GPU-ACCELERATED INVERSE LITHOGRAPHY TOWARDS HIGH QUALITY CURVY MASK GENERATION

HAOYU YANG & HAOXING REN / ISPD'2025

- Computational Lithography
  - Lithography Basis
  - Mask Optimization
- Methodology
  - Curvy Design Retargeting
  - MRC Aware ILT
- Experiments

COMPUTATIONAL LITHOGRAPHY

Lithography Basis and Hopkin's Diffraction Model

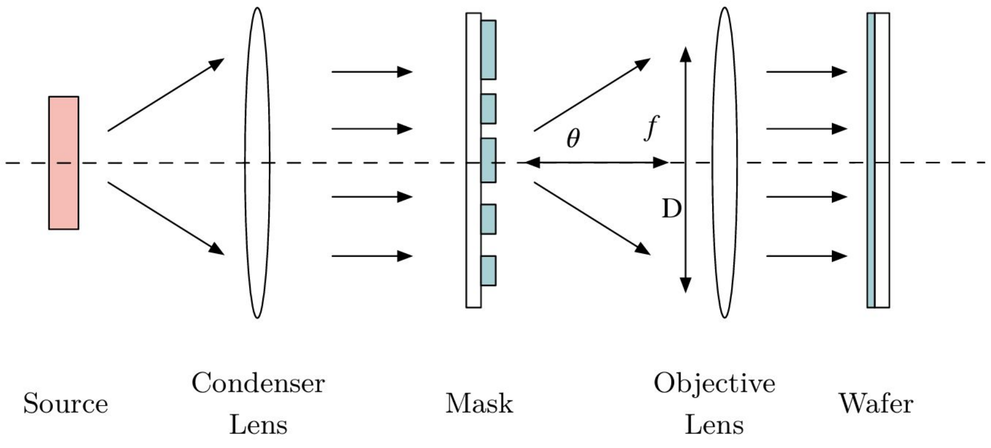
- Lithography
  - Transfers chip design pattern onto silicon wafer.
  - Key technology in chip manufacturing.
- Challenges
  - Extreme shrinking of design 14nm->7nm->N3->… is not aligned with lithography system DUV (193) / EUV (13.5). Making lithography a low pass filter.
  - Designs need to be optimized to compensate distortion.
- Lithography modeling
  - Numerically describe the lithography behavior.
  - Consider simple case with thin mask model.

$$\boldsymbol{I}(m,n) = \tilde{\boldsymbol{s}}^{\mathsf{H}} \boldsymbol{A} \tilde{\boldsymbol{s}}$$

$$\tilde{\boldsymbol{s}} = \tilde{\boldsymbol{M}}(p,q) e^{j(2pm+2qn)\pi}$$

  - Taking SVD of A

$$\boldsymbol{I} = \sum_{i=1}^{k} \alpha_i ||\mathcal{F}^{-1}(\mathcal{M} \odot \mathcal{H}_i)||_2^2,$$



- Resist approximation

$$\boldsymbol{Z}(i,j) = \begin{cases} 1, & \text{if } \boldsymbol{I}(i,j) \geq D_{\text{th}}, \\ 0, & \text{if } \boldsymbol{I}(i,j) < D_{\text{th}}. \end{cases}$$

COMPUTATIONAL LITHOGRAPHY

Inverse Lithography and Mask Optimization

- Lithography and Inverse Lithography

$$I = \sum_{i=1}^{k} \alpha_i ||\mathcal{F}^{-1}(\mathcal{M} \odot \mathcal{H}_i)||_2^2, \quad (1)$$

$$Z(i,j) = \begin{cases} 1, & \text{if } I(i,j) \geq D_{\text{th}}, \\ 0, & \text{if } I(i,j) < D_{\text{th}}. \end{cases} \quad (2)$$

$$M_c = \frac{1}{1 + \exp[-\beta_1 (M - M_s)]}, \quad (3)$$

$$Z = \frac{1}{1 + \exp[-\beta_2 (I - D_{\text{th}})]}. \quad (4)$$

- ILT Objectives

$$\min_{M} f(M, Z^*),$$
$$\text{s.t. Equations (1), (3) and (4),}$$

- OPC vs ILT vs Curvy ILT

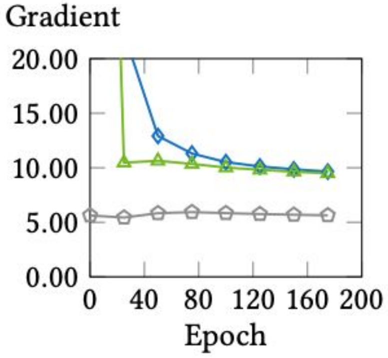
Solution	OPC	ILT	**Curvy ILT**
Mask Writer	Variable Shaped Beam	Variable Shaped Beam	Multibeam
Mask Rule	Manhattan Geometry Constraints	Manhattan Geometry Constraints	Width, Area, Curvature
Efficiency	Fast	Slow	Slow
Solution Space	Small	Medium	Large
Optimizer	Heurestic	Gradient	Gradient
Example			

MASK OPTIMIZATION AND ITS CHALLENGES

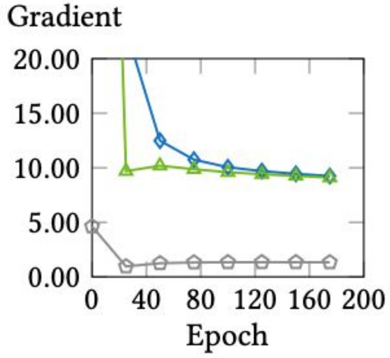
- Challenges in OPC
  - Manhattan constraints provides very low solution space, making the design less optimal.
- Challenges in ILT
  - Pixeled optimization target easily creates rule-violation artifacts.
  - Hard to honor mask rules.
  - Post processing loses optimality.
  - Manhattan design target is hard to achieve.
- Our solution
  - Consider mask rules during the optimization runtime.
  - Release non-achievable objectives for faster and better convergence.

CURVY DESIGN RETARGETING

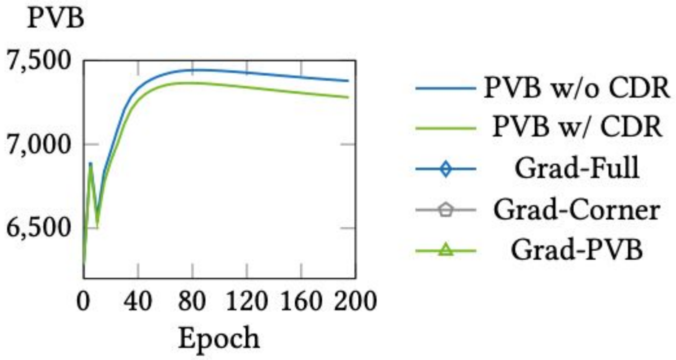
- Curvy Design Retargeting
  - Sharp corners in post routing design shapes cannot be manufactured, considering lithography is a low pass filter.
  - Over optimize these corners will sabotage process variations.



(a) w/o CDR Gradient



(b) w/ CDR Gradient



(c) PVB Loss

CURVY DESIGN RETARGETING

- Curvy Design Retargeting
  - Sharp corners in post routing design shapes cannot be manufactured, considering lithography is a low pass filter.
  - Over optimize these corners will sabotage process variations.
  - We leverage Morphological Operator
- Morphological Operator
  - Dilation
  - Erosion
  - Opening
  - Closing

$$f_e(\boldsymbol{A}, \boldsymbol{B}) = \boldsymbol{A} \ominus \boldsymbol{B}, \text{ (Erosion)}$$
$$f_d(\boldsymbol{A}, \boldsymbol{B}) = \boldsymbol{A} \oplus \boldsymbol{B}, \text{ (Dilation)}$$
$$f_o(\boldsymbol{A}, \boldsymbol{B}) = f_d(\boldsymbol{A}, \boldsymbol{B}) \ominus \boldsymbol{B}, \text{ (Opening)}$$
$$f_c(\boldsymbol{A}, \boldsymbol{B}) = f_e(\boldsymbol{A}, \boldsymbol{B}) \oplus \boldsymbol{B}, \text{ (Closing)}$$

(a) Reference (b) Dilation (c) Erosion

(d) Opening (e) Closing (f) CDR

CURVY DESIGN RETARGETING

- Curvy Design Retargeting
  - Sharp corners in post routing design shapes cannot be manufactured, considering lithography is a low pass filter.
  - Over optimize these corners will sabotage process variations.
  - We leverage Morphological Operator
- Morphological Operator
  - Dilation
  - Erosion
  - Opening
  - Closing

$$f_e(\boldsymbol{A}, \boldsymbol{B}) = \boldsymbol{A} \ominus \boldsymbol{B}, \text{ (Erosion)}$$
$$f_d(\boldsymbol{A}, \boldsymbol{B}) = \boldsymbol{A} \oplus \boldsymbol{B}, \text{ (Dilation)}$$
$$f_o(\boldsymbol{A}, \boldsymbol{B}) = f_d(\boldsymbol{A}, \boldsymbol{B}) \ominus \boldsymbol{B}, \text{ (Opening)}$$
$$f_c(\boldsymbol{A}, \boldsymbol{B}) = f_e(\boldsymbol{A}, \boldsymbol{B}) \oplus \boldsymbol{B}, \text{ (Closing)}$$

Algorithm 1 CDR

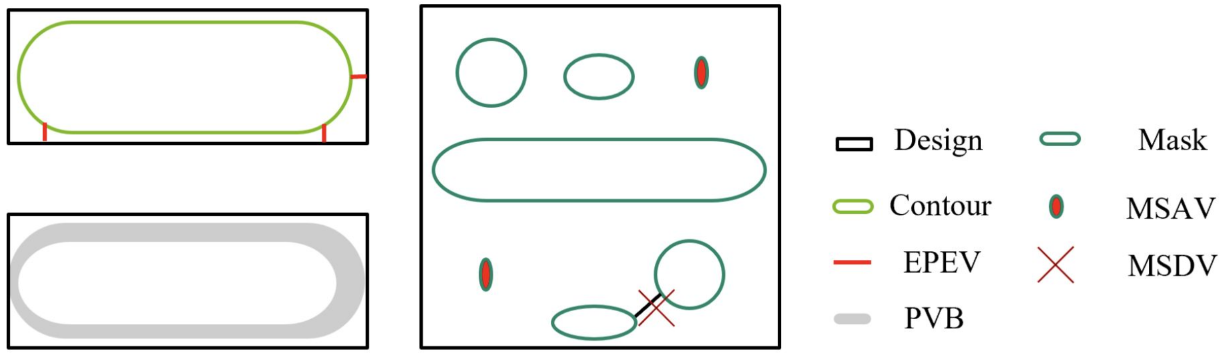
Input: Manhattan design $\boldsymbol{Z}^*$, convex corner smoothness coefficient k_{cvx}, concave corner smoothness coefficient k_{ccv}.

Output: Corner retargeted design.

1: $\boldsymbol{B}_{\text{cvx}} \leftarrow$ Disc-shaped structuring element with size k_{cvx};
2: $\boldsymbol{B}_{\text{ccv}} \leftarrow$ Disc-shaped structuring element with size k_{ccv};
3: $\boldsymbol{Z}^*_{\text{cvx}} \leftarrow \boldsymbol{Z}^* \oplus \boldsymbol{B}_{\text{cvx}} \ominus \boldsymbol{B}_{\text{cvx}}$;
4: $\boldsymbol{Z}^*_{\text{ccv}} \leftarrow \boldsymbol{Z}^* \ominus \boldsymbol{B}_{\text{ccv}} \oplus \boldsymbol{B}_{\text{ccv}}$;
5: $\boldsymbol{Z}^* \leftarrow \boldsymbol{Z}^*_{\text{cvx}} + \boldsymbol{Z}^*_{\text{ccv}} - \boldsymbol{Z}^*$.

MASK OPTIMIZATION QUALITY IN CURVY ERA

- QoR Evaluation
  - EPE
  - Process Viation
  - Mask Shape Area
  - Mask Shape Distance
  - Mask Smoothness



- Cost Function

EPE

Process Variation

$$f(\boldsymbol{M}, \boldsymbol{Z}^*) = ||Z_{\text{nom}} - \boldsymbol{Z}^*||_2^2 + ||Z_{\max} - Z_{\min}||_2^2 + \beta_3 ||\mathcal{F}(\boldsymbol{M})(k :; k :)||_2^2,$$

Mask Smoothness

MASK OPTIMIZATION QUALITY IN CURVY ERA

- QoR Evaluation
  - EPE
  - Process Viation
  - Mask Shape Area
  - Mask Shape Distance
  - Mask Smoothness

- Leverage Morphological Operator
  - Implemented in a differentiable manner
  - PlugnPlay for GPU-Accelerated ILT
  - Erosion removes small artifacts
  - Dilation merge close shapes

Algorithm 2 Differentiable Morphological Operator

function Dilation_Forward($\boldsymbol{A} \in \mathbb{R}^{N \times N}, \boldsymbol{B} \in \mathbb{R}^{k \times k}$)
    $\boldsymbol{A}^+ \leftarrow$ ZeroPadding($\boldsymbol{A}, \lfloor \frac{k}{2} \rfloor$);
    for each thread $i, j \in [0, N-1]$ **do**
        $\boldsymbol{A}(i, j) = \max(\boldsymbol{A}^+(i : i+k, j : j+k) \odot \boldsymbol{B})$;
    return $\boldsymbol{A}$.
function Erosion_Forward($\boldsymbol{A} \in \mathbb{R}^{N \times N}, \boldsymbol{B} \in \mathbb{R}^{k \times k}$)
    $\boldsymbol{A}^+ \leftarrow$ ZeroPadding($\boldsymbol{A}, \lfloor \frac{k}{2} \rfloor$);
    for each thread $i, j \in [0, N-1]$ **do**
        $\boldsymbol{A}(i, j) = \min(\boldsymbol{A}^+(i : i+k, j : j+k) \odot \boldsymbol{B})$;
    return $\boldsymbol{A}$.
function Opening_Forward($\boldsymbol{A} \in \mathbb{R}^{N \times N}, \boldsymbol{B} \in \mathbb{R}^{k \times k}$)
    $\boldsymbol{A} \leftarrow$ Dilation_Forward(Erosion_Forward($\boldsymbol{A}, \boldsymbol{B}$), $\boldsymbol{B}$);
    return $\boldsymbol{A}$.
function Closing_Forward($\boldsymbol{A} \in \mathbb{R}^{N \times N}, \boldsymbol{B} \in \mathbb{R}^{k \times k}$)
    $\boldsymbol{A} \leftarrow$ Erosion_Forward(Dilation_Forward($\boldsymbol{A}, \boldsymbol{B}$), $\boldsymbol{B}$);
    return $\boldsymbol{A}$.

EXPERIMENTS

Dataset

- ICCAD'13 CAD Contest
  - 10 Toy Examples with 10~20 shapes in each clip
- LithoBench (NeurIPS'23 Dataset and Benchmark)
  - https://github.com/shelljane/lithobench
  - A large collection of metal and via clips for ML purpose
  - We only take the testing clips extracted from standard cells.

Benchmark	Layer	Statistic
ICCAD13	Metal	10
LithoBench	Metal Via	271 165

EXPERIMENTS

Comparison with SOTA

- ICCAD'13

Case	A2-ILT [6]			MultiILT [8] Ref			MultiILT [8] Reimp					Ours				
	MSE	PV	EPE	MSE	PV	EPE	MSE	PV	EPE	MSA	MSD	MSE	PV	EPE	MSA	MSD
1	45824	59136	7	38495	47015	3	39533	44887	3	832	1	38066	44447	3	1062	16
2	33976	52054	3	28173	37555	0	32516	37374	0	640	12	28623	36914	0	2029	28
3	94634	82661	62	67949	69361	22	65315	75011	23	768	9	61650	70580	15	981	10
4	20405	29435	2	10307	21514	0	9099	21484	0	704	1	9211	21584	0	1342	10
5	37038	62068	1	28482	49683	0	30015	48696	0	896	9	27859	47870	0	865	11
6	40701	54842	2	30334	44127	0	33400	42788	0	896	9	30391	42288	0	1088	17
7	21840	48474	0	14635	36961	0	17419	36241	0	768	9	12791	34389	0	2061	12
8	14912	24598	0	11194	20985	0	11552	18987	0	640	9	11468	18649	0	621	9
9	47489	68056	2	34900	54948	0	37219	54792	0	640	1	32720	54387	0	3940	13
10	9399	20243	0	7266	16581	0	7180	14979	0	2304	9	7130	15014	0	2964	68
Avg	36621.8	50156.7	7.9	27173.5	39873.0	2.5	28324.8	39523.9	2.6	908.8	6.9	**25990.9**	**38612.2**	**1.8**	**1695.3**	**19.4**

- LithoBench

Case	MultiILT [8] Ref					Ours				
	MSE	PV	EPE	MSA	MSD	MSE	PV	EPE	MSA	MSD
Metal	13814.2	24928.2	0.03	720.4	16.1	14643.3	21631.3	0.0	1709.0	24.4
Via	34813.4	39997.4	8.6	464.2	19.2	29183.8	36172.4	3.8	1072.9	11.3
Avg	24313.8	32462.8	4.31	592.3	17.6	21913.5	28901.9	1.9	1390.9	17.8

- Throughput and Efficiency

Solver	A2-ILT [6]	MultiILT [8]	Ours
Throughput	4.51	3.45	2.11
OptPeakMemory	-	7.2GB	0.6GB

EXPERIMENTS

Side Effects of Post Processing

- Post processing sabotages optimality
  - EPE violation +20%
  - MSE +5%

Case	MultiILT w/o PP					MultiILT w/ PP				
	MSE	PV	EPE	MSA	MSD	MSE	PV	EPE	MSA	MSD
1	37976	45423	3	64	1	39533	44887	3	832	1
2	31070	37754	0	576	12	32516	37374	0	640	12
3	63036	73396	19	64	1	65315	75011	23	768	9
4	8498	21561	0	64	1	9099	21484	0	704	1
5	28478	49400	0	64	9	30015	48696	0	896	9
6	29666	43162	0	64	1	33400	42788	0	896	9
7	17333	36319	0	256	9	17419	36241	0	768	9
8	11486	19048	0	64	1	11552	18987	0	640	9
9	34459	55688	0	64	1	37219	54792	0	640	1
10	7180	14979	0	2304	9	7180	14979	0	2304	9
Avg	26918.2	39673	2.2	358.4	4.5	28324.8	39523.9	2.6	908.8	6.9

- Note: This conclusion could differ as the multiILT is implemented following the manuscript. We tried our best to reproduce the result in the paper.

CONCLUSION

Side Effects of Post Processing

- A Fast GPU-Accelerated ILT Alogrithm is Implemented
- Curvy Design Retargeting Allows Faster and Better Convergence
- Diff Morphological Operator Eliminates Rule-Violating Artifacts During Optimization Runtime Which Avoids Post Processing
- We are undergoing internal process to release the source code, follow https://github.com/phdyang007/curvyILT it could be ready anytime.