A Unified Framework for Simultaneous Layout Decomposition and Mask Optimization

Yuzhe Ma\textsuperscript{1}, Jhih-Rong Gao\textsuperscript{2}, Jian Kuang\textsuperscript{2}, Jin Miao\textsuperscript{2}, Bei Yu\textsuperscript{1}

\textsuperscript{1}\textit{The Chinese University of Hong Kong}
\textsuperscript{2}\textit{Cadence Design Systems}
Outline

Introduction

Algorithms

Experimental Results

Conclusion
Outline

Introduction

Algorithms

Experimental Results

Conclusion
VLSI Chip Design Flow

System Specification
- Architectural Design
  - Functional Design and Logic Design (RTL)
    - Logic Synthesis
      - Physical Design
        - Physical Verification and Signoff
          - Fabrication
            - Packaging and Testing
              - Chip

Mask Synthesis
- Fill Optimization
- Layout Decomposition
  ...

Module test
input in[3];
...
endmodule

AND
OR
DRC
LVS
STA
Layout Decomposition (LD)

- **Conflict**: two features with the same color, while distance \(< d_{\text{min}}\)

![Diagram showing layout decomposition](image)

(a) LELE

(b) LELELE

**Problem Formulation**

**Input**: layout and \(d_{\text{min}}\)

**Output**: decomposed layout, minimizing conflict #
Mask Optimization (MO)

- The quality of printed image may be poor due to the diffraction effect of the light.
- **Optical Proximity Correction (OPC):** Refine the mask to compensate the diffraction effect.
- **Method for OPC:**
  - rule-based [Park+, ISQED’2010];
  - model-based [Kuang+, DATE’2015][Su+, TCAD’2016];
  - inverse lithography technique [Gao+, DAC’2014].
Mask Optimization (cont.)

- **Edge Placement Error (EPE):** Geometric displacement between the image contour and the edge of target image on the layout.
- **EPE Violation:** The perpendicular displacement is greater than an EPE threshold value.

```
Problem Formulation

Input: target layout
Output: refined mask, minimizing EPE violation #.
```
Two-Stage Flow for Layout Optimization

Two stages:

▶ Layout Decomposition (LD)
▶ Mask Optimization (MO)
Issues

Solution 1: \#EPE Violation = 1

Solution 2: \#EPE Violation = 3
Options?

- **Exhaustive MO** for all LD solutions.
  - Running time overhead due to thousands of LD solutions.
Options? (cont.)

- **Heuristic selection** among LD solutions.
  - Local region density [Yu+, ICCAD’13]: balance the pattern density on each mask.
  - Spacing vector [Chen+, ISQED’13]: maximize minimum distance between patterns.
  - Limited effectiveness.
How about combining **LD** and **MO** together?

- It is an open problem.
- It is expected to be more effective and more efficient.
Outline

Introduction

Algorithms

Experimental Results

Conclusion
Lithography model:
- The aerial image is formed by a series of convolution operation between mask $M$ and lithography kernel $h$.

$$ I = f_{optical}(M) = \sum_{k=1}^{K} w_k \cdot |M \otimes h_k|^2 $$

Photo-resist model
- Set a threshold $I_{th}$ to binarize aerial image.

$$ Z(x, y) = f_{resist}(I) = \begin{cases} 
1, & \text{if } I(x, y) \geq I_{th}, \\
0, & \text{otherwise}.
\end{cases} $$
Problem Formulation

**LDMO**: Given a target image $Z_t$, find two masks $M_1$ and $M_2$ which can form printed image with high fidelity.

\[
\begin{align*}
\min_{M_1, M_2} & \quad F = \|Z_t - Z\|_2^2 \\
\text{s.t.} & \quad M_1(x, y) \in \{0, 1\}, \quad \forall x, y, \\
& \quad M_2(x, y) \in \{0, 1\}, \quad \forall x, y, \\
& \quad I_1 = \sum_{k=1}^{K} w_k \cdot |M_1 \otimes h_k|^2, \\
& \quad I_2 = \sum_{k=1}^{K} w_k \cdot |M_2 \otimes h_k|^2, \\
& \quad Z = f_{\text{resist}}(I_1) \lor f_{\text{resist}}(I_2).
\end{align*}
\]
Overall Flow

Input Cell Layout

- Grid Construction
  - Numerical Layout Optimization
  - Discrete Layout Optimization

- SDP
- Pruning
- Gradient-based Mask Update
- Violation Detection
- Initialization

converge?

Output Optimized Masks
Overall Flow

- Input Cell Layout
- Grid Construction
  - Initialization
  - Gradient-based Mask Update
  - Violation Detection
  - SDP
  - Pruning
- Numerical Layout Optimization
- Discrete Layout Optimization
- Output Optimized Masks

converge? [Y] [N]
Grid Construction

- Extract target pattern.
- Add bounding box.
- Construct grid.
- Merge grid.

Pattern grid  Spacing grid  V  H  D  D  Vertical  Horizontal

Merged spacing grid
Overall Flow

Input Cell Layout

Grid Construction

Numerical Layout Optimization

Discrete Layout Optimization

Initialization

Gradient-based Mask Update

Violation Detection

SDP

Pruning

converge?

Output Optimized Masks

Y

N
Formulation Relaxation

- Relaxation on binary constraints with *sigmoid* function.

\[
M_1(x, y) \in \{0, 1\} \rightarrow M_1(x, y) = \text{sig}(P_1(x, y)) = \frac{1}{1 + \exp[-\theta_M P_1(x, y)]}
\]

\[
Z_1(x, y) = f_{\text{resist}}(I_1) \rightarrow Z_1(x, y) = \text{sig}(I_1(x, y)) = \frac{1}{1 + \exp[-\theta_Z (I_1(x, y) - I_{th})]}
\]

- Relaxation on \( Z \).

\[
Z = f_{\text{resist}}(I_1) \lor f_{\text{resist}}(I_2) \rightarrow Z(x, y) = \min\{Z_1(x, y) + Z_2(x, y), 1\}
\]
Algorithm 1 Gradient-Based Mask Update

1: function MaskUpdate($P_1, P_2$)
2:   Initialize stepsize $t$;
3:   Compute the relaxed masks $M_1, M_2$;
4:   Compute $Z$ according to current $P_1$ and $P_2$;
5:   Compute the gradient $\nabla_{P_1} F, \nabla_{P_2} F$
6:   $P_1 \leftarrow P_1 - t \times \nabla_{P_1} F$;
7:   $P_2 \leftarrow P_2 - t \times \nabla_{P_2} F$;
8:   return $P_1, P_2, \nabla_{P_1} F, \nabla_{P_2} F$;
9: end function
Overall Flow

Input Cell Layout

Grid Construction

Numerical Layout Optimization

Discrete Layout Optimization

Initialization

Gradient-based Mask Update

Violation Detection

SDP

Pruning

converge?

Output Optimized Masks

Y

N
Violation Graph

\[ w_{ij} = \begin{cases} 
1, & \text{if } v_i \text{ and } v_j \text{ have conflict,} \\
\beta, & \text{if } v_i \text{ and } v_j \text{ have large } \#EPEV, \\
0, & \text{otherwise.} 
\end{cases} \]

\[
W = \begin{bmatrix}
0 & 0 & 0 & 0 & \beta \\
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\beta & 0 & 0 & 1 & 0
\end{bmatrix}
\]
Semidefinite Programming

- Use $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$ to denote the grid assignment solution.
- Max-Cut:

$$\max_{x_i} \sum_{(i,j) \in E} w_{ij} (1 - x_i x_j)$$

s.t. $x_i \in \{-1, 1\}$, $\forall v_i \in V$

Relax to Semidefinite Programming:

$$\min_{\mathbf{X}} \mathbf{W} \cdot \mathbf{X}$$

s.t. $\text{diag}(\mathbf{X}) = \mathbf{e}$,

$\mathbf{X} \succeq 0$
Randomized rounding [Goemans+, JACM’1995]

- Obtain $X^*$ by solving SDP.
- Cholesky decomposition with $X^*$.

$$X^* = U^T U$$

- Get $x_i$ as follows. $u_i$ is the $i$-th column of $U$ and $r$ is random unit vector.

$$x_i = \text{sgn}(u_i^T r) = \begin{cases} 
1, & \text{if } u_i^T r \geq 0, \\
-1, & \text{otherwise}.
\end{cases}$$
Pruning

- Obtain multiple solutions by randomized rounding.

- Efficient pruning.
Outline

Introduction

Algorithms

Experimental Results

Conclusion
#EPE Violation Convergence Curve

![Graph showing #EPE violations vs #Iteration for OR2_X1, NAND4_X1, and BUF_X1](image-url)
Comparison – EPE Violation Num

(Number of EPE Violations)

- INV_X1
- NOR_X1
- BUF_X1
- OR2_X1
- AOI211_X1

- ENUM+[DAC’14]
- [ICCAD’13]+[DAC’14]
- [ISQED’13]+[DAC’14]
- Ours

Comparison Table:

<table>
<thead>
<tr>
<th>Component</th>
<th>ENUM+[DAC’14]</th>
<th>[ICCAD’13]+[DAC’14]</th>
<th>[ISQED’13]+[DAC’14]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV_X1</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>NOR_X1</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>BUF_X1</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>OR2_X1</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>AOI211_X1</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>
Comparison – Runtime

![Bar chart showing runtime comparison for different components: INV_X1, NOR_X1, BUF_X1, OR2_X1, AOI211_X1. The y-axis represents runtime in seconds (\(10^4\)) and the x-axis lists components. The chart compares runtime across different methods: ENUM+[DAC'14], [ICCAD'13]+[DAC'14], [ISQED'13]+[DAC'14], and Ours. Each component has bars indicating the runtime for each method, with the tallest bar for ENUM+[DAC'14] and the shortest for [ISQED'13]+[DAC'14].]
Distribution of #EPE violations

(a) BUF_X1

(b) NAND4_X1

(c) OR2_X1

Flow-2 [ICCAD’13] + [DAC’14];
Flow-3 [ISQED’13] + [DAC’14];
Examples of Printed Image

(a) [ICCAD’13] + [DAC’14];
(b) [ISQED’13] + [DAC’14];
(c) Ours.

BUF_X1
Examples of Printed Image

(a) [ICCAD'13] + [DAC'14];
(b) [ISQED'13] + [DAC'14];
(c) Ours.

BUF_X1

OR2_X1
Outline

Introduction

Algorithms

Experimental Results

Conclusion
Conclusion

- A unified framework is proposed for solving LDMO problem.
- Two collaborative flows are designed:
  - A gradient-based numerical optimization
  - A set of discrete optimization.
- Effectiveness and efficiency are verified.
Conclusion

▶ A unified framework is proposed for solving LDMO problem.

▶ Two collaborative flows are designed:
  ▶ A gradient-based numerical optimization
  ▶ A set of discrete optimization.

▶ Effectiveness and efficiency are verified.

Future Exploration

- More advanced lithography process, e.g., triple patterning lithography.
- More optimization targets, such as process variation band.
Thank You

Yuzhe Ma (yzma@cse.cuhk.edu.hk)
Jhih-Rong Gao (jgao@cadence.com)
Jian Kuang (jkuang@cadence.com)
Jin Miao (jmiao@cadence.com)
Bei Yu (byu@cse.cuhk.edu.hk)