Linear Time Algorithm to Find All Relocation Positions for EUV Defect Mitigation

Yuelin Du[1], Hongbo Zhang[2], Qiang Ma[2]
and Martin D. F. Wong[1]
[1]University of Illinois at Urbana-Champaign
[2]Synopsys Inc.

Outline

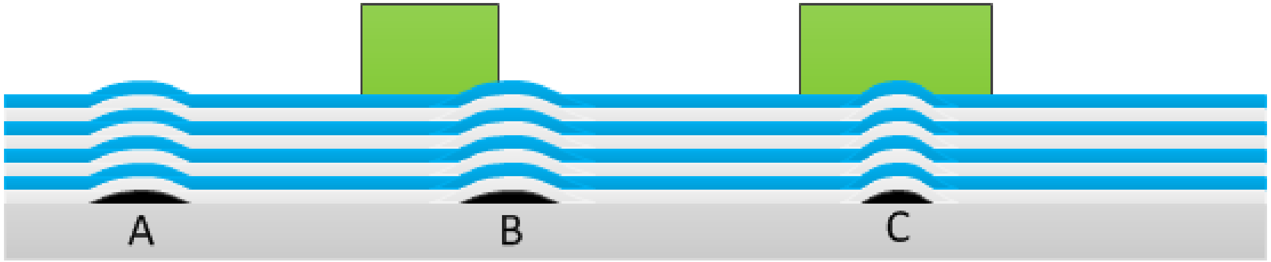
- Background
  - EUV defect mitigation by feature coverage
  - Covering defects by layout relocation
- Problem Formulation
  - Feasible regions to locate a single die
  - Problem definition
- Problem Solution
  - Blank region partitioning
  - Find all feasible regions in each blank region
- Experimental Results

Outline

- Background
  - EUV defect mitigation by feature coverage
  - Covering defects by layout relocation
- Problem Formulation
  - Feasible regions to locate a single die
  - Problem definition
- Problem Solution
  - Blank region partitioning
  - Find all feasible regions in each blank region
- Experimental Results

Defect Mitigation by Feature Coverage

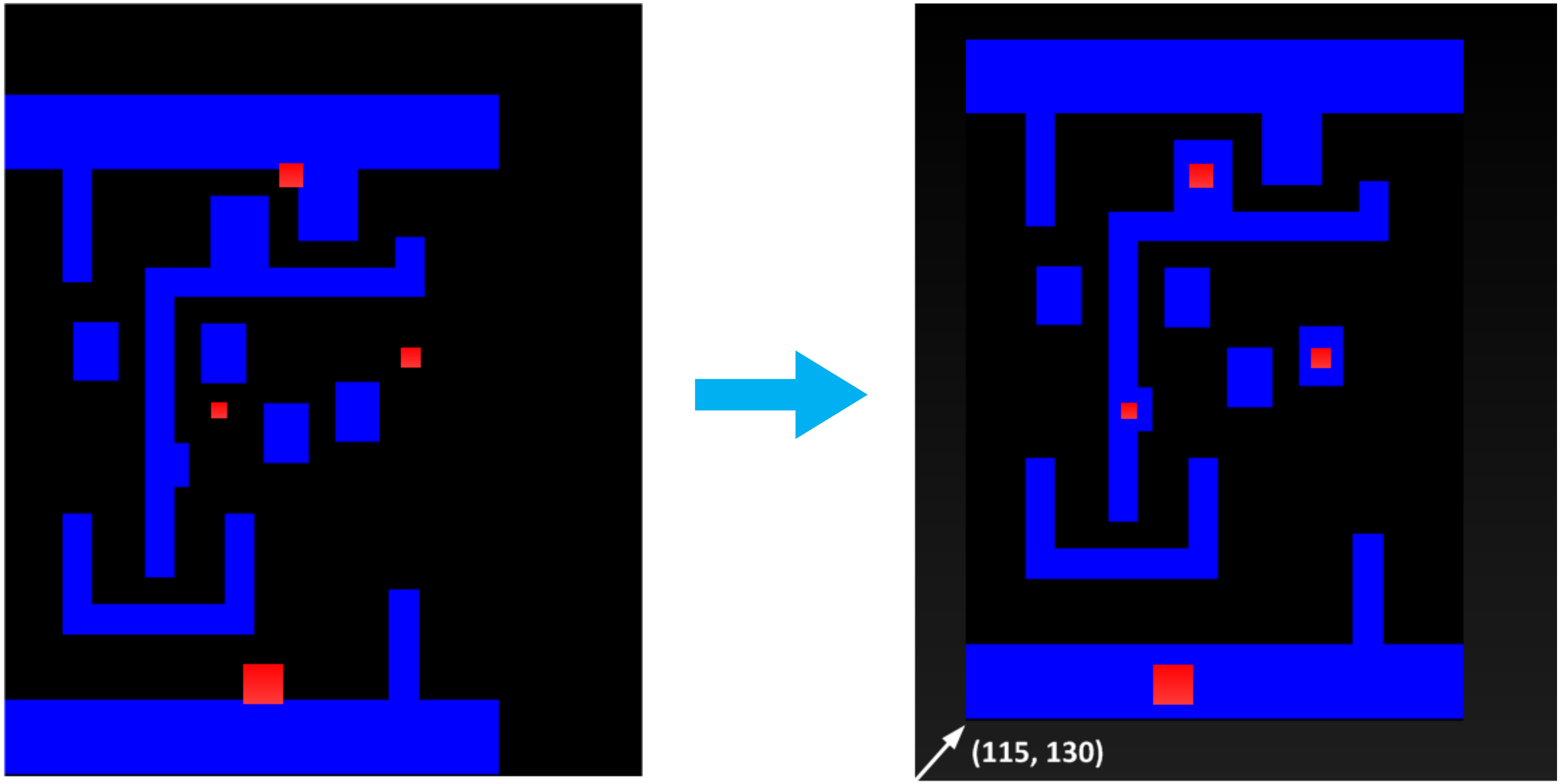
- Difficult to achieve **0** defect due to the complex Multi-layer (ML) structure.
- Covering the defects with features is an effective way to mitigate defect impact.



Defect A is not covered; defect B is partially covered; defect C is completely covered. Therefore, only defect C is successfully mitigated.

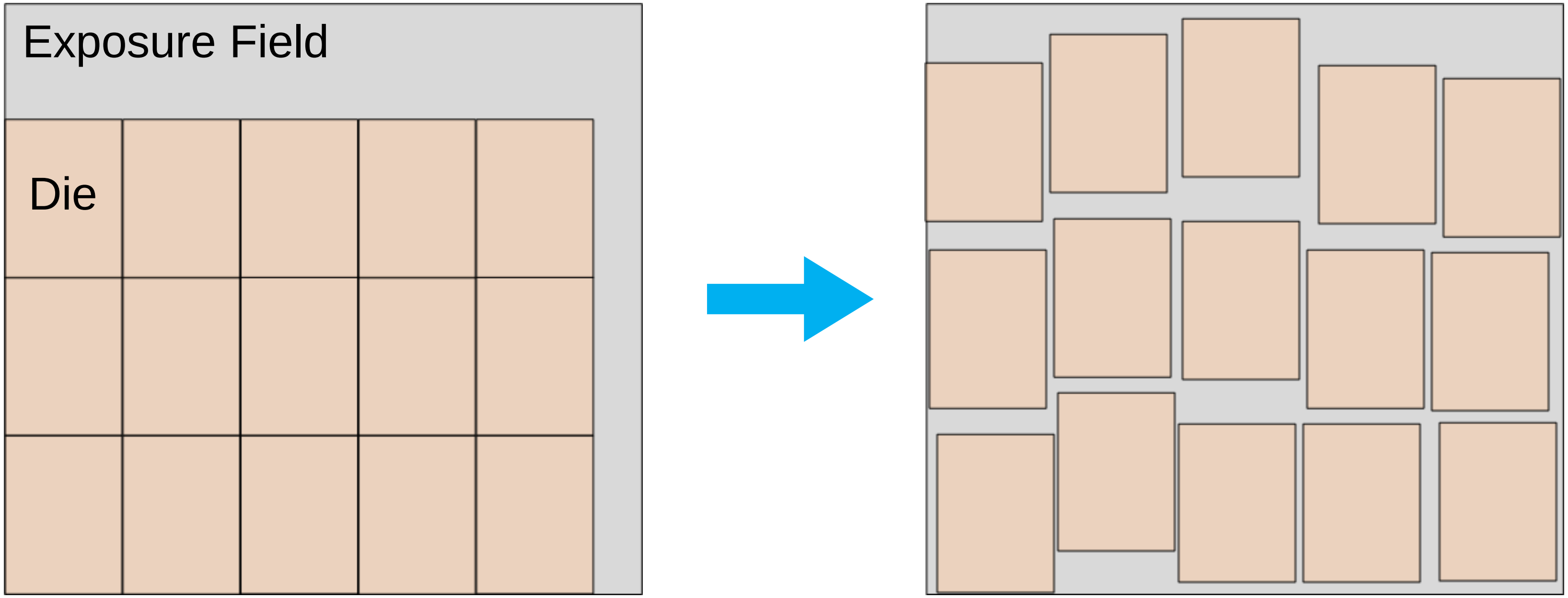
Covering Defects by Layout Relocation

- All defects are successfully covered by shifting the layout to position (115, 130).



Full Field Die Relocation

- Each die can be relocated independently within the exposure field.
- All feasible locations to locate a single die must be obtained first.

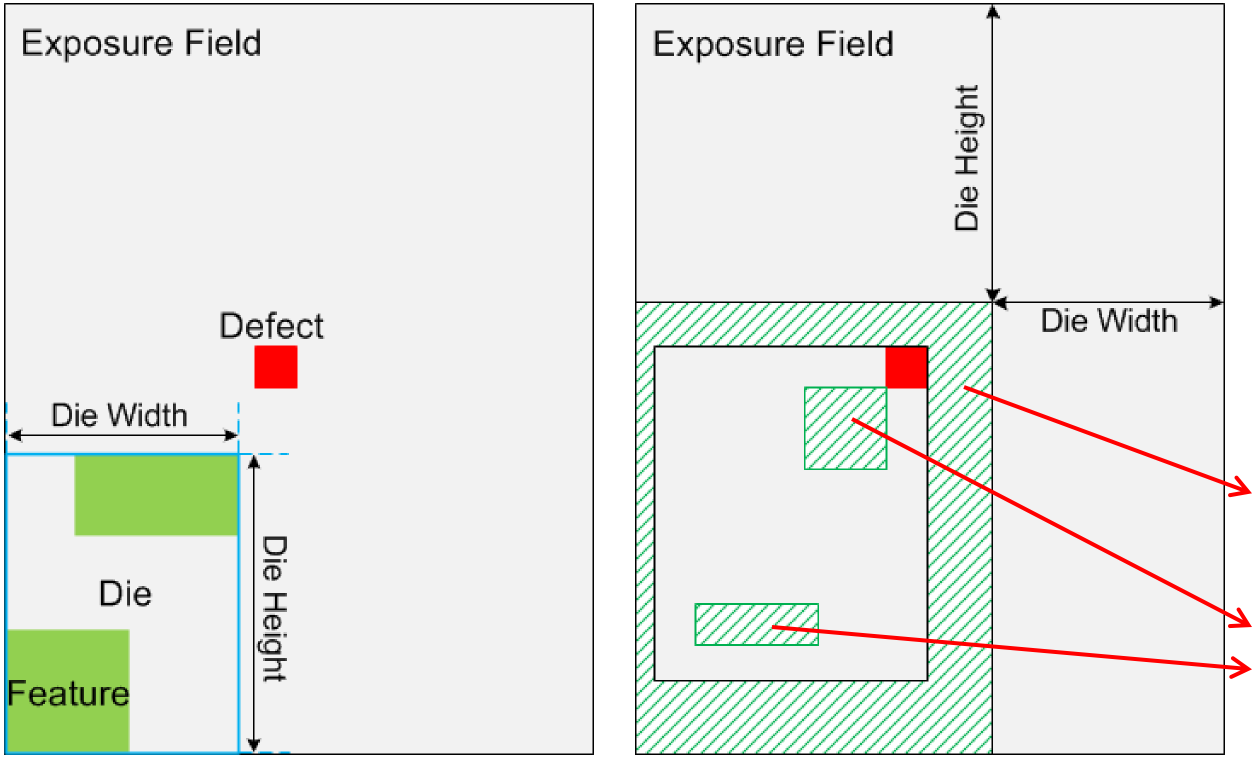


Outline

- Background
  - EUV defect mitigation by feature coverage
  - Covering defects by layout relocation
- Problem Formulation
  - Feasible regions to locate a single die
  - Problem definition
- Problem Solution
  - Blank region partitioning
  - Find all feasible regions in each blank region
- Experimental Results

Feasible Regions to Locate a Single Die

- Defect impact is mitigated when the defect is either located outside the die area or covered by a feature.



▨ The feasible regions to place the bottom left corner of the die

Defect outside the die area

Defect covered by a feature

Problem Definition

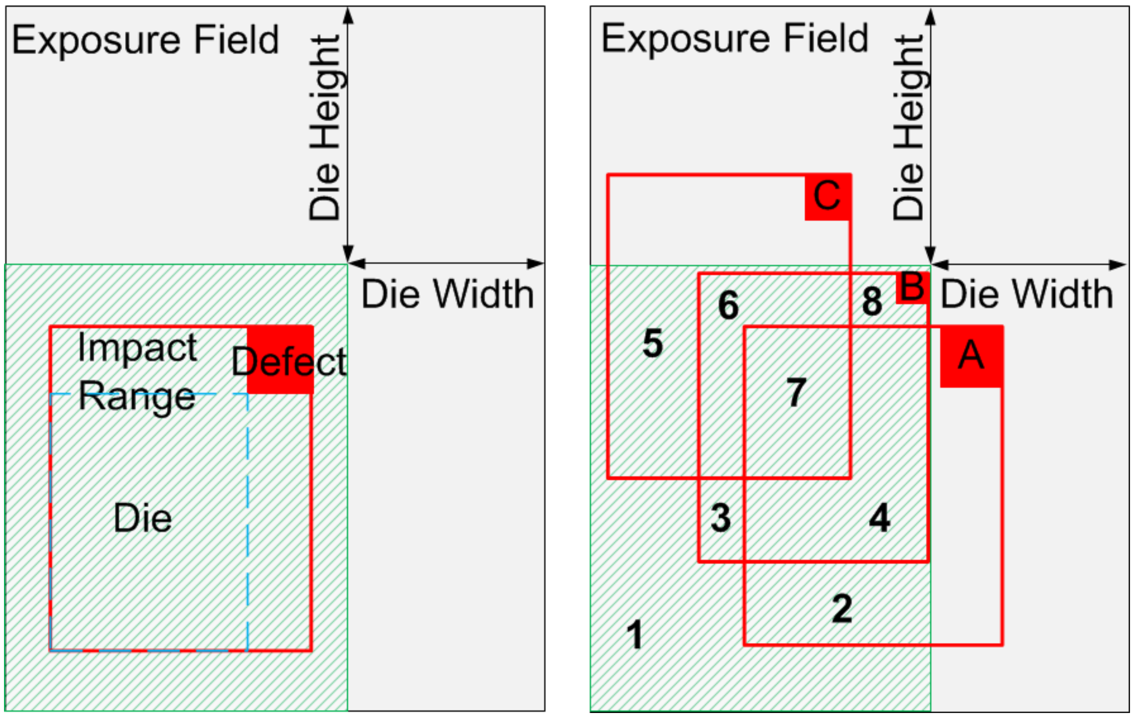
- Given a full die layout and an exposure field with a certain number of rectangular defects, the objective is to find all feasible regions, such that as long as the bottom left corner of the die lies within any feasible region, all the defects are either outside the die area or completely covered by the features in the die, and the whole die is located within the exposure field.

Outline

- Background
  - EUV defect mitigation by feature coverage
  - Covering defects by layout relocation
- Problem Formulation
  - Feasible regions to locate a single die
  - Problem definition
- Problem Solution
  - Blank region partitioning
  - Find all feasible regions in each blank region
- Experimental Results

Blank Region Partitioning

- The valid region is partitioned into 8 blank regions based on the impact range of each defect.



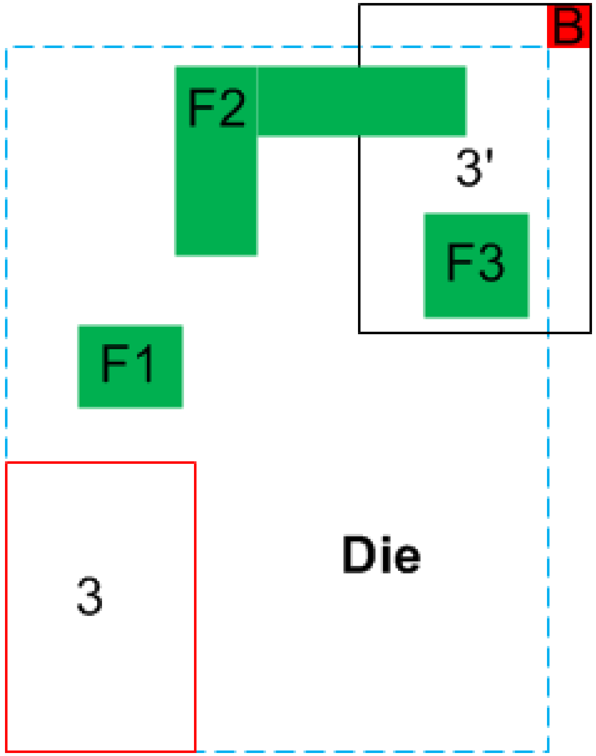
# of effective defects for each blank region:

- Region 1 : **0**
- Region 2, 3, 5 and 8 : **1**
- Region 4 and 6: **2**
- Region 7: **3**

Find Feasible Regions in Each Blank Region

- Blank Region with No Effective Defect
  - The whole blank region is a feasible region
- Blank Region with Single Effective Defect
  - Three steps to find all feasible regions
    - Impacted feature extraction
    - Impacted feature shrinking
    - Shrunk region rotation and shift
- Blank Region with Multiple Effective Defect
  - First find the feasible regions for each single defect
  - Then all sets of feasible regions intersect with each other to obtain the final feasible regions

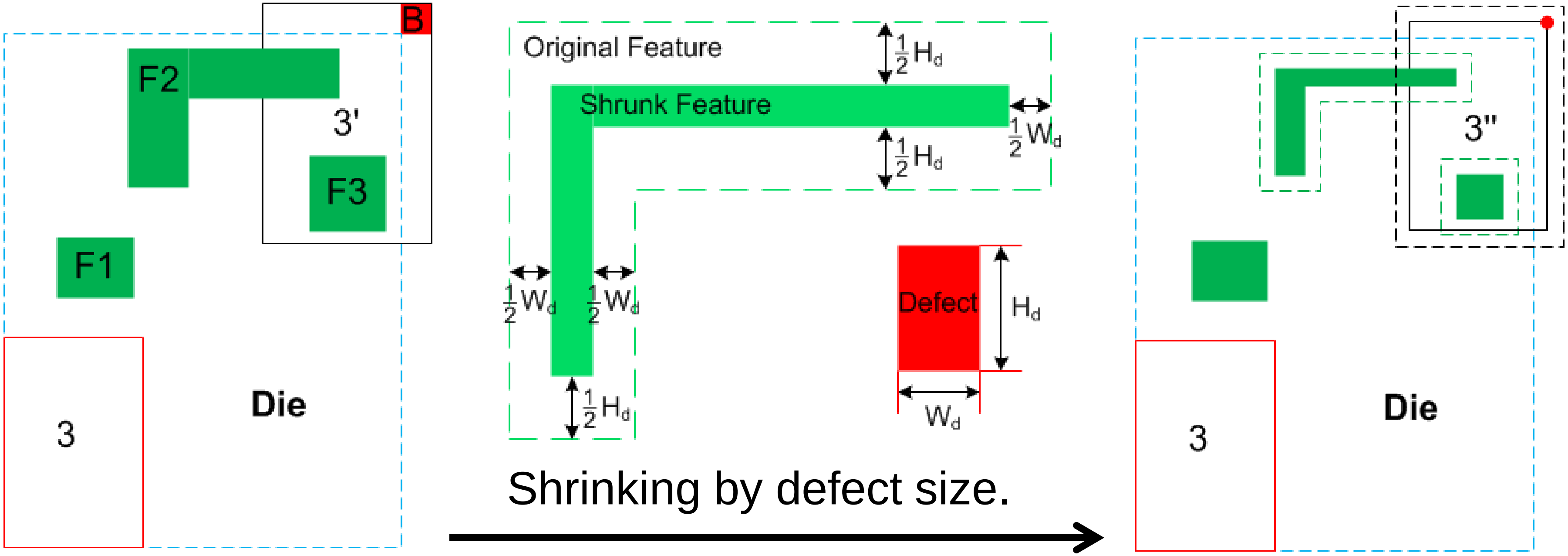
Blank Region with Single Effective Defect



- Take region *3* as an example.
- When the bottom left corner of the die moves in region *3*, region *3′* is impacted by defect *B* – defined as **impacted region**.
- $W_{3'} = W_3 + W_B$
- $H_{3'} = H_3 + H_B$
- Feature F2 and F3 in the die can potentially cover defect *B* – defined as **impacted features**.

Impacted Feature Shrinking

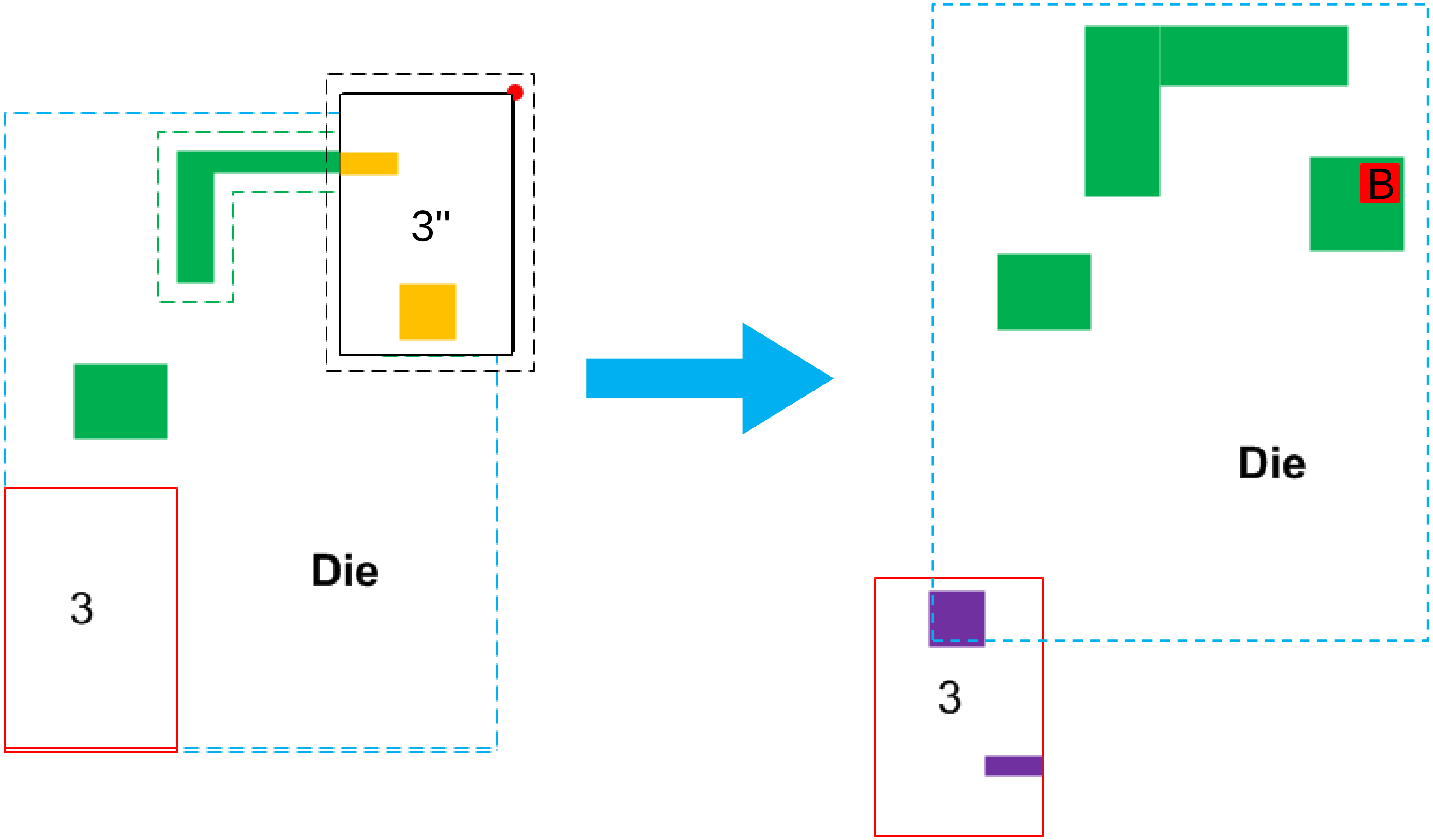
- Impacted features (F2 and F3) and impacted region (*3′*) are shrunk by defect size.
- Shrunk region *3″* is the same size as region *3.*



Shrinking by defect size.

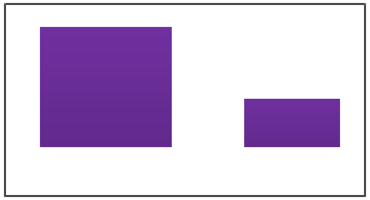
Shrunk Region Rotation and Shift

- Rotate and shift *3″* to obtain final feasible regions.

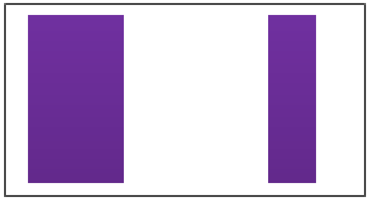


Blank Region with Multiple Effective Defect

- First find the feasible regions for each single defect separately.

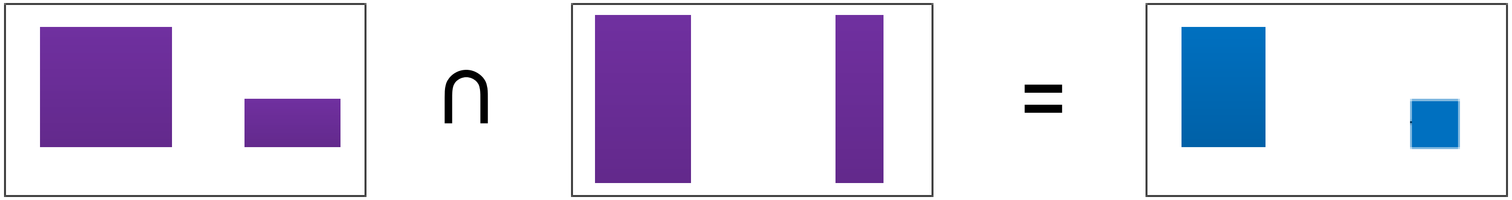


Feasible regions for defect **1**



Feasible regions for defect **2**

- Then all sets of feasible regions intersect with each other to obtain the final feasible regions.

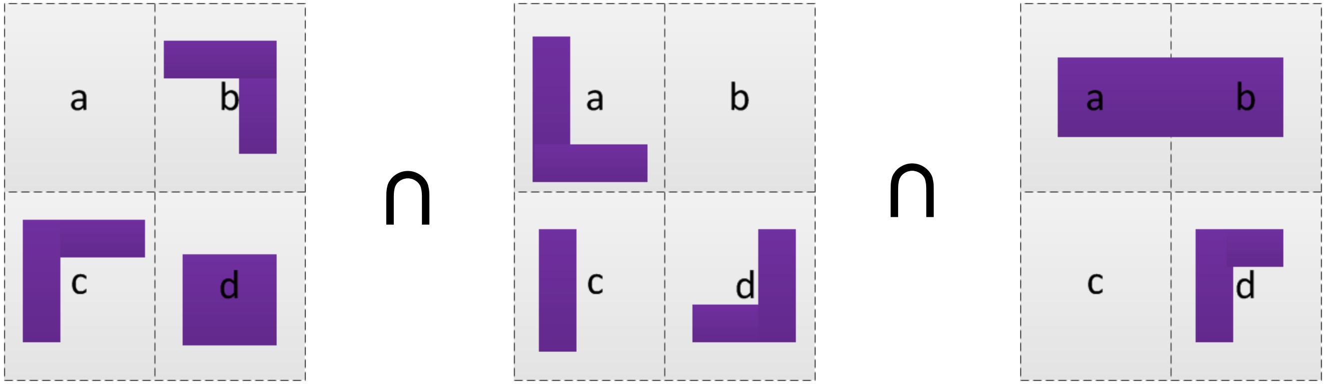


Improved Strategy for Region Intersection

- After shrinking, impacted regions by each defect are of the same size.
- Partition each region into equally sized small tiles, each with limited number of patterns.
- A tile is a **valid tile** if and only if it is not empty.
- Build a truth table to remember the validation of each tile.
- Only consider the tile that is valid for all defects during intersection.
- Update the truth table after an intersect operation.

Improved Strategy for Region Intersection

- Only tile *d* is valid for all defects.
- Only tile *d* is considered for intersection.



Feasible regions intersection

$$\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 1 \\ \hline \end{array} \;\&\; \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 1 \\ \hline \end{array} \;\&\; \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 1 \\ \hline \end{array}$$

Truth table to remember valid tiles.

Time Analysis

- # of defects: c_1
- # of blank regions: c_2
- # of features in the die: n
- # of tiles in each impacted region: $\approx n$
- # of patterns in each tile: c_3
- c_1, c_2 and c_3 are very small compared to n, and hence they can be consider as constant values.

Time Analysis

- Impacted feature extraction: $O(c_1 * c_2 * n)$
- Impacted feature shrinking: $O(c_1 * n)$
- Shrunk region rotation and shift: $O(c_1 * c_2 * n)$
- Distribute features into tiles: $O(c_1 * c_2 * n)$
- Intersect two small tiles: c_i (constant time)
- Feasible regions intersection: $O(c_1 * c_2 * n * c_i)$

- The overall time complexity is the summation of the above terms: $O(n)$

Outline

- Background
  - EUV defect mitigation by feature coverage
  - Covering defects by layout relocation
- Problem Formulation
  - Feasible regions to locate a single die
  - Problem definition
- Problem Solution
  - Blank region partitioning
  - Find all feasible regions in each blank region
- Experimental Results

Experimental Results

Die Size (cm x cm)	Defect #	Runtime Comparison	
		Ours (s)	Grid Based (s)
1 x 1	1	871.81	> 1 week
1.5 x 1.5	1	1980.91	> 1 week
1 x 1	2	1635.54	> 1 week
1.5 x 1.5	2	3747.33	> 1 week
1 x 1	4	3375.12	> 1 week
1.5 x 1.5	4	9161.14	> 1 week
1 x 1	6	5634.87	> 1 week
1.5 x 1.5	6	15058.00	> 1 week
1 x 1	8	9086.35	> 1 week
1.5 x 1.5	8	26037.1	> 1 week

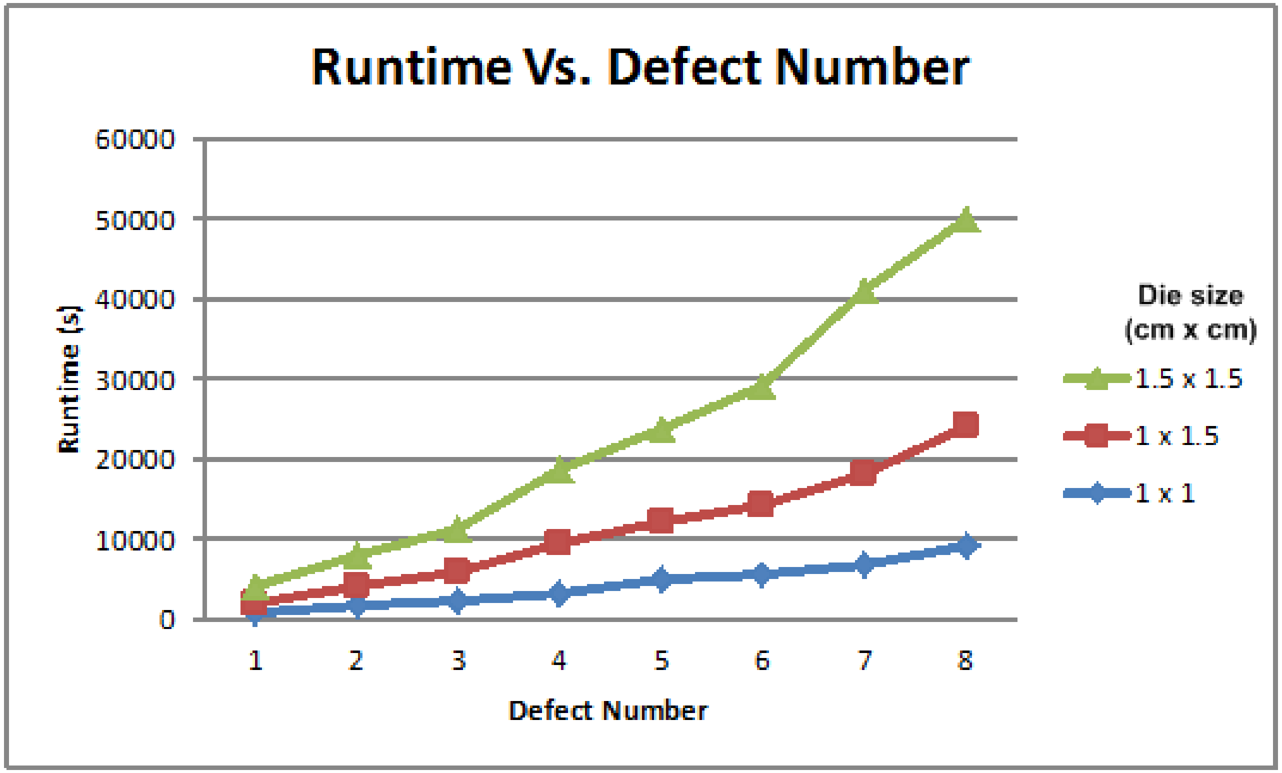
- CPU: *2.4GHz*
- RAM: *36GB*

- Technology: *11 nm*
- Design: *Logic*
- Layer: *M1*

- Defect Size: *50 ~ 200 nm*
- Exposure Size: *10.4 x 13.2 cm^2*

Experimental Results

- # of blank regions increases nonlinearly with defect #.
- Nonlinear runtime in defect #, especially for large dies.



Experimental Results

- Assuming constant defect #, Runtime is linear with respect to # of features in the die.