A New Approach in Design for Manufacturing for Nanoscale VLSI Circuits

Thesis submitted in partial fulfillment of the requirement

for the Degree of

Master of Computer Science and Engineering

By

DEBASIS PAL

Class Roll. No.: 001410502032
University Registration No.: 129013 of 2014-15
Examination Roll No.: M4CSE1623

Under the Guidance of

Prof.(Dr.) Debesh Kumar Das

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University, Kolkata-700032, WB, India

May,2016

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY
Jadavpur, Kolkata-32

CERTIFICATE OF APPROVAL

This is to certify that the thesis entitled ***"A New Approach in Design for Manufacturing for Nanoscale VLSI Circuits"*** has been carried out by **Debasis Pal** (Registration No.: 129013 of 2014-15, Examination Roll No.: M4CSE1623) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the "Degree of Master of Computer Science and Engineering".

……………………………………………………

Prof. (Dr.) Debesh Kumar Das (***Supervisor)***
Department of Computer Science and Engineering
Jadavpur University, Kolkata-32

……………………………………………………

Dr. Debesh Kumar Das, **Professor and Head**
Department of Computer Science and Engineering
Jadavpur University, Kolkata-32

……………………………………

Dr. Sivaji Bandyopadhyay
Professor & Dean
Faculty of Engineering and Technology,
Jadavpur University, Kolkata-32.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY, KOLKATA-32

ACKNOWLWDGEMENTS

I would hereby like to avail this opportunity to express my deepest gratitude to my sir, advisor and thesis guide **Dr. Debesh Kumar Das**, **Professor and Head**, Department of Computer Science and Engineering, Jadavpur University for his admirable guidance, incessant encouragement, sincere supervision, constant support, patience ,without which it would not have been possible to complete the thesis in time. Our numerous scientific discussions and his many constructive comments have greatly improved this work.

Words cannot express my indebtedness to **Prof. Parthasarathi Dasgupta**, **IIMC**, Department of Management Information Systems for his amazing guidance and supervision. I am deeply grateful to him for the long discussions that helped to enrich the technical content of this manuscript. Without his enthusiasm, encouragement, support and continuous optimism, this thesis would hardly have been completed. Unfortunately since 1st February, 2016 his soul is resting in peace.

I am highly grateful specially to my lab-mate Mr. Abir Pramanik (JRF,JU) who took the important role in helping me in understanding the primaries of VLSI design despite his other commitments. In this regard, I would also like to acknowledge other lab-mates Mr. Arindam Banerjee, Mr. Subrata Das, Ms. Joyati Mondal and Ms. Debobani Choudhury for helping me in different tasks.

Most importantly none of these would have been possible without the love and support of my family. I extend my thanks to my parents, my beloved and especially to my kids whose forbearance and whole hearted support helped this endeavor succeed.

This thesis would not have been completed without the inspiration and support of a number of wonderful individuals — my thanks and appreciation to all of them for being part of this journey and making this thesis possible.

…………………………………………….

Debasis Pal (Roll -M4CSE1623)

Dept. of CSE, Jadavpur University

ABSTRACT

Manufacturing of VLSI chips in the sub-wavelength technology using the conventional lithography has become extremely challenging. A major reason for this is the difficulty to print sub-30 nm half-pitch patterns with stepper size of 193 nm.

Double Patterning Lithography (DPL) is considered to be potential for process nodes at or below 32 nm. In this method, a layout is decomposed into two masks and two features are assigned different colors corresponding to different exposures if the spacing between them is less than a minimum value defined by design-specific rules. But, generally, there are cases where such different colors assignment may not be possible even though the inter-feature spacing is less than the specified minimum. This condition is often known as color conflict of adjacent features. Color conflicts are traditionally resolved by splitting the features. This problem can be modeled as a graph-theoretic problem with color conflicts identified as odd-cycle detection. In case of graph with coloring conflict or the conflict graph, the required solution is that no two adjacent nodes can be assigned on same color.

Double Patterning Lithography (DPL) has been considered widely as good method for the sub-22nm technology to enhance pattern printability. As minimum feature size and pitch spacing further decrease, Triple Patterning Lithography (TPL) is coming up as the most promising technique in the 14nm technology and beyond. TPL is required for the layout where the features are too complex in design and tough to be split always into only two masks (DPL). In common, there may be the cases where such different colors assignment is not possible even though the inter-feature spacing is less than the minimum. The condition often is noted as color conflict among adjacent features. In this dissertation, we propose a *merge* technique with the application of *de-compaction* so that the overall area of the layout is minimally affected to resolve the coloring conflict and also minimize the number of stitches in case of DPL and as well as TPL. We consider layouts having rectilinear features present in the layout and apply the proposed algorithm to obtain a DPL-compliant layout and TPL-compliant layout with selective use of stitches keeping the overall layout area fixed.

Experimental results with layouts illustrate the effectiveness of the proposed algorithm.

Contents

Contents

List of Figures

List of Tables

List of Abbreviations

Chapter 1 Introduction **1**

1.1 Design for Manufacturing (DFM) in submicron VLSI Design 2
1.2 VLSI – Flowchart 3
1.3 Physical Design Flow 4
1.3.1 Partitioning 5
1.3.2 Floorplanning 6
1.3.3 Placement 7
1.3.4 Routing 9
1.3.5 Compaction 10
1.4 Lithography 10
1.5 Photolithography 11
1.5.1 Basic steps of Photolithography 12
1.5.2 Types of Optical Lithography 13
1.5.3 Optical issues (Resolution) in Projection lithography 14
1.5.4 Issue (Resolution) in lithography for diffraction 15
1.6 Resolution Enhancement Technique (RET) 17
1.6.1 Phase Shift Masking (PSM) 17
1.6.2 Optical Proximity Correction (OPC) 18
1.6.3 Sub-Resolution Assist Features (SRAFs) 18
1.6.4 Off-axis illumination (OAI) 19
1.6.5 Multiple Exposure System / Multiple patterning System 19

Chapter 2 Double Patterning Lithography (DPL) & Triple Patterning Lithography (TPL) 21

2.1 Introduction 21
2.2 Double Patterning Lithography (DPL) 22
2.3 Triple Patterning Lithography (TPL) 23
2.4 Critical issues in DPL & TPL 24

2.5 Layout Splitting and Coloring Algorithm 25
2.6 Techniques Used in DPL / TPL 25
2.6.1 Lithography-Etch, Lithography-Etch (LELE) 25
2.6.2 Lithography-Freeze, Lithography Etch (LFLE) 26
2.6.3 Self-Alignment Double Patterning (SADP) 26
2.7 Applications of DPL /TPL 27
2.8 Summary 27

Chapter 3 Double Patterning Lithography (DPL)-Compliant Layout Construction (DCLC) with Area-Stitch Usage Tradeoff 28

3.1 Introduction 28
3.2 Review and Motivation 29
3.3 DPL Layout Decomposition 29
3.4 Problem Formulation 33
3.5 Proposed Algorithm 34
3.6 Complexity Analysis 38
3.7 Examples 38
3.8 Experimental Results 40
3.9 Observation 42

Chapter 4 An Approach to Construct a Triple Patterning Lithography Compliant Layout Using Area-Stitch Tradeoff 43

4.1 Introduction 43
4.2 Preliminaries and Background 44
4.3 Problem Formulation 45
4.4 Proposed Techniques 46
4.5 Examples 47
4.6 Proposed Algorithm 49
4.7 Complexity Analysis 50
4.8 Experimental Results 50
4.9 Observation 52

Chapter 5 Conclusion and Future Work 53

5.1 Conclusion 53
5.2 Future Work 53

Chapter 6 Bibliography 54

List of Figures

Figure 1-1: Block Diagram of DFM 3
Figure 1-2: VLSI Design Flow Diagram 4
Figure 1-3 : Physical Design Flow........ 5
Figure 1-4 : A Circuit and Partitioning of the Circuit........ 6
Figure 1-5 (a) Example of Slicible and (b) Non-slicible floorplanning........ 7
Figure 1-6 : (a) A bad placement and (b) A good placement 8
Figure 1-7: Examples of Global and Detailed Routing........ 9
Figure 2-1: Contact, Proximity and Projection lithography........ 14
Figure 2-2 : An example of effect of diffraction: Image resolution decreases. 16
Figure 2-3: Resolution of layout features for different intensity of light source. 16
Figure 2-4 : (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. 17
Figure 2-5 : Example of Optical Proximity Correction (OPC)........ 18
Figure 2-6 : Examples of Sub-Resolution Assist Features (SRAFs) 18
Figure 2-7 : Example of (a) On-axis and (b) off-axis illumination........ 19
Figure 3-1: Decomposition of a design in Double Patterning Lithography (DPL) 22
Figure 3-2: Process of DPL different two masks........ 23
Figure 3-3 : Triple Patterning Lithography (TPL) 23
Figure 3-4 Example of Coloring Conflict, Stitch and Overlay error 24
Figure 3-5: Decomposition of feature layout to corresponding masks........ 24
Figure 3-6: Double Patterning with Double Exposure and Double Etch (DE/DE) 26
Figure 3-7: Double Patterning with Double Exposure and Single Etch (DE/SE) 27
Figure 3-1: An example of Conflict graph........ 30
Figure 4-2: Concave Corner , Merge 30
Figure 4-3: Issues related to layout decomposition. 31
Figure 4-4: Use of stitches for decomposition in layout........ 32
Figure 4-5: An example of De-compaction 32
Figure 4-6: A weighted graph with its layout 33
Figure 4-7: Removal of color conflict of a weighted layout........ 33
Figure 4-8: Example of removal of conflicts using merging and de-compaction. 39
Figure 4-9: An GSRC benchmark with conflict among modules (1,2,4,6,5) & (3,4,6 ,7) 39
Figure 4-10: Removal of color conflicts in GSRC benchmark circuit. 39
Figure 4-11: An n10c benchmark circuit without color conflict by merges and de-compaction 40
Figure 4-12: Apte benchmark 41
Figure 4-13: Removal of conflicts by decompaction of Apte benchmark........ 42
Figure 5-1: A weighted conflict graph with its layout having color conflict........ 44
Figure 5-2: Issues related to TPL-layout decomposition........ 45
Figure 5-3: An example of De-compaction 46
Figure 5-4: Removal of color conflict of a weighted layout. 47
Figure 5-5: A layout with 11 modules with conflict and its removal by decompaction........ 47
Figure 5-6: An example of removal of color conflict using merging. 48
Figure 5-7: A TPL layout with color conflict & its removal by merging and De-compaction. 48
Figure 5-8 : TPL coloring conflict and its removal by De- compaction of a GSRC benchmark 50

List of Tables

Table 1: Comparison of parameters of different benchmark layouts, before and after applying De-Compaction 41

Table 2: Comparison of parameters of different benchmark layouts, before and after applying De-compaction 51

List of Abbreviations

Abbreviations

ASIC	Application Specific Integrated Circuit
PCB	Printed Circuit Board
DFM	Design for Manufacturing
DFM	Design for Manufacturing
DOF	Depth of Focus
CD	Critical Dimension
DPL	Double Patterning Lithography
TPL	Triple Patterning Lithography
QPL	Quadruple Patterning Lithography
MPL	Multiple Patterning Lithography
RET	Resolution Enhancement Technique
OPC	Optical Proximity Correction
PSM	Phase Shift Mask
OAI	Off -Axis Illumination
SRAF	Sub-Resolution Assist Feature
LELE	Lithography-Etch- Lithography-Etch
LFLE	Lithography-Freeze-Lithography- Etch
SADP	Self-Alignment Double Patterning
OL	Overlay Error
$\mathbf{Min_{cs}}$	Minimum Coloring Spacing

Chapter-1

Introduction

For an ASIC (Application Specific Integrated Circuit) design, it is 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 submicron 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, mechanisms are 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, we will attempt to present some of the new technologies to reduce the gap between design and manufacturing.

Current VLSI technology has the minimum transistor feature size of 90nm and 65nm, which is remarkably below the lithography wavelength of 192nm. The technology trend for future indicates that this gap will become even larger. In the sub-wavelength lithography, Resolution Enhancement Techniques (RET) 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. 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 on photo-mask may result in printed feature with distortion on the silicon. A circuit layout with poor printability implies that it is difficult to make the printed features on wafers to follow designed shapes without distortions. Currently, the printability of devices with sub-wavelength sizes is usually improved by using Resolution Enhancement Techniques (RET)

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.

1.1 Design for Manufacturing (DFM) in submicron VLSI Design

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 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. 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, we 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.

Cutting-edge Design for Manufacturability Technique for Nanoscale CMOS VLSI circuits covering defect analysis, equipment, and lithographic control evaluations, is a holistic approach for VLSI circuit designers to evaluate and analyze IC circuit designs from the manufacturability point of view. This practical guide is ideal for design engineers, managers, students, and academics interested in understanding the sources of semiconductor chip failures and how these problems can be mitigated through design.

In the PCB design process, DFM leads to a set of design guidelines that attempt to ensure manufacturability. By doing so, probable production problems may be addressed during the design stage. DFM is usually used to reduce these costs.

Ideally, DFM guidelines take into account the processes and capabilities of the manufacturing industry. Therefore, DFM is constantly evolving.

Achieving high-yielding designs in the state of the art VLSI technology has become an extremely challenging task due to the miniaturization as well as the complexity of leading-

edge products. Here, the DFM methodology includes a set of techniques to modify the design of integrated circuits (IC) in order to make them more manufacturable, i.e., to improve their functional yield, parametric yield, or their reliability. As manufacturing companies evolve and automate more and more stages of the processes, these processes tend to become cheaper.

In general, Design for Manufacturing (DFM) means new design technologies, tools & methodologies that ensure enhanced printability of patterns control the parametric variation & enhance yield.

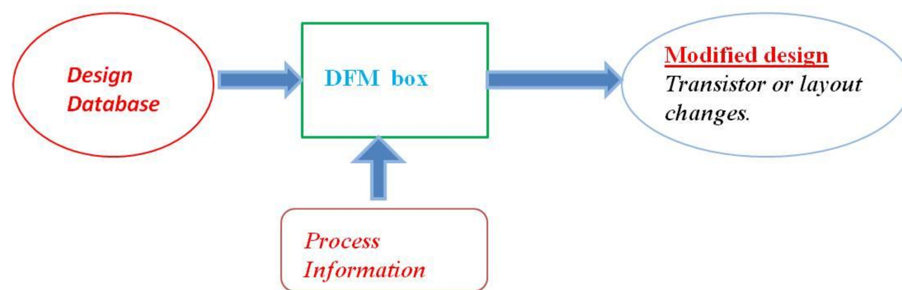


Figure 1-1: Block Diagram of DFM

1.2 VLSI – Flowchart

The functionality of electronics equipment and gadgets has achieved a phenomenal while their physical sizes and weights have come down drastically. The major reason is due to the rapid advances in integration technologies, which enables fabrication of millions of transistors in a single Integrated Circuit (IC) or chip. With the rise in functionality of VLSI ICs, design problem has become huge and complex. To address this complexly issue, after the design specifications are complete almost all the other steps are automated using CAD tools. However, even designs automated using CAD tools may have bugs.

Also, due to extremely large size of the design space it is not possible to verify correctness of the design under all possible situations. So technique are required that can verify, without exercising exhaustive input-output combinations, that the design meets all the input specifications; this technique is called formal verification.

In VLSI designs millions of transistors are packed into a single chip. This leads to manufacturing defects and all the chips need to be physically tested by giving input signals from a pattern generator and comparing responses using a logic analyzer; this process is called Testing. So, in the process of manufacturing a VLSI IC there are three broad steps: DESIGN-VERIFICATION-TEST.

In this thesis we will deal only with digital VLSI circuits. Henceforth, in this course VLSI IC would imply digital VLSI ICs only.

The flow chart of design is as follow.

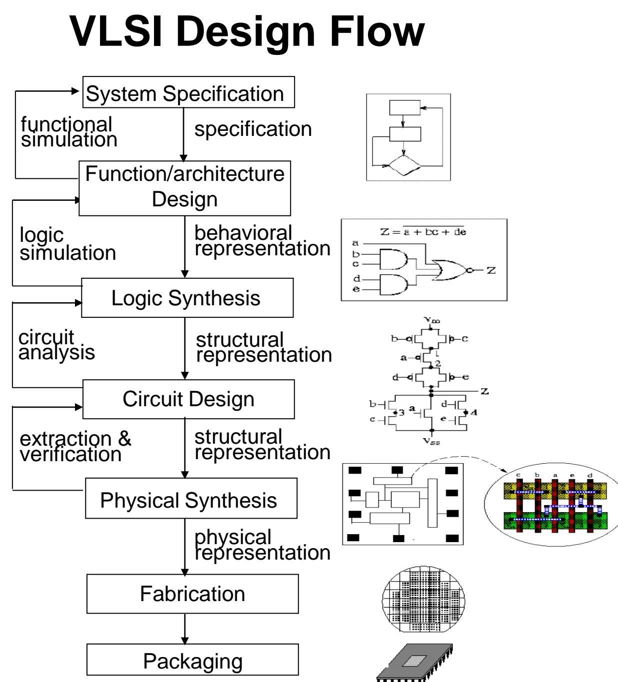


Figure 1-2: VLSI Design Flow Diagram

1.3 Physical Design Flow

VLSI Physical Design Flow is an algorithm with several objectives. Some of them include minimum area, wire length and power optimization. It also involves preparing timing constraints and making sure, that net list generated after physical design flow meets those constraints. The steps of *physical design flow* are ***Partitioning, Floorplanning, Placement, Routing & Compaction***.

The important step in chip design is **floorplanning**, in which the width and height of the chip, basically the area of the chip, is defined.

During **placement** and **routing**, most of the placement tools, place/move logic cells based on floorplan specifications. Some of the important or critical cell's locations have to be pre-defined before actual placement and routing stages. The critical cells are mostly the cells related to clocks, viz. clock buffers, clock mux, etc. and also few other cells such as RAM's, ROM's etc. Since, these cells are placed in to core before placement and routing stage, they are called 'preplaced cells'. The above diagram describes the same.

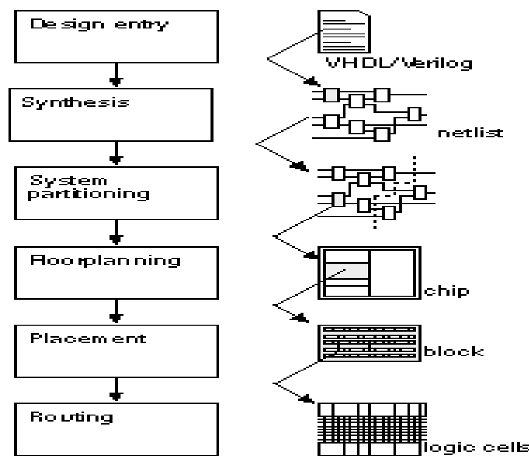


Figure 1-3 : Physical Design Flow

1.3.1 Partitioning

The process of decomposition of a large system into independent manageable subsystems is called Partitioning of a circuit. This is done mainly to separate different functional blocks and also to make placement and routing easier.

Partitioning is required for

- A Complex system with a large number of components
- Efficient design by breaking into smaller subsystems
- Each subsystem can be designed independently and concurrently
- To Decompose the system so as to preserve the original functionality
- To Minimize interface across the partition
- Time for decomposition should be small

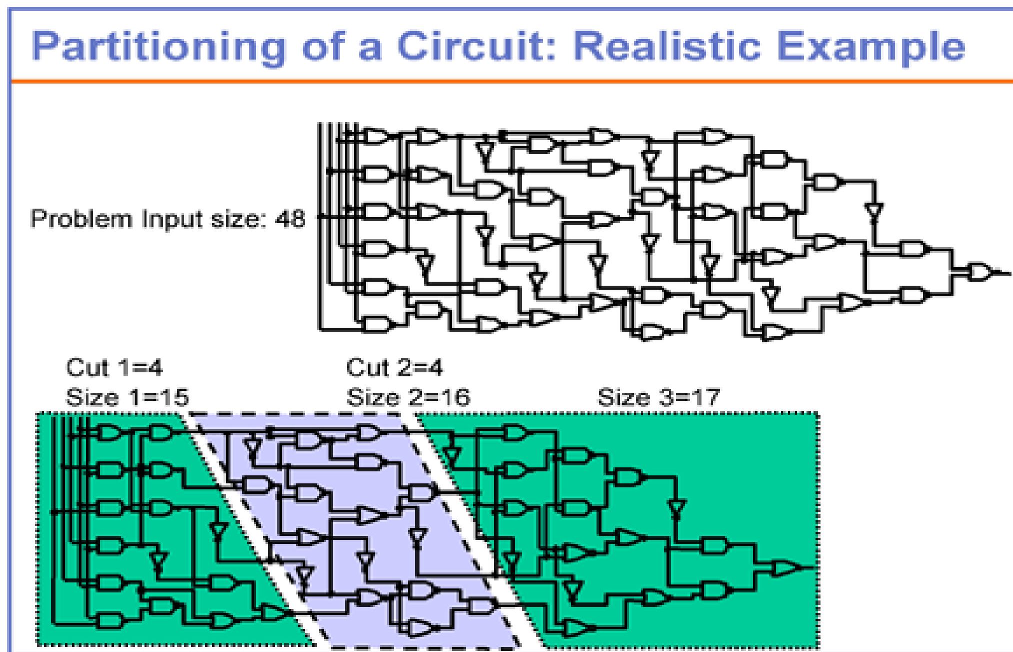


Figure 1-4 : A Circuit and Partitioning of the Circuit.

Circuit Partitioning Complexity

- For *2n circuit nodes*, # of possibilities = $\binom{2n}{n}$
- For a 100-gate circuit, there are 5×10^{28} possibilities; It will take around 1.59×10^{13} years if one checks 100 M possibilities per second

1.3.2 Floorplanning

Floorplanning is the placement of (flexible) blocks with fixed area but unknown dimension.

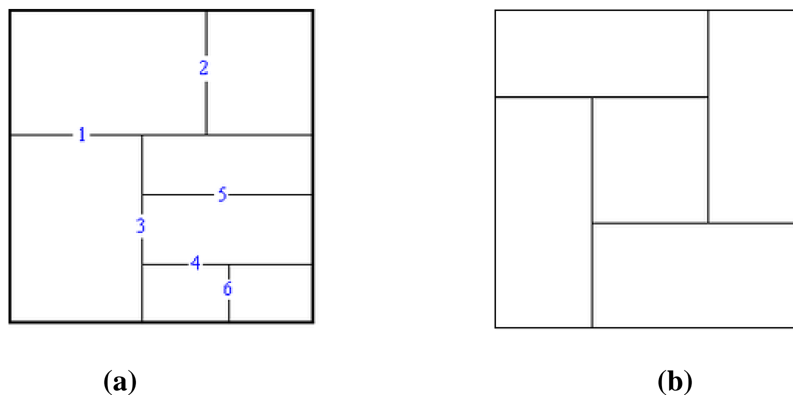
The first important step in the physical design flow is *Floorplanning*. Floorplanning is the process of identifying structures that should be placed close together, and allocating space for them in such a manner as to meet the sometimes conflicting goals of available space (cost of the chip), required performance, and the desire to have everything close to everything else.

Based on the area of the design and the hierarchy, a suitable floorplan is decided upon. Floorplanning takes into account the macros used in the design, memory, other IP cores and their placement needs, the routing possibilities and also the area of the entire design. A bad floorplan will lead to wastage of die area and routing congestion.

In many design methodologies, *Area* and *Speed* are considered to be things that should be traded off against each other.

Floorplanning is of two types:

- **Slicible Floorplanning**: *A floorplanning is slicible if it is obtained by recursively bipartitioning a rectangle into two slicible floorplans either by a horizontal or a vertical line.*
- **Non-Slicible Floorplanning** : *A floorplanning is non-slicible if it is not possible to bipartition a rectangle into two floorplans either by a horizontal or a vertical line.*



(a) **(b)**

Figure 1-5 (a) Example of Slicible and (b) Non-slicible floorplanning

1.3.3 Placement

It is a very important step in physical design cycle.

A process of arranging a set of modules on the layout surface is called ***Placement***. Each module has fixed shape and fixed terminal locations. A poor placement requires larger area and results in performance degradation. A subset of modules may have pre-assigned positions.

Objectives:

- Minimize layout area.
- Reduce the length of critical nets.
- Completion of routing

Placement Problems at Different Levels

System-level placement – Place all the PCBs together such that

- Area occupied is minimum.
- Heat dissipation is within limits.

Board-level placement – All the chips have to be placed on a PCB.

- Area is fixed
- All modules of rectangular shape

Chip-level placement

- Normally, floorplanning / placement carried out along with pin assignment.
- Limited number of routing layers (2 to 4).
- Minimization of area.

Placement is a fundamental problem for physical design. We can see this in the figures as below.

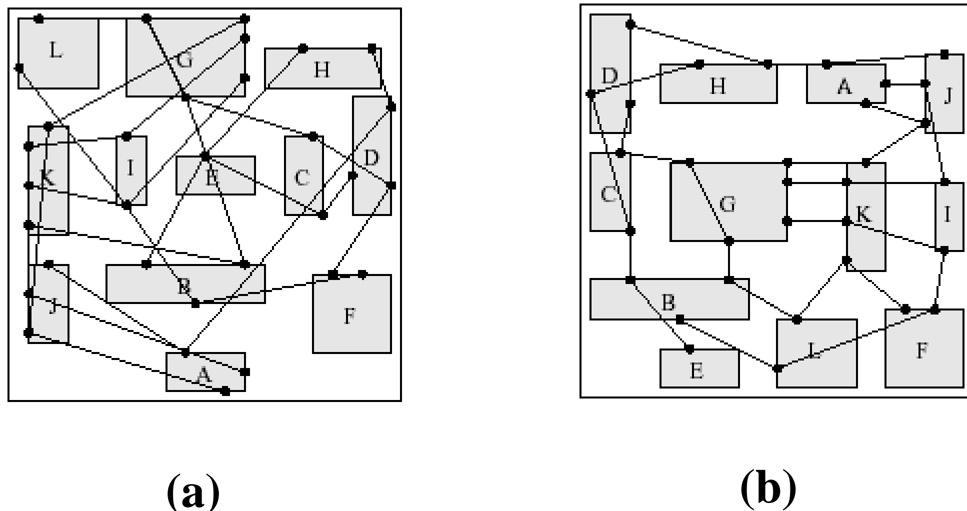


Figure 1-6 : (a) A bad placement and (b) A good placement

The *Placement* phase performs the following tasks.

- Exact locations of circuit blocks and pins are determined.
- A netlist is generated which specifies the required interconnections.
- Region for interconnections (routing region) is also generated in this phase.

1.3.4 Routing

Then ***Routing*** phase completes the interconnections among the modules of a PCB considering the factors like critical path, clock skew, crosstalk, congestion, repeater placement, wire spacing.

Routing is of two types.

- *Global Routing*
- *Detailed Routing*

Global Routing: Global routing is done to provide instructions to the detailed router about where to route every net. It provides the channels for interconnect to be routed.

Detailed Routing: Detailed routing is where we specify the exact location of the wires/interconnects in the channels specified by the global routing.

Routing Regions: Regions through which interconnecting wires are laid out.

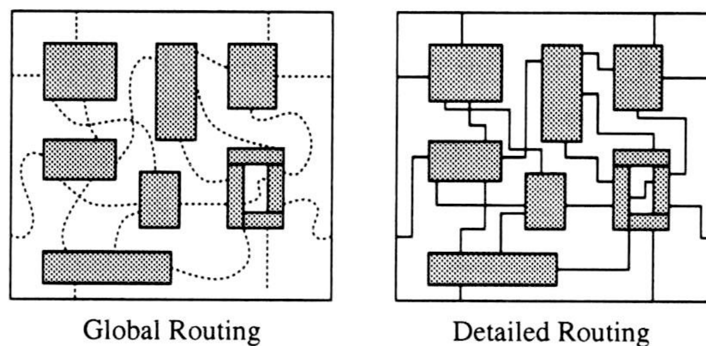


Figure 1-7: Examples of Global and Detailed Routing

Objective of Routing:

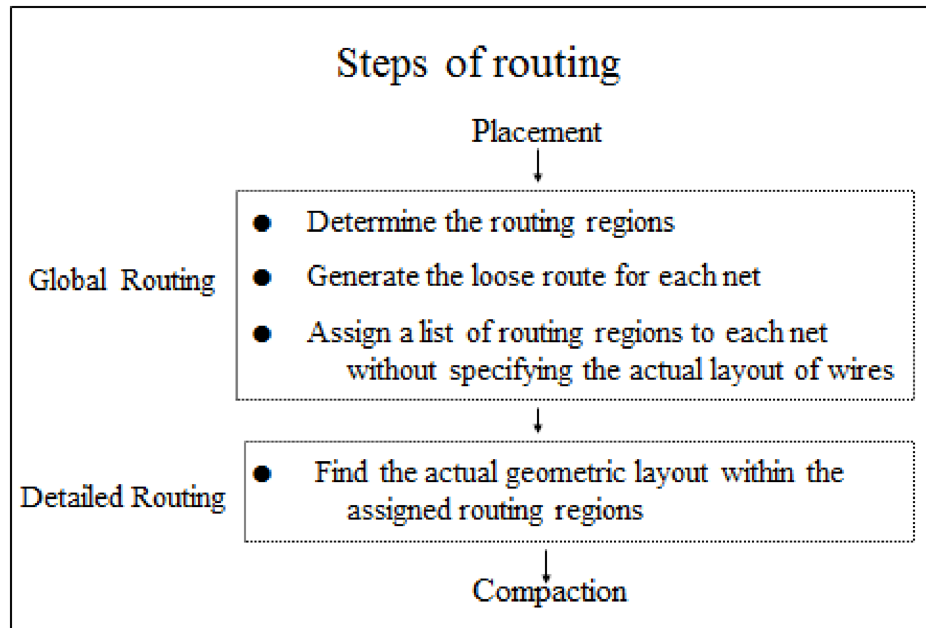
- To complete the interconnections among the blocks according to the specified netlist.
- While interconnecting, each net satisfy different constraints
- Optimize various criteria for better performance

Constraint of Routing:

While interconnecting (Routing) it must follow some constraints. These are as follows.

- Design specification
- Design style

Steps of Routing



Around 60% of the overall design time is spent on routing!!

1.3.5 Compaction

The operation of layout area minimization is called layout *compaction*.

It is the Compaction of:

– Space between the features.

– Size of the features.

– Shape of the features.

Compaction accepts symbolic layout as the input, and generates the final layout as output.

1.4 Lithography

A light sensitive *photoresist* is spun onto the wafer forming a thin layer on the surface. The resist is then selectively exposed to shining light through a *mask* which contains the information for the particular pattern being fabricated. The resist is then developed which completes the pattern transfer from the mask to the wafer.

Lithography comes from two Greek words, "lithos" which means stone and "graphein" which means write. *"Writing a pattern on stone"*

Lithography transforms complex circuit diagrams into patterns which are defined on the wafer in a series of steps of exposure and processing to form a number of superimposed layers of insulator, conductor, and semiconductor materials. Typically *8-25 lithography steps* and several hundred processing steps between exposures are required to fabricate a packed IC. The minimum feature size (the minimum line width or line to line separation) that can be printed on the surface controls the number of circuits that can be placed on the chip and has a direct impact on circuit speed**.**

Types of Lithography:

1. Photolithography
2. E-beam Lithography
3. EUV(Extreme Ultra Violet) lithography
4. X-Ray Lithography

Only Photolithography is cost effective as compared to three other lithography.

1.5 Photolithography

Photolithography, also termed **optical lithography** or UV lithography, is a process used in micro- fabrication to pattern parts of a thin film or the bulk of a substrate. It uses light to transfer a geometric pattern from a photo mask to a light-sensitive chemical "photoresist", or simply "resist," on the substrate. A series of chemical treatments then either engraves the exposure pattern into, or enables deposition of a new material in the desired pattern upon, the material underneath the photo resist. For example, in complex integrated circuits, a modern CMOS wafer will go through the photolithographic cycle up to 50 times.

Photolithography is one of the most important technologies in the production of advanced integrated circuits. In order to make extremely small features, on the order of the wavelength of the light, advanced optical techniques are used to transfer a pattern from a mask onto the surface. A polymeric film or *resist*, is modified by the light and records the information in a process not dissimilar to ordinary photography**.**

Photolithography shares some fundamental principles with photography in that the pattern in the etching resist is created by exposing it to light, either directly (without using a mask) or with a projected image using an optical mask. This procedure is comparable to a high precision version of the method used to make printed circuit boards. Subsequent stages in the process have more in common with etching than with lithographic printing. It is used

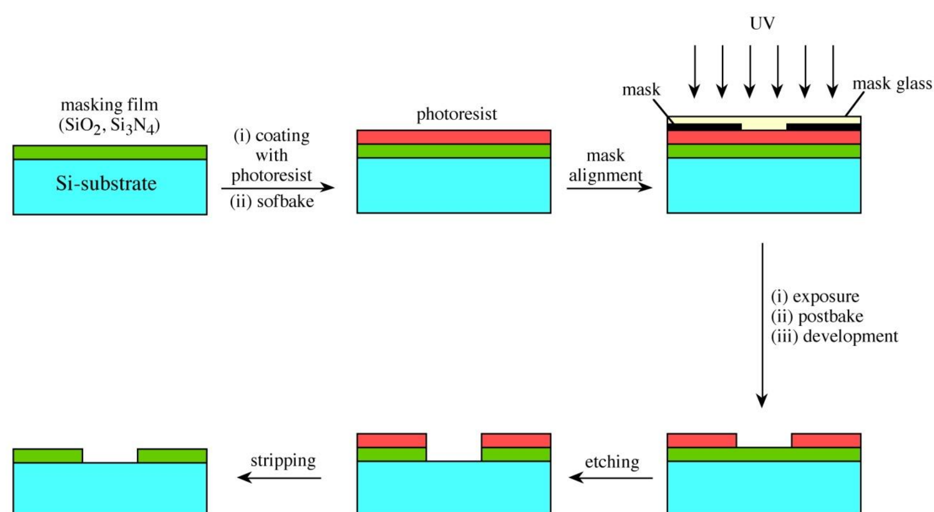
because it can create extremely small patterns (down to a few tens of nanometers in size), it affords exact control over the shape and size of the objects it creates, and because it can create patterns over an entire surface cost-effectively. Its main disadvantages are that it requires a flat substrate to start with, it is not very effective at creating shapes that are not flat, and it can require extremely clean operating conditions.

1.5.1 Basic steps of Photolithography

The process follows the following basic steps:

1. The wafer is spin coated with resist to form a uniform ~1 μm thin film of resist on the surface.
2. The wafer is exposed with ultraviolet light through a mask which contains the desired pattern. In the simplest processes the mask is simply placed over the wafer, but advanced sub-micron technologies require the pattern to image through a complex optical system.
3. The photoresist is developed and the irradiated area is washed away (positive resist or negative resist).
4. Processing (etching, deposition etc.)
5. Remaining resist is stripped.

Steps in optical printing using photolithography:



In addition to being possibly the most important semiconductor process step, photolithography is also the most expensive technology in semiconductor manufacturing. This expense is the result of two considerations:

1. The optics in photolithography tools are expensive where a single lens can cost a $1 million or more
2. Each chip must be exposed individually unlike other semiconductor processes .

This means that not only the photolithography machines are most expensive, but more of them are needed in order to maintain throughput.

1.5.2 Types of Optical Lithography

There are mainly three types of optical lithography.

They are as follows:

1.*Contact Lithography*

2.Proximity Lithography

3.Projection Printing Lithography

Contact Lithography: A contact printer, the simplest exposure system, puts a photomask in direct contact with the wafer and exposes it to a uniform light. Contact printing is liable to damage both the mask and the wafer, and this was the primary reason it was abandoned for high volume production.

Proximity Lithography: A proximity printer puts a small gap between the photomask and wafer.

Projection Lithography: Very-large-scale integration (VLSI) lithography uses projection systems. Unlike contact or proximity masks, which cover an entire wafer, projection masks show only one die or an array of dies (known as a "field"). Projection exposure systems project the mask onto the wafer many times to create the complete pattern.

- **Comparison between Contact Lithography and Proximity Lithography**

In both cases, the mask covers the entire wafer, and simultaneously patterns every die.Both contact and proximity lithography require the light intensity to be uniform across an entire

wafer, and the mask to align precisely to features already on the wafer. As modern processes use increasingly large wafers, these conditions become increasingly difficult.

Research and prototyping processes often use contact or proximity lithography, because it uses inexpensive hardware and can achieve high optical resolution. The resolution in proximity lithography is approximately the square root of the product of the wavelength and the gap distance. Hence, except for projection lithography (see below), contact printing offers the best resolution, because its gap distance is approximately zero (neglecting the thickness of the photoresist itself).

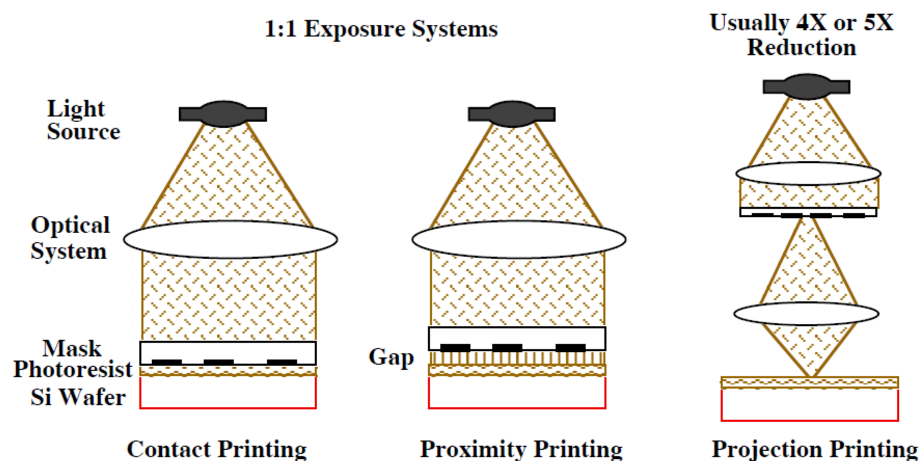


Figure 1-8: Contact, Proximity and Projection lithography

1.5.3 Optical issues (Resolution) in Projection lithography

The ability to project a clear image of a small feature onto the wafer is limited by the wavelength of the light that is used, and the ability of the reduction lens system to capture enough diffraction orders from the illuminated mask. Current state-of-the-art photolithography tools use extreme ultraviolet (EUV) light from excimer lasers with wavelengths of 248 and 193 nm (the dominant lithography technology today is thus also called "excimer laser lithography"), which allow minimum feature sizes down to 50 nm. Excimer laser lithography has thus played a critical role in the continued advance of the so-called Moore's Law for the last 20 years.

The minimum feature size that a projection system can print is given approximately by

$$CD = k_1 . \frac{\lambda}{NA}$$

Where, *CD* is the **minimum feature size** (also called the **critical dimension**). It is also common to write 2 *times* the *half-pitch*.

k1 (commonly called k_1 *factor*) is a coefficient that encapsulates process-related factors, and typically equals 0.4 for production. The minimum feature size can be reduced by decreasing this coefficient through computational lithography.

λ is the wavelength of light used and *NA* is the numerical aperture of the lens as seen from the wafer.

Numerical aperture (NA) of an optical system is a measure of the ability of the lens to collect light.

From this equation, we begin to see what can be done to reduce the critical dimension of a lithography system:

1. Change the wavelength of the source.
2. Increase the numerical aperture (NA).
3. Reduce k_1.

In modern systems, the depth of focus (DOF) is also a concern .The depth of focus is the length along the axis in which a sharp image exists. Naturally a large DOF is desirable for ease of alignment, since the entire chip must with lie within this region. In reality, however, the more meaningful constraint is that the DOF must be thicker than the resist layer so that the entire volume of resist is exposed and can be developed. The depth of focus can also be expressed as a function of numerical aperture and wavelength:

$$DOF = k_2 \frac{\lambda}{[n\sin(\theta)]^2}$$

These two quantities, DOF and CD, provide the direction in lithography and semiconductor processing as a whole.

1.5.4 Issue (Resolution) in lithography for diffraction

Diffraction is a concern in photolithography. This phenomenon is also referred to as the "deviation" of light from a rectilinear path. Diffraction is the foremost phenomenon in projection imaging, because the mask, with its stencil pattern of lines and holes, becomes a diffraction grating. So, the light image that passes through the mask diffracts into a larger image. This is a problem since minimizing feature size is a goal for IC fabrication. To

minimize feature size, we need a minimum size of the mask. However, smaller apertures result in an increase in diffusion. To counteract this, the wavelength of light used is minimized. This relationship is called the diffraction limit. In effect, the minimum feature size is directly proportional to the wavelength of light. Over time, semiconductor equipment makers have steadily decreased the light wavelengths used in lithographic systems from 365 nm in the 1980s to 193 nm today. Using resolution enhancement techniques (RETs), these systems produce ICs with feature sizes of 100 nm.

As diffraction is the spread of light/radiation into unexposed region, it has great effect to the photoresist wafer in a lithography process.

In below diagram we can see the effect of diffraction i.e. how it decreases the resolution of an object image.

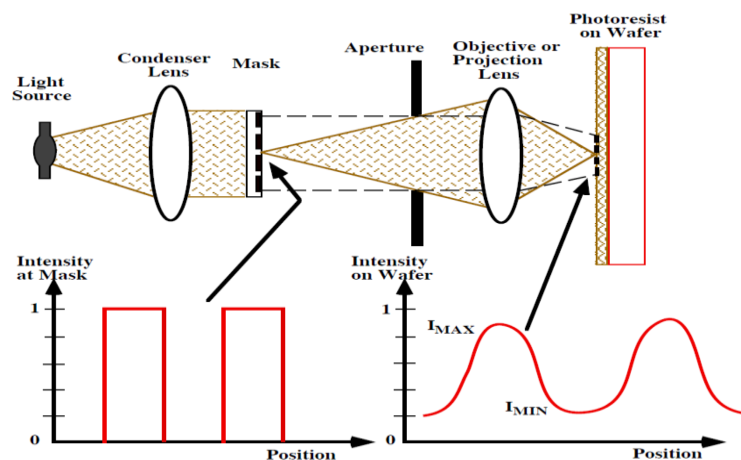


Figure 1-9 : An example of effect of diffraction: Image resolution decreases.

The following figure shows the deceased resolution of layout printing.

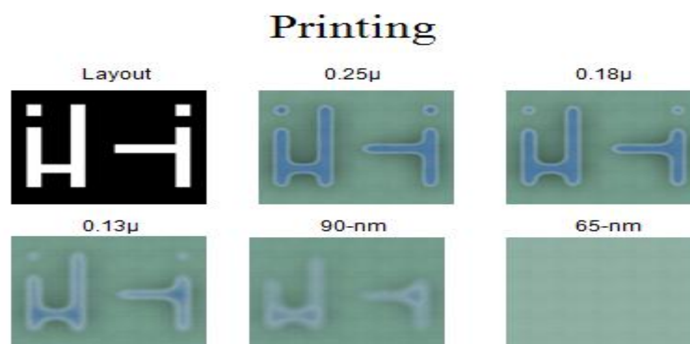


Figure 1-10: Resolution of layout features for different intensity of light source.

1.6 Resolution Enhancement Technique (RET)

Resolution enhancement techniques (RETs) are methods used to improve the resolution of the lithography system by manipulating various parameters of the system. Most RETs aim to manipulate the patterns on the mask. The resolution of a feature being printed depends on neighboring features and the spacing between them.

The principle of diffraction governs the interaction of light waves that pass through the mask patterns on the mask while being projected onto the wafer.

Resolution enhancement techniques include

- Phase Shift Masking (PSM)
- Optical Proximity Correction (OPC)
- Sub-Resolution Assist Feature (SRAF)
- Off-Axis Illumination (OAI)
- Multiple Exposure System / Multi Patterning System [such as *Double Patterning Lithography (DPL), Triple Patterning Lithography (TPL)*]

1.6.1 Phase Shift Masking (PSM)

Phase Shift Masking (PSM) utilizes the superposition principle of light waves to improve the resolution by creating phase changes in spaces between the features. 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.

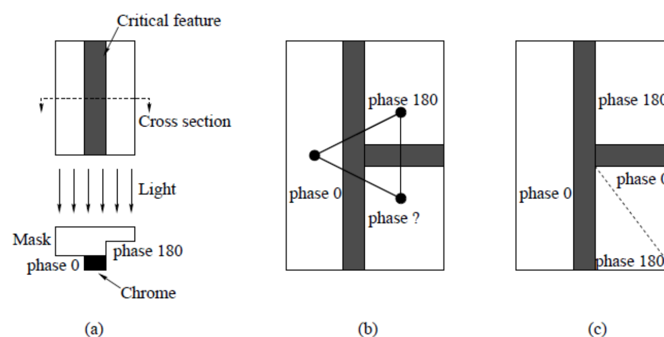


Figure 1-11 : (a) Alternating phase Shifting Mask (Alt-PSM) (b) Phase Conflict occurs for a T-shaped critical feature (c) Phase conflict removal by splitting a phase region.

1.6.2 Optical Proximity Correction (OPC)

Interference of diffraction patterns changes in feature widths (line width reduction, line end shortening, corner rounding). Optical Proximity Correction (OPC) changes the shape of the feature by adding extra jogs and serifs to improve the resolution. It divides each polygon into segments. Remove /Add features from/to segments to improve printability in the presence of proximity effects which gives better approximation of the mask patterns and modifies amplitude of the diffraction pattern of mask features.

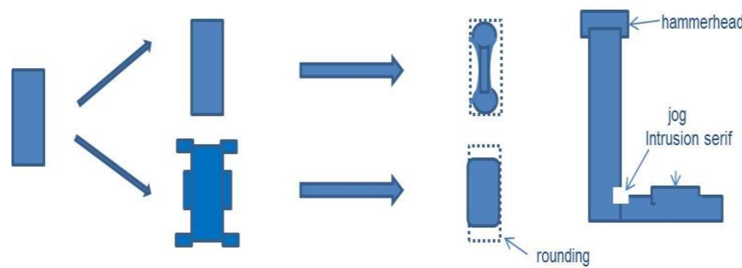


Figure 1-12 : Example of Optical Proximity Correction (OPC)

1.6.3 Sub-Resolution Assist Features (SRAFs)

Sub-Resolution Assist Features are assist features or scattered bars that are drawn adjacent to the mask polygons in order to enhance the diffraction pattern. Addition / insertion of extra features improve the diffraction pattern of the main features. SRAFs cause destructive interference due to phase difference and destructive interference improves the contrast of the image on the wafer. Phase depends on the pitch between the main feature and the SRAF

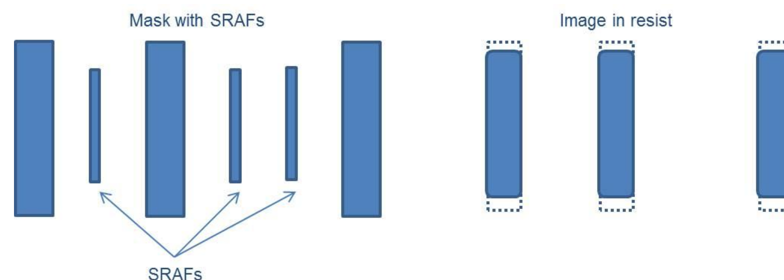


Figure 1-13 : Examples of Sub-Resolution Assist Features (SRAFs)

1.6.4 Off-axis illumination (OAI)

Off-axis illumination is based on the principle that, if the light rays are incident at an angle on the mask, then higher order diffraction patterns can be made to pass through the lens and thereby improve resolution. It is used to print features that are oriented in a particular direction of a mask. Because OAI prescribes what lenses to use for a given feature orientation, this techniques is categorized as “lens” RET.

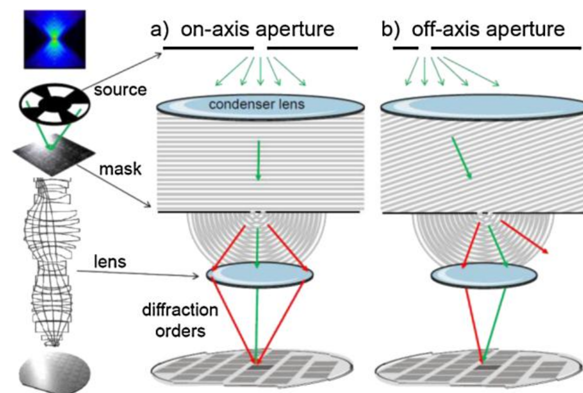


Figure 1-14 : Example of (a) On-axis and (b) off-axis illumination

1.6.5 Multiple Exposure System / Multiple patterning System

Another promising technique that improves the resolution of patterns being printed is Multiple Exposure System/Multiple patterning (or multi-patterning). It is a class of technologies for manufacturing integrated circuits (ICs), developed for photolithography to enhance the feature density. It is expected to be necessary for the upcoming 10 nm and 7 nm node semiconductor processes and beyond. The premise is that a single lithographic exposure may not be enough to provide sufficient resolution. Hence additional exposures would be needed, or else positioning patterns using etched feature sidewalls (using spacers) would be necessary.

Although EUV has been projected to be the next-generation lithography of choice, it would still require more than one lithographic exposure, due to the foreseen need to first print a series of lines and then cut them. It is also likely more than one cut would be needed, even for EUV.

Multi-exposure lithography systems are provided for improved overlay accuracy. In one aspect, a method for multi-exposure lithography operates by determining overlay parameters corresponding to each of a plurality of sub-layouts, inputting the overlay parameters into an exposure system, exposing each sub-layout to photoresist on a wafer by using the exposure system, wherein prior to the exposure process for a given sub-layout, a correction process is performed for the sub-layout using a corresponding overlay parameter to correct an overlay of the sub-layout, and developing the exposed photoresist after exposing all of the sub-layouts.

Multiple Patterning System is of different types. Mostly used technique is *Double Patterning Lithography (DPL)* then *Triple Patterning Lithography (TPL).*

Double Patterning Lithography (DPL): In DPL, the layout patterns are decomposed into two masks. Two features have to be assigned opposite colors if their spacing is less than certain minimum coloring distance.

Triple Patterning Lithography (TPL) : In case of Triple Patterning Lithography, the total layout is partitioned into three different masks. Each mask is assigned different colors. In TPL manufacturing process, there are three exposure/etching steps, through which the layout can be produced.

Chapter-2

Double Patterning Lithography (DPL) & Triple Patterning Lithography (TPL)

2.1 Introduction

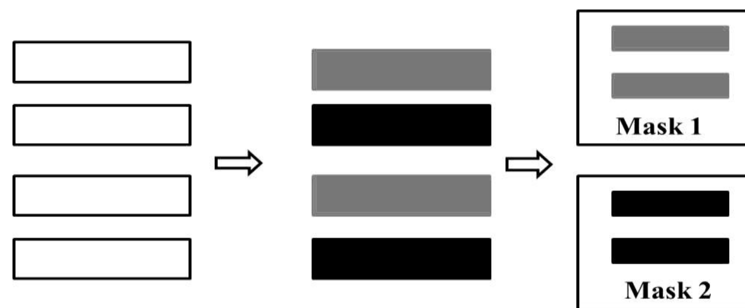
As semiconductor technology advances toward sub- 20-nm nodes, the dimensions of the back end of line (BEOL) decrease according to the scaling of the device. Thus, patterning of narrow interconnects has become a major bottleneck for the success of the manufacturing of integrated circuits (ICs). To resolve BEOL patterning problems, several pitch splitting techniques have been proposed and considered as efficient solutions. Double patterning lithography (DPL), or litho-etch-litho-etch (LELE), has been used for 20-nm logic nodes to pattern interconnect dimensions less than 100 nm . In the DPL process, the original design is decomposed into two separate masks (or colors), and then interconnect lines are printed by the two continuous litho-etch processes to print narrow lines reusing existing 193-nm and 1.35-NA lithography. Recently, the triple patterning lithography (TPL), or litho-etch-litho-etch-litho-etch (LELELE), has been proposed for backend patterning of a 14-nm node in which the minimum metal pitch can be below 50 nm. In TPL, three individual layers or masks can be generated in layout decomposition. Both DPL and TPL are good solutions for printing highly dense and tiny interconnects. However, there are several inherent side effects in multi-patterning lithography. The critical dimension (CD), or metal width, or metal height (H) between two adjacent lines (or three parallel lines in TPL) varies as the result of two (or three) separate lithography and etching processes. In addition, overlay (OL) errors occur because of the misalignment between two or three consecutive litho- etch processes. The OL errors cause non-uniform spacing between neighboring interconnect

lines and are attributed to the change in the coupling capacitance. As a result, electrical parameters such as resistance and capacitance may differ among identical parallel lines in the same layers. Thus, understanding of the performance impact of the structural variations and OL errors by multiple pattern lithography have become extremely important for the success of future integrated circuit designs.

2.2 Double Patterning Lithography (DPL)

Double patterning lithography (DPL) is considered as a most likely solution for 32nm/22nm technology. In DPL, the layout patterns are decomposed into two masks (colors). Two features have to be assigned opposite colors if their spacing is less than certain minimum coloring distance. However, a proper coloring is not always feasible because two neighboring patterns within the minimum distance may be in the same mask due to complex pattern configurations. In that case, a feature may be split into two parts to resolve the conflict but the resulting stitch causes yield loss due to overlay error and increases manufacturing cost.

In DPL, no new technology is introduced. This may be viewed as a short term solution to keep pace with Moore's Law. When DPL is used with immersion techniques it can produce feature sizes of 32nm and beyond.



One single design is decomposed into two masks and pitch size is effectively increased in DPL

Figure 2-1: Decomposition of a design in Double Patterning Lithography (DPL)

The following figure will definitely explain how a feature is produced using two masks.

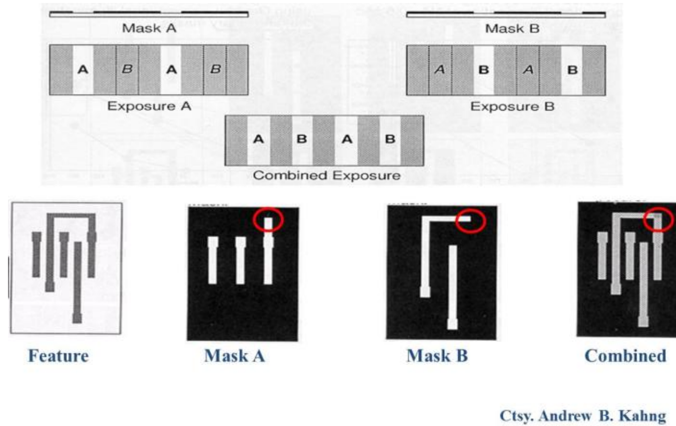


Figure 2-2: Process of DPL different two masks.

2.3 Triple Patterning Lithography (TPL)

Triple patterning lithography (TPL) is one of the most promising techniques in the 14nm logic node and beyond. Instead of two colors if we are allowed to use three colors, we may have more flexibility and better printability. This method of lithography technique is known as Triple Patterning Lithography (TPL). In case of TPL, the total layout is partitioned into three different masks. But, not all conflicts between all the features can necessarily be resolved. In triple patterning lithography manufacturing process, there are three exposure/etching steps, through which the layout can be produced. The advantage of this process is that the effective pitch can be improved, which enhances the lithography resolution. Like in double patterning, the key challenge of triple patterning lithography is the layout decomposition, where input layout is divided into three masks.

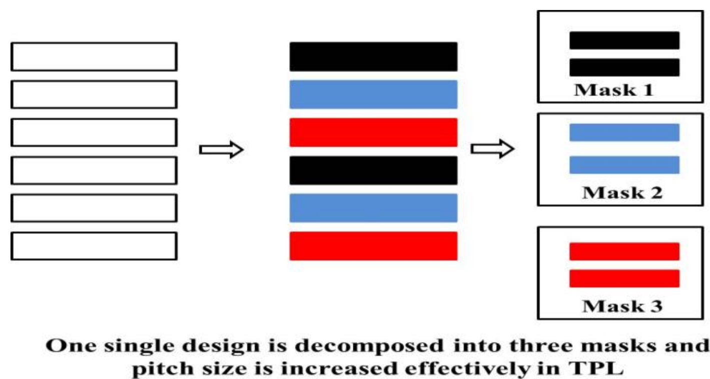


Figure 2-3 : Triple Patterning Lithography (TPL)

2.4 Critical issues in DPL & TPL

There are two critical issues with DPL layout decomposition: **Coloring Conflict and Splitting Stitch**.

Coloring Conflict: If the distance between two separate features in the same mask is less than minimum coloring spacing means, they should be assigned to different masks (colors). Otherwise, there will be a *coloring conflict*. This is common in complex design patterns. Figure 3-4 (a) shows a layout with three features, and any two of them are required to have different colors because of the insufficient spacing.

Splitting Stitch: The *stitch* (Figure 3-4(b)) exists when two touched features are assigned to different masks. However, stitch can have negative effects such as yield loss due to *overlay error* (Figure 3-4(c)) between the two masks and increase of manufacturing cost.

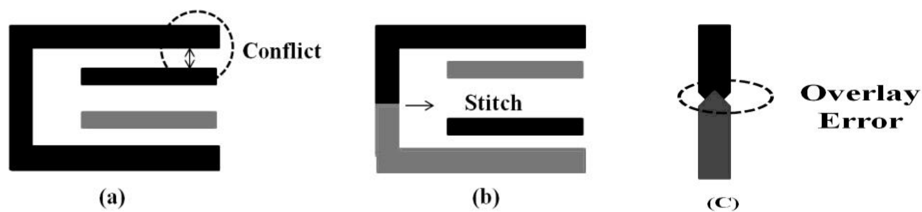


Figure 2-4 Example of Coloring Conflict, Stitch and Overlay error

Mask Cost: Mask cost is linearly proportional to the data volume. If feature complexity increases, obviously mask design will be complex in nature resulting increased mask cost. The following figure demonstrates the fact.

i. e. **Mask Cost ∝ Data Volume**

OPC, PSM→ increased feature complexity → increased mask cost

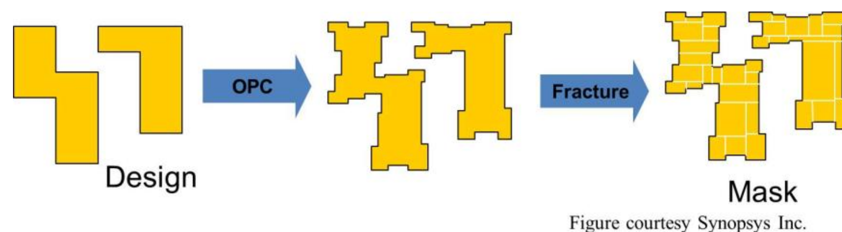
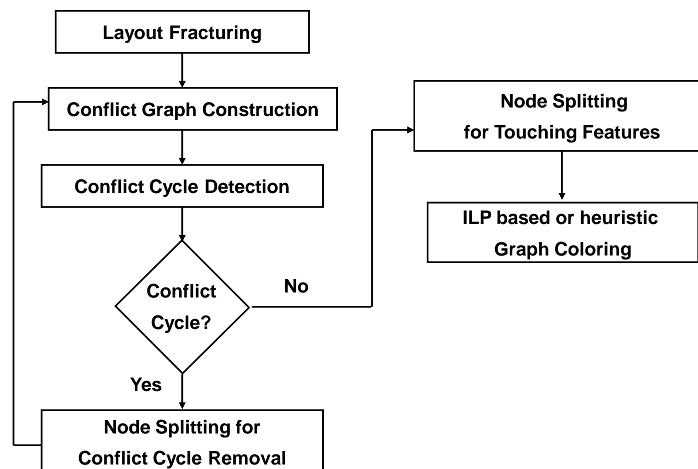


Figure 2-5: Decomposition of feature layout to corresponding masks.

2.5 Layout Splitting and Coloring Algorithm



2.6 Techniques Used in DPL / TPL

Three main techniques

1. Lithography-Etch, Lithography-Etch (LELE)
2. Lithography-Freeze, Lithography Etch (LFLE)
3. Self-Alignment Double Patterning (SADP)

2.6.1 Lithography-Etch, Lithography-Etch (LELE)

Advantages

- No new technology
- Allows for greater resolution
- Uses existing technology
- Straightforward process

Disadvantages

- Requires 5 process steps
- Expensive as litho-etch process twice
- Low throughput
- Small tolerance for pattern overlay

2.6.2 Lithography-Freeze, Lithography Etch (LFLE)

Advantages

- Four process steps (five for LELE)
- Reduced cost
- Increased throughput

Disadvantages

- Faces same issues with small overlay tolerance

2.6.3 Self-Alignment Double Patterning (SADP)

Advantages

- Eliminates trouble with pattern overlay tolerance

Disadvantages

- Increased process steps – increased cost
- Optimized for processes with uniform patterns

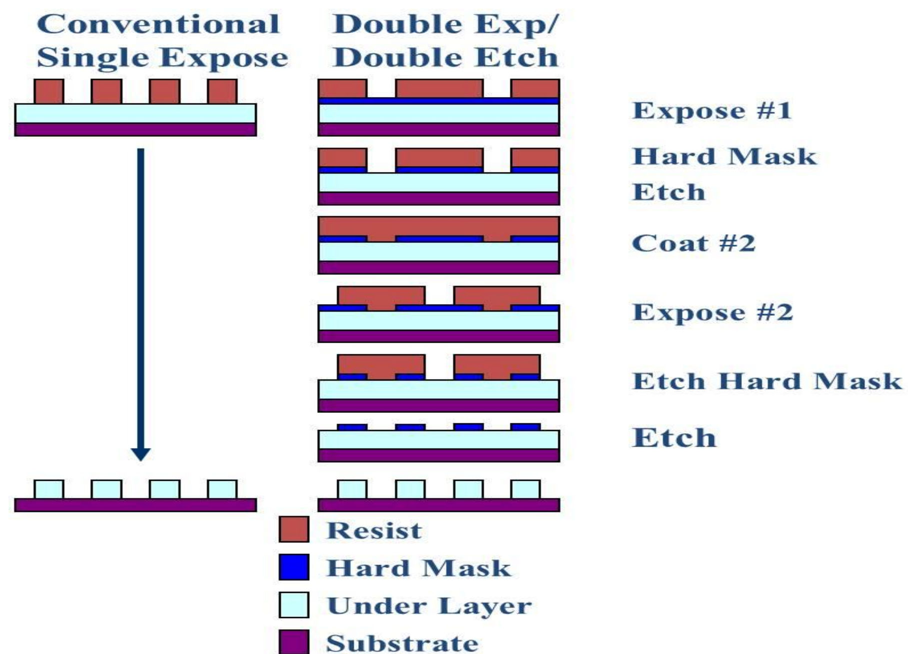


Figure 2-6: Double Patterning with Double Exposure and Double Etch (DE/DE)

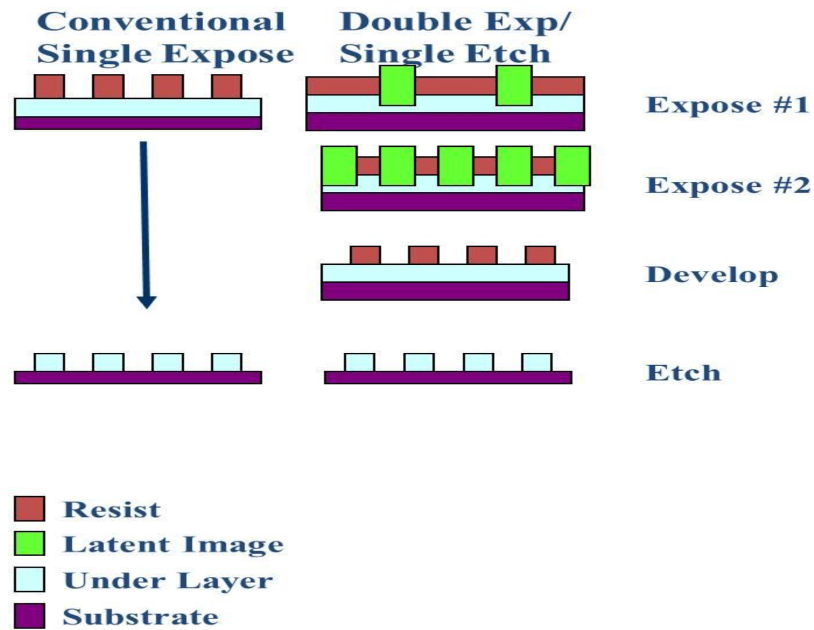


Figure 2-7: Double Patterning with Double Exposure and Single Etch (DE/SE)

2.7 Applications of DPL /TPL

- Memory Devices
  - Self-Aligned Double Patterning (SADP)
    - Used because these devices typically have uniform patterns
    - Used by Hynix, Micron, Renesas, and Samsung
- Logic Devices
  - Litho-Etch, Litho-Etch (LELE) and Litho-Freeze, Litho-Etch (LFLE)
    - Used because these devices typically have non-uniform patterns
    - Used by Intel, Sony, TI, Toshiba, and TSMC

2.8 Summary

Physical designers and mask community need to work together to maintain cost (value) trajectory of Moore’s Law. Bidirectional interaction between design and manufacturing based on cost, value to transfer of functional intent to mask and foundry flows and transfer of constraints of mask and foundry flows up to design. DPL and TPL is the improved Manufacturing-aware Physical Design.

Chapter-3

Double Patterning Lithography (DPL)-Compliant Layout Construction (DCLC) with Area-Stitch Usage Tradeoff

3.1 Introduction

Manufacturing of VLSI chips in the sub-wavelength technology using the conventional lithography has become extremely challenging. A major reason for this is the difficulty to print sub-30 nm half-pitch patterns with stepper size of 193 nm. One possible remedy is the use of immersion lithography which uses high numerical aperture (NA) systems (high index fluids). However, it is difficult to find new liquid material to increase NA beyond 1.35. Alternative solution is the use of extreme ultra-violet (EUV) lithography with illumination source of 13.5 nm. However, this too has serious limitations owing to the lack of power sources, resists, and defect free masks. Resolution enhancement techniques are traditionally used to achieve better quality designs.

A primary lithography candidate proposed in recent past for such technology nodes is *Double Patterning Lithography* (DPL). It involves partitioning dense circuit patterns into two separate masks. The decreased pattern density in each exposure improves the resolution and the depth-of-focus (DOF) of the pattern. DPL can be implemented using several techniques called as either of litho-etch-litho-etch (LELE), litho-freeze-litho-etch (LFLE) or Self-Aligned Double Patterning (SADP) techniques. Though, subsequent to DPL, several advanced strategies of triple-patterning, quad-patterning have already appeared in literature, the former is still of research interest owing to its simplicity and effectiveness.

In DPL if any two features are separated by a distance less than a minimum specified value given by the design rule, they are said to have a *conflict* between them. In such a case, for

better printability, in DPL, these two features are considered independently in two different exposures. However, not all conflicts between all the features can necessarily be resolved. A possible way to resolve conflicts is to decompose one or more features. Every type of DPL requires layout decomposition before manufacturing. Subsequent to exposure and printing of the (decomposed) features, a common strategy to connect the fragments of identical features is to insert stitches between them. Insertion of stitches, however, may cause degradation of printability due to overlay errors and line-end effect. Thus, for layout decomposition, it is preferable to have fewer stitches. A DPL-compliant layout using minimum stitches is done here.

3.2 Review and Motivation

The problem of stitch minimization for DPL has been considered in several existing works. The use of layout modification and de-compaction for the DPL problem has been suggested in several previous works. However, most of these existing works consider stitch minimization and conflict minimization by modeling the conflicts as graphs with vertices representing the modules in the layout and edges between them representing their mutual conflicts in terms of the separation between the modules. For DPL, 2-colorability of the conflict graph is achieved using decomposition of the modules. The existing works also do consider fracturing of the polygons even in the body causing overlay errors, and do not use de-compaction effectively.

In this dissertation paper, I use a restricted version of the *conflict graph* (it is restricted in the sense that we consider all the polygons are rectangular in shape) model and apply certain techniques to gradually modify it into a bipartite graph so as to achieve 2-colorability. The proposed method attempts to achieve an optimal combination of number of stitches, and area overhead due to de-compaction.

3.3 DPL Layout Decomposition

Multi-patterning in general and double patterning in particular are well-researched techniques in lithography, due to their capability of achieving better printability. Double Patterning Lithography involves the decomposition of a mask into two masks subject to certain minimum spacing rule constraints. The existing design flow in DPL initially partitions rectilinear layout features into a set of non-overlapping rectangles using a minimum-sliver fracturing algorithm.

We define a few terminologies below which would be used in subsequent discussions.

***Definition 1**:* A *conflict graph* is constructed by connecting the rectangular features according to a minimum spacing rule, where a vertex of the graph corresponds to a rectangular feature and an edge of the graph corresponds to the connections with the distance between neighbouring rectangular features (see Figure 4-1).

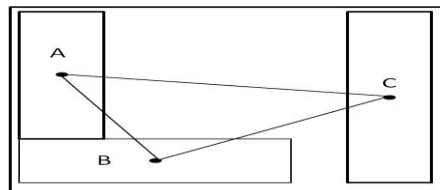


Figure 3-3-1: An example of Conflict graph.

There are two types of edges between the pairs of vertices depending on whether the vertices correspond to the same original feature or whether they belong to different polygons with a separation by a distance less than the minimum spacing as defined by the design rules. In the latter case, the corresponding vertices in the graph model are said to have a conflict between them.

Definition 2.*Yield loss* is the proportion of the number of chips produced from a wafer that are functionally incorrect.

Definition 3. The *concave corner* of a polygon is one at which the internal angle between a pair of adjacent sides is 270° (see Figure 4-2(a)).

Definition 4. A *fracturing* of a polygon is a horizontal or vertical line segment at least one of whose endpoints is incident on a concave corner. The other endpoint is obtained by extending the line segment within the polygon until it encounters the boundary of the polygon. Fracturing of the polygons is done only at the concave corners to avoid fabrication issues.

Definition 5. A *merge* is joining of two rectangular blocks either vertically or horizontally and erasing the line segment between them (see Figure 4-2(b)).

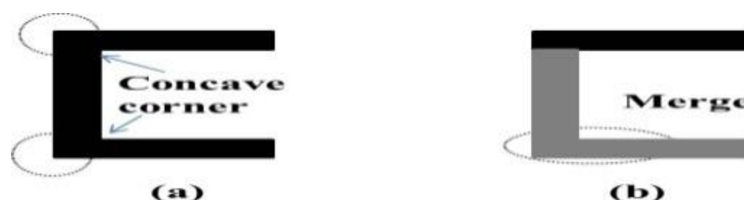


Figure 3-2: Concave Corner , Merge

However, a proper coloring is not always feasible because two neighboring patterns within the minimum distance may be in the same mask due to complex pattern configurations. In that case, a feature may be fractured into two parts to resolve the conflict but the resulting stitch causes yield loss due to overlay error and increases manufacturing cost.

We introduce certain concepts and operators through the Figure 4-3 for specific purposes used in our proposed algorithm below:

A. Coloring Conflict—If the distance between two separate features is less than minimum coloring spacing, they should be assigned to different masks (colors). Otherwise, there will be a coloring conflict.

This is common in complex design patterns. Figure 4-3(a) shows a layout with three features, and any two of them are required to have different colors because of the insufficient spacing. A coloring conflict is unavoidable as in Figure 4-3(b). Sometime, violations can be eliminated by appropriately splitting as in Figure 4-3(e). Splitting may shift the conflict position as in Figure 4-3(c) and 4-3(d).

B. Splitting Stitch—The stitch exists when two touched features are assigned to different masks.

The stitch can be inserted to split some features to resolve the conflict as indicated in Figure 4-3(e) and Figure 4-3(f). However, stitch can have negative effects such as yield loss due to overlay error, etc. and increases the manufacturing cost. Without altering layout in the scope, the traditional objective of layout decomposition in DPL has been minimizing the unresolved conflicts by introducing as few stitches as possible.

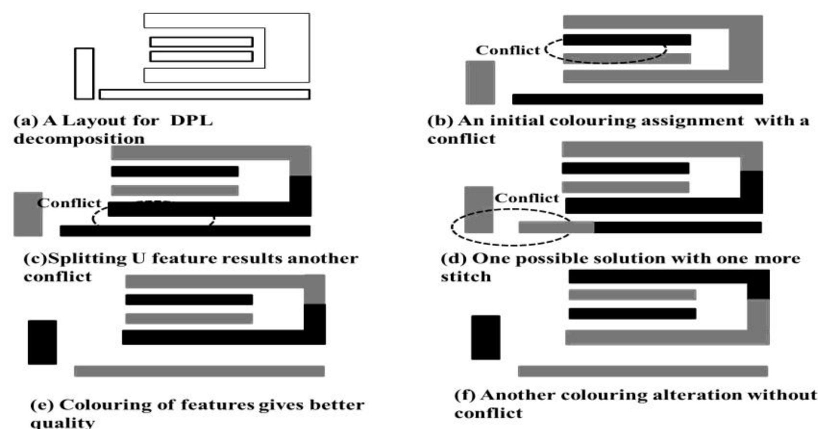


Figure 3-3: Issues related to layout decomposition.

C. Minimization of Stitches- Our main objective is to minimize the number of stitches in a given layout. At the same time the position of the stitch is also important. It may be at middle of a rectangular polygon, as in Figure 4-4(a) or concave corner of a L-shaped polygon, as in Figure 4-4(b).

Figure 3-4: Use of stitches for decomposition in layout.

In several of the existing works, stitches have been inserted either at a concave corner or in the middle of a rectilinear feature in layout. However, the use of stitches in the middle of a feature has the drawback of introducing misalignment. In our proposed work, stitches, if introduced, are at concave corners only (see Figure 4-4(b)).

D. De-Compaction— In certain situations, conflicts may occur in both vertical and horizontal directions, which cannot be resolved by the use of splits. To resolve such conflicts, some blocks may have to be moved in either vertical or horizontal direction but not both simultaneously. This is called *de-compaction* (see Figure 4-5).

Figure 3-5: An example of De-compaction

Let us consider a layout as an example and draw its weighted graph of the layout in which weights of the edges are defined as the minimum distance between the vertices (i.e. midpoint of each blocks) and the minimum coloring conflict spacing is less than or equal to 0.3 unit. The weight of the edges (or between the vertices) AB,AC,BC,BE,CD,CE,DE,DF,EF,EG,FG are 0.1,0.7,0.3,0.25,0.25,0.3,0.3,0.5, 0.5,0.3,0.8 respectively. In our example, the conflict occurs among the blocks B, C, E and C, D, E as shown in Fig.8. We resolve the conflicts by using de-compaction of blocks C, D and E. This

changes the weight of edges between certain vertices as in below *De-compaction* does not change the total layout area.

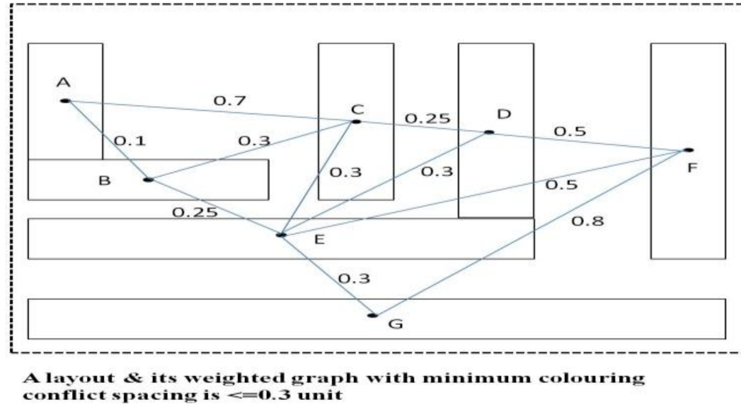


Figure 3-6: A weighted graph with its layout

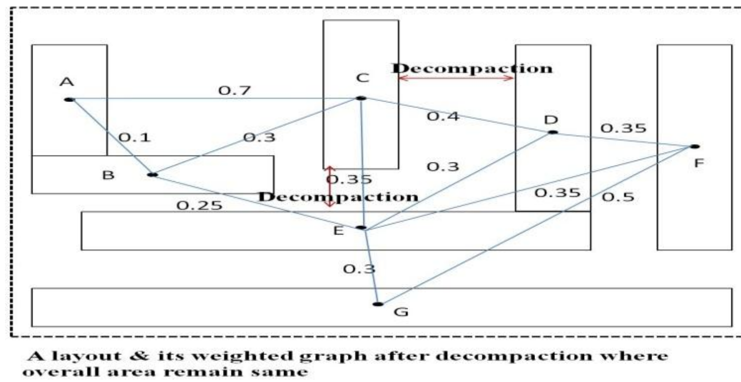


Figure 3-7: Removal of color conflict of a weighted layout.

3.4 Problem Formulation

Given: A layout L and the minimum allowed spacing *d* between two features.

Objective: To obtain a fracturing of *L* with appropriate assignment of the fractured features into two masks and the positions of the fractured features so that the combination of the number of fractured features of the input polygons and increase in total layout area is optimal.

Constraints: (i) two non-touching modules or fractured features f_i and f_j separated by $d_{i,j}$ $d_{i,j}$, $0 < d_{i,j} < d$ must be assigned to different masks, and (ii) two touching features with $d_{i,j} = 0$, are assigned the same mask.

In our formulation of the problem, we consider a graph *G* whose vertices represent the rectangular modules or rectangular fractures of rectilinear polygons of a given layout. There can be either of two types of edges between a pair of vertices.

Definition 6**:* If a pair of rectangular modules or fractured features f_i and f_j are separated by distance $d_{i,j}$, $0 < d_{i,j} < d$, their corresponding vertices in G have ***Type-1 edge, whereas if they are touching features with $d_{i,j} = 0$, then their corresponding vertices have a ***Type-2*** edge between them.

The *DPL* problem is then to assign each of two colors to the vertices of *G* such that every pair of vertices with *Type-1* edge between them do not have identical color, referred to as *coloring conflict*. Two vertices connected with a *Type*-2 edge may have identical or different colors. For convenience of algorithm design, two vertices connected with a Type-2 edge will also have a Type-1 edge between them. If two such vertices are assigned different colors it corresponds to fracturing of a rectilinear polygon in the layout, and the Type-2 edge between them is deleted. For 2-colorability of *G*, we usually merge the vertices. However, in certain complex cases, a rectangular polygon may have to be split in its body, which corresponds to splitting of vertex of the conflict graph. Splitting of a vertex may, in some cases, be not enough, as the layout pattern is such that it inserts new odd cycles. Layout decompaction is required in such cases.

3.5 Proposed Algorithm

We define following terms in the context of our proposed algorithm.

Definition 7. A *hard conflict* is the coloring conflict corresponding to features having minimum spacing below the critical distance but not touched with one another.

Definition 8. A *soft conflict* is the coloring conflict corresponding to two feature fragments touching each other.

Definition 9. In a conflict graph, a cycle is defined to be *constrained* when there is *only* hard conflict among its vertices.

Definition 10. An *unconstrained cycle* appears when there is at least one soft conflict among the vertices of the cycle.

Our proposed algorithm *DCLC* uses seven procedures FRACTURING (), LAYOUT-GRAPH (), MERGING (), H-V-CONSTRAINT-GRAPH-CONSTRUCTION(*G, G_c),* DE-COMPACT () and BI-COLOR-THE-GRAPH (), GRAPH-LAYOUT ().

Formal description of the algorithm is as follows:

Algorithm: DCLC

Input: A set of polygons *P* in a layout

Output: A conflict graph with all even cycles or no cycle

1. *P' = fracturing (P)* [4]   // *P'*: the set of rectangles //
2. *P'* is converted into conflict graph *G* where each rectangle is a vertex, and adjacent rectangles are connected by an edge by LAYOUT-GRAPH(*G*)
3. If there exists any odd cycle then MERGING (*G*)
4. If there still exists any odd cycle then
   // construct the horizontal (and, if required, vertical) constraint graph G_c
   H-V-CONSTRAINT-GRAPH-CONSTRUCTION(G, G_c)
   DE-COMPACT (G, G_c)
5. endif
6. BI-COLOR-THE-GRAPH (*G*) by Chitin's Algorithm [11]
7. Convert the graph *G* into layout by GRAPH-LAYOUT()

Procedure:DE-COMPACT (G,G_c)

Find all the planar graphs in the Conflict Graph (*G*) (considering both soft and hard conflicts) using depth first search traversal.

For every planar graph in graph (*G) d*o

Case-I: For a hard conflict horizontally

case(a): Move the vertices towards horizontally opposite direction of the vertices produce conflicts.

case(b): During movement towards horizontally, if the vertex faces any obstacle by neighboring vertices, then the neighbored vertices is moved either horizontally or vertically without affecting the total area of the layout.

Case-II: For a *hard conflict* vertically

case(a): Move the vertices towards vertically opposite direction of the vertices produce conflicts.

case(b): During movement towards vertically, if the vertex faces any obstacle by neighboring vertices, then the neighbored vertices is moved either horizontally or vertically without affecting the total area of the layout.

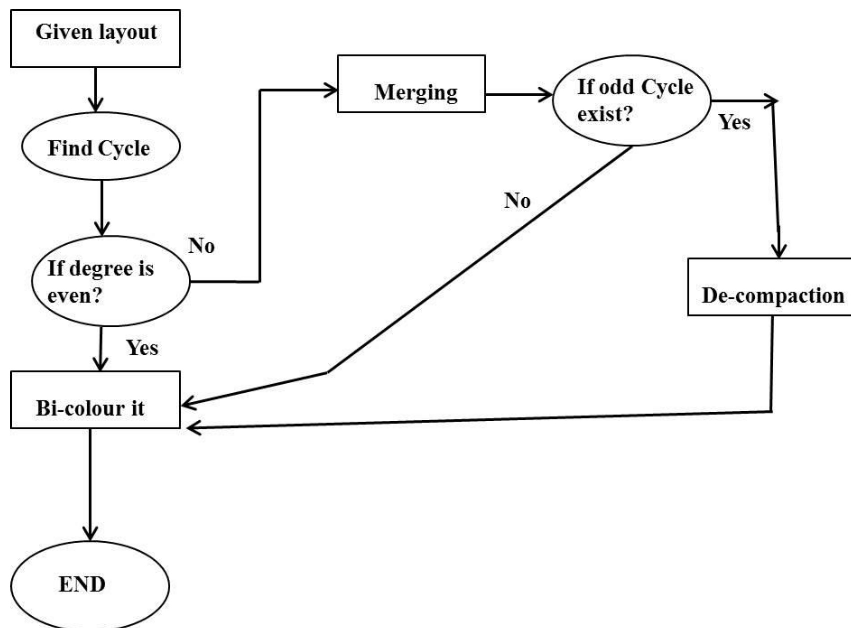
Case-III: : For a *hard conflict* vertically or horizontally, if the concerned vertex is unable to move vertically or horizontally due to obstacle by neighbored vertices, then total layout area may increase.

Update *G* and return by above steps

Procedure: MERGING (*P',G*)
Find all the cycles in the Conflict Graph *G* (considering both soft and hard conflicts) using depth-first search traversal. For every cycle in graph *G* do *Case-I* : For a disjoint *constrained odd cycle* delete an edge (DE- COMPACT(G)). *Case-II* : For a disjoint *constrained even cycle* go to return. *Case-III*: For a disjoint *unconstrained odd cycle* merge two vertices to reduce it to a constrained even cycle. *Case-IV*: For a disjoint *unconstrained even cycle* delete the *Type-II* edge for a pair of vertices having soft conflict (fracture of a rectilinear feature at concave corner). *Case-V*: If a *constrained odd cycle* is adjacent to a *Constrained even cycle* then delete the *Type-I* edge common to them (DE- COMPACT(*G, G_c*)). A new constrained odd cycle may be formed. *Case-VI*: If a *constrained odd cycle* is adjacent to an *unconstrained odd cycle* then delete the *Type-I* edge common to them (DE- COMPACT (G, G_c)). *Case-VII*: If a *constrained even cycle* is adjacent to an *unconstrained odd cycle* then simply merge the pair of vertices for an edge with a *Type-II* edge in the latter (apply merge). *Case-VII*: If a *constrained even cycle* is adjacent to an *unconstrained even cycle* then simply delete the *Type-II* edge for an edge with soft conflict (split). *Case-VIII*: If a *constrained odd cycle* is adjacent to an *unconstrained even cycle* then delete a *Type-I* edge of the former not shared by both (DE- COMPACT (*G, G_c*)). *Case-IX*: If a *constrained odd cycle* is adjacent to another *constrained odd cycle* then simply delete their common *Type*-II edge (DE- COMPACT (*G, G_c*)). Return.

Procedure: H-V-CONSTRAINT-GRAPH-CONSTRUCTION(*G, Gc*)
Begin Construct a weighted horizontal constraint graph, whose vertices represent the vertices of G and edges represent their adjacencies. Each edge is assigned a weight defined by the distance between them in the layout. End

The main algorithm DCLC can be depicted in simpler way as below.



The main algorithm DCLC shown above uses merging and de-compaction of polygons as these are sufficient to remove all the color conflicts between the polygons for DPL compliant layout.

The input to the proposed algorithm is a set of polygons which may or may not have concave corners. The proposed algorithm initially splits the entire polygon into rectangles. A graph is constructed based on the conflicts among the rectangles. The objective is to remove all the *hard* conflicts between the polygons. In its graph theoretic formulation, the objective is to remove all the odd cycles such that the output may have even cycles or no cycle.

It is observed that merging of rectangles with *soft conflict* only between them may cause removal of odd cycles, and de-compaction of the modules result in removal of odd cycles. Thus merging and de-compaction of vertices (corresponding to the polygons) are sufficient to attain the desired objectives.

3.6 Complexity Analysis

The time complexity of the proposed algorithm is O (n^2), where n is the number of given rectangular polygons.

The proposed algorithm uses the following procedures in sequence.

FRACTURING, LAYOUT-GRAPH, MERGING, H-V-CONSTRAINT-GRAPH-CONSTRUCTION, DE-COMPACT, BI-COLOR-THE-GRAPH and GRAPH-LAYOUT.

1. Time Complexity of the procedure FRACTURING is O(n)
2. Time Complexity of the procedure LAYOUT-GRAPH is O(n^2) assuming each polygon is at most L-shaped.
3. Time Complexity of the procedure MERGING is O($2n$)
4. Time Complexity of the procedure H-V-CONSTRAINT-GRAPH-CONSTRUCTION and DE- COMPACT is O($2n$)
5. Time Complexity of the procedure BI-COLOR-THE-GRAPH is O($2n$).
6. Time Complexity of the procedure GRAPH-LAYOUT is O(n^2) .

3.7 Examples

Example1: We consider a layout structure (looks like a comb) where all the features are rectangles (Figure 4-8(a)). The *conflict graph* of the structure is with odd degree cycle as in Figure 4-8(b). A color conflict in both vertical and horizontal direction occurs. So, next we merge two rectangles into an L-shape as in Figure 4-8(c).We draw conflict graph (Figure 4-8(d)), where we find two odd degree cycles. Introduction of an extra split does not resolve the problem. Additionally, presence of an extra stitch as well as overlay error will occur. Finally, we are compelled to apply de-compaction which resolves the problem with decreased number of stitches. A rectangular feature is de-compacted (Figure 4-8(e)). The conflict graph is drawn as in Figure 4-8(f) which shows even cycle i.e. DPL is possible. So, De-compaction along with split-merge technique is sufficient to obtain a DPL compliant layout.

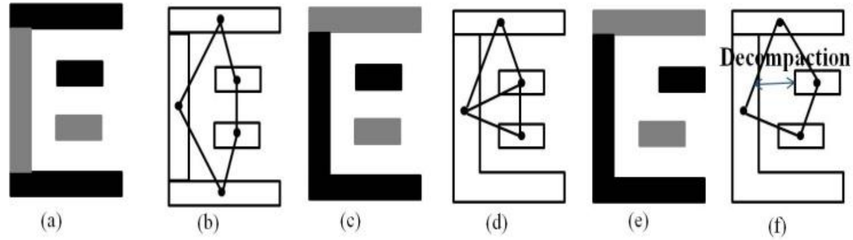


Figure 3-8: Example of removal of conflicts using merging and de-compaction.

Example 2: In case of a **GSRC** benchmark [12] with 10 modules, where we find two odd degree cycles among the modules (1,2,4,5,6) and (4,6,7) as in Figure 4-9. We first fracture module 7 and merge the modules (1,2) into L-shape as in Figure 4-10(a). The problem of DPL is resolved but with an extra stitch. We next, merge the rectangular modules (1,2) and (9,10) into L shape. Next we de-compact the module 7. The conflict graph now shows even degree cycle among the modules(1,3,7,9,6,4) as in Figure 4-10(b).This does not increase the total layout area without any extra stitch and we obtain the DPL compliant layout.

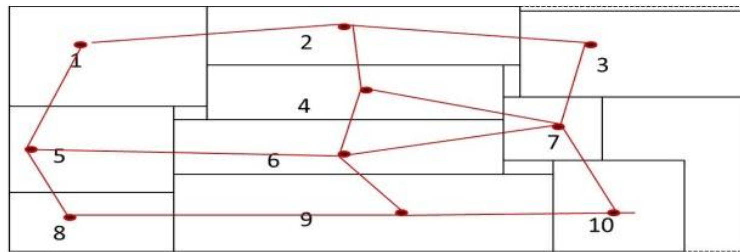
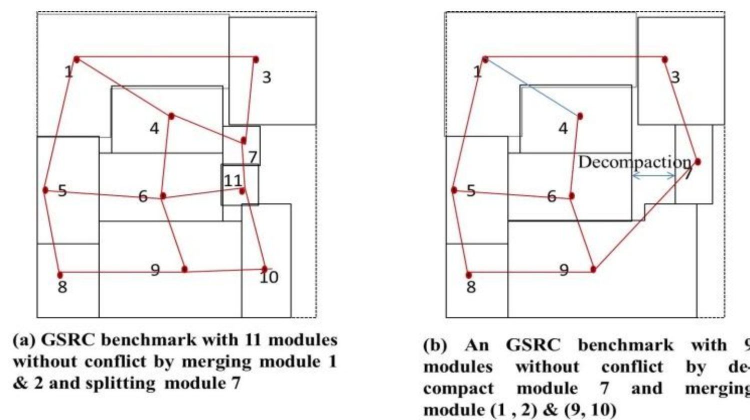


Figure 3-9: An GSRC benchmark with conflict among modules (1,2,4,6,5) & (3,4,6 ,7)



(a) GSRC benchmark with 11 modules without conflict by merging module 1 & 2 and splitting module 7

(b) An GSRC benchmark with 9 modules without conflict by de-compact module 7 and merging module (1 , 2) & (9, 10)

Figure 3-10: Removal of color conflicts in GSRC benchmark circuit.

Example 3: In case of **n10c** benchmark as in Figure 4-11(a), if we want to get DPL-compliant layout, the solution would not be obtained by using merge operator as it introduces more number of odd degree cycles among modules along with number of stitches and creates further complicacy. We obtain a solution by merging modules 7 and 8 to form L-shape and then by de-compaction of the L-shaped module and module 6 as in Figure 4-11(b).

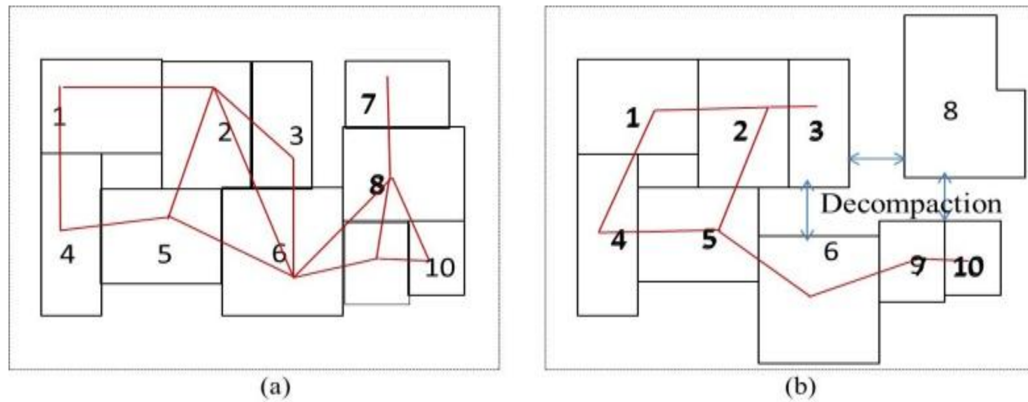


Figure 3-11: An n10c benchmark circuit without color conflict by merges and de-compaction

3.8 Experimental Results

We test our approach on ten (10) benchmark designs and one artificial design marked as X* (Figure 4-8(a)). Our test results are not comparable with any other work as existing results on these bench marks are not found. Results are summarized in Table I. The results are evaluated based on (1) number of rectangles present in the benchmark layout (2) number of stitches present in the benchmark layout (3) number of L-shaped features present in DPL compliant layout (4) number of rectangles present in DPL compliant layout (5) number of stitches present in DPL compliant layout (6) percentage of reduction in number of stitch (7) whether de-compaction is required or not and (8) overall change in layout area in percentage. The experimental data in Table I show that *decompaction* does not change the overall layout area

Table 1: Comparison of parameters of different benchmark layouts, before and after applying De-Compaction

BENCHMARKS	*Number of rectangles present in the*		*Number of stitches present in the*		*Percentage of reduction in number of stitches*	*Number of L shaped features present in DPL compliant layout*	*Whether de-compaction is required or not*	*Overall change in layout area in percentage*
	Benchmark	**DPL compliant layout**	**Benchmark**	**DPL compliant layout**				
n10a	**10**	**9**	**14**	**12**	**14.29**	**1**	**N**	**0.00**
Hp	**11**	**9**	**16**	**12**	**25.00**	**2**	**Y**	**0.00**
Apte	**9**	**8**	**13**	**8**	**38.46**	**1**	**Y**	**0.00**
n10b	**10**	**8**	**13**	**10**	**23.07**	**2**	**Y**	**0.00**
n10c	**10**	**9**	**14**	**10**	**28.57**	**1**	**Y**	**0.00**
GSRC	**10**	**9**	**13**	**8**	**38.46**	**1**	**Y**	**0.00**
Xerox	**10**	**10**	**13**	**9**	**30.77**	**0**	**Y**	**0.00**
n30a	**30**	**24**	**39**	**30**	**23.07**	**6**	**Y**	**0.00**
n30b	**30**	**28**	**43**	**38**	**11.90**	**2**	**Y**	**0.00**
Ami33	**34**	**31**	**53**	**47**	**11.32**	**3**	**Y**	**0.00**
X*	**5**	**1**	**2**	**1**	**50.00**	**1**	**Y**	**0.00**

So, if a layout can afford decompaction, this heuristics may be used to reduce the number of stitches. We considered that the outer modules of a layout have a scope to move minimum conflict spacing to outward direction but within the layout area. We find below the case of **Apte** benchmark [13](see Figure 4-12) and stitch minimization by decompaction of the said benchmark (see Figure 4-13).

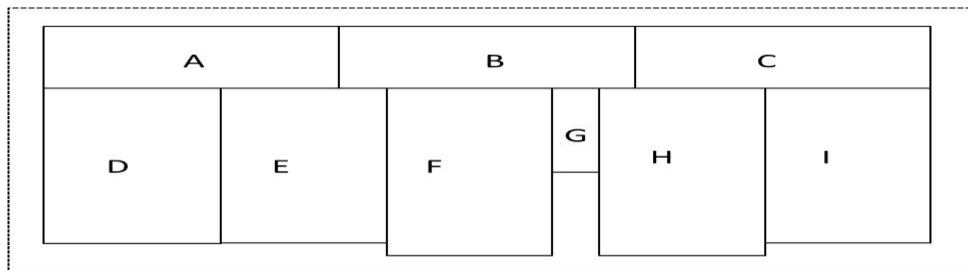


Figure 3-12: Apte benchmark

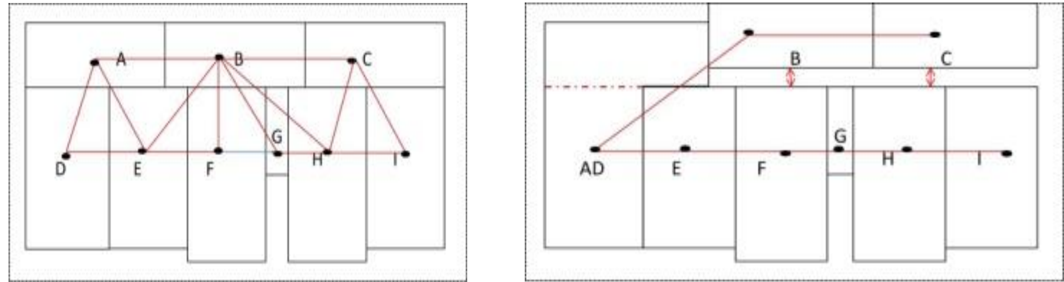


Figure 3-13: Removal of conflicts by decompaction of Apte benchmark

3.9 Observation

Following observations are clear from the description of the proposed algorithm:

Observation 1. All the fracturing of rectilinear polygons in the output DPL-complaint layout are at the concave corners only.

Observation 2. *De-Compaction* does not result in the use of any additional crossing of routing paths as the blocks are shifted either in horizontally or vertically but not in both directions simultaneously.

Chapter-4

An Approach to Construct a Triple Patterning Lithography Compliant Layout Using Area-Stitch Tradeoff

4.1 Introduction

As per conventional lithography, manufacturing of VLSI chips in the sub-wavelength technology has become extremely challenging. The difficulty to print sub-30 nm half-pitch patterns with stepper size of 193 nm is the major problem. Immersion lithography that uses high numerical aperture (NA) systems (high index fluids) is one of the possible solutions in these cases. But it is difficult to find new liquid material to increase numerical aperture (NA) beyond 1.35. Extreme ultra- violet (EUV) lithography with illumination source of 13.5 nm may be the alternative remedy. Again, this too has number of limitations owing to the lack of power sources, resists, and defect free masks.

Multi-patterning in general is well-researched techniques in lithography, due to their capability of achieving better printability. Instead of two colors if we are allowed to use three colors, we may have more flexibility and better printability. This method of lithography technique is known as Triple Patterning Lithography (TPL). Obviously, it minimizes the stitches in a layout as compared to DPL. In case of TPL, the total layout is partitioned into three different masks. But, not all conflicts between all the features can necessarily be resolved. *Merging* of two features and/or decompaction of one or more features may be the possible way to resolve the conflicts. The problem of stitch minimization for a TPL compliant layout has been recently considered in some works. The use of layout modification for the TPL problem also has been suggested in several papers. However, most of these existing works consider stitch minimization and conflict

minimization by modeling the conflicts as graphs with vertices representing the modules in the layout and edges between them representing their mutual conflicts in terms of the separation between the modules. A TPL-compliant layout having minimum stitches are constructed in this dissertation also.

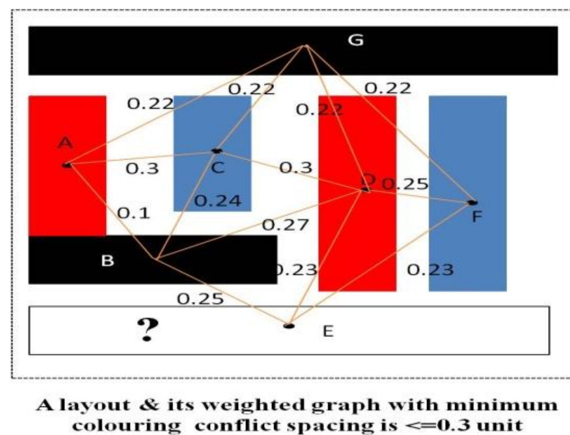
4.2 Preliminaries and Background

The stitch minimization problem for DPL has been considered in several existing works. Graph-theoretic formulation of the problem and its solutions appear in some papers. The layout modification for the DPL problem has been suggested in several papers.

We introduce certain concepts used in our proposed algorithm.

Conflict Graph in TPL: A *conflict graph* is defined by a graph which is drawn by connecting the rectangular features according to a minimum spacing rule, where a vertex of the graph corresponds to a rectangular feature.

For an example as shown in Figure 5-1, we consider a layout with 7 rectangular features A,B,C,D,E,F,G in which weights of the edges are defined as the minimum distance between the features (i.e. midpoint of each blocks) and the minimum coloring conflict spacing is less than or equal to 0.3 unit. The weight of the edges AB,AC,AG,BC,BD,BE,CD,CG,DE,DF,DG,EF,FG are 0.1,0.3,0.22,0.24,0.27,0.25,0.3, 0.22, 0.23,0.25,0.22,0.23,0.22 respectively. We draw a weighted conflict graph of the said layout. The graph shows conflicts occur among the blocks B, D, E and F. The block E is undetermined about its color.



A layout & its weighted graph with minimum colouring conflict spacing is <=0.3 unit

Figure 4-1: A weighted conflict graph with its layout having color conflict

Coloring Conflict in TPL—As already mentioned, if the distance between two separate features is less than minimum coloring spacing, they should be assigned to different masks (colors). Otherwise, there will be a coloring conflict.

Figure 5-2(a) shows a layout with six features. In TPL, any three neighbors of a feature are required to have different colors as the features are less than the minimum spacing rule with each other. So, a coloring conflict is unavoidable as shown in Figure 5-2(b). The conflict may be removed by splitting *the* right features as shown in Figure 5-2(c).

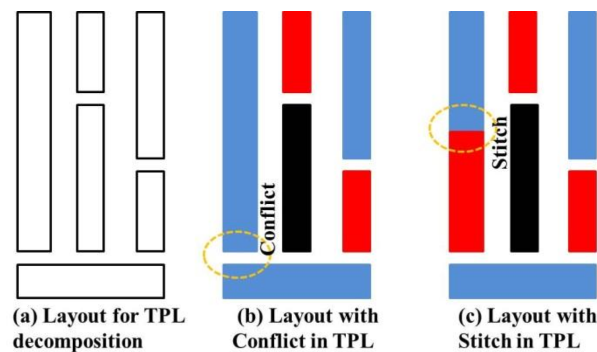


Figure 4-2: Issues related to TPL-layout decomposition.

Splitting Stitch in TPL--The stitch exists when two touched features are assigned to different masks. The stitch can be inserted to split some features to resolve the conflict as indicated Figure 5-2(c). However, stitch can have negative effects such as generation of another conflict, yield loss due to overlay error, etc. and thus increase of the manufacturing cost. Without altering layout, the traditional objective of layout decomposition is to minimize the unresolved conflicts by introducing as few stitches as possible.

Minimization of Stitches- See Chapter-3

In this dissertation, our proposed method attempts to achieve an optimal number of stitches using TPL. Our method may increase total area layout in some special cases.

4.3 Problem Formulation

Input: *L* is a layout in which the minimum allowed spacing between two features is *d*.

Objective: To obtain a fracturing of *L* with appropriate assignment of the fractured features to three masks and positions of the fractured features so that the combination of the number of fractures of the input features and the increase in total layout area is optimal.

Constraints: (i) two non-touching modules or fractured features *fi* and *fj* separated by *di,j*, 0 < *di,j* < *d* must be assigned to different masks (ii) two touching features with *di,j* = 0, are assigned the same mask and (iii) In some cases, we may not achieve the decrease in number of stitches without increasing area. In these cases, we have to trade-off area and stitches.

In our formulation of the problem, we consider a graph G whose vertices represent the rectangular modules or rectangular fractures of rectilinear polygons of a given layout.

4.4 Proposed Techniques

We use a technique named as *De-Compaction* to remove the color conflict in a TPL layout. The method is discussed below.

De-Compaction— In certain situations, conflicts may occur in both vertical and horizontal directions, which cannot be resolved by the use of splits. To resolve such conflicts, some blocks may have to be moved in either vertical or horizontal direction or both simultaneously. This is called *de- compaction*.

In Figure 5-3, we have taken a layout having four features which are separated with each other by less than $\mathbf{min_{CS}}$. So, if features1,3,4 are colored by three different color, then, obviously, any color at feature2 produces color conflict. The TPL coloring conflicts can be removed by using *de- compaction* i.e. by moving the feature2 beyond $\mathbf{min_{cs}}$ distance opposite direction from feature1 or feature3. It results no conflict with feature1 & 3. Hence feature2 can be colored by the color of feature1 or feature3.

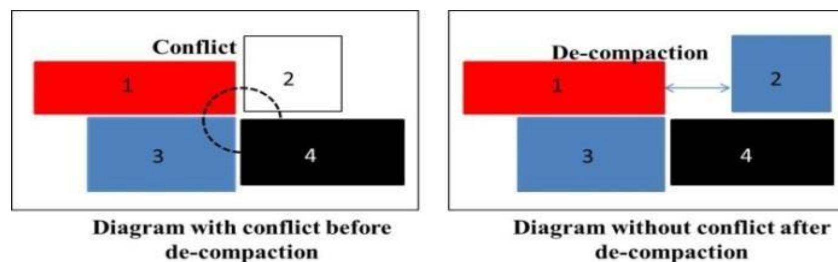


Figure 4-3: An example of De-compaction

We considered a layout with its weighted conflict graph as an example earlier as shown in Figure 5-1. In our example; the conflict occurred among the blocks B, D, E and F. We resolve the conflict by using de-compaction only one block, E.

De- compaction does not change the total layout area. This changes the weight of edges between the corresponding vertices. We can see all the detail in Figure 5-4.

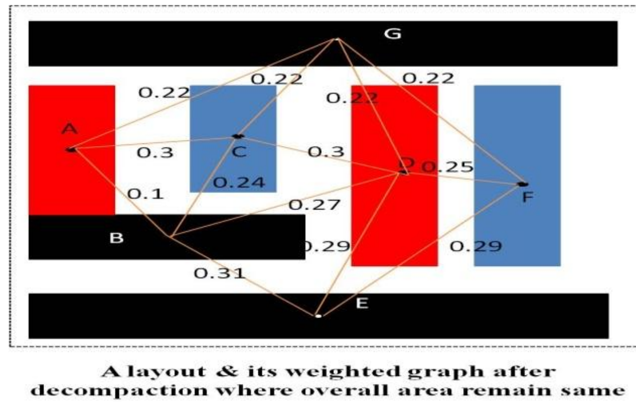


Figure 4-4: Removal of color conflict of a weighted layout.

4.5 Examples

Example1: We consider another layout structure with 11 modules, marked by X1 as in Figure 5-5(a), as an example where we find module-6 faces TPL color conflict with the modules (2,3,7). The conflicts of TPL coloring is resolved by de- compaction of the module-6 vertically downward with a spacing more than minimum color spacing (min_{CS}) distance without affecting layout area as in Figure 5-5(b).This does not increase the total layout area without any extra stitch to obtain the TPL compliant layout.

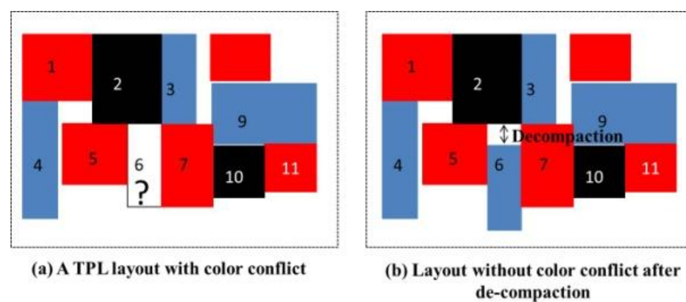


Figure 4-5: A layout with 11 modules with conflict and its removal by decompaction

We also use the merge techniques in our approach which is shown by an example below.

Example2: We consider a layout structure with its conflict *graph* (corresponding to TPL coloring) where all the features are rectangles. This is marked by X2 and shown in Figure

5-6(a). A color conflict occurs among the features (2,4,5,6) . So, next we merge features (1,2) into a L-shape as in Figure 5-6(b). So, by merging easily we can resolve the conflicts.

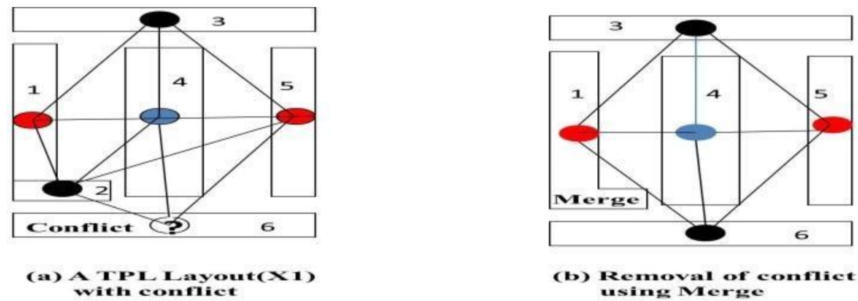


Figure 4-6: An example of removal of color conflict using merging.

Sometimes, we may have to use both decompaction and merging as given in the following example.

Example 3: Consider another layout structure with 8 modules, marked by X3, as an example as in Figure 5-7(a), where we find the TPL color conflict at module-6 with the modules (2,3,4,5,7,8). To remove the TPL color conflict we first merge the modules (6,8). But still there exist coloring conflict between the modules (5,6) Which is shown in Figure 5-7(b). Now, problem of TPL coloring is resolved by de-compaction of the module-5 horizontally to left direction more than minimum color spacing distance. This is shown in Figure 5-7(c). It does not affect the total layout area. Finally we obtain the TPL compliant layout without any insertion of stitches by both merging and de-compaction.

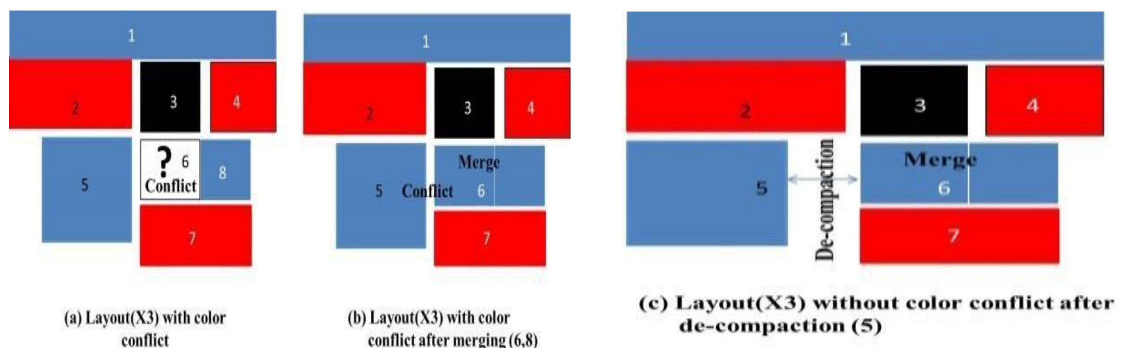


Figure 4-7: A TPL layout with color conflict & its removal by merging and De-compaction.

We first fracture the total layout features into rectilinear shapes. Then the features are converted into graph. The algorithm searches the soft and hard conflicts. The vertices with

soft conflicts are merged and vertices with hard conflicts are separated (*de-compacted)* towards horizontal or vertical direction depending on horizontal /vertical constraints graph. Then the vertices are tri-colored and converted to layout from graph.

4.6 Proposed Algorithm

Formal description of the algorithm as named ***TPLCL*** is as follows:

Our proposed algorithm ***TPLCL*** uses seven procedures- (i) FRACTURING () : Fractures the total layout into rectilinear feature, (ii) LAYOUT-GRAPH(): Converts layout to graph, (iii) MERGING(): Merges the vertices, (iv) H-V-CONSTRAINT-GRAPH-CONSTRUCTION(*G,Gc)*:Construct horizontal-vertical graph (v) DE-COMPACT(): Moves the vertices (vi) TRI-COLOR-THE-GRAPH(): Makes Tricolor the graph, (vii) GRAPH-LAYOUT(): Construct graph to layout.

Algorithm: TPLCL

Input: A set of polygons *P* in a layout

Output: A conflict graph with all even cycles or no cycle

---

1. *P' = fracturing (P)* [4] // *P'*: the set of rectangles //
2. *P'* is converted into conflict graph *G* where each rectangle is a vertex, and adjacent rectangles are connected by an edge by LAYOUT-GRAPH(*G*)
3. If there exists any planar graph among more than three vertices then MERGING (*G*) the vertices having *soft conflicts*
4. If still there exists any planar graph among more than three vertices then
// construct the horizontal (and, if required, vertical)
constraint graph *Gc*
H-V-CONSTRAINT-GRAPH-CONSTRUCTION (*G,Gc)*
DE-COMPACT (*G, Gc*)
5. End if
6. TRI-COLOR-THE-GRAPH (*G*) [16]
7. Convert the graph *G* into layout by GRAPH-LAYOUT()

The *procedures* used in the algorithm above are described in the *Chapter-4.*

It is sufficient to use merging and de-compaction of polygons to remove all the conflicts between polygons for TPL compliant layout also.

4.7 Complexity Analysis

The time complexity of the proposed algorithm is O (n^2), where n is the number of given rectangular polygons.

The proposed algorithm uses the following procedures in sequence.

FRACTURING, LAYOUT-GRAPH, MERGING, H-V-CONSTRAINT-GRAPH-ONSTRUCTION, DE-COMPACT, BI-COLOR-THE-GRAPH and GRAPH-LAYOUT.

1. Time Complexity of the procedure FRACTURING is O(n)
2. Time Complexity of the procedure LAYOUT-GRAPH is O(n^2) assuming each polygon is at most L-shaped.
3. Time Complexity of the procedure MERGING is O($2n$)
4. Time Complexity of the procedure H-V-CONSTRAINT-GRAPH-CONSTRUCTION and DE- COMPACT is O($2n$)
5. Time Complexity of the procedure BI-COLOR-THE- GRAPH is O($2n$).
6. Time Complexity of the procedure GRAPH-LAYOUT is O(n^2) .

4.8 Experimental Results

We find below a GSRC benchmark with 10 modules as in Figure 5-8(a). The TPL coloring conflict between modules (4,5) is avoided by splitting module 5 into two modules (5,11)which in turns produces one extra stitch as shown in Figure 5-8(b). The said conflict problem also may be resolved by using de-compaction horizontally towards left direction without any stitch as in Figure 5-8(c).

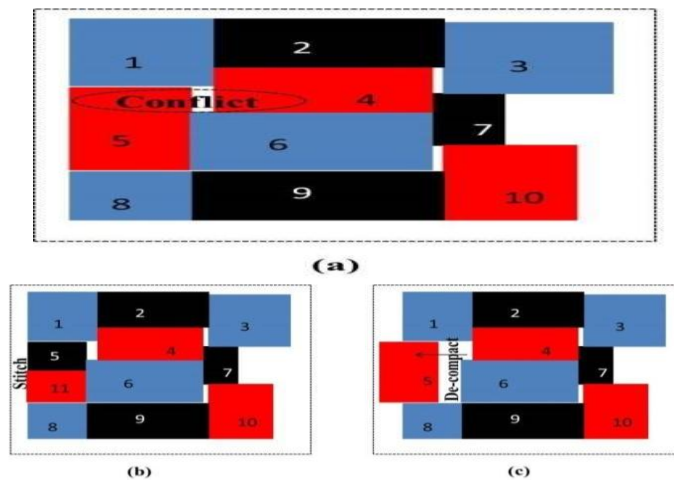


Figure 4-8 : TPL coloring conflict and its removal by De- compaction of a GSRC benchmark

We test our approach on four benchmark designs and three assumed designs in our example marked as X1,X2,X3 (Figure 5-5, Figure 5-6,Figure 5-7) Our test results are not comparable with any other work as existing results on these bench marks are not found. Results are summarized in Table I. The results are evaluated based on (1) number of rectangles present in the benchmark layout (2) number of stitches present in the benchmark layout (3) number of L-shaped features present in TPL compliant layout (4) number of rectangles present in TPL compliant layout (5) number of stitches present in TPL compliant layout (6) percentage of reduction in number of stitches (7) whether merge is required or not (8) whether de-compaction is required or not & (9) overall change in layout area in percentage.

Table 2: Comparison of parameters of different benchmark layouts, before and after applying De-compaction

Benchmarks	***Number of rectangles present in the***		***Number of stitches present in the***		***Percentage of reduction in number of stitches***	***Whether merging is required or not***	***Number of conflicts present in***		***Whether de-compaction is required or not***	***Overall change in layout area in percentage***
	Benchmark	***TPL compliant layout***	**Benchmark**	***TPL compliant layout***			**Benchmark**	***TPL compliant layout***		
GSRC	**10**	**10**	**13**	**11**	**15.40**	**N**	**1**	**0**	**Y**	**0.00**
n30a	**30**	**30**	**43**	**42**	**02.30**	**N**	**4**	**0**	**Y**	**0.00**
n30b	**32**	**32**	**47**	**47**	**00.00**	**Y**	**5**	**0**	**Y**	**0.00**
Ami33	**34**	**34**	**51**	**49**	**04.00**	**N**	**4**	**0**	**Y**	**0.00**
X1	**11**	**11**	**12**	**11**	**08.30**	**N**	**1**	**0**	**Y**	**0.00**
X2	**6**	**5**	**1**	**1**	**00.00**	**Y**	**1**	**0**	**N**	**0.00**
X3	**8**	**7**	**2**	**1**	**50.00**	**Y**	**1**	**0**	**Y**	**20.00**

So, if a layout can afford decompaction (i.e. the outer modules of a layout have a scope to move beyond minimum conflict distance to outward direction without affecting layout boundary and there is sufficient space to move the features), it may be used to reduce the number of stitches. For the layout X3, the total area increases by 20%, but the stitches reduce by 50%.

4.9 Observation

Following observations are clear from the description of the examples.

Observation 1. All the fracturing of rectilinear polygons in the output TPL-complaint layout is at the concave corners only.

Observation 2. *De-Compaction* does not result in the use of any additional crossing of routing paths as the blocks are shifted either in horizontally or vertically but not in both directions simultaneously.

Chapter-5

Conclusion and Future Work

5.1 Conclusion

In this dissertation, I propose and attempt to use a combination of stitches and de-compaction of features to obtain a DPL-compliant layout and TPL-compliant layout for double exposure patterning at 45nm and below. The proposed method uses a graph-theoretic model. Two graph-based operators, namely, merging of vertex-pairs and selective deletion of edges have been used. Horizontal and vertical constraint graphs are used for de-compaction, and a trade-off may be obtained between area-overhead during de-compaction and stitches during fracturing of features. It may be used to reduce the number of stitches.

These approaches practically and effectively improve lithography yield. Experimental results with benchmark design and artificial test cases show that with all necessary figures designed and provided, confirming the effectiveness of the said approaches.

5.2 Future Work

There are lot of scope for future works. Our ongoing research is in the following directions.

- The work can be extended to Quadruple Patterning Lithography (QPL)-Compliant layout.
- The work can be extended by using a convex combination of number of stitches and area-overhead.
- Gradually the work can be extended to Multiple patterning Lithography (MPL)-compliant layout.

Chapter-7

Bibliography

[1] Sandip Kundu, Aswin Sreedhar, *Nanoscale CMOS VLSI Circuits*, McGraw Hill Professional , 2010, ISBN: 0071635203, 9780071635202, Length:316 pages

[2] A. B. Kahng, X. Xu and A. Zelikovsky, "Fast Yield-Driven Fracture for Variable Shaped-Beam Mask Writing", *Proc. SPIE Conf. on Photomask and Next-Generation Lithography Mask Technology,* 2006, pp. 62832R-1 - 62832R-9.

[3] Kun Yuan, Jae-seok Yang, Pan, D.Z, " Double Patterning Layout Decomposition for Simultaneous Conflict and Stitch Minimization," *Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions* ,2010, Volume: 29, pp. 185 – 196.

[4] Bei Yu, Jhih-Rong Gao, Pan, D.Z. "L-Shaped based Layout Fracturing for E-Beam Lithography," *Proc. of The 18th Asia and South Pacific Design Automation Conference,* 2013,pp. 249-254, 2013.

[5] David Z. Pan, Jae-Seok Yang , Kun Yuan and Minsik Cho,"CAD for Double Patterning Lithography," *International Conference on IC Design and Technology (ICICDT),* 2010, IEEE, DOI:10.1109/ICICDT. 2010.5510279.

[6] Andrew B. Kahng, Chul-Hong Park, Xu Xu, and Hailong Yao,"Layout Decomposition for Double Patterning Lithography,"*International Conference on Computer-Aided Design (ICCAD'08),* November 10-13,2008, DOI: 10.1145/1509456.1509563.

[7] David Z. Pan, Jae-seok Yang, Kun Yuan, Minsik Cho and Yongchan Ban, "Layout Optimization for Double Patterning Lithography," *IEEE 8th International Conference on ASIC(ASICON '09)*,DOI: 10.1109/ ASICON.2009.5351308, pp.726-729,2009.

[8] A. B. Kahng, Chul-Hong Park , Hailong Yao. "Layout decomposition approches for Double Patterning Lithography*," IEEE Transactions on Computer-Aided Design*, vol. 29 No.6 June 2010 page. 939.

[9] Burn J Lin, "Lithography till the end of Moore's law", *Proc. of the 2012 ACM Int'l Symposium on Physical Design*, pp 1-2, San Jose, USA.

[10] Andrew B. Kahng, Chul-Hong Park, Xu Xu, and Hailong Yao,"Layout Decomposition for Double Patterning Lithography," *International Conference on Computer-Aided Design (ICCAD),* November 10-13,2008, DOI: 10.1145/1509456.1509563.

[11] Bei Yu, Kun Yuan,Boyang Zhang,Duo Ding,David Z. Pan,"Layout decomposition for Triple Patterning Lithography," *Proceeding of the 2011ACM International Conference on computer-Aided design(ICCAD'11)*,pp. 1-8.

[12] Bei Yu, Jhih-Rong Gao,David Z. Pan,"Triple Patering Lithography(TPL) layput decomposition using End-Cutting," *Proceedings of SPIE - The International Society for Optical Engineering,* vol. 8684, June 2013, DOI: 10.1117/12.2011355.

[13] Bei Yu, Gilda Garreton,David Z. Pan," Layout Compliance for Triple Patering Lithography: An Iterative Approach " *Proceeding of SPIE, Photomask Technology 2014*, 923504 (8 October 2014), vol. 9235 DOI: 10.1117/12.2066034.

[14] Shao-Yun Fang,Yao-Wen Chang,Wei-Yu Chen,"A Novel layout Decomposition algorithm for Triple Paterning Lithography", *Proceeding of 49th ACM Design Automation Conference(DAC)*, 2012, pp. 1181-1186.

[15] Burn J Lin, "Lithography till the end of Moore's law", *Proceedings of the 2012 ACM International Symposium on Physical Design*, pp 1-2, San Jose, USA.

[16] H Ayanegui, A Chávez-Aragón, "A complete algorithm to solve graph coloring problem" *LA-NMR*, 2009 – Citeseer, pp. 107-117.

[17] http://www.cse.ucsc.edu/research/surf/GSRC/progress.html.

[18] http://www.cse.ucsc.edu/research/surf/GSRC/MCNC bench.html.

[19] http://www.google.com/patents/US8709684

[20] http://www.umsl.edu/~meap6d/Algorithm.html

Accepted Paper

Debasis Pal, Abir Pramanik, Parthasarathi Dasgupta, and Debesh Kumar Das, "Double Patterning Lithography(DPL)-Compliant Layout Construction(DCLC) with Area-Stitch Usage Tradeoff", *Proceedings of the 20th International Symposium on VLSI Design and Test (VDAT), 24th-27th* May, 2016.