Cohesive Techniques for Cell Layout Optimization Supporting 2D Metal-1 Routing Completion

Kyeongrok Jo, Seyong Ahn, Taewhan Kim and Kyumyung Choi

School of Electrical and Computer Engineering

Seoul National University, Korea

SNUCAD

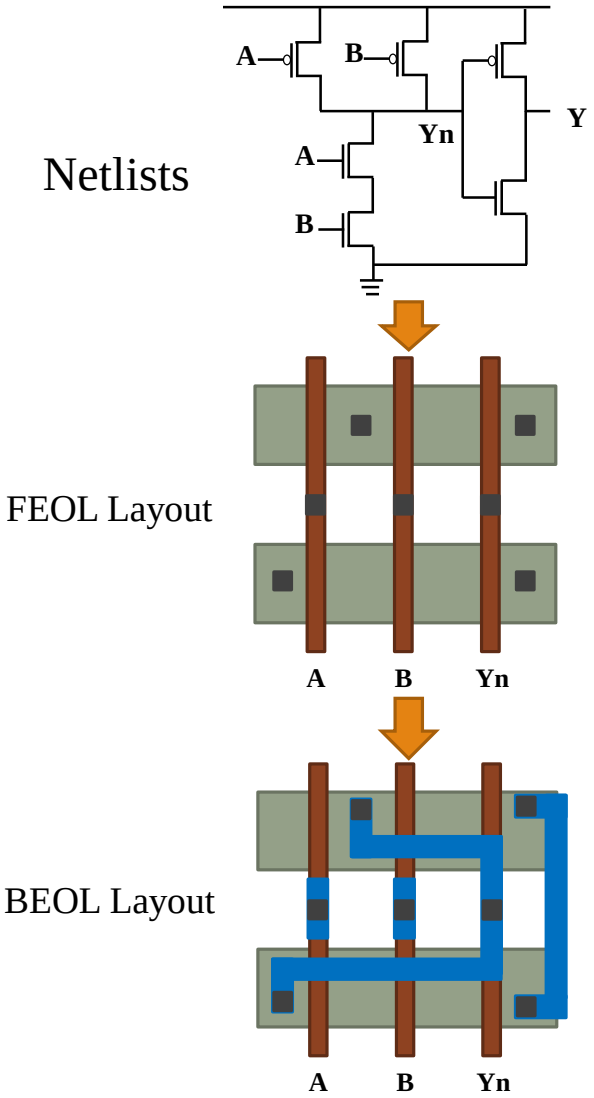
Motivation

- **Till today: Standard cell layout is done manually in industry**
  - Cell topology generation & internal net routing are not tightly coupled
    → Produce inferior layouts wrt. cell size and routing completion

- **Now: Increase the demand for fast and automatic exploration & generation of standard cell layouts**
  - Need a quick evaluation vehicle of DTCO(design-technology co-optimization)
  - Need quick and quality competitive layout generation
    - Multiple sets of library to cope with broad range of product groups
    - Libraries for derivative processes

Cell Layout Generation

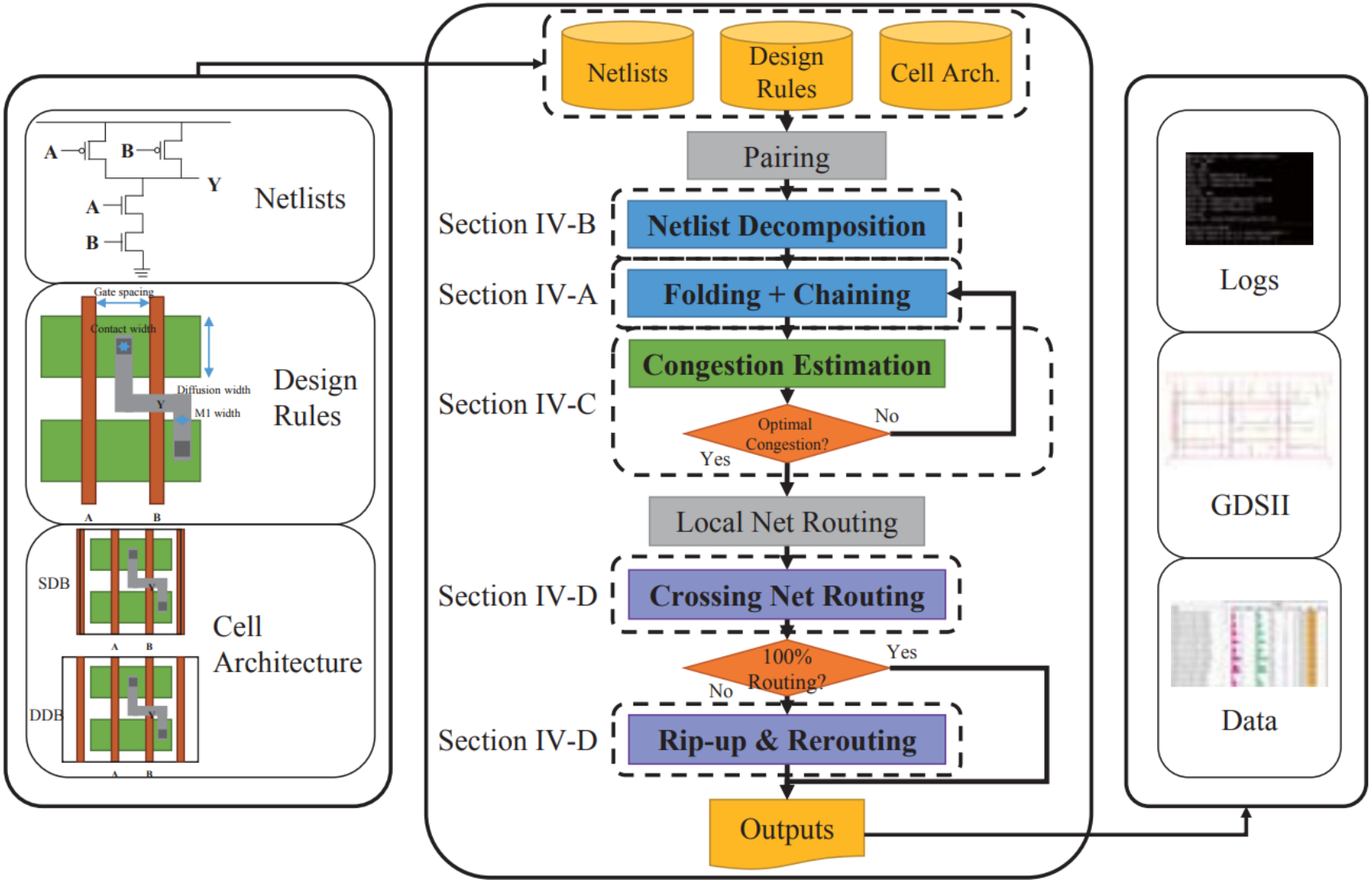
- **Cell topology generation (Front End Of Line layout)**
  - Determine gate poly order
  - Minimize area & enable 100% of internal routing
- **Internal routing (Back End Of Line layout)**
  - Assume 2D metal-1 routing
  - Route internal nets
  - Minimizing M2 usage



Related Works

- **Design Rule Evaluator** [1]
  - Evaluate design rules and cell layout simultaneously
  - BEOL layout is not generated
- **Standard cell routing via SAT** [4]
  - Find legal routing among candidate routes by SAT
  - No consideration of FEOL to enhance routability
  - Run time and memory explosion
- **BonnCell: Automatic layout of leaf cells** [5]
  - Generate cell layouts through recursive enumeration
  - Internal net routing using mixed-integer programming
  - No consideration of correlation between FEOL & BEOL

Proposed Flow of Cell Layout Generation



Cohesive Technologies for Cell Layout Optimization

Newly devised following methodologies

1. Chaining combined with folding

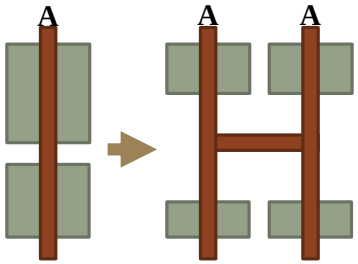
2. Netlist decomposition

3. Gate poly ordering combined with routing congestion estimation

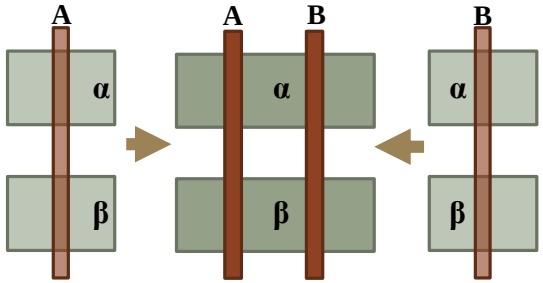
4. 2D routing with minimal resource

1. Chaining Combined with Folding

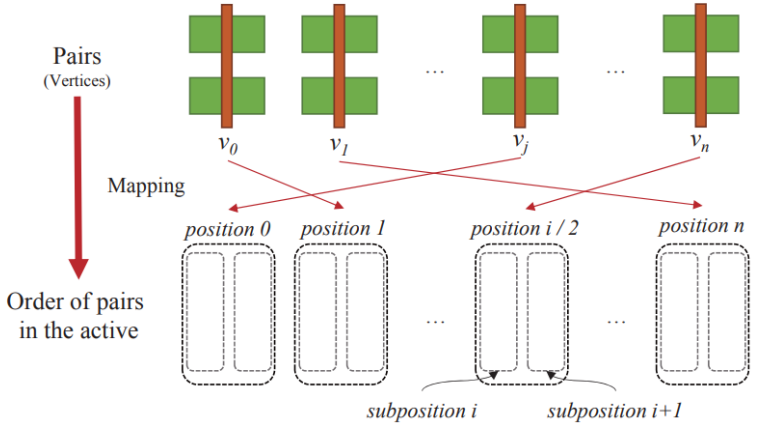
- **Folding**
  - Splitting a transistor with large width into multiple smaller ones in parallel connection



- **Chaining**
  - Abutting transistor pairs by sharing the same active area

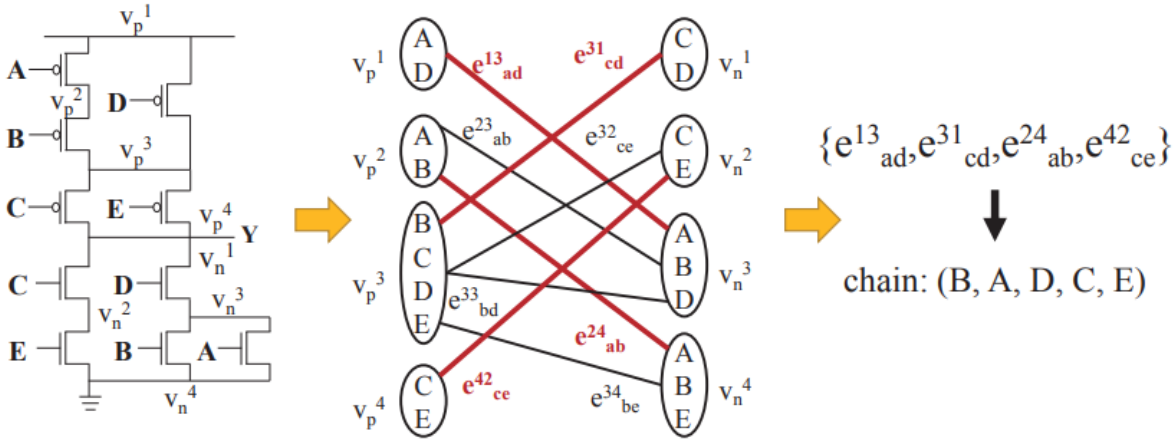


- **Problem formulation**
  - Given: transistor pairs including folded ones
  - Objective: abut the pairs as a chain



1. Chaining Combined with Folding

- **Previous method [6]**
  - Fold all transistors larger than maximum width
  - Netlist → Bipartite graph → Transistor chain



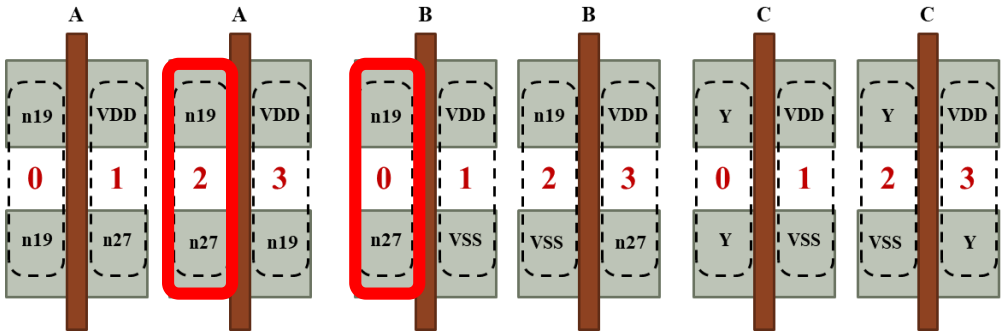
  - Significant runtime sacrifices
    - Edges of bipartite graph increase significantly when transistors are folded
      → excessive exploration of redundant folded transistors
    - ≥12 hours for cells having more than 20 folds

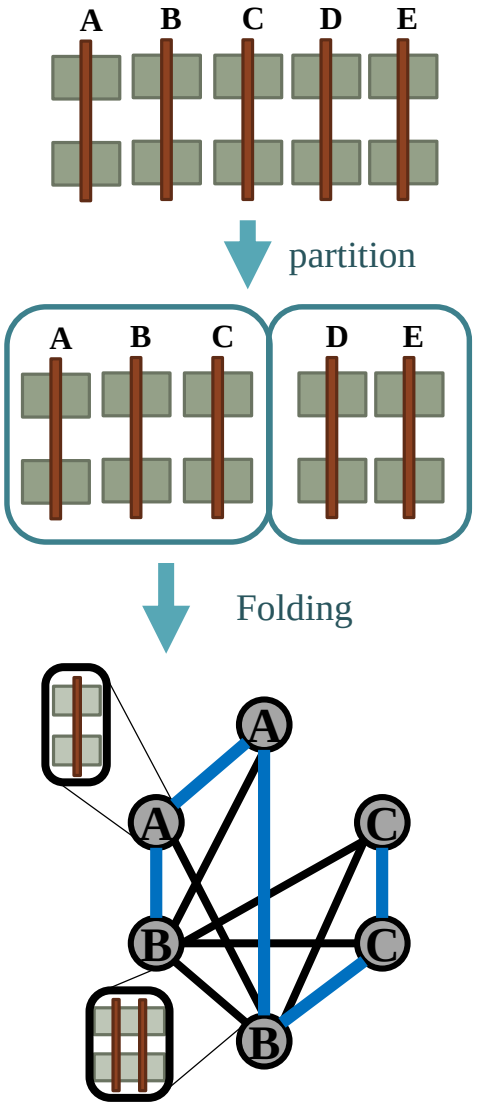
1. Chaining Combined with Folding

- **Proposed method**

1. Partition transistor pairs to be abutted into a single chain
   - Use previous *Bipartite graph method*[6]
   - Consider transistor pairs as unfolded

2. Derive *transistor abutment graph*
   - Vertex: transistor pair including folded one
   - Edge: possibility of abutment between tr. Pairs
   - Type: source/drain combination of a pair





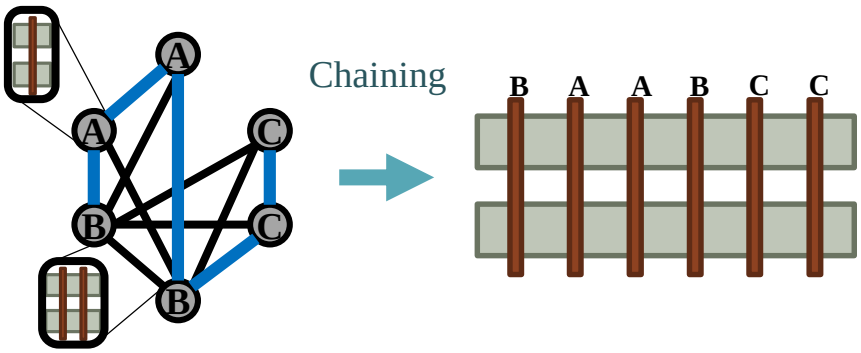
1. Chaining Combined with Folding

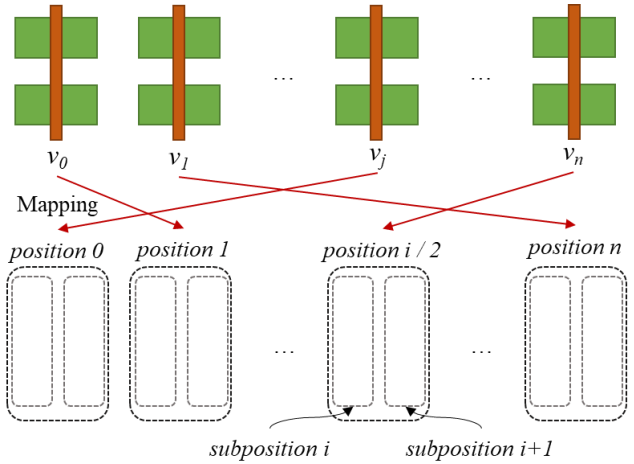
3. Formulate as Hamiltonian path problem

- *Hamiltonian path* represents a chain with abutted transistors

4. Solve the problem with SAT formulation

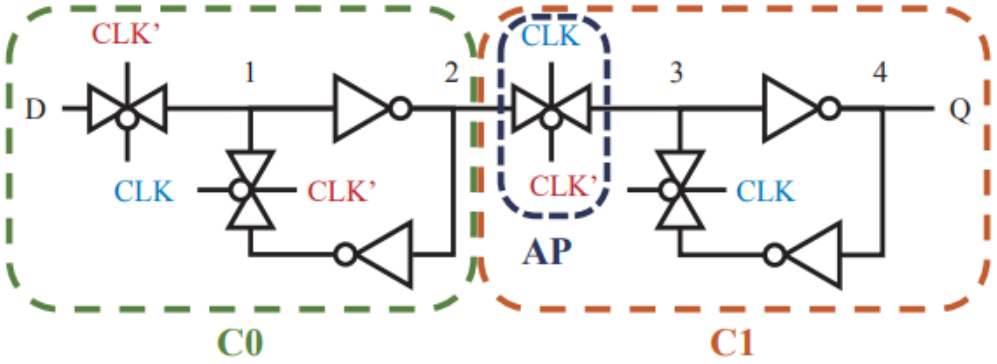
- Integration of gate poly ordering and abutment constraints
- SAT solver reduce run time
- Generate multiple gate poly ordering candidates
  → higher possibility of internal routing completion





2. Netlist Decomposition

- **Combine theoretical knowledge and designer's experience**

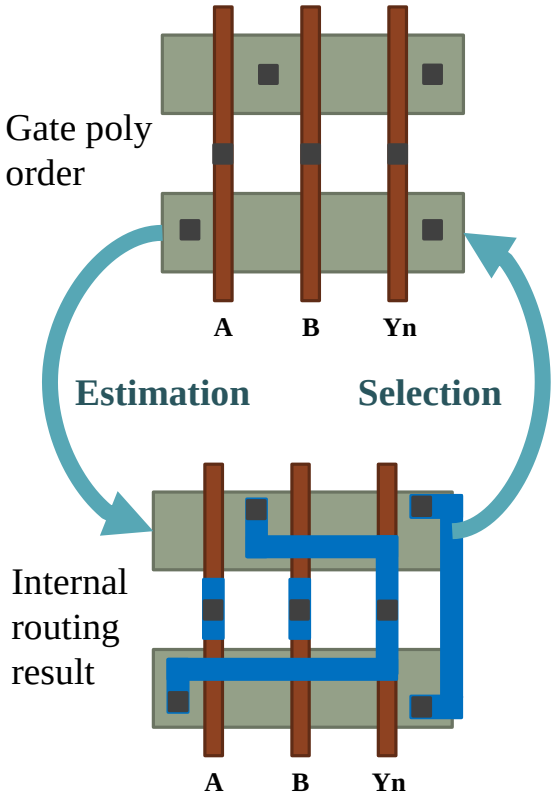


- **Partition a netlist if multiple data flow feedback loops exist**

- **Results**
  - Trade-off between area optimization and internal routing completion
  - Proof of effectiveness by experiments
    → increased routing completion for sequential logic cells (flip-flops)

3. Gate Poly Ordering Combined with Routing Congestion Estimation

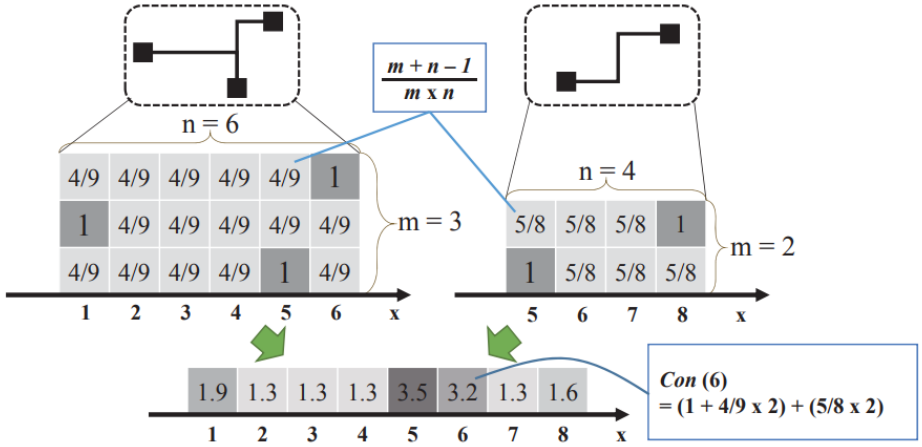
- **Correlation of gate poly order and internal routing result**
  - Not fully considered in previous related works
- **Ordering gate polys to have higher possibility of routing completion**
- **Estimate routing congestion for every instance of FEOL layout**

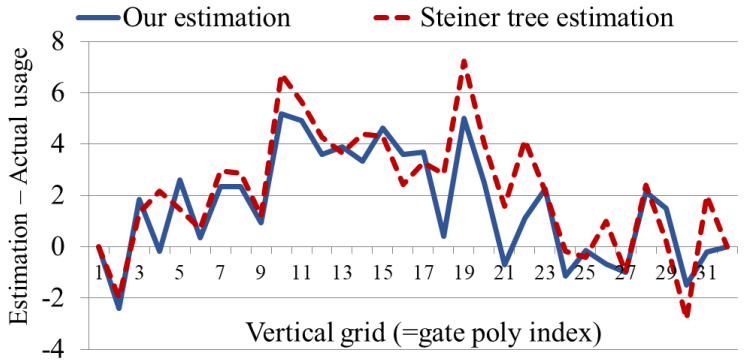


- **Find ordering instances with less routing congestion**

3. Gate Poly Ordering Combined with Routing Congestion Estimation

- **Proposed routing congestion estimation**

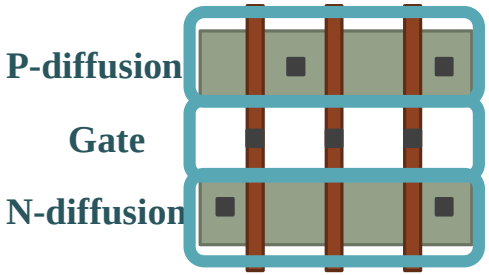




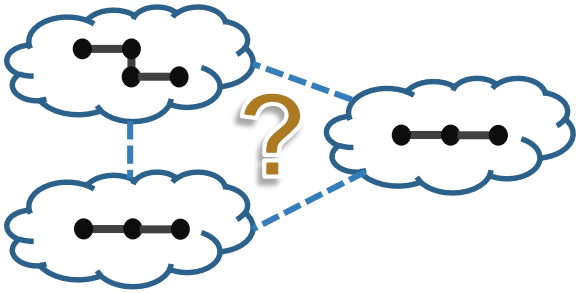
  - Estimation for each net
    - Bounding box of all pins
    - Possibility of grid occupation
      - Contact: 1
      - Otherwise: (Half-perimeter / # of bins)
  - More precise estimation compared to STST (Single Trunk Steiner Tree) method
    - Actual vertical track usage vs. estimated usage

4. 2D Routing with Minimal Resource

- **Routing Strategy**
  - Local nets
    - Connection of contacts belongs to a single region
  - Crossing nets
    - Connection of contacts in between multiple regions
  - Local nets → Crossing nets

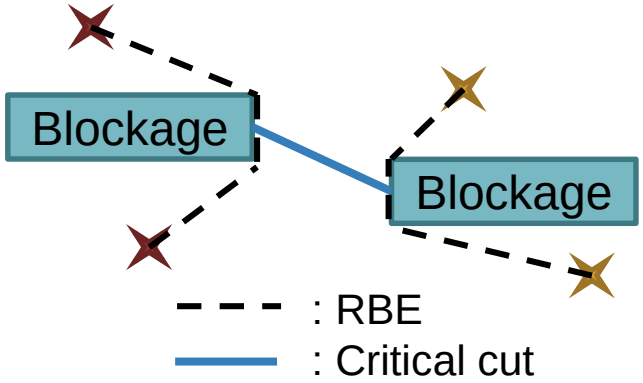


- **Issues on crossing net routings**
  - Planarity
    - Possibility of routing to be done in a single plane
  - Sequentiality
    - Effect of routing order on routing completion

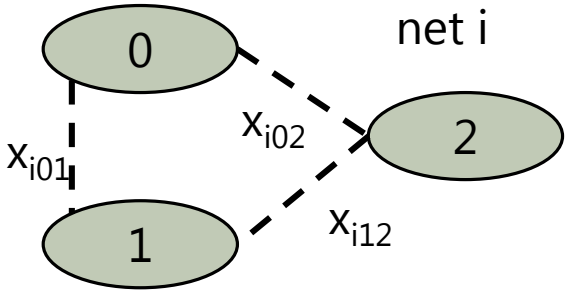


4. 2D Routing with Minimal Resource

- **Rubber Band Equivalent (RBE)**
  - Shortest path connecting two points, detouring blockages
- **Critical Cut**
  - A shortest line between two blockages' corner or tip
- **Planarity Check**
  - All critical cuts → *flow* ≤ *capacity*



- **Solved with an ILP**
  - Minimized routing length with
    - Sub nets should be connected
    - Planarity must be satisfied



Experimental Results

- **PDK & Spice netlist**
  - 45nm FreePDK & NanGate standard cell library
  - 28nm Industry partner's PDK & cell library
  - Net specification of 28nm cell library

	#Cells	#Nets	#Nets per cell
Comb. logic	48	354	4-12
Flip-flops	8	151	14-24
Total	56	505	4-24

- **Results**
  - Cell generation
    - **100%** routing completion with 9 & 11 tracks
  - Run time:
    - About **1 hour** for 28nm library (56 representative cells)
    - $\leq$ **3 hours** for NanGate cell library (all cells)

Experimental Results

- **Effectiveness of our techniques**
  - Netlist decomposition
    → Routing completion increase(M1: **22%p**, total: **17%p**) with run time reduction
  - Chaining combined with folding
    → 83% run time reduction, with **7%p** total routing completion increase
  - New congestion estimation method
    → Ensure 100% routing completion
  - Cross routing with planarity
    → Ensure 100% routing completion

Techniques	Netlist	Routing completion		Run time (sec.)
		M1	M1+M2	TG / Rout' / Total
All-applied	Combs.	96%	100%	930 / 50 / 980
	FFs	70%	100%	683 / 267 / 950
	Total	89%	100%	1613 / 317 / 1930
Netlist Decomp. (x)	Combs.	-	-	- / - / -
	FFs	48%	83%	1124 / 163 / 1287
Chaining + Folding (x)	Combs.	85%	89%	10394 / 36 / 10430
	FFs	70%	100%	687 / 261 / 948
	Total	81%	93%	11081 / 297 / 11378
New Congestion Estimation (x)	Combs.	97%	100%	933 / 111 / 1044
	FFs	61%	91%	687 / 261 / 948
	Total	86%	97%	1522 / 266 / 1788
Planar Cross Routing (x)	Combs.	95%	99%	932 / 47 / 979
	FFs	70%	100%	683 / 260 / 943
	Total	88%	99%	1615 / 307 / 1922

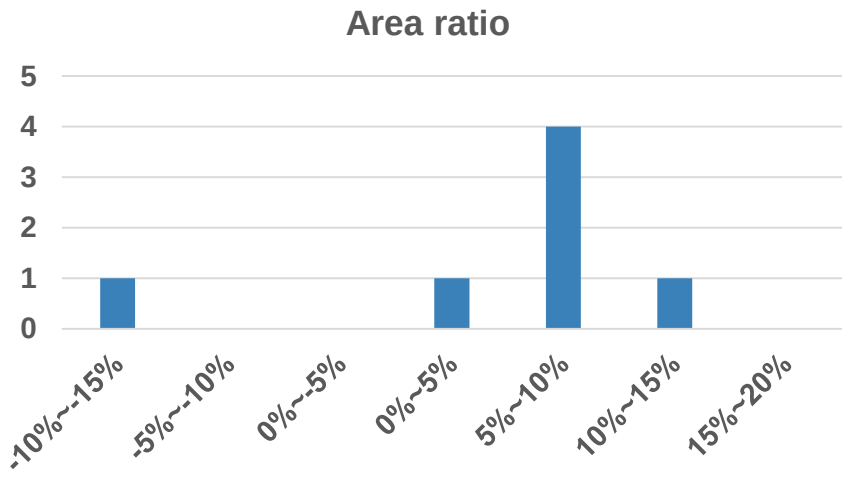
Experimental Results

- **Evaluation of layout quality**
  - Comparison with industry partner's library developed manually

- **Layout area**
  - Flip-flop area is larger than that of manual by **5~10%**
  - In digital block, comb. Cell (60%) + flip-flops (40%)
    → Area overhead in digital block = **2~4%**

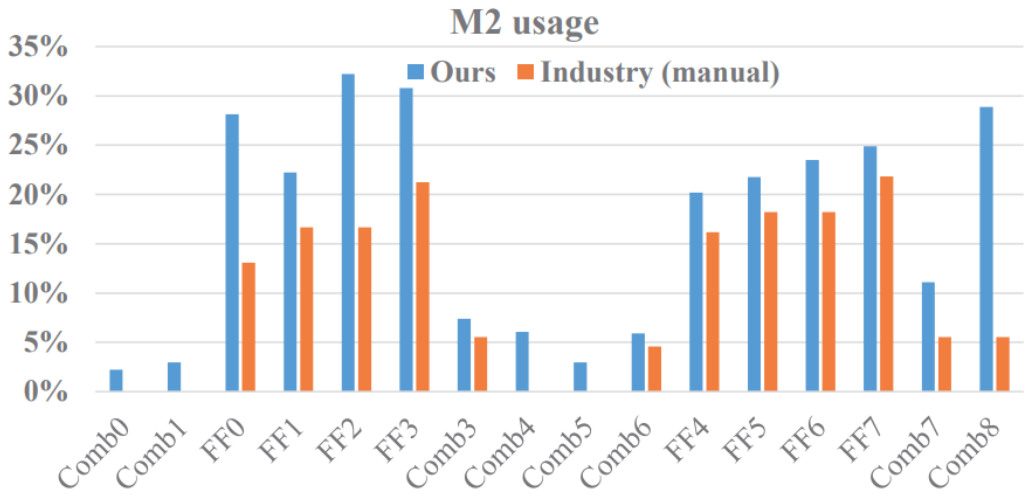
Cell type	Cell count	Area ratio
Comb. logic	48	0%
Flip-flops	8	5~10%

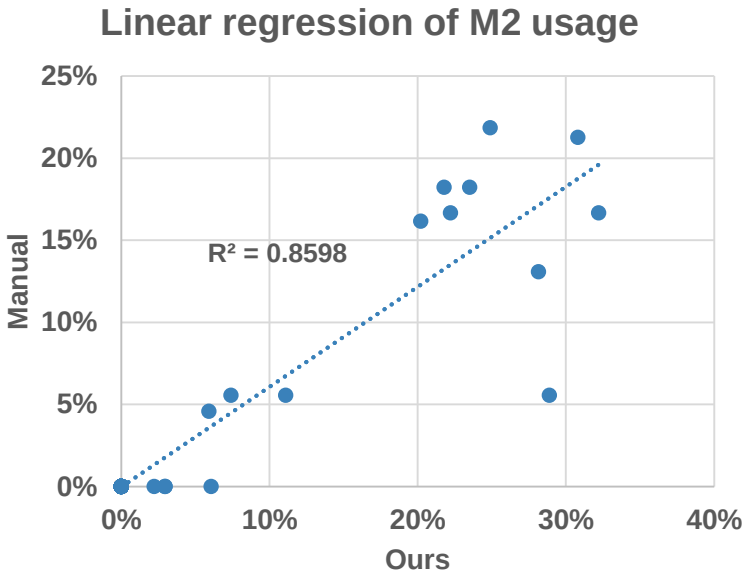
$$Area\ Ratio = \frac{Ours - Industry\ partner's}{Industry\ partner's}$$



Experimental Results

- **M2 Usage**





- Occupy **7%p** more M2 resource on average
- Quickly generated (several days *vs.* within an hour)
- Positive correlation with manual

Conclusions & Ongoing works

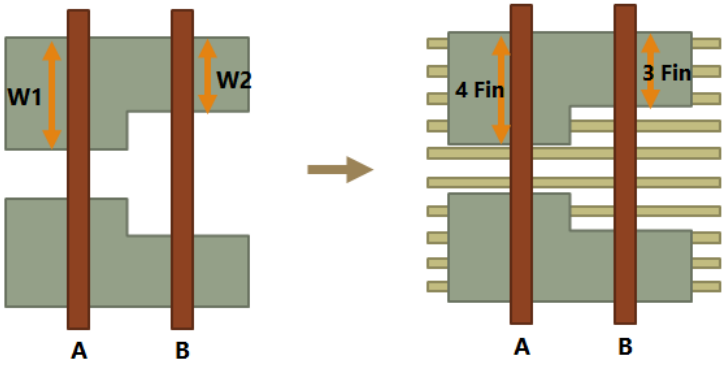
- **Full automation for generating cell layouts**
  - Cohesive techniques are effective to improve quality of the cell generator
  - Comparable to the industry partner's manual layouts with a extremely short time

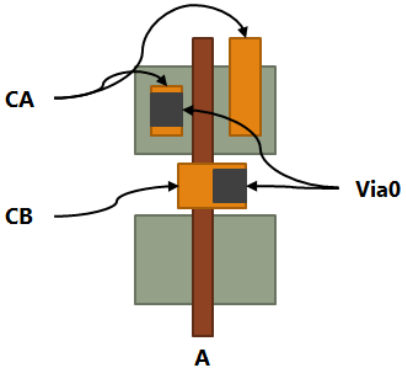
  - Can be used as a cornerstone for migrating from planar technology to 3D FinFET technology
  - Can be used valuably as core engine in DTCO

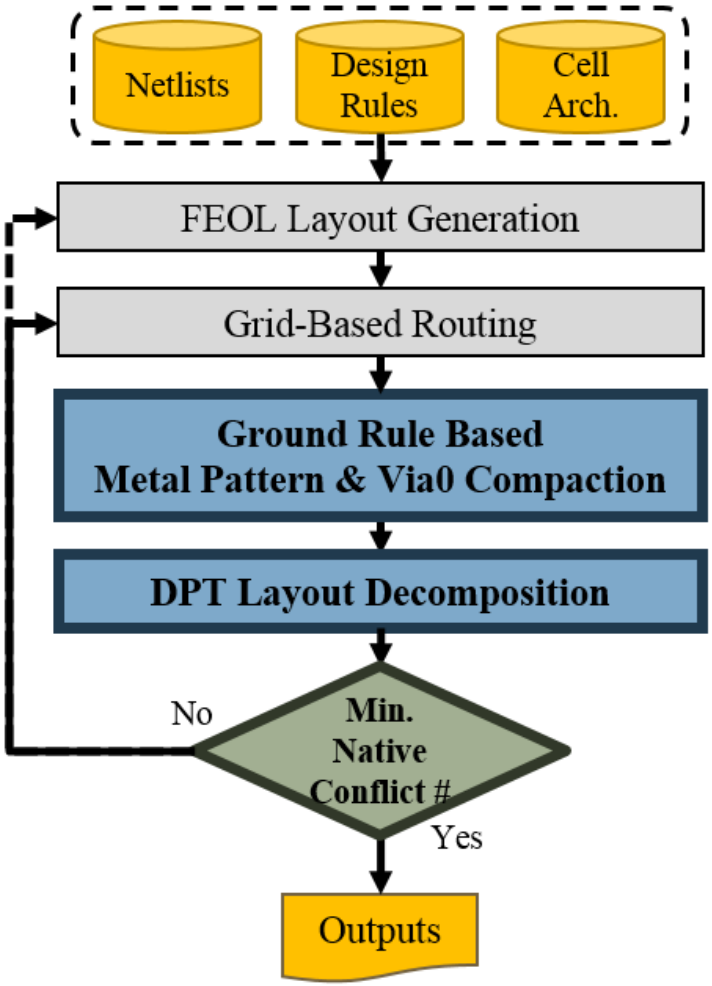
} Ongoing works

Ongoing works

- **Adaptation for FinFET technology**
  - Discrete Fin number & MEOL
  - Increased design rule complexity
  - DPT (Double patterning) is needed

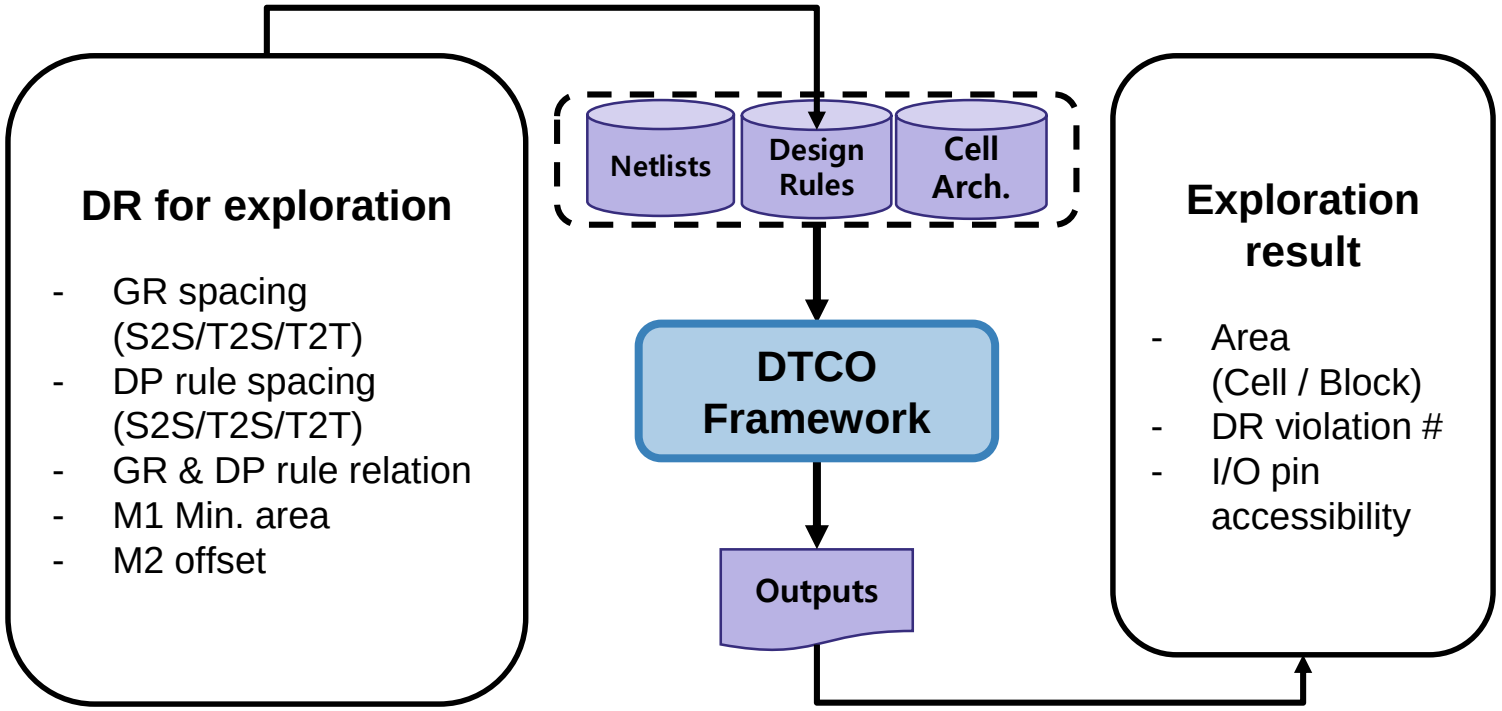






Ongoing works

- **DTCO (Design Technology Co-Optimization) Framework**



- Can be used as exploration tool of design rule and cell architecture
- Enable co-optimization of design & process technology

Thank you!

Q & A