ASP-DAC : 8A-1

A New Graph-Theoretic, Multi-Objective Layout Decomposition Framework for Double Patterning Lithography

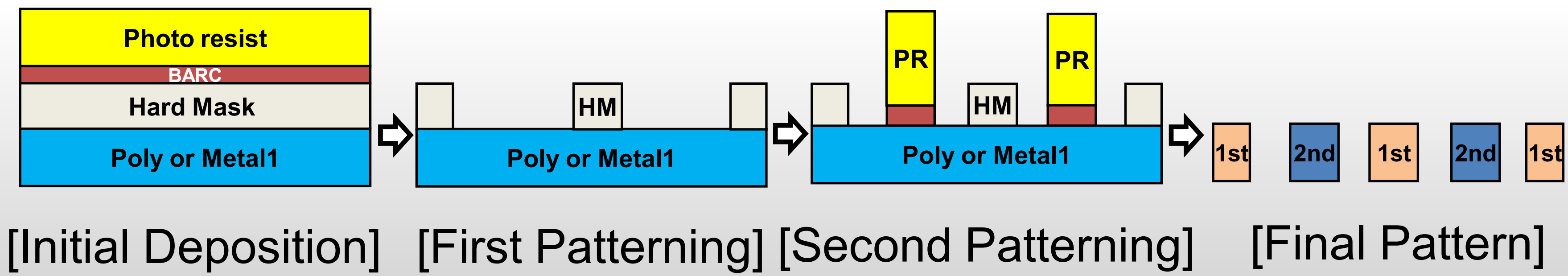
Jae-Seok Yang, Katrina Lu[1], Minsik Cho[2], Kun Yuan, and David Z. Pan
Dept. of ECE, The University of Texas at Austin, Austin, TX USA
[1]Intel Corporation, Hillsboro, OR USA
[2]IBM T. J. Watson Research Center, Yorktown Heights, NY USA

Outline

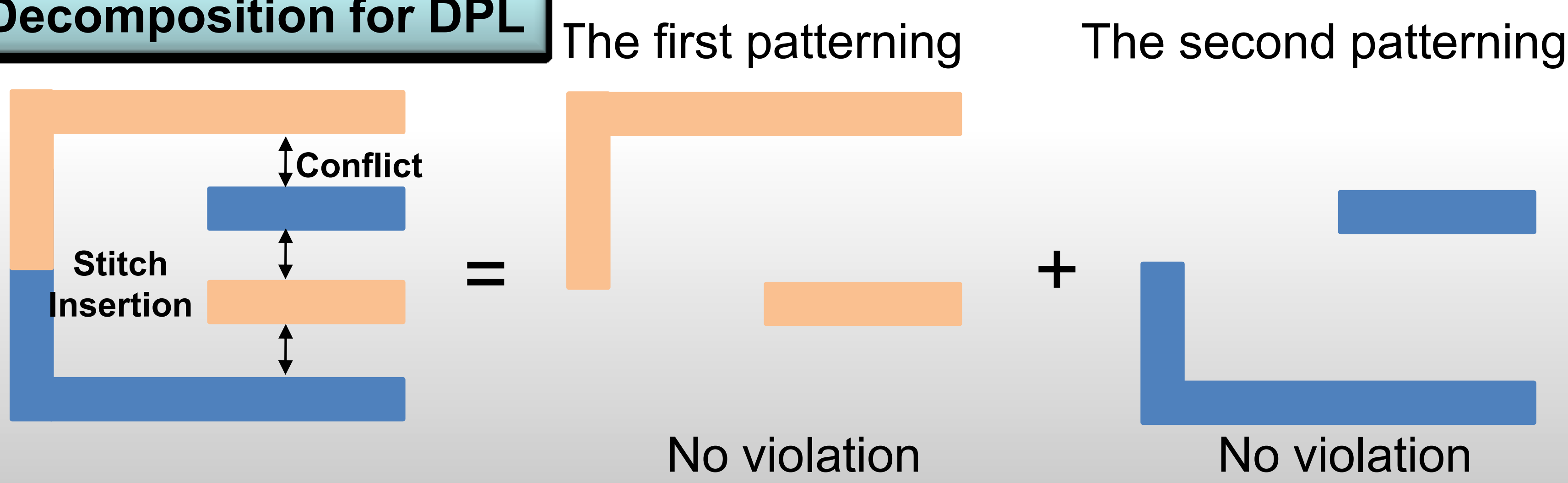
- Motivation
- Bi-Partitioning Based Decomposition
- Timing Driven Decomposition
- Experimental Results
- Future Works

Layout Decomposition

LELE DPL Process

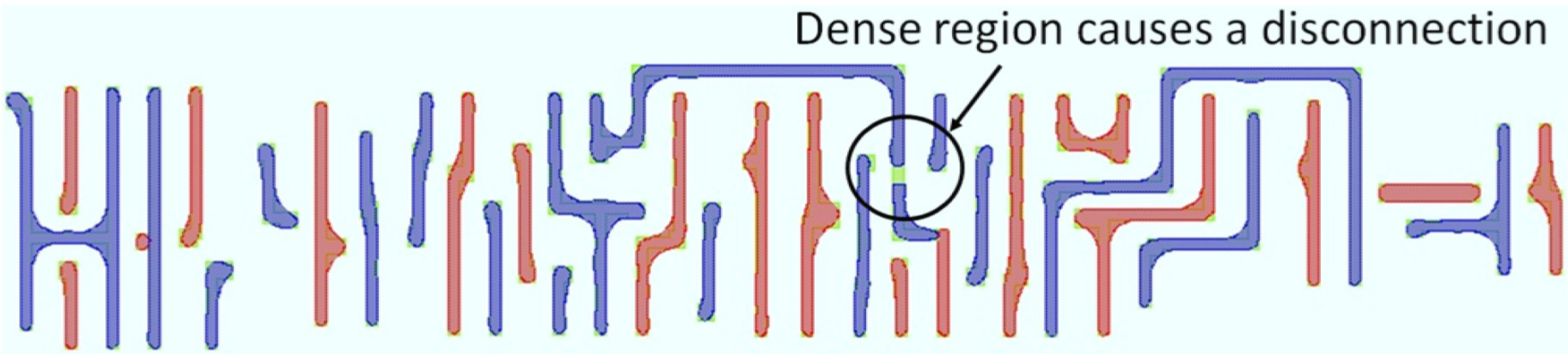


Decomposition for DPL

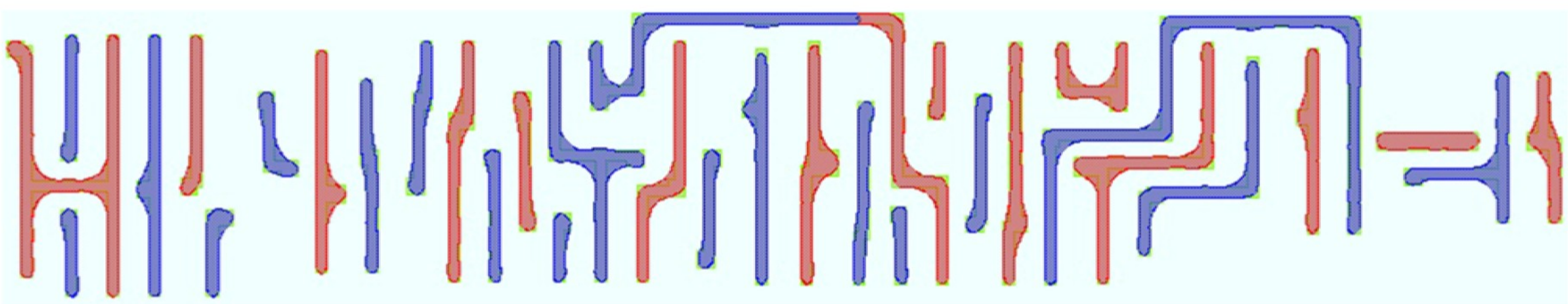


Balanced Density

Unbalanced decomposition: 38%(Red) and 62% (Blue)



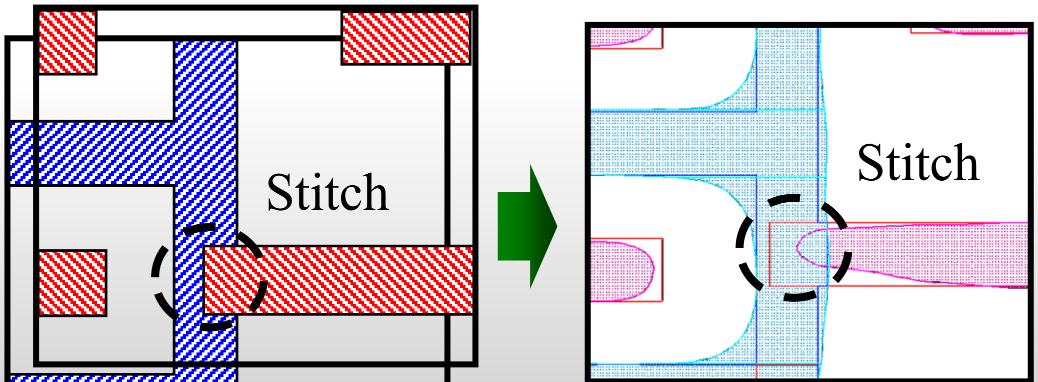
Balanced decomposition: 48%(Red) an 52% (Blue)



Balanced density is preferred during layout decomposition

More Decomposition Requirements

Minimum Stitch Insertion

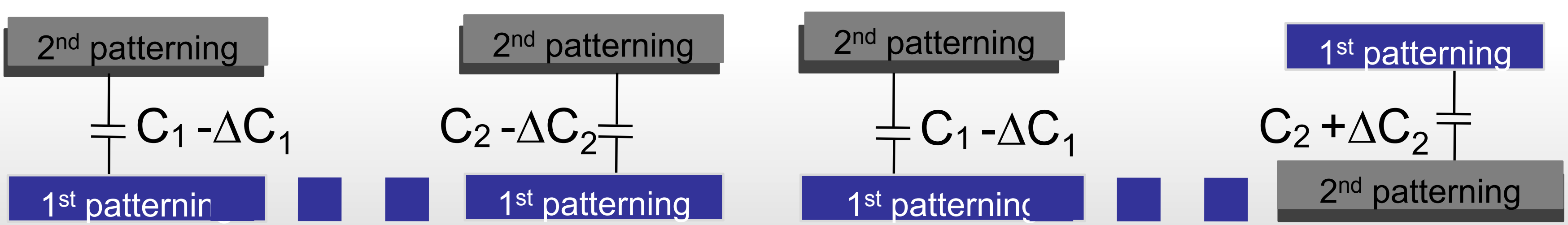


1) Yield loss with overlay

2) Area increase due to overlap margin

[Lucas SPIE'08]

Overlay Compensation



Without Overlay Compensation **With Overlay Compensation**

Comparisons & Complexity

Previous Approach

	Balanced Density	Overlay Compensation	Stitch Minimization	Complexity
ICCAD08[1]	No	No	Yes (ILP)	NP-Complete
Our Framework	Yes	Yes	Yes (Bi-Partitioning)	Polynomial Time O(NlogN)

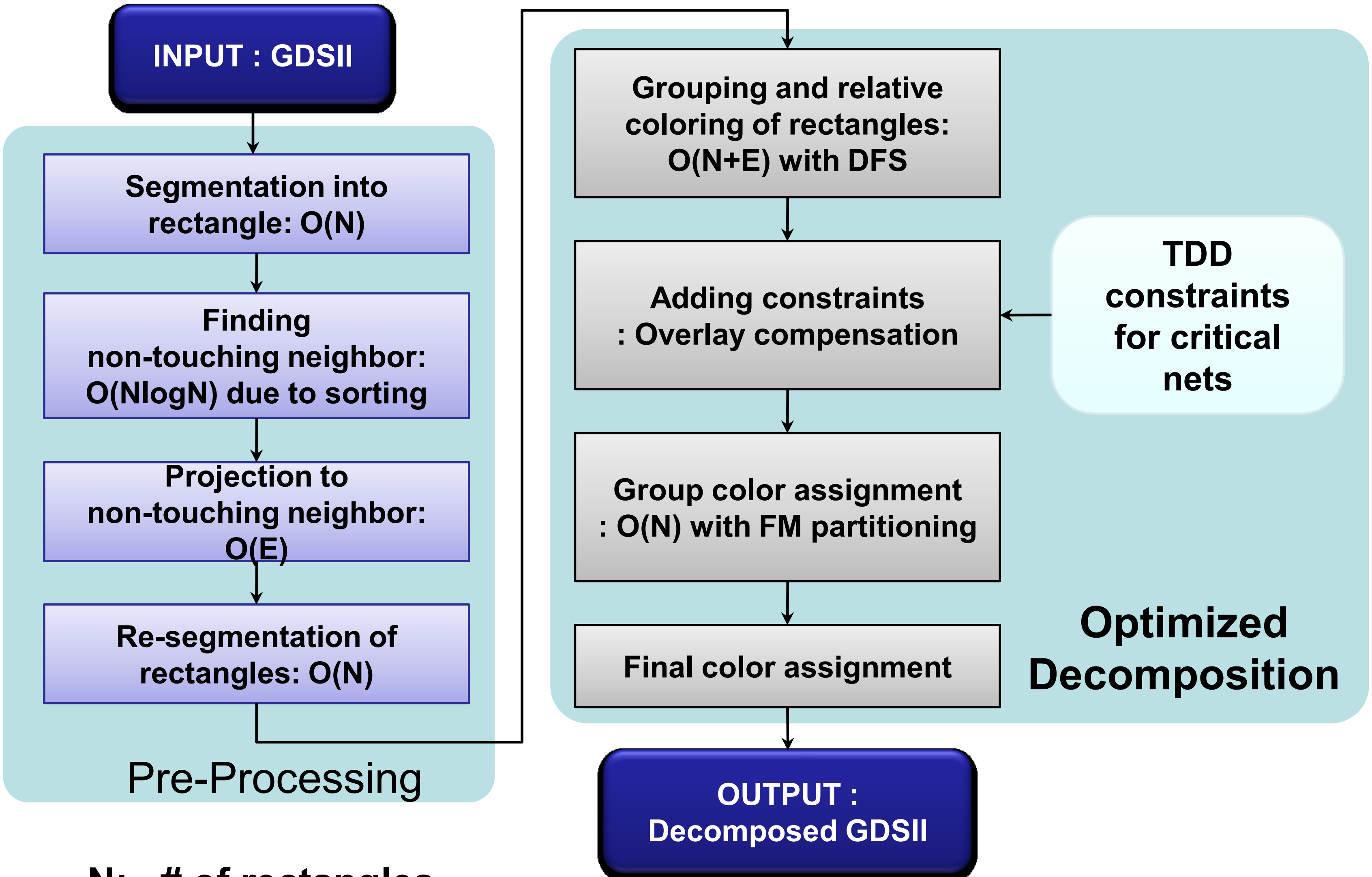
Complexity of Our Decomposition Algorithm

N # of rectangles, and E # of neighboring pairs.

1) Segmentation from polygon to rectangles ➔ O(N)
2) Finding neighbors (sorting according to coordinate) ➔ O(NlogN)
3) The complexity of projection to non-touching neighbor ➔ O(E)
4) Grouping and relative coloring using DFS ➔ O(N+E)
5) Group color assignment with min-cut partitioning ➔ O(N)

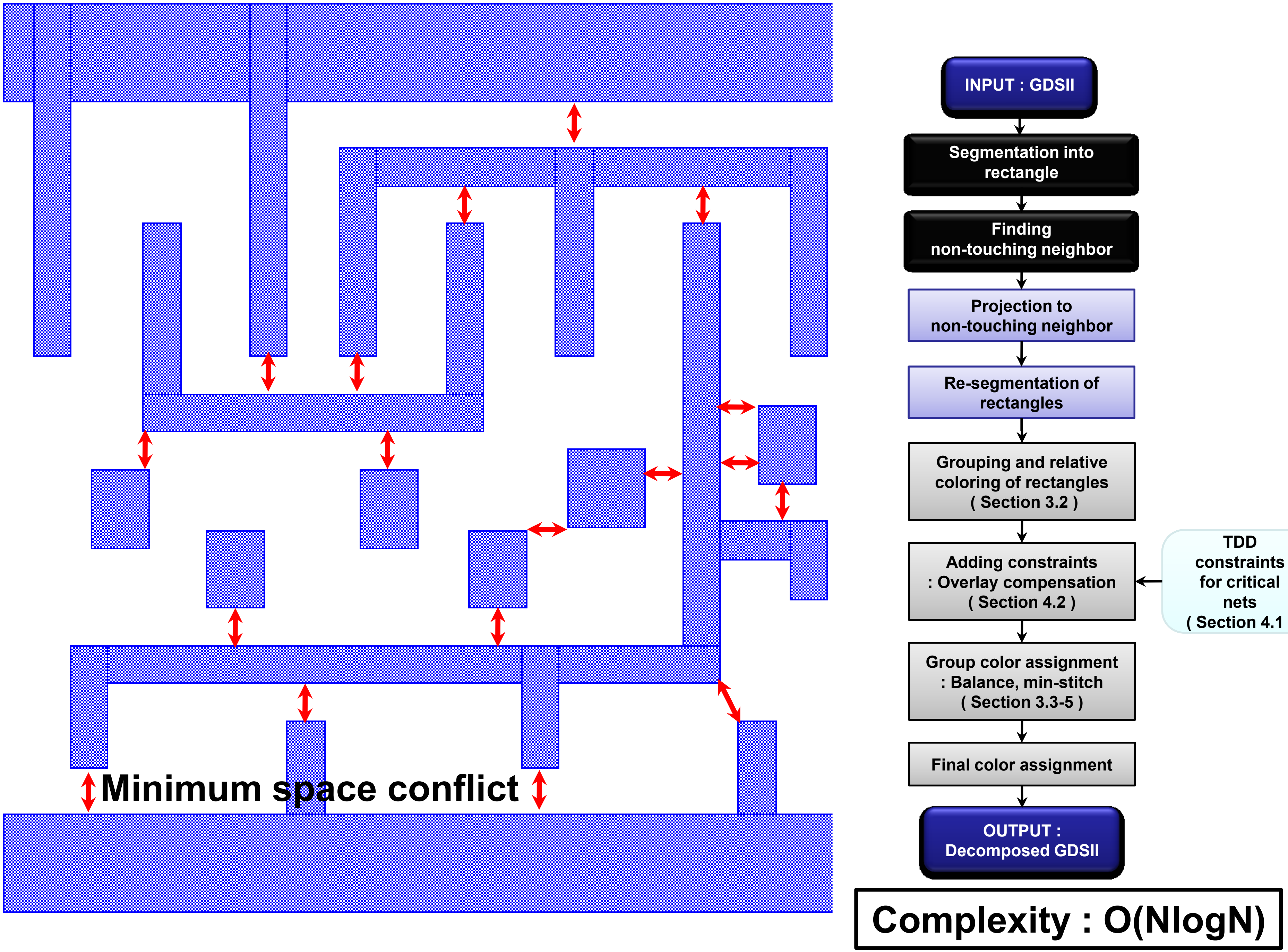
Overall complexity ➔ O(NlogN)

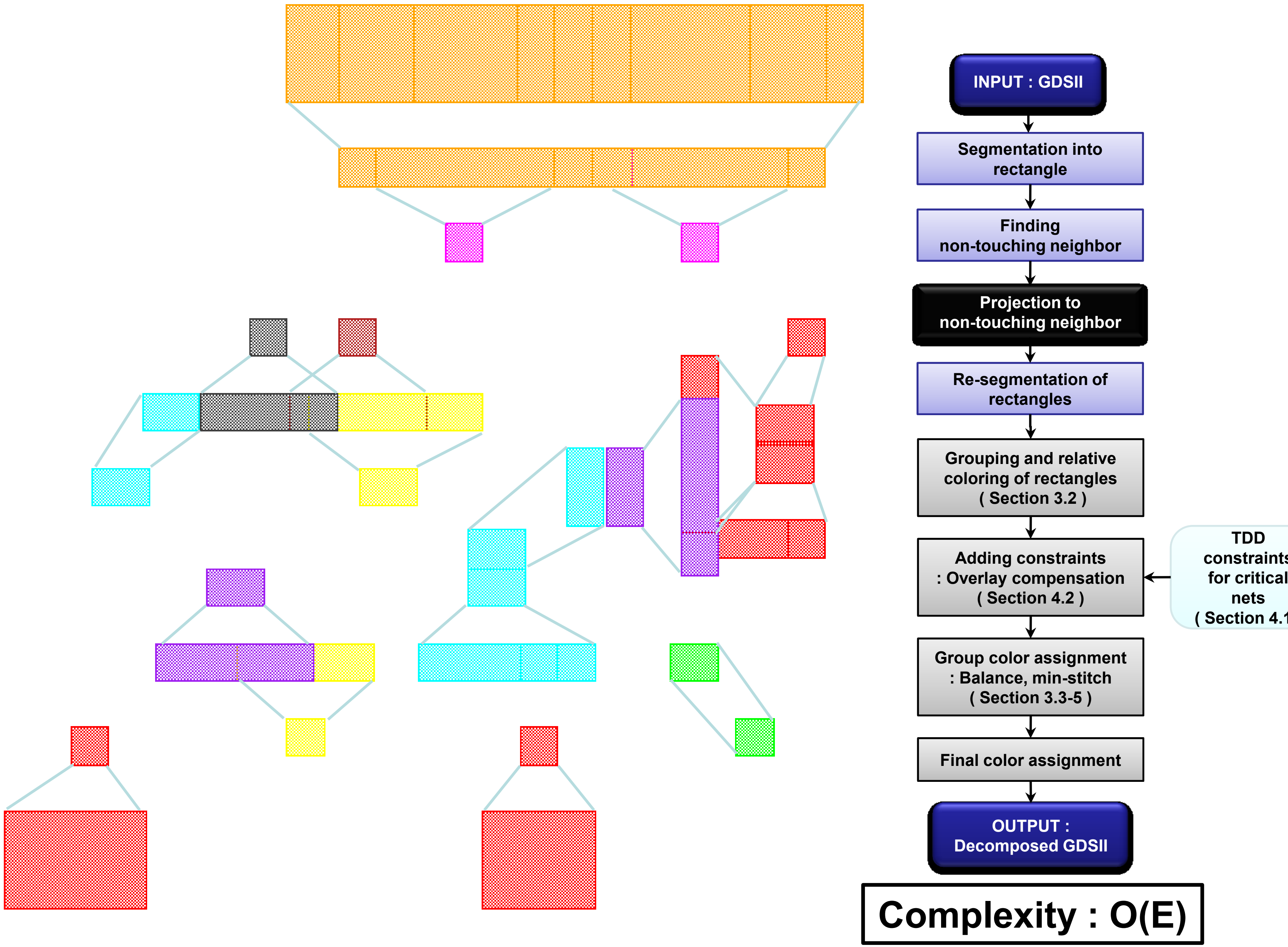
Overall decomposition flow



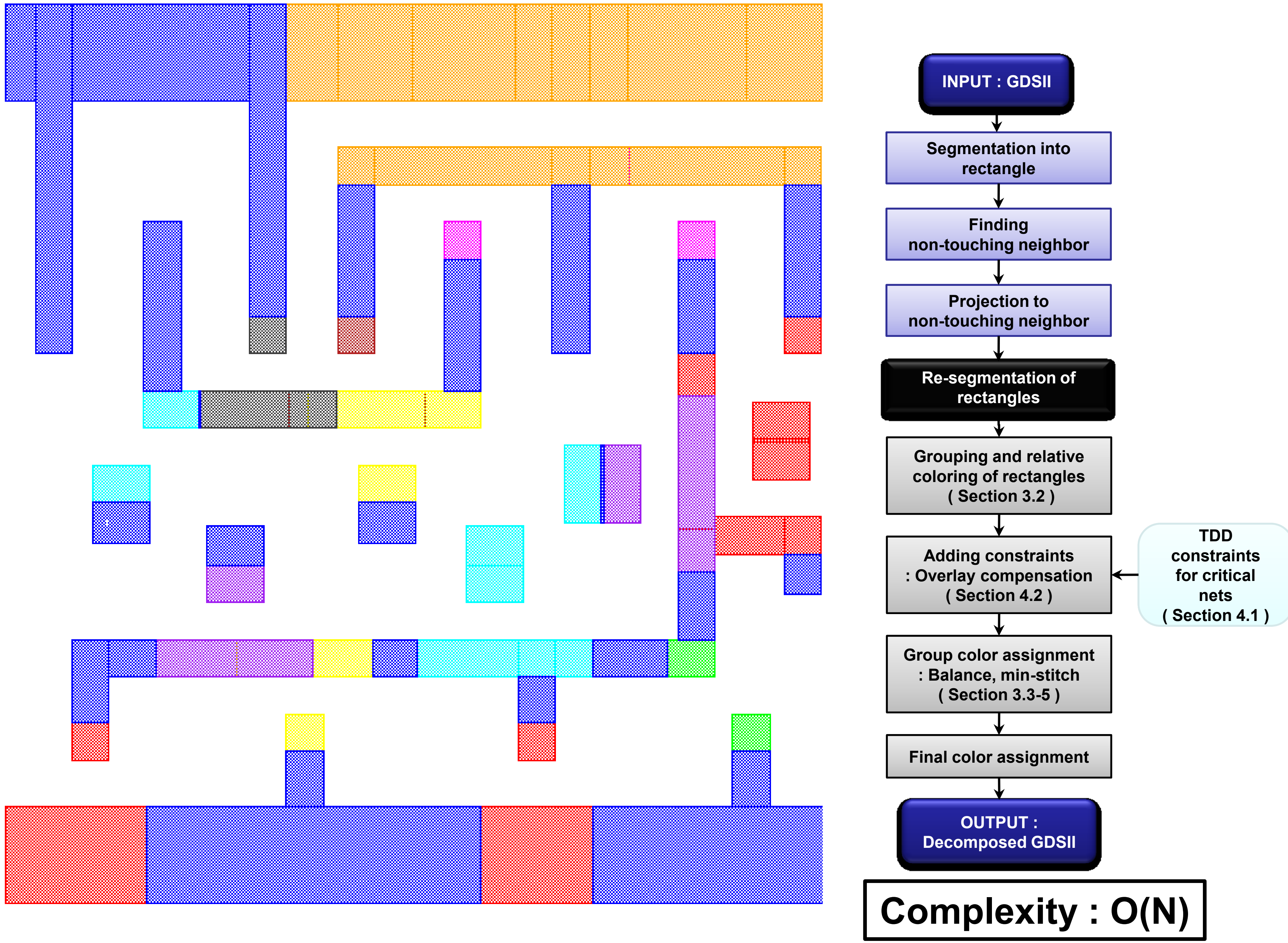
N: # of rectangles,
E: # of neighboring pairs

Overall complexity is O(NlogN)

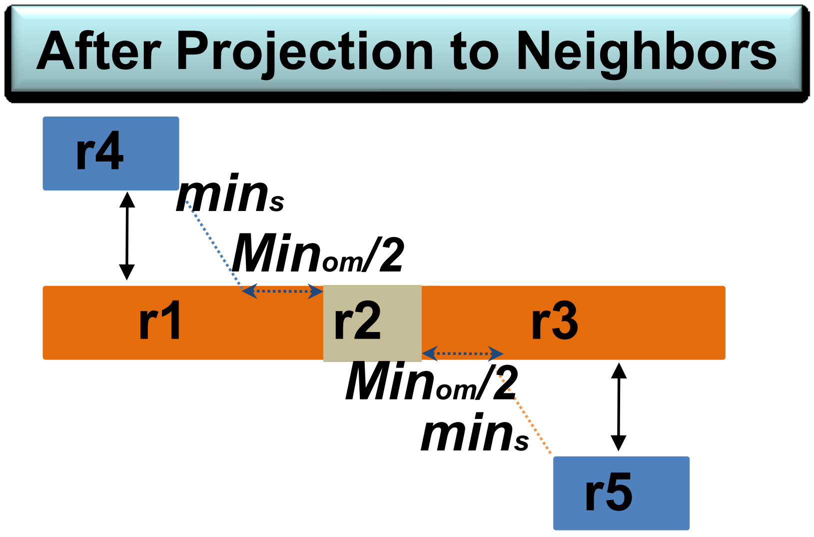

Minimum space conflict
INPUT : GDSII
Segmentation into rectangle
Finding non-touching neighbor
Projection to non-touching neighbor
Re-segmentation of rectangles
Grouping and relative coloring of rectangles (Section 3.2)
Adding constraints : Overlay compensation (Section 4.2)
TDD constraints for critical nets (Section 4.1)
Group color assignment : Balance, min-stitch (Section 3.3-5)
Final color assignment
OUTPUT : Decomposed GDSII
Complexity : O(NlogN)

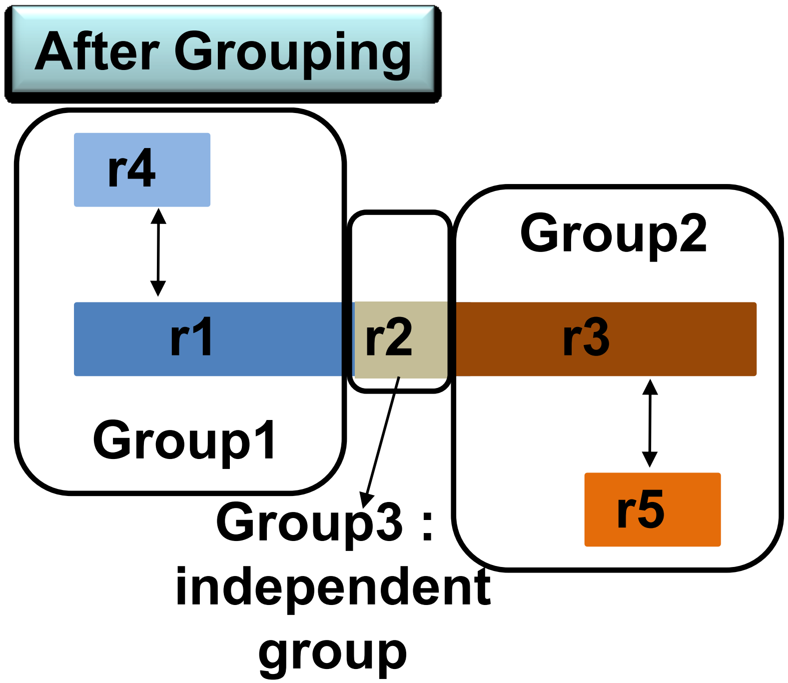


INPUT : GDSII
Segmentation into rectangle
Finding non-touching neighbor
Projection to non-touching neighbor
Re-segmentation of rectangles
Grouping and relative coloring of rectangles (Section 3.2)
Adding constraints : Overlay compensation (Section 4.2)
TDD constraints for critical nets (Section 4.1
Group color assignment : Balance, min-stitch (Section 3.3-5)
Final color assignment
OUTPUT : Decomposed GDSII
Complexity : O(E)


INPUT : GDSII
Segmentation into rectangle
Finding non-touching neighbor
Projection to non-touching neighbor
Re-segmentation of rectangles
Grouping and relative coloring of rectangles (Section 3.2)
Adding constraints : Overlay compensation (Section 4.2)
TDD constraints for critical nets (Section 4.1)
Group color assignment : Balance, min-stitch (Section 3.3-5)
Final color assignment
OUTPUT : Decomposed GDSII
Complexity : O(N)

Grouping and Relative Coloring

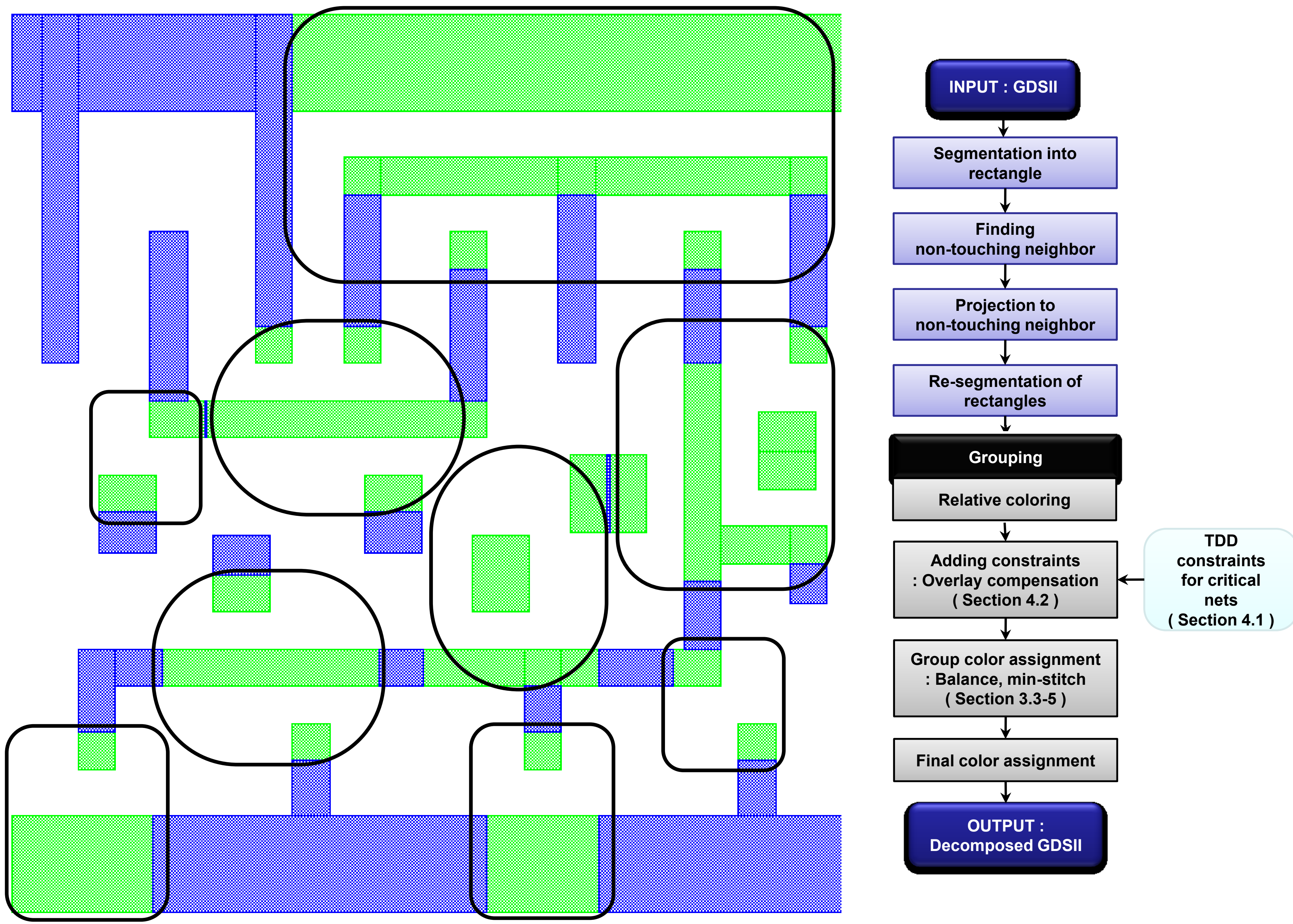


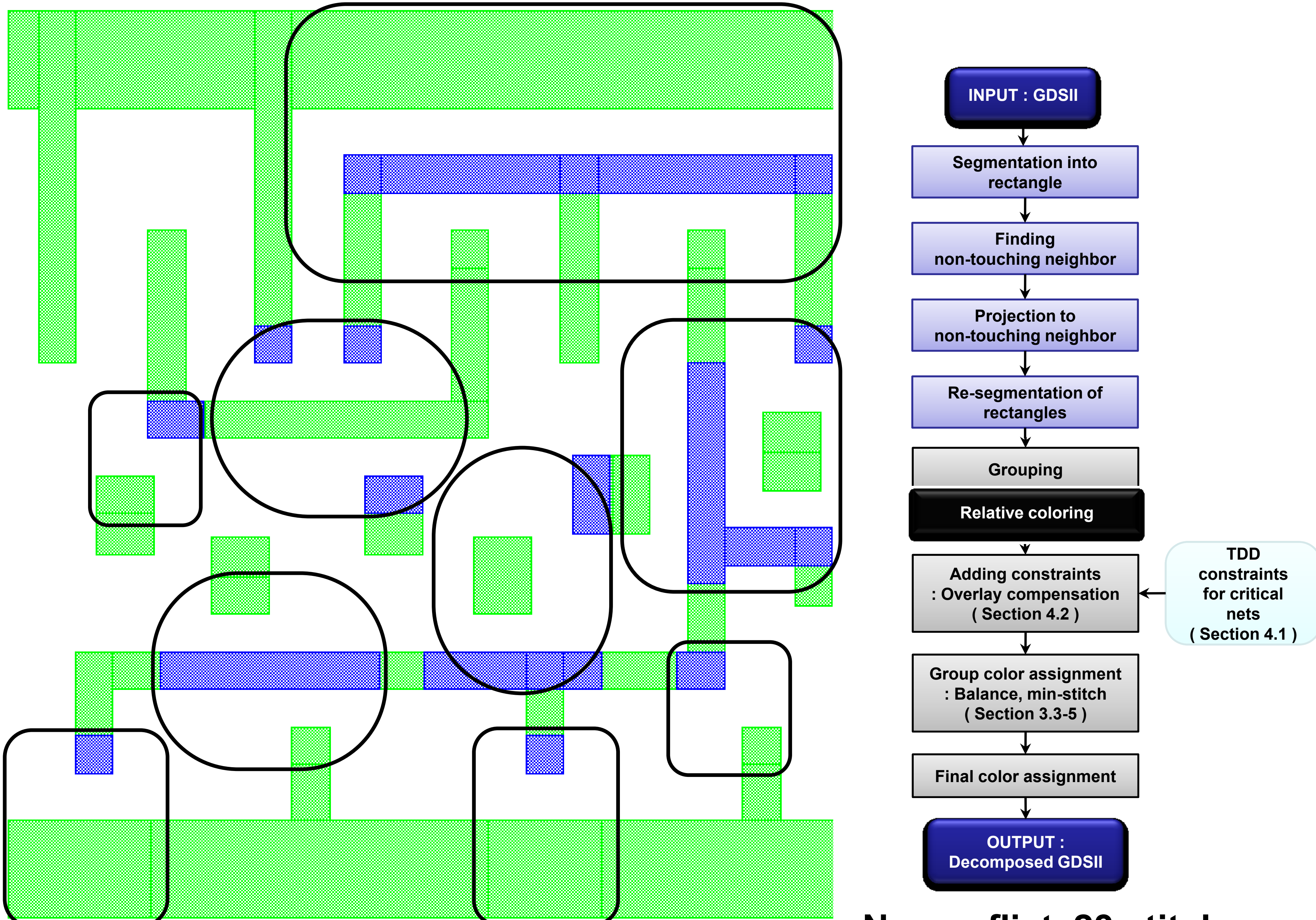


Grouping and Relative Coloring is done by DFS(Complexity : O(N+E))

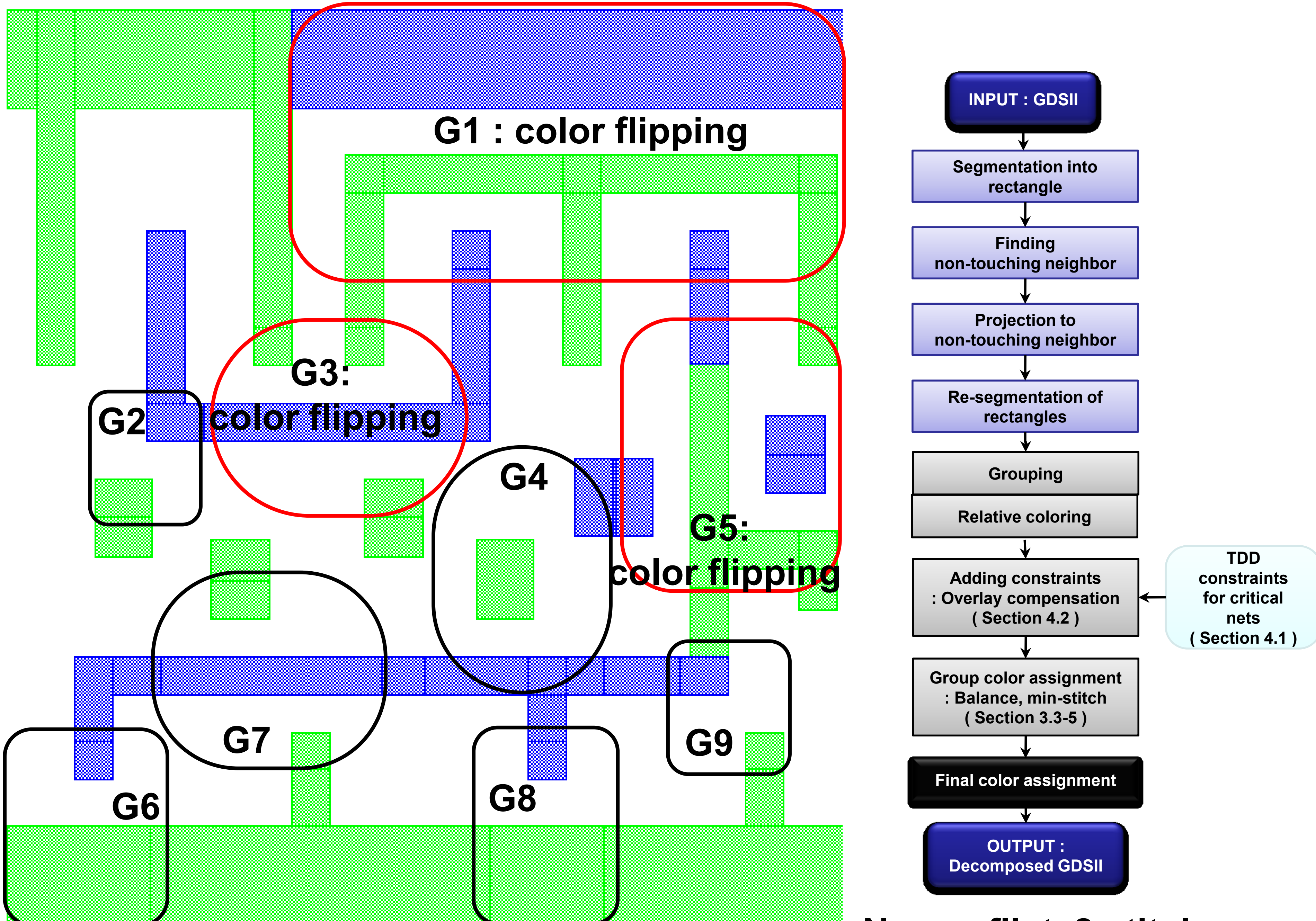
r1 and r4 should have different color
r3 and r5 should have different color
r2 can have any color

Relative coloring is a procedure assigning a color to remove conflicts


INPUT : GDSII
Segmentation into rectangle
Finding non-touching neighbor
Projection to non-touching neighbor
Re-segmentation of rectangles
Grouping
Relative coloring
Adding constraints : Overlay compensation (Section 4.2)
TDD constraints for critical nets (Section 4.1)
Group color assignment : Balance, min-stitch (Section 3.3-5)
Final color assignment
OUTPUT : Decomposed GDSII



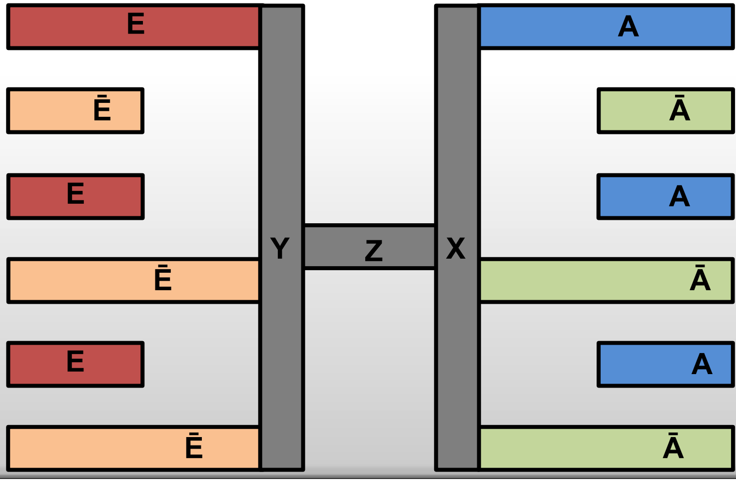
No conflict, 23 stitches



No conflict, 2 stitches

Color Assignment – Exact Solution

Example of Stitch Minimization



Minimize :

A⊕X + X⊕Z + Y⊕Z + E⊕Y + 2(Ā⊕X) + 2(Ē⊕Y)

Subject to :

A∈{0,1}, E∈{0,1}, X ∈{0,1}, Y ∈{0,1}, Z ∈{0,1}

ILP formulation for Stitch Minimization (Exact Solution)

$$Minimize : \sum X_w$$

$$Subject\ \ To :$$

$$: A_w - B_w \le X_w$$

$$: B_w - A_w \le X_w$$

N groups in layout

➔2^N solution

➔NP-Complete

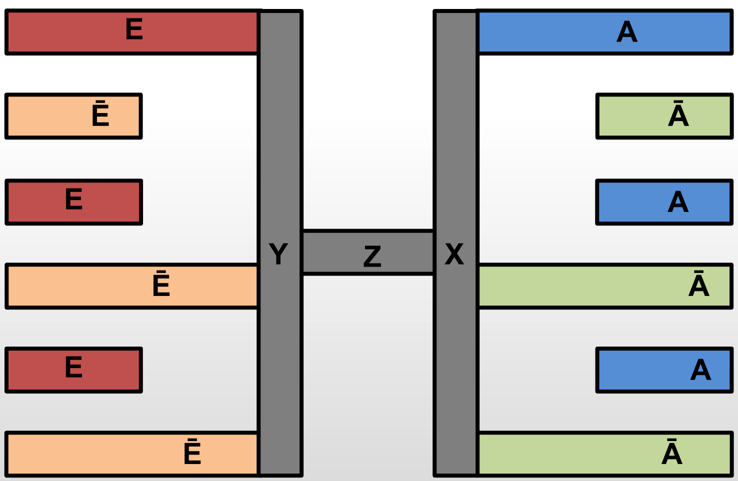
➔Need a heuristic method

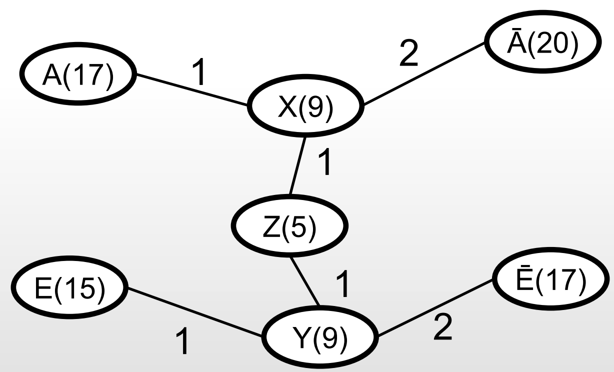
Color Assignment – Heuristic Solution

Theorem 1 : Min-Cut Based Stitch Minimization

The number of stitches in layout decomposition is equal to the cut size of the bi-partitioning problem in graph theory.

Example of Graph Based Stitch Minimization





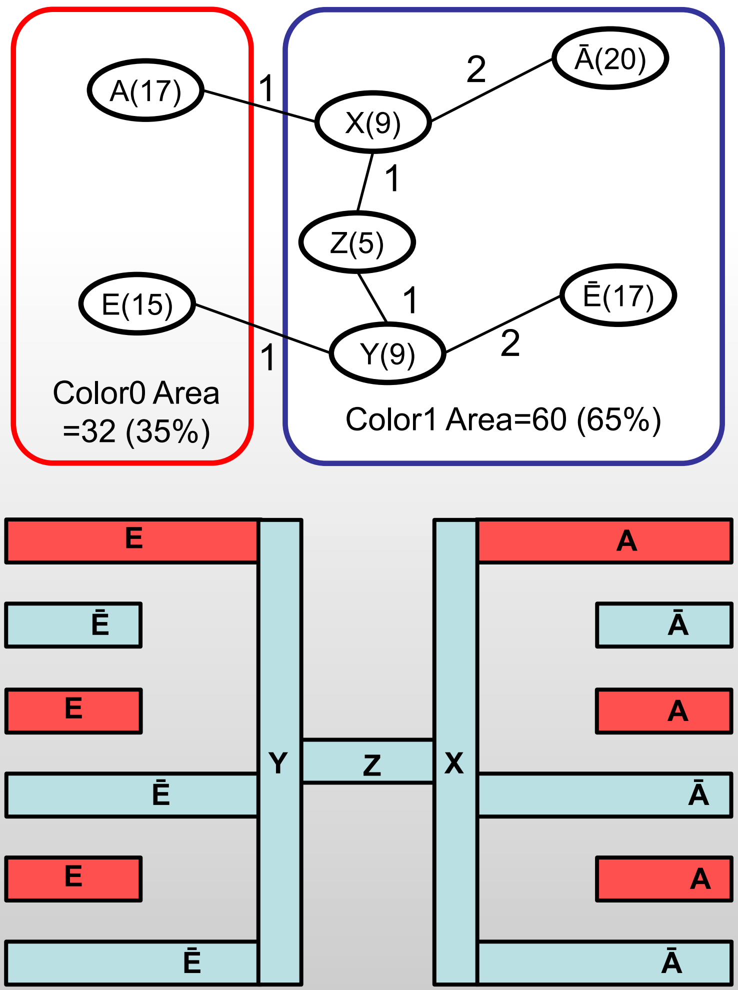
Minimize :

$$A\oplus X + X\oplus Z + Y\oplus Z + E\oplus Y + 2(\bar{A}\oplus X) + 2(\bar{E}\oplus Y)$$

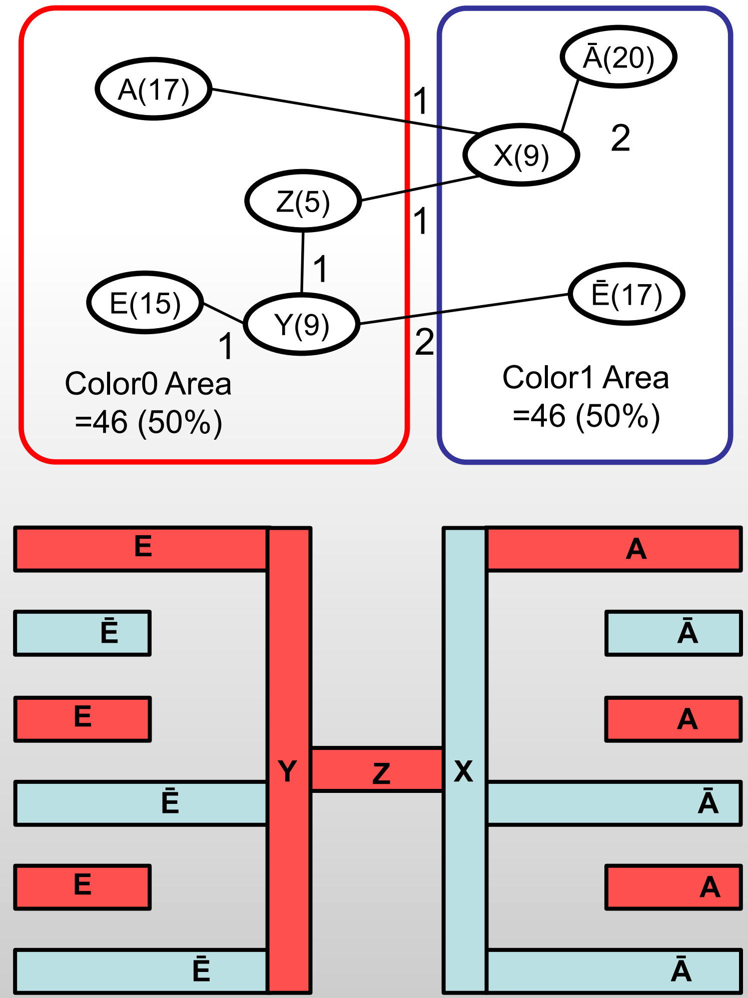
Constraint:
$(A, \bar{A})$ and $(E, \bar{E})$ are repulsive pairs.

Graph Partitioning Based Decomposition

Min-Stitch Coloring

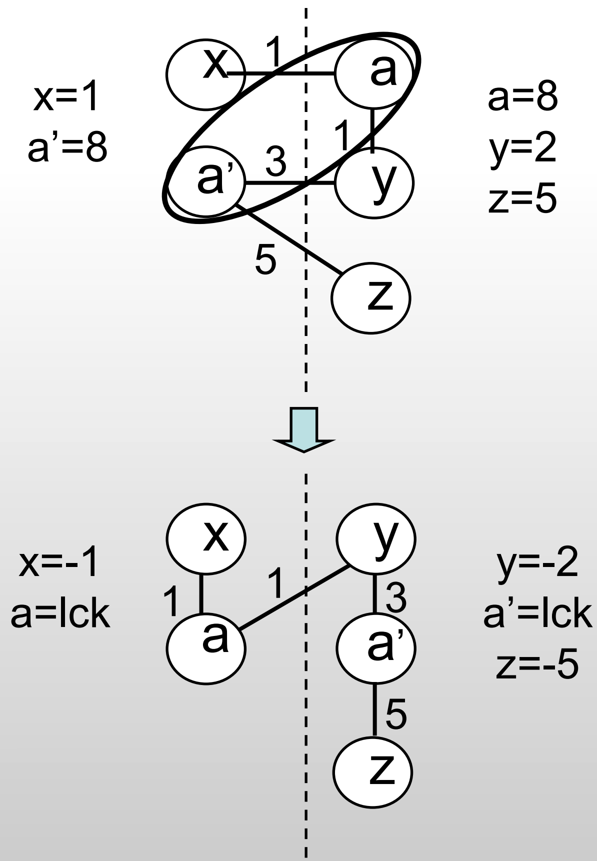


Balanced Coloring



Modification of FM partitioning

Repulsive Pair Consideration



Local Density Consideration

R_{11}	R_{12}	R_{13}	…	R_{1i}
R_{21}	R_{22}	R_{23}	…	R_{2i}
R_{31}	R_{32}	R_{33}	…	R_{3i}
…	…	…	…	…
R_{j1}	R_{j2}	R_{j3}	…	R_{ji}

$$rW_{11} - smax_{11} \le |A_{11}| \le rW_{11} + smax_{11}$$

$$rW_{12} - smax_{12} \le |A_{12}| \le rW_{12} + smax_{12}$$

$$\vdots$$

$$rW_{ji} - smax_{ji} \le |A_{ji}| \le rW_{ji} + smax_{ji}$$

We implemented FM partitioning with the two new features

Minimize ΔDelay due to Overlay

1st Order Expression of ΔDelay

$$\sqrt{\alpha^2+\beta^2}sin(\theta+\phi)$$

$$\text{where,}\quad \alpha = 2AX^T - \sum_{n=1}^{i} a_n, \quad \beta = 2BY^T - \sum_{n=1}^{j} b_n$$

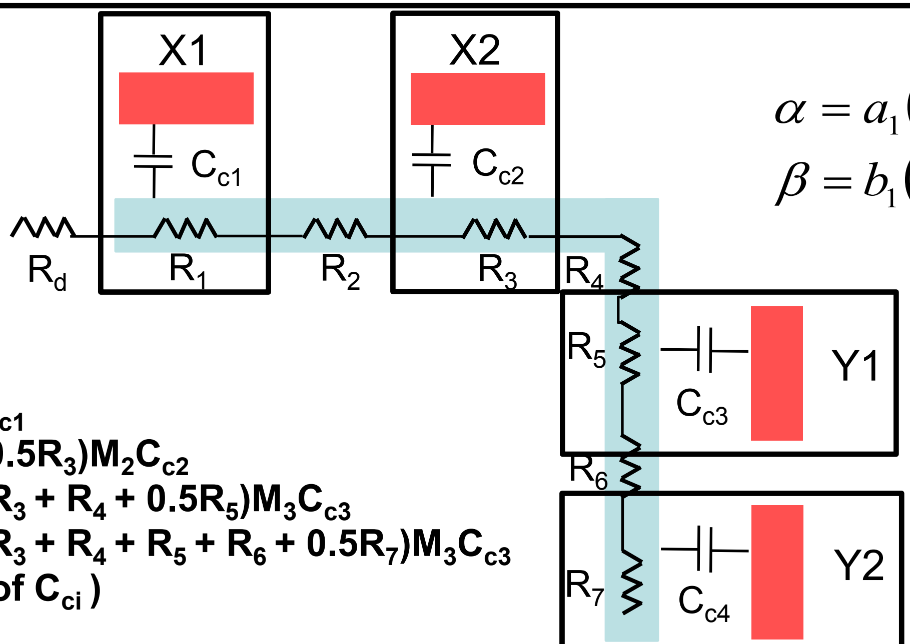
$$X = [x_1, x_2, \ldots, x_i], \quad Y = [y_1, y_2, \ldots, y_j]$$

$$\phi = sin^{-1}\left(\frac{\beta}{\sqrt{\alpha^2+\beta^2}}\right)$$

Theoram2

Horizontal direction and Vertical direction can be optimized independently to minimize ΔDelay due to Overlay

Example



$$\alpha = a_1(2x_1-1)+a_2(2x_2-1)$$
$$\beta = b_1(2y_1-1)+b_2(2y_2-1)$$

Minimize α² and β² independently to minimize ΔDelay due to overlay

$a_1 = (R_d + 0.5R_1)M_1C_{c1}$
$a_2 = (R_d + R_1 + R_2 + 0.5R_3)M_2C_{c2}$
$b_1 = (R_d + R_1 + R_2 + R_3 + R_4 + 0.5R_5)M_3C_{c3}$
$b_2 = (R_d + R_1 + R_2 + R_3 + R_4 + R_5 + R_6 + 0.5R_7)M_3C_{c3}$
(M_i : Miller Factor of C_{ci})

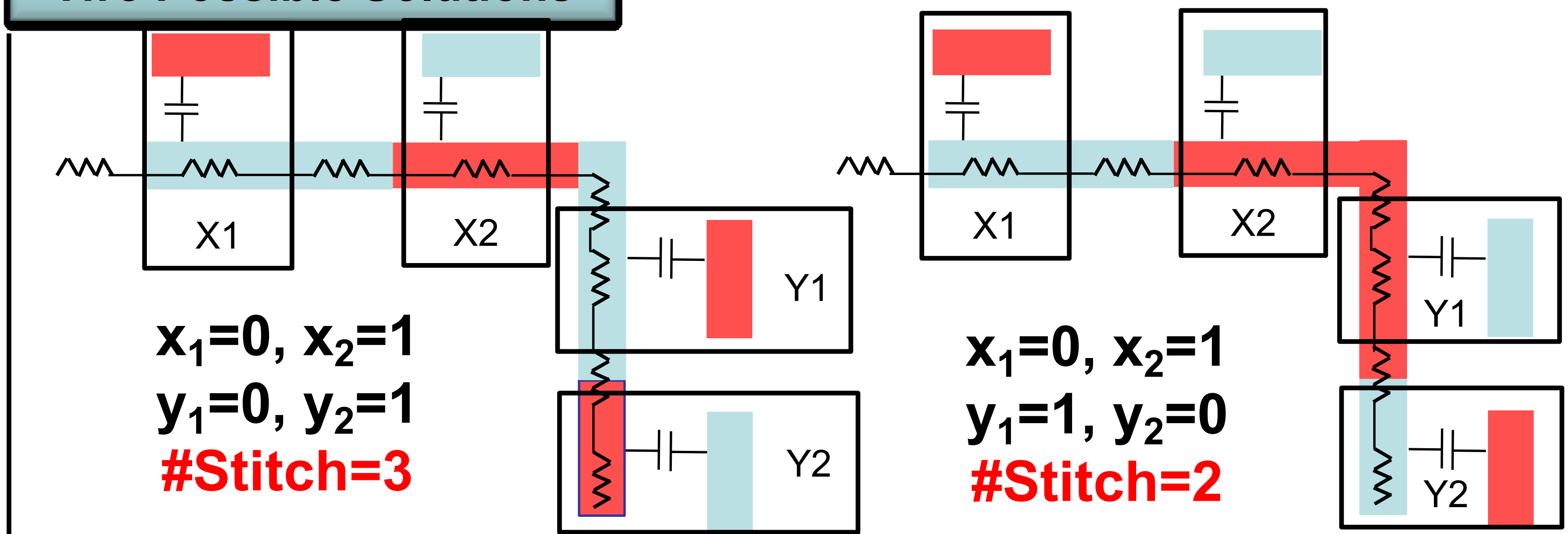
Timing Driven Decomposition(TDD)

ILP Formulation to minimize α^2

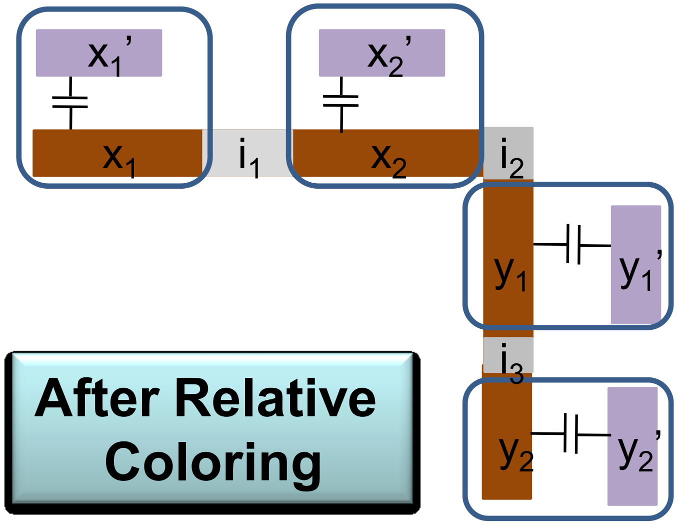
$$\text{minimize} \quad 4\sum_{n=1}^{i}\left\{a_n\left(AW_n^T - w_{nn}\sum_{p=1}^{i}a_p\right)\right\} + \left(\sum_{p=1}^{i}a_p\right)^2$$

$$\begin{aligned} s.t. \quad & w_{ii} = x_i \\ & 1 + w_{ij} \geq x_i + x_j \\ & x_i \geq w_{ij} \\ & x_j \geq w_{ij} \end{aligned}$$

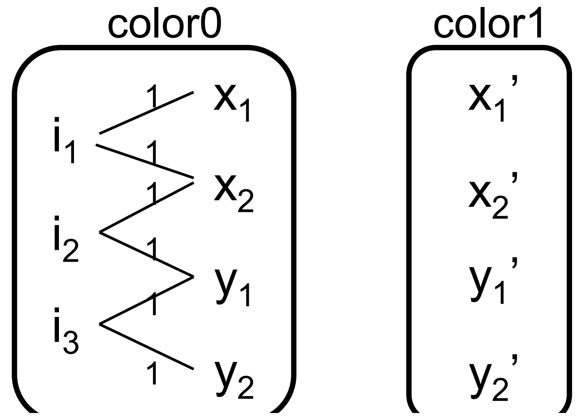
Two Possible Solutions



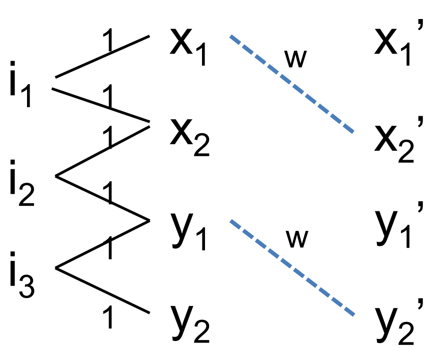
Decomposition with TDD constraint



After Relative Coloring

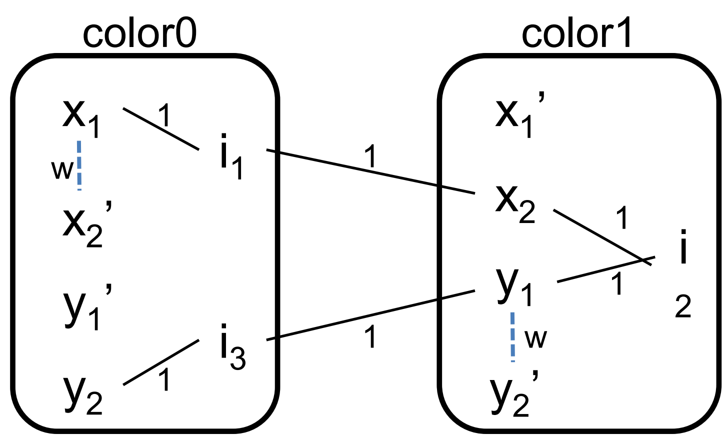


Group color assignment without TDD constraints: Zero stitch



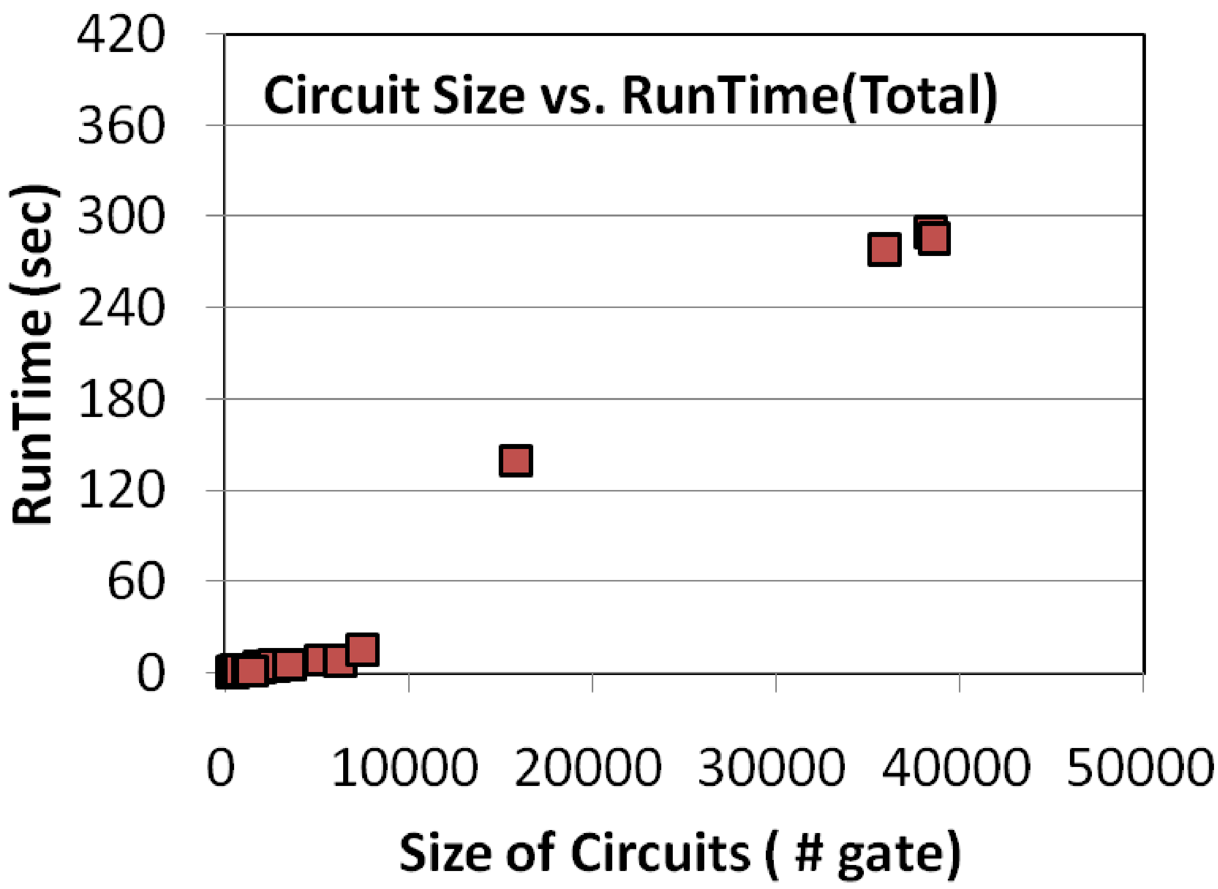
$X_0=\{x_1, x_2'\}$, $Y_0=\{y_1, y_2'\}$ to minimize α^2,β^2

TDD Constraints insertion



Group color assignment when the edge weight(w) is bigger than one: Two stitches

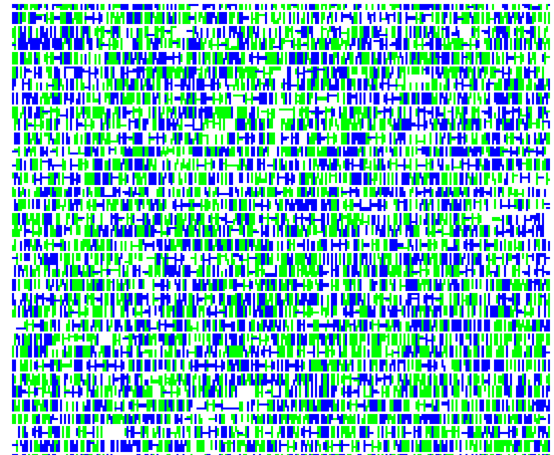
Runtime Result



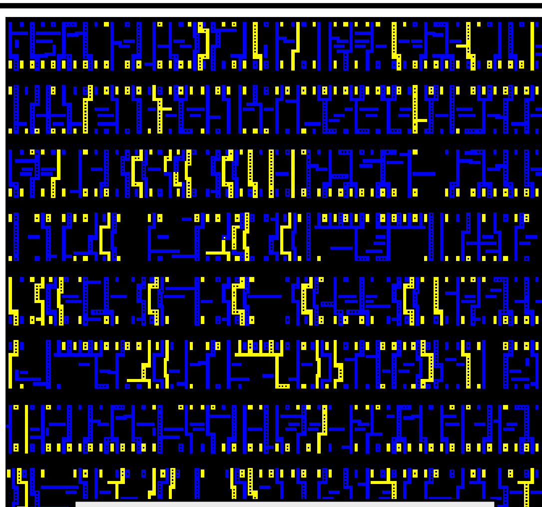
Complexity: O(NlogN) ➔ Don't need layout partitioning

Balanced Density Result

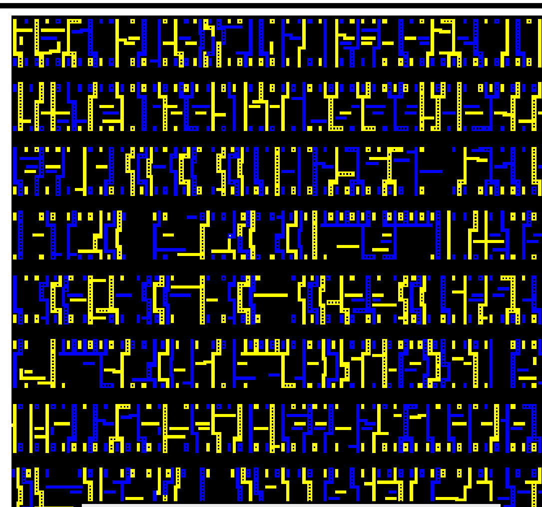
S38584:13% and 87%



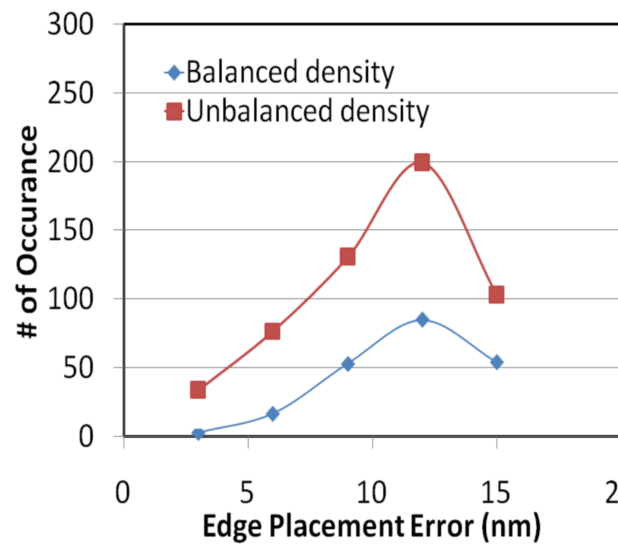
S38584: 50% and 50%



C432:27% and 73%
(7 stitches)



C432:50% and 50%
(17 stitches)



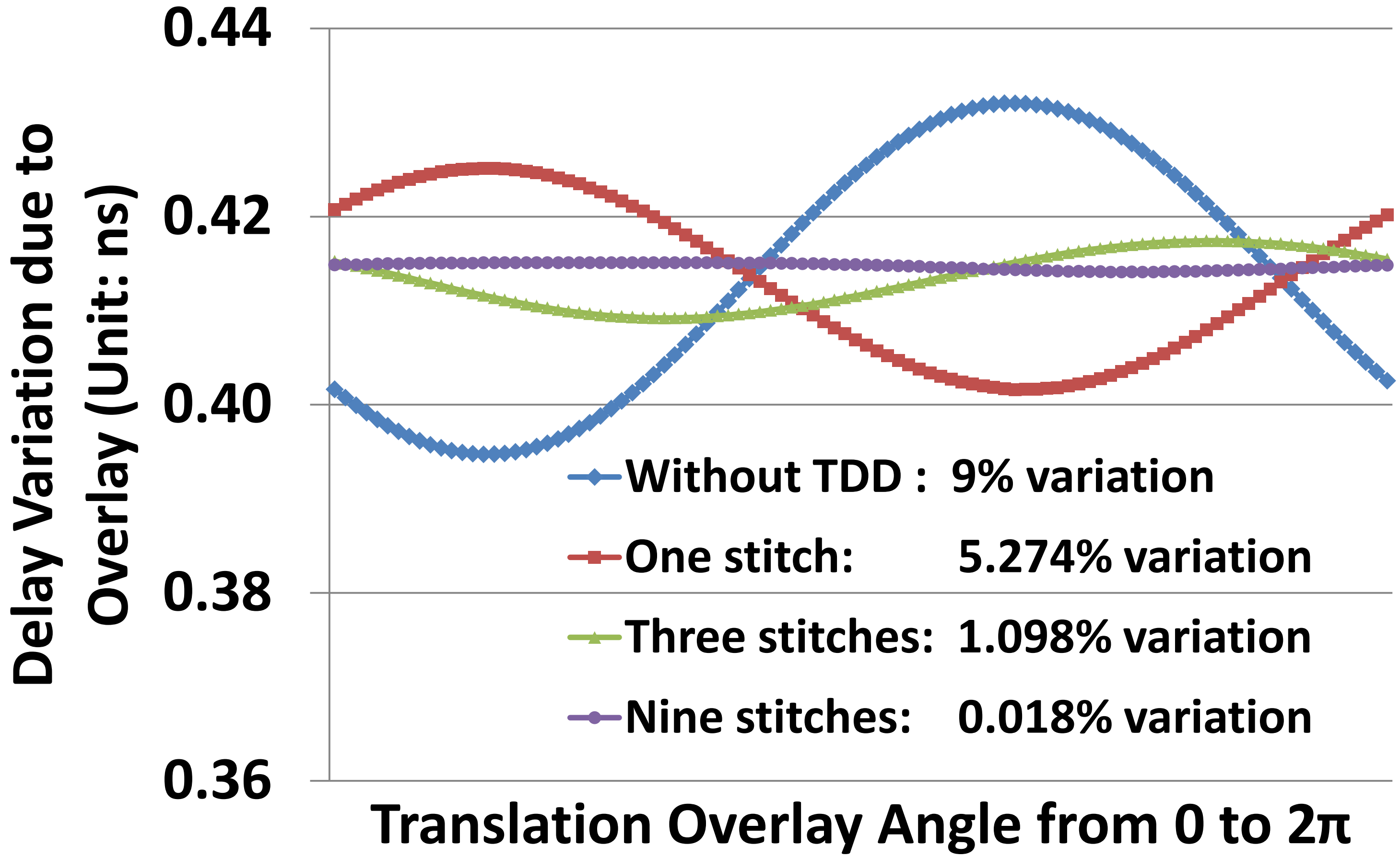
Exact vs. Heuristic comparison

Circuit	#Groups	#Touching neighbors	No balance, ILP (Exact)				No balance, Graph Partition (Proposed heuristic)					48% balance, Graph Partition (Proposed heuristic)			
			# Partitions for ILP	RunTime (total)	Inserted stitches	Balanced ratio(%)	RunTime comparison	RunTime (total)	Inserted stitches	Balanced ratio(%)	RunTime comparison	RunTime (total)	Inserted stitches	Balanced ratio(%)
C432	1512	1098	1	0.63	1	20.35	x1.4	0.46	1	33.60	x1.0	0.65	2	48.12
C499	3103	3280	12	100.85	50	24.01	x49.9	2.02	50	46.47	x49.9	2.02	50	48.50
C880	3758	2631	14	4525.57	198	30.09	x2773.0	1.63	198	47.12	x2807.4	1.61	198	48.87
C1355	4836	3083	18	702.4	114	18.91	x347.4	2.02	114	36.12	x344.0	2.04	114	48.00
C1908	7795	5472	18	37019.7	371	22.09	x9762.6	3.79	372	46.78	x10422	3.55	373	48.66
C2670	12863	9905	-	> 24Hr	-	-	-	6.7	947	43.51	-	6.87	948	49.30
C3540	16638	12021	-	> 24Hr	-	-	-	9.85	1034	41.46	-	10.07	1034	49.39
C5315	24483	18373	-	> 24Hr	-	-	-	17.43	1546	40.87	-	18.5	1549	48.00
C6288	19922	11577	-	> 24Hr	-	-	-	11.57	256	30.81	-	11.25	256	48.13
C7552	34309	24789	-	> 24Hr	-	-	-	30.89	2058	41.97	-	31.52	2060	48.02

Runtime : Bi-partitioning based decomposition is up to 10K faster than ILP based decomposition.

Accuracy : C1908 has two more stitches in our heuristic algorithm. All benchmarks except C1908 have the same #stitches.

Overlay compensation result



We could compensate overlay effect on timing
More stitches➔ Less overlay effect on timing

Conclusion & Future Works

- Graph-based multi-objective decomposition
  - Super linear time complexity : O(NlogN)
  - Stitch minimization
  - Balanced density
  - Constraint insertion : overlay compensation
- Future work
  - Multiple Decomposition for Multiple Patterning
  - Correlation Aware Decomposition