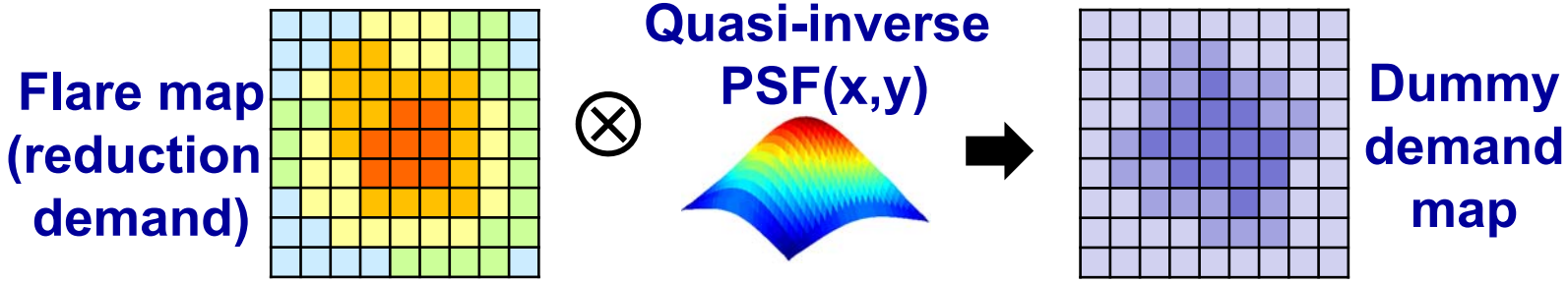
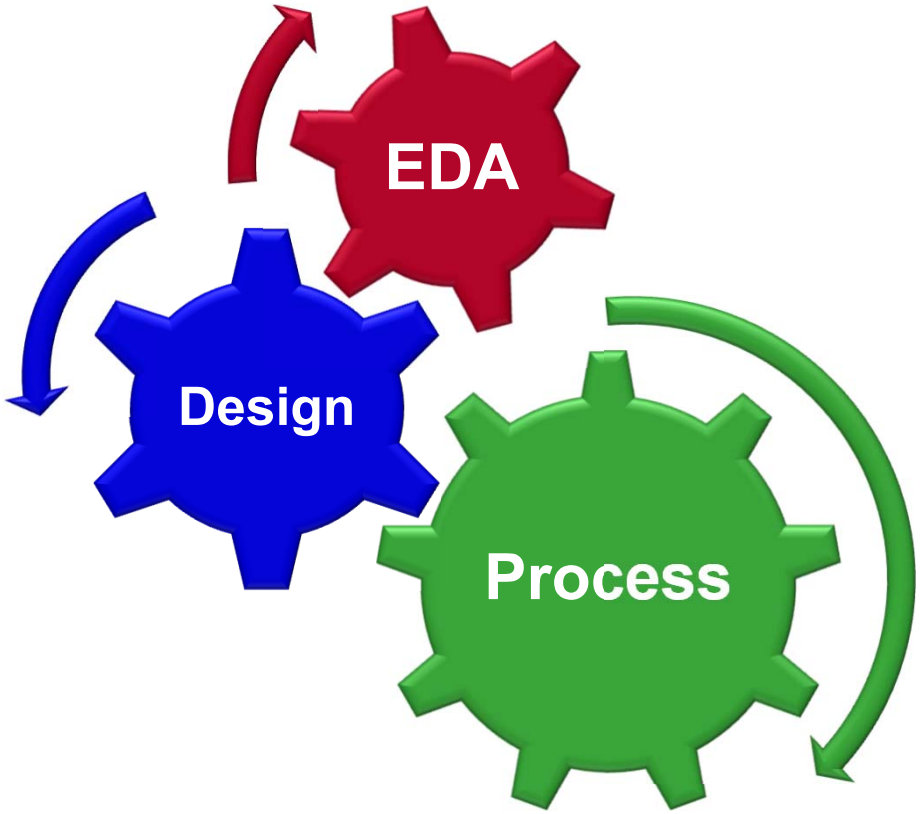
Unit 8: Design for Manufacturability

- Course contents:
  - Lithography basics
  - Double/multiple pattering
  - Extreme ultraviolet (EUV)
  - Electron beam (e-beam)
  - Directed self-assembly (DSA)
- Readings
  - W&C&C: Chapter 12



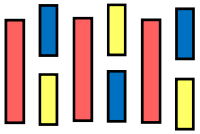
Collaboration for Nanometer Design

- Crucial to have 3-way close collaboration among process, design, and EDA to achieve high-quality nanometer circuit design

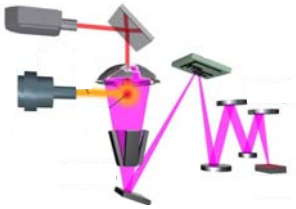


Outline

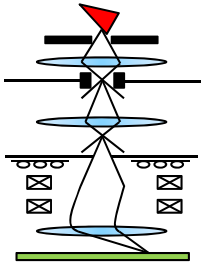
Lithography Basics



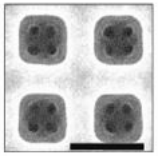
Multiple Patterning



Extreme Ultraviolet (EUV)



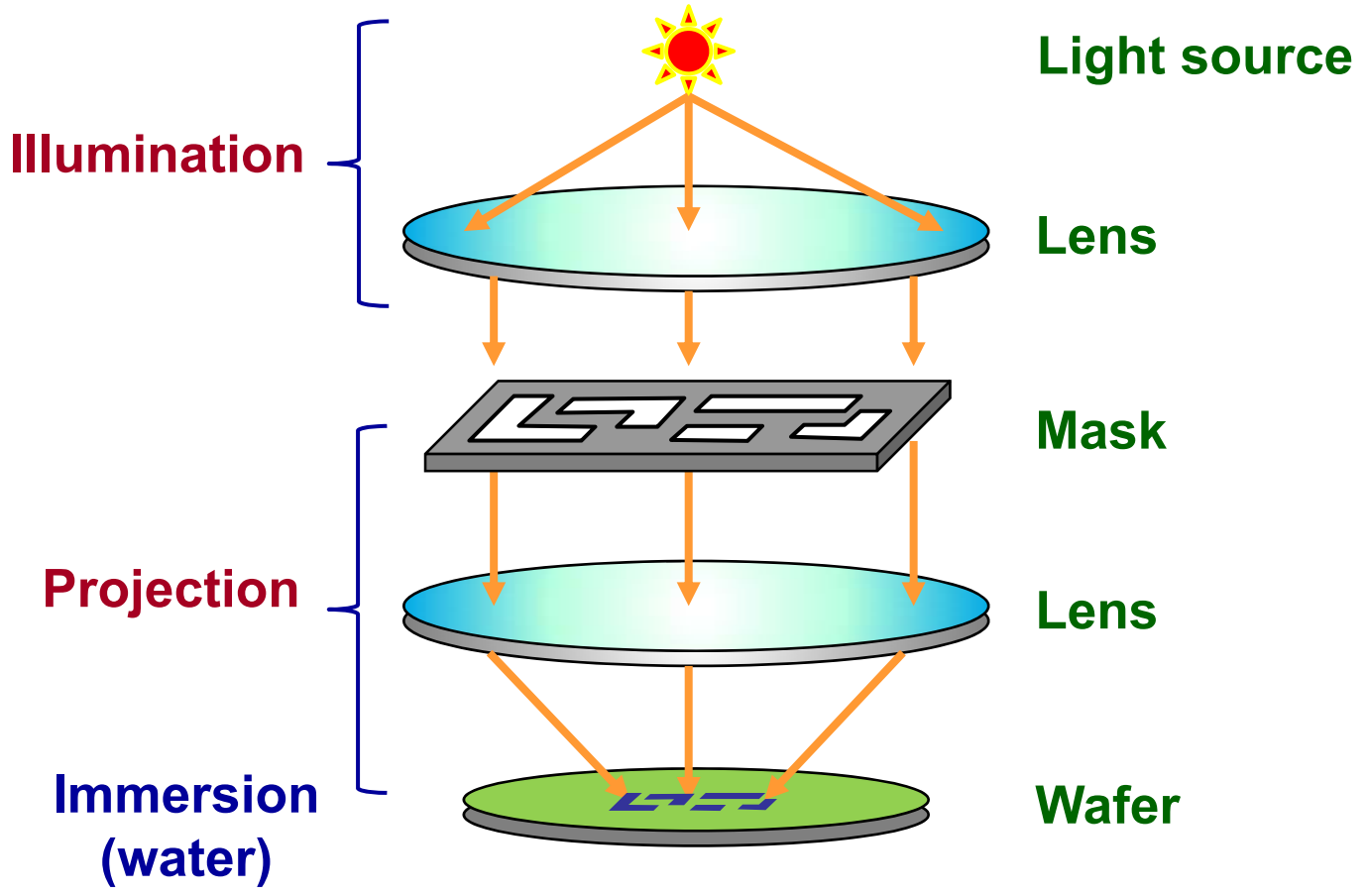
Electron Beam (e-beam)



Directed Self-Assembly (DSA)

Optical Lithography System

- Patterns on a mask are transferred onto a wafer



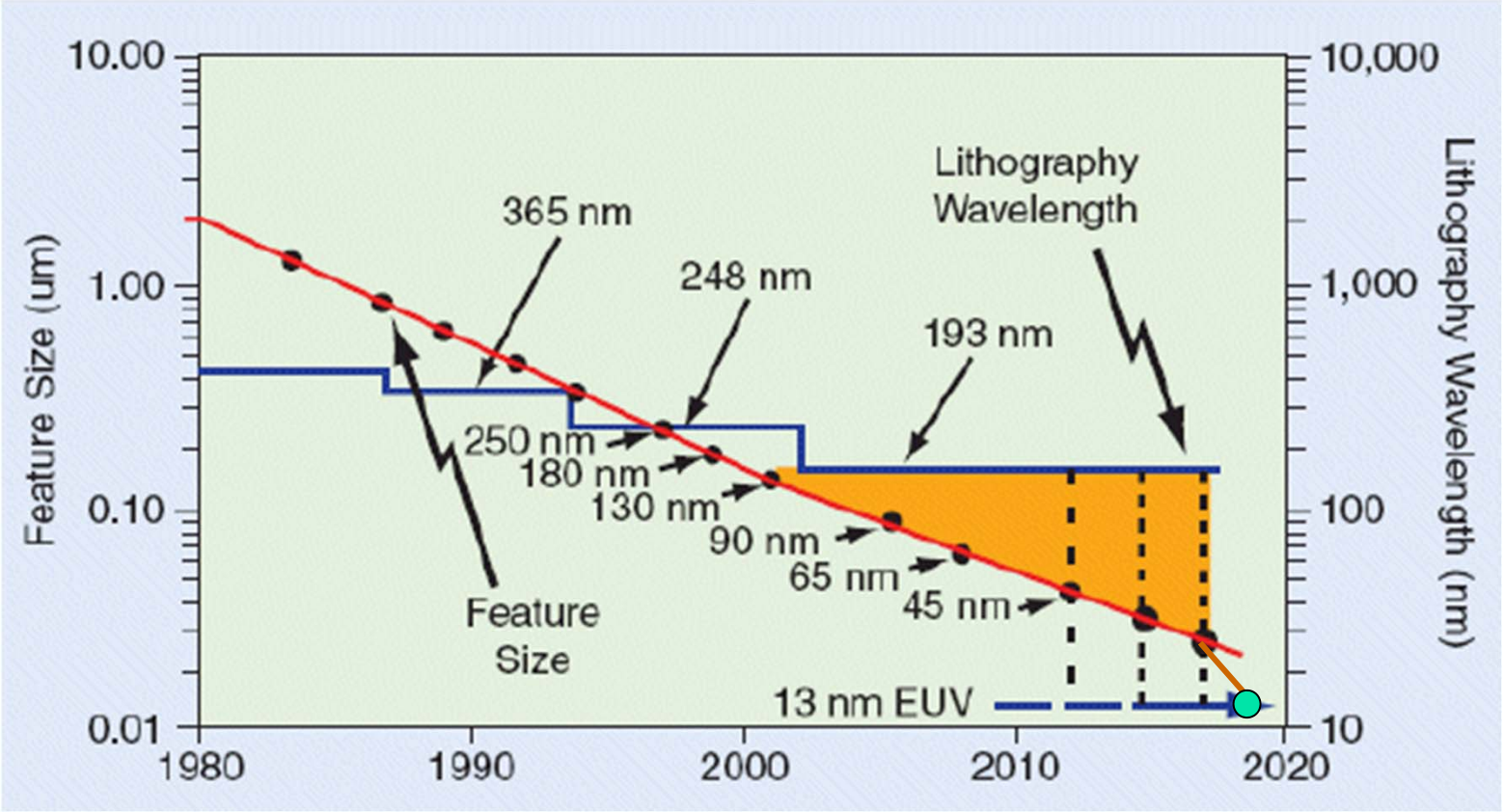
$R = k_1 \lambda / NA$ 0.25 * 193 nm / 1.35 = 36 nm

R: resolution; k_1: resolution constant (>= 0.25); λ: wavelength

NA: numerical aperture = *f*(lens, refractive index)

Sub-Wavelength Lithography Gap

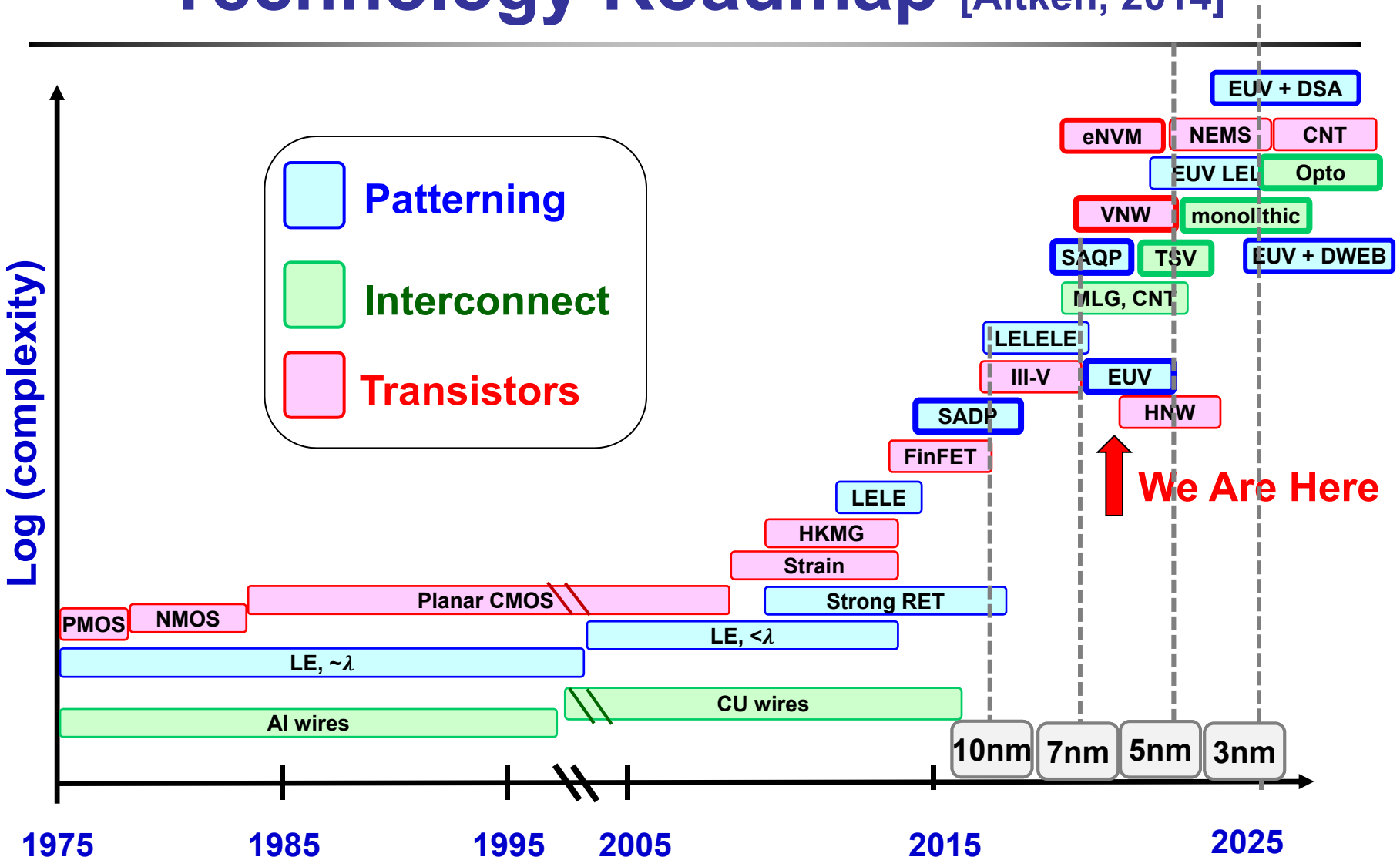
- ***Sub-wavelength lithography:*** use light of larger wavelength (193nm) to print features of smaller sizes



[S. Borkar, MICRO'04]

EUV (13.5nm)?
e-beam?

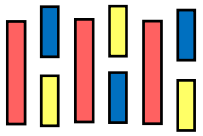
Technology Roadmap [Aitken, 2014]



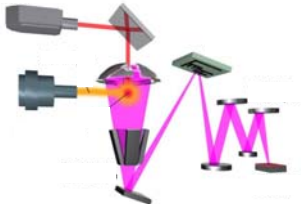
Source: R. Aitken @ ISPD'14 Keynote & S. Segars @ 2014 Kaufman Award dinner
(with revision by Y.-W. Chang)

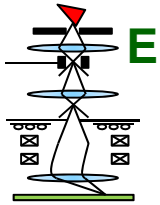
Most Expected Patterning Technologies

Multiple patterning lithography (MPL)



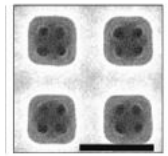
Extreme ultraviolet lithography (EUVL)





Electron beam lithography (EBL)

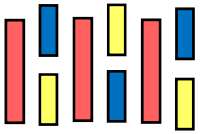
Directed Self-Assembly (DSA)



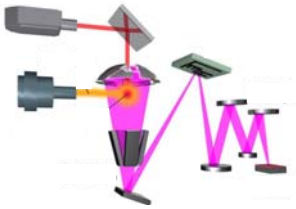
Each technology has different difficulties and requires solutions for a breakthrough

Outline

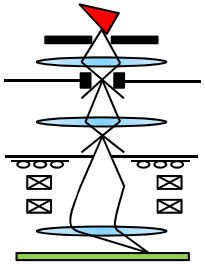
Lithography Basics



Multiple Patterning

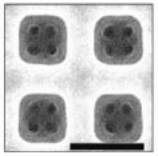


Extreme Ultraviolet (EUV)



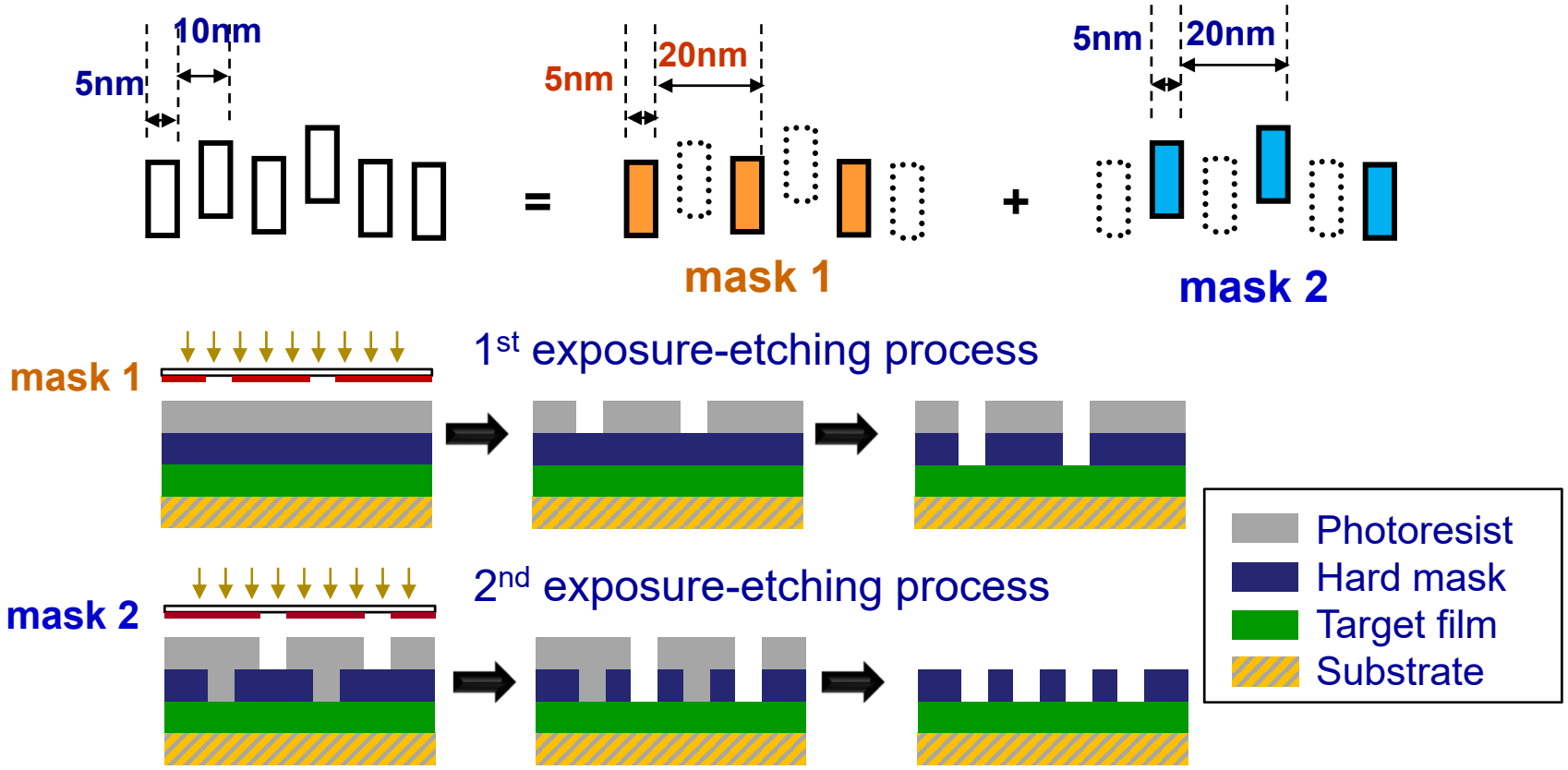
Electron Beam (e-beam)

Directed Self-Assembly (DSA)



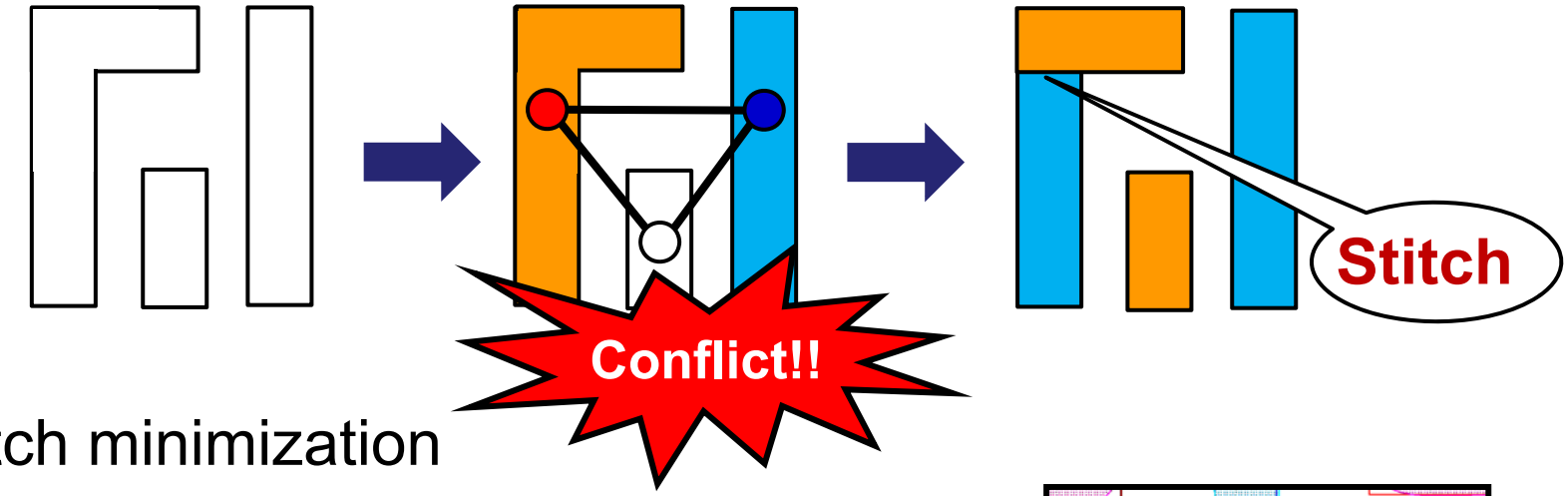
Litho-Etch-Litho-Etch (LELE) Double Patterning

- Pro: Simpler layout decomposition into masks
- Con: **overlay error** with misalignment between masks
- Hsu, Chang & Nassif, ICCAD-09; Chen & Chang, ICCAD-10

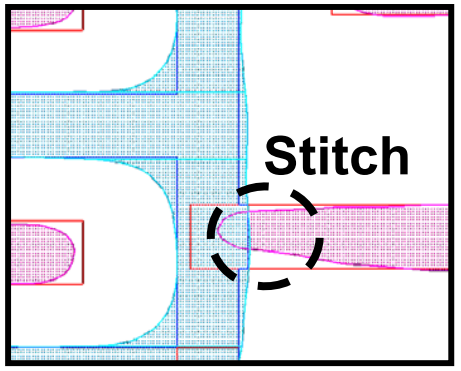


Major LELE Patterning Challenges

- Layout decomposition (LD): graph 2-coloring problem
  - Not always feasible to decompose a 2D layout
  - Some conflicts can be resolved with stitch insertion

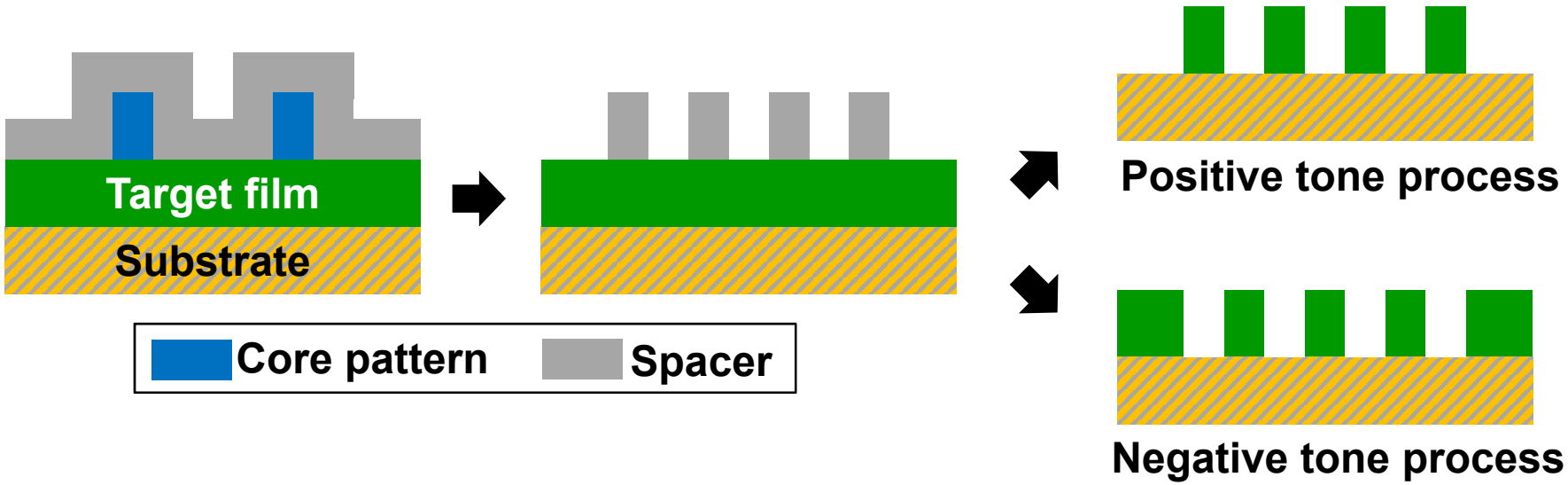


- Stitch minimization
  - Stitches may cause yield loss, due to overlay error



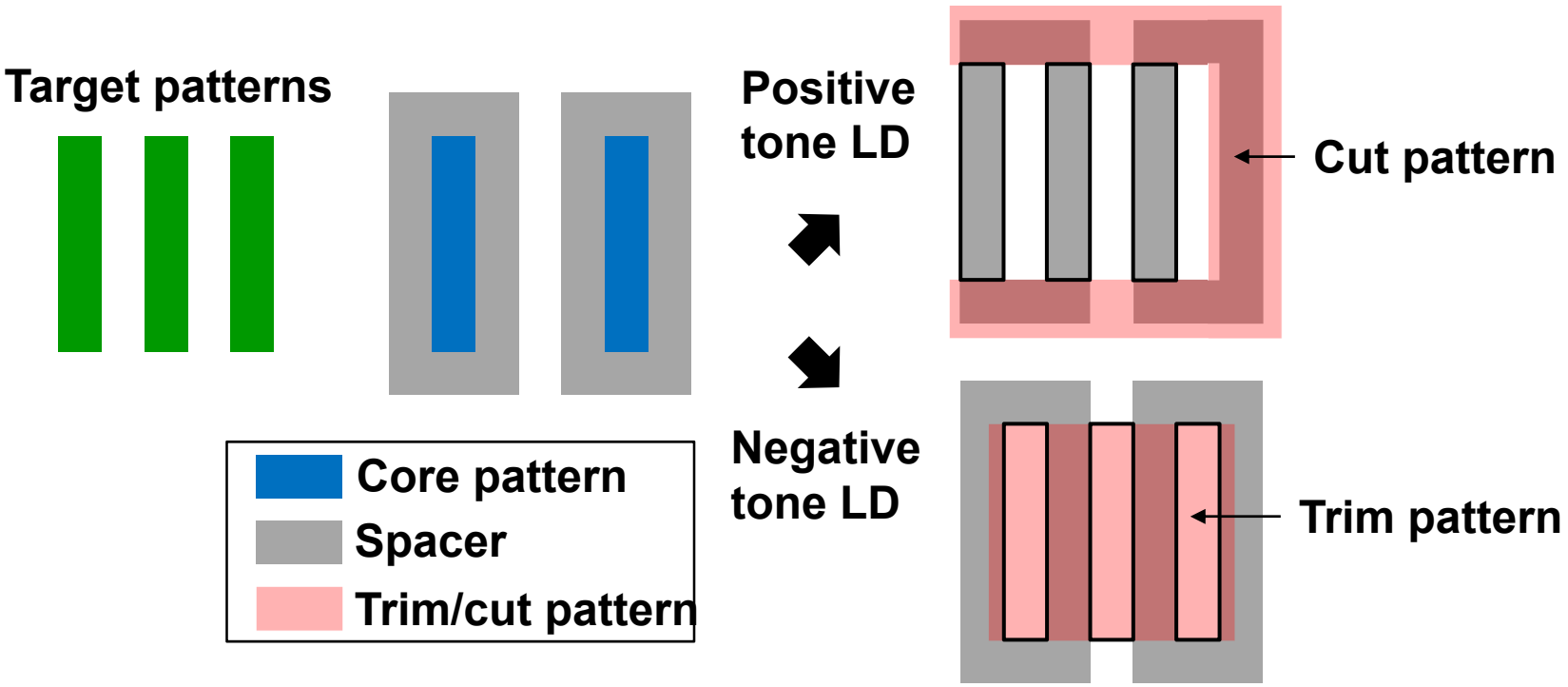
Self-Aligned Double Patterning (SADP)

- Liu, Fang & Chang, DAC-14; Fang, Tai & Chang, ASP-DAC-15
- Becomes more popular due to its better overlay and critical dimension (CD) controllability
- Processes in SADP
  - Positive tone: spacers define lines
  - Negative tone: spacers define trenches



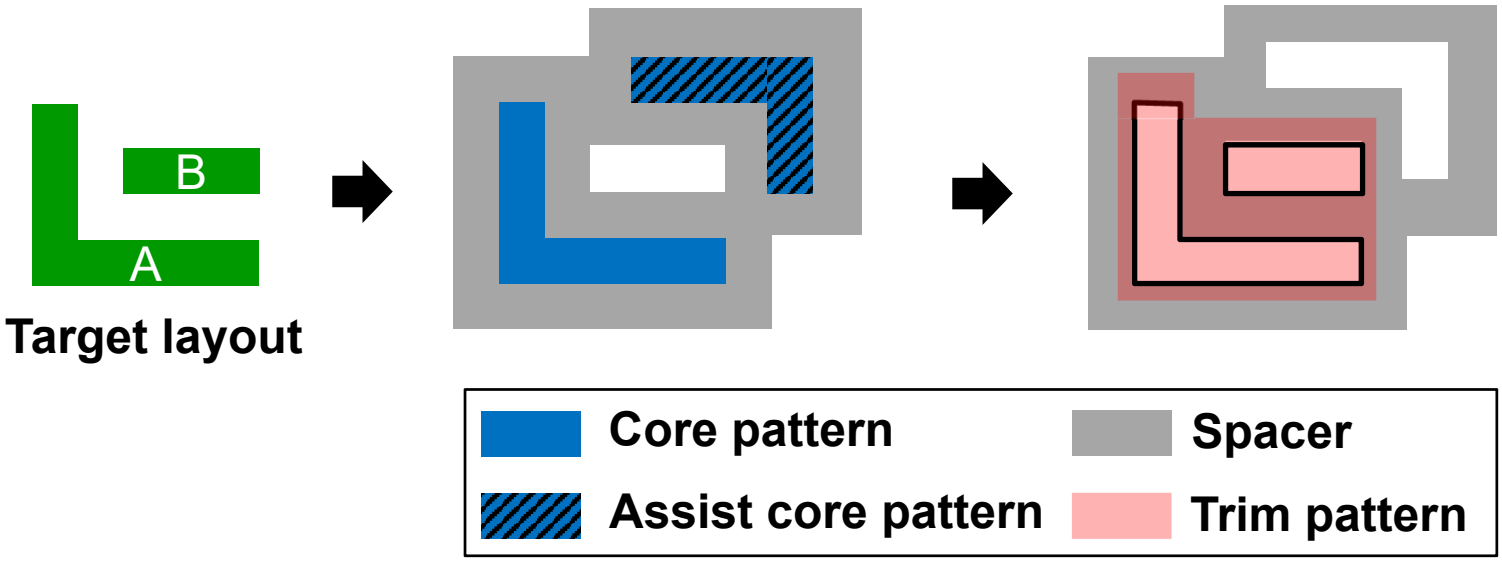
SADP Challenges: Layout Decomposition

- SADP Layout decomposition (LD) is not intuitive
  - Positive tone: spacers define patterns
  - Negative tone: spacers define spacing among patterns



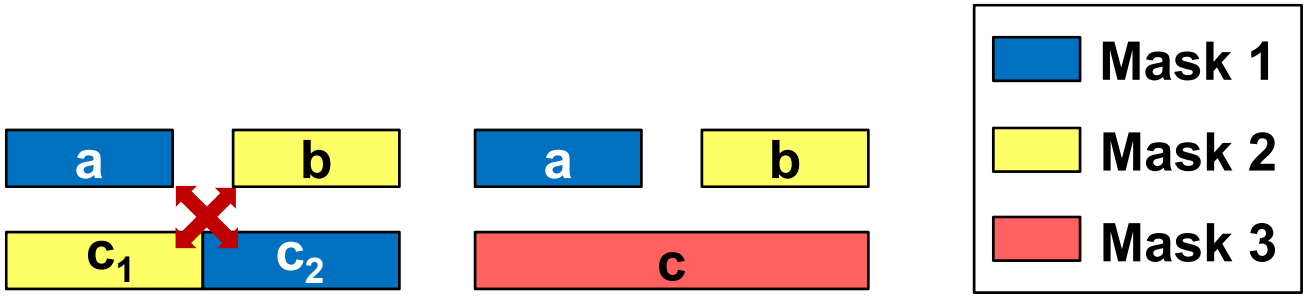
2D SADP LD Is Harder

- Negative tone 2D LD
  - Patterns are formed by either core patterns or assist core patterns
  - Patterns with arbitrary spacing values may be distorted after LD



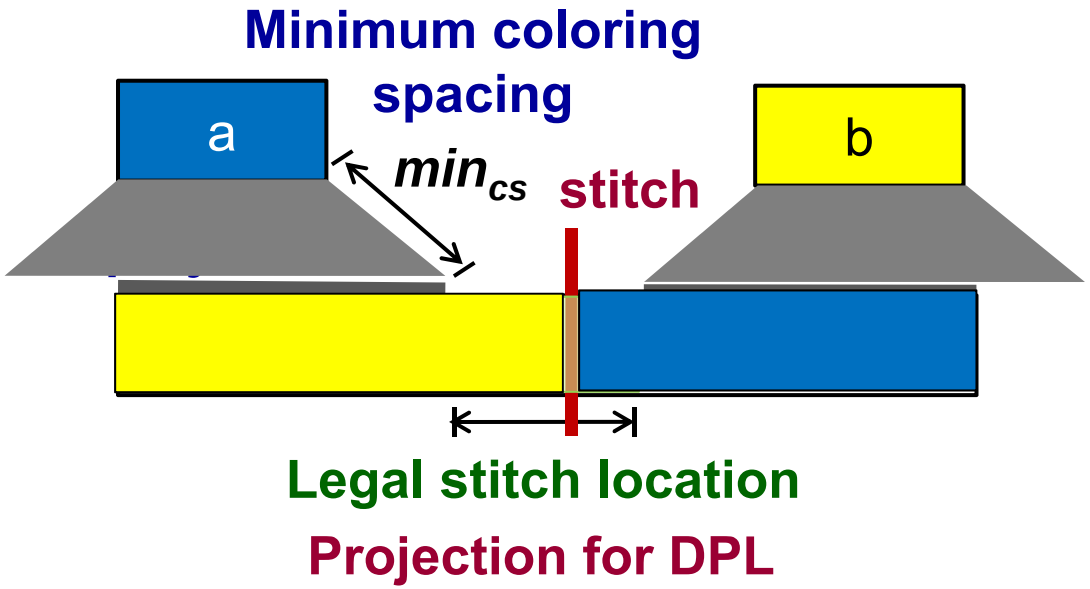
LD in Triple Patterning Lithography (TPL)

- Fang, Chang, Chen, DAC-12, TCAD-14; Hsu & Chang, ASP-DAC-15
- TPL is required for more complex layouts
- TPL LD is modeled as a 3-coloring problem
  - Determine if a graph is 3-colorable is NP-complete
  - TPL LD is NP-complete even for a 3-colorable, planar graph



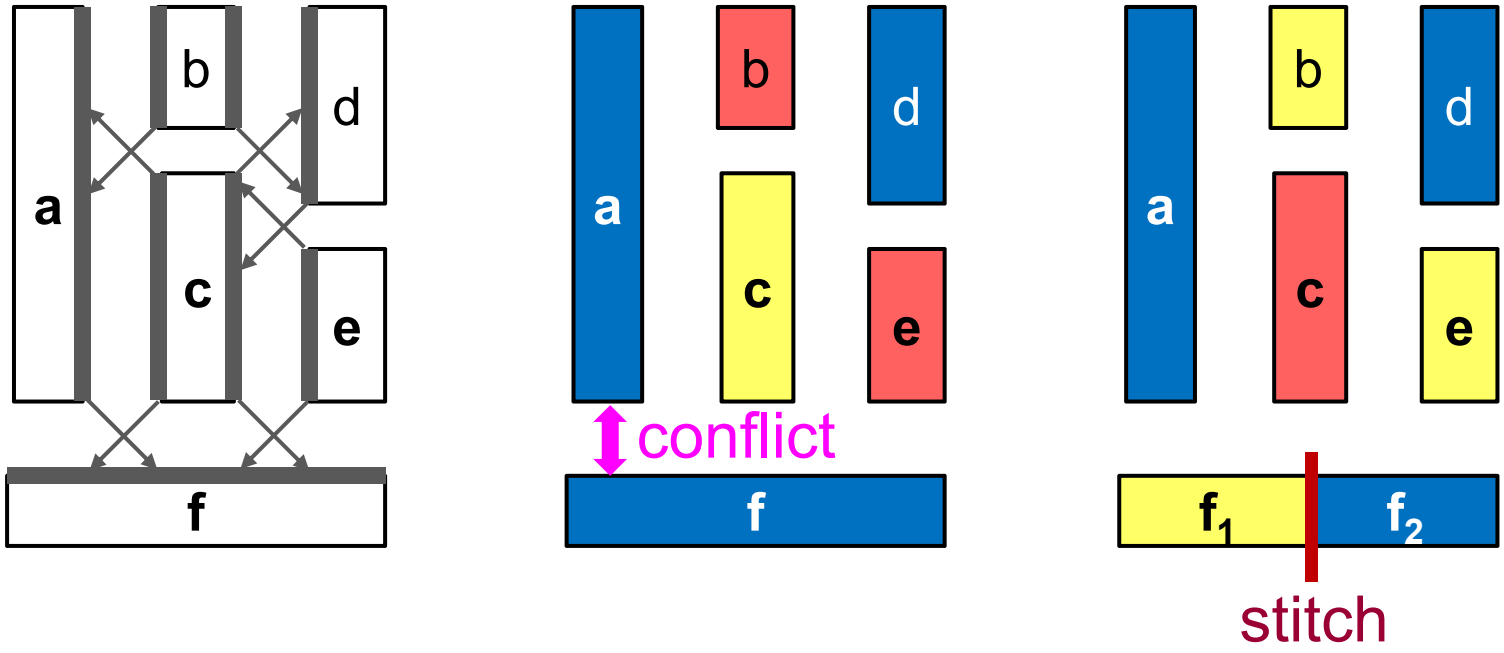
Stitch Finding in TPL

- Yu et al. [ICCAD'11] directly applied the projection method used in DPL to find stitch candidates
- We observed that the projection method may miss legal stitches for TPL [DAC'12, TCAD'14]
  - Some generated conflicts can be resolved with stitch insertion

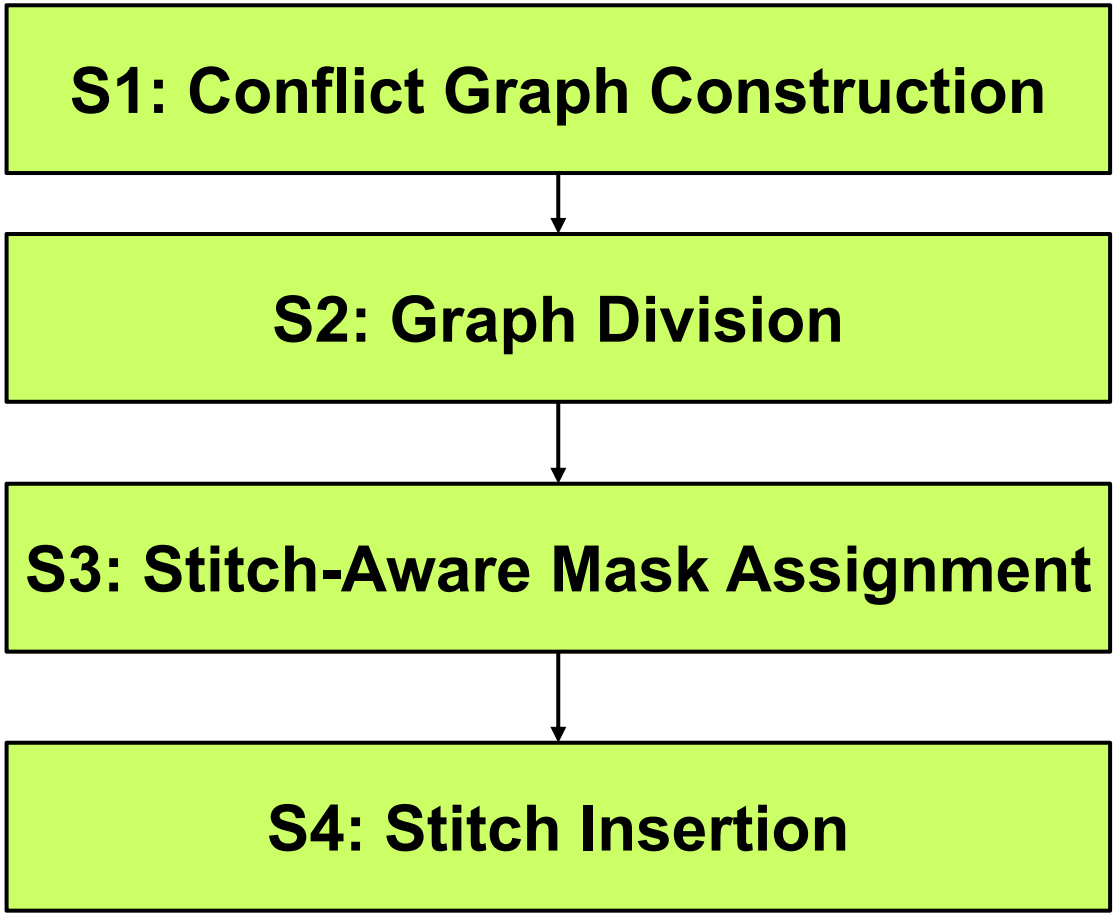


Stitch Finding in TPL

- Yu et al. [ICCAD'11] directly applied the projection method used in DPL to find stitch candidates
- Observed that the projection method may miss legal stitches for TPL [DAC'12, TCAD'14]
  - Some conflicts can be resolved with stitch insertion

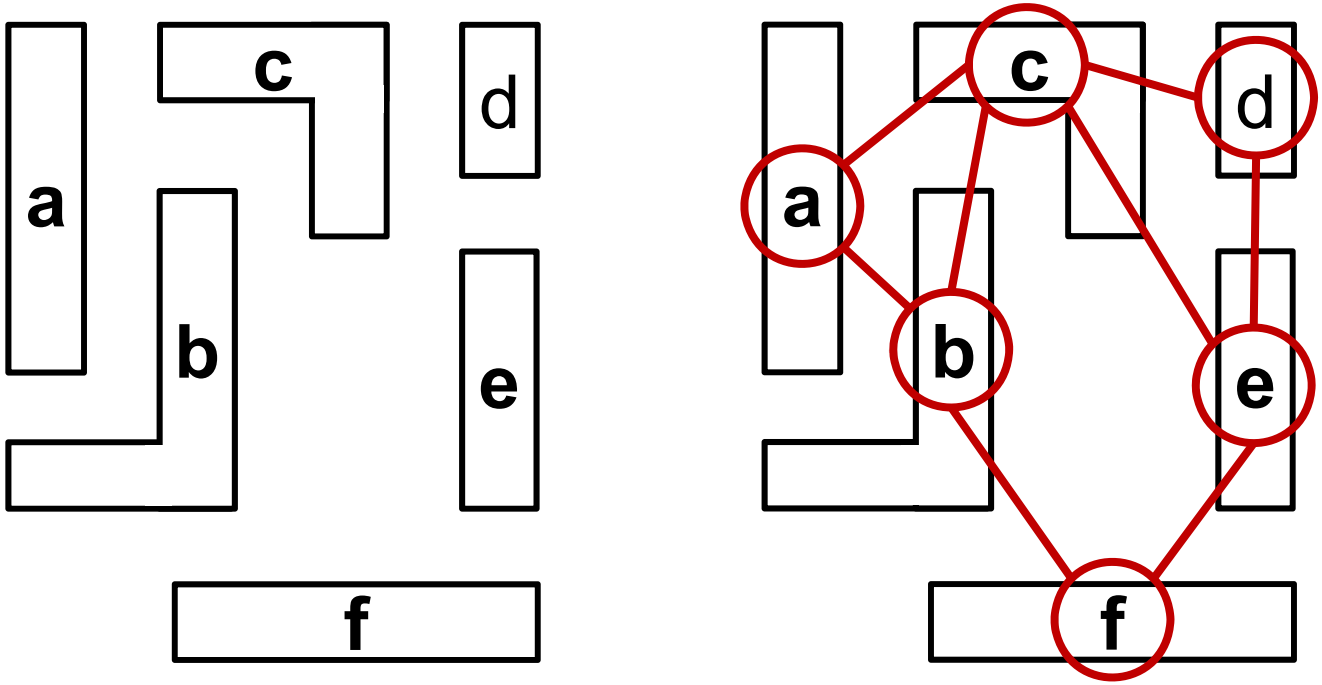


Layout Decomposition for TPL



S1: Conflict Graph Construction

- A conflict graph G=(V,E) is constructed
  - A node: a pattern
  - An edge: the distance between the two patterns < min_{CS}



Huge graph size?

Divide and Conquer! Graph Division!!

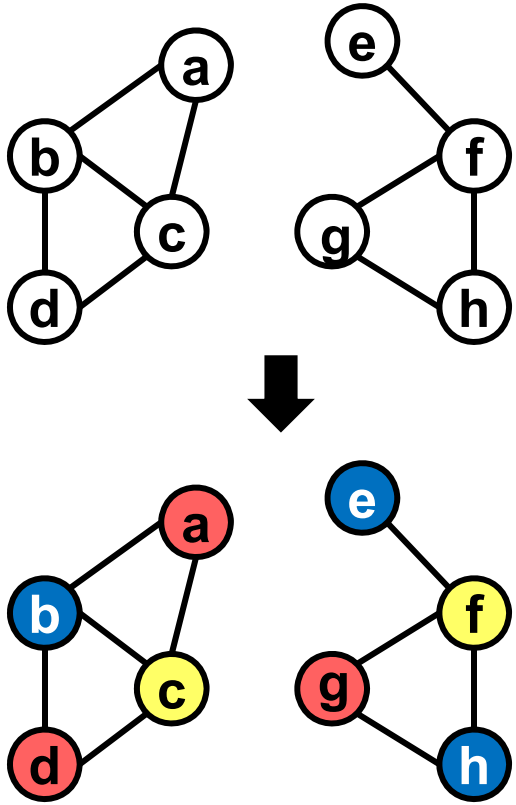
S2: Graph Division Techniques

[ICCAD'11]

1. Connected Component Computation
2. Vertex with Degree Less than Three Removal
3. 2-Edge-Connected Component Computation

Ours

4. 2-Connected Component Computation
5. 3-Edge-Connected Component Computation



S2: Graph Division Techniques

[ICCAD'11]

1. Connected Component Computation

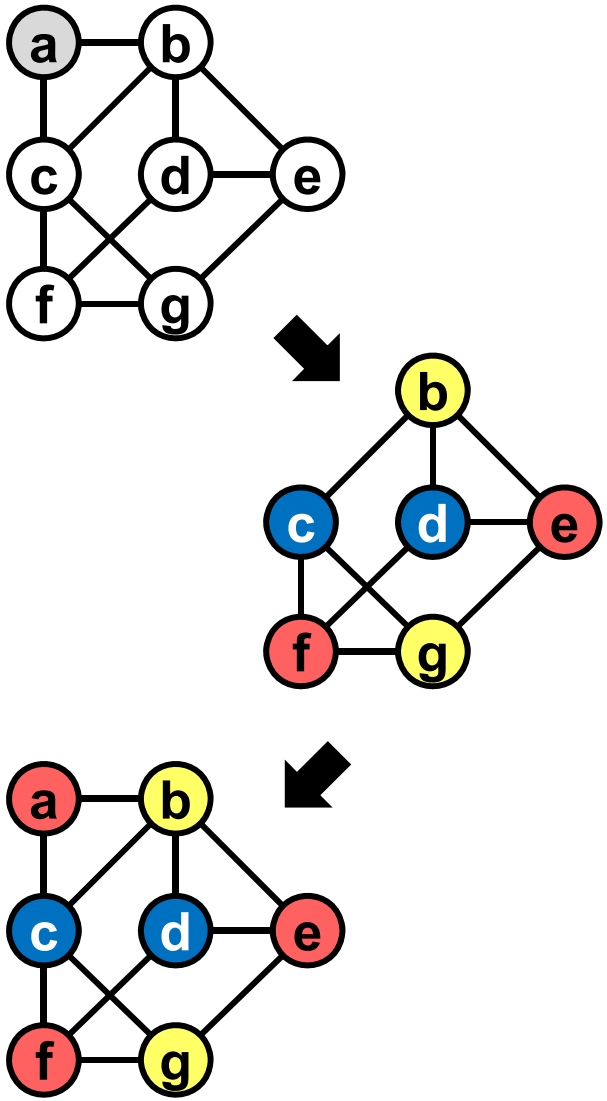
2. Nodes with Degree Less than Three Removal

3. 2-Edge-Connected Component Computation

Ours

4. 2-Connected Component Computation

5. 3-Edge-Connected Component Computation



S2: Graph Division Techniques

[ICCAD'11]

1. Connected Component Computation

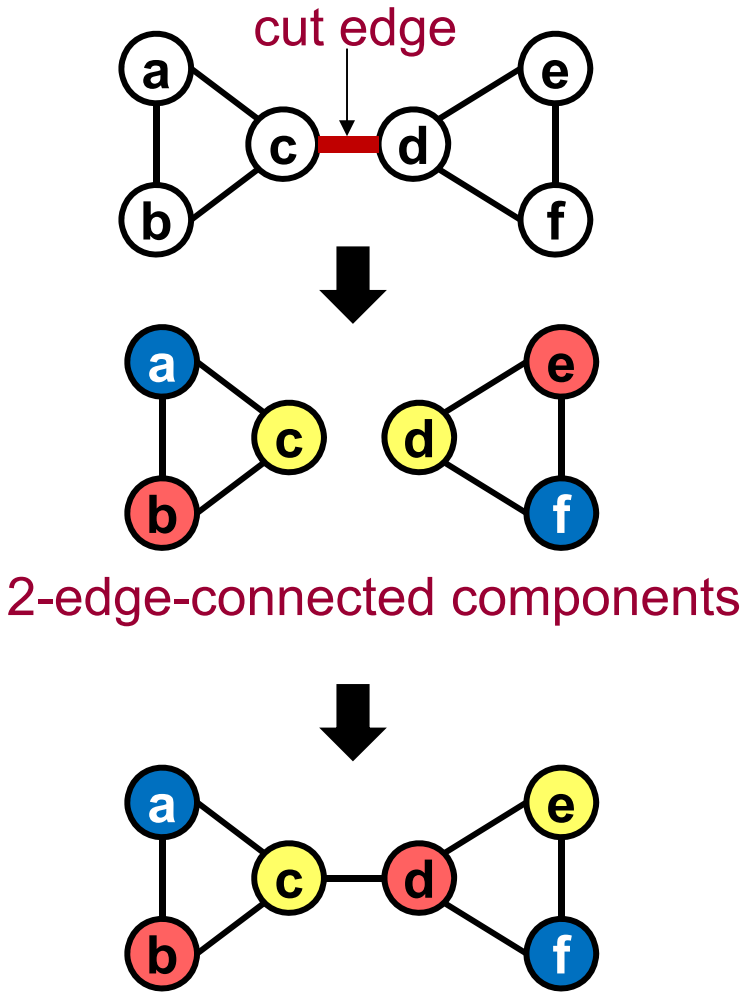
2. Vertex with Degree Less than Three Removal

3. 2-Edge-Connected Component Computation

Ours

4. 2-Connected Component Computation

5. 3-Edge-Connected Component Computation



S2: Graph Division Techniques

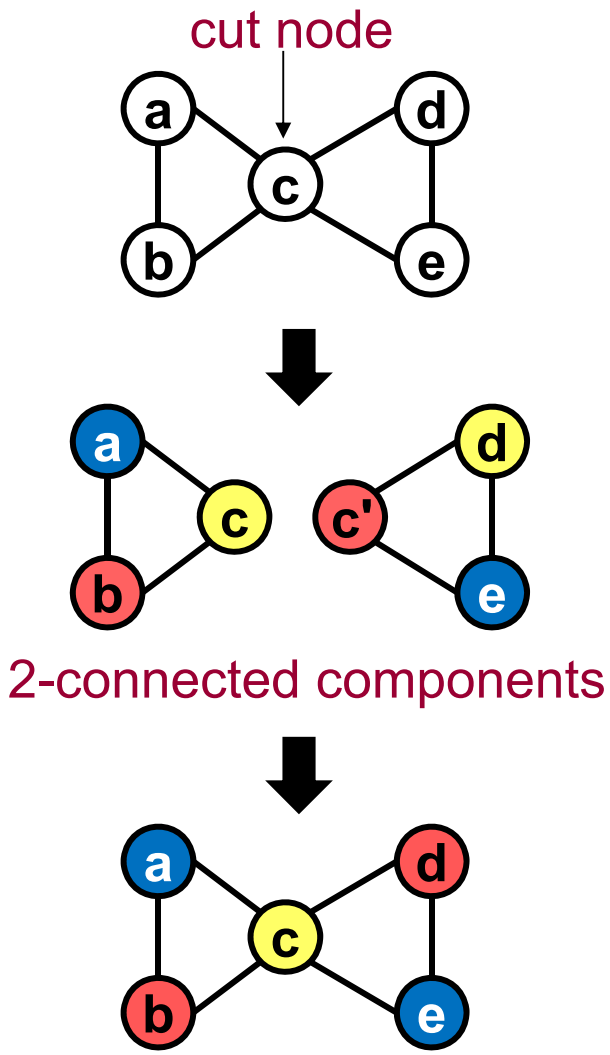
[ICCAD'11]

1. Connected Component Computation
2. Vertex with Degree Less than Three Removal
3. 2-Edge-Connected Component Computation

Ours

4. 2-Connected Component Computation

5. 3-Edge-Connected Component Computation



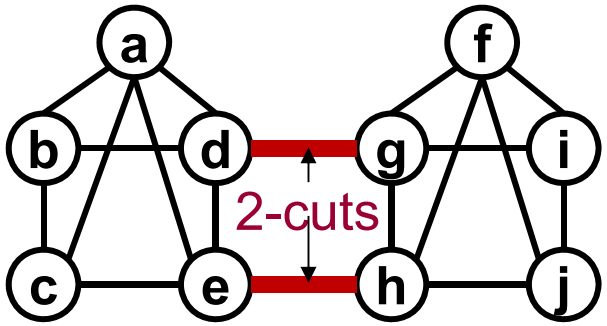
S2: Graph Division Techniques

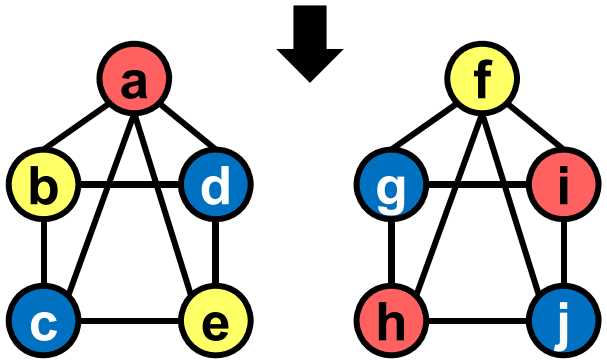
[ICCAD'11]

1. Connected Component Computation
2. Vertex with Degree Less than Three Removal
3. 2-Edge-Connected Component Computation

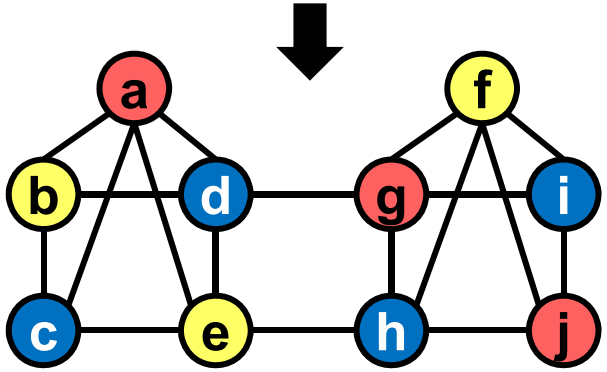
Ours

4. 2-Connected Component Computation
5. **3-Edge-Connected Component Computation**



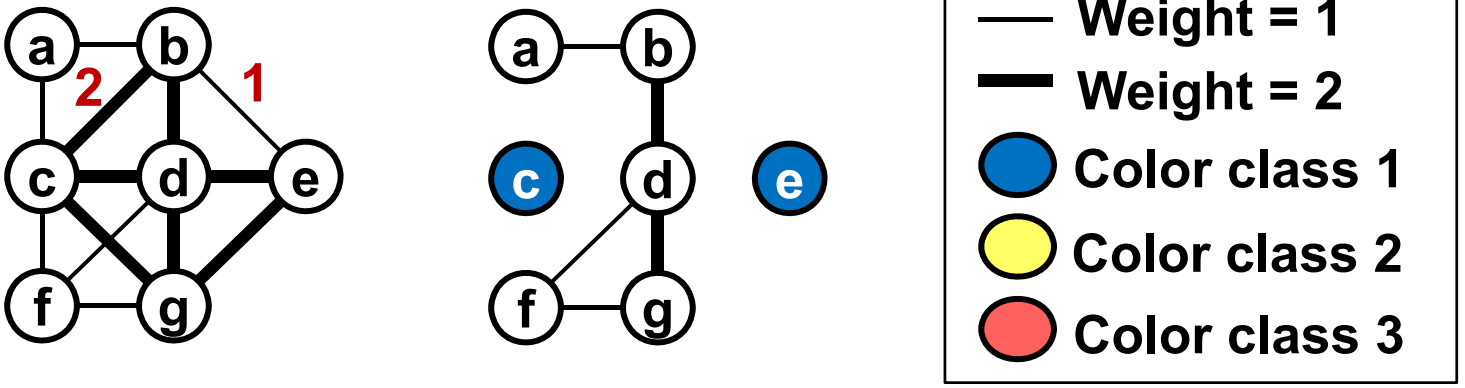


3-edge-connected components



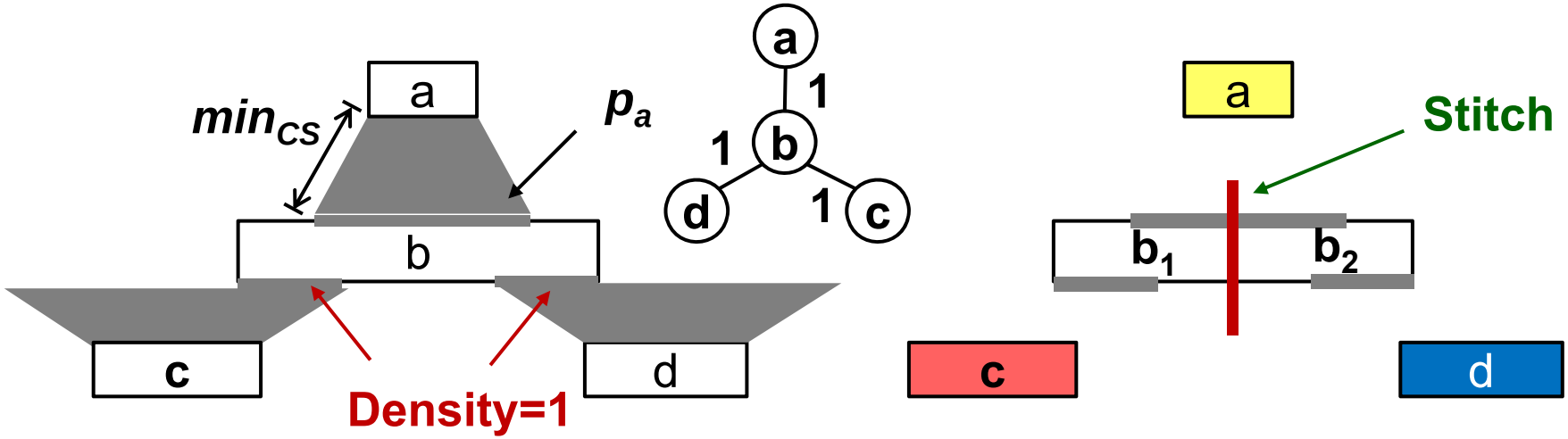
S3: Stitch-Aware Mask Assignment

- Find a mask assignment such that the conflicts are more likely to be resolved with stitch insertion
  - Construct 3 color classes (independent sets) to minimize the total edge weight of the residual graph
    - weight: difficulty of resolving a conflict by stitch insertion
  - Assign each remaining node to a color class such that the corresponding conflict edge has the smallest edge weight

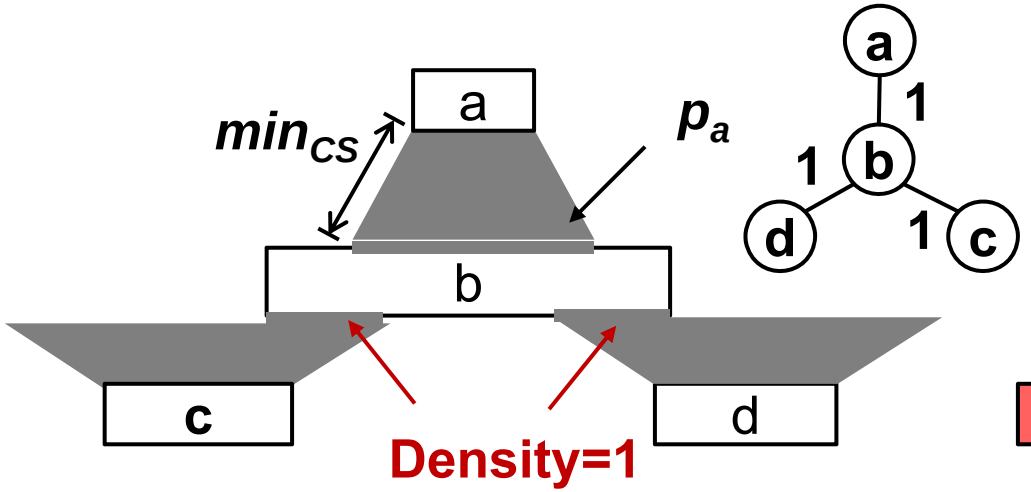


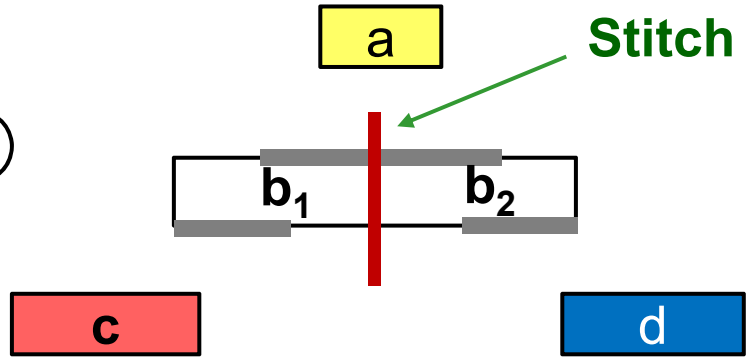
Edge Weight Computation

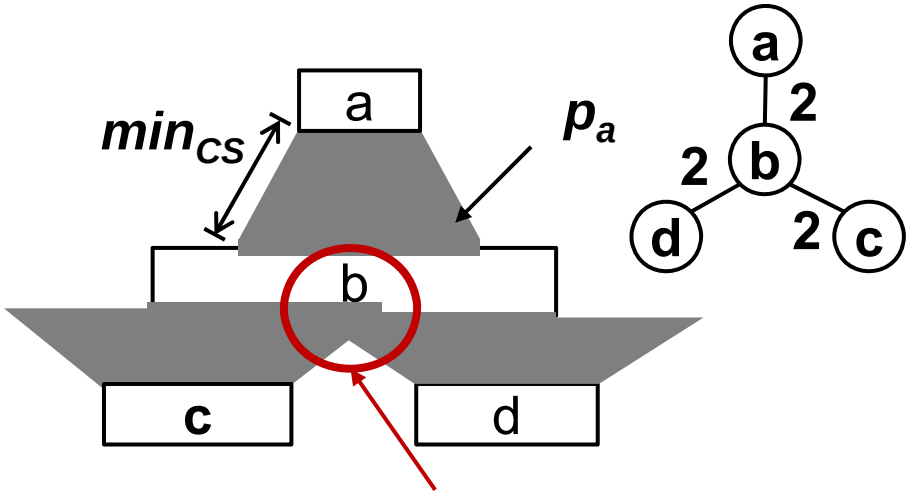
- Weight of a conflict edge reflects how hard the conflict can be resolved by stitch insertion
  - The more projections overlap on a pattern, the more difficult the conflict can be resolved with stitch insertion
  - Edge weight: maximum density of other overlapped projections

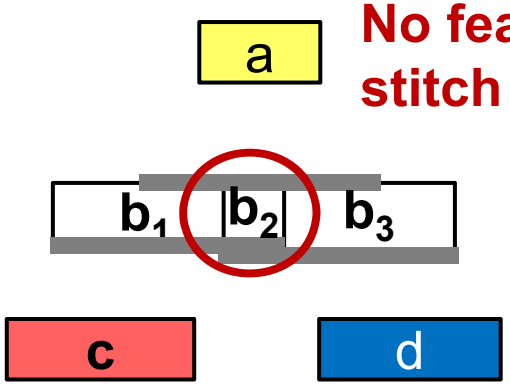


Edge Weight Computation



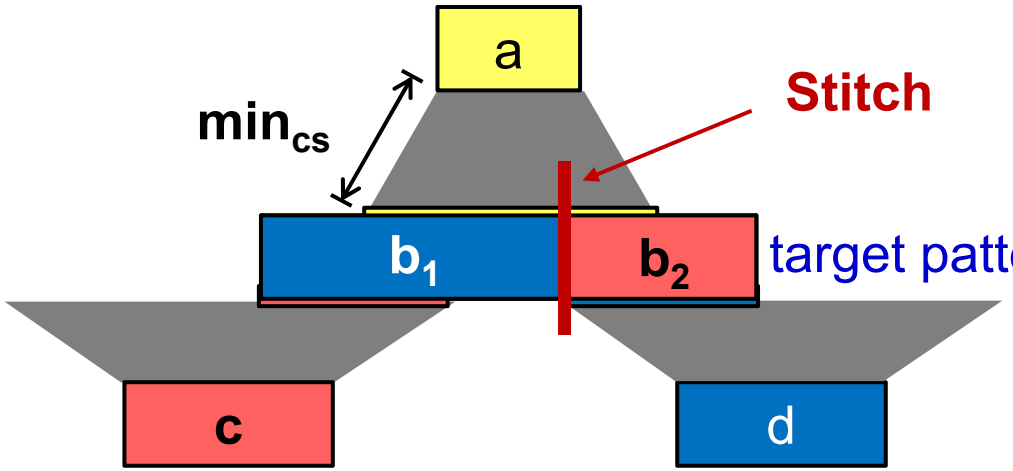


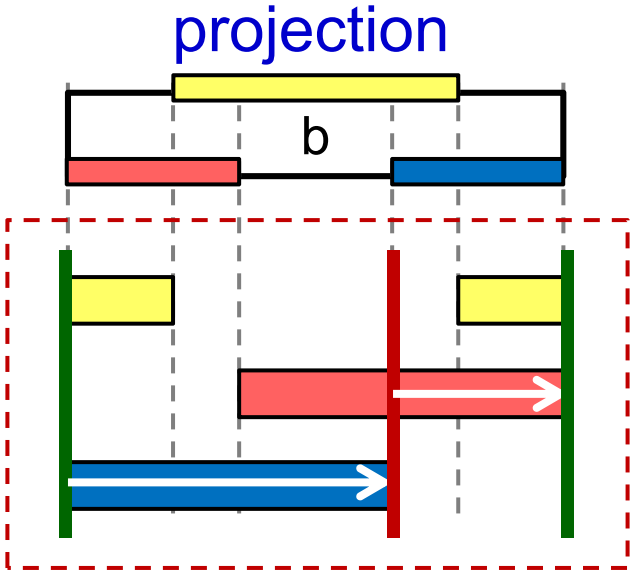




S4: Stitch Insertion

- Solve conflicts with stitch insertion
  - Partition a target pattern into segments with available colors
  - Apply a plane sweep method to solve a conflict with fewer stitches





Experimental Results

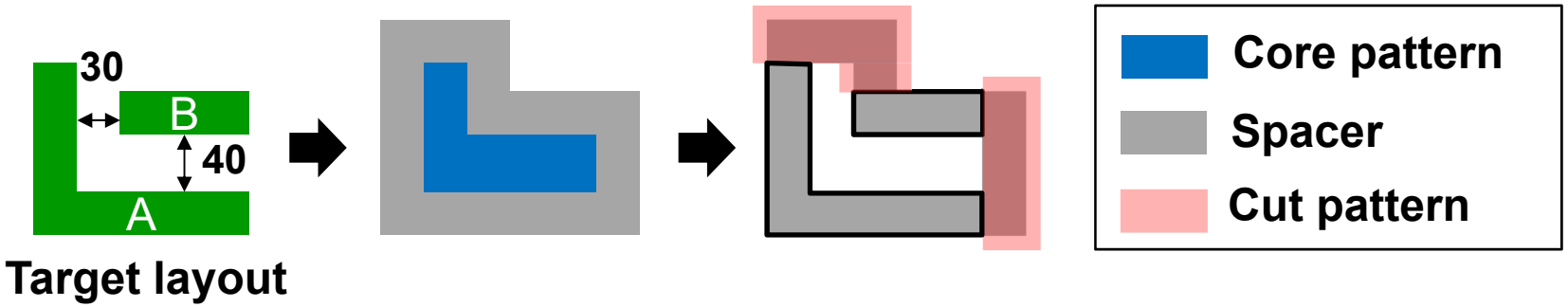
- 56% conflict reduction, 43% cost reduction, 40X speedup
  - Cost = #conflicts * 10 + #stitches [ICCAD'11]

Circuit	SDP Based [ICCAD'11]				Ours [DAC'12, TCAD'13]			
	#C	#S	Cost	CPU (s)	#C	#S	Cost	CPU (s)
C432	3	1	31	0.14	0	4	4	0.01
C499	0	0	0	0.19	0	0	0	0.01
C880	1	6	16	0.27	1	8	18	0.01
C1355	1	6	16	0.21	1	4	14	0.02
C1908	0	1	1	0.29	1	0	10	0.04
C2670	2	4	24	0.53	2	11	31	0.05
C3540	5	6	56	0.72	2	12	32	0.08
C5315	7	7	77	1.01	3	11	41	0.11
C6288	82	131	951	4.49	19	248	438	0.13
C7552	12	15	135	1.72	3	37	67	0.17
S1488	1	1	11	0.33	0	4	4	0.03
S38417	44	55	495	21.67	20	82	282	0.63
S35932	93	18	948	96.45	46	63	523	1.98
S38584	63	122	752	99.80	36	176	536	2.03
S15850	73	91	821	87.22	36	146	506	1.91
Avg.	**2.28**	**0.58**	**1.73**	**39.98**	**1.00**	**1.00**	**1.00**	**1.00**

SDP: semidefinite programming

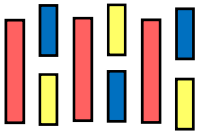
Positive Tone SADP 2D LD

- Fang, Tai, Chang, “Layout decomposition for spacer-is-metal (SIM) self-aligned double patterning”, ASP-DAC-15
- Positive tone 2D layout decomposition
  - Patterns are formed by spacers
  - Only for designs with uniform pattern width
  - May have better decomposability since non-uniform spacing can be controlled by core patterns
- Positive-tone SADP routing

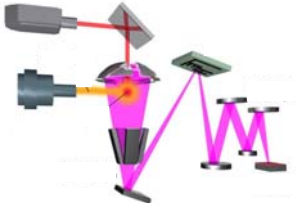


Outline

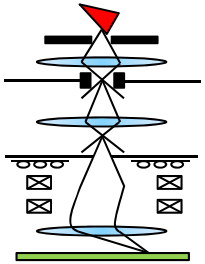
Lithography Basics



Multiple Patterning

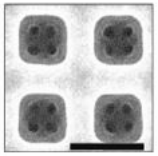


Extreme Ultraviolet (EUV)



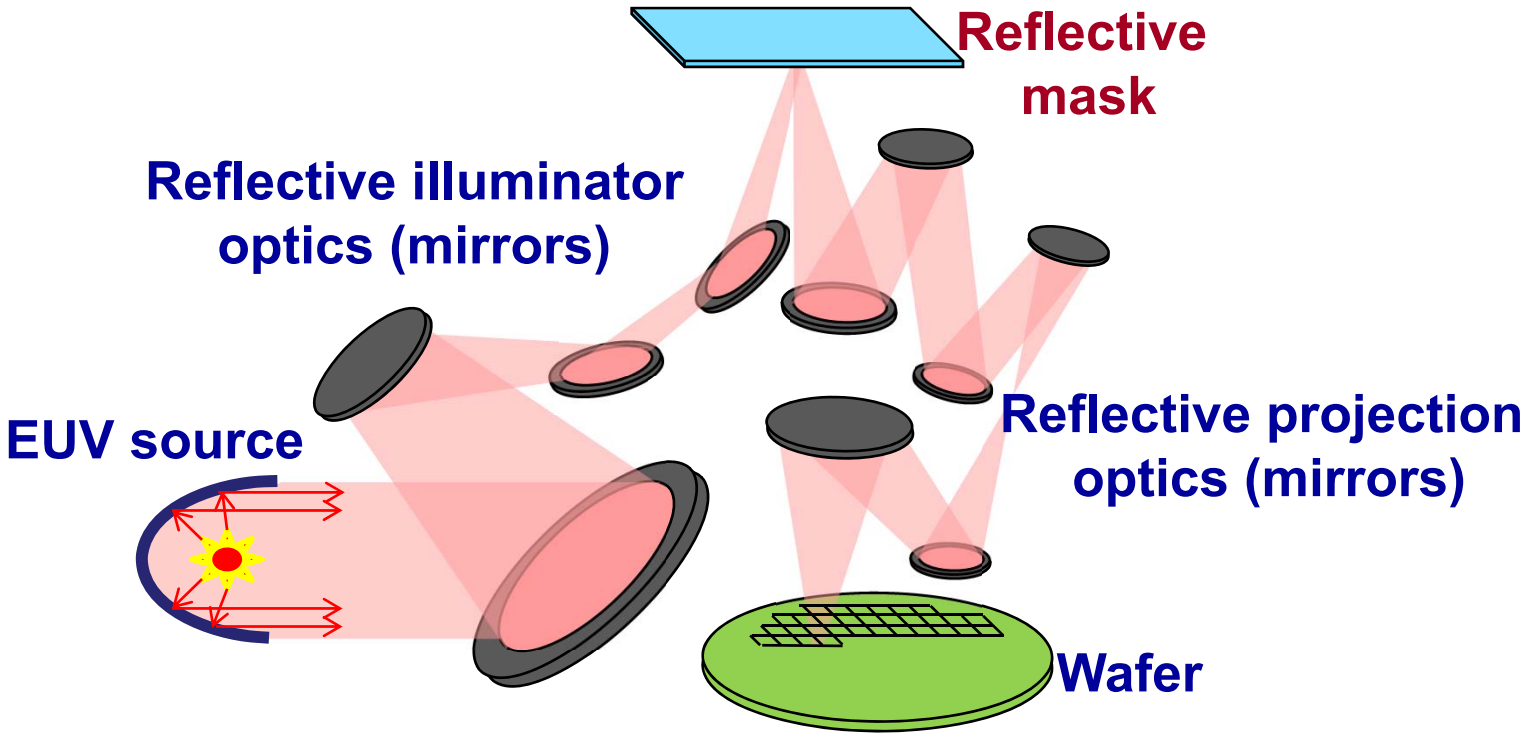
Electron Beam (e-beam)

Directed Self-Assembly (DSA)

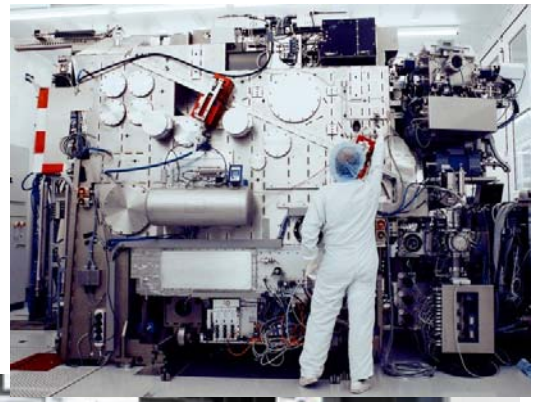


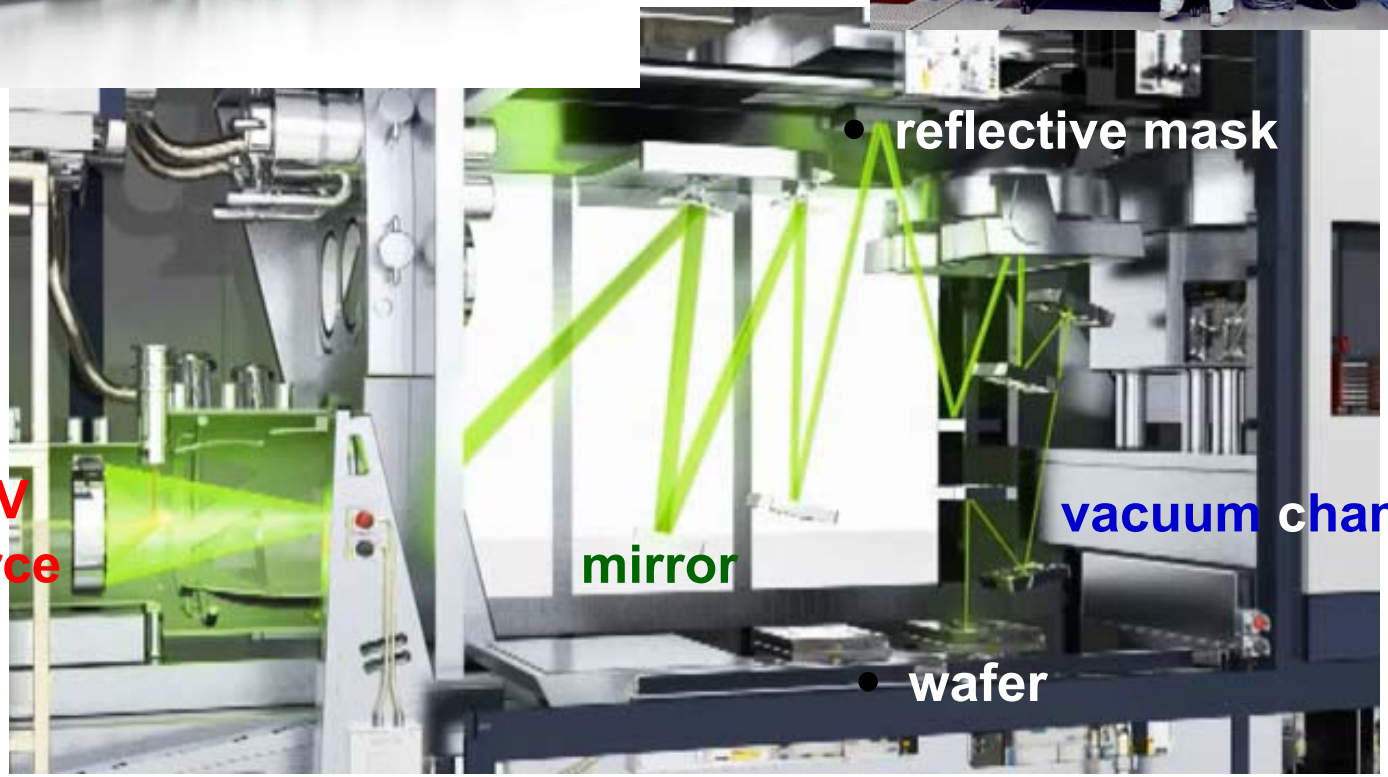
Extreme Ultraviolet Lithography (EUVL)

- EUVL is the most invested next-generation lithography technology
  - Its wavelength is only 13.5 nm
  - Reflective optical components and masks are used



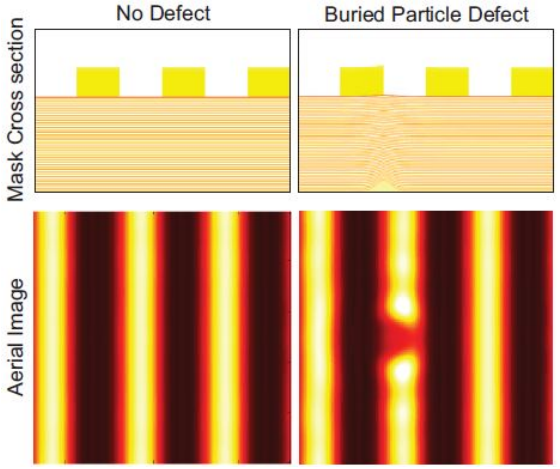
ASML EUV Lithography System





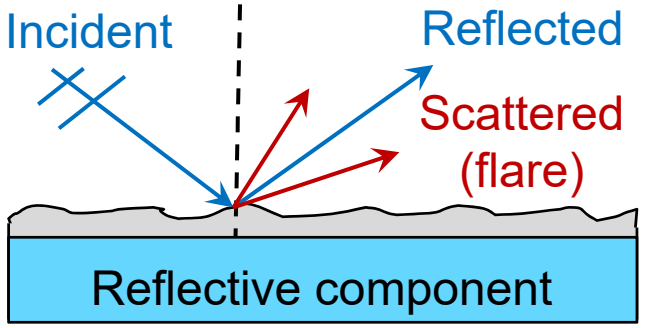
Layout Techniques for EUV Effects

1. **Blank defect mitigation:** pattern relocation
   - Prof. Martin Wong' group: SPIE'11, ASP-DAC'12, ICCAD'12, ASP-DAC'13
   - Prof. Andrew Kahng's group & Prof. Puneet Gupta's group: ASP-DAC'14, DAC'15 Tutorial

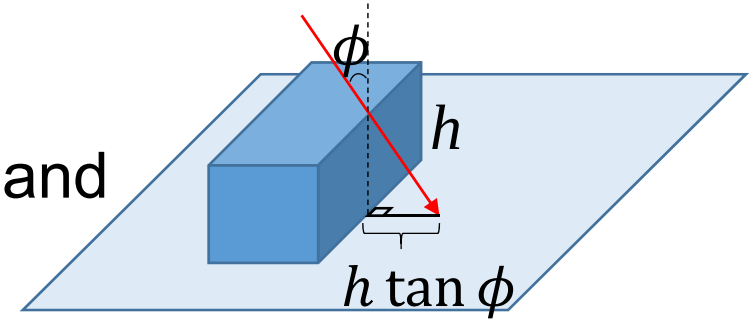


Clifford, et al. [SPIE'10]

2. **Flare mitigation:** pattern density adjustment & dummification
   - Ours: DAC'12, DAC'14, ICCD'14

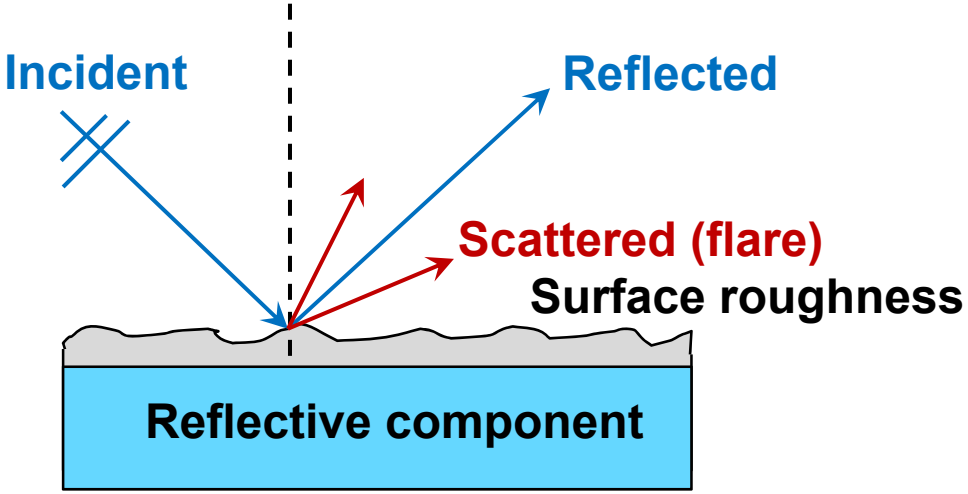


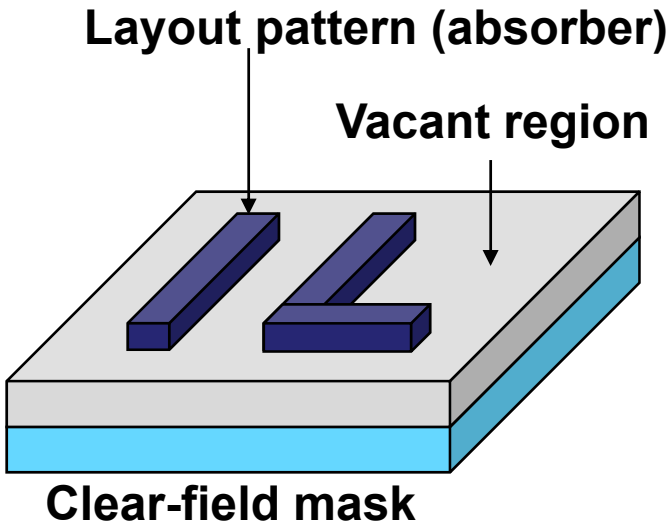
3. **Shadowing effect handling:** pattern shape, position, size, and pitch adjustment



Flare in EUVL

- Fang & Chang, DAC-12; Liu, et al., DAC-14
- Reflective masks are used
- **Flare**: scattered light due to the surface roughness of the optical system
- **Clear field mask**: layout patterns are formed by absorbers
- Flare will be distributed from vacant regions and reduce image contrast (and thus pattern quality)



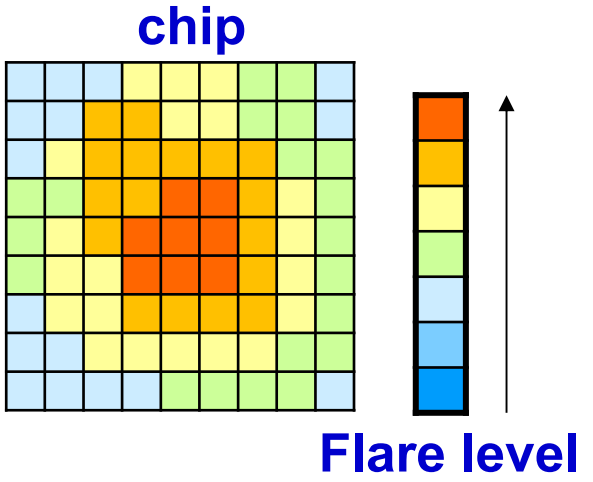


Flare Level and Flare Variation

- High flare level in EUVL [Myers et al. 2008]

$$flare \propto \frac{surface\ roughness}{wavelength^2}$$

- Large flare variation
  - Non-uniformity of layout patterns
  - **Flare periphery effect**: regions at the periphery of a chip have much less flare

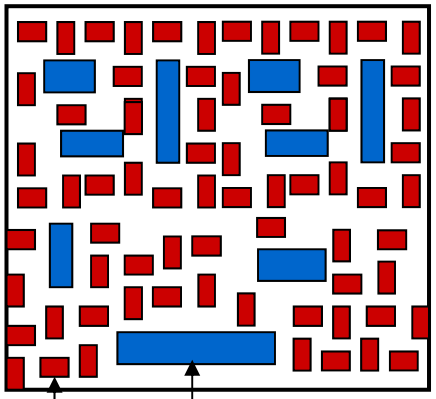


- Both the high flare level and the large flare variation could damage the pattern quality

Flare Mitigation with Dummification [DAC'12]

- Flare can be mitigated with dummification (dummy fill)
  - Existing dummy fill algorithms are for CMP optimization to maximize layout uniformity
  - Need to consider global flare distribution for flare mitigation

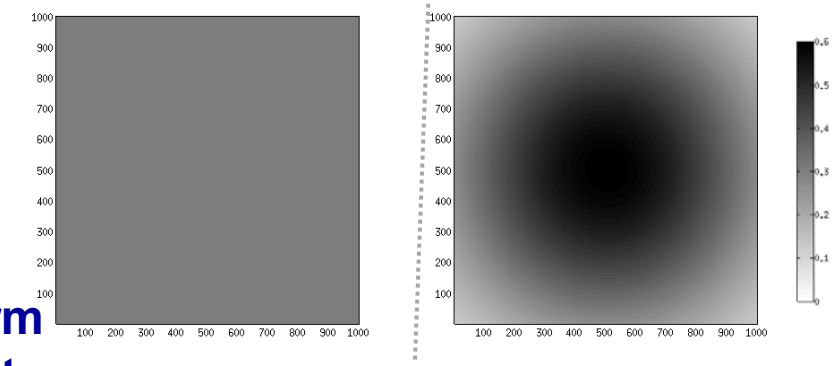
Dummy fill



Layout pattern

Dummy

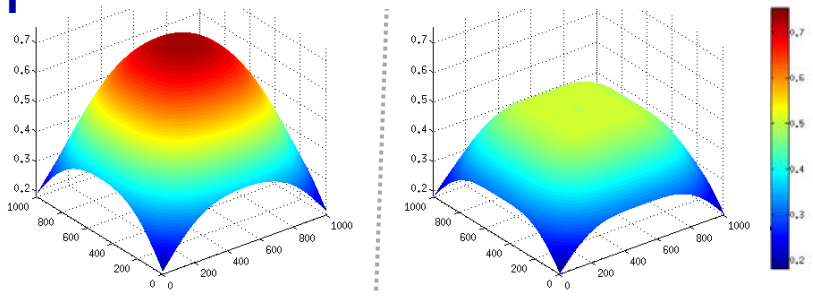
Pattern density distribution



Uniform density distribution

Density distribution conforming to flare distribution

EUV Flare distribution

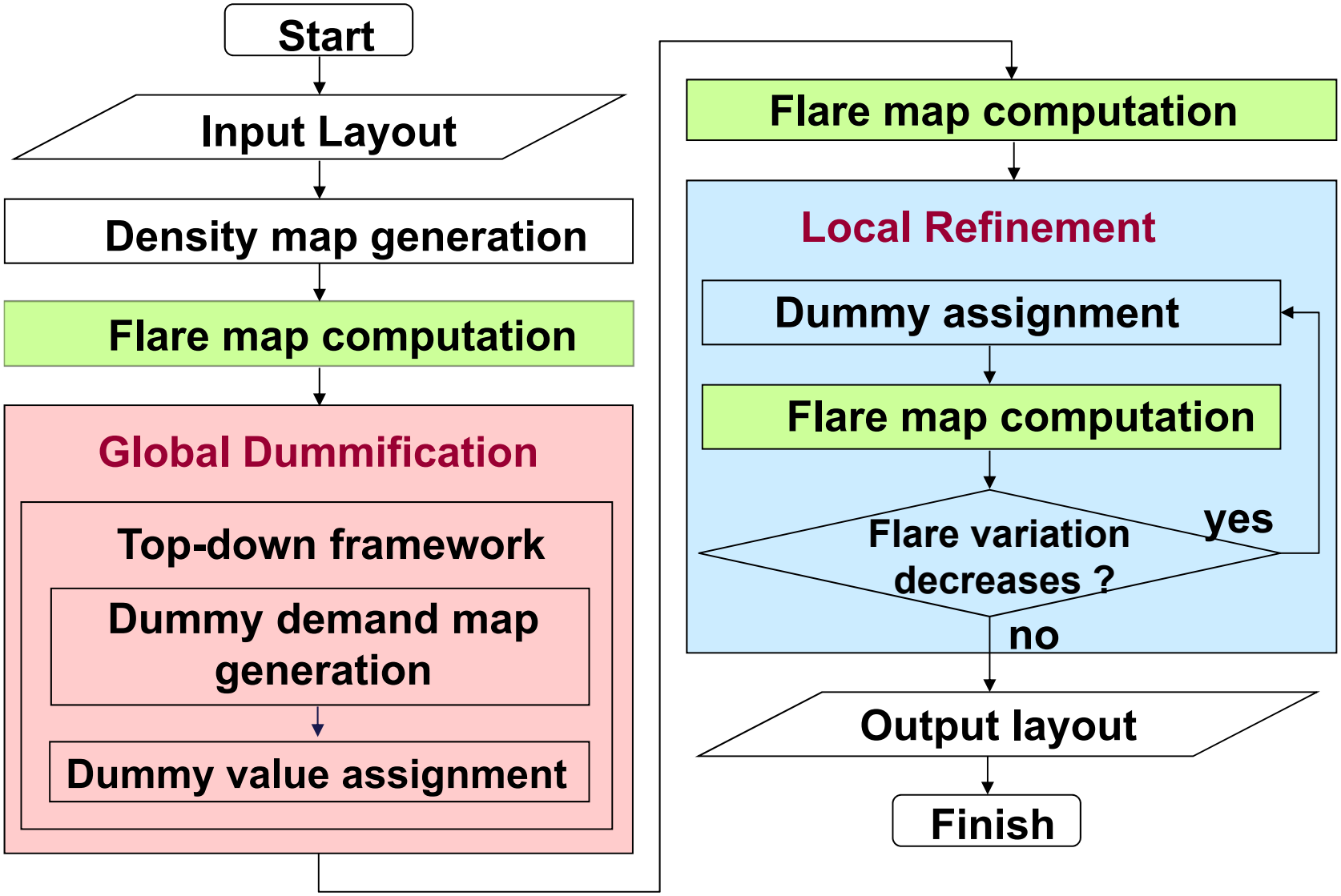


Flare Minimization with Dummification

- Input
  - A grid-based layout
  - Maximum available dummy density of each grid
- Objective
  - Assign a dummy value to each grid to simultaneously minimize the flare level and the flare variation
- Output
  - Inserted dummies in each grid without conflicting to the original design

1st work on flare minimization with dummification [DAC'12, DAC'14]

Dummification Algorithm

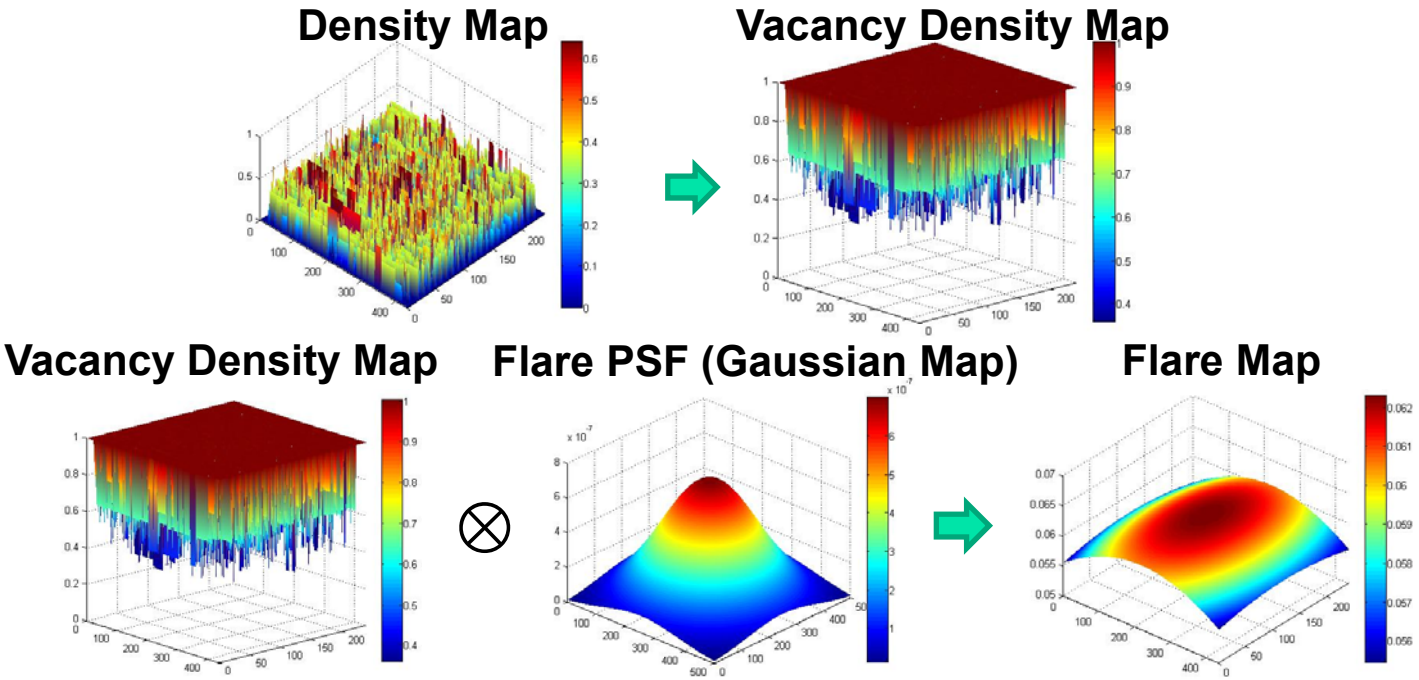


EUV Flare Cost Evaluation

- Use a point spread function (PSF) to model the flare
- A flare map can be computed as

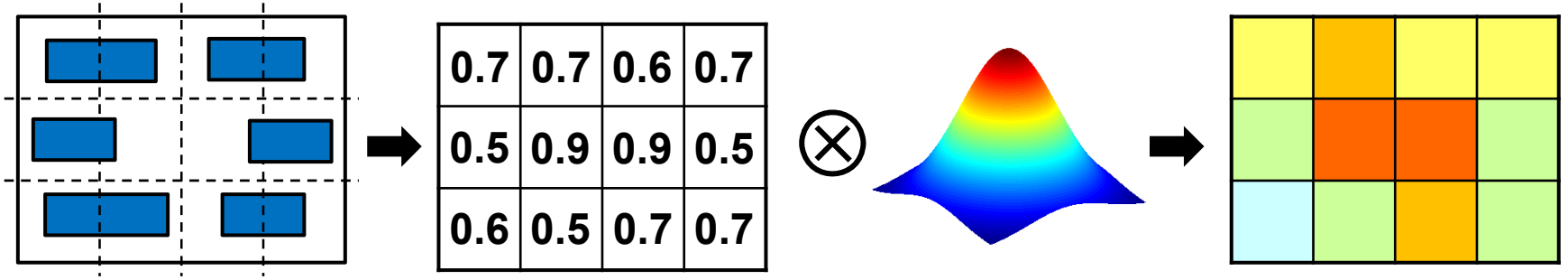
Flare map $I_F(x,y)$ = $I_D(x,y)$ $\otimes$ $PSF(x,y)$ **Flare PSF**

Vacancy density map

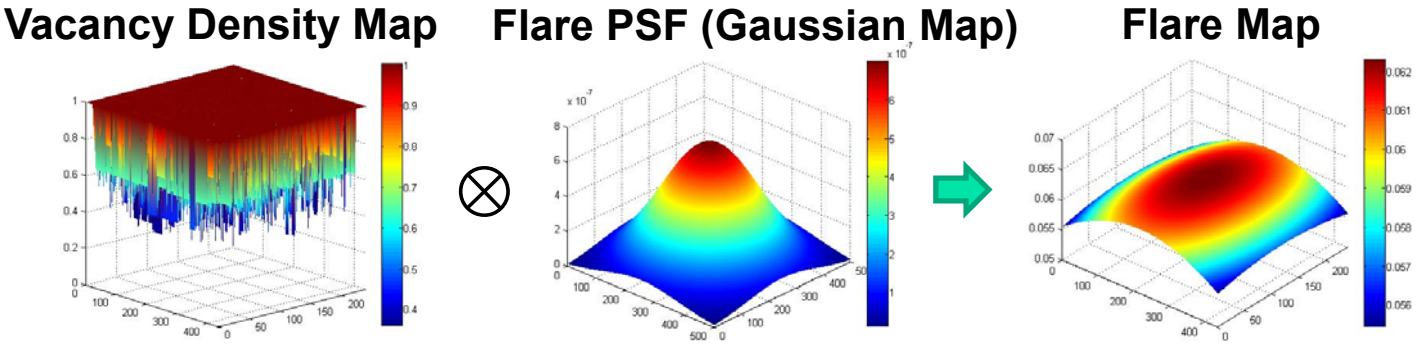


Example Flare Map Computation

Flare map: $I_F(x,y)$ = Vacancy density map: $I_D(x,y)$ $\otimes$ Flare PSF: $PSF(x,y)$



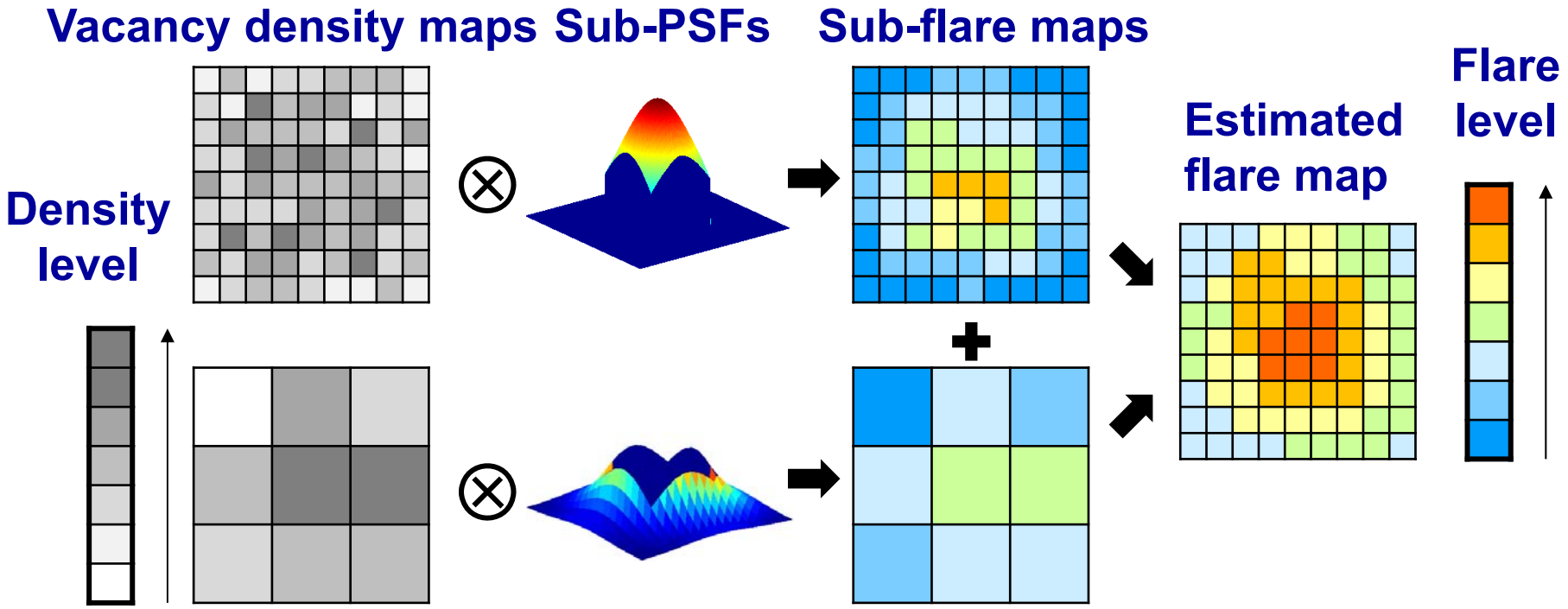
Input layout | Vacancy density map (1 – pattern density) | Flare PSF | Flare map



Full-chip flare map computation is time-consuming!!

Estimated Flare Map Computation

- Apply multiple convolutions with coarsened grids of different sizes to speed up the computation
  - A sub-PSF with larger variation ➡ use smaller grid size
  - A sub-PSF with smaller variation ➡ use larger grid size



$$I_D(x,y) \otimes PSF(x,y) = I_F(x,y)$$

Dummy Demand Map Generation

- A flare reduction map due to dummy fill

$$\underbrace{I_F(x,y) - I'_F(x,y)}_{\text{Flare reduction}} = \underbrace{I_{dummy}(x,y)}_{\text{Dummy density map}} \otimes PSF(x,y)$$

- A dummy demand map is generated by

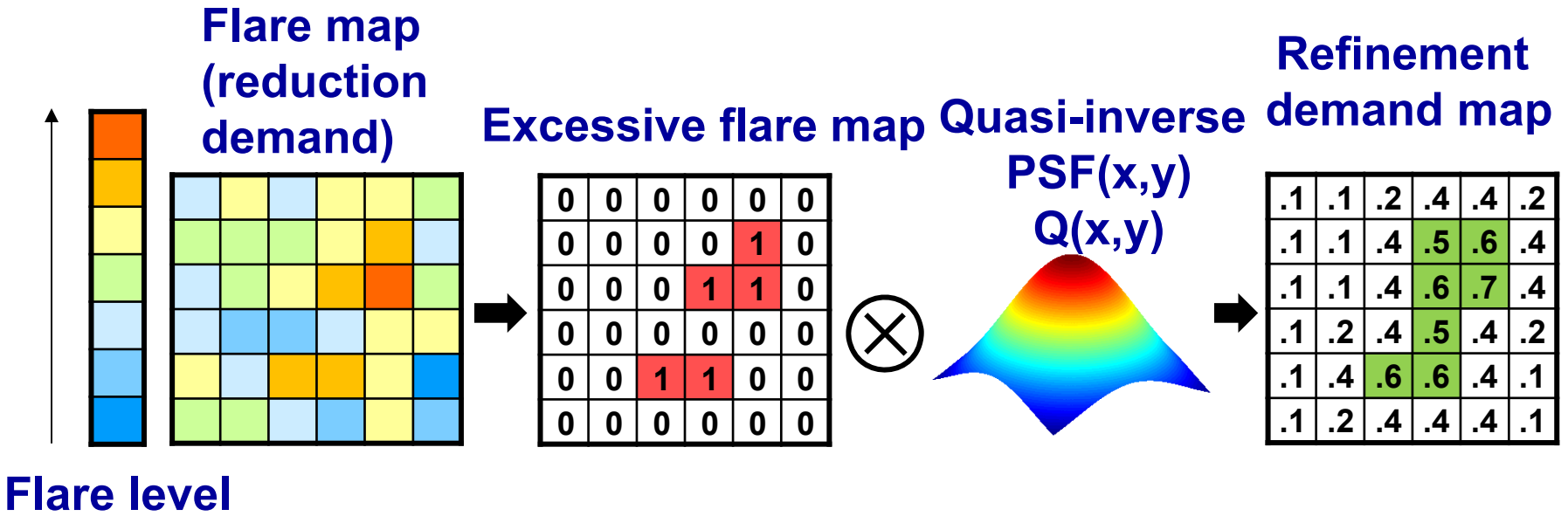
$$\underbrace{D(x,y)}_{\text{Dummy demand map}} = I_F(x,y) \otimes \underbrace{Q(x,y)}_{\text{Quasi-inverse PSF}}$$

$$Q(x,y) = \iint_{-\infty}^{\infty} PSF(x-f, y-g)PSF(x,y)\,df\,dg$$

Insert dummies guided by the dummy demand map

Local Refinement

- Further minimize the flare variation
  - Identify grids with excessive flare values
  - Propagate flare reduction demand of a grid to the refinement demand of neighboring girds
  - Assign available dummy values for those grids with larger demands



Comparison of Flare Mitigation Results

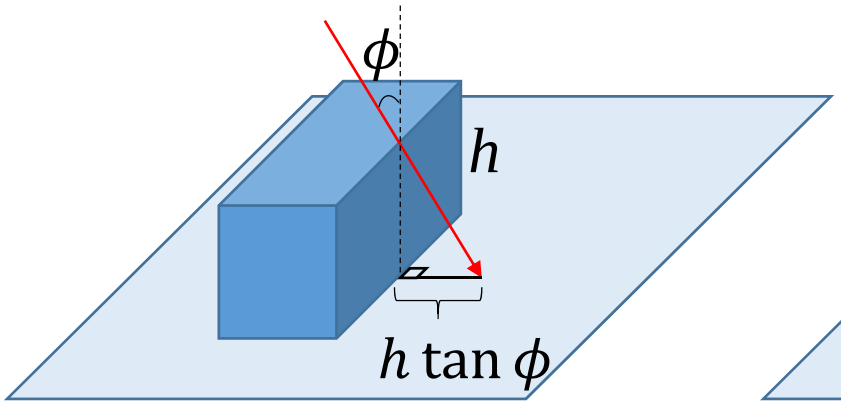
- **36% flare level reduction and 37% flare variation reduction**

Circuit	Original		Linear			Ours		
	Avg.	Var	Avg.	Var.	CPU (s)	Avg.	Var.	CPU (s)
Struct	0.850	0.743	0.606	0.535	66	0.382	0.288	155
Primary1	0.877	0.748	0.626	0.588	93	0.381	0.282	221
Primary2	0.898	0.745	0.645	0.627	157	0.388	0.283	417
S5378	0.061	0.007	0.045	0.005	16	0.029	0.003	23
S9234	0.054	0.005	0.040	0.004	17	0.025	0.003	22
S13207	0.129	0.031	0.094	0.022	18	0.061	0.015	37
S15850	0.143	0.038	0.104	0.028	17	0.068	0.018	39
S38417	0.289	0.161	0.207	0.110	19	0.139	0.077	78
S38584	0.322	0.205	0.230	0.138	21	0.159	0.099	95
Dma	0.101	0.016	0.073	0.011	19	0.043	0.007	31
Dsp1	0.258	0.102	0.184	0.071	20	0.115	0.045	59
Dsp2	0.224	0.076	0.160	0.054	19	0.098	0.033	52
Risc1	0.407	0.262	0.288	0.160	22	0.190	0.122	24
Risc2	0.386	0.235	0.274	0.161	22	0.183	0.109	100
Avg.	**-**	**-**	**1.00**	**1.00**	**-**	**0.64**	**0.63**	**-**

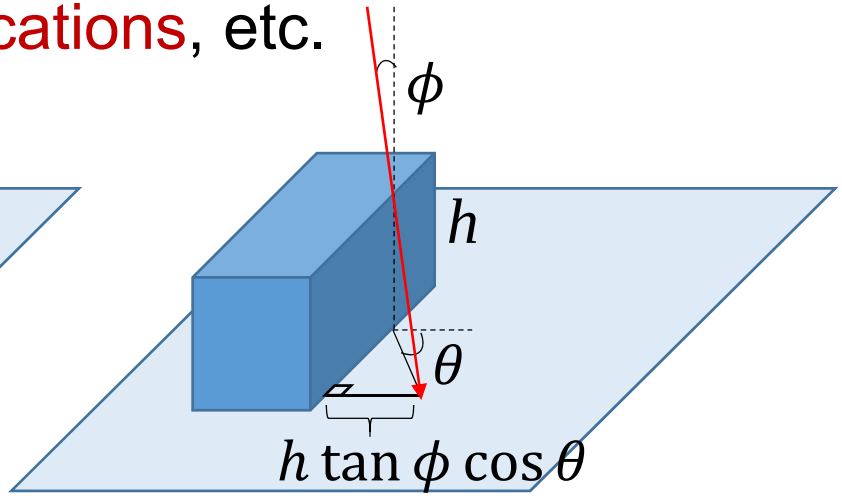
- **linear dummification: assign the maximum available dummy values to the central grids, and zero dummy value to the farthest grids**

EUV Shadowing Effect

- EUV incident light has an angle ($\phi \approx 6^{o}$) relative to the axis perpendicular to the mask plane.
  - Oblique illumination and absorber (pattern) thickness result in shadows around a pattern and thus make a pattern wider during exposure
- Pattern distortion due to shadowing effect depends on pattern shapes, pattern locations, etc.



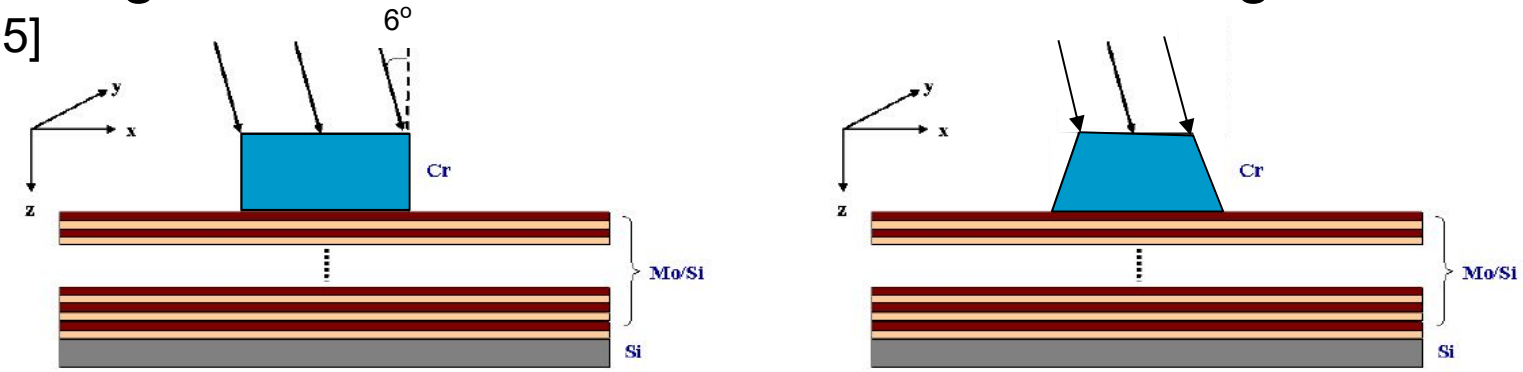
ray is perpendicular to pattern edge



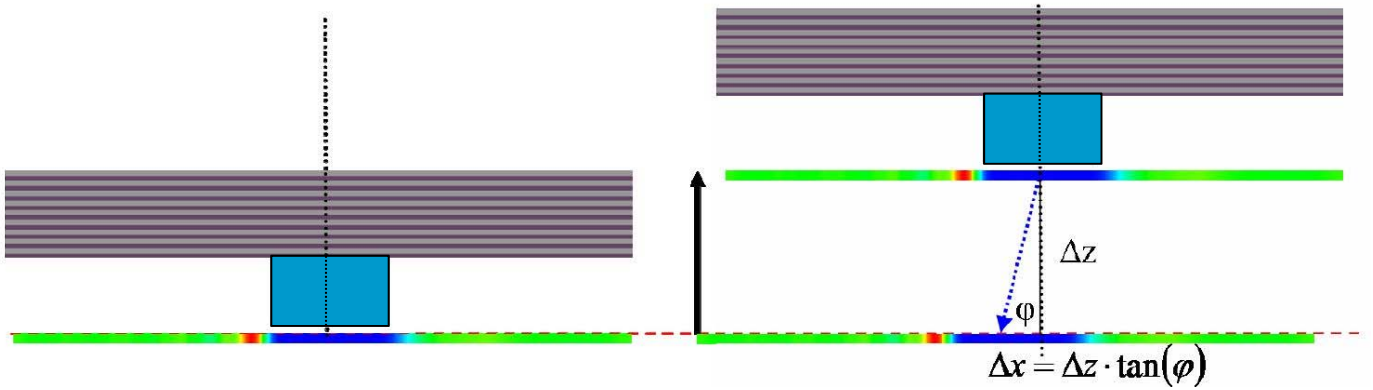
non-zero angle θ between the chief ray and the normal of pattern edge

Rule-Based Shadowing Effect Handling

- Still not much work on layout design to optimize the shadowing effect
- Change mask structure/absorber sidewall angle [Woo, et al., '05]



- Shift patterns [Schmoeller, et al., '08]



Research Directions on EUV

Objectives

Techniques:

Wire sizing/shaping

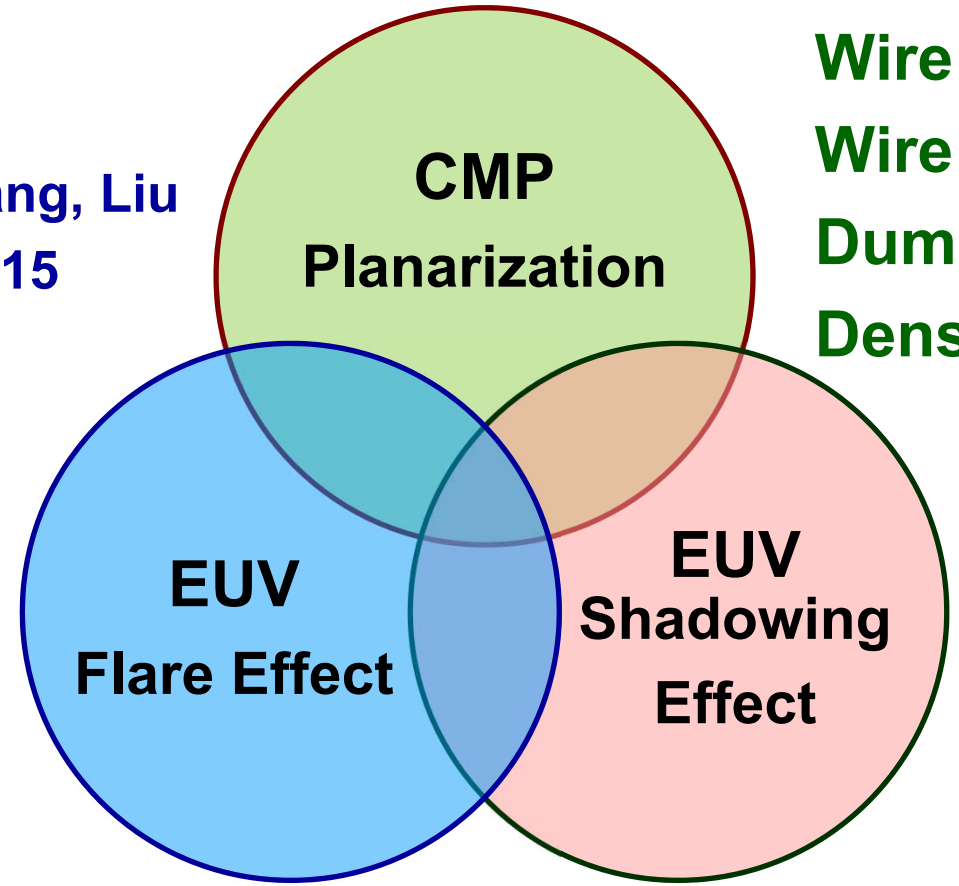
Wire perturbation

Dummification

Density control

Chang, Fang, Liu

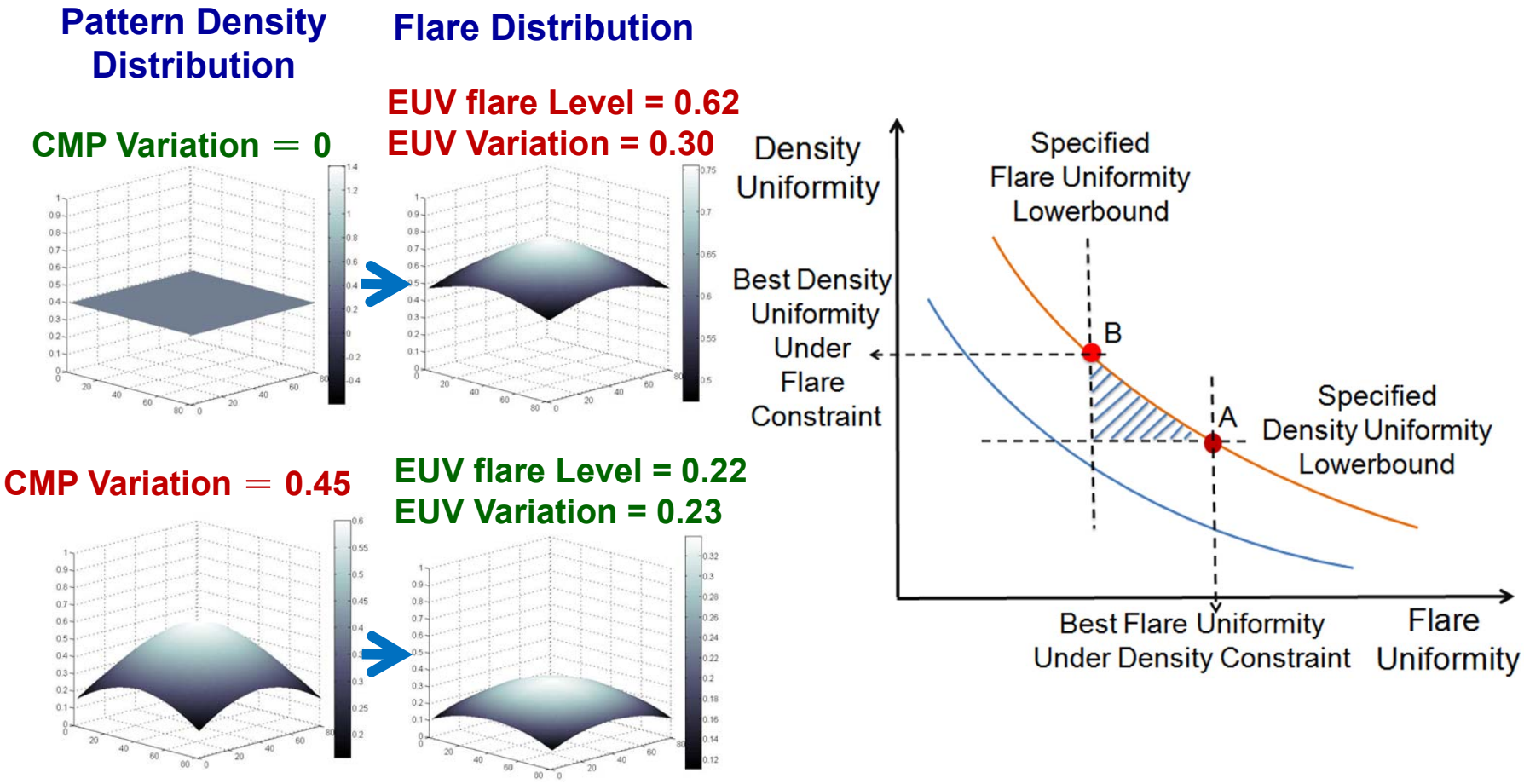
DAC-15



Flow: Placement, Routing, Post-layout optimization

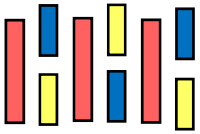
Example: EUV Flare and CMP Co-Optimization

- Conflicting pattern density requirements between CMP planarization and EUV flare optimization

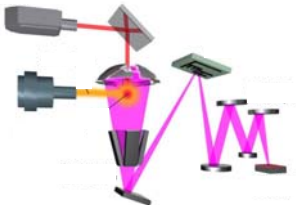


Outline

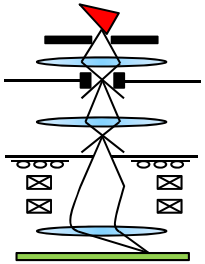
Lithography Basics



Multiple Patterning

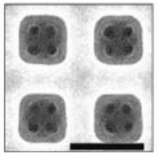


Extreme Ultraviolet (EUV)



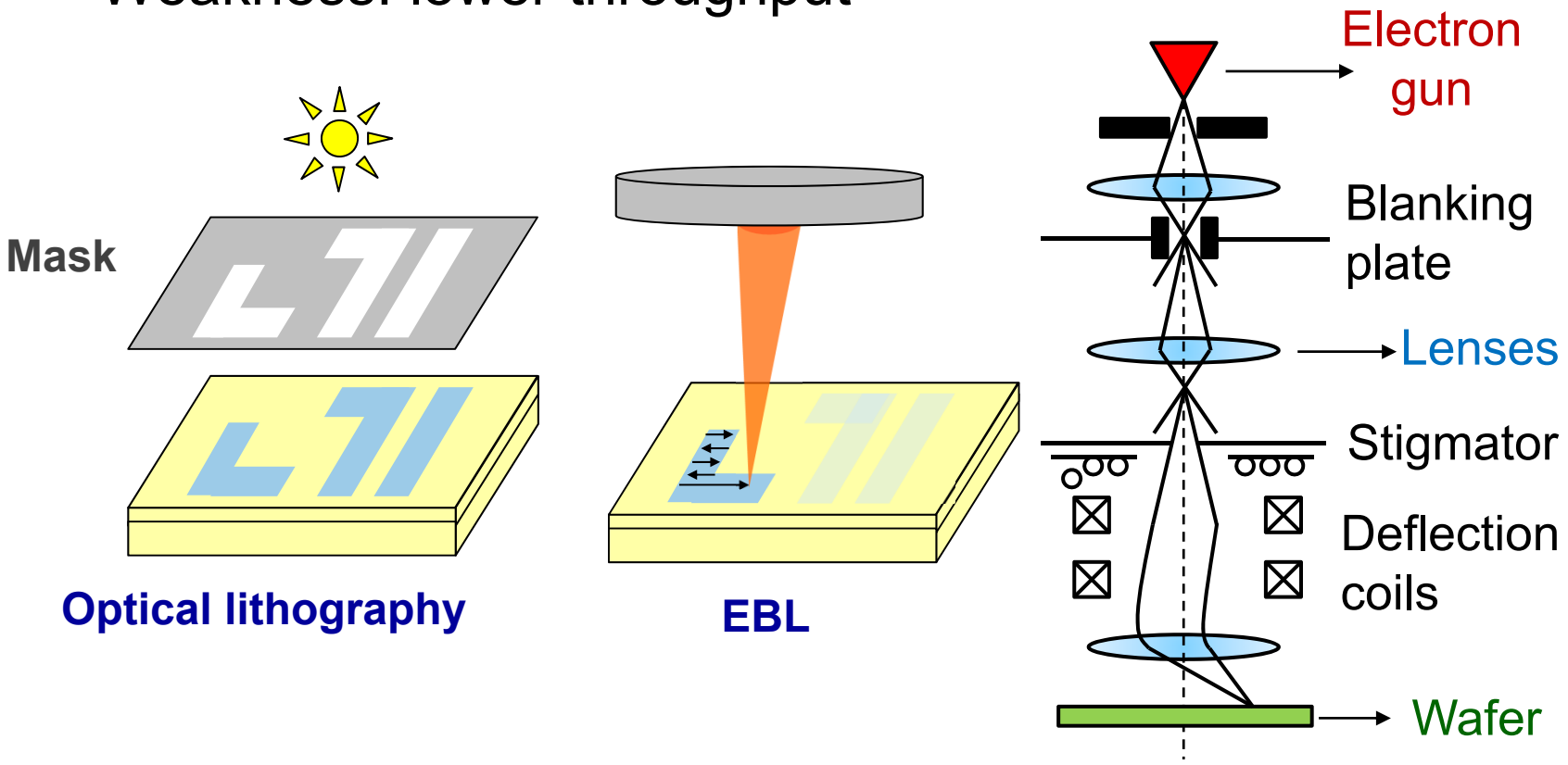
Electron Beam (e-beam)

Directed Self-Assembly (DSA)



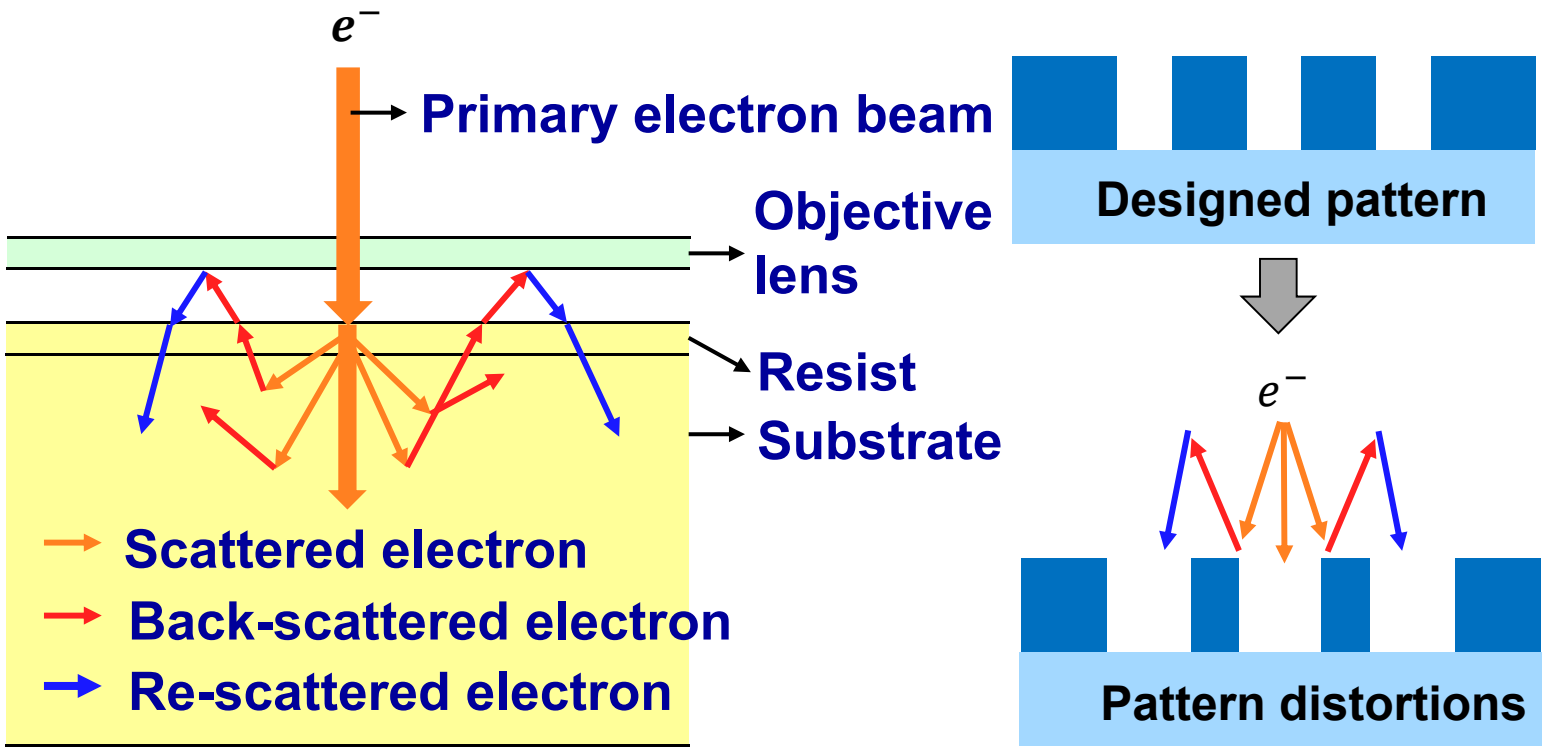
Electron Beam Lithography (EBL)

- E-beam direct write is one of the most promising next-generation lithography technologies
- Advantages: maskless, high resolution
- Weakness: lower throughput



Scattered Electrons

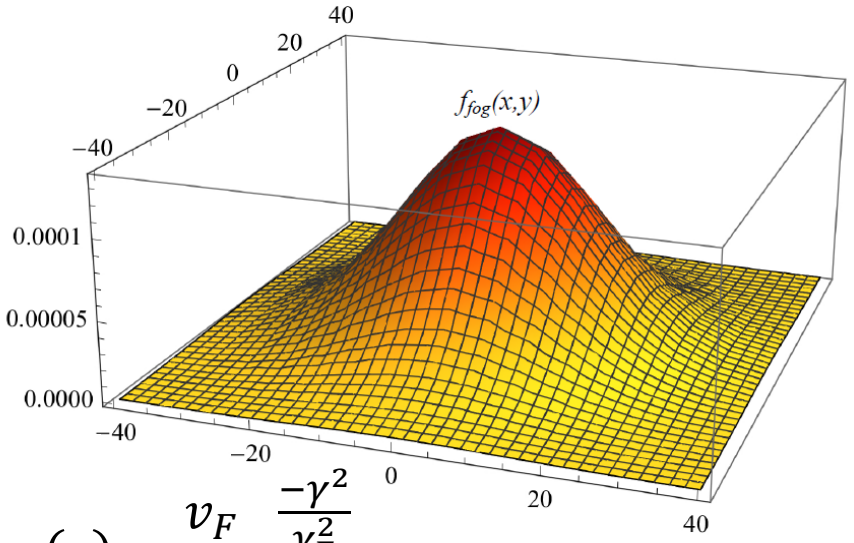
- Electrons are scattered in the resist and substrate, with various directions



- Re-scattered electrons might induce fogging effects
- The effect causes over exposure & pattern variations

Fogging Effect

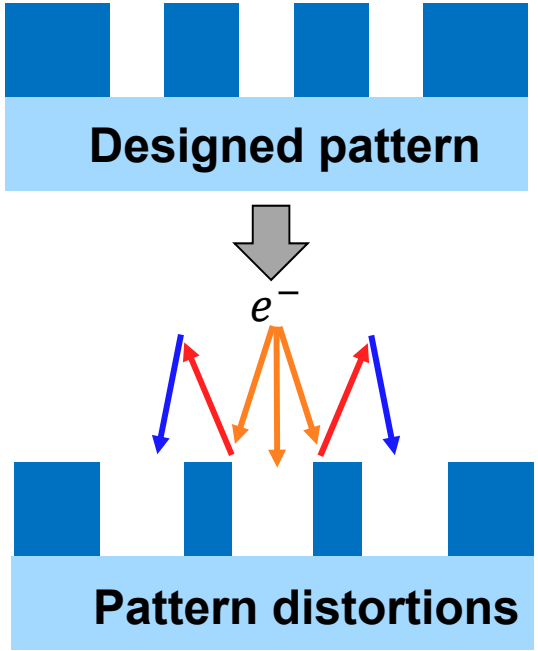
- Fogging effect is long-range (mm), low-intensity
- Accumulated over exposure causes pattern variations
- A point spread function (PSF): Gaussian distribution



$$f_{fog}(r) = \frac{v_F}{\gamma_F^2} e^{\frac{-\gamma^2}{\gamma_F^2}}$$

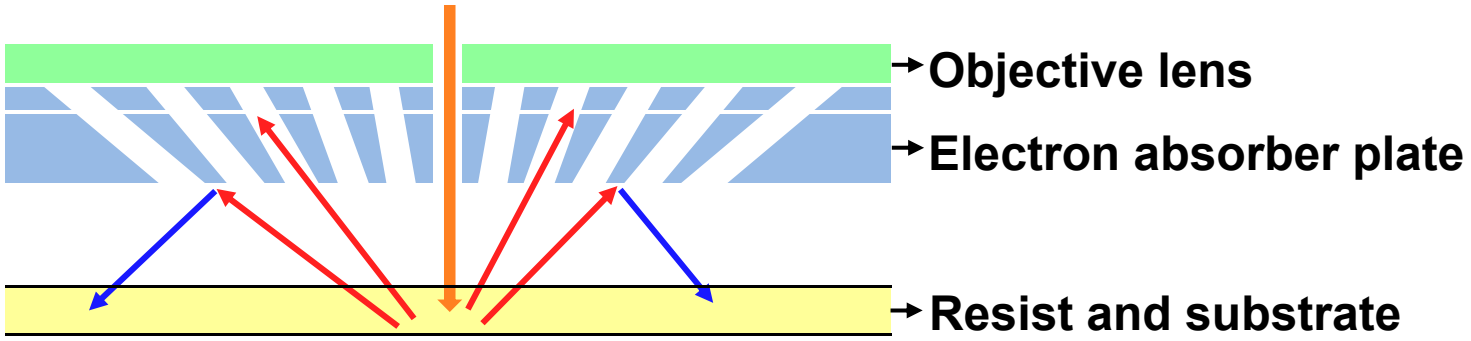
r : distance from exposure point

v_F: Weight of the effect; γ_F : Range of the effect



Prior Work on Correction

- Fogging effect correction in manufacturing process
  - Very thin substrates: [Pease, *Contemporary Physics*, 1981]
  - Electron absorber plate: [Shimomura *et al.*, *SPIE*, 1999]
  - Cannot fully eliminate the re-scattered electrons



- Fogging effect correction during the design stage
  - Dose modulation: [Pease, *Contemporary Physics*, 1981], [Figueiro *et al.*, *SPIE,* 2013]
  - Geometry modulation: [Hudek *et al.*, *MNE*, 2006], [Figueiro *et al.*, *SPIE,* 2013]
  - Convolution computation is time-consuming

New Contributions

- Huang & Chang, “Fogging effect aware placement in electron beam lithography,” DAC-17
- Chen, Chang, Huang, “Analytical placement considering the electron beam fogging effect,” TCAD 2020
- Present the first work to consider the e-beam fogging effect during placement
- Propose a movable fogging source model and adopt a fast Gauss transform with Hermite expansion to approximate the fogging effect efficiently, yet sufficiently accurate
- Experimental results show that our algorithm can effectively reduce fogging variation by 12%

Fogging Effect Computation

- Traditional: Convolve pattern density map and PSF

$$I_f(x,y) = I_d(x,y) * f_{fog}(x,y)$$

fogging **pattern density map** **PSF**

  - Huge quadratic time for large-scale designs
  - Not feasible for placement optimization

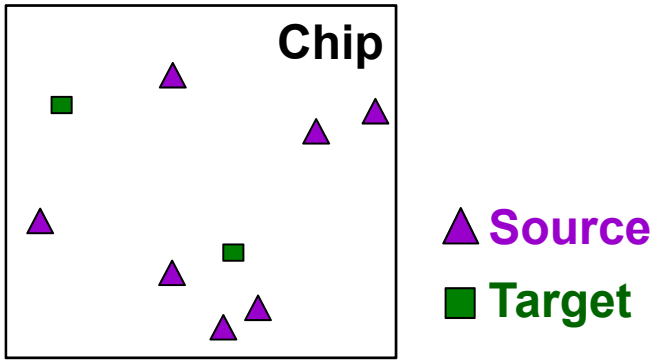
- This work: Fast Gauss transform for approximation + Hermite expansion for runtime reduction
  - [Greengard and Strain, *SISC*, 1991]
  - Linear time practically, good for placement optimization

Fast Gauss Transform + Hermite Expansion

- For every target t_i, the fast Gauss transform computes
  - Sources $S = \{s_1, s_2, \dots, s_{N_s}\}$: Gaussian distributions
  - Targets $T = \{t_1, t_2, \dots, t_{N_t}\}$: Summations of Gaussian distributions

$$G_{t_i}(s) = \sum_{j=1}^{N_s} q_j e^{\frac{-|t_i - s_j|^2}{\delta}}$$

Weight of sources



- Hermite function: $h_n(t) = (-1)^n D^n e^{-t^2}$
- Generating function for Hermite polynomials:

$$e^{-(t-s)^2} = \sum_{n=0}^{\infty} \frac{s^n}{n!} h_n(t)$$

- Expansion at s_0: $e^{\frac{-(t-s_0-(s-s_0))^2}{\delta}} = \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{s-s_0}{\sqrt{\delta}}\right)^n h_n\left(\frac{t-s_0}{\sqrt{\delta}}\right)$

Hermite Expansion

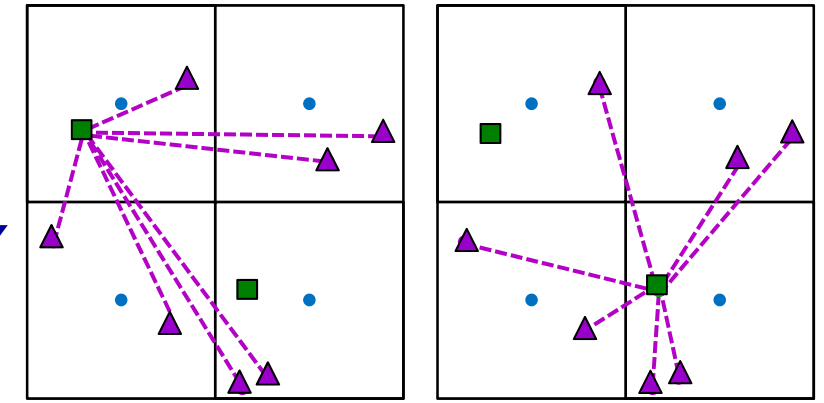
- By $e^{-(t-s)^2} = \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{s-s_0}{\sqrt{\delta}}\right)^n h_n\left(\frac{t-s_0}{\sqrt{\delta}}\right)$

- For every target t_i,

$$G_{t_i}(s) = \sum_{j=1}^{N_s} q_j e^{\frac{-|t_i - s_j|^2}{\delta}}$$

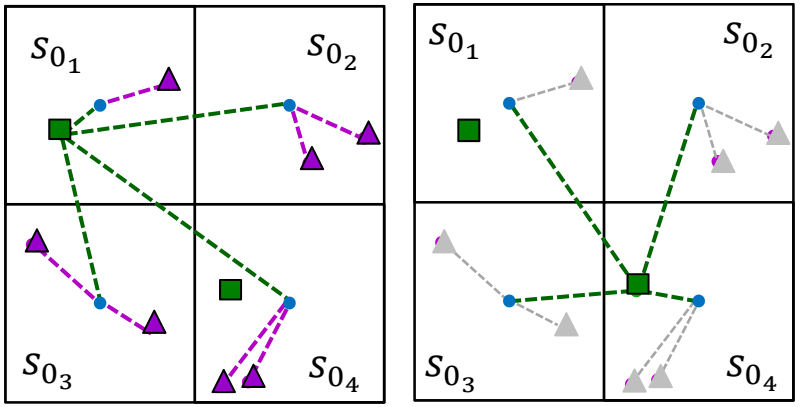
$$= \sum_{n=0}^{\infty} A_n \, h_n\left(\frac{t_i - s_0}{\sqrt{\delta}}\right)$$

$$A_n = \frac{1}{n!} \sum_{j=1}^{N_s} q_j \left(\frac{s_j - s_0}{\sqrt{\delta}}\right)^n$$



$O(N_s N_t)$ **time**

▲ **Source** ■ **Target** ● **Expansion point**

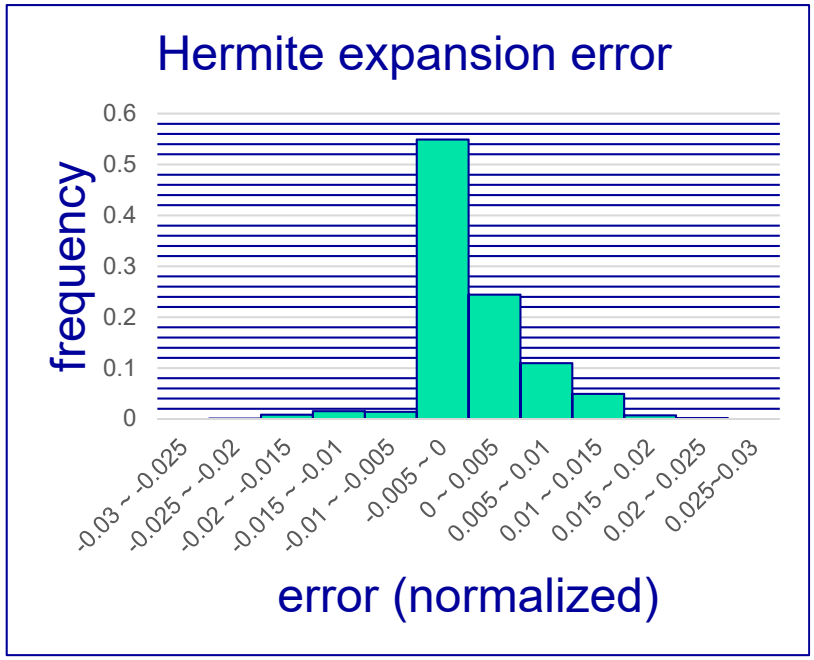


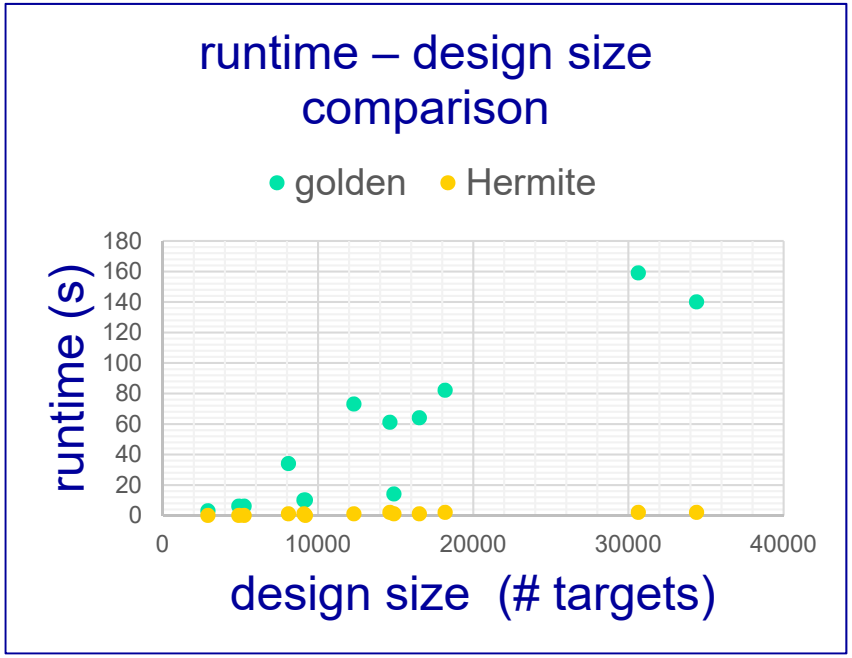
$O(N_s + N_t)$ **time**

- Error:
  - $|E_{Hermite}(p)| = \sum_{n>p} A_n h_n\left(\frac{t_i - s_0}{\sqrt{\delta}}\right) \leq \frac{1}{p!}(1.09)^2 \sum_{j=1}^{N_s} q_j \left(\frac{r^{p+1}}{1-r}\right)^2, \; r = \frac{w_g}{\sqrt{2\delta}} \leq \frac{1}{2}$

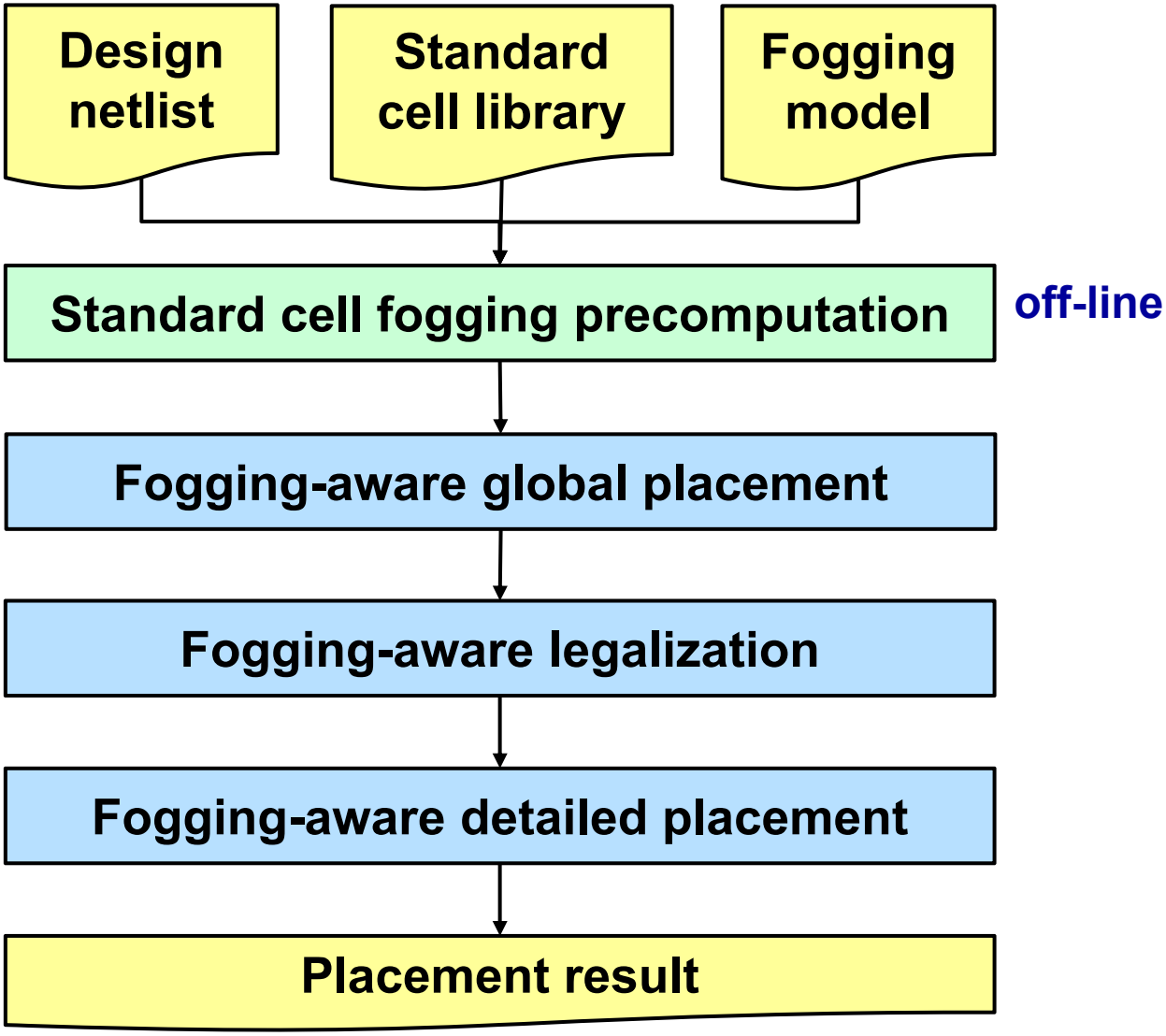
Accuracy and Efficiency

- Absolute average error: 0.33%
  - Most computation errors lie in the range [-0.5%, 0.5%]
- Average speedup: 26.5X



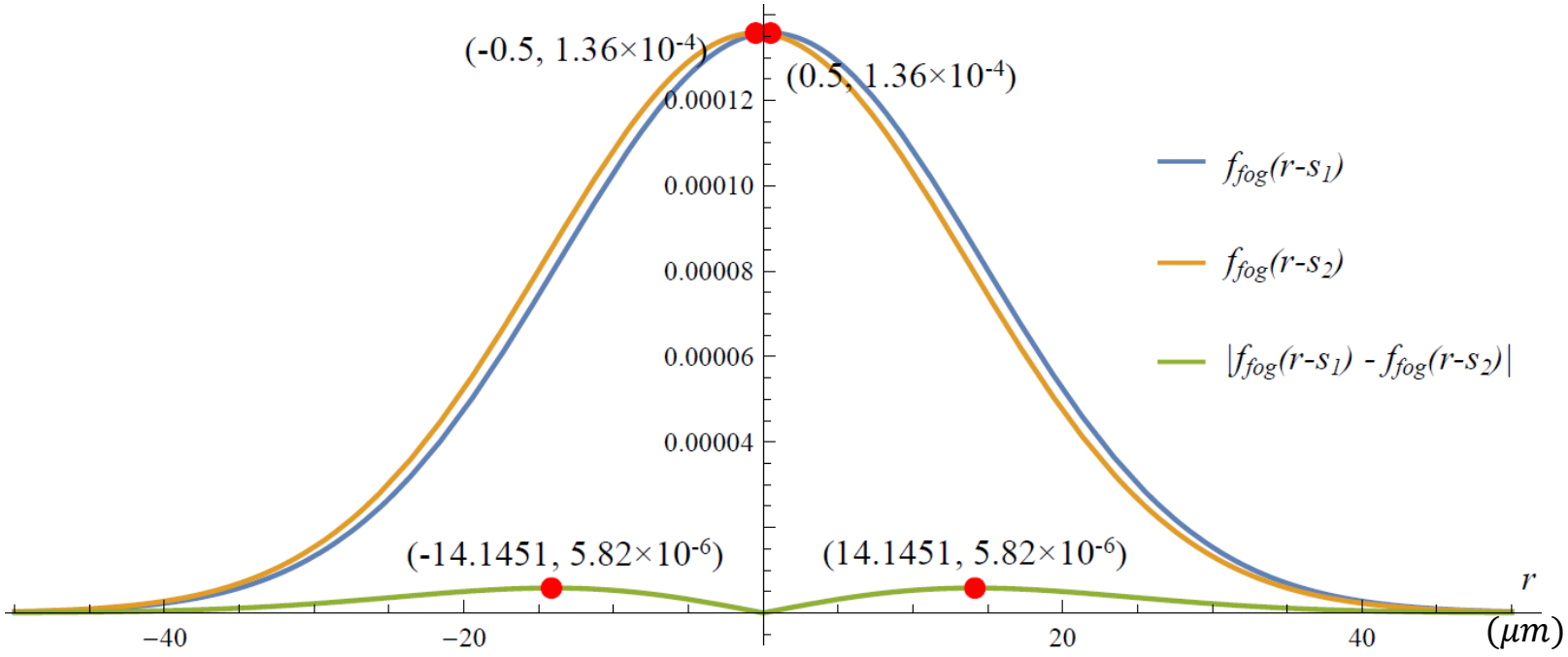


Algorithm Flow



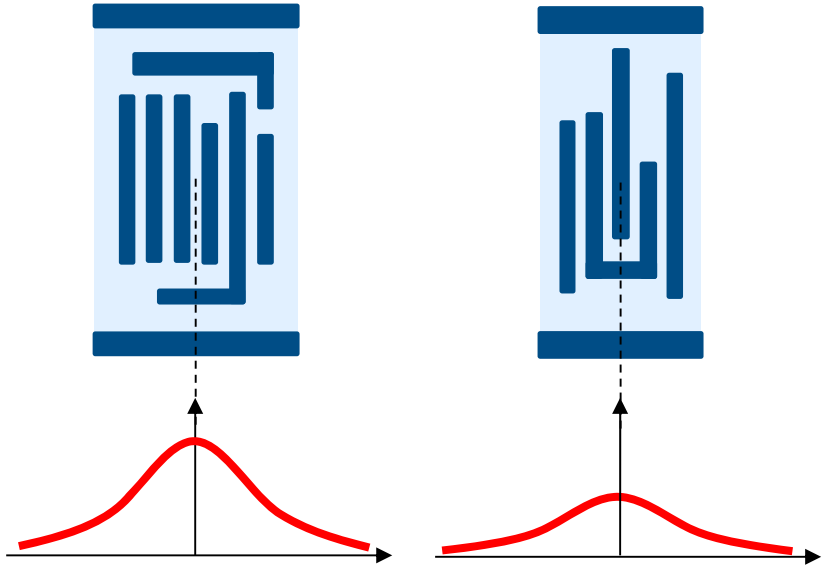
Fogging Source Model

- Fogging effect is a long range effect
- Energy distribution changes slightly in nearby locations
- Two exposures centered at nearby locations have similar energy distributions



Fogging Source Model Approximation

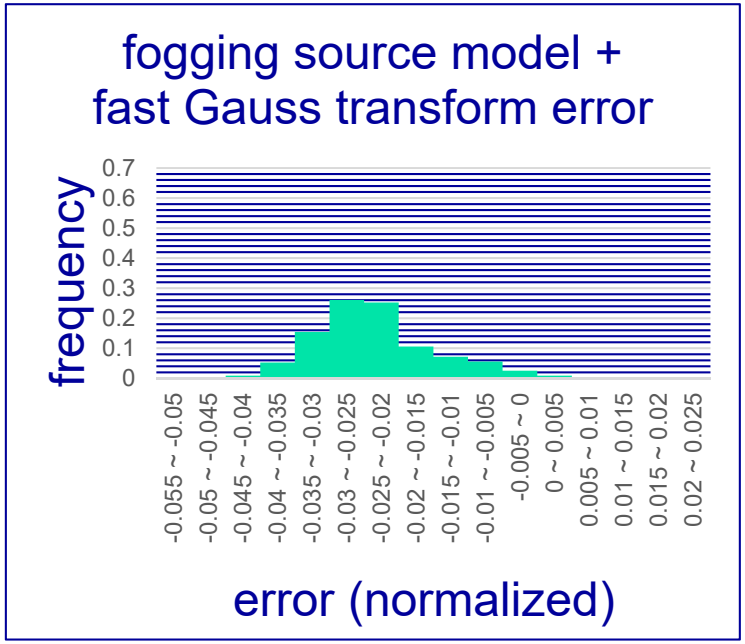
- Cell sizes are about 1 um [NanGate 15 nm cell library]
  - Very small, compared to mm for fogging range
- Exposures to write a cell are all considered at its center
- Sufficient to measure the pattern area of a cell for fogging intensity modeling (off-line computation!!)

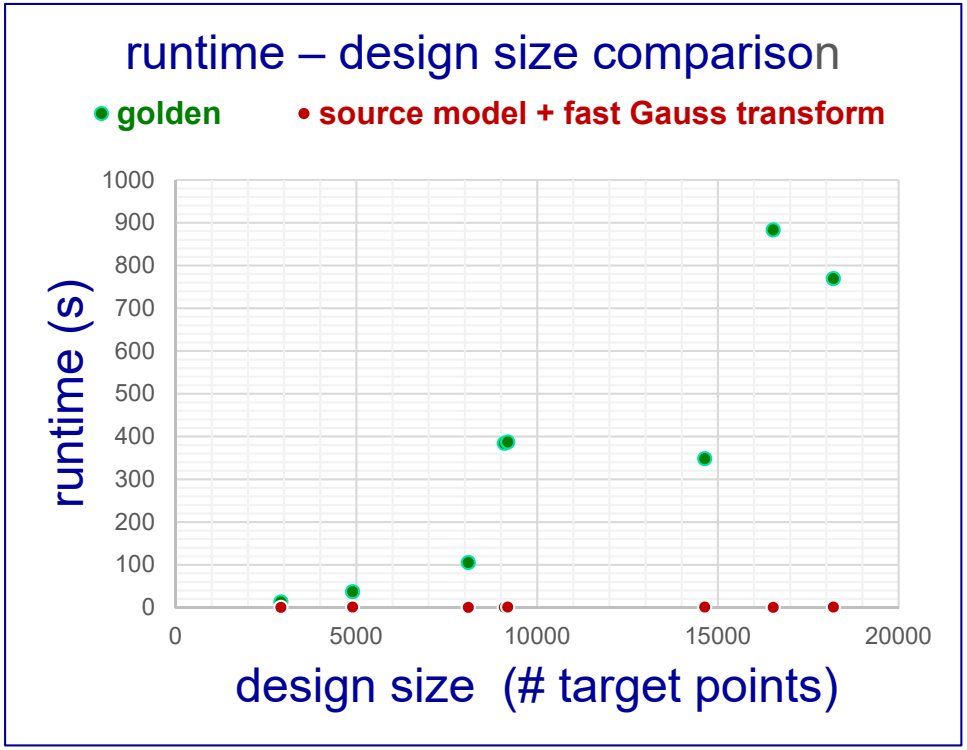


fogging effect intensity ∝ pattern area

Modeling Accuracy & Efficiency

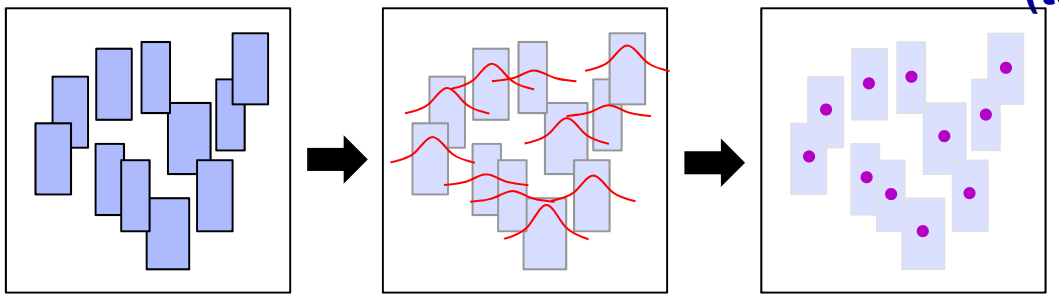
- **Golden:** Conventional convolution
- Fogging source model + fast Gauss transform
  - 2.35% absolute average error, 30.2X speedup

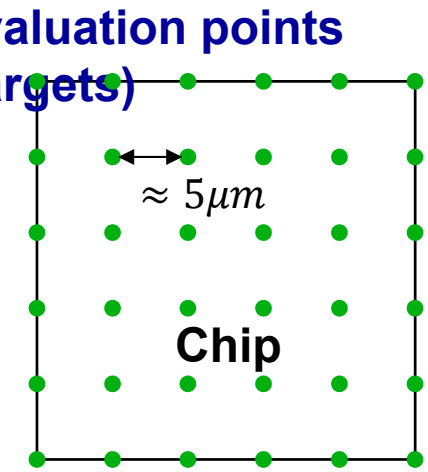




Global Placement

- Main stage to tackle the global fogging effect
- Movable cells (fogging sources): “sources” of the fast Gauss transform $S = \{s_1, s_2, \dots, s_{N_s}\}$
- Uniform evaluation points: “targets” of the fast Gauss transform $T = \{t_1, t_2, \dots, t_{N_t}\}$



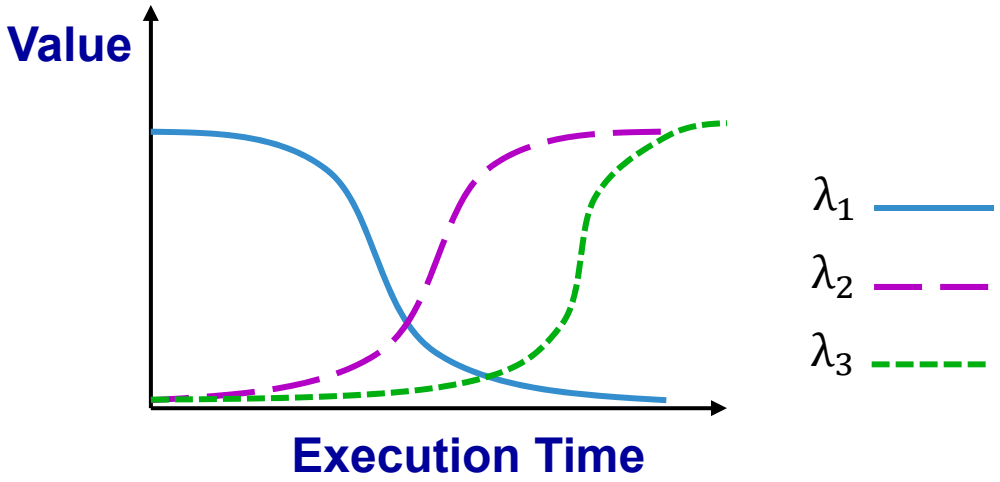


- Minimize fogging variation $S_f(x, y)$
  - $S_f(x, y) = \frac{1}{N_t} \sum_{t_i \in T} \left(G_{t_i}(x, y) - \frac{\sum_{t_i \in T} G_{t_i}(x,y)}{N_t}\right)^2$
  - Fogging effect at target t_i: $G_{t_i}(x, y) = \sum_{n \le p} A_n \, h_n\left(\frac{t_i - s_0}{\sqrt{\delta}}\right)$

Global Placement: Objective Function

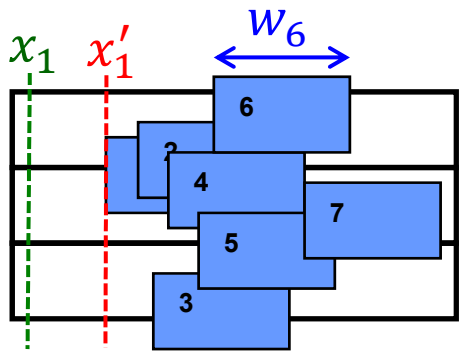
$$\min_{x,y} \lambda_1 W(x,y) + \lambda_2 \sum_{b \in B} (D_b(x,y) - A_b)^2 + \lambda_3 S_f(x,y)$$

- λ_1 decreases along the execution, while λ_2 and λ_3 increase
- The fogging variation is considered in later iterations
  - Not to degrade the wirelength



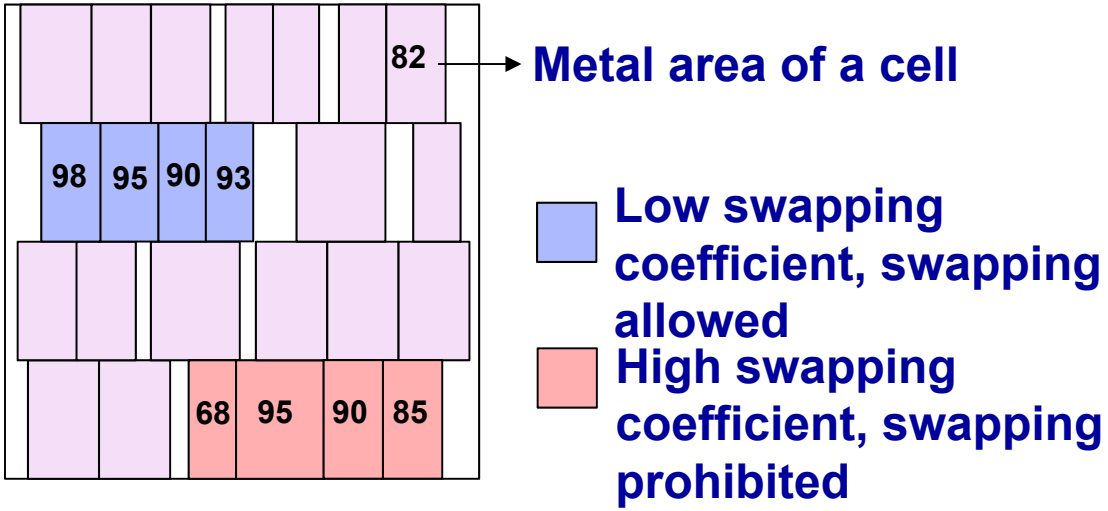
Legalization

- Legalize the global placement result with minimized displacement
  - Extend Abacus [Spindler *et al.*, *ISPD*, 2008]
- Decide the positions for standard cells in a row
  - $\min_{x} \sum_{i=1}^{N_r} k_i (x_i - x_i')^2$
  - subject to $x_i - x_{i-1} \geq w_{i-1}, i \in N, 2 \leq i \leq N_r$
- Make the weight of a cell (k_i) proportional to its pattern area



Detailed Placement: Cell Swapping

- Consider all permutations of m selected cells, and pick the best for swapping
- Swap cells with low swapping coefficient to minimize fogging variations
  - $\delta = \frac{\max(a_i) - \min(a_i)}{\max(a_i)}, 1 \leq i \leq m, a_i$ is the pattern area in cell i
  - The smaller the coefficient δ, the smaller the fogging variation



Experimental Setup

- Platform: C++ programming language on Intel Xeon 2.93GHz Linux workstation with 49GB memory
- 16 Benchmarks: 2015 ISPD Blockage-Aware Detailed Routing-Driven Placement Contest, w/o fence regions

Circuit	#blocks	#movable cells	#nets	util
mgc_des_perf_1	113018	112644	112878	0.906
mgc_des_perf_a	108666	108288	110281	0.567
mgc_des_perf_b	113018	112644	112878	0.563
mgc_edit_dist_a	129993	127413	131134	0.541
mgc_fft_1	35291	32281	33307	0.835
mgc_fft_2	35291	32281	33307	0.5
mgc_fft_a	33641	30625	32088	0.285
mgc_fft_b	33641	30625	32088	0.309
mgc_matrix_mult_1	160127	155325	158527	0.802
mgc_matrix_mult_a	154457	149650	154284	0.449
mgc_matrix_mult_b	151244	146435	151612	0.342
mgc_pci_bridge32_a	29882	29517	29985	0.64
mgc_pci_bridge32_b	29281	28914	29417	0.273
mgc_superblue11_a	984445	925616	935567	0.351
mgc_superblue12	1292945	1286948	1293410	0.449
mgc_superblue16_a	698367	680450	697224	0.502

Placement Comparisons with NTUplace4dr

- Reduce the average **fogging variation** by **12%**, at the costs of 1.7% longer wirelength and 2.4X runtime

Circuit	Fogging variation		Wirelength		Runtime (s)	
	NTUplace4dr	Ours	NTUplace4dr	Ours	NTUplace4dr	Ours
mgc_des_perf_1	1744.02	1667.78	1241925098	1262736148	392	987
mgc_des_perf_a	2238.26	1680.15	2777735834	2777989749	398	729
mgc_des_perf_b	808.14	767.44	1755427203	1677313973	462	844
mgc_edit_dist_a	1142.05	1232.46	5141106266	4886104918	550	933
mgc_fft_1	4234.82	3260.09	466703004	487011129	81	207
mgc_fft_2	1941.73	1530.54	444506960	479525095	108	213
mgc_fft_a	3847.82	2794.8	695359371	798126451	121	456
mgc_fft_b	7071.04	6000.65	871813697	977414060	99	424
mgc_matrix_mult_1	3059.94	2890.65	2292304916	2324407986	366	863
mgc_matrix_mult_a	6965.44	7514.71	3725961351	3620128812	431	823
mgc_matrix_mult_b	4848.77	4856.71	3501375316	3459048487	346	519
mgc_pci_bridge32_a	2603.29	1089.27	415101912	430491647	147	547
mgc_pci_bridge32_b	1407.18	1265	638911584	647428485	109	411
mgc_superblue11_a	8870.58	8489.38	60278170217	57422591986	21651	25963
mgc_superblue12	3506.19	3328.99	37576928082	37480090255	10634	14523
mgc_superblue16_a	2157.55	2202.61	32592590628	32033223346	5034	6607
Comp.	1	**0.884**	1	**1.017**	1	**2.349**

Runtime Analysis

- Runtime comparison between small and large cases
  - Much more efficient for large testcases (only 1.29X slower)

Small testcases	runtime **ratio**	**Large** testcases	runtime **ratio**
mgc_des_perf_1	2.52	mgc_superblue11_a	1.20
mgc_des_perf_a	1.83	mgc_superblue12	1.37
mgc_des_perf_b	1.82	mgc_superblue16_a	1.31
mgc_edit_dist_a	1.70		
mgc_fft_1	2.56		
mgc_fft_2	1.97		
mgc_fft_a	3.77		
mgc_fft_b	4.28		
mgc_matrix_mult_1	2.36		
mgc_matrix_mult_a	1.91		
mgc_matrix_mult_b	1.50		
mgc_pci_bridge32_a	3.72		
mgc_pci_bridge32_b	3.77		
average	**2.59**	average	**1.29**

< 500K cells **> 500K cells**

Proximity and Fogging Modeling

- The deposited energy distribution can be modeled by the point spread function (PSF):

$$PSF(r) = \frac{1}{\pi(1+\eta)}\left(\frac{1}{\beta_f^2}\exp\left(\frac{-r^2}{\beta_f^2}\right) + \frac{\eta}{\beta_b^2}\exp\left(\frac{-r^2}{\beta_b^2}\right) + \frac{\nu}{\beta_F^2}\exp\left(\frac{-r^2}{\beta_F^2}\right)\right)$$

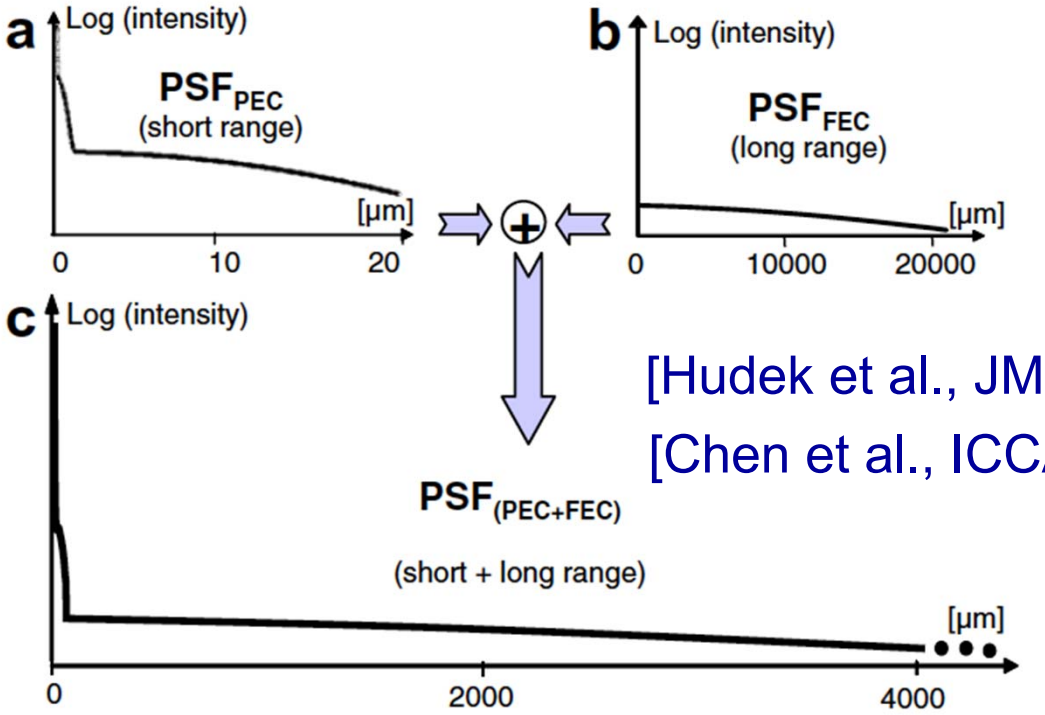
Proximity

Fogging

Forward

Backward

Re-scattering



Scattering ranges:

$\beta_f = 0.06\ \mu m$

$\beta_b = 30\ \mu m$

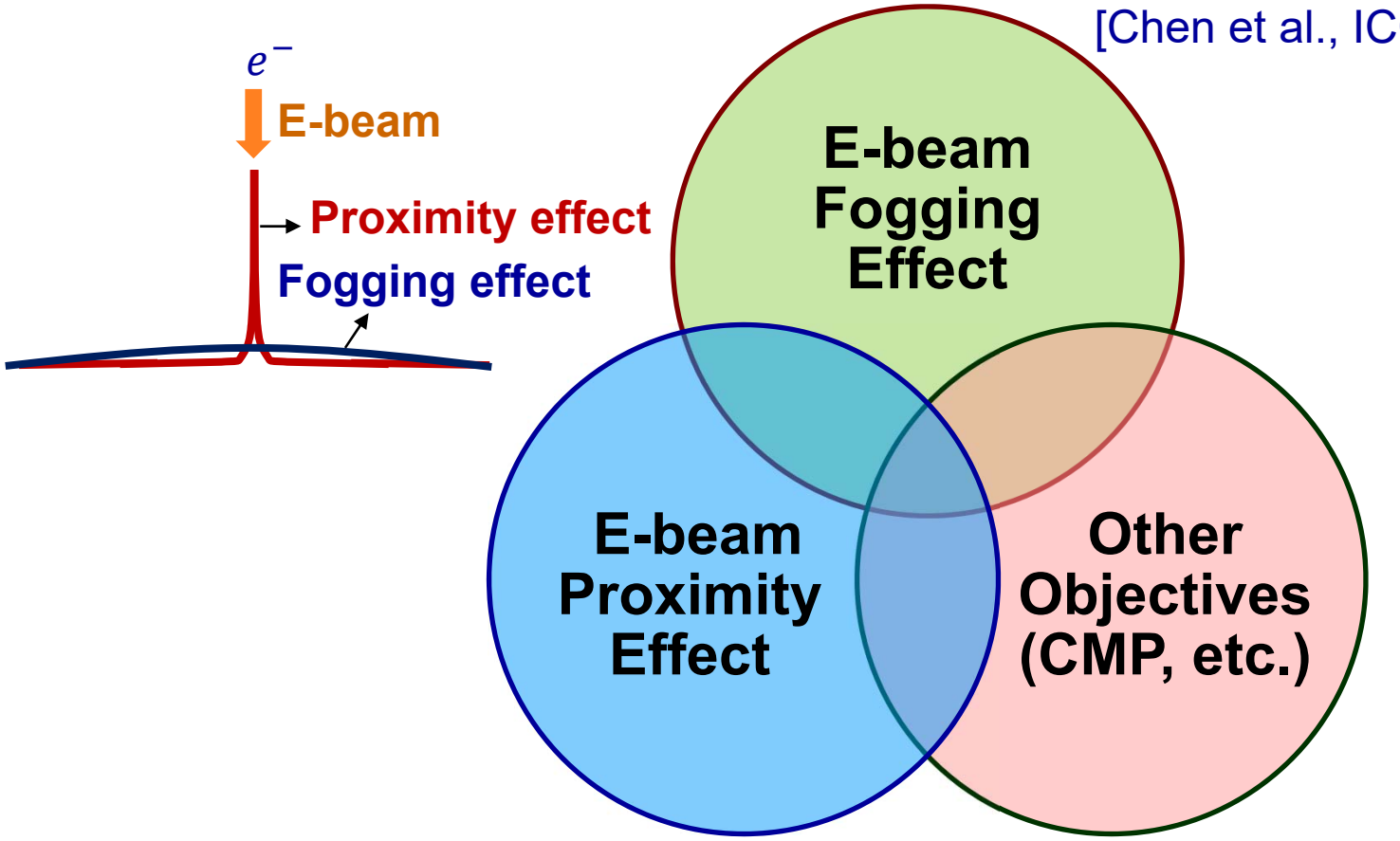
$\beta_F = 20000\ \mu m$

[Hudek et al., JME-07]

[Chen et al., ICCAD-18]

Future Work

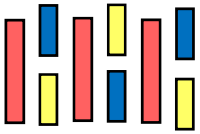
Fogging + proximity + placement
[Chen et al., ICCAD-18]



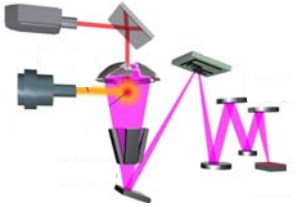
Placement, Routing, Post-Layout Optimization

Outline

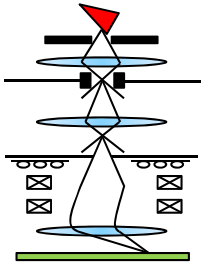
Lithography Basics



Multiple Patterning

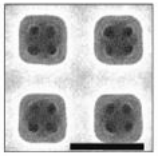


Extreme Ultraviolet (EUV)



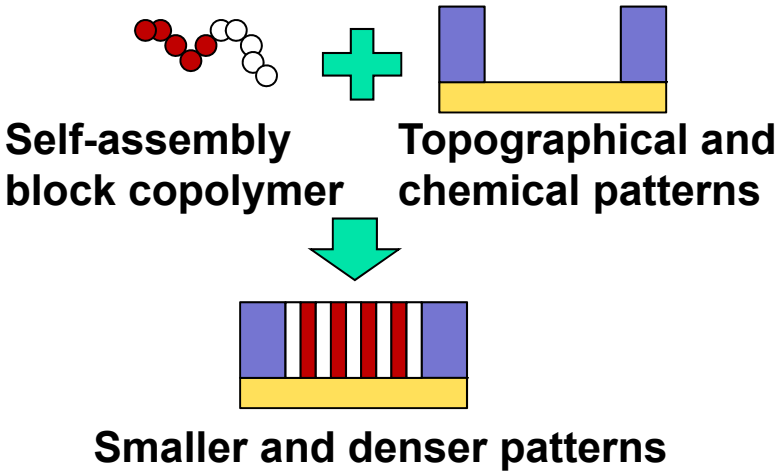
Electron Beam (e-beam)

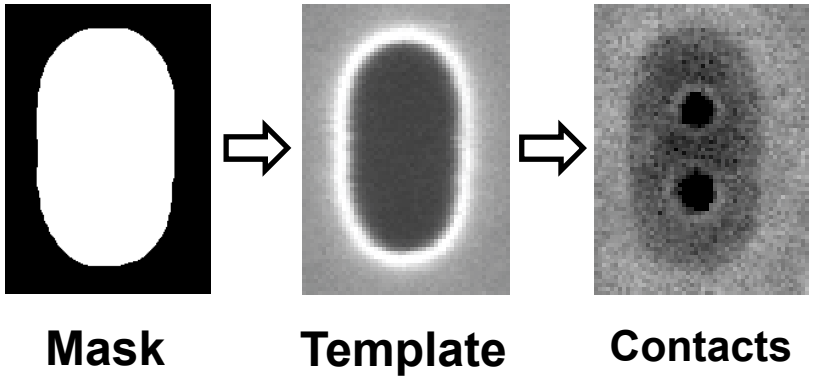
Directed Self-Assembly (DSA)



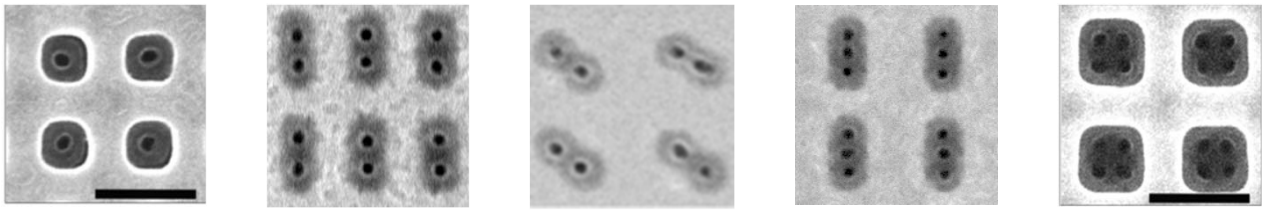
Directed Self-Assembly (DSA)

- Block copolymer DSA for contact/via patterning
  - Groups of contacts/vias are patterned by guiding templates with traditional 193i lithography





Contact patterning with DSA

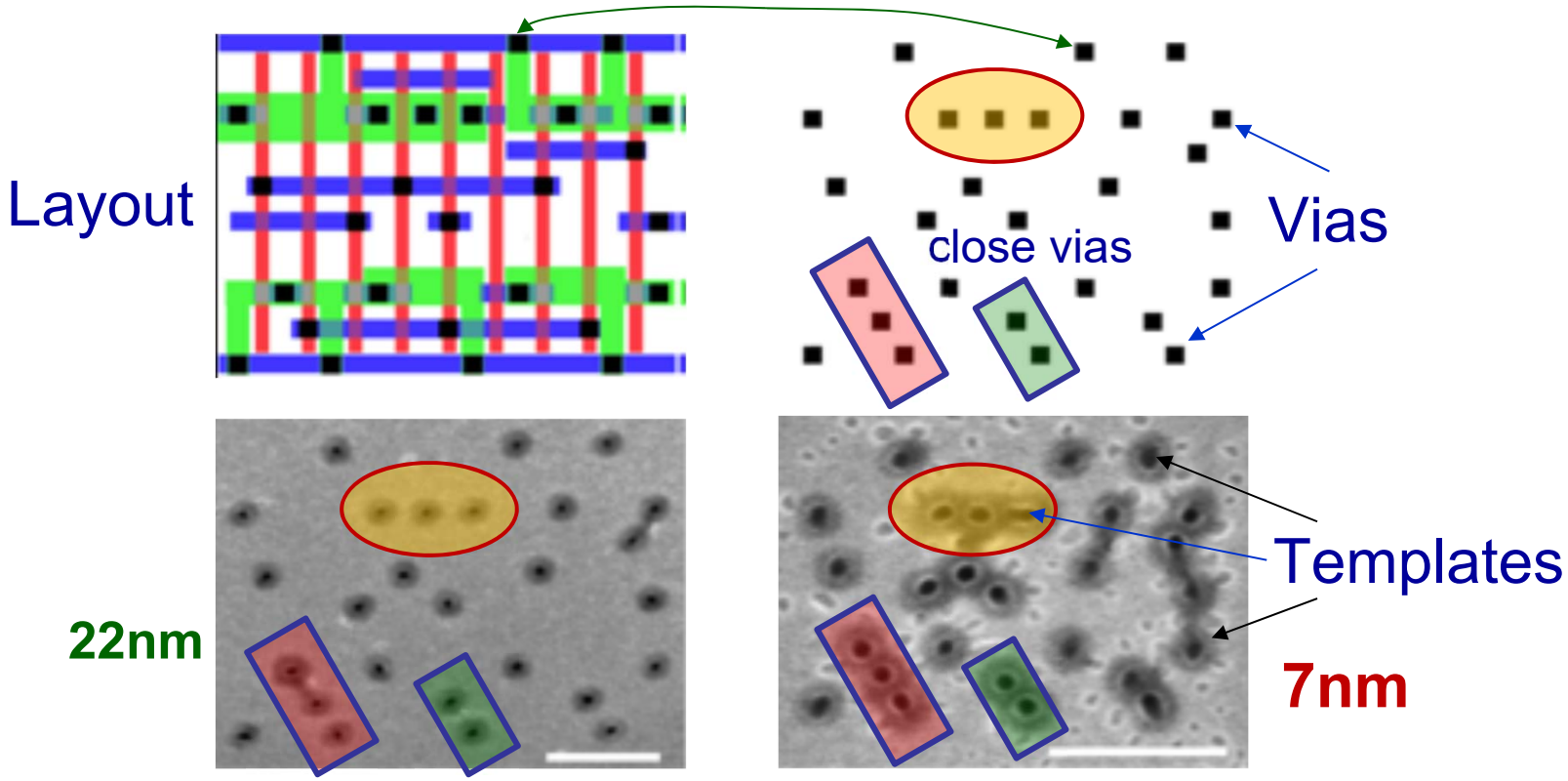


Contact patterns formed by various DSA templates

[Xiao, et al., ASP-DAC'15]

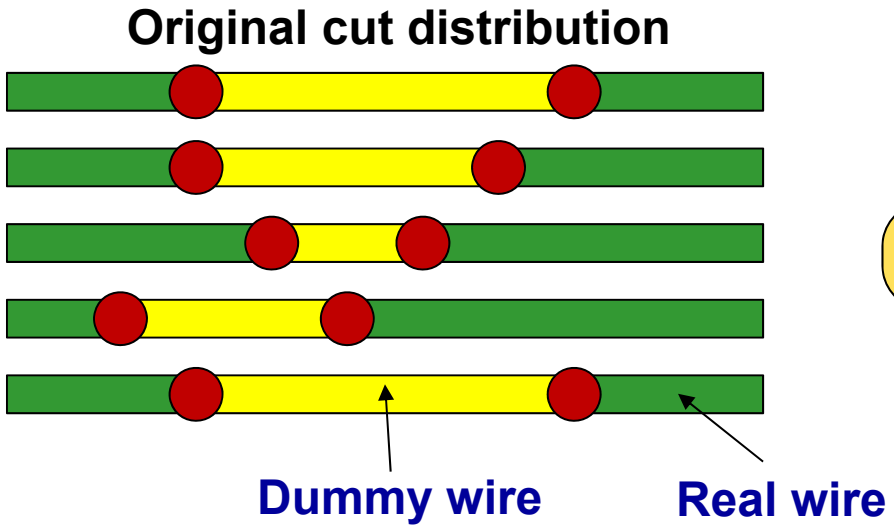
Cut/Via Pattering with DSA

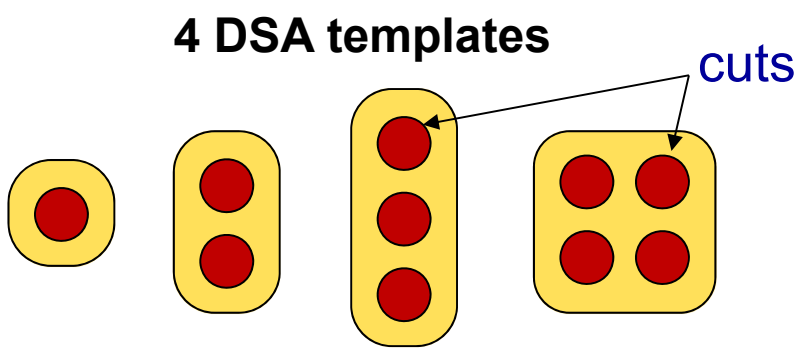
- A large template can be used to pattern multiple close contacts even for sub-7nm nodes

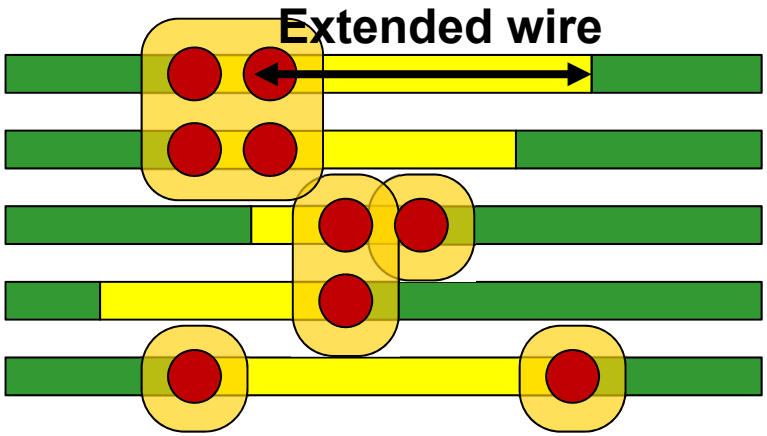


[Xiao, et al., DAC'14]

1D DSA Cut Redistribution

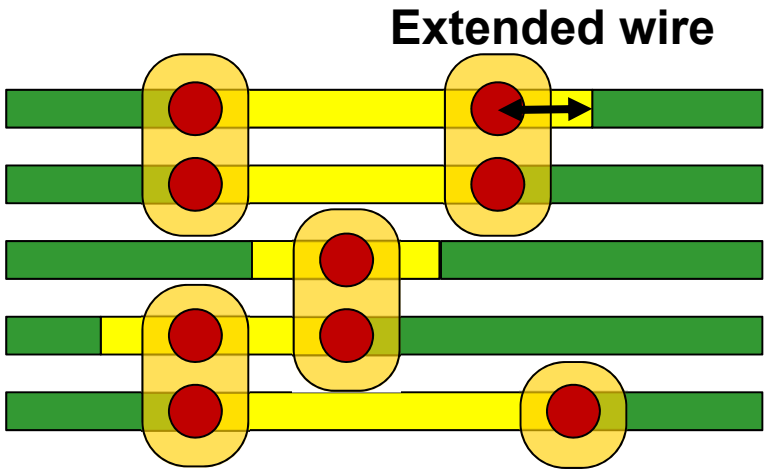






Cut distribution A

conflicts: 2, wire cost: 11

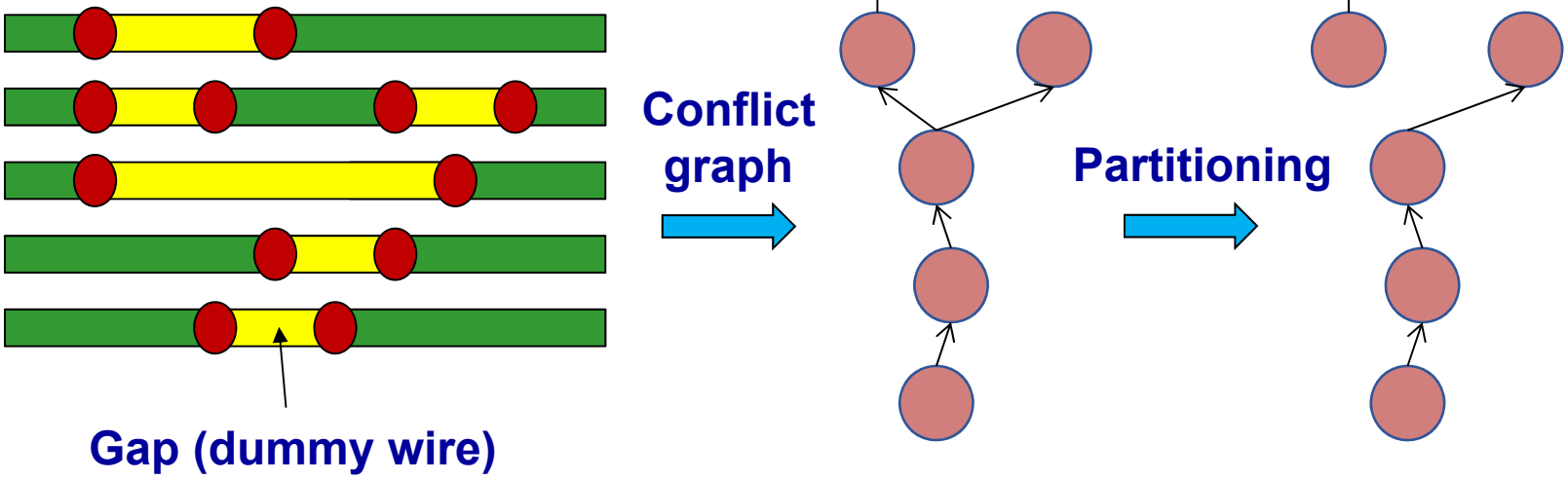


Cut distribution B

Conflicts: 0, wire cost: 4

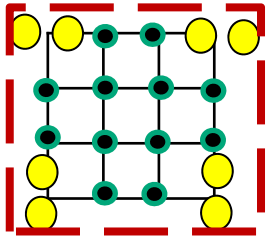
Cut Redistribution for 1D Layouts

- Lin & Chang, ASP-DAC-16
- Construct a conflict graph G = (V,E)
  - A node denotes a gap
  - An edge is between two gaps if their cuts conflict each other
- Partition G into vertex-disjoint paths considering conflicts
- Each vertex-disjoint subproblem can be solved in linear time (dynamic programming)

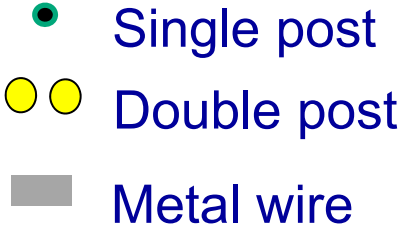


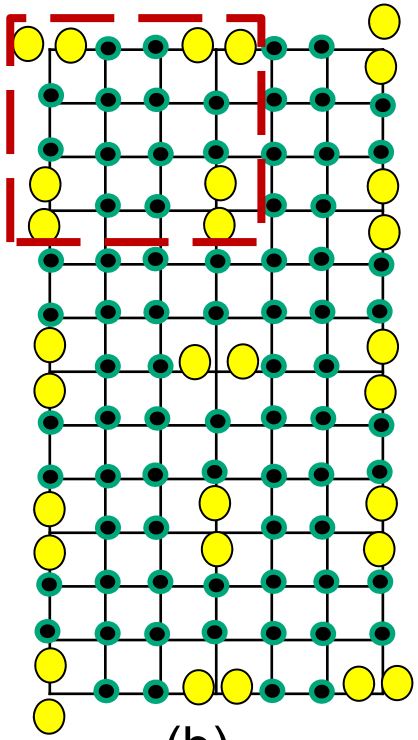
2-D Directed Self-Assembly

- A square lattice of topographic features (posts), guides block copolymers (BCPs) to form 2-D metal wires
- A layout is divided into an array of square tiles
  - Each tile contains 4 X 4 grids with 4 double posts at the corner grids and single posts at the other grids

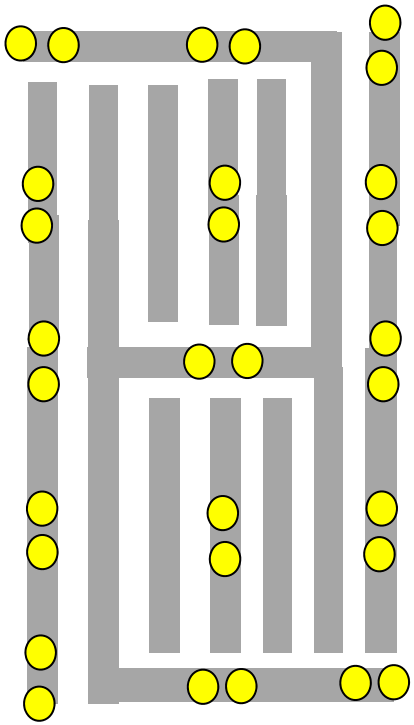


(a)





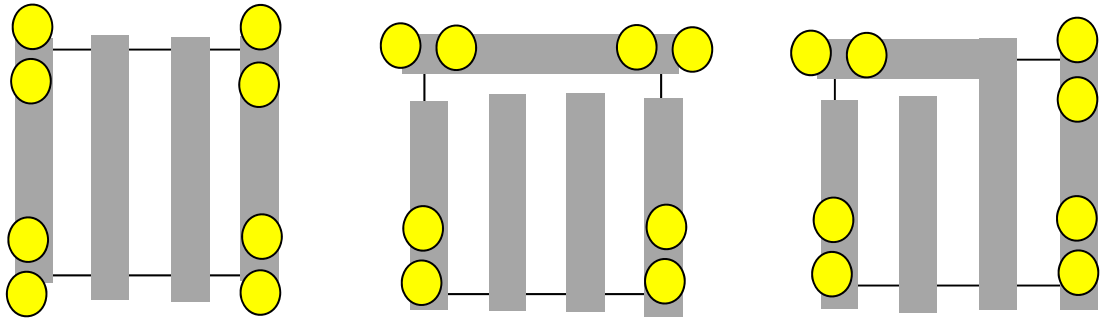
(b)



(c)

2D-DSA Wire Patterns

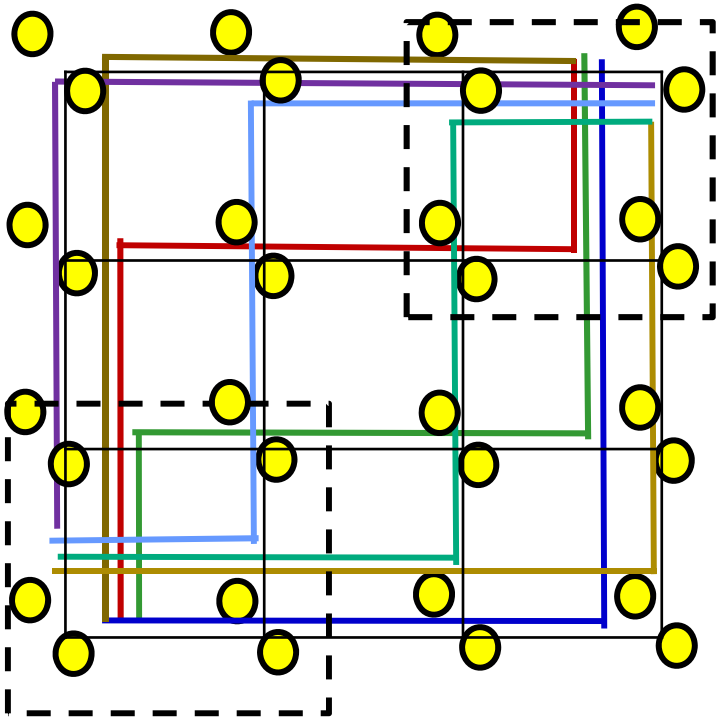
- Double post orientations determine wire patterns
- Only some combinations of double orientations can have high yields [Chang, et al., NatureCommunication-14]

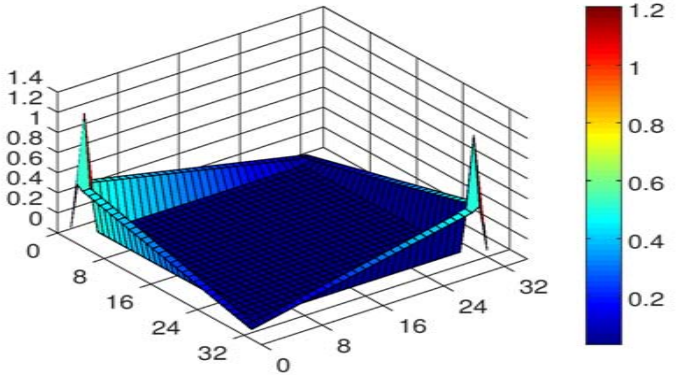


- Physical design for 2D-DSA
  - Placement for 2D-DSA [Lin & Chang, DAC-17]
  - Routing algorithm [Su & Chang, ICCAD-16; Yu & Chang, DAC-18]
  - 2D DSA cut redistribution [Wang & Chang, ICCAD-19]

2D-DSA Detailed Placement [DAC'17]

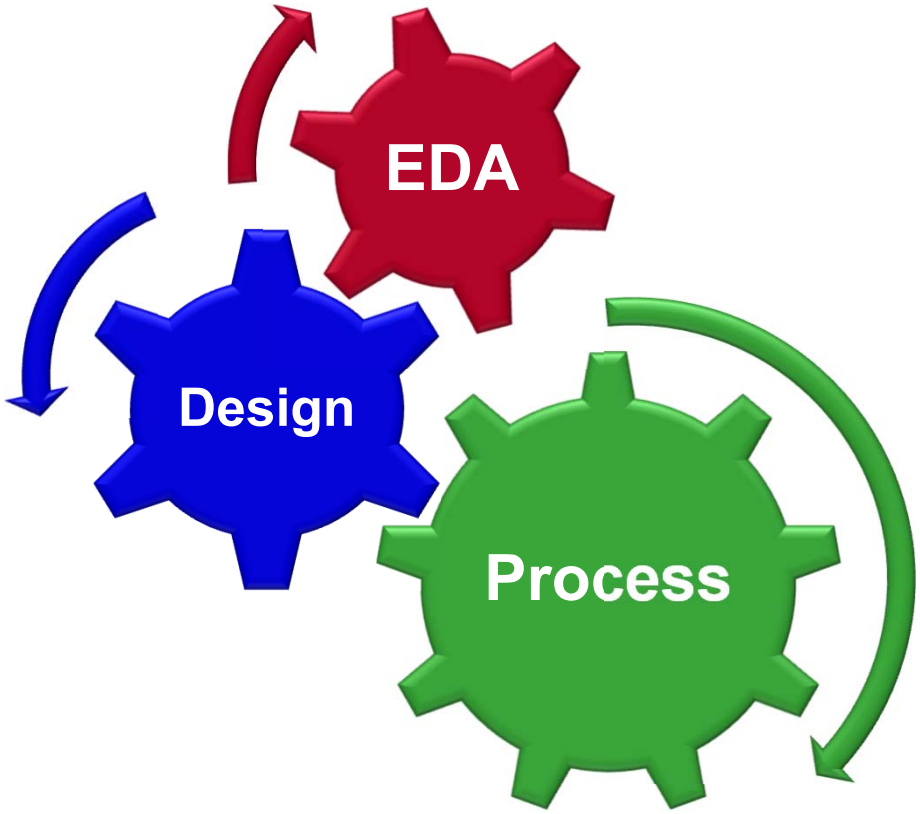
- There exist only L-shaped or Z-shaped patterns in non-detour routing for 2D-DSA
- A probability-based routability metric to efficiently estimate the impact of pin locations on post orientations
- A 2D-DSA-routability-driven detailed placement framework based on the proposed routability metric





Collaboration Model Revisited

Tremendous opportunities for process/design co-optimization



Keys to Research Solutions: CAR

Criticality

Abstraction

Restriction

Thank You!!