Layout Decomposition for Quadruple Patterning Lithography and Beyond

Bei Yu, David Z. Pan
ECE Department, University of Texas at Austin, Austin, USA

Introduction
- Natural extension of triple patterning lithography (TPL)
- But with one more mask

Why QPL?
- Delay of EUVL
- Research perspective: need to be prepared
- Resolve native conflict from triple patterning

Problem Formulation
Input:
- Input layout patterns
- Minimum coloring distance $m_{\min}$

Output:
- Decomposed layout
- The conflict number & the stitch number

Overall Flow

SDP based Color Assignment

Vector based Color Representation
- Four vectors
  - same color: $\vec{v}, \vec{v} = 1$
  - different color: $\vec{v}, \vec{v} = -1/3$

Vector Programming:
\[
\begin{align*}
\min & \sum_{i < j, (i,j) \in E} \frac{3}{2} \vec{v}_i \cdot \vec{v}_j + \frac{3}{2} \sum_{i \in V} (1 - \vec{v}_i) \\
\text{s.t.} & \quad \vec{v}_i \in \{0, 1\} \cup \{-1, 1/3\} \\
& \quad \vec{v}_i \cdot \vec{v}_j = 1 \\
& \quad \forall \vec{v}_i, \vec{v}_j \in \vec{v} \in CE
\end{align*}
\]

Relax to Semidefinite Programming (SDP)
\[
\begin{align*}
\min & \sum_{i < j, (i,j) \in E} \frac{3}{2} \vec{v}_i \cdot \vec{v}_j + \frac{3}{2} \sum_{i \in V} (1 - \vec{v}_i) \\
\text{s.t.} & \quad \vec{v}_i \cdot \vec{v}_j = 1 \\
& \quad \forall \vec{v}_i, \vec{v}_j \in \vec{v} \in CE
\end{align*}
\]

Mapping: Continuous solutions to discrete $\vec{v}, \vec{v}$

Algorithm: Backtrack based mapping

Require: SDP solution $x_i$, threshold value $\epsilon_{x_i}$
1. for all $i, j \in V$ do
2. Combine vertices $\vec{v}_i$, $\vec{v}_j$ into one larger vertex;
3. end for
4. Construct merged graph $G = (V', C', E')$;
5. BACKTRACK($G$);
6. return color assignment result in $G$;
7. function BACKTRACK($G$)
8. if $t > 0$ then
9. if Find a better color assignment then
10. Store current color assignment;
11. end if
12. else
13. for all legal color $c$ do
14. $G[e] = c$
15. BACKTRACK($G$ + 1)
16. $G[e] = -1$
17. end for
18. end if
19. end function

GH-Tree based 3-Cut Removal

An example of 3-Cut Detection and Removal:

Algorithm: GH-tree based 3-Cut Removal

Require: Decomposition graph $G = (V, C, E)$
1. Construct GH-tree;
2. Remove the edges with weight $> 4$;
3. Compute connected components on remaining GH-tree;
4. for each component do
5. Color assignment on this component;
6. end for
7. Color rotation to interconnect all components;

Conclusion
- First layout decomposition framework for Quadruple Patterning and Beyond
- Our algorithm is effective and efficient to obtain high quality solution
- MPL may be a promising manufacturing solution for sub-10nm technology node
- Facilitating the advancement of MPL technology

Acknowledgements
- The authors would like to thank Tsung-Wei Huang for helpful discussions.
- This work is supported in part by Oracle, NSF, and NSFRC.