IEEE/ACM ICCAD 2025 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN 44th Edition

H3D: Heterogeneous Resources Aware Global Router for Face-to-Face Bonded 3D ICs

Yuxuan Zhao, Feng Gu, Siting Liu, Peiyu Liao, Bei Yu

Department of Computer Science & Engineering
The Chinese University of Hong Kong

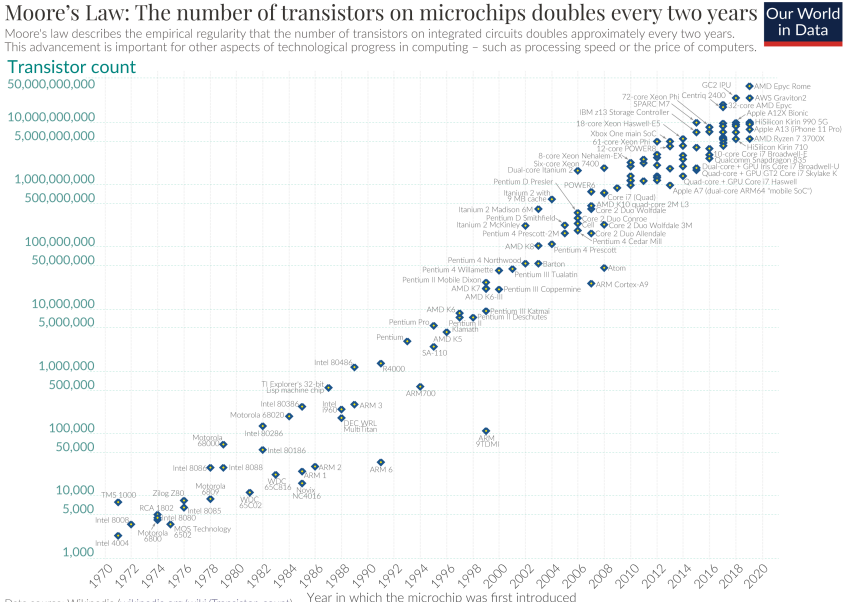
{yxzhao21,byu}@cse.cuhk.edu.hk

October 28, 2025

Outline

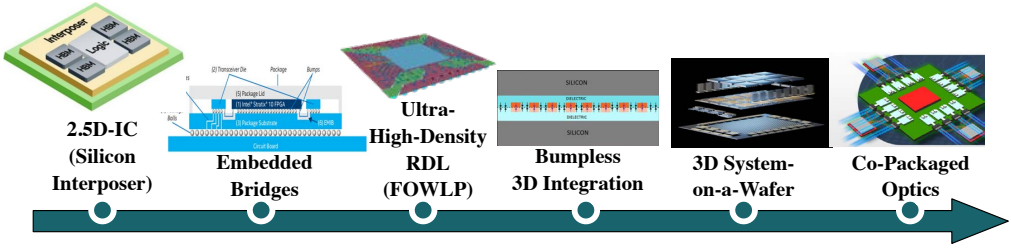
Introduction

Cramming More Components onto Integrated Circuits



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.

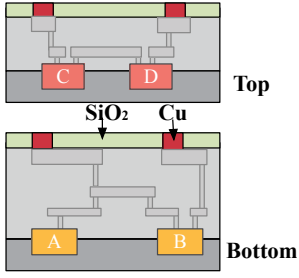
More-than-Moore Heterogeneous Integration



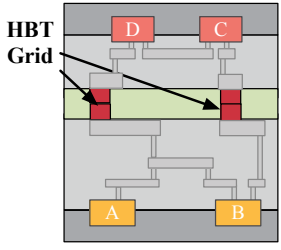
Source: Cadence

- Heterogeneous integration (HI) is a promising solution for better PPA, cost, and memory bandwidth.
- 3D integration is a key technology for HI to achieve Moore style gains.

Face-to-Face Bonded 3D ICs



(a) Prefabricated Dies

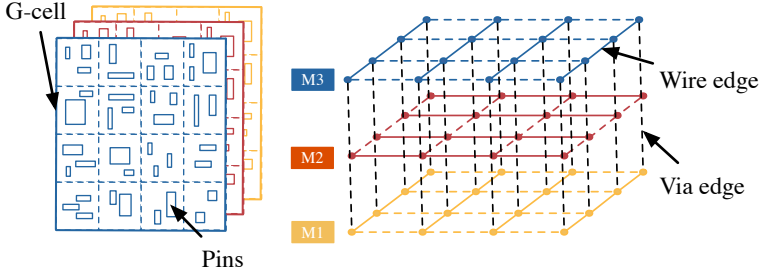


(b) F2F Bonded 3D ICs

- Hybrid bonding technology enables direct metal-to-metal and dielectric-to-dielectric bonding at BEOLs of prefabricated dies.
- With hybrid boning terminal (HBT) pitches below 2 μm becoming feasible, we can achieve ultra-high vertical integration density.

Global Routing

- Signal routing establishes connections between circuit elements using metal tracks across multiple layers.
- Routing problem is divided into two phases:
  - **Global routing**: partitions the layout into a grid map and constructs a rough routing solution in grid units.
  - **Detailed routing**: connects all the nets using actual metal tracks and vias while satisfying all the design rules.



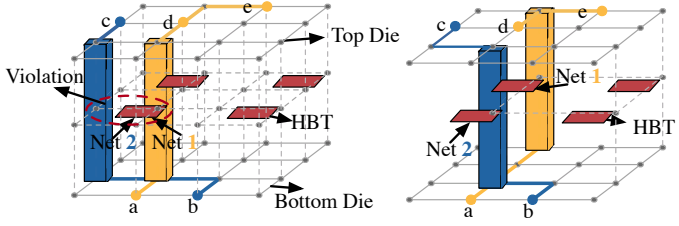
Conventional grid graph model in global routing.

How to Handle the HBT in Routing?

Connection Type	C4 Bump	TSV & μbump	Hybrid Bond	Metal Via*
Typical Pitch	250 μm	50 μm	<10 μm	0.1 μm
# of Connection†	16	400	$> 10^4$	10^8

*: *Via below top-most metal in 28 nm node.* †: *In mm^2 area.*

- Existing pseudo-3D flows [TCAD'19][1] using commercial tools require a post-routing legalization [TCAD'24][2] step to avoid spacing violations.
- Existing 3D pattern routing methods face similar issues.
- We propose a heterogeneous grid graph to model both routing and HBT resources.



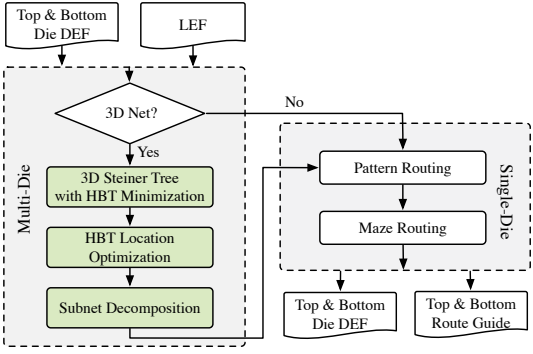
(a) Uniform Grid Graph **(b) Heterogeneous Grid Graph**

[1] **Bon Woong Ku, Kyungwook Chang, and Sung Kyu Lim (2019). "Compact-2D: A physical design methodology to build two-tier gate-level 3-D ICs".** In: *IEEE TCAD* **39.6, pp. 1151–1164.**

[2] **Yen-Hsiang Huang et al. (2024). "On Legalization of Die Bonding Bumps and Pads for 3D ICs".** In: *IEEE TCAD*.

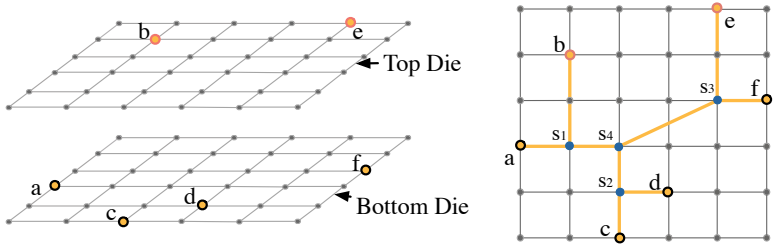
H3D Global Router

Our 3D Global Router Framework



- H3D is a sequential global router that handles both 3D and 2D nets.
- H3D processes nets in ascending order of their HPWL.
- H3D efficiently minimizes the HBT counts and optimizes HBT locations to achieve minimal wirelength without spacing violations.

- **Planar RSMT Construction:**
  - Project the 3D pins to a 2D plane.
  - Generate the rectilinear Steiner minimum tree (RSMT) using FLUTE [TCAD'08][3].

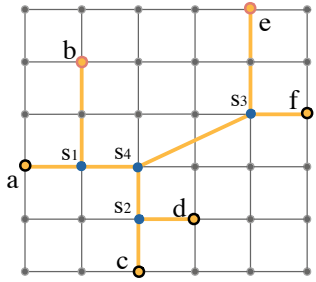


(a) Pins for 3D Net **(b) Planar RSMT**

[3] **Chris Chu and Yiu-Chung Wong (2008). "FLUTE: Fast Lookup Table Based Rectilinear Steiner Minimal Tree Algorithm for VLSI Design".** In: *IEEE TCAD* **27.1, pp. 70–83.**

Dynamic Programming-based HBT Minimization

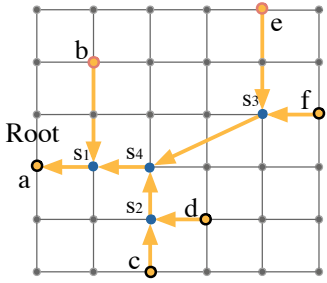
- Similar to layer assignment for via minimization [TCAD'08][4], we use a dynamic programming (DP) algorithm to minimize the HBTs.
- The two-pin net order is determined by a depth-first search (DFS) traversal.



[4] **Tsung-Hsien Lee and Ting-Chi Wang (2008). "Congestion-constrained layer assignment for via minimization in global routing".** In: *IEEE TCAD* **27.9, pp. 1643–1656.**

Dynamic Programming-based HBT Minimization

- Similar to layer assignment for via minimization [TCAD'08][4], we use a dynamic programming (DP) algorithm to minimize the HBTs.
- The two-pin net order is determined by a depth-first search (DFS) traversal.

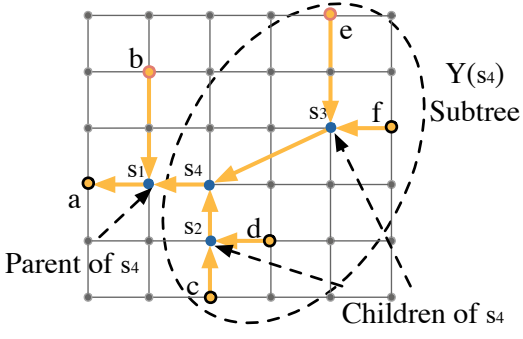


[4] **Tsung-Hsien Lee and Ting-Chi Wang (2008). "Congestion-constrained layer assignment for via minimization in global routing".** In: *IEEE TCAD* **27.9, pp. 1643–1656.**

Dynamic Programming-based HBT Minimization

- Let $mhc(v,l)$ denote the minimal HBT cost of $Y(v)$ when the edge $e(v, par(v))$ is assigned to die l,

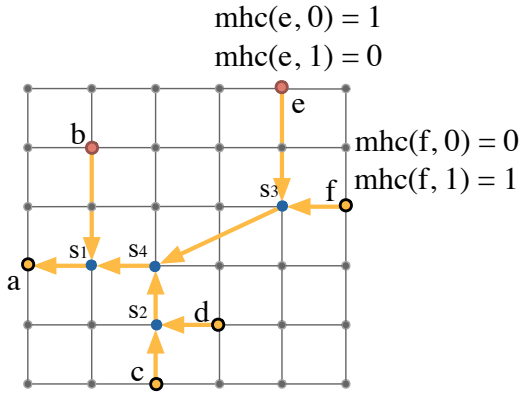
$$mhc(v,l) = \min_{\substack{1\le l_1\le L \\ \cdots \\ 1\le l_k\le L}} \Big(\underbrace{hc(v)}_{\substack{\#\text{HBTs} \\ \text{connecting} \\ k \text{ children}}} + \sum_{j=1}^{k} \underbrace{mhc(v_{c(j)}, l_j)}_{\substack{\text{HBT cost of} \\ \text{child } v_{c(j)} \text{ at die } l_j}} \Big), \tag{1}$$



Dynamic Programming-based HBT Minimization

- Let $mhc(v, l)$ denote the minimal HBT cost of $Y(v)$ when the edge $e(v, par(v))$ is assigned to die l,

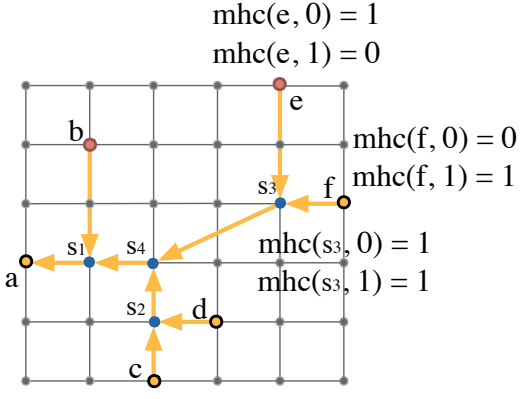
$$mhc(v,l) = \min_{\substack{1\le l_1\le L \\ \cdots \\ 1\le l_k\le L}} \Big(\underbrace{hc(v)}_{\substack{\#\text{HBTs} \\ \text{connecting} \\ k \text{ children}}} + \sum_{j=1}^{k} \underbrace{mhc(v_{c(j)}, l_j)}_{\substack{\text{HBT cost of} \\ \text{child } v_{c(j)} \text{ at die } l_j}} \Big), \tag{1}$$



Dynamic Programming-based HBT Minimization

- Let $mhc(v, l)$ denote the minimal HBT cost of $Y(v)$ when the edge $e(v, par(v))$ is assigned to die l,

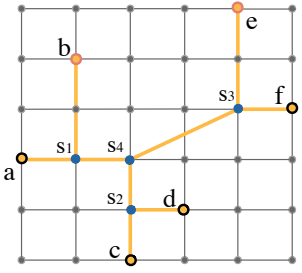
$$mhc(v,l) = \min_{\substack{1\le l_1\le L \\ \cdots \\ 1\le l_k\le L}} \Big(\underbrace{hc(v)}_{\substack{\#\text{HBTs} \\ \text{connecting} \\ k \text{ children}}} + \sum_{j=1}^{k} \underbrace{mhc(v_{c(j)}, l_j)}_{\substack{\text{HBT cost of} \\ \text{child } v_{c(j)} \text{ at die } l_j}} \Big), \tag{1}$$

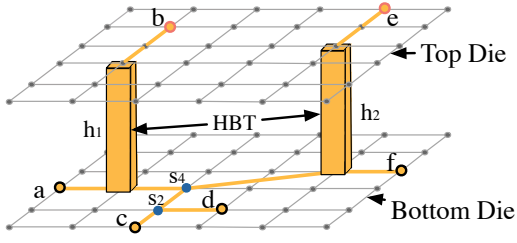


Dynamic Programming-based HBT Minimization

- Let $mhc(v, l)$ denote the minimal HBT cost of $Y(v)$ when the edge $e(v, par(v))$ is assigned to die l,

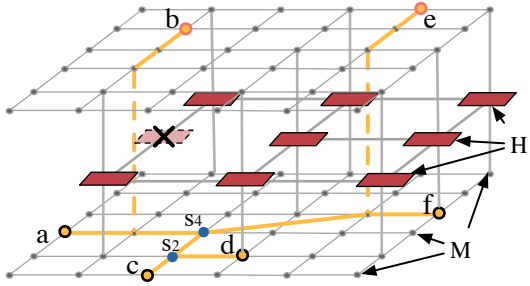
$$mhc(v,l) = \min_{\substack{1\le l_1\le L \\ \cdots \\ 1\le l_k\le L}} \Big(\underbrace{hc(v)}_{\substack{\#\text{HBTs} \\ \text{connecting} \\ k \text{ children}}} + \sum_{j=1}^{k} \underbrace{mhc(v_{c(j)}, l_j)}_{\substack{\text{HBT cost of} \\ \text{child } v_{c(j)} \text{ at die } l_j}} \Big), \tag{1}$$





Dynamic Programming-Based HBT Location Optimization

- **Heterogeneous Grid Graph**:
  - We use a quad-tree to represent the coarse grid for the HBT resources.
  - Some HBTs are occupied by already routed 3D nets.
- Rectilinear Steiner Tree Problem with Fixed Topology (**RSTPFT**) [MATH PROGRAM'75][5] [Oper. Res. Lett.'16][6]: compute the (x, y) of Steiner nodes, $V(T) \setminus P$, to minimize the wirelength.



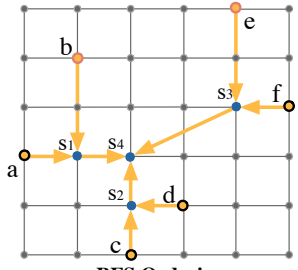
[5] **David Sankoff and Pascale Rousseau (1975). "Locating the vertices of a Steiner tree in an arbitrary metric space". In:** *Mathematical Programming* **9, pp. 240–246.**

[6] **Annika Kristina Kiefner (2016). "Minimizing path lengths in rectilinear Steiner minimum trees with fixed topology". In:** *Operations Research Letters* **44.6, pp. 835–838.**

Dynamic Programming-Based HBT Location Optimization

- Let $g_v(x_v)$ denote the minimal x-wirelength of the subtree $Y(v)$, given x-coordinate of v has value x_v,

$$g_v(x_v) = \sum_{w \in pch(v)} \underbrace{|x_v - x_w|}_{\text{distance to child pin } w} + \sum_{w \in sch(v)} \min_{x_w} \{ \underbrace{g_w(x_w)}_{\text{wirelength of } Y(w)} + \underbrace{|x_v - x_w|}_{\text{distance to child Steiner node } w} \}. \quad (2)$$

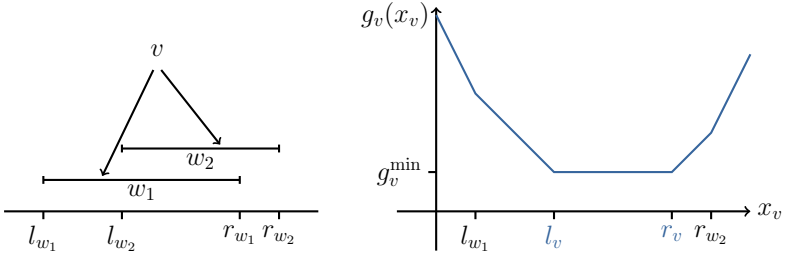


BFS Ordering

Dynamic Programming-Based HBT Location Optimization

- Let $g_v(x_v)$ denote the minimal x-wirelength of the subtree $Y(v)$, given x-coordinate of v has value x_v,

$$g_v(x_v) = \sum_{w \in pch(v)} \underbrace{|x_v - x_w|}_{\substack{\text{distance to} \\ \text{child pin } w}} + \sum_{w \in sch(v)} \min_{x_w} \{ \underbrace{g_w(x_w)}_{\substack{\text{wirelength} \\ \text{of } Y(w)}} + \underbrace{|x_v - x_w|}_{\substack{\text{distance to child} \\ \text{Steiner node } w}} \}. \tag{2}$$



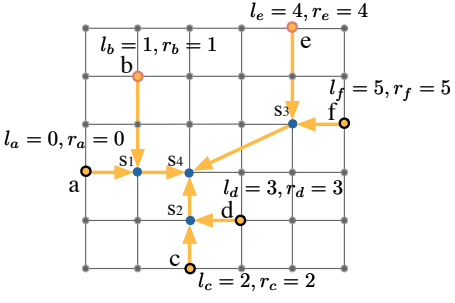
- Define the interval $[l_v, r_v] := \{x_v \in \mathbb{R} \mid g_v(x_v) = g_v^{\min}\}$,

$$g_v(x_v) = \sum_{w \in ch(v)} \left(g_w^{\min} + \max\{0, x_v - r_w\} + \max\{0, l_w - x_v\} \right), \tag{3}$$

Dynamic Programming-Based HBT Location Optimization

- Let $g_v(x_v)$ denote the minimal x-wirelength of the subtree $Y(v)$, given x-coordinate of v has value x_v,

$$g_v(x_v) = \sum_{w \in pch(v)} \underbrace{|x_v - x_w|}_{\text{distance to child pin } w} + \sum_{w \in sch(v)} \min_{x_w} \{ \underbrace{g_w(x_w)}_{\text{wirelength of } Y(w)} + \underbrace{|x_v - x_w|}_{\text{distance to child Steiner node } w} \}. \tag{2}$$



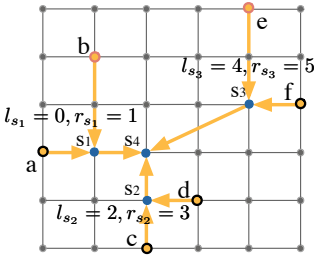
- Define the interval $[l_v, r_v] := \{x_v \in \mathbb{R} \mid g_v(x_v) = g_v^{\min}\}$,

$$g_v(x_v) = \sum_{w \in ch(v)} \left(g_w^{\min} + \max\{0, x_v - r_w\} + \max\{0, l_w - x_v\} \right), \tag{3}$$

Dynamic Programming-Based HBT Location Optimization

- Let $g_v(x_v)$ denote the minimal x-wirelength of the subtree $Y(v)$, given x-coordinate of v has value x_v,

$$g_v(x_v) = \sum_{w \in pch(v)} \underbrace{|x_v - x_w|}_{\text{distance to child pin } w} + \sum_{w \in sch(v)} \min_{x_w} \{ \underbrace{g_w(x_w)}_{\text{wirelength of } Y(w)} + \underbrace{|x_v - x_w|}_{\text{distance to child Steiner node } w} \}. \tag{2}$$



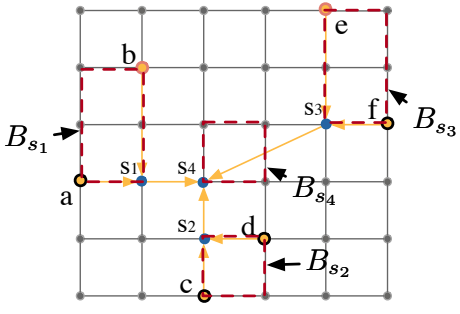
- Define the interval $[l_v, r_v] := \{x_v \in \mathbb{R} \mid g_v(x_v) = g_v^{\min}\}$,

$$g_v(x_v) = \sum_{w \in ch(v)} \left(g_w^{\min} + \max\{0, x_v - r_w\} + \max\{0, l_w - x_v\} \right), \tag{3}$$

Dynamic Programming-Based HBT Location Optimization

- Let $g_v(x_v)$ denote the minimal x-wirelength of the subtree $Y(v)$, given x-coordinate of v has value x_v,

$$g_v(x_v) = \sum_{w \in pch(v)} \underbrace{|x_v - x_w|}_{\text{distance to child pin } w} + \sum_{w \in sch(v)} \min_{x_w} \{ \underbrace{g_w(x_w)}_{\text{wirelength of } Y(w)} + \underbrace{|x_v - x_w|}_{\text{distance to child Steiner node } w} \}. \quad (2)$$



- Define the interval $[l_v, r_v] := \{x_v \in \mathbb{R} \mid g_v(x_v) = g_v^{\min}\}$,

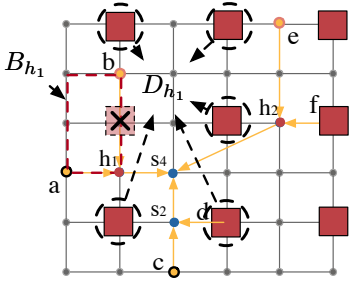
$$g_v(x_v) = \sum_{w \in ch(v)} \left(g_w^{\min} + \max\{0, x_v - r_w\} + \max\{0, l_w - x_v\} \right), \quad (3)$$

Discrete HBT Location Optimization

- An HBT node v can only be placed within the set of currently available HBT array locations, denoted as $S_v := H$,

$$g_v(s) = \sum_{w \in pch(v)} \|s - w\|_1 + \sum_{w \in sch(v)} \min_{t \in S_w} \{g_w(t) + \|s - t\|_1\}, \quad \forall s \in S_v. \tag{4}$$

- Considering all the candidate locations results in $O(|V||S_v|^2)$ time.
- We propose considering the available locations ($B_v \cap S_v$) in the optimal region B_v and the closest points (D_v) to B_v in 8 directions.
- It takes $O(|V||S_v^c|^2)$ time, where $|S_v^c| \ll |S_v|$.

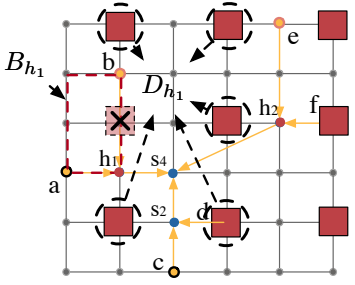


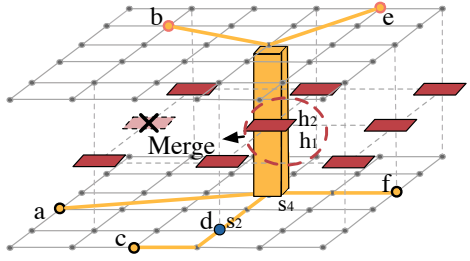
Discrete HBT Location Optimization

- An HBT node v can only be placed within the set of currently available HBT array locations, denoted as $S_v := H$,

$$g_v(s) = \sum_{w \in pch(v)} \|s - w\|_1 + \sum_{w \in sch(v)} \min_{t \in S_w} \{g_w(t) + \|s - t\|_1\}, \quad \forall s \in S_v. \tag{4}$$

- Considering all the candidate locations results in $O(|V||S_v|^2)$ time.
- We propose considering the available locations ($B_v \cap S_v$) in the optimal region B_v and the closest points (D_v) to B_v in 8 directions.
- It takes $O(|V||S_v^c|^2)$ time, where $|S_v^c| \ll |S_v|$.





Subnet Decomposition, Pattern and Maze Routing

- Decompose the 3D net into single-die subnets by splitting at the HBT nodes.
- Conventional pattern routing and maze routing [DAC'24][7] are used to route the single-die nets.

[7] **Jinwei Liu and Evangeline FY Young (2023). "EDGE: Efficient DAG-based Global Routing Engine".** In: *Proc. DAC*. IEEE, pp. 1–6.

Experimental Results

Experimental Setup

- **Benchmarks:** 7 real-world designs from OpenCores and open-source RISC-V designs. 3D placement results are generated by a true 3D placer[8].
- **Metrics:** Wirelength after detailed routing, HBT count, and runtime.
- **Platform:** Intel Xeon Silver 4210R CPU (2.40GHz).

Table: The statistics of 7 evaluated real-world designs.

Bench.	#Cells	#Macros	#Nets	#3D Nets	ratio
rocket	24647	2	26084	1940	0.07
aes	13158	0	13158	381	0.03
ethmac	23714	0	23900	3487	0.15
jpeg	175352	0	205921	16 001	0.08
morlkx	56951	0	57521	2927	0.05
ariane	145684	132	157129	12 718	0.08
BP	273187	24	265585	3615	0.01

Table: The physical information of the technology node (NanGate 45nm PDK).

Name	Metal6	Via5	HBT	G-cell
pitch (μm)	0.28	0.3	3.0	2.1

[8] **Yuxuan Zhao et al. (2024). "Analytical Heterogeneous Die-to-Die 3D Placement with Macros".** In: *IEEE TCAD*.

Experimental Results

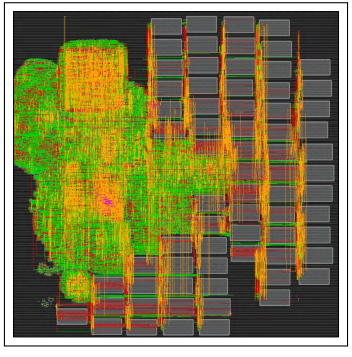
Table: Comparison of experimental results between our router and SOTA global routers.

Bench.	FastRoute [ASPDAC'09][9] + Legal [TCAD'24]				CUGR 2.0 [DAC'23] + Legal [TCAD'24]				Ours			
	WL	#Vias	#HBTs	RT	WL	#Vias	#HBTs	RT	WL	#Vias	#HBTs	RT
rocket	374 421	200 585	2800	9.163	369 069	210 891	2581	4.370	**336830**	205 913	**2179**	**2.441**
aes	156 045	119 885	703	4.996	157 048	125 413	831	2.466	**151657**	123 480	**526**	**1.335**
ethmac	557 884	221 480	5912	13.436	545 503	238 256	5738	6.338	**439322**	223 486	**4954**	**2.975**
jpeg	3 056 841	1 402 528	25 552	64.726	2 679 224	1 441 034	22 064	33.901	**2514129**	1 413 384	**19507**	**18.648**
mor1kx	1 516 476	559 389	10 443	26.012	1 380 750	577 274	9413	14.769	**1330973**	562 741	**7664**	**7.684**
ariane	3 684 778	1 377 210	25 837	72.921	3 506 201	1 459 846	25 163	38.048	**3142161**	1 396 919	**19504**	**18.736**
BP	6 401 377	2 018 460	34 430	113.263	5 634 172	2 094 227	28 428	72.662	**4700736**	1 986 244	**24315**	**35.960**
Average	1.186	**0.989**	1.318	3.701	1.116	1.036	1.249	1.937	**1.000**	1.000	**1.000**	**1.000**

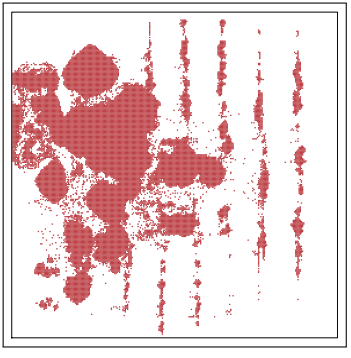
- **12%** shorter wirelength, **25%** fewer HBTs, and **1.9**× speedup compared to the baseline.

[9] **Yue Xu, Yanheng Zhang, and Chris Chu (2009). "FastRoute 4.0: Global router with efficient via minimization".** In: *Proc. ASPDAC*. IEEE, pp. 576–581.

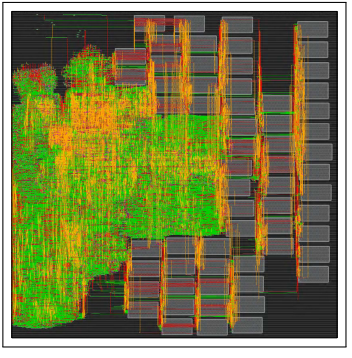
Routing Results Visualization



(a) Top Die



(b) Hybrid Bonding Terminals



(c) Bottom Die

The layouts for top, HBT, and bottom dies of `ariane`.

Impact of HBT Pitch Size

Table: Comparison of different HBT pitch sizes.

Bench.	Ours-2 µm				Ours-3 µm			
	WL	#Vias	#HBTs	RT	WL	#Vias	#HBTs	RT
rocket	**317907**	204 640	2275	2.427	336 830	205 913	**2179**	2.441
aes	**150940**	123 640	628	1.398	151 657	123 480	**526**	1.335
ethmac	**382196**	221 734	5563	2.666	439 322	223 486	**4954**	2.975
jpeg	**2442417**	1 413 078	20 297	18.067	2 514 129	1 413 384	**19507**	18.648
morlkx	**1315771**	561 775	8030	7.708	1 330 973	562 741	**7664**	7.684
ariane	**3037041**	1 394 232	20 739	18.645	3 142 161	1 396 919	**19504**	18.736
BP	**4300681**	1 974 723	27 324	35.543	4 700 736	1 986 244	**24315**	35.960
Average	**0.950**	**0.997**	1.091	**0.985**	1.000	1.000	**1.000**	1.000

- With the 2 µm pitch, H3D achieved a 5% improvement in wirelength while using 9% more HBTs.

Q&A