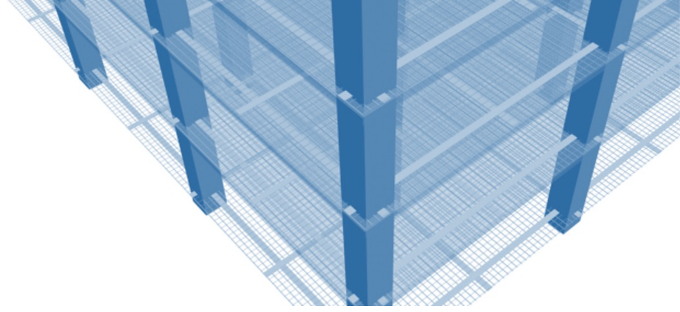
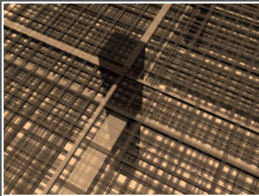
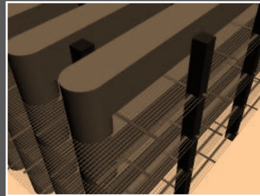
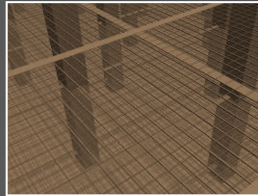
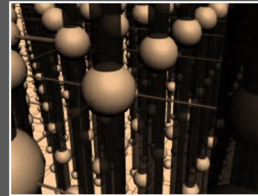
BUU BURAPHA UNIVERSITY WISDOM OF THE EAST

GT Georgia Tech

SYNOPSYS



Placement-Aware 3D Net-to-Pad Assignment for Array-Style Hybrid Bonding 3D ICs

Pruek Vanna-iampikul[1], Junsik Yoon[3], Chaeryung Park[3], Gary Yeap[3], and Sung Kyu Lim[2]

1. Burapha University

2. Georgia Institute of Technology

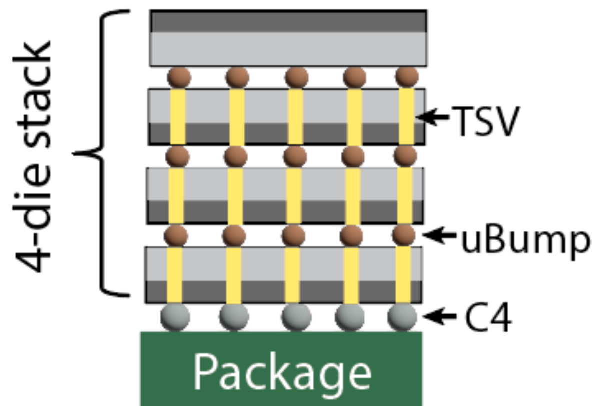
3. Synopsys Inc.

Outline

- Introduction: 3D Integration / 3D Terminal Assignment
- AnchorGrid: Placement-Aware 3D Net-to-Pad Assignment for Array-Style Hybrid Bonding 3D ICs
- Experimental Results
- Conclusion

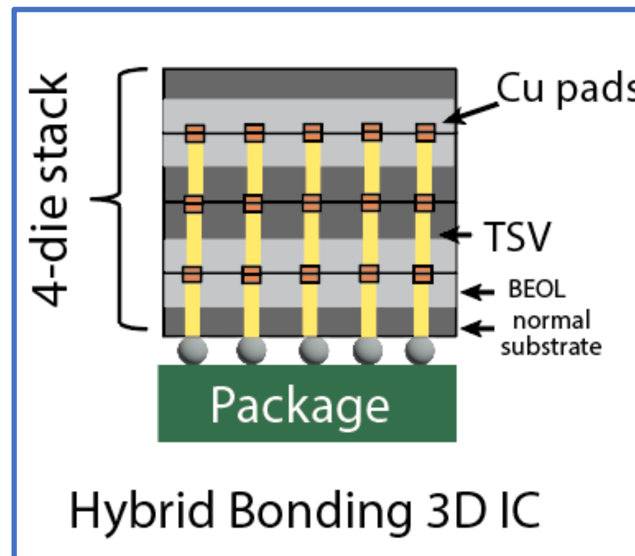
Introduction: 3D Integration Technologies

- Die Stacking with various 3D Integration Technologies
  - TSV + micro bumping 3D IC
  - Hybrid Bonding 3D IC
  - Monolithic 3D IC



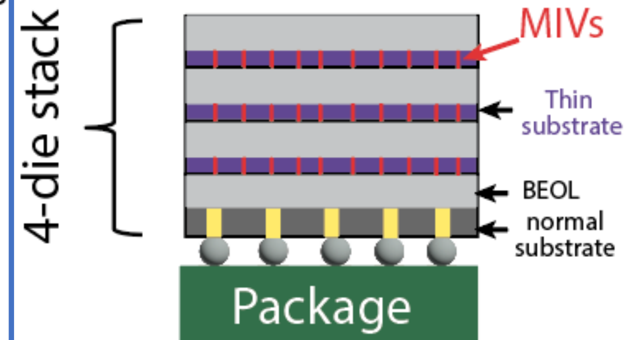
TSV-based 3D IC

Early days



Hybrid Bonding 3D IC

Our focus

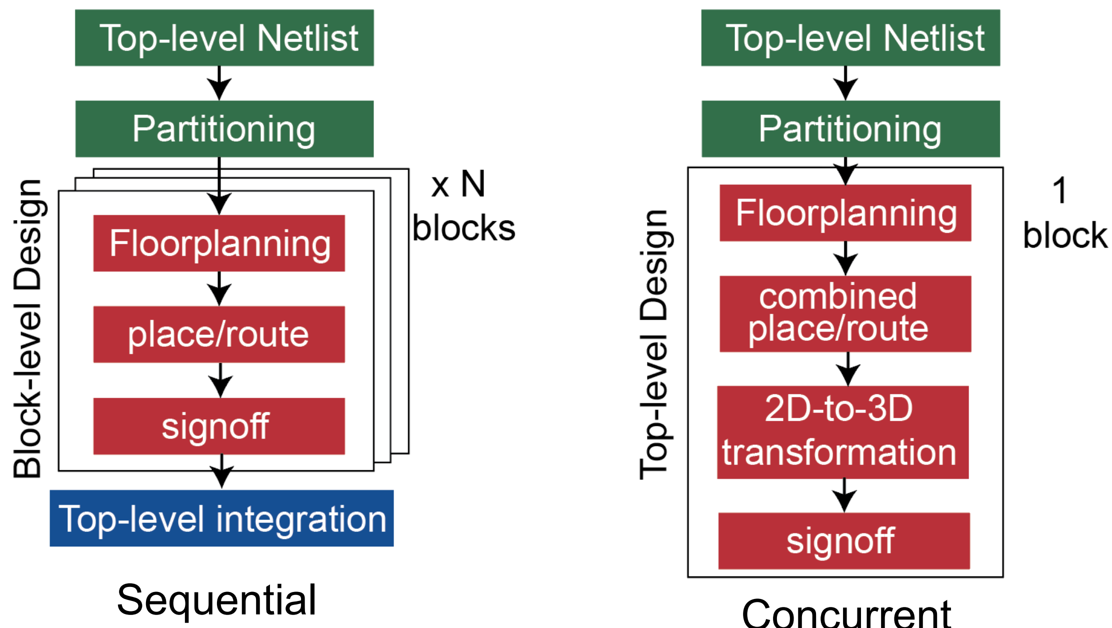


Monolithic 3D IC

Coming soon

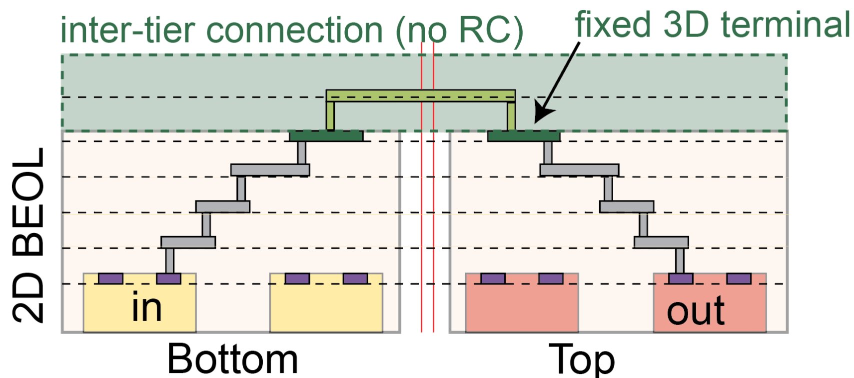
Introduction: EDA Tools for 3D ICs

1. **True-3D EDA Tools** → Actual 3D Place & Route (PnR)
   But commercial tool NOT READY
2. **Pseudo-3D Flow** → Utilize 2D Commercial tool to build 3D IC
   - **Sequential** : Design each die separately
   - **Concurrent**: Co-design & optimize multiple dies together

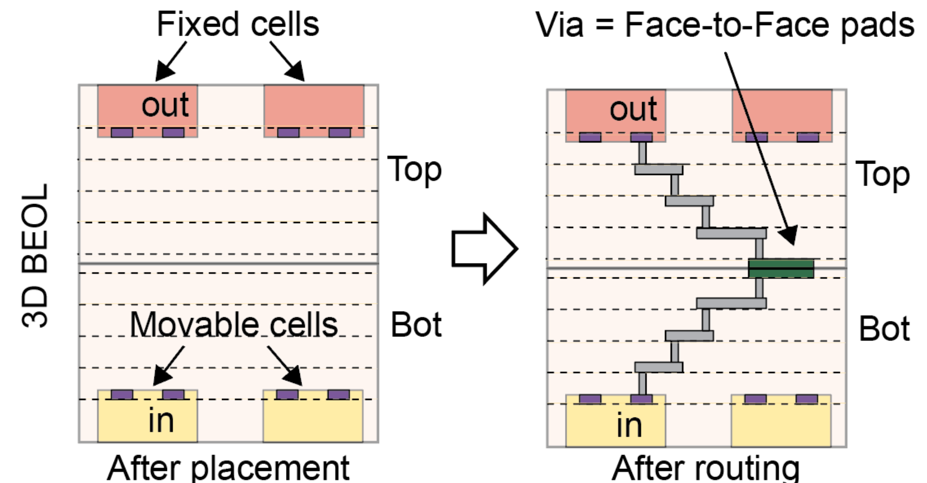


Introduction: Terminal Assignment in 3D IC

1. **Assign-First Flows :** Fixed 3D Terminal before PnR (Cascade-2D)
   Key: Cascaded 2D design with no-RC bridge connections (fixed)
2. **Assign-Last Flows :** Assign 3D Position in the final routing stage (Compact-2D, Pin-3D, Snap-3D, etc.)
   Key: Optimize one die at a time with 3D BEOL



Assign-first flow (Cascade-2D)



Assign-last flow (Pin-3D)

Introduction: Motivation & Problem Statement

- **Assigned-First Flow (Cascade-2D)**
  - **Limitations:** Fixed 3D Terminal may lead to sub-optimal result
  - **Strength:** DRC-free, Scalability and No Library changes
- **Assigned-Last Flow (Pin-3D, Snap-3D)**
  - **Limitations:** Require Library changes, May not be scalable (>2 Dies)
  - **Strength:** Dynamic 3D Terminal location (Based on routing)

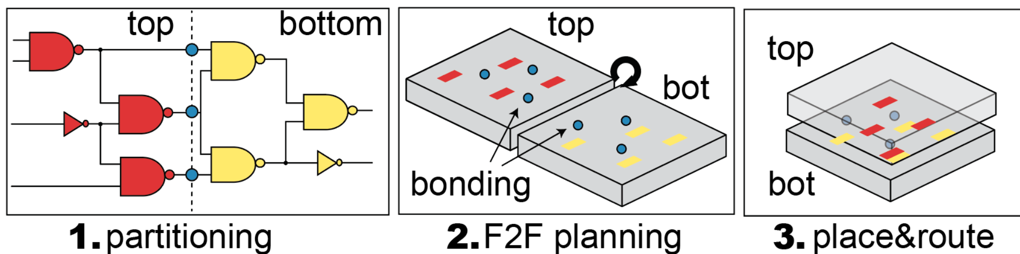
- **Proposed Solution**
  - **Enable flexible 3D Terminal location** in Assigned-First flow
  - Aim to **enhance PPA** and **reduce Design rule check (DRC) violation** (especially in **Advance Node 3D ICs**)
  - Support **Array-Style Hybrid bonding for multi-die integration**

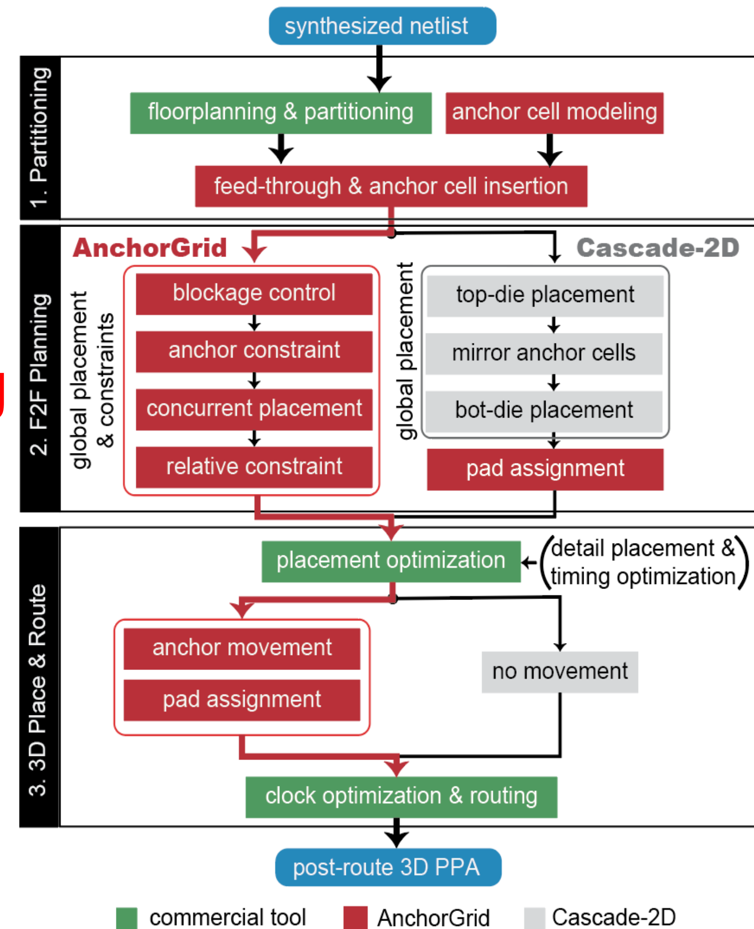
Outline

- Introduction: 3D Integration / 3D Terminal Assignment
- AnchorGrid: Placement-Aware 3D Net-to-Pad Assignment for Array-Style Hybrid Bonding 3D ICs
- Experimental Results
- Conclusion

AnchorGrid: Overview

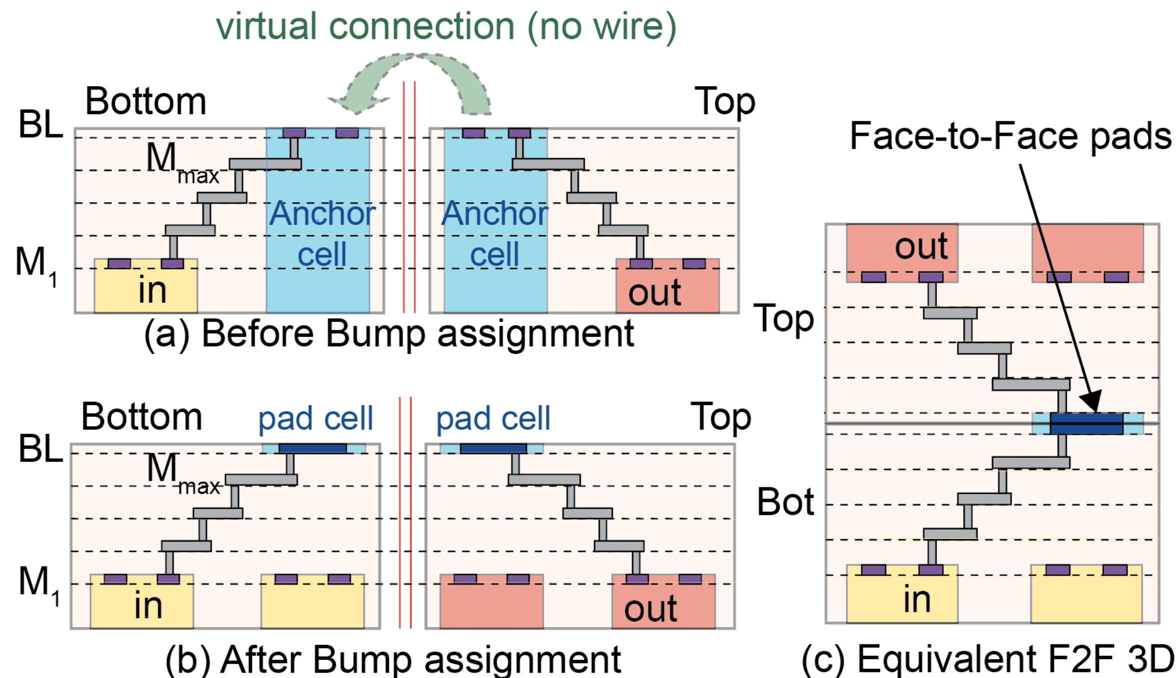
- **Advance Cascade-2D** by
  - Enabling the **3D Terminal movement**
    - Anchor cell modeling
    - Anchor Constraints
  - Supporting **Full-Array Hybrid bonding**
    - **Assign to grid location**





AnchorGrid: Anchor Cell Modeling

- **Anchor Cells → Represent 3D Net**
- **Two types of anchor cell**
  1. Anchor cells : Movable
  2. Pad cells : Fixed



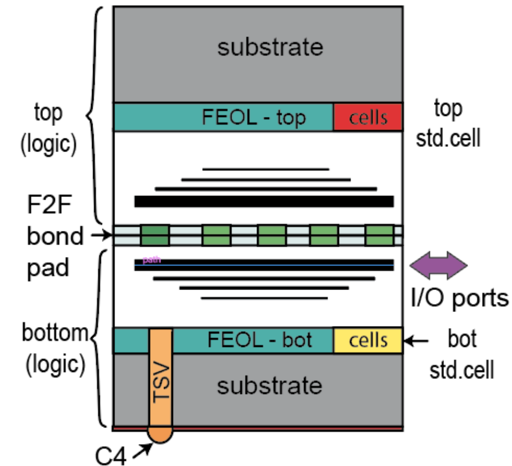
(a) Before Bump assignment

(b) After Bump assignment

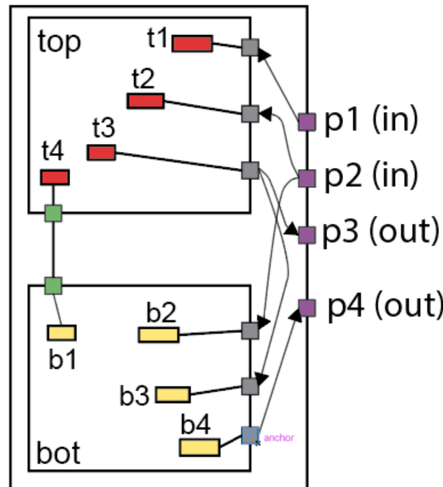
(c) Equivalent F2F 3D

AnchorGrid: Feed-through & Anchor Insertion

- **Given partitioning (top/bottom)**
- **Avoid multi-port Input/Output**
  - Add Feedthrough net
- **Transform 3D net to anchor pairs**
  - Insert movable anchor cells

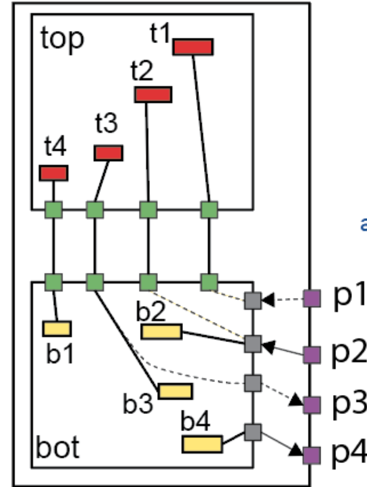


net	ports/pins
n1	t1, p1
n2	t2, b2, p2
n3	t3, b3, p3
n4	b4, p4
n5	b1, t4



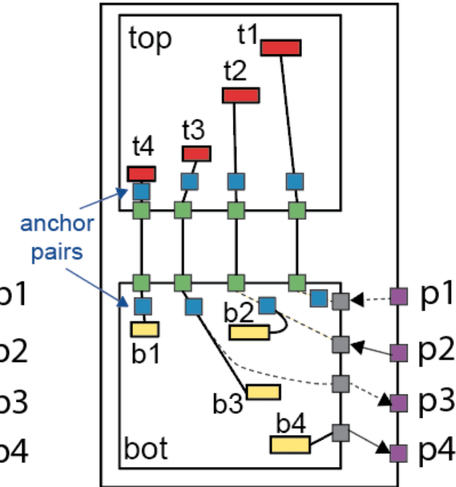
top-level

(a) Initial connections



top-level

(b) Feed-through insertion

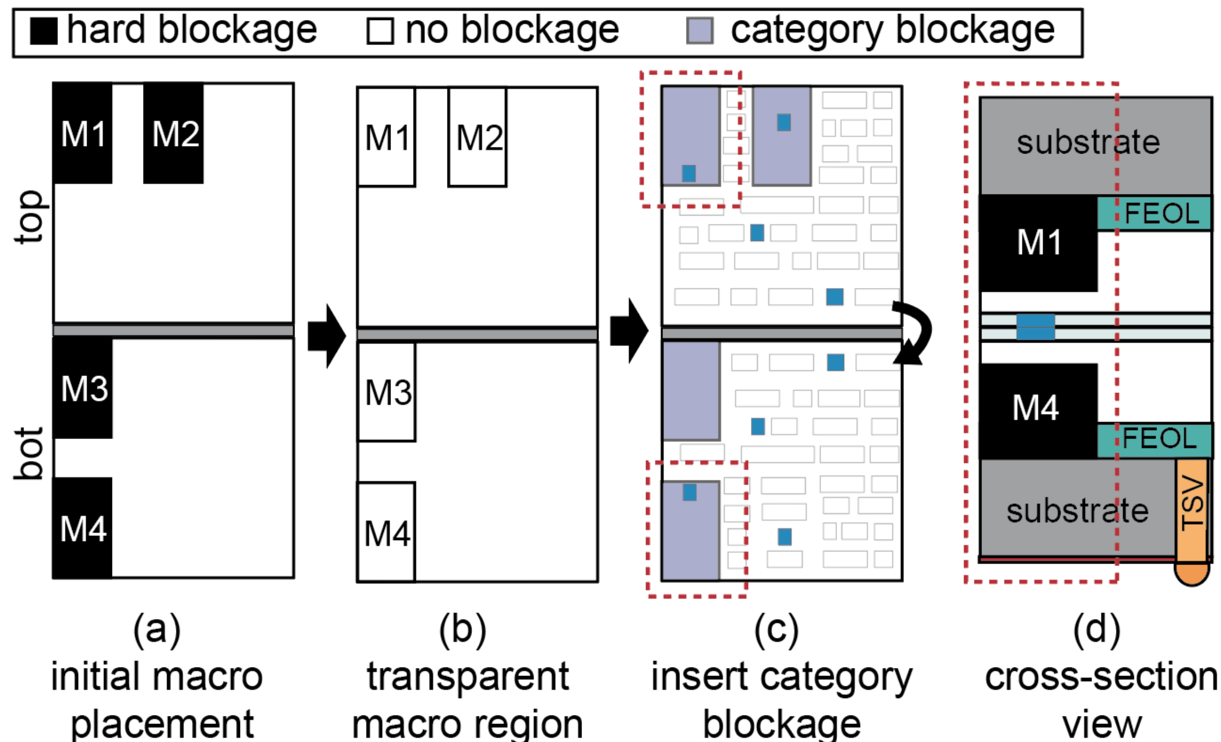


top-level

(c) Anchor cell insertion

AnchorGrid: Blockage Control

- **Goal: Allow Only anchor cells to be placed within memory regions**
- **Generate special blockages:** Block Std Cells
- **Transparent the memories:** Allow the region to be placed



(a) initial macro placement

(b) transparent macro region

(c) insert category blockage

(d) cross-section view

AnchorGrid: Anchor constraint

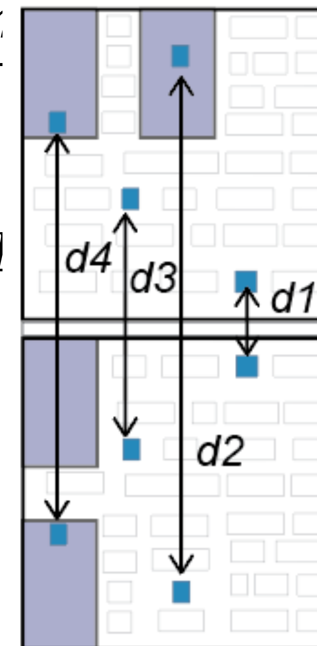
- Align the correct locations (non-uniform distance)
- Flip the top die to simplify the constrain
- **Uniform constraint:**

$$D = die_height + \frac{cutli}{}$$

- **Uniform movement**

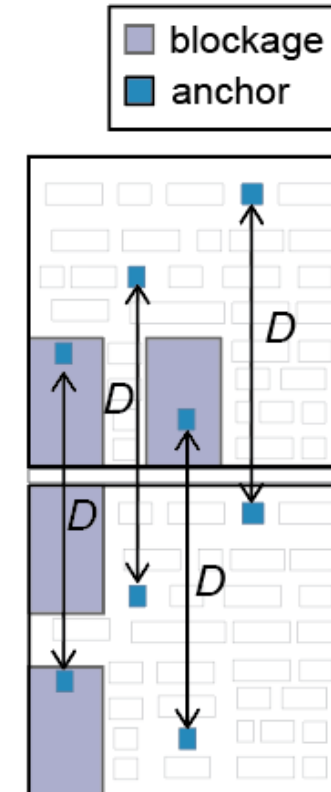
$$\Delta y_t = \Delta y_b + l$$

where cutline_height is gap height
y_t = y position of top anchor cell
y_b = y position of bottom anchor cell



(a) Non-uniform constraint (hard)

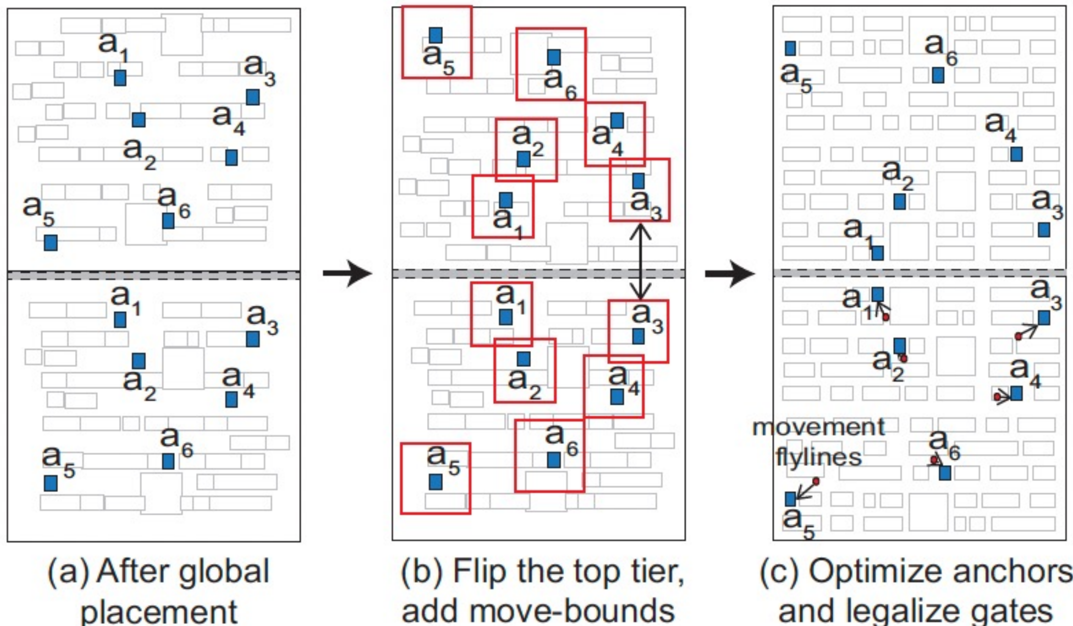
flip top



(b) Uniform constraint (easy)

AnchorGrid: Relative Constraint

- **After global placement**
  - Anchor are aligned (**But not snap into rows**)
- **To support Detail placement:**
  - **Replace anchor constraint with Move bound**
  - Allow legalization Based on Slack value of each 3D net



(a) After global placement

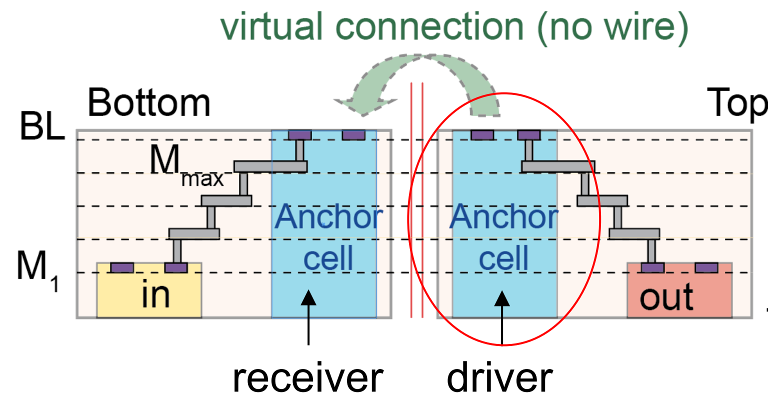
(b) Flip the top tier, add move-bounds

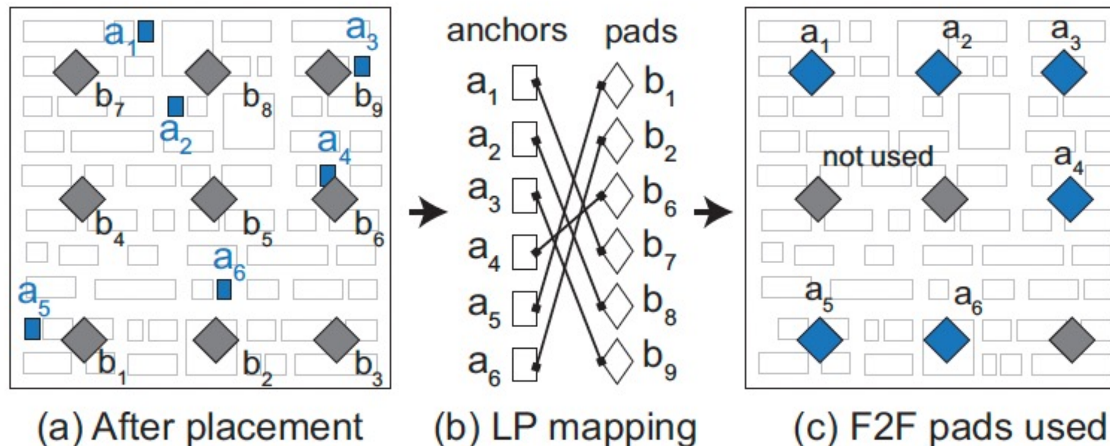
(c) Optimize anchors and legalize gates

Maximum distance is 4X of F2F pitch (can be adjusted)

AnchorGrid: Pad Assignment

- **Goal: Legalize anchor to Grid location**
- **Choose driver cell from each anchor pairs as representation**





(a) After placement (b) LP mapping (c) F2F pads used

Outline

- Introduction: 3D Integration / 3D Terminal Assignment
- AnchorGrid: Placement-Aware 3D Net-to-Pad Assignment for Array-Style Hybrid Bonding 3D ICs
- Experimental Results
- Conclusion

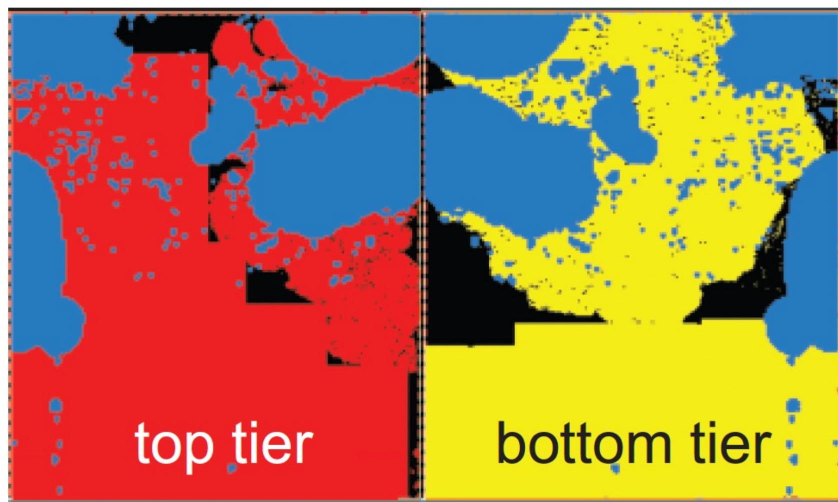
Experimental Results:
Design Benchmark

- **3 commercial processor benchmarks**
- **3 advance technology Nodes** (**5nm** and **7nm**)
- Different memory macro utilization (**Mem-on-Logic** / **Logic-on-Logic**)

Design	Technology Node	Memory macro Utilization (%) (top/bottom)	# 3D nets	F2F pitch (um)
CPU1	5nm (1)	90% / 0% Memory on Logic	7,966	1.2
CPU2	5nm (2)	50% / 45% Logic on Logic	6,233	2.0
CPU3	7nm	10% / 15% Logic on Logic	17,132	3.0

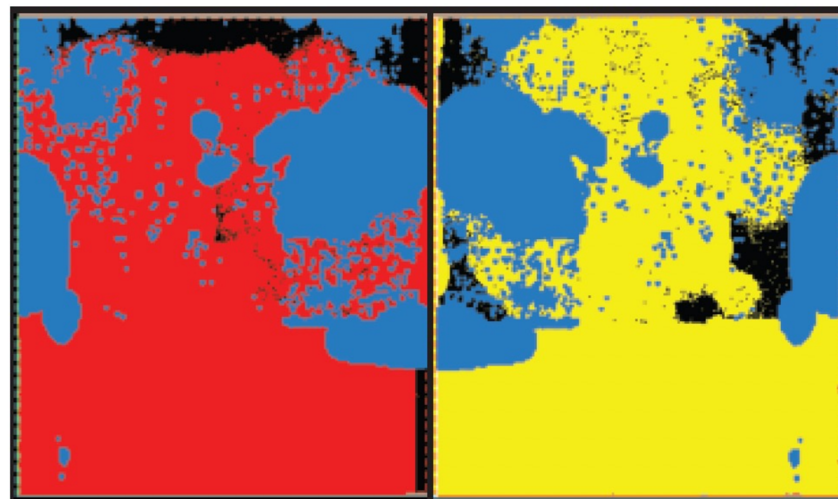
Experimental Results:
F2F locations Before Vs After Place Opt.

- **Anchor cells are moved along with Std. cells**
  - Buffer Insertion to close the timing
  - Slight changes in location based on move bound (relative constraint)



Before Placement Optimization

Cascade-2D



After Placement Optimization

AnchorGrid (Ours)

Experimental Results:
PPA Impact from Flexible 3D Terminal

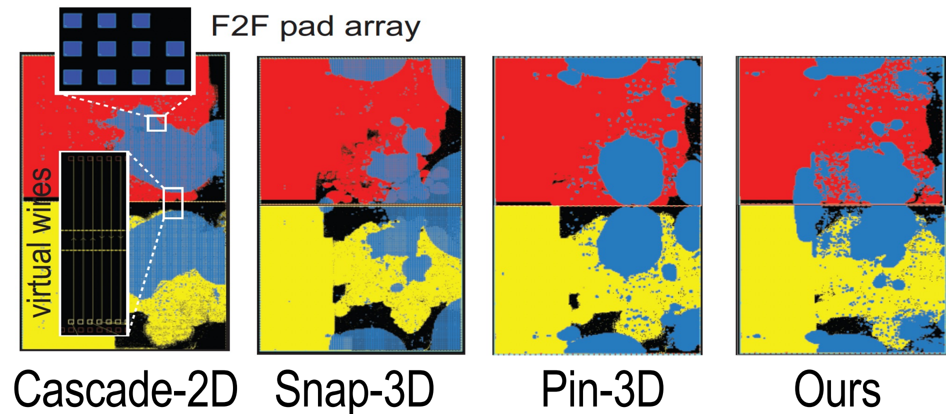
- **PPA Comparison** between Cascade-2D and **Ours**
  - Better Setup/Hold timing, Shorter wirelength, Power-Delay-Product

		CPU1 - 4.5GHz			CPU2 – 2.5GHz			CPU3 - 0.54GHz		
		C-2D	Ours	%Impv	C-2D	Ours	%Impv	C-2D	Ours	%Impv
# Metal layer (top+bot)		11 + 11			14 + 14			12 + 12		
# cells		641K	639K	0.35%	439K	430K	2.1%	1M	1.1M	-1.4%
Setup	WNS (ns)	-0.143	-0.072	49.6%	-0.23	-0.21	8.7%	-0.31	-0.013	95.8%
	TNS (ns)	-76.6	-38.1	50.2%	-192.8	-156	18.7%	-65	-0.3	99.5%
	#vio EPs	4,964	3,734	24.7%	9,775	8,404	14%	706	195	72.4%
Hold	HWNS (ns)	-0.116	-0.113	2.5%	-0.1	-0.07	30%	-1.41	-0.102	92.7%
WL (m)		5.26	5.19	1.3%	6.04	6.17	-2.1%	20.1	18.8	6.47%
3D Net WL (m)		1.09	0.98	10.1%	0.49	0.43	12.2%	9.38	6.97	25.7%
Power (mW)		1,334	1,312	1.7%	287	270	5.9%	73.3	64.2	12.4%
PDP (mW ns)		484.2	383.1	20.9%	180.8	164.7	8.9%	159.8	120.9	24.3%

Experimental Results:
PPA Comparison with State-of-the-Art

- **C**ascade-**2D** **vs.** **S**nap-**3D** **vs.** **P**in-**3D** **and Our** AnchorGrid
- Global Placement + Pad Assignment + Cascade PnR

		CPU 2 – 2.5GHz			
		Final design			
		C-2D	S-3D	P-3D	Ours
# cells		439K	434K	437K	430K
WL		6.04	5.82	6.05	6.17
Setup	WNS	-0.23	-0.21	-0.21	-0.21
	TNS	-192	-173	-171	-156
	#vio	9.7K	9.5K	9.8K	8.4K
Hold	HWNS	-0.1	-0.07	-0.07	-0.06
	HTNS	-4.3	-5.6	-12.6	-2.6
Power (mW)		287	281	286	270
DRC		2.3K	3.5K	2.5K	224
-> Shorts		1146	1566	1339	90



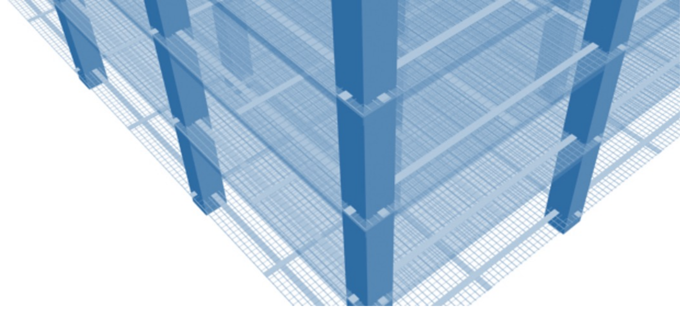
Outline

- Introduction: 3D Integration / 3D Terminal Assignment
- AnchorGrid: Placement-Aware 3D Net-to-Pad Assignment for Array-Style Hybrid Bonding 3D ICs
- Experimental Results
- Conclusion

Conclusions

- **AnchorGrid key ideas**
  - Enable the F2F movement until actual routing stage
    (Account for Buffer insertion & Avoid unnecessary cells by moving F2F)
  - Enable the F2F location within memory macro regions
  - Assign F2F locations to Array-style (Grid) for multi-die integration

- **AnchorGrid key benefits**
  - Reduce Design Rule Check (DRC) Violation in Advance Node Technology
  - Support Array-style Hybrid bonding 3D IC
  - Outperforms Cascade-2D, Pin-3D and Snap-3D
    - Setup/Hold timing, Power, and Wirelength



Thank you for your attention