_b \in Des(e_i)} \mu_b + \sum_{e_h, e_i \in Des(e_j)} \frac{rl_j}{w_j} \sum_{s_b \in Des(e_j)} \mu_b \\ & + \sum_{e_h, e_i \in Des(R_k)} R_k \sum_{s_b \in Des(R_k)} \mu_b] \\ U(\mathbf{w}) &= \sum_{e_i \in E} \frac{\alpha_i}{w_i} + \sum_{e_i \in E} \beta_i w_i + \sum_{e_i, e_j \in E} \gamma_{ij} \frac{w_i}{w_j} \end{aligned}$$

where $Des(e)$ indicates the set of descendant edges of e, α_i, β_i and γ_{ij} are positive constants.

According to [25], $U(\mathbf{w})$ is a unary posynomial and therefore can be optimized through convex programming. In order to optimize $F(\mathbf{w})$, the convexity of $J(\mathbf{w})$ needs to be investigated as well.

Theorem 1: *Function $J(\mathbf{w})$ is convex.*

Proof: It can be observed that $J(\mathbf{w})$ is a positively linear combination of the following base functions:

$$\begin{aligned} \phi(x,y,z) &= \frac{1}{z(D-x-y)} \\ \psi(x,y) &= \frac{1}{x(D-x-y)} \\ \zeta(x,y) &= \frac{1}{D-x-y} \end{aligned}$$

where D is a positive constant and $x > 0, y > 0, z > 0, D - x - y > 0$. Therefore, $J(\mathbf{w})$ is convex if all of the above base functions are convex.

Let $A = \frac{1}{z}$ and $B = \frac{1}{D-x-y}$, then the Hessian matrix for $\phi(x,y,z)$ can be derived as:

$$\mathbf{H} = \begin{bmatrix} 2A^3B & -A^2B^2 & -A^2B^2 \\ -A^2B^2 & 2AB^3 & 2AB^3 \\ -A^2B^2 & 2AB^3 & 2AB^3 \end{bmatrix}$$

For an arbitrary vector $\mathbf{v} = (a,b,c)^t$, we can obtain:

$$\mathbf{v^tHv} = 2AB[a^2A^2 + (b+c)^2B^2] \geq 0$$

Hence, $\phi(x,y,z)$ is a convex function. Similarly, it is straightforward to show that $\psi(x,y) = \frac{1}{x(D-x-y)}$ and $\zeta(x,y) = \frac{1}{D-x-y}$ are also convex functions. $\square$

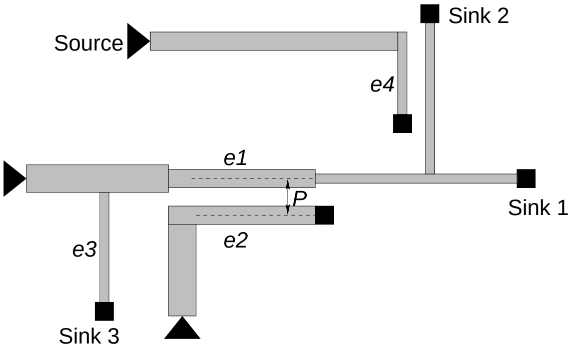


Fig. 22. Illustration on relative positions among nodes and edges.

Since $J(\mathbf{w})$ is convex and $U(\mathbf{w})$ is a posynomial, there is a unique global minimum solution for $F(\mathbf{w})$ [25]. Therefore, the global minimum solution can be reached through iterative local optimization as in [25]. In each local optimization, wire sizing is performed for only one edge e_i while the widths of the other edges are fixed. We use $e_i \bowtie s_j$ to denote that edge e_i and sink s_j are in the same Steiner tree. Thus, the delay function $F(\mathbf{w})$ in the local optimization becomes

$$\begin{aligned} f(w_i) &= \sum_{s_j \in Des(e_i)} \frac{\alpha_{ij}}{w_i} + \sum_{e_i \bowtie s_j} \beta_{ij} w_i \\ &\quad + \sum_{s_j \in Des(e_i), e_k \in Adj(e_i)} \frac{\eta_{ij}}{w_i(Q_{ik} - w_i)} \\ &\quad + \sum_{s_j \bowtie e_i, s_j \bowtie e_k, e_k \in Adj(e_i)} \frac{\rho_{ij}}{Q_{ik} - w_i} \end{aligned}$$

where η_{ij}, ρ_{ij} and Q_{ik} are positive constants. In the above equation, the first term represents the impact of wire resistance of e_i on its descendant nodes. For example, the effect of e_1 on sink 1 and sink 2 in Figure 22. The second term reflects the effect of wire e_i self capacitance on sink nodes in the same tree, as e_3 for sink 1, 2 and 3 in Figure 22. The third term represents the product of wire resistance of e_i and coupling capacitance of e_i and the effect on its descendant nodes. For the example in Figure 22, the wire resistance of e_1 and its coupling capacitance with e_2 affect the delay at sink 1 and 2. The last term shows that the coupling capacitance due to wire e_i presents a capacitive load to its Steiner tree and the Steiner tree it is coupled with. For example, the width of edge e_4 affects the delay of sink 1, 2 and 3 in Figure 22. Figure 22 also shows that a wire segment could be part of the Steiner tree edge. Here, we define the portion of the Steiner tree as a wire segment if the coupling remains the same across the whole segment. For example, e_1 is a segment, but the rest of the wire toward sink 1 is another segment. This way, we are much more flexible for optimization since a single physical wire can have different width across it.

Lemma 1 *Function $f(w_i)$ is convex.*

Proof: It can be shown that $\frac{df(w_i)^2}{d^2 w_i} > 0$. □

If there is value $\tilde{w}_i$ satisfying $\frac{df(w_i)}{dw_i} = 0$, then $f(w_i)$ has a unique minimum solution at $w_i^* = \min\{U_i, \max\{L_i, \tilde{w}_i\}\}$. Please note that $\frac{df(w_i)}{dw_i} = 0$ is a fourth order equation which has closed form solutions. Based on this local optimization, the algorithm of minimizing $F(\mathbf{w})$ subject to wire width constraints is shown in Figure 23.

```
Coupling aware timing driven wire sizing
1. Initialize $(w_1, w_2, ...)$ with $(L_1, L_2, ...)$
2. Do {
3.    For each edge $e_i$
4.       $w_i = \min\{U_i, \max\{L_i, \tilde{w}_i\}\}$
5. } while $F(\mathbf{w})$ is improved
```

Fig. 23. Timing driven wire sizing algorithm.

Theorem 2: *The timing driven wire sizing algorithm can converge to the optimal solution and the complexity of the algorithm is* $O(n)$*, with n being the number of wire segments.*

Proof: The proof is similar as the proof of Lemma 2 and Lemma 3 in [25] and is omitted here. □

E. Printability Optimization

The feasible solution obtained in phase 1 is fed to phase 2 in which the printability $\Theta(\mathbf{w},\mathbf{x},\mathbf{y})$ is maximized by adjusting wire width and spacing. The feasibility in terms of (3.2), (3.3) and (3.4) is maintained, which ensures that there is no violation on timing and wire width/spacing rules.

Equation (3.9) and (3.10), indicate that the printability function is an eighth order polynomial in general, although this is an approximated model. We employ a sensitivity based heuristic similar as in [18] to solve this complicated problem. The sensitivity of the printability function for a certain wire *i* can be obtained as:

$$\psi_{i,w}(\mathbf{w},\mathbf{x},\mathbf{y}) = \frac{\Theta(\mathbf{w_i},\mathbf{x},\mathbf{y}) - \Theta(\mathbf{w},\mathbf{x},\mathbf{y})}{\delta} \tag{3.14}$$

$$\psi_{i,xy}(\mathbf{w},\mathbf{x},\mathbf{y}) = \frac{\Theta(\mathbf{w},\mathbf{x_i},\mathbf{y}) - \Theta(\mathbf{w},\mathbf{x},\mathbf{y})}{\delta} \tag{3.15}$$

where $\psi_{i,w}$ is the sensitivity with respect to the wire width w_i, $\psi_{i,xy}$ is the sensitivity with respect to the center location of the wire. Here, we use vertical wires as an example, $\mathbf{w_i} = (w_1, w_2, ..., w_i + \delta, ...w_n)^t$, $\mathbf{x_i} = (x_1, x_2, ..., x_i + \delta, ...x_n)^t$. Horizontal wire segments can be handled in a similar way. By changing wire width and center location of the wire, we adjust both the width of the wire and space of the wire to its neighboring wires to improve the printability function while the wire size and spacing rules are satisfied. Also, we do incremental timing analysis to make sure the required arrival time stays accurate.

Even though the printability function $\Theta(\mathbf{w}, \mathbf{x}, \mathbf{y})$ is based on the light intensity of every wire segment in the entire layout, the computation of the sensitivity $\psi_{i,w}(\mathbf{w}, \mathbf{x}, \mathbf{y})$ and $\psi_{i,xy}(\mathbf{w}, \mathbf{x}, \mathbf{y})$ can be limited to a geometrically local region. This is because a change on wire width or wire space affects light intensity of only a local region close to that wire segment. Usually, the effect of a change decays to negligible level at a location more than 2λ (λ is the lithographic wavelength) away from the location of the change. Hence, we generate a window by expanding the wire segment i by 2λ on each of its four sides. The computation of sensitivity $\psi_{i,w}(\mathbf{w}, \mathbf{x}, \mathbf{y})$ and $\psi_{i,xy}(\mathbf{w}, \mathbf{x}, \mathbf{y})$ can be limited within this window. After the sensitivity for every wire is obtained, the wire with the maximal value of $\psi_i(\mathbf{w}, \mathbf{x}, \mathbf{y}) = max(|\psi_{i,w}(\mathbf{w}, \mathbf{x}, \mathbf{y})|, |\psi_{i,xy}(\mathbf{w}, \mathbf{x}, \mathbf{y})|)$ is selected to be changed. The pseudo code of the printability optimization is outlined in Figure 24.

When we attempt to tune the width or center location of a wire, the constraints on wire size and wire spacing are enforced. In addition, we need to ensure that there is no timing violations due to this change. Even though there is analytical formula for sink delay $t_j(w_i)$ of a sink $s_j \in S$ with respect to edge $e_i \in E$, the delay constraint function sometimes is equivalent to a third order polynomial of w_i and the resultant feasible range for w_i is not necessarily continuous. Therefore, we just check the timing feasibility of each change instead of finding an analytical bound for a change. If a change of wire width and/or space causes any delay constraint violation, this change is forbidden. The pseudo code for the

Printability optimization

```
Input: $\mathbf{w} = (\overline{w}_1, \overline{w}_2, \ldots)$,
       $\mathbf{x} = (\overline{x}_1, \overline{x}_2, \ldots)$ for vertical wires
       $\mathbf{y} = (\overline{y}_1, \overline{y}_2, \ldots)$ for horizontal wires
       such that $t_j \leq q_j \forall s_j \in S$,
       wire width and spacing rules are satisfied

Do {
    $e_j \leftarrow NULL$
    For each edge $e_i$
        Let $\varphi_i$ be $w_i$, $x_i$ or $y_i$ that maximizes $\psi_i(\mathbf{w}, \mathbf{x}, \mathbf{y})$
        If $\varphi_i \leftarrow \varphi_i + sign(\psi_i(\mathbf{w}, \mathbf{x}, \mathbf{y})) \cdot \delta$ satisfies
                wire width, spacing and timing constraints
            $e_j \leftarrow e_i$
    If $e_j \neq NULL$
        change $\varphi_j$ by $sign(\psi_j(\mathbf{w}, \mathbf{x}, \mathbf{y})) \cdot \delta$
} while $\Theta(\mathbf{w}, \mathbf{x}, \mathbf{y})$ is improved
```

Fig. 24. Printability optimization heuristic.

sensitivity based heuristic is shown in Figure 24. We change every wire segment at most once during the optimization. After each change, we need to update the timing for the path that contains this target segment, as well as timing of the paths that contain any wire segment that couples into the target segment. Let P be the maximum number of wire segments that couples into any given wire segment in the design, G be the maximum number of wire segments in any given timing paths in the design, the complexity for printability optimization is $O(PGn)$. Typical value of P would be 2 as we only consider closest neighboring wire segment of the target segment. Also G is typically much less than the total number of wire segment n.

F. Experimental Results

Our method is implemented in C++ and the experiment is performed on a SUN Sparc Ultra-80 workstation with four 450MHz CPU and 4Gb RAM. Table III shows the number of nets, horizontal wires and vertical wires for the benchmark circuits. Based on 90*nm* technology, the wire width is allowed in a range between 100*nm* and 300*nm* while the wire pitch is 400*nm*. The light intensity threshold I_{th} at wire segment boundary is chosen as 0.3, since this is usually where light intensity slope is the greatest. Here, light intensity is relative intensity. Light intensity at the wafer when there is no mask between light source and the wafer is defined as 1.

Since there is no previous work on this timing and printability optimization problem, we list the results of the 2 phases of our method in Table IV. The number of sinks with timing violations for each case is in column 2. After phase 1, all timing violations are eliminated as indicated in column 3. The printability function Θ values after phase 1 are shown in column 4. The printability after phase 2 and the percentage improvement are in column 5 and column 6, respectively. It can be seen that our sensitivity based heuristic can yield

Table III. Benchmark circuit specification.

Circuit	#nets	# hori edges	# vert edges
apte	73	205	168
hp	66	248	208
ami33	107	412	341
xerox	154	651	548
ac3	193	712	564
ami49	297	1016	844
hc7	365	1746	1430
a9c3	664	2848	2450
xc5	919	3586	2911
playout	958	3777	3121

4% to 22% improvement on the printability function Θ without timing violation. Meanwhile, all wire width/spacing rules are also satisfied. The runtime information is shown in the rightmost column. This computation speed is reasonable for practical applications.

In order to validate our approach with a precise model, lithographic simulation [24] is performed on the layout result of Phase 1 and Phase 2. SPLAT is a simulation program to model projection printed images using a two-dimensional optical image. Developed in UC Berkeley, it is one of the first tools in this field. One of the most important and obvious metric for design printability is the average EPE (Edge Placement Error) [16]. EPE is the distance between a printed edge to the location where this edge is intended to be in the design. Smaller EPE means that the printed edge is closer to its intended location. Designs with smaller EPE will be easier for OPC/RET in the manufacturing stage to further reduce EPE to spec. We compare the average EPE for the designs before and after printability optimization. The simulations for wires on Metal 1 and Metal 2 are conducted separately

Table IV. Experimental results. # vio is the number of delay violations. Θ is the printability. # vio remains 0 in phase 2.

	Input	Phase 1		Phase 2		Total
Circuit	# vio	#vio	Θ	Θ	improv	CPU(s)
apte	11	0	-245.4	-217.8	11.2%	14.7
hp	17	0	-264.5	-239.7	9.4%	11.4
ami33	21	0	-415.5	-397.5	4.3%	58.6
xerox	70	0	-868.1	-833.8	3.9%	98.7
ac3	36	0	-777.3	-723.2	7.0%	113.34
ami49	39	0	-1123.4	-1034.1	7.9%	90.2
hc7	65	0	-1886.0	-1738.3	7.8%	236.3
a9c3	163	0	-3097.5	-2878.1	7.1%	328.1
xc5	166	0	-3971.2	-3746.2	5.7%	824.0
playout	131	0	-5900.2	-4584.2	22.3%	1462.9

and the results are summarized in Table V. These data show that our method can improve the average EPE by about 20%-40%. Considering that no OPC has been applied yet, such improvement is quite significant.

G. Conclusion

Lithography friendly design is a concept that aims to reduce litho process complexity so that volume production of the design can be achieved faster in the fab while reducing mask cost. In this chapter, we propose an approach that adjusts wire size and space to optimize layout printability while maintaining design performance. A new printability model is proposed to handle partially coherent illuminations. The complicated printability optimization problem is solved in a 2-phase heuristic. A coupling aware timing driven continuous wire

Table V. Lithographic simulation results on average EPE.

Circuit	Metal 1 (hori)			Metal 2 (vert)		
	Phase 1 Ave EPE (nm)	Phase 2 Ave EPE (nm)	Phase 2 improv	Phase 1 Ave EPE (nm)	Phase 2 Ave EPE (nm)	Phase 2 improv
apte	4.35	2.87	34.0%	5.04	3.57	29.2%
hp	4.20	3.04	27.6%	5.64	4.27	24.3%
ami33	3.97	2.67	32.7%	4.35	2.91	33.1%
xerox	3.98	2.33	41.4%	4.65	3.04	34.6%
ac3	3.31	2.34	29.3%	4.12	2.73	33.7%
ami49	3.72	2.65	28.8%	4.14	2.78	32.8%
hc7	4.10	2.98	27.3%	4.65	3.04	34.6%
a9c3	3.99	2.89	27.6%	4.15	2.98	28.2%
xc5	3.79	3.01	20.6%	4.19	3.02	27.9%
playout	3.70	2.80	33.3%	4.08	3.01	26.2%

sizing algorithm is also introduced. Experimental results from lithographic simulations confirm the effectiveness of our method.

CHAPTER IV

ASIC DESIGN FLOW CONSIDERING LITHOGRAPHY INDUCED EFFECTS

In deep submicron VLSI designs, both timing and power performance of integrated circuits are increasingly affected by process variations. In practice, people often treat systematic components of the variations, which are generally traceable according to process models, in the same way as random variations in process corner based methodologies. In particular, lithography induced process variations are usually estimated by a universal worst case value without considering their layout environment. Consequently, the process corner models based on such estimation are unnecessarily pessimistic. In this chapter, we propose a new ASIC design methodology which captures lithography induced polysilicon gate length variations including both the layout dependent systematic components and random components. Our methodology also shows that look-up table methodology is sufficient to handle BEOL (Back End Of Line) lithography process variations in timing analysis. In addition, a new technique of dummy poly insertion is suggested to shield inter-cell optical interferences. This technique together with standard cells characterized using our methodology will let current design flows comprehend the variations almost without any changes. More importantly, by separating systematic lithography effect from random process variations, our methodology greatly reduces pessimism in timing analysis, thus enables both aggressive design implementation and easier timing signoff. Experimental results on industrial designs indicate that our methodology can averagely reduce timing variation window by 11%, power variation window by 55% when compared to a worst case approach.

A. Introduction

The main contributions in this work are as follows.

- A new technique of dummy poly insertion is suggested to handle the dependence of gate length variations on inter-cell spacing. In general, the gate length variations in the boundary regions of a cell depends on its spacing with outskirt poly of neighboring cells. However, the neighboring cell information is not available before cell placement is completed. Our dummy poly insertion technique can generally avoid such dependence on unknown information. Therefore, we do not need to characterize different variants of a cell as in [31]. Moreover, the pattern matching based variant selection [31] is not necessary any more in later design stages. In other words, the lithography aware cell characteristics can be utilized without affecting current standard cell based design flows.

- We utilize a poly segmentation technique to accurately evaluate arbitrarily irregular poly shapes. The printed poly shapes on wafer not only have deviations on gate length, but also have *different* deviations on different spots of a same poly. In other words, the deviations are by and large non-uniform. Our approach is in contrast to many previous works which treat the deviation of a poly uniformly.

- We verified that for lithography effect on wire width of interconnect, traditional lookup table method is still valid.

Accurate lithography simulations, including OPC, photoresist and etch simulations, are timing consuming in general. For practical design use in timing analysis, it is thus very difficult to run lithography simulations on the full chip level. However, the size of a library cell is very small and the characterization is usually performed only once. Therefore, the expensive lithography/OPC simulations are affordable in this scenario. More importantly, cell-based timing annotation methodology is used in typical timing analysis for standard cell based ASIC design. In order not to disturb this flow, it is therefore, very desirable for a methodology to capture lithography effect in a cell-based fashion. Traditional ASIC

STA (Static Timing Analysis) flow utilizes process corner conditions for timing signoff. This approach introduced high level of pessimism. For example, in slow process corner, all transistor gates are assumed to have the largest gate lengths. In reality, that will never happen. Because of the systematic nature for lithography effects, our methodology predicts this portion of the variations and take it into consideration for STA to reduce pessimism significantly. We applied our methodology to industrial library cell designs. The experimental results indicate that our methodology can averagely reduce timing variation window by 11%, power variation window by 55% when compared to an existing approach.

B. Overview of Methodology

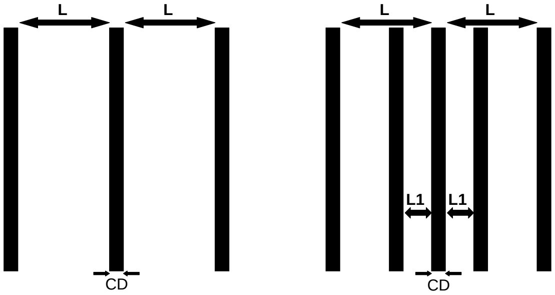
1. FEOL Lithography Variations

a. Standard Cell Architecture

It is claimed in [30] that litho simulation on individual standard cell does not represent accurate lithographic effect for that standard cell in block level designs because lithographic effect depends on proximity of that standard cell. However, increasing the distance between one shape and other shapes will reduce the impact of the lithographic effect of other shapes tremendously. It is also very important to notice that the closest neighbors of a shape are the dominating factors to model based OPC process and Sub-Resolution Assist Features (SRAFs) generation for a particular shape.

We ran Mentor Graphics Calibre LFD on two sets of test structures shown in Figure 25. The first set of the structures have three shapes with distance L between them. The second set of structures have five shapes where the distance between the middle shape and the shapes on both sides is also L. The shapes next to the middle shape are $L1$ away from it. In both sets, L is changing and $L1$ in second set is fixed. The CD (Critical Dimension) data for the middle shape is recorded from our lithography simulation tool.

The result is shown in Figure 26. We can see that the CD of test structure 2 has much less variations than that of test structure 1, i.e., having neighboring shapes with a fixed distance really helps reducing printability variations of that shape. Also, other test structures we ran showed that a shape has minimal impact on another shape's printing image if there are two or more shapes between them.



a. Test structure 1: variable space L.

b. Test structure 2: variable space L with fixed space L1.

Fig. 25. Test structures for CD variations through pitch.

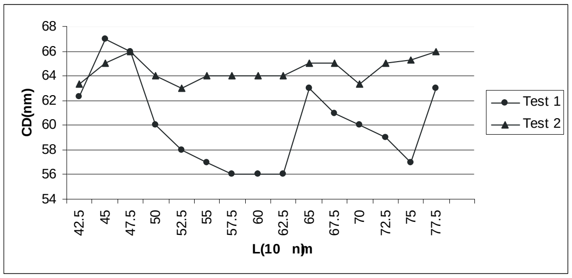


Fig. 26. Critical dimensions of the two test structures when *L* changes.

The range of *L* and *L*1 that we chose to exercise the test structures is hardly random. In our 65nm standard cell library implementation, *L* is the range of the possible poly gate spacing if two standard cells are placed adjacent to each other. If the two neighbor cells have gate space larger than *L*, a filler cell with a dummy poly will be inserted between these two cells for Design Rule Check (DRC) and power connectivity purposes. For the first set of test structures, we observe over 10% of CD variability over the range of *L*. However, in our standard cell library architecture, we can put a dummy poly shape at the cell border without introducing any area penalty. In this case, the value of *L*1 in our test structures represents the minimum spacing between the dummy poly and active transistors in the standard cell. When two cells are placed side by side, the dummy poly shapes of both cells overlap exactly (see Figure 27). We also would like to point out that the dummy poly shapes we insert are field poly, i.e., they do not form new devices as they fall in the gap of the diffusions between two closely placed standard cells. Thus, these shapes do not cause extra LVS verification efforts. Our standard cell designs ensure DRC of the dummy poly lines as the gap of the diffusion is large enough. By adding dummy poly shapes into the original standard cells, we introduce fixed closest neighbors to the poly gates that are at the cell boundary, thus greatly reduce the CD variations introduced by various proximity of this standard cell in the design as shown in Figure 26. In fact, they effectively “shield” all the internal transistors from lithographic effects of neighboring structures.

b. Calculation of Effective Transistor with Litho Effects

In lithography process, there are several contributors of the poly gate length variations (see Figure 28). One of them is the variation caused by poly pitch to neighbors. The other is the L-shaped poly cornering effect, which is particularly important for small transistors. This effect is clearly shown in the middle transistor in Figure 28. To obtain an accurate prediction of the transistor behavior, we need to account for multiple sources of poly gate length variations.

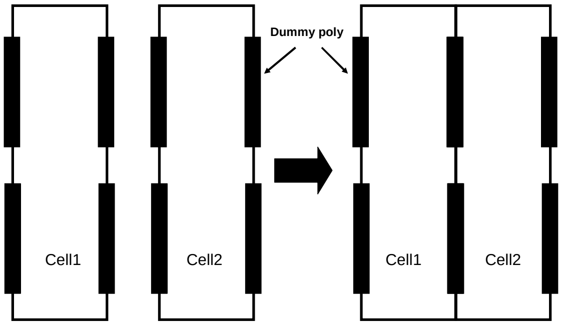


Fig. 27. Dummy poly of two neighboring cells can be merged.

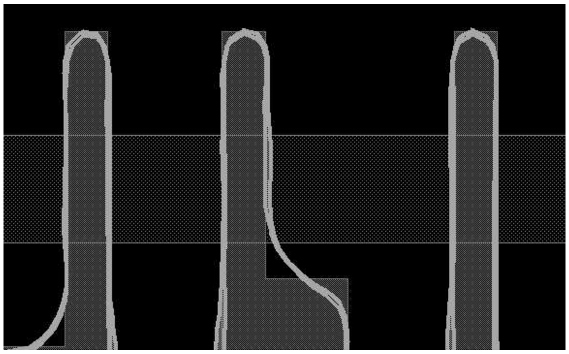


Fig. 28. Systematic poly gate CD variations from lithography.

The lithography induced deviations of the critical dimension, which is the poly gate length, are usually in the order of a few nanometers. For other larger shapes, the relative shape deviation is much smaller. In a typical 65nm design, poly gate width and diffusion dimensions are at least 2 or 3 times of the gate length. That means that lithography induced circuit performance variations are mostly due to gate length deviation. It is therefore sufficient to extract the lithography information for only poly gate length. The printed image of gate poly shapes across process window will be employed to replace gate length image offset parameters introduced by traditional process corner models. We keep all the other process corner parameters unchanged, such as threshold voltage variation, gate oxide variation, etc.

After lithography/OPC simulation, we have an estimation of the printed images of poly shapes. Ideally, we wish to run SPICE simulations to obtain timing and leakage power profiles of the cell. However, current device models in SPICE can handle only rectangular shaped transistors while the lithography/OPC simulation results are often irregular shapes. For example, a gate length is relatively large at the location of a jog, but is small right before reaching that jogging region. In order to solve this mismatch, we try to compute an effective gate length which may provide the same timing/power performance of a post lithography/OPC simulation poly shape. In general, the on current I_{on} of a transistor determines the timing performance of this transistor. The leakage power of a transistor is mostly dependent on the off current I_{off} of the transistor. Since on and off currents of a transistor usually have different sensitivities to gate length variations, we need to use different effective gate length for timing and leakage.

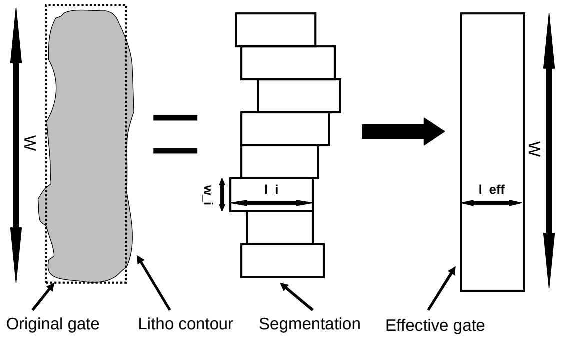


Fig. 29. Calculation of effective gate length for timing and leakage.

We utilize a segmentation technique to compute the effective gate length for timing and leakage. First, we construct two lookup tables for transistor I_{on} and I_{off}. For both tables, each row corresponds to a specific transistor gate width and the columns are for different transistor gate length. Each entry of the table represents I_{on} or I_{off} of a transistor with gate width and length specified by the row and column indices. The ranges of transistor width and gate length, i.e., the ranges of row and column indices, are based on typical transistor sizes allowed in fabrication. The values of I_{on} and I_{off} are obtained through SPICE simulations.

Next, we chop a poly shape from lithography/OPC simulation into multiple segments and each segment can be approximated by a rectangle. This is illustrated in Figure 29. The I_{on} and I_{off} of each small segment can be obtained from a simple calculation based on the lookup tables. Please note that the width of a segment is usually much smaller than fabrication allowed size. Thus, it cannot be matched to any row index in the lookup tables. We suggest to solve this discrepancy through scaling. For example, consider a transistor with nominal gate length $65nm$ and width $200nm$. We chop its gate poly shape from lithography/OPC simulation into 10 segments. Thus, each segment i has a length l_i and width of $20nm$. Then, we can find the on current $I_{on}(l_i, 200nm)$ from the lookup tables. The on current of this segment can be approximated as $I_{on}(l_i, 20nm) = I_{on}(l_i, 200nm)/10$. The off current $I_{off}(l_i, 20nm)$ can be calculated in the same way.

Once the on and off currents of all segments are available, the overall currents of the entire transistor based on the lithography/OPC simulated poly shape can be calculated as:

$$I_{on,shape} = \sum_{i=1}^{n} I_{on}(l_i, w) \tag{4.1}$$

$$I_{off,shape} = \sum_{i=1}^{n} I_{off}(l_i, w) \tag{4.2}$$

where n is the number of segments and w is the width of each segment.

Last, the effective gate length $L_{eff,timing}$ can be found based on the estimated $I_{on,shape}$

and the lookup table of on current. From the nominal transistor width, we can find its corresponding row in the lookup table. We search for the entry in the row with value closest to the $I_{on,shape}$ obtained above. The column index for this entry is the $L_{eff,timing}$ for this transistor. The effective length $L_{eff,leakage}$ can be obtained in the same way based on $I_{off,shape}$ and the lookup table for off current. Similar technique has been proposed in [32] with detailed analysis of this modeling method.

c. Netlist Back Annotation and Standard Cell Characterization

After we calculate the effective gate length $L_{eff,timing}$ and $L_{eff,leakage}$ of each poly gate shape, we need to back annotate the standard cell SPICE netlist with these effective gate lengths. The layout of a transistor may consist of multiple fingers and lithography usually has different effects on each finger depending on the layout environment. Therefore, we need treat these fingers separately even though they belong to the same transistor. When an LVS (Layout Versus Schematic) tool runs in its normal mode, it automatically merges multiple fingers of a transistor into a single gate. To avoid this merging, we perform a special LVS that takes the x and y coordinates of each poly gate shape into a layout netlist even when some poly gate shapes are the fingers of the same transistor. The layout netlist will be fed into our extraction tool to generate a netlist with parasitics. The extracted netlist also keeps each poly gate shape as a separated device. We then use the x and y coordinates to match the poly shape in the extracted netlist with poly shape contours from lithography/OPC simulations. We back annotate the $L_{eff,timing}$ and $L_{eff,leakage}$ into the extracted netlist. Thus, for each standard cell, we generate one netlist for timing simulation and another netlist for leakage power simulation. By including dose and focus variations in the lithography/OPC simulations, we can have the extracted netlist for each cell at the worst and the best process corners.

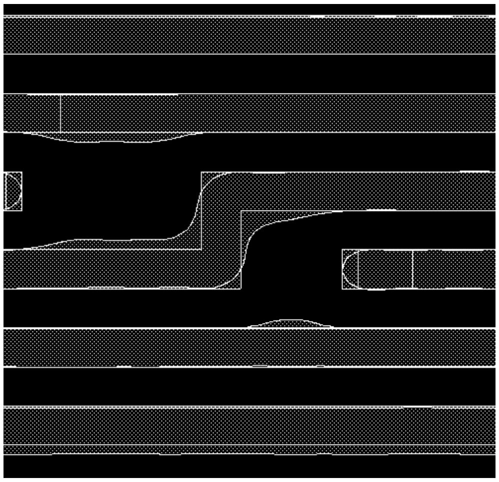


Fig. 30. Litho contours for routing metal.

2. BEOL Lithography Variations

For interconnections, lithography also introduces variations, mainly on the width of the wire. In Figure 30, we show the contour of the metal wire on top the intended drawn shape. The distortion of the routing metal also introduces timing variations. Therefore, we will also need to consider these effects for STA. However, it is obviously true that metal shape distortions do not impact timing in a noticeable way locally in an individual standard cell because of the small size of the cells. For interconnection, signal routing is often done in a grid based fashion, meaning most portion of the wire will have stable and predictable environment. It is interesting to notice that even though part of the interconnection will be distorted by lithography process significantly, the wire stays close to the drawn shape across most of the its length. In this section, we will verify that traditional parasitic extraction methodology with a lookup table still applies for STA analysis considering lithography effects. We compare the STA results from two different approaches, one with full lithography simulation on BEOL interconnection wire width and one using look-up table extraction.

a. Lithography Simulation On BEOL

After performing lithography simulation on BEOL, we obtain the contour of the wire. However, the contour is presented by a polygon with a large number of vertexes. It would be extremely difficult for the extraction tool to handle these kinds of polygons. To solve this problem, we convert these polygons with smoother ones. This process is described as follows.

- We convert the polygon to a Manhattan polygon whose segments are either vertical or horizontal.
- We define a threshold value for wire width change between adjacent wire segments. If the change for adjacent edges is smaller, we merge the two segments into one with the average wire width for the segment, as shown in Figure 31.
- We use the bounding box of the VIA shapes for VIA area.

b. STA Comparison

Two separate STAs are performed, one use drawn shape routing metal for extraction, the other use polygons converted from litho contours of the routing metals. We verified the timing difference is very minor. The results are presented in the experiment section.

3. Litho-aware STA Flow

Based on our analysis of FEOL and BEOL litho effects, we conclude that we should focus most of the efforts on poly gate litho effects of standard cells for litho-aware timing analysis flow. With accurate litho simulation and device characterization, we are able to extract

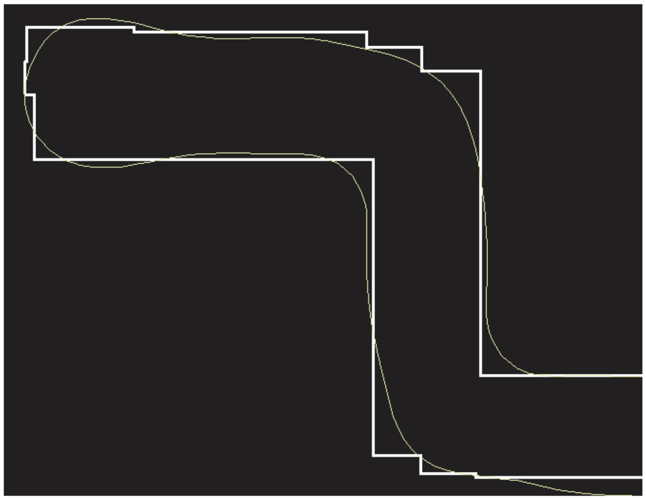


Fig. 31. Convert litho contour to Manhattan polygon.

timing impact of litho process for a given standard cell. At the same time, with an effective technique such as dummy poly insertion, we are able to minimize timing variation caused by standard cell context for a placed design. For BEOL routing metals, we show that we can continue to use current parasitic extraction methodology. In summary, the litho-aware timing analysis flow will reduce the pessimism of the traditional corner based signoff methodology. At the same time, the evolution to the litho-aware timing analysis flow from the current STA flow mostly happens in the standard cell development and characterization. The costly litho simulations can be avoided in the block and chip level.

C. Experiment

1. Standard Cells

We follow the flow shown in Figure 32 for standard cell characterization. From the original standard cell, we first insert dummy poly on the boundary of the standard cells, we stream out the gds to feed into our litho simulation tool. We then use the result of the litho simulation to generate the new netlist with the lookup table for timing and leakage. Last we run the standard cell characterization with our standard flow and tools.

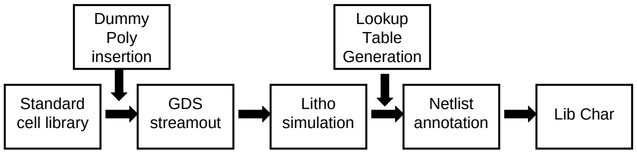


Fig. 32. Litho aware standard cell characterization flow.

We run litho simulation with Mentor Graphics Calibre LFD on our original standard cell library layout. Since our litho simulator provides the printed images of transistor poly gates across process window, we can calculate the longest and shortest effective gate length for each gate. With our original standard cell netlist at the worst timing corner, which has the worst *RC* parasitic extraction, we change the length of each gate to the longest L_{eff} of that specific gate, this gives us the annotated cell netlist at the worst timing corner. We do the same for original cell netlist at best timing corner except that we use the shortest L_{eff} of each gate to replace the original gate length in the netlist and we get the annotated cell netlist at the best timing corner. We repeat the process for leakage corners and we get the annotated cell netlists at best leakage corner and worst leakage corner.

In Figure 33 and Figure 34, we show the distributions of lithography induced gate length deviation for both best and worst timing corners for all our library cells. The x-axises in Figure 33 and Figure 34 are the difference between L_{eff} of that gate and the draw length, the y axises are the number of poly transistor gates with the specified gate length deviation. We show that gate length variations can be as much as 16*nm* across process window. Although the data reveals the level of immaturity for current 65nm Resolution

Enhancement Techniques (RET) including OPC and SRAF generation, it further confirms the value of our methodology which considers lithography induced gate length variations in a systematic fashion.

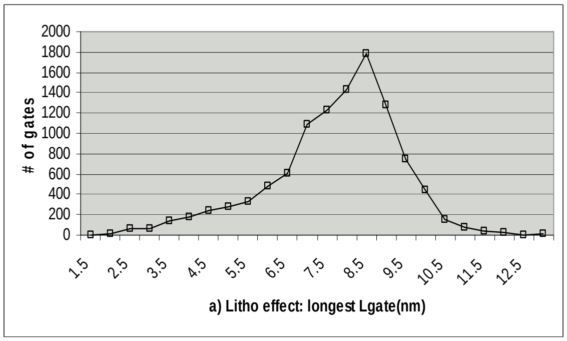


Fig. 33. Lithography induced Lgate deviation distribution: worst corner.

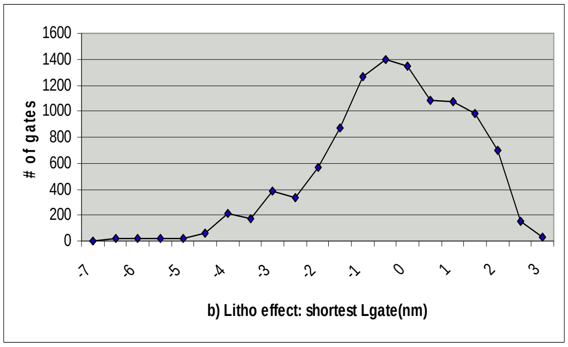


Fig. 34. Lithography induced Lgate deviation distribution: best corner.

All the generated netlists have been characterized with our standard cell characterization flow. We present the timing and leakage variabilities of a set of representative standard cells in Table 1. In column 2 and 3, we report the timing variation between two timing signoff corners with original standard cell netlist and with our new netlist, the percentage change is reported in column 4. In column 5 and 6, we report the leakage ratio between two leakage analysis corners with original standard cell netlist and with our new netlist. All data are presented with the an input slew of 180ps and output load of 4.7 ff. We see an average of 11% decrease for the variabilities of delay. As we performed our litho simulation across process window, through exposure dose and depth of focus rather than at a normal process condition, we think that is a significant source of the variability. However, we strongly believe that litho simulation through process window is absolutely necessary in order to capture lithography effect properly in process corner based design flow. The leakage analysis shows that the new netlist of the standard cells have far less variability for leakage, with the average ratio less than half of that of the original netlist. We would also want to point out that because transistor leakage is exponational to the transistor gate length, doing leakage calculation with this methodology can help identify lithography sensitive design patterns for leakage and thus help improve standard cell design robustness in terms of leakage variability. With the improvement of OPC and SRAF generation from the foundry, we believe we will see better design variability control for both timing and leakage.

2. Design Implementation

We apply our newly characterized standard cell library to one of the low power, high speed hard macros of our 65nm designs for timing analysis. We use our standard timing signoff flow for this analysis, with the same timing constraints as the original design.

Figure 35 shows the timing variability reduction for the 400 most timing critical paths in the design. Timing variability for a path is defined as the path delay difference between best timing corner and worst timing corner. Use our methodology, we are able to reduce the variability on an average of about $330ps$. With the average path delay variability at $3ns$, we reduce the variability by 11%, which is consistent with our standard cell analysis.

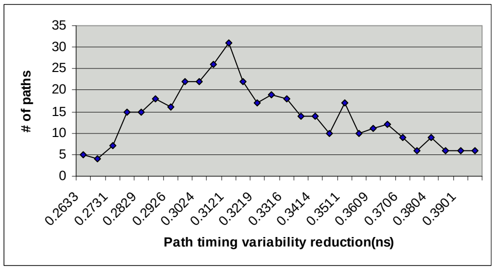


Fig. 35. Timing variability reduction for critical paths.

We present data for leakage analysis for several 65nm hard macros in table 2. Column 3 and 4 are the leakage ratio between two leakage analysis corners for original and new netlists respectively. The data is once again, very consistent with cell level analysis.

3. BEOL Comparison

We use a 65nm high speed IP block implemented with standard cells as our test case. There are about 2k placed standard cell instances in this test case. As described in previous section, we performed two STA simulations based on different approaches for lithography effects on interconnection, one with routing metal drawn shape and the other with litho contour converted polygon for routing metals. We generated STA reports for all endpoints of the design for both setup and hold delays. We then compare the difference of the delays with the two STA results for each endpoint. The difference is shown in Figure 36(a) and Figure 36(b).

The experiment shows that the timing impact for metal litho effects is within 1%-2% for both setup and hold delay paths except one hold delay path, which is about 3.6%. Additionally, litho effects cause some paths to have larger delays, other paths to have smaller delays. In fact, the small differences caused by litho effects on routing metals are in the same order of the errors in STA flow and the parasitic extraction.

D. Conclusion

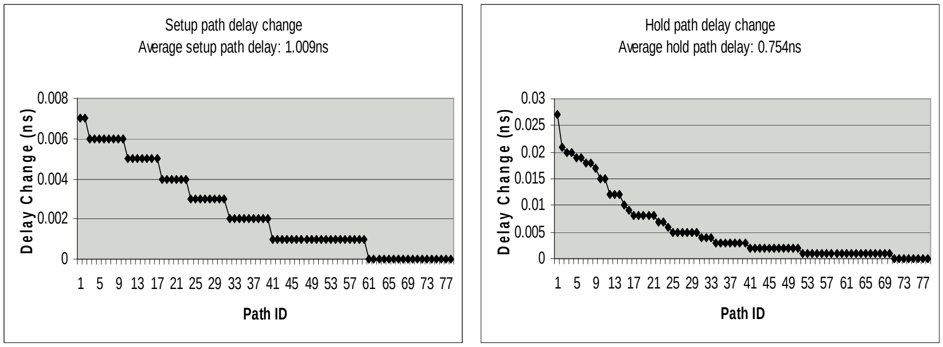
In this chapter, we proposed a new lithography aware design methodology. The systematic lithography effects are considered in the design flow to reduce design pessimism. For polysilicon transistor gates, we introduced dummy poly into the standard cell to achieve context independency of the standard cell timing model; for interconnection, we verified that traditional lookup table based methodology is still applicable considering lithography effect. Our methodology can be easily incorporated into current design flow with virtually no impact on design schedule. We performed a lithography simulation with foundry validated and calibrated production lithography models across process window, we extracted the electrical parameters from the lithography images and applied STA on the design. As a result, we have reduced the pessimism introduced by the traditional process corners methodology for timing and leakage analysis for real 65nm designs.

Table VI. Experimental results for standard cell characterization.

Cell	**Timing**			**Leakage**	
	Orig timing variation (ps)	**New timing variation (ps)**	**%**	**Orig leakage ratio**	**New leakage ratio**
inv	142.0	127.4	10.3	16.5	9.8
and2	268.2	243.2	9.3	24.5	11.4
or2	203.9	182.9	10.3	15.8	7.2
nand2	117.4	104.4	11.1	14.1	6.1
nor2	119.3	105.7	11.4	14.9	6.4
xor2	304.6	268.8	11.7	15.9	6.3
xnor2	393.9	361.3	8.3	27.8	13.4
oai211	176.9	159.8	9.7	13.7	7.0
aoi21	169.4	155.2	8.4	15.3	7.0
d flop	360.9	270.0	25.2	20.3	8.4
ao21	211.3	189.6	10.3	16.5	6.5
buffer	228.2	203.4	10.9	21.0	8.0
ha	303.2	273.6	9.8	15.9	7.6
mux	238.3	211.2	11.4	17.2	8.1
oa21	254.6	223.4	12.3	17.3	7.3
Ave	232.8	205.3	11.4	17.8	8.0

Table VII. Leakage variability analysis.

block	**instance count**	**Orig leakage ratio**	**New leakage ratio**
block 1	87265	14.4	6.7
block 2	20582	15.0	6.5
block 3	38515	19.8	8.8
block 4	94165	19.3	8.3
block 5	236832	18.5	8.0
Ave	95471.8	17.4	7.7



(a) Delay change of setup paths. (b) Delay change of hold paths

Fig. 36. Delay path timing changes.

CHAPTER V

CONCLUSION

Design For Manufacturability (DFM) is seen as an essential channel of communication between design and manufacturing for 65nm VLSI technology and beyond. Our proposed methodologies serve for the very purpose by translating manufacturing information into design guidances that can be utilized by designers. We addressed the issue of Alt-PSM phase error in the standard cell development, we improved the printability of the routing metals without introducing timing degradations, we enabled the litho aware STA flow with litho aware standard cell characterization methodology. Hopefully, our efforts help designers successfully capture some of the most important manufacturing effects and therefore facilitate them to improve both manufacturing and parametric yield of the design.

REFERENCES

[1] A. K-K. Wong, *Resolution Enhancement Techniques in Optical Lithography*. Bellingham, WA: SPIE Press, 2001.

[2] C. Pierrat, M. Cote, and K. Patterson, "New alternating phase-shifting mask conversion methodology using phase conflict resolution," in *Proc. of SPIE*, vol. 4691, July 2002, pp. 325-335.

[3] P. Berman, A. B. Kahng, D. Vidhani, H. Wang, and A. Zelikovsky, "Optimal phase conflict removal for layout of dark field alternating phase shifting masks," *IEEE Trans. on Computer-Aided Design*, vol. 19, pp. 175-187, 2000.

[4] A. B. Kahng, S. Vaya, and A. Zelikovsky, "New graph bipartizations for double-exposure, bright field alternating phase-shift mask layout," in *Proc. of Asia and South Pacific Design Automation Conf.*, Yokohama, Japan, 2001, pp. 113-118.

[5] T. Uehara and W. M. van Cleemput, "Optimal layout of CMOS functional arrays," *IEEE Trans. on Computers*, vol. 30, pp 305-312, May 1981.

[6] T. Iizuka, M. Ikeda, and K. Asada, "High speed layout synthesis for minimum-width CMOS logic cells via Boolean satisfiability," in *Proc. of Asia and South Pacific Design Automation Conf.*, Yokohama, Japan, 2004, pp. 149-154.

[7] E. Zarpas, "Siege vs. zChaff and Berkmin561 on BMC Formulas from the IBM Formal Verification Benchmarks Library,"
http://www.haifa.il.ibm.com/projects/verification/
RB_Homepage/bmcbenchmarks_illustrations.html, July 2004.

[8] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik, "Chaff: engineering an efficient SAT solver," in *Proc. of the ACM/IEEE Design Automation Conf.*, Las Vegas, NV, June 2001, pp. 530-535.

[9] K. Ooi, S. Hera, and K. Koyama, "Computer aiede design software for designing phase shifting masks," *Jpn J. Appl. Phys.*, vol 32, pp. 5887-5891, 1993.

[10] B. Korte, J. Vygen, *Combinatorial Optimization, Theory and Algorithms.* New York, NY: Springer, 2000.

[11] L. Liebmann, J. Lund, F.-L. Heng, and I. Graur, "Enabling alternating phase shifted mask designs for a full logic gate level: design rules and design rule checking," in *Proc. of the ACM/IEEE Design Automation Conf.*, Las Vegas, NV, June 2001, pp. 79-84.

[12] W. Grobman, M. Thompson, R. Wang, C. Yuan, R. Tian, and E. Demircan, "Reticle enhancement technology: Implications and challenges for physical design," in *Proc. of the ACM/IEEE Design Automation Conf.*, Las Vegas, NV, June 2001, pp. 73-78.

[13] L. W. Liebmann, "Layout impact of resolution enhancement techniques: impediment or opportunity?" in *Proc. of the ACM Int. Symp. on Physical Design*, Monterey, CA, Apr. 2003, pp. 110-117.

[14] L.-D. Huang and M. D. F. Wong, "Optical proximity correction (OPC)-friendly maze routing," in *Proc. of the ACM/IEEE Design Automation Conf.*, San Diego, CA, Jun. 2004, pp. 186-191.

[15] P. Gupta and A. K. Kahng, "Manufacturing-aware physical design," in *Proc. of the IEEE/ACM Int. Conf. on Computer-Aided Design*, San Jose, CA, Nov. 2003, pp. 681-687.

[16] J. Mitra, P. Yu and D. Z. Pan, "RADAR: RET-aware detailed routing using fast lithography simulations," in *Proc. of the ACM/IEEE Design Automation Conf.*, Anaheim, CA, Jun. 2005, pp. 369-372.

[17] J. Cong and K.-S. Leung, "Optimal wiresizing under the distributed Elmore delay model," *IEEE Trans. on Computer-Aided Design*, vol. 14, pp. 321-336, 1995.

[18] S. S. Sapatnekar, "RC interconnect optimization under the Elmore delay model," in *Proc. of the ACM/IEEE Design Automation Conf.*, San Diego, CA, Jun. 1994, pp. 392-396.

[19] J. Lillis, C. K. Cheng, and T. Y. Lin, "Optimal wire sizing and buffer insertion for low power and a generalized delay model," *IEEE J. of Solid-State Circuits*, vol. 31, pp. 437-447, Mar. 1996.

[20] J. Cong and L. He, "Theory and algorithm of local refinement based optimization with application to device and interconnect sizing," *IEEE Trans. on Computer-Aided Design*, vol. 19, pp. 406-420, Apr. 1999.

[21] C.-P. Chen, C. C.-N. Chu, and D. F. Wong, "Fast and exact simultaneous gate and wire sizing by Lagrangian relaxation," *IEEE Trans. on Computer-Aided Design*, vol. 18, pp. 1014-1025, July 1999.

[22] J. Cong, L. He, C.-K. Koh, and Z. Pan, "Interconnect sizing and spacing with consideration of coupling capacitance," *IEEE Trans. on Computer-Aided Design*, vol. 20, pp. 1164-1169, Sep. 2001.

[23] R. Socha, "Propagation effects of partially coherent light in optical lithography and inspection," Ph.D. dissertation, Univ. of California, Berkeley, CA, 1997.

[24] D. Lee, *SPLAT v5.0 Users' Guide,* University of California, Berkeley, 1995.

[25] C. Chu and D. F. Wong, "VLSI circuit performance optimization by geometric programming," *Annals of Operations Research*, vol. 105, pp 37-60, July 2001.

[26] C. Visweswariah, "Death, taxes and failing chips," in *Proc. of ACM/IEEE Design Automation Conf.*, Anaheim, CA, Jun. 2003, pp. 343-348.

[27] S. Postnikove, and S. Hector, *ITRS CD Error Budgets: Proposed Simulation Study Methodology.* Austin, TX: International Technology Roadmap for Semiconductors, May 2003.

[28] A. Balasinski, L. Karklin and V. Axelrad, "Impact of subwavelength CD tolerance on device performance," in *Proc. of SPIE*, vol. 4692, July 2002, pp. 361-368.

[29] B. Stine, D. Boning, J. Chung, D. Ciplickas, and J. Kibarian, "Simulating the impact of poly_CD wafer-level and die-Level variation on circuit Performance," in *Int. Workshop on Statistical Metrology*, Kyoto, Japan, 1997, pp. 24-27.

[30] J. Yang, L. Capodieci and D. Sylvester, "Advanced timing analysis based on post-OPC extraction of critical dimensions," in *Proc. of ACM/IEEE Design Automation Conf.*, Anaheim, CA, Jun. 2005, pp. 359-364.

[31] P. Gupta and F. Heng, "Toward a systematic-variation aware timing methodology," in *Proc. of ACM/IEEE Design Automation Conf.*, San Diego, CA, Jun. 2004, pp. 321-326.

[32] W. J. Poppe, L Capodieci, J. Wu, and A. Neureuther, "From poly line to transistor: building BSIM models for non-rectangular transistors," in *Proc. of SPIE*, vol. 6156. Mar. 2006, pp. 235-243.

VITA

Ke Cao was born in Changsha, China. He received B.S. and M.S. degrees from University of Science and Technology of China in 1996 and University of Minnesota in 2000 respectively. He received his Ph.D. in Computer Engineering in August 2007 from Texas A&M University. He was with Integrated Device Technology Inc. from 2000 to 2003 as an IC design engineer. He worked at Qualcomm Inc. from 2004 to 2006 as a VLSI engineer. He is now with Marvell Semiconductor as a staff physical design engineer. His research interests include physical design methodology and EDA development, especially in the area of Design for Manufacturing (DFM), signal integrity and low power design methodology. His permanent address is: 520 Mansion Court, Apt 305, Santa Clara, CA 95054.

The typist for this dissertation was Ke Cao.