Nanosheet FET standard cell library synthesis with pin access optimization on M0 layer

Meng-Yu Shih, Ting-Xin Lin, Yih-Lang Li
Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

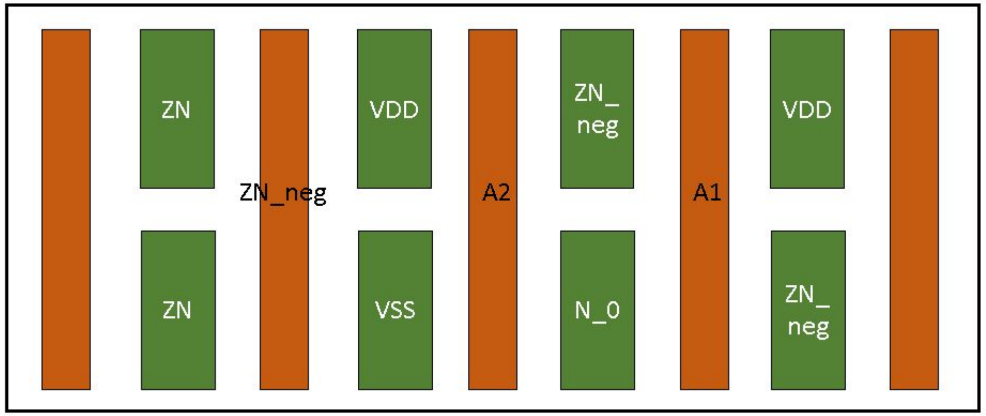
International Symposium on Physical Design 2026

Outline

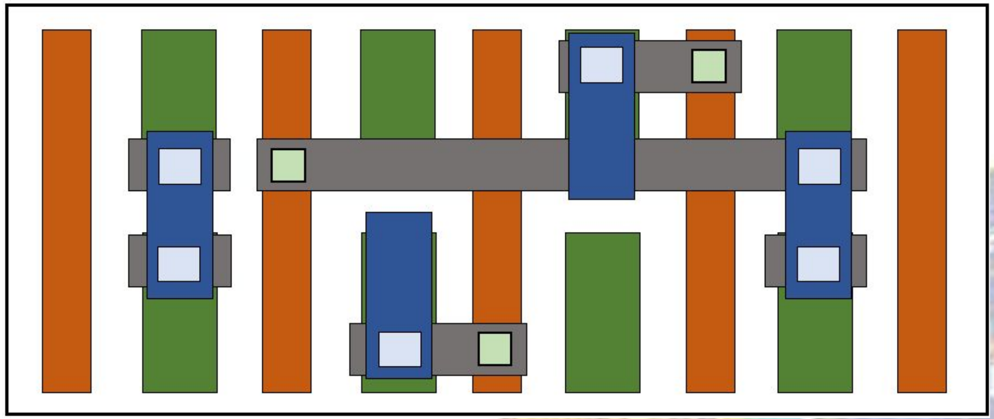
- Introduction
  - Standard cell synthesis
  - Problems to solve in nanosheet FET cell synthesis
- Preliminaries
- Synthesis Flow
- Algorithm
- Experimental Results
- Conclusions

Standard cell synthesis

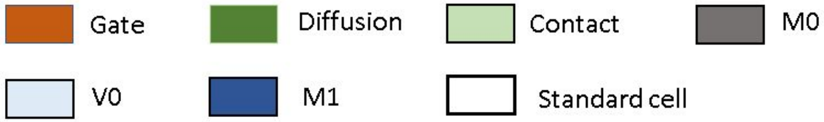
- Physically arrange transistors and connect signals with non-overlapping wires to form a logic gate.
- Algorithm
  - Placement
  - Routing



placement

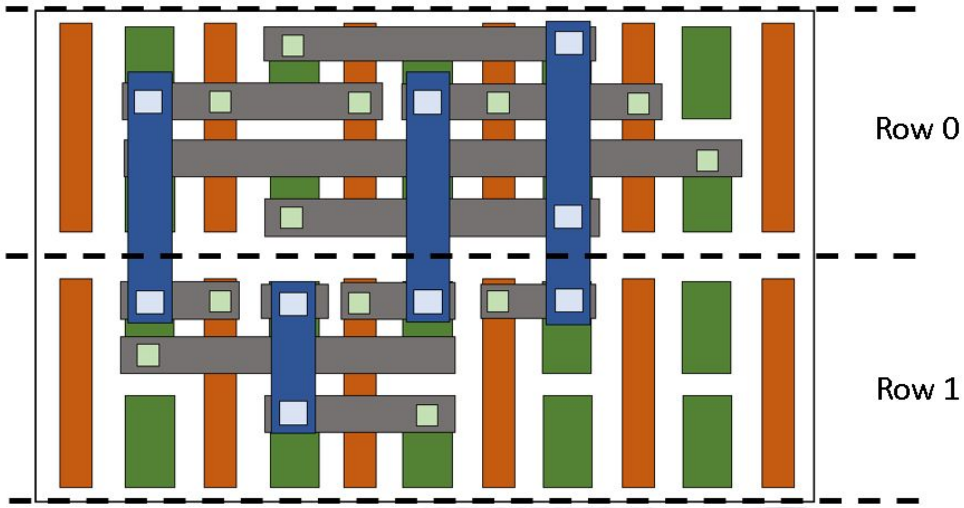


routing



Problems to solve in nanosheet FET cell synthesis

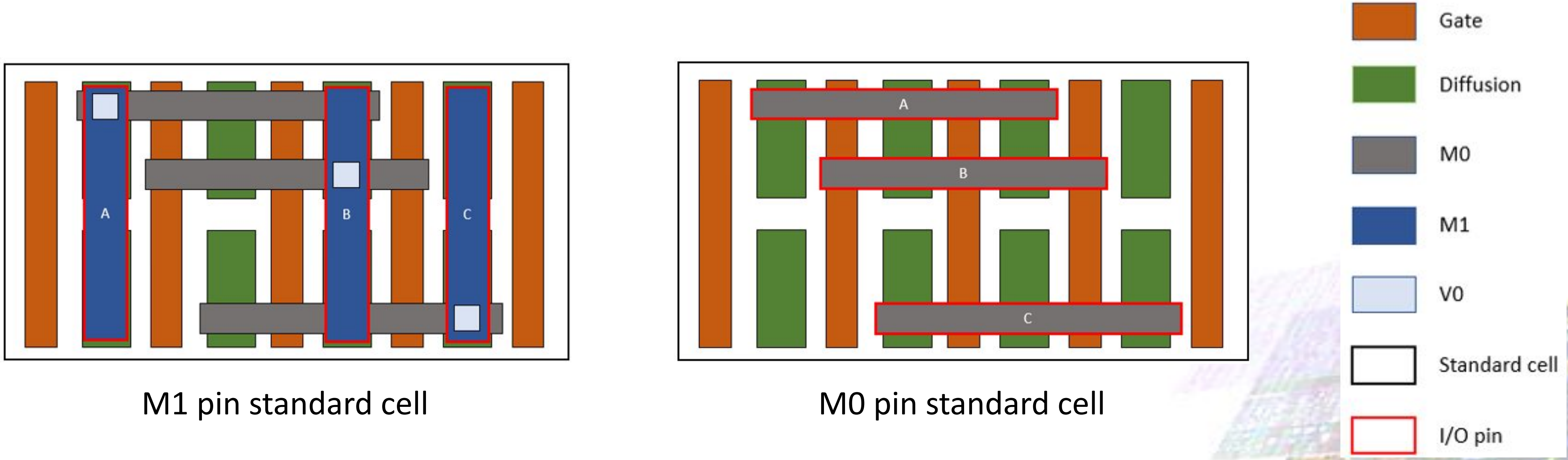
- Horizontal routing track reduction.
- Metal 0 (M0) / Metal 1 (M1) can be used in block-level routing.
  - Routing grids need to be redesigned.
- Metal 2 (M2) track coordinates in a row are unknown during cell synthesis.
  - M2 usage is forbidden.
- A cell has to be multi-row if it is unroutable in the single row structure.



Double-row standard cell

Problems to solve in nanosheet FET cell synthesis

- IO pins could be allocated on both M0 and M1 layers.
  - Currently, there is no algorithm to allocate IO pins on M0 layer.



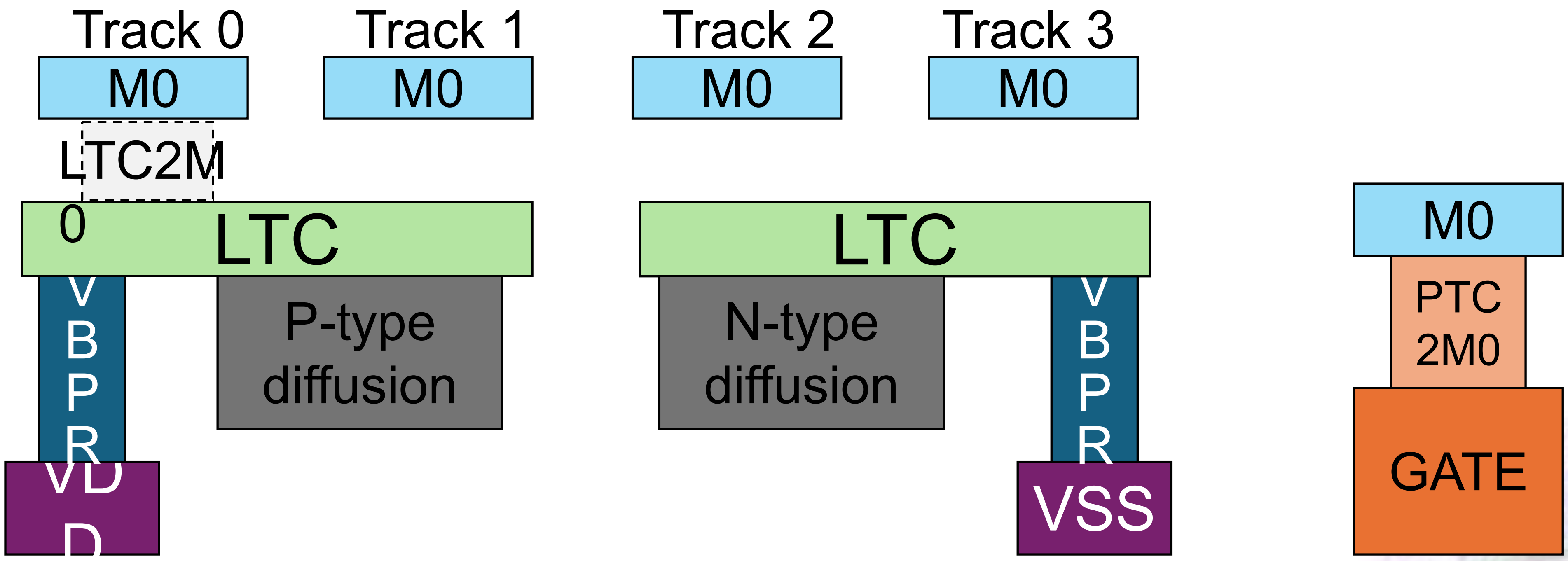
M1 pin standard cell

M0 pin standard cell

Outline

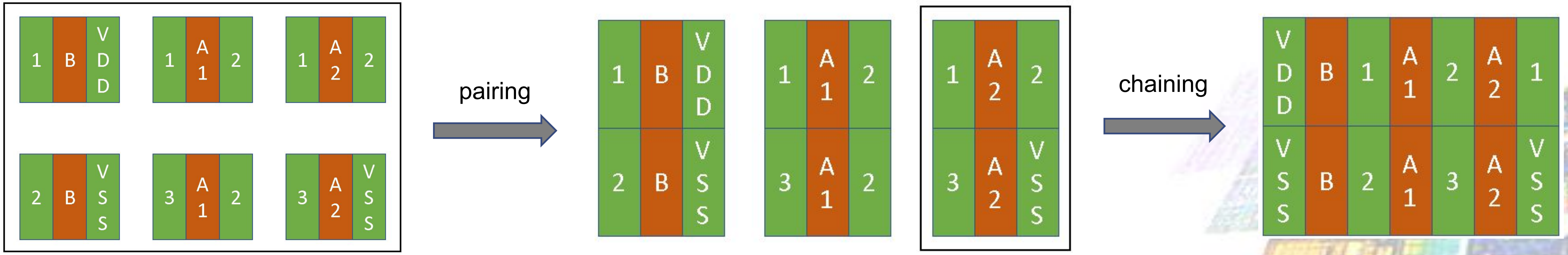
- Introduction
- Preliminaries
    - Nanosheet FET Structure
    - Placement
    - Routing
- Synthesis Flow
- Algorithm
- Experimental Results
- Conclusions

Nanosheet FET Structure



Placement

- Single-row placement (Paper: “LiB: a CMOS cell compiler”)
  1. Cluster devices using common signals in P and N-type devices
  2. Pair P/N-type devices with same gate signals.
  3. Chain transistor pairs to form partial placement.
  4. Permute chains to find optimized final placement.



P / N transistor cluster   P / N transistor pair   placement

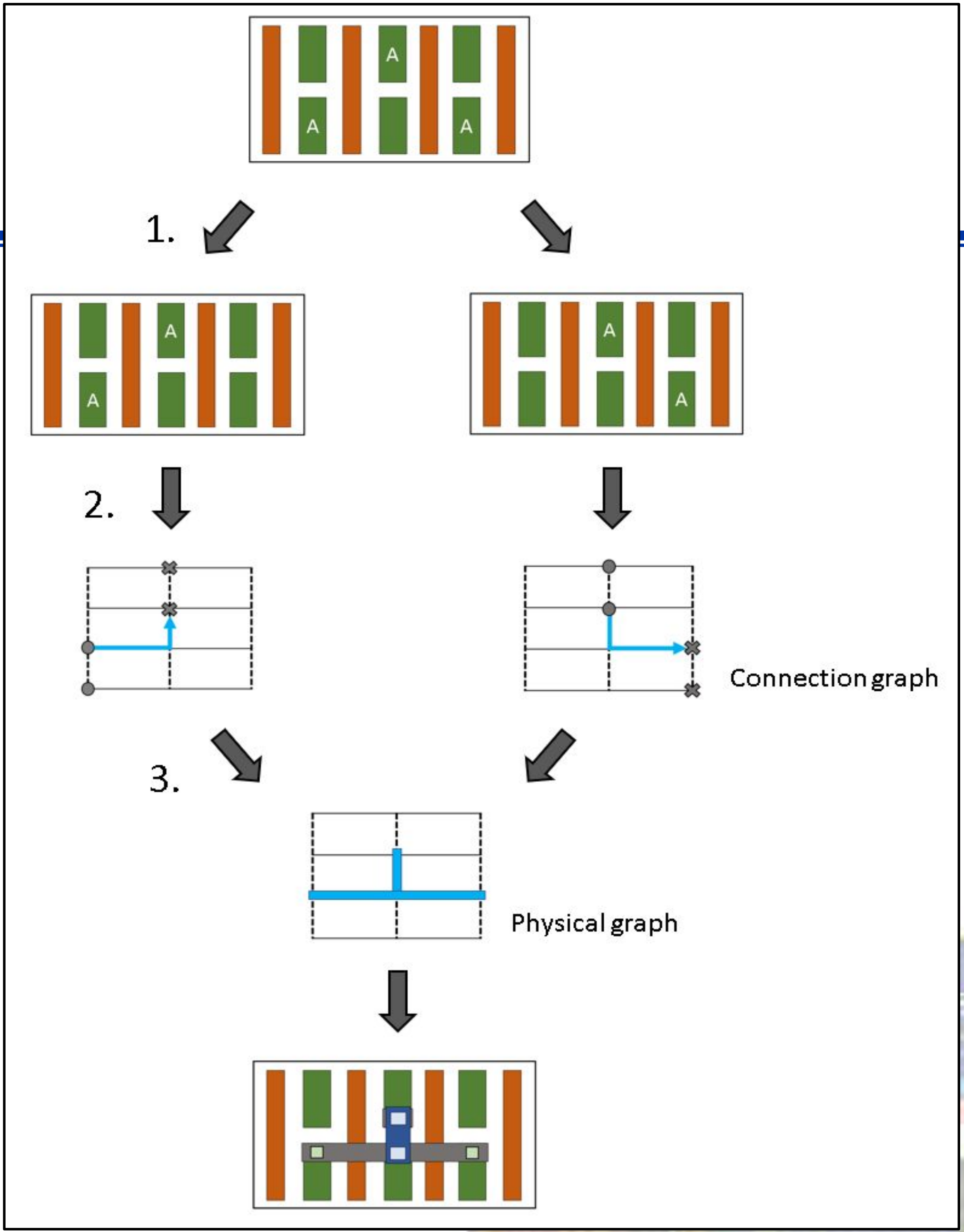
Placement

- Multi-row placement
  - Allocate transistors into N rows using Fiduccia–Mattheyses (FM) algorithm
  - Perform transistor placement algorithm on each row.
  - Exhaustively combine placements from each row and minimize weighted HPWL.

HPWL: Half Perimeter Wire Length

Routing

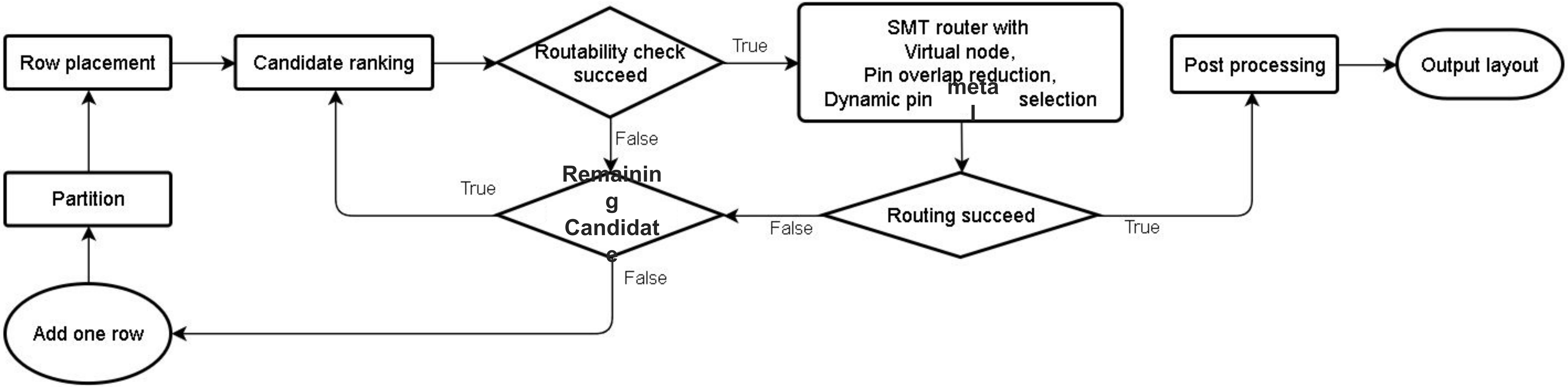
- Paper
  - Scalable CFET cell library synthesis with a DRC-aware lookup table to optimize valid pin access
- Method
  1. Decompose each multi-pin net into two-pin nets.
  2. Route every two-pin net with SMT in connection graphs.
  3. Results of connection graphs will be copied to the physical graph concurrently during routing.



Outline

- Introduction
- Previous work
- Preliminaries
- Synthesis Flow
- Algorithm
- Experimental Results
- Conclusions

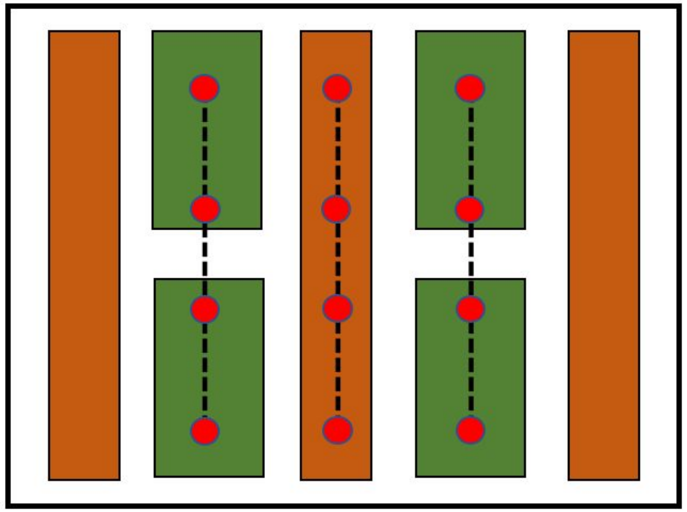
Synthesis Flow



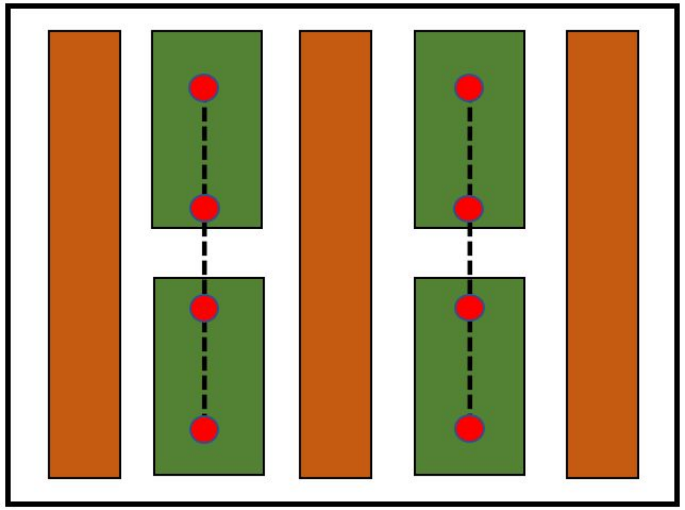
Outline

- Introduction
- Preliminaries
- Synthesis Flow
- Algorithm
  - Grid Graph Formulation
  - Virtual Node
  - Pin Overlap Reduction
- Experimental Results
- Conclusions

Grid Graph Formulation

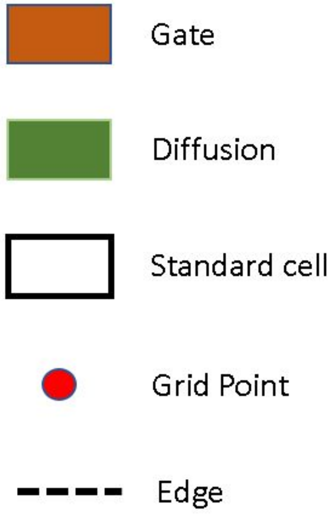


Normal M1 grid graph



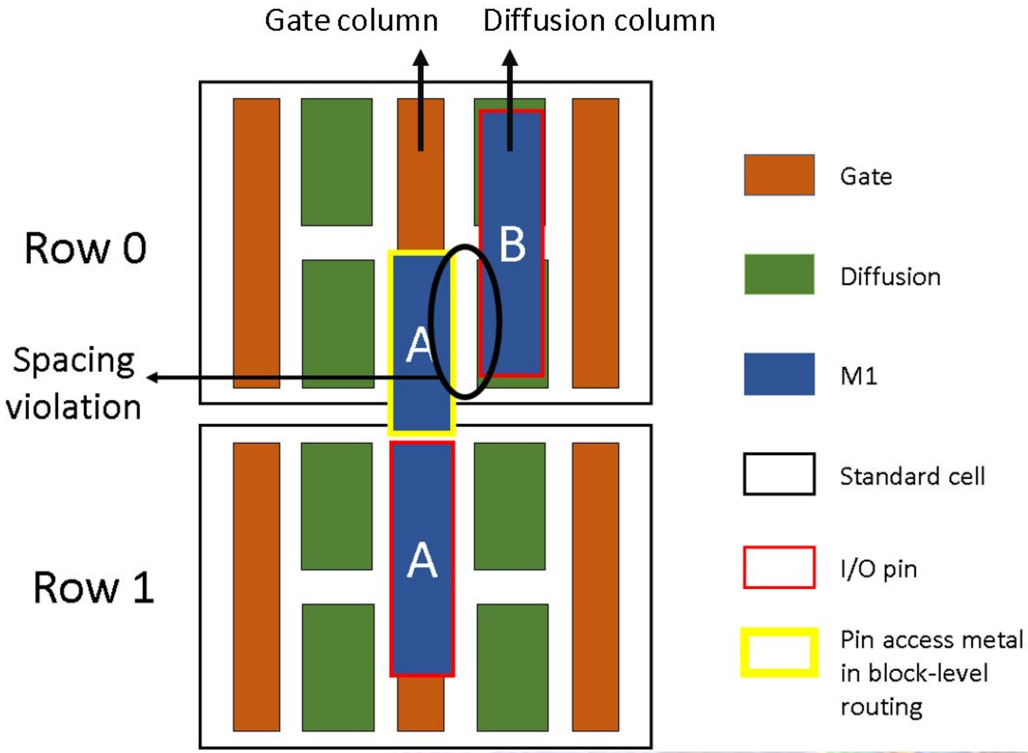
Diffusion column-only

M1 grid graph



Grid Graph Formulation

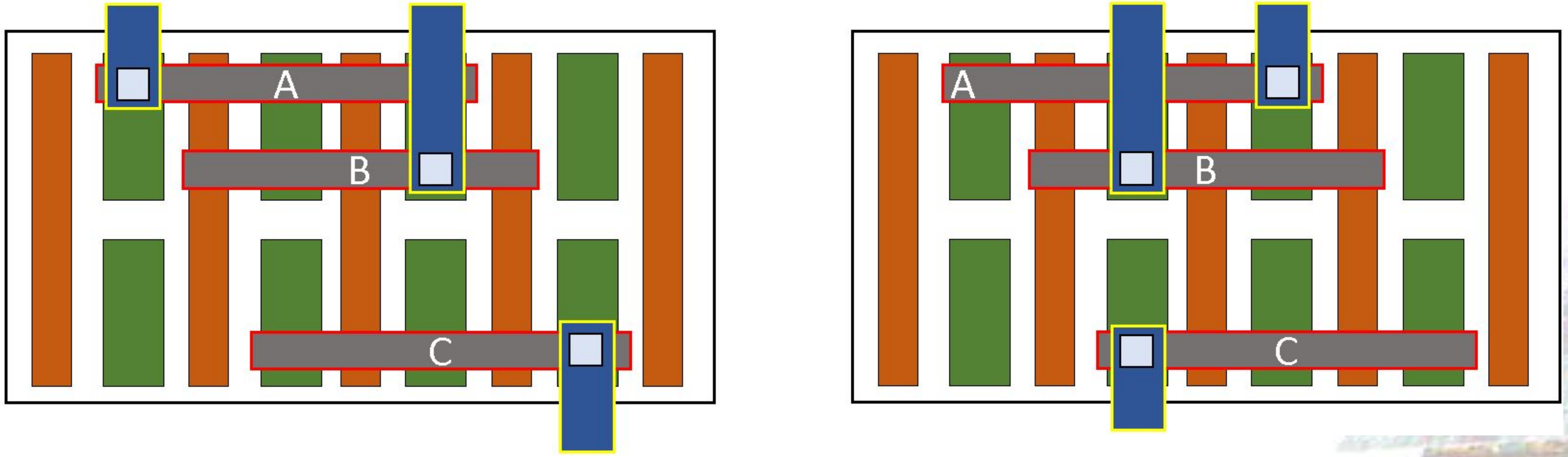
- Diffusion column-only M1 grid graph
  - M1 usage in block-level routing.
  - Cross row M1s on diffusion column and gate column will easily cause spacing violations.



Spacing violation

M0 pin allocation

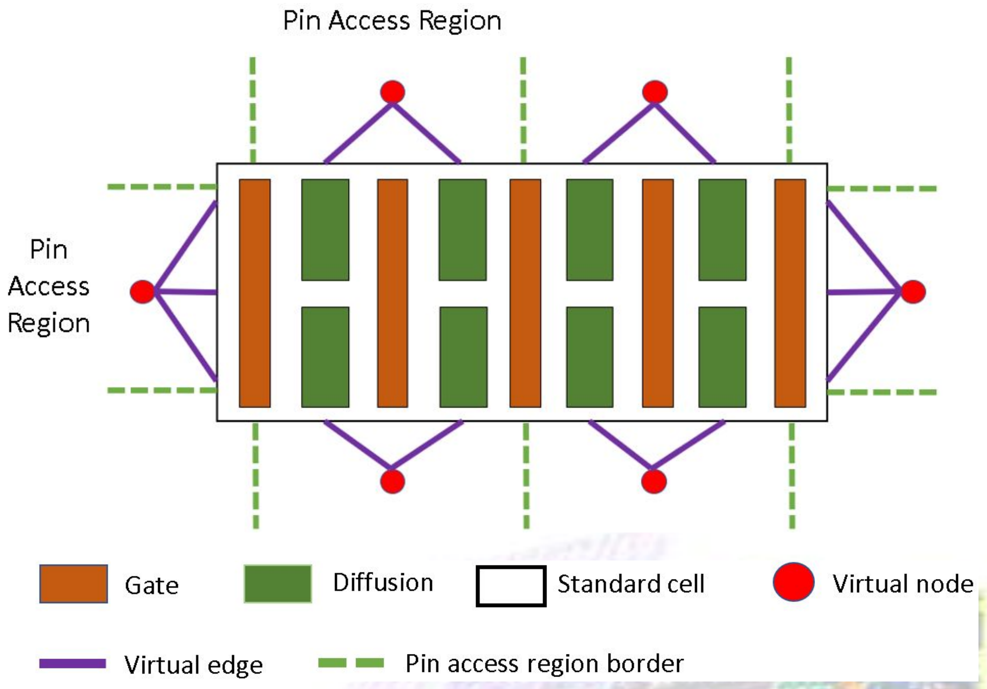
- Standard cells may be accessed from different regions or directions.
- Simulate potential pin access patterns during routing and extract M0 pins from those patterns.



Different pin access patterns in block-level routing for a standard cell

Virtual Node

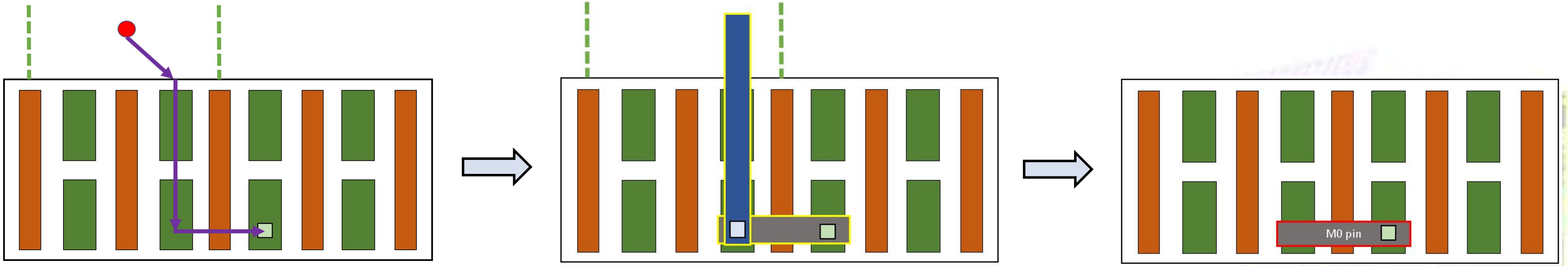
- M0 pin allocation
  - Allocate virtual nodes around the cell.
    - Horizontal: one for each row on left / right.
    - Vertical:
      - Divide horizontal span of the cell into $\lceil nCol/nIO \rceil$ regions.
      - Allocate a virtual node to each region on top and bottom of the cell.
  - Connect virtual nodes to corresponding tracks/columns with virtual edges in its region.



- $nCol$: number of diffusion column
- nIO: number of I/O signal

Virtual Node

- M0 pin allocation
  - During routing, concurrently route each virtual node to I/O contacts.
  - After routing, remove routes starting from virtual nodes until the last M0 connecting to diffusion of an I/O signal with depth-first search (DFS).
    - Set the remaining M0 as an I/O pin.



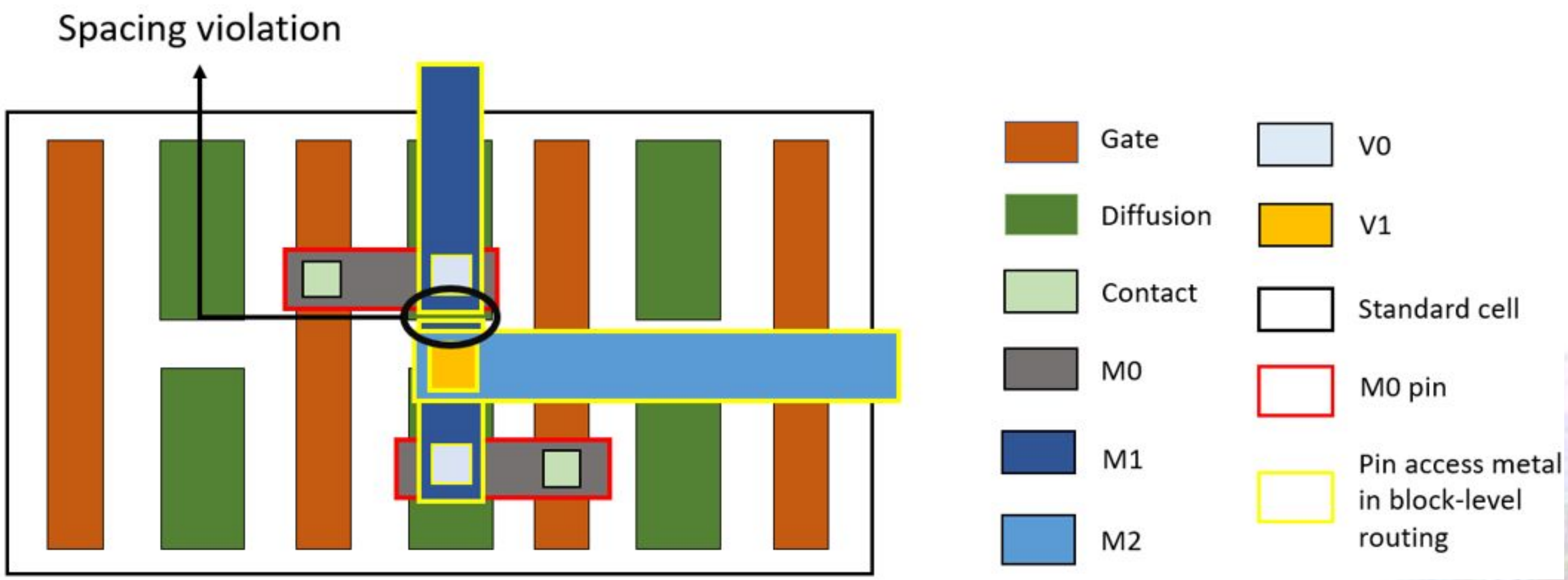
Virtual Node

- M0 pin allocation
  - Maximize the number of successful connections to virtual nodes.
    - $\text{Maximize } \sum_{n \in N_{IO}} \sum_{\forall G_{n,i}^{VCN} \in VCN(n)} (\wedge \mathcal{C}(G_{n,i}^{VCN}))$
  - Each IO signal has at least one successful connection to virtual node.
    - $\sum_{\forall G_{n,i}^{CN} \in VCN(n)} \wedge \mathcal{C}(G_{n,i}^{VCN}) \geq 1, \forall\ n\ \in N_{IO}$

N / N_{IO}	The set of nets. / The set of nets for IO signals.
VCN	The set of connection graphs used to route to virtual nodes. VCN(n): The set of connection graphs used to route to virtual nodes for signal n.
G^{PH} / G^{CN}	Physical graph / Connection graph. $G_{n,i}^{VCN}$: i^{th} connection graph in VCN(n).
$\mathcal{C}(G)$	The set of routing constraints for graph G

Pin Overlap Reduction

- Motivation
  - Vertically overlapped horizontal pins may easily lead to design rule violations in pin access.



Pin Overlap Reduction

- Method

a. Create vector "*AP*" (access potential).

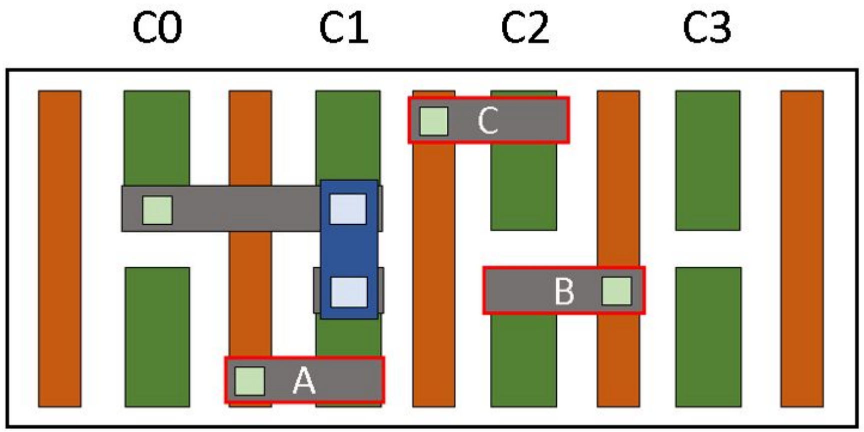
- i^{th} element corresponds to the possibility that i^{th} diffusion column is used in pin access .

b. Update each I/O pin to *AP*.

- For I/O signal i, update "1/*PL*[i]" to *AP*[j].
- j: diffusion column covered by pins of signal i.

c. Soft constraints limiting *AP*[j] < *cap*[j] for all column j.

- *cap*[j] = *r* * (1 - *m1Len*[j] / *nTrack*)
  - *r* = 1.5 (resource on a column)

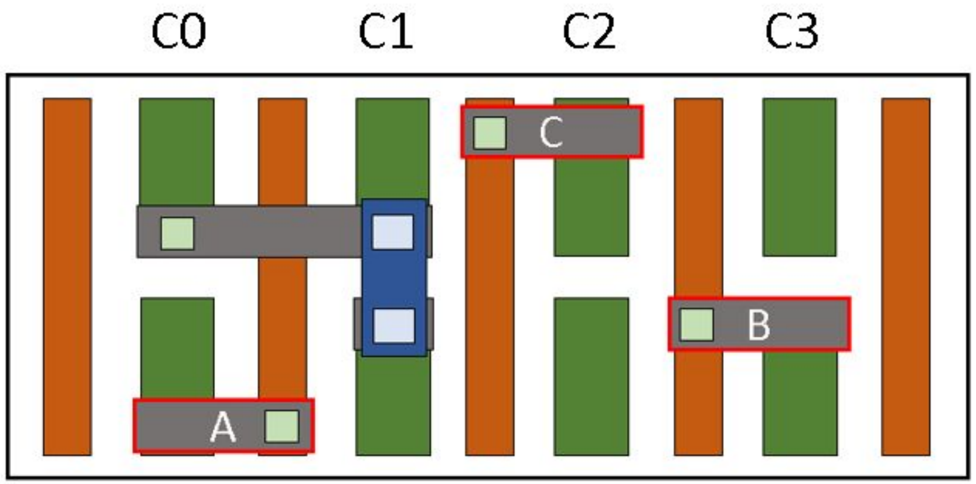


	C0	C1	C2	C3
Cap	1.5	0.75	1.5	1.5
AP	0	1	2	0

illegal pin pattern

Pin Overlap Reduction

- Legal layout pattern



	C0	C1	C2	C3
Cap	1.5	0.75	1.5	1.5
AP	1	0	1	1

Legal pin pattern

Outline

- Introduction
- Previous work
- Preliminaries
- Synthesis Flow
- Algorithm
- Experimental Results
- Conclusions

Cell Synthesis Result Comparison

- NS3K: A 3-nm Nanosheet FET Standard Cell Library Development and Its Impact

	#CPP			#Row	Runtime (sec)
Cell Name	NS3K	Ours	Imp.	NS3K&Ours	Ours
AND2_X1	5	4	1	1	1.0
AND2_X2	8	7	1	1	4.2
AND2_X4	14	12	2	1	22.6
AND3_X1	6	5	1	1	1.9
AOI21_X1	4	4	0	1	0.7
AOI221_X1	6	6	0	1	3.1
AOI22_X1	5	5	0	1	1.7
BUF_X1	3	3	0	1	0.3
BUF_X16	25	25	0	1	2.3
BUF_X2	4	4	0	1	0.8
BUF_X32	49	49	0	1	8.5
BUF_X4	7	7	0	1	1.7
BUF_X8	13	13	0	1	1.3
DFF_X1	20	20	0	2	109.8
DFF_X2	22	20	2	2	109.2
FA_X1	22	18	4	2	153.9
INV_X1	2	2	0	1	0.2
INV_X16	17	17	0	1	0.8
INV_X2	3	3	0	1	0.2
INV_X32	33	33	0	1	1.9

	#CPP			#Row	Runtime (sec)
Cell Name	NS3K	Ours	Imp.	NS3K&Ours	Ours
INV_X4	5	5	0	1	0.6
INV_X8	9	9	0	1	0.4
MUX_X1	9	7	2	1	2.9
NAND2_X1	3	3	0	1	0.3
NAND2_X2	5	5	0	1	0.9
NAND2_X4	9	9	0	1	8.9
NAND3_X1	4	4	0	1	0.9
NOR2_X1	3	3	0	1	0.4
NOR2_X2	5	5	0	1	0.9
NOR2_X4	9	9	0	1	8.4
NOR3_X1	4	4	0	1	0.8
OAI21_X1	4	4	0	1	0.6
OAI221_X1	6	6	0	1	2.7
OAI22_X1	5	5	0	1	1.6
OR2_X1	5	4	1	1	0.8
OR2_X2	7	6	1	1	2.3
OR2_X4	8	7	1	1	3.9
OR3_X1	6	5	1	1	1.8
XNOR2_X1	6	6	0	1	1.0
XOR2_X1	6	6	0	1	1.1

CPP: contact poly pitch

Cell Synthesis Result Comparison

- NS3K: A 3-nm Nanosheet FET Standard Cell Library Development and Its Impact

	#CPP			#Row		Runtime (sec)
Cell Name	NS3K	Ours	Imp.	NS3K	Ours	Ours
AOI222_X1	10	8	2	2	1	9.6
HA_X1	16	10	6	2	1	10.7
MUX_X2	14	10	4	2	1	9.0
OAI222_X1	10	8	2	2	1	9.8
XNOR2_X2	12	9	3	2	1	9.3
XOR2_X2	12	9	3	2	1	7.7

Cells with different number of rows

Block-level Place and Route Comparison

- NS3K: A 3-nm Nanosheet FET Standard Cell Library Development and Its Impact

util. = 70%	Area (um^2)			#DRV			WL (um)		
Benchmark	NS3K	Ours	Imp. (%)	NS3K	Ours	Imp. (%)	NS3K	Ours	Imp. (%)
b19	687.7	627.6	8.7	6	0	100.0	30344.8	24423.3	19.5
marochinno	2954.1	2830.5	4.2	94	0	100.0	207631.7	181064.6	12.8
mblite	914.7	898.8	1.7	269	0	100.0	101038.6	78695.1	22.1
aes	398.3	395.4	0.7	9	0	100.0	38285.4	33892.6	11.5
lxp32	5934.8	5899.1	0.6	215040	26085	87.9	950659.6	821606.8	13.6
Avg.			3.2			97.6			16.5

Block-level P&R results under core utilization = 70%

Block-level Place and Route Comparison

- NS3K: A 3-nm Nanosheet FET Standard Cell Library Development and Its Impact

	#DRV			WL (um)		
Benchmark	NS3K	Ours	Imp. (%)	NS3K	Ours	Imp. (%)
b19	6	0	100.0	29585.8	24423.3	17.4
marochinno	74	0	100.0	205468	181064.6	11.9
mblite	289	0	100.0	98850.4	78695.1	20.4
aes	9	0	100.0	38735.3	33892.6	12.5
lxp32	209005	26085	87.5	948899	821606.8	13.4
Avg.			97.5			15.1

Block-level P&R results under same chip area
(area = 70% utilization using our library)

Outline

- Introduction
- Previous work
- Preliminaries
- Framework
- Algorithm
- Experimental Results
- Conclusions

Conclusions

- A standard cell synthesis framework for nanosheet FET structure.
- Pin access optimization
  - Virtual nodes to allocate I/O pins on M0.
  - A pin overlap reduction technique to further reduce conflicts in pin access.
- In block-level P&R
  - Improve area by 3.2%, DRV counts by 97.6%, wirelength by 16.5% compared to a hand-crafted cell library (NS3K).