

Triple Patterning Aware Detailed Placement Toward Zero Cross-Row Middle-of- Line Conflict

Yibo Lin¹, Bei Yu^{1,2}, Biying Xu¹, David Z. Pan¹

¹ECE Department, University of Texas at Austin

²CSE Department, Chinese University of Hong Kong

This work is supported in part by NSF and SRC

Outline



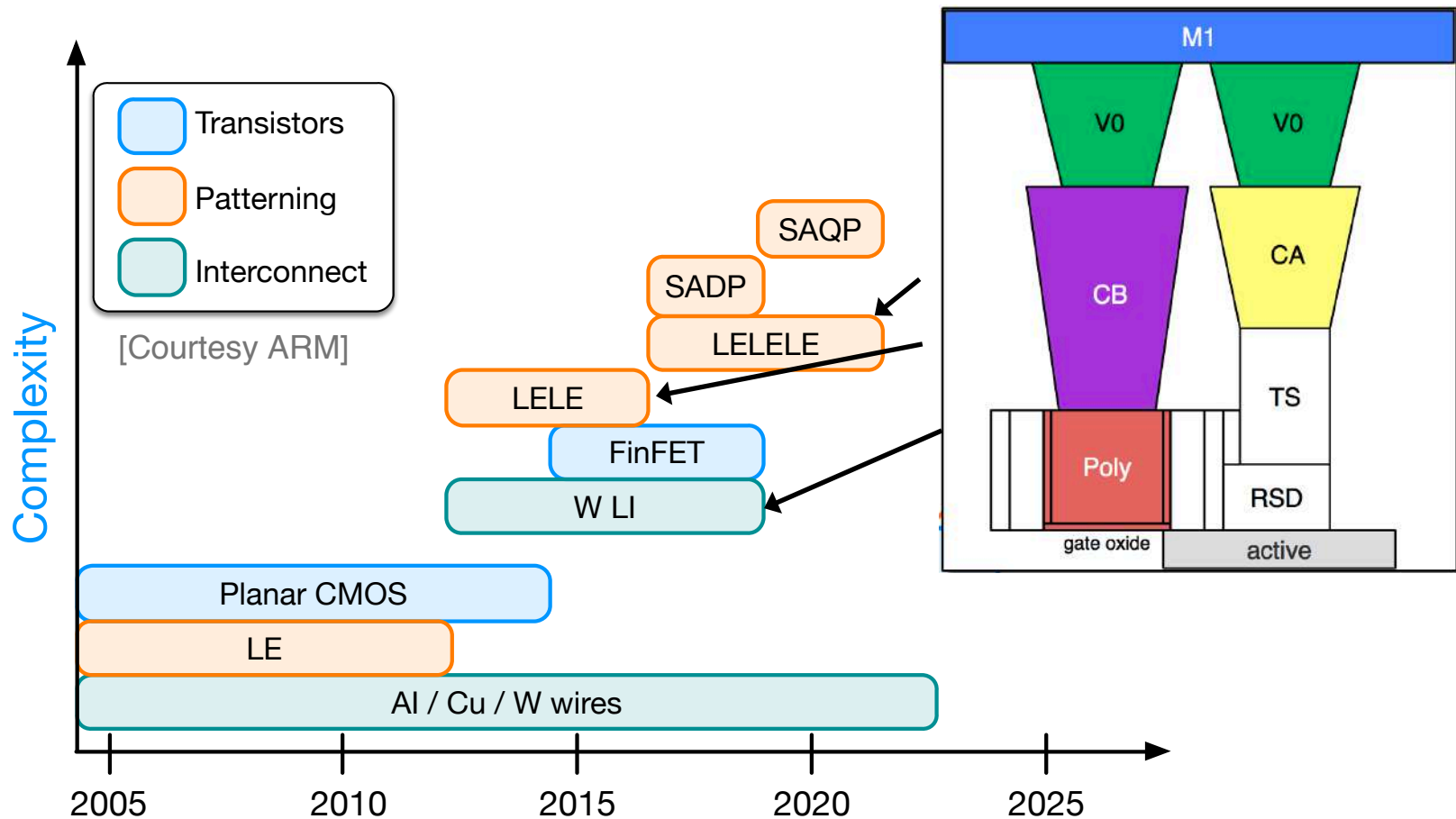
- Introduction
- Previous Work
- Problem Formulation
- Conflict Detection
- TPL Aware Detailed Placement
- Experimental Results
- Conclusion

Introduction



• Technology Scaling

- Middle-of-line (MOL) layers: CA, CB
- Multiple patterning lithography [Lucas+, SPIE'12]

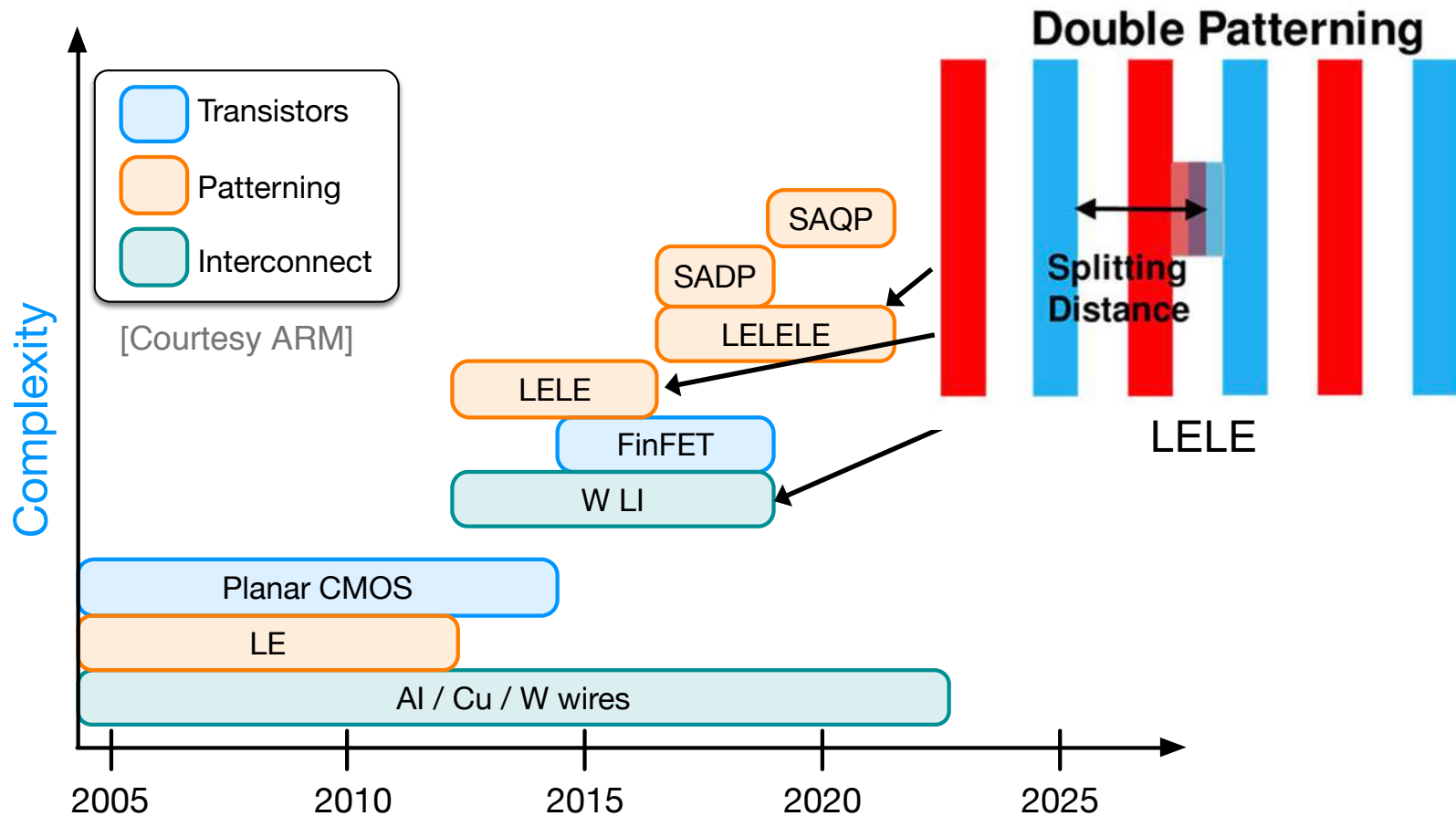


Introduction



- Technology Scaling

- Middle-of-line (MOL) layers: CA, CB
- Multiple patterning lithography [Lucas+, SPIE'12]

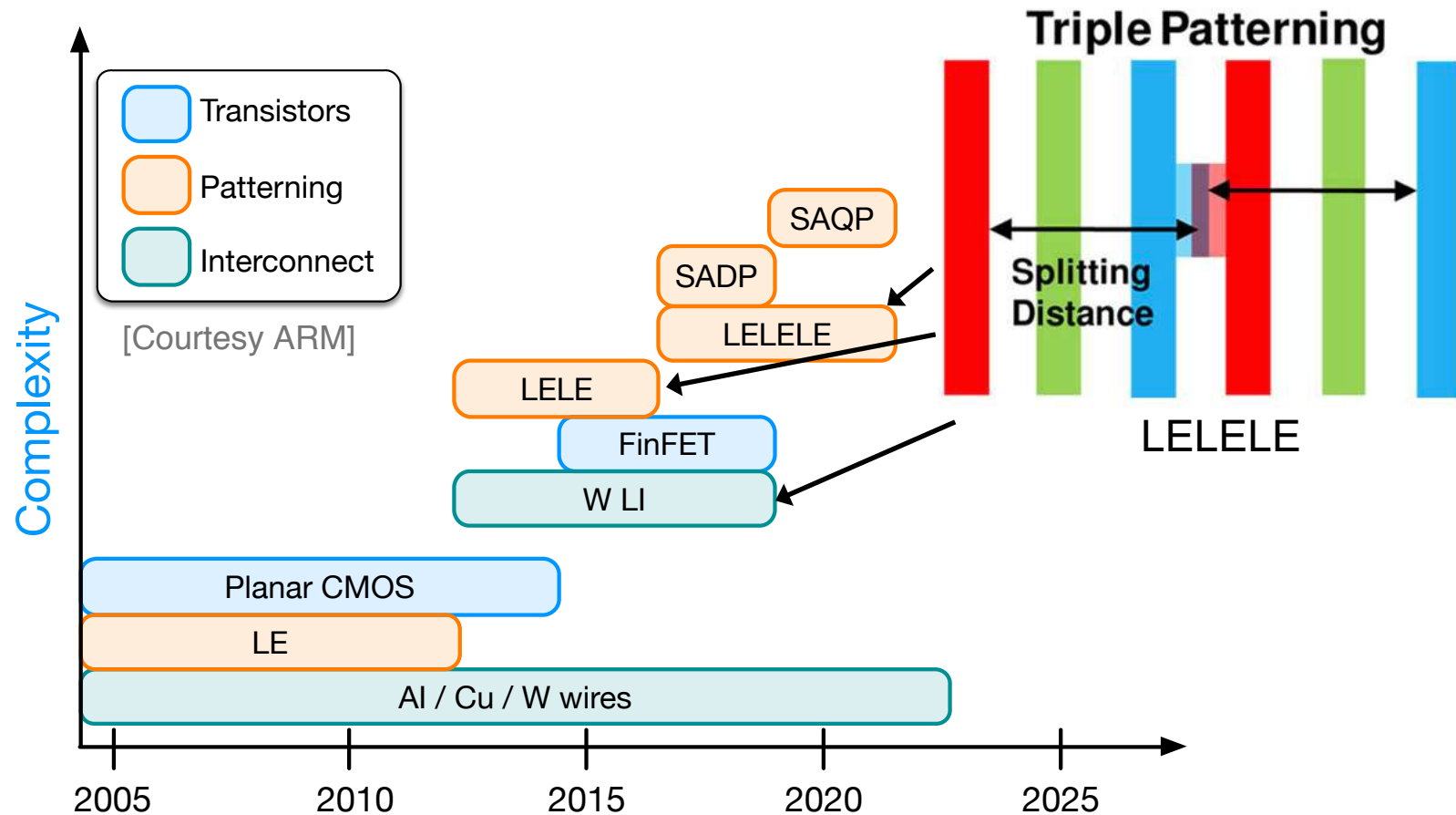


Introduction

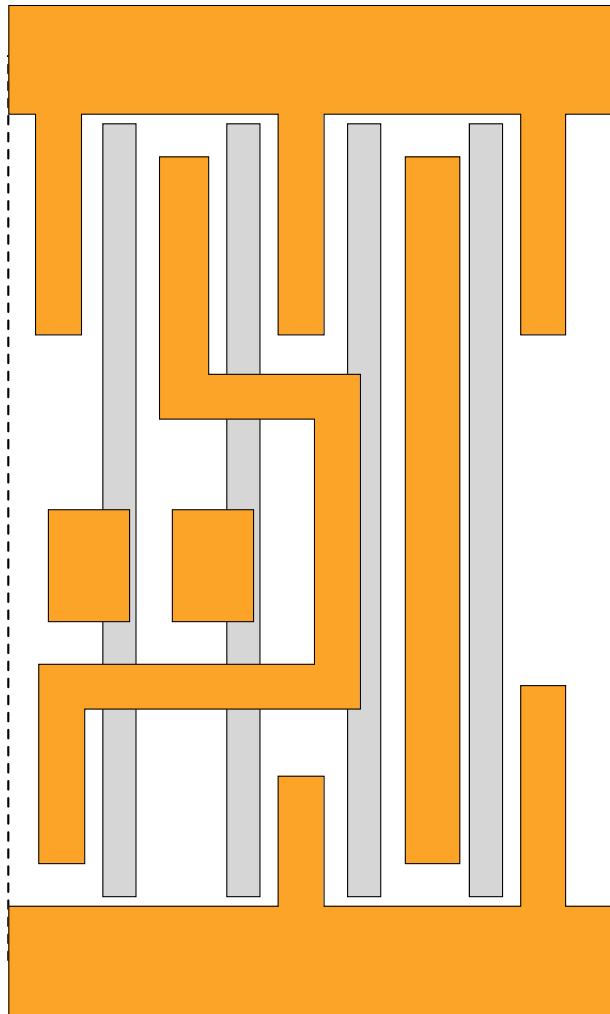
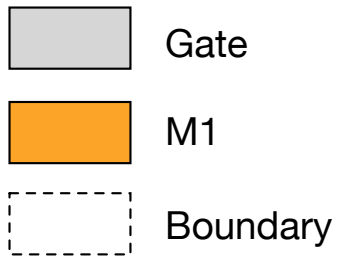


• Technology Scaling

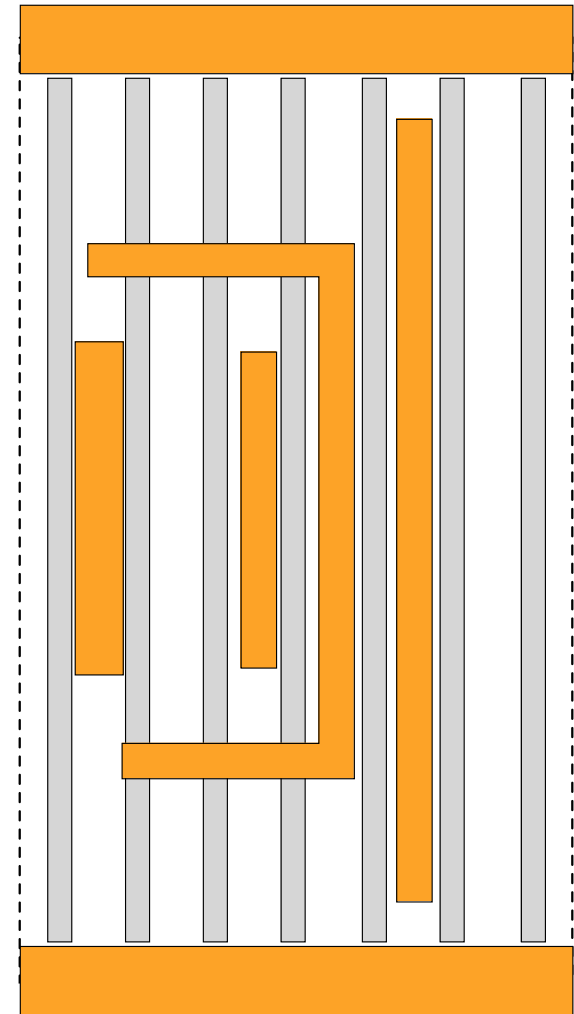
- Middle-of-line (MOL) layers: CA, CB
- Multiple patterning lithography [Lucas+, SPIE'12]



MOL to Simplify Metal-1 Cell Routing

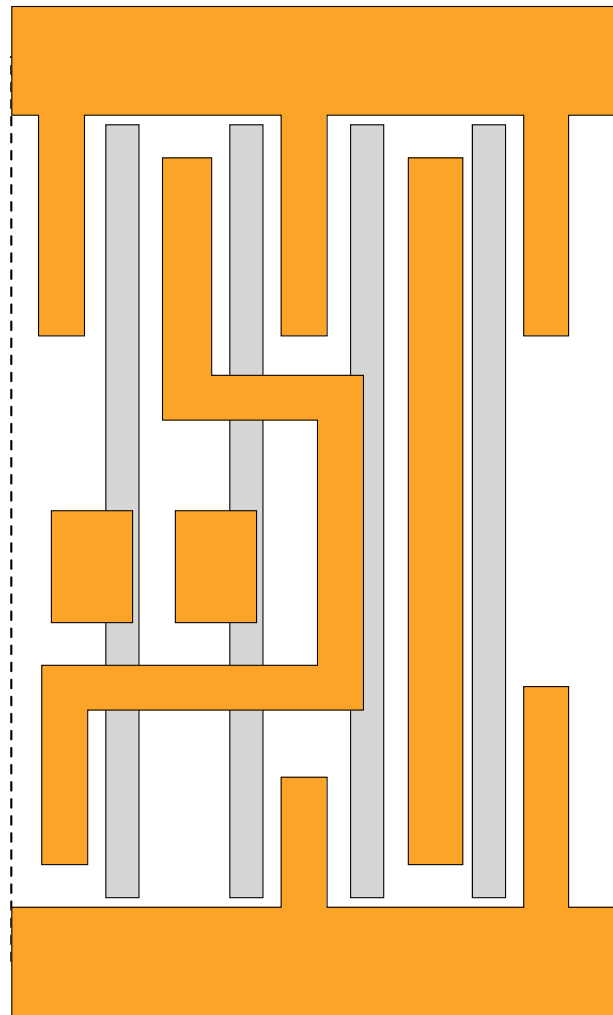
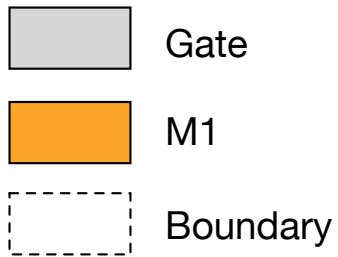


AND2_X2 in Nangate 45nm Library

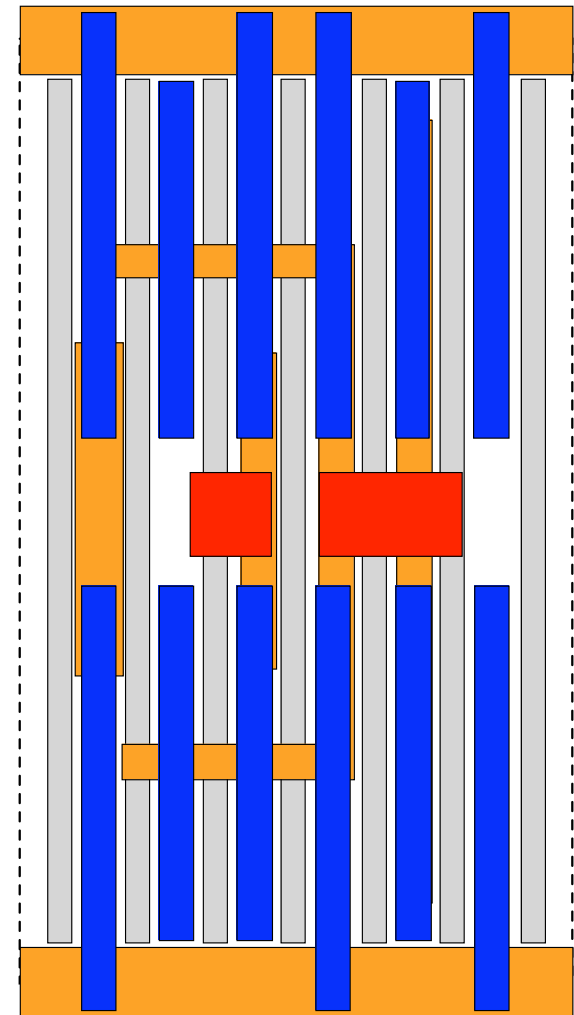


AND2_X2 in Nangate 15nm Library

MOL to Simplify Metal-1 Cell Routing



AND2_X2 in Nangate 45nm Library

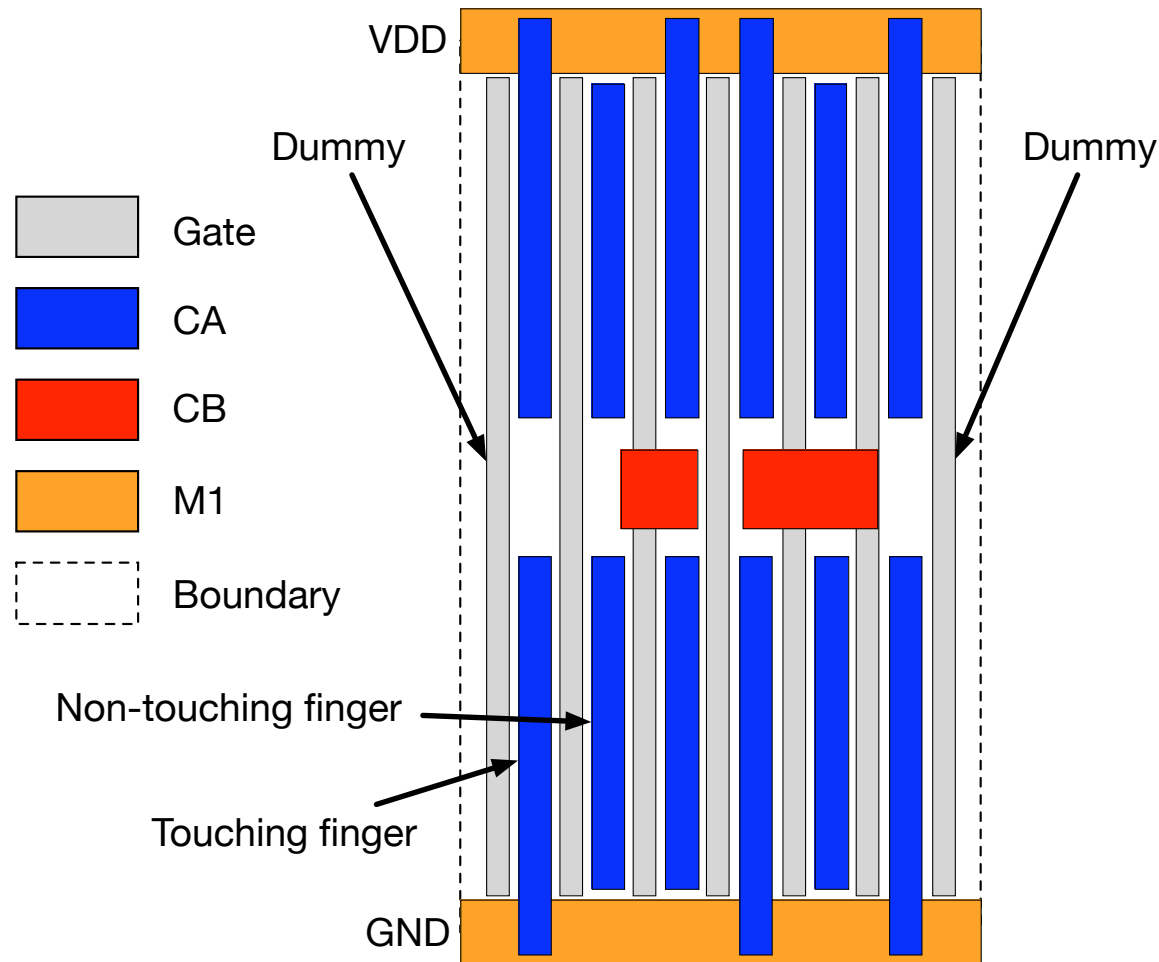


AND2_X2 in Nangate 15nm Library

Example of MOL Layer



- Tungsten layer
- CA (drain \leftrightarrow source), CB (via0 \leftrightarrow gate)

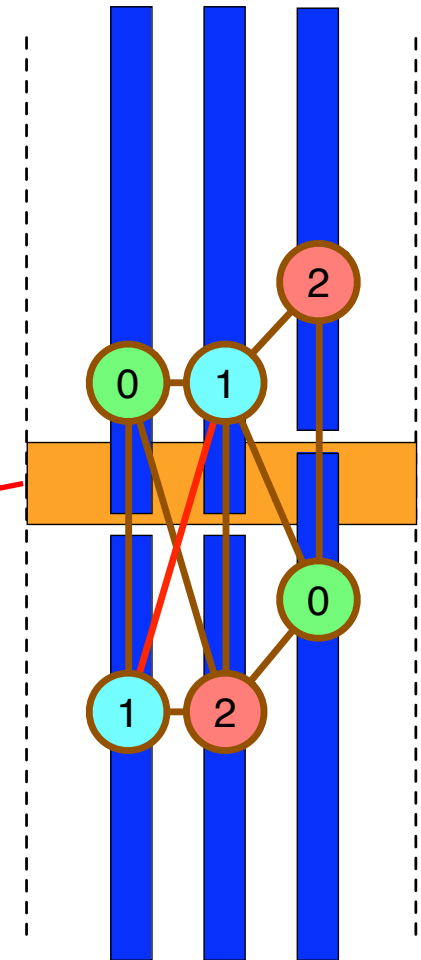
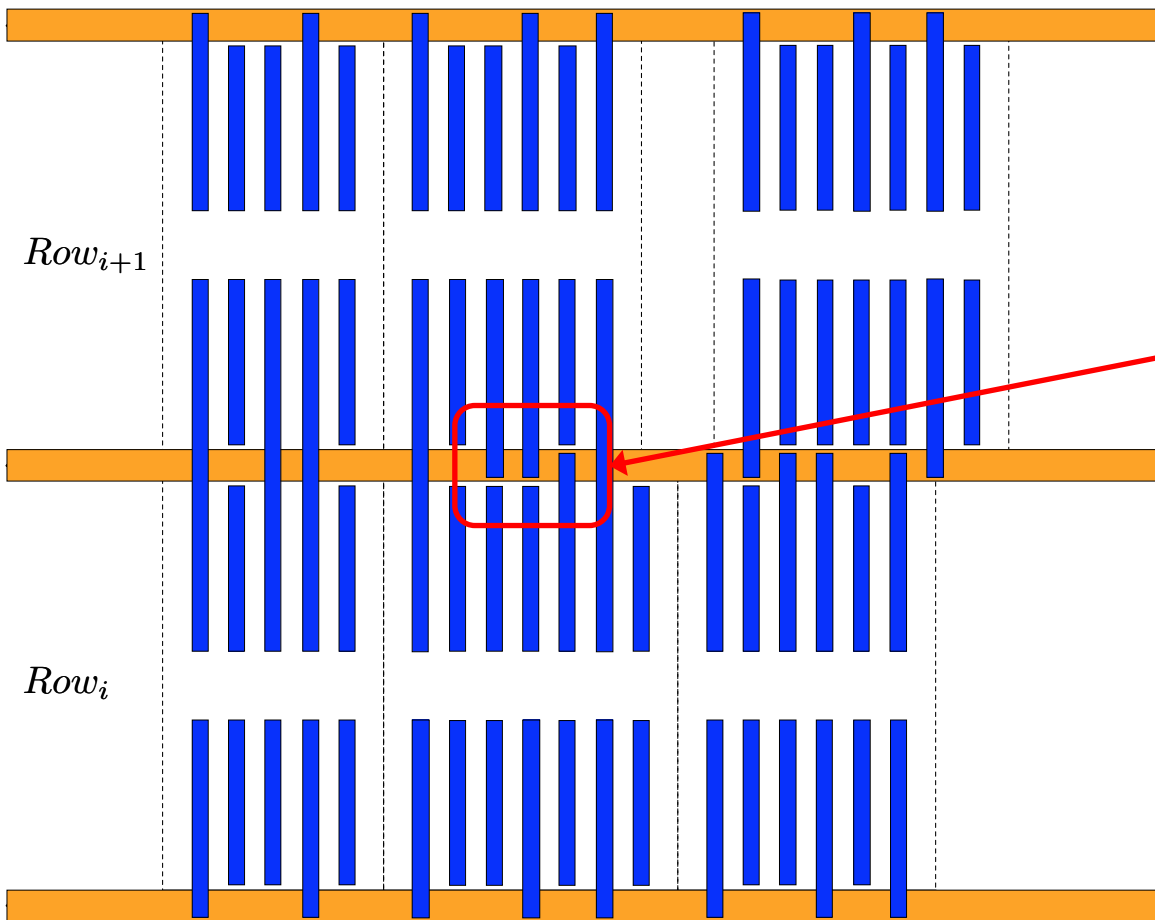


MOL layer in 15nm Nangate library

Cross-Row Conflict for MOL Layer



- 4-clique (K4) structure

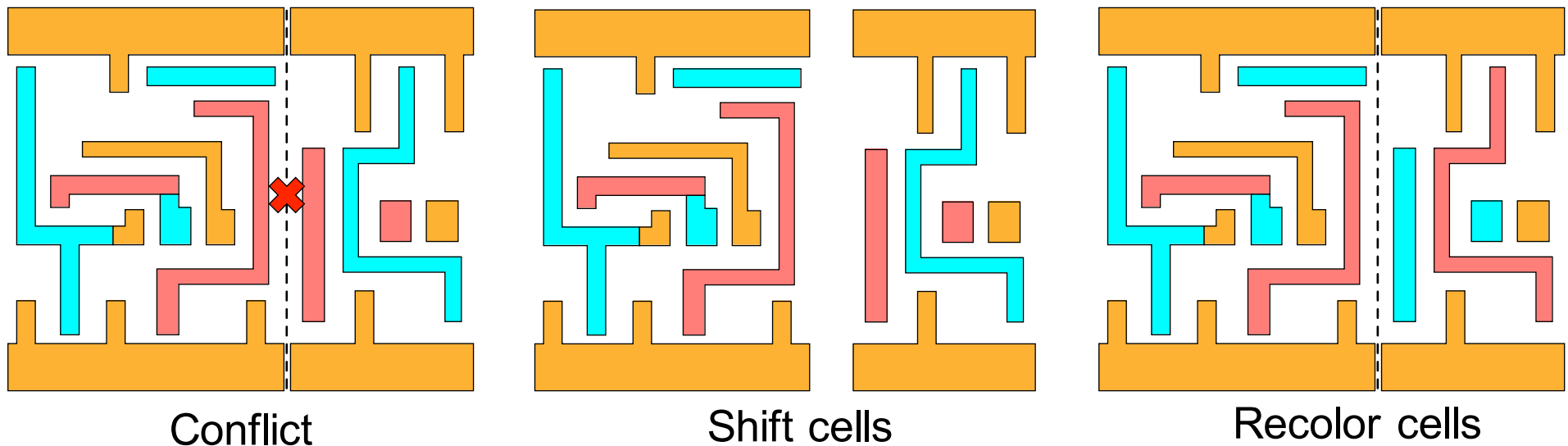


Possible cross row conflicts at MOL layer

Previous Work



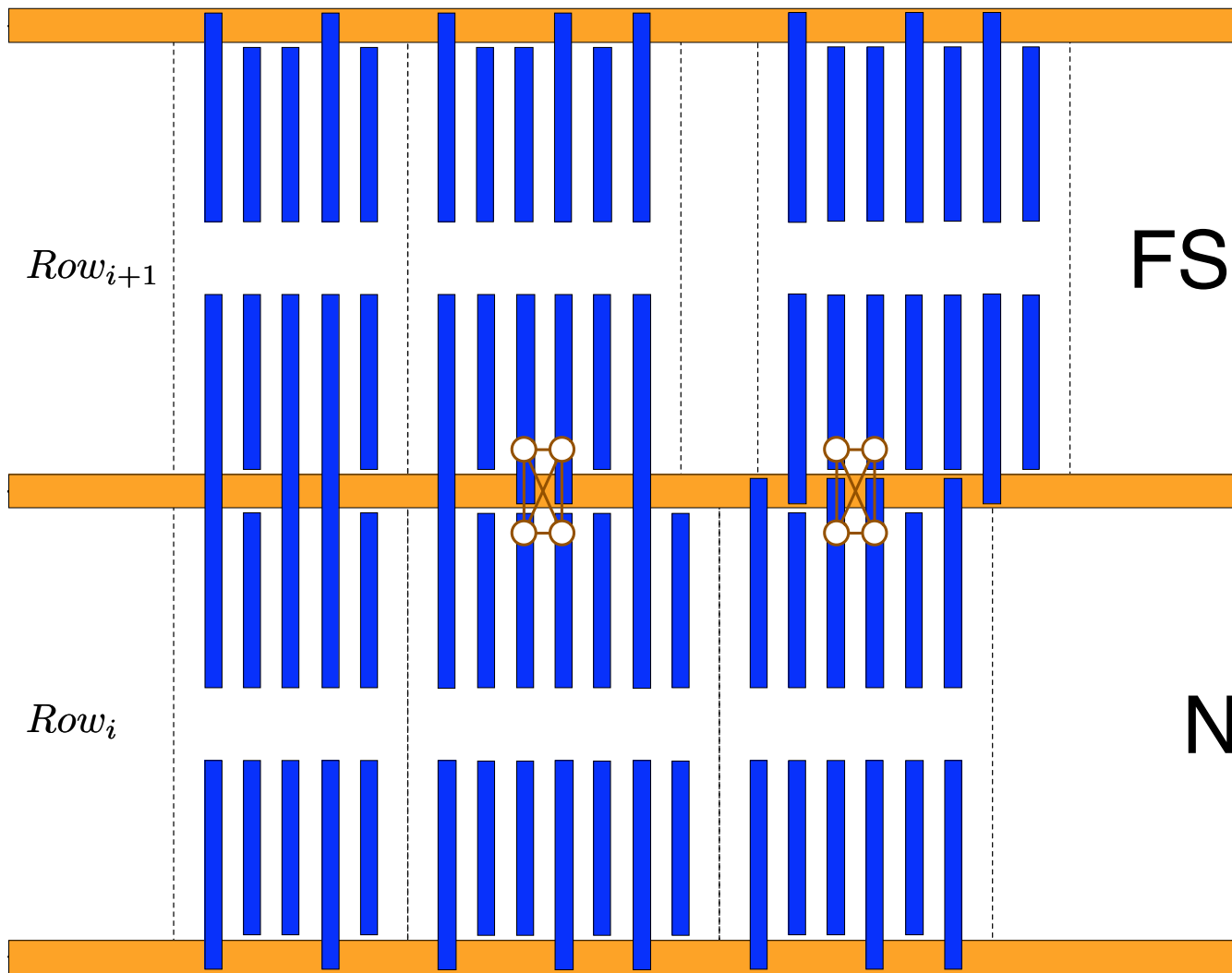
- TPL aware detailed placement for Metal-1 layer
 - [Yu+, ICCAD'13, TCAD'15], [Kuang+, ICCAD'14, TVLSI'15], [Chien+, TCAD'15]
 - [Tian+, ICCAD'14], [Lin+, ISPD'15]
- Limited to inner-row conflict
 - Assume rows are isolated by wide PG grids



Cross-Row Conflict for TPL



- MOL layer



Problem Formulation

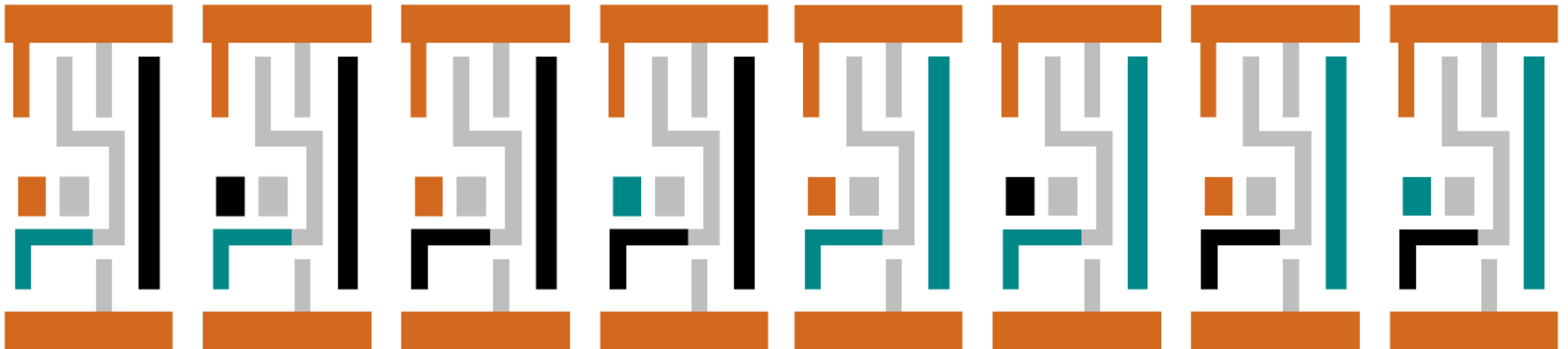


- Input
 - Standard cell library
 - Initial placement
- Output
 - New placement with optimized wirelength and minimum conflicts
 - Coloring solution for Metal-1 and MOL layer at standard cell level
 - TPL friendliness

Precoloring for Metal-1 Layer



- Inner-row conflicts
- Only features at cell boundary will result in conflicts
- Construct LUT to store candidate coloring solutions
- Limited number of candidate coloring solutions

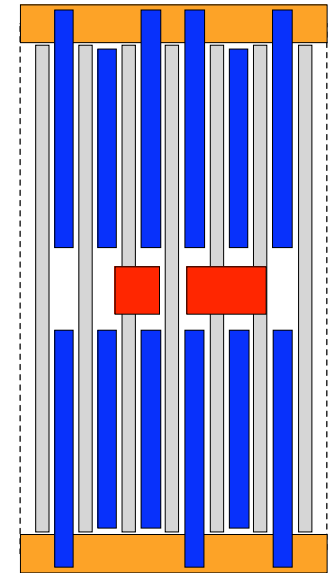
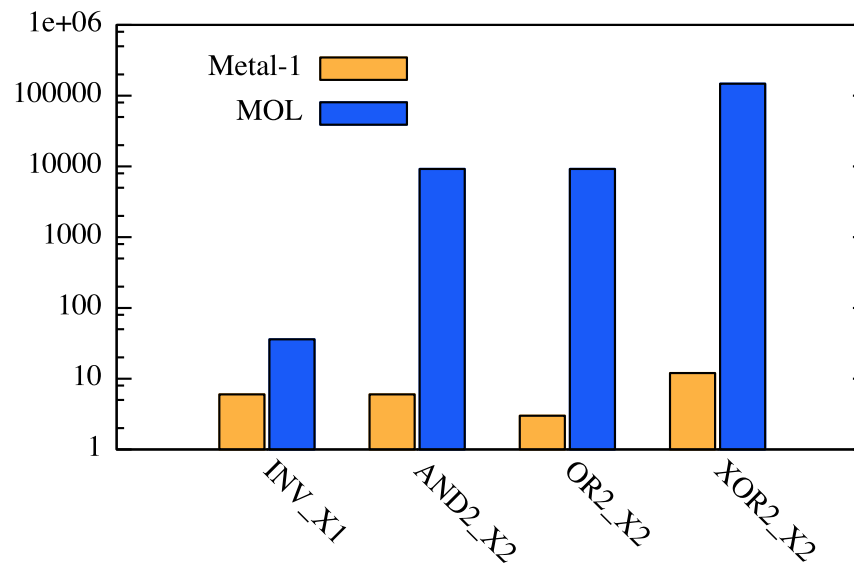


Precoloring solution for a single cell [Yu+, TCAD'15]

Precoloring for MOL Layer

- Conventional LUT like Metal-1 layer is infeasible
- Too many candidate coloring solutions

TPL candidates: $(3 \times 2^5) \times (3 \times 2^5) = 9216$

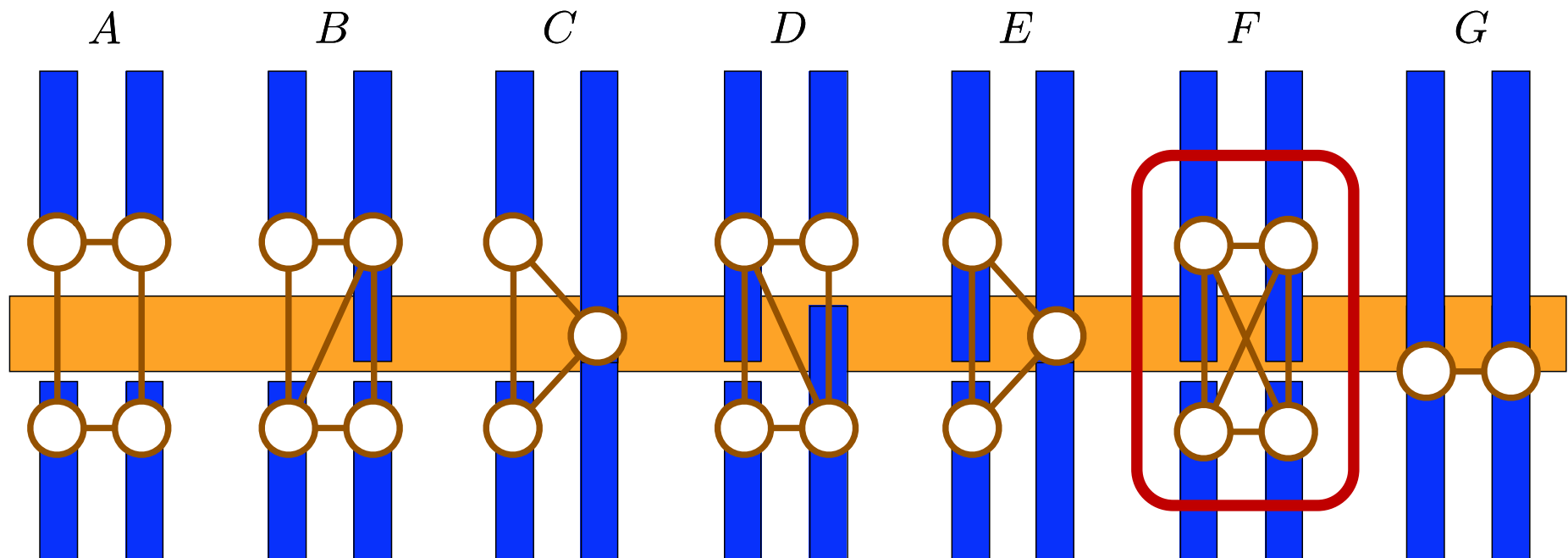


Number of MOL candidate coloring solutions for different cells

K4 Avoidance for MOL Layer



- For MOL layer, try to avoid K4 structures and assign colors after placement
 - Limited types of patterns
 - Due to the regularity of MOL layer, zero conflict is guaranteed if no K4 exists



Different types of patterns for MOL layer

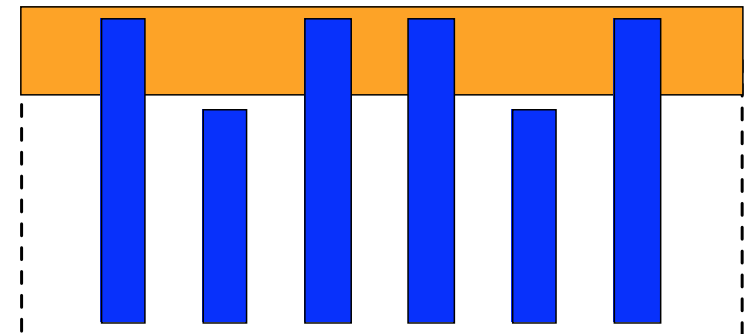
K4 Detection – BLUT



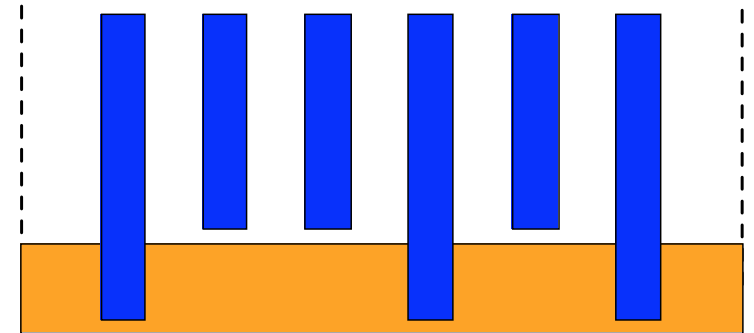
• Bitwise LUT (BLUT) for MOL Layer

- Goal: easier K4 detection
- Limited types of MOL fingers
- Bitwise operation to detect K4

B^{T2}	1	0	1	1	0	1
B^{N2}	0	1	0	0	1	0

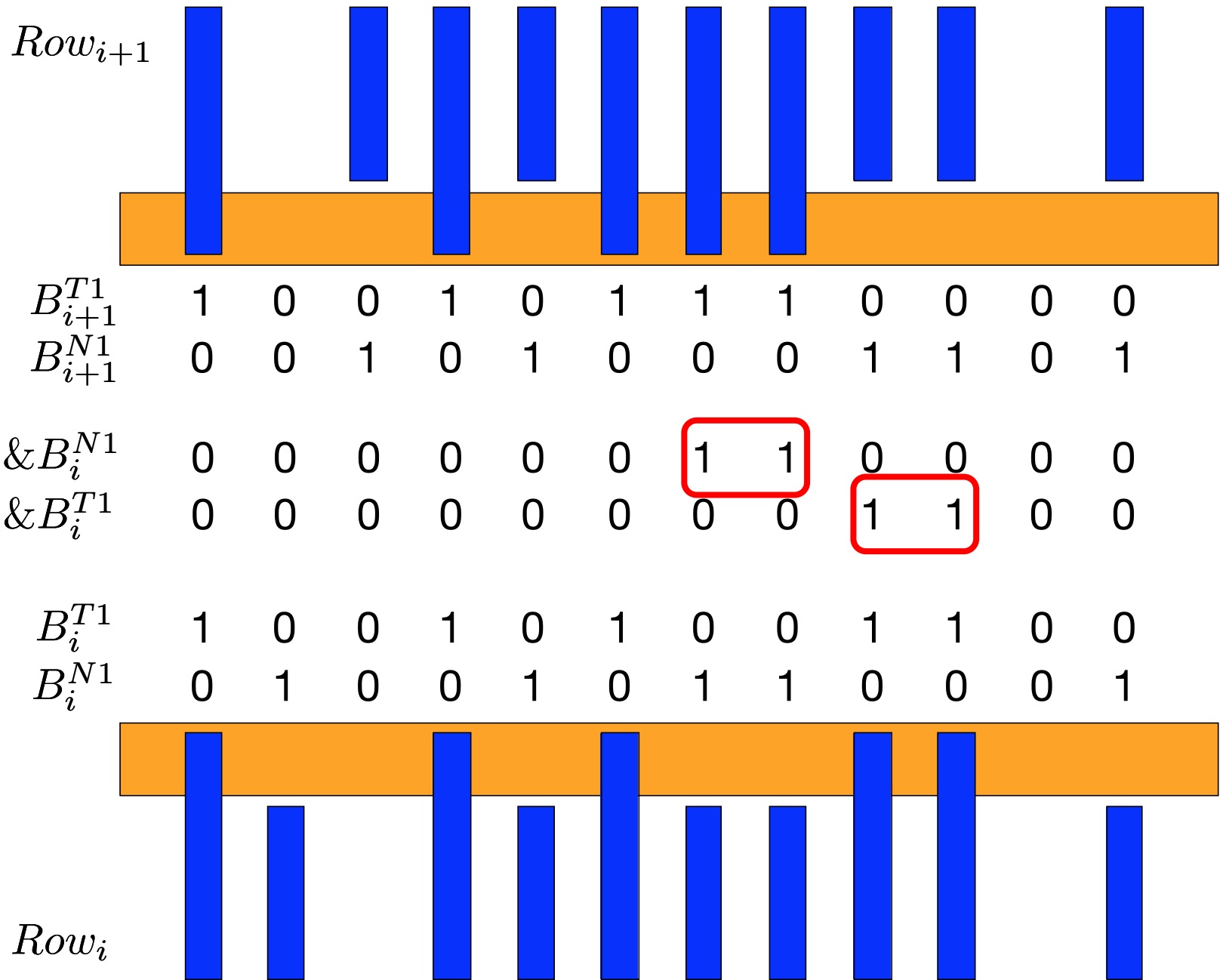


B^{T1} : touching finger bit array at cell bottom
 B^{N1} : non-touching finger bit array at cell bottom
 B^{T2} : touching finger bit array at cell top
 B^{N2} : non-touching finger bit array at cell top

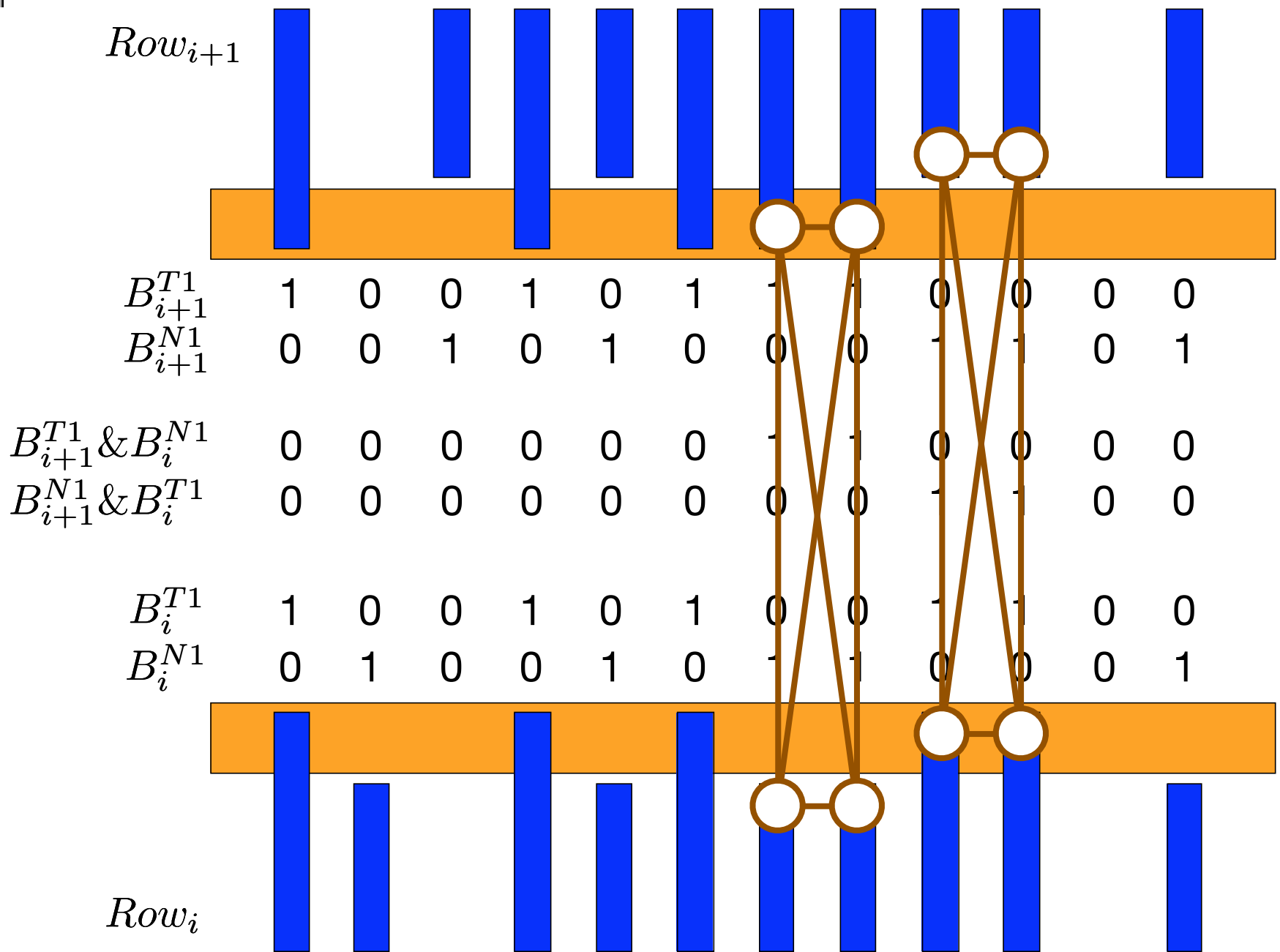


B^{T1}	1	0	0	1	0	1
B^{N1}	0	1	1	0	1	0

Example of BLUT Usage



Example of BLUT Usage

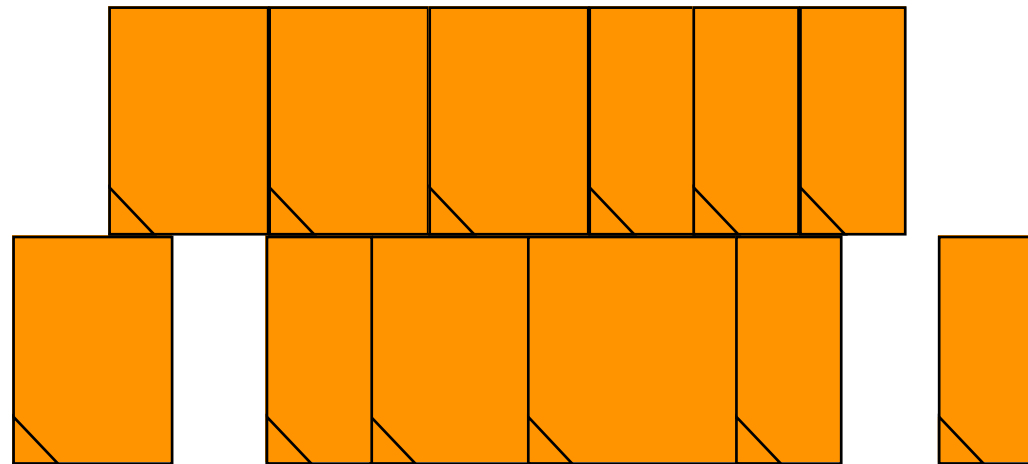


TPL Aware Detailed Placement



• Single Row Placement

- Construct a graph model to allow cell flipping, local reordering and local shifting
- Avoid Metal-1 conflict and MOL K4 (count as conflict)
- A graph model generalizes all three kinds of movement; edge cost determined by movement and conflicts (BLUT)
- Solve with graph model by shortest path



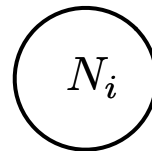
Cell flipping Local reordering Local shifting

$$edge\ cost = C_{move} + C_{stitch} + C_{conflict}$$

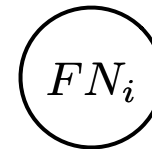
Cell Flipping



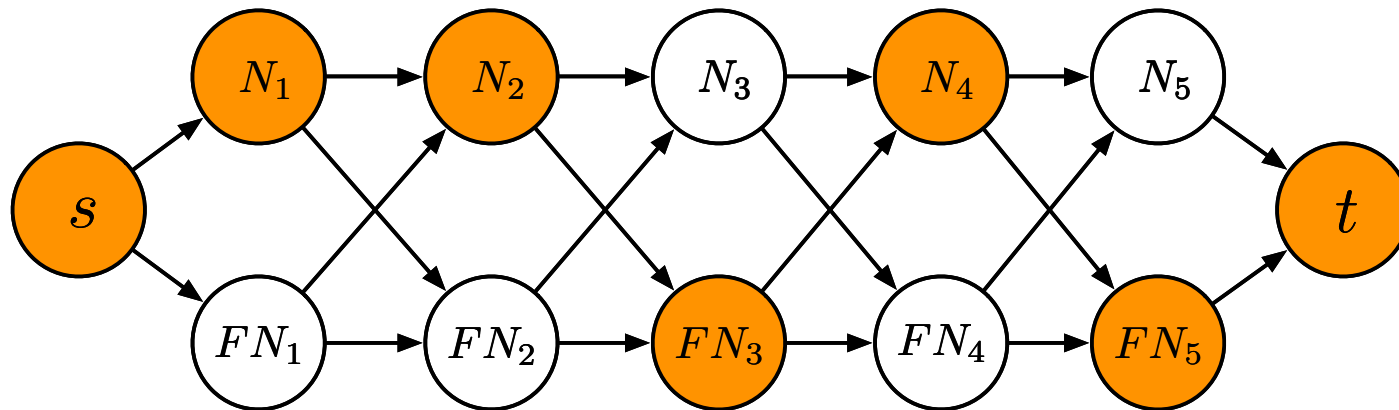
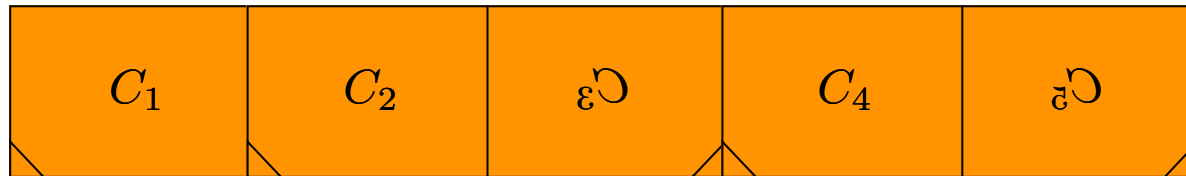
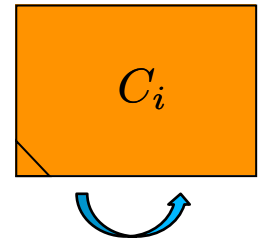
- Single Row Placement
 - An example for cell flipping
 - Edge cost determined by stitches and conflicts (BLUT)



No flip



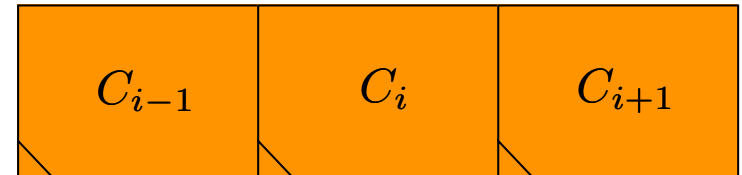
C_i flipped



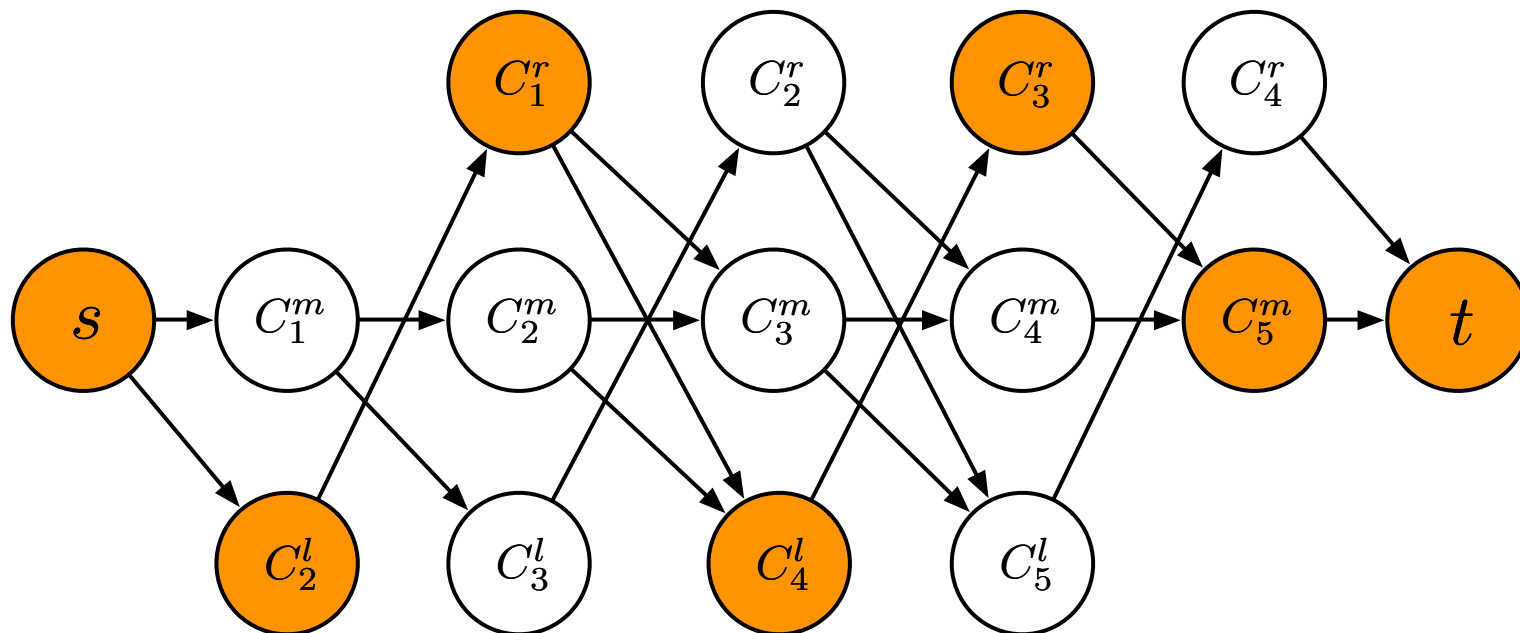
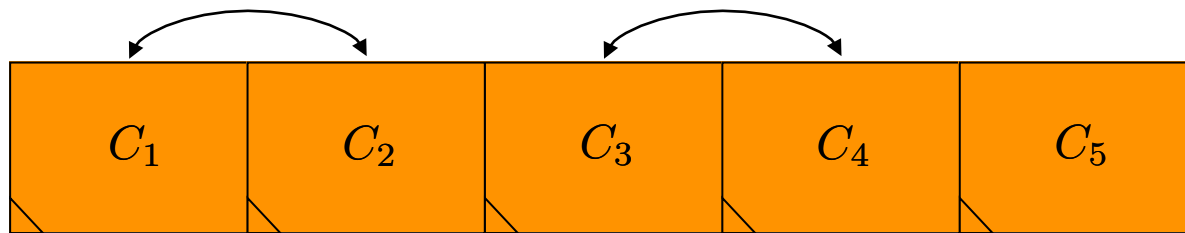
Local Reordering



- Single Row Placement
 - An example for local reordering



C_i^l C_i swaps with C_{i-1} C_i^m No swap for C_i C_i^r C_i swaps with C_{i+1}

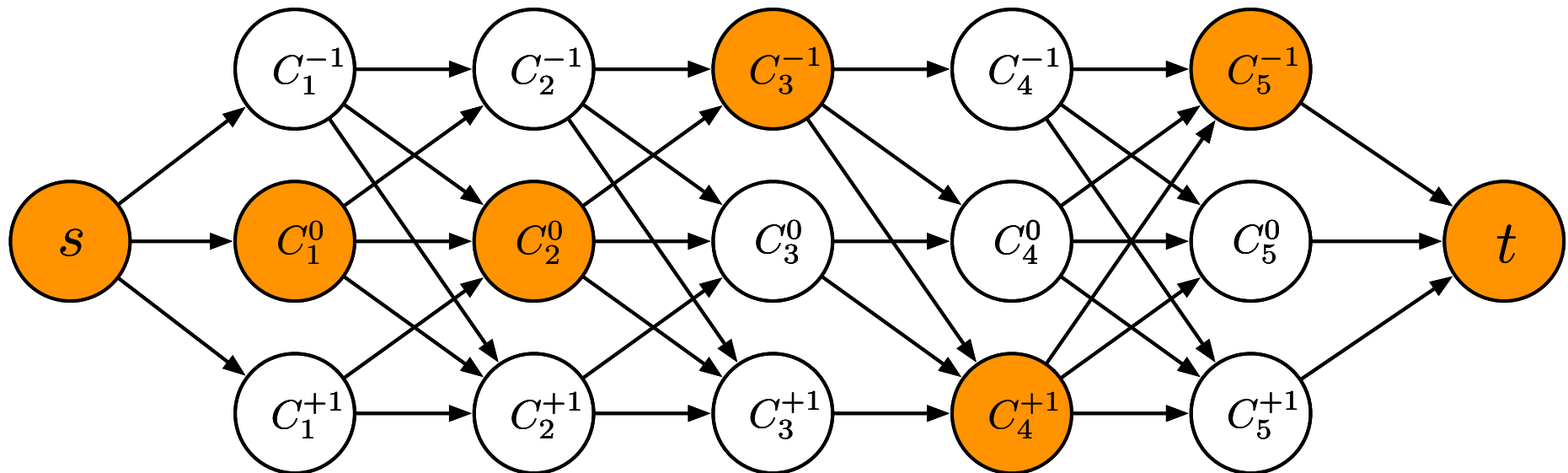
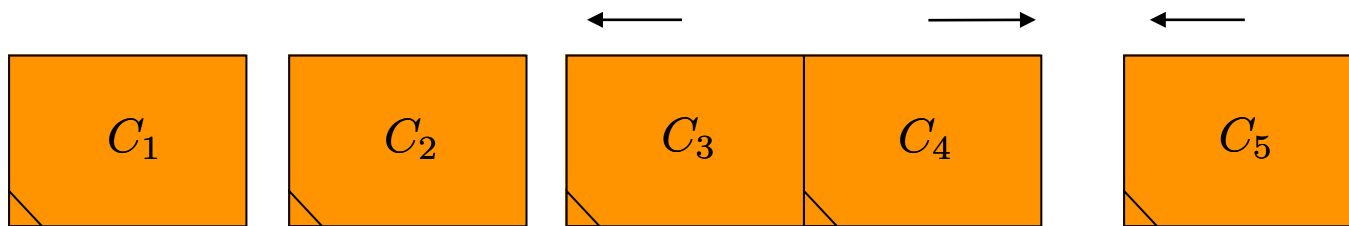
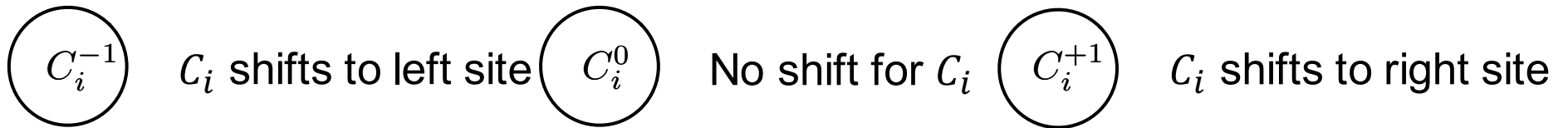
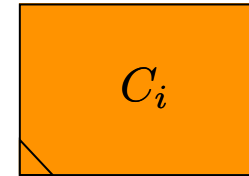


Local Shifting



- Single Row Placement

- An example for local shifting

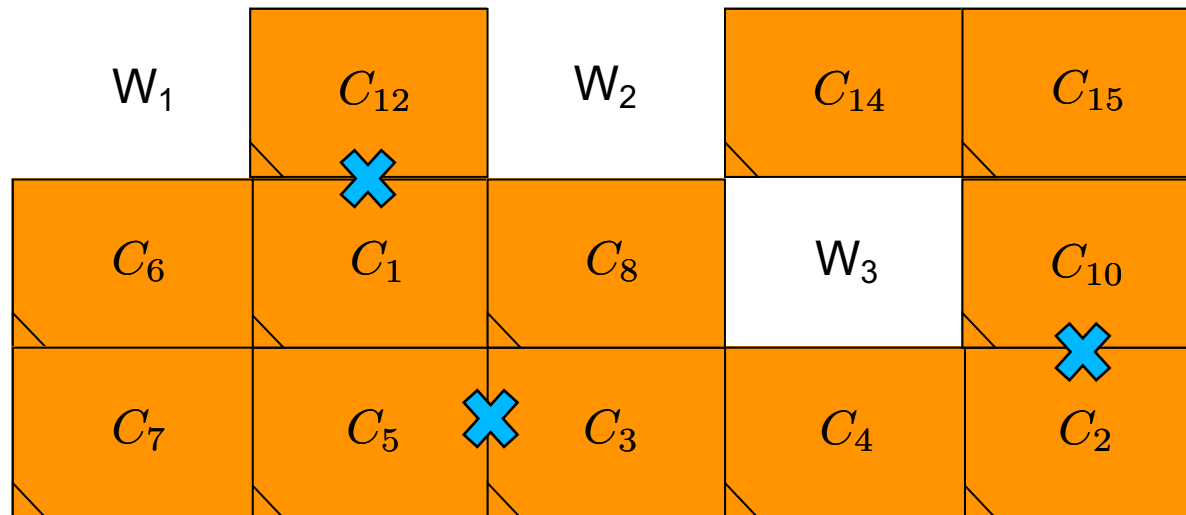


Whitespace Assignment



- Multi-Row placement

- Extract cells with conflicts and re-assign to whitespaces
- Remove conflicts

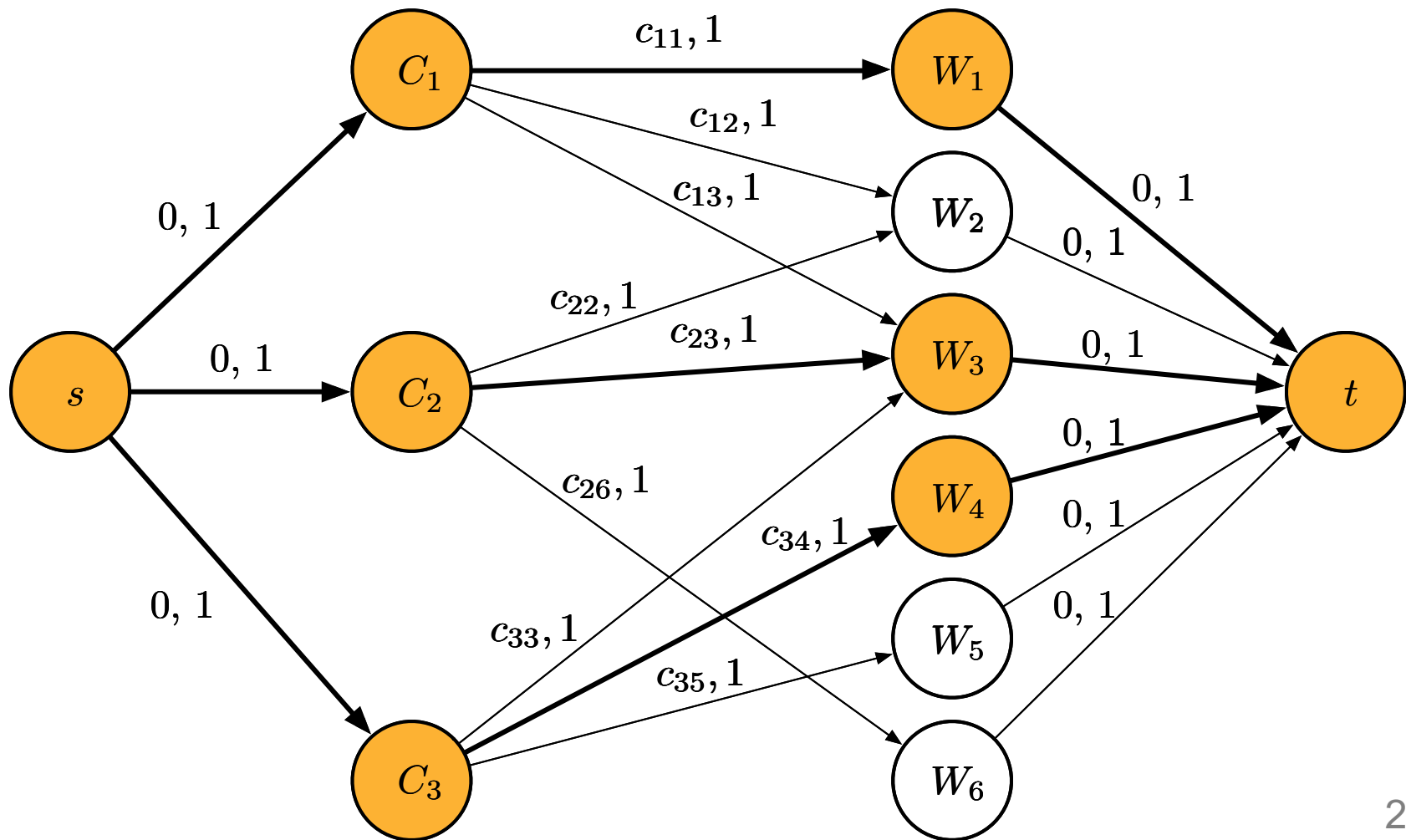


Min-Cost Flow Formulation



- Whitespace assignment

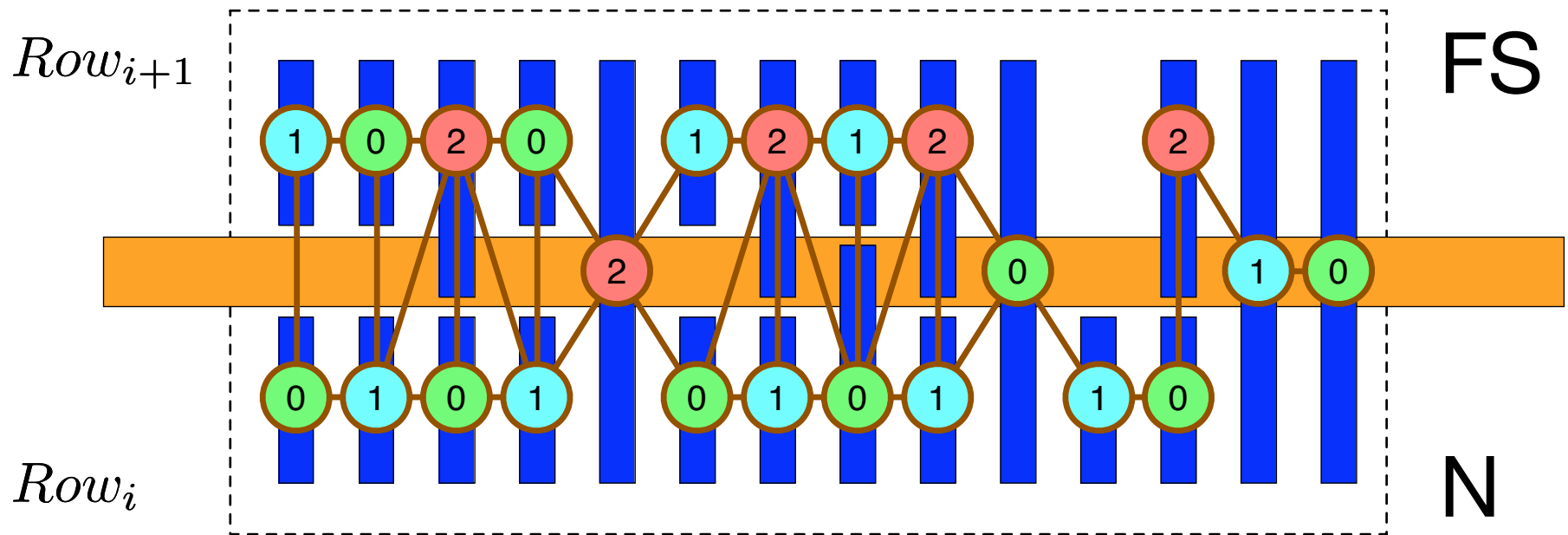
- Min-cost flow formulation
- c_{ij} denotes the cost of assigning cell C_i to whitespace W_j



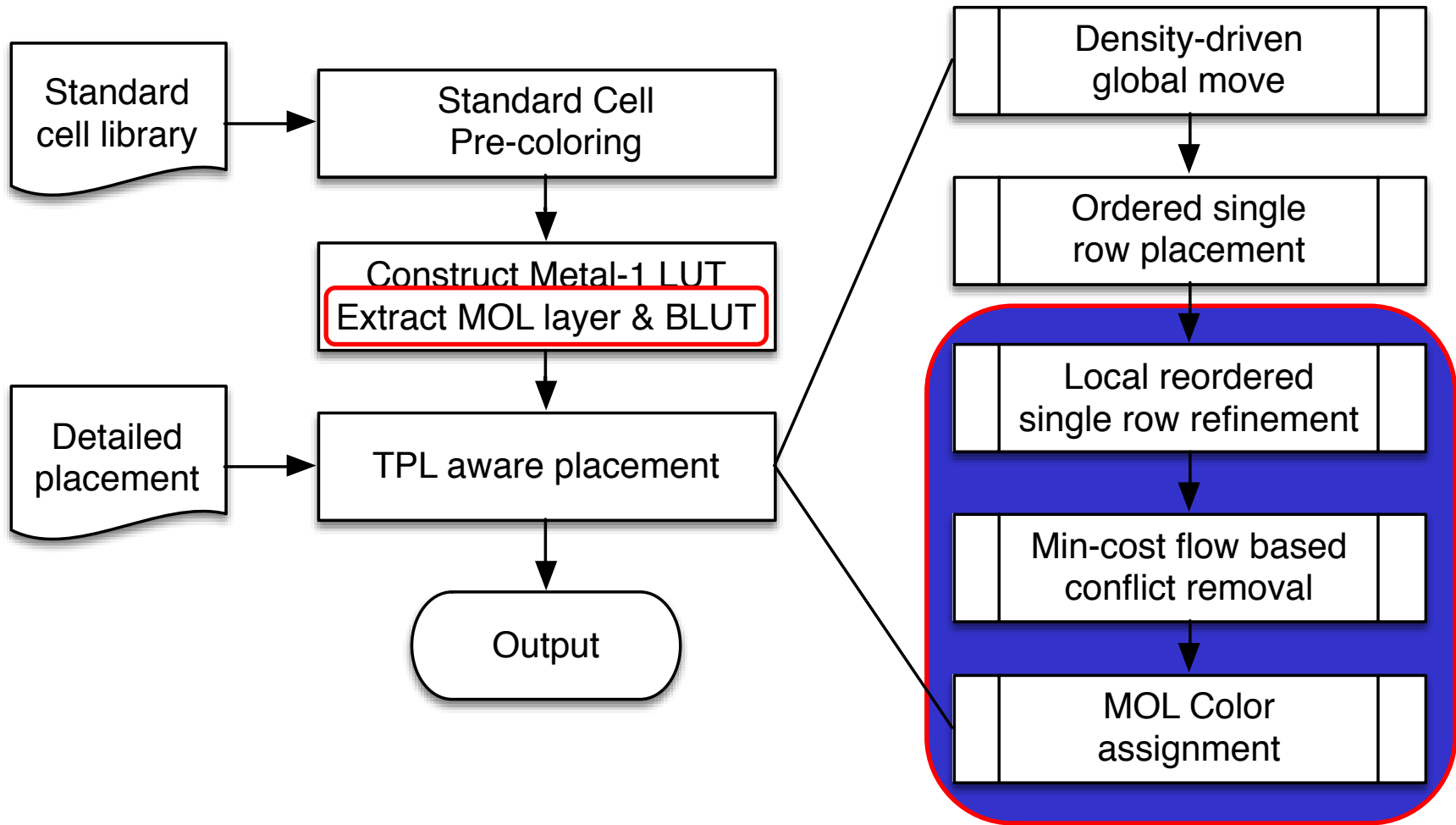
Post Color Assignment for MOL Layer



- Assign colors to MOL Layers
 - Conflict free if no K4 exists due to regularity
 - Scan from left to right in the interaction region between two neighboring rows
 - Linear time



Overall Flow



Experimental Environment Setup

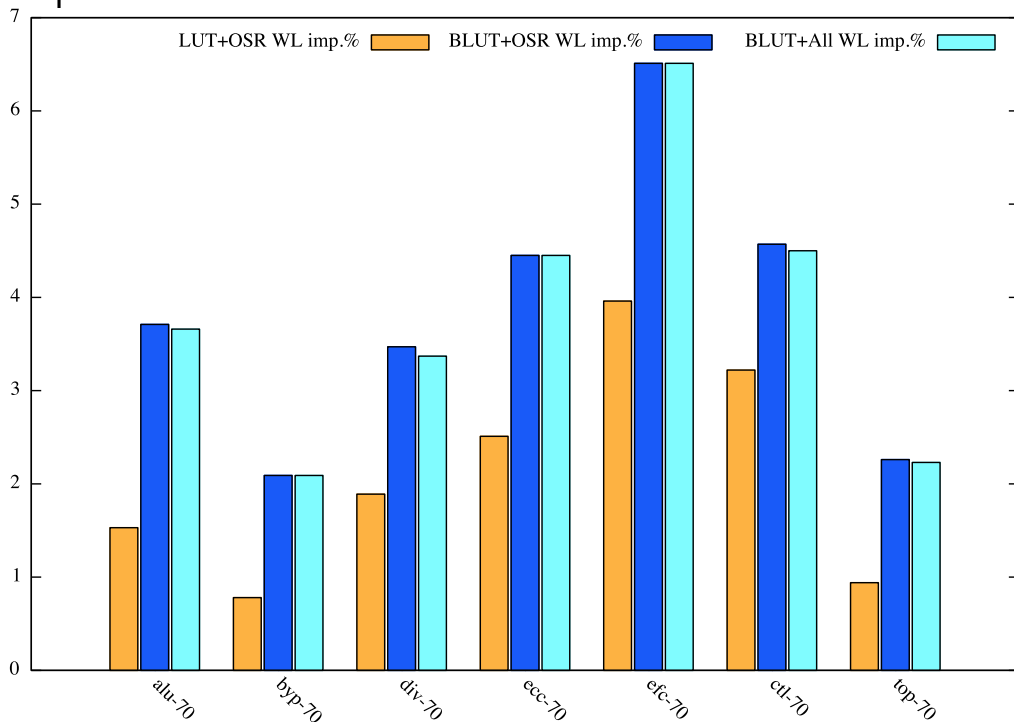


- Implemented in C++
- 8-Core 3.4GHz Linux server
- 32GB RAM
- OpenSparc benchmark
 - Nangate 15nm Standard Cell Library
 - Synthesis: Synopsys Design Compiler
 - Initial placement tool: Cadence Encounter

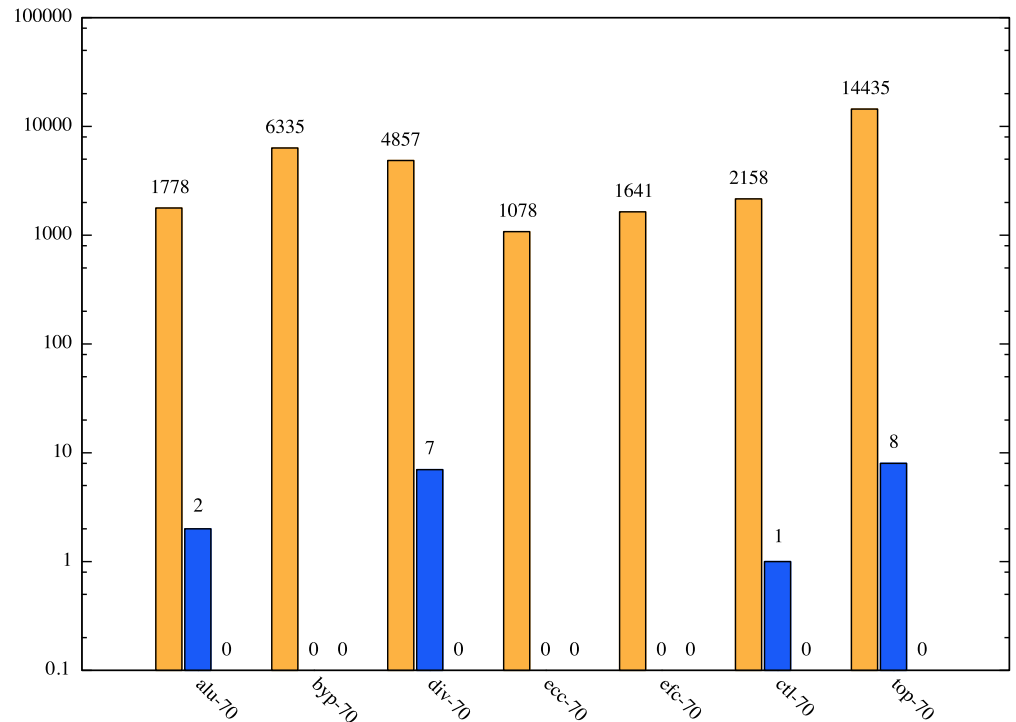
Experimental Results on Sparse Benchmarks



Wirelength Improvement



Final Conflicts



Baseline : LUT+OSR (Ordered Single Row Placement) [Yu+, TCAD'15]

Algorithm 1: BLUT+OSR

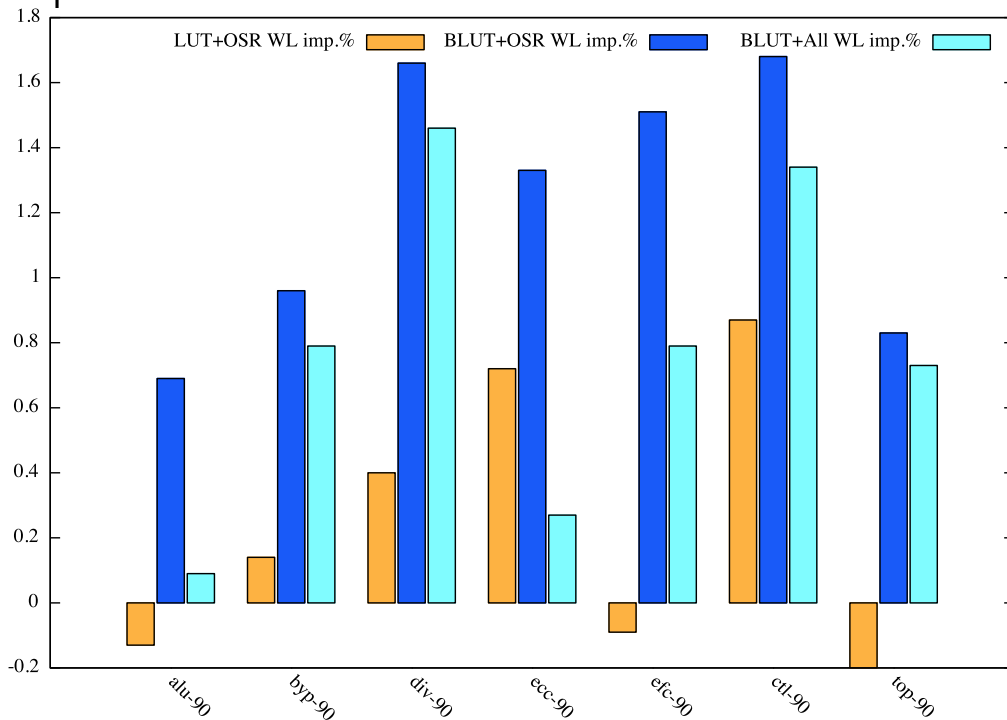
Algorithm 2: BLUT+All

Benchmarks with 70% utilization

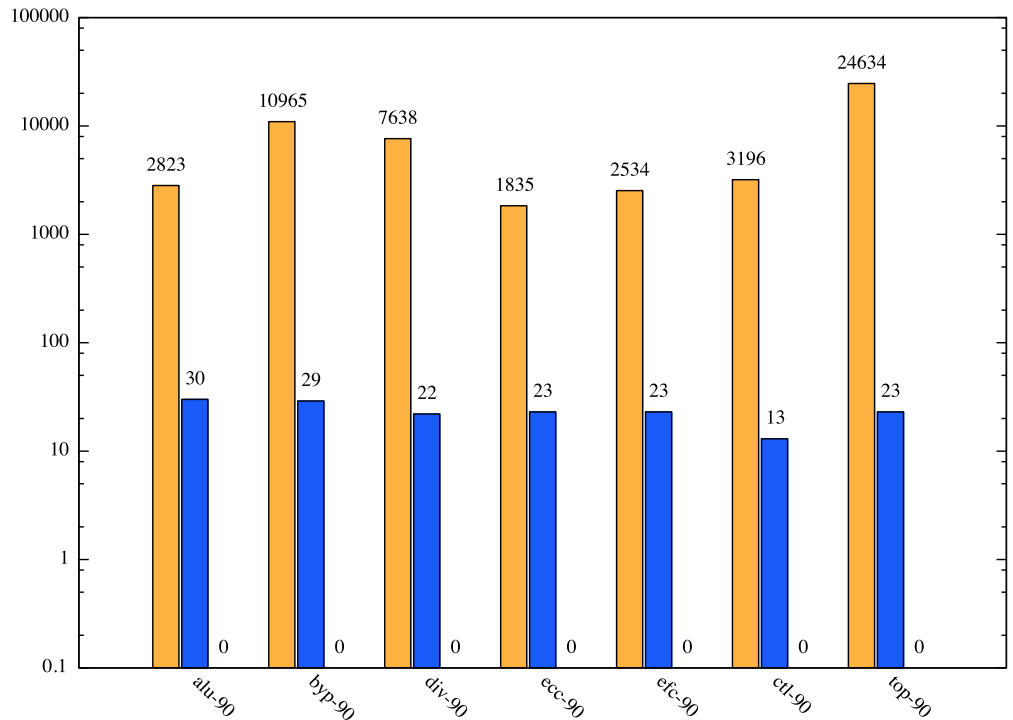
Experimental Results on Dense Benchmarks



Wirelength Improvement



Final Conflicts



Baseline : LUT+OSR (Ordered Single Row Placement) [Yu+, TCAD'15]

Algorithm 1: BLUT+OSR

Algorithm 2: BLUT+All

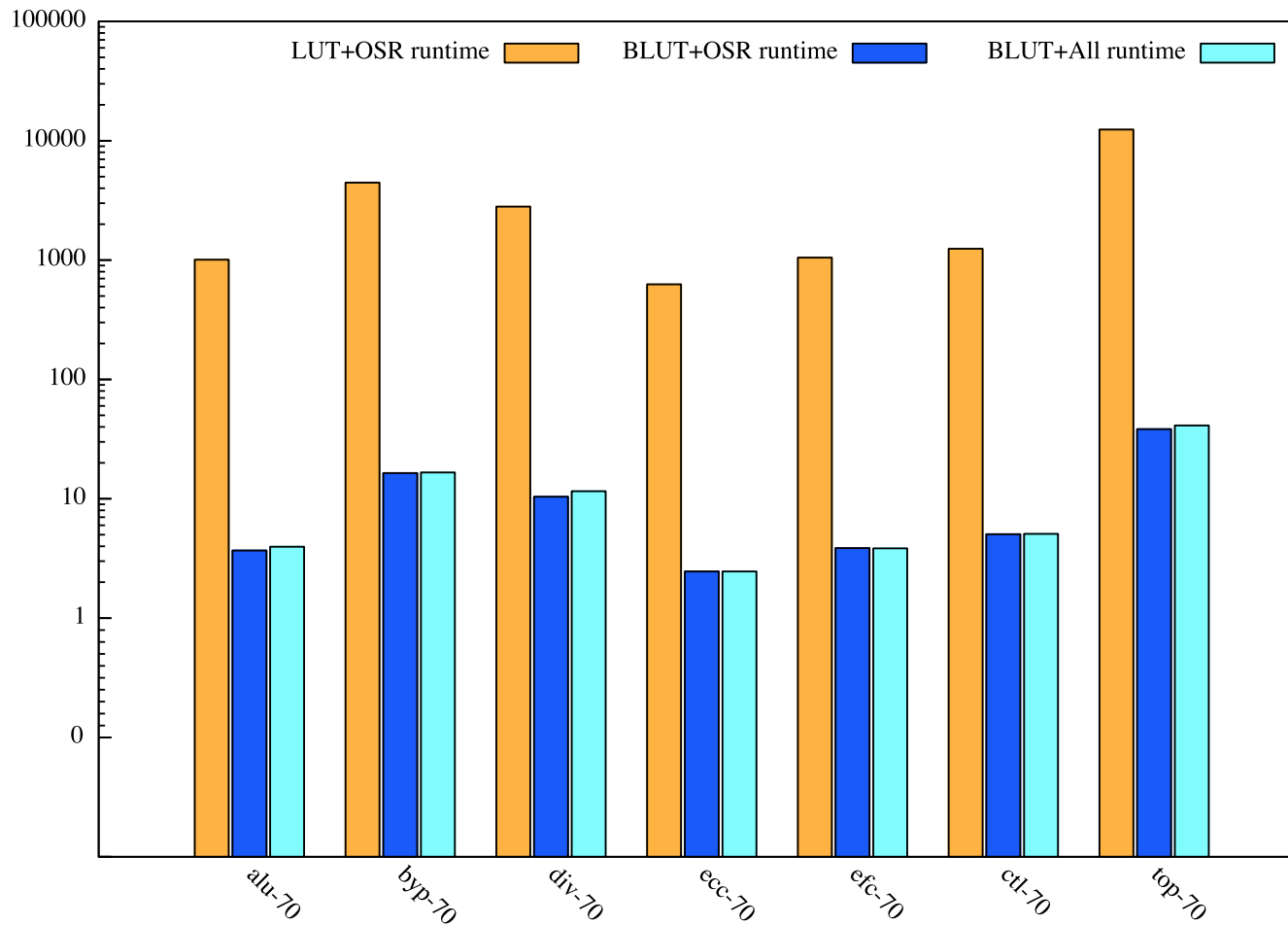
Benchmarks with 90% utilization

Runtime Comparison



- Algorithm 1&2: 100x faster than Baseline

Runtime (s)



Conclusion



- Methodology to handle cross-row MOL conflicts during physical design flow
- A placement framework to optimize both Metal-1 and MOL layer conflicts along with traditional objective
- Better TPL friendliness
- Future work
 - Consider quadruple patterning lithography (QPL) impacts



Thanks