

# HBM Roadmap Ver 1.7 Workshop by KAIST TERALAB

June 11<sup>th</sup>, 2025

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory

School of Electrical Engineering, KAIST

<https://tera.kaist.ac.kr/>

# HBM Roadmap Ver 1.7 Workshop Agenda [1/2]

	순번	Contents	발표자
Intro	1	Overview of HBM Roadmap Ver. 1.7 by KAIST Teralab	김정호 교수
HBM4/5	2	HBM5-LPDDR Architecture with Customized Base Die	윤지원
	3	Design of 3D Near Memory Computing Architecture in HBM5 for High Performance and Power Efficient Computing	윤지원
	4	Hybrid Vision Transformer Based Chip Design Agent for Fast Estimation of Multi-layer and Multi-power PDN Impedance in Customized Base Die in HBM5	안현준
	5	Transformer-based Reinforcement Learning for TSV Placement and Design Optimization considering IR Drop in HBM5	서은지
	6	Mamba-Reinforcement Learning-based HBM5 Design Agent for Fast PDN Optimization considering Power Integrity	김병목
	7	Devformer with Collaborative Distillation for Optimal Decoupling Capacitor Placement in HBM5 Custom Base Die	김혜연
	8	Reinforcement Learning-based Decap Placement Optimization considering Diverse I/O Channel Interfaces in Custom Base Die of HBM5 Memory Pooling Architecture	박준호
	9	Power Supply Noise Induced Jitter (PSIJ) Modeling and Reinforcement-Learning based PI Optimization for HBM5 I/O Interface	신태인

# HBM Roadmap Ver 1.7 Workshop Agenda [2/2]

	순번	Contents	발표자
HBM6	10	Quad-Tower (QT)-HBM6 Architecture for High-Throughput and Low-Latency Inference with Signal Integrity Considerations	김태수
	11	Large-Scale Hybrid Interposer for Multi-Tower HBM6 Architecture	서해석
	12	L3 Cache Embedded (L3E) HBM6 Architecture for LLM Inference	서해석
	13	HBM6 Cluster Architecture with Crossbar Network Switch for High Throughput and Low Latency LLM Inference	윤영수
	14	HBM6-Centric Network Design under Traffic Asymmetry in Heterogeneous HBM Module based Systems	안효원
	15	Conditional Diffusion Model-based Imitation Learning for Placement and Interconnection Optimization for HBM6	김지훈
	16	Generative Adversarial Learning-Based Power Noise Induced Eye Diagram Estimation Agent for HBM6	이정현
HBM7/8	17	HBM-HBF with Storage Network Architecture	안현준
	18	NMC-HBM with HBF for Large-Scale AI Inference	이현이
	19	HBM7 Architecture Integrated with High-Capacity 3D Stacked LPDDR	최인영
	20	Embedded Cooling Structure for HBM7 Architecture	손기영
	21	3D Memory Expansion Architecture for HBM8 with Double-Sided Interposer and HBM-HBF-LPDDR Integration	양채민
	22	AI Design Agent for 3D Placement and Routing Optimization for HBM8 using Reinforcement Learning considering Thermal-Signal Integrity	엄현서
	23	LLM-aided Interactive Reinforcement Learning (IRL) with Switch Transformer for PSIJ Reduction in HBM7	배재근
	24	LLM-based HBM7 Design Agent using Interactive Reinforcement Learning (IRL) for Decoupling Capacitor Placement	김근우

# [HBM Roadmap ver 1.7 by KAIST Teralab] Overview of Next Generation HBM Architectures

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

2025. 06. 11



- Basics of HBM
- Overview of HBM Roadmap
- Next-Generation HBM Roadmap by KAIST TERALAB
  - ✓ HBM4 2026
  - ✓ HBM5 2029
  - ✓ HBM6 2032
  - ✓ HBM7 2035
  - ✓ HBM8 2038

# Part1: Basics HBM

# DALL·E 2: OpenAI's Image Generation AI System

Text Prompt

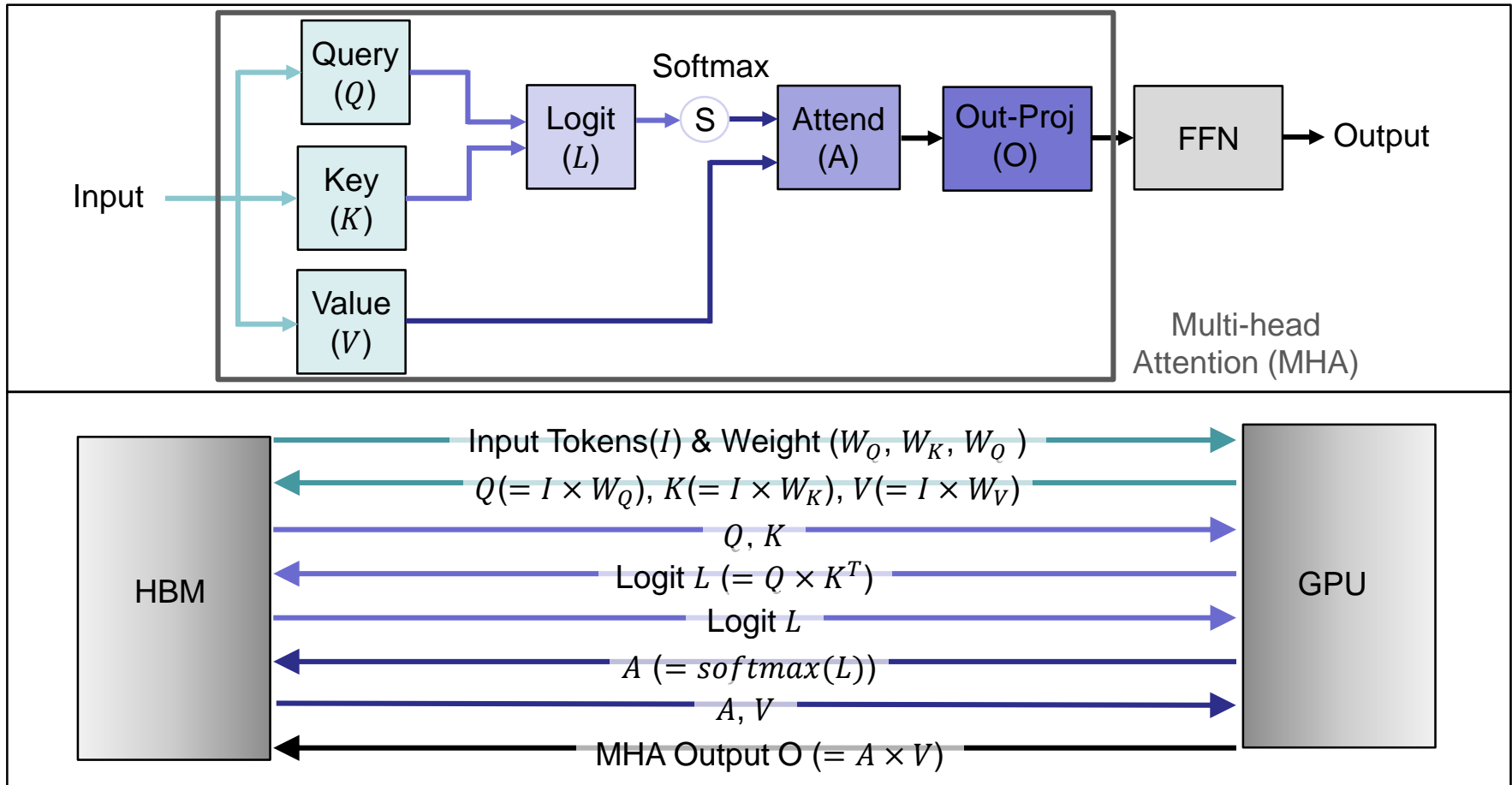
Vibrant portrait painting of Salvador Dalí with a robotic half face.



< Salvador Dalí image generated by DALL·E 2 >

- DALL·E 2 is an AI system that can create realistic images and art from a description in natural language.
- OpenAI released DALL·E 2 on April 6, 2022.
- The parameter of DALL·E 2 is 3.5 billion, which is 1/4 of the previous version.

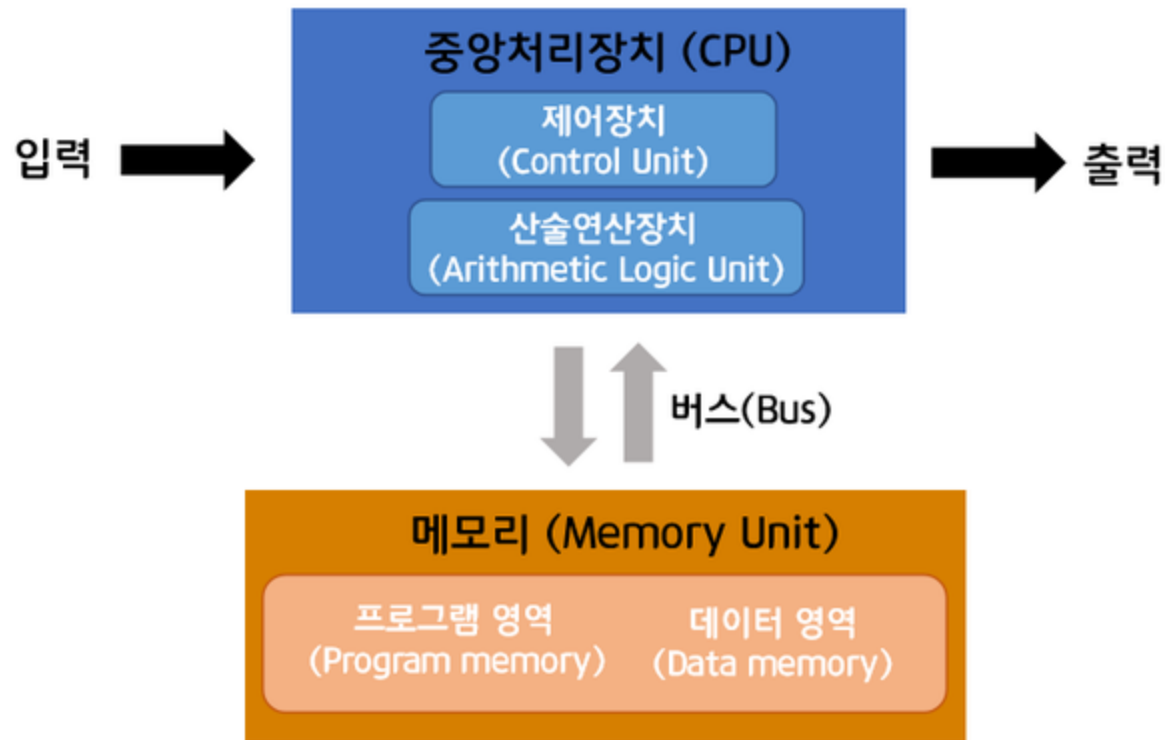
# Massive Data Movement between GPU and HBM during Transformer-based AI Computing



< Structure of MHA and FFN in Transformer and Conceptual Data Movement between GPU and HBM >

- Transformer model consists of an encoder and decoder, which are composed of two main components: multi-head attention (MHA) and feed forward network (FFN).
- During these processes, significant data movement occurs between GPU and HBM.

# Von Neumann Architecture

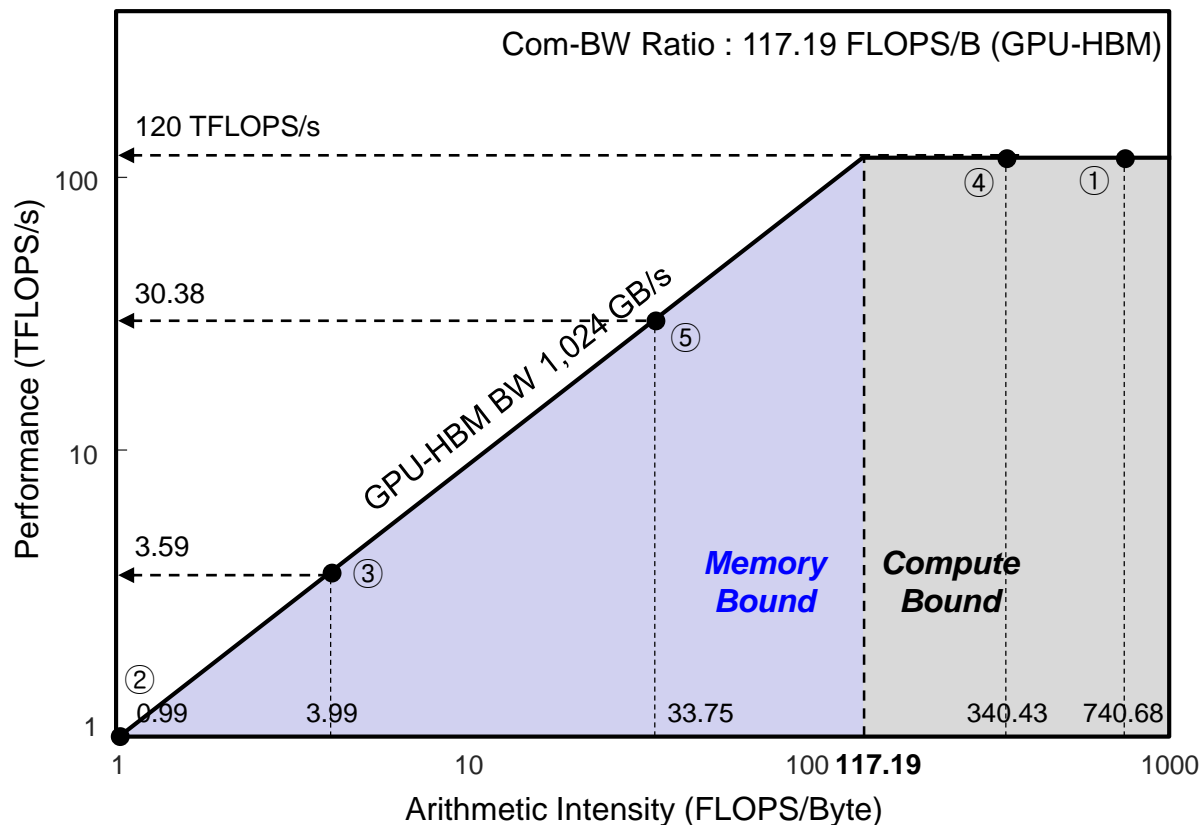


# Performance Evaluation of GEMM Workloads with GPU-HBM Roofline Model

$$\text{Arithmetic Intensity} = \frac{M \cdot N \cdot K}{M \cdot K + N \cdot K + M \cdot N}$$

$$\text{Com-BW Ratio} = \frac{\text{Peak Performance}}{\text{Bandwidth}}$$

GEMM	Arithmetic Intensity [FLOPS/Byte]	Achieved Performance [TFLOPS/s]
①	740.68	120
②	0.99	0.89
③	3.99	3.59
④	340.43	120
⑤	33.75	30.38

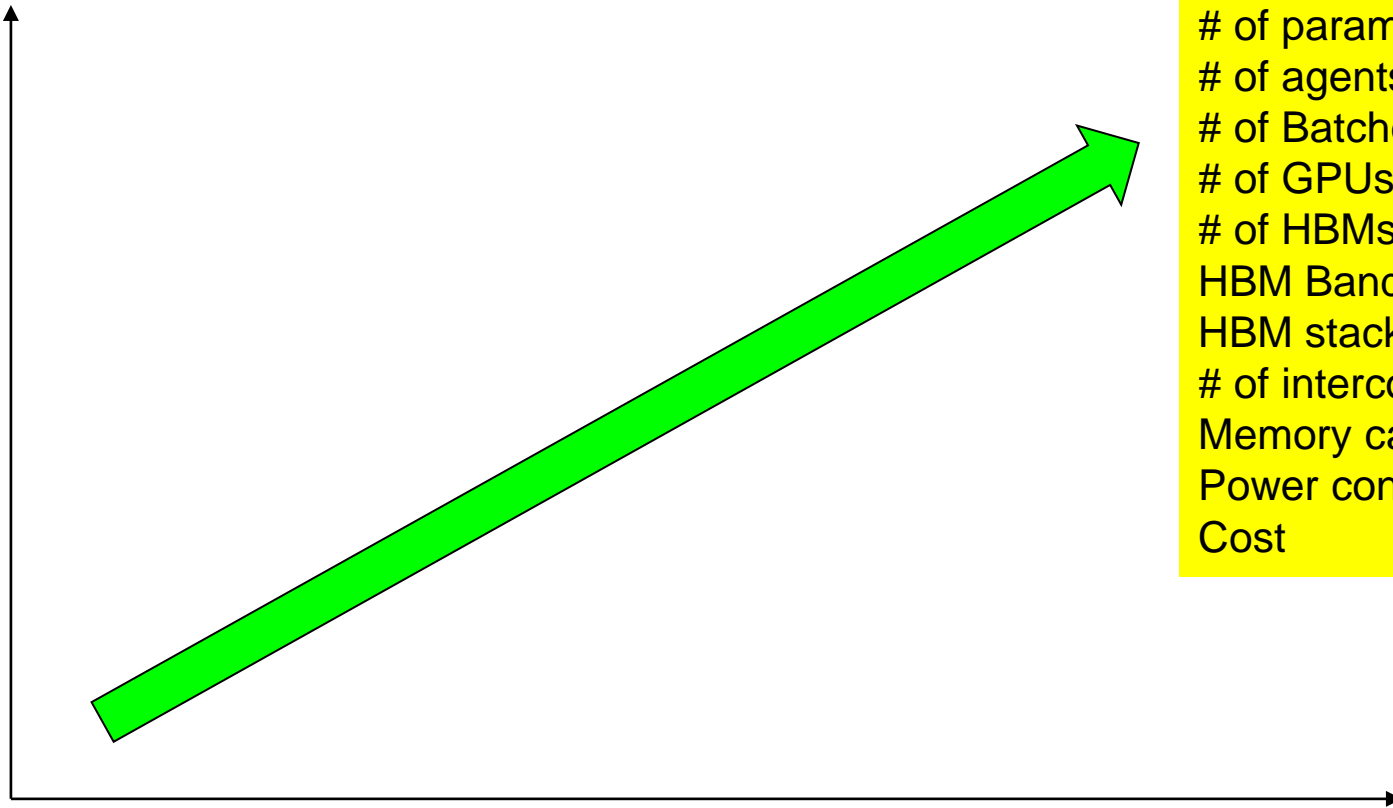


## < Roofline Model of Simulated GEMM Workloads >

- Assuming the GPU-HBM architecture meets the system's peak computation capabilities, the roofline model of GPU-HBM and GEMM workloads was plotted based on V100.
  - ✓ GEMM ①, ④ : Arithmetic Intensity > 117.19 FLOPS/B → **Compute-bound**
  - ✓ GEMM ②, ③, ⑤ : Arithmetic Intensity < 117.19 FLOPS/B → **Memory-bound**

# Scaling 법칙은 계속된다

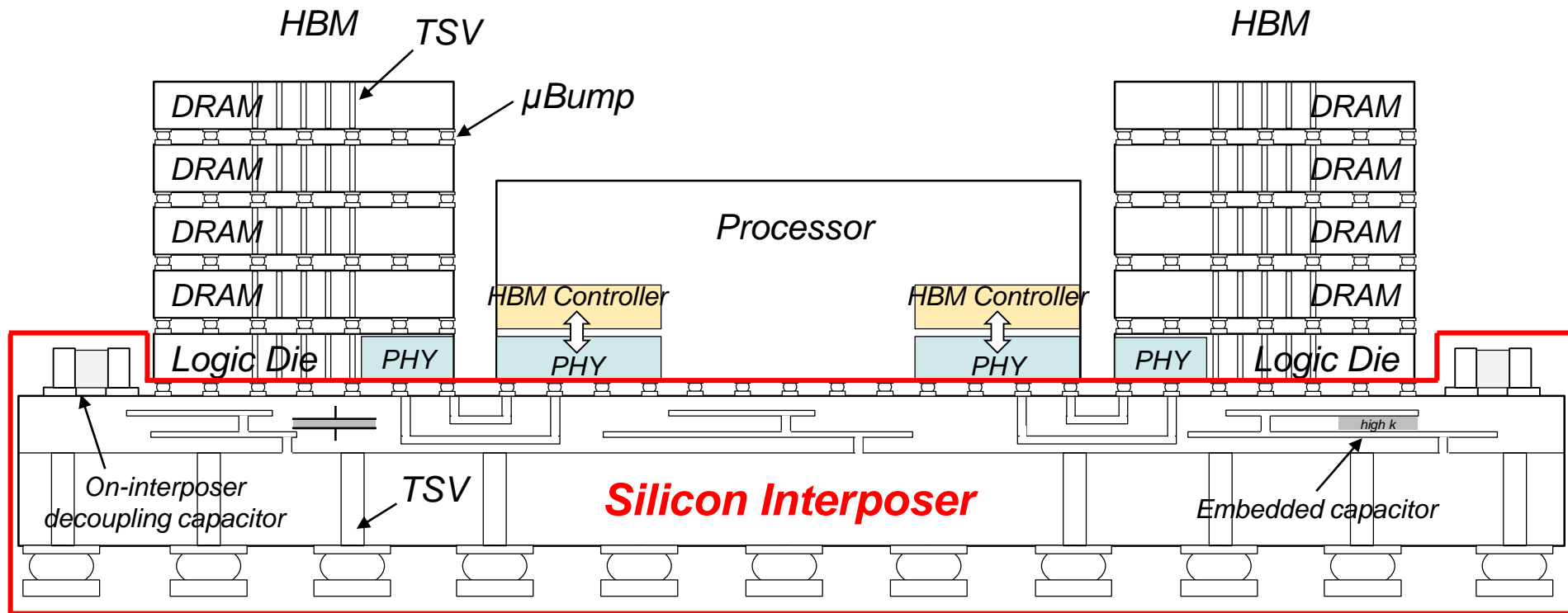
Size



Model Size  
# of parameters  
# of agents  
# of Batches  
# of GPUs  
# of HBMs  
HBM Bandwidth  
HBM stacks  
# of interconnections  
Memory capacity  
Power consumptions  
Cost

Year

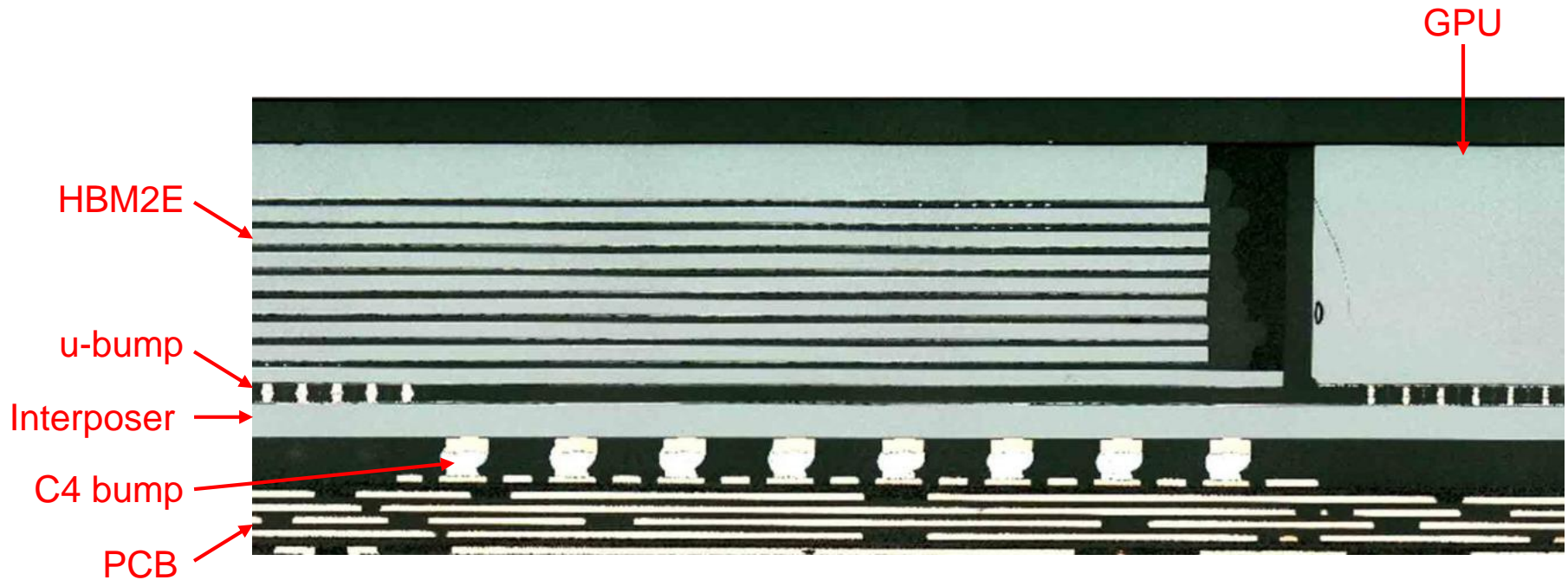
# HBM Interposer Design: Cross-sectional View



< Cross-sectional View of Silicon Interposer based HBM Module >



# Cross-section of NVIDIA A100



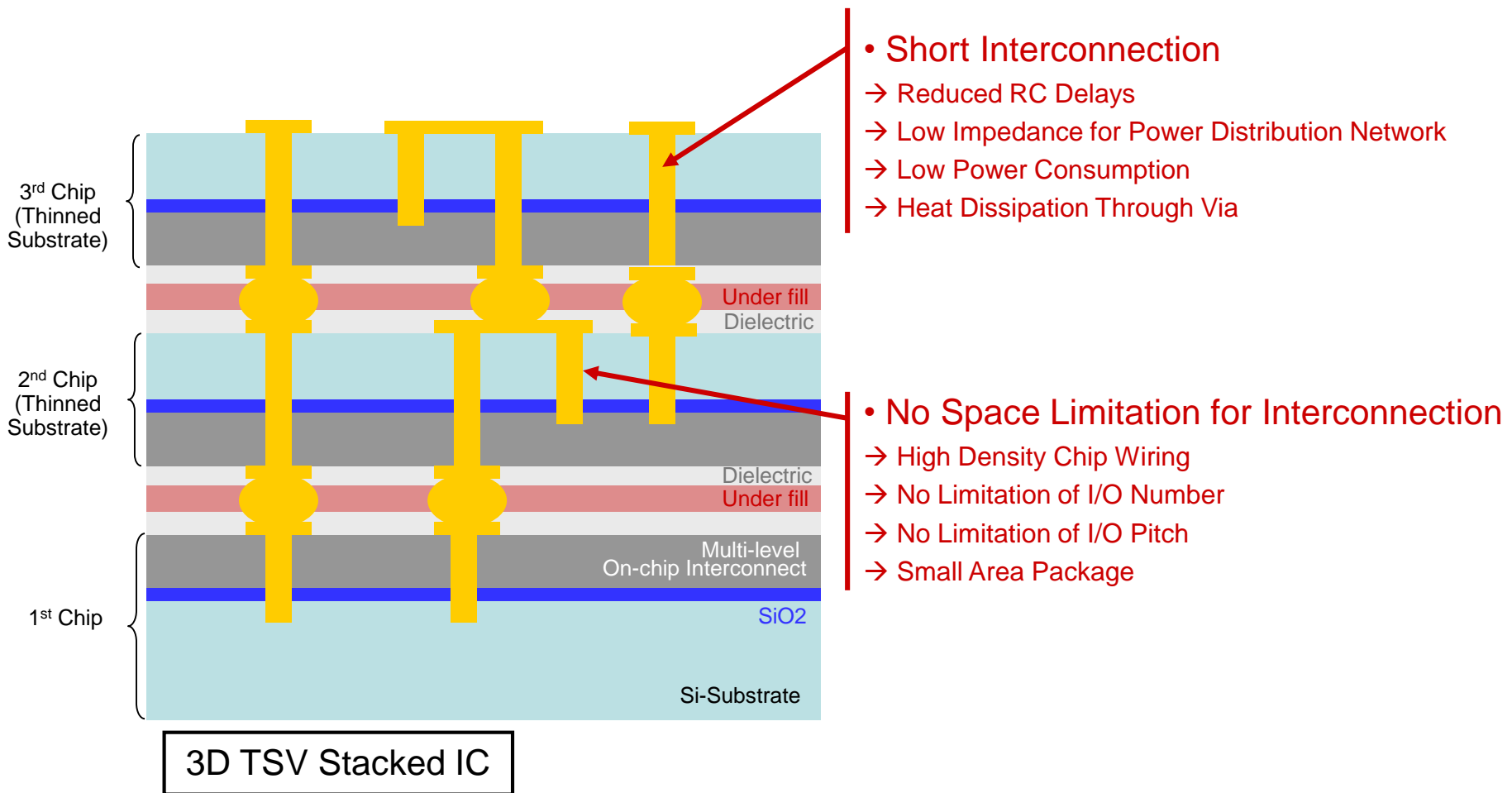
< Cross-section of NVIDIA A100 >

- HBM2E와 GPU를 연결하기 위해서 Interposer를 사용하는 CoWoS 기술이 적용되었음.

# 주상복합 건물



# Key Technology : TSV (Through Silicon Via)

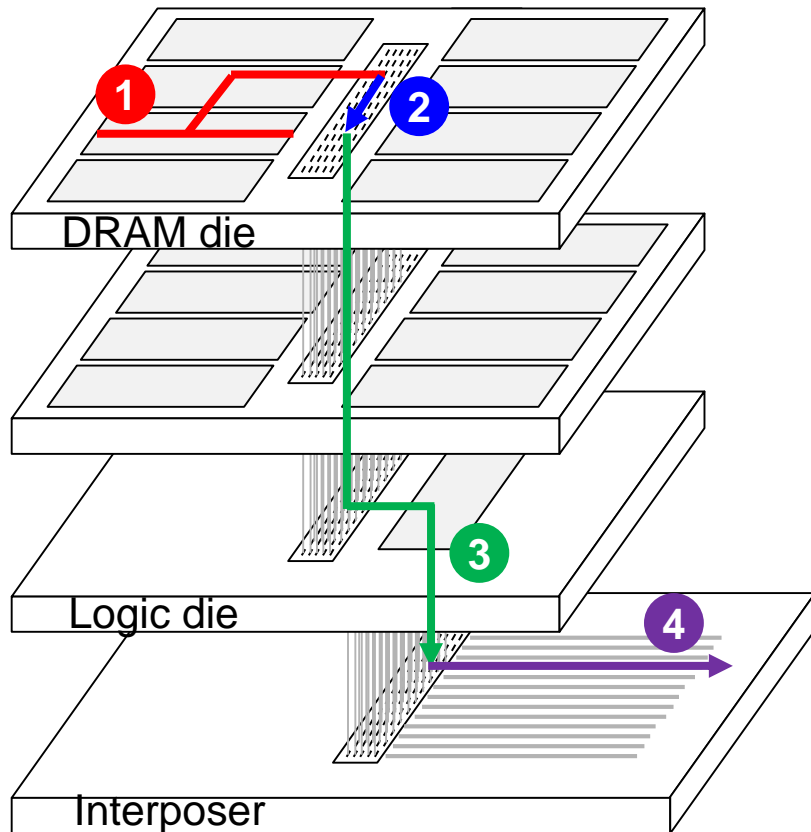


# TSV Elevator





# DRAM Bandwidth Teardown in HBM3



< Data path in HBM3 >

Data path	Configuration	Bandwidth
Bank group (on-chip)	16384 I/O x 0.4375 Gbps (4nCK)	896 GB/s
Global (on-chip)	8192 I/O x 0.875 Gbps (2nCK)	896 GB/s
Die-to-die (TSV)	4096 I/O x 1.75 Gbps (1nCK)	896 GB/s
Logic die (on-chip)	4096 I/O x 1.75 Gbps (1nCK)	896 GB/s
Interface (interposer)	1024 I/O x 7 Gbps (0.25 nCK)	896 GB/s

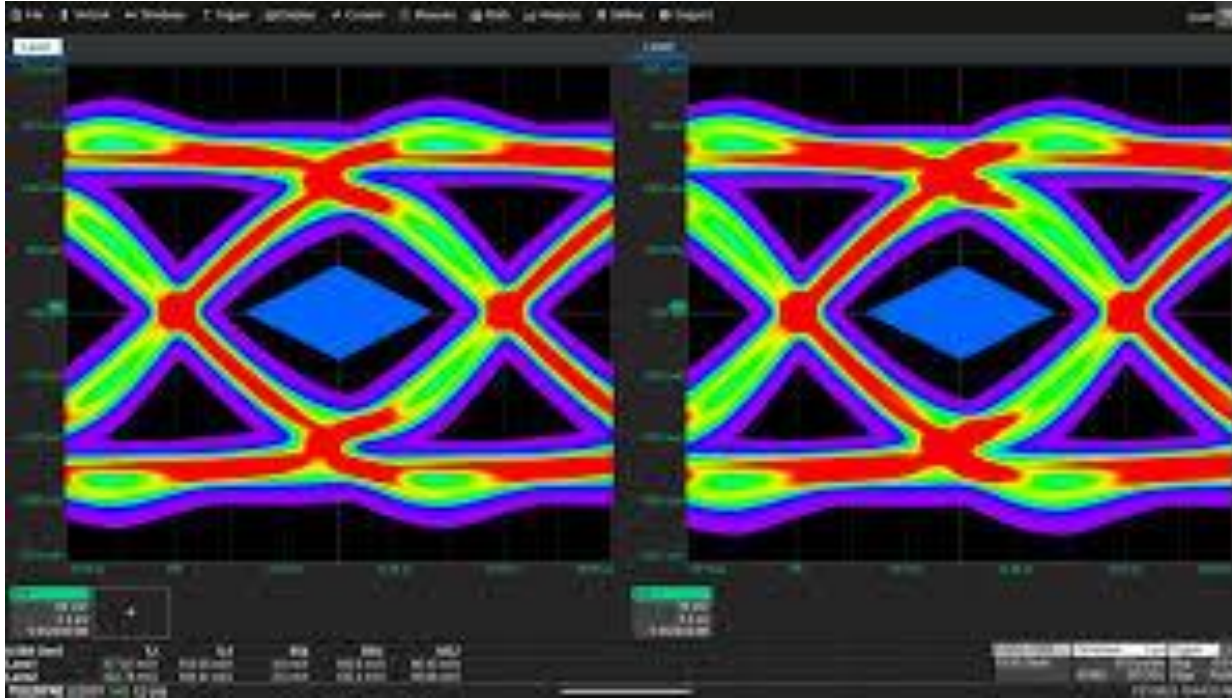
< DRAM bandwidth of each data path in HBM3 >

- From bank group I/O to interface I/O, the number of I/O and data rate are hierarchical.
- I/O bus window is extended by time-division multiplexing.

# 인천 공항



# 시간의 불확실성: ultimate bandwidth limit



Deterministic Jitter  
Random jitter  
ISI jitter  
PDN jitter  
PSIJ

# Interconnection Future(10): Design Innovations: 5 Is

## Design and Analysis for High-performance System

Co-Simul. Co-Design

System on Chip  
(On-Chip Level)

System in Package  
(On-Package Level)

PCB, Cable, Module  
(System Level)



Signal **I**ntegrity  
(**SI**)

Power  
**I**ntegrity  
(**PI**)

Thermal  
**I**ntegrity  
(**TI**)

**5I**  
for High-  
Performance  
System

Electromagnetic  
**I**nterference  
(**EMI**)

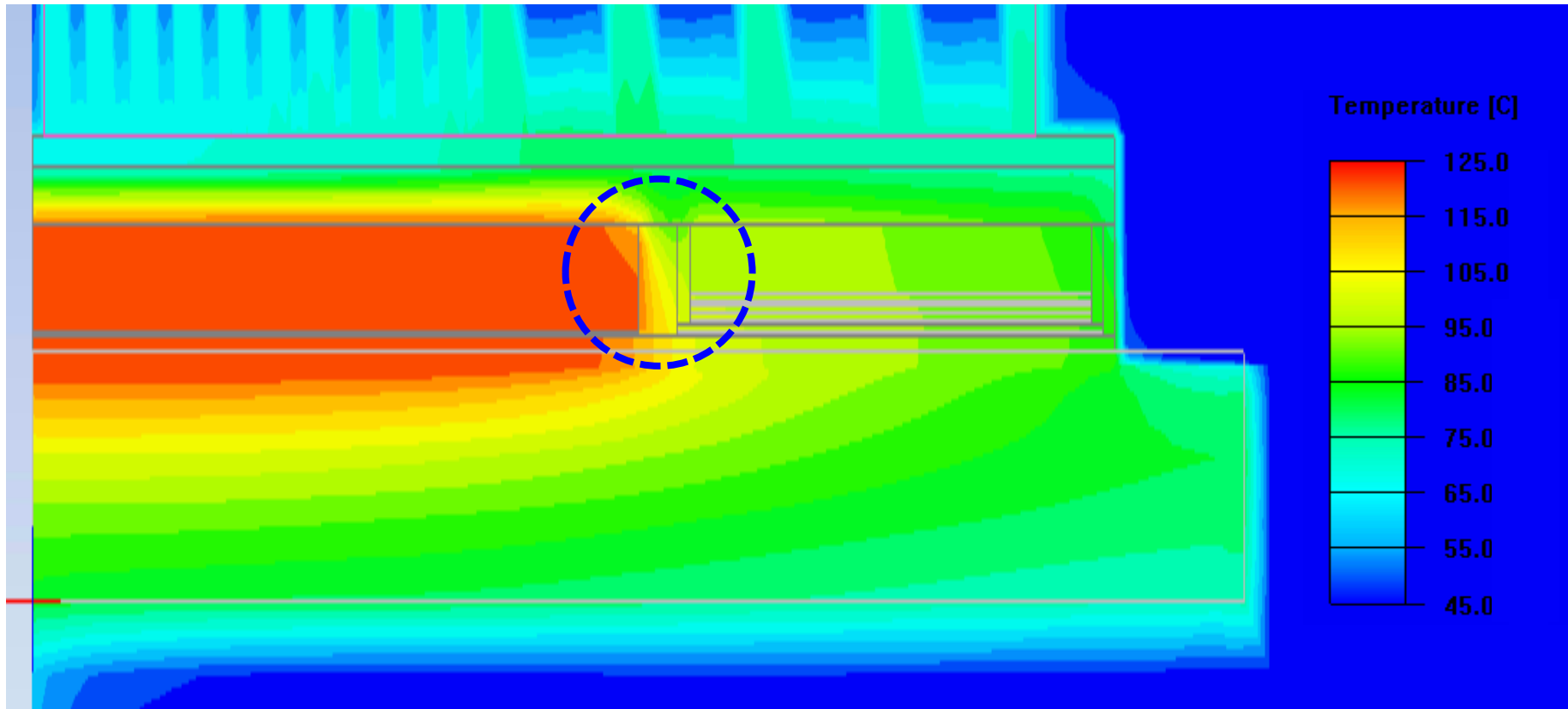
Artificial  
**I**ntelligence  
(**AI**)



Improvement of System Integrity, Performance, Reliability, Cost, ...

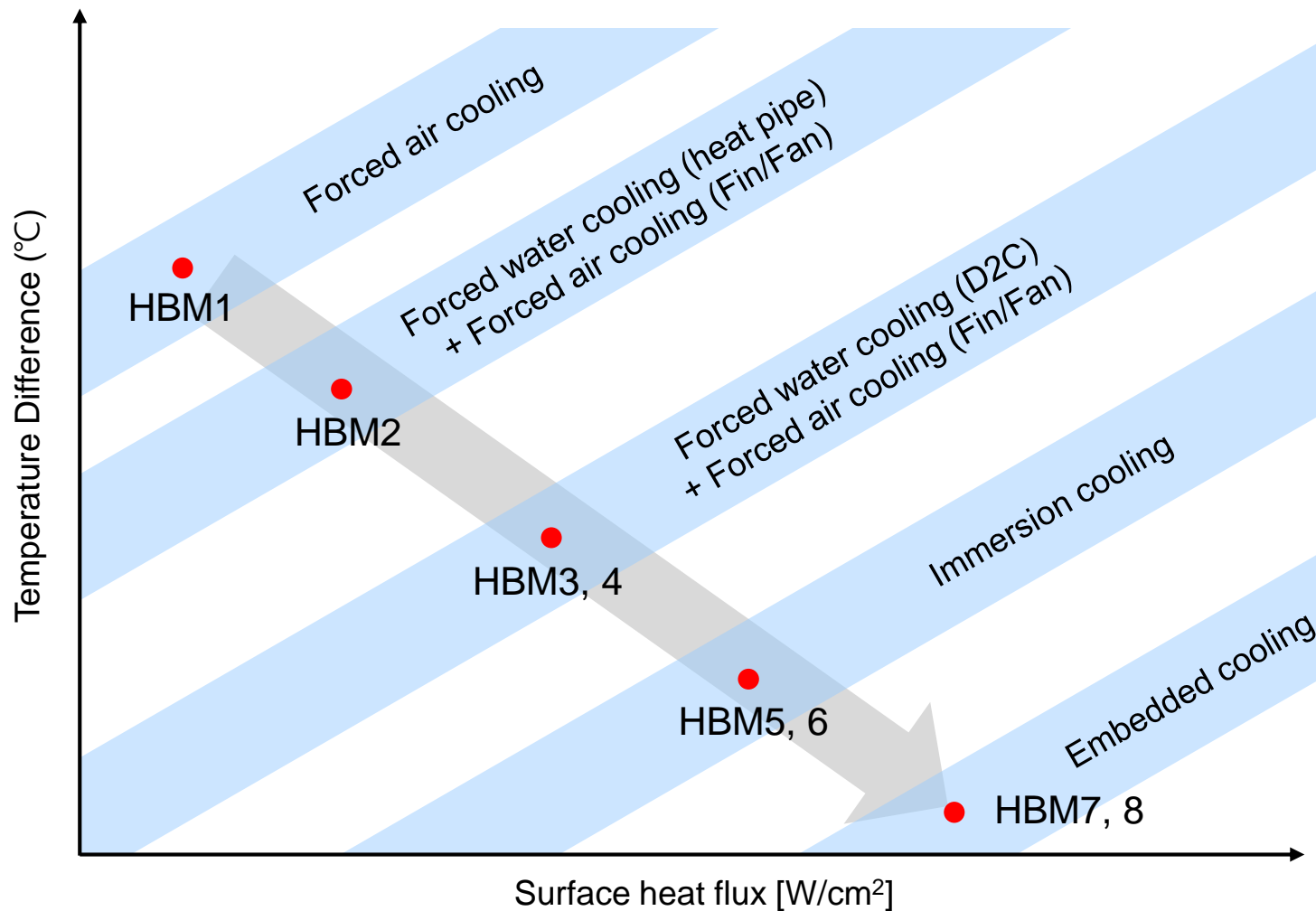


# Thermal Coupling on Operating HBM System



<Thermal distributions of operating HBM system>

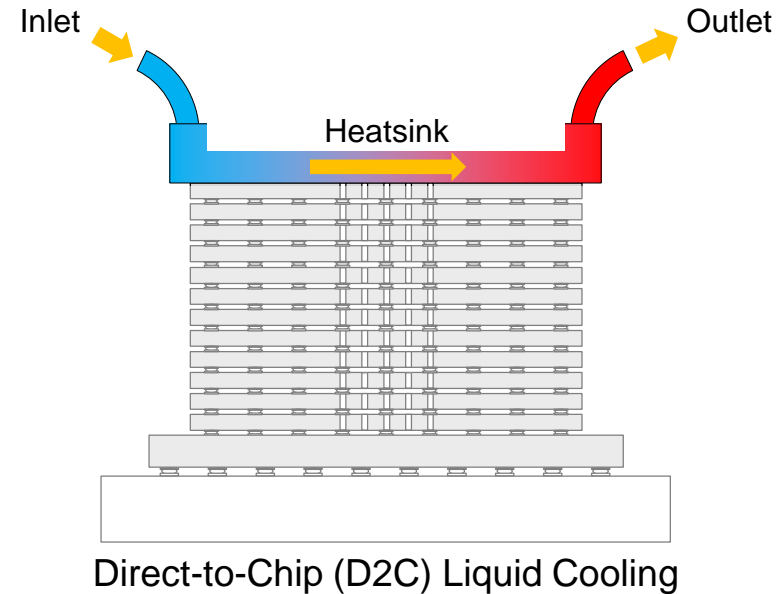
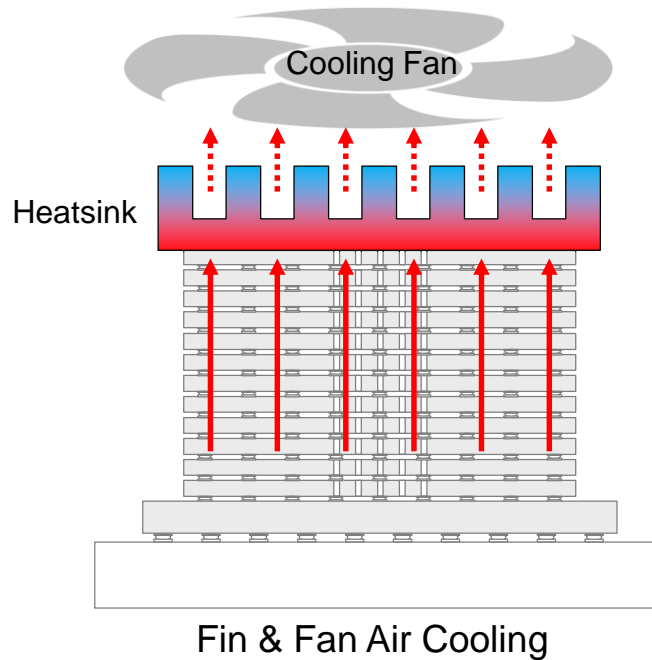
- GPU's heat thermal coupling to HBM through molding compound, heat sink, interposer and etc.
- GPU's heat is dominantly affected to HBM's thermal gradient.
- Only molding compound part can be customized for reducing thermal coupling effects.



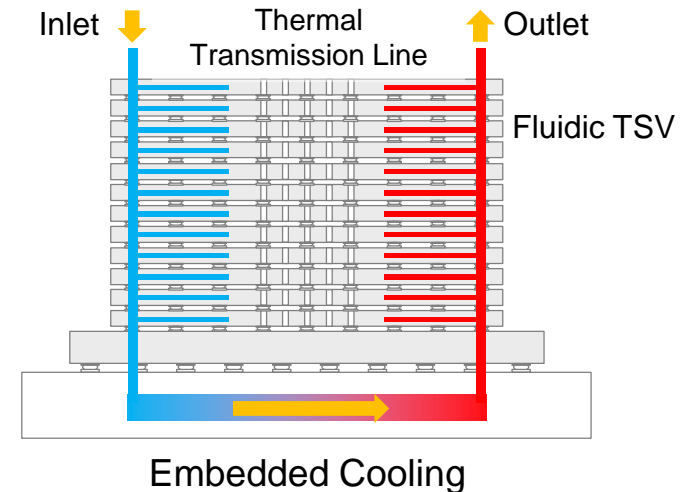
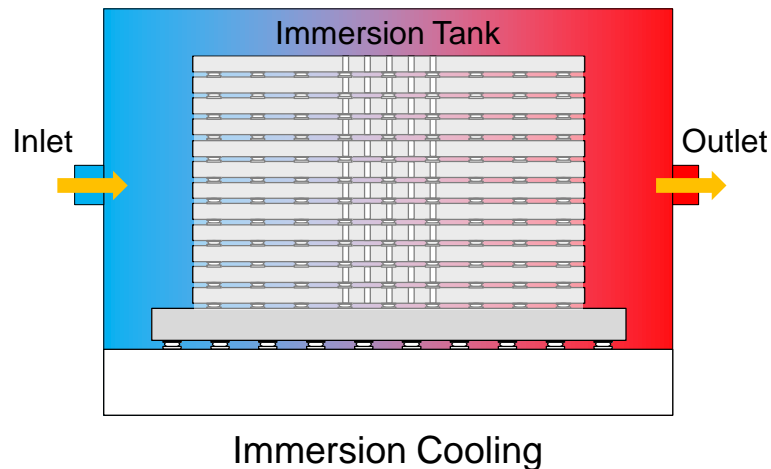
< Thermal Management & Cooling Methods for Next-Generation HBM >

# Next-Generation HBM Roadmap : Thermal Management & Cooling Method

*(Present)  
Conventional  
HBM Cooling  
Method*



*Next-Generation  
HBM Cooling  
Method*



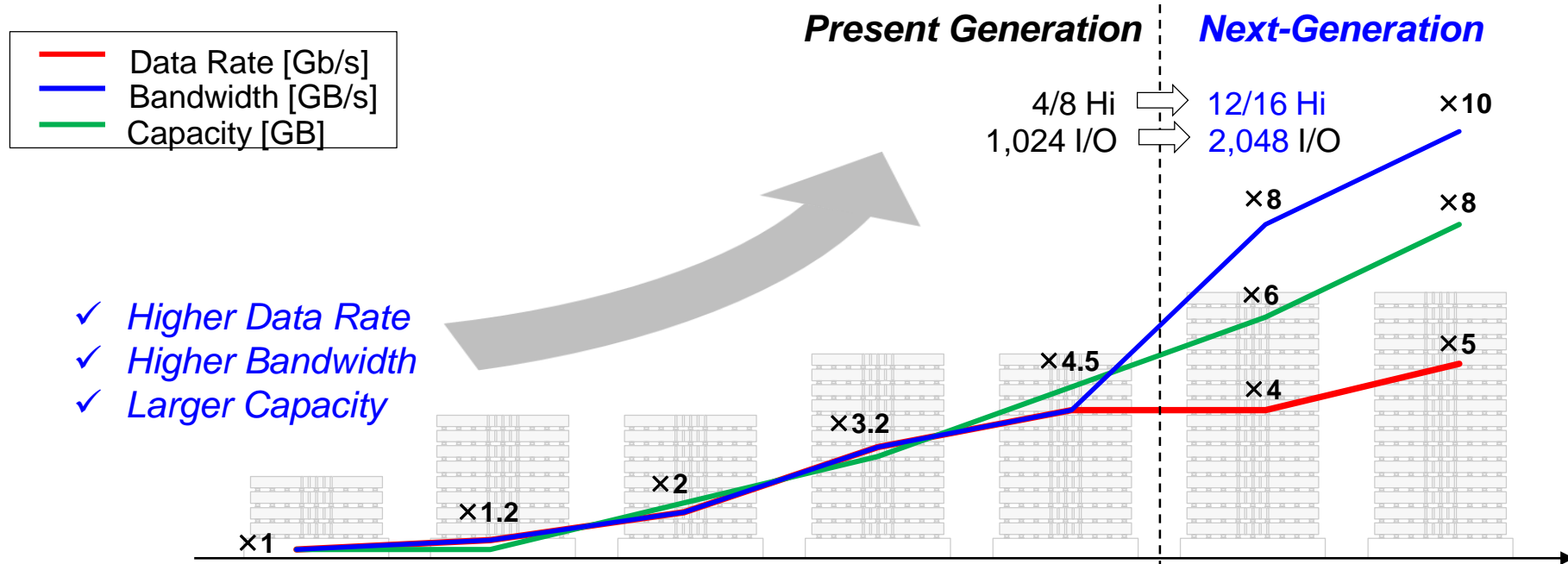
## Part2: Overview of HBM Roadmap

- Higher Bandwidth
  - ✓ Larger Number of Interconnects
  - ✓ Higher Gbps/line
  - ✓ Larger Number of TSV
  - ✓ Hybrid bonding, Narrow pitch,
- Higher Memory Capacity
  - ✓ Integration with LPDDR-HBM, HBF
  - ✓ Hierarchical Memory Architecture
  - ✓ Integrated Memory network, CXL
- Computing in HBM
  - ✓ Near Memory Computing
  - ✓ Streaming Multi-processor
  - ✓ Data Compression
  - ✓ Error Corrections

- 3D Integration
  - ✓ Stacked Cash
  - ✓ Active/Embedded Interposer
  - ✓ Hybrid interposer (Si/Glass)
- Innovative cooling architecture
  - ✓ Liquid cooling
  - ✓ Immersion Cooling
  - ✓ Embedded cooling
- HBM Centric Computing
  - ✓ Full 3D Integration
  - ✓ CPU, GPU, HBM, HBF integration
  - ✓ Instruction set, programming

# Technical Trend of AI-Specialized HBM in AI Semiconductor Industry

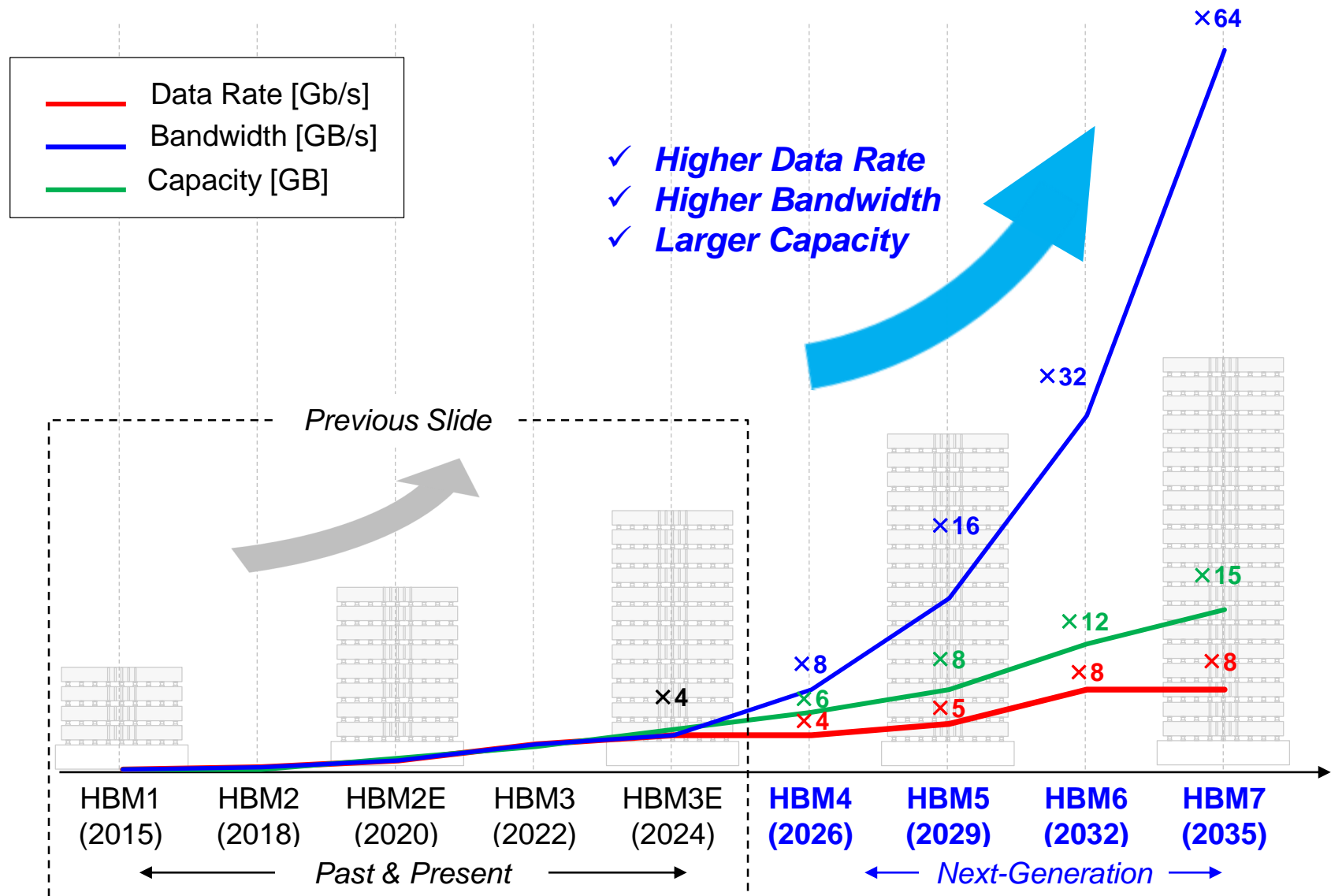
## : Increasing Bandwidth, Data Rate, Capacity



	HBM (2015)	HBM2 (2018)	HBM2E (2020)	HBM3 (2022)	HBM3E (2024)	HBM4 (2026)	HBM4E (*2028)
Data Rate	2 Gbps	2.4 Gbps	3.6 Gbps	6.4 Gbps	8 Gbps	8 Gbps	10 Gbps
# of I/O	1,024					2,048	
Bandwidth	256 GB/s	307 GB/s	461 GB/s	819 GB/s	1.0 TB/s	2.0 TB/s	2.5 TB/s
Capacity/die	8 Gb		16 Gb		24 Gb		32 Gb
# of stack die	4/8-Hi			8/12-Hi		12/16-Hi	
Capacity/HBM	4/8 GB		8/16 GB	16/24 GB	24/36 GB	36/48 GB	48/64 GB
Power/HBM	4 W	10 W	19 W	25 W	32 W	43 / 75 W	48 / 80 W
Cooling Method	Thermo-Electric Cooling (TEC) w/ Heatsink		Direct-to-Chip (D2C) Liquid Cooling				

# Technical Trend of AI-Specialized HBM in AI Semiconductor Industry

## : Next Generation HBM Technology Trend by KAIST TERALAB





# Next-Generation HBM Roadmap by KAIST TERALAB

HBM Spec. Packaging/Cooling  
Architecture AI Design Agent

Ver 1.2 / updated.250521

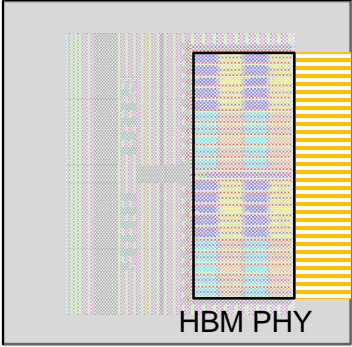
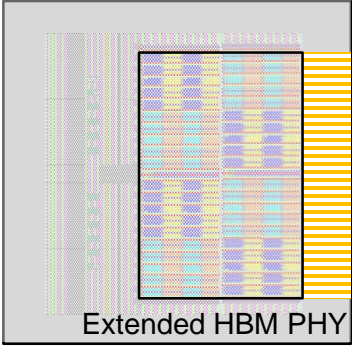
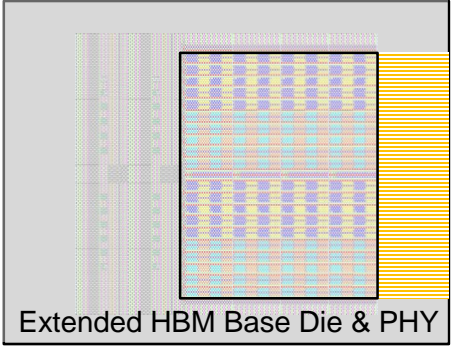
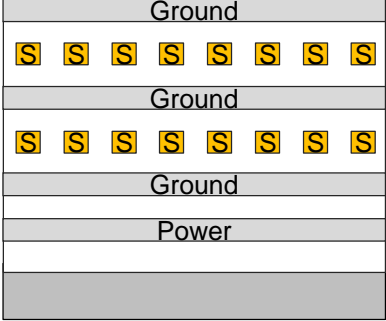
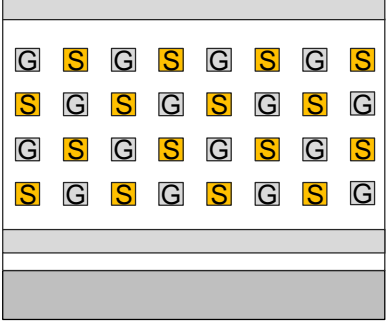

	HBM4 (2026)	HBM5 (2029)	HBM6 (2032)	HBM7 (2035)	HBM8 (2038)
Data Rate	8 Gbps	8 Gbps	16 Gbps	24 Gbps	32 Gbps
# of I/O	2,048	4,096	4,096	8,192	16,384
Bandwidth	2.0 TB/s	4 TB/s	8 TB/s	24 TB/s	64 TB/s
Capacity/die	24 Gb	40 Gb	48 Gb	64 Gb	80 Gb
# of die stack	12/16-Hi	16-Hi	16/20-Hi	20/24-Hi	20/24-Hi
Capacity /HBM	36/48 GB	80 GB	96/120 GB	160/192 GB	200/240 GB
Power/HBM	75 W	100 W	120 W	160 W	180 W
Die stacking	Microbump (MR-MUF)		Bump-less Cu-Cu Direct bonding		
Cooling Method	Direct-to-Chip (D2C) Liquid Cooling	Immersion Cooling		Embedded Cooling	
HBM Architecture	Custom HBM Base Die HBM-LPDDR	3D NMC-HBM & stacked cache / decap	Multi-tower HBM Active / Hybrid Interposer	Hybrid HBM Architecture HBM-HBF HBM-3D LPDDR	Full-3D / HBM Centric Computing Architecture
Additional Features (Patent)	NMC processor + LPDDR Ctrl	+ Cache + CXL + on-die/stacked decap + HBM shielding	+ Network switch + Bridge die + Asymmetric TSV	+ HBF/LPDDR Ctrl + Storage network	+ HBM Centric Interposer + Double side Cooling + Edge-expand Stack
AI Design Agent	ubump & TSV-array Decap placement Optimization	I/O Interface Optimization considering PSIJ	Hybrid Equalizer + Generative AI based SI/PI Metric Estimation	LLM based Human Interactive AI Design Agent	

# Next-Generation GPU-HBM Roadmap

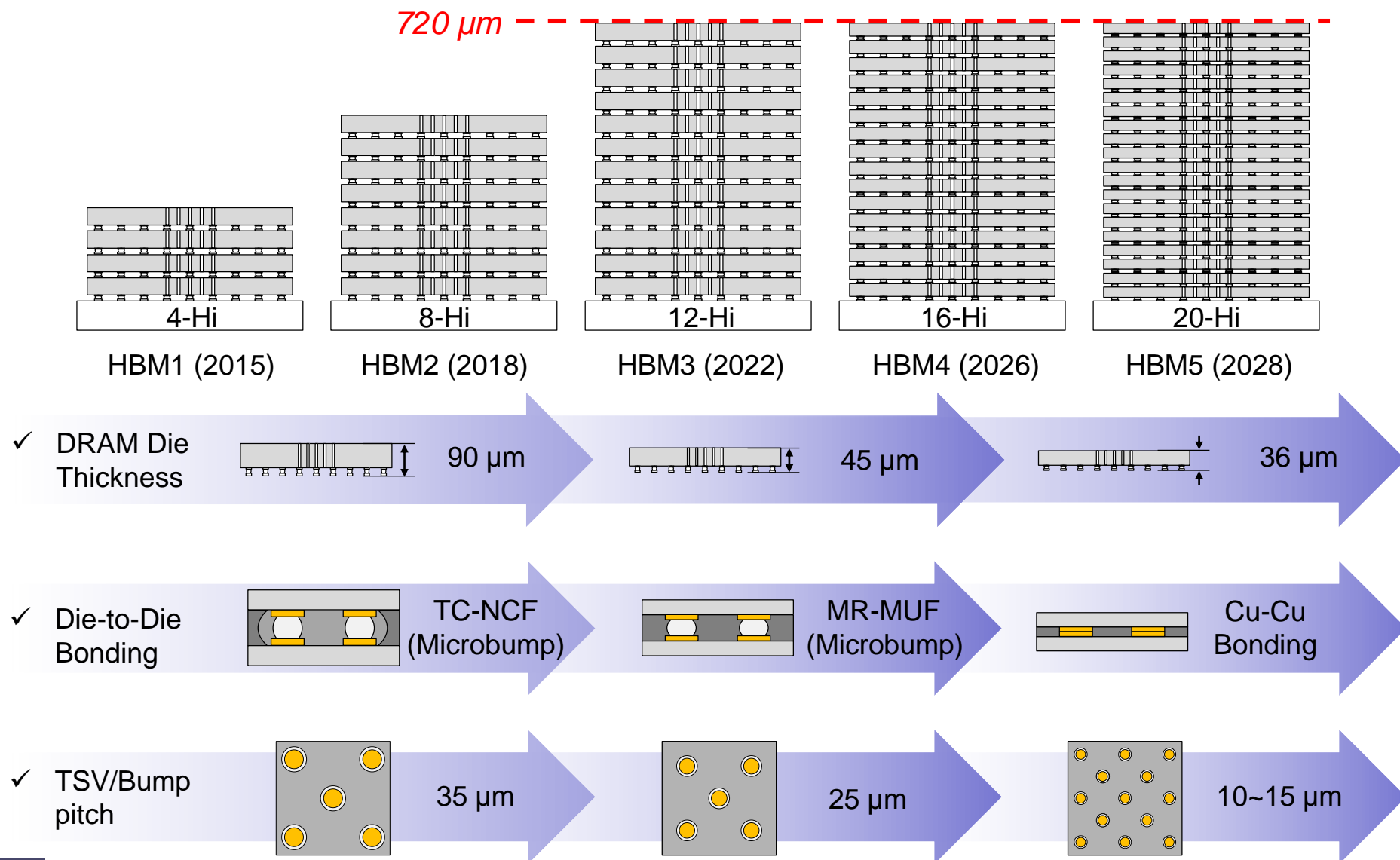
## : More GPU & HBM Integrated Above Interposer

GPU Architecture	Rubin (2026)	Feynman (2029)	Post Feynman (2032)	Next-Gen Architecture (2035)
GPU Die Size	728 mm <sup>2</sup>	750 mm <sup>2</sup>	700 mm <sup>2</sup>	600 mm <sup>2</sup>
GPU Power	800 W	900 W	1,000 W	1,200 W
GPU-HBM Module	R200	F400	Post Feynman GPU-HBM Module	Next-Gen GPU-HBM Module
Interposer Size	<p>47.5 mm 46.2 mm</p>	<p>56.2 mm 85.2 mm</p>	<p>58.5 mm 102.8 mm</p>	<p>95.9 mm 96.4 mm</p>
# of GPU Dies	×2	×4	×4	×8
# of HBM Stack	HBM4×8	HBM5×8	HBM6×16	HBM7×32
Interposer Die Size	2,194 mm <sup>2</sup> (46.2 mm x 48.5 mm)	4,788 mm <sup>2</sup> (85.2 mm x 56.2 mm)	6,014 mm <sup>2</sup> (102.8 mm x 58.5 mm)	9,245 mm <sup>2</sup> (96.4 mm x 95.9 mm)
Total Bandwidth	16 / 32 TB/s	48 TB/s	128/256 TB/s	1,024 TB/s
Total HBM Capacity	288/384 GB	400/500 GB	1,536/1,920 GB	5,120/6,144 GB
Total Power	2,200 W	4,400 W	5,920 W	15,360 W

## : Increased I/O Channels for Bandwidth Extension

	HBM3 (2022)	HBM4 (2024)	Next-Generation HBM
I/O #	1,024	2,048	4,096 ~ 8,192
HBM PHY Ball map			
microbump pitch	55 um	45 um	40 um
Interposer Channel Cross Section			
Metal w/s	2um ~ 3um	2um	< 2um
# RDL	Microstrip + Stripline (x2)	GSG Interleaved (x4)	GSG Interleaved (x6 ~ x8)

# Next-Generation HBM Roadmap : 3D Integration with Advanced Packaging Technology



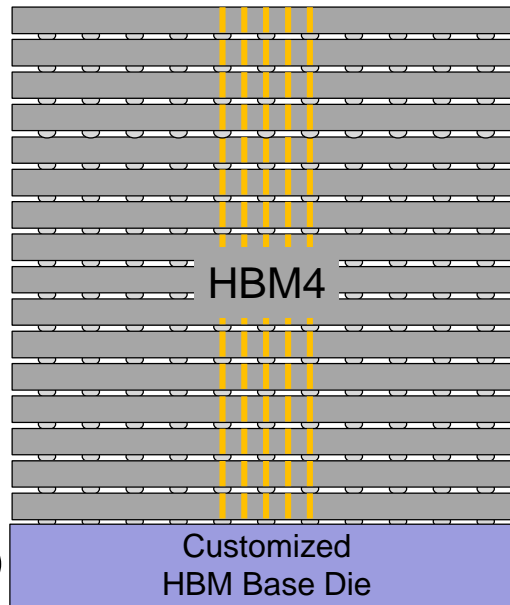
TSV(실리콘 관통전극)



1

HBM 적층 수, 저장 용량 증가  
8 ~ 12 단 → 12 ~ 16 단  
16 ~ 24 GB → 36 ~ 48 GB

1



HBM4

GPU

Customized  
HBM Base Die

2

2

고객 맞춤형 (Customized)  
Base Die 설계  
일부 GPU 계산 기능이 이동

3

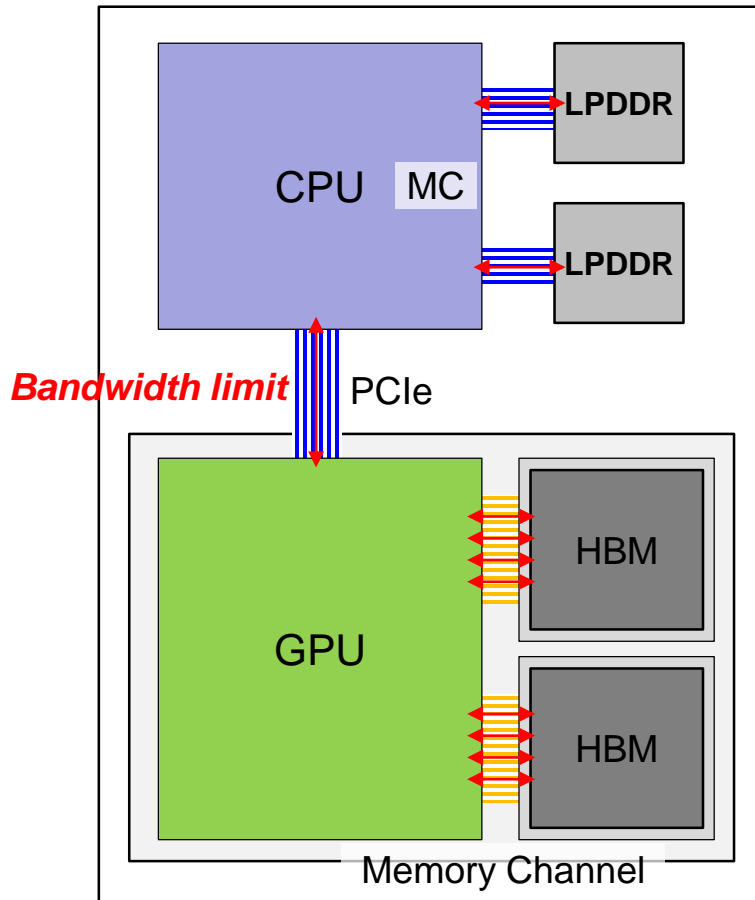
3

HBM3/3E 대비 2배 늘어난  
HBM-GPU 연결선 (I/O)  
2,048 개

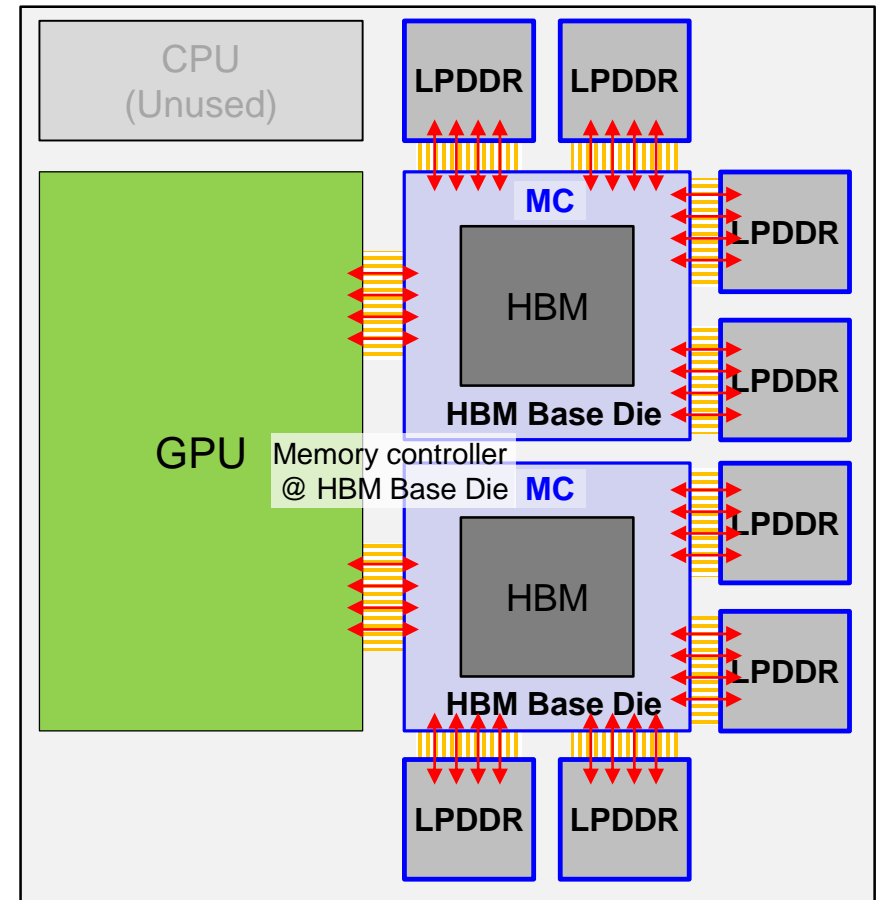
실리콘 인터포저 (기판)

# Custom HBM Base Die Design : LPDDR Memory Channel for High Capacity & Memory Bandwidth

↔ : Low-Bandwidth      ↔↔↔ : High-Bandwidth

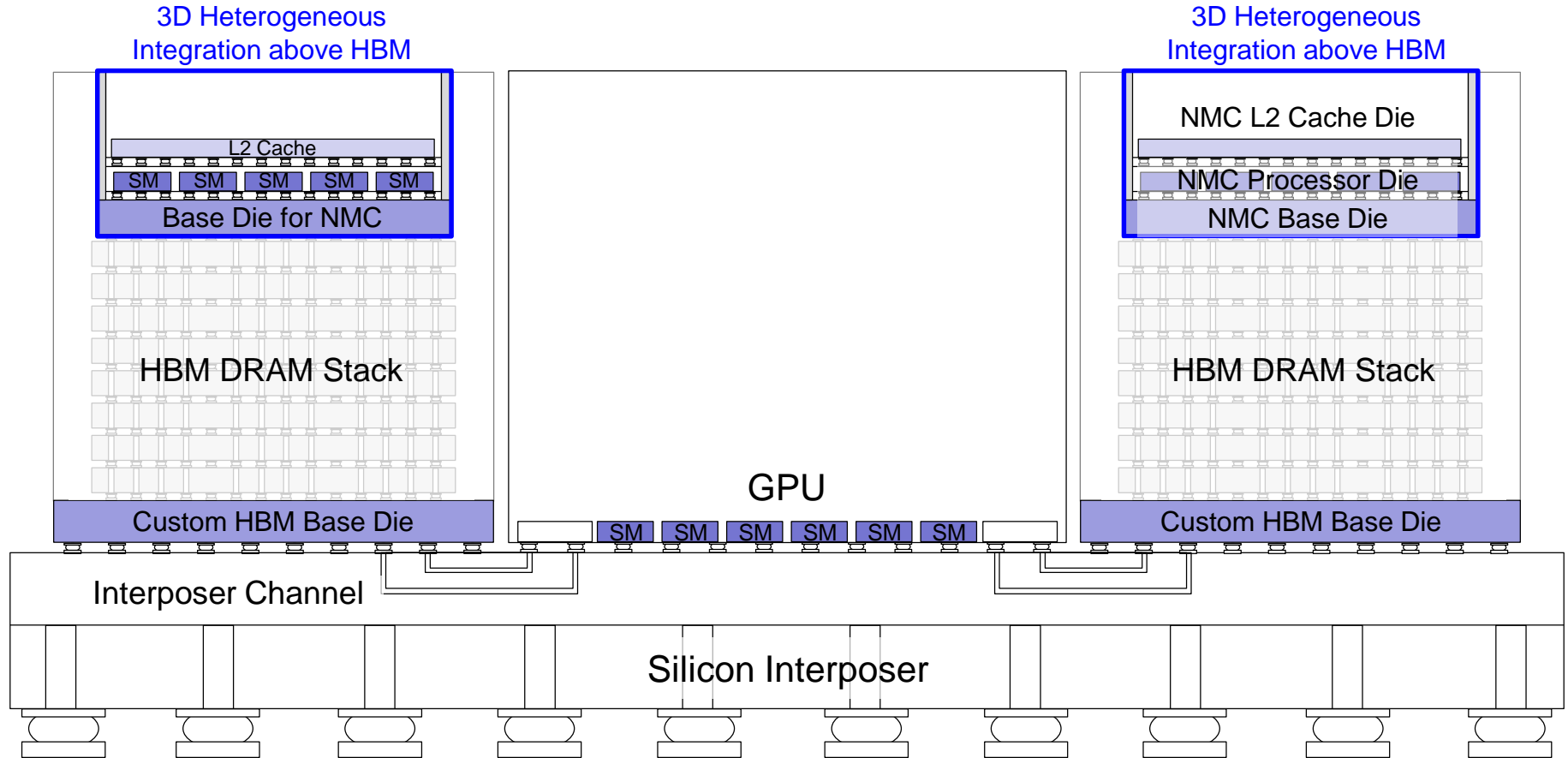


< Conventional CPU-GPU Architecture with HBM3 >



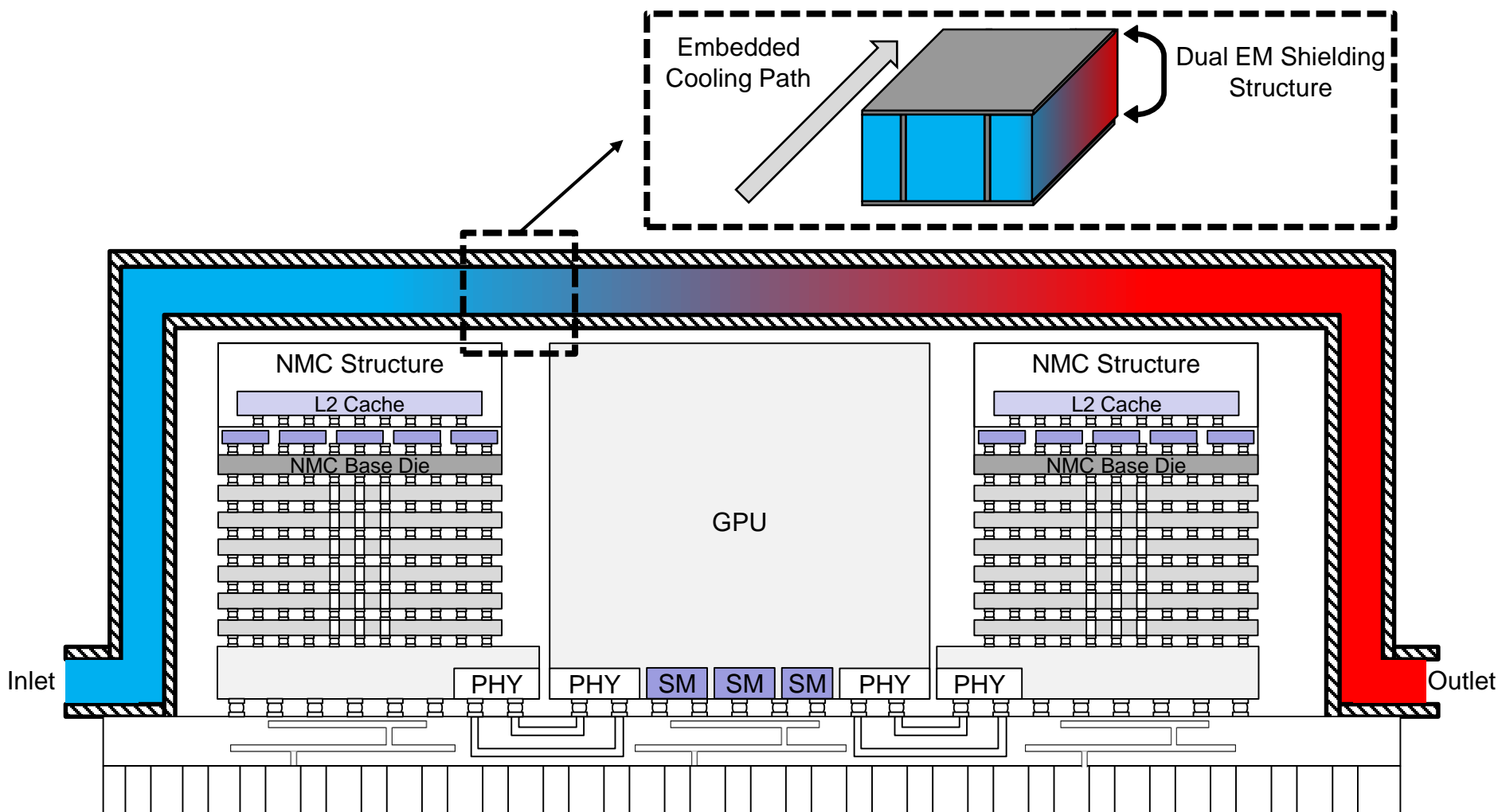
< HBM4 Architecture with LPDDR >

- The custom base die of HBM4 enables direct access to HBM and LPDDR, providing improved memory capacity without the CPU.



## < 3D Integrated NMC-HBM Architecture above HBM DRAM >

- By integrating a NMC processor die and cache die above HBM, the proposed 3D NMC-HBM achieves high performance and energy efficient computing through dedicated TSV interconnection and power supply network.



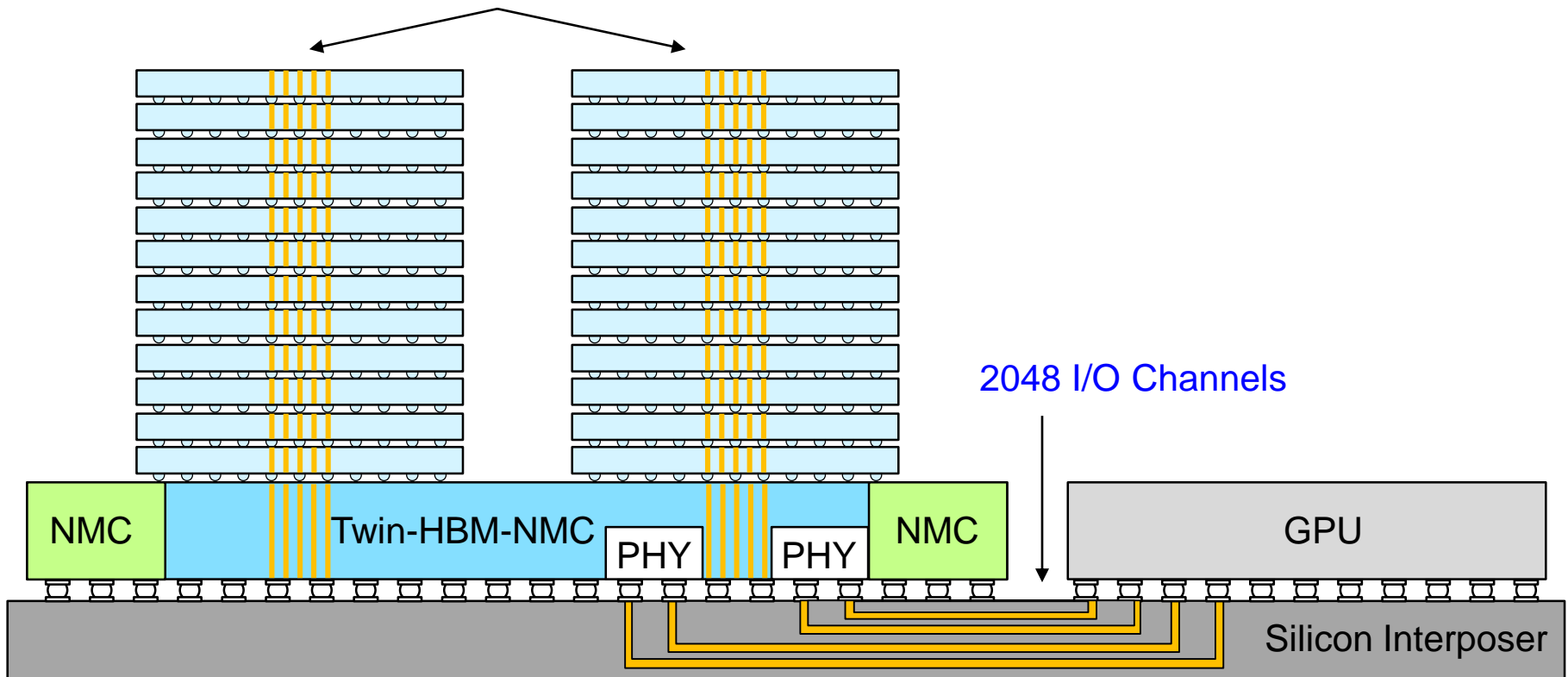
< Electromagnetic Shielding Structure with Liquid Cooling System for HBM5 Architecture >



# Proposal of Twin Tower High Bandwidth Memory with Near-Memory Computing (Twin-HBM-NMC) Architecture

HBM6  
Architecture

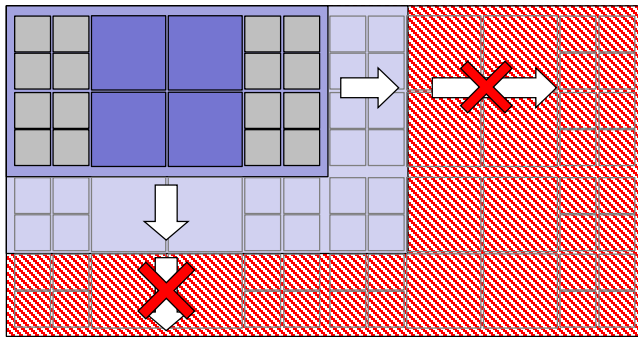
Two DRAM Stacks



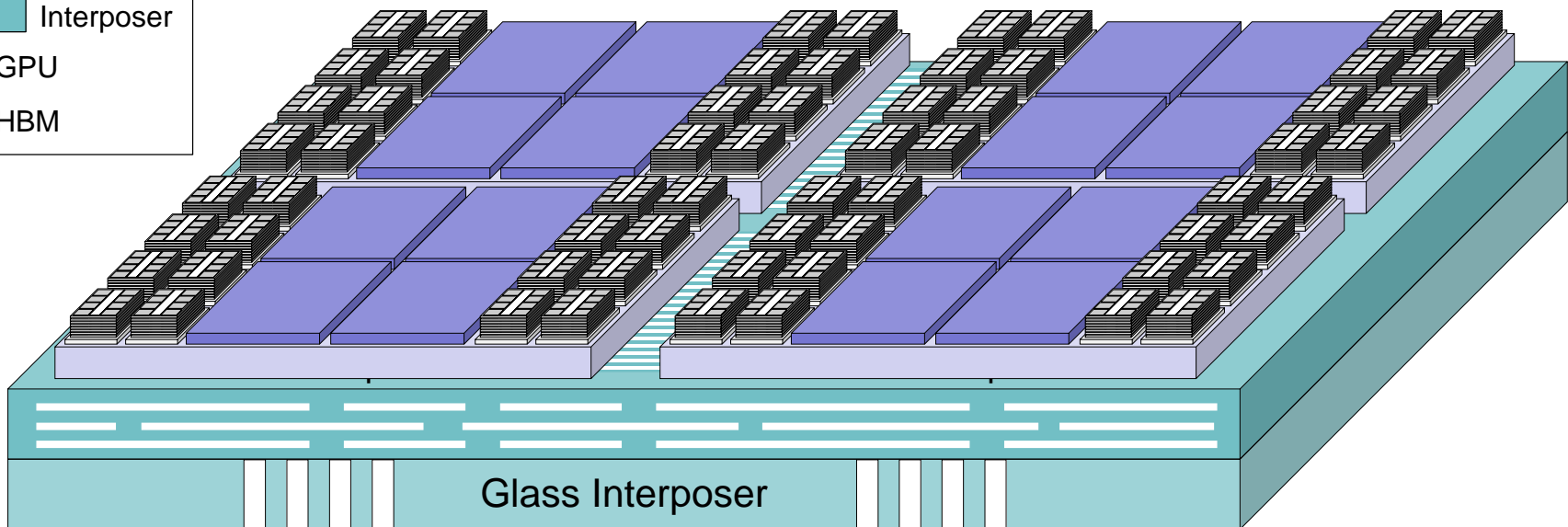
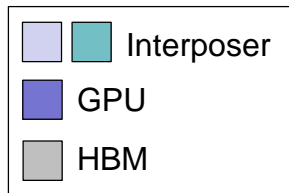
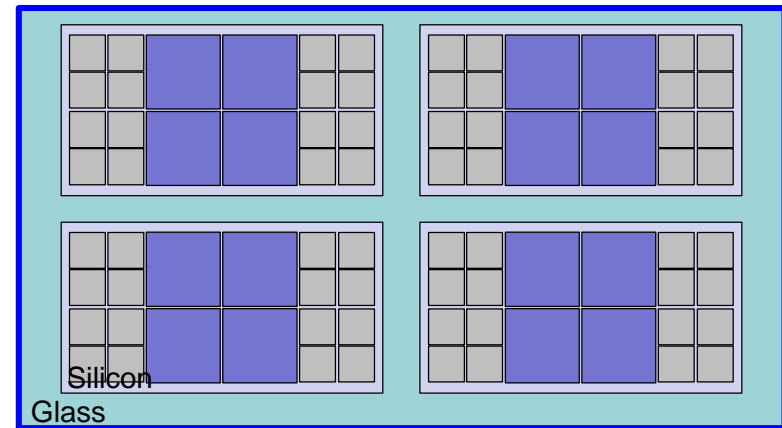
< Overview of Twin Tower HBM with NMC Architecture >

- In twin tower high bandwidth memory with near-memory-computing (Twin-HBM-NMC) architecture, two DRAM stacks are located on top of the large logic die.
- The logic die include NMC units and is connected to the GPU via 2048 interposer channels.

Limitation in increasing  
silicon interposer die size

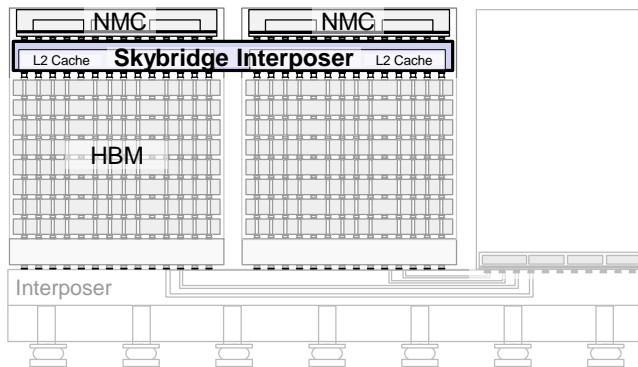


Large Scale Hybrid (Silicon+Glass) interposer

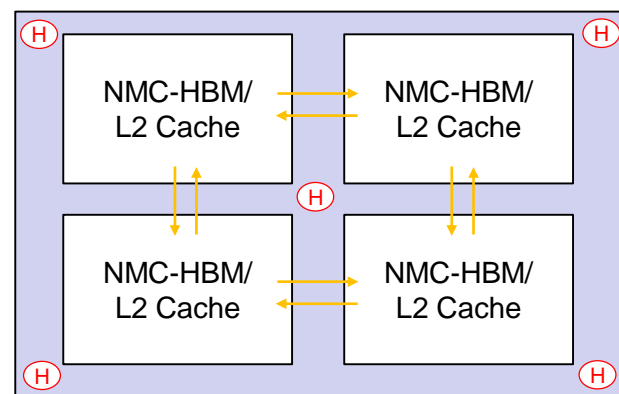


< Silicon-Glass Hybrid Interposer for Ultra Large-Scale Next Generation HBM >

# Multi HBM-HBM Skybridge Interposer for Efficient Near Memory Computing Performance



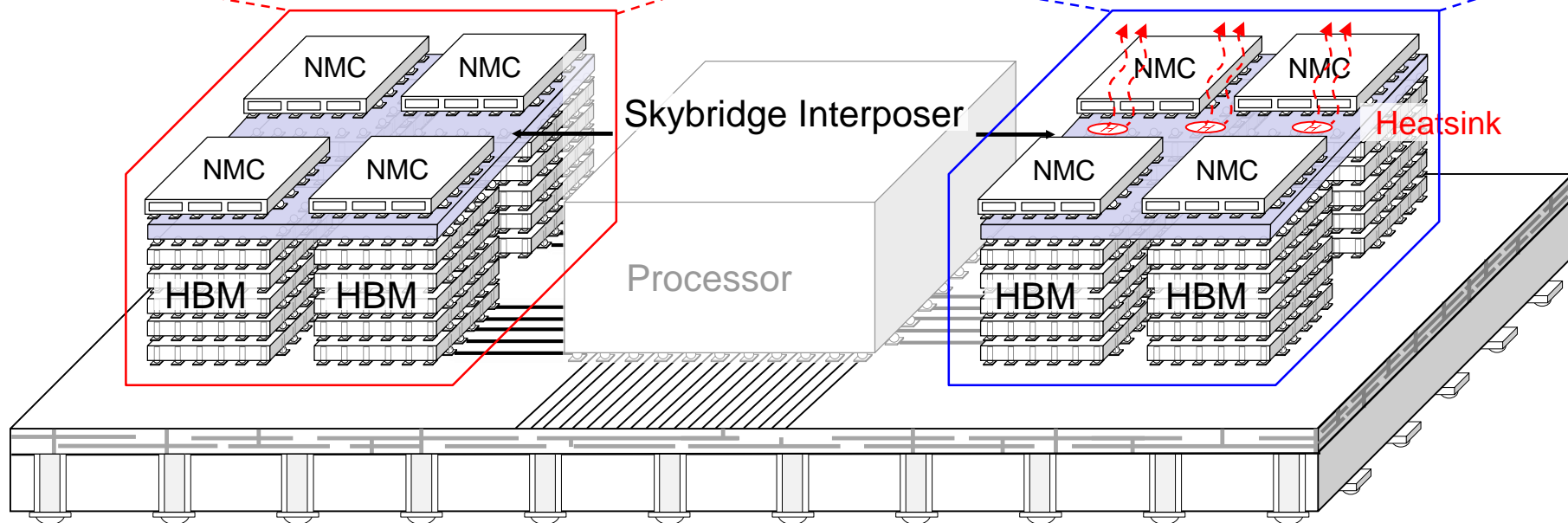
< Side view of Skybridge Interposer Architecture >

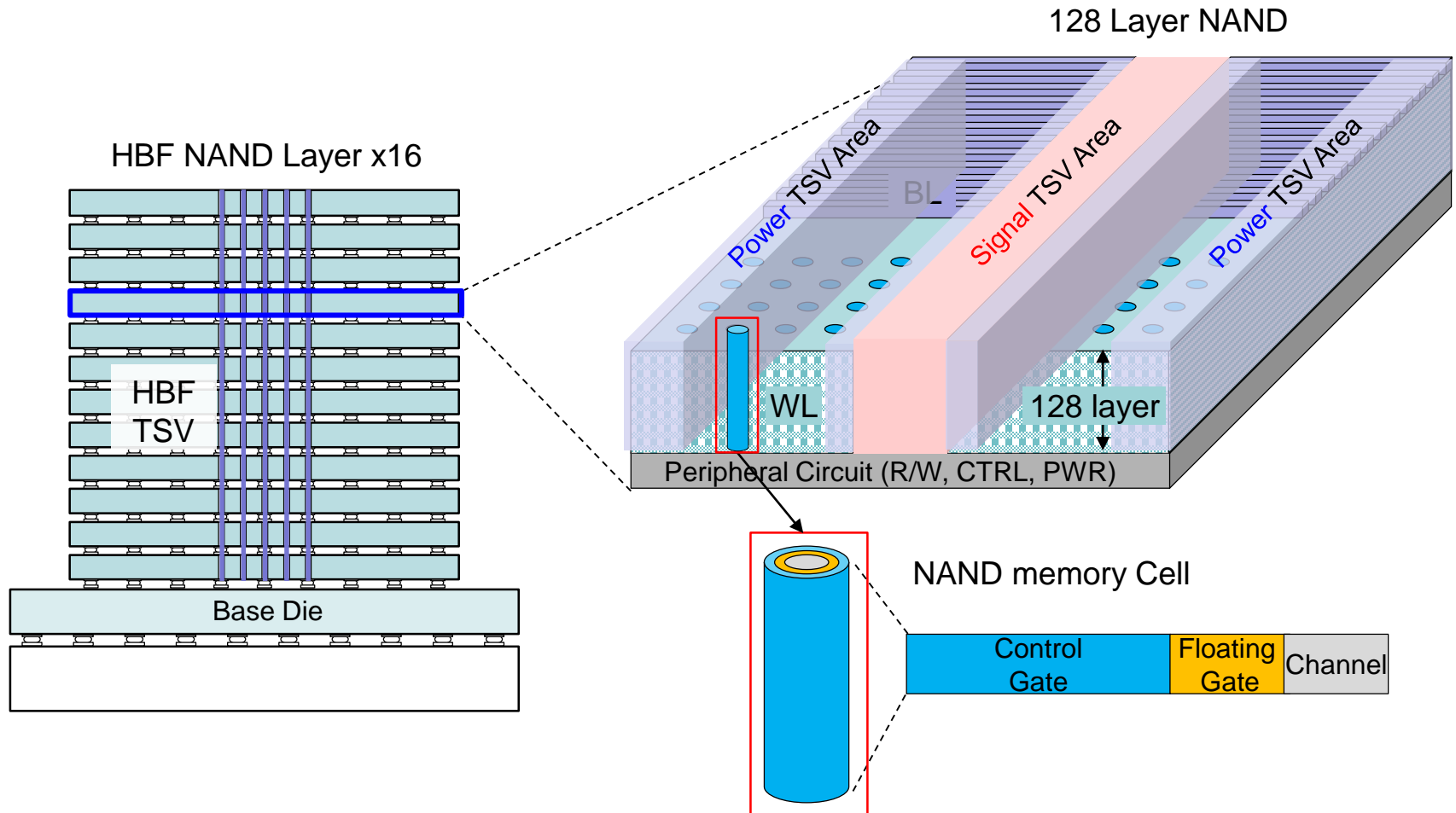


< Top view of Skybridge Interposer >

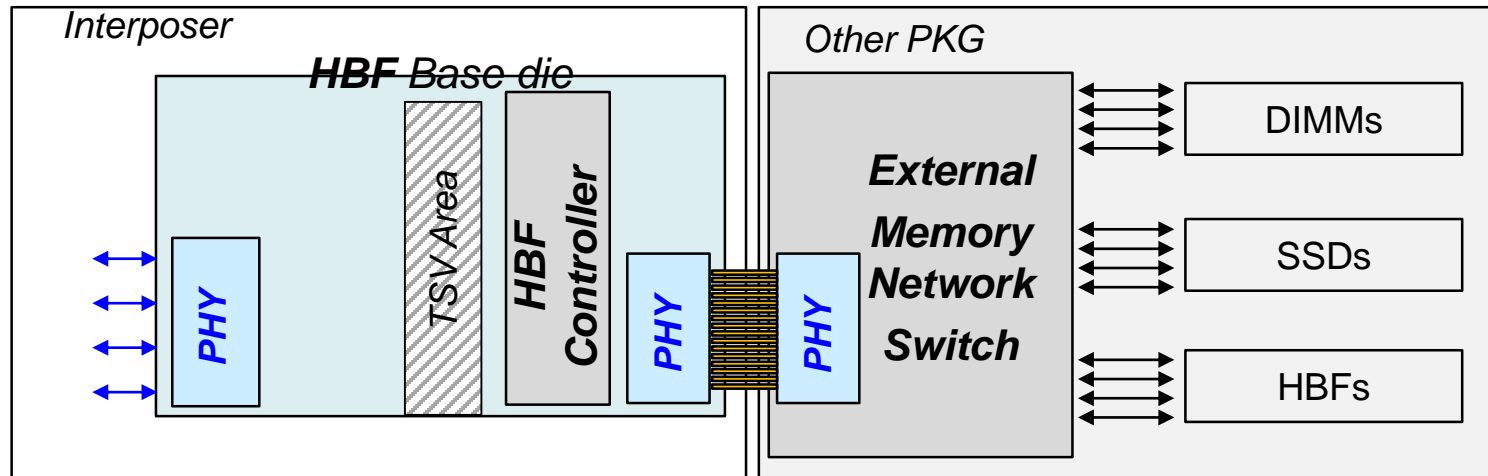
Through  
Skybridge  
Interposer...

- ✓ NMC-NMC
- ✓ HBM-HBM
- ✓ NMC-HBM
- ✓ L2 cache
- ✓ Heatsink



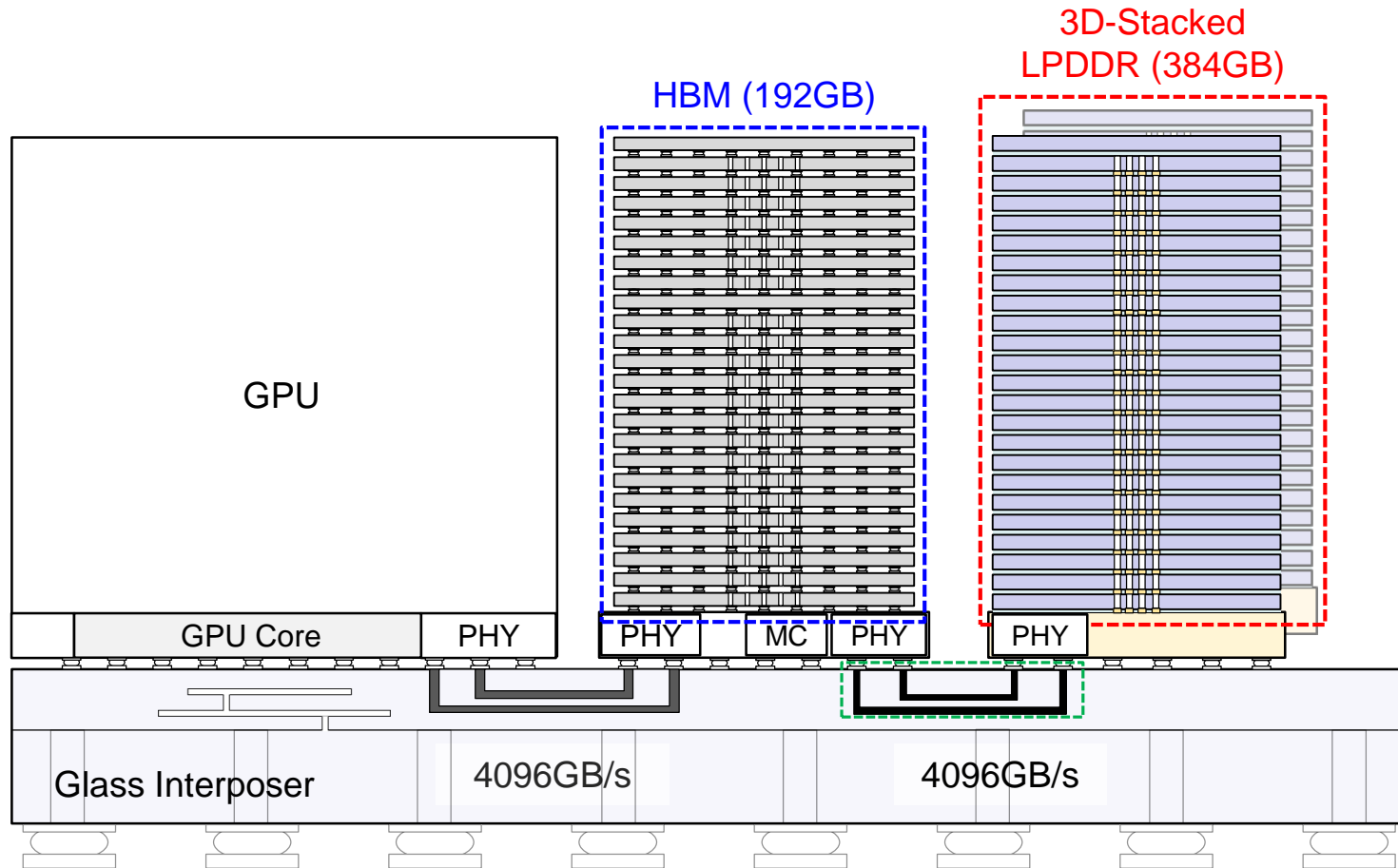


< High Bandwidth Flash (HBF) Architecture for Memory Intensive LLM Inference >

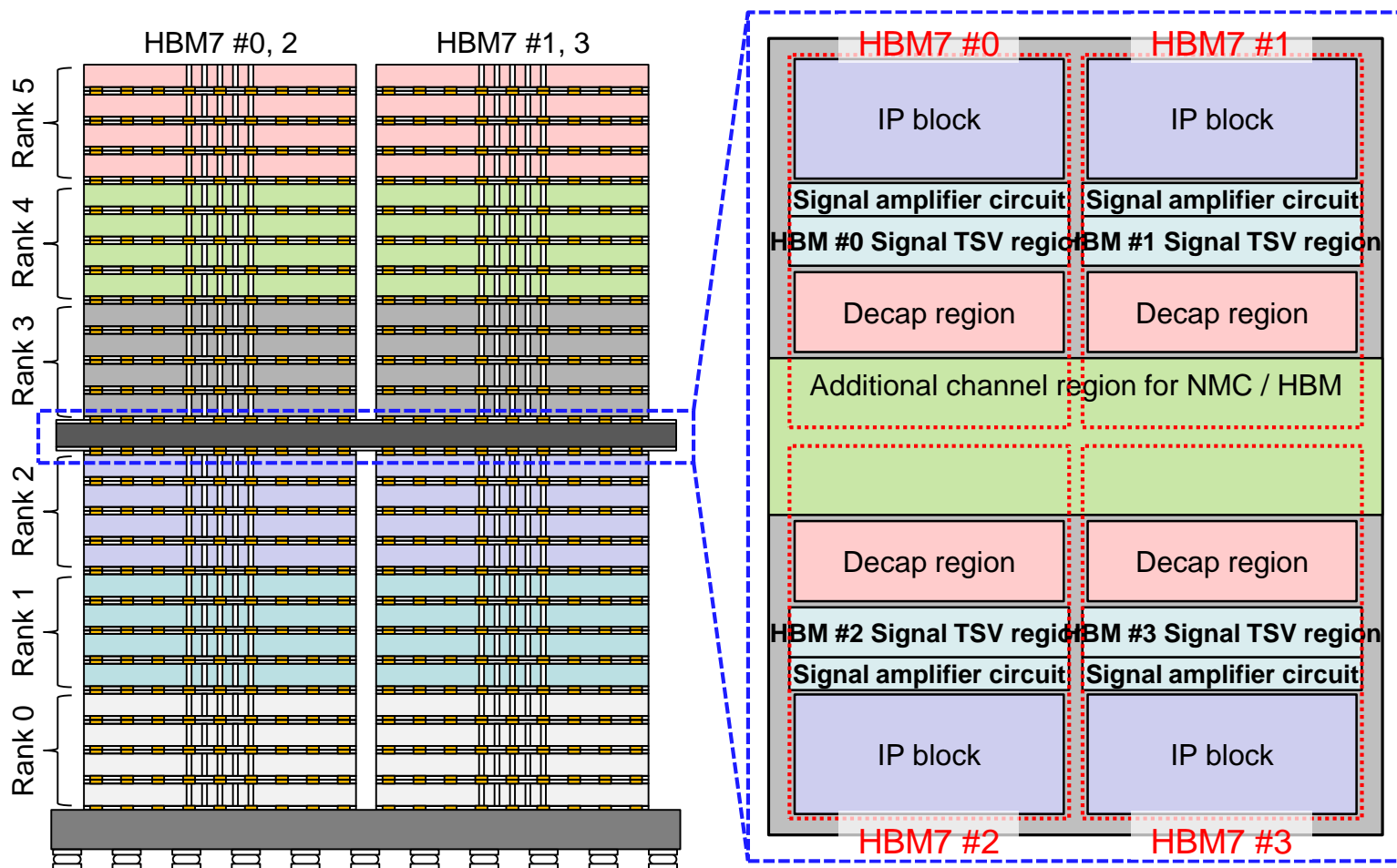


# HBM7 Architecture Integrated with High-Capacity 3D-Stacked LPDDR on Glass Interposer

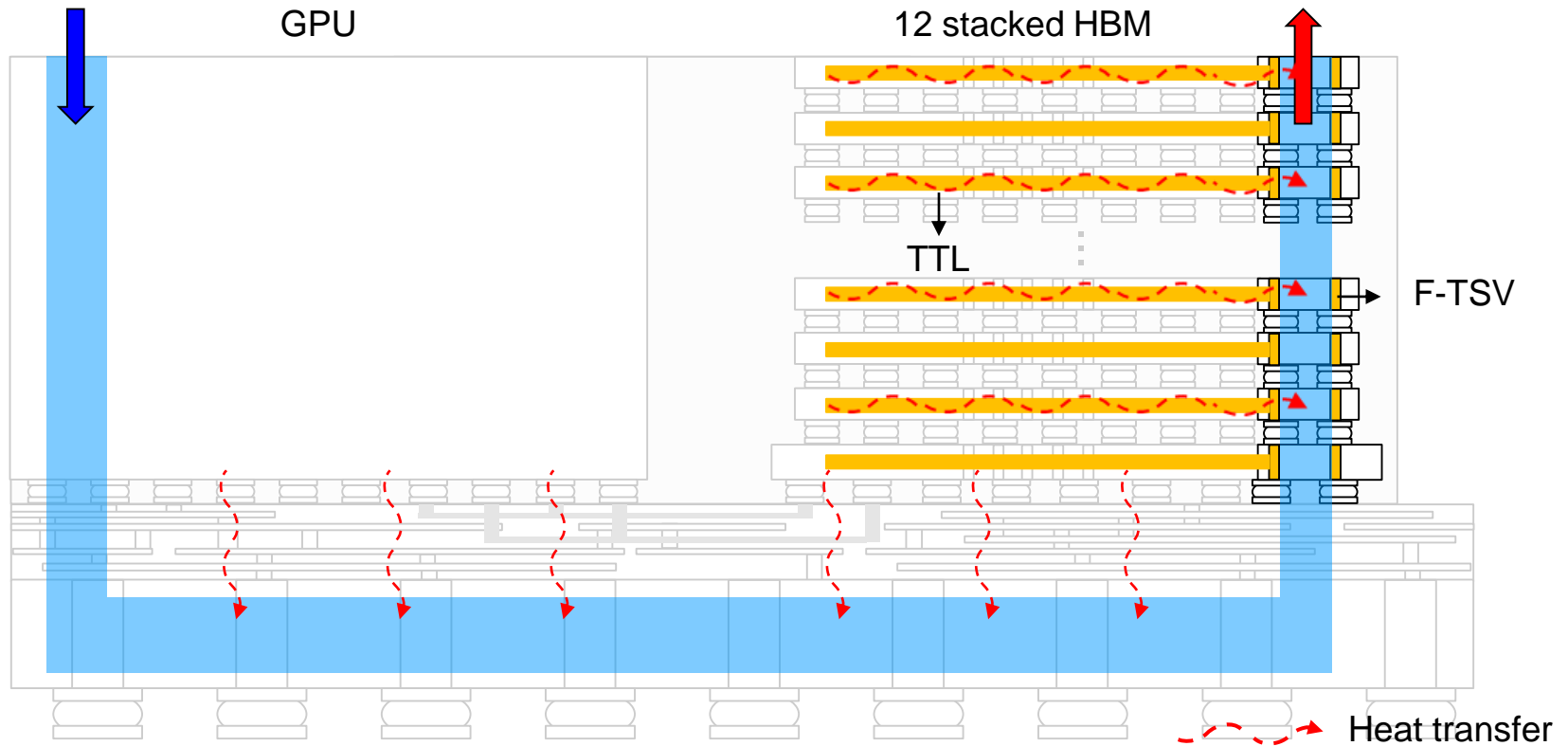
HBM7  
Architecture



< HBM7 Architecture Integrated with High-Capacity 3D-Stacked LPDDR on Glass Interposer >



< Side and top view of proposed multi functional bridge-die >



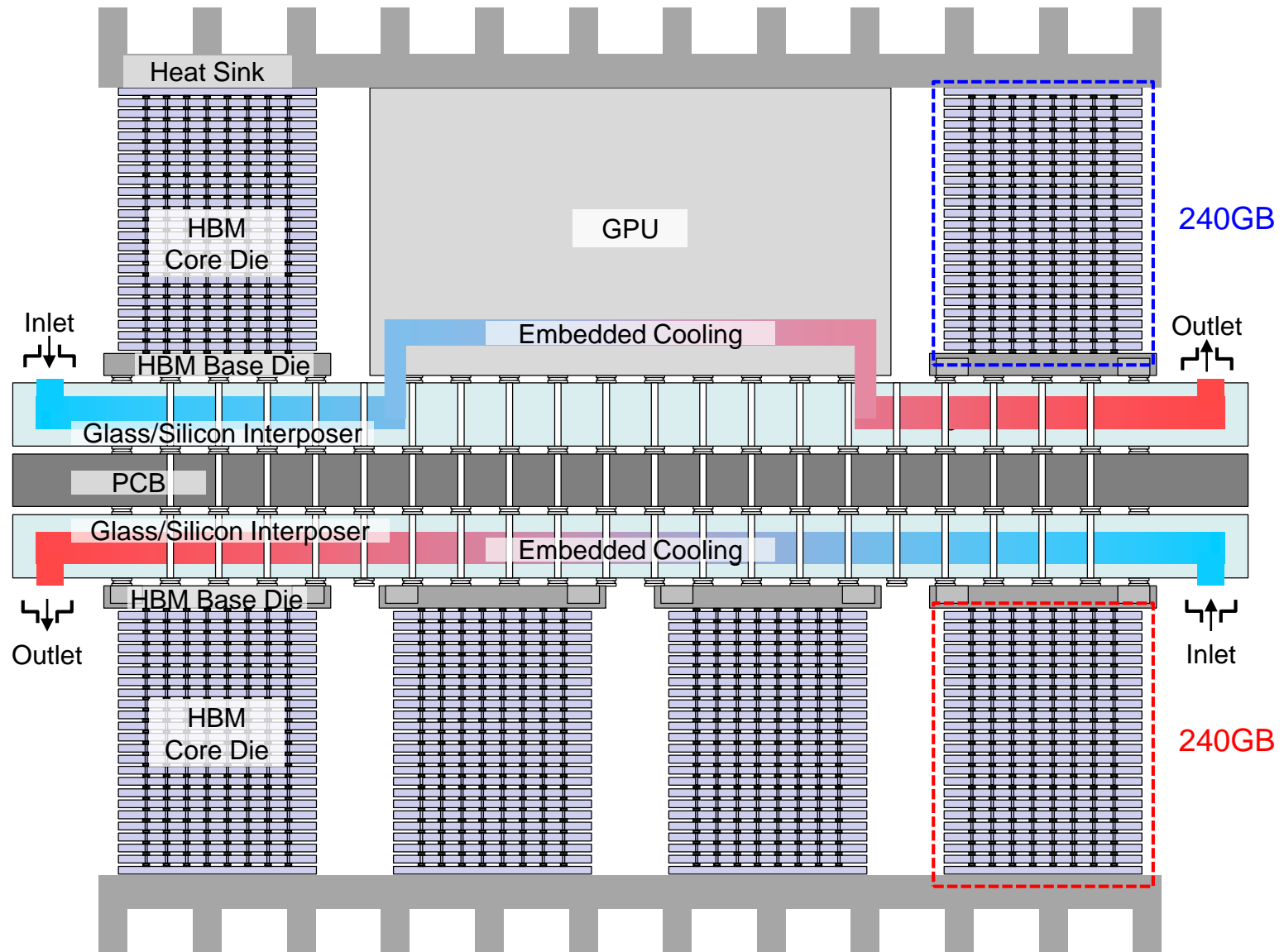
< Concept of the proposed embedded cooling structure for GPU-HBM module >

- The proposed Thermal Transmission Line (TTL) and Fluidic TSV (F-TSV) can cool the HBM module efficiently by circulating cooling fluid through the GPU to the interposer and HBM.
- The proposed TTL transfers the internal heat within HBM die to the fluid flowing in F-TSV

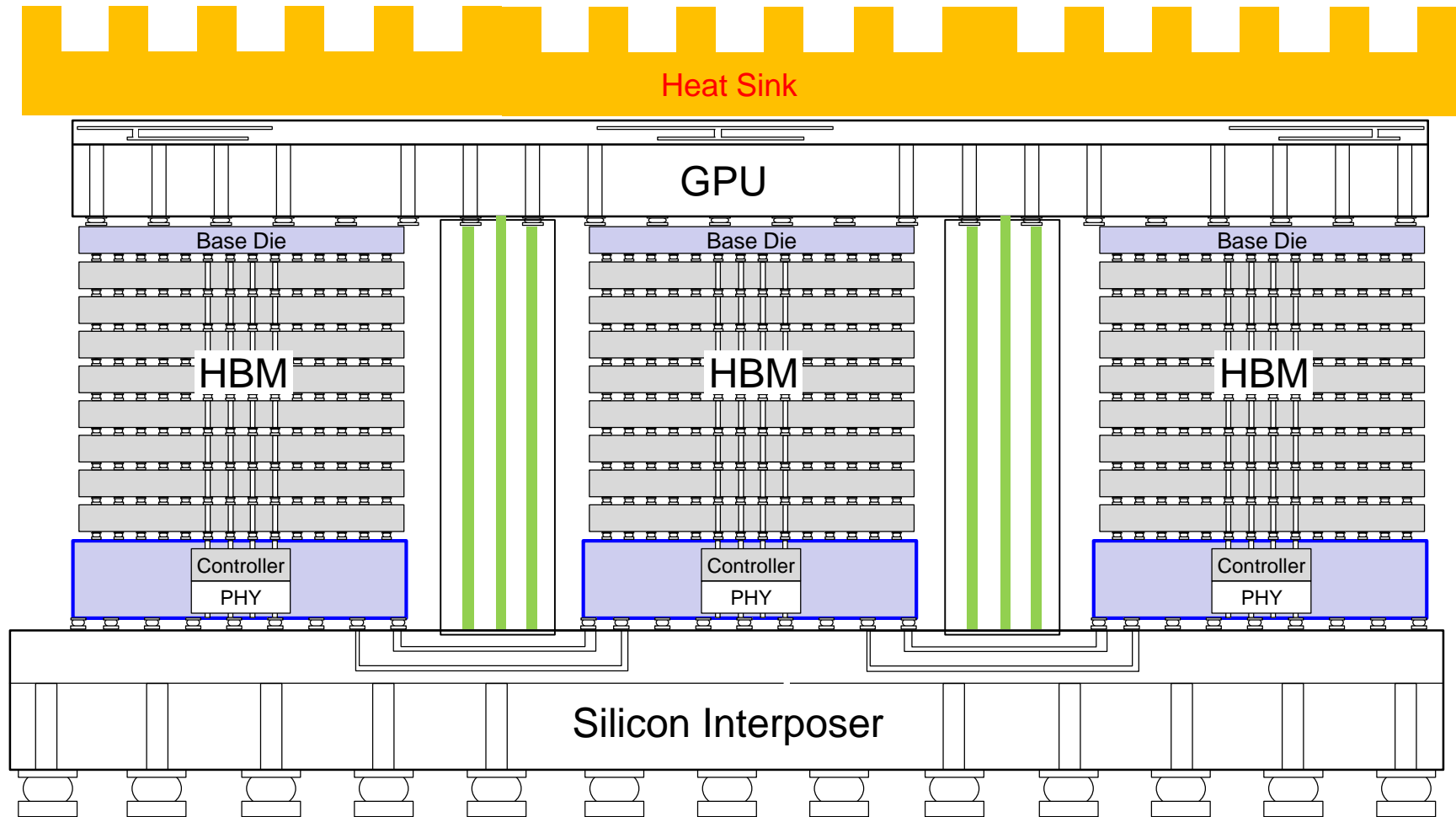


# 3D Memory Expansion Architecture with Embedded Cooling Structure for HBM8 with Double-Sided Interposer [1/3]: GPU-HBM-HBM

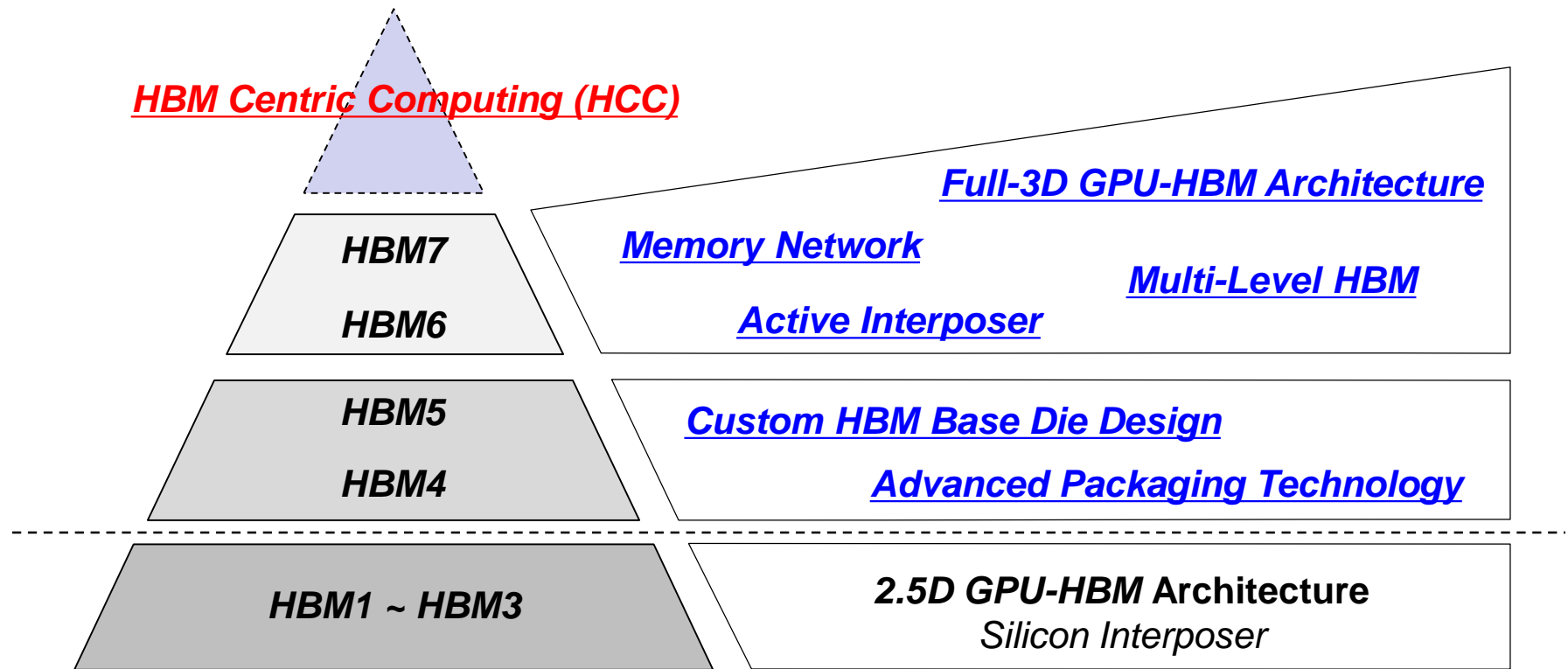
HBM8  
Architecture



< 3D Memory Expansion Architecture with Embedded Cooling for HBM8 with Double-Sided Interposer using HBM >



- GPU is implemented at top layer of memory stack for heat dissipation. (Thermal Issue ↓)
- Additional silicon interconnect pillar die is embedded between HBMs to support power to GPU.



< Next-Generation HBM Architecture Roadmap >

## Part3: Key Features in HBM4

# Key Features of HBM4

## 1. Electrical Specification

- Data Rate : ~ 8 Gbps
- Number of I/Os : 2,048 (4096)
- Total Bandwidth : 2.0 TB/s
- Number of die stack : 12/16-Hi
- Capacity/die : 24 Gb

## 2. Packaging/Cooling Method

- Microbump (MR-MUF)
- Direct-to-Chip (D2C) Liquid Cooling

## 3. HBM Architecture

- Custom HBM Base Die
- NMC processor + LPDDR in Base Die

## 4. AI Design Agent

- AI assistant
- Microbump & TSV array, and Decap placement Optimization based on Reinforcement Learning

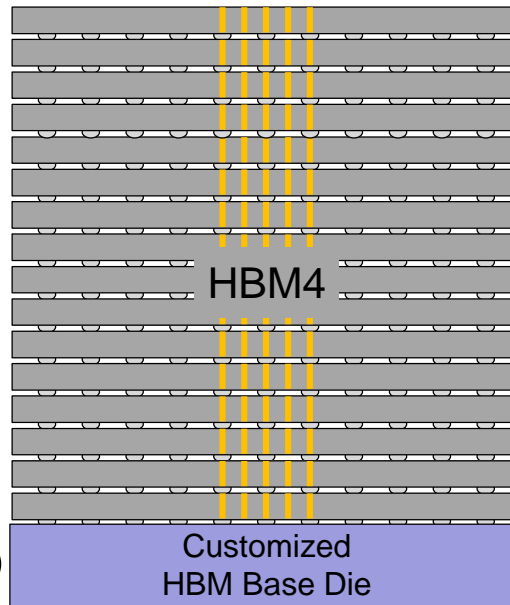
TSV(실리콘 관통전극)



1

HBM 적층 수, 저장 용량 증가  
8 ~ 12 단 → 12 ~ 16 단  
16 ~ 24 GB → 36 ~ 48 GB

1



HBM4

GPU

2

2

고객 맞춤형 (Customized)  
Base Die 설계  
일부 GPU 계산 기능이 이동

Customized  
HBM Base Die

3

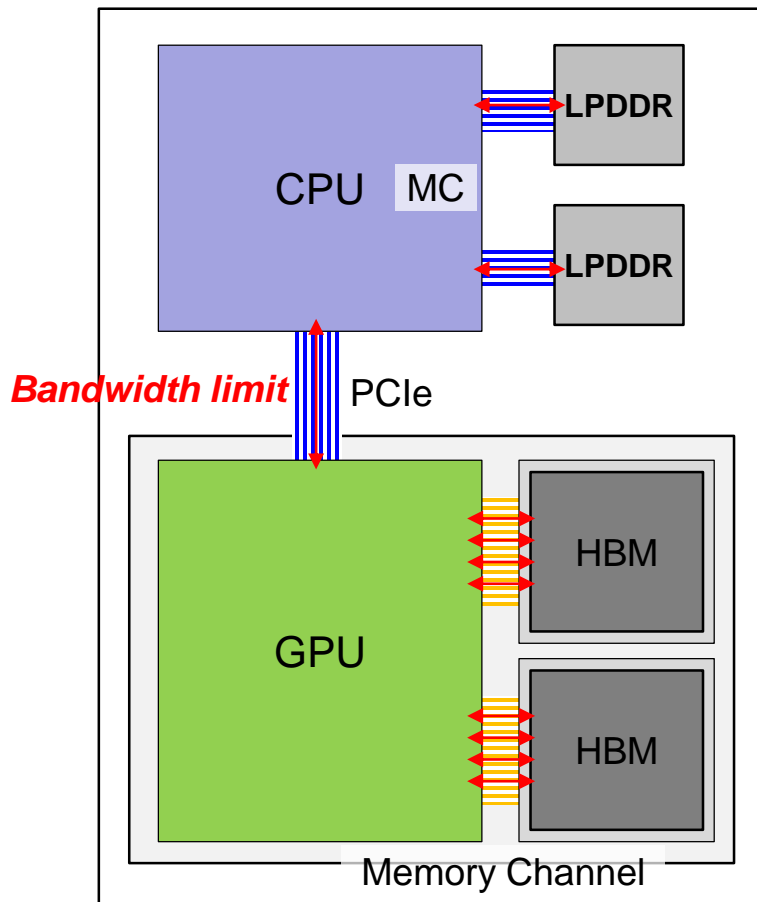
실리콘 인터포저 (기판)

3

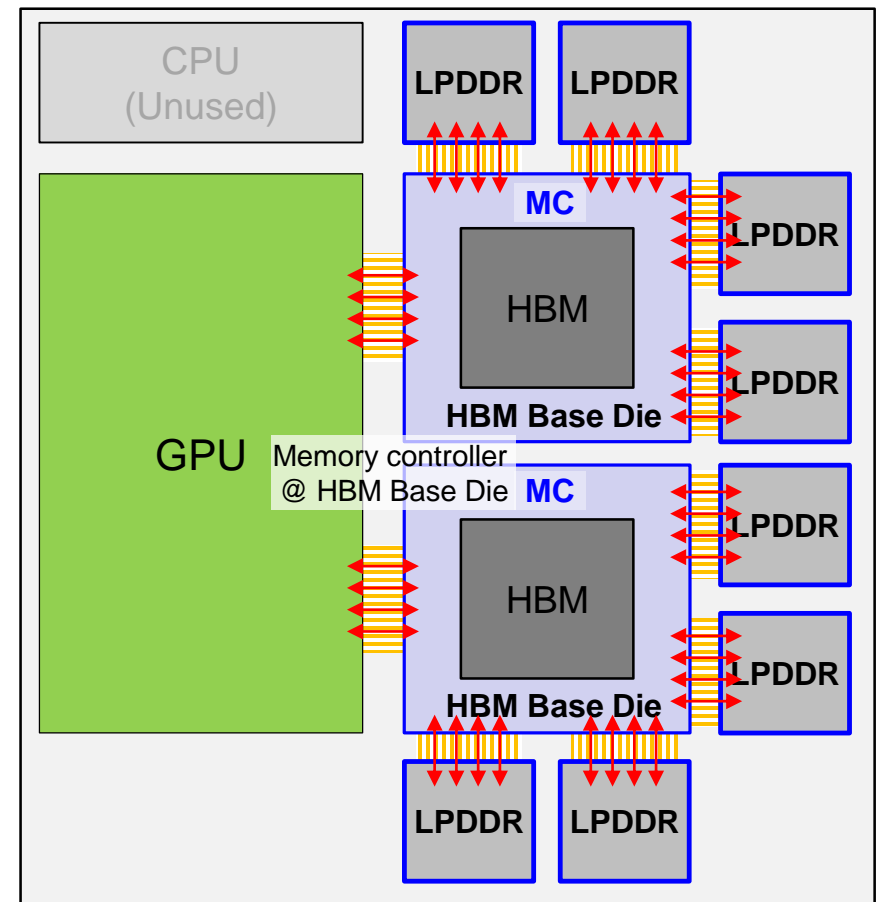
HBM3/3E 대비 2배 늘어난  
HBM-GPU 연결선 (I/O)  
2,048 개

# Custom HBM Base Die Design : LPDDR Memory Channel for High Capacity & Memory Bandwidth

↔ : Low-Bandwidth      ↔↔↔ : High-Bandwidth

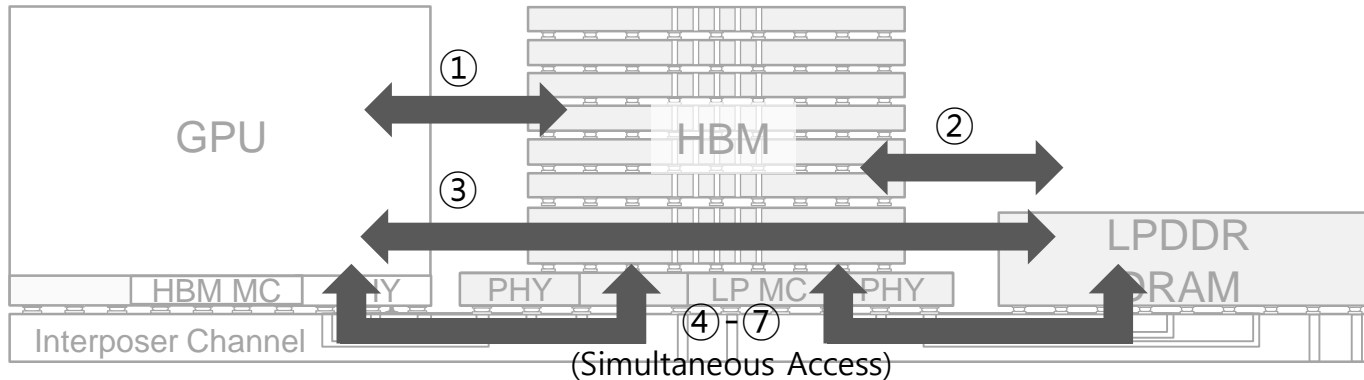


< Conventional CPU-GPU Architecture with HBM3 >



< HBM4 Architecture with LPDDR >

- The custom base die of HBM4 enables direct access to HBM and LPDDR, providing improved memory capacity without the CPU.



< Side-view of HBM-LPDDR Structure and its data path >

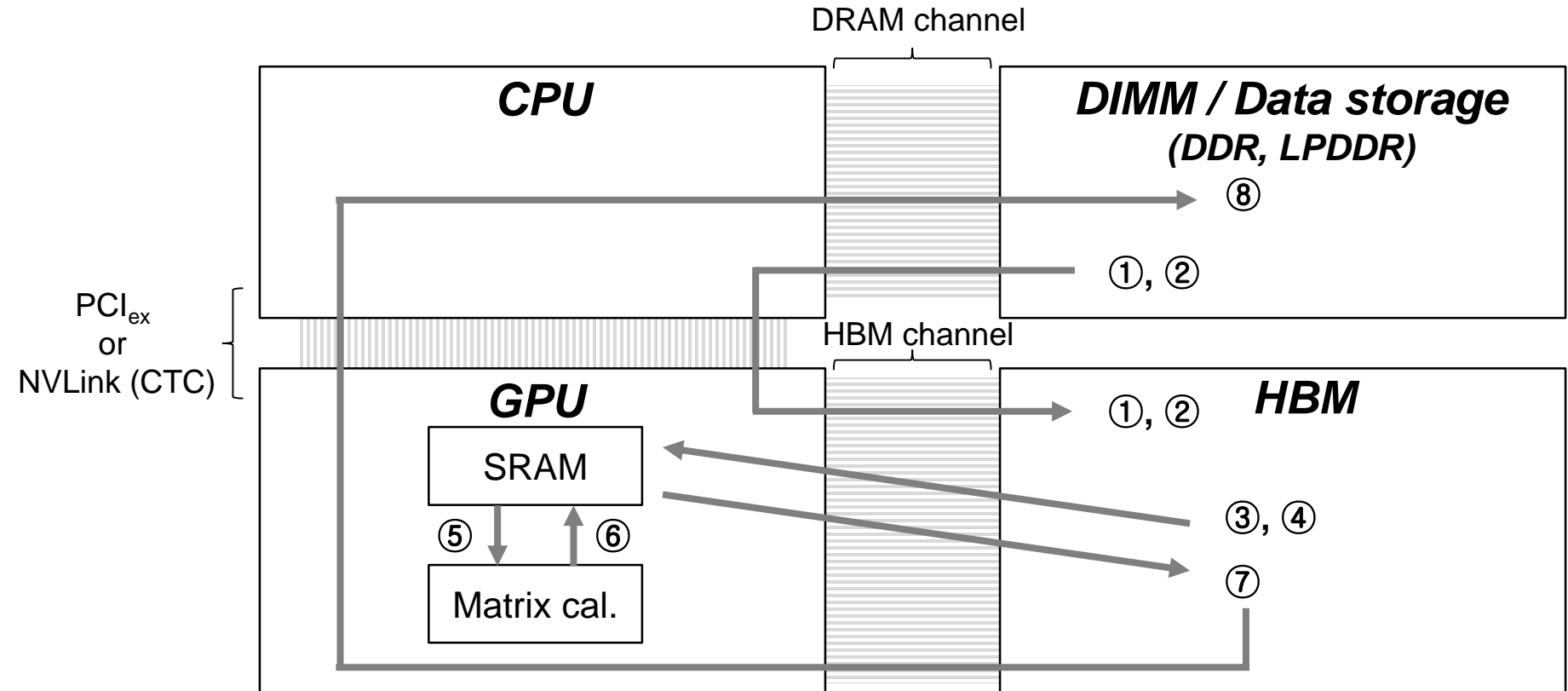
- Case 1: GPU ↔ HBM (Read/Write)
  - Case 2: HBM ↔ LPDDR (Read/Write)
  - Case 3: GPU ↔ LPDDR (Read/Write)
  - Case 4: GPU → HBM (Write) & HBM ← LPDDR (Read)
  - Case 5: GPU → HBM (Write) & HBM → LPDDR (Write)
  - Case 6: GPU ← HBM (Read) & HBM ← LPDDR (Read)
  - Case 7: GPU ← HBM (Read) & HBM → LPDDR (Write)
- }

Single-Command  
Execution
- }

Dual-Command  
Execution  
(GPU ↔ HBM ↔ LPDDR)

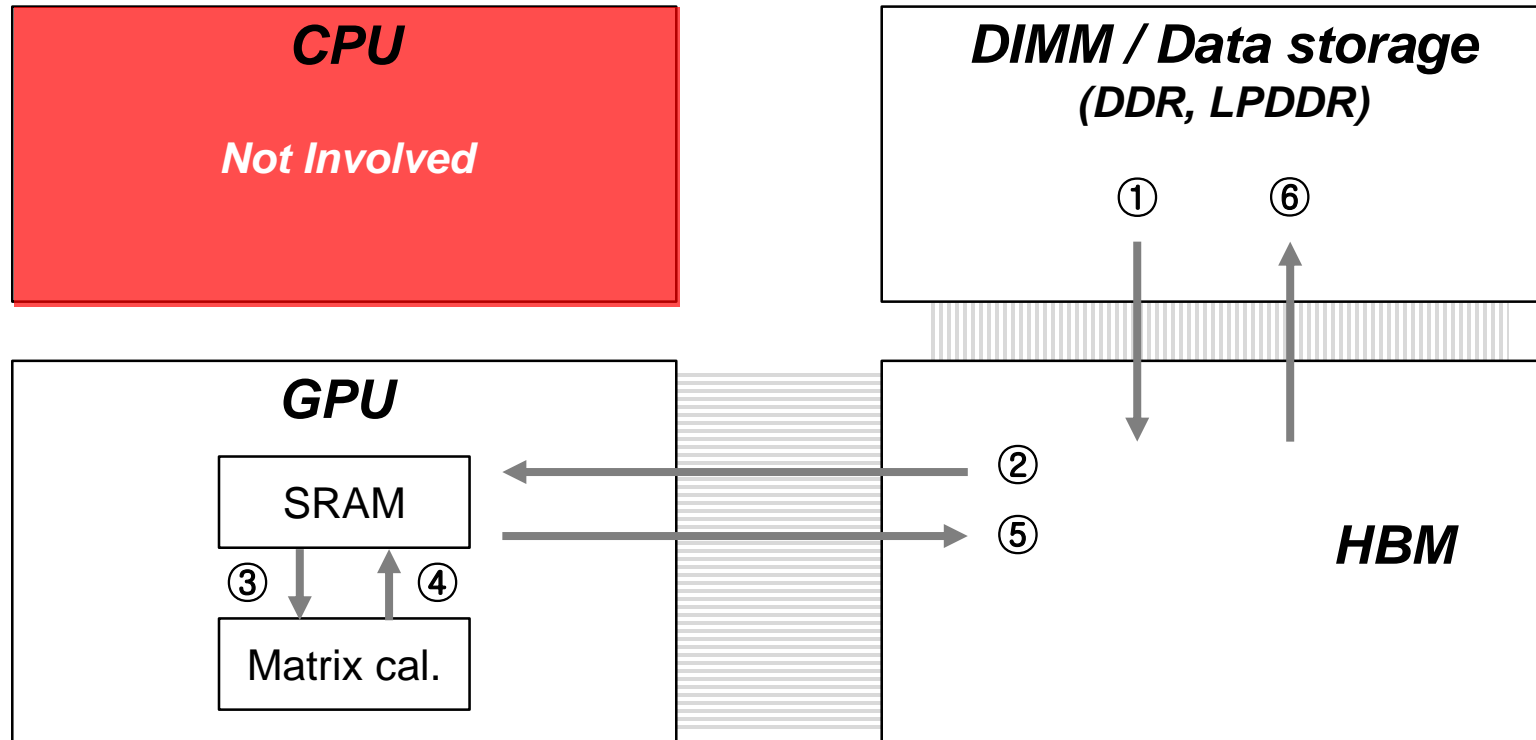


## Conventional Data Flow in AI Computer Architecture



- ①, ② Data (Input  $x$ , Input  $h$ , Weight  $w_{ij}$ ) from DIMM → copy to → HBM  
→ Path: DIMM → CPU → GPU → HBM
- ③, ④ Data (Input  $x$ , Input  $h$ , Weight  $w_{ij}$ ) from HBM to cache in GPU  
→ Path: HBM → GPU
- ⑤ Matrix calculation in GPU
- ⑥ Save matrix multiplication results to SRAM
- ⑦ Copy ⑥ → to HBM / Path: GPU → HBM
- ⑧ Copy ⑦ → to DIMM / Path: HBM → GPU → CPU → DIMM

## New Data Flow in HCC with GPU Co-Existence



- ① Data (Input  $x$ , Input  $h$ , Weight  $w_{ij}$ ) from DIMM → copy to → HBM / Path: DIMM → HBM
- ② Data (Input  $x$ , Input  $h$ , Weight  $w_{ij}$ ) from HBM to cache in GPU / Path: HBM → GPU
- ③ Matrix calculation in GPU
- ④ Save matrix multiplication results to SRAM
- ⑤ Save it to HBM / Path: GPU → HBM
- ⑥ Copy it to DIMM / Path: HBM → DIMM

→ Since the **CPU is not involved** in the memory transfer path, **delays can be reduced**, and it has the advantage of **fewer interconnection steps and shorter lengths**.

## Part4: Key Features in HBM5

# Key Features of HBM5

## 1. Electrical Specification

- Data Rate : 8 Gbps
- Number of I/Os : 4,096
- Total Bandwidth : 4.0 TB/s
- Number of die stack : 16-Hi
- Capacity/die : 40 Gb

## 2. Packaging/Cooling Method

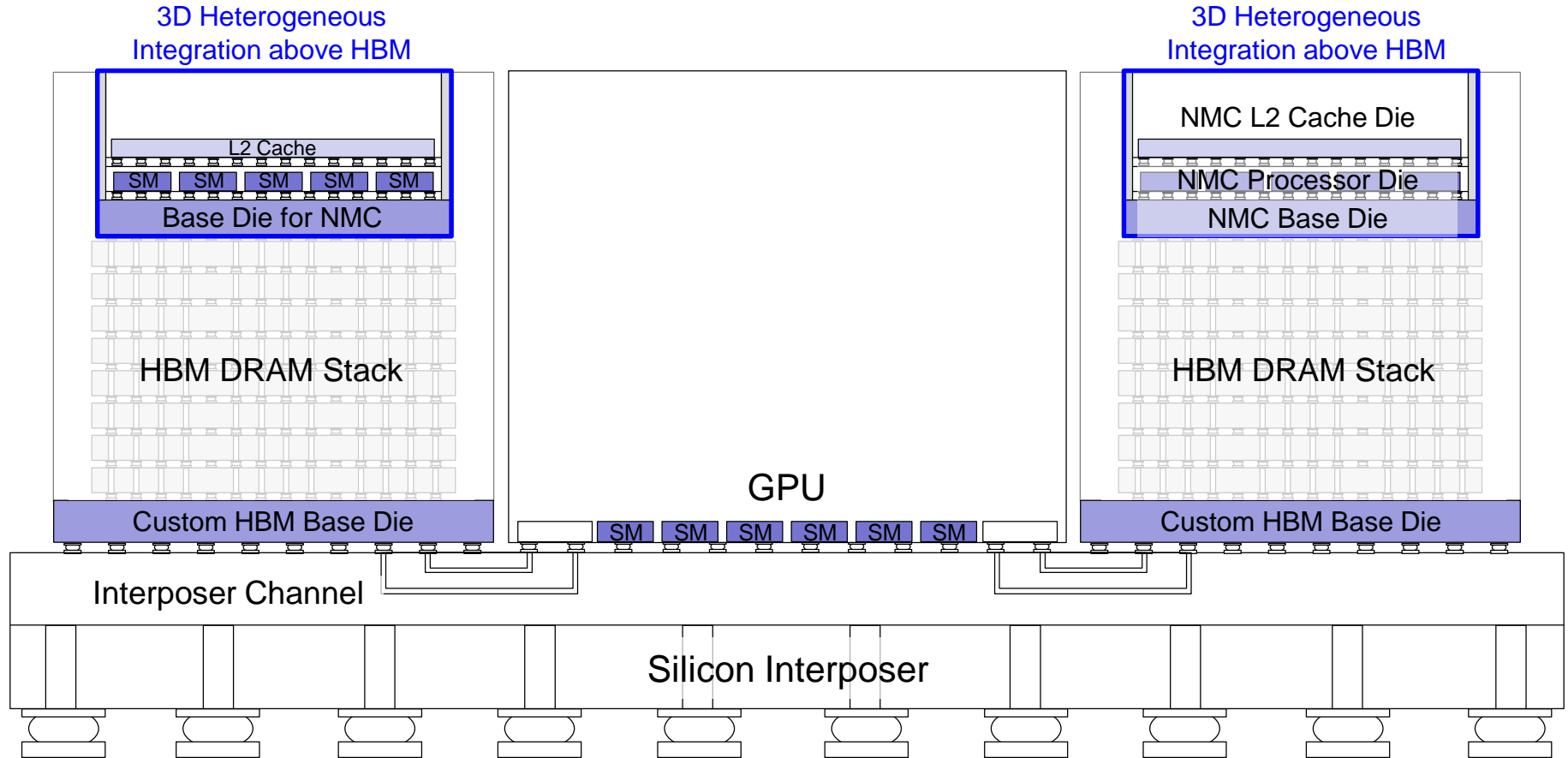
- Microbump (MR-MUF)
- Immersion Cooling, Thermal Via (TTV), Thermal Bonding
- Temperature sensors in base die

## 3. HBM Architecture

- Dedicated decoupling capacitor chip die stack
- Custom HBM Base Die w/ 3D NMC-HBM & Stacked Cache
- NMC + Cache in HBM PKG
- LPDDR + CXL in Base Die
- Separated TSV, TGV, TPV, TTV designs

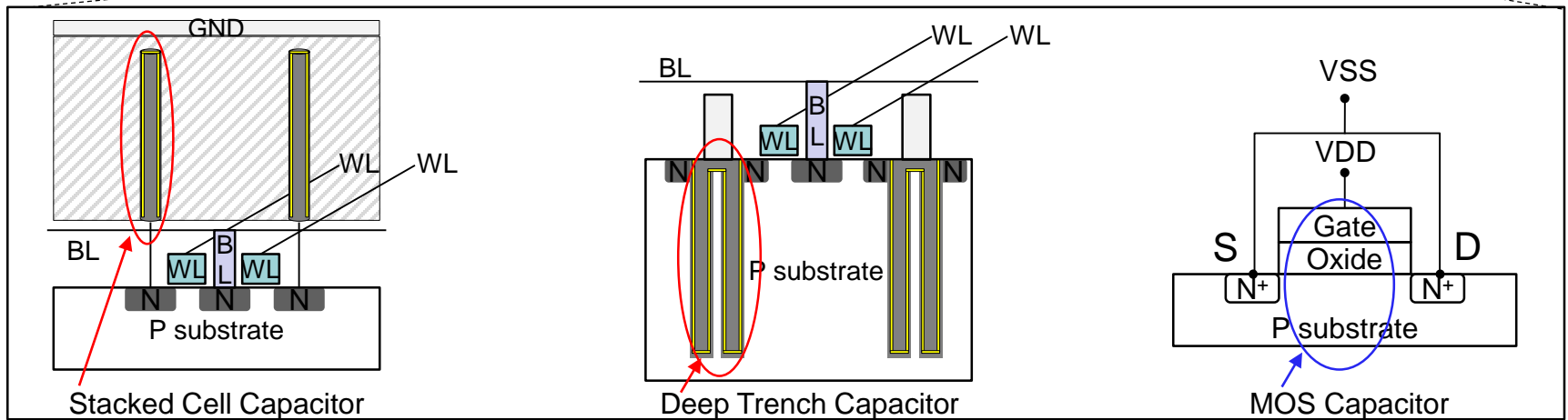
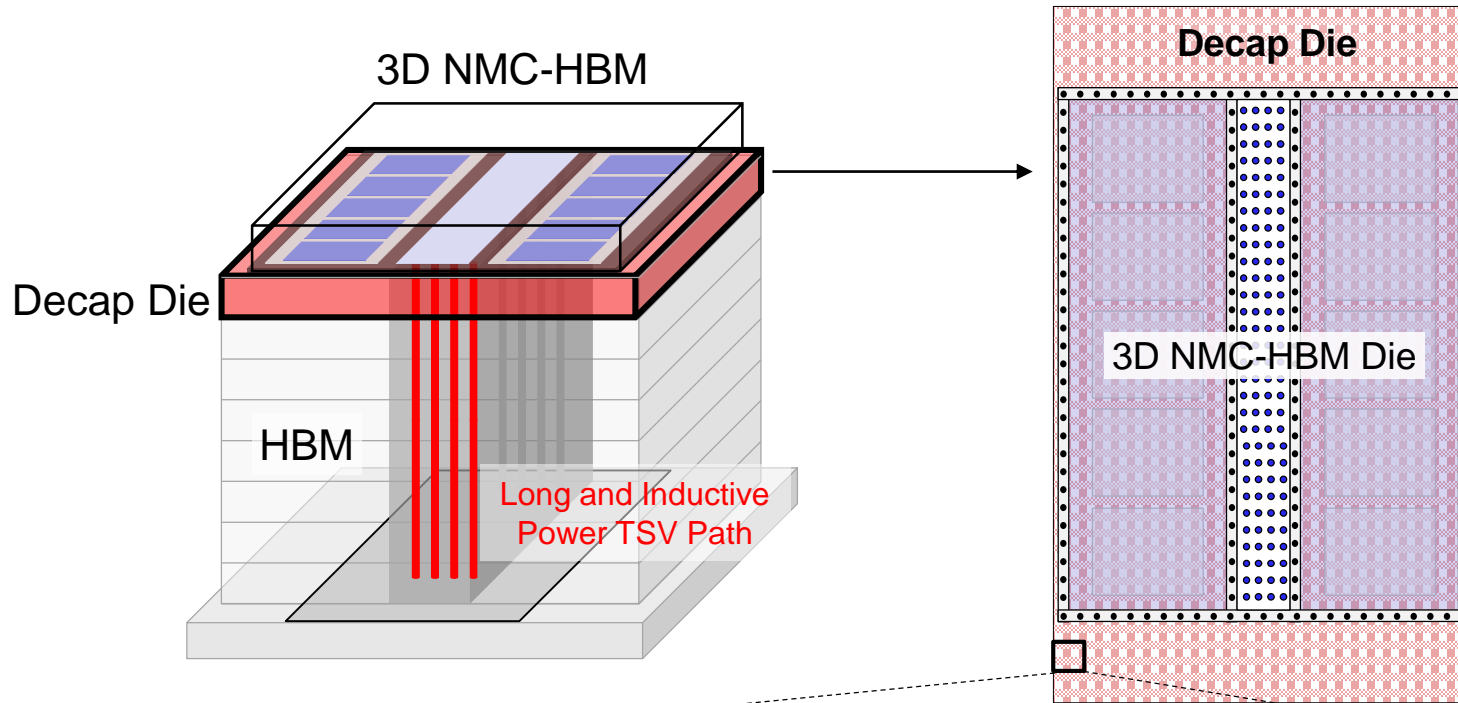
## 4. AI Design Agent

- AI Agent
- I/O Interface Optimization considering PSIJ based on Reinforcement Learning

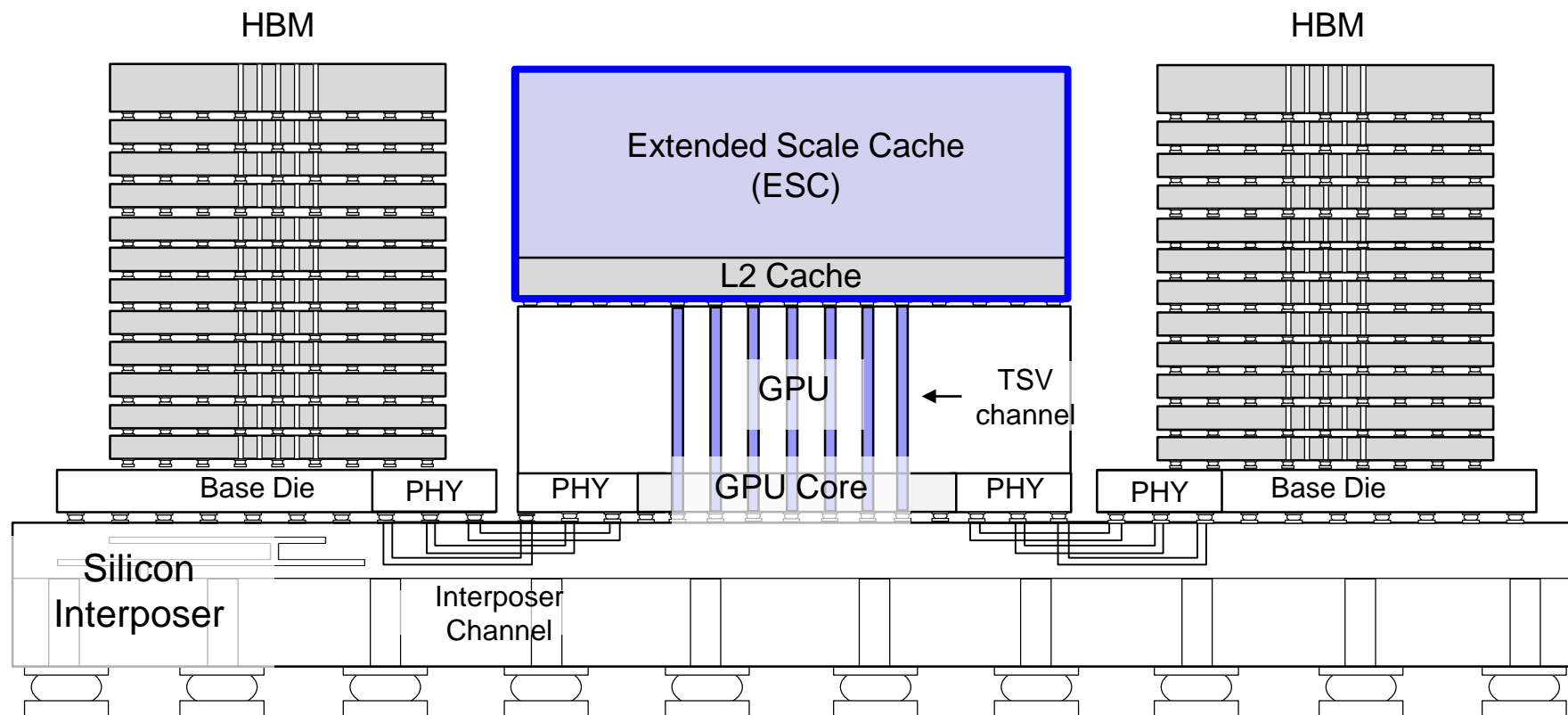


## < 3D Integrated NMC-HBM Architecture above HBM DRAM >

- By integrating a NMC processor die and cache die above HBM, the proposed 3D NMC-HBM achieves high performance and energy efficient computing through dedicated TSV interconnection and power supply network.

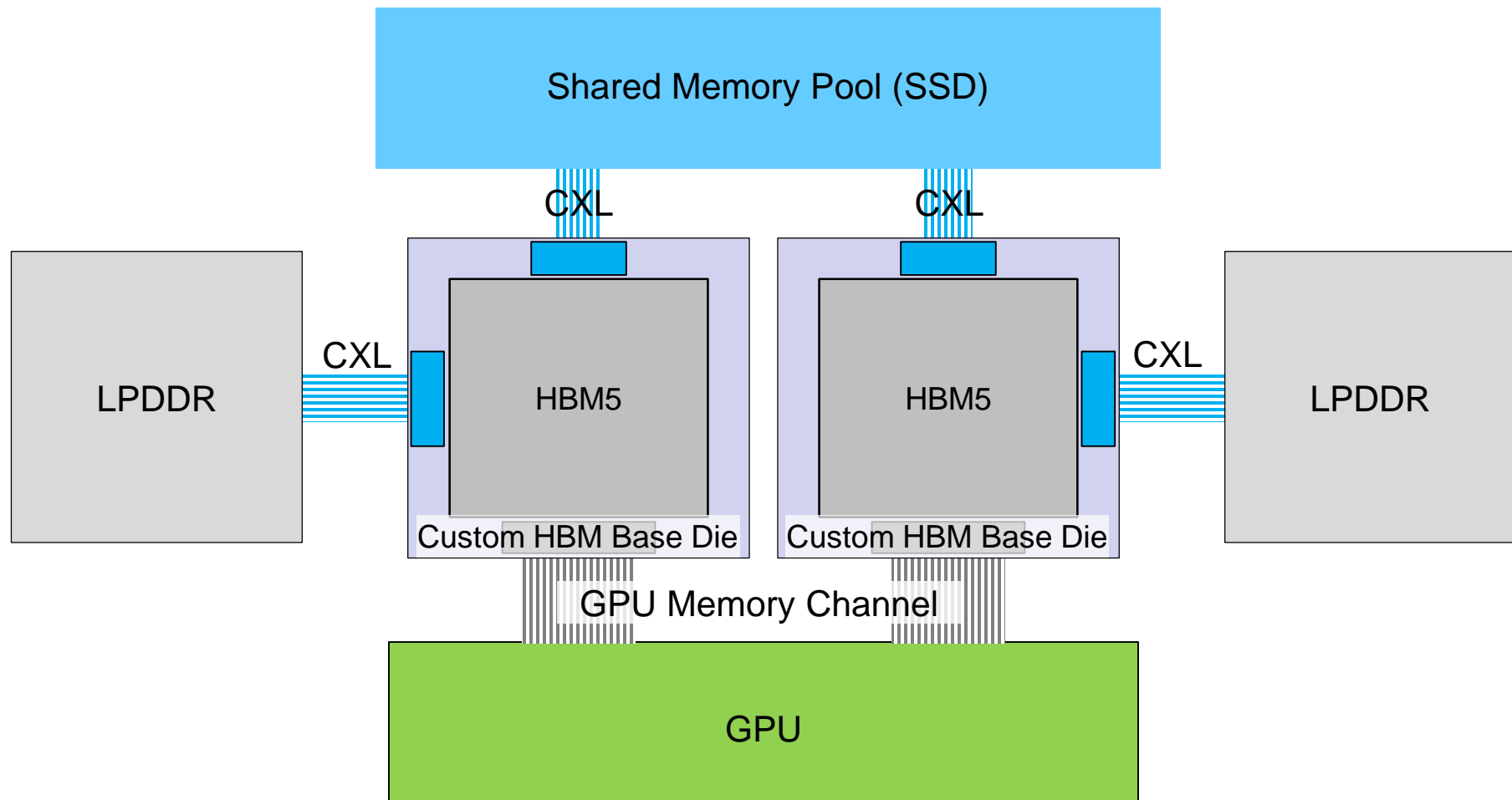


< Designated Decoupling Capacitor Die in 3D Stacked NMC-HBM5 Architecture >



< Extended Scale Cache (ESC) Stacked-GPU HBM Architecture >

- The proposed architecture utilizes an extended L2 cache, which is the last-level cache of the GPU, stacked above the GPU integrated using through silicon vias (TSVs).



< HBM5 Architecture with CXL >

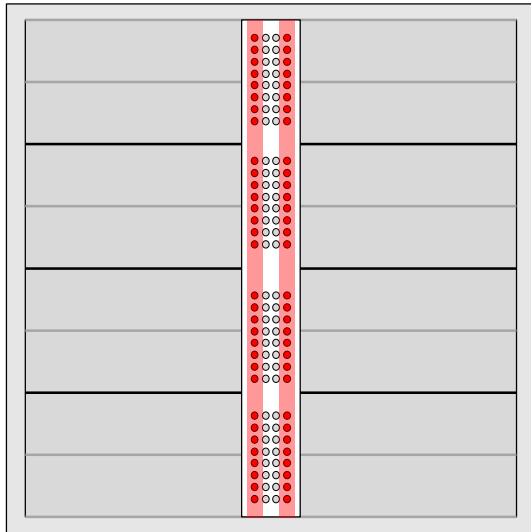
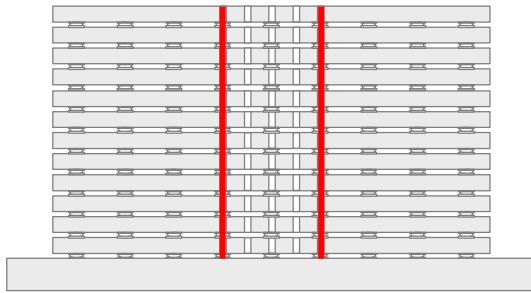
- The custom base die of HBM5 enables direct access to shared memory pool through CXL, providing a single unified memory with improved bandwidth and capacity.



# Next-Generation HBM Roadmap : Distributed Power + Thermal HBM TSV Array Placement

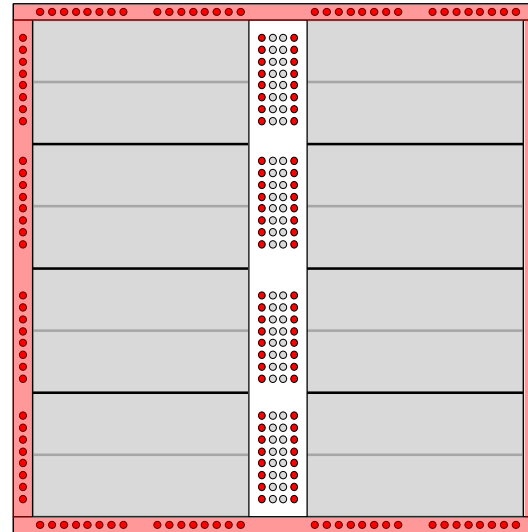
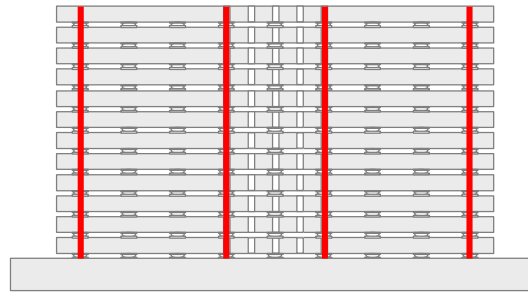
**HBM5  
Architecture**

HBM3



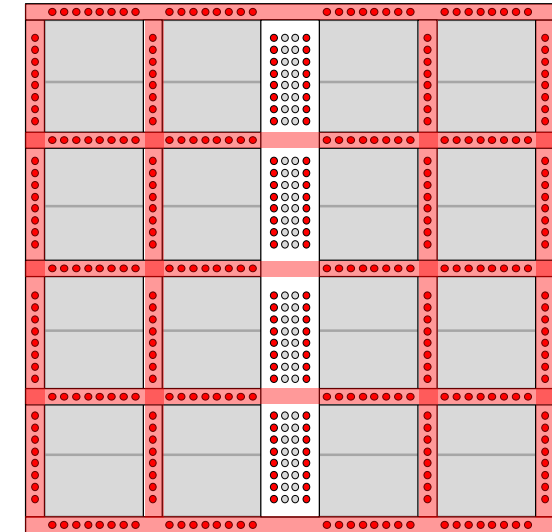
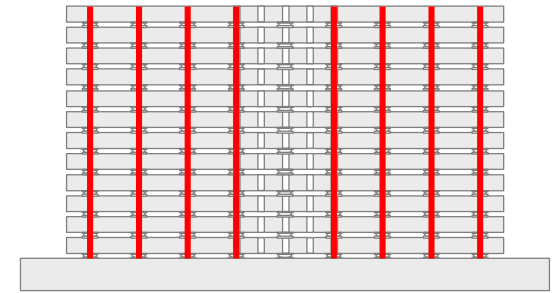
Center Array  
P/G + Thermal TSV

HBM4

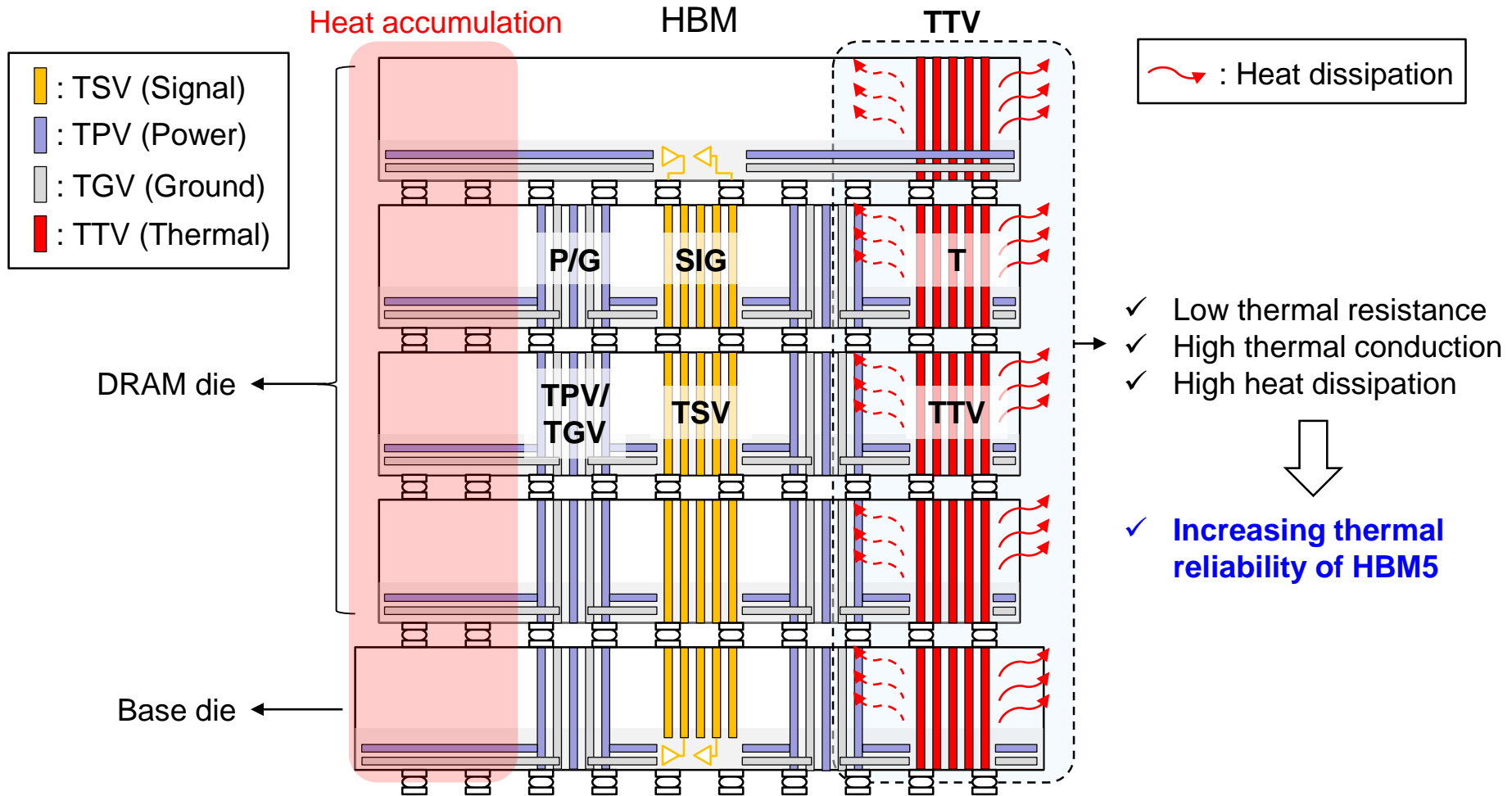


Center Array + Ring  
P/G + Thermal TSV

HBM5 ~ HBM8



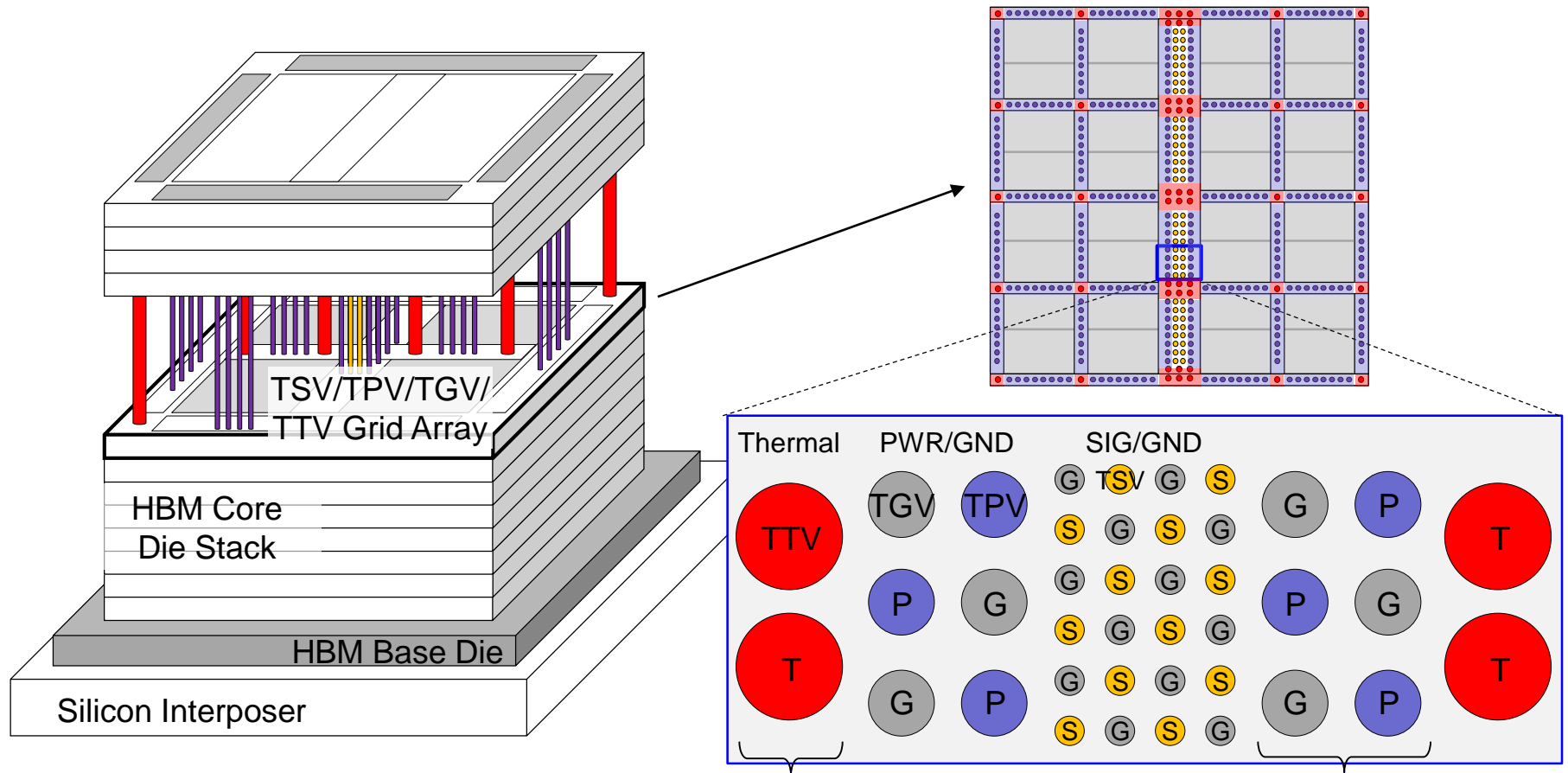
Distributed Grid Array  
P/G + Thermal TSV



## < Through Thermal Via (TTV) for Increasing Heat Dissipation in HBM5 >

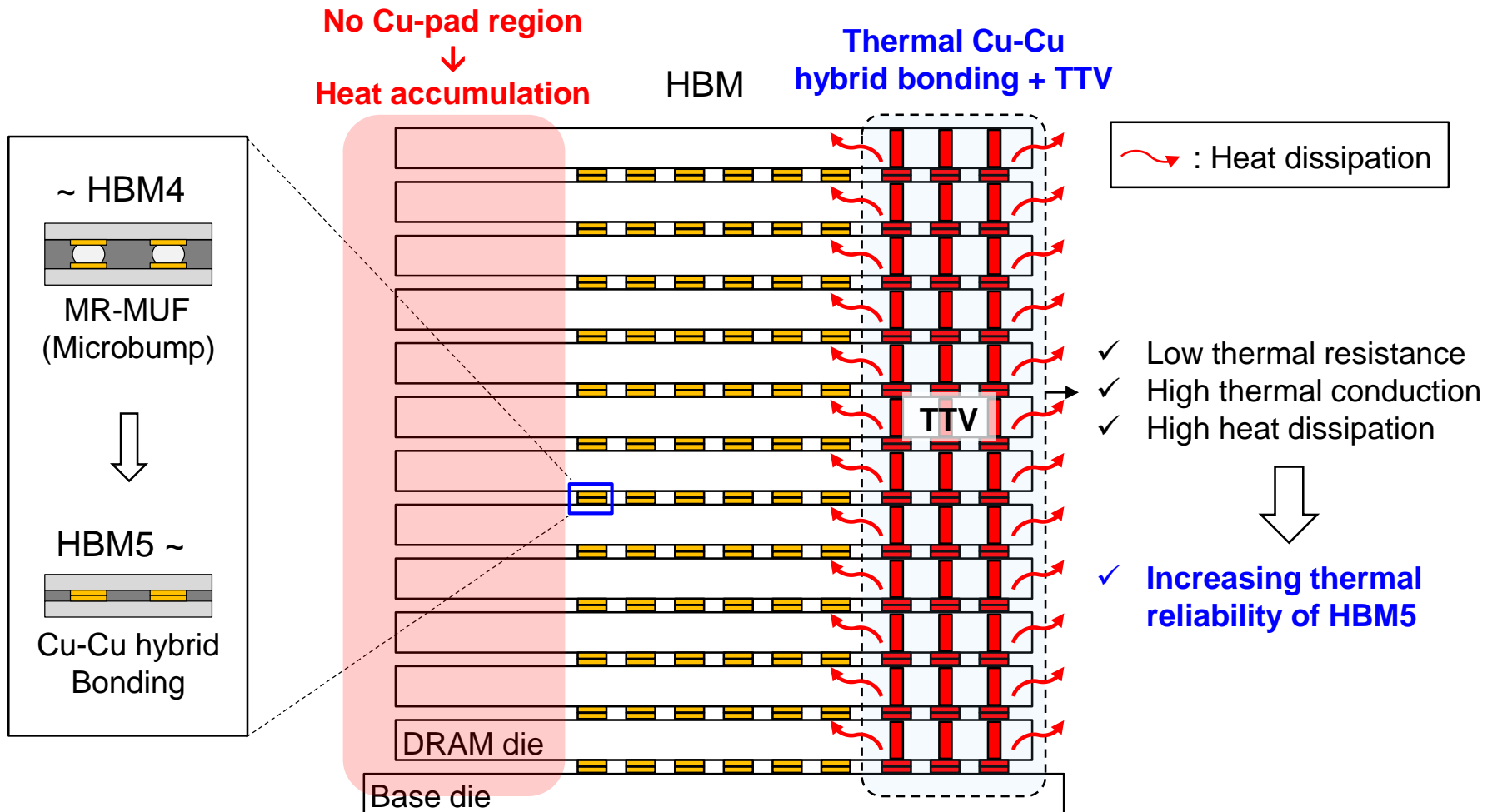
- Through Thermal Vias (TTVs) are implemented in the unused areas of the HBM silicon substrate, excluding the regions occupied by signal, power, and ground vias.

Distributed S/P/G/T Via Grid Array at HBM5



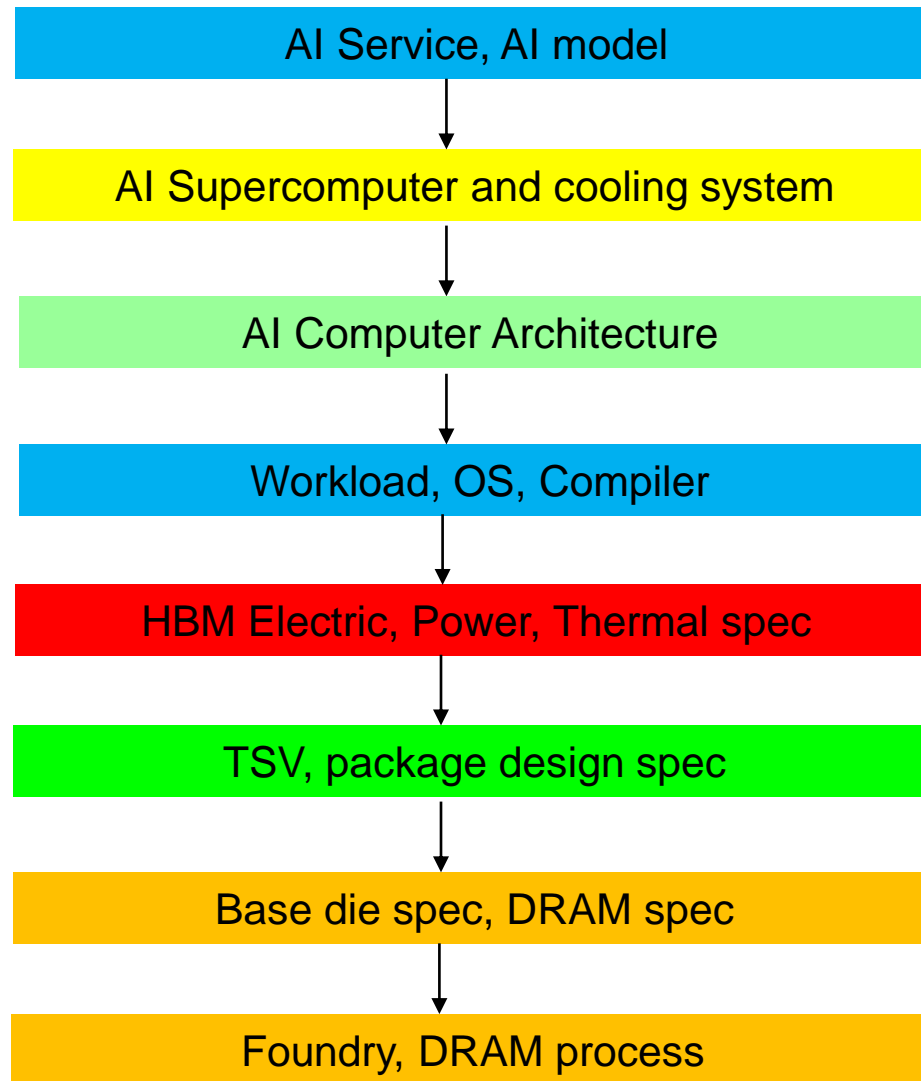
- ✓ Low thermal resistance
- ✓ High thermal conduction
- ✓ Low inductance
- ✓ Low electrical resistance

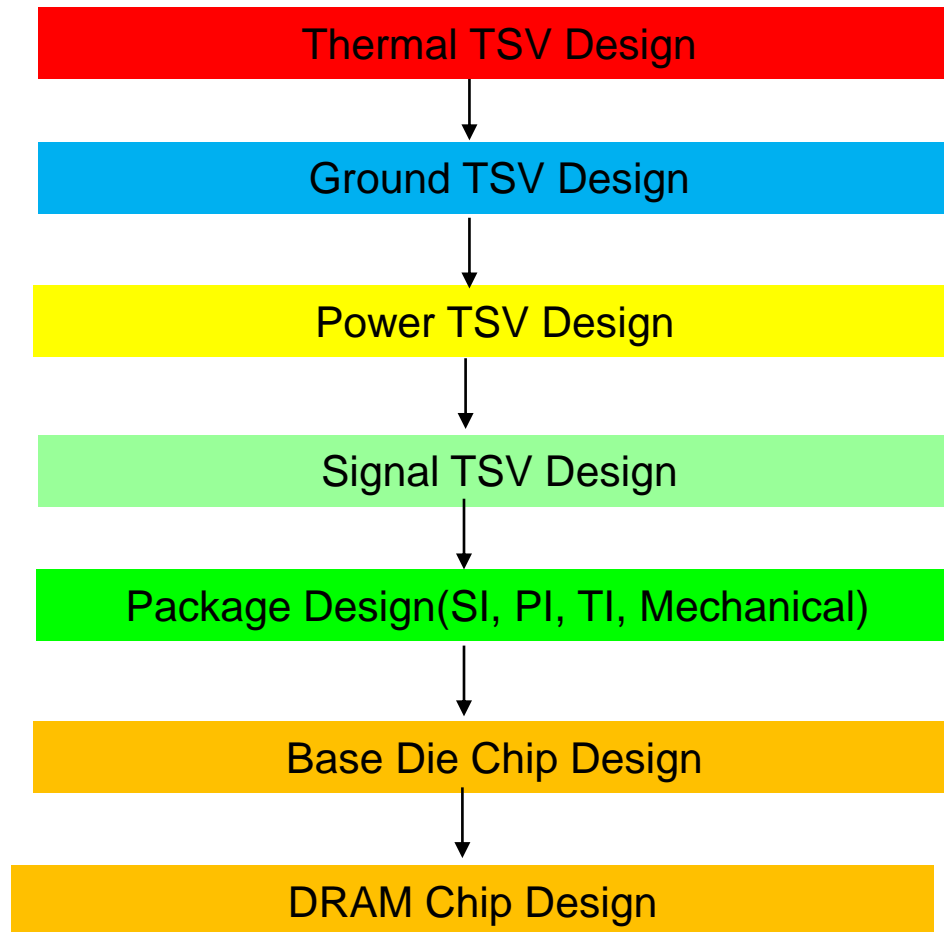
< Asymmetric Distributed TSV/TPV/PGV/TTV Grid Array Design for HBM5 >

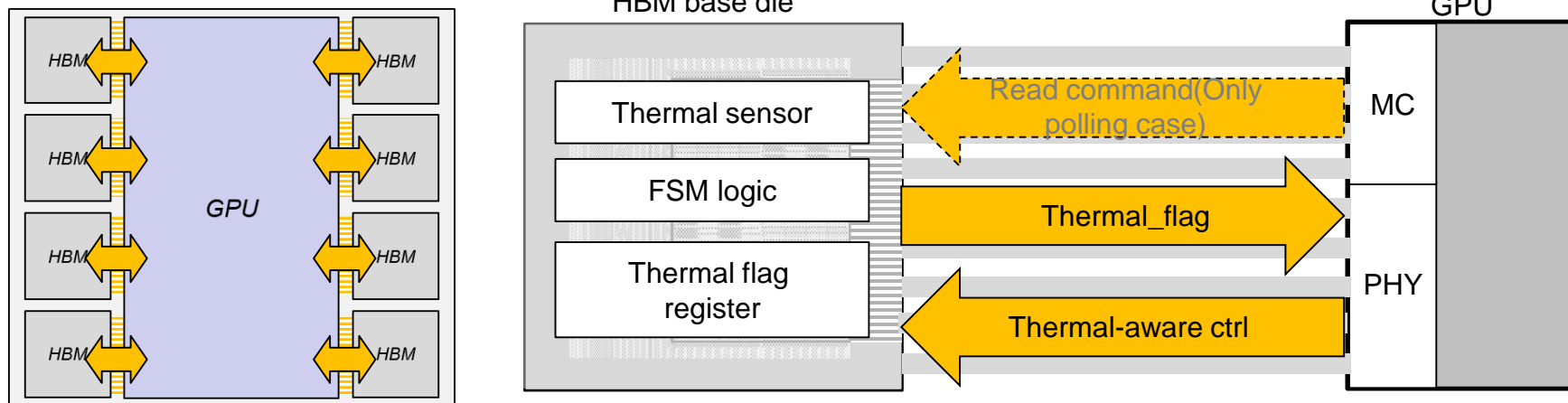


## < Thermal Cu-Cu Hybrid Bonding with TTVs for Thermal Reliability in HBM5 >

- Thermal Cu-Cu hybrid bondings are implemented on no Cu-pad (no bump) region of HBM except S/P/G Cu-pad.
- Thermal Cu-Cu hybrid bondings are interconnected to TTVs to maximize heat dissipation of HBM.







## <HBM base die thermal sensing and FSM-based control flag interface>

- HBM base die에는 Thermal sensor와 FSM이 통합되어 있으며, 온도를 실시간으로 모니터링해 thermal level (Level 0/1/2)을 판단하고 해당 상태에 따라 내부 thermal flag register에 값을 기록함
    - ✓ Base die 내 thermal-aware block 예상 면적:약 0.02–0.05 mm<sup>2</sup> 이하 < 0.5% 추정
  - GPU 내 Memory Controller는 다음 두 가지 방식 중 하나로 thermal flag를 인지하여 정책을 적용할 수 있음
    - ✓ Polling 방식: Controller가 HBM command bus를 통해 주기적으로 register를 읽는 구조
    - ✓ Push 방식: Base die FSM에서 상태 변화 시 thermal\_flag를 data path에 실어 controller로 즉시 전달 (interrupt-like signaling)
- 이 구조는 JEDEC burst protocol을 유지하면서도, thermal 대응을 위한 light-weight signaling 경로로 사용 가능함.



- The proposed Thermal-Flow-Uniformized (TFU)-based immersion cooling structure provides a uniform cooling solution to the GPU-HBM module by submerging it in a immersion tank.
- The proposed TFU is designed to ensure uniform flow in order to maintain temperature uniformity across the high-dense GPU-HBM module.



## Part5: Key Features in HBM6

# Key Features of HBM6

## 1. Electrical Specification

- Data Rate : 16 Gbps
- Number of I/Os : 4,096
- Total Bandwidth : 8.0 TB/s
- Number of die stack : 16/20-Hi
- Capacity/die : 48 Gb

## 2. Packaging/Cooling Method

- Bump-less Cu-Cu Direct Bonding
- Immersion Cooling

## 3. HBM Architecture

- Custom Multi-tower HBMs
- Active / Hybrid (Silicon+Glass) Interposer
- Network Switch + Bridge Die
- BS-PDN

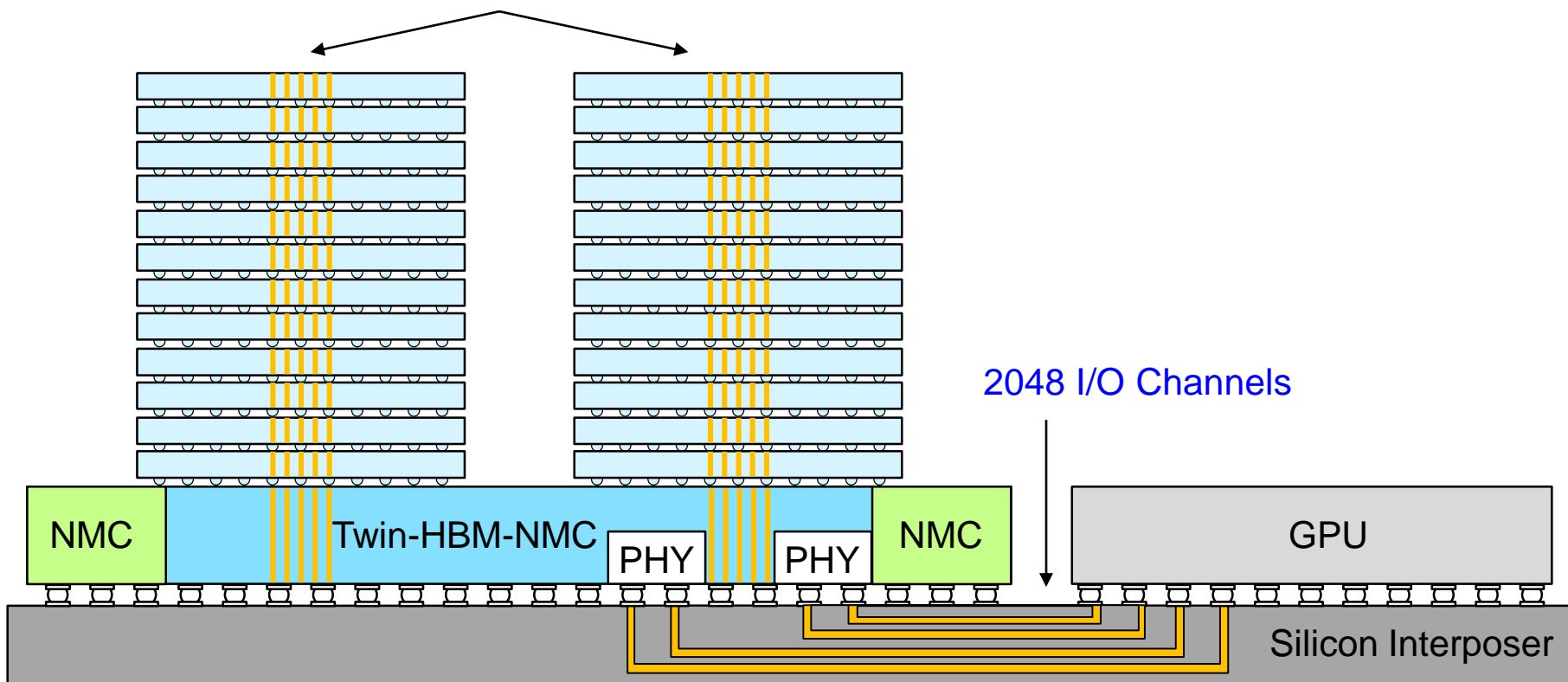
## 4. AI Design Agent

- Hybrid Equalizer + Generative AI based SI/PI Metric Estimation
- AGI (Artificial General Intelligence)

# Proposal of Twin Tower High Bandwidth Memory with Near-Memory Computing (Twin-HBM-NMC) Architecture

HBM6  
Architecture

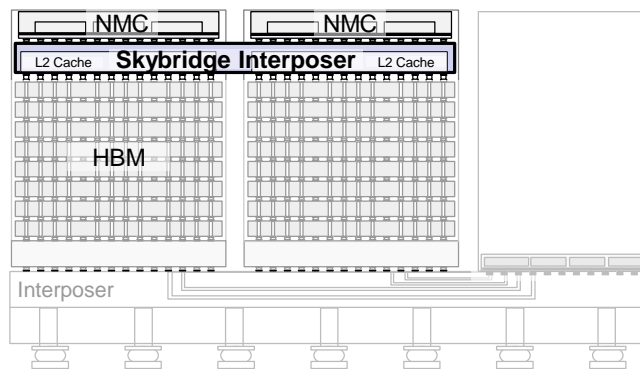
Two DRAM Stacks



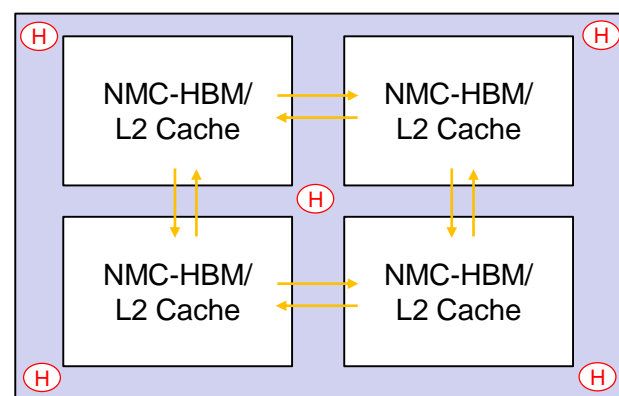
< Overview of Twin Tower HBM with NMC Architecture >

- In twin tower high bandwidth memory with near-memory-computing (Twin-HBM-NMC) architecture, two DRAM stacks are located on top of the large logic die.
- The logic die include NMC units and is connected to the GPU via 2048 interposer channels.

# Multi HBM-HBM Skybridge Interposer for Efficient Near Memory Computing Performance



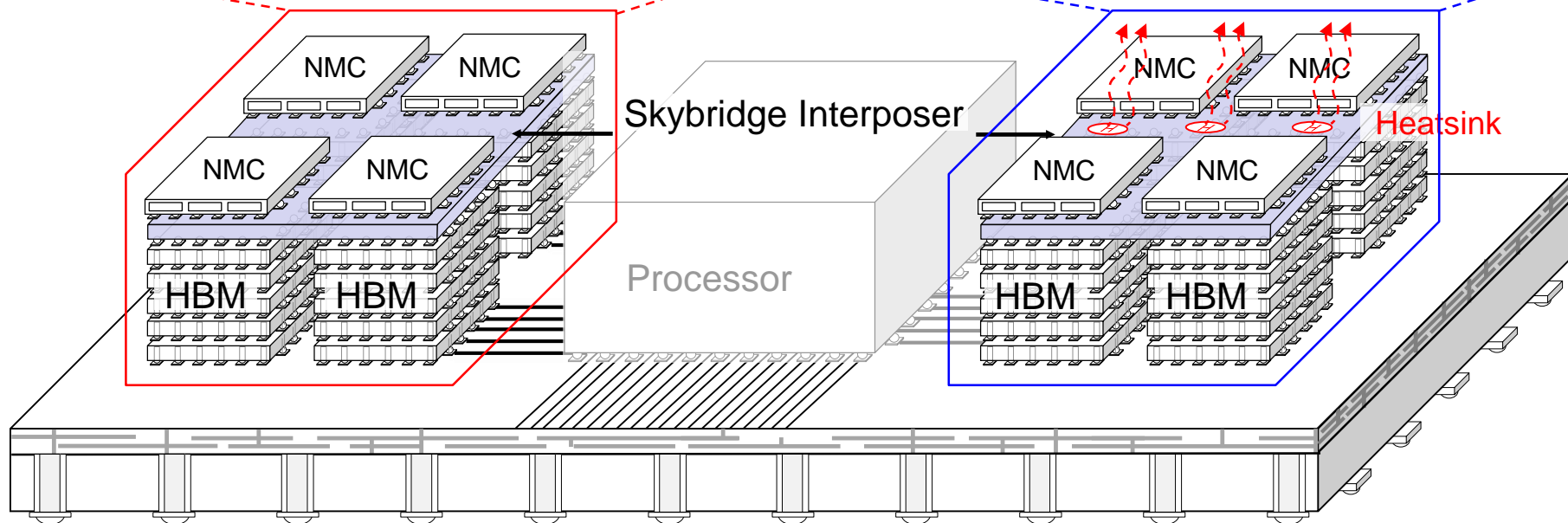
< Side view of Skybridge Interposer Architecture >

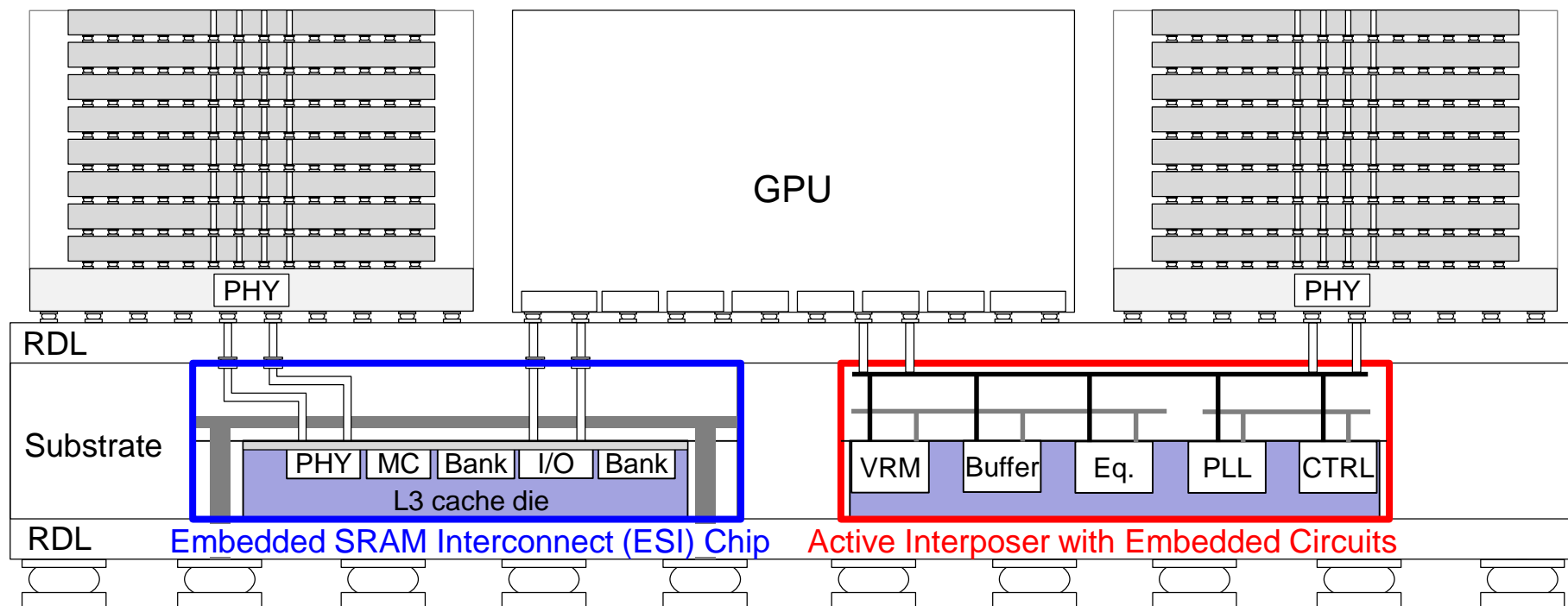


< Top view of Skybridge Interposer >

Through  
Skybridge  
Interposer...

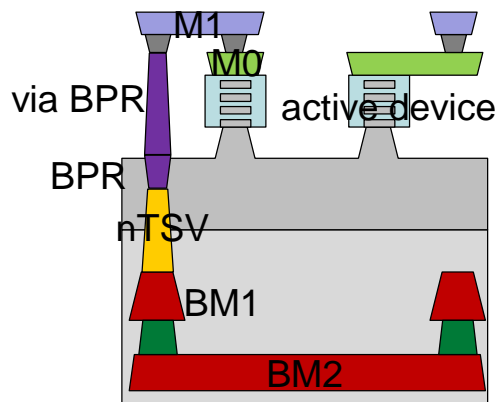
- ✓ NMC-NMC
- ✓ HBM-HBM
- ✓ NMC-HBM
- ✓ L2 cache
- ✓ Heatsink



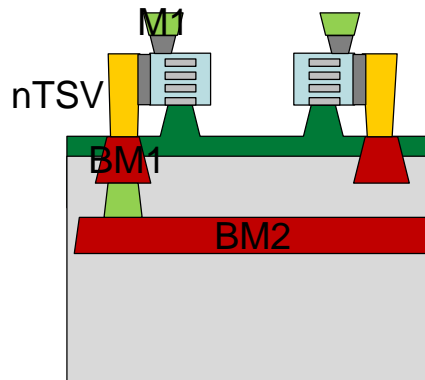


## < Embedded GPU-HBM Architecture with Active Interposer >

- The local silicon interconnect (LSI) die and L3 cache die is hybrid bonded to form embedded SRAM interconnect (ESI) chip, which is placed inside the interposer using COWOS-L technology.
- Embedded circuit components including VRM, equalizer, PLL, and control circuits are embedded in active interposer to enhance electrical performance.

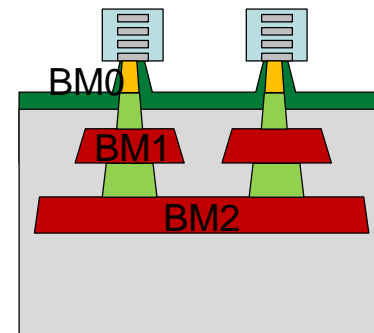


Buried Power Rail



Power Via

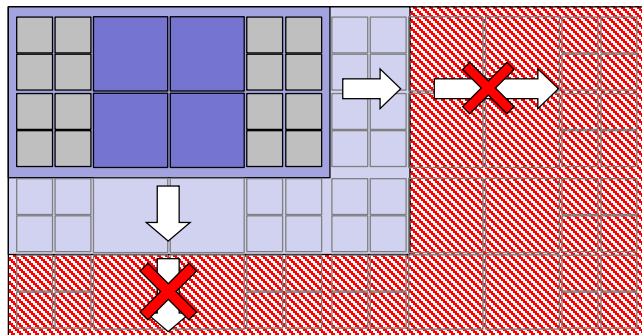
< 기본적인 BSPDN 구조 >



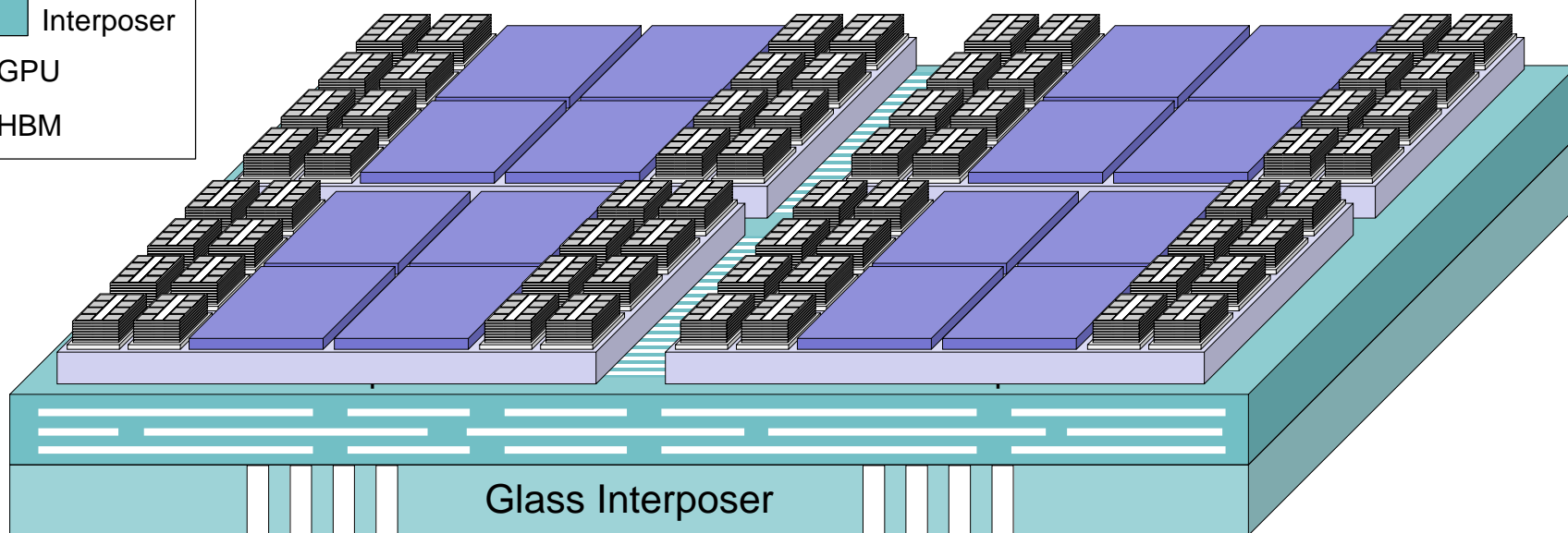
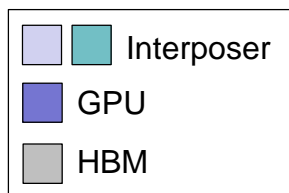
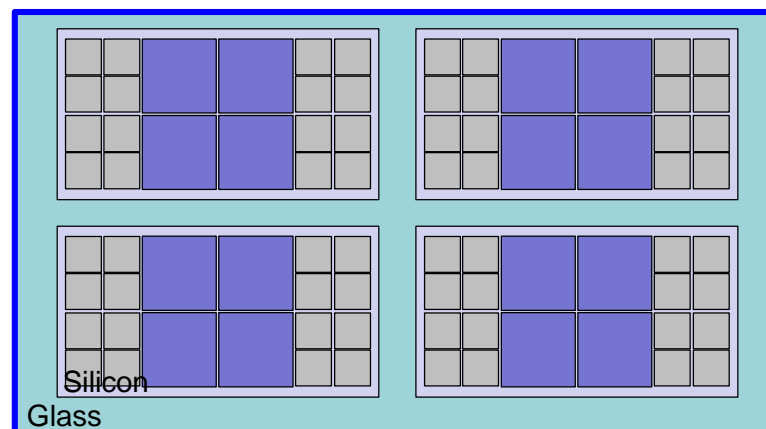
Super Power Rail

- 산업에서 개발된 Active 소자와 backside metal을 interconnection 하는 방식의 차이로 총 3가지 구조 Buried Power Rail(BPR), Power Via, Super Power Rail(SPR)로 나뉘어 짐
  - BPR : backside metal – nTSV - BPR – VBPR – frontside metal(M0) – active devices
  - Power Via : backside metal(BM1) – nTSV – active devices
  - SPR : backside metal(BM1) – BM0 – active devices

Limitation in increasing  
silicon interposer die size



Large Scale Hybrid (Silicon+Glass) interposer



< Silicon-Glass Hybrid Interposer for Ultra Large-Scale Next Generation HBM >

## Part6: Key Features in HBM7



# Key Features of HBM7

## 1. Electrical Specification

- Data Rate : 24 Gbps
- Number of I/Os : 8,192
- Total Bandwidth : 24.0 TB/s
- Number of die stack : 20/24-Hi
- Capacity/die : 64 Gb

## 2. Packaging/Cooling Method

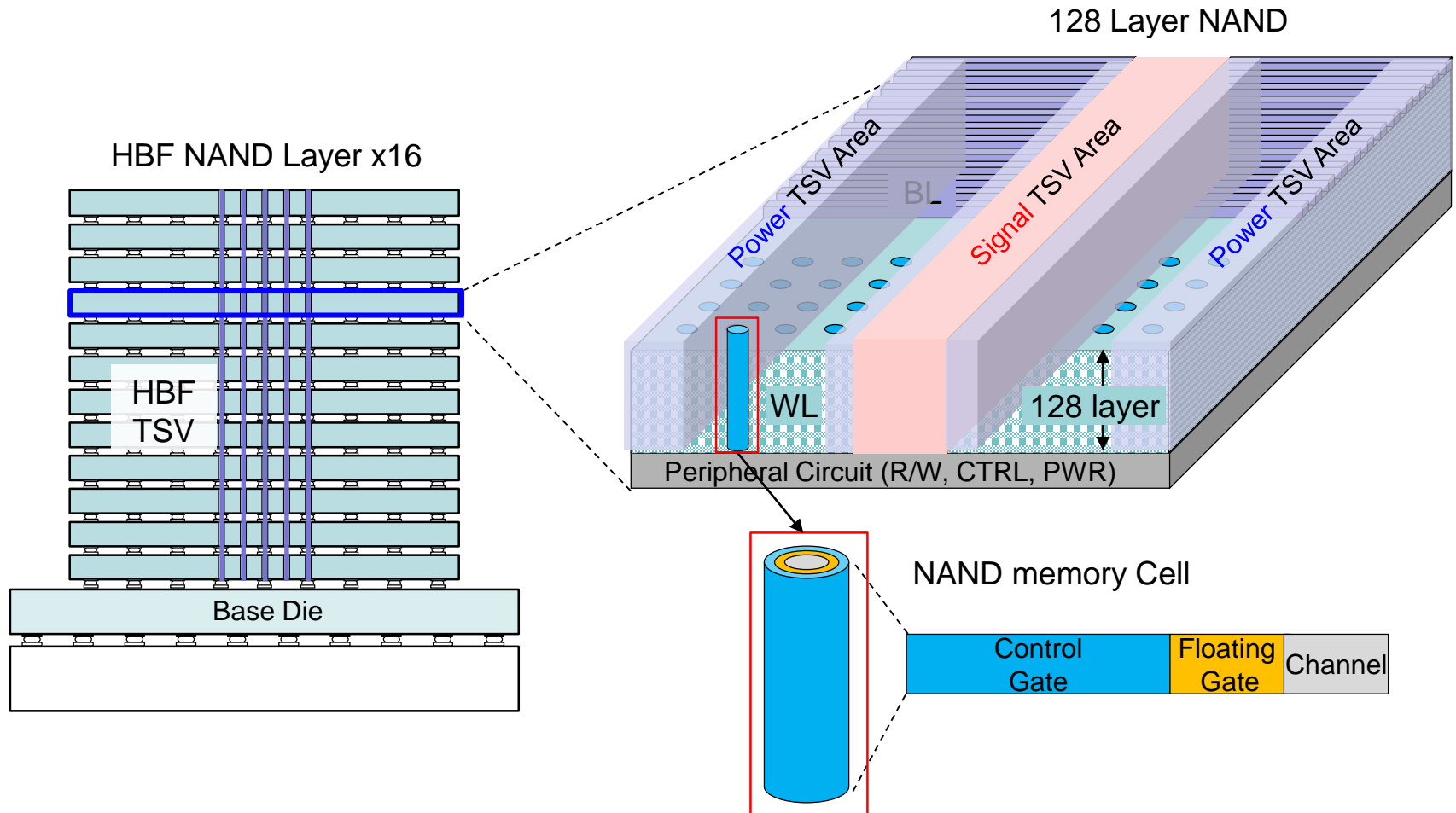
- Bump-less Cu-Cu Direct Bonding
- Embedded Cooling

## 3. HBM Architecture

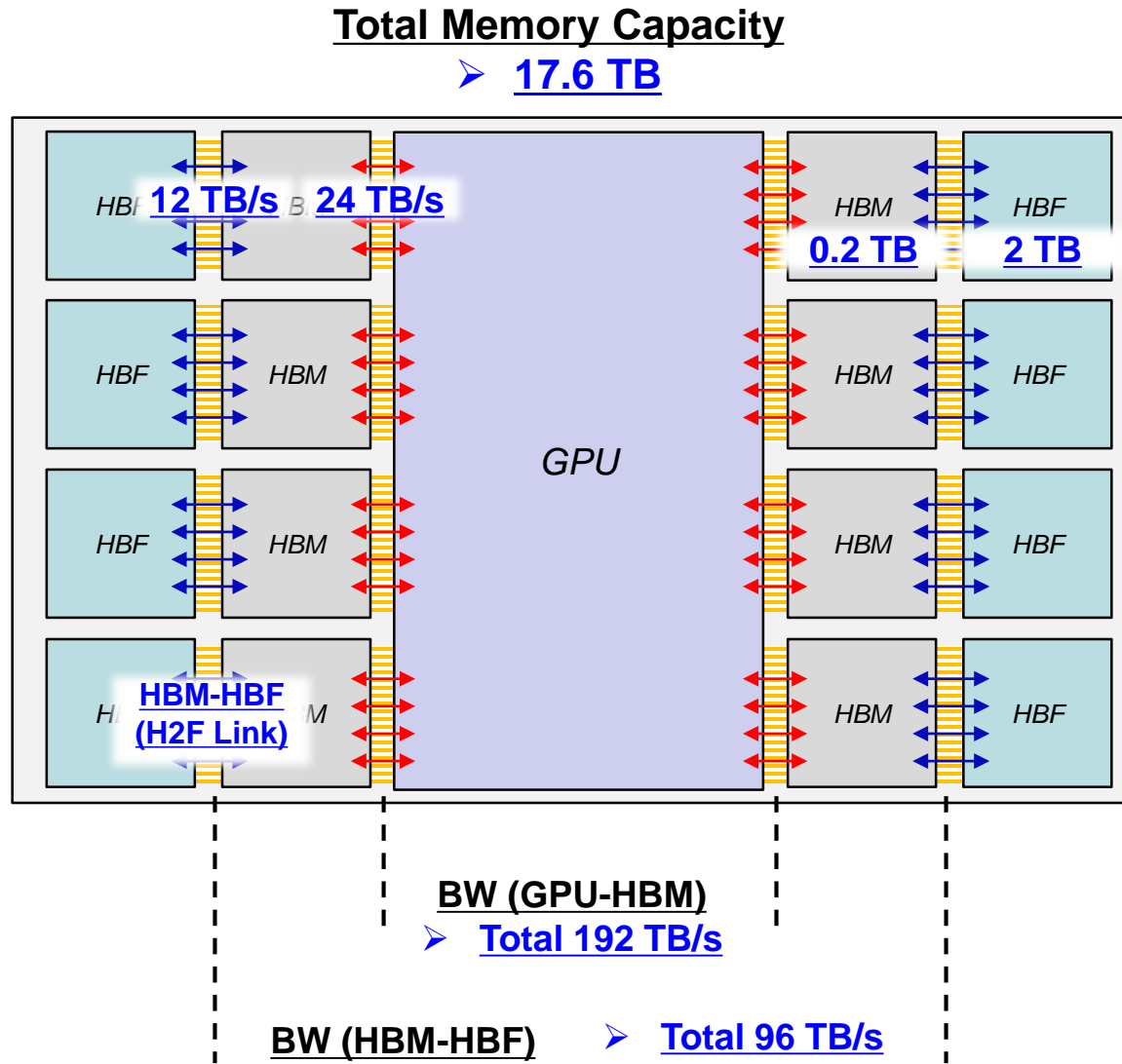
- Hybrid HBM Architecture
- HBM-HBF
- HBM-LPDDR
- Buffer dies in HBM stack

## 4. AI Design Agent

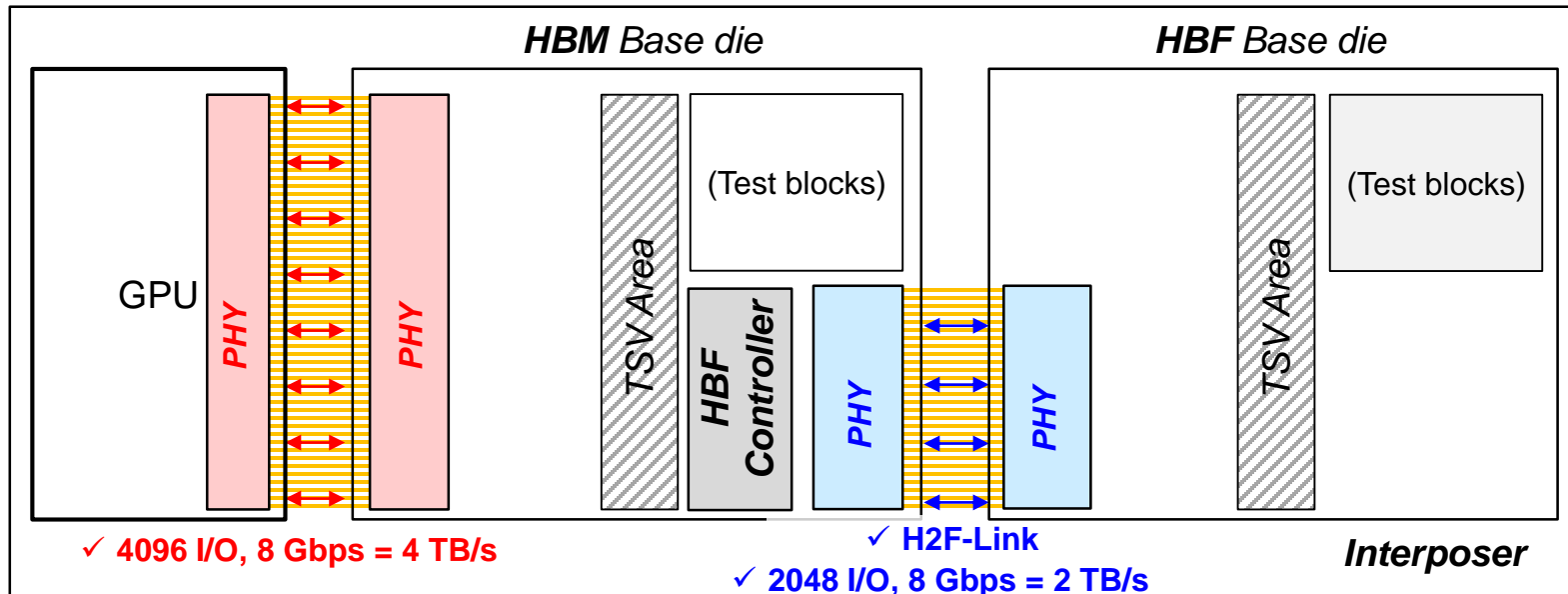
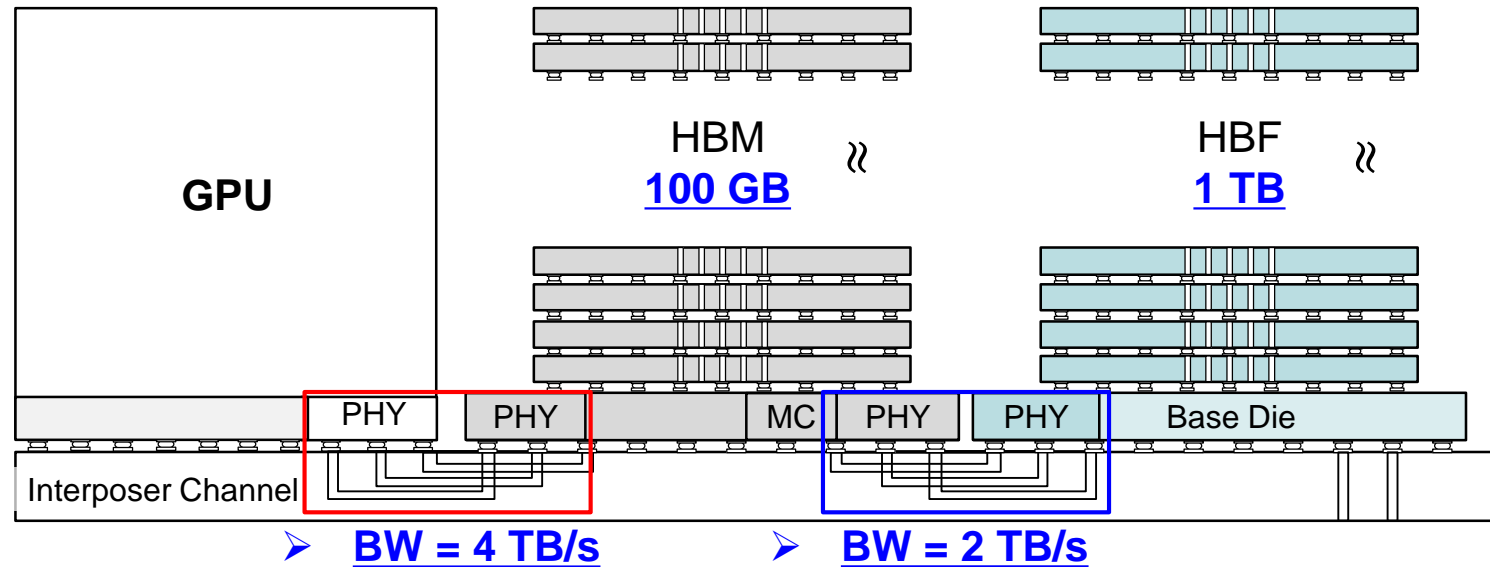
- LLM based Human Interactive AI Design Agent



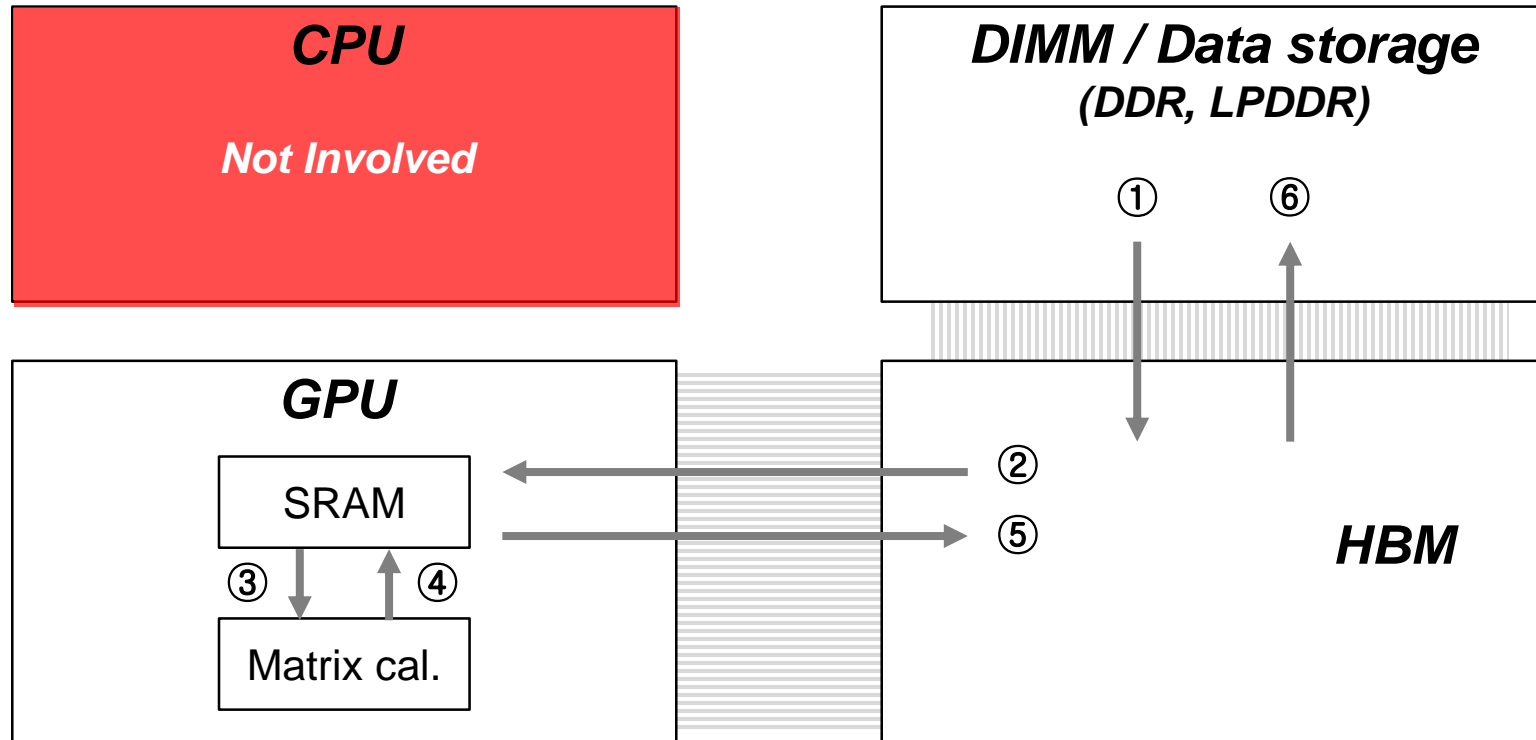
< High Bandwidth Flash (HBF) Architecture for Memory Intensive LLM Inference >



< HBM-HBF Architecture : Top View >

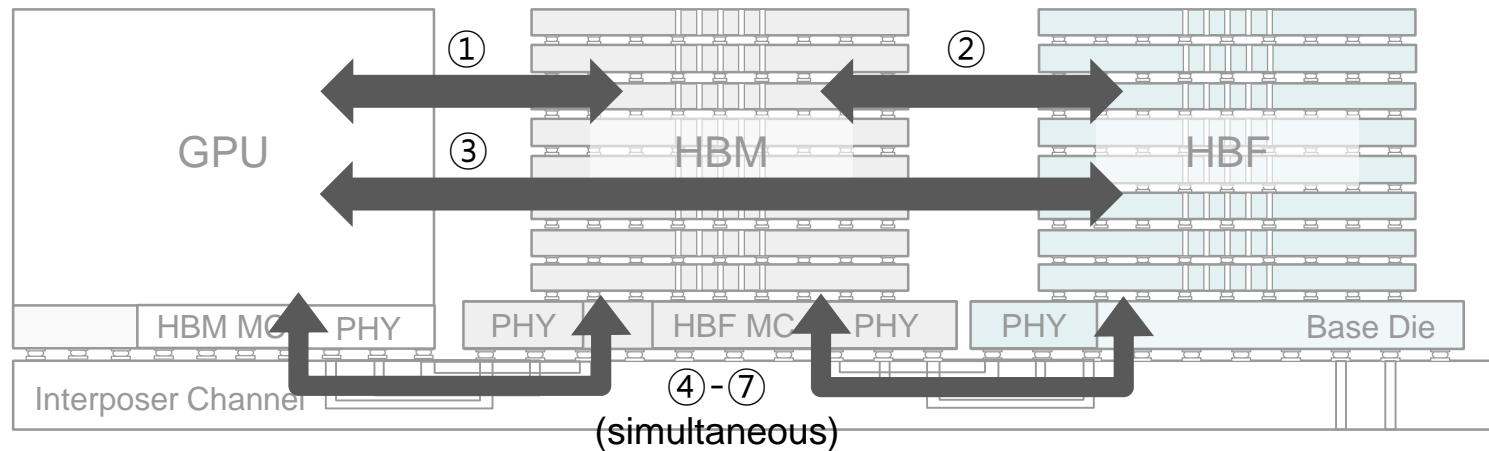


## New Data Flow in HCC with GPU Co-Existence



- ① Data (Input  $x$ , Input  $h$ , Weight  $w_{ij}$ ) from DIMM → copy to → HBM / Path: DIMM → HBM
- ② Data (Input  $x$ , Input  $h$ , Weight  $w_{ij}$ ) from HBM to cache in GPU / Path: HBM → GPU
- ③ Matrix calculation in GPU
- ④ Save matrix multiplication results to SRAM
- ⑤ Save it to HBM / Path: GPU → HBM
- ⑥ Copy it to DIMM / Path: HBM → DIMM

→ Since the **CPU is not involved** in the memory transfer path, **delays can be reduced**, and it has the advantage of **fewer interconnection steps and shorter lengths**.



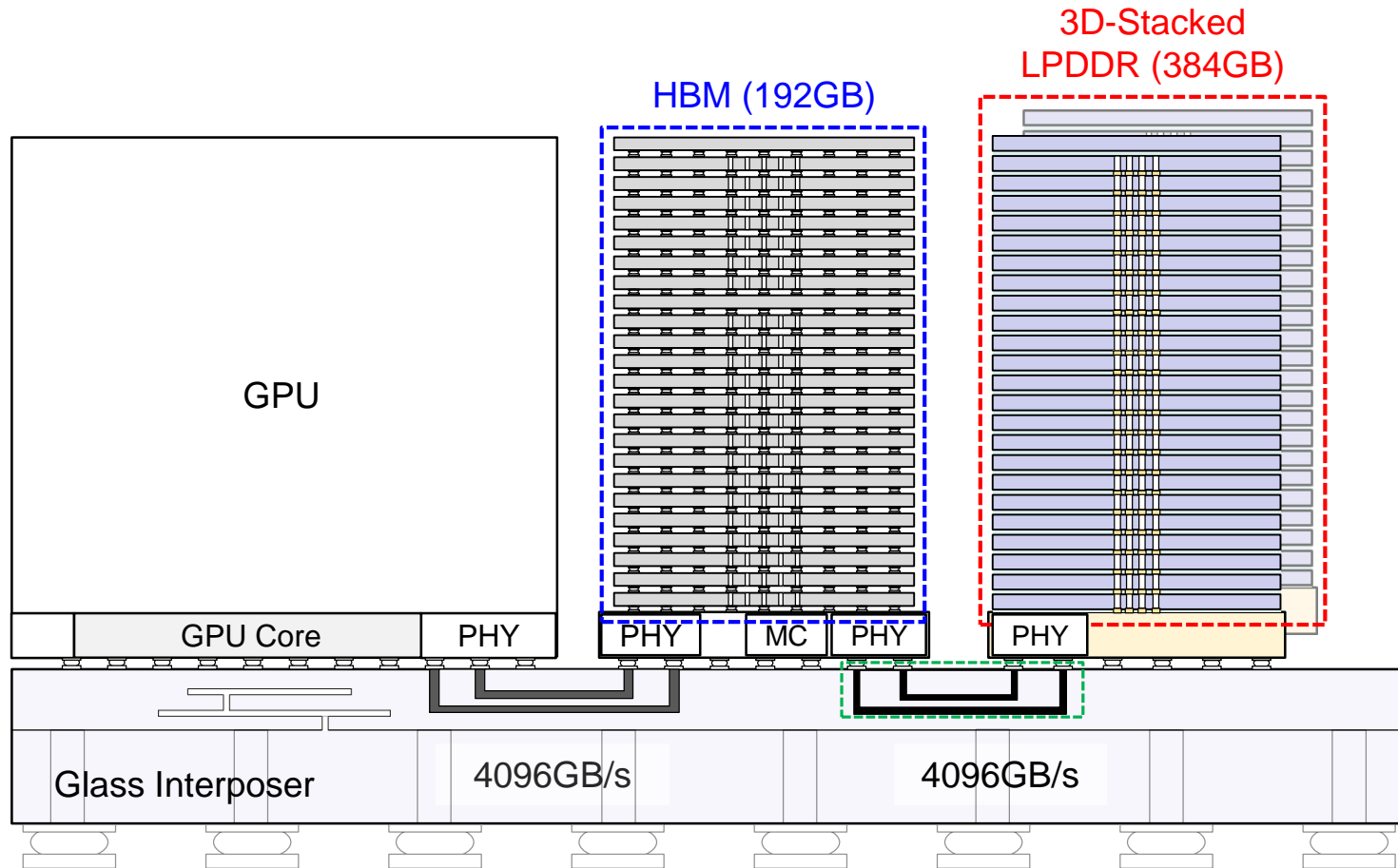
< Side-view of GPU-HBM-HBF Structure and Its Data Path Cases >

- Case 1: GPU ↔ HBM (Read/Write)
  - Case 2: HBM ↔ HBF (Read/Write)
  - Case 3: GPU ↔ HBF (Read/Write)
- Single-Command Execution
- 
- Case 4: GPU → HBM (Write) & HBM ← HBF (Read)
  - Case 5: GPU → HBM (Write) & HBM → HBF (Write)
  - Case 6: GPU ← HBM (Read) & HBM ← HBF (Read)
  - Case 7: GPU ← HBM (Read) & HBM → HBF (Write)
- Dual-Command Execution  
(GPU ↔ HBM ↔ HBF)



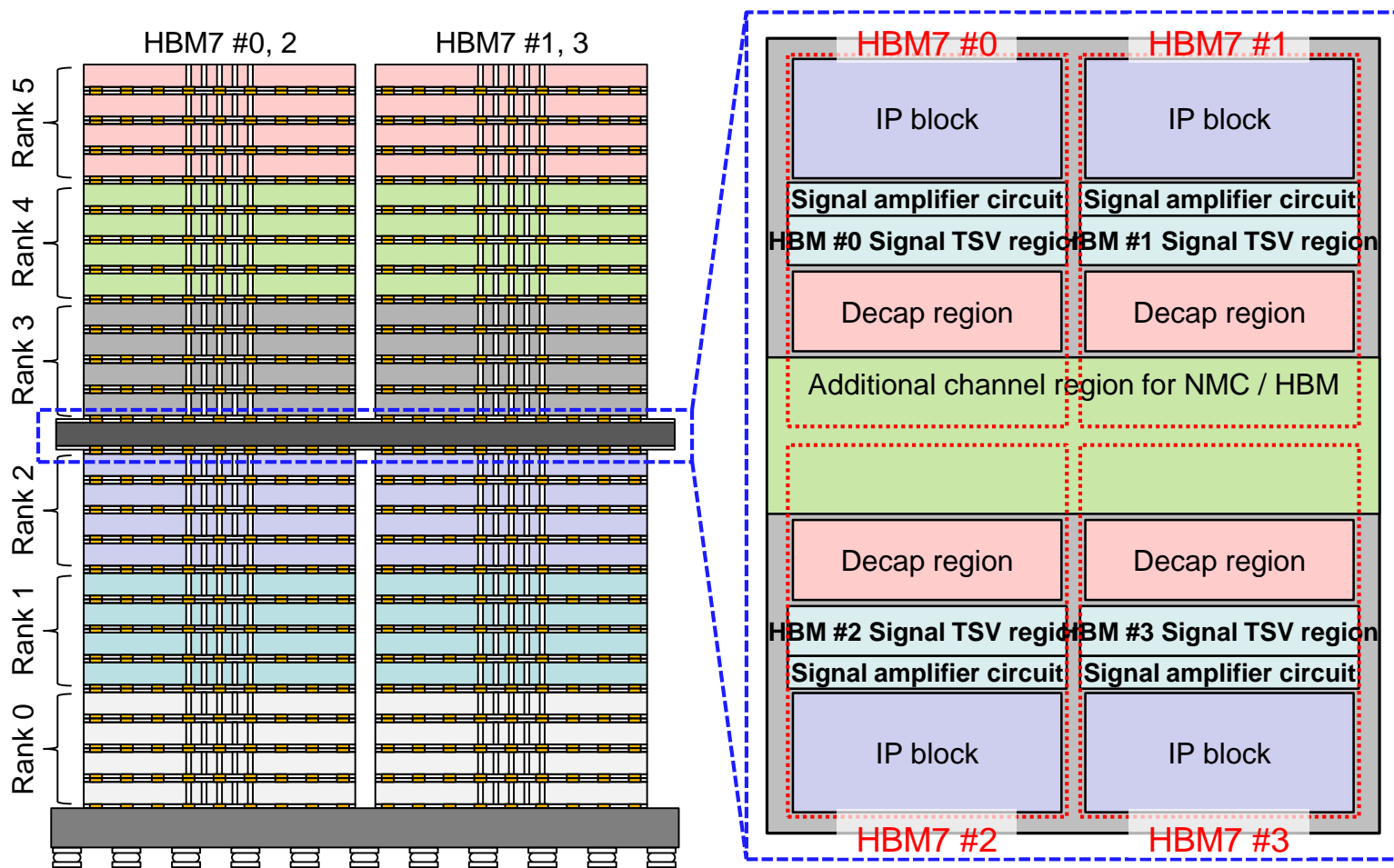
# HBM7 Architecture Integrated with High-Capacity 3D-Stacked LPDDR on Glass Interposer

HBM7  
Architecture

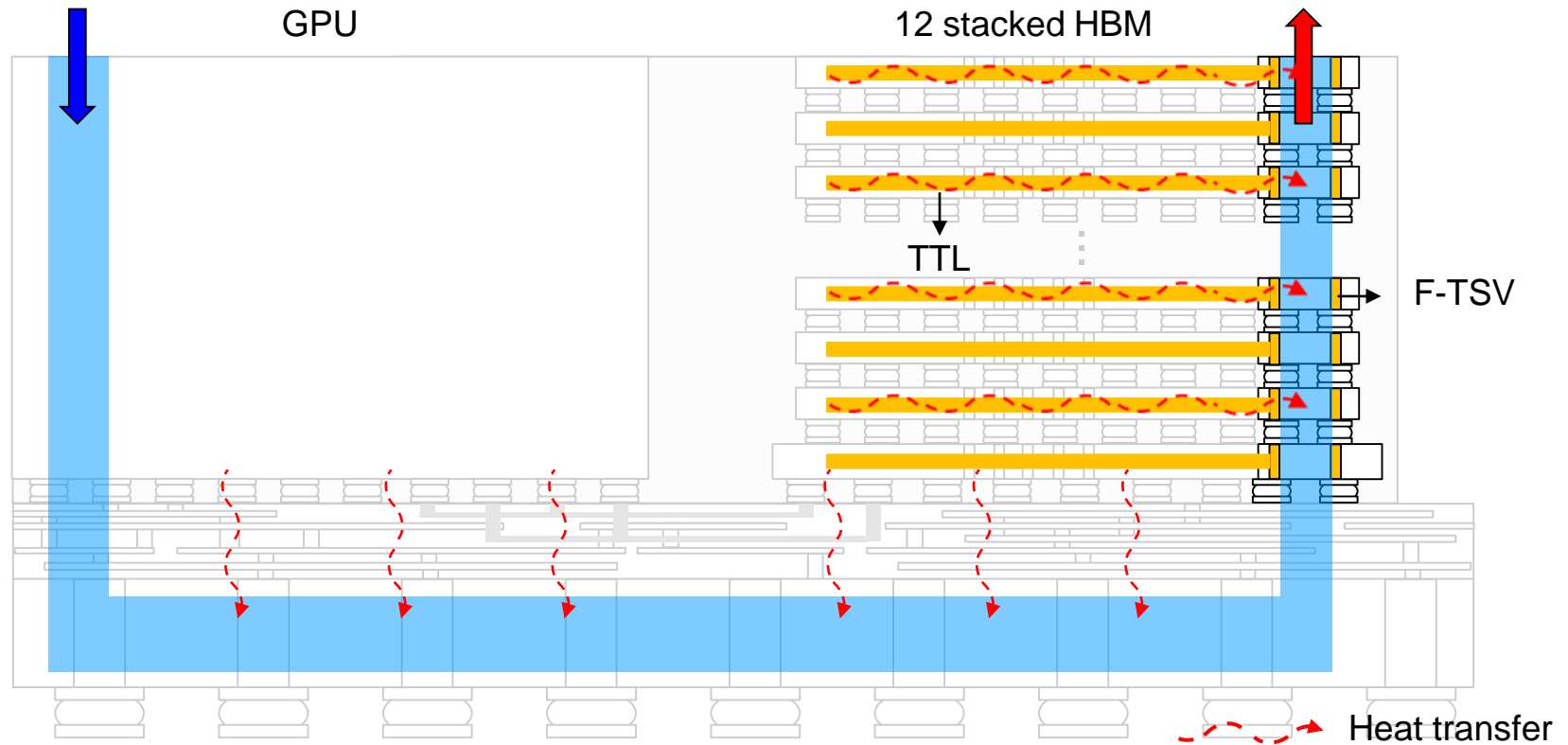


< HBM7 Architecture Integrated with High-Capacity 3D-Stacked LPDDR on Glass Interposer >





< Side and top view of proposed multi functional bridge-die >



< Concept of the proposed embedded cooling structure for GPU-HBM module >

- The proposed Thermal Transmission Line (TTL) and Fluidic TSV (F-TSV) can cool the HBM module efficiently by circulating cooling fluid through the GPU to the interposer and HBM.
- The proposed TTL transfers the internal heat within HBM die to the fluid flowing in F-TSV

## Part7: Key Features in HBM8

# Key Features of HBM8

## 1. Electrical Specification

- Data Rate : 32 Gbps
- Number of I/Os : 16,384
- Total Bandwidth : 64.0 TB/s
- Number of die stack : 20/24-Hi
- Capacity/die : 80 Gb

## 2. Packaging/Cooling Method

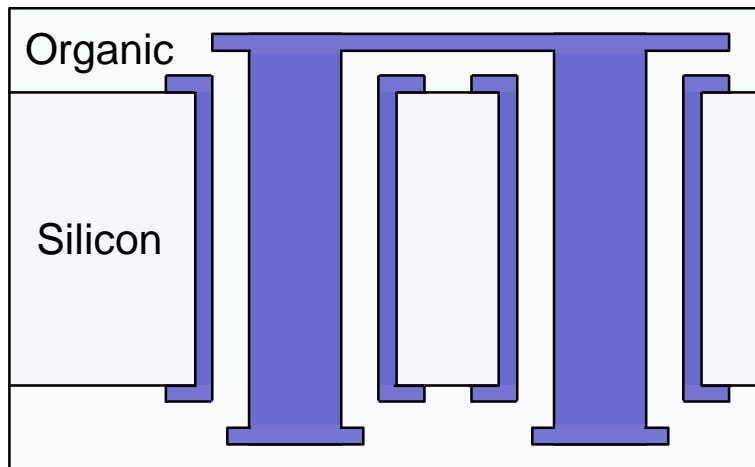
- Bump-less Cu-Cu Direct Bonding
- Embedded Cooling

## 3. HBM Architecture

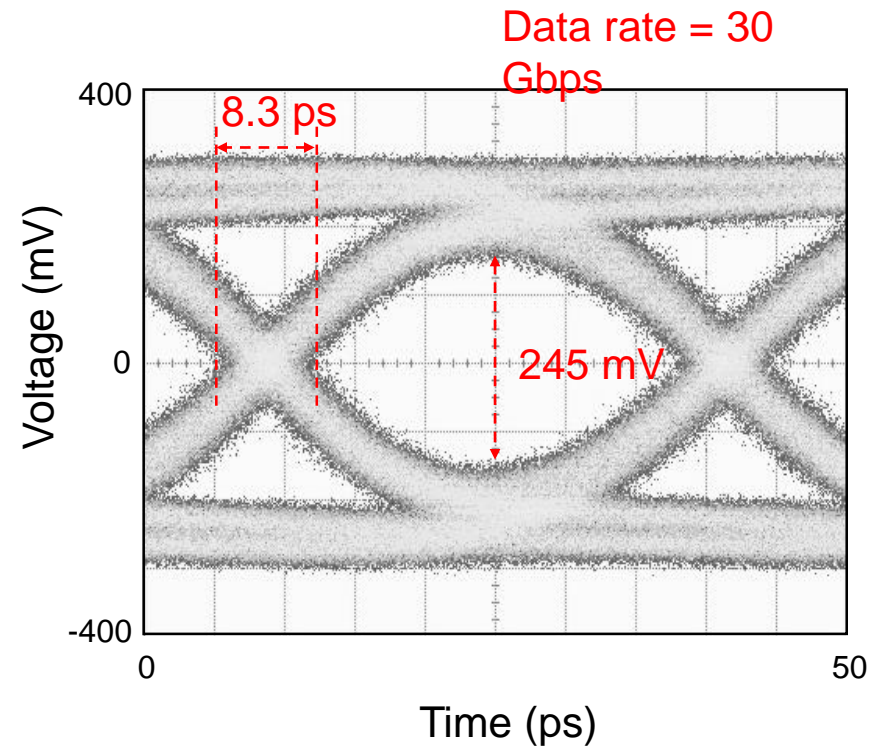
- Coaxial TSV
- Full-3D GPU-HBM
- HBM Centric Computing
- Full Memory Network
- Double Sided Interposer

## 4. AI Design Agent

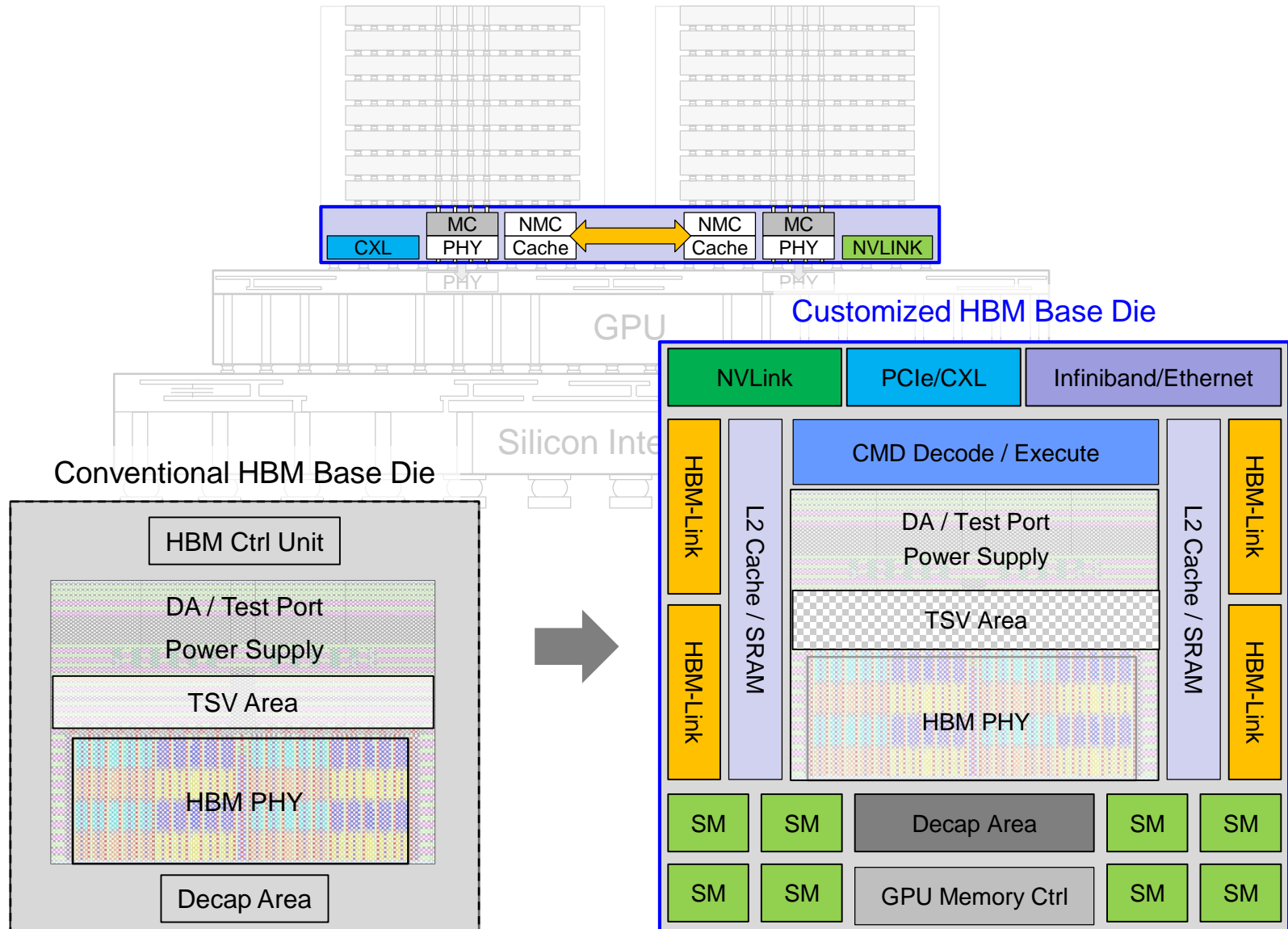
- LLM based Human Interactive AI Design Agent

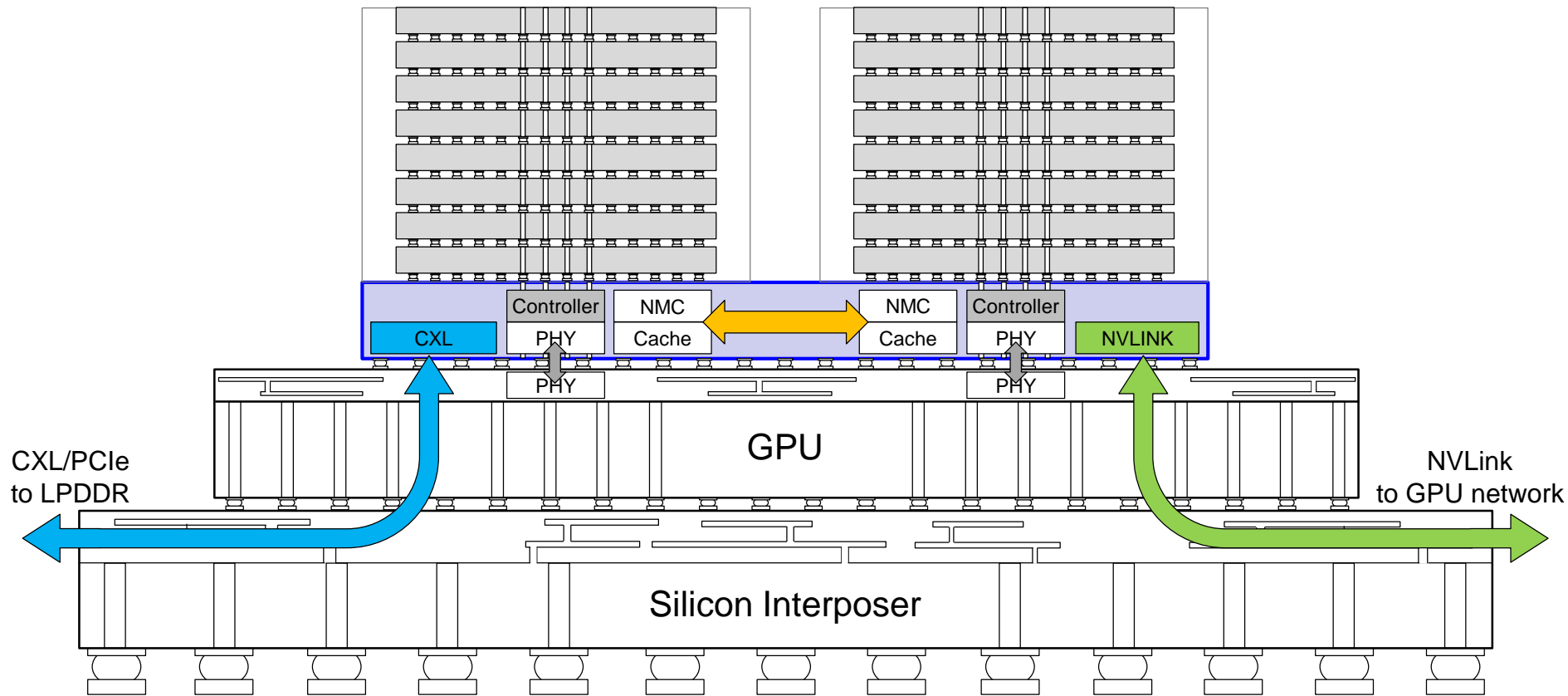


< Coaxial Organic Line Via (COLV) >



< Eye-diagram of Coaxial TSV channel >



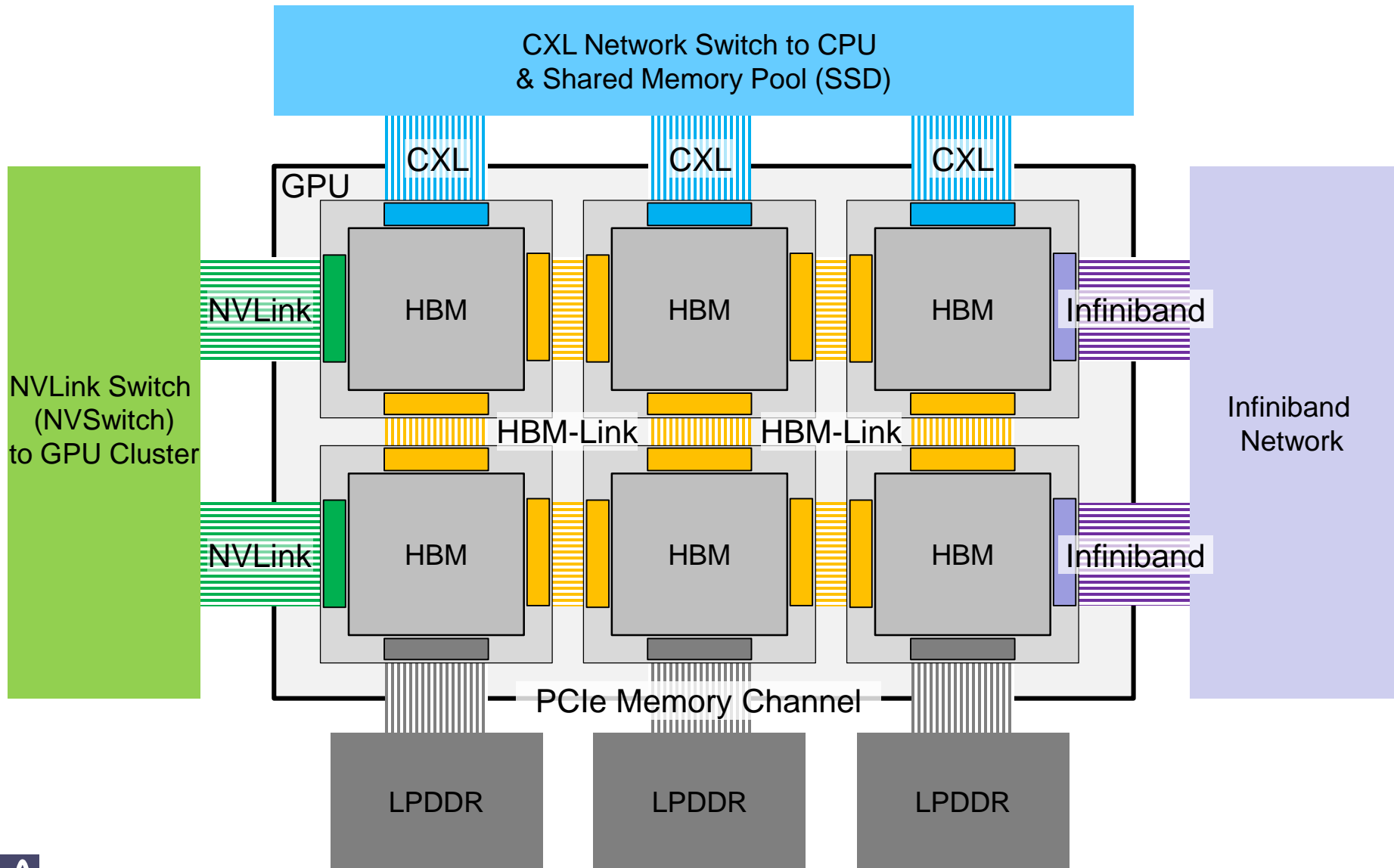


< HBM7 : Full 3D Integration of HBM and GPU with Custom Base Die >

- Through design customization between memory and processor companies, HBM7 is expected to be integrated directly above the GPU(processor).

# Next-Generation HBM Roadmap : HCC Architecture with System-Level Memory Network

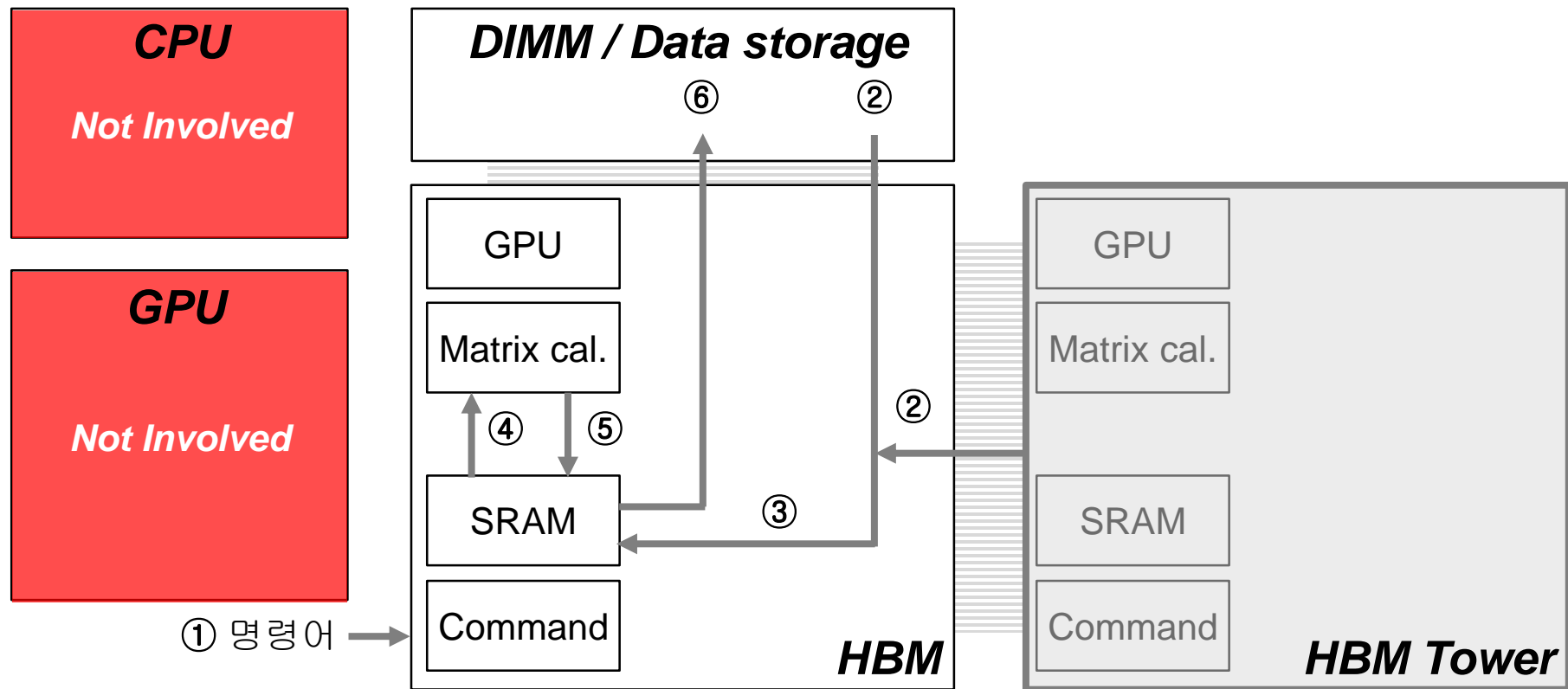
HBM8  
Architecture



< System-Level Memory Network for HCC Architecture >



## New Data Flow in HCC with HBM Centric Computing

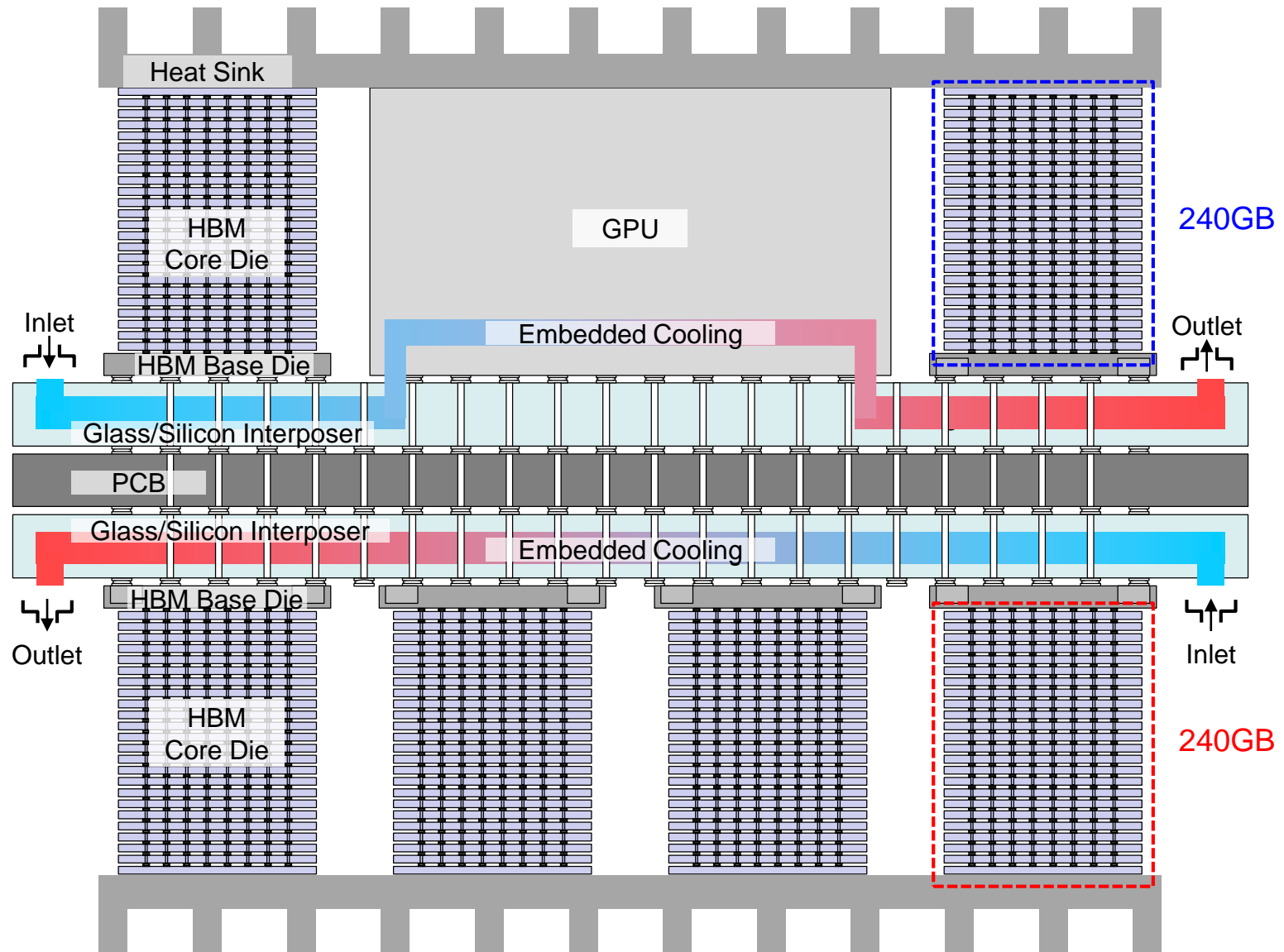


- ① Command decoder
- ② Data (Input  $x$ , Input  $h$ , Weight  $w_{ij}$ ) from HBM tower or DIMM → copy to → HBM
- ③ Data (Input  $x$ , Input  $h$ , Weight  $w_{ij}$ ) from HBM to SRAM in base die
- ④ Matrix calculation
- ⑤ Save it to SRAM in base die
- ⑥ Save it to DIMM

→ With no **CPU** and **GPU** in the memory transfer path, *delays are greatly reduced*, and there are *very few interconnection steps with a much shorter length*.

# 3D Memory Expansion Architecture with Embedded Cooling Structure for HBM8 with Double-Sided Interposer [1/3]: GPU-HBM-HBM

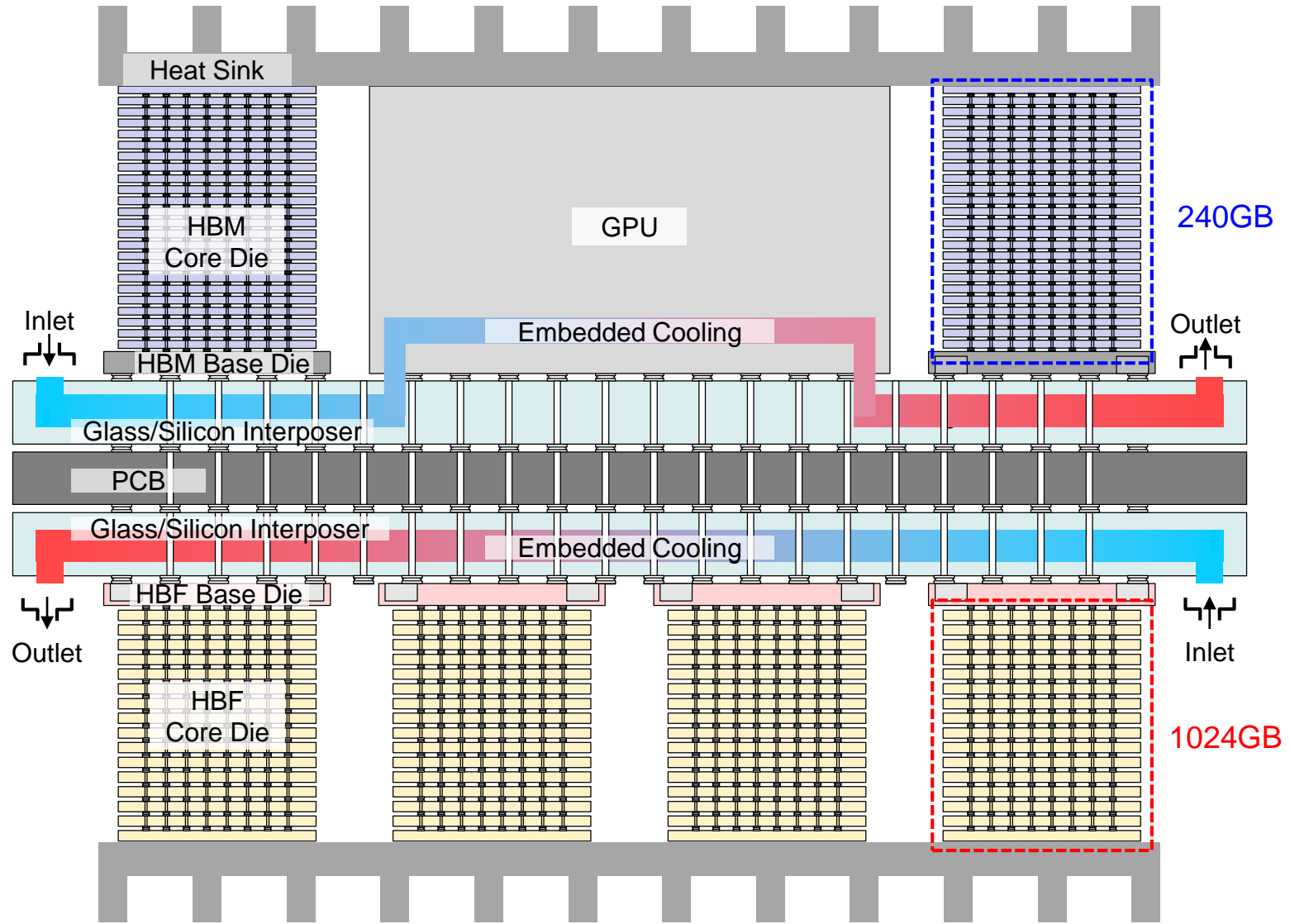
HBM8  
Architecture



< 3D Memory Expansion Architecture with Embedded Cooling for HBM8 with Double-Sided Interposer using HBM >

# 3D Memory Expansion Architecture with Embedded Cooling Structure for HBM8 with Double-Sided Interposer [2/3]: GPU-HBM-HBF

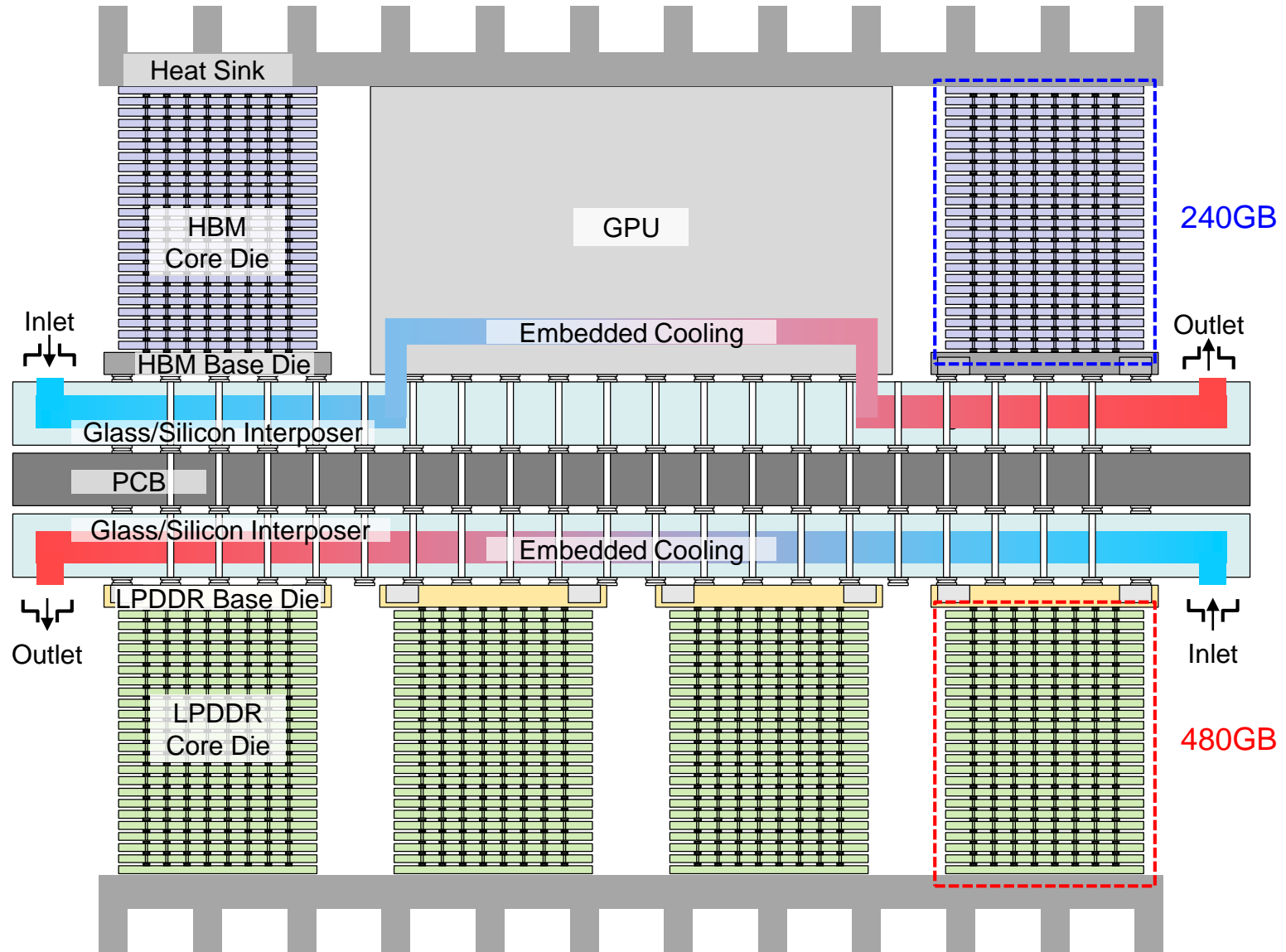
HBM8  
Architecture



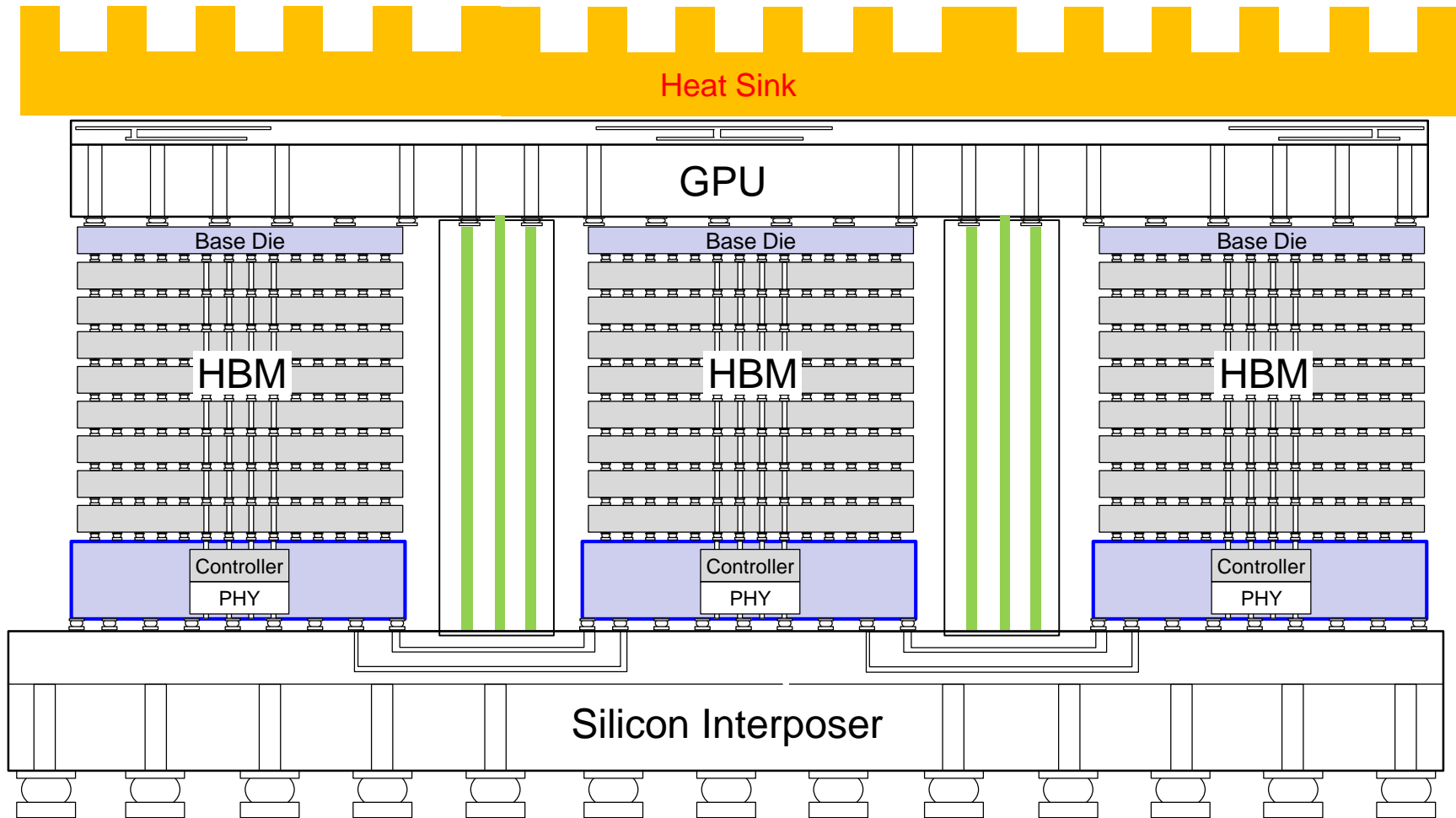
< 3D Memory Expansion Architecture with Embedded Cooling for HBM8 with Double-Sided Interposer using HBF >

# 3D Memory Expansion Architecture with Embedded Cooling Struct for HBM8 with Double-Sided Interposer [3/3]: GPU-HBM-LPDDR

HBM8  
Architecture



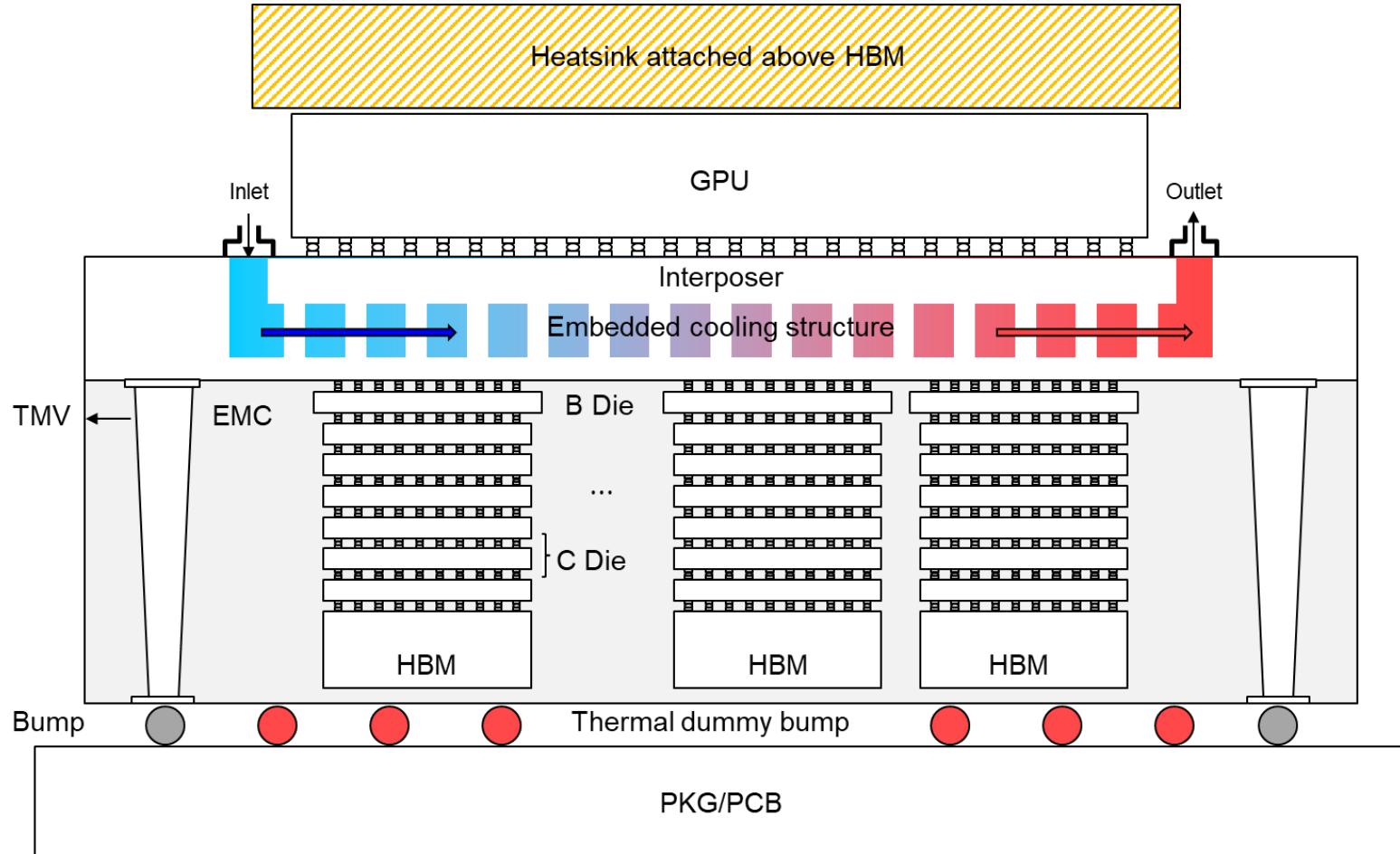
< 3D Memory Expansion Architecture with Embedded Cooling for HBM8 with Double-Sided Interposer using LPDDR >



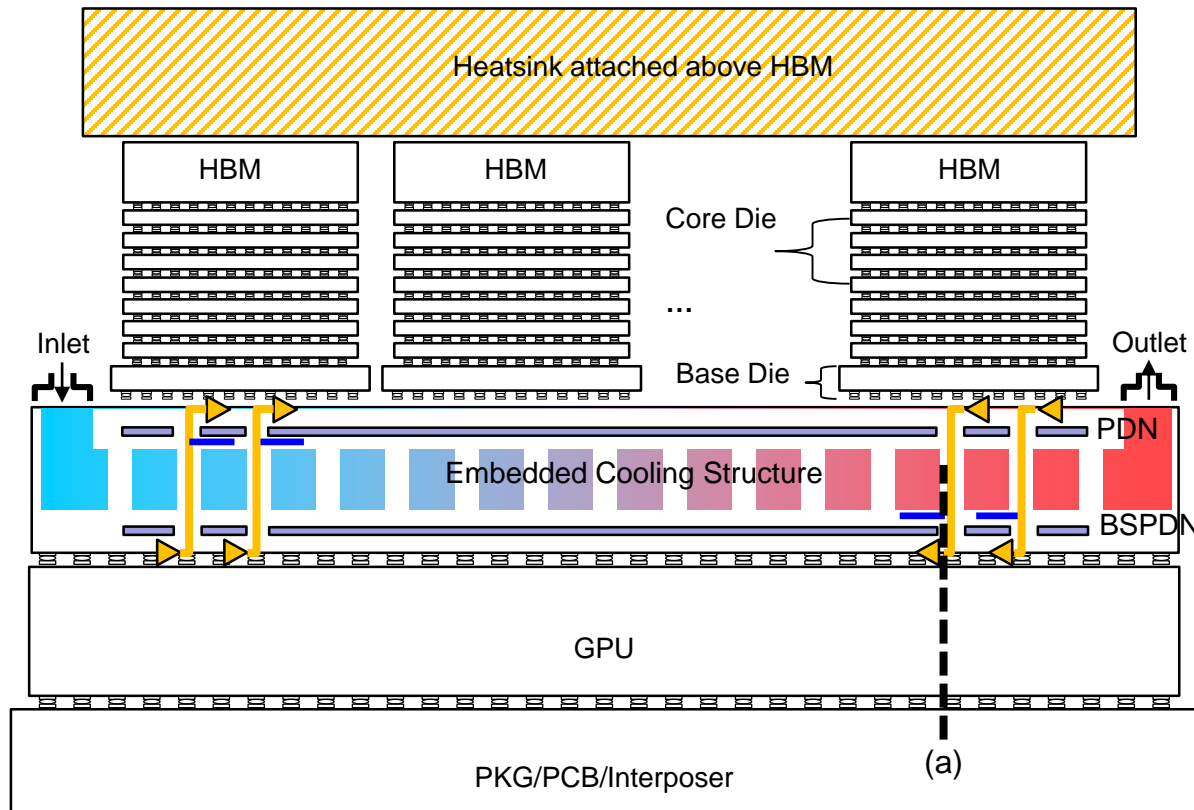
- GPU is implemented at top layer of memory stack for heat dissipation. (Thermal Issue ↓)
- Additional silicon interconnect pillar die is embedded between HBMs to support power to GPU.

# 3D Package-on-Interposer (PoI) Architecture Using a Double-Sided Interposer with Embedded Cooling Structure for HBM8

HBM8  
Architecture

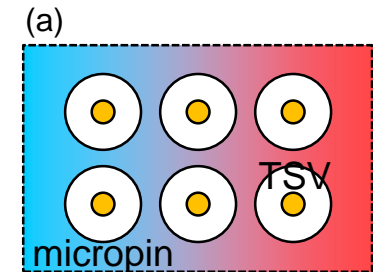


<Proposed 3D Package-on-Interposer (PoI) Architecture Using a Double-Sided Interposer with Embedded Cooling Structure for HBM8 >

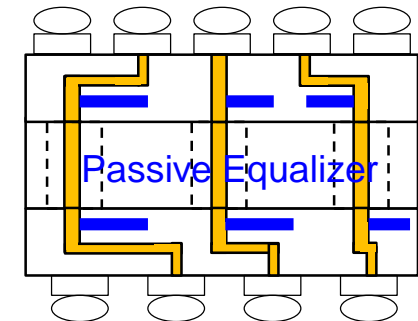


<Proposed Double-Side Interposer Architecture with Embedded Cooling and Top Heatsink for 3D GPU-HBM>

- Both the top heatsink and the double-side interposer are designed with inlet and outlet channels to enable efficient liquid cooling for the entire 3D GPU-HBM stack.

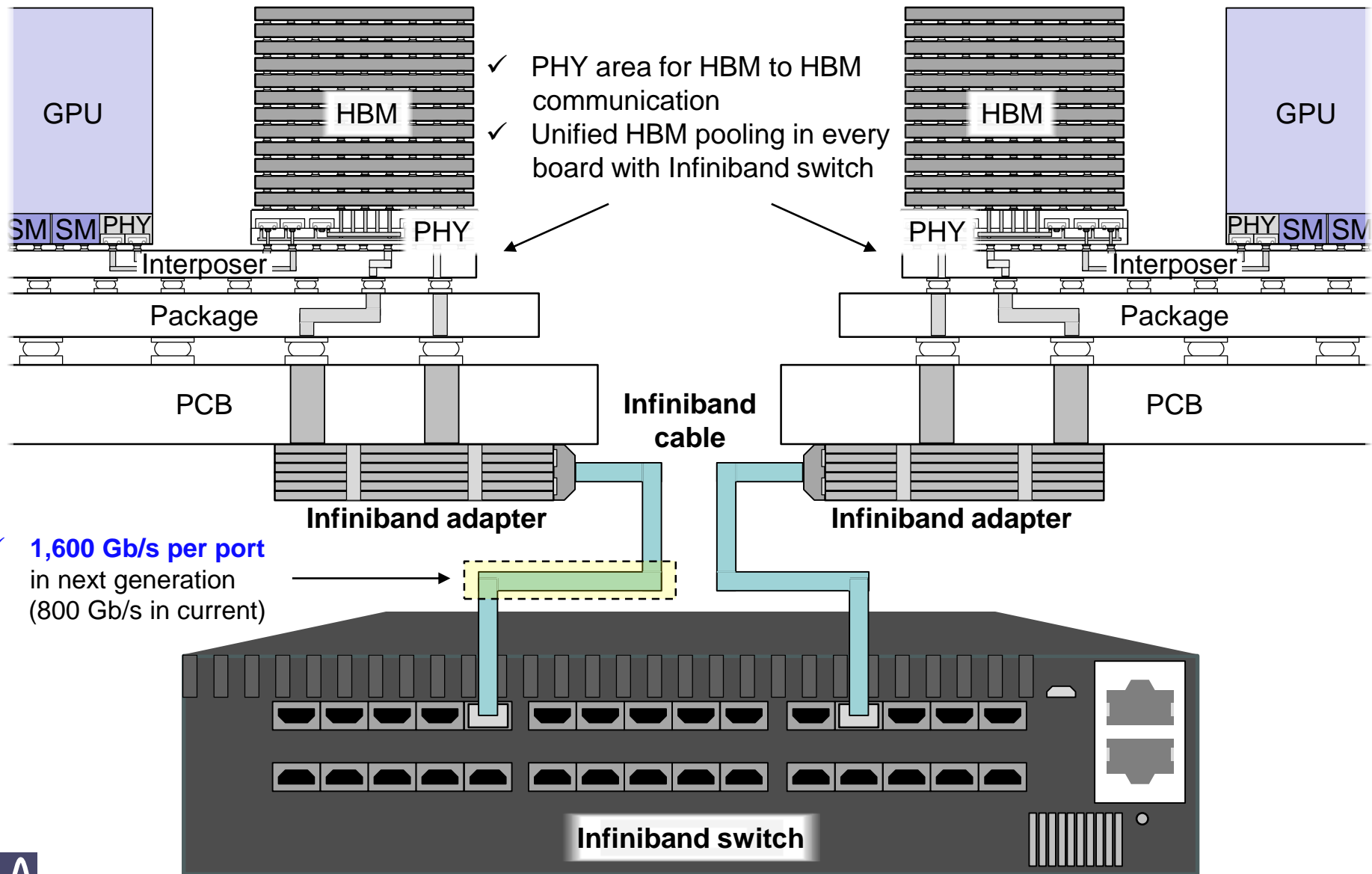


<Embedded Cooling Structure with micropin-fin and tsv>



<Passive Equalizer in double-side interposer>

# HBM-to-HBM Communication with Optical Cable (Infiniband) in HBM9





# Thank You!

## HBM

# HBM Roadmap Ver 1.7 Workshop

HBM 세대	순번	Time	Contents	Presenter
Intro	1	09:00 ~ 10:30	Overview of HBM Roadmap Ver. 1.7 by KAIST Teralab	김정호 교수
		10:30 ~ 10:40	Break (10분)	
HBM4/5	2	10:40 ~ 10:55	HBM5-LPDDR Architecture with Customized Base Die	윤지원
	3	10:55 ~ 11:20	Design of 3D Near Memory Computing Architecture in HBM5 for High Performance and Power Efficient Computing	윤지원
	4	11:20 ~ 11:30	Hybrid Vision Transformer Based Chip Design Agent for Fast Estimation of Multi-layer and Multi-power PDN Impedance in Customized Base Die in HBM5	안현준
	5	11:30 ~ 11:40	Transformer-based Reinforcement Learning for TSV Placement and Design Optimization considering IR Drop in HBM5	서은지
	6	11:40 ~ 11:50	Mamba-Reinforcement Learning-based HBM5 Design Agent for Fast PDN Optimization considering Power Integrity	김병목
	7	11:50 ~ 12:00	Devformer with Collaborative Distillation for Optimal Decoupling Capacitor Placement in HBM5 Custom Base Die	김혜연
	8	12:00 ~ 12:10	Reinforcement Learning-based Decap Placement Optimization considering Diverse I/O Channel Interfaces in Custom Base Die of HBM5 Memory Pooling Architecture	박준호
	9	12:10 ~ 12:20	Power Supply Noise Induced Jitter (PSIJ) Modeling and Reinforcement-Learning based PI Optimization for HBM5 I/O Interface	신태인
		12:20 ~ 13:30	Lunch (70분)	

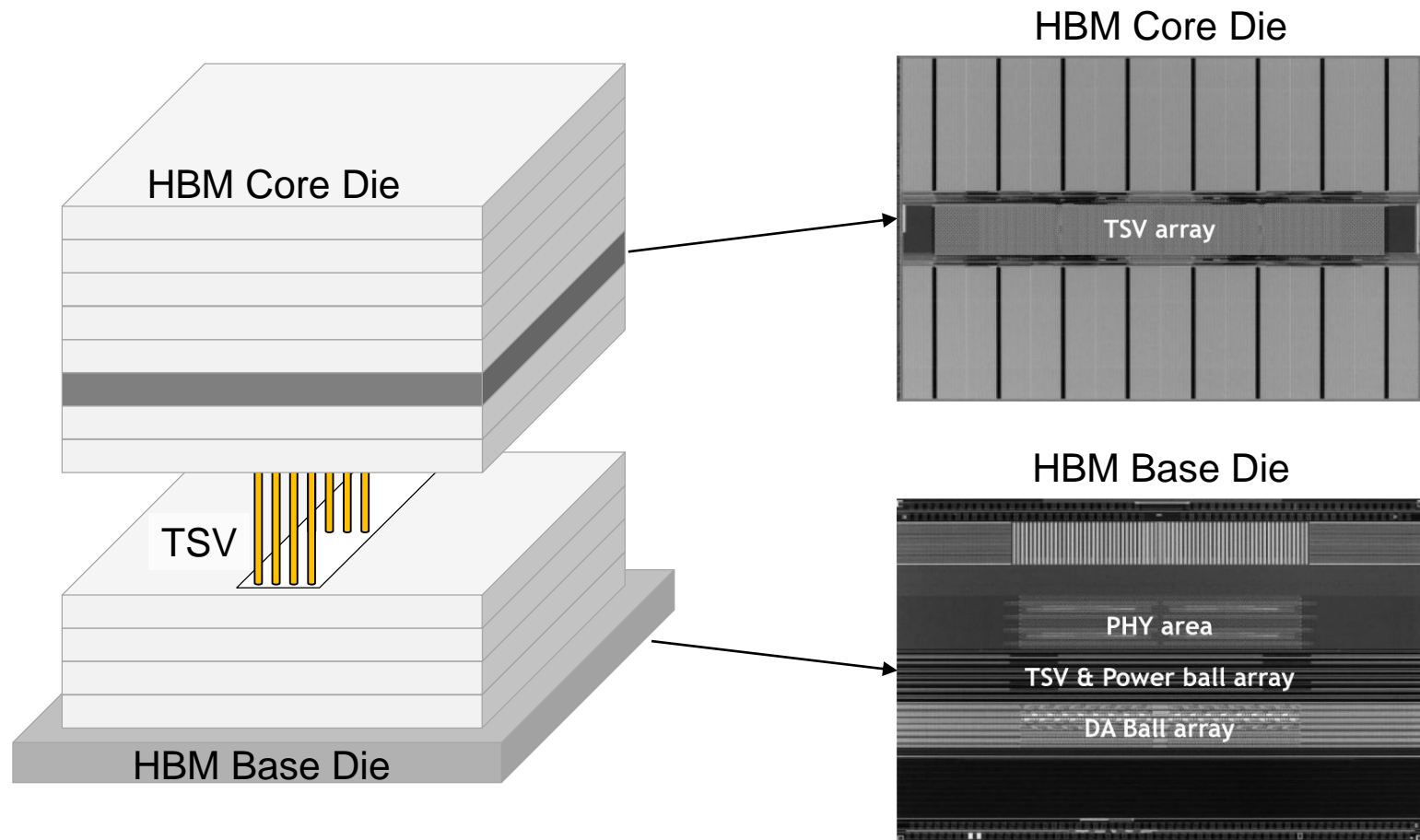
# HBM-LPDDR Architecture with Customized Base Die

Jiwon Yoon

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

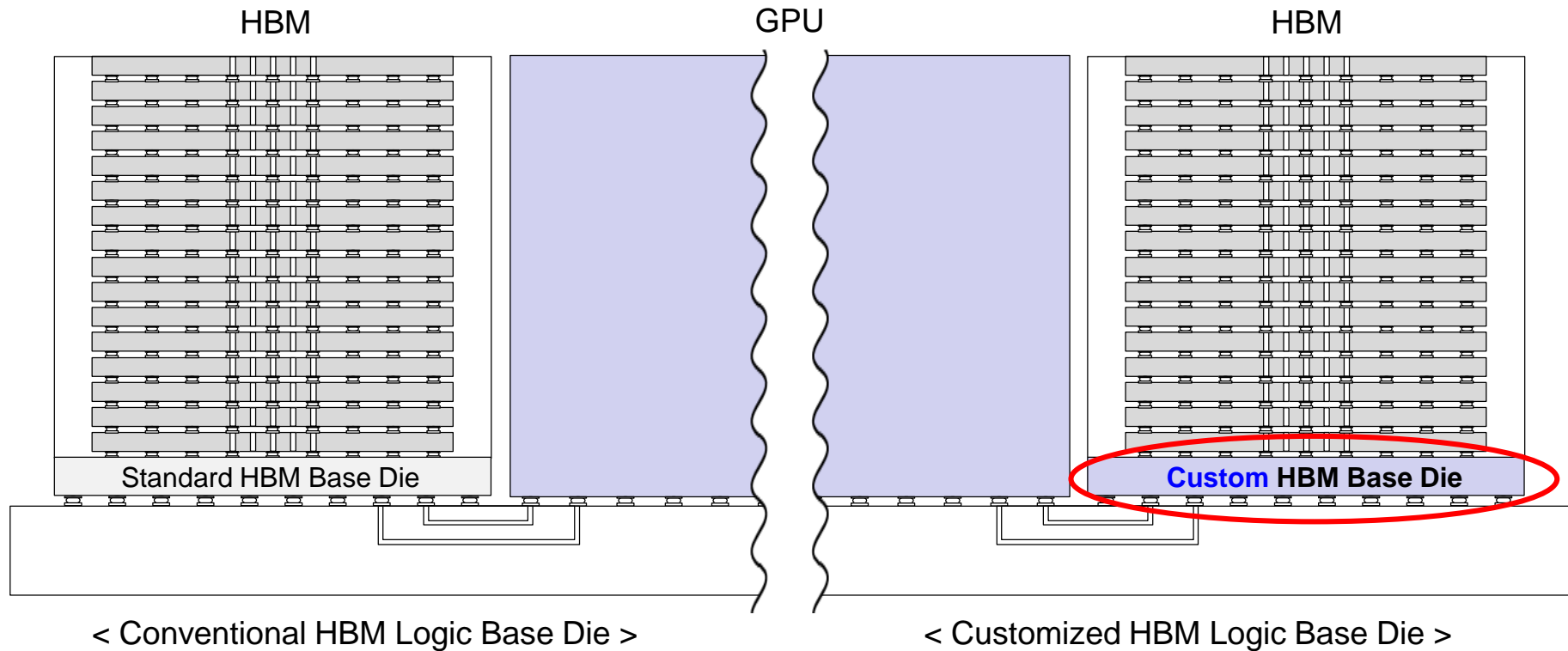
# Basic Structure of HBM (High Bandwidth Memory)



## < HBM DRAM Architecture with Base die and Stacked Core dies >

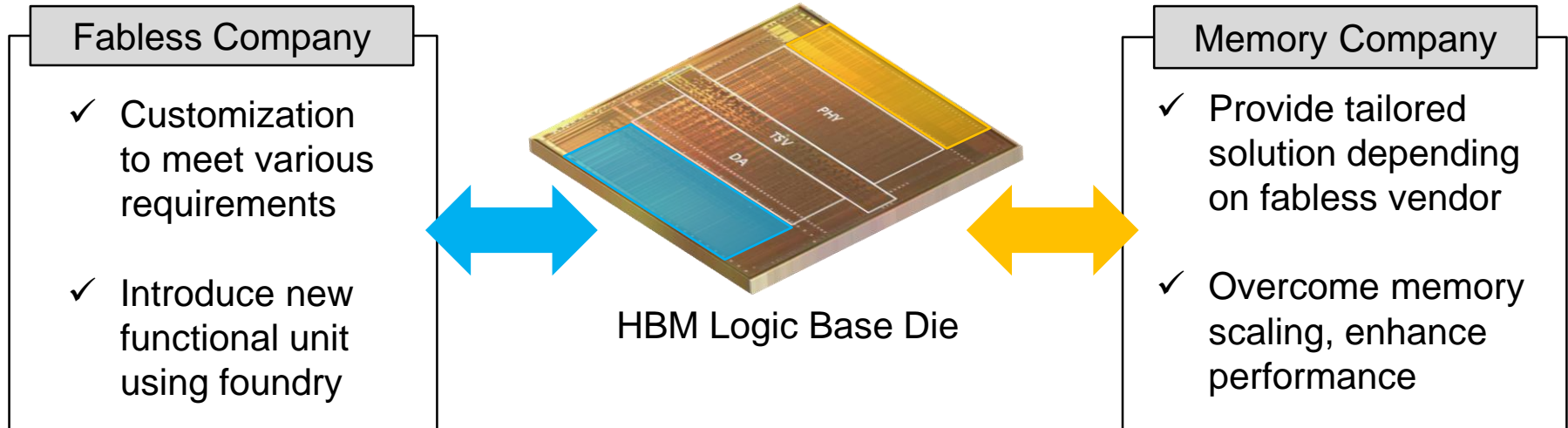
- HBM Core die : Memory cell array + peripheral logic, TSV array
- HBM Base (Buffer) die : PHY, signal + power TSV/ball array, direct access, decap etc.

# Customized HBM Logic Base Die for Next Generation HBM



	Logic (Foundry) Process	Memory Process
Conventional HBM-GPU Module	Processor (GPU)	HBM Core Die HBM Logic Base Die
<b>Future</b> HBM-GPU Module	Processor (GPU) <b>HBM Logic Base Die</b> ←	HBM Core Die

# Custom HBM Base die Manufactured in Logic Process

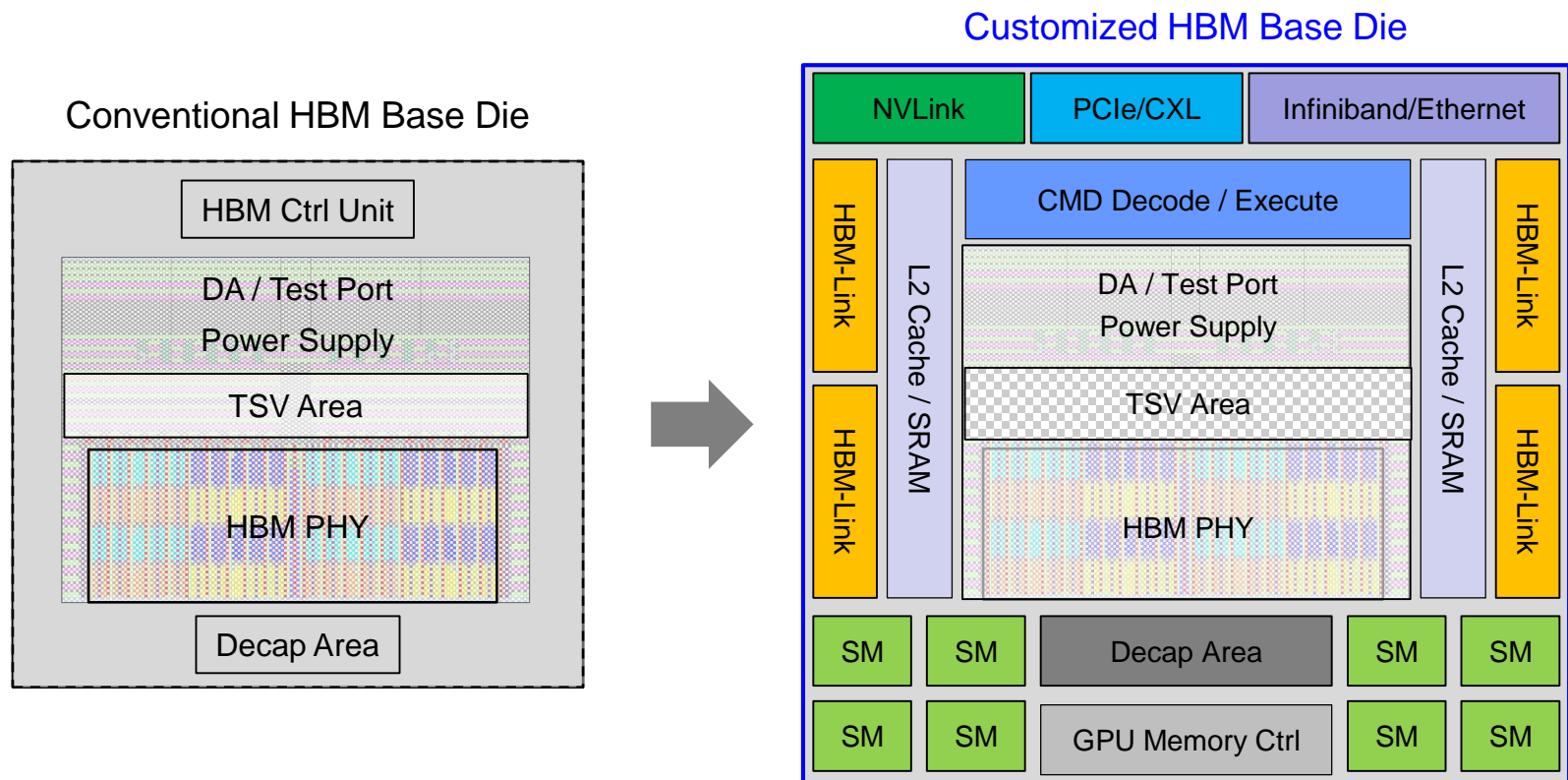


< HBM Base Die : Where Fabless and Memory Meet >

- HBM Base die is co-designed by both Fabless vendors and Memory companies, and manufactured using logic process.
  - ✓ Active collaboration opportunities between Fabless and Memory.
  - ✓ Enables application of various IP used in logic process to base die.



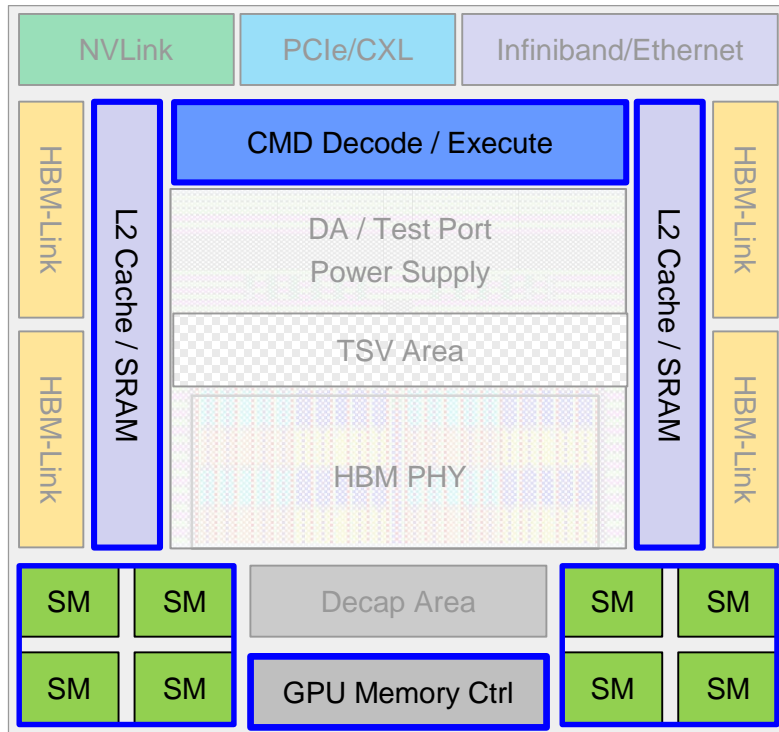
# HBM Centric Architecture with Custom HBM Base Die



	HBM4 (2026)	HBM5 (2029)	HBM6 (2032)	HBM7 (2035)	HBM8 (2038)
Architecture	Custom HBM Base Die HBM-LPDDR	3D NMC-HBM & stacked cache / decap	Multi-tower HBM Active / Hybrid Interposer	Hybrid HBM Architecture HBM-HBF HBM-3D LPDDR	Full-3D / HBM Centric Computing Architecture
HBM Base Die Functions	NMC processor + LPDDR Ctrl	+ Cache + CXL + on-die/stacked decap + HBM shielding	+ Network switch + Bridge die + Asymmetric TSV	+ HBF/LPDDR Ctrl + Storage network	+ HBM Centric Interposer + Double side Cooling + Edge-expand Stack

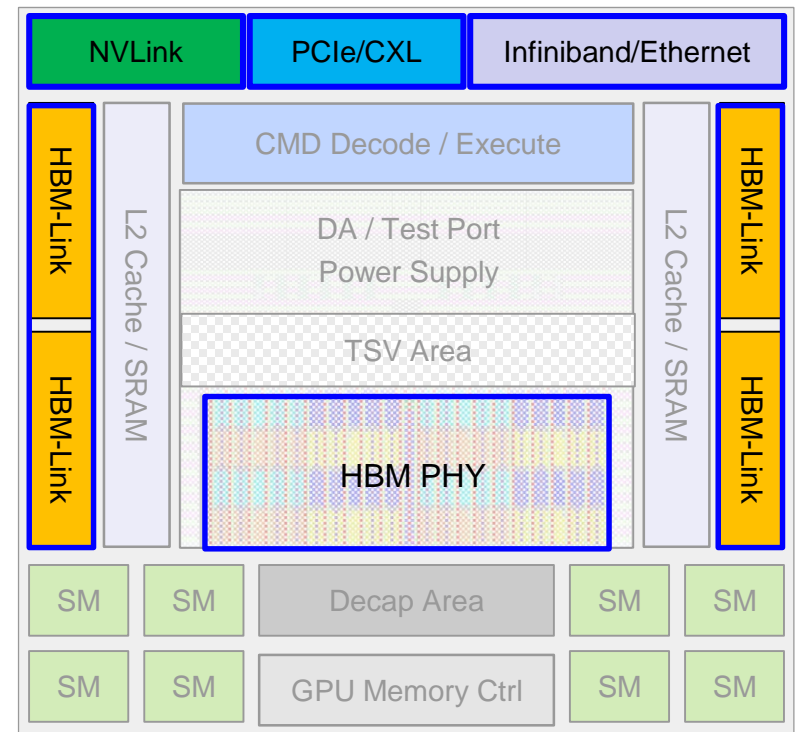
< Additional Functions Included in Custom HBM Base Die for Next-Generation HBM >

# Offloading CPU/GPU Function & Interconnect to Custom HBM Base Die



< CPU/GPU Functions in HBM Base Die >

- ① Command decode unit (instruction)
- ② Command execute unit (ALU)
- ③ L2 Cache / SRAM for frequent data access
- ④ GPU SM cores for near-memory-computing & bandwidth extension
- ⑤ GPU-HBM memory controller

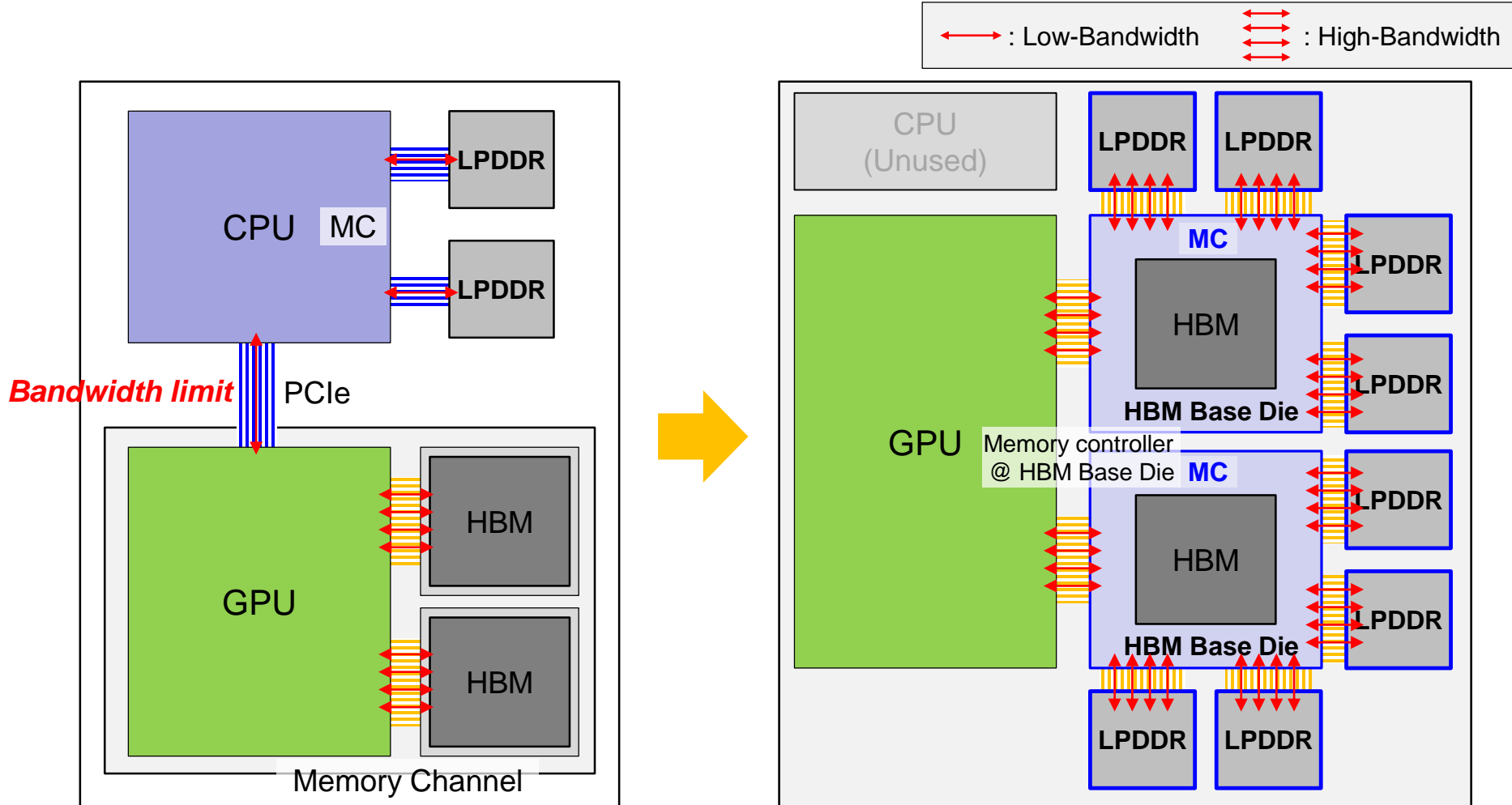


< Interconnect Network in HBM Base Die >

- ① HBM-GPU Interconnect
- ② NVLink for GPU-GPU Channel Network
- ③ PCIe+CXL for CPU / Memory Expansion
- ④ Infiniband / Ethernet for Node&Rack Level Supercomputer Network
- ⑤ HBM-Link for HBM-HBM Network



# Custom HBM Base Die Design (1/3) : LPDDR Memory Channel for High Capacity & Memory Expansion

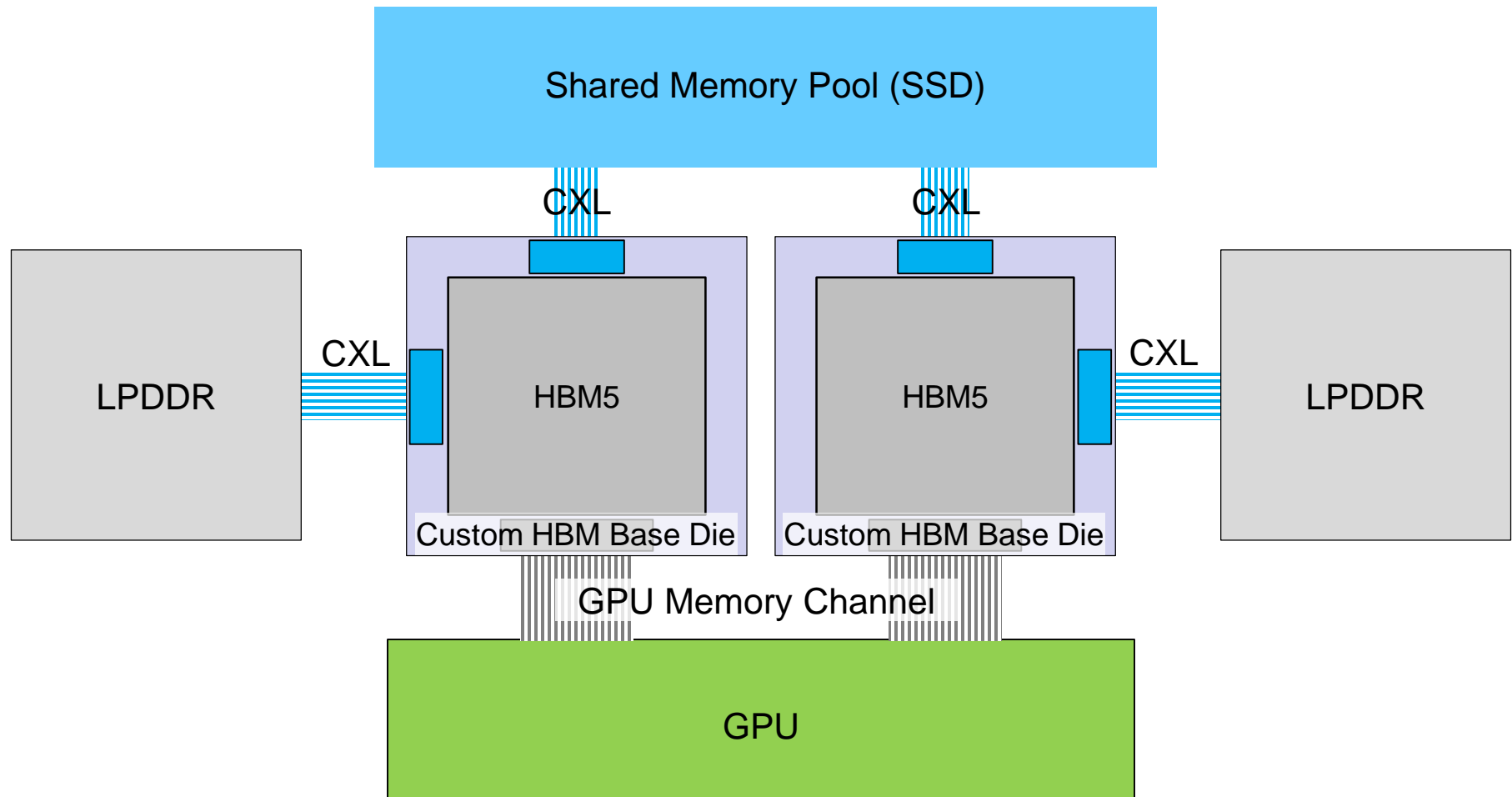


< Conventional CPU-GPU Architecture with HBM3 >

< HBM4 Architecture with LPDDR >

- The custom base die of HBM4 enables direct access to HBM and LPDDR, providing improved memory capacity without the CPU.

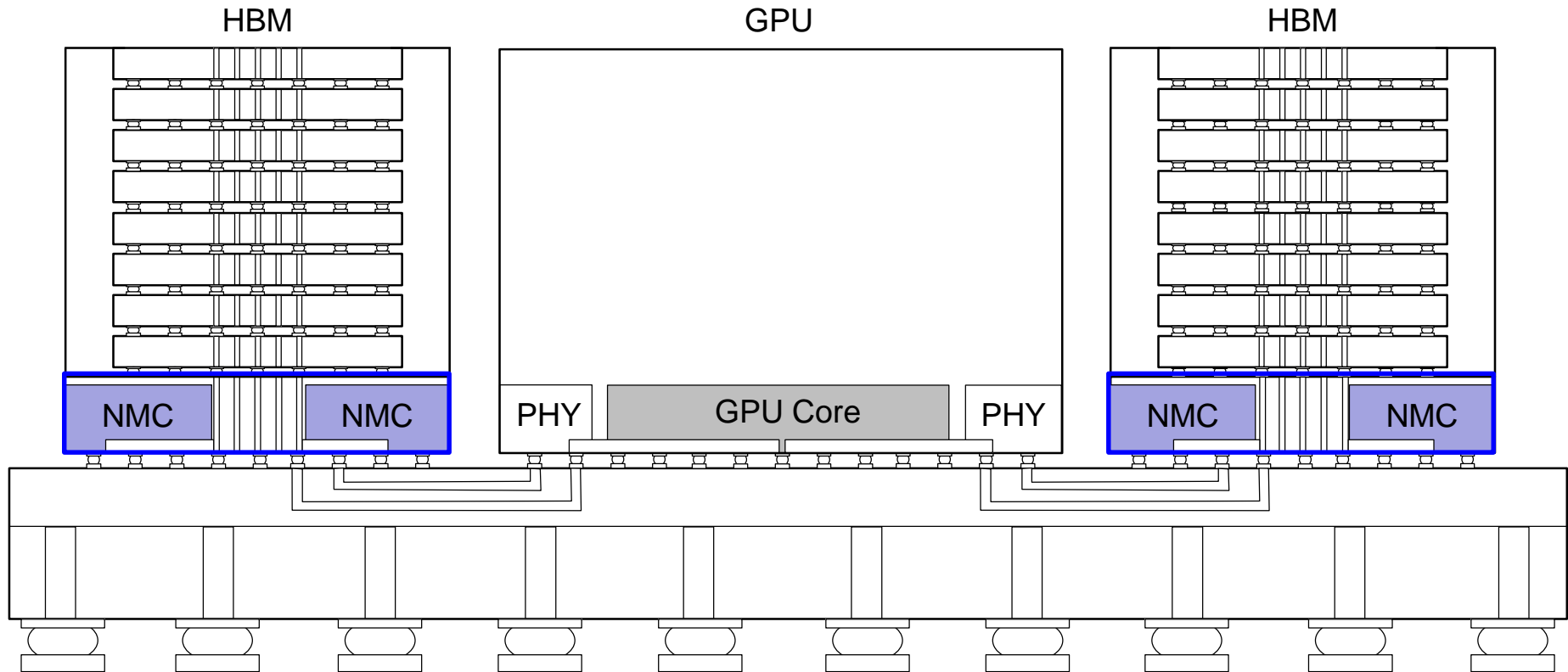
# Custom HBM Base Die Design (2/3) : CXL Memory Channel for Shared Data Storage & Memory Expansion



< HBM5 Architecture with CXL >

- The custom base die of HBM5 enables direct access to shared memory pool through CXL, providing a single unified memory with improved bandwidth and capacity.

# Custom HBM Base Die Design (3/3) : Near-Memory-Computing (NMC) HBM Architecture



## < NMC-HBM Architecture embedded at Custom HBM Base Die >

- HBM4 adopts a custom-HBM base die architecture, introducing various compute resources/functions within the base die.
- Through the NMC unit in HBM base die, the proposed NMC-HBM architecture achieves high bandwidth extension through the TSV and energy efficient channels.

- Sixth-generation HBM, HBM4 is expected to be revealed near 2026, with advanced technology innovations compared to previous HBM generations.
- The memory architecture of HBM4 is expected to change from the standard HBM base die to the customized base die.
- Possible examples of custom HBM base die functions :
  - ✓ Memory controller for HBM-LPDDR memory expansion
  - ✓ CXL interconnect for shared data storage
  - ✓ Near Memory Computing for energy efficient computing
- Following this trend, memory-fabless-foundry companies are preparing for the change in the semiconductor eco-system.
  - ✓ Collaboration & partnership between memory-fabless vendors
  - ✓ Innovation through advanced 3D packaging technology

# Thank You!

## HBM

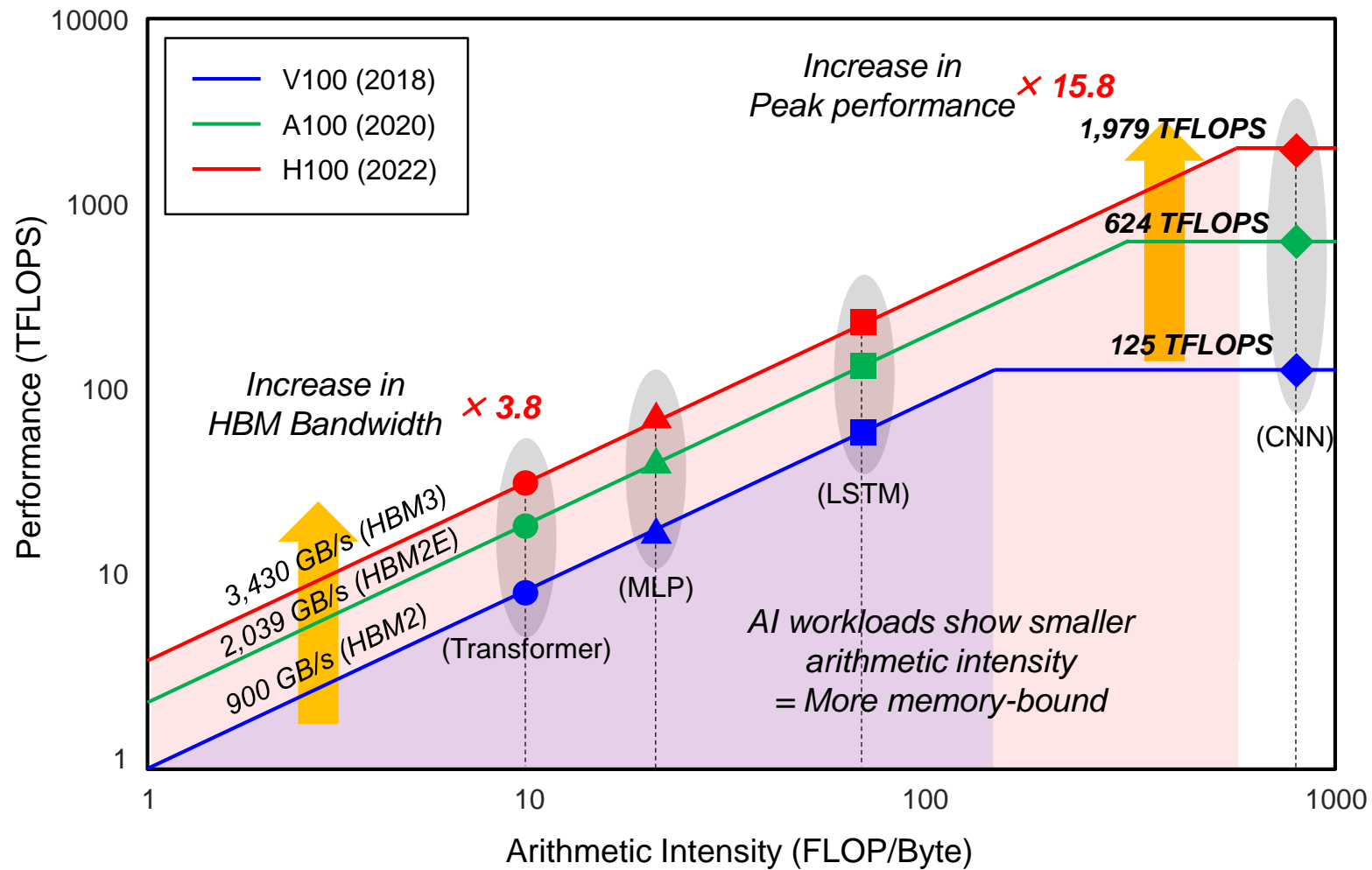
# Design of 3D Near Memory Computing Architecture in HBM5 for High Performance and Power Efficient Computing

Jiwon Yoon

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

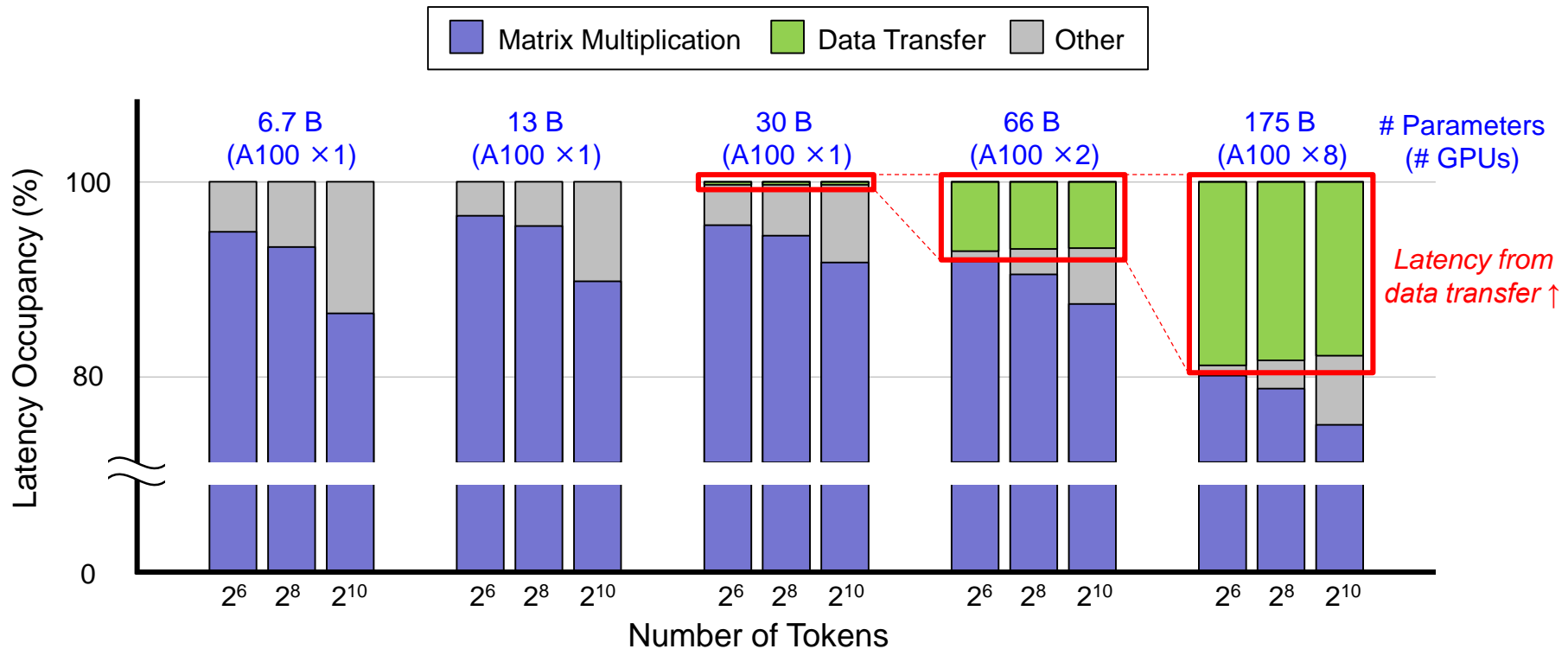
# Compute Characteristics of AI Workloads with Advance in GPU Performance & Increased Memory Bandwidth



< Roofline model by GPU-HBM generation >

# Characteristics of Transformer-based AI Models :

## High Latency from Frequent Data Access Between GPU and HBM



### < Inference Latency Occupancy of Transformer models >

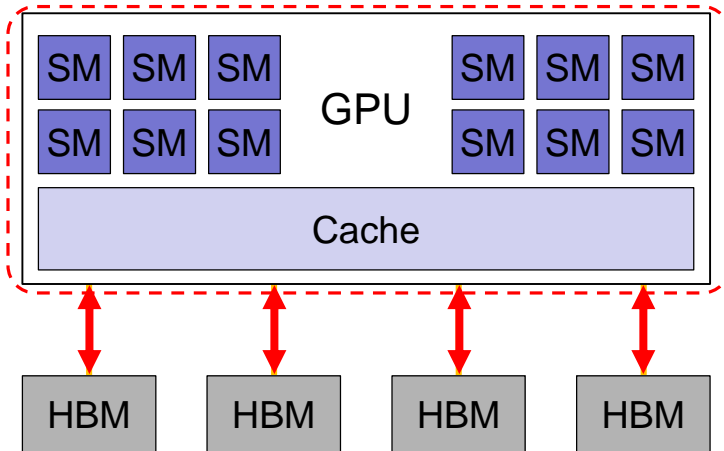
- The latency during transformer-based model inference is dominated by matrix multiplication.
- With the increase in model (parameter) size, latency from frequent data transfer is significant.
  - ✓ Overall performance of the current GPU-HBM system is expected to be limited by the excess data transfer between multiple GPUs and memory.

Ref) Park, Gunho, et al. "Lut-gemm: Quantized matrix multiplication based on luts for efficient inference in large-scale generative language models." arXiv:2206.09557 (2023).



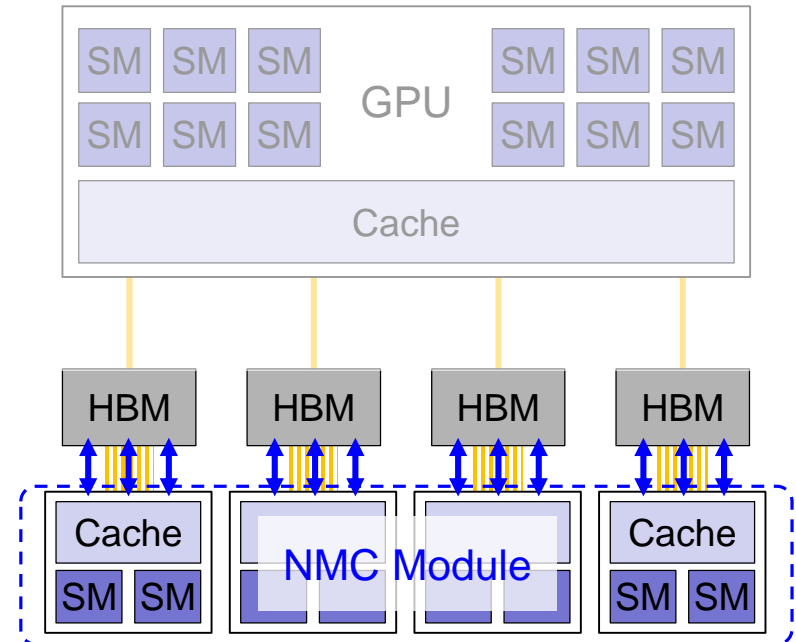
# Near Memory Computing (NMC) Architecture : Integration of Computing Resources Near Memory

Von Neumann architecture



- ✓ Long data movement path
- ✓ High interconnect latency / energy

Near-memory-computing (NMC) architecture

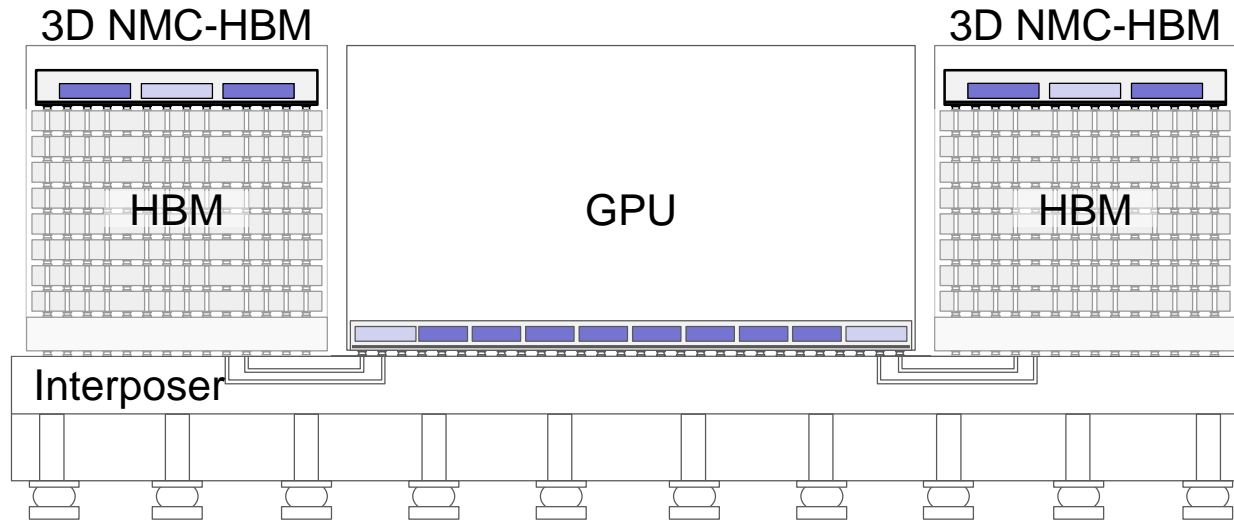


- ✓ Short data movement path
- ✓ Low interconnect latency / energy

< Comparison of Von Neumann and Near-memory-computing (NMC) architecture >

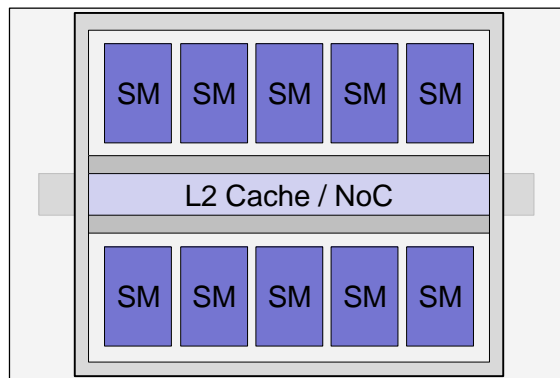
- Near-memory-computing architecture based on GPU-HBM is achieved by integrating GPU compute unit (SM & L2 cache) near HBM DRAM.

# Design of 3D Near-Memory-Computing Architecture in HBM5 : 3D NMC-HBM

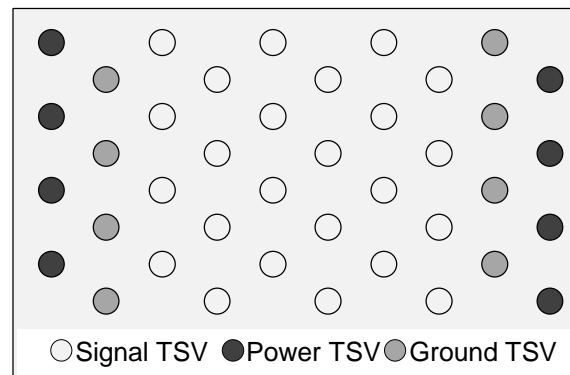


< Key Features of the designed 3D NMC-HBM Architecture >

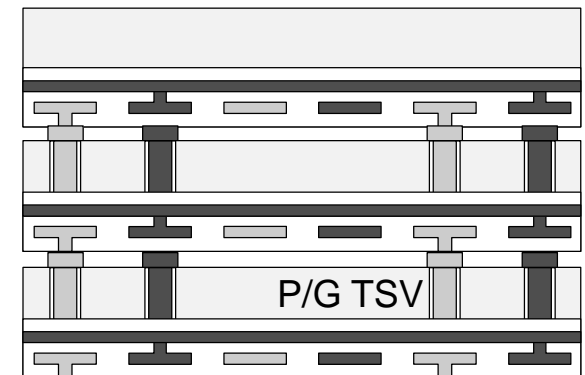
3D NMC Processor Die



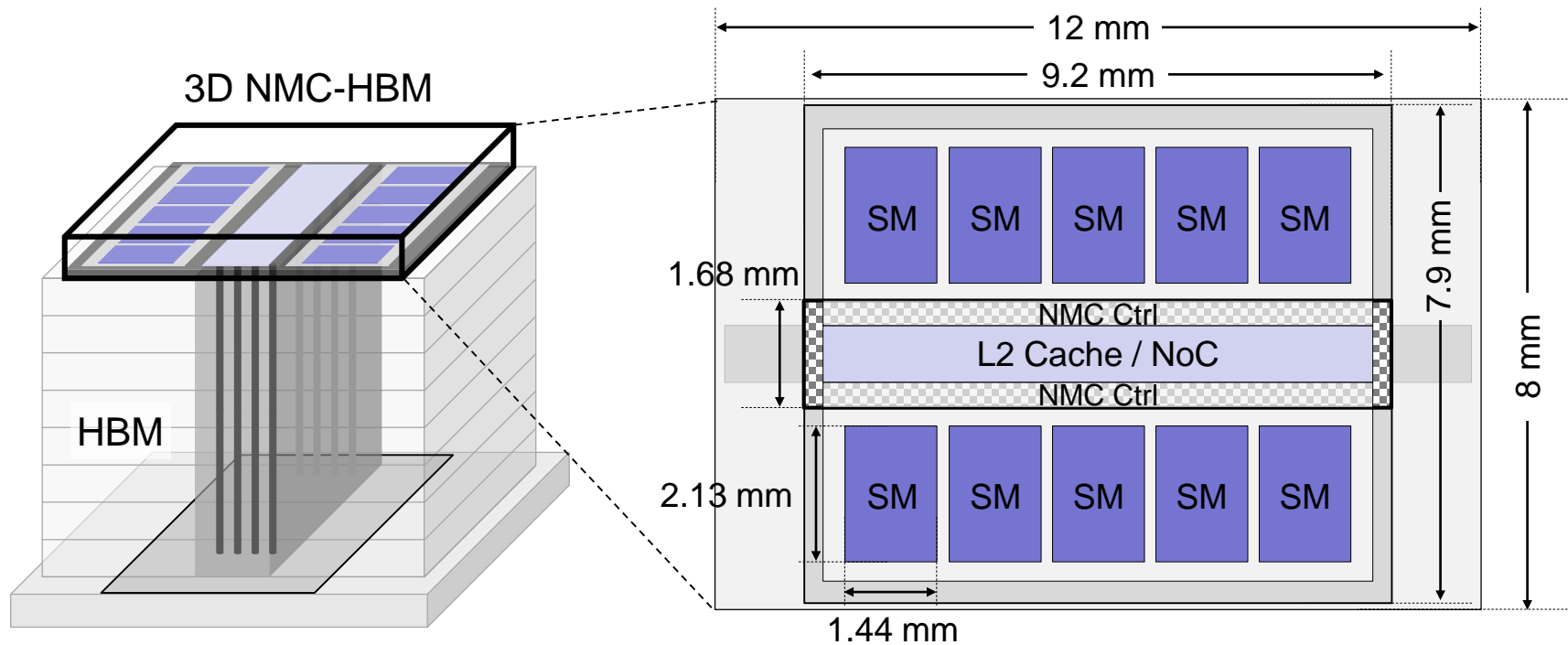
3D NMC-HBM TSV



3D NMC-HBM PDN



# Die Area Assumption of 3D NMC-HBM Architecture

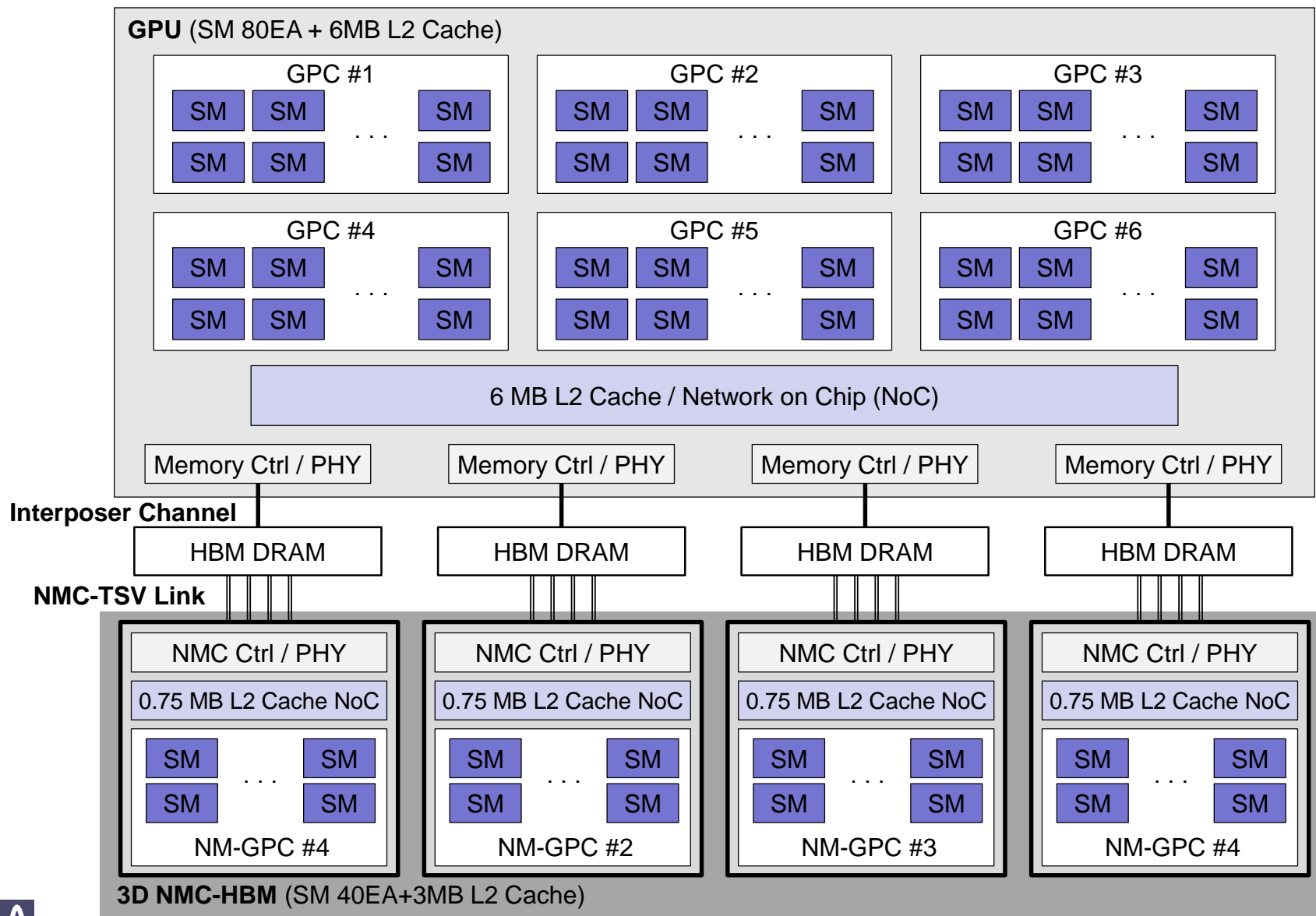


< 3D NMC-HBM PDN Structure >

< Physical Layout of 3D NMC Processor Die >

- The design of the 3D NMC-HBM architecture includes a NMC processor die stacked above HBM DRAM stack.
- The NMC processor die area is assumed based on physical dimensions of HBM.
  - ✓ 10 processing cores (GPU SM) + 0.75 MB L2 Cache + NMC Controller

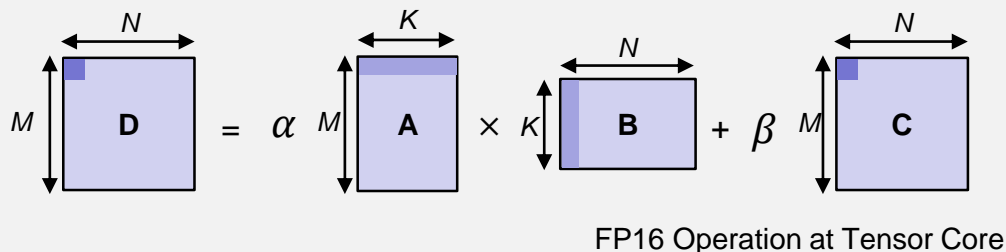
# System Configuration of 3D NMC-HBM Architecture



< System Configuration of NMC-HBM Architecture based on V100 GPU >

# Calculating Arithmetic Intensity of GEMM Workloads

$$D = \alpha A \cdot B + \beta C$$



Deepbench Benchmark (Inference)

GEMM	Matrix Dimension			Arithmetic Intensity [FLOPS/Byte]
	M	N	K	
①	5124	1500	2048	740.68
②	7680	1	2560	0.99
③	6144	4	2048	3.99
④	512	3000	1536	340.43
⑤	35	1500	2560	33.75

< Basic Structure of GEMM Operation >

< Arithmetic Intensity of GEMM workloads >

- General Matrix Multiplication (GEMM) is the fundamental building block for many operations in transformer network.
- A basic GEMM operation defined as  $D = \alpha AB + \beta C$ , requires a total of  $M \cdot N \cdot K$  multiply and accumulate (FP16 MAC) operations, equal to a total of  $2 \cdot M \cdot N \cdot K$  FLOPS.
- The number of data access from local memory is  $2 \cdot (M \cdot K + N \cdot K + M \cdot N)$ .

$$\text{Arithmetic Intensity [FLOPS/byte]} = \frac{\text{number of FLOPS}}{\text{number of byte accesses}} = \frac{2 \cdot (M \cdot N \cdot K)}{2 \cdot (M \cdot K + N \cdot K + M \cdot N)} = \frac{M \cdot N \cdot K}{M \cdot K + N \cdot K + M \cdot N}$$

Ref) <https://docs.nvidia.com/deeplearning/performance/dl-performance-matrix-multiplication/index.html>

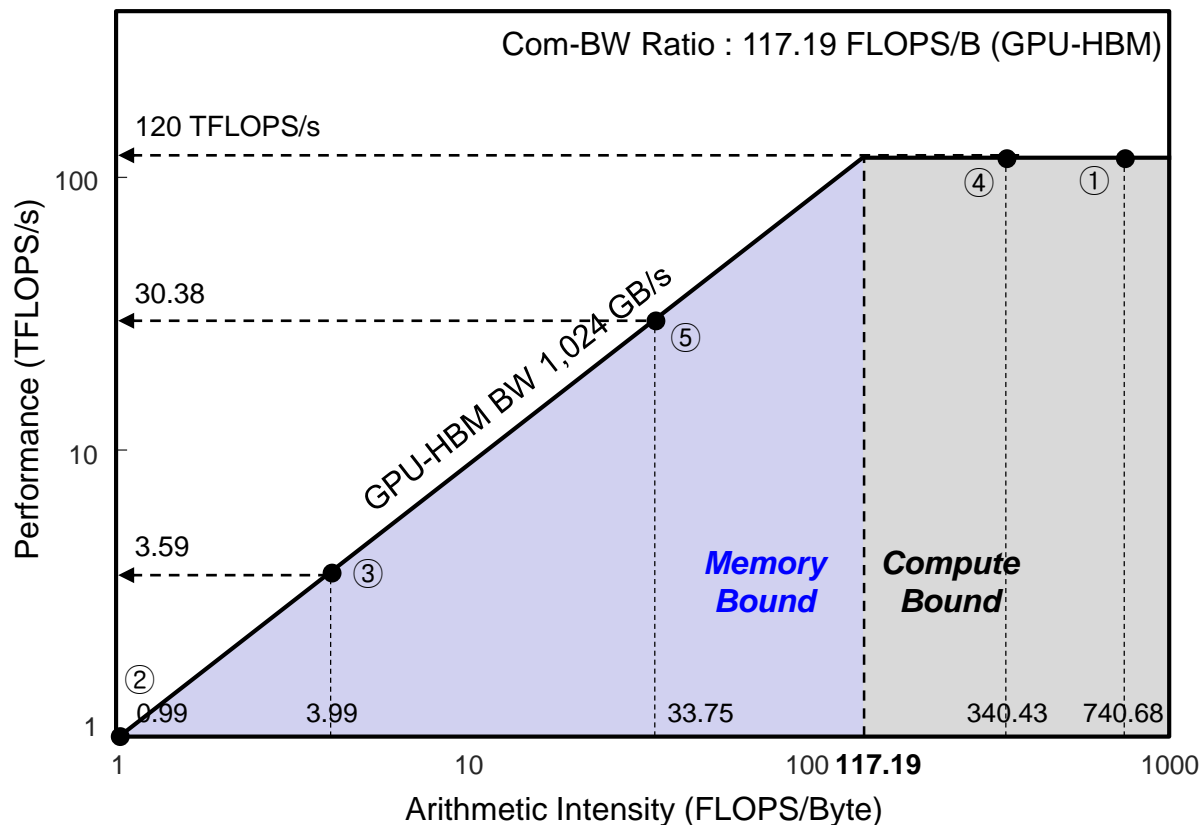
Ref) Baidu Research - Deepbench

# Performance Evaluation of GEMM Workloads with GPU-HBM Roofline Model

$$\text{Arithmetic Intensity} = \frac{M \cdot N \cdot K}{M \cdot K + N \cdot K + M \cdot N}$$

$$\text{Com-BW Ratio} = \frac{\text{Peak Performance}}{\text{Bandwidth}}$$

GEMM	Arithmetic Intensity [FLOPS/Byte]	Achieved Performance [TFLOPS/s]
①	740.68	120
②	0.99	0.89
③	3.99	3.59
④	340.43	120
⑤	33.75	30.38



## < Roofline Model of Simulated GEMM Workloads >

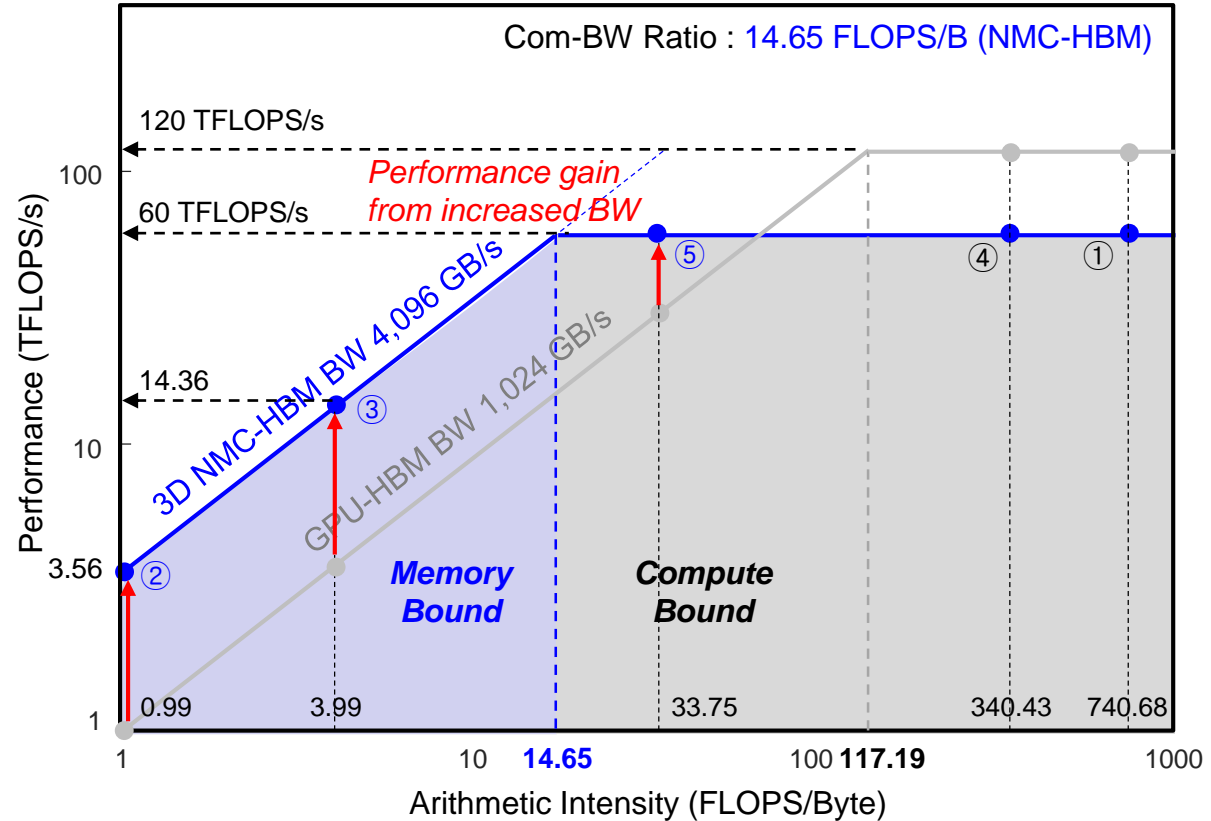
- Assuming the GPU-HBM architecture meets the system's peak computation capabilities, the roofline model of GPU-HBM and GEMM workloads was plotted based on V100.
  - ✓ GEMM ①, ④ : Arithmetic Intensity > 117.19 FLOPS/B → **Compute-bound**
  - ✓ GEMM ②, ③, ⑤ : Arithmetic Intensity < 117.19 FLOPS/B → **Memory-bound**

# Performance Evaluation of GEMM Workloads with NMC-HBM Roofline Model

$$\text{Arithmetic Intensity} = \frac{M \cdot N \cdot K}{M \cdot K + N \cdot K + M \cdot N}$$

$$\text{Com-BW Ratio} = \frac{\text{Peak Performance}}{\text{Bandwidth}}$$

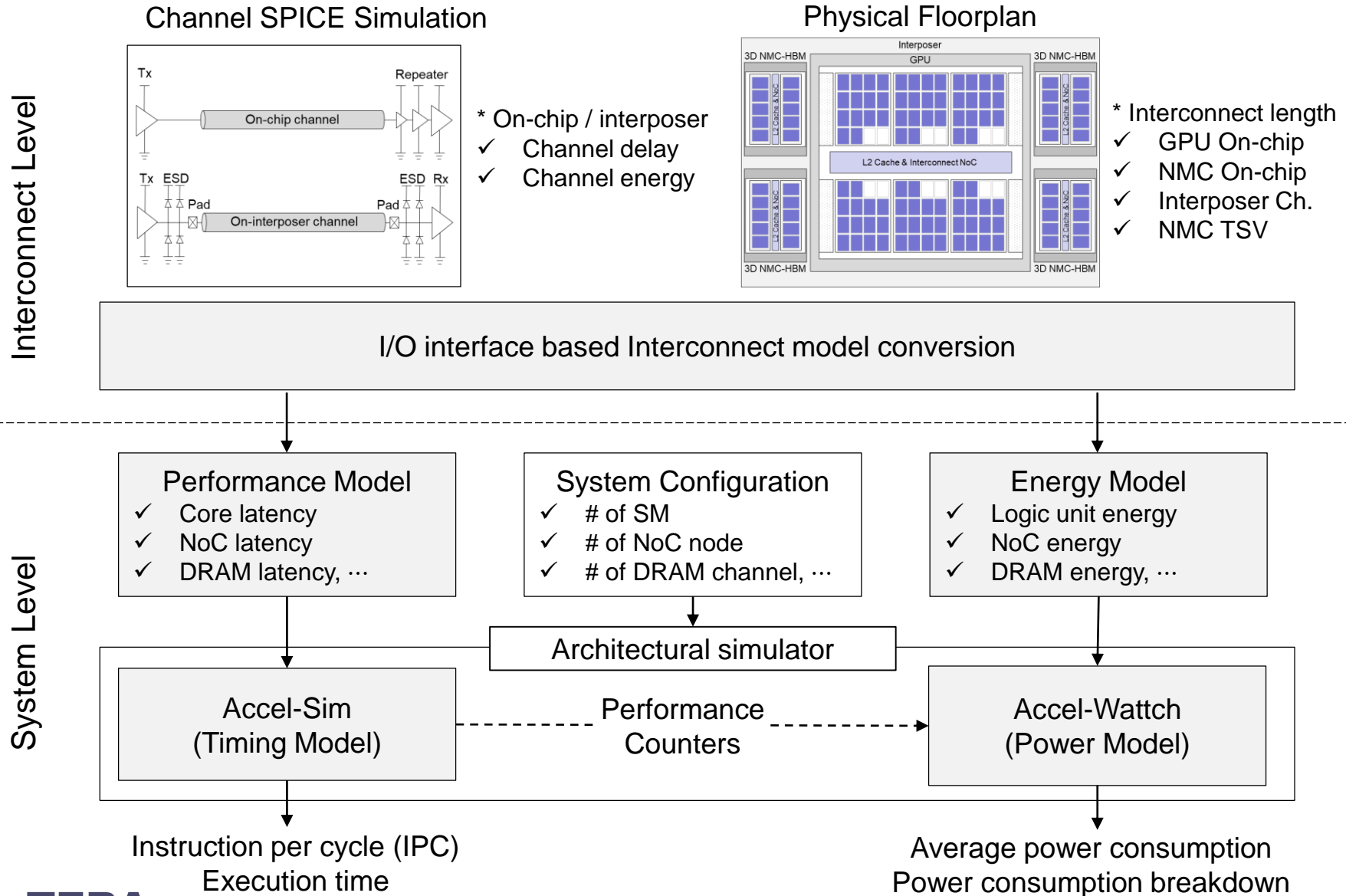
GEMM	Arithmetic Intensity [FLOPS/Byte]	Achieved Performance [TFLOPS/s]
①	740.68	120
②	0.99	0.89 → 3.56
③	3.99	3.59 → 14.364
④	340.43	120
⑤	33.75	30.38 → 60



## < Roofline Model of Simulated GEMM Workloads >

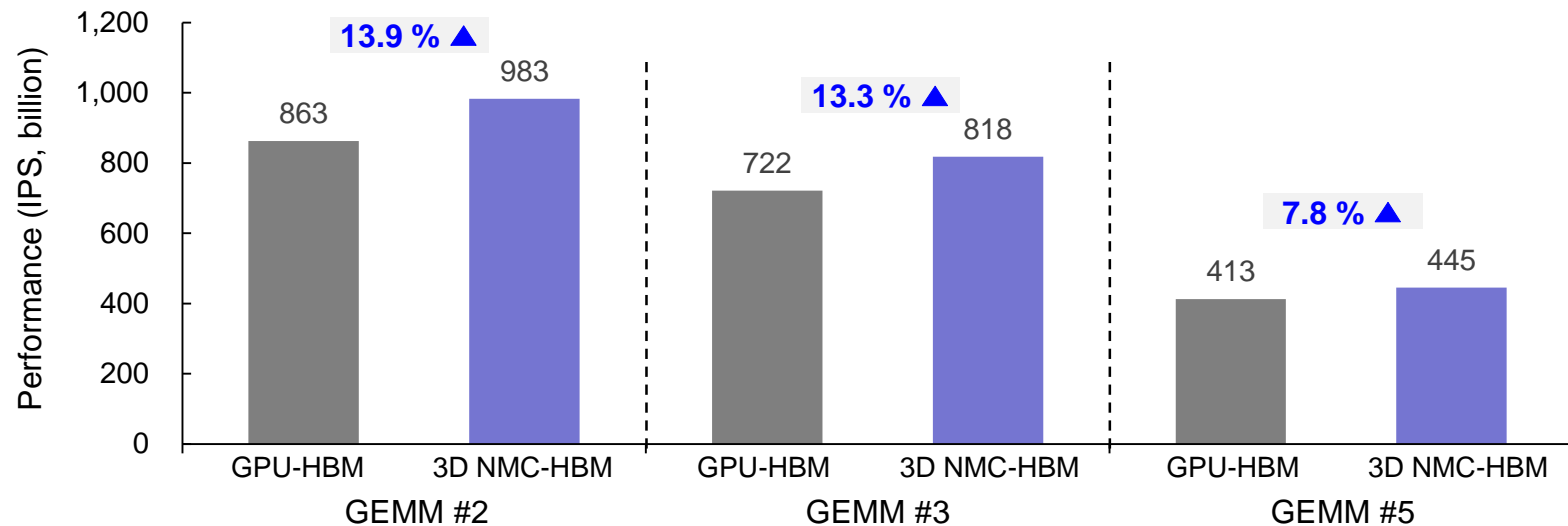
- The proposed NMC-HBM architecture providing higher bandwidth (4,096 GB/s) show lowered Compute-BW-Ratio (117.19 → 14.65)
- Memory-bound GEMM workloads (②, ③, ⑤) benefit from higher memory bandwidth within NMC-TSV, showing higher performance (FLOPS) when offloaded to 3D NMC-HBM.

# Performance and Power Evaluation Method of the Proposed NMC-HBM Architecture

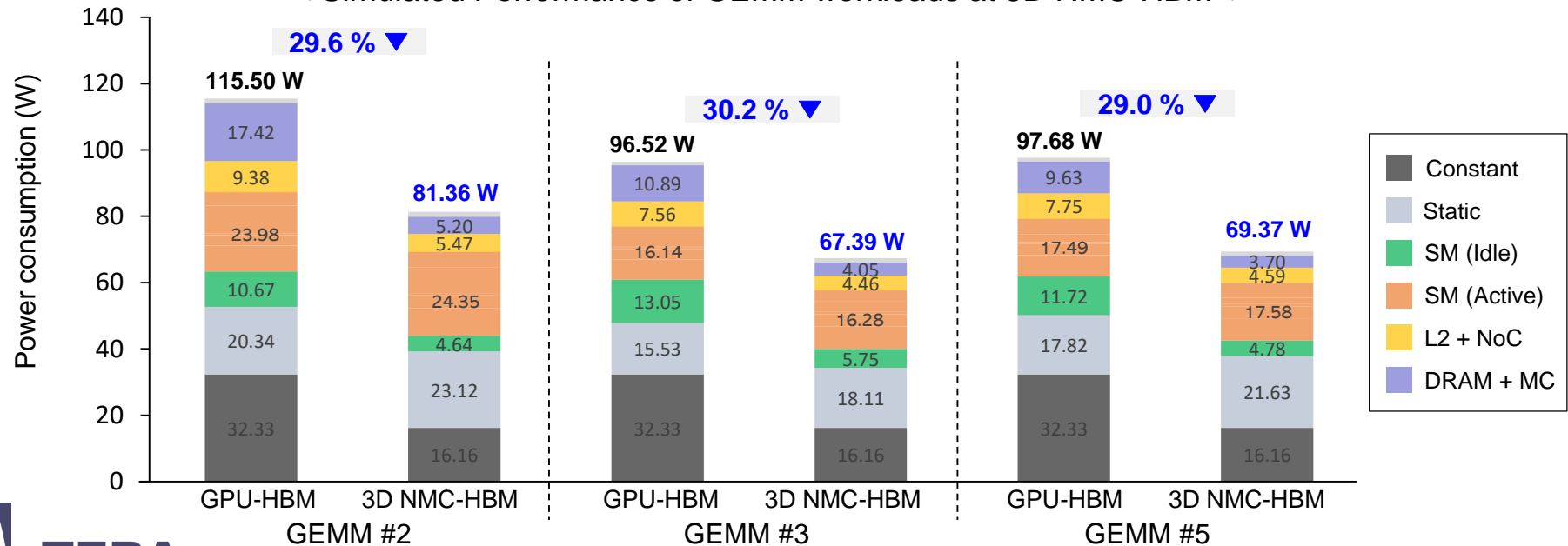




# Performance and Power Evaluation of 3D NMC-HBM Architecture

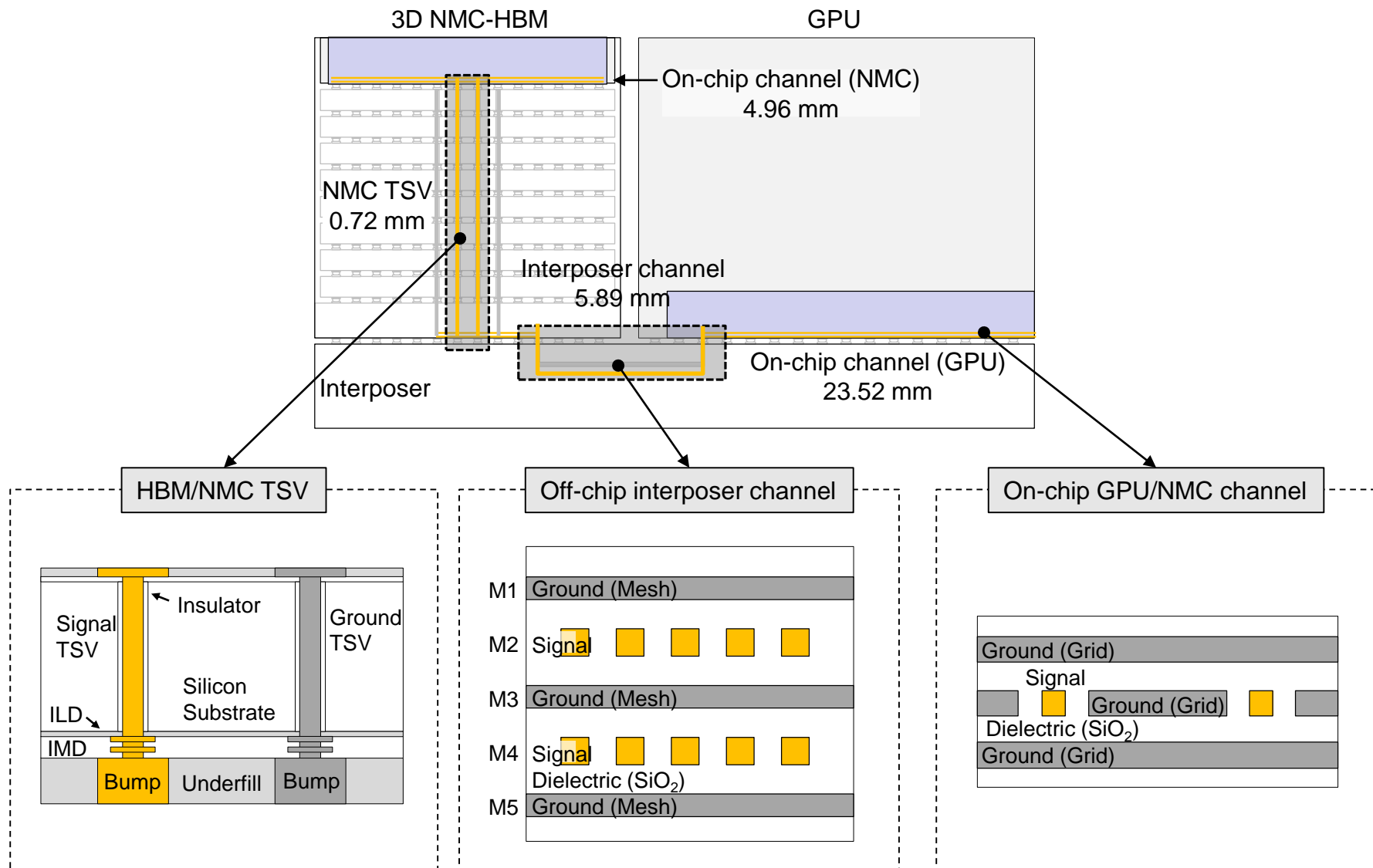


< Simulated Performance of GEMM workloads at 3D NMC-HBM >



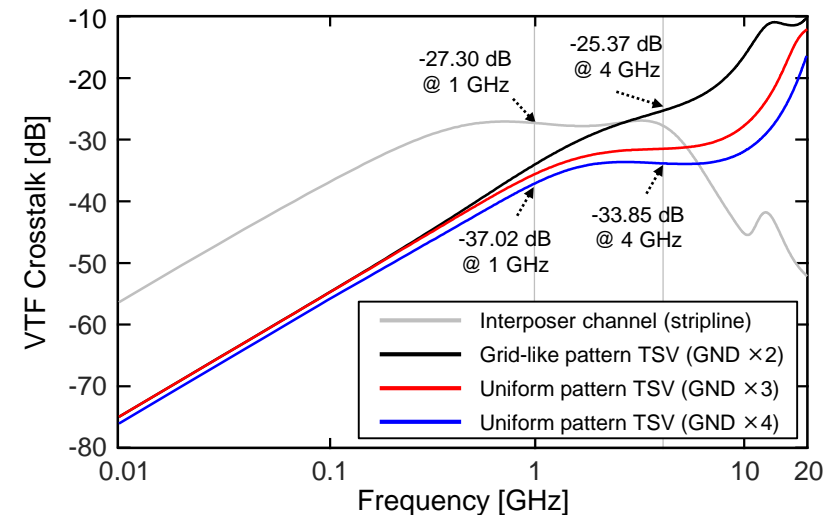
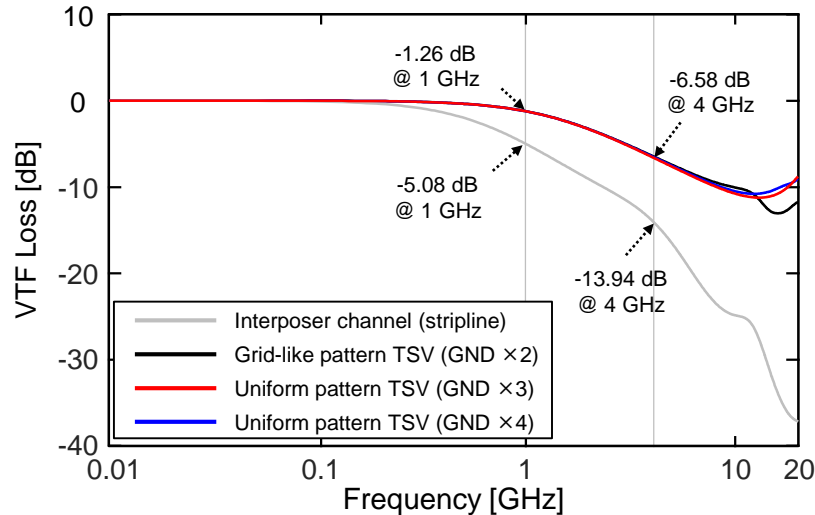
< Simulated Power Consumption of GEMM workloads at 3D NMC-HBM >

# Design of Interconnect Structure within 3D NMC-HBM Architecture



< Cross-sectional view of 3D NMC-HBM interconnect structures >

# VTF Loss & Crosstalk, Eye Diagram Results of 3D NMC-HBM Interconnect

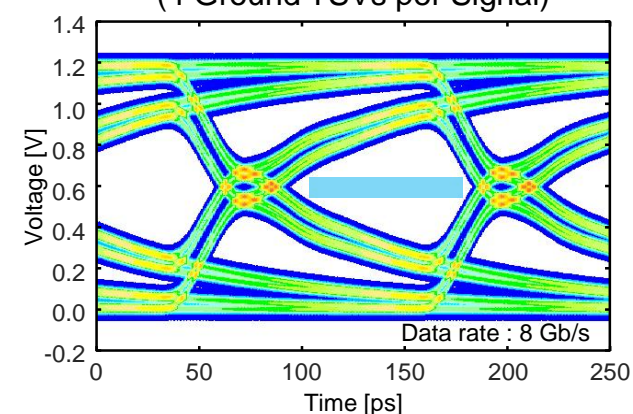
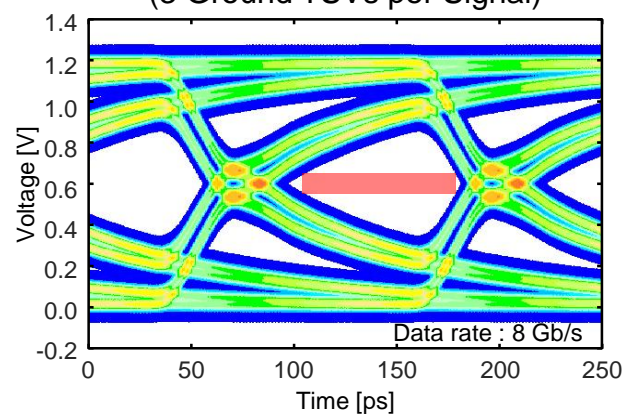
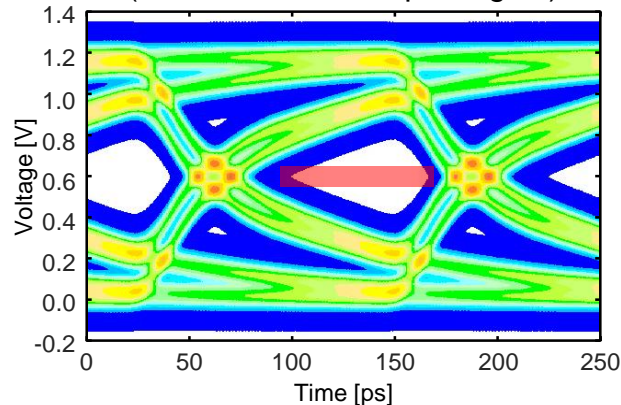


< Simulated Voltage Transfer Function loss and crosstalk of the 3D NMC-HBM TSV >

Conventional Grid TSV Array  
(0 ~ 2 Ground TSV per Signal)

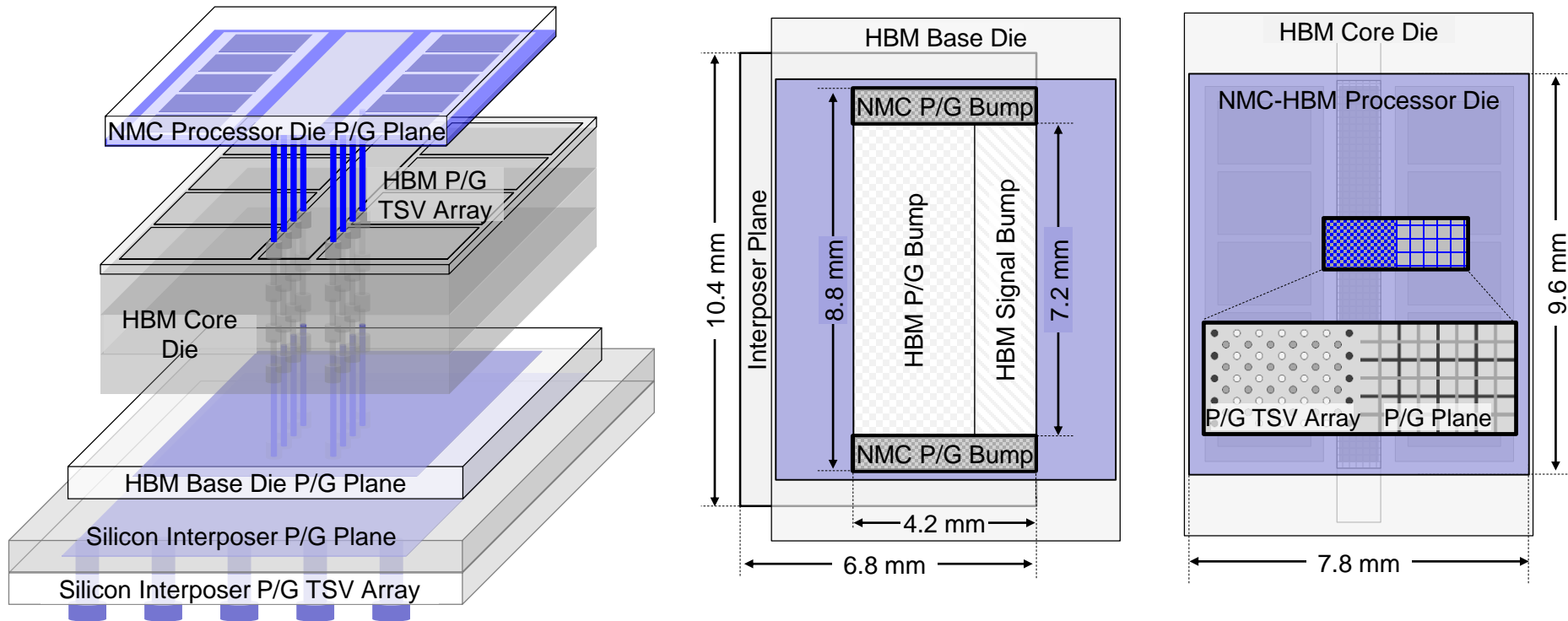
Uniform Distributed TSV Array  
(3 Ground TSVs per Signal)

Uniform Distributed TSV Array  
(4 Ground TSVs per Signal)



< Simulated eye-diagram of the 3D NMC-HBM TSV configurations >

# Design of 3D Power Distribution Network for 3D NMC-HBM Architecture



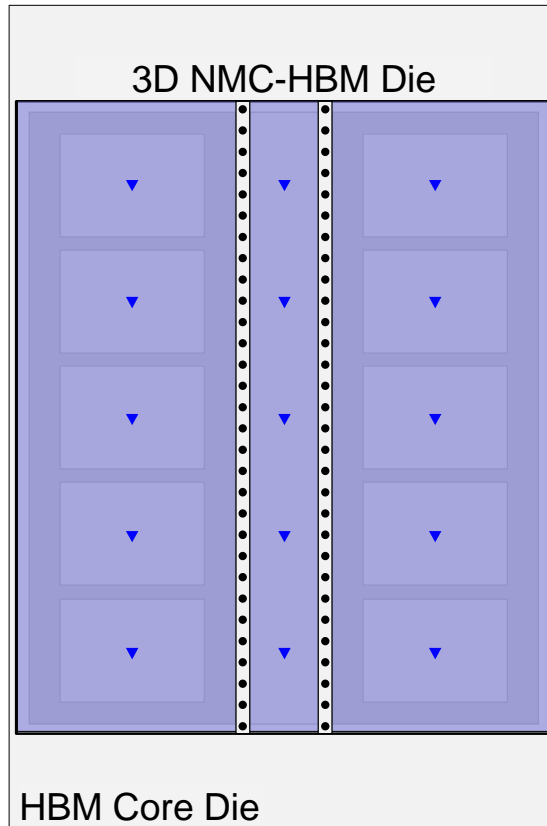
## < Hierarchic PDN Configuration of 3D NMC-HBM Architecture with Physical Layout >

- The power for 3D NMC-HBM architecture is supplied through the 3D NMC PDN consisting of a dedicated power TSV array and a NMC-base die above HBM DRAM.

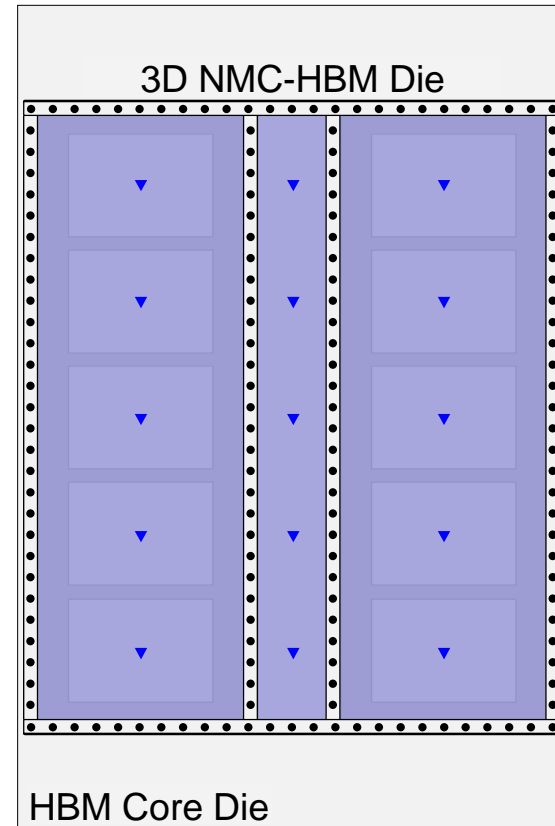
# Additional P/G TSV Array Placement at HBM Die Edge

■ P/G TSV placement area    ▼ Probing port

P/G TSV @ Die Center



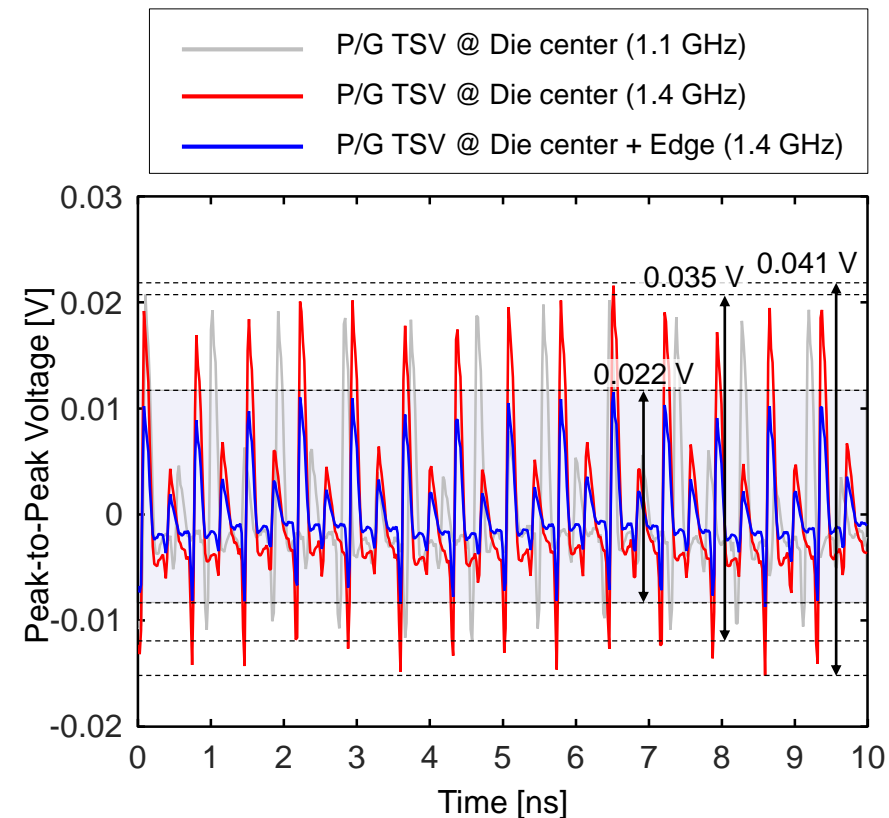
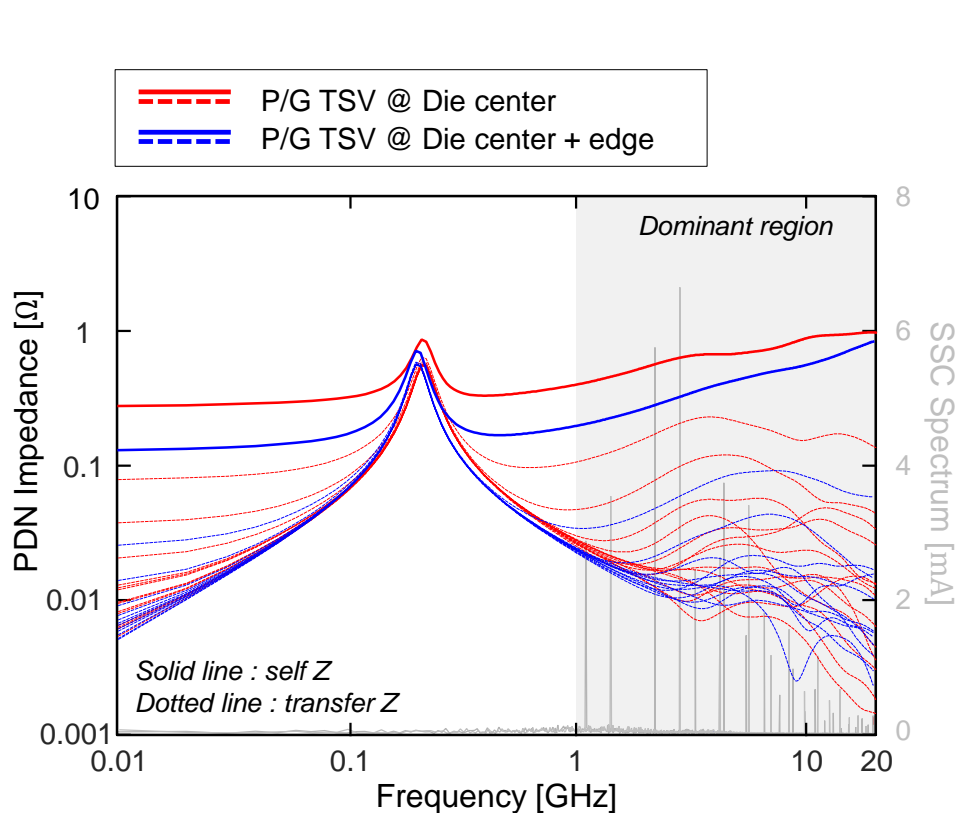
P/G TSV @ Die Center + Edge



## < P/G TSV Array Placement Design Cases for NMC-HBM PDN >

- Power supply to the NMC processor die is provided through the P/G TSVs routed through the HBM die.
- The baseline P/G TSV array at HBM is located at the center of the HBM die, near the HBM TSV area
- Additional P/G TSVs at the die edge are utilized for improving power integrity within NMC-HBM PDN

# Modeled Hierarchic PDN Impedance of NMC-HBM VDD Domain



< Simulated Impedance of 3D NMC-HBM PDN >

< Power Switching Noise at 3D NMC-HBM PDN >

- Through placing additional P/G TSVs at the NMC processor die edge, the overall resistance and loop inductance is reduced from additional parallel PDN loops within P/G HBM TSV.
- As a result, SSN  $V_{\text{peak-to-peak}}$  of the NMC PDN operating at higher clock frequency is suppressed.
  - ✓ Peak-to-peak voltage noise @ 1.4 GHz Clock freq : 0.041 V (5.13%)  $\rightarrow$  0.022 V (2.75%)

- A 3D Near-Memory-Computing Architecture based on High Bandwidth Memory (NMC-HBM) is designed through 3D heterogeneous integration of compute die above HBM.
- Analyzed the advantages of proposed 3D NMC-HBM architecture considering compute characteristics of processor and arithmetic intensity of GEMM workloads.
- Designed and analyzed NMC-TSV for high bandwidth, energy efficient inter-NMC interconnect in terms of signal integrity.
- Modeled and analyzed hierarchic 3D PDN components of 3D NMC-HBM architecture for suppressed impedance and stable power supply noise.
- IR drop & PSIJ from HBM I/O at the 3D NMC-HBM architecture is expected to have a significant effect in performance.



# Thank You!

## HBM



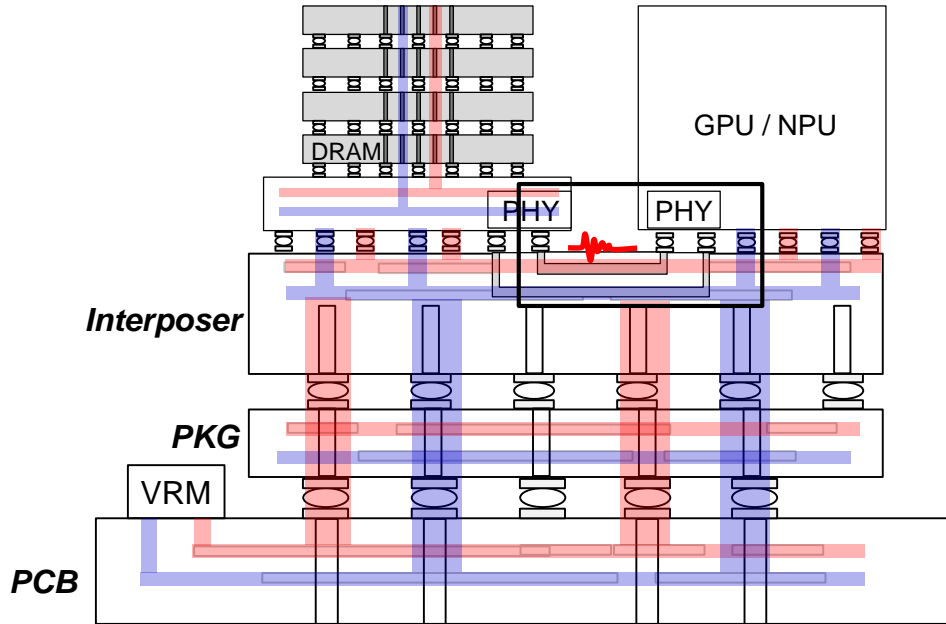
# PDNFormer Based Chip Design Agent for Fast Estimation of Multi-layer and Multi-power PDN Impedance in Customized Base Die in HBM5

Hyunjun An

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

# Power Integrity Issue of HBM5



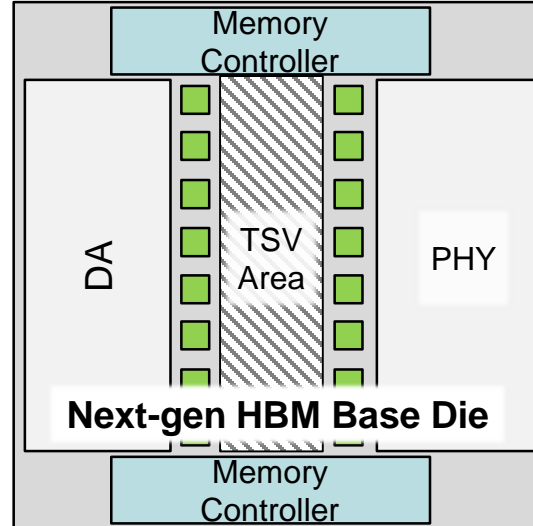
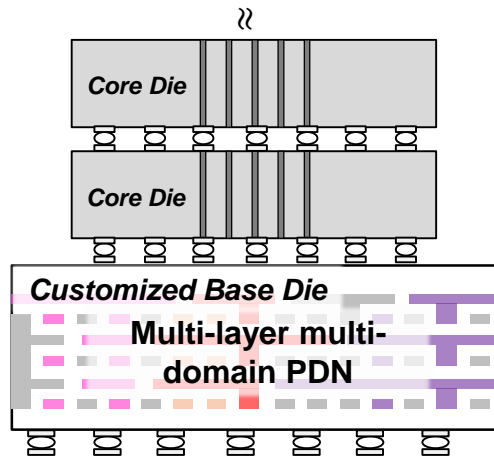
Power noise will be much more severe due to

- ✓ Increased number of I/Os
  - ✓ Lower supply power trend
  - ✓ More functional blocks and less available PDN area
  - ✓ More power domains and interferences
- Tightening power noise margin
  - Causing severe power supply induced jitter (PSIJ) and leading to eye degradation

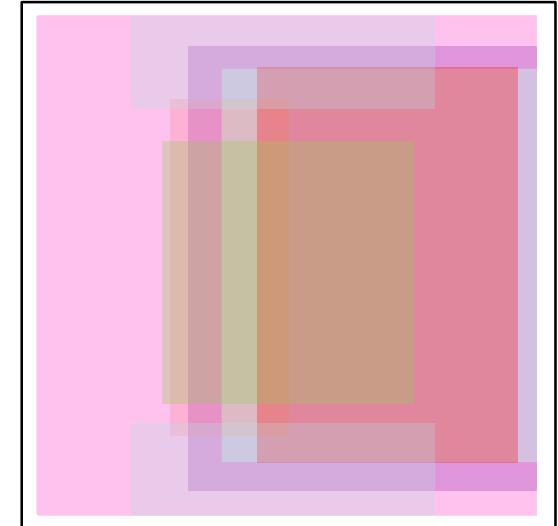


< Power integrity issue of next-generation HBM >

# Multi-layer and Multi-Power Domain of Customized Base Die



■ : SM Core    ■ : Memory controller

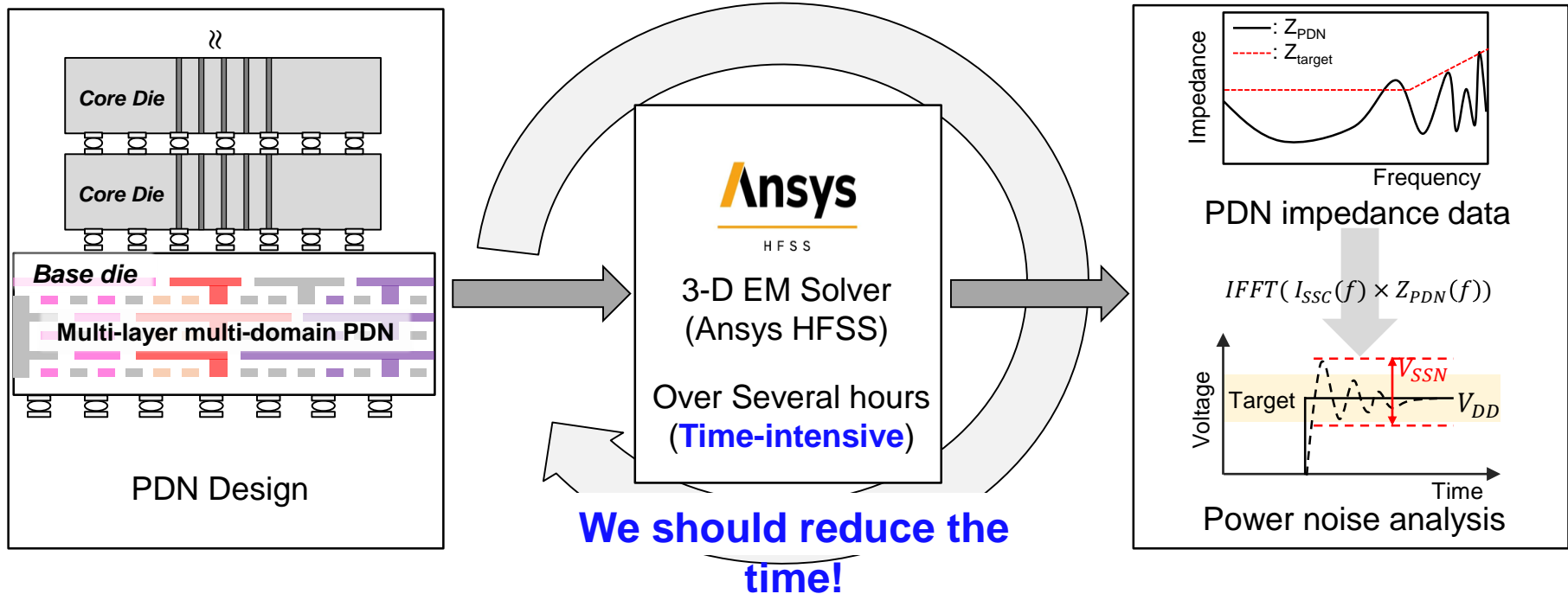


■ : VDDC    ■ : VPP    ■ : VDDQ  
■ : VDDQL    ■ : VCC<sub>MC</sub>    ■ : VCC<sub>SM</sub>

< Multi-layer and multi-power domain PDN in customized HBM base die >

- In customized HBM base die, there are additional power domains such as VCC<sub>MC</sub> and VCC<sub>SM</sub>.
- They occupy their PDNs with different location and various shape and size, even being intersected each others.
  - ✓ PDN design complexity increases dramatically
  - ✓ Increased SSC induces severe SSN
    - PDN design is more challenging in next-gen customized HBM

# Time-consuming Process of PDN Design Cycle and Demand for Fast and Accurate PDN Impedance Estimator

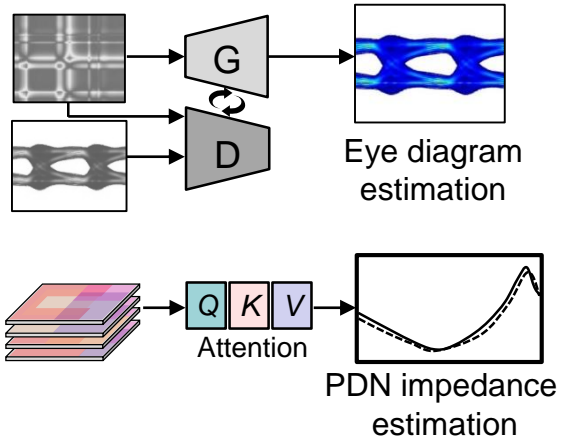


< Time-intensive design cycle of PDN design using 3-D EM solver >

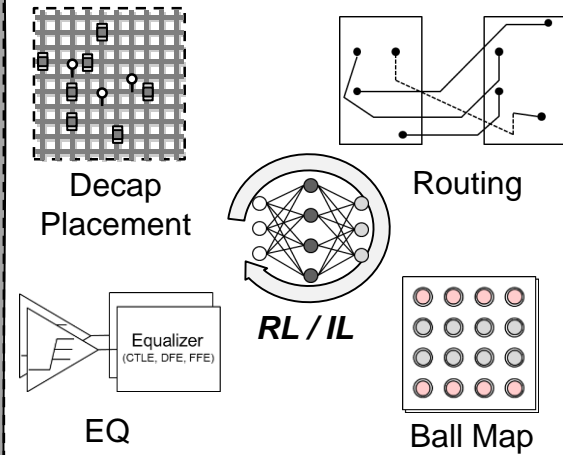
- After PDN design,  $Z_{PDN}$  is calculated by 3-D EM solver, and power noise is obtained by multiplying SSC and  $Z_{PDN}$ .
- However, 3-D EM solver is too time-intensive. So, it's **important to develop fast and accurate PDN impedance estimator** in order to reduce the time for design cycle.

# AI Agent for HBM Design in TeraLab

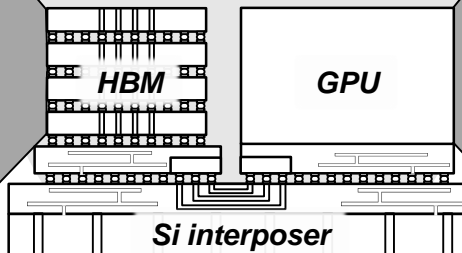
## Simulation Agent



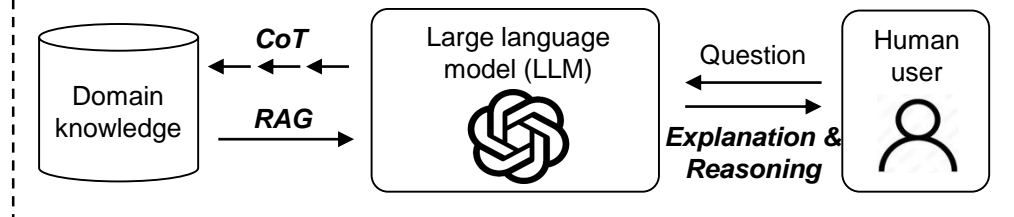
## Optimization Agent



## HBM Design AI Agent



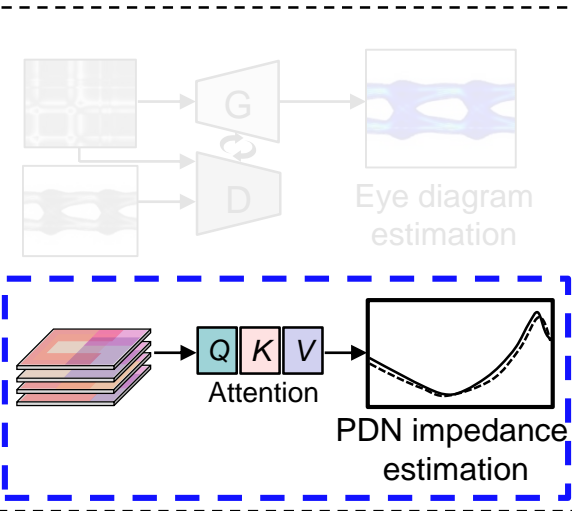
## Human interactive Agent



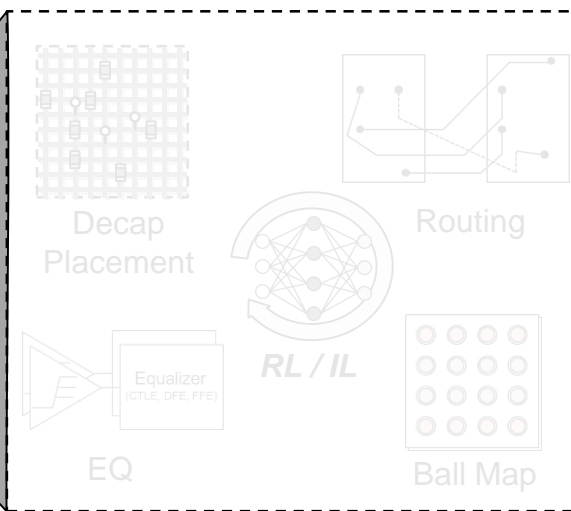
< AI agent for HBM Design in TERA Lab >

# Proposal of Power Integrity Specialized Agent : $Z_{PDN}$ Estimator

## Simulation Agent



## Optimization Agent



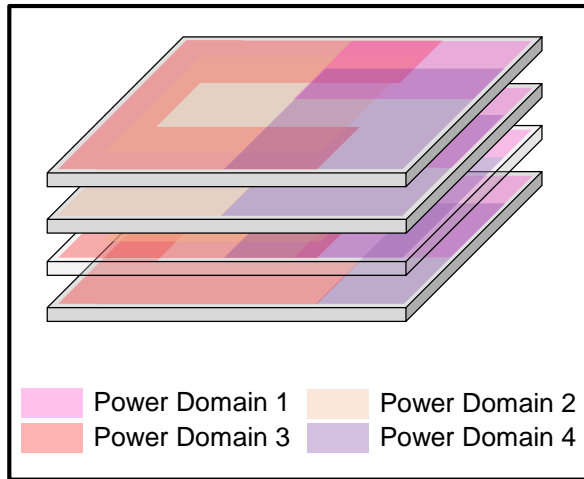
## Human interactive Agent



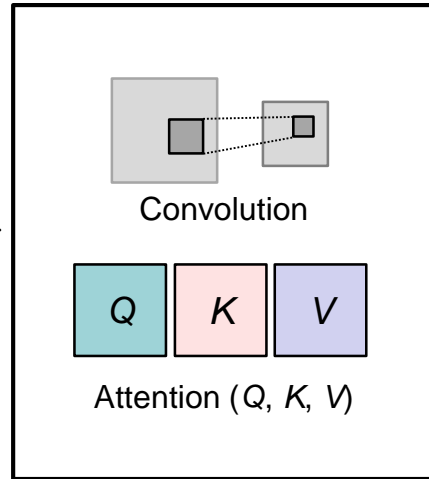
< AI agent for HBM Design in TERALab >

# Proposal of PDNFormer : Multi-layer and Multi-domain PDN Impedance Estimation Network

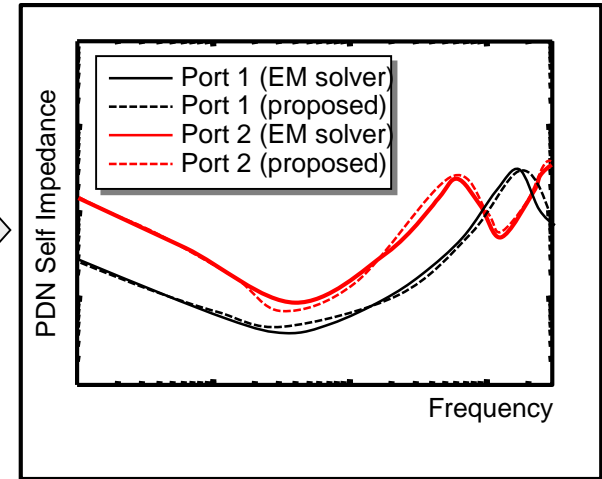
Design of multi-layer PDN with multi power domain



PDNFormer model based PDN Impedance Estimation Network



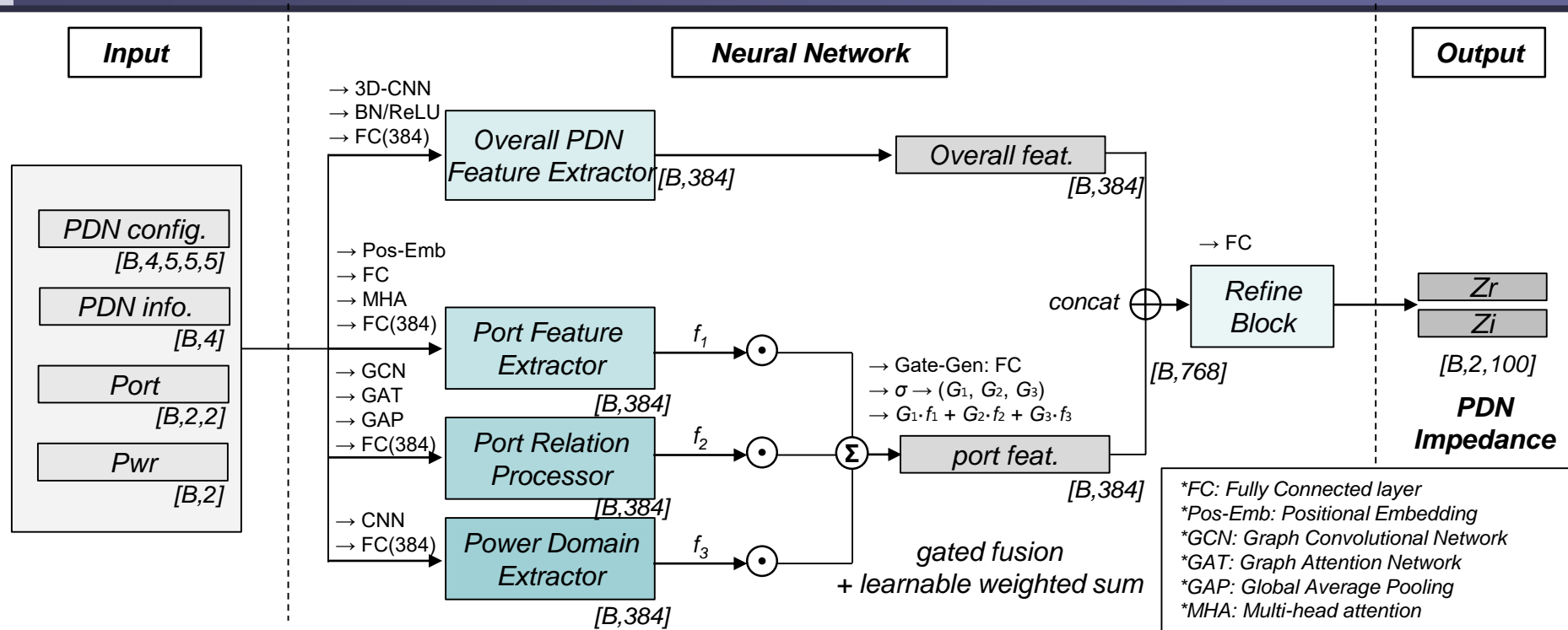
PDN Impedance Estimation



< Overall process for generative model based PDN impedance estimation for multi-layer PDN with multi-power domain >

- Given PDN information, the proposed neural network predicts PDN Impedance in a fast and accurate manner.
- PDNFormer model based multi-layer and multi-power domain PDN impedance estimation is proposed.
  - ✓ CNN effectively captures spatial relations between different power domains
  - ✓ Attention block understands global relations and integrate all the information from CNN

# Overall Block Diagram of Proposed Neural Architecture



< Block diagram of proposed neural architecture >

- Overall PDN configurations, geometrical properties are extracted through:
  - ✓ Overall PDN Feature Extractor Block
- Port information (including port position and interactions) and power domain features (to which ports are assigned) are extracted through
  - ✓ Port Feature Extractor, Port Relation Processor, Power Domain Extractor
  - ✓ Through gated fusion with learnable weights, the most effective information is emphasized.
- Refine block integrate these features and converts them into PDN impedance.



# Result : Performance Comparison Table

Table I. Performance comparison by data scale type

Training Data Scale Type (Model type : unified)	Avg. MAE of Mag/Phase	
	Mag [dBΩ]	Phase [rad]
PDNFormer	3.44	0.25

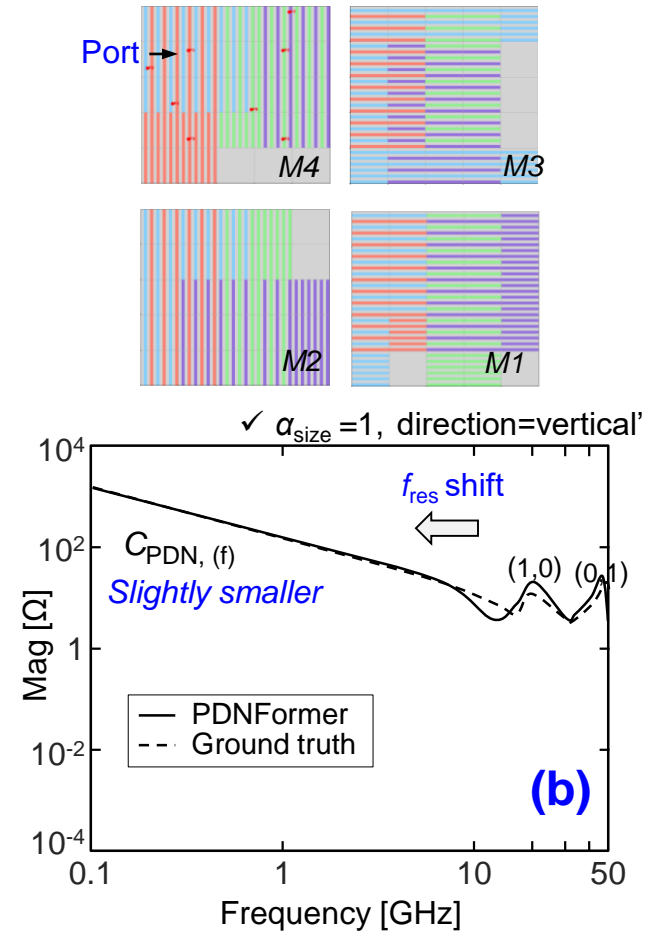
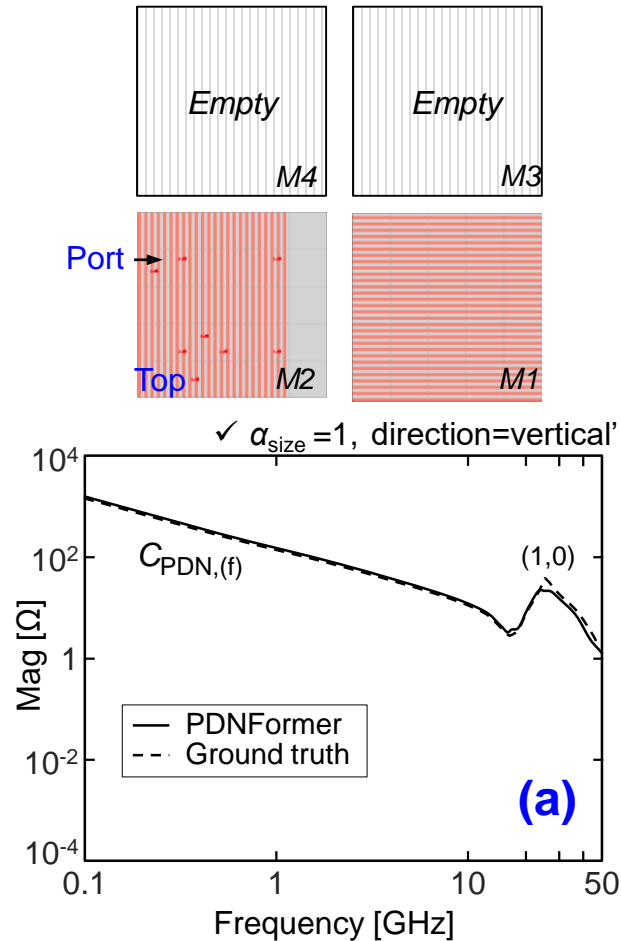
Table II. Inference time comparison

Computation method	Computing device	Inference time for one data sample
Full 3-D EM solver (Ansys HFSS)	Intel i3-14100 CPU	10,000 s
Proposed	Intel i3-14100 CPU	< 1 ms
	NVIDIA RTX3090 GPU	< 1 ms

- Performance comparison:

- ✓ **Accuracy:** The proposed model achieves an average MAE of 3.44 dBΩ for magnitude and 0.25 rad for phase.
- ✓ **Time efficiency:** Compared to a conventional EM solver (Ansys HFSS, ~10,000 seconds), the proposed method reduces inference time to **under 1 ms**.

# Result : Data Sample



< Data sample of PDN impedance >

- It can be clearly observed that the predicted impedance profile aligns well with the theoretical PDN behavior, capturing both the overall impedance trend and specific mode resonances.

# Thank You!

## HBM

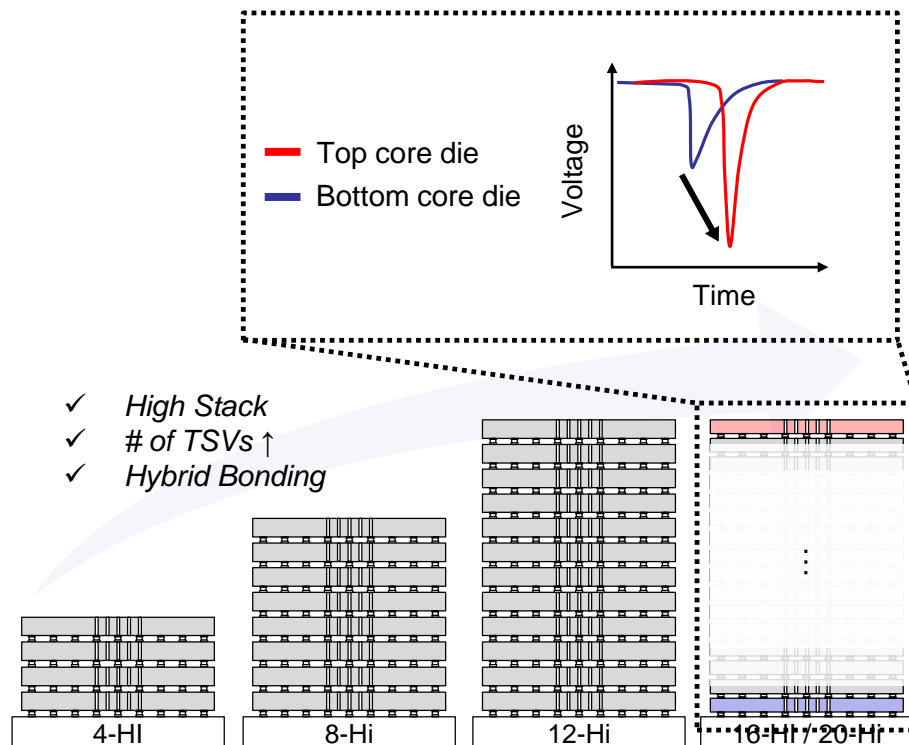
# Transformer-based Reinforcement Learning for TSV Placement and Design Optimization considering IR Drop in HBM5

Eunji Seo

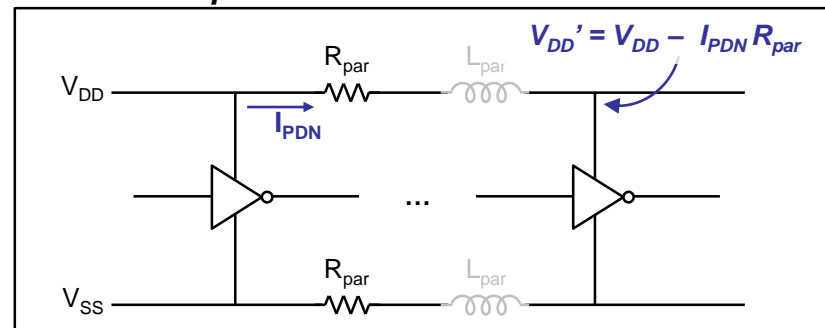
Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

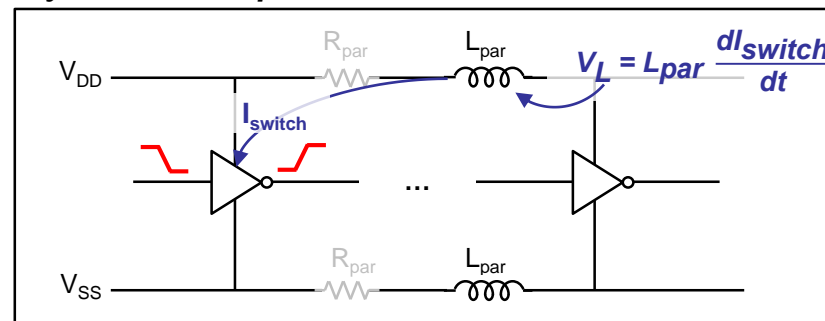
# Die-to-Die IR Drop Issues in Highly Stacked Next Generation HBM



## Static IR drop



## Dynamic IR drop

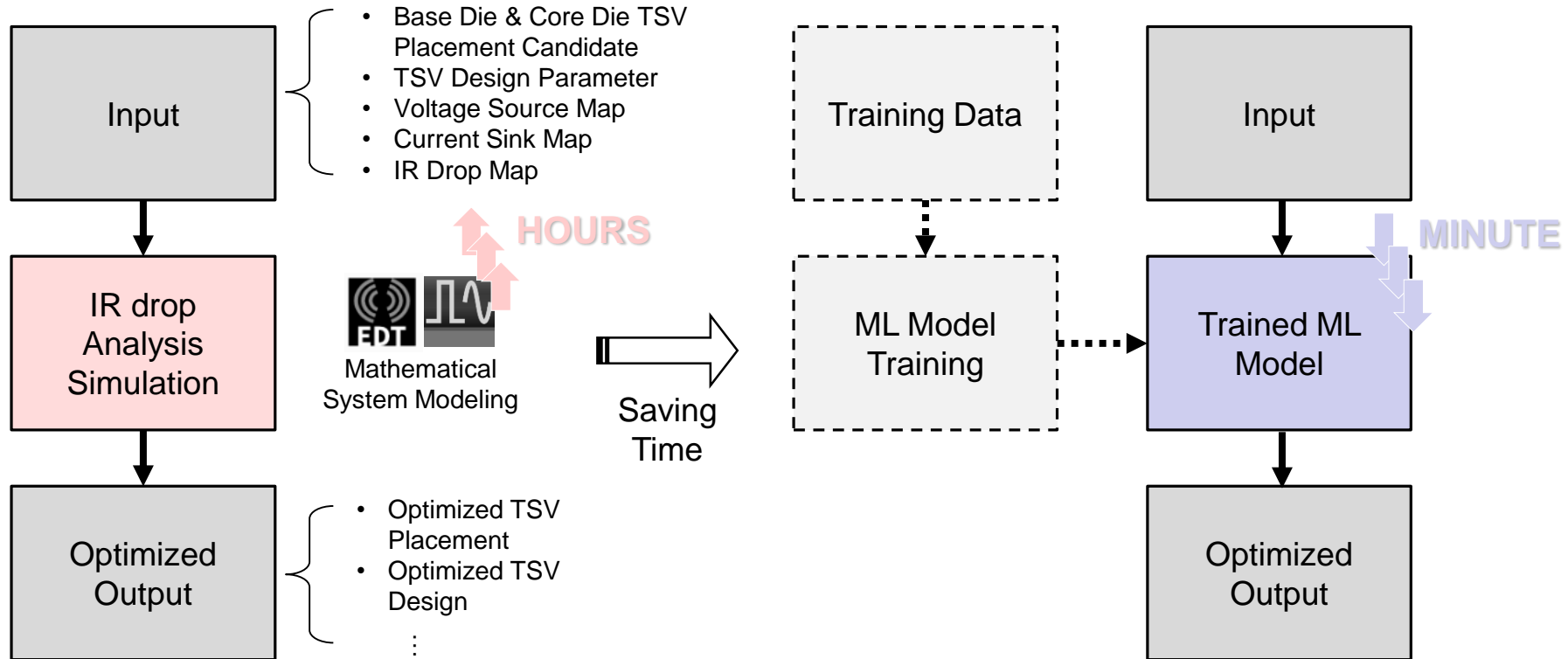


< Increased IR drop in highly stacked HBM generations >

< Static IR drop and Dynamic IR drop >

- As HBM technology evolves, height of HBM increases and IR drop between the bottom and top core dies occurs.
- Static IR drop refers to the voltage drop caused by resistive components under steady-state current conditions.
- Dynamic IR drop represents the voltage drop induced by rapid changes in current over time.

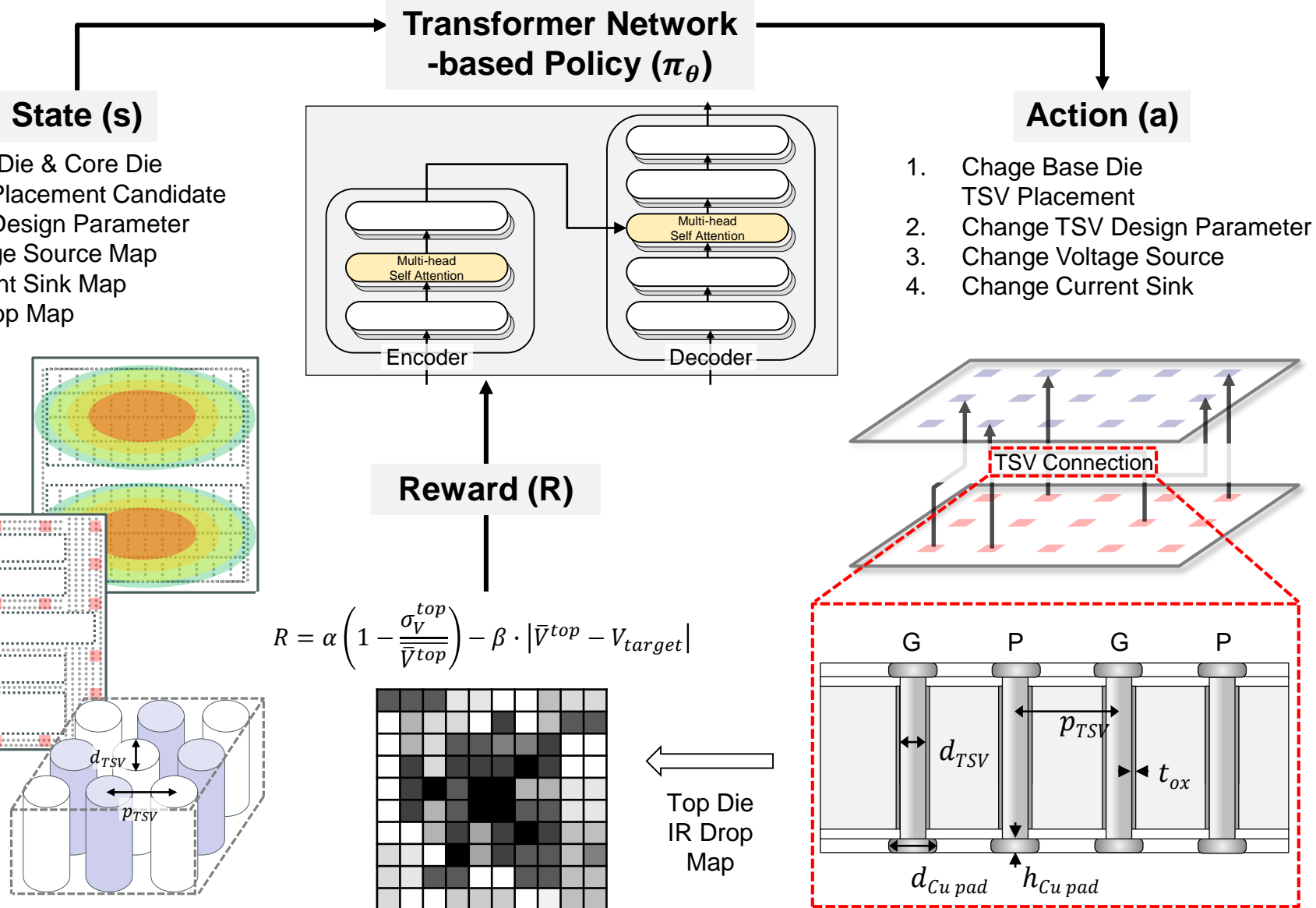
# Decreased Simulation Time by using AI for TSV optimization considering IR Drop



< Conventional & ML-based TSV optimization process considering IR drop >

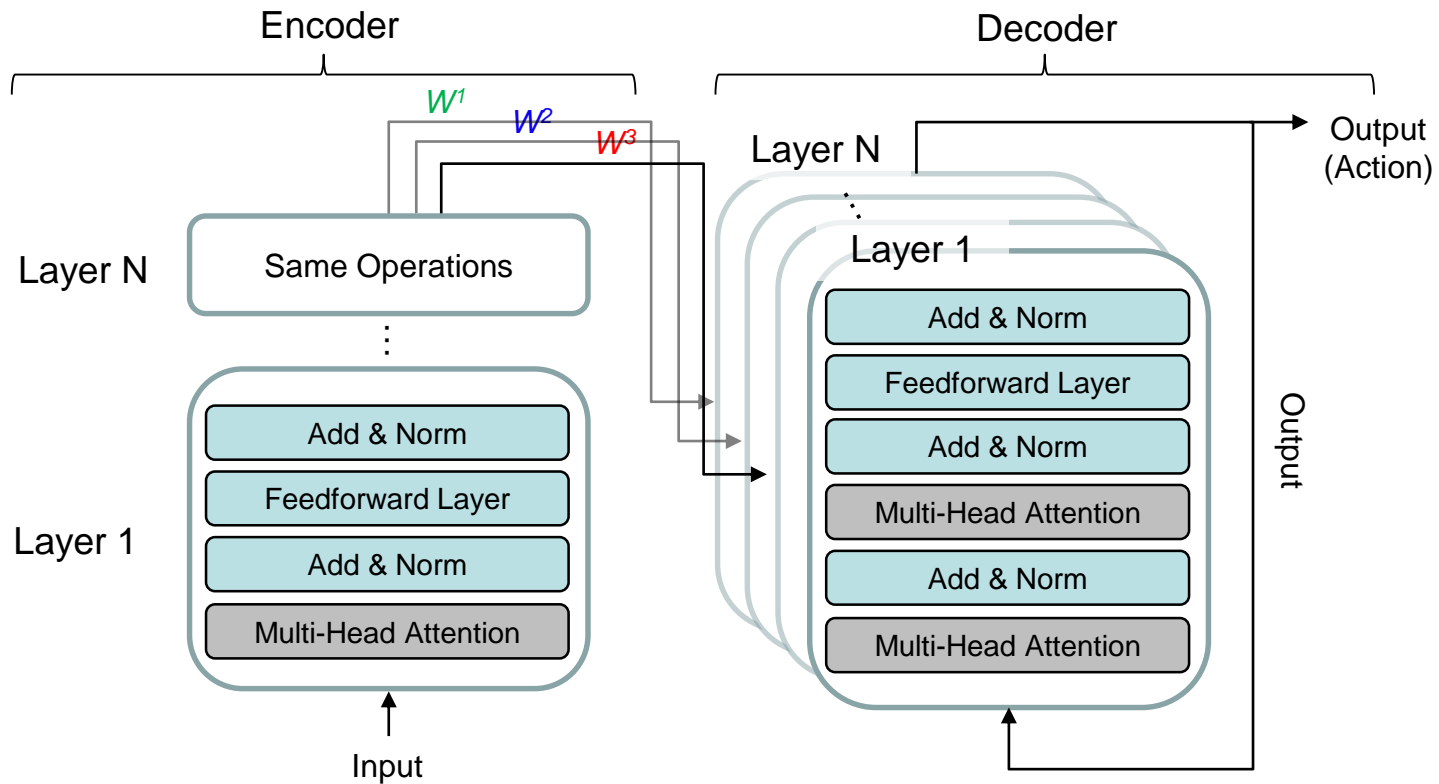
- TSV placement and design require simulation tools such as Q3D and ADS.
- Each simulation run takes over 3 hours, so applying ML-based network can significantly reduce the overall computation time.

# Proposal of Transformer Network-based Reinforcement Learning for TSV Placement and Design Optimization considering IR Drop



< Overview of proposed Transformer network-based RL for TSV optimization >

# Defined MDP Policy – Transformer Model Network

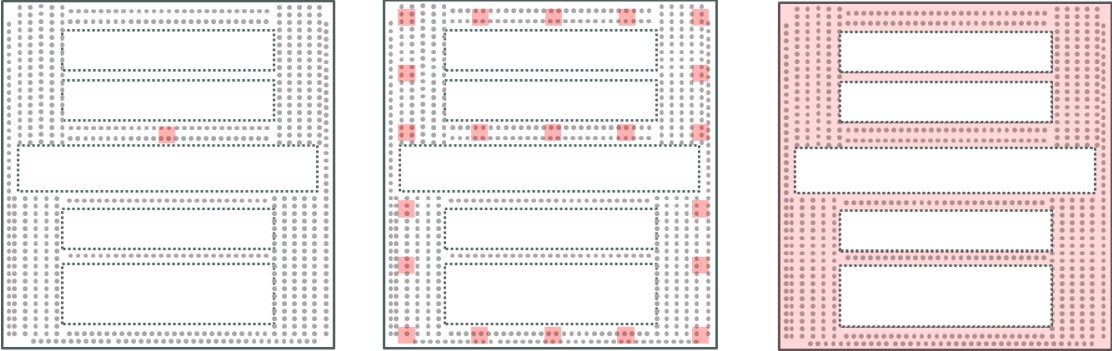
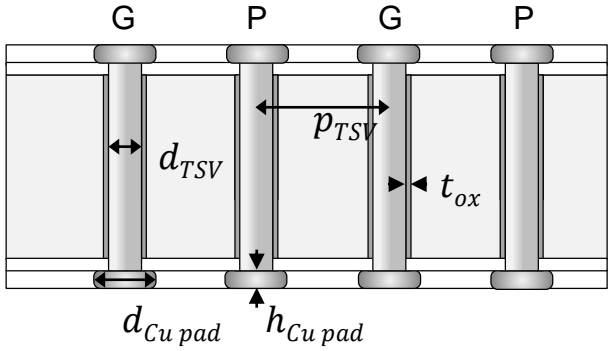
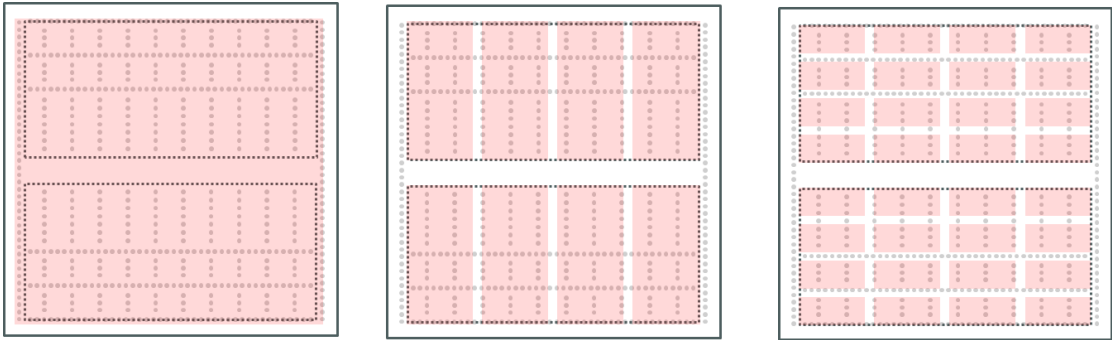
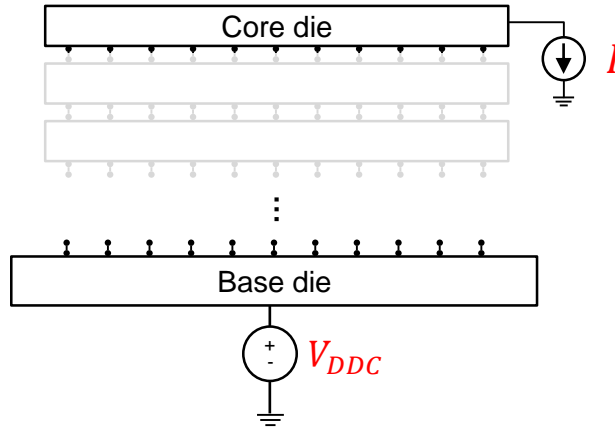


< Architecture of Transformer network >

- The agent is implemented using a transformer model to predict the optimal TSV placement and design.
- Due to the attention mechanism, the model effectively captures spatial dependencies in the PDN layout.
- This allows scalable and accurate design optimization without relying on domain-specific heuristics.



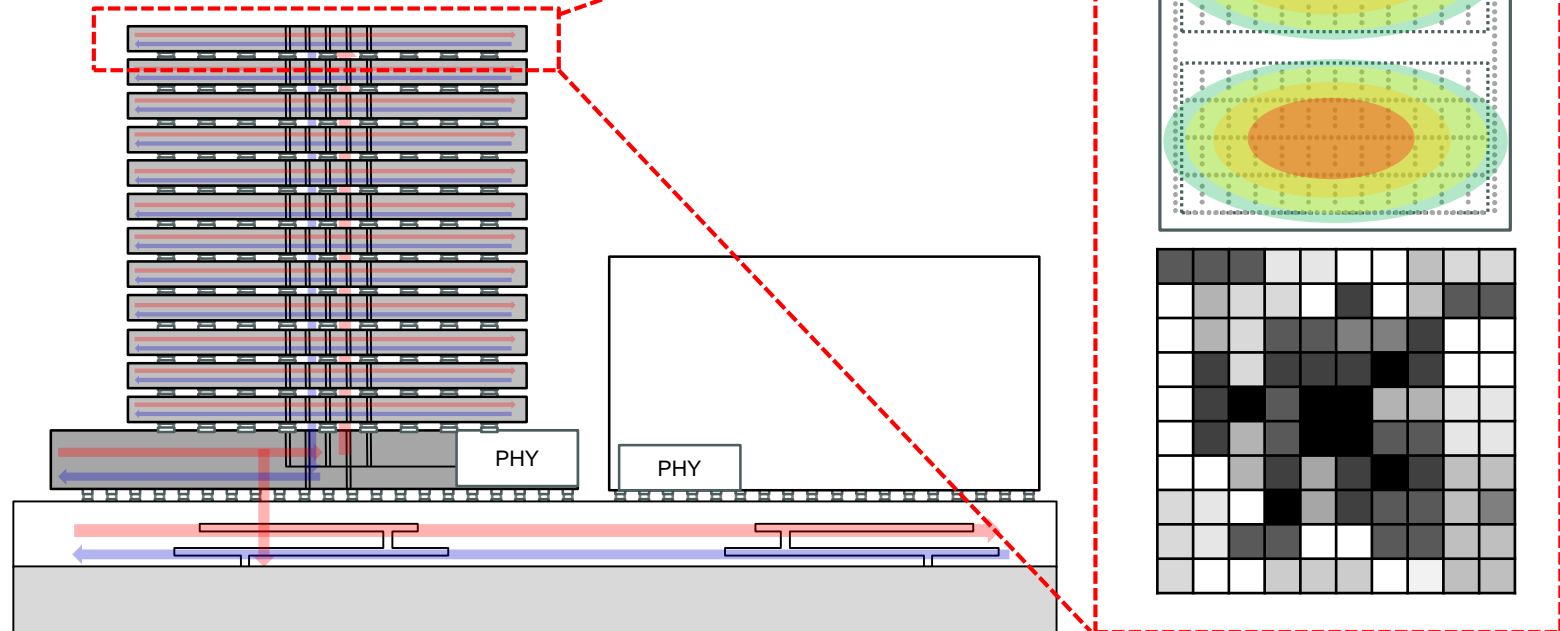
# Defined MDP Action – TSV/PDN Design Variables

Base Die TSV Placement Candidate / Voltage Supply Location	TSV Design Parameter
	
Core Die TSV Placement Candidate	Voltage Supply / Current Sink Values
	

< Action space of the proposed method, including TSV placement, design parameters, and PDN configuration >

# Defined MDP Reward – Uniform IR Drop Distribution

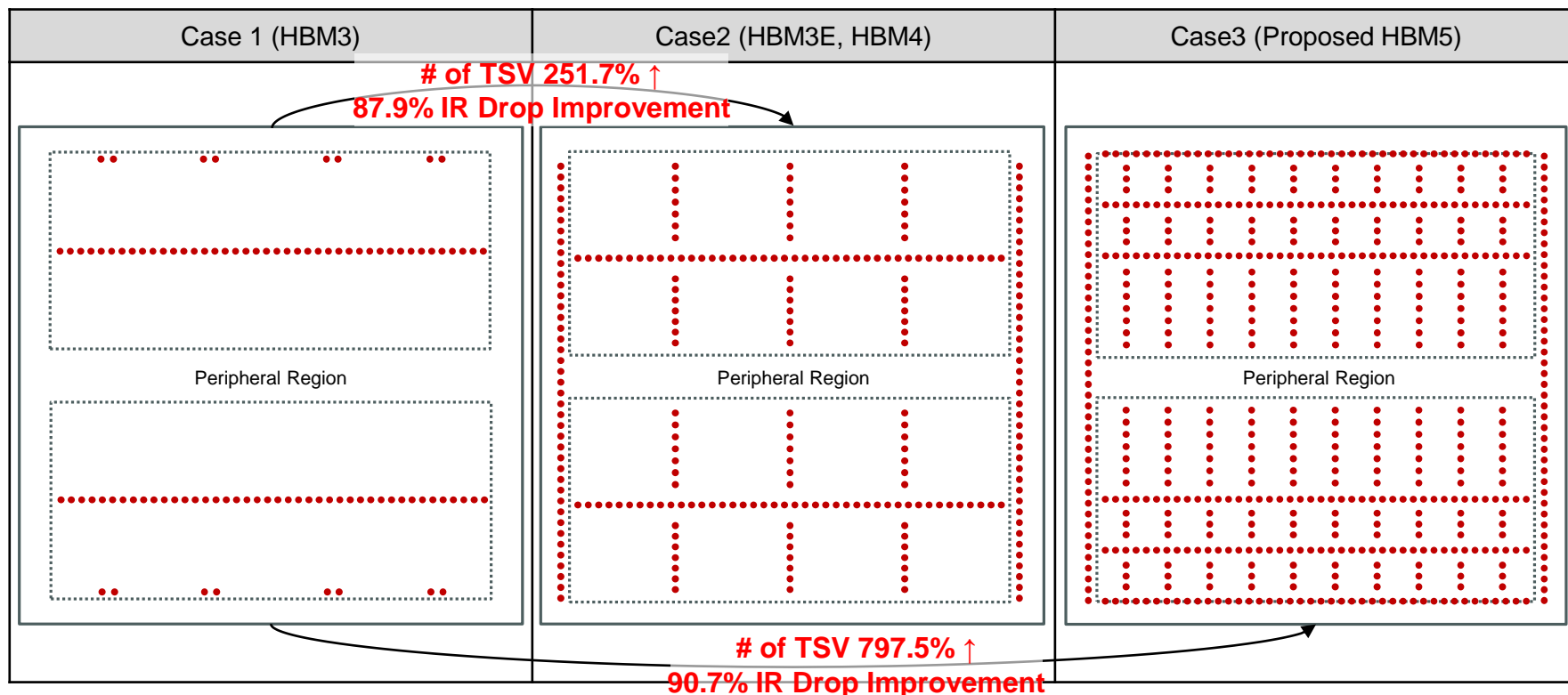
$$Reward = \alpha \left( 1 - \frac{\sigma_V^{top}}{\bar{V}^{top}} \right) - \beta \cdot |\bar{V}^{top} - V_{target}|$$



< Voltage distribution at the top die visualized as heatmap and grid-based IR drop map >

- The first term assigns a weight to uniformity, increasing the reward as the voltage becomes more evenly distributed.
- The second term introduces a penalty for deviation from the average voltage, preventing the voltage from skewing too high or too low.

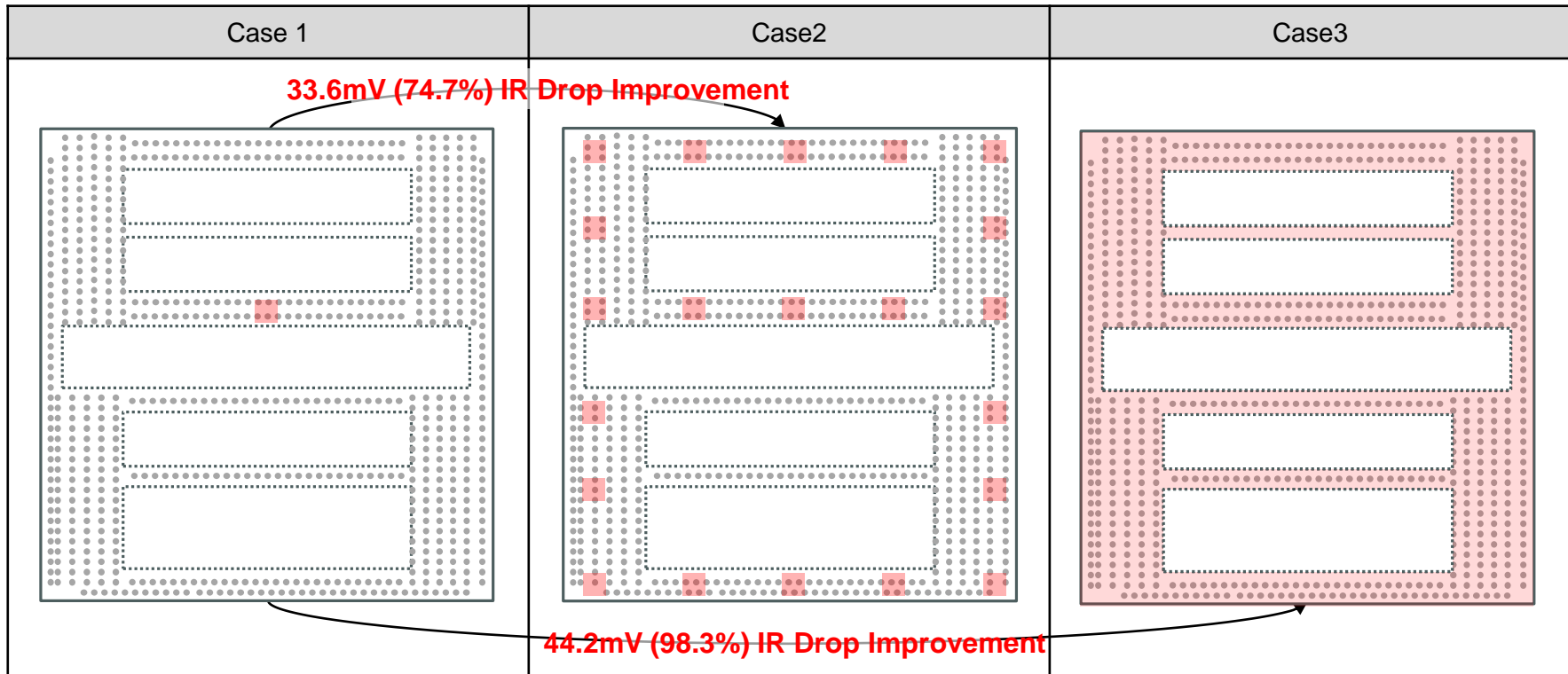
# IR Drop Analysis [1/2] – Based on the TSV Placement Candidate According to the HBM Generations



	TSV Design		Core Die TSV Candidate	Base Die TSV Candidate	Current Sink [A]	Voltage Source [V]	Top Core Die Voltage [V]	Max IR Drop [mV]
	$d_{\text{TSV}}$	$p_{\text{TSV}}$						
Case1	5um	30um	118*20=2360	402	6.7	1.05	0.9263	123.75
Case2			414*20=8280	1148			1.0351	14.863
Case3			1060*20=21200	1782			1.0385	11.463

< Number of TSV candidate and IR drop results based on placement map at the core & base die >

# IR Drop Analysis [2/2] – Based on the Number of Voltage Sources at the base die



	TSV Design		Number of Base Die Voltage Supply	Current Sink [A]	Voltage Source [V]	Top Core Die Voltage [V]	Max IR Drop [mV]
	d <sub>TSV</sub>	p <sub>TSV</sub>					
Case1	5um	30um	1	6.7	1.05	1.00498	45.0
Case2			38			1.03854	11.4
Case3			1782			1.04234	0.766

< IR drop reduction according to the number of voltage source at the base die >

# Conclusion

- As HBM continues to scale toward higher stacking, IR drop becomes a critical issue. To mitigate this, optimization of TSV design and placement is essential.
- Moreover, IR drop is significantly influenced not only by TSVs themselves, but also by how voltage sources and current sinks are distributed across the base die and core dies.
- Considering all these factors while minimizing simulation time requires AI-based optimization.
- Therefore, this study proposes an AI agent framework that derives the optimal TSV placement for 16- and 20-layer HBM stacks, using IR drop uniformity as the reward criterion.

# Thank You!

## HBM

# Mamba-based Reinforcement Learning (RL) method for HBM5 PDN Optimization Agent

**Byeongmok Kim**

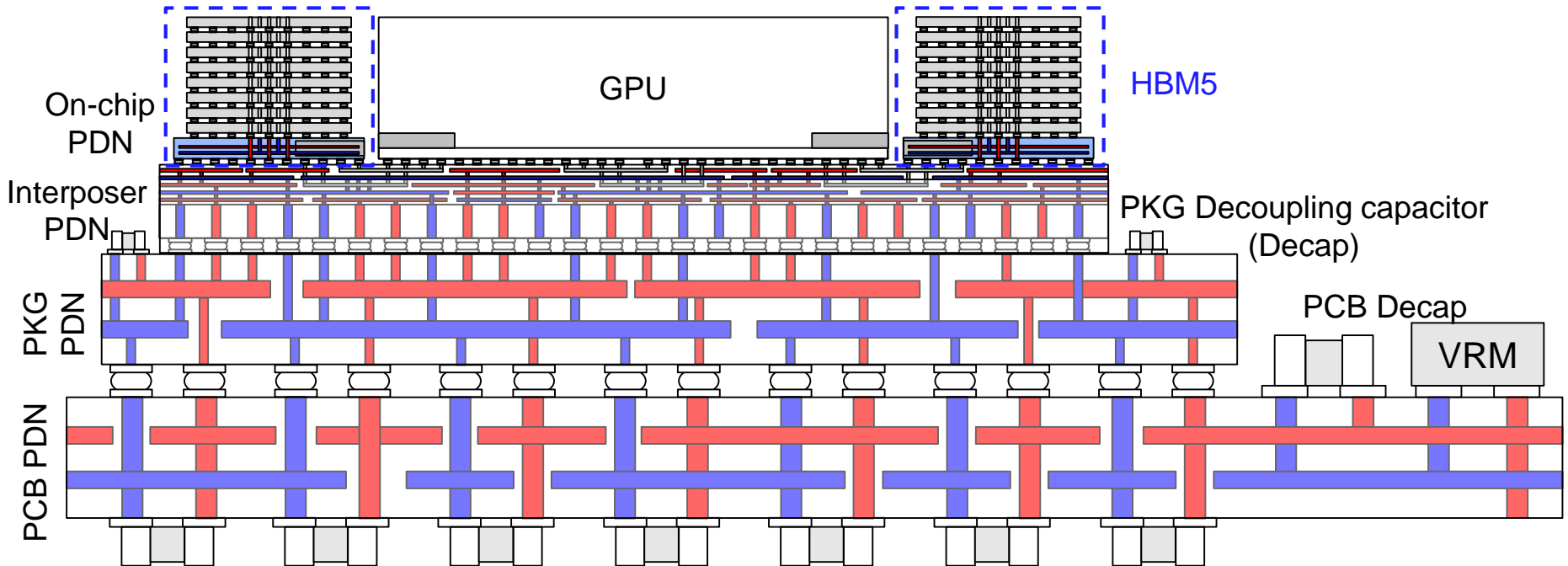
Advising Professor : Prof. Jounggho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering, KAIST

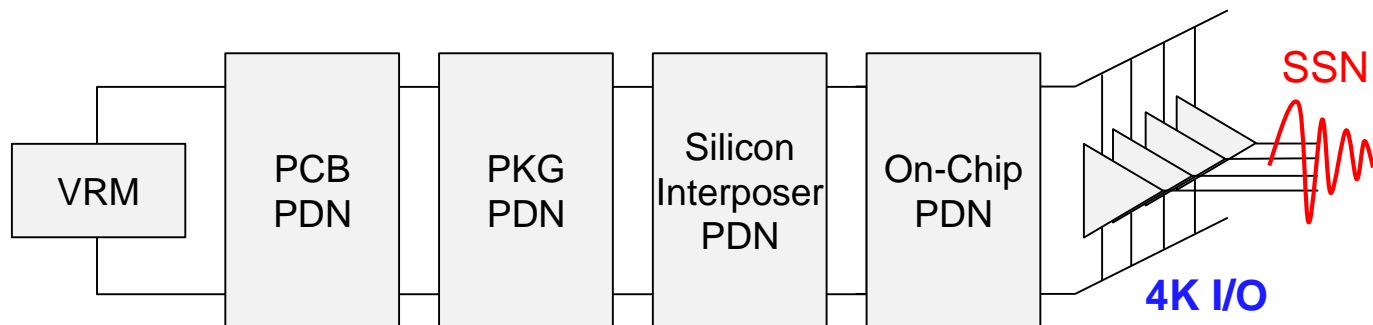
June 11<sup>st</sup>, 2025

# Challenge in HBM5 Design :

## The Hierarchical PDN & Simultaneous Switching Noise (SSN)



< Cross-sectional view of PDN in GPU >

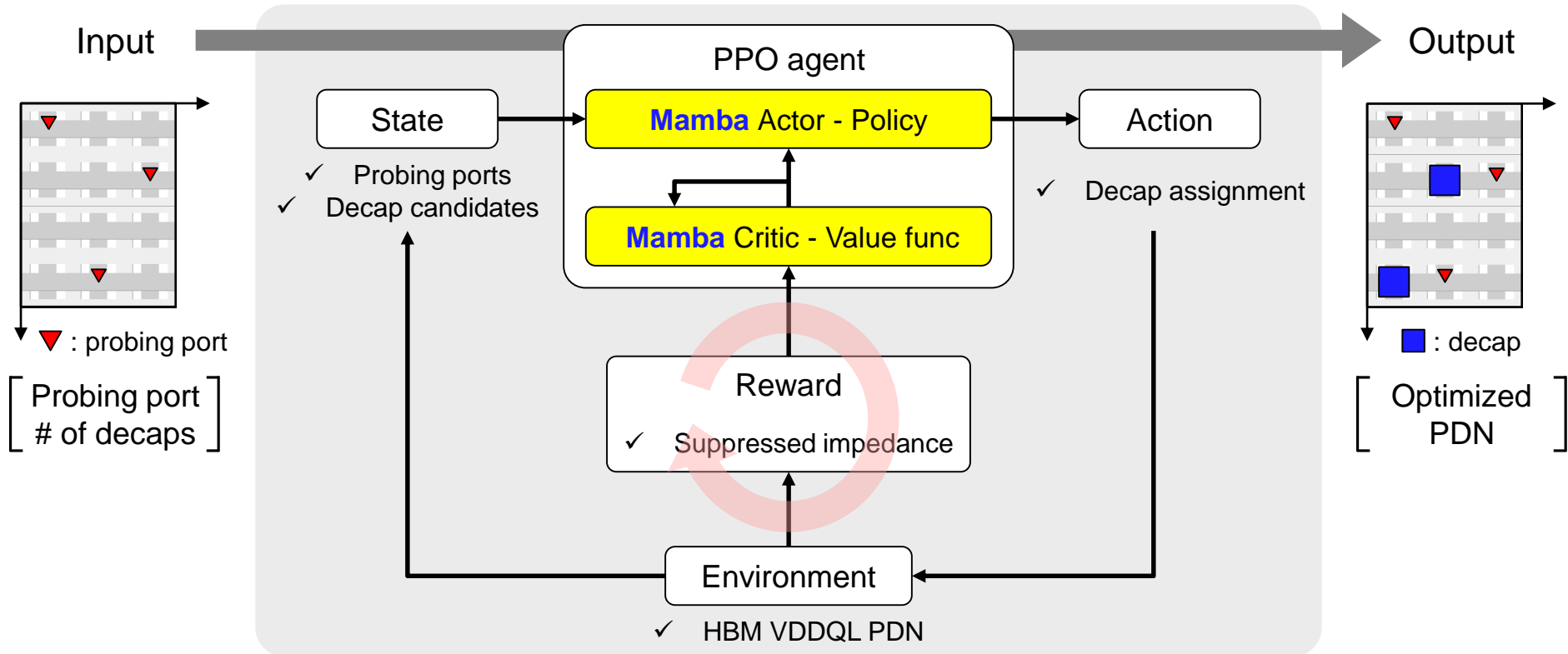


< Conceptual view of SSN >

✓ Too many constraints and intertwined objectives to optimize



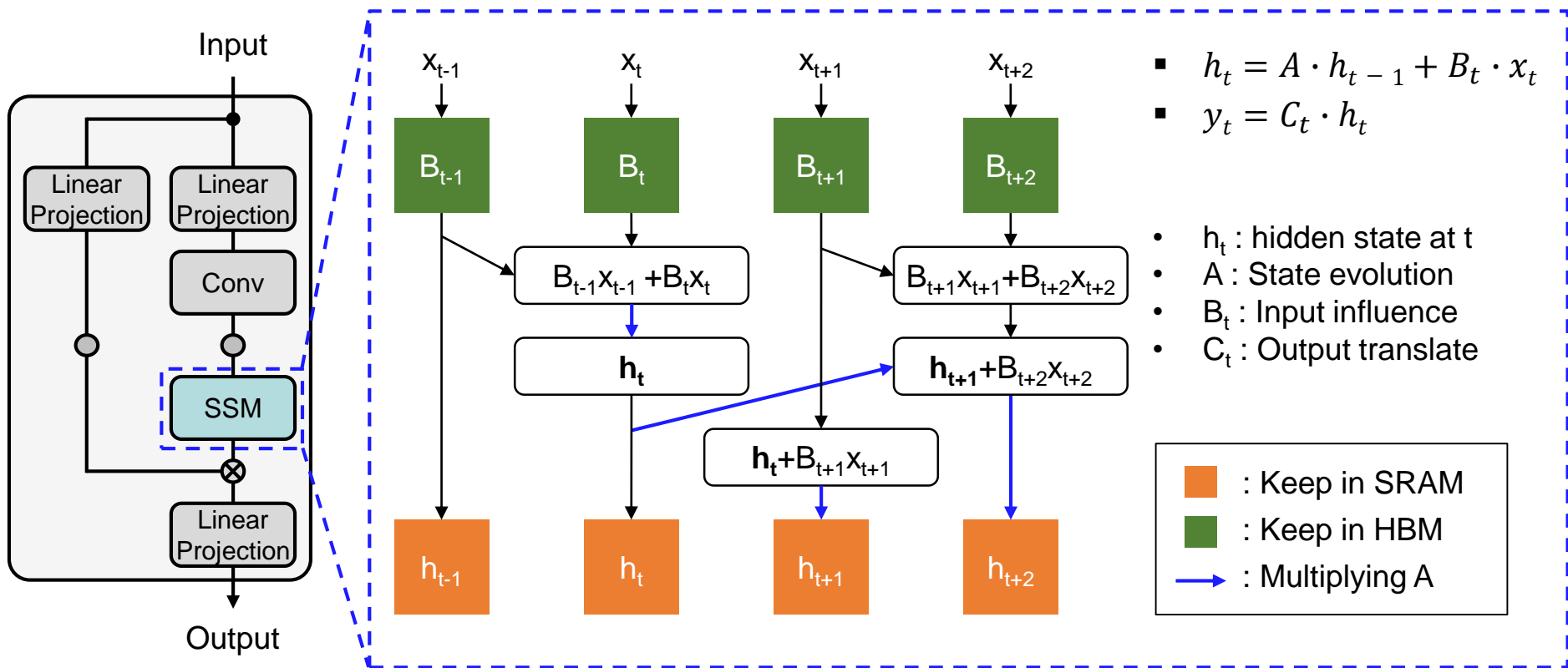
# Proposal of Mamba-based RL Method for HBM5 PDN Optimization Agent



< Markov decision process (MDP) of the proposed Mamba-based RL method >

- The inputs include the location of probing ports and the number of decaps to be assigned ( $m$ ).
- The output is completed HBM5 VDDQL PDN map with assigned decaps.
- Proximal policy optimization (PPO) algorithm is used for the RL pipeline.

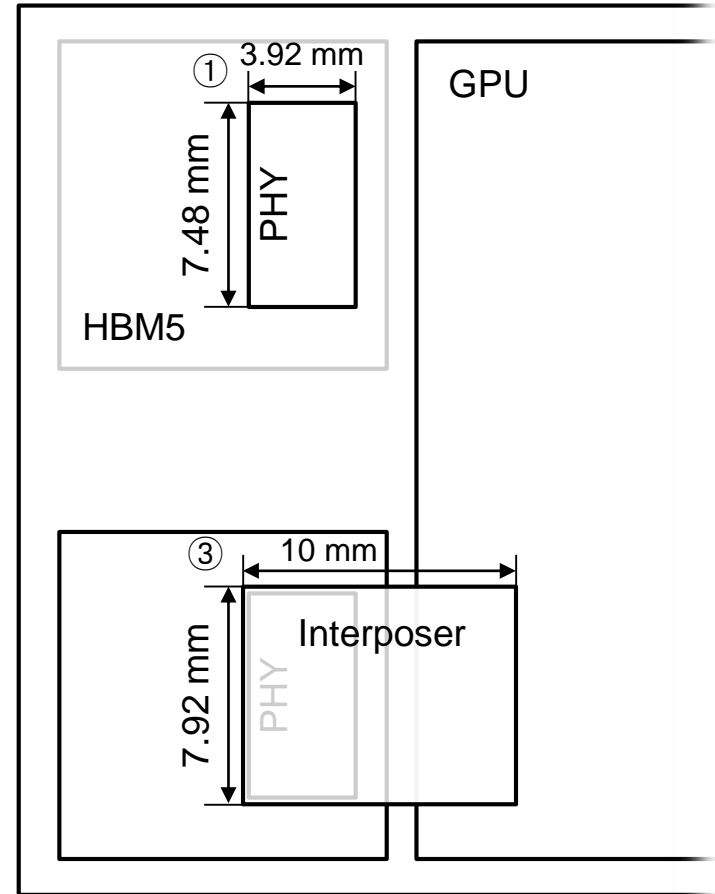
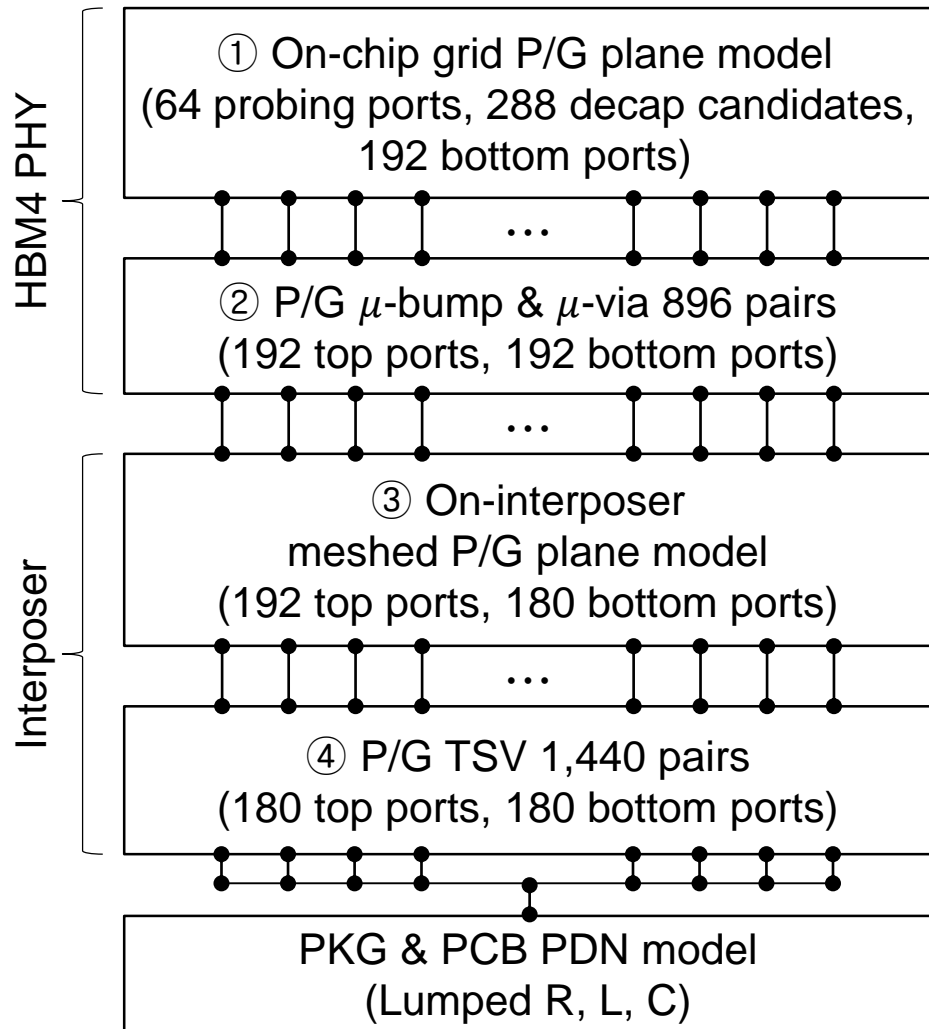
# Introduction to Mamba Architecture



## < Introduction to Mamba architecture >

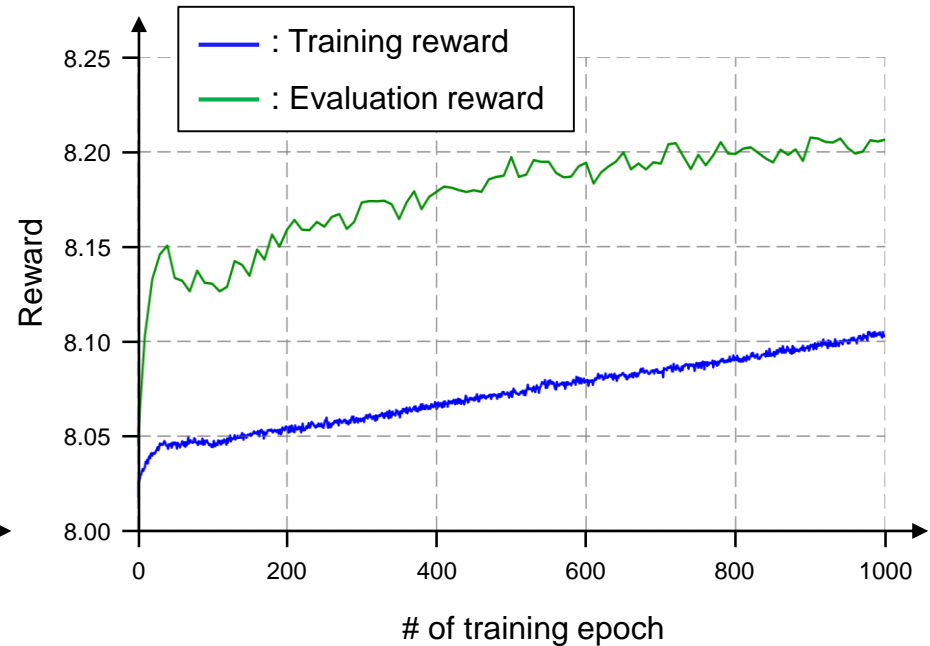
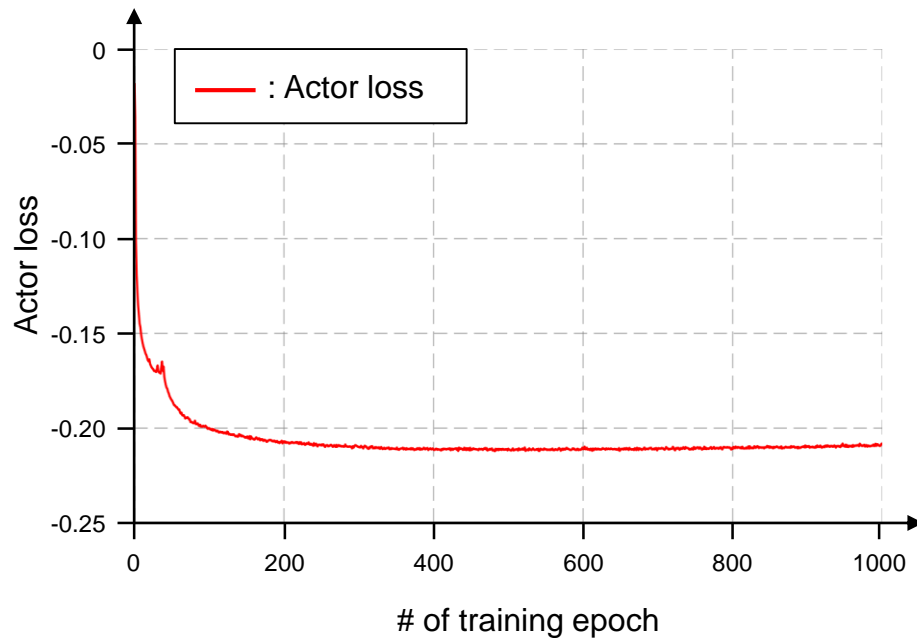
- Mamba is based on SSM with selective scheme and parallel scan algorithm.
  - **Parallel training and linear inference** with  $O(N)$  linear computing.
  - **No need to use several GBs of KV cache** like Transformer architecture.
- Mamba -block can be stacked and the output of one block can be used as the input for the next Mamba -block.

# Modeling of Hierarchical HBM5 VDDQL PDN



< Modeling and configuration of HBM5 VDDQL PDN >

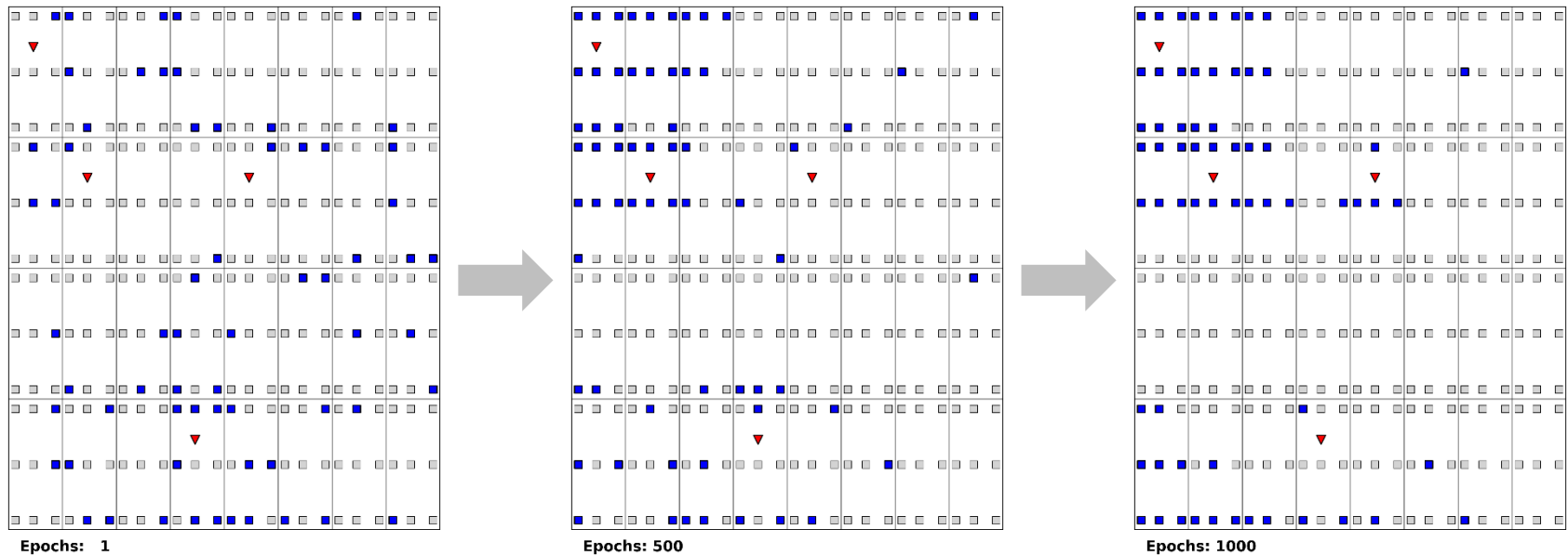
# Training Loss and Reward Convergence Verification of the Proposed Method



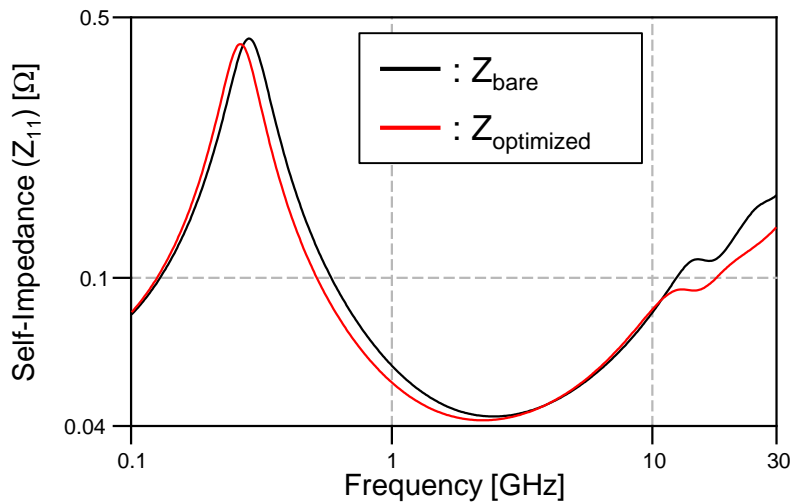
< Training loss and reward convergence graphs of the proposed method >

- The loss curves show convergence to -0.2, which means the policy optimization process stabilizes over training epochs.  
→ Negative actor loss indicates that the policy is improving in an optimal direction.
- The training reward curve also steadily increases and evaluation reward curve stabilizes after 800 epochs.

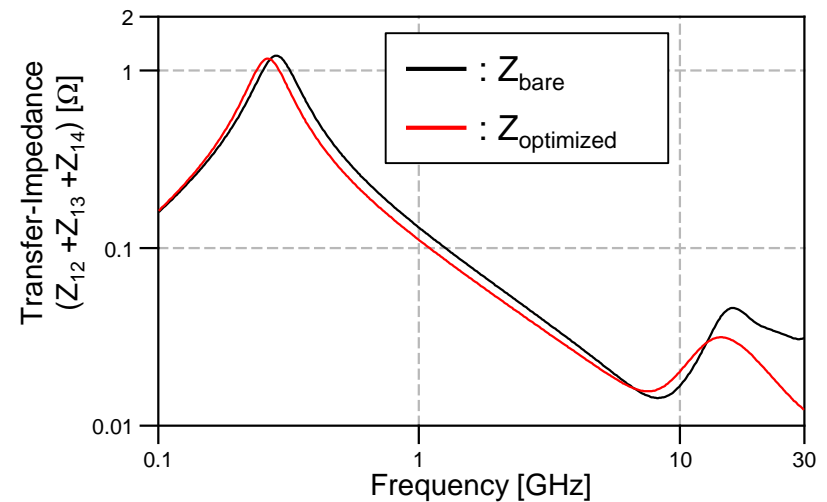
# Optimized Result in HBM5 VDDQL PDN : Decap Assignment and Impedance Suppression



< 64 decaps assignment by proposed method >



< Self-impedance curve comparison >



< Transfer-impedance curve comparison >

# Performance Verification by Comparison to Conventional Optimization Algorithm and Transformer-based RL

64 decaps assignment	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Avg Reward	Time [sec]
RS {500}	8.127	8.010	8.488	8.320	8.081	8.061	8.140	8.149	8.172	28.5
GA {500}	8.146	8.026	8.651	8.452	8.085	8.097	8.168	8.186	8.226	30.9
Transformer-based RL {1}	8.199	7.988	8.787	<b>8.535</b>	7.955	8.029	<b>8.202</b>	8.168	8.233	0.1
<b>Proposed {1}</b>	<b>8.257</b>	<b>8.034</b>	<b>8.858</b>	8.520	<b>8.129</b>	<b>8.127</b>	8.162	<b>8.233</b>	<b>8.290</b>	<b>0.1</b>

※ { } is # of reward calculation, RS : random search, GA : genetic algorithm

< Comparison between RS, GA, Transformer-based RL and proposed method >

- For performance verification, the proposed method is applied to unseen test probing port data sets.
- With a single inference, the proposed method outperforms conventional optimization methods with less computation time, verifying its reusability.  
→ **99.65 %** ↓ than RS {500} and **99.68 %** ↓ than GA {500}
- The proposed method requires 17.9% less training time than Transformer-based RL.  
→ 3, 200 minutes compared to 3, 900 minutes.

- ✓ For the first time, I proposed a Mamba-based RL method for HBM5 PDN optimization agent.
- ✓ The proposed method can instantly find optimal decap assignment solution while suppressing PDN impedance.
- ✓ The proposed method outperforms the RS and GA by achieving better performance and Transformer-based method with lower computational cost, with reusability.
- I have proven that the Mamba network can be used as a policy network for optimization problems.

# Thank You!

## HBM



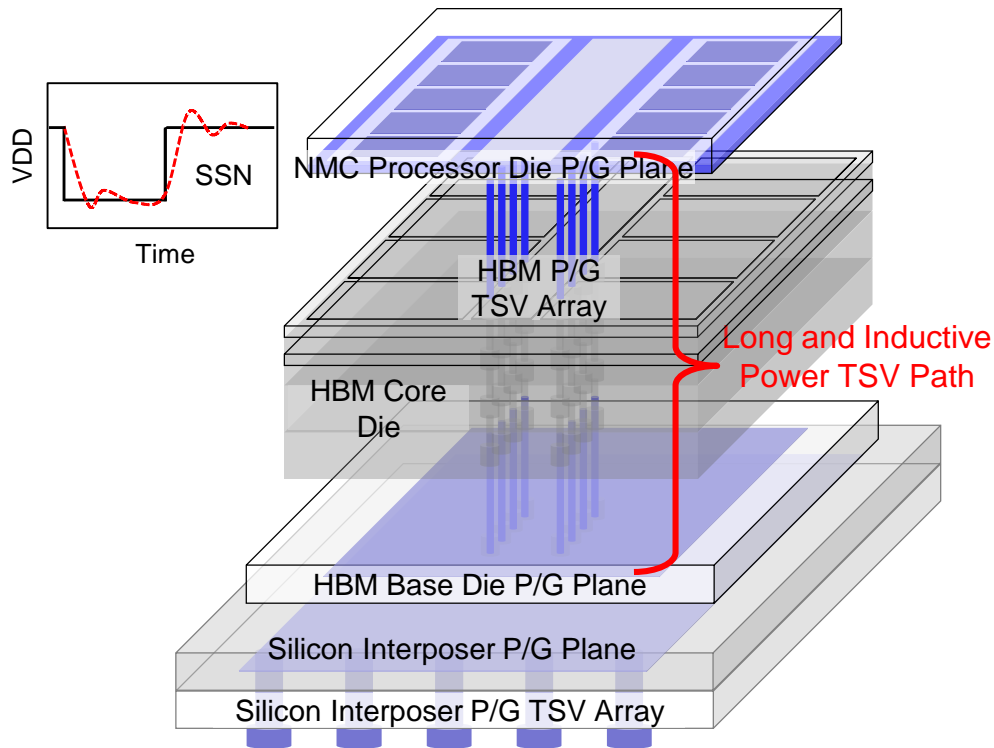
# Devformer with Collaborative Distillation for Optimal Decoupling Capacitor Placement in HBM5 Custom Base Die

Haeyeon Kim

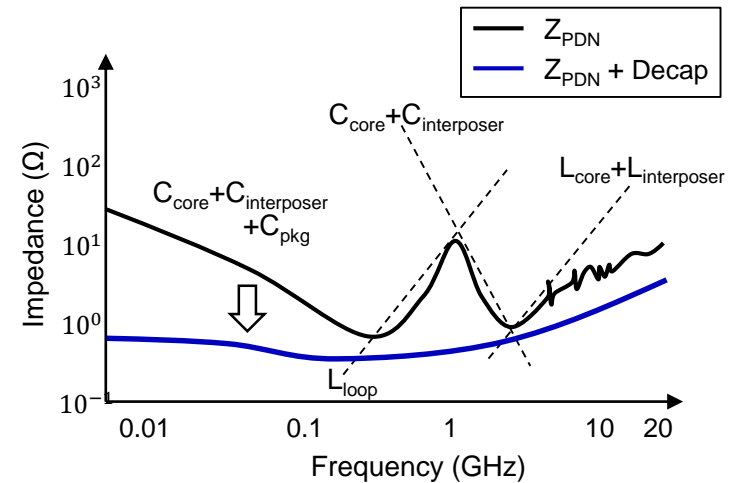
Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

# Power Integrity Issues in Hierarchical Power Distribution Network of HBM Gen. 5 NMC-Integrated Architecture



< Power Integrity Issues in HBM Gen. 5 NMC-Integrated Architecture >

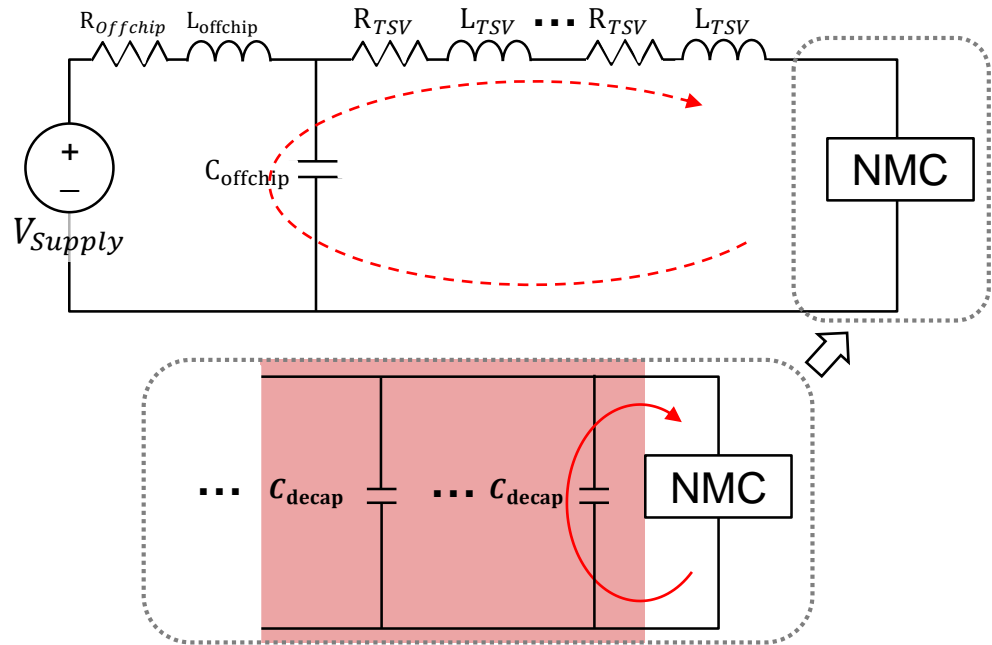
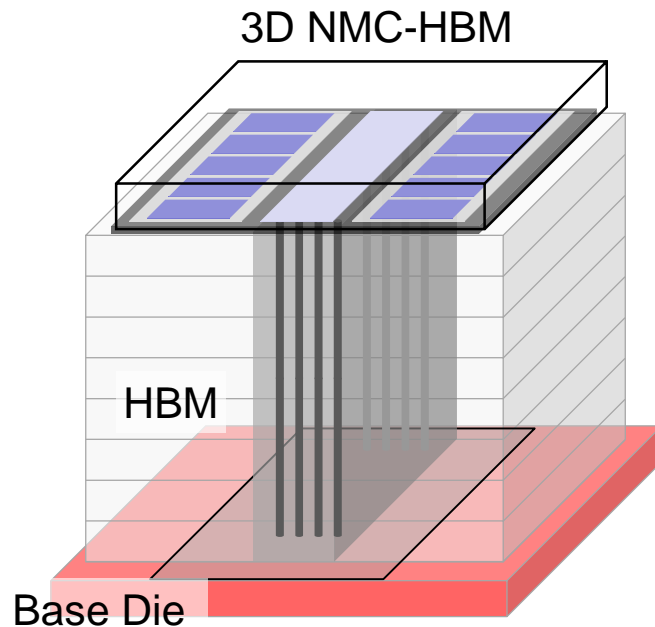


$$\text{Simultaneous Switching Noise (SSN)} \downarrow = \mathbf{Z(f)} \downarrow \times \text{SSC}$$

< SSN Reduction through Decoupling Capacitor Placement >

- In 3D NMC-HBM Gen. 5 architectures, power delivery to the NMC processor die is often unstable due to the resistance and inductance along the power supply path.
- This instability becomes particularly severe when the processor die rapidly switches and demands high current, leading to significant simultaneous switching noise (SSN) issues.
- Therefore, it is essential to strategically place sufficient on-chip decoupling capacitors, such as MIM, MOS, or cell capacitors, to effectively lower the power delivery impedance.

# Importance of Decoupling Capacitors for Reliable HBM Power Supply



## < DRAM Process-based Decoupling Capacitor Die for Reliable HBM Power Supply >

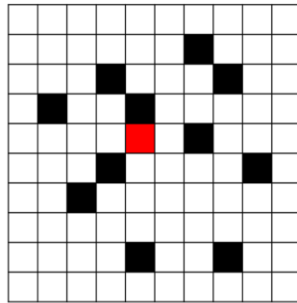
- Decoupling capacitors (Decaps) are essential for ensuring reliable power delivery in 3D NMC-HBM.
- Placing decaps close to the die shortens the power delivery path, minimizing voltage fluctuations caused by power grid resistance and inductance. This is particularly effective in suppressing simultaneous switching noise (SSN) during fast switching and high current demand.
- Although having more Decaps is beneficial, their number is limited by spatial constraints and cost. Therefore, it is crucial to optimally place a minimal number of Decaps in the most effective locations.
  - A combinatorial optimization problem for the optimal placement of a given set of decaps.

# Necessities of AI Agent for Decap Placement Optimization [1/2]: NP-Hard Nature of Decap Placement Problem

For any given  $N \times M$  PDN,

$M$

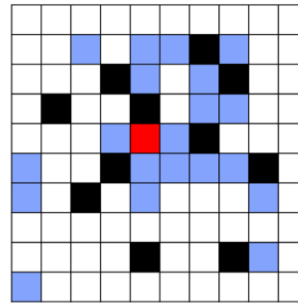
$N$



■ : Probing point

■ : Keep-Out Region

Find the best placement  
for given number of decaps



■ : Decap

- ✓ Size of PDN:  $N \times M$
- ✓ Number of Probing Point: 1
- ✓ Number of Keep-out region:  $0 \leq ko \leq 15$
- ✓ Number of decap:  $K$

- Decap placement task is to place given number of decaps for  $N \times M$  PDN with an arbitrary probing port and random keep-out region.
- Decap placement problem is formulated as an **NP-hard problem**, which has an infinite search space (i.e, infinite number of possible solutions)

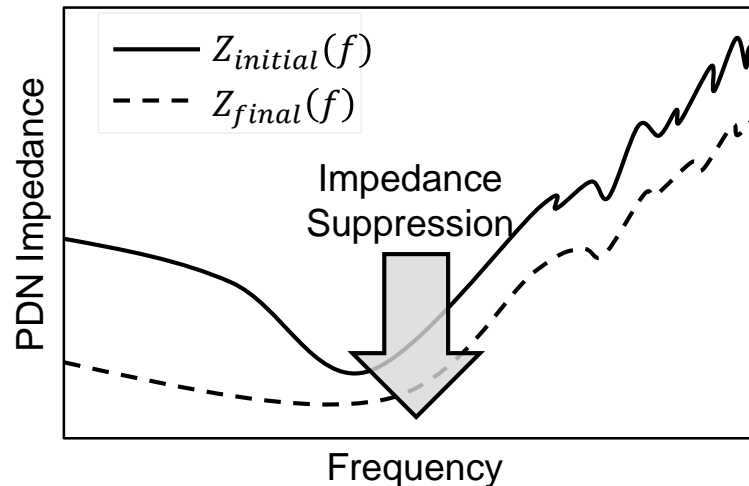
$$\text{Number of possible problems} = \underbrace{NM}_{\text{Probe}} \times \underbrace{\sum_{k=0}^{15} \binom{NM-1}{ko}}_{\text{Keep-out regions}}$$

$$\text{Solution space for each problem} = \binom{NM-1-ko}{K}$$

$$\text{Full Search Space} = \text{Number of possible problems} \times \text{solution space for each problem} \approx \infty$$

- Therefore, leveraging an **AI agent is essential to efficiently explore solution spaces far beyond what human engineers can visualize or enumerate**, enabling the discovery of novel and highly effective decap placement strategies.

# Necessities of AI Agent for Decap Placement Optimization [2/2]: High-Cost Reward Calculation through Time-Intensive Simulation



< Graphical Representation of Reward Metric for PDN Decap Placement >

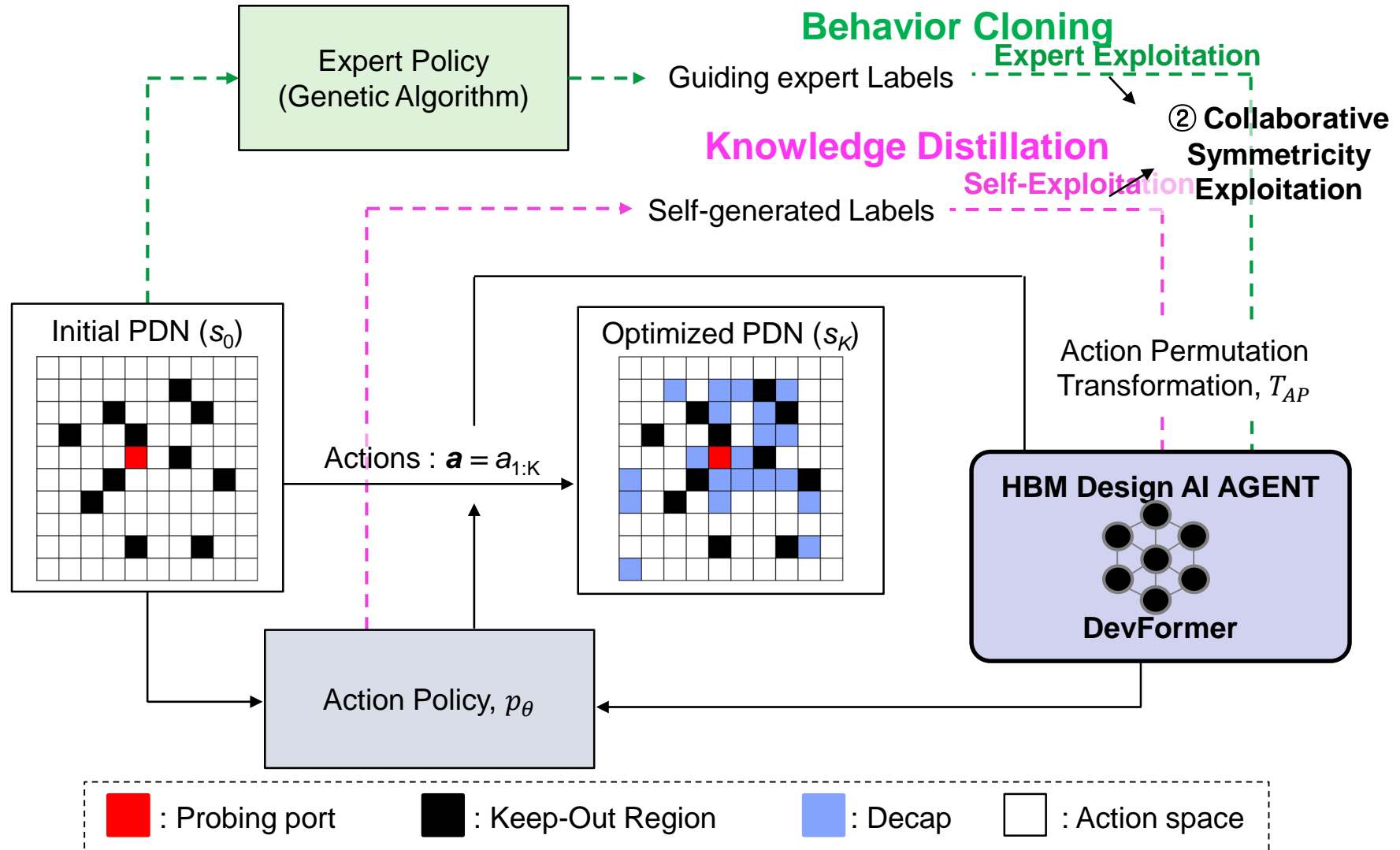
- The value of reward is a quantitative representation of impedance suppression and is computed as:

$$\text{Reward} = \sum_{f \in F} (Z_{initial}(f) - Z_{final}(f)) \cdot \frac{1 \text{ GHz}}{f}$$

where  $F$  is a set of 201 frequency points linearly distributed over the range 100MHz - 20GHz.

- Reward calculation involves
  - 1) multiple matrix-multiplication (high computational demand → time-intensive)
  - 2) frequency- and position- dependent large matrix (high memory usage)
- Therefore, **minimizing the number of optimization iterations is crucial**. This can be achieved by training an AI agent, which can efficiently parameterize and learn the underlying relationships.

# Imitation Learning of DevFormer with Symmetricity Exploitation

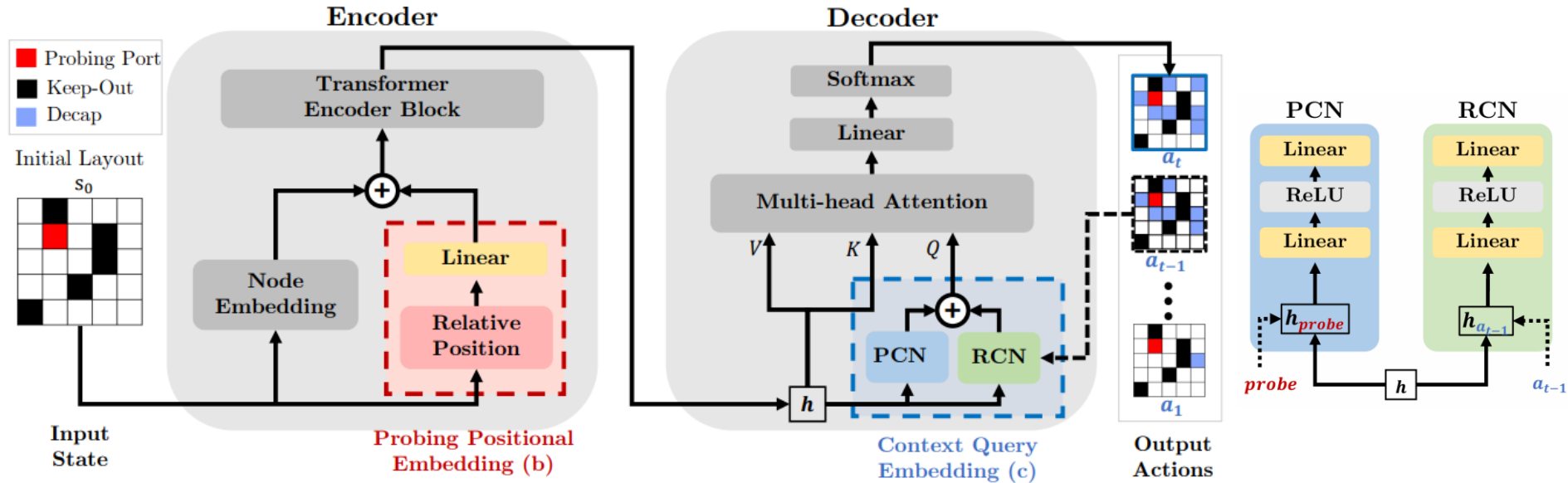


< Imitation Learning of DevFormer-Collaborative Symmetricity Exploitation (CSE) Framework >

Ref) H. Kim et al., "Collaborative Symmetricity Exploitation for Offline Learning of Hardware Design Solver," NeurIPS 2022 Offline RL Workshop. <https://openreview.net/forum?id=FR9NkGgaLw>

Ref) H. Kim et al., "DevFormer: A Symmetric Transformer for Context-Aware Device Placement," ICML 2023

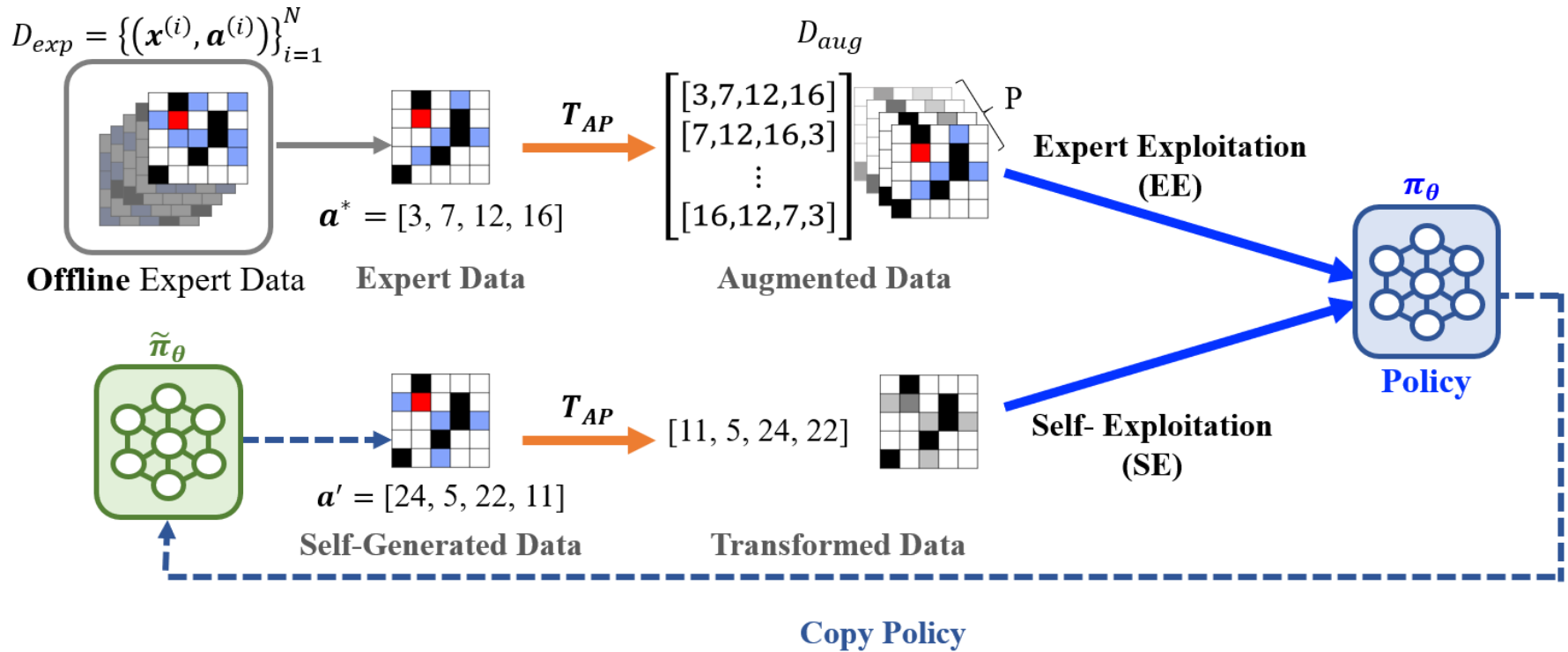
# Architecture of Device Transformer (DevFormer)



## < Overall DevFormer Architecture for Decap Placement Optimization Agent >

- Devformer is a variant of transformer, specifically designed for decap placement optimization agent:
  - Devformer is composed of an encoder block and multiple decoder blocks.
  - Instead of **positional encoding** in Transformer, DevFormer has a **node embedding and probing positional embedding**, which embeds the distances of each port relative to the probing port.
  - DevFormer has two extra context neural networks: **PCN (probing port context network)** and **RCN (recurrent context network)** to use extra embedding to capture contextual information in initial design conditions and stages of the partial solution, respectively.

# Collaborative Symmetricity Exploitation (CSE): Learning Scheme for Symmetricity Distillation to DevFormer



## < A Novel Collaborative Symmetricity Exploitation (CSE) Learning Scheme and Loss Function >

- The decoder of DevFormer outputs the decap placement actions auto-regressively while the final design is not affected by the order of placement; The DevFormer model inheritably has placement order bias.
- To improve generalization capability of DevFormer, CSE was devised to induce action-permutation symmetricity so that the model understands that orders don't matter and improve sample efficiency.



# Performance Optimality and Reusability Verification

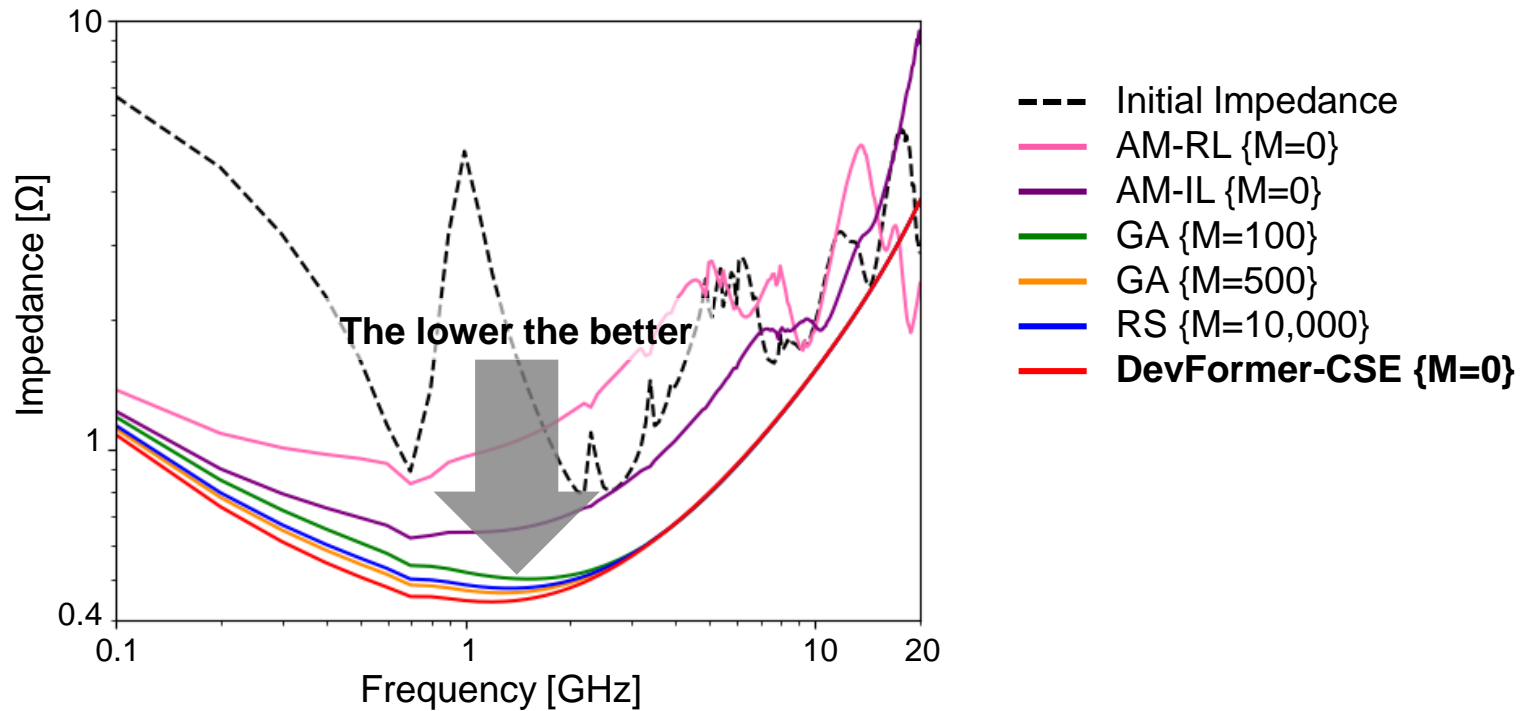
		Method	Number of Samples {M}	Average Score
<i>Online Search Heuristics</i>		GA, <i>Expert Policy</i>	100	$12.56 \pm 0.017$
		RS	10,000	$12.70 \pm 0.000$
<i>Online Test-Time Adaptation Learning</i>		Pointer-PG	10,000	$9.66 \pm 0.206$
		AM-PG	10,000	$9.63 \pm 0.587$
		CNN-DQN	10,000	$9.79 \pm 0.267$
		CNN-DDQN	10,000	$9.63 \pm 0.150$
<i>Online Pretrained</i>	RL	Pointer-RL	<b>Zero-shot</b>	$9.59 \pm 0.232$
		AM-RL	<b>Zero-shot</b>	$9.56 \pm 0.471$
<i>Offline Pretrained</i>	IL	Pointer-IL {N=2,000}	<b>Zero-shot</b>	$10.49 \pm 0.119$
		AM-IL {N=2,000}	<b>Zero-shot</b>	$11.74 \pm 0.075$
	<b>Novel Neural Net</b>	<b>DevFormer-CSE {N=1,000}</b>	<b>Zero-shot</b>	<b><math>12.88 \pm 0.003</math></b>

\*N: number of expert data used for training / M: number of samples used for a single problem inference

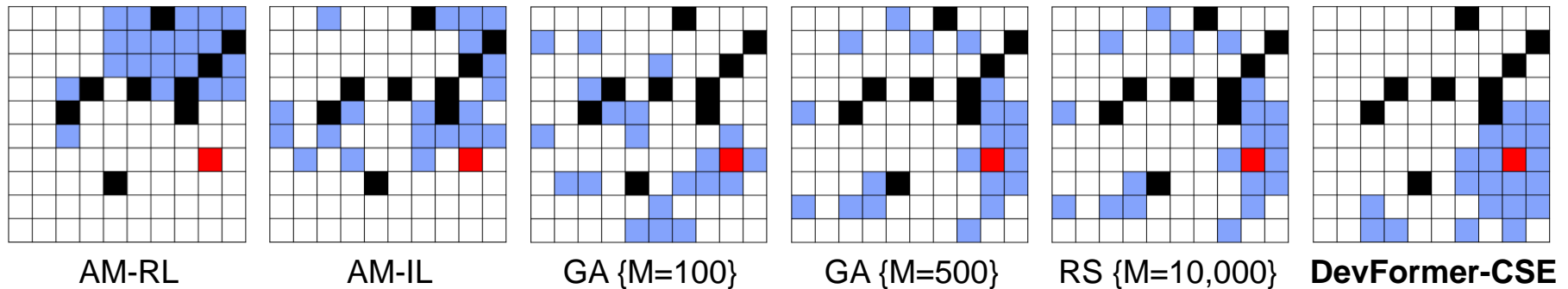
\* Average score of 10 unseen problems are reported.

< Performance Evaluation of the DevFormer-CSE Framework and Baseline Methods >

# Verification in terms of PDN Impedance Suppression



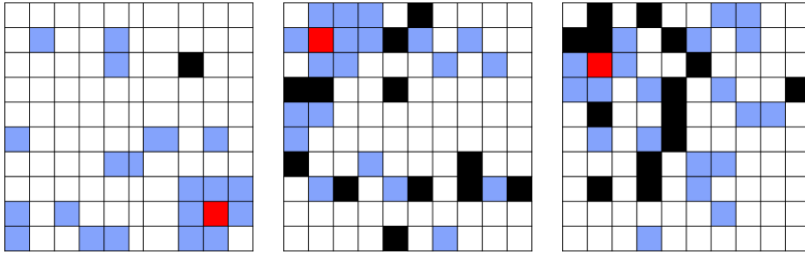
< Resulting Impedance Suppressed by Decap Placement by Each Method for Test Case 1 >



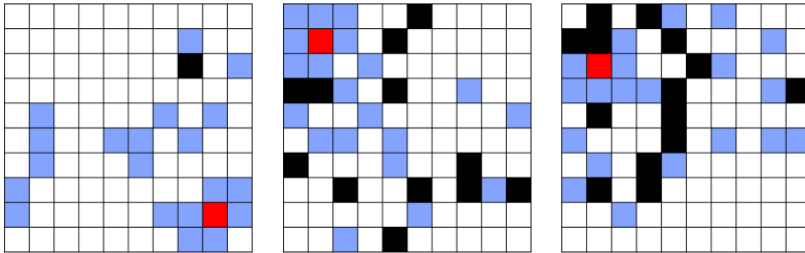
< Corresponding Decap Placement Solutions by Each Method for Test Case 1 >

# Solution Tendency Analysis

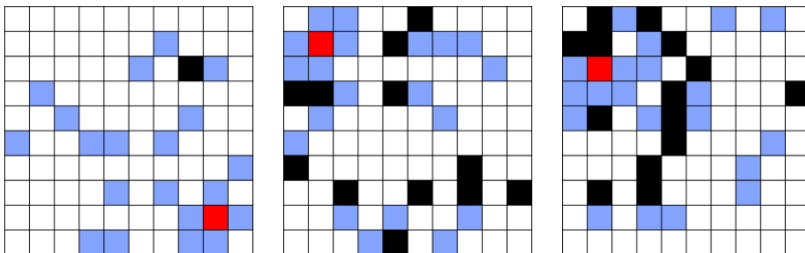
## Search-Heuristic Methods



GA {M=100}

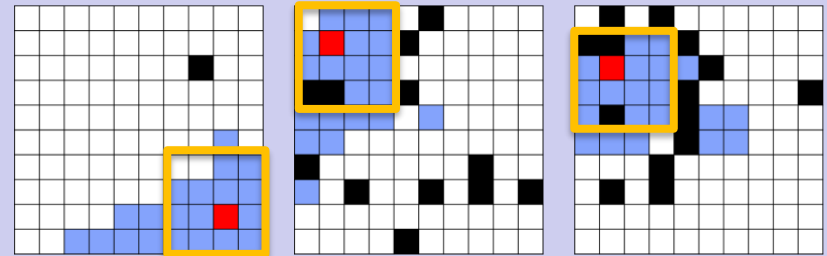


GA {M=500}

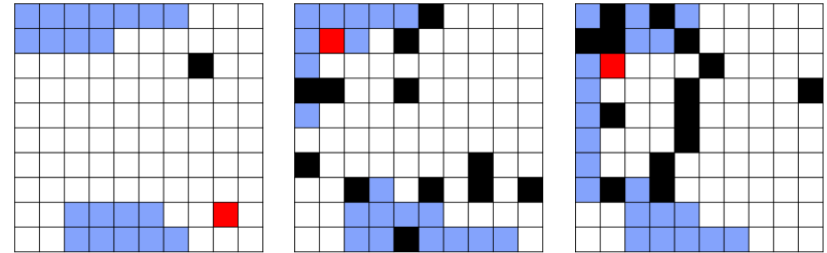


RS {M=10,000}

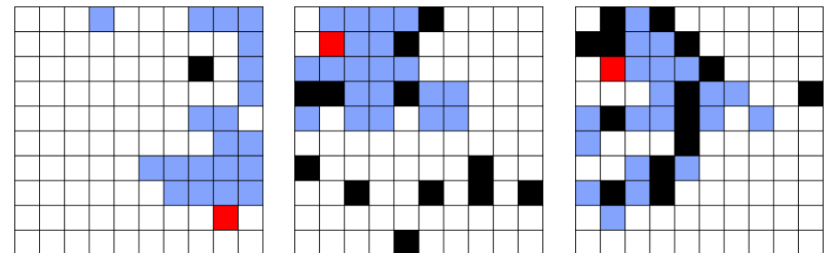
## Pretrained Methods



DevFormer-CSE {M=0}



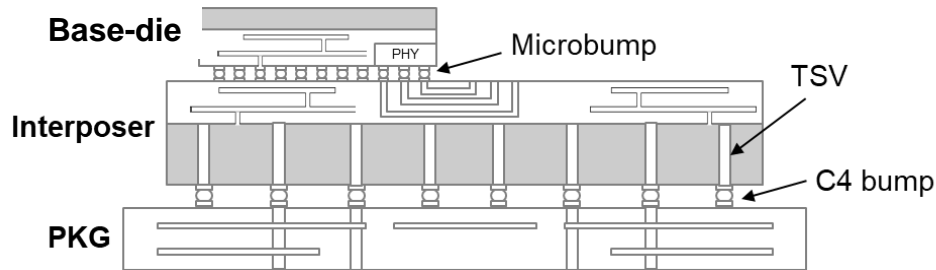
AM-RL {M=0}



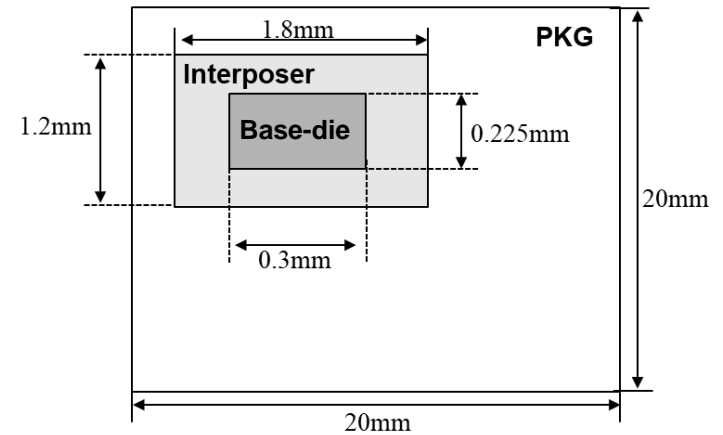
AM-IL {M=0}

- Solutions by the search-heuristic methods do not show clear tendency while pretrained methods produce clustered solutions. DevFormer-CSE tends to place decaps near the probing port.

# Application of DevFormer to Power Distribution Network of HBM



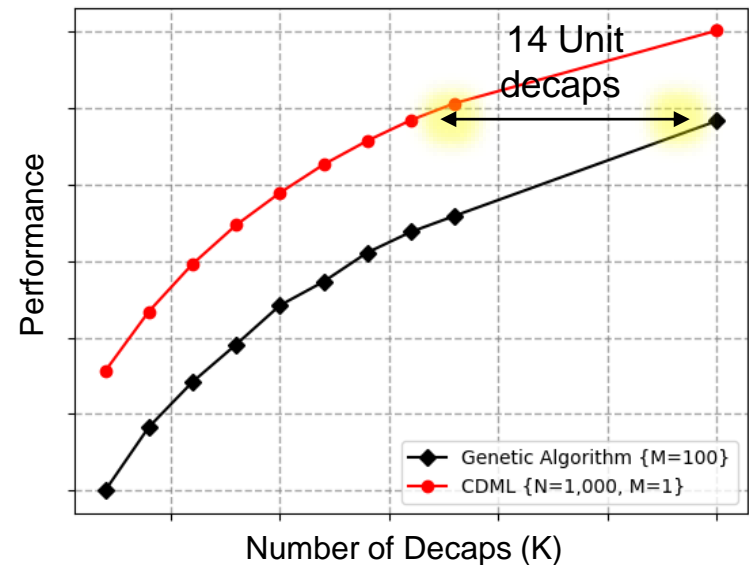
< Side-view of the Target Hierarchical PDN >



< Top-view of the Target Hierarchical PDN >

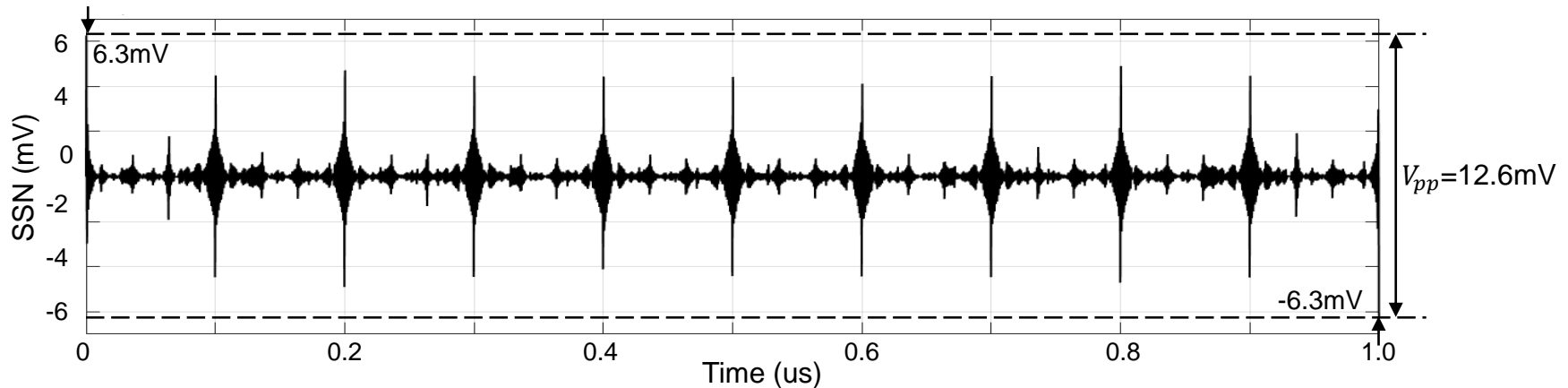
Method (K=20)	Average Score
GA {M=100}	26.44
GA {M=200}	26.46
GA {M=500}	26.51
RS {M=100}	26.38
RS {M=500}	26.40
RS {M=500}	26.42
RS {M=1,000}	26.43
<b>DevFormer-CSE {N = 1,000, M=1}</b>	<b>26.59</b>

< Performance Verification on HBM PDN >

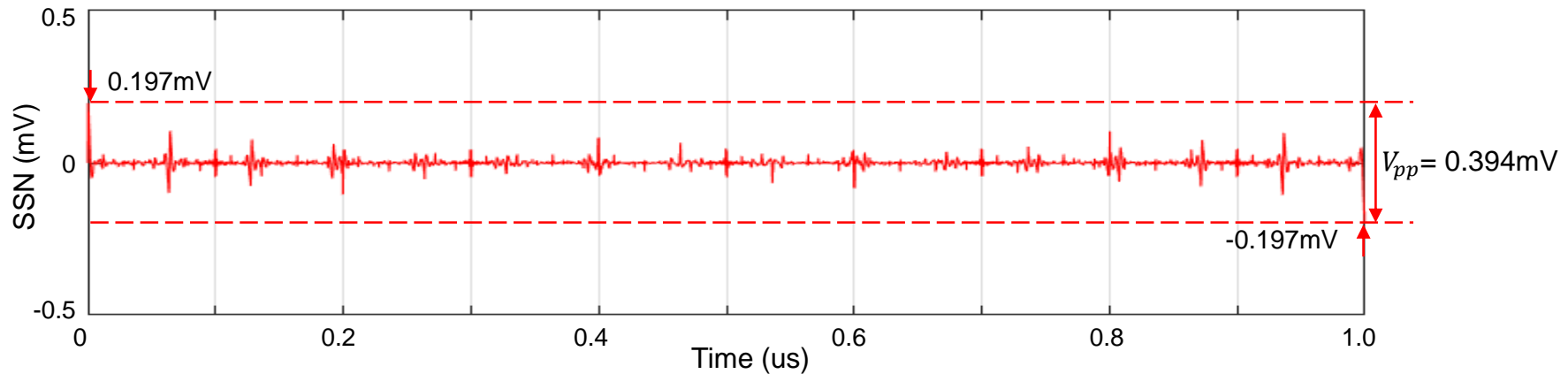


< Decap Scalability Verification on HBM PDN >

# Time-Domain SSN Analysis at a Logic Block on HBM



< Initial Time-Domain SSN at a Logic Block on HBM PDN >



< Resulting Time-Domain SSN by DevFormer-CSE at a Logic Block on HBM PDN >

- The initial peak-to-peak SSN was 12.6mV, which was reduced to 0.394mV (**96.8% reduction**) after 20 0.1nF decap placement on HBM PDN by DevFormer-CSE.
- This verifies that the decap placement by DevFormer-CSE significantly reduces SSN at a logic block on HBM.

# Thank You!

## HBM

# Reinforcement Learning-based Decap Placement Optimization considering Diverse I/O Channel Interfaces in Custom Base Die of HBM5 Memory Pooling Architecture

**Junho Park**

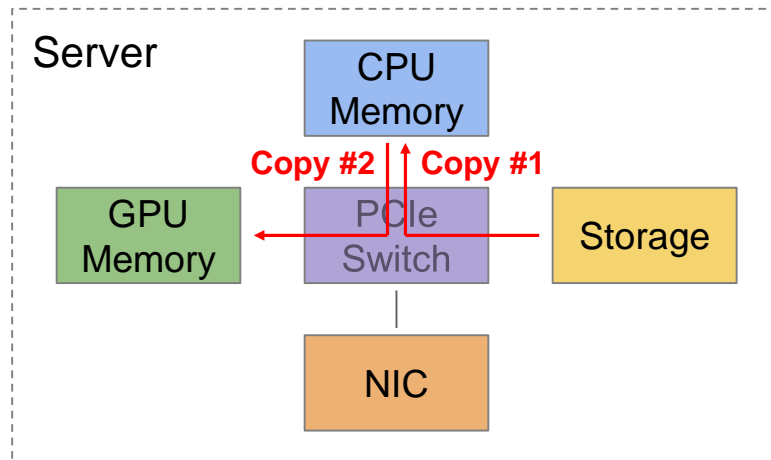
Advising Professor: Prof. Jounggho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering, KAIST

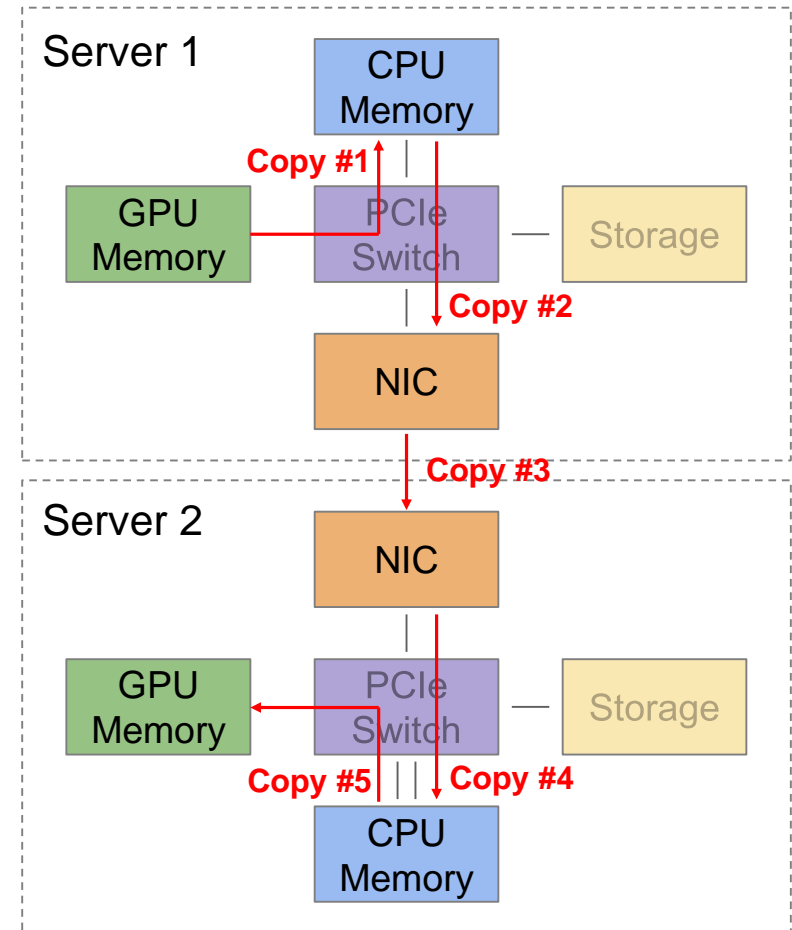
June 11<sup>th</sup> 2025

# Limitation of Data Movement in Conventional Computing Systems

## Intra-server Data Copy



## Inter-server Data Copy

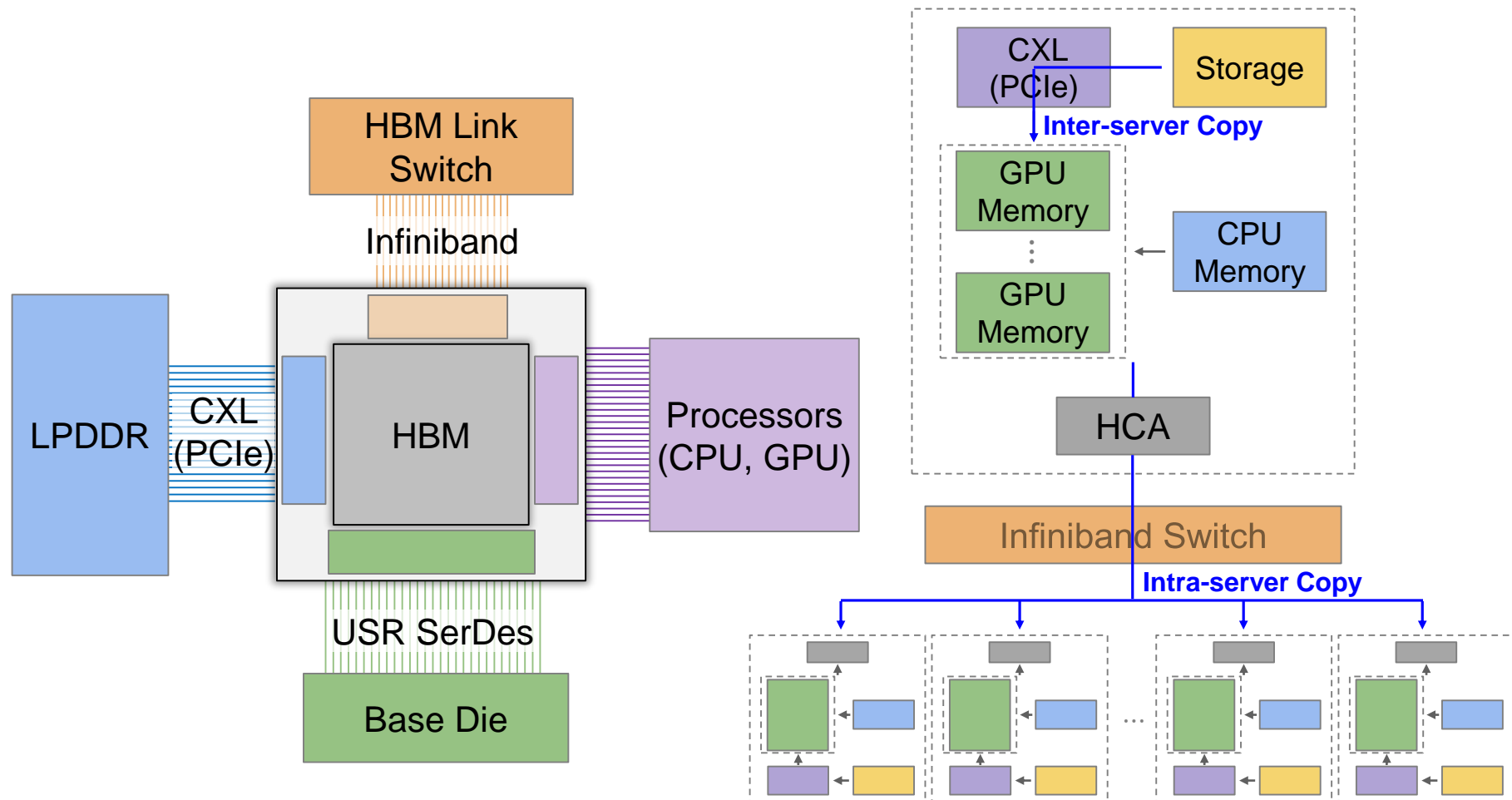


< Excessive data movement overhead makes AI workloads memory-bound >

- Latency caused by data copies across host memory leads to memory-bound issues during AI training and inference.



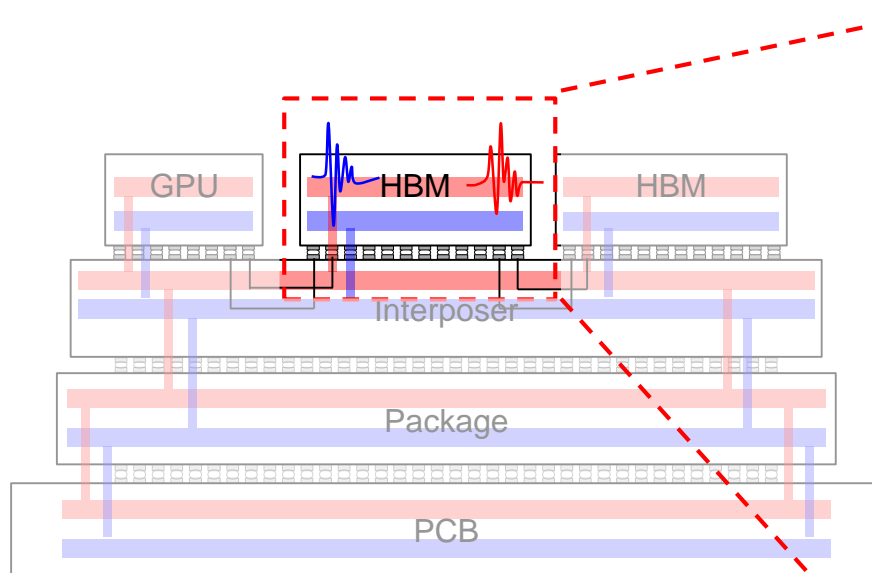
# Overview of HBM-centric Memory Pooling Computing Architecture



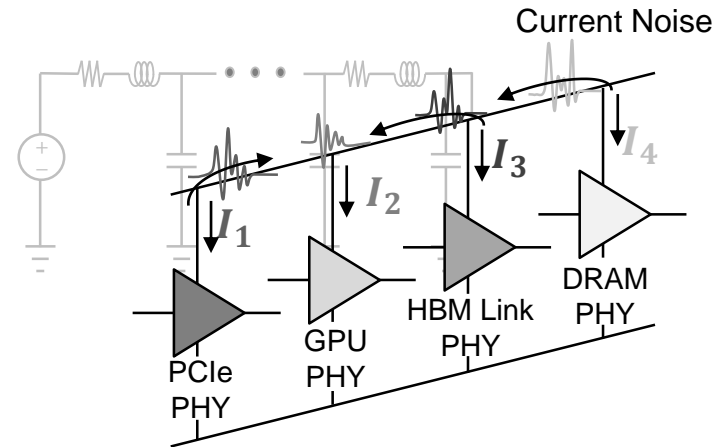
< Conceptual design of HBM-centric Architecture >

- This system enables a memory pooling architecture where all processors can access the entire memory.

# Power Integrity Challenges in an HBM-Centric Customized Base Die with Diverse I/O Interfaces



## Diverse I/O Interface Modeling



$$V_{iSSN} = \mathbf{I}_i \times Z_{i\ self} + \sum_{i \neq j} \mathbf{I}_j \times Z_{i,j\ transfer}$$

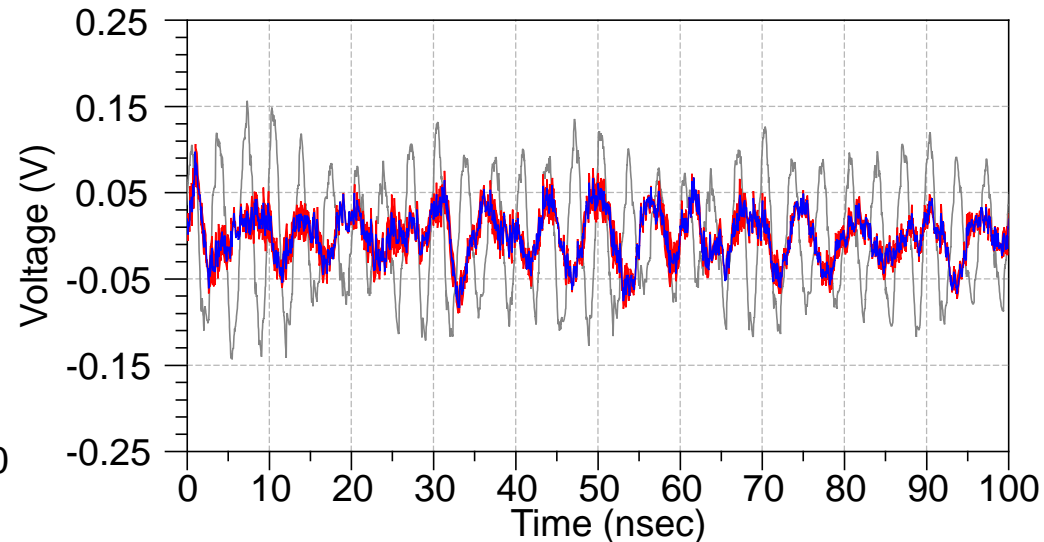
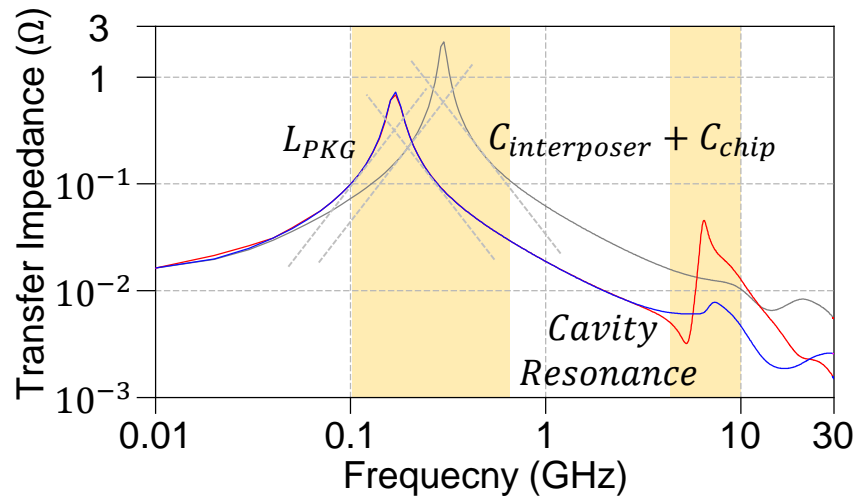
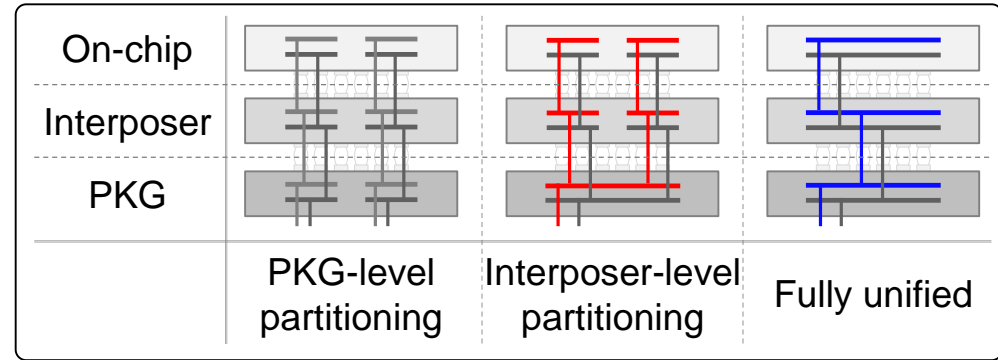
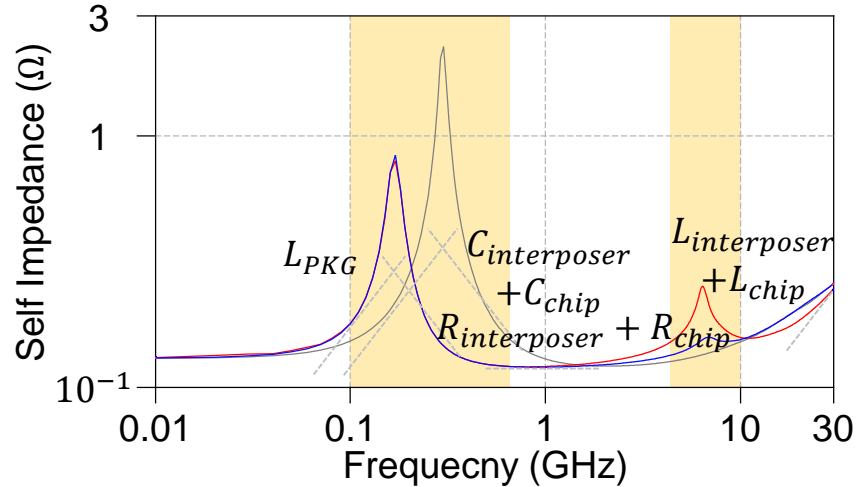
< Comparison of PDN design with considering  $I_{SSC}$  >

- Simultaneous switching current ( $I_{SSC}$ ) varies depending on the number of transistors and switching frequency.

$$I_{SSC}(f) = N_{transistor} \cdot C_{load} \cdot V_{DD}(f) \cdot f_{switching}$$

- The conventional PDN design overlooks various  $I_{SSC}$  types in a single power domain, limiting its suitability for optimizing the proposed customized HBM's PDN.

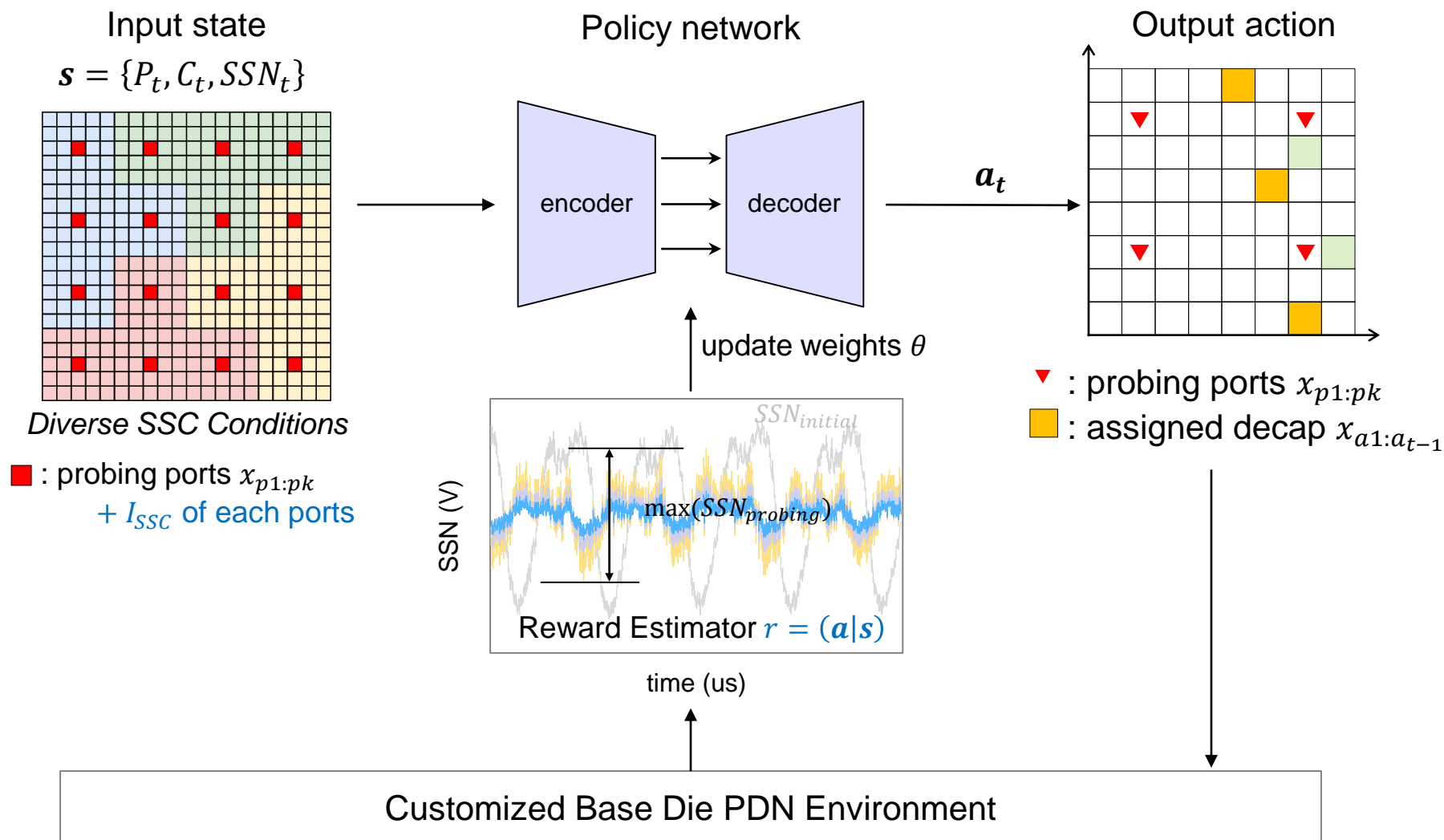
# Verification of Power Integrity Issue for Expanded VDDQ PDN with heterogeneous I/O interfaces



< Comparison of impedances and SSNs among various PDN designs >

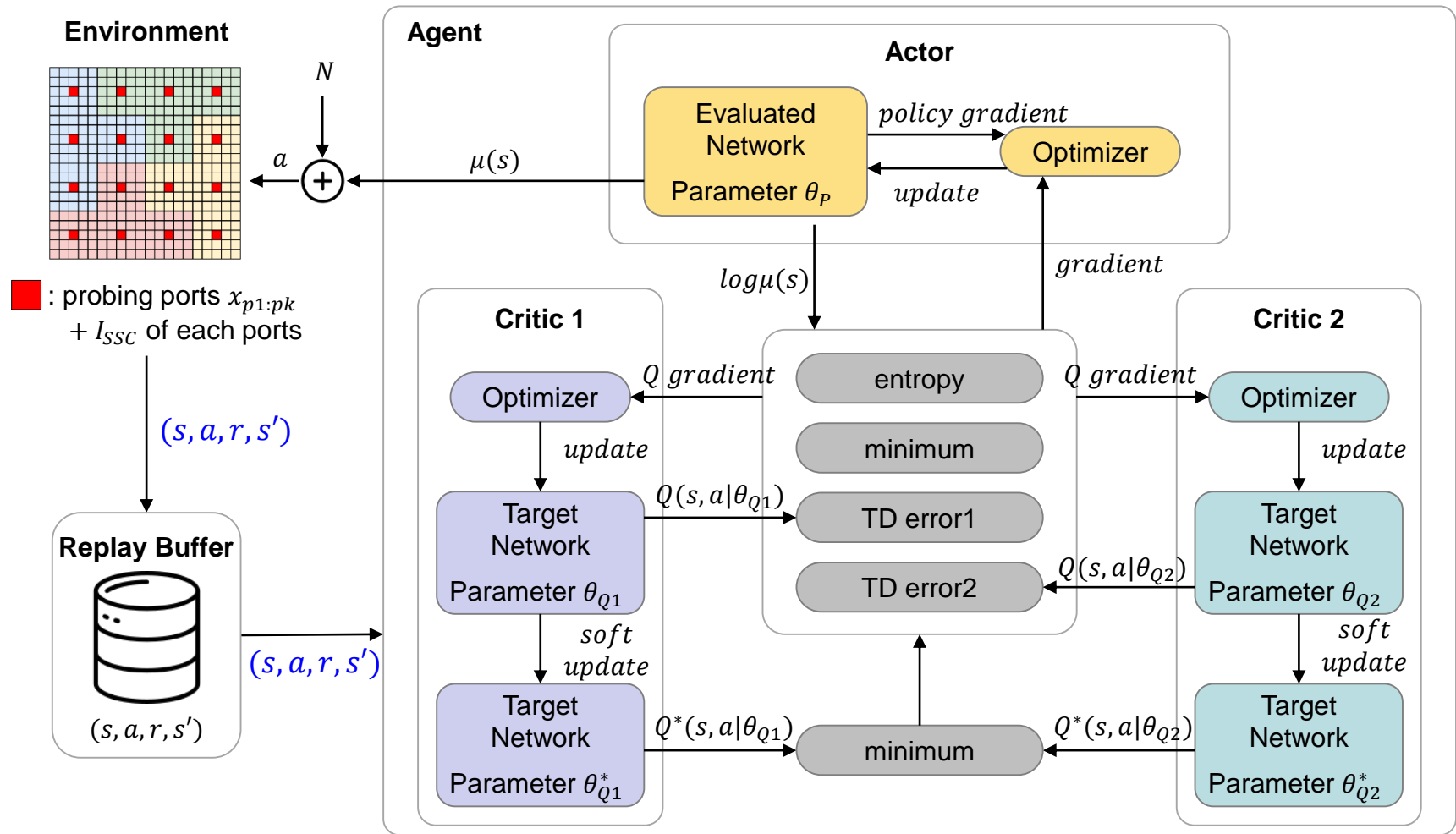
- The impedance varies depending on the PDN design methodology, which consequently results in different levels of SSN.

# Proposal of Reinforcement Learning-based Decap Placement Optimization considering Diverse SSC



< Overall reinforcement learning method for PDN optimization >

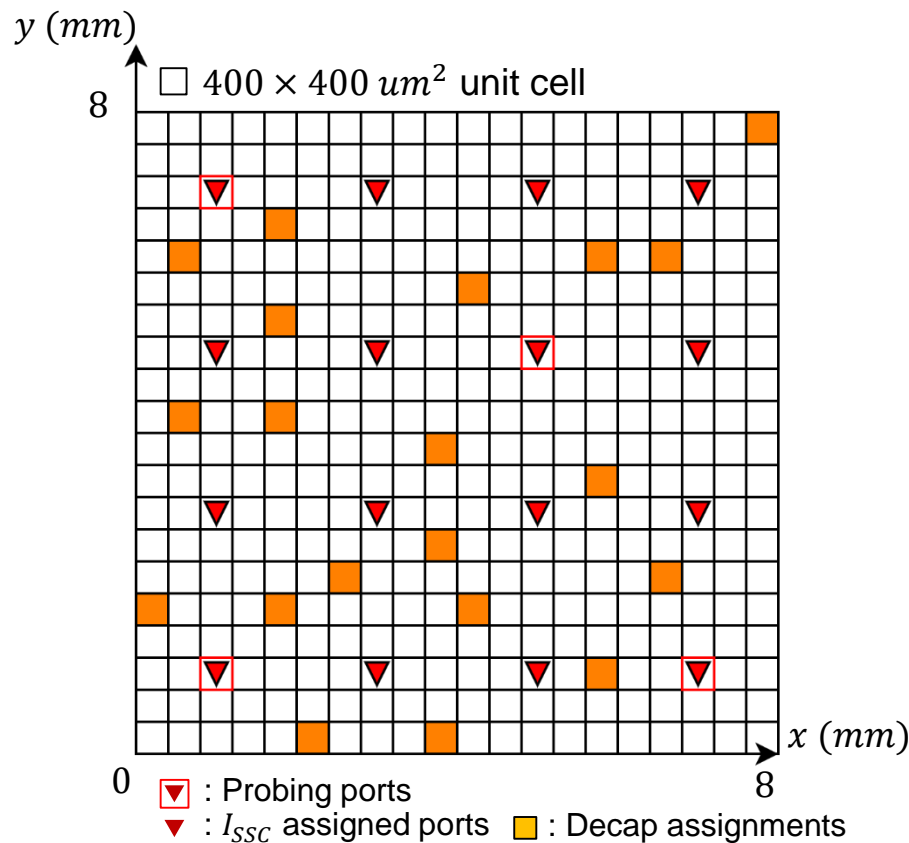
# Soft Actor-Critic (SAC) Algorithm-based Diverse SSC considered Decap Placement Design Agent Framework



< Overall architecture of PDN optimization with SAC algorithm >

- The state incorporates  $I_{SSC}$  variations at each probing port,

# Markov Decision Process (MDP) Setting: State



## State

$$S_t = \{P_t, C_t, SSN_t\}$$

$P_t \in \mathbb{R}^{N \times 3}$  : Port state matrix for N ports

- $P_{t,i,0} = 1$  if port  $i$  is a probing port,
- $P_{t,i,1} = 1$  if port  $i$  has a decap placed
- $P_{t,i,2} = 1$  if port  $i$  is an on-chip port

$C_t \in \mathbb{Z}^M$  :  $I_{SSC}$  type assignment

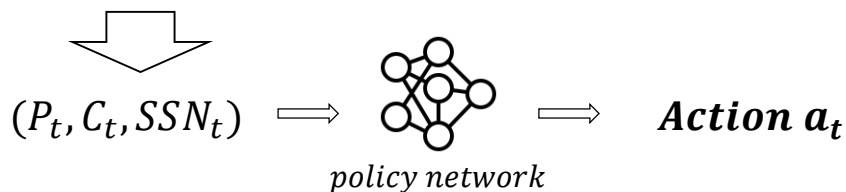
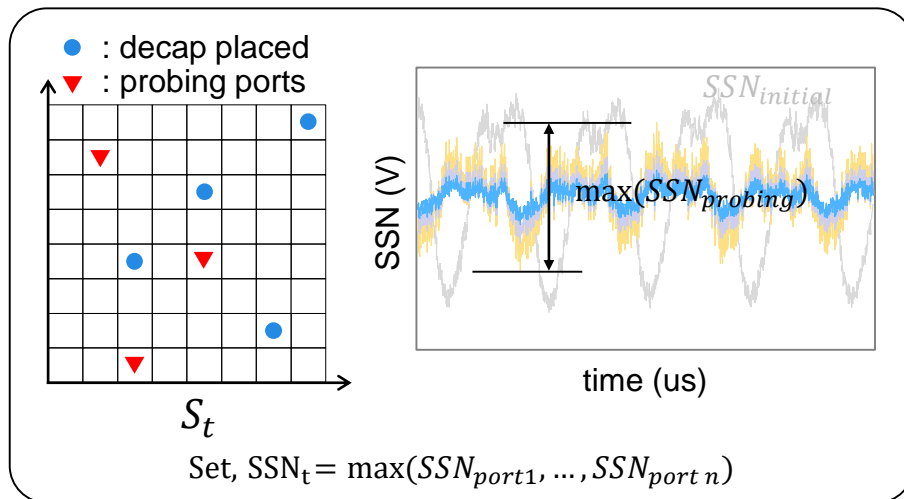
- One of the 4 types of  $I_{SSC}$

$SSN_t \in \mathbb{R}$  : Current  $SSN$  value

- Current state's calculated  $SSN$

< Coordinate based state and action representation >

# Markov Decision Process (MDP) Setting: Action



< Feedback-driven decap placement using latest SSN information >

- Each action decision reflects the current state including previously placed decaps and current SSN value.
- Action sequence terminates when the target SSN is satisfied or maximum decap count is reached.

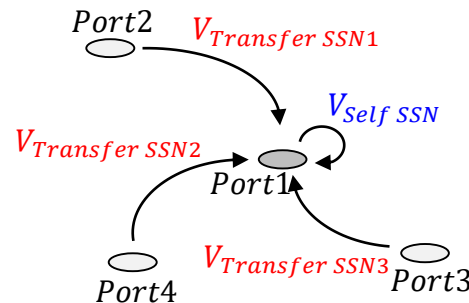
Action
$a_t \in \{0, 1, \dots, N - 1\} \setminus (\mathcal{P} \cup \mathcal{D}_t)$
<b>Action:</b> Port index to place a decap <ul style="list-style-type: none"> <li><math>N</math> : Total number of ports</li> <li><math>\mathcal{P}</math> : Set of porbing port indices</li> <li><math>\mathcal{D}_t</math> : Set of ports that already have decaps</li> </ul>

# Markov Decision Process (MDP) Setting: Reward

## Reward

$$Reward = \frac{SSN_{t-1} - SSN_t}{SSN_{t-1}} - \alpha \quad \alpha: Decap penalty$$

- Justifying simultaneous switching noise (SSN)



$$SSN = \max_{p_i \in \text{observation ports}} \left[ IFFT_{peak\ to\ peak} \left[ \sum_{p_j \in \text{all probing ports}} \left( I_{SSC@p_j}(f) \times Z_{p_j p_i}(f) \right) \right] \right]$$

- Reward can quantifies the SSN reduction at each step and penalizes decap usage to achieve maximum performance with minimal resources.
- It maintains training stability and balances reward sensitivity from early to late stages.



# Conclusion

- In conventional computing architectures, latency-induced memory-bound issues arise. To address this, an HBM-centric memory pooling architecture is required to minimize repetitive data copies.
- However, due to the use of heterogeneous I/O interfaces, the Simultaneous Switching Current (SSC) varies by location, leading to location-dependent noise issues.
- Rather than relying solely on target impedance-based decap placement, performing continuous SSN (Simultaneous Switching Noise) analysis enables optimized decap placement even in environments where multiple SSCs operate concurrently.

# Thank You!

## HBM

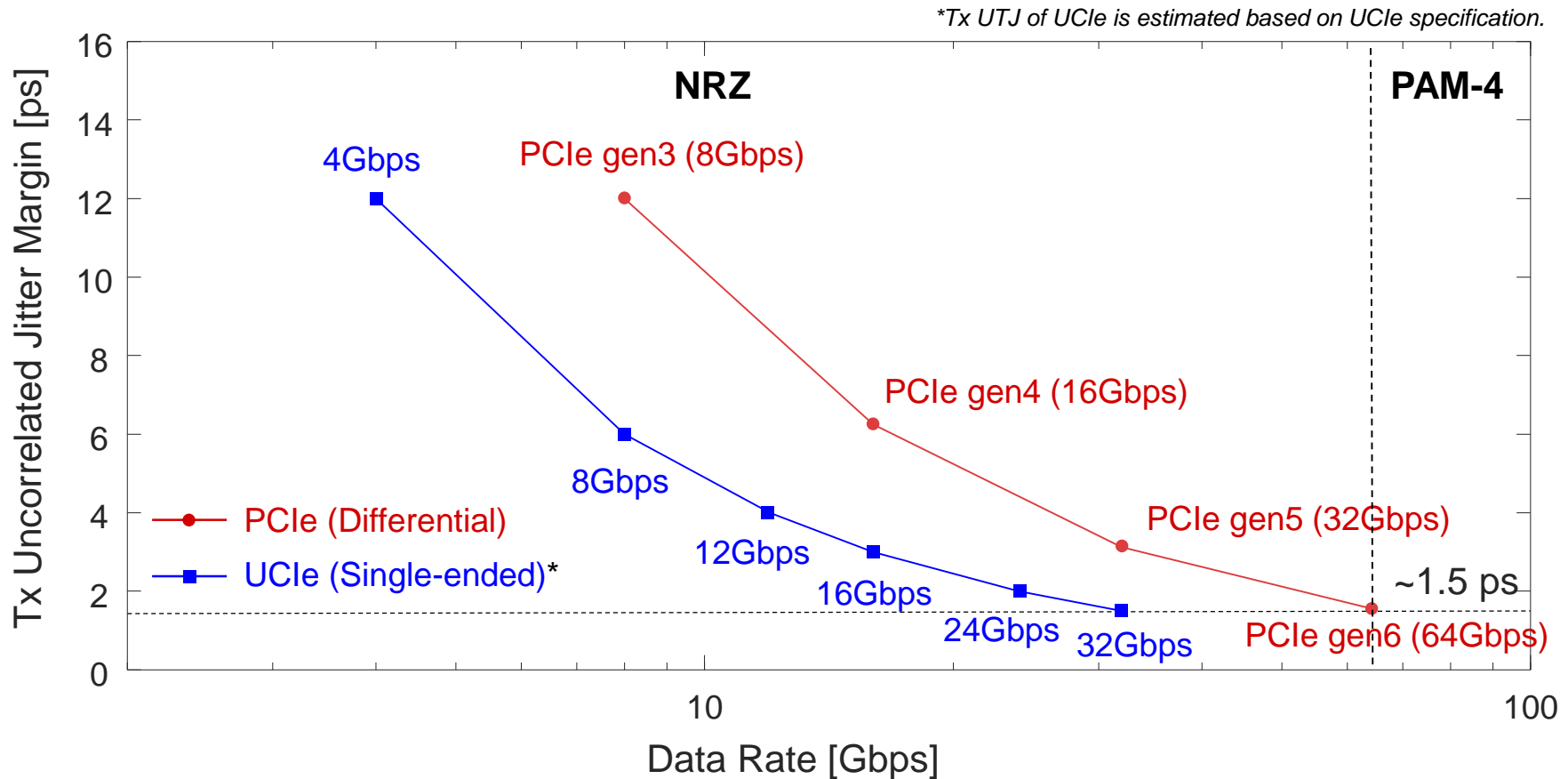
# Power Supply Noise Induced Jitter (PSIJ) based HBM5 I/O Interface Optimization using Reinforcement-Learning

Taein Shin

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

# Tighter Jitter Margin : Limitation of the Data Rate in AI Semiconductor and Super-Computer System

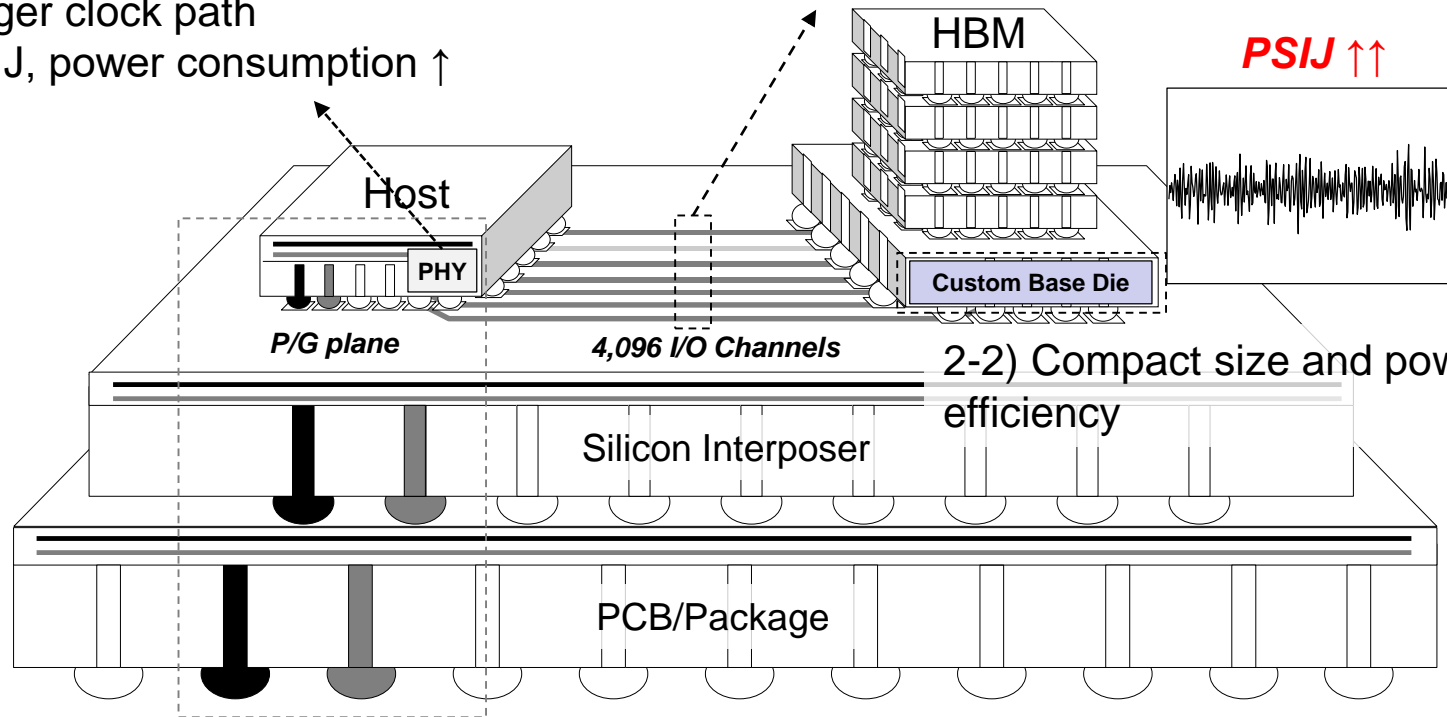


- As the data rate increased, the jitter margins of both PCIe and UCle became tighter.
- The most recent data rate is 32Gbps and 64Gbps, and both have the same Nyquist frequency of 16 GHz, and a similar Tx jitter margin ~1.5 ps.

# Increased Power Supply Noise Induced Jitter (PSIJ) Proportion of Total Jitter in HBM5

1-1) Chip process shrinks w/ high speed  
→ longer clock path  
→ PSIJ, power consumption ↑

2-1) Huge transient current (4,096 I/Os)



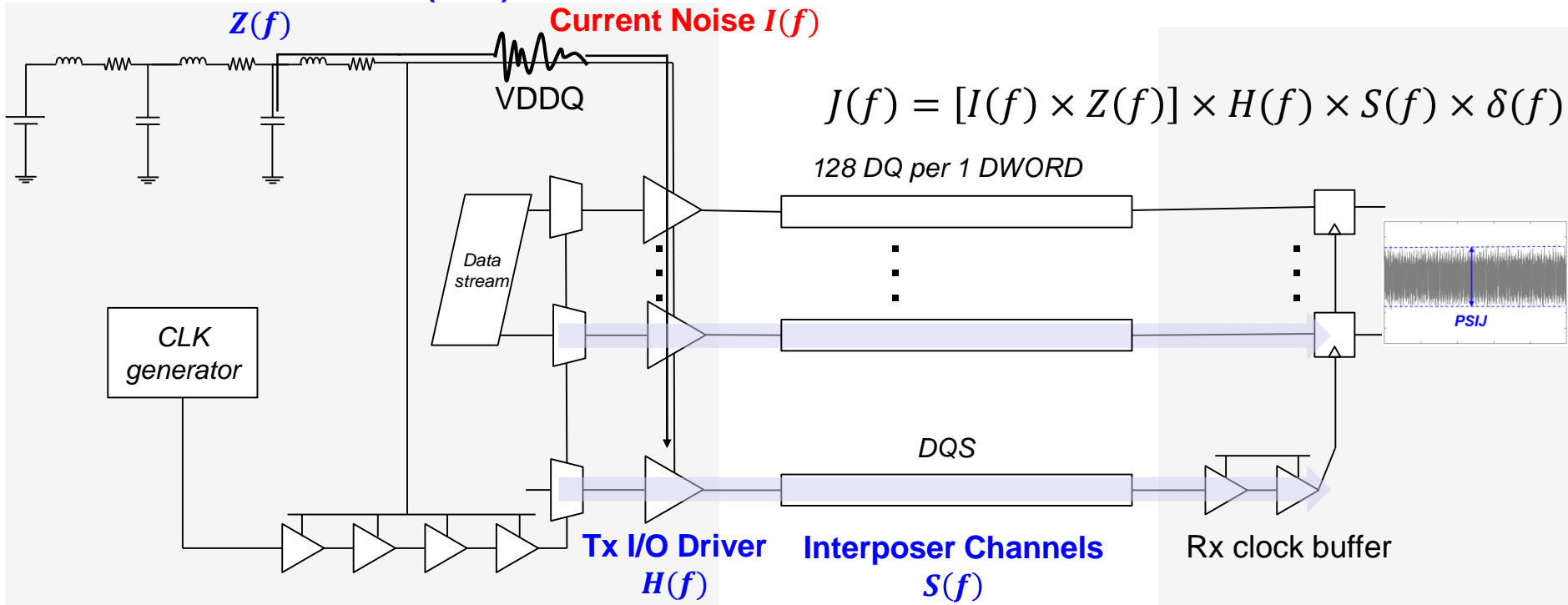
2-2) Compact size and power efficiency

1-2) No scaling down of package technology

1. Power domain is not scaled down with the increasing data rate.
2. HBM I/O interface is vulnerable to power supply noise problem than other memory or chip interconnect system.

# Target : HBM5 I/O Interface for System-Level PSIJ Reduction

## Power Distribution Network (PDN)

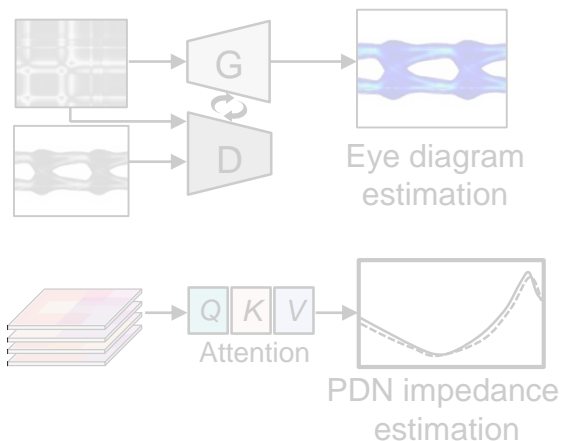


- To satisfy the target PSIJ, design parameters are assigned by optimally utilizing resources for each I/O domain (PDN, Tx, Channel) based on the given random current noise.

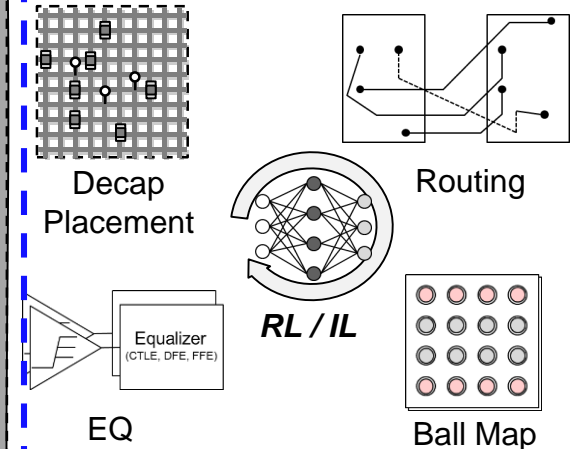
	PSIJ Factor	Resource
<b>PDN</b>	PDN Impedance : $Z(f)$	De-cap area
<b>Tx I/O Driver</b>	Jitter sensitivity : $H(f)$	Circuit power
<b>Channel</b>	Jitter amplification : $S(f)$	Routing area

# Proposal of PSIJ based Optimization Agent for HBM5

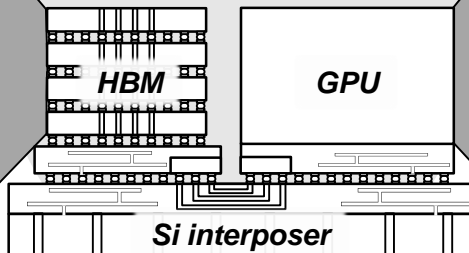
## Simulation Agent



## Optimization Agent



## HBM Design AI Agent

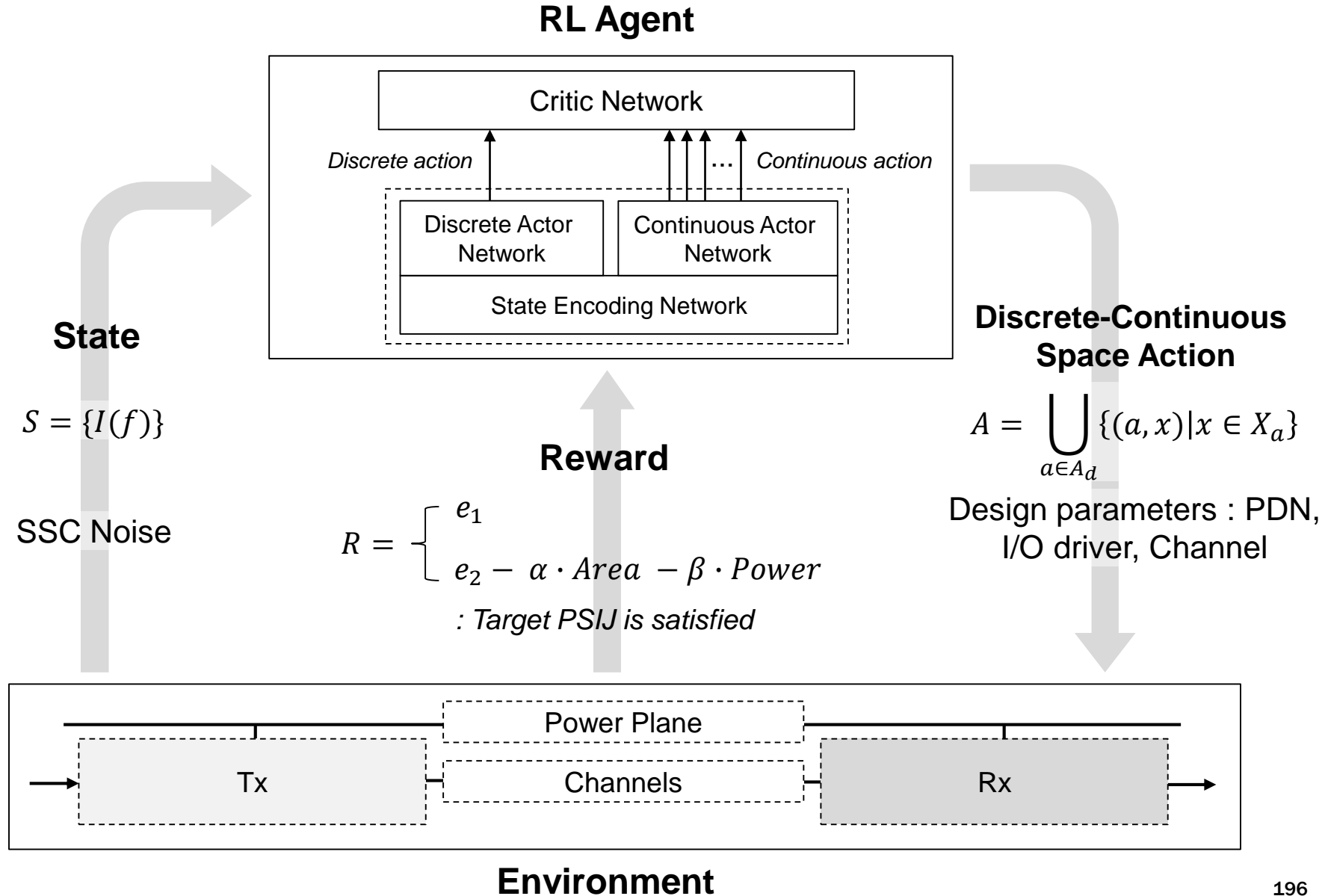


## Human interactive Agent



< AI agent for HBM Design in TERALab >

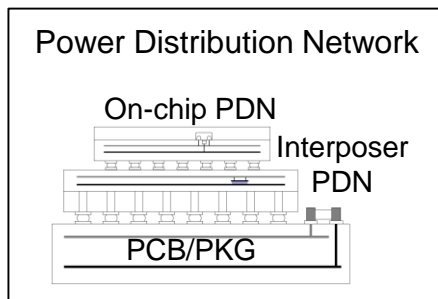
# Proposal of PSIJ based HBM I/O Interface Optimization Method using Discrete-Continuous Space Reinforcement Learning



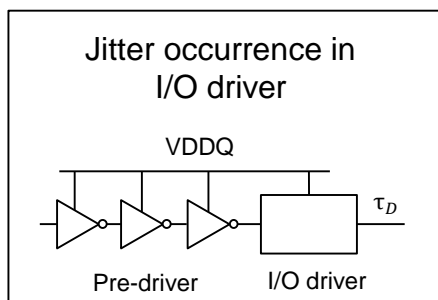


# Optimization Target in HBM5 I/O Interface

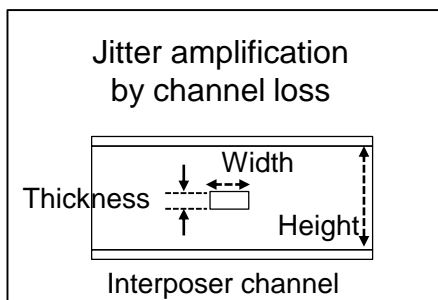
**Power**



**Tx**



**Channel**

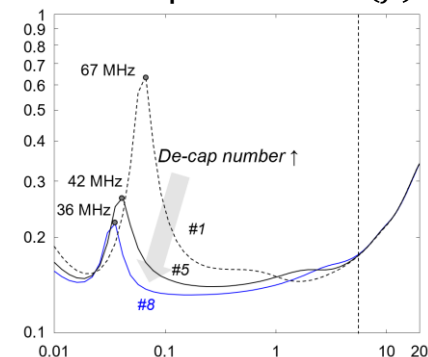


- Number of De-cap
- Position of De-cap

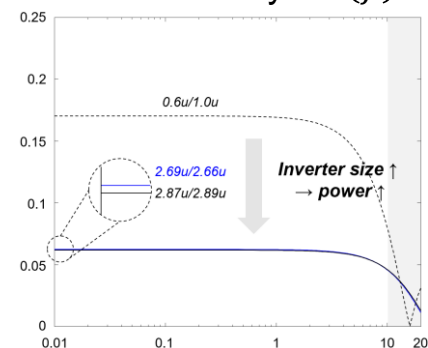
- Number of driver stage
- Transistor size

- Channel dimension

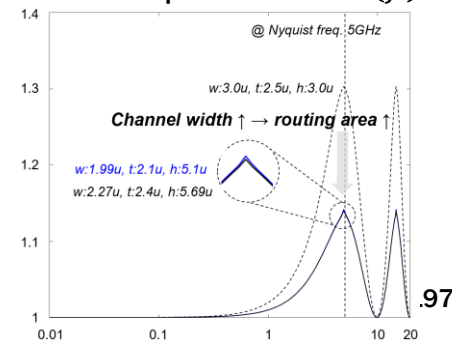
PDN impedance :  $Z(f)$



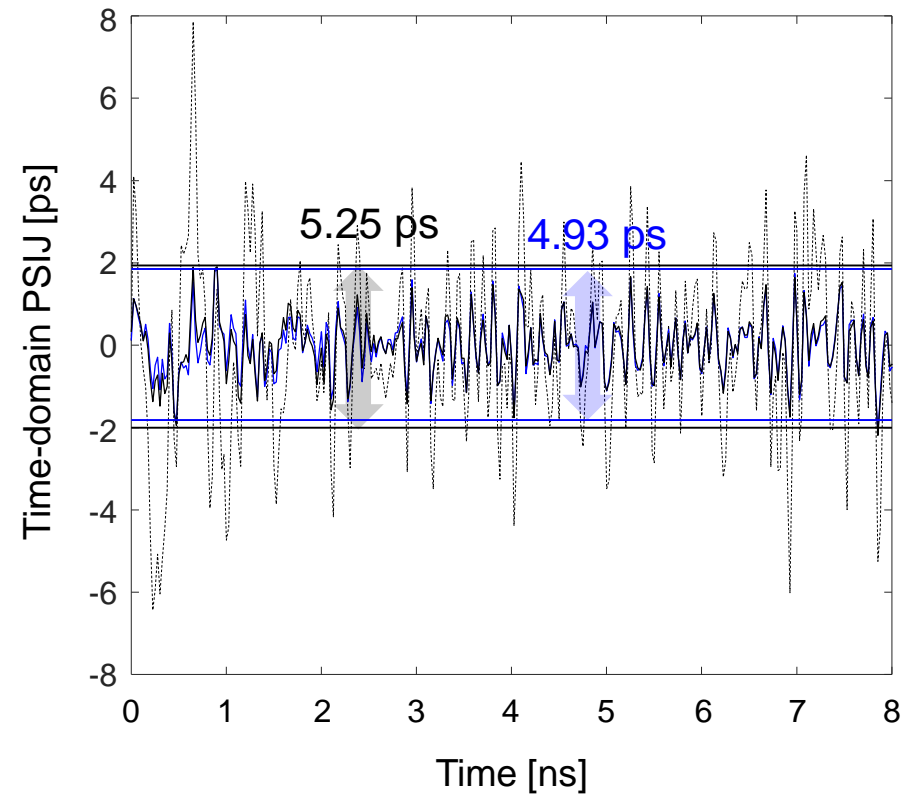
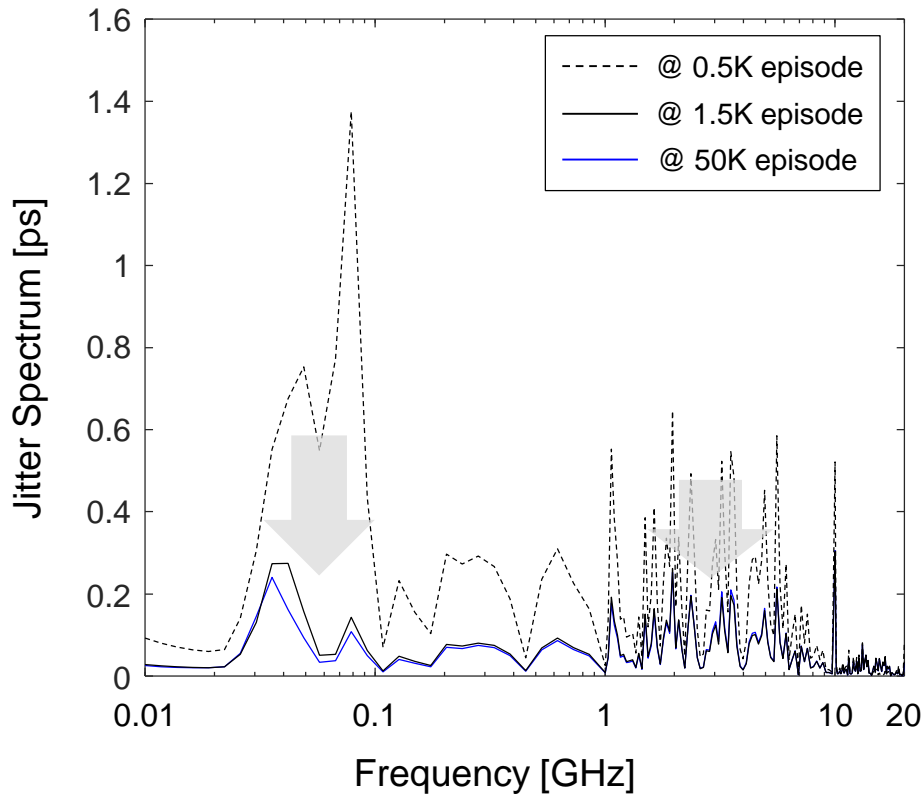
Jitter sensitivity :  $H(f)$



Jitter amplification :  $S(f)$

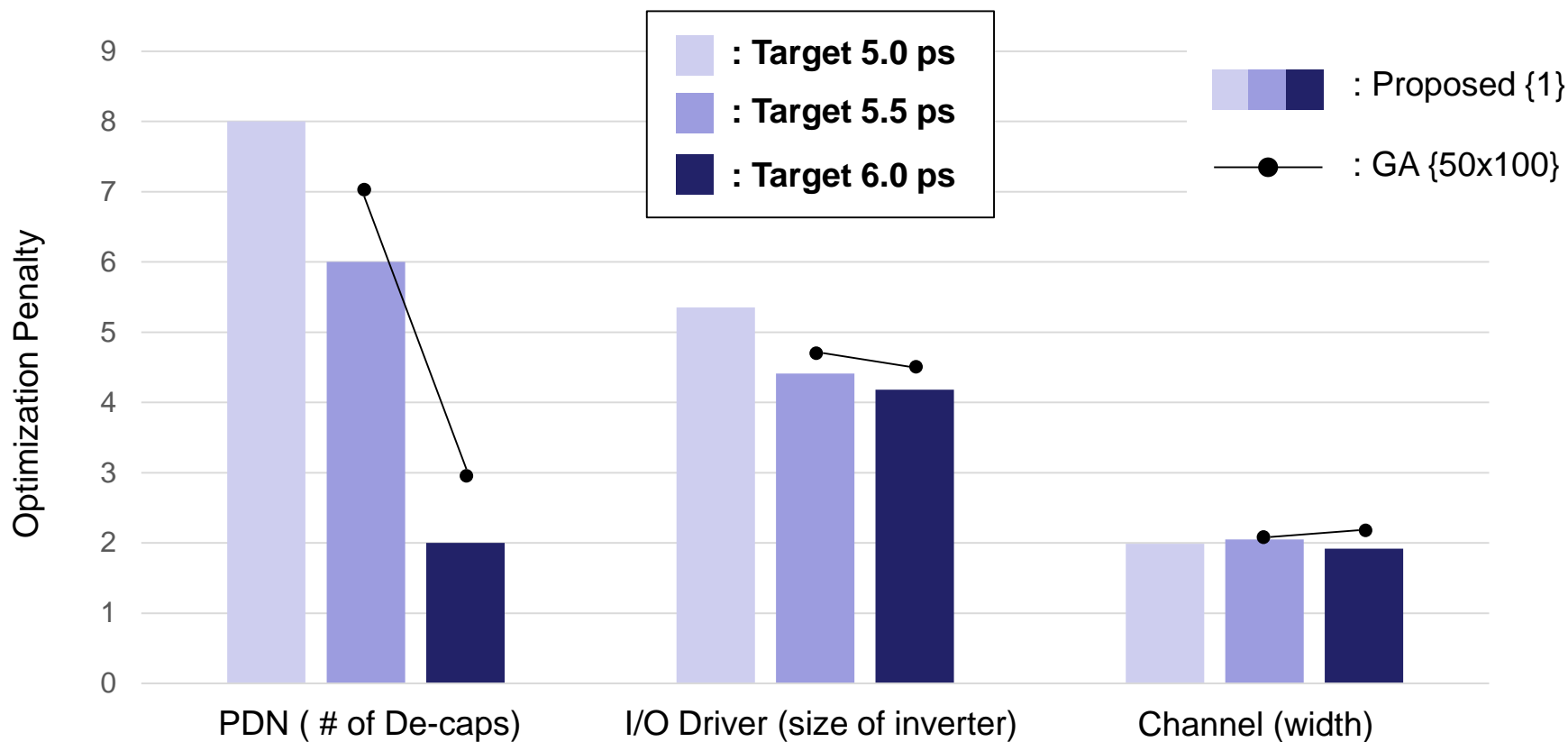


# Jitter Spectrum and Time-domain PSIJ Tendency according to the Performance Improvement



- PSIJ in all frequency ranges is gradually reduced.
- The agent efficiently reduces the PSIJ by maximizing the reward to satisfy the target 5.0 ps PSIJ at the same time.
- As a result, relatively low-frequency jitter is more reduced, and high-frequency jitter is mainly remained.

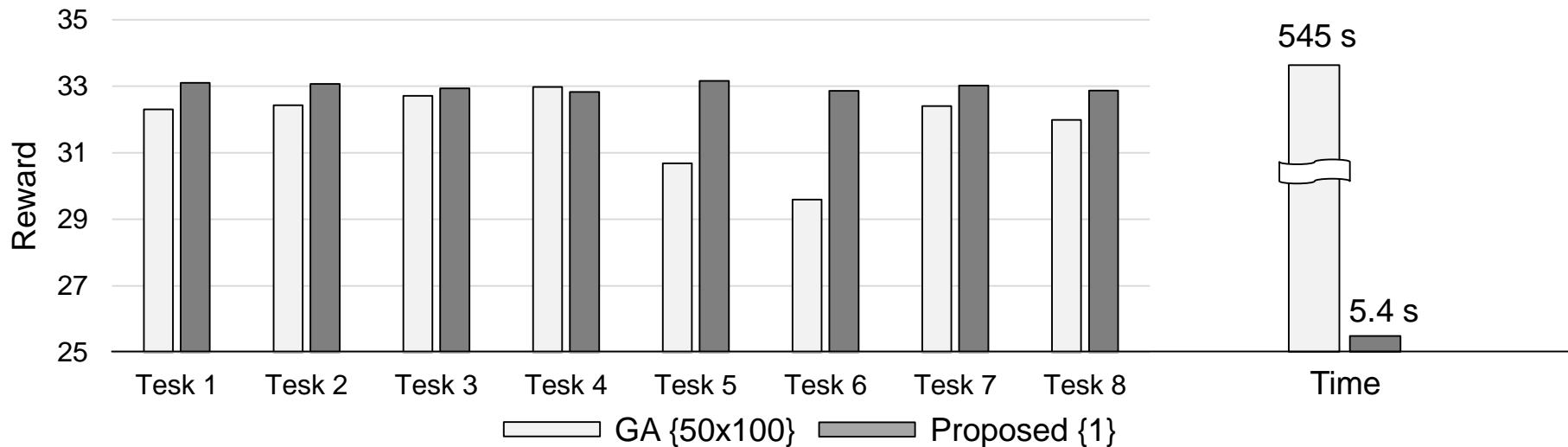
# Optimization Penalty and Trend to Satisfy the Target PSIJ



- The proposed RL-based optimization method has lower penalty than GA for target PSIJ.
- PDN is the main factor in determining whether the agent can satisfy the target PSIJ.
- I/O driver size can be reduced for lower PSIJ.

# Performance Verification on the HBM5 Current Noises

Method	Reward (Target : 5.5ps)							
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
GA {50x100}	32.31	32.43	32.71	<b>32.98</b>	30.68	29.59	32.40	31.99
Proposed {1}	<b>33.10</b>	<b>33.07</b>	<b>32.94</b>	32.83	<b>33.16</b>	<b>33.08</b>	<b>33.02</b>	<b>32.87</b>



< Performance evaluation by rewards of the proposed RL method comparing with GA >

- **Optimality:** The proposed reinforcement learning model has higher optimal rewards with faster computing time than the GA algorithm.
- **Computing time:** Conventional GA algorithm requires approximately ~100 times more computing time cost than proposed RL-based method.
- **Reusability:** verified that it was optimized through RL learned once for 8 different random tasks.

# Thank You!

## HBM

# HBM Roadmap Ver 1.7 Workshop

HBM 세대	순번	Time	Contents	Presenter
HBM6	10	13:30 ~ 13:45	Quad-Tower (QT)-HBM6 Architecture for High-Throughput and Low-Latency Inference with Signal Integrity Considerations	김태수
	11	13:45 ~ 14:00	Large-Scale Hybrid Interposer for Multi-Tower HBM6 Architecture	서해석
	12	14:00 ~ 14:15	L3 Cache Embedded (L3E) HBM6 Architecture for LLM Inference	서해석
	13	14:15 ~ 14:30	HBM6 Cluster Architecture with Crossbar Network Switch for High Throughput and Low Latency LLM Inference	윤영수
	14	14:30 ~ 14:45	HBM6-Centric Network Design under Traffic Asymmetry in Heterogeneous HBM Module based Systems	안효원
	15	14:45 ~ 14:55	Conditional Diffusion Model-based Imitation Learning for Placement and Interconnection Optimization for HBM6	김지훈
	16	14:55 ~ 15:05	Generative Adversarial Learning-Based Power Noise Induced Eye Diagram Estimation Agent for HBM6	이정현
		15:05 ~ 15:30	Break (25분)	

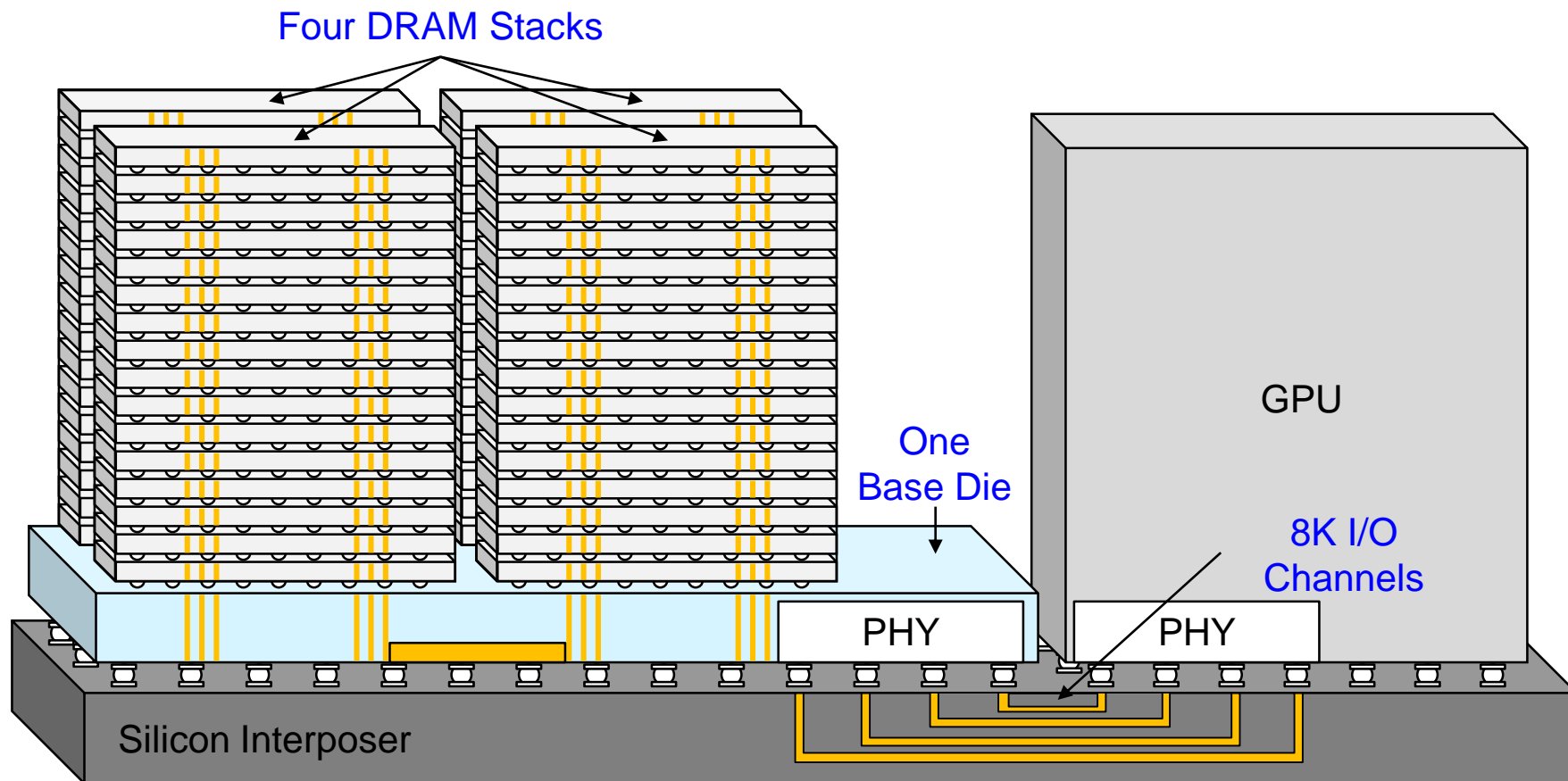
# [HBM6] Quad-Tower HBM (QT-HBM) Architecture for Enhanced Memory Capacity and Bandwidth

Taesoo Kim

Advising Professor : Prof. Jounggho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

# Quad-Tower High-Bandwidth Memory Architecture

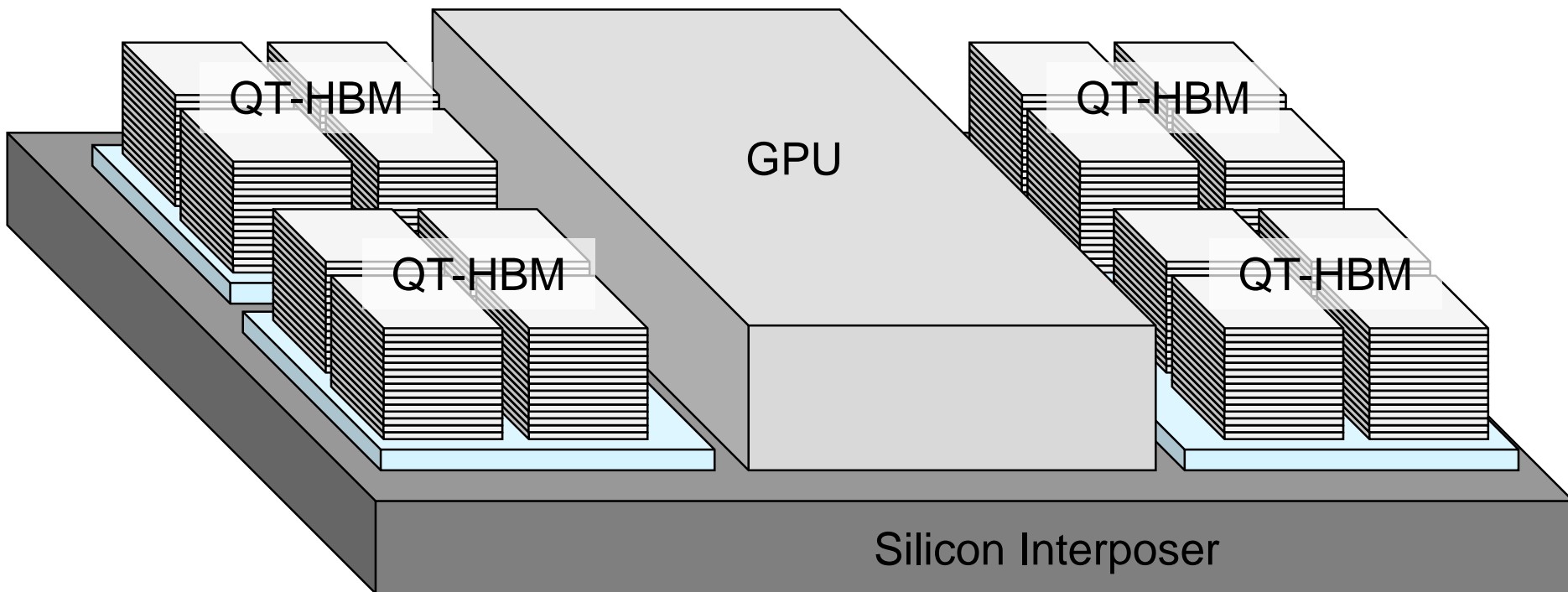


< Overview of Quad-Tower HBM Architecture >

- In Quad-Tower HBM (QT-HBM) architecture, four DRAM stacks are arranged in a  $2 \times 2$  configuration on a single base die.
- The QT-HBM is connected to the GPU via a silicon interposer with 8,096 I/O channels.

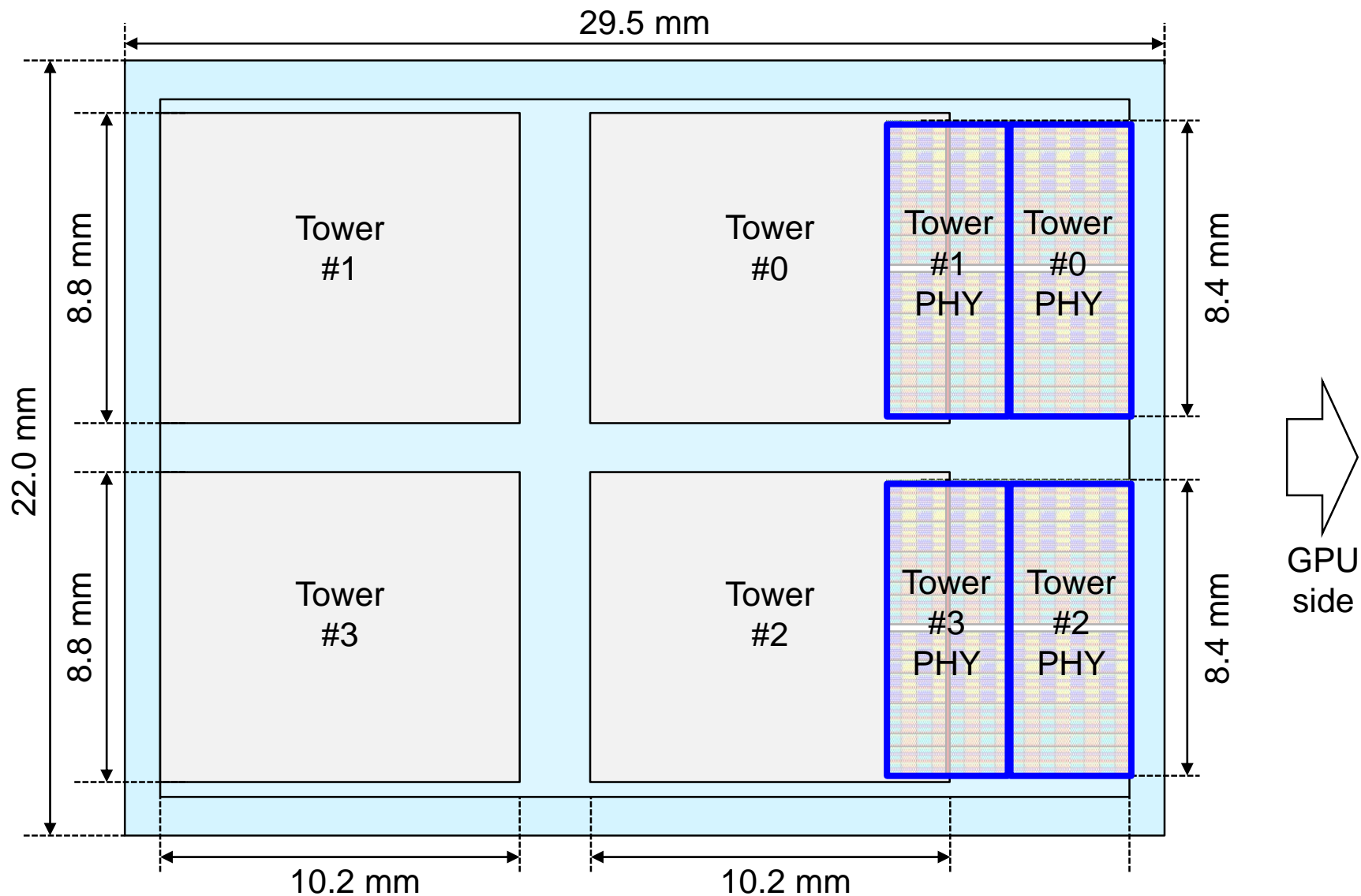


# Overview of the QT-HBM with GPU

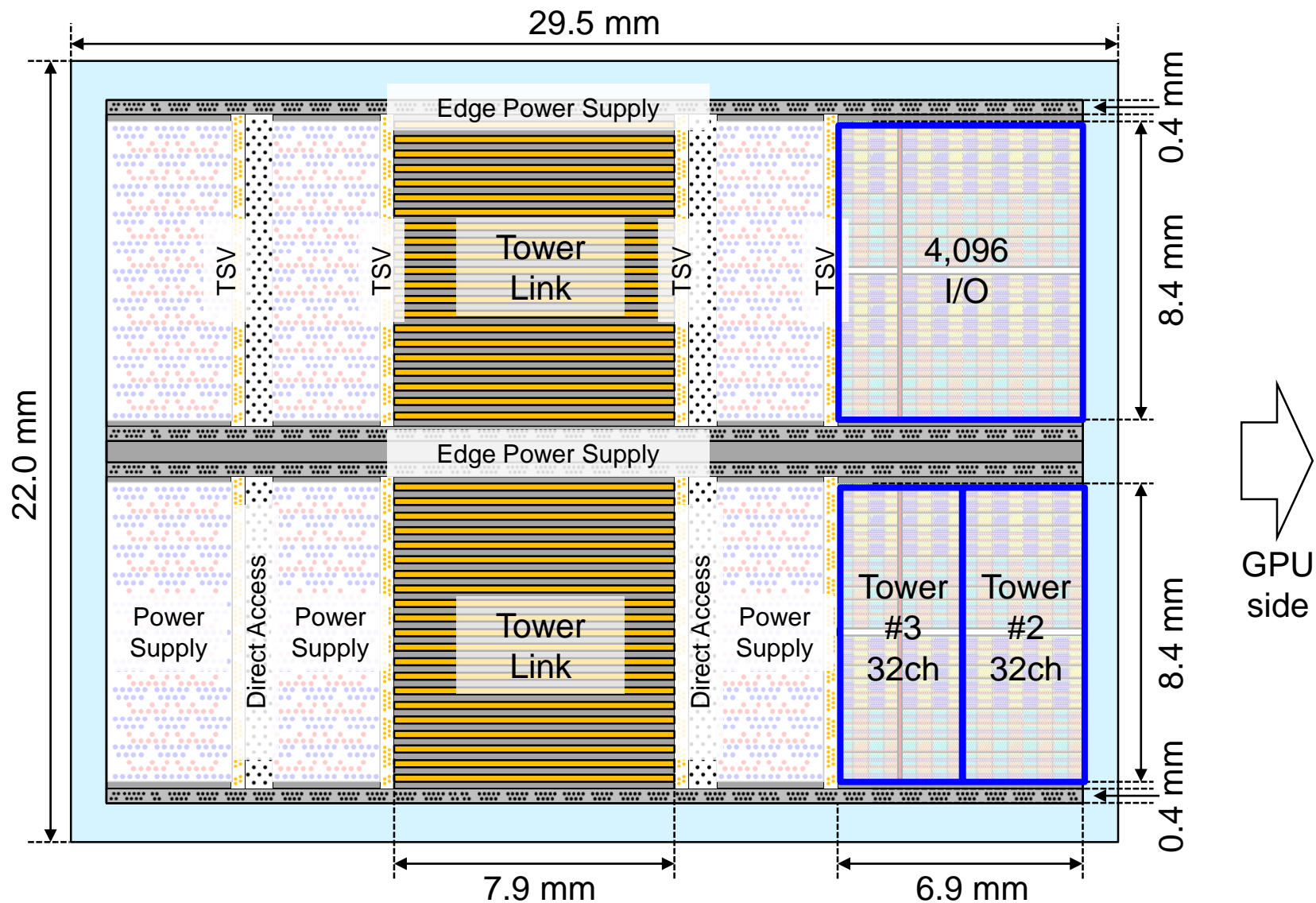


< Overview of the QT-HBM with GPU >

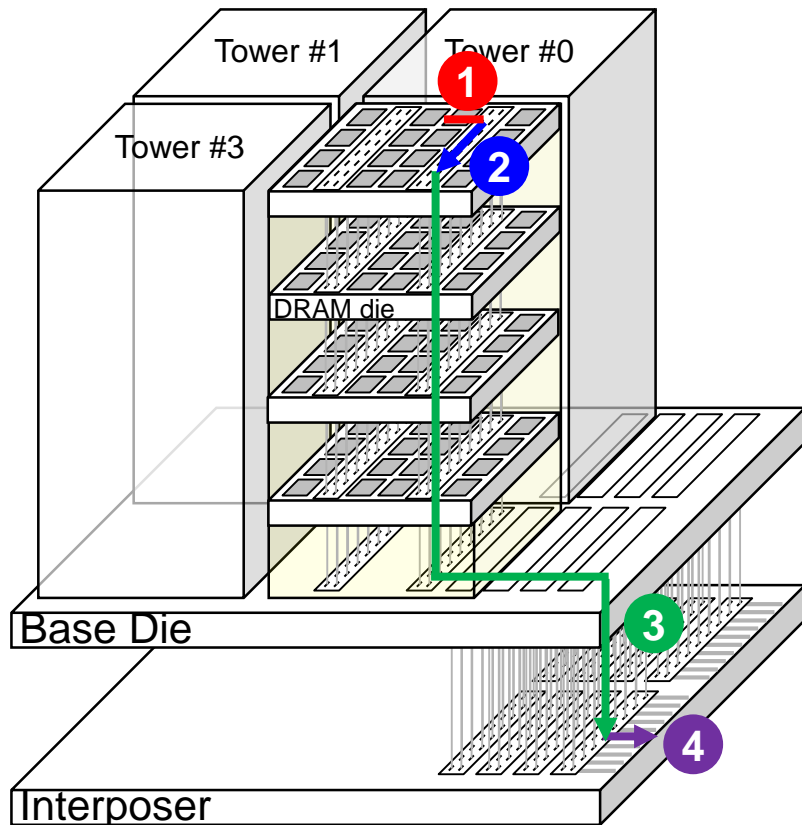
# Floor Plan of QT-HBM Base Die [1/2]



# Floor Plan of QT-HBM Base Die [2/2]



# DRAM Bandwidth Teardown in QT-HBM



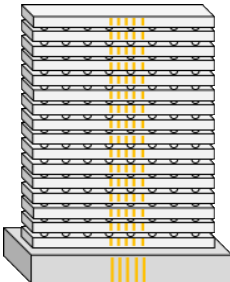
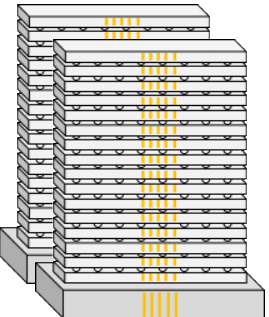

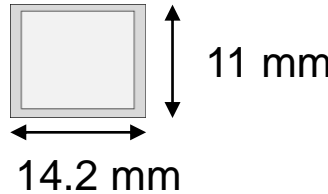
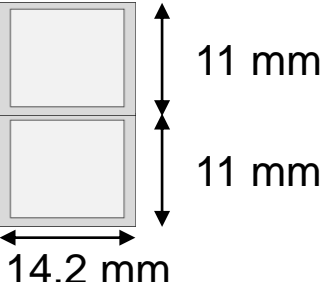
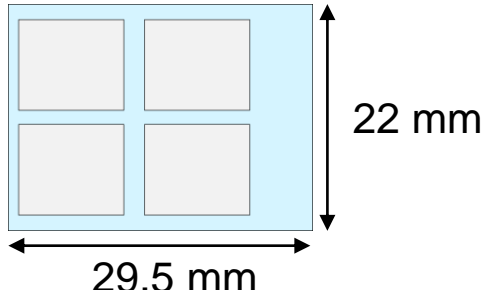
< Data path in QT-HBM >

Data path		Configuration	Bandwidth
Each Tower	Bank group (on-chip)	32,768 I/O x 0.375 Gb/s (4nCK)	1,536 GB/s
	Global (on-chip)	16,384 I/O x 0.75 Gb/s (2nCK)	1,536 GB/s
	Die-to-die (TSV)	8,192 I/O x 1.5 Gb/s (1nCK)	1,536 GB/s
	Logic die (on-chip)	8,192 I/O x 1.5 Gb/s (1nCK)	1,536 GB/s
	Interface (interposer)	2,048 I/O x 6 Gb/s (0.25 nCK)	1,536 GB/s
QT-HBM (= 4 x Tower)		8,192 I/O x 6 Gb/s (0.25 nCK)	6 TB/s (= 4 x 1,536 GB/s)

< DRAM bandwidth of each data path in QT-HBM >

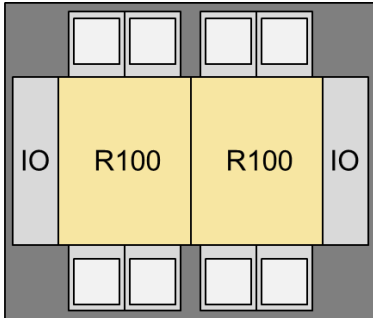
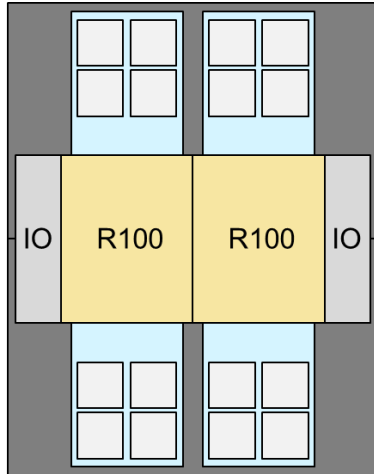
- From bank group I/O to interface I/O, the number of I/O and data rate are hierarchical.
- I/O bus window is extended by time-division multiplexing.

# Comparison between QT-HBM and HBM4

	HBM4	2 * HBM4	Proposed Quad-Tower HBM
Overview			
Form Factor			
DRAM Capacity	64 GB (= 16 hi x 32 Gb)	128 GB (= 2 * 64 GB)	↑ 100% 256 GB (= 4 * 64 GB)
# of I/O	2,048	4,096 (= 2 * 2,048)	8,192 (= 2 * 4,096)
Data rate	8 Gb/s	8 Gb/s	6 Gb/s
DRAM Bandwidth	2 TB/s (= 2,048 * 8 Gb/s)	4 TB/s (= 2 * 2 TB/s)	↑ 50% 6 TB/s (= 8,192 * 6 Gb/s)

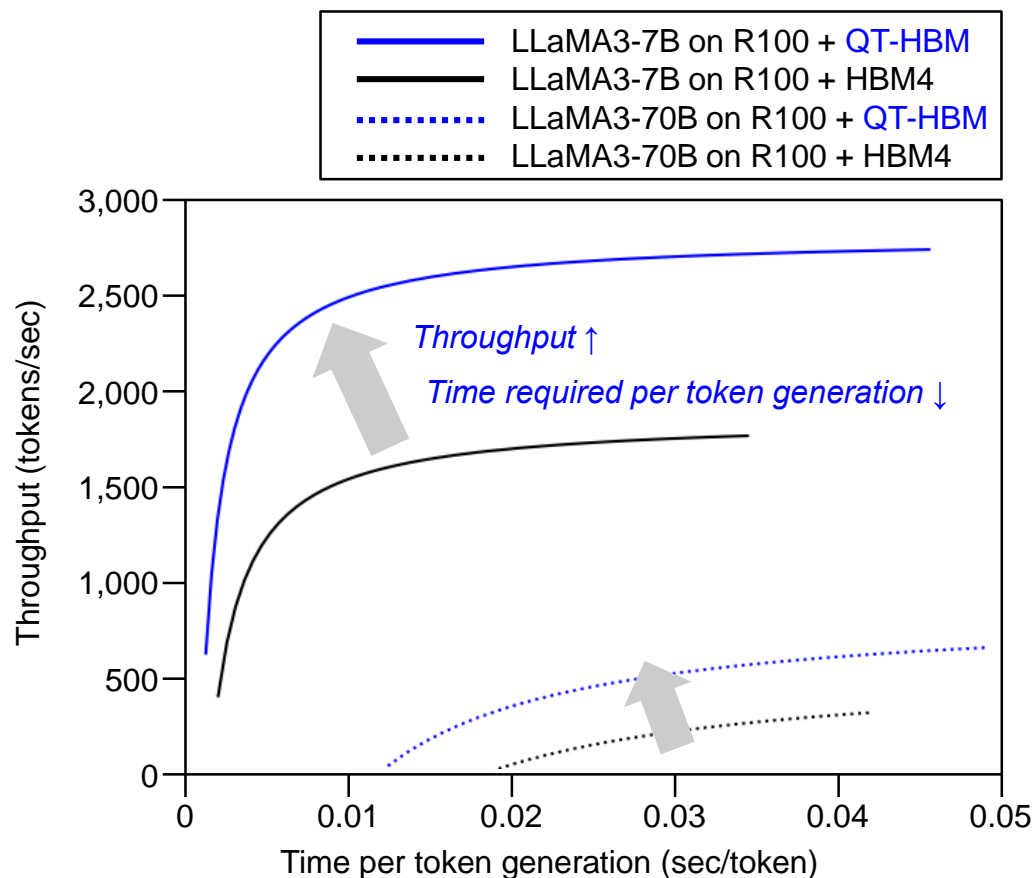
< Comparison between Quad-Tower HBM and HBM4 >

# GPU-HBM Module Comparison: QT-HBM vs HBM4

	HBM4-GPU module	Proposed QT-HBM-GPU
Figure		
DRAM Capacity	512 GB (= 8 x 64 GB)	<div>↑ 100%</div> 1 TB (= 16 x 64 GB)
DRAM Bandwidth	16 TB/s (= 8 x 2 TB/s)	<div>↑ 50%</div> 24 TB/s (= 4 x 6 TB/s)
Memory Density	0.106 GB/mm <sup>2</sup> (= 512GB / 74*65.4 mm <sup>2</sup> )	<div>↑ 36%</div> 0.144 GB/mm <sup>2</sup> (= 1,024GB / 74*96 mm <sup>2</sup> )

< Comparison of the QT-HBM versus HBM4 with the NVIDIA R100 GPU >

# Inference Throughput Comparison: QT-HBM vs HBM4



Throughput @Latency: 0.03

# param	HBM4	QT-HBM
7B	1,752	2,684
70B	239	541

< Inference throughput comparison: QT-HBM vs HBM4 >

- QT-HBM demonstrates higher throughput and lower latency in LLM inference compared to HBM4, due to its higher bandwidth and greater capacity.

# Thank You!

## HBM



# Large-Scale Hybrid Interposer for Multi-Tower HBM6 Architecture

Haeseok Suh

Advising Professor : Prof. Joungho Kim

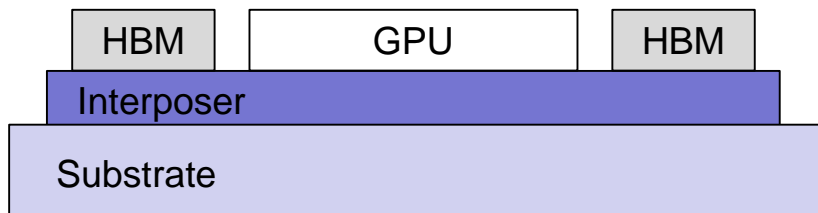
TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering

KAIST

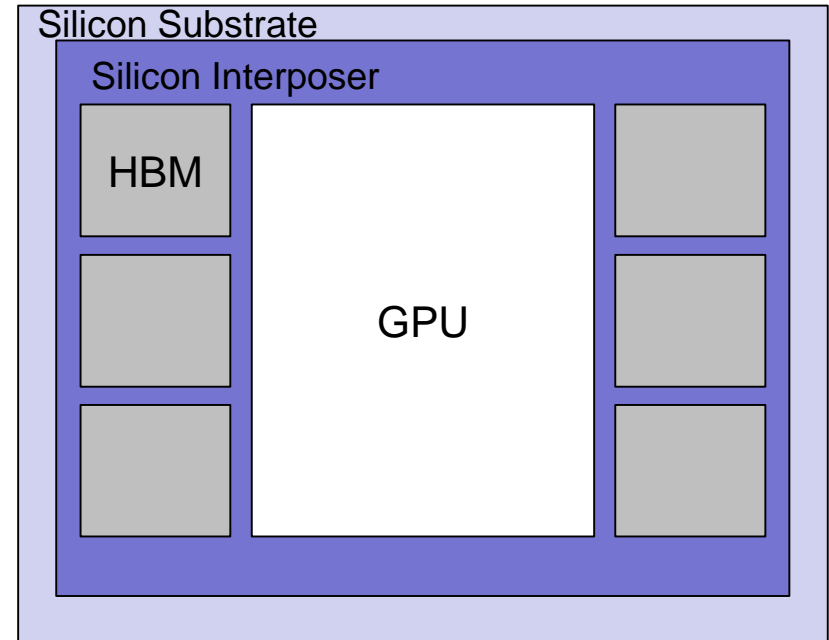
# Limitation of Current GPU-HBM Architecture: Lack of Scalability in Silicon Interposer

GPU: x1 reticle size (858 mm<sup>2</sup>)

HBM: 11 mm by 11 mm (121 mm<sup>2</sup>)



Interposer: x3.3 reticle size (2830 mm<sup>2</sup>,  
current max: 2 GPU + 8 HBM)

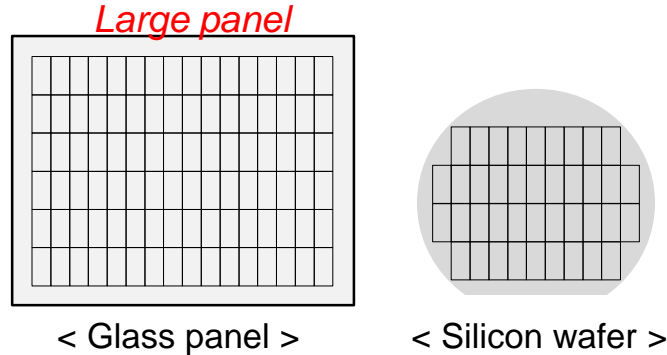


Substrate: 80 mm by 80 mm (6400 mm<sup>2</sup>)

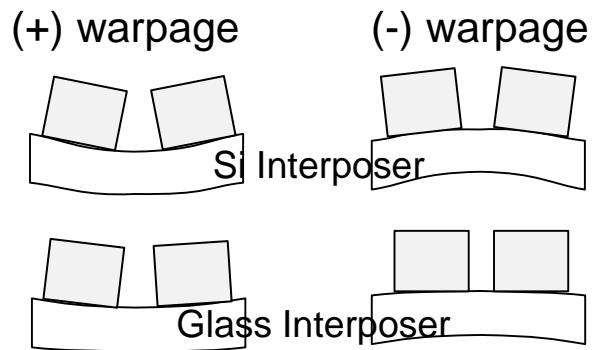
< Cross-sectional and top view of conventional GPU-HBM using CoWoS Method >

- GPU-HBM needs to be scaled out more and more due to increasing requirement of AI workloads.
- Interconnections outside of interposer goes through substrate, which has much higher latency and lower bandwidth compared to interposer channels.
- Scaling is limited by both the interposer area and the substrate area.
- Silicon interposer scaling is extremely difficult due to warpage/yield/cost issue.

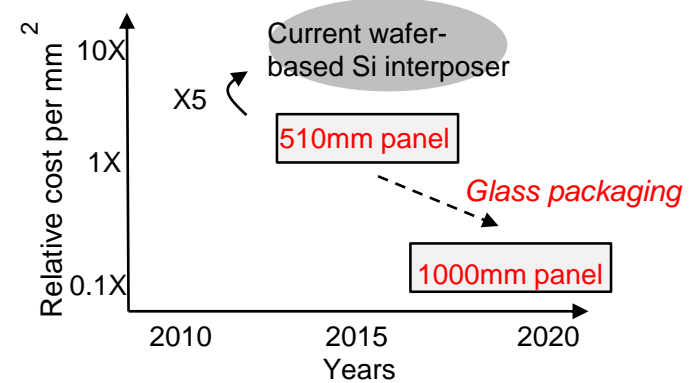
# Advantages of Glass Interposer



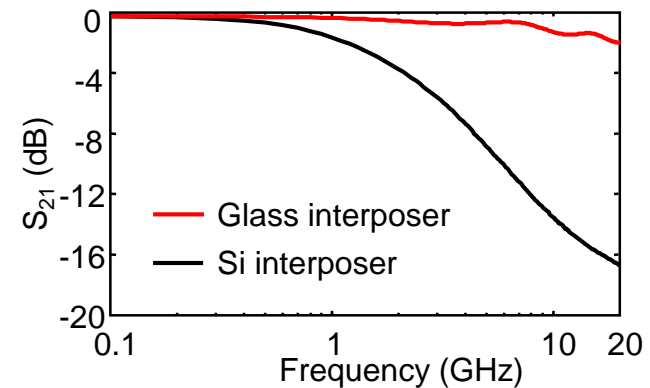
< Panel size difference >



< Warpage Issue >



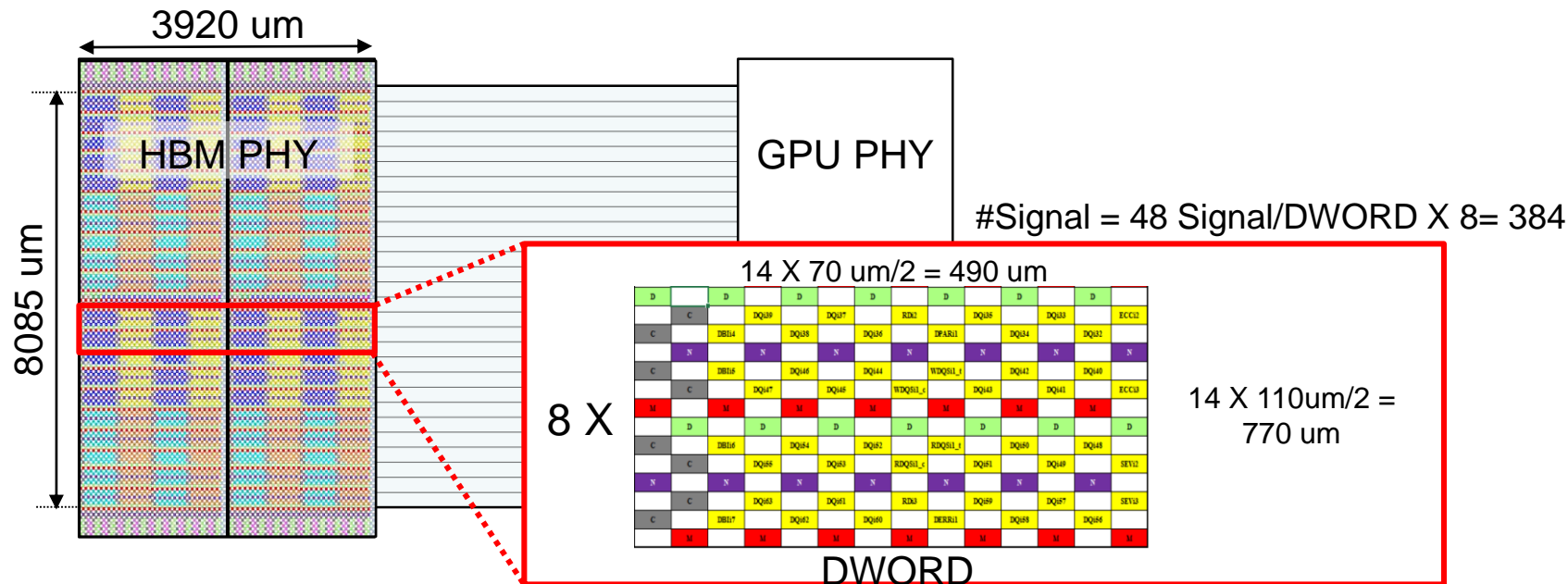
< Cost reduction >



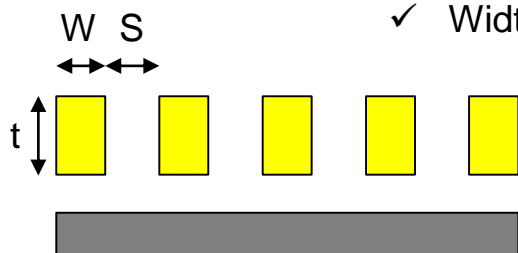
< Low-loss characteristic >

- Glass interposer can be made bigger than silicon interposer, providing scalability.
  - Panel process with higher yield, lower cost
  - Lower warpage issues
  - Lower loss

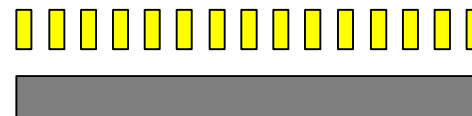
# Disadvantage of Glass Interposer: Coarse Pitch Routing



✓ Width+Space limitation from routability:  $770 \text{ um} * \text{\#Layer} / (\text{\#S} + \text{\#G})$



Glass Interposer:  $W/S = 2 \text{ um}$ ,  $t = 4 \text{ um}$

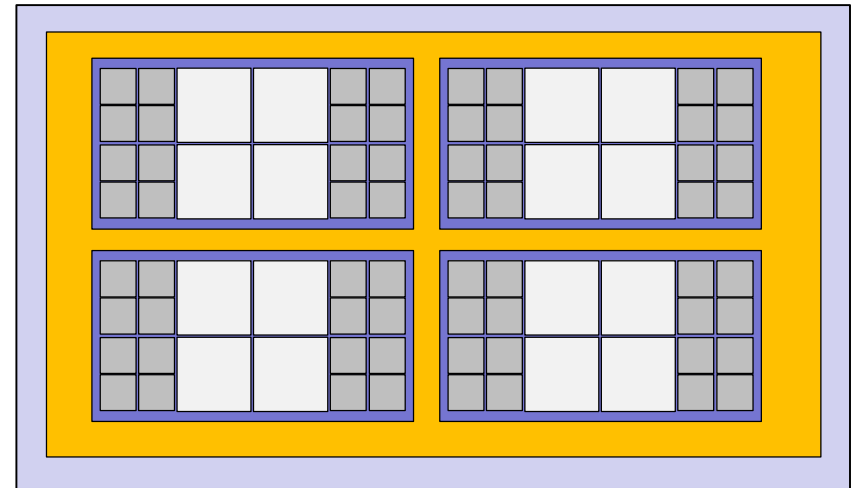
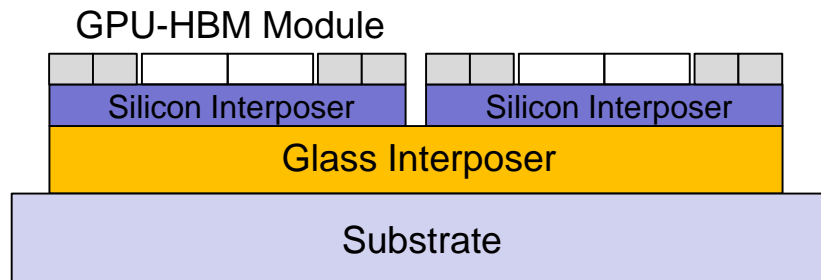


Si Interposer:  $W/S = 0.4 \text{ um}$ ,  $t = 1 \text{ um}$

< Width, space, thickness comparison of glass and Si interposer >

- Due to routability from width and space, the number of interconnection between GPU and HBM is limited.
- Minimum width/space/metal thickness for Si interposer is much smaller than glass.
- So, silicon interposer enables much more finer routing, which leads to more bandwidth.

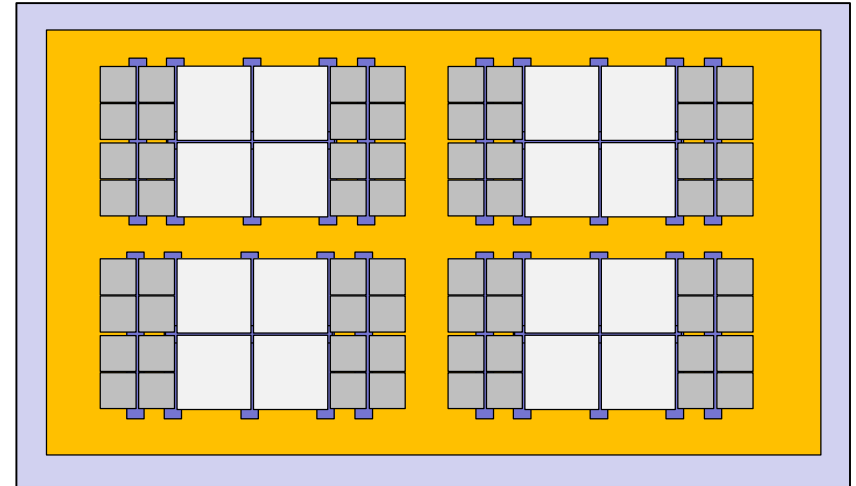
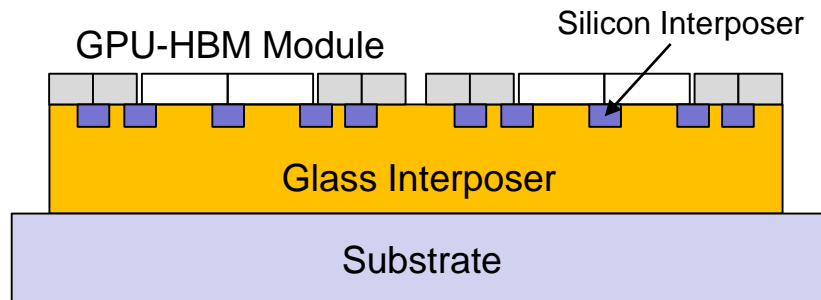
# Next Generation Hybrid Interposer [1/2]: Silicon-Glass 2 Layer Hybrid Interposer



< Cross-sectional and top view of Si-Glass Hybrid 2 Layer Interposer >

- 2 layer hybrid interposer uses both large scale glass interposer and fine pitch silicon interposer for GPU-HBM module.
- GPU-GPU, GPU-HBM, HBM-HBM interconnection is done with fine I/O silicon interposer.
- GPU module (such as B100)-GPU module interconnection is done with glass interposer.
  - Conventional structure required substrate interconnection, which has low bandwidth.
- Using the hybrid structure, the advantage of both glass and silicon can be used.
- However, two layers of interposer require additional bumps, leading to more cost and possible electrical performance issue.

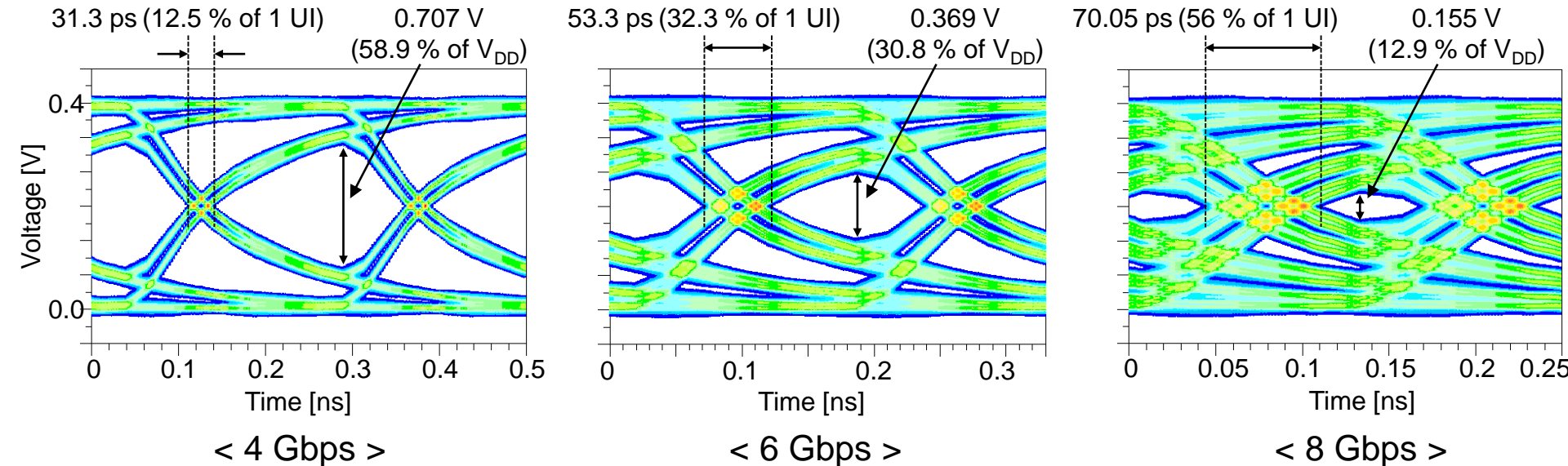
# Next Generation Hybrid Interposer [2/2]: Silicon Embedded Glass Hybrid Interposer



< Cross-sectional and top view of Si-embedded glass hybrid 2 layer interposer >

- Another viable option is embedding silicon interposer inside glass interposer.
- Silicon interposer is placed inside glass interposer cavity.
- This can reduce the usage of low yield, high cost Si interposer while maintaining fine-pitch I/O in necessary regions.
- Also, this structure does not need additional bumps between the two interposers.
- GPU-GPU, GPU-HBM, HBM-HBM interconnection is done with fine I/O embedded silicon interposer.
- GPU module (such as B100)-GPU module interconnection is done with glass interposer.
- However, difference in thermal expansion coefficient may cause issues.

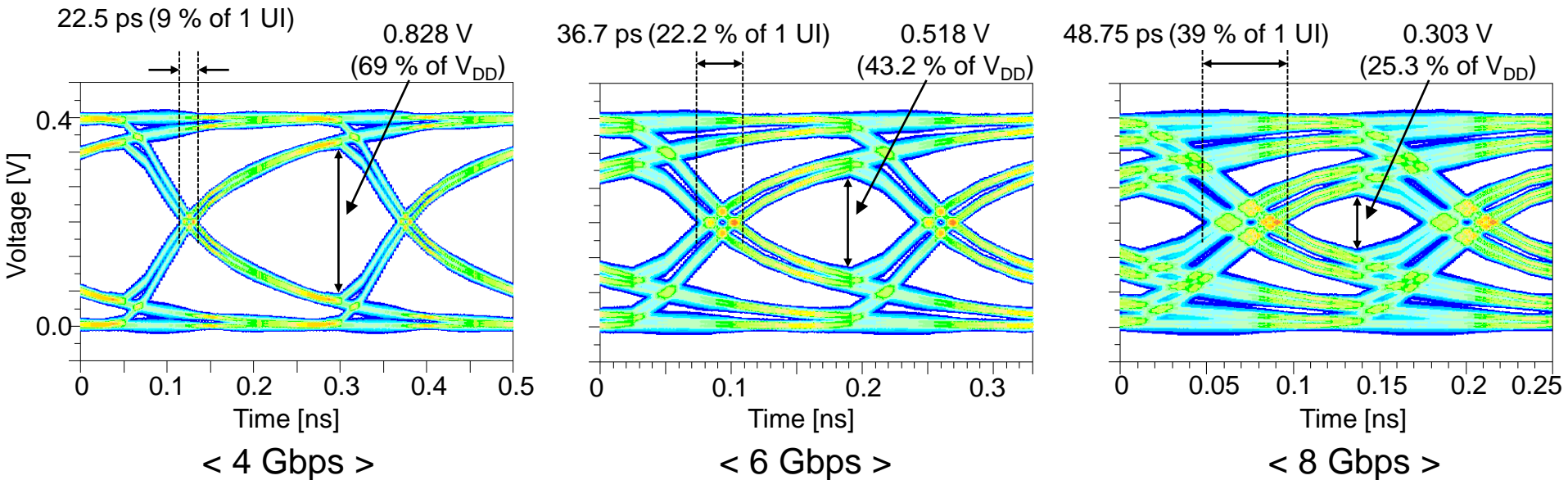
# Signal Integrity Analysis: Eye-diagram of Microstrip Line based on Silicon Interposer (Length = 5 mm)



< Eye-diagram of microstrip line based on silicon Interposer >

- To show the electrical performance of silicon and glass interposer, eye diagram test is done based on GPU-HBM channel (5 mm).
- Based on a silicon interposer, it is hard to achieve the data rate up to 6 Gbps without equalizer for data channels.
- This is due to high loss and fine pitch traces.

# Signal Integrity Analysis: Eye-diagram of Microstrip Line based on Glass Interposer (Length = 5 mm)

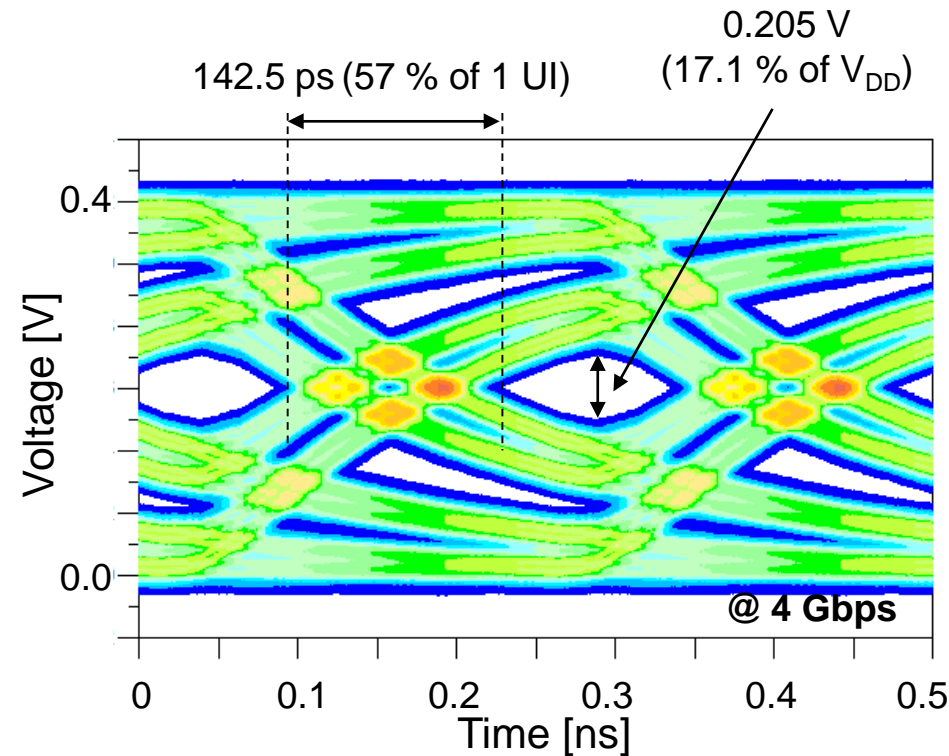


< Eye-diagram of designed microstrip line based on glass Interposer >

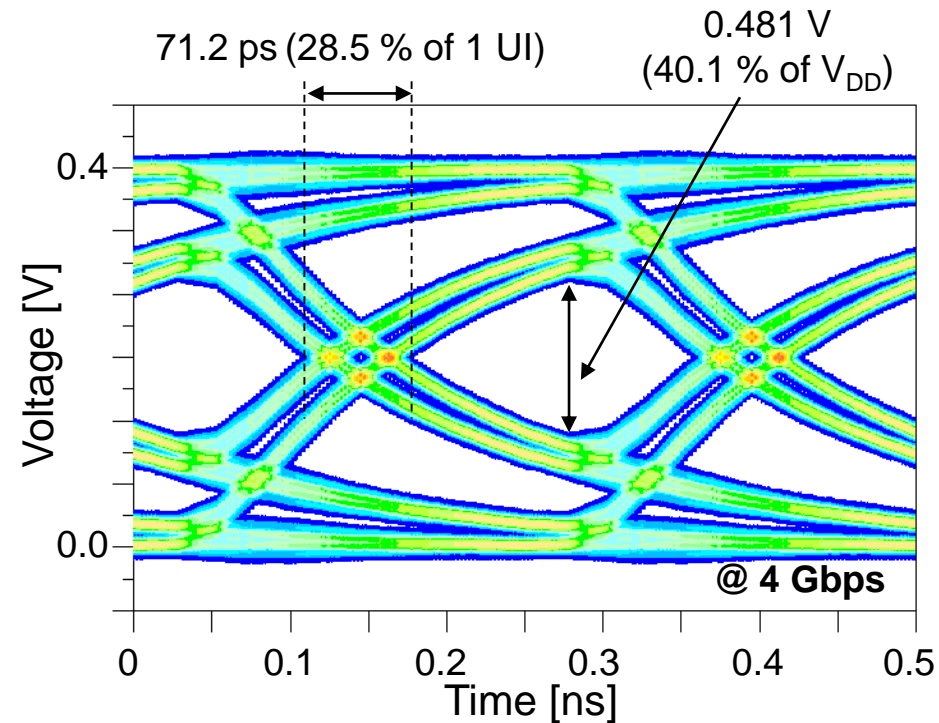
- Due to low loss characteristics of a glass interposer, designed microstrip line based on a glass interposer show better SI performance than that of a silicon interposer.
- Based on a glass interposer, it can be possible to achieve the data rate up to 6 Gbps without equalizer for data channels.



# Signal Integrity Analysis: Eye-diagram of Microstrip Line depending on Interposer Material (Length = 10 mm)



< Silicon Interposer >



< Glass Interposer >

< Eye-diagram of microstrip line (10mm) depending on interposer material >

- Based on a glass interposer, it can be possible to achieve the data rate up to 4 Gbps without equalizer for 10 mm data channel.
- So, it is appropriate for long channels between GPU modules, with high bandwidth.

# Thank You!

## HBM

# L3 Cache Embedded (L3E) HBM6 Architecture for LLM Inference

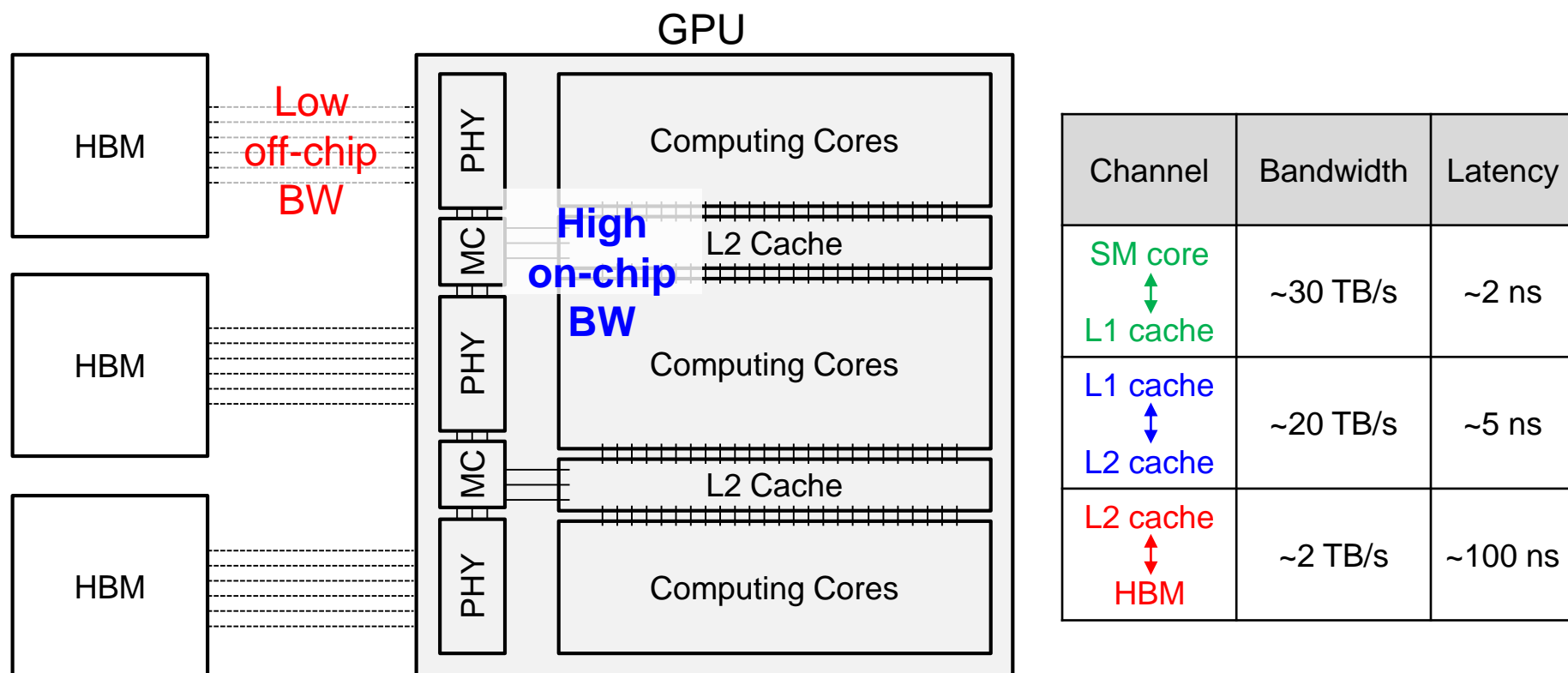
Haeseok Suh

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering

KAIST

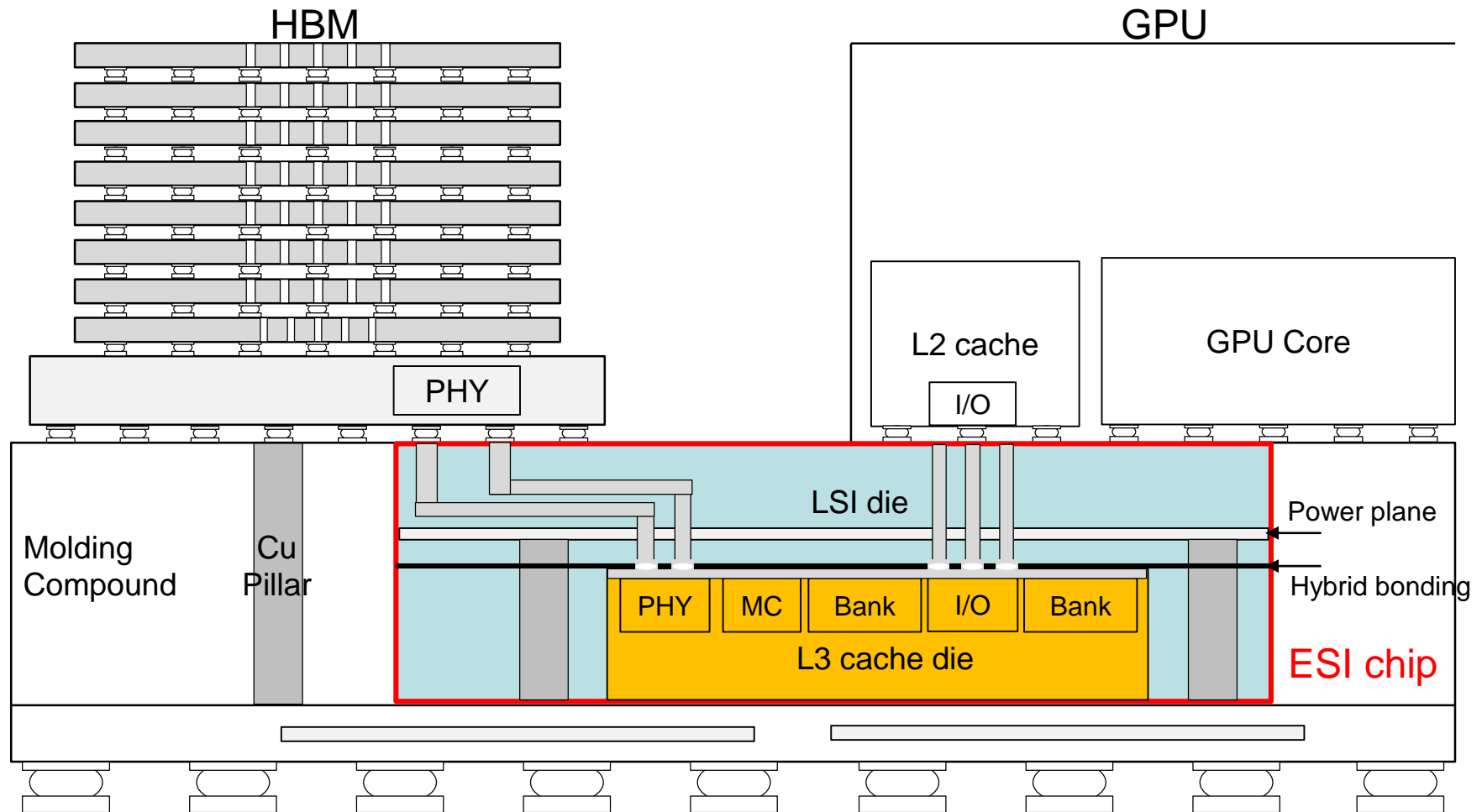
# Limitation of Current GPU-HBM Architecture



## < GPU-HBM architecture on-chip and off-chip bandwidth >

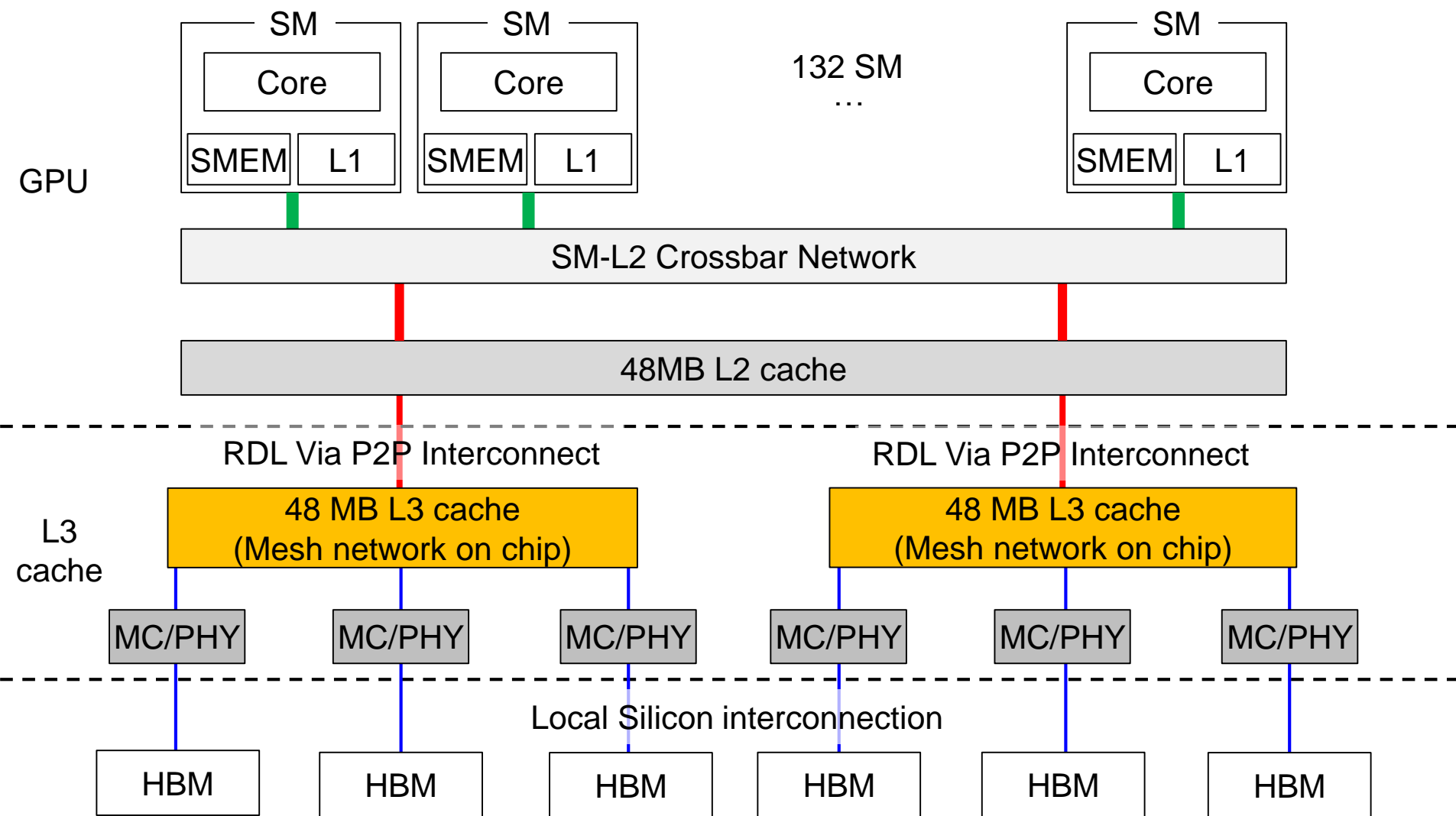
- HBM enables higher memory bandwidth than GDDR, but still provides much less bandwidth compared to on-chip bandwidth between processing clusters and cache.
- This discrepancy creates performance degradation, as LLMs need lots of memory access.

# Overview of Proposed L3E-GPU-HBM Architecture



< Cross-sectional view of L3E-GPU-HBM >

# High-level Block Diagram of Proposed L3E-GPU-HBM Architecture

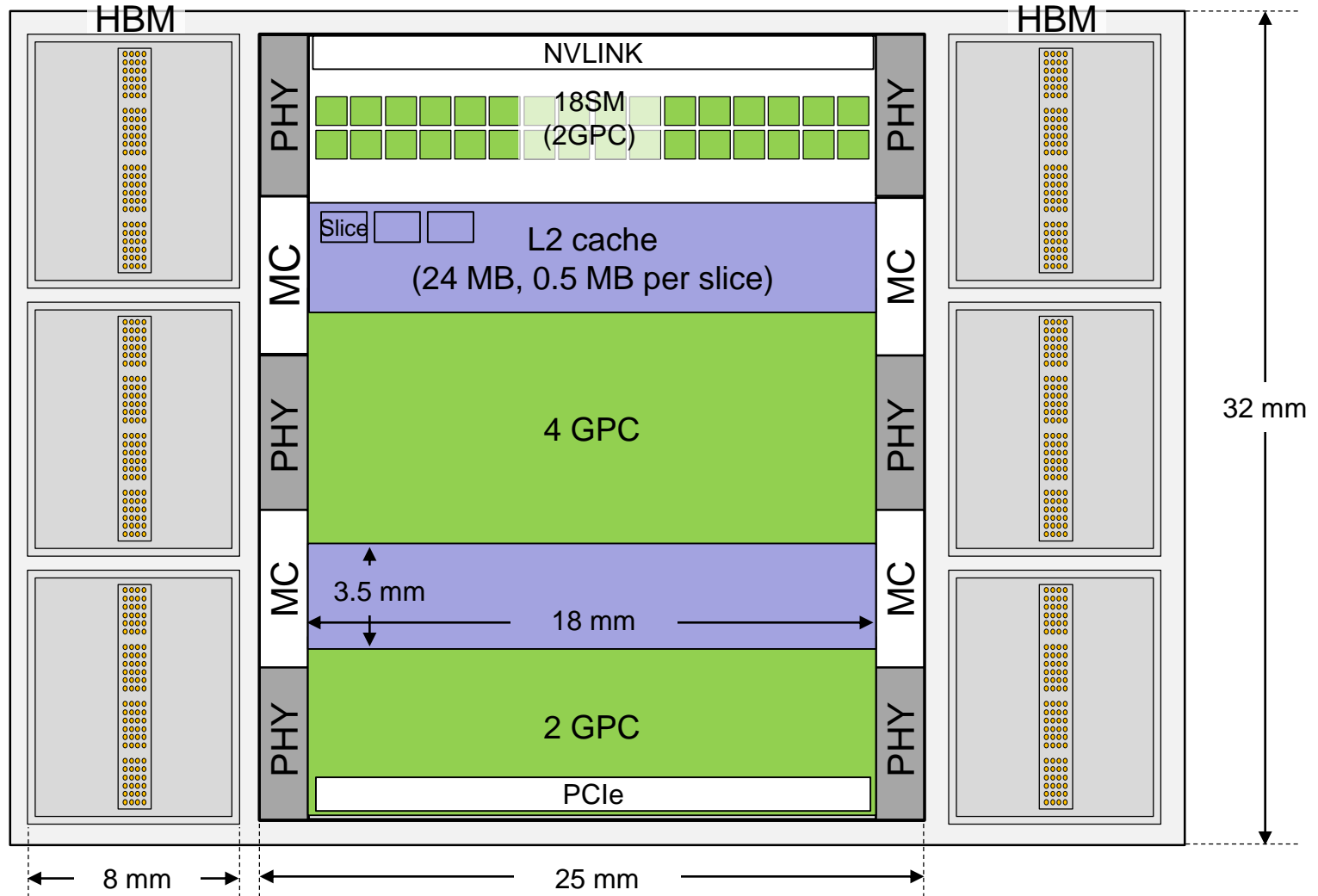


< Block Diagram of L3E-GPU-HBM >

# Floorplan of Conventional GPU-HBM Architecture

Total size: 32mm \* 42mm = 1344mm<sup>2</sup>

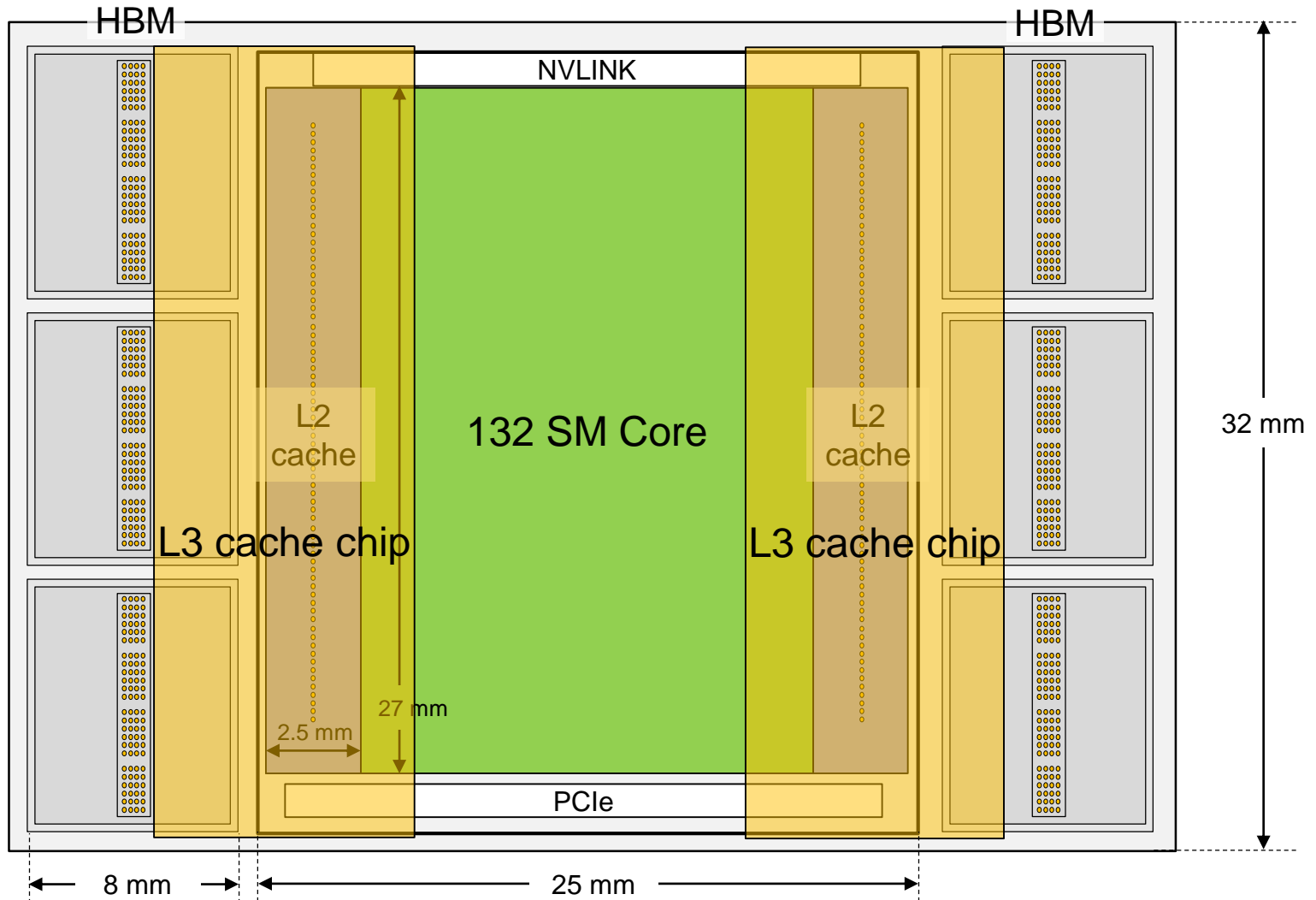
GPC: Graphic Processing Cluster  
MC: Memory controller PHY: Physical Layer



< Floorplan of H100 GPU >

# Floorplan of Proposed GPU-L3E-HBM Architecture

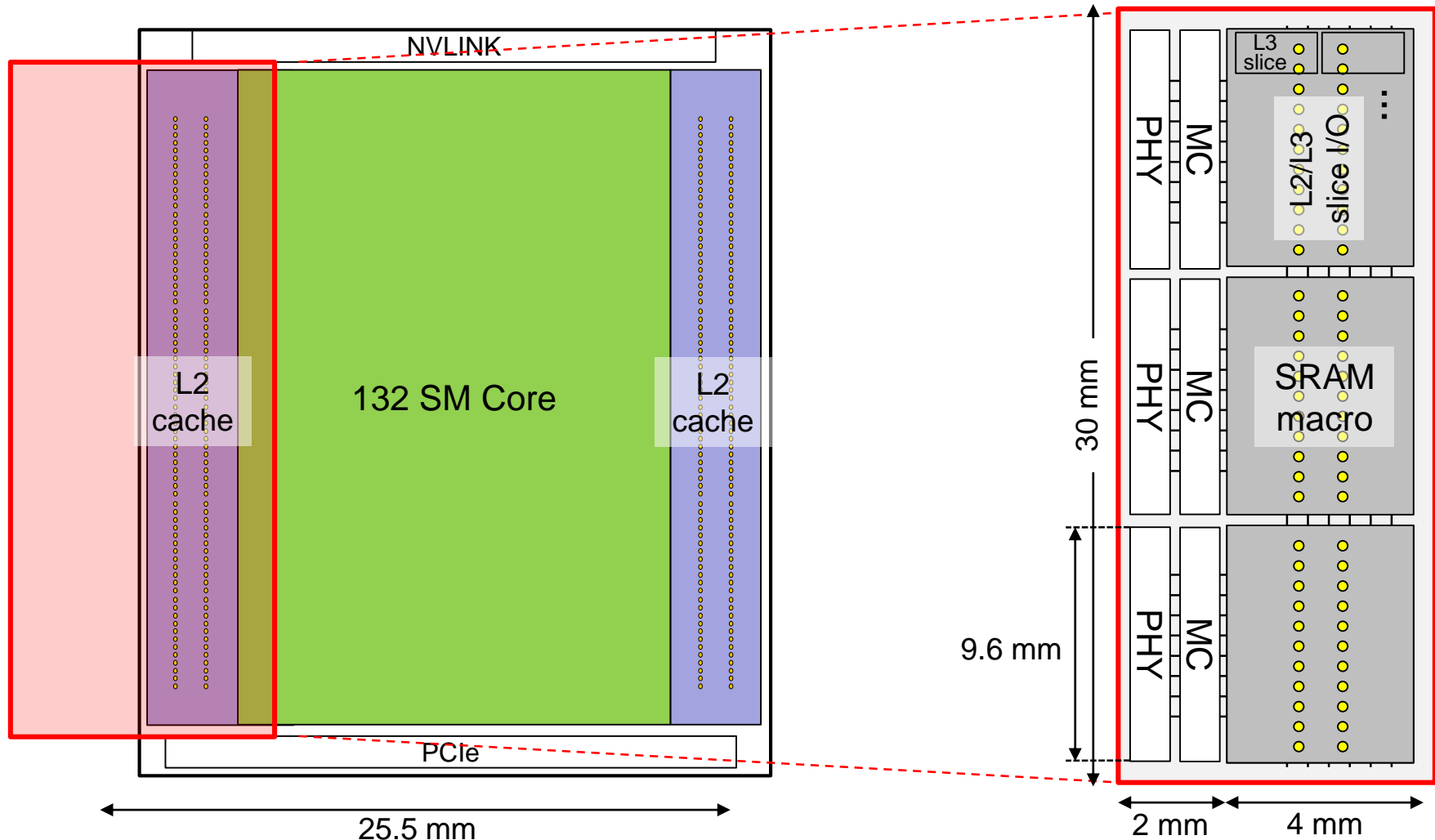
Total size: 32mm \* 42mm = 1344mm<sup>2</sup>



< Revised floorplan of proposed L3E-GPU-HBM >



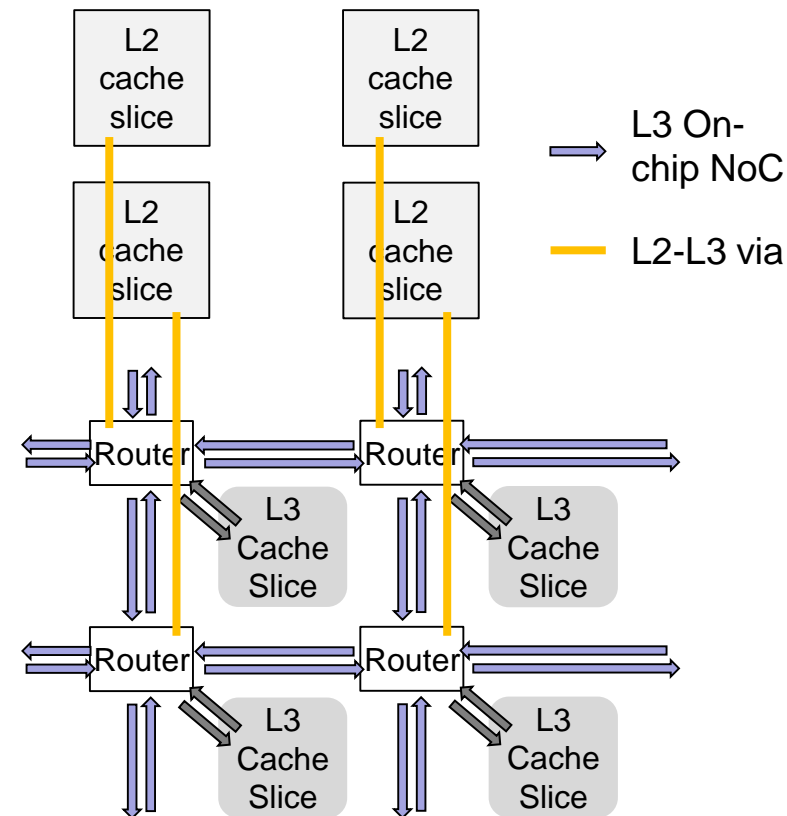
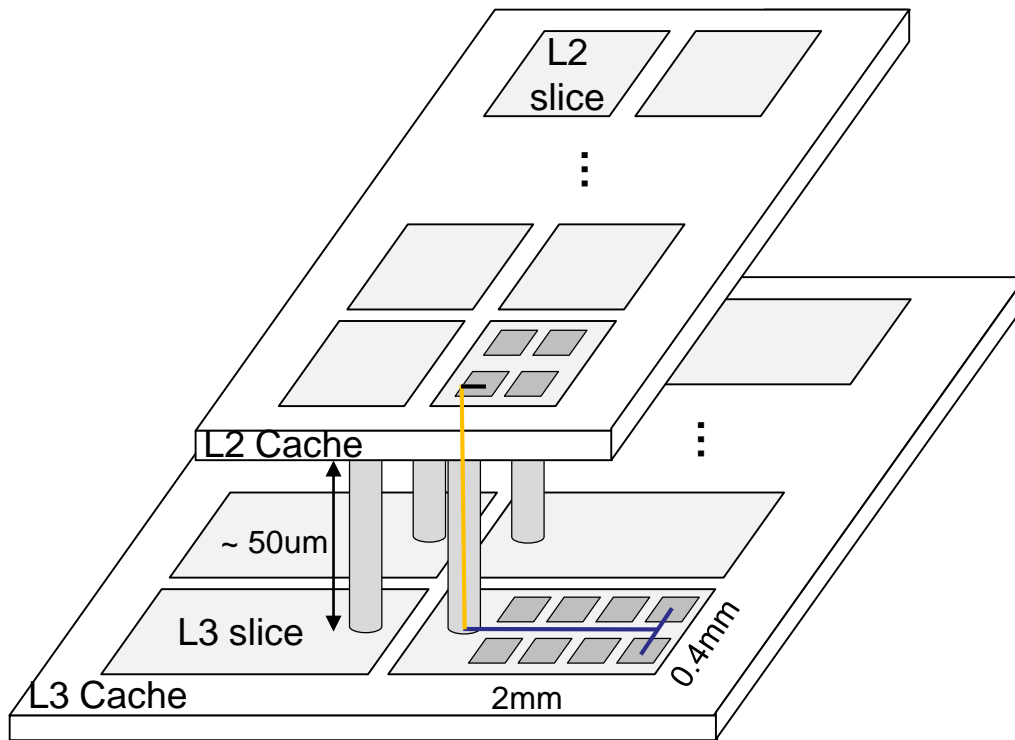
# Design of L3 Cache Chip Considering the Area of GPU-HBM



< Embedded L3 cache design >

- L3 cache is 48 MB on each side, with each slice connected to corresponding L2 slice.
- L2 slice ( $1\text{mm}^2$ ) is 0.5MB, so L3 cache slice is set as 1MB, total of  $32\text{mm}^2$  per side.

# L2 Cache – L3 Cache Interconnect: Via + Mesh Network

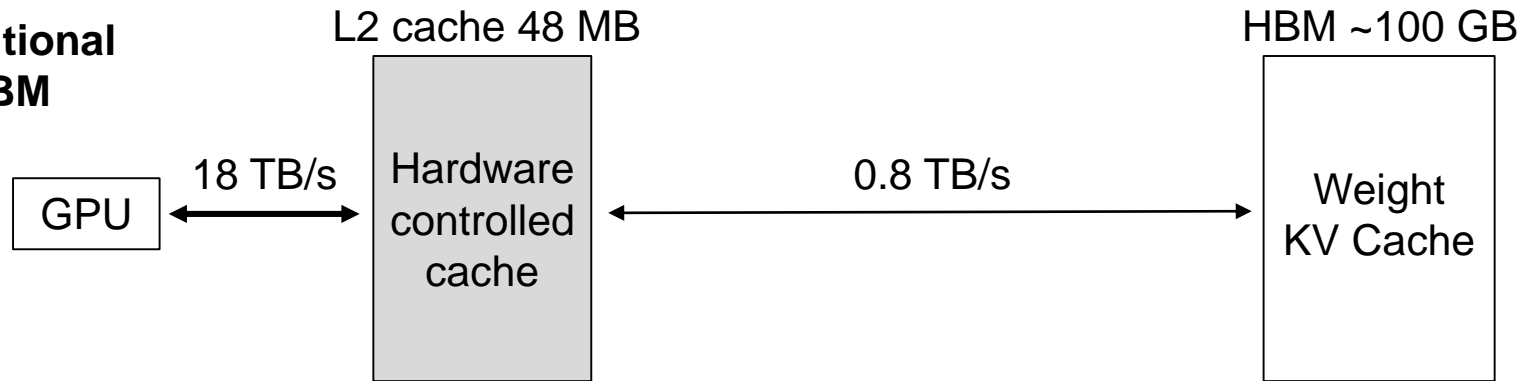


## < Interconnection scheme of L2 – L3 and L3 cache >

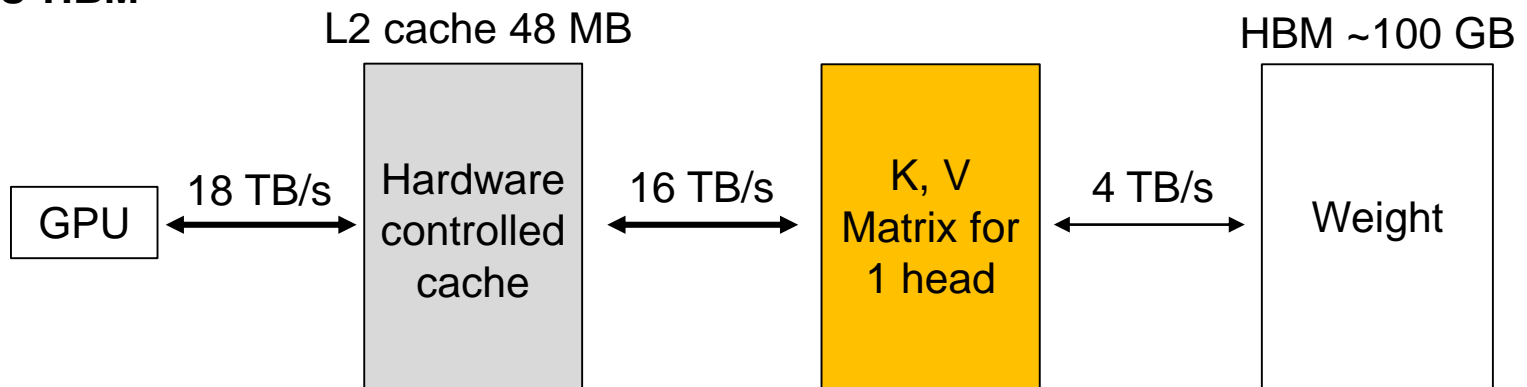
- A key to L3E-GPU-HBM is enabling fast, high-bandwidth interconnect between L2 cache of GPU and L3 cache in interposer.
- L2 and L3 cache is connected with short via, enabling ultra-high bandwidth.
- L3 cache is connected in mesh, so the data can be read from L3 cache to GPU efficiently.

# Reduction of HBM Access by Using L3 Cache as Shared Memory

## Conventional GPU-HBM



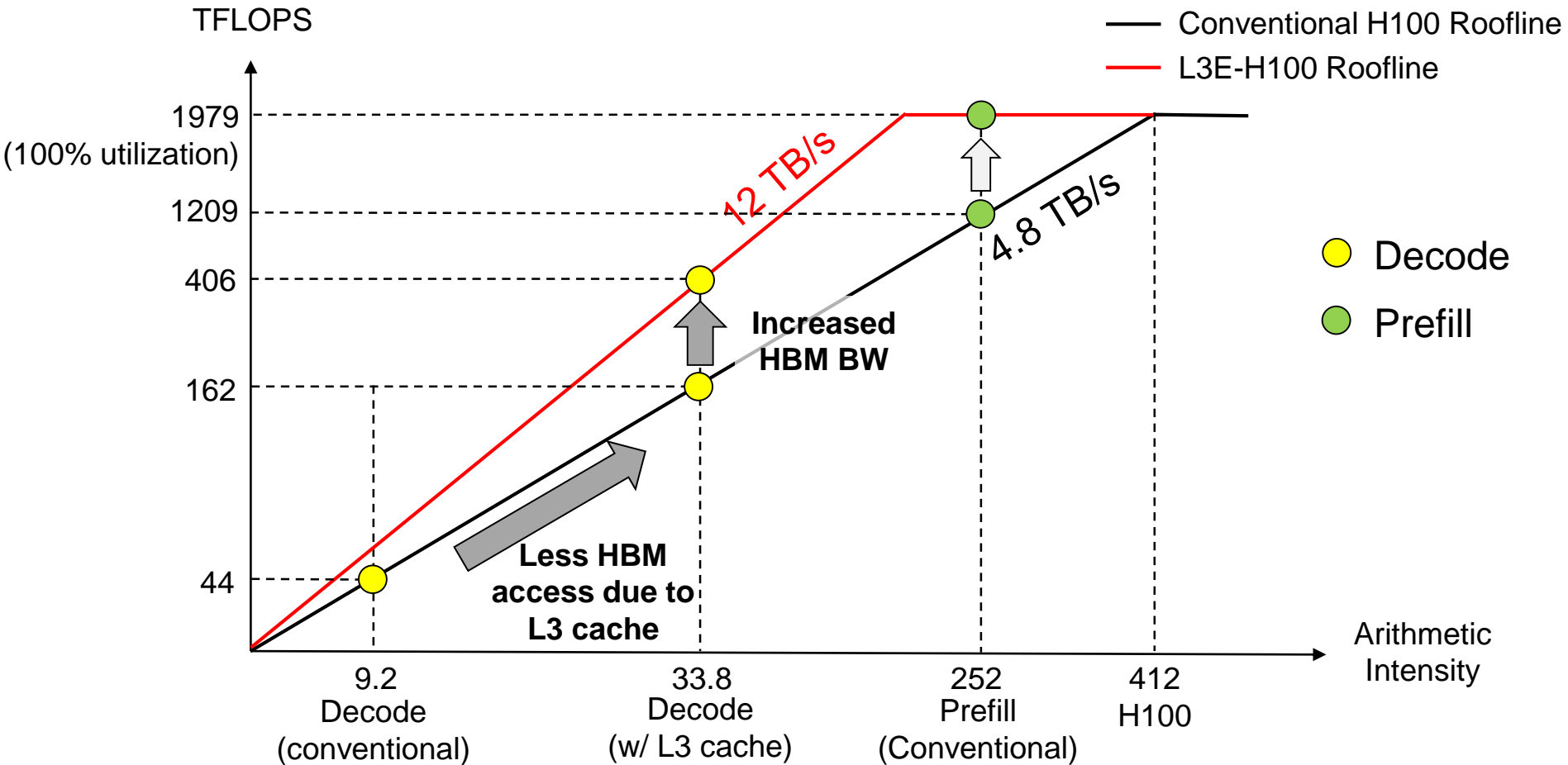
## L3E-GPU-HBM



### < Cache utilization for L3E-GPU-HBM >

- L2 cache of GPU is managed by “hardware”, which means programmer cannot control the contents.
- In L3E-GPU-HBM, L3 cache can be used as “shared memory”, which is programmable.
- If K,V matrix is programmed to be stored in L3 cache, HBM access will be most efficiently reduced.
- HBM access will decrease from  $O(d_m d_h + B s d_h)$  to  $O(d_m d_h)$  with 96 MB L3 cache.
  - ✓ 73% ↓ for transformer inference with 4096 tokens input and 4096 tokens generated.

# Computing Performance Improvement of Proposed Architecture due to Increased GPU Utilization



## < Utilization improvement of L3E-GPU-HBM

- **73% decrease of HBM access** by storing data in L3 cache and **x2.5 increase of HBM BW** results in performance boost of decode from 44 TFLOPS to 406 TFLOPS.
- Prefill process increases from 1209 TFLOPS to 1979 TFLOPS, which is 100% utilization.

# Token Per Second of Proposed L3E-GPU-HBM Architecture

Used model: Llama3-400B

Required FLOP	Prefill (input s=4096)	19900 TFLOP
	Decode (1 token)	5.1 TFLOP
	FFN (1 token)	0.54 TFLOP

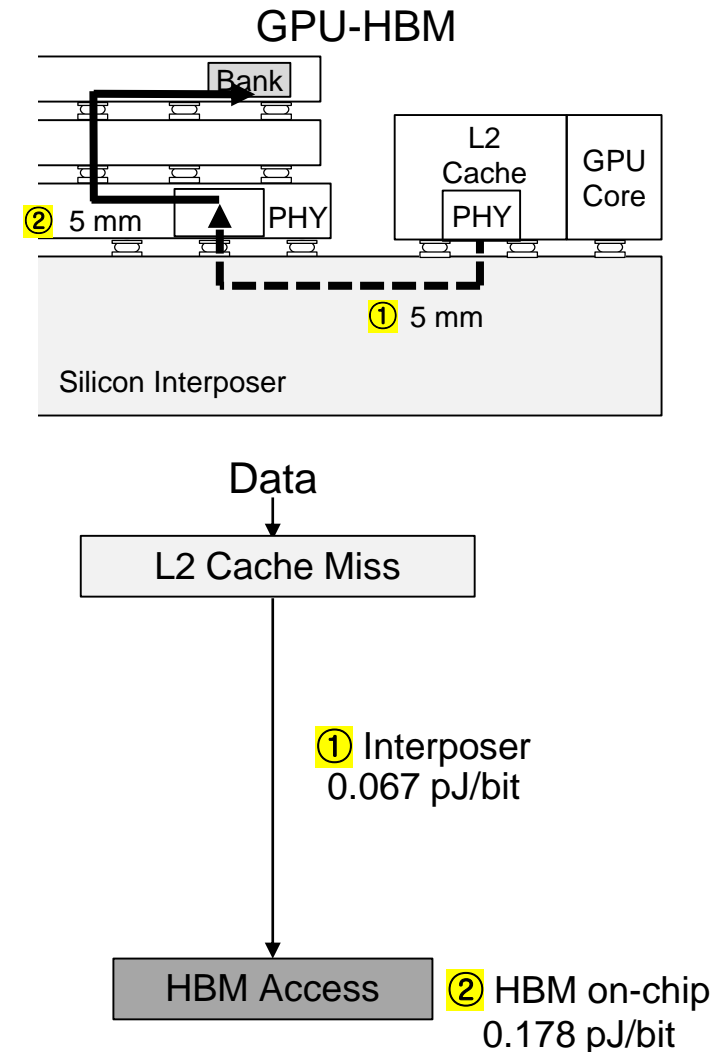
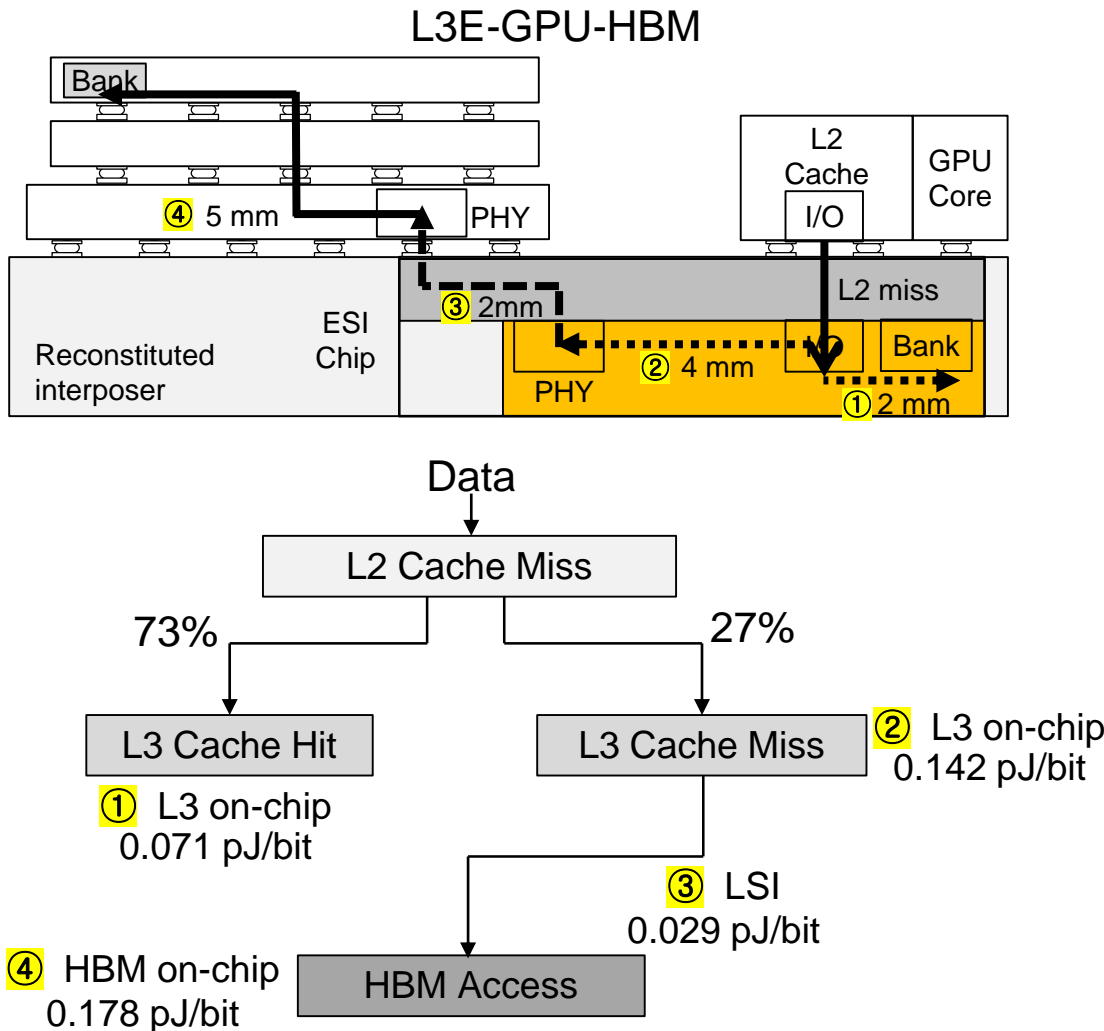
\* 4096 prefilled token, 4096 generated token, on 128 GPU with ideal scalability

	L3E-GPU-H100	H100	Improvement
Time-to-first-token	0.079 s	0.13 s	38.8%
Time-to-4096-token	0.49 s	3.88 s	87.3%
<b>Token/sec</b>	<b>8313 tokens</b>	<b>1057 tokens</b>	<b>686%</b>

< Token per second considering required FLOP and improved performance >

- Since the FLOPS and required FLOP of inference process is calculated, token per second can be obtained by calculating the time of each generated token.
- Time-to-4096-token: FLOP/FLOPS of all 3 stage  $\rightarrow \frac{19900}{1979*128} + \frac{5.14*4096}{406*128} + \frac{0.54*4096}{1979*128} = 0.290 \text{ s}$ 
  - $\rightarrow \frac{\text{Token}}{\text{second}} = \frac{4096}{0.493} = 8313 \text{ tokens per second for 128 GPU system.}$

# Energy Consumption Analysis of L3 Cache and LSI: Dynamic Energy Considering Hit Ratio and Length

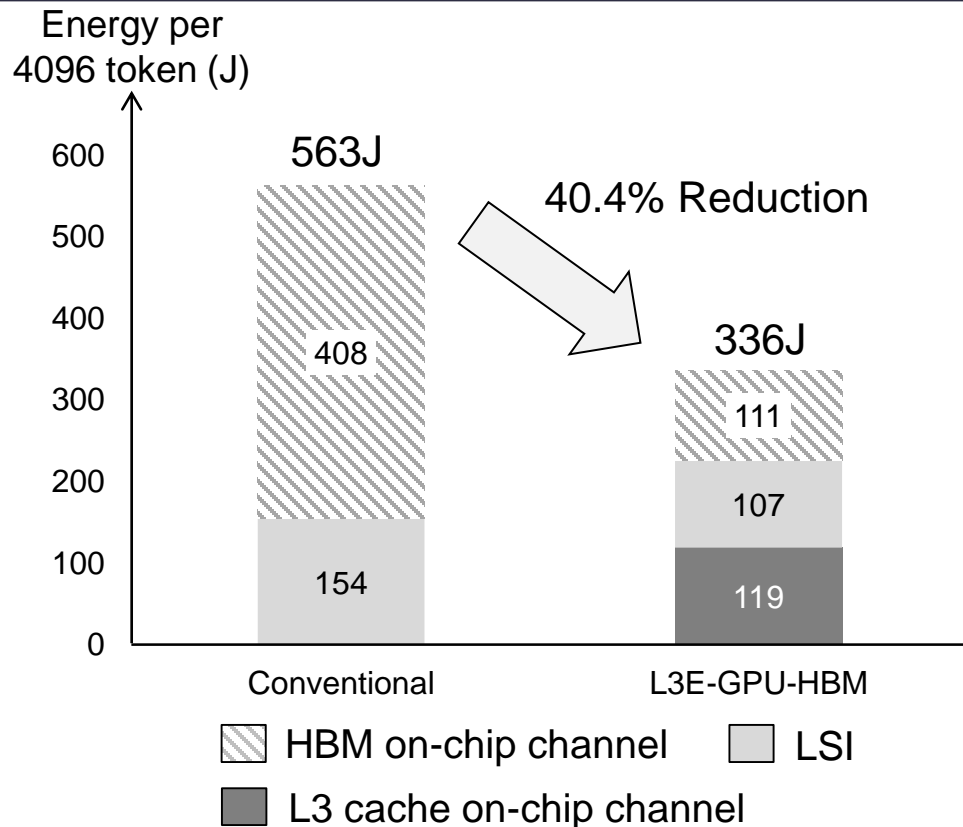


< Interconnect energy breakdown of proposed architecture >

- Depending on distance travelled per hit/miss, energy per access bit is obtained.
- With access bits, the total energy consumption can be achieved.

# Energy per Token Reduction of L3E-GPU-HBM

Access bits per 4096 token	H100	L3E-GPU-H100
HBM	2300 Tb	623 Tb
L3 Cache	-	1676 Tb



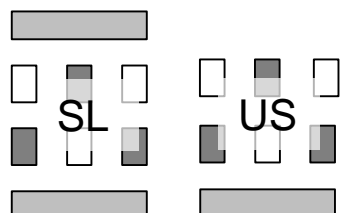
< Required HBM access for 4096 token >

< Energy per token reduction for L3E-GPU-HBM >

- To obtain energy per token of proposed architecture, three factors are considered.
  - 1) Required HBM access per token, 2) reduced HBM access, 3) energy per bit of each channel
- The proposed architecture decreased energy consumption by 40.4%.

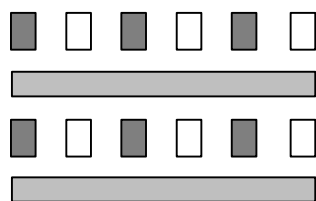
# Signal Integrity Analysis: Width and Space by Routability of Signal and Ground

Checker



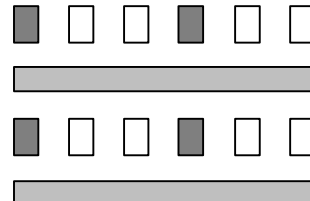
2 Layer  
 $W + S < 2\mu\text{m}$

1S to 1G



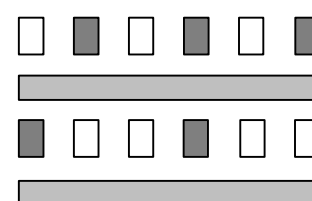
2 Layer  
 $W + S < 2\mu\text{m}$

2S to 1G



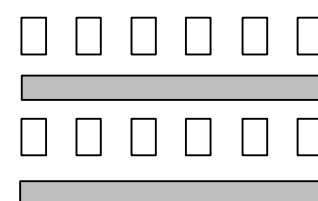
2 Layer  
 $W + S < 2.66\mu\text{m}$

Hybrid

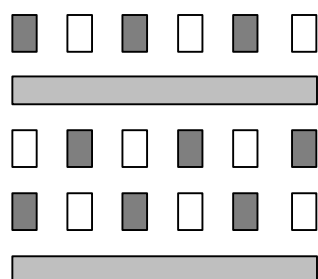


2 Layer  
 $W + S < 2.34\mu\text{m}$

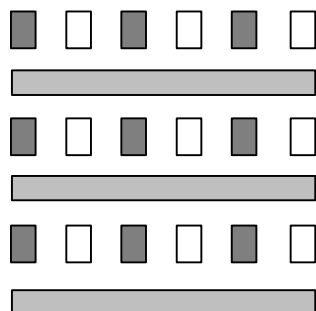
Conventional



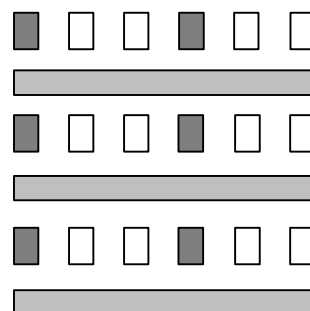
2 Layer  
 $W + S < 4\mu\text{m}$



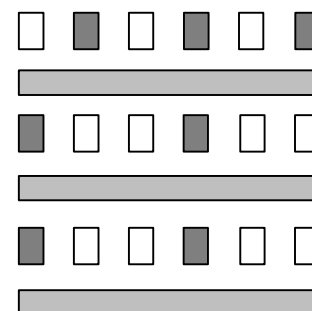
3 Layer  
 $W + S < 3\mu\text{m}$



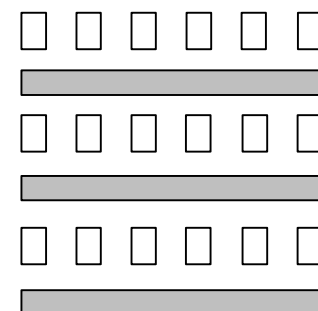
3 Layer  
 $W + S < 3\mu\text{m}$



3 Layer  
 $W + S < 4\mu\text{m}$



3 Layer  
 $W + S < 3.67\mu\text{m}$



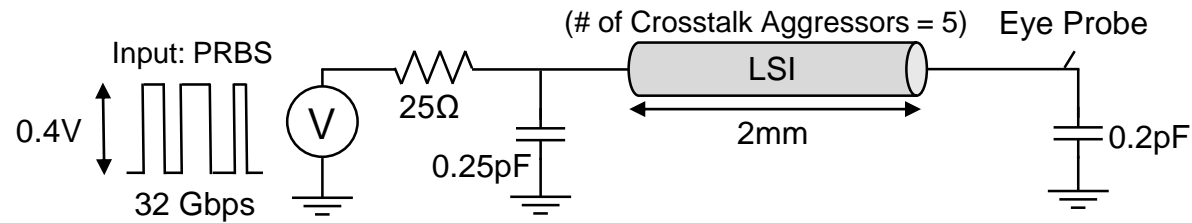
3 Layer  
 $W + S < 6\mu\text{m}$

< LSI stack-up considering routability and signal integrity >

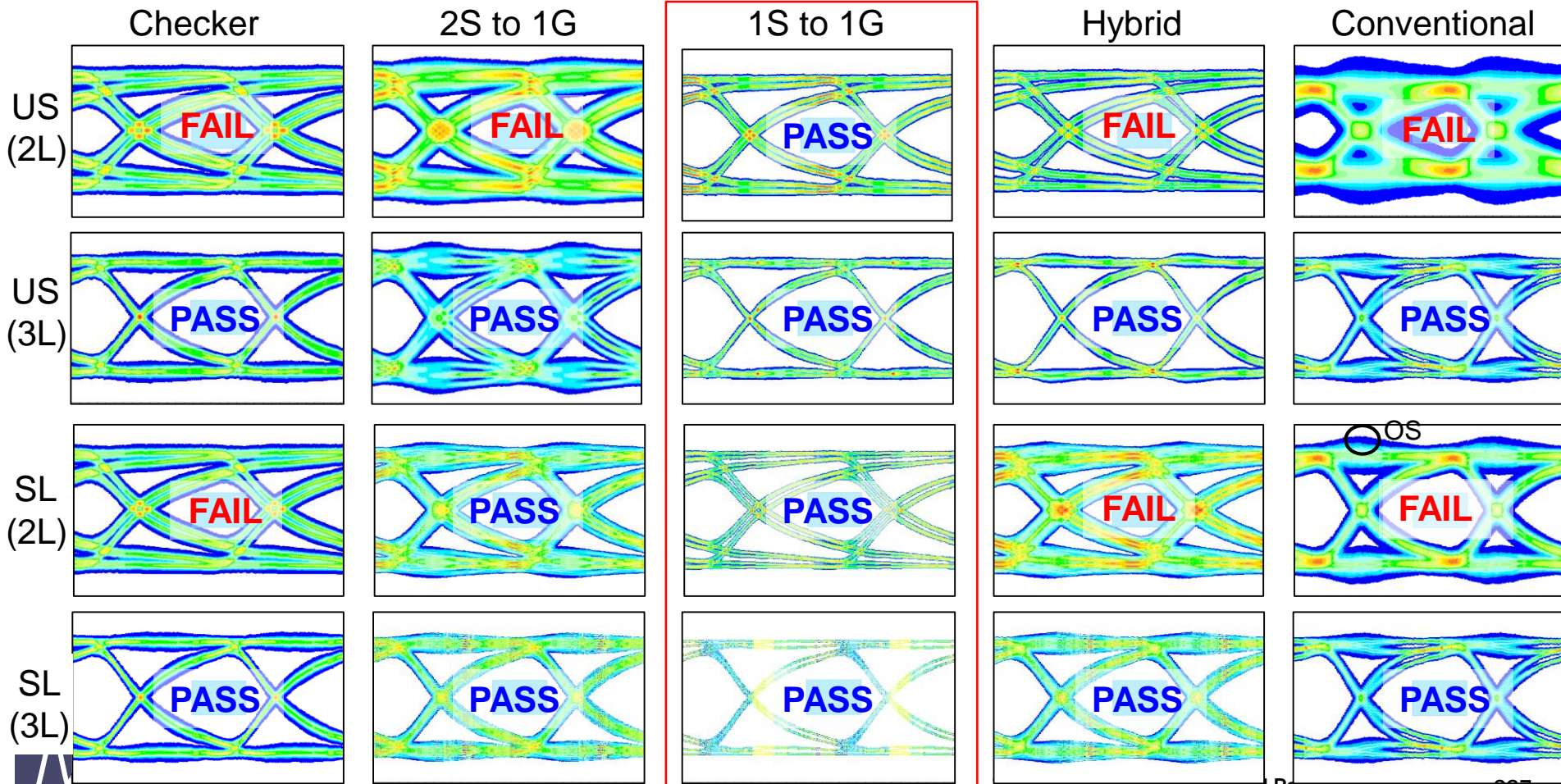
- To verify signal integrity, various stack-ups of LSI is analyzed.
- The sum of width and space is chosen under routability constraint.



# Signal Integrity Analysis: Eye Diagram of Various LSI Stack-ups



Eye mask:  
120mV, 0.6 UI,  
Overshoot  $\pm 0.1V$



- For 2 layer, only 1 to 1 passes, and for 3 layers, all cases pass the eye diagram test.

# Thank You!

## HBM

# HBM6 Cluster Architecture with Crossbar Network Switch for High Throughput and Low Latency LLM Inference

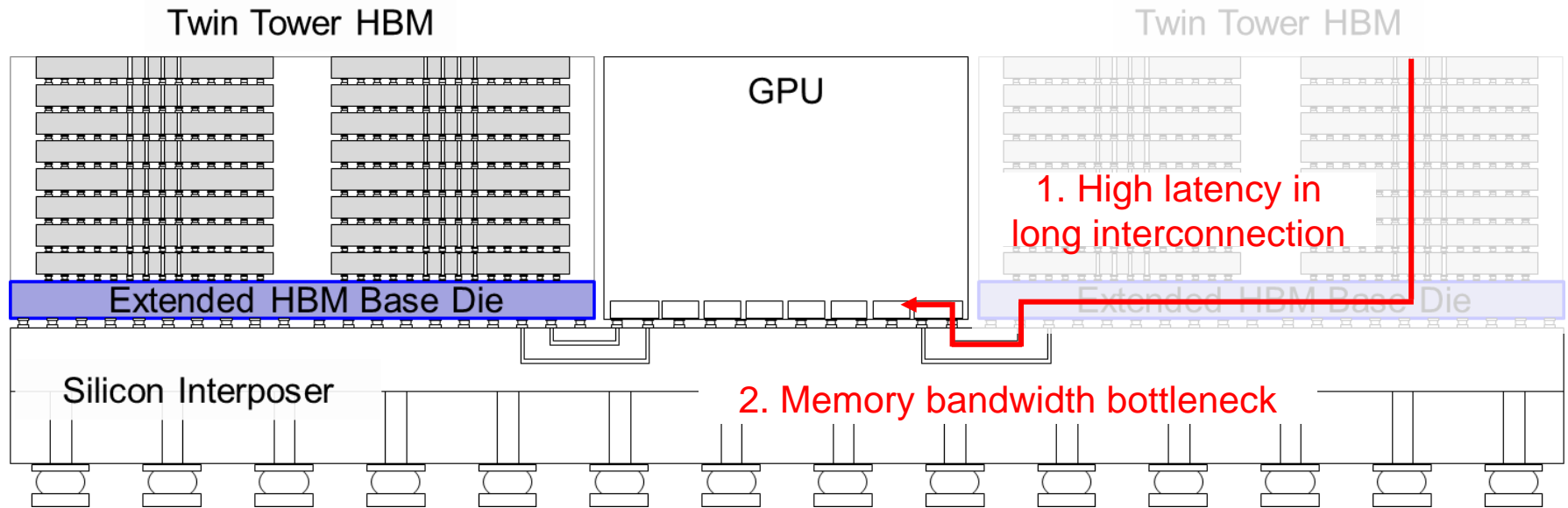
**Youngsu Yoon**

Advising Professor : Prof. Jounggho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering, KAIST

June. 11<sup>th</sup>, 2025

# Predictive Problem in HBM6 Generation

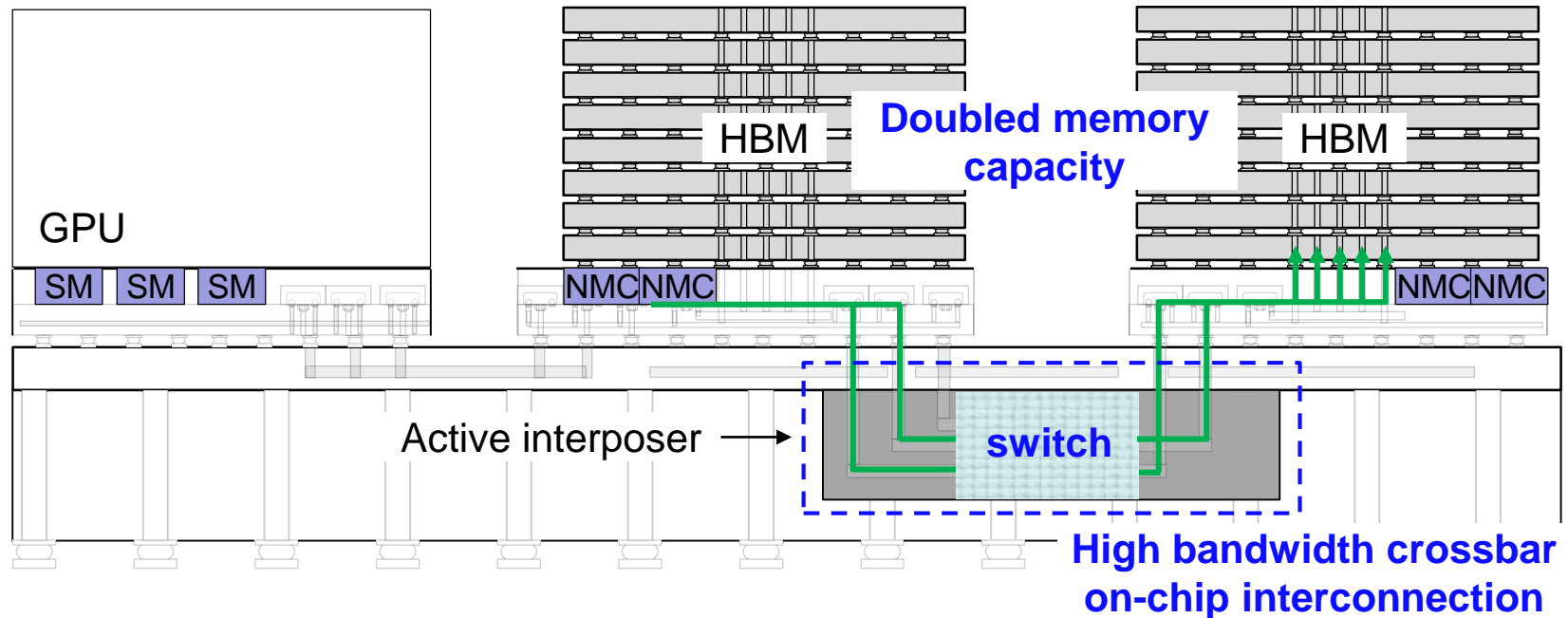


## < Next-Generation HBM6 Architecture based Research Roadmap by Teralab >

- In the HBM6, memory capacity is doubled to save more matrix data of attention mechanism.
- However, high latency is inevitable in the process of exchanging data with HBMs which are in the farthest away from the GPU.
- It is still vulnerable to memory intensive workload due to the memory bandwidth bottleneck.



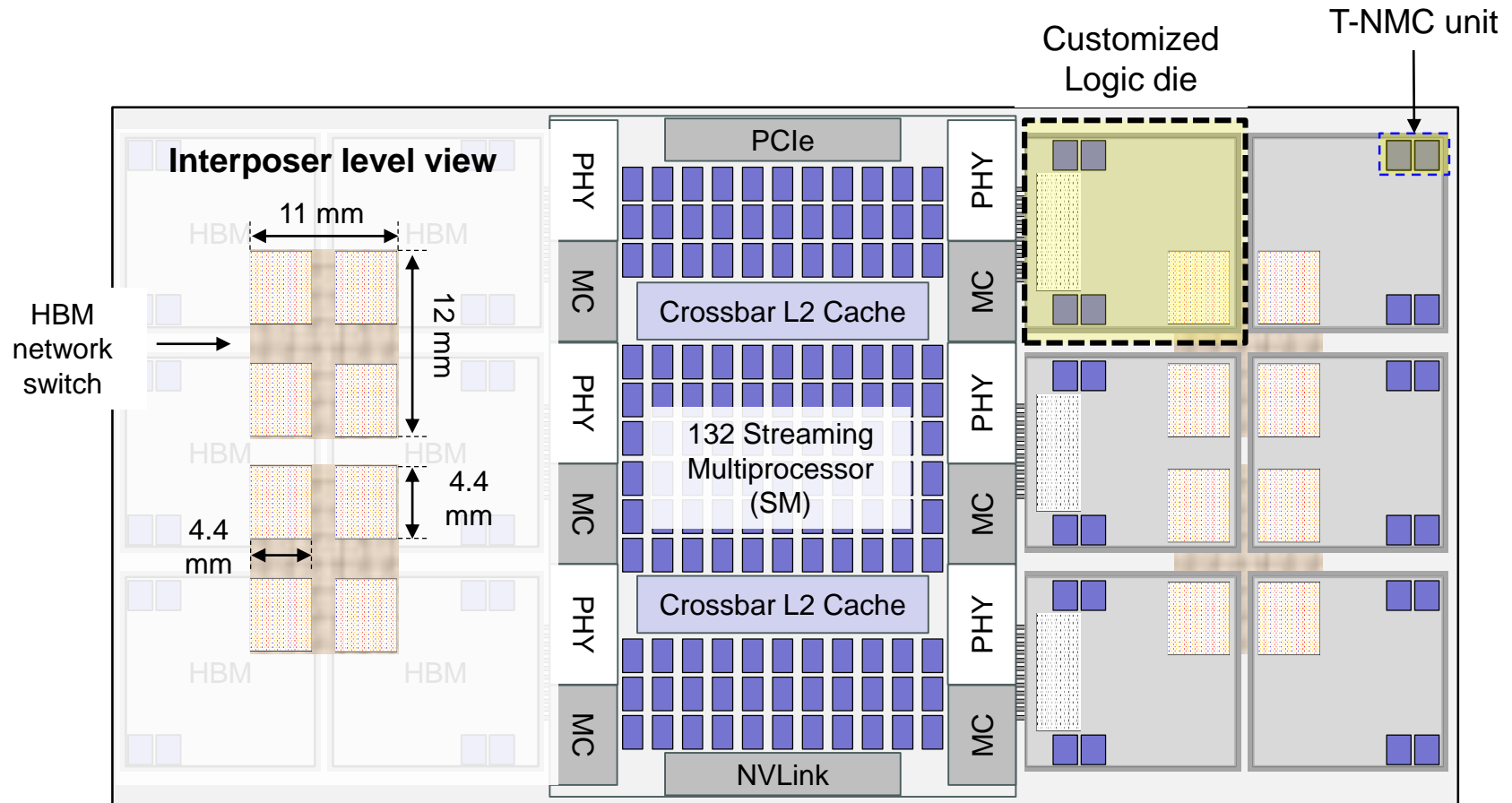
# Main Concept of the Proposed High-Bandwidth Memory Cluster Architecture with HBM Network Switch



< Side view of HBM cluster architecture with HBM network switch >

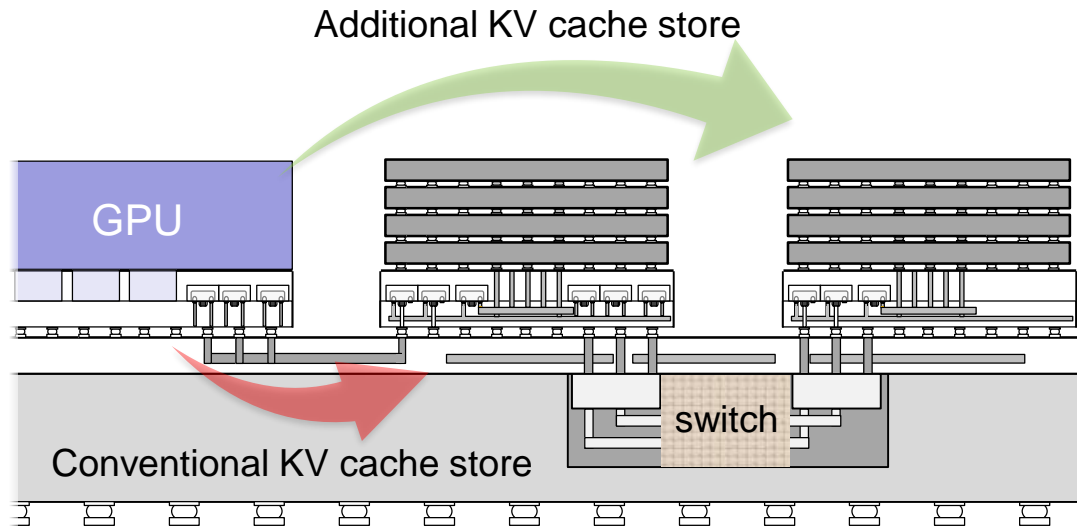
- The proposed HBM cluster architecture integrated crossbar type HBM network switch in the active interposer level to provide HBM to HBM communication.
- For the HBM network switch, customized logic die is applied to reallocate the location of HBM elements.
- HBM network switch is connected to 4 HBM's corner PHY and designed to provide 9.6 TB/s in maximum.

# Floorplan of the HBM Cluster Architecture with HBM Network Switch



< Floorplan of the proposed HBM cluster with GH100 GPU Architecture >

# Key Advantages of HBM Cluster Architecture [1/3] : Increased Memory Capacity

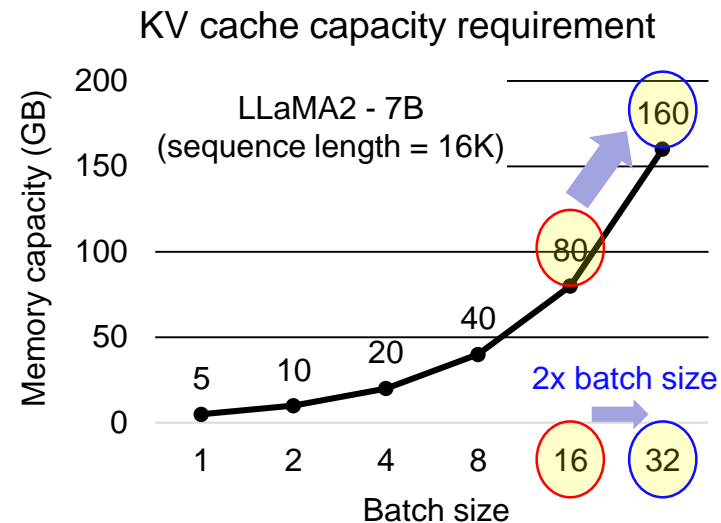


< Memory capacity increase by additional HBMs >

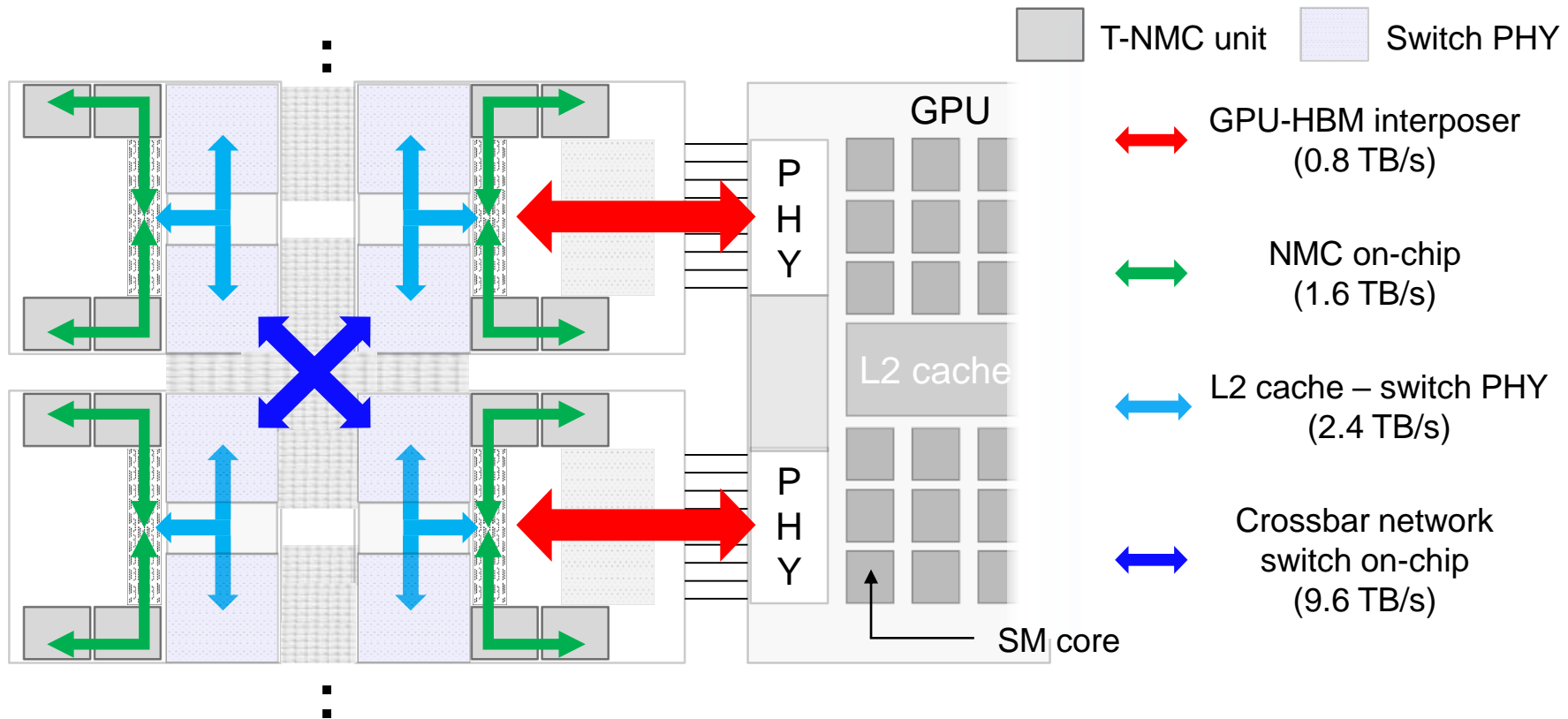
- By extending the number of HBMs, proposed HBM cluster architecture can store twice larger KV cache in HBMs.
- For example, the LLaMA2-7B model requires the memory capacity required to store KV cache based on the following formula.

$$KV\ Cache\ capacity_{LLaMA2-7B} = 0.52\ MB \times batch\ size \times sequence\ length$$

- Therefore, the batch size can be doubled at the same sequence length with our HBM cluster architecture.



# Key Advantages of HBM Cluster Architecture [2/3] : High Bandwidth Utilization of Crossbar type HBM Network Switch

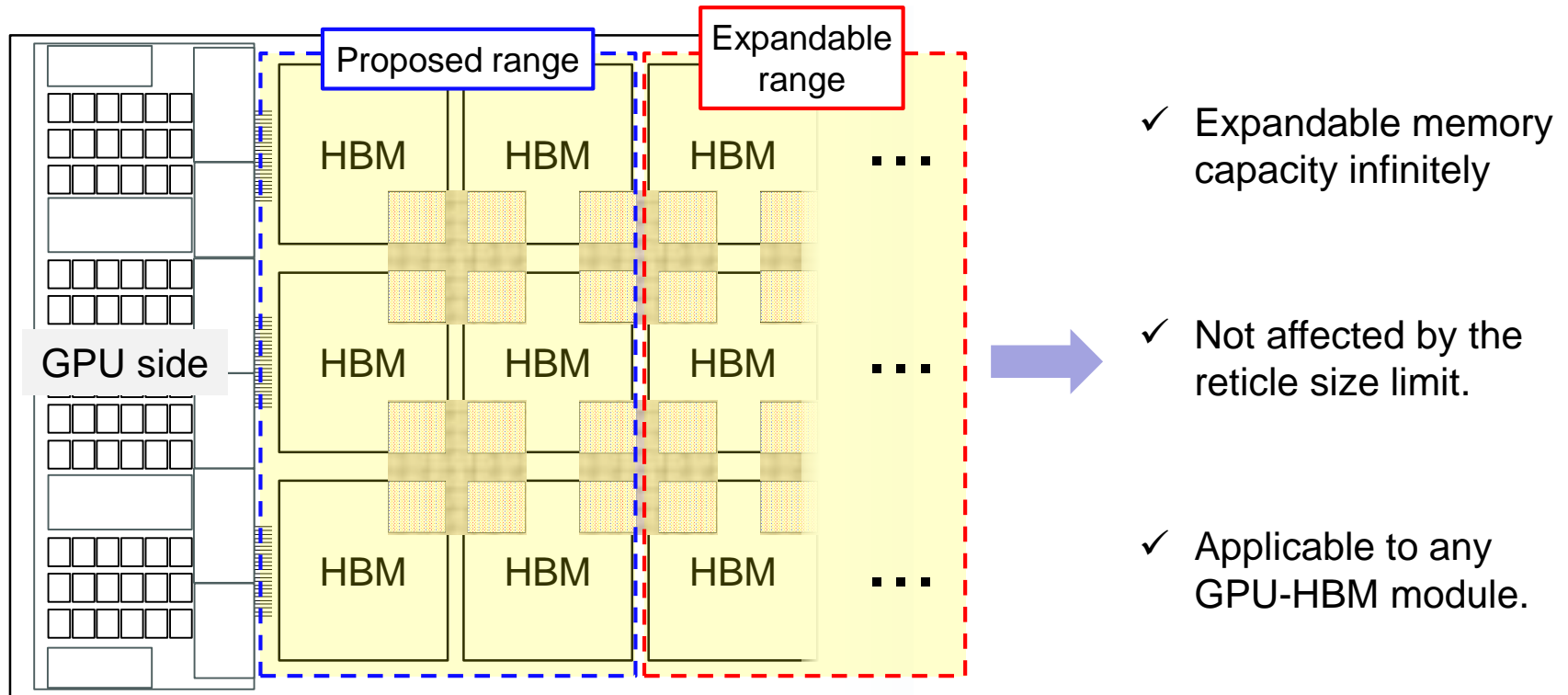


< High bandwidth data movement from HBM network switch >

- In our proposed architecture, GPU-HBM channel and NMC on-chip channel totally requires 2.4 TB/s.
- However, conventional HBM's TSV only provides 0.8 TB/s → **Bandwidth bottleneck**
- Our proposed HBM network switch provides 2.4 TB/s to each HBM and communicates with other HBM network switch by 9.6 TB/s.



# Key Advantages of HBM Cluster Architecture [3/3] : Scalability

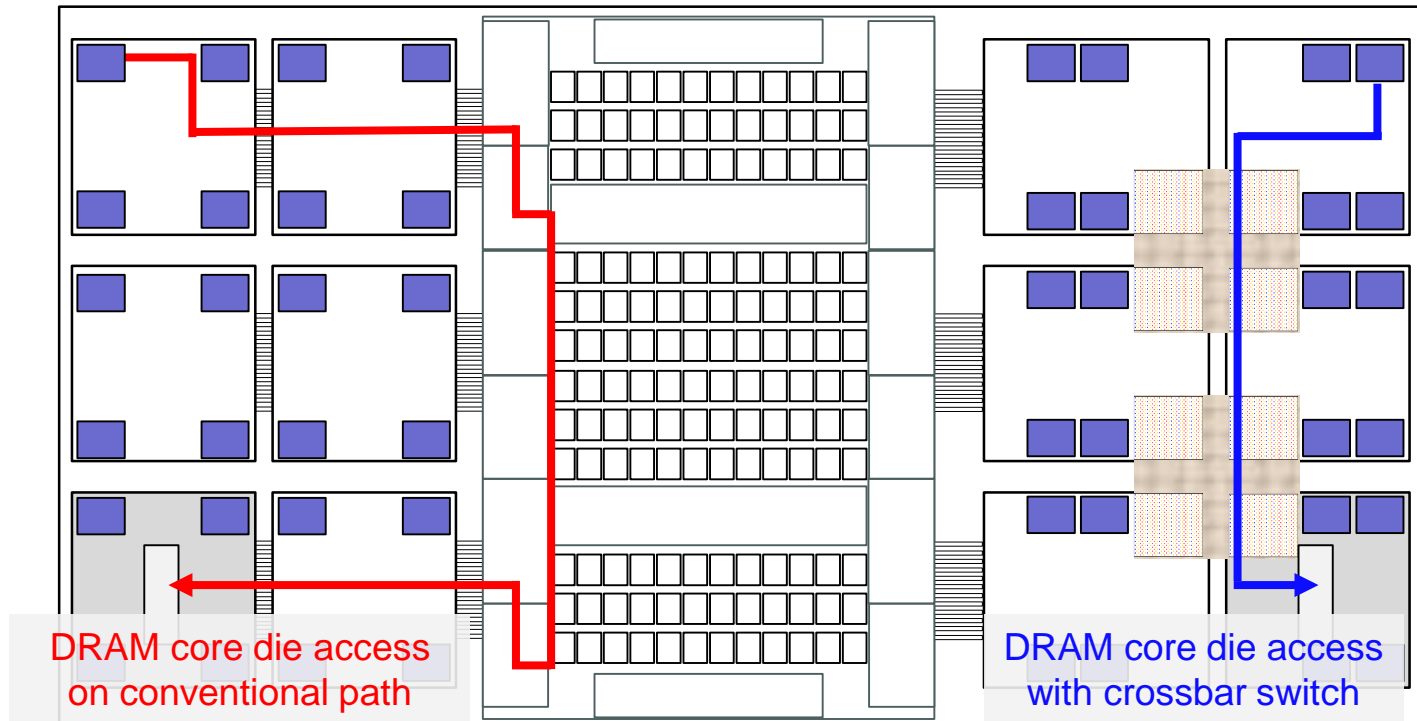


## < Expansion scalability of HBM network switch >

- Our HBM network switch connect 4 HBMs in each corner PHY.
- Therefore, HBM network switch can be applied in any GPU-HBM module architecture to expand memory capacity. → [High scalability](#)

# Data Path Analysis of HBM Cluster Architecture in terms of HBM network switch usage

- HBM core die access w/o HBM network switch
- HBM core die access w HBM network switch

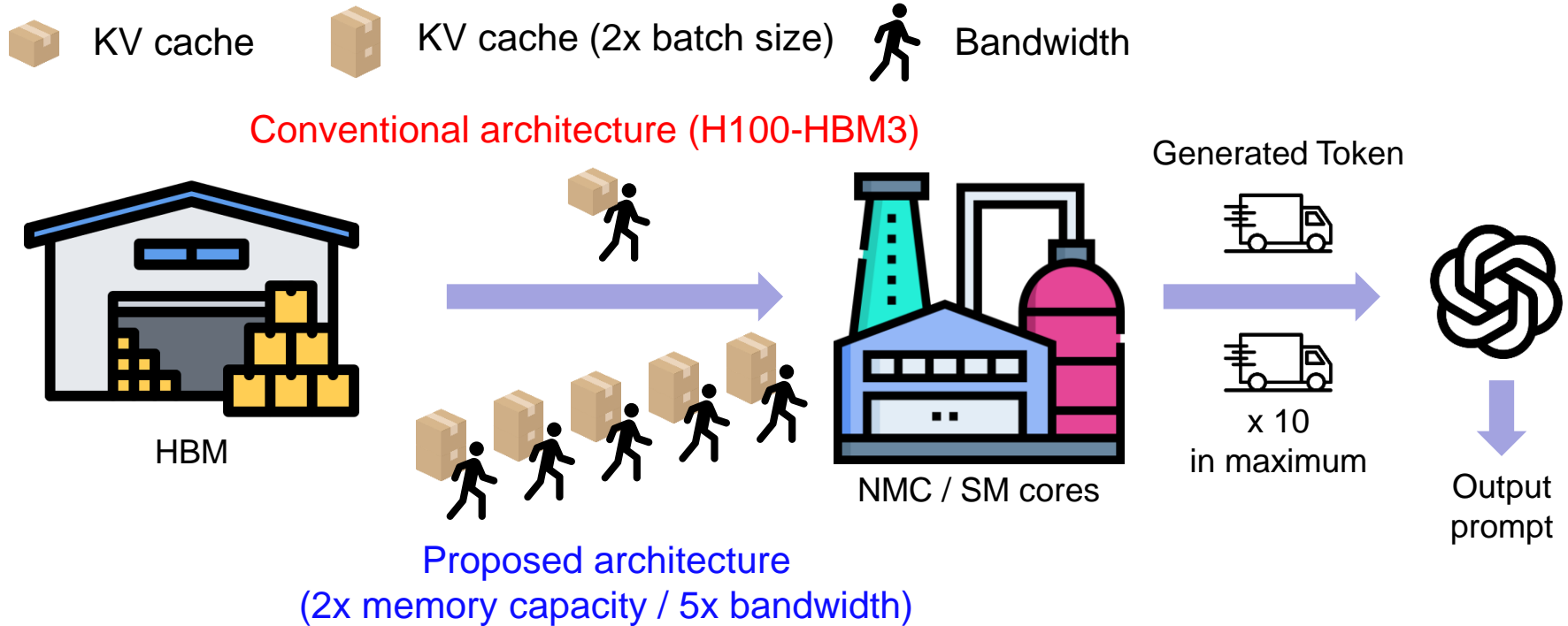


< Dram core die access path comparison in terms of HBM network switch usage >

Scenarios	Other HBM core die access when L2 cache miss	
	Without switch	With switch
Interconnect length	16 mm (interposer) + 75.1mm (on-chip) = 91.1 mm	44.92 mm (on-chip)

51% reduce

# Performance Analysis of HBM Cluster Architecture : Increased Token Throughput



< Token generation process comparison of two architectures >

- In the ideal condition (perfect memory bottleneck, bandwidth utilization and no overhead), proposed HBM cluster architecture can increase throughput by up to **10** times.

$$B_{max} = \frac{MC}{SL \times \alpha} \quad t_{token} = \frac{\alpha}{BW} \quad R = \frac{B_{max}}{t_{token}} \propto (BW \times MC) \quad \frac{R_{proposed}}{R_{convent}} = \frac{5BW \times 2MC}{BW \times MC} = \mathbf{10}$$

B : batch size

MC : memory capacity

$\alpha$  : KV cache utilization per token

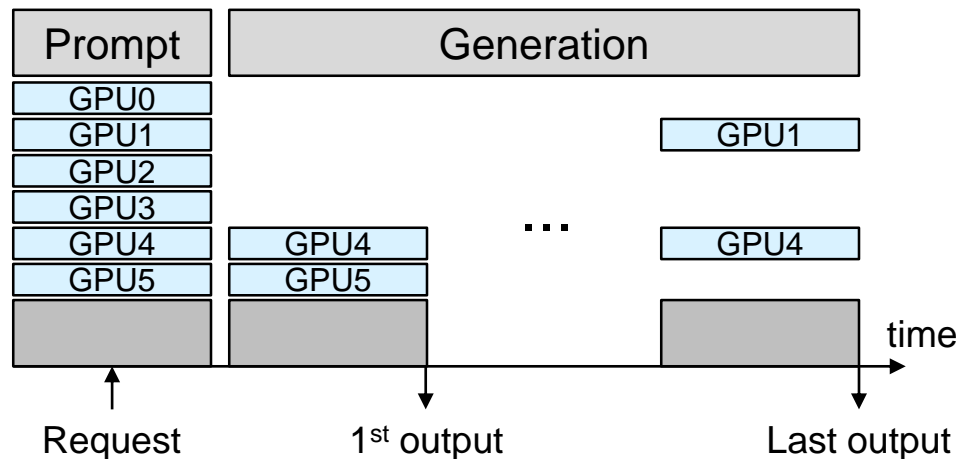
t : token generation time

SL : sequence length

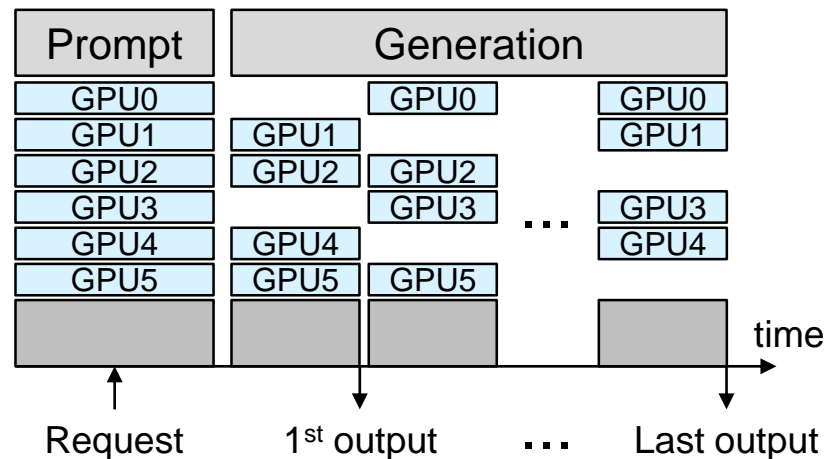
R : token throughput

# Detailed Token Throughput Analysis in Future LLM Service

\* Future AI model assumption : 20 Trillion



→ **Low GPU utilization** in conventional architecture



→ **High GPU utilization** in HBM6 cluster architecture

$$TPOT = 2 \text{ Byte} \times (\# \text{ of model parameter}) \times \max\left(\frac{1}{BW}, \frac{1}{FLOPS}\right) \rightarrow \text{Memory bound}$$

$$\text{Throughput (Token per second)} \approx \frac{1}{TPOT} = \frac{\text{Total memory bandwidth}}{2 \text{ Byte} \times (\# \text{ of model parameter})}$$

AI Rack scale performance	GPU-HBM module bandwidth	TPOT	Token per second @ single GPU	Token per second @ Rack scale
Conventional H200 NVL72	4.8 TB/s	115.74 ms	0.12 Token/sec	8.64 Token/sec
Proposed HBM Cluster NVL 72	24 TB/s	23.15 ms	0.6 Token/sec	43.2 Token/sec

# Conclusion

- In the HBM6, memory capacity is doubled to save more matrix data of attention mechanism.
- However, high latency and low memory bandwidth are still bottleneck of the GPU-HBM module.
- Therefore, we propose a high-bandwidth memory cluster architecture with HBM network switch.
- Our HBM network switch connect every HBM logic die to reduce interconnect length and provide high bandwidth for near memory computing cores.
- In the ideal condition, our proposed architecture can generate 10x larger tokens.

# Thank You!

## HBM

# HBM6-Centric Network Design under Traffic Asymmetry in Heterogeneous Module-Based Systems

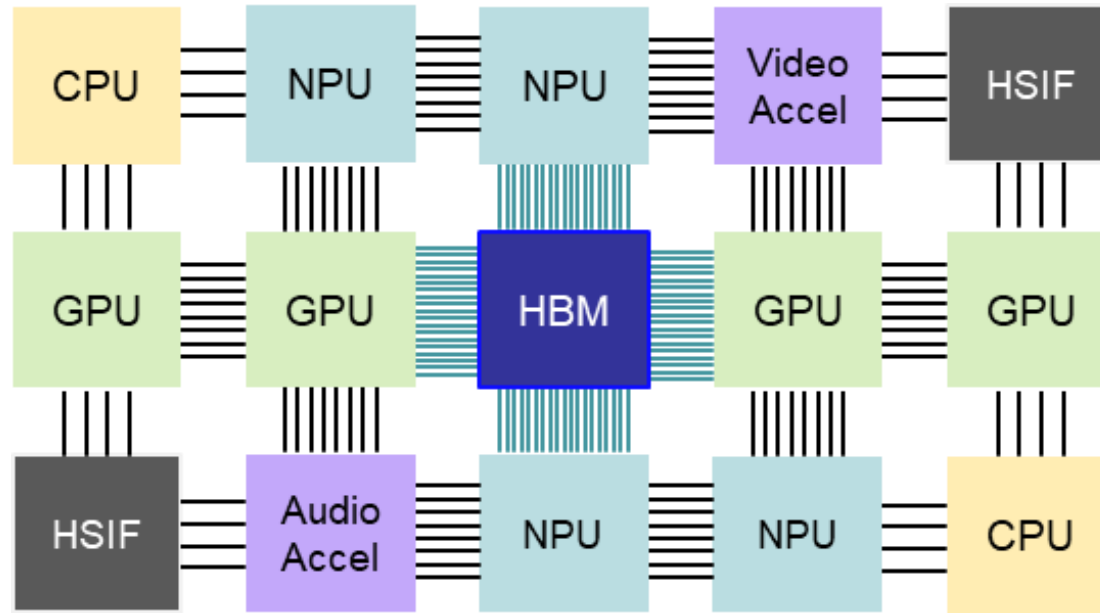
**Hyowon An**

Advising Professor : Prof. Jounggho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering, KAIST

June 11<sup>th</sup>, 2025

# Beyond HBM Enables Direct Access to Shared Memory Pool: Sharable HBM in Heterogeneous Chiplet System

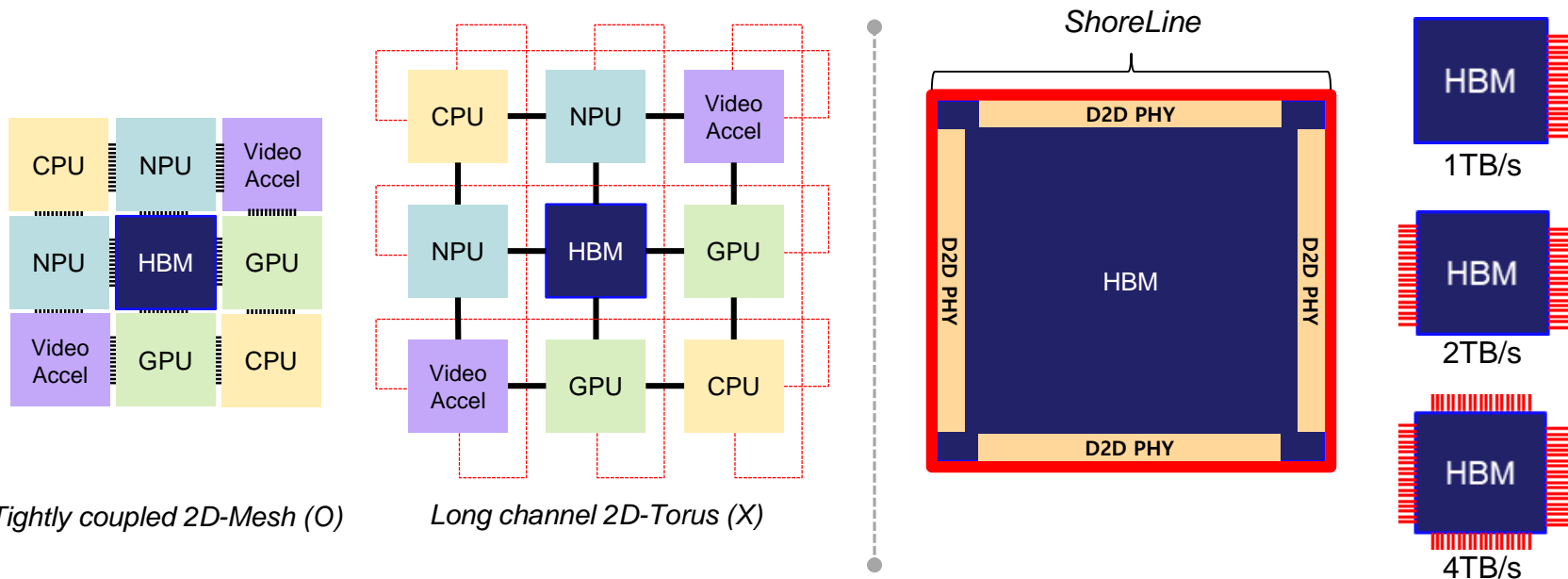


< Centric HBM6 in Heterogeneous Module Based Chiplet System >

- GPU-HBM fits large-scale AI in datacenters—but not small edge devices.
- In compact devices like AR glasses, GPU drawbacks (size, cost) dominate.
- Systems will likely evolve toward a modular chiplet-based architecture, where each function is implemented as a specialized module—similar to AP SoCs.
- Congestion occurs near HBM nodes due to the many-to-few traffic pattern.
- Each heterogeneous node has distinct bandwidth requirements toward the HBM.



# Interconnection Network Constraints of Module based Chiplet System including Shared HBM: Physical View



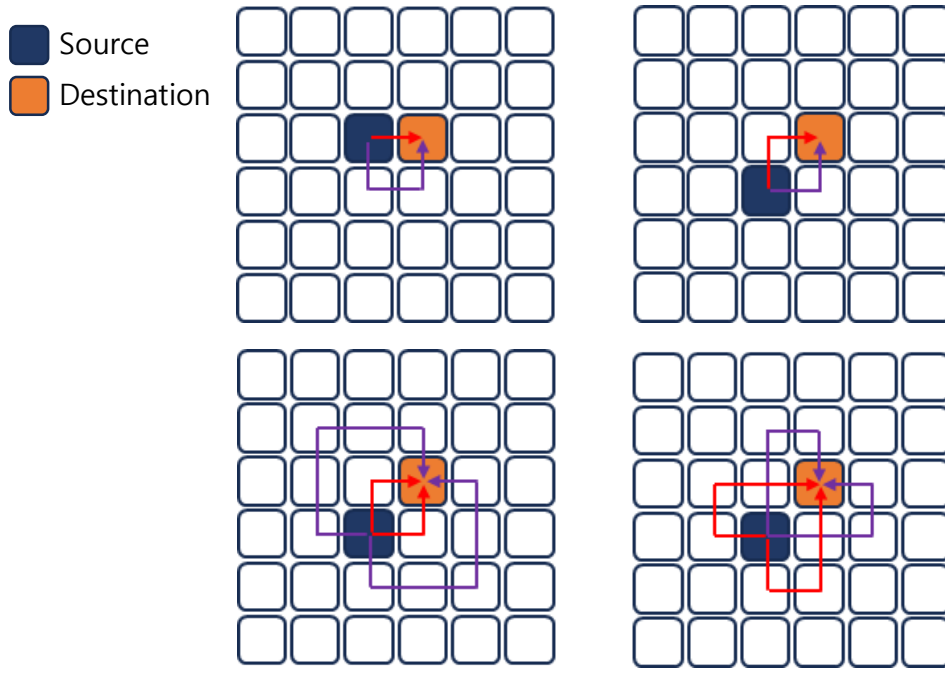
< Topology for High-Bandwidth Chiplet >

< ShoreLine Limitation >

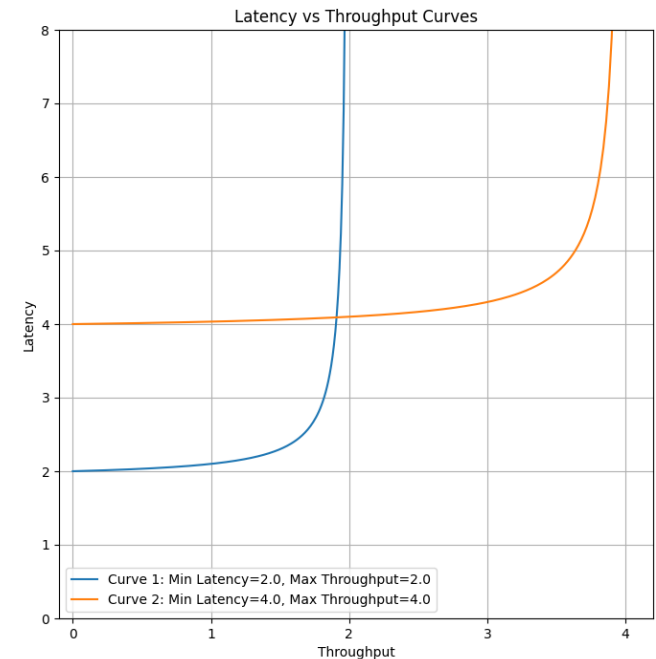
- Signal Integrity Challenges Hinder Topology Design: 2D-Mesh
- 3D ICs concentrate bandwidth at base-die edges — but shoreline I/O is limited.
- Reticle/Shoreline constraints limit I/O bandwidth; achieving higher throughput requires multi-directional edge interconnects.

$$\{Length\ of\ ShoreLine\} \propto \{Number\ of\ Pad\ on\ Edge\} \propto \{Bandwidth\}$$

# Example Issue Study of Detoured Routing by Shoreline Limitation



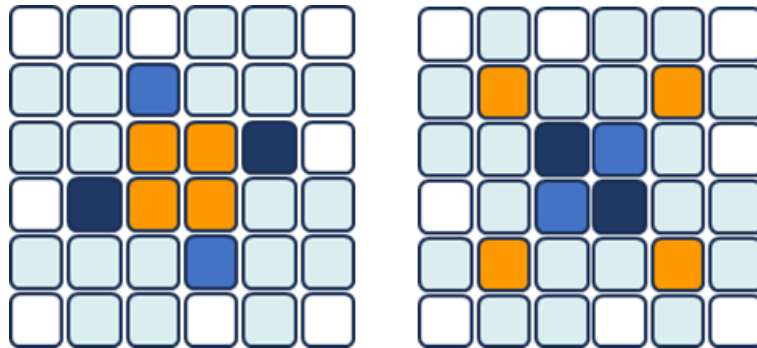
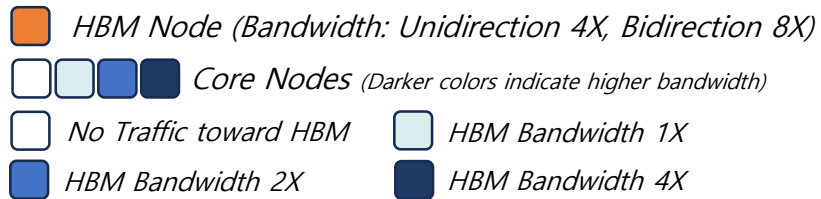
< Simple Example between 2 node >



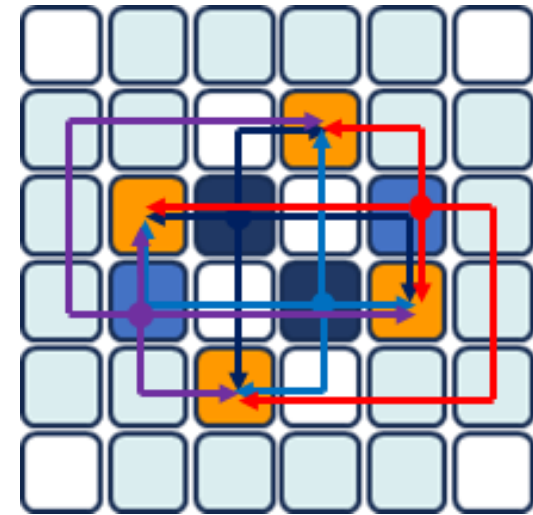
< Latency-Throughput Curve >

- Case 1. Find Best Placement considering required bandwidth and directions of traffic
- Case 2. Find Best Routing Strategy considering required bandwidth and directions of traffic
- Case 3. Find Proper Adaptive Routing condition based on Latency-Throughput Curve

# Practical Example including HBM6



< Practical Case Example >

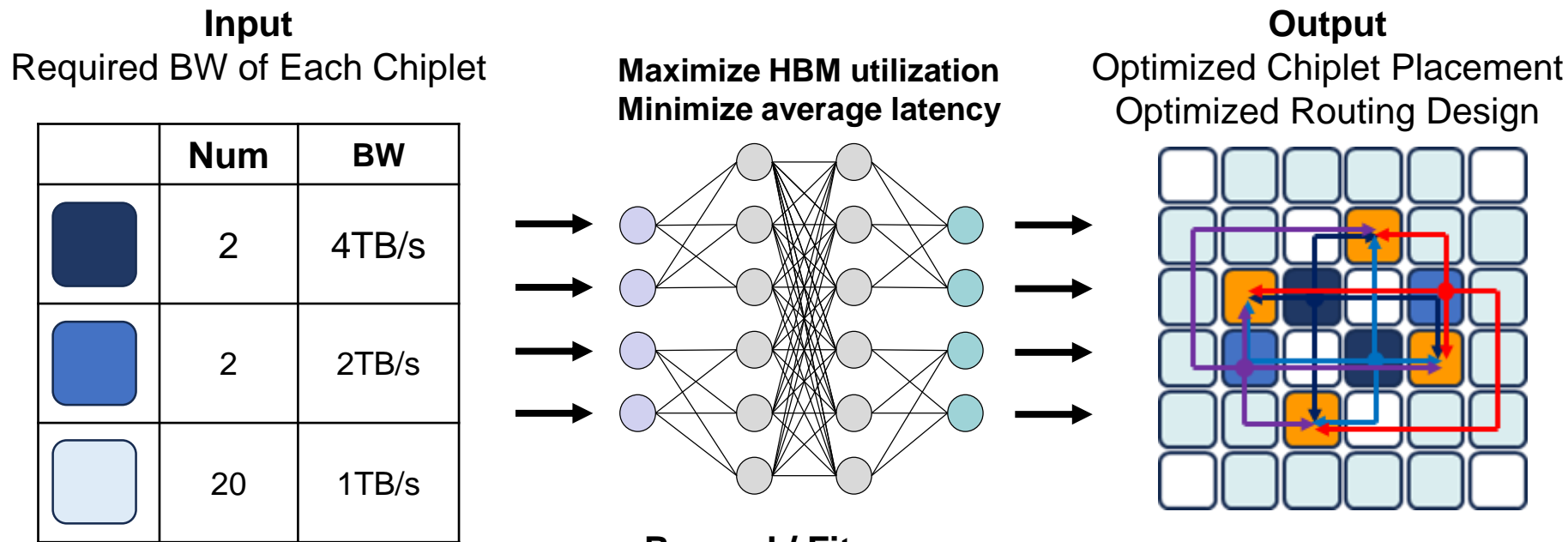


< Potential Solution >

- Rule 1. HBM nodes must not be placed adjacent to each other.
- Rule 2. Avoid placing heavy-bandwidth source nodes adjacent to each other.
- Rule 3. Heavy-bandwidth source should be positioned near the center of the HBM nodes.

→ Objective: Minimize average hop count (latency), Maximize HBM channel utilization

# HBM-Centric Optimization of Chiplet Placement and Interconnect Design



**Reward / Fitness**

$$\omega1 \cdot \text{Routing Ratio} - \omega2 \cdot \text{AvgHop} - \omega3 \cdot \text{MaxHop}$$

- Step 1: (GA Outer Loop) Place chiplets based on Ground Rule
- Step 2: (GA Inner Loop) Find Optimal Routing Considering QoS, Deadlock Free
- Step 3: Calculate Average Hop-Count of Each Node
- Step 4: Simulation under Various Traffic Condition

→ The iterative loop between Step 1 to Step 3 results in a long design cycle, which can be significantly shortened through an ML-based approach.



# HBM CENTRIC

# Thank you!

# Conditional Diffusion Model-based Imitation Learning for HBM6 Placement and Interconnection Optimization

Jihun Kim

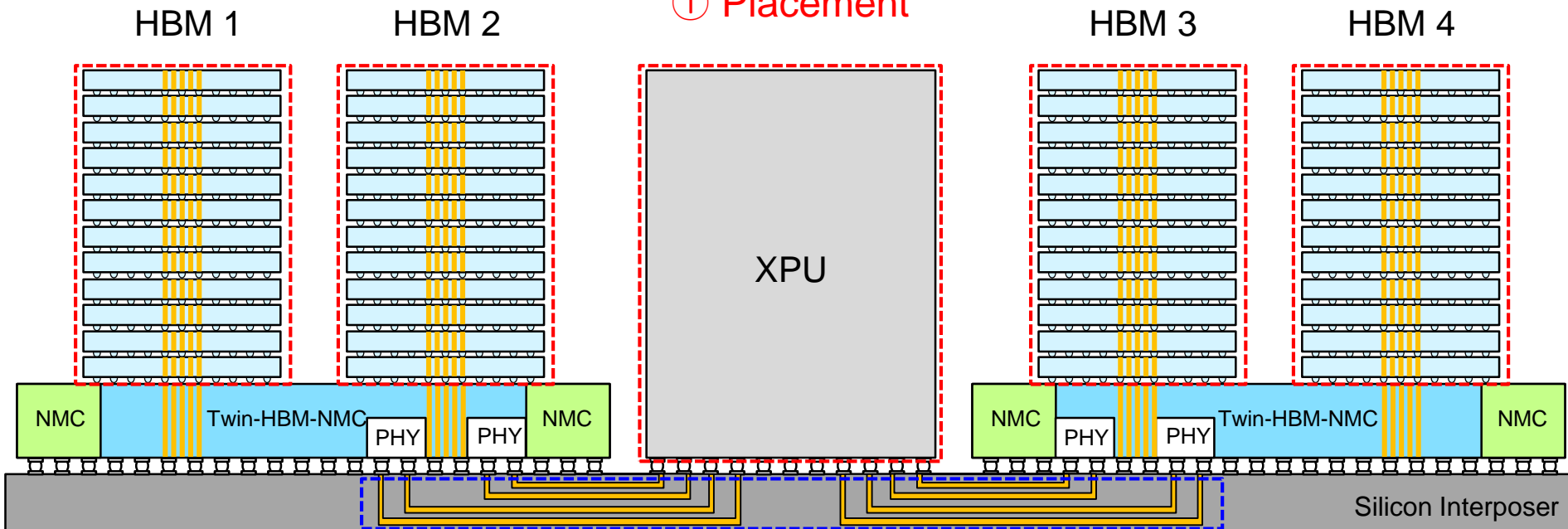
Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

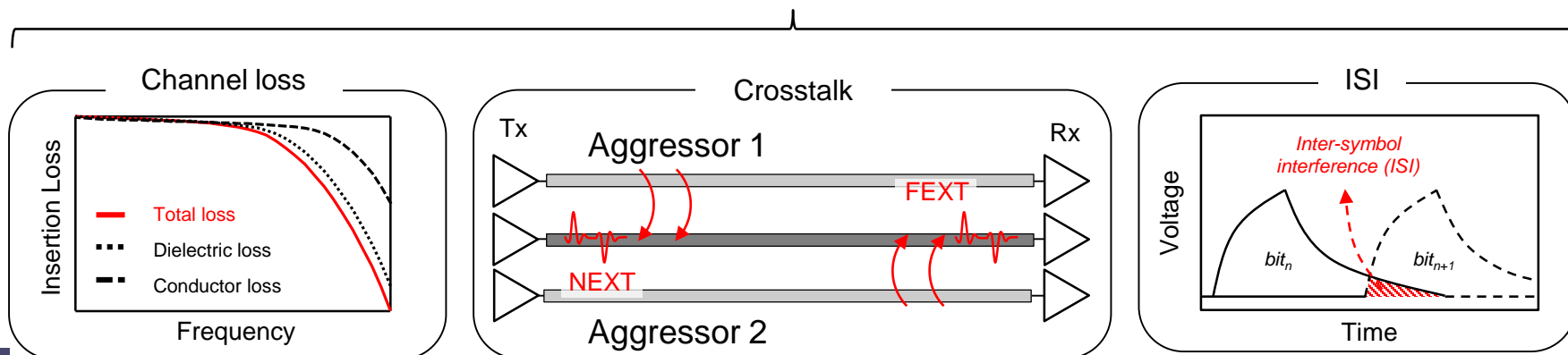
2025. 06. 11

# Signal Integrity Problem on HBM6 Generation

## ① Placement



## ② Interconnection



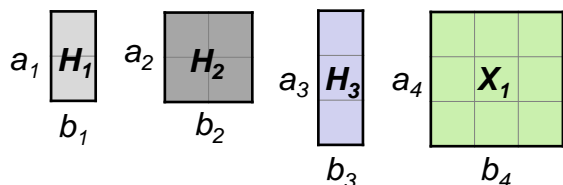
< Signal integrity problem on HBM6 Generation >



# AI Agent-based HBM6 Placement and Interconnection Optimization Solver

## Input:

XPU & HBM6 /  
Interconnections



HBM6 & XPU

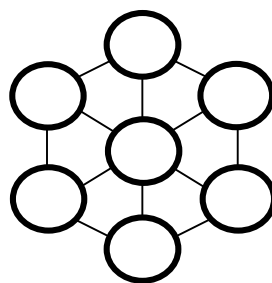
$i_1 = \{h_1, h_3, 16Gbps\}$

$i_2 = \{h_1, x_1, 48Gbps\}$

$i_3 = \{h_2, h_3, 16Gbps\}$

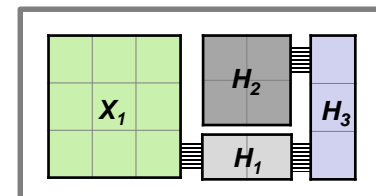
Interconnections

Fast and Optimal Solver  
based on AI Agent



## Output

Optimized  
XPU & HBM6 Design



$(h_1, h_3): \{Channel_1, EQ_1, Term_1\}$

$(h_1, x_1): \{Channel_2, EQ_2, Term_2\}$

$(h_2, h_3): \{Channel_3, EQ_3, Term_3\}$

Optimized Interconnections

< AI Agent-based HBM6 placement and interconnection optimization >

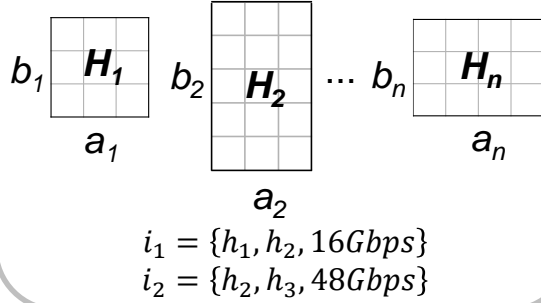
- AI agent-based HBM6 placement and interconnection optimization is necessary to reduce the design cycle considering signal integrity.
- By training the AI in terms of the relationship among SI performance, and the HBM6 placement and interconnection solution, the SI design guide can be extracted fast with inference of trained neural network.



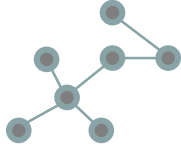
# Proposal of Conditional Diffusion Model-based Imitation Learning for HBM6 Placement and Interconnection Optimization

## Stage 1: Placement

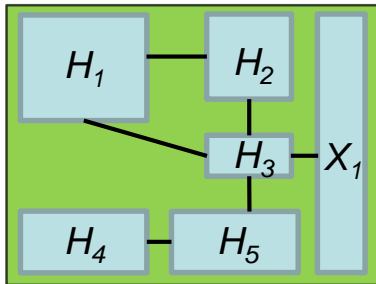
HBM6s & XPU / Interconnections



HBM Placement Algorithm

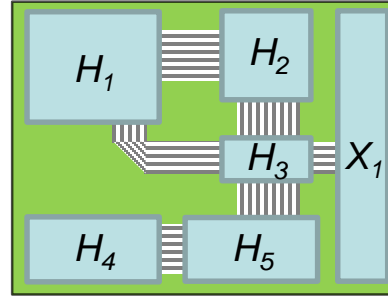


Optimized XPU & HBM6s  
**Placement**

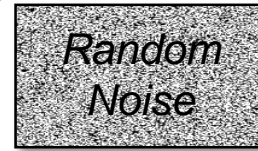


State

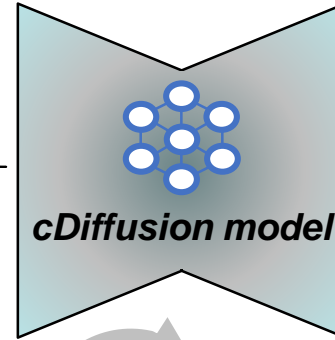
## Stage 2: Interconnection with Tx/Rx



Optimized HBM & XPU  
**placement / interconnection**  
design with driver condition



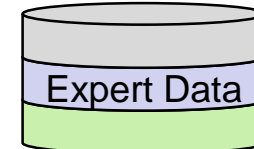
Condition:  
HBM & XPU  
placement



Action

Optimized HBM & XPU  
**Interconnection** Design

Imitation  
Learning

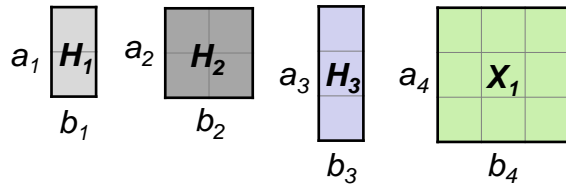


$$\text{Objective function} = \underbrace{\gamma \nabla \log p(x_t | c)}_{\text{Conditional score}} + \underbrace{(1 - \gamma) \nabla \log p(x_t)}_{\text{Unconditional score}}$$

< Detail procedure of HBM6 placement algorithm >

# Detail Procedure of HBM Placement Algorithm (HPA)

## ① HBMs & XPU's Interconnections

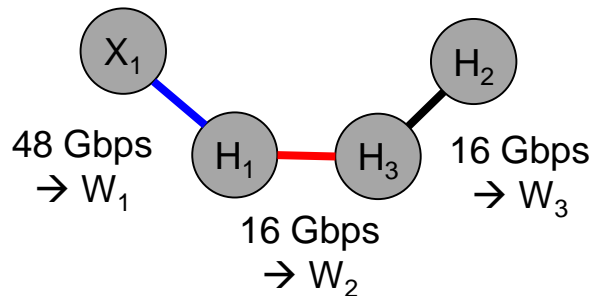


$$i_1 = \{h_1, h_3, 16\text{Gbps}\}$$

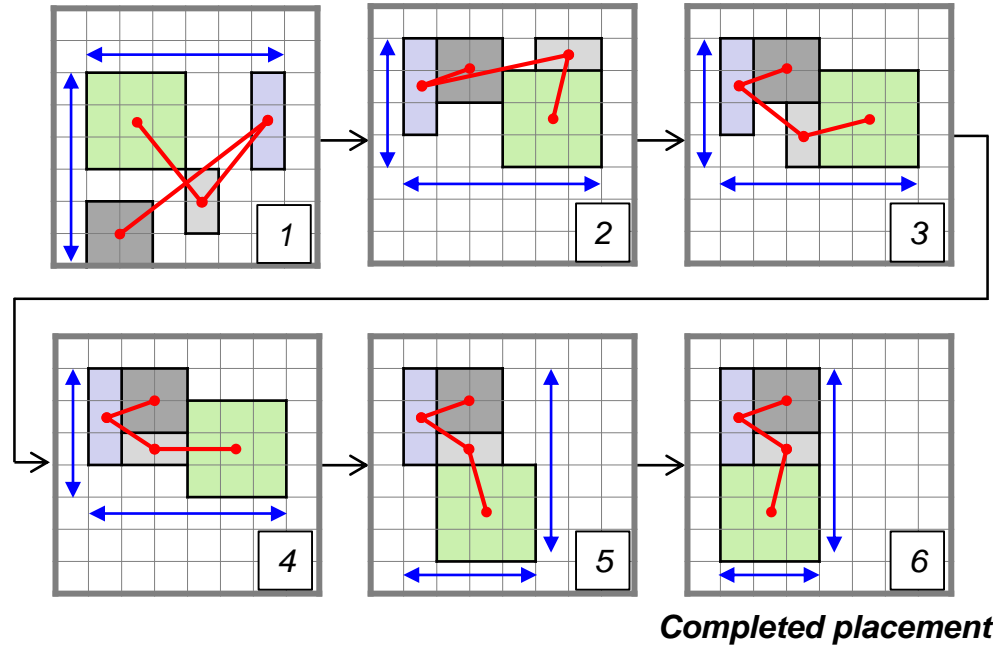
$$i_2 = \{h_1, x_1, 48\text{Gbps}\}$$

$$i_3 = \{h_2, h_3, 16\text{Gbps}\}$$

## ② Simplified Placement Order



## ③ HBMs & XPU's Placement



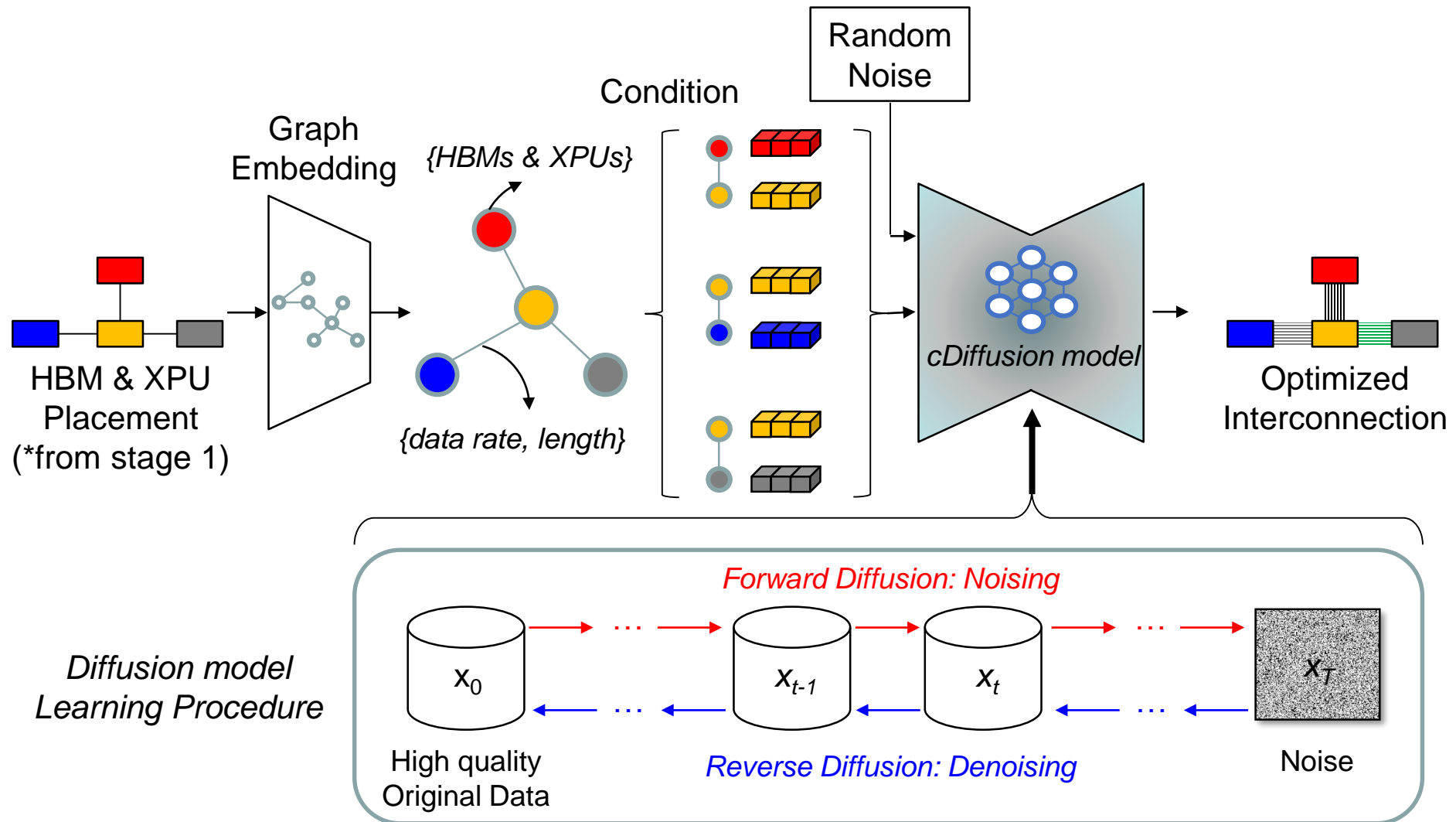
**Completed placement**

Objective function:

$$R_n^m = - \sum_{(h_a, h_b) \in \mathcal{E}} \left( \underbrace{w_{(h_a, h_b)} \times (d(h_a, h_b) + 1000 [1 - adj(h_a, h_b)])}_{\text{Length}} + \underbrace{[(x_{max} - x_{min})(y_{max} - y_{min})]}_{\text{Area}} \right)$$

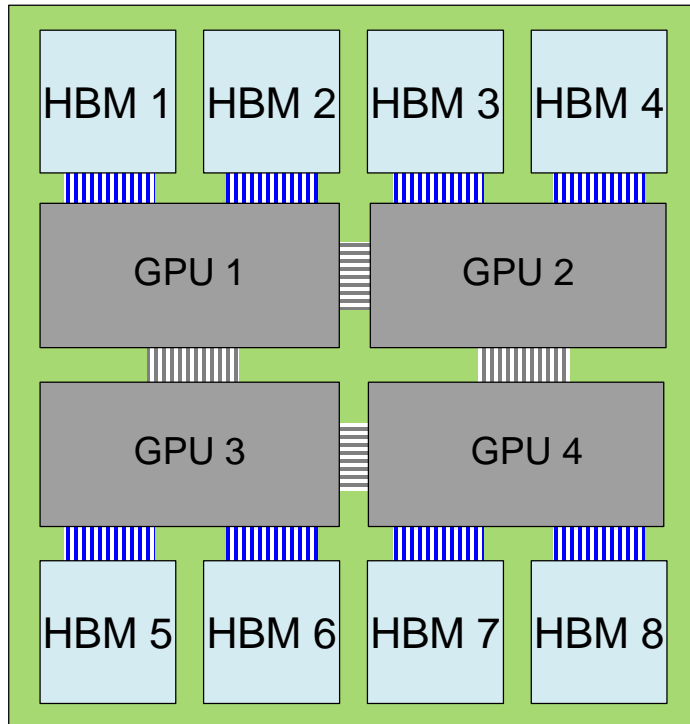
< Detail procedure of HBM placement algorithm >

# Stage 2: Conditional Diffusion Model-based HBM & XPU Interconnection Optimization



< Conditional diffusion model-based HBM interconnection optimization >

# Definition of HBM6 Placement and Interconnection Problem



Interconnections	Data rate
GPU – GPU	48 Gbps
GPU – HBM	16 Gbps

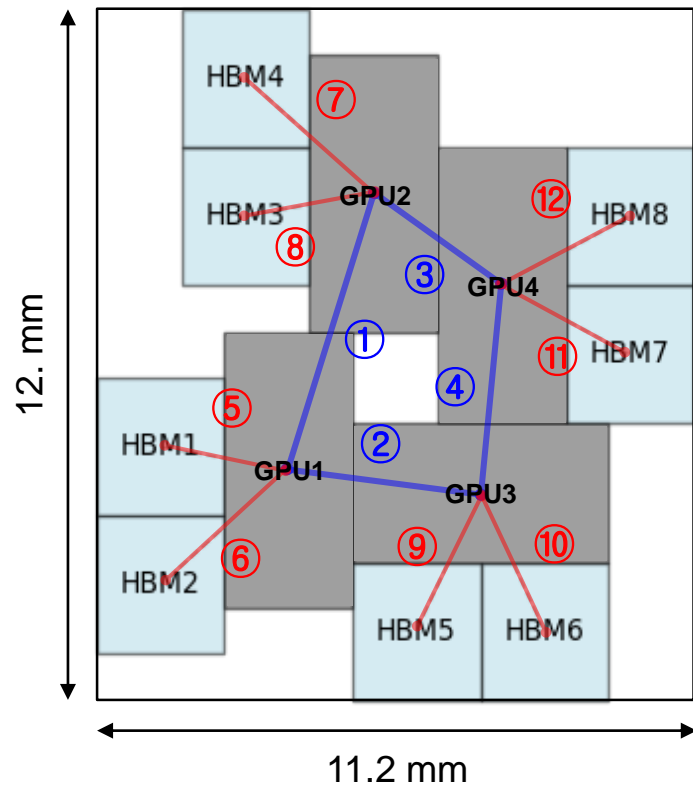
- # of I / O: 4096
  - Bandwidth: 24 TB / s
  - **Inter-GPU BW: 48 TB/s**
- 
- # of I / O: 4096
  - Bandwidth: 8 TB/s
  - **Remote HBM BW: 48 TB/s**

**$Inter\text{-}GPU\ BW \geq Remote\ HBM\ BW$**   
(Requirement of latency reduction)

< Problem instance: HBMs & XPU's / interconnections >

- A problem is defined to apply the proposed method to an AI specialized HBM6 package. (comprising 4 GPUs and 8 HBM6).
- The HBM6 and GPU sizes for placement are determined with relative dimensions.
- Data rates and the number of I/Os for each interconnection are chosen to satisfy conditions aimed at reducing GPU compute latency.

# Optimized HBM Placement Results using Proposed HPA (Stage 1)



Net ID	Interconnections	Data rate [Gbps]	Length [mm]	Cost
①	GPU1 – GPU2	48 Gbps	5.0 mm	<b>171.376</b> (Total Area 134.4 Total Length 36.976)
②	GPU1 – GPU3	48 Gbps	3.624 mm	
③	GPU2 – GPU4	48 Gbps	2.884 mm	
④	GPU3 – GPU4	48 Gbps	3.624 mm	
⑤	GPU1 – HBM1	16 Gbps	2.432 mm	
⑥	GPU1 – HBM2	16 Gbps	3.124 mm	
⑦	GPU2 – HBM3	16 Gbps	2.432 mm	
⑧	GPU2 – HBM4	16 Gbps	3.124 mm	
⑨	GPU3 – HBM5	16 Gbps	2.683 mm	
⑩	GPU3 – HBM6	16 Gbps	2.683 mm	
⑪	GPU4 – HBM7	16 Gbps	2.683 mm	
⑫	GPU4 – HBM8	16 Gbps	2.683 mm	

< Optimized HBMs & GPUs placement results >

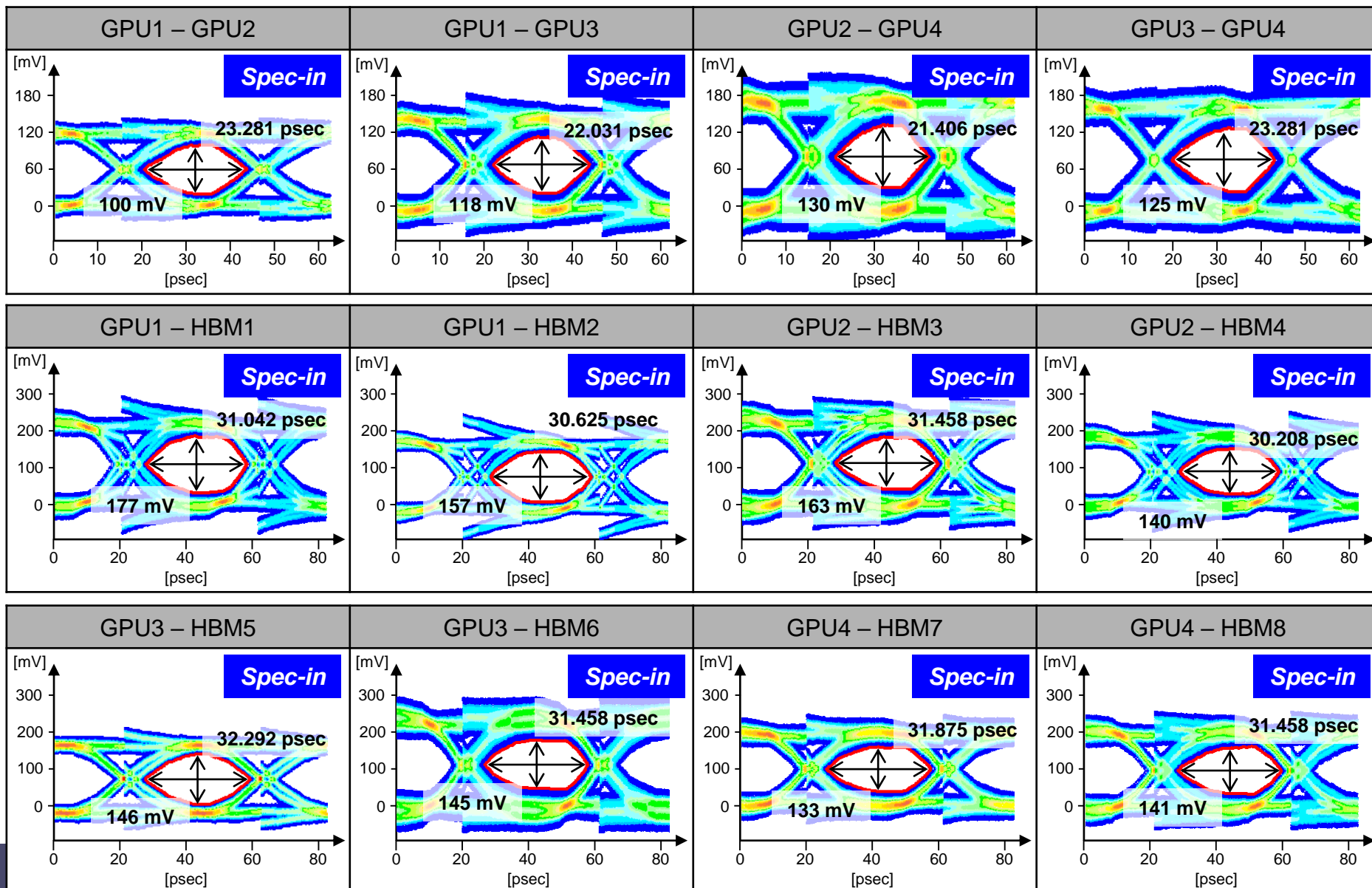
- HPA optimizes HBM and XPU placement by jointly minimizing total wire length and layout area.
  - ✓ Total Cost: 171.376 – Total layout area: 134.4 / Total wire length: 36.976
- Each GPU is placed in direct adjacency to an HBM module.
- The placed blocks do not overlap, and all connected blocks are positioned adjacently.
- The completed HBM placement is used as a condition for the stage 2 diffusion model.

# Optimized HBM Interconnection Results using Proposed Conditional Diffusion Model (Stage 1-based **Stage 2**)

	Length [mm]	W [um]	S [um]	H [um]	T [um]	C1	C-1	Tx_R [Ω]	Tx_Cap [pF]	Rx_Cap [pF]	ADC	P1 [GHz]	P2 [GHz]	P3 [GHz]	P4 [GHz]	Z1 [GHz]
G <sub>1</sub> & G <sub>2</sub> 32 Gbps	5.0	1.81	2.85	1.16	1.49	0.15	0.09	24.83	145.86	177.12	0.49	0.71	9.85	26.27	26.75	0.53
G <sub>1</sub> & G <sub>3</sub> 32 Gbps	3.624	1.55	2.92	1.03	1.25	0.22	0.06	27.82	123.2	144.49	0.54	0.73	10.32	31.3	29.27	0.45
G <sub>2</sub> & G <sub>4</sub> 32 Gbps	2.884	1.11	2.98	1.1	1.22	0.11	0.09	26.46	145.74	124.99	0.51	0.76	10.66	29.97	31.45	0.5
C <sub>3</sub> & G <sub>4</sub> 32 Gbps	3.624	1.55	2.74	1.0	1.27	0.07	0.1	25.59	137.39	119.21	0.55	0.68	11.39	26.16	32.35	0.47
G <sub>1</sub> & H <sub>1</sub> 24 Gbps	2.432	1.87	2.58	1.2	1.19	0.13	0.07	24.44	132.21	108.79	0.39	0.75	9.22	29.2	30.14	0.49
G <sub>1</sub> & H <sub>2</sub> 24 Gbps	3.124	2.08	2.74	1.44	1.07	0.13	0.06	21.08	146.43	138.7	0.27	0.78	10.55	28.31	28.87	0.5
G <sub>2</sub> & H <sub>3</sub> 24 Gbps	2.432	1.75	2.58	1.18	1.35	0.05	0.11	23.35	124.49	175.19	0.54	0.66	7.94	27.11	28.11	0.5
G <sub>2</sub> & H <sub>4</sub> 24 Gbps	3.124	1.69	2.81	1.07	1.67	0.16	0.08	26.22	132.19	104.02	0.32	0.77	10.62	28.45	29.72	0.46
G <sub>3</sub> & H <sub>5</sub> 24 Gbps	2.683	1.37	2.44	1.14	1.55	0.12	0.04	24.96	162.34	181.2	0.5	0.60	9.74	25.68	31.8	0.5
G <sub>3</sub> & H <sub>6</sub> 24 Gbps	2.683	2.25	2.34	1.21	1.26	0.06	0.12	26.62	181.25	118.92	0.52	0.7	9.19	32	31.91	0.46
G <sub>4</sub> & H <sub>7</sub> 24 Gbps	2.683	2.45	2.31	1.16	1.96	0.12	0.07	27.1	110.37	142.33	0.49	0.64	9.85	28.47	29.3	0.5
G <sub>4</sub> & H <sub>8</sub> 24 Gbps	2.683	2.26	2.27	1.0	1.51	0.09	0.09	24.14	120.61	133.08	0.46	0.69	8.67	26.64	24.32	0.5

< Optimized HBM6 interconnection results >

# Eye Diagram Verification of the Solution



< Spec-compliant all eye diagram results >



# Thank You!

## HBM



# Generative Adversarial Learning-Based Power Noise Induced Eye Diagram Estimation Agent for HBM6

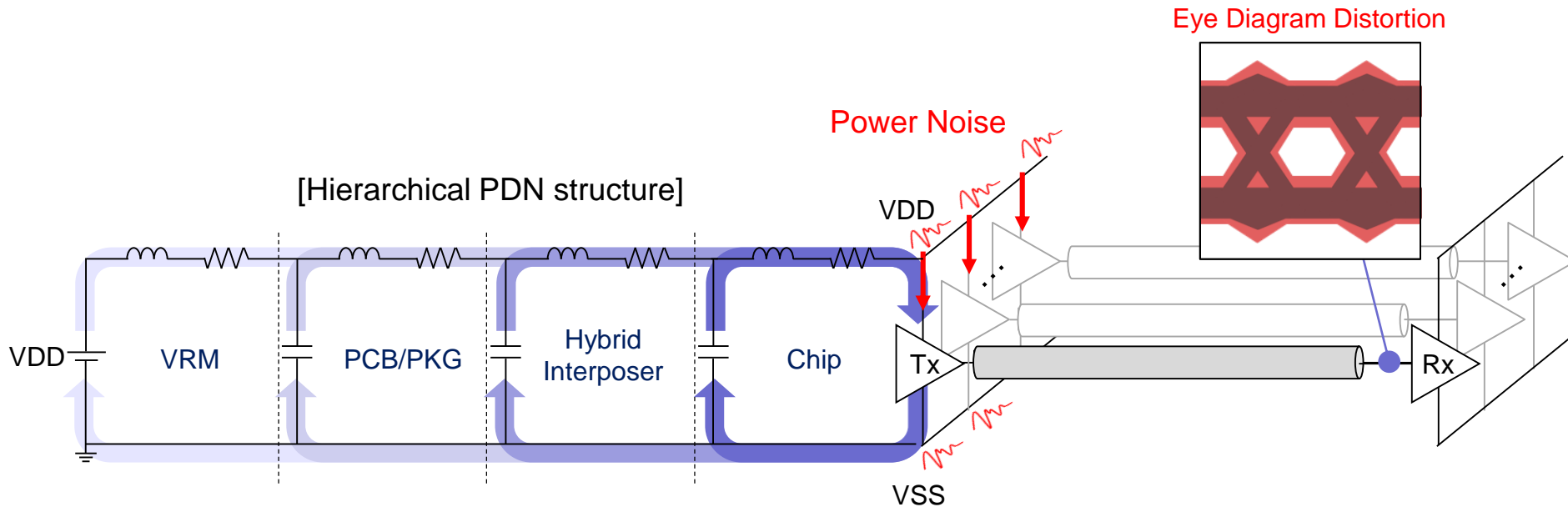
June 11<sup>th</sup>, 2025

Junghyun Lee

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering, KAIST

# Power Noise Impact on Eye Diagram in Hierarchical Power Distribution Network (PDN) Configurations of HBM6

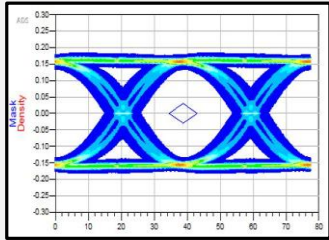


< Eye diagram distortion at Rx due to SSN of HBM I/O interface >

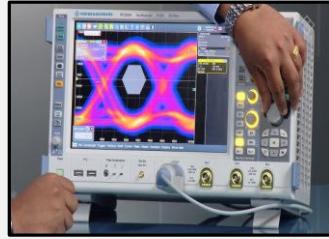
- Power noise induced by hierarchical power distribution network (PDN) causes simultaneous switching noise (SSN) which degrades eye diagram and leads to signal distortion.
- Due to the large number of core and I/O drivers in HBM, the importance of signal integrity/power integrity (SI/PI) co-simulation considering SSN is ever-increasing.
- By utilizing eye diagram simulations, we can effectively observe and understand these complex interactions and their impact on system performance.

# Generative AI-based Eye Diagram Estimation Agent

## Conventional Design Analysis Method



Simulation



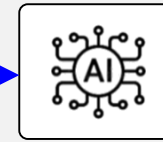
Measurement

*time-consuming*

## Design Analysis with Generative AI Agent



Design



AI Agent

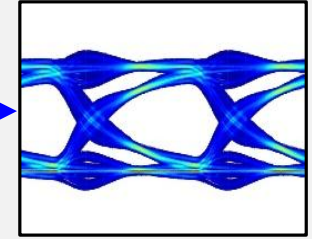
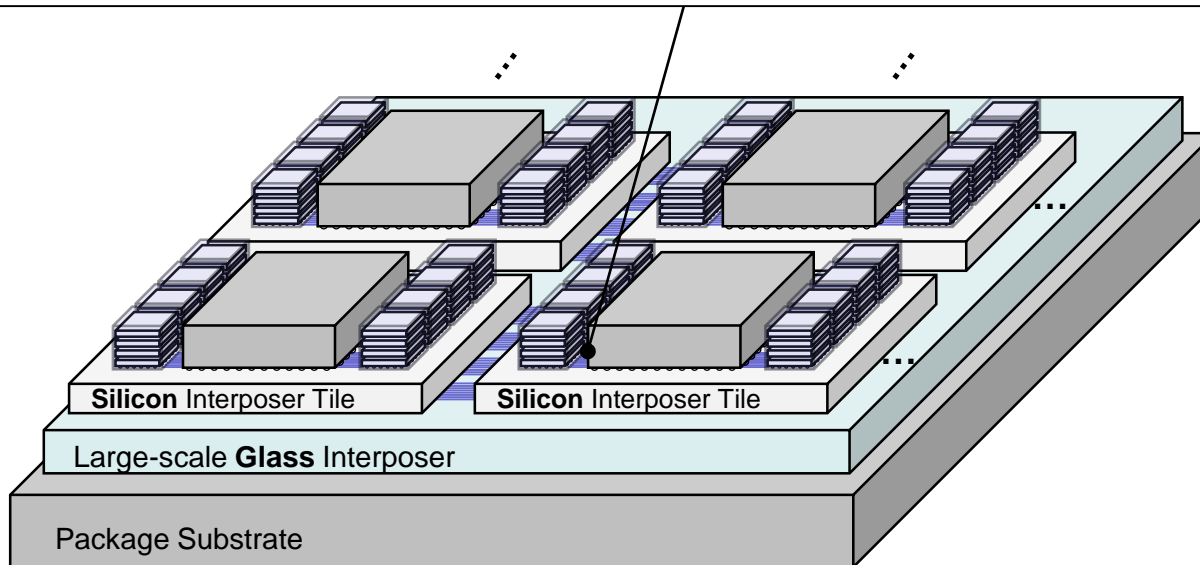


Image Generation (Estimation)

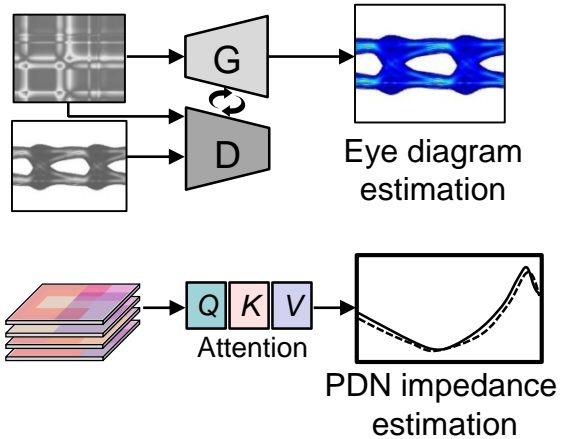
*time-efficient*



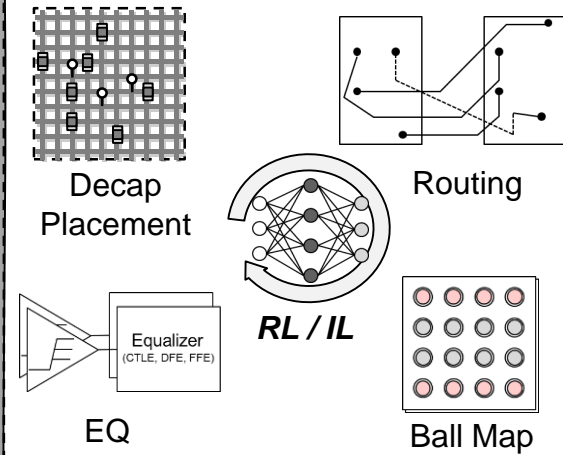
< Silicon-Glass Hybrid Interposer for HBM6-GPU Module >

# AI Agent for HBM Design in TeraLab

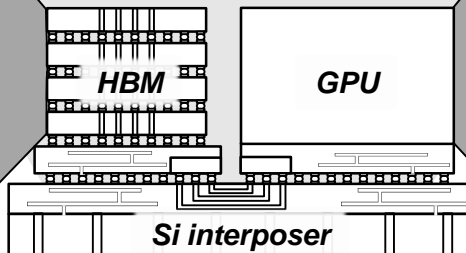
## Simulation Agent



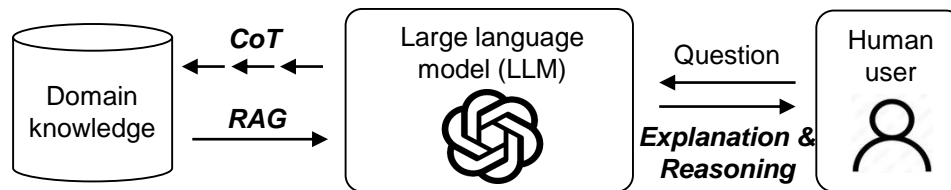
## Optimization Agent



## HBM Design AI Agent



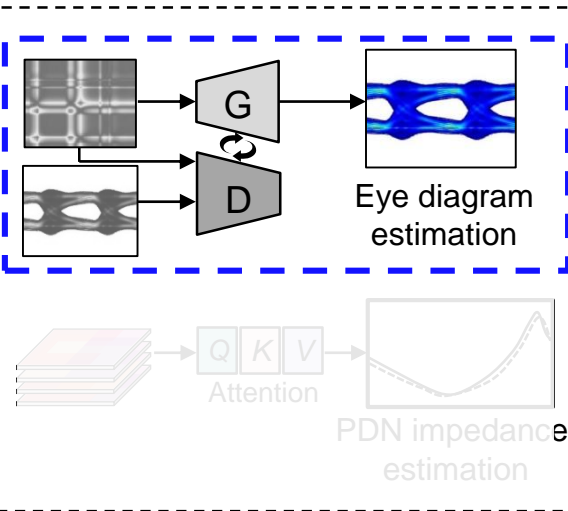
## Human Interactive Agent



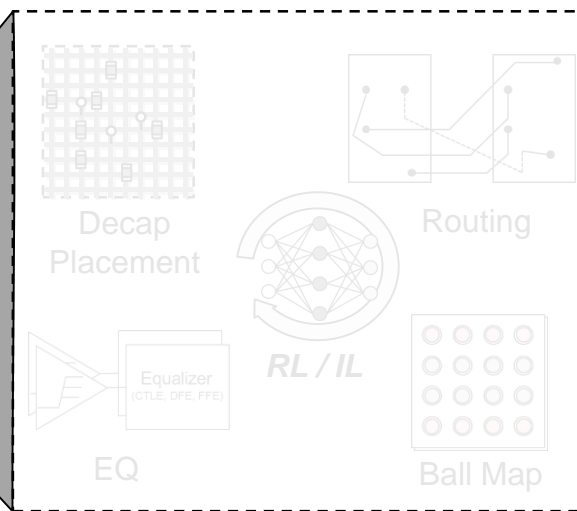
< AI agent for HBM Design in TERA Lab >

# AI Agent for HBM Design in TeraLab

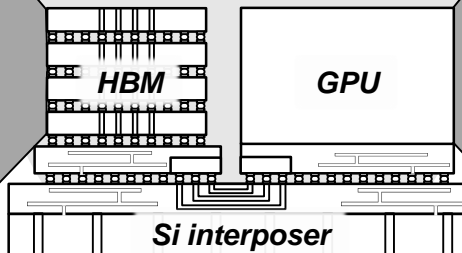
## Simulation Agent



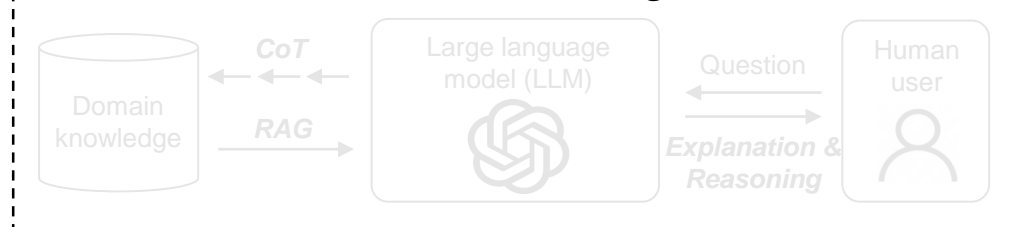
## Optimization Agent



## HBM Design AI Agent

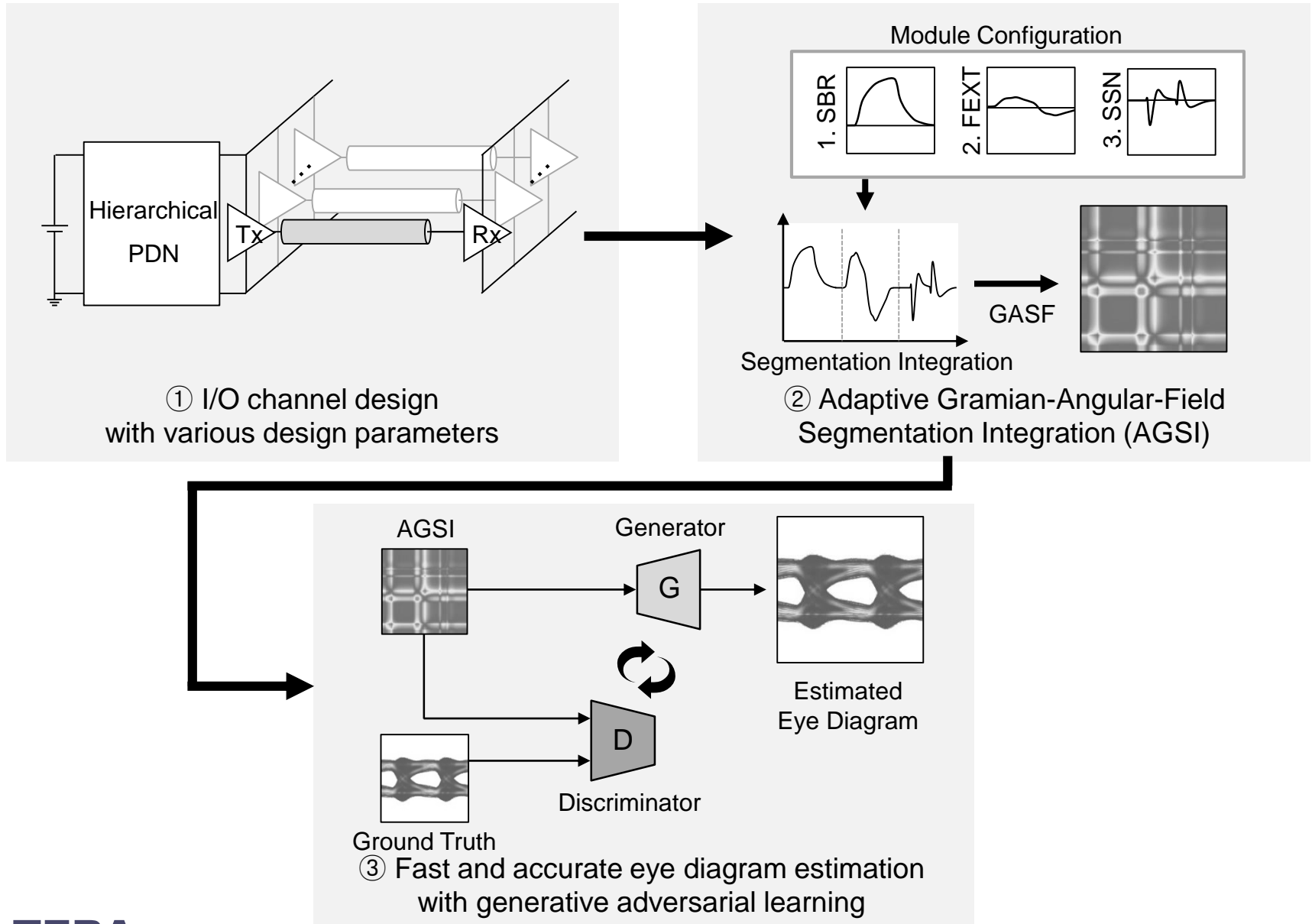


## Human Interactive Agent

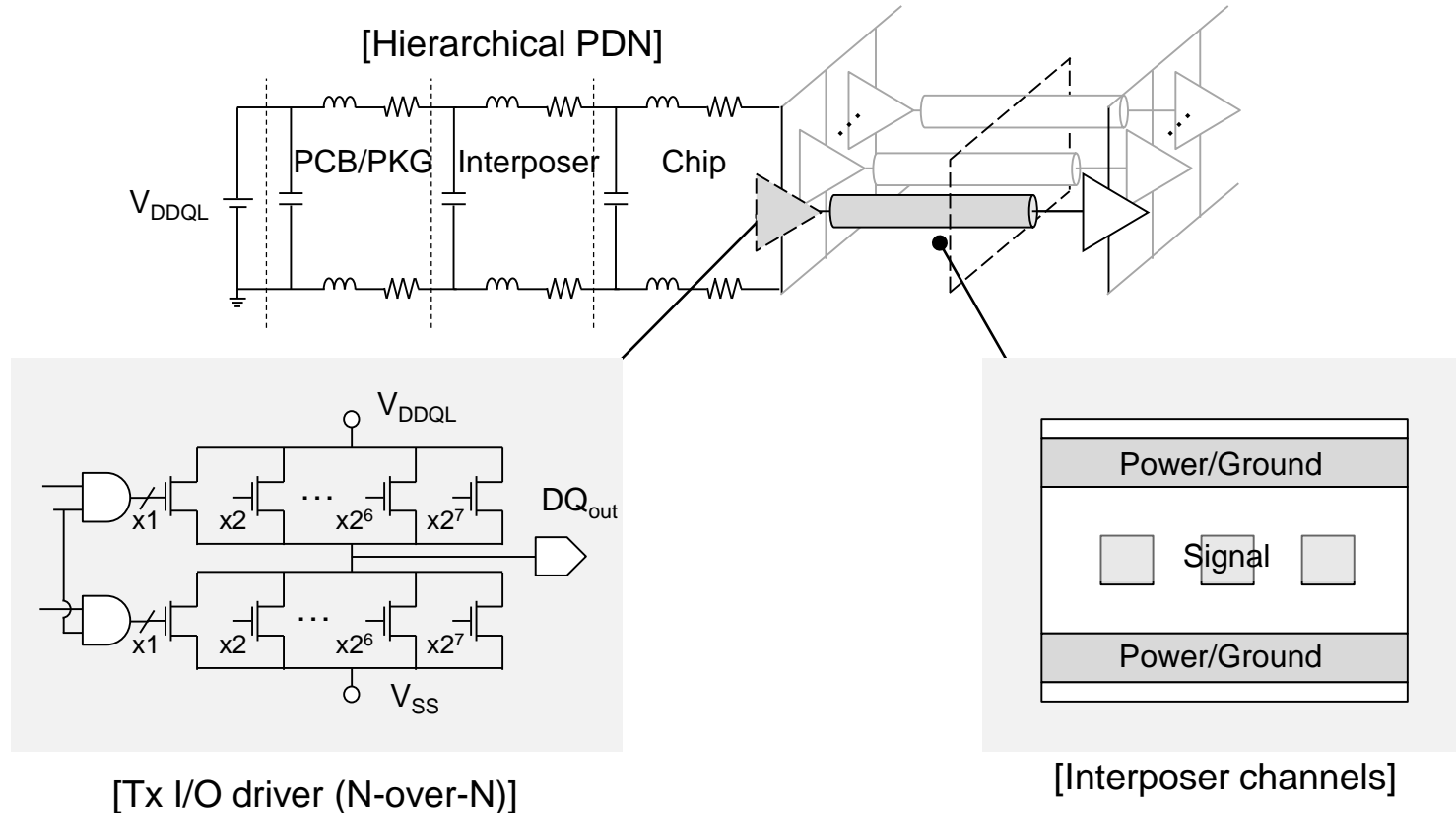


< AI agent for HBM Design in TERALab >

# Proposal of Adaptive Gramian-Angular-Field Segmentation Integration Based Generative Adversarial Network (AGSI-GAN) for Eye Diagram Estimation



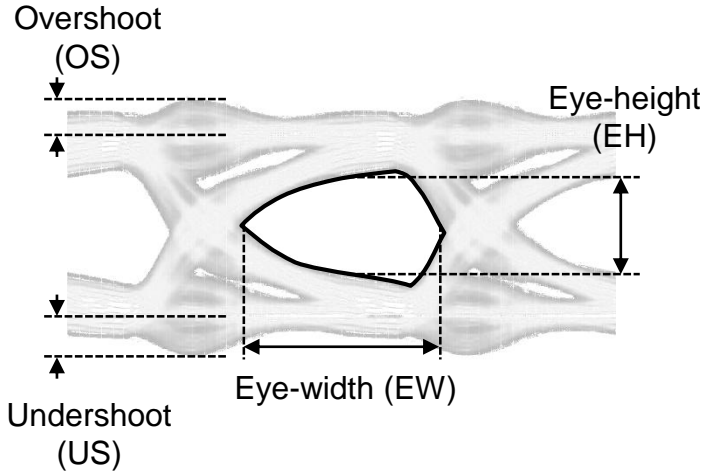
# Designed components of HBM6 I/O Interface



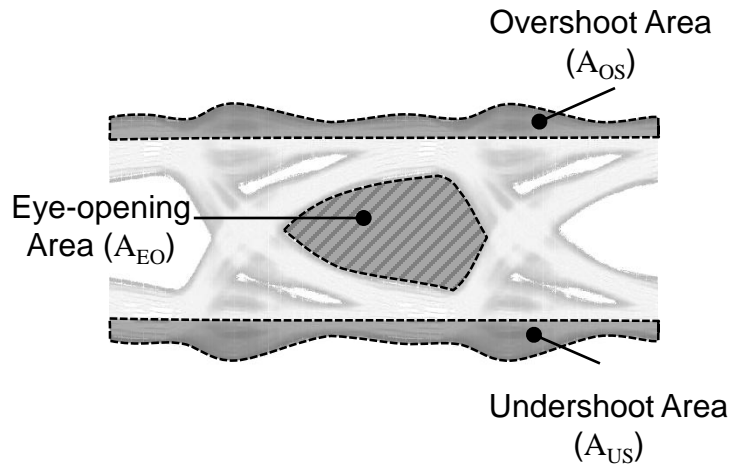
< Designed HBM I/O interface >

- Hierarchical PDN incorporates PCB, package, interposer, and chip layers.
- N-over-N driver for transmitter (Tx) I/Os, and stripline memory channels for DQ signaling are designed.

# Estimation Performance of the Proposed Method on the Test Set



< Conventional point-to-point metrics >

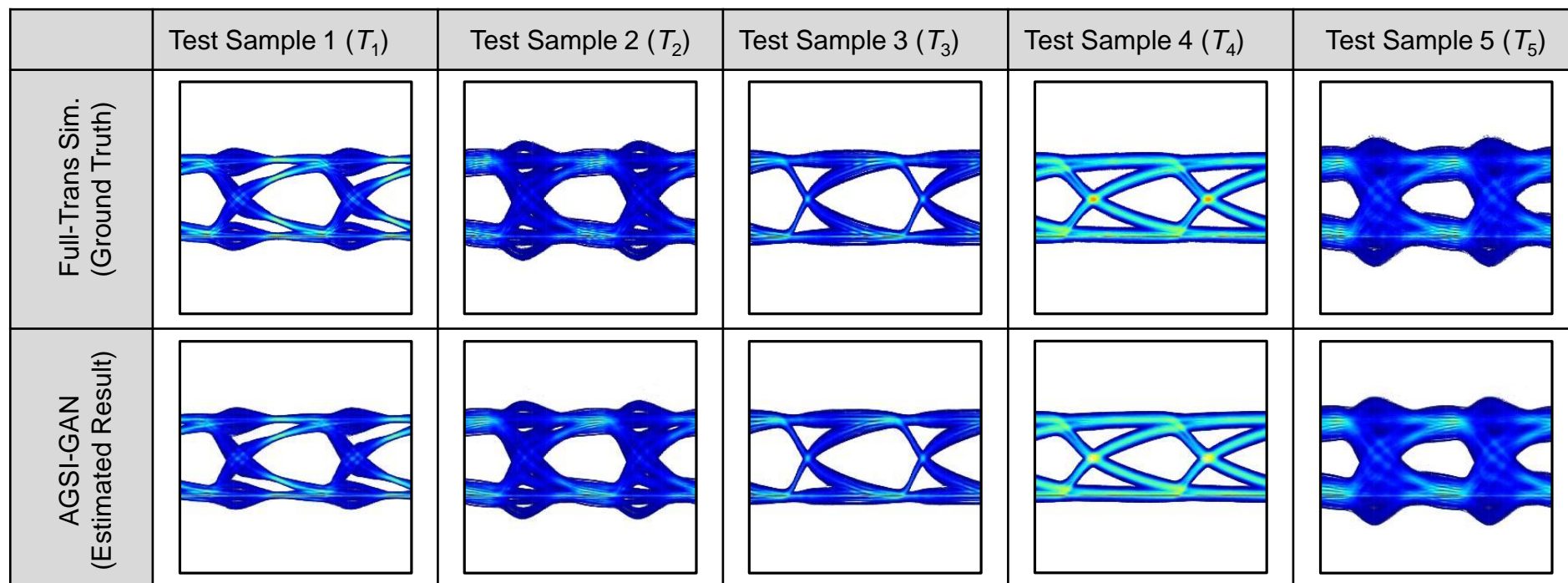


< Cumulative metrics for further comprehensive evaluation >

The number of test dataset	120	
Method	Avg. time	Time efficiency
Full-Transient Simulation	216.8 s	-
AGSI-GAN (proposed)	24.7 s	88.6 %
Measured Eye Diagram Metrics	Mean absolute percentage error (MAPE)	
	EW	0.84 %
	EH	1.08 %
	OS	3.14 %
	US	4.19 %
	$A_{EO}$	1.24 %
	$A_{OS}$	2.36 %
	$A_{US}$	2.93 %

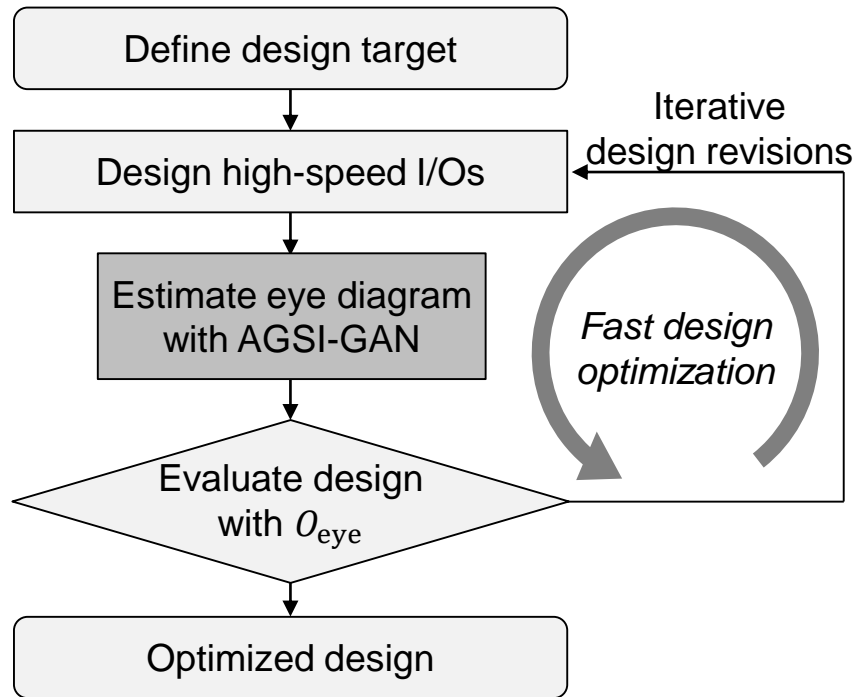


# Estimation Results on Test Samples

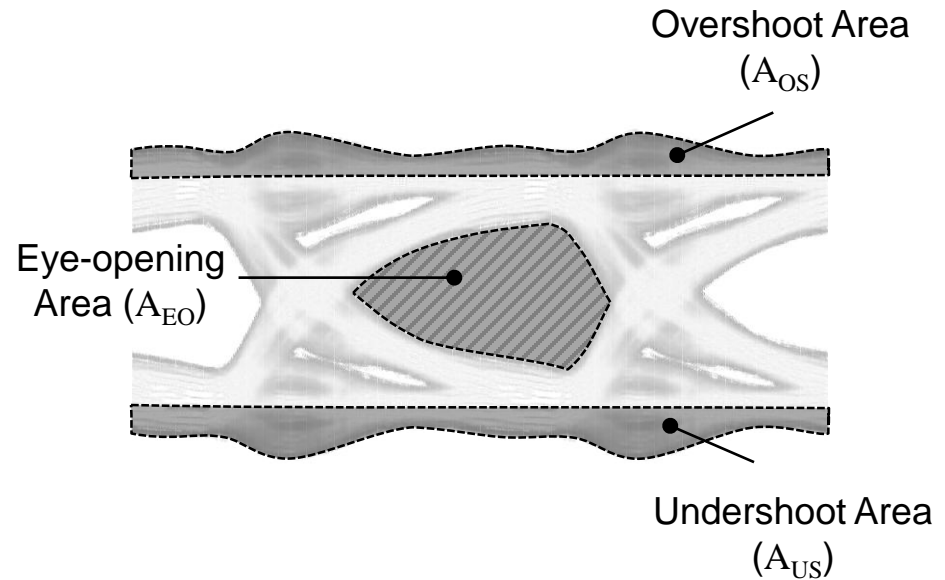


Method	Factor		Test Sample				
			$T_1$	$T_2$	$T_3$	$T_4$	$T_5$
Full-Transient Sim.	IQM	SSIM	-	-	-	-	-
	Time	t [s]	197.6	235.7	211.4	213.0	212.9
AGSI-GAN (Proposed)	IQM	SSIM	<b>0.97</b>	<b>0.95</b>	<b>0.93</b>	<b>0.94</b>	<b>0.93</b>
	Time	t [s]	25.1	24.8	24.4	24.8	24.2

# Accelerated Design Optimization with the Proposed Framework



< Flowchart of high-speed channel design optimization with the AGSI-GAN framework >



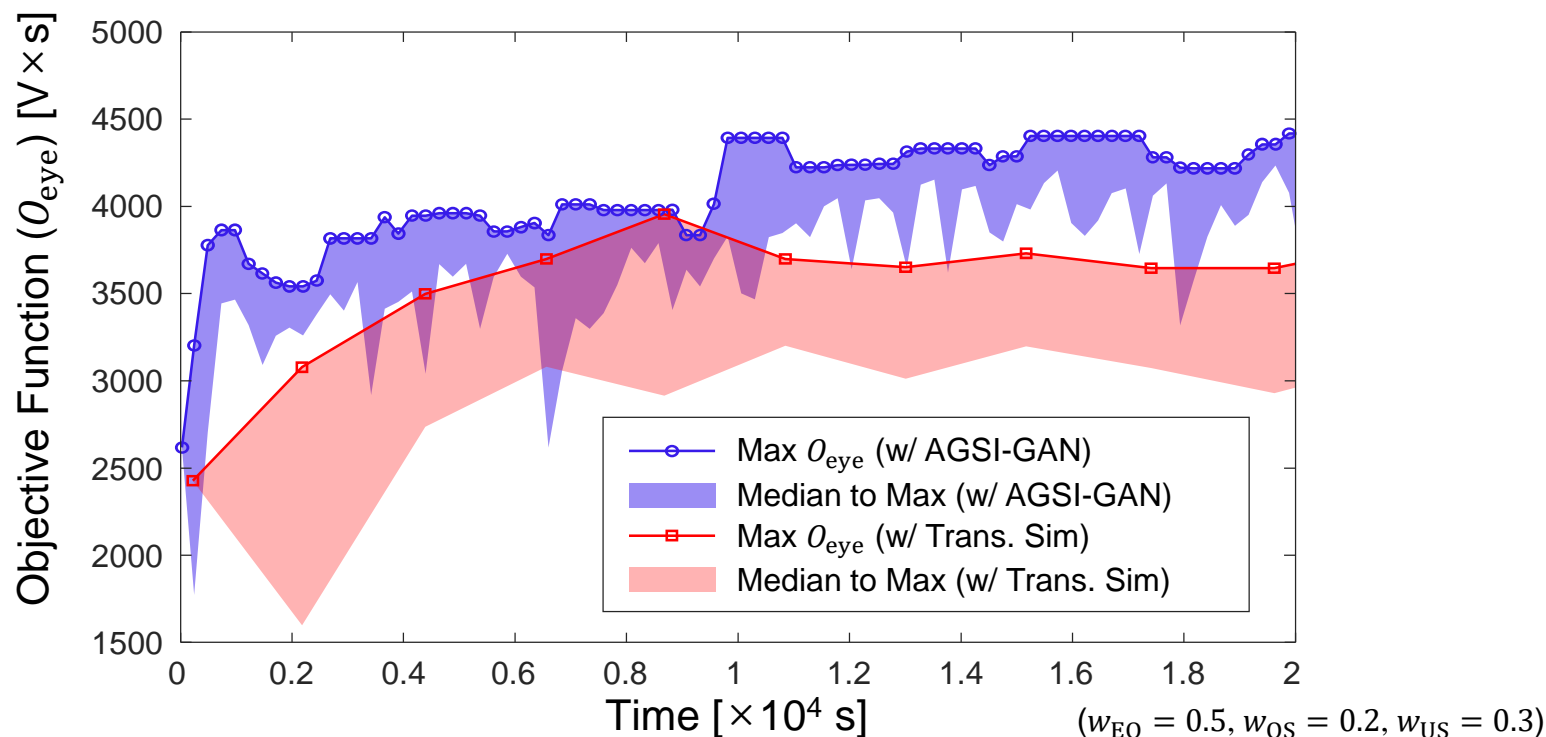
Objective Function

$$O_{eye} = w_{EO} \cdot A_{EO} - w_{OS} \cdot A_{OS} - w_{US} \cdot A_{US}$$

< Objective function for the eye diagram ( $O_{eye}$ ) considering design trade-offs >

- By automating the estimation process with AGSI-GAN, design revision iterations and computing resources can be reduced.
- To address limitation of point-to-point metrics based design evaluation, we propose an objective function for the eye diagram ( $O_{eye}$ ) that maximizes the  $A_{EO}$  while minimizing  $A_{OS}$  and  $A_{US}$ , considering power noise sensitivity.

# Accelerated Optimization Result using the Genetic Algorithm (GA)



< Objective function values over time during the optimization process >

Method	Target $O_{eye}$	Evaluation Tool	Time-to-Target
Genetic Algorithm	3800	Full-Trans. Sim	8672 s
		AGSI-GAN	732 s

- Within the given time budget, full-transient simulation was only able to complete 9 generations, whereas AGSI-GAN allowed for 81 generations.
- Despite GA requiring many iterations, AGSI-GAN significantly accelerates the iterative refinement process by rapidly estimating the eye diagrams required for objective evaluation.

# Thank You!

## HBM

# HBM Roadmap Ver 1.7 Workshop Agenda [3/3]

HBM 세대	순번	Time	Contents	Presenter
HBM7/8	17	15:00 ~ 15:15	HBM-HBF with Storage Network Architecture	안현준
	18	15:15 ~ 15:30	NMC-HBM with HBF for Large-Scale AI Inference	이현이
	19	15:30 ~ 15:45	HBM7 Architecture Integrated with High-Capacity 3D Stacked LPDDR	최인영
	20	15:45 ~ 16:00	Embedded Cooling Structure for HBM7 Architecture	손기영
	21	16:00 ~ 16:15	3D Memory Expansion Architecture for HBM8 with Double-Sided Interposer and HBM-HBF-LPDDR Integration	양채민
	22	16:15 ~ 16:25	AI Design Agent for 3D Placement and Routing Optimization for HBM8 using Reinforcement Learning considering Thermal-Signal Integrity	엄현서
	23	16:25 ~ 16:35	LLM-aided Interactive Reinforcement Learning (IRL) with Switch Transformer for PSIJ Reduction in HBM7	배재근
	24	16:35 ~ 16:45	LLM-based HBM7 Design Agent using Interactive Reinforcement Learning (IRL) for Decoupling Capacitor Placement	김근우
		16:45 ~ 17:00	Closing	김정호 교수

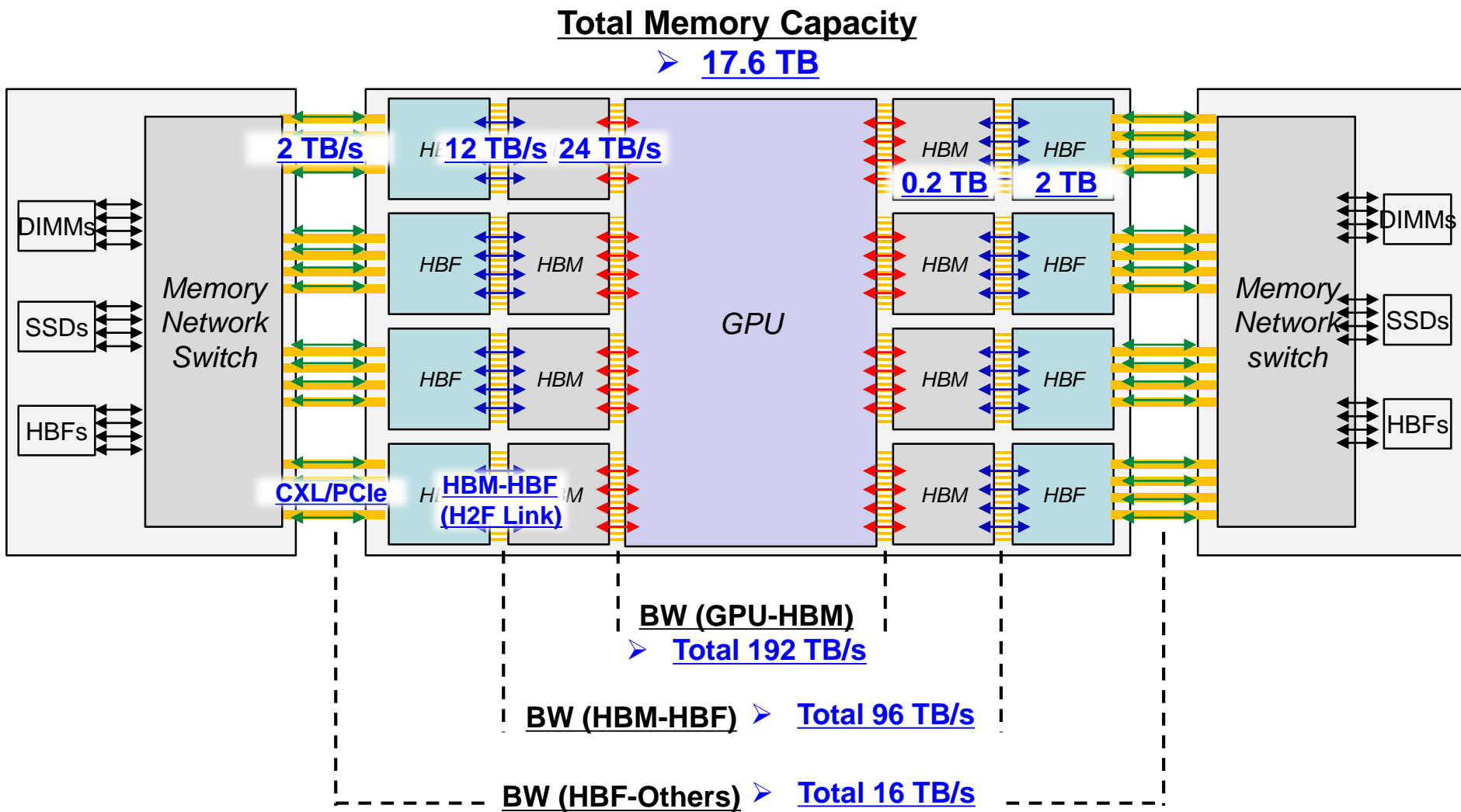
# HBM-HBF with Storage Network Architecture

Hyunjun An

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

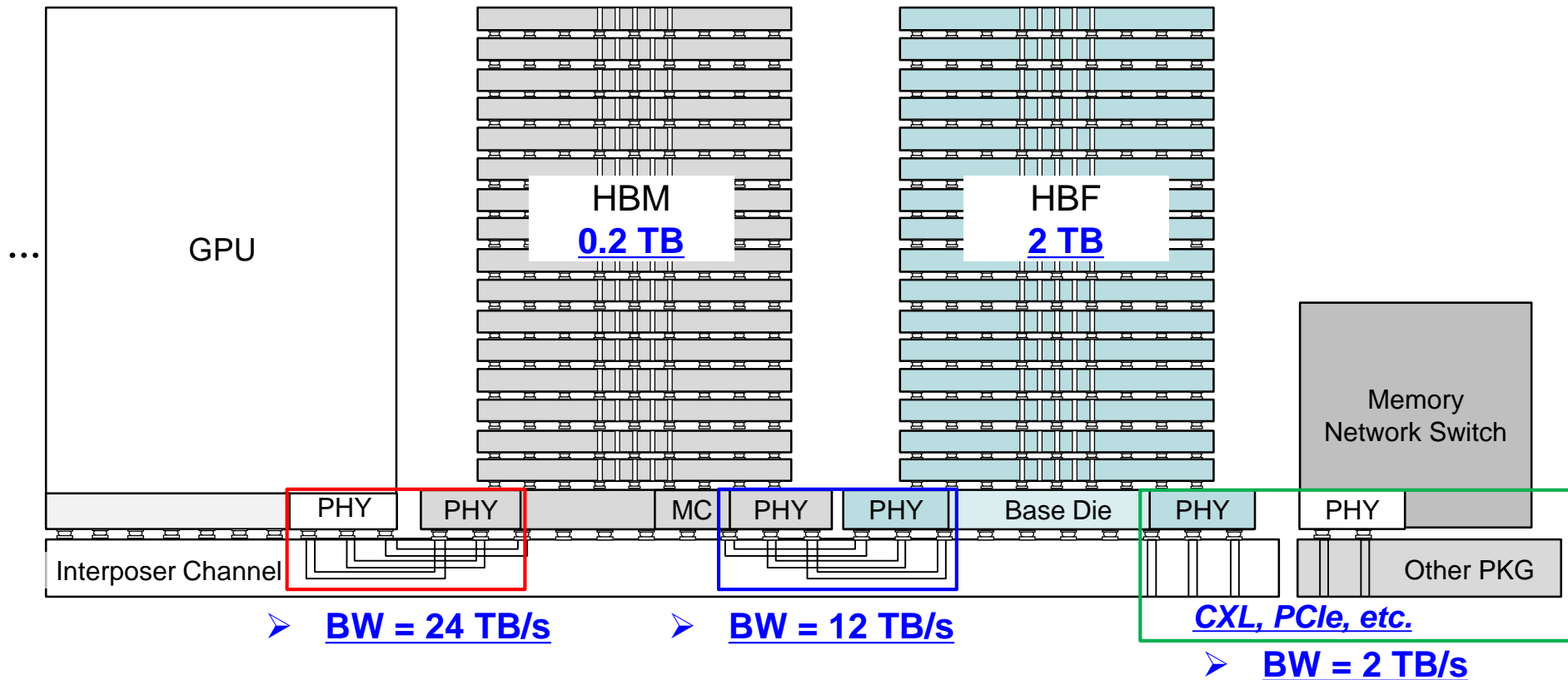
## Key Features of HBM-HBF-Storage Network Architecture – [1/2]



### < HBM-HBF-Storage Network Architecture : Top View >



# Key Features of HBM-HBF-Storage Network Architecture – [2/2]



**[GPU-HBM]**

- ✓ Parallel interface
- ✓ 8,192 x 24 Gbps = **24 TB/s**

**[HBM-HBF (H2F Link)]**

- ✓ Parallel interface
- ✓ 4,096 x 24 Gbps = **12 TB/s**

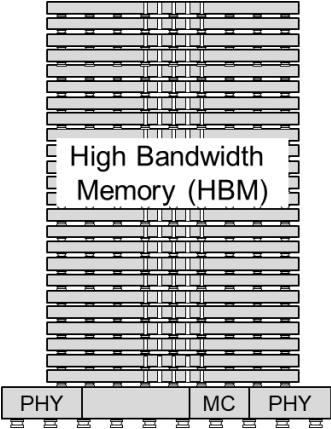
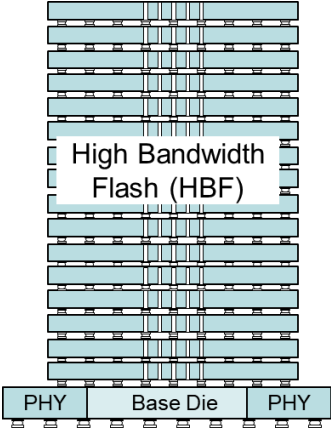
**[HBF Link]**

- ✓ Serial interface (PCIe 7.0, CXL 4.0)
- ✓ 128 Gbps x 128 lanes
- **2 TB/s**

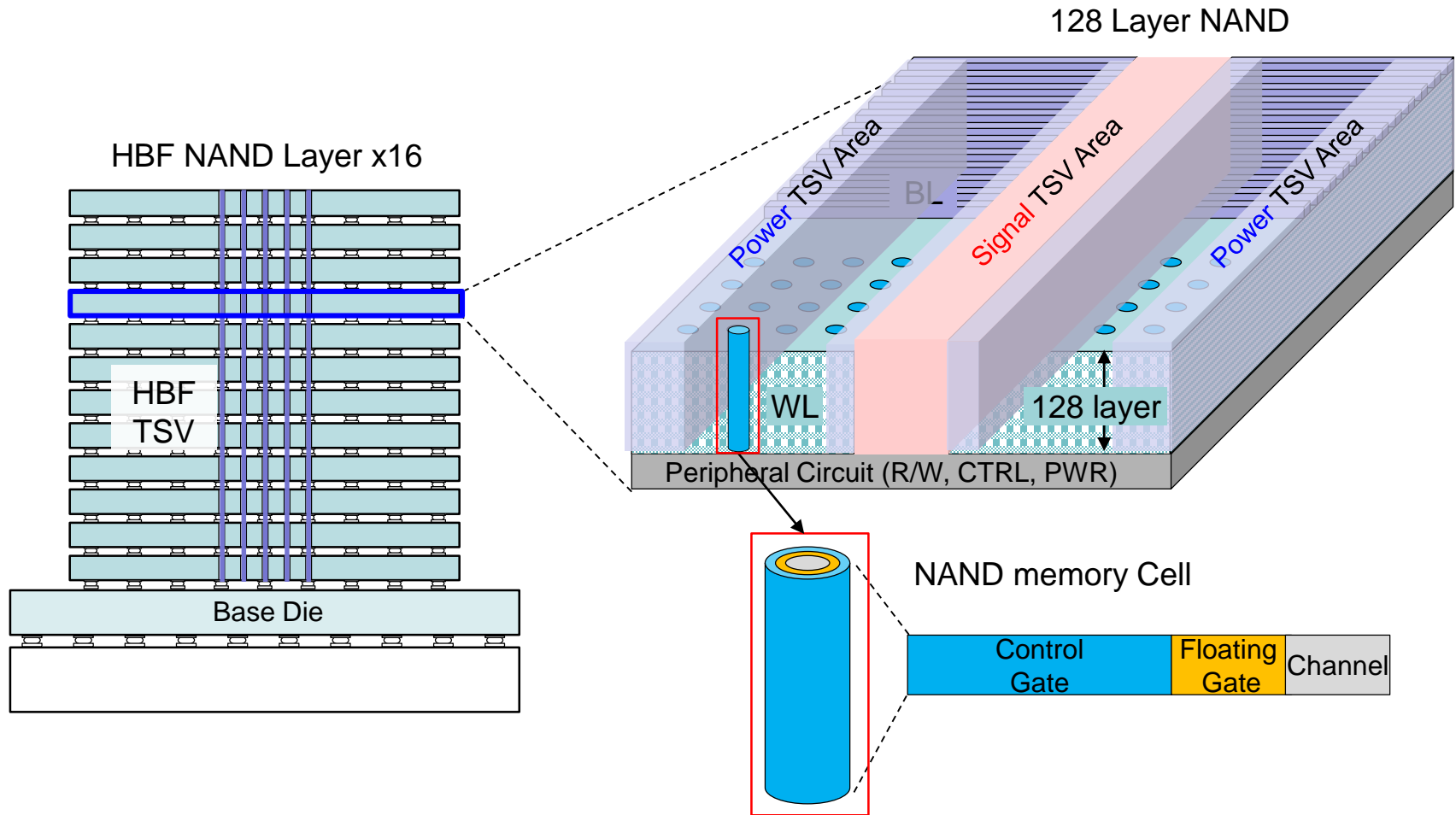
\*Assume PCIe 7.0 : 128 Gbps  
\*Current PCIe 6.0 : 64 Gbps



# Comparison Between HBM and HBF

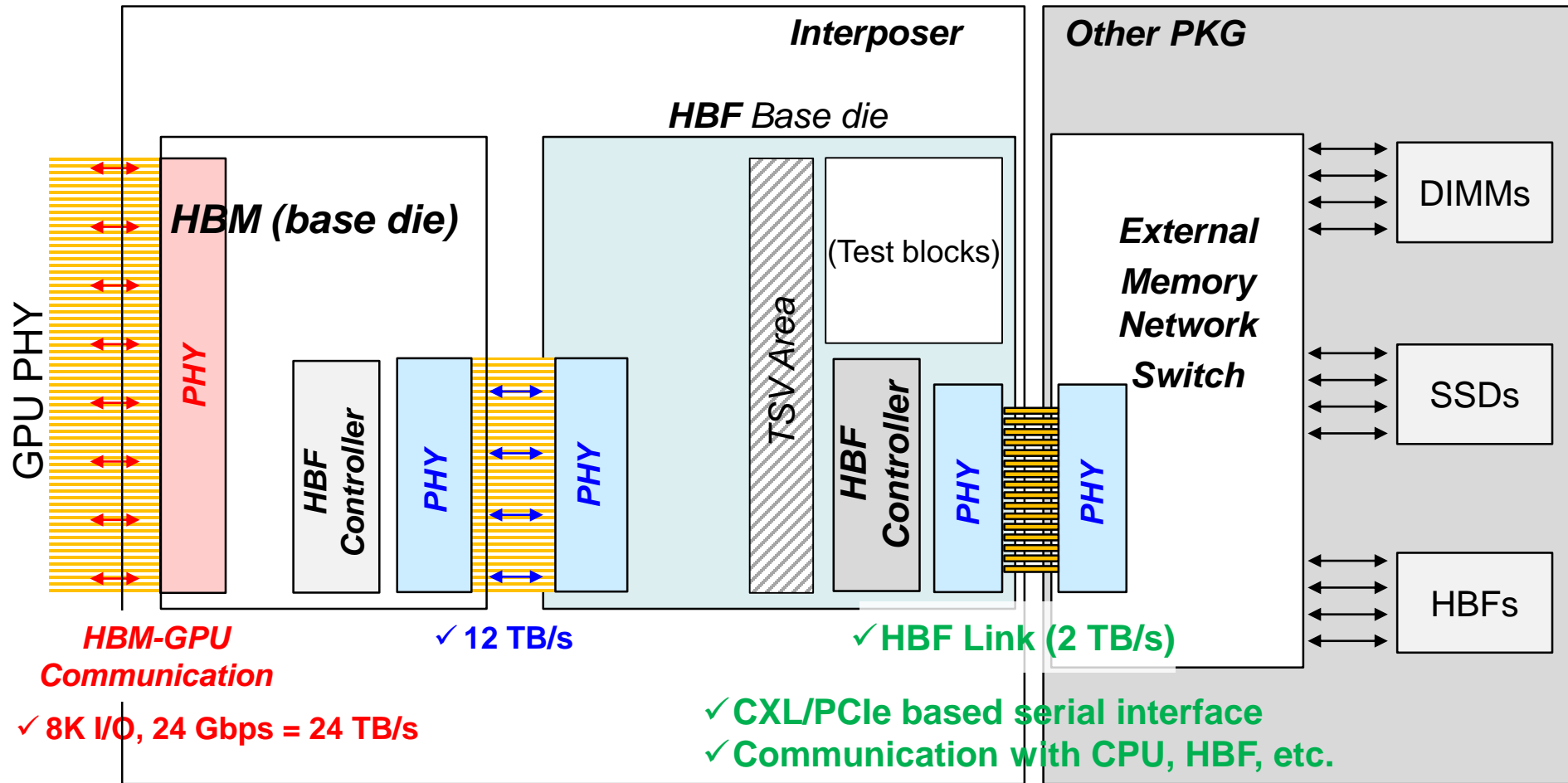
	HBM 7	HBF
Architecture (Side view)		
Stack numbers	24	16
Memory Capacity per die	64 Gb (=8 GB)	512 Gb (=64 GB)
<b>Total Memory Capacity</b>	<b>0.2 TB</b>	<b>2 TB</b>
I/O numbers	8,192	4,096
Signal TSV numbers	32,768	16,384
Diameter / Pitch of TSV	$d_{TSV}=5\text{ }\mu\text{m}$ , $p_{TSV}=20\text{ }\mu\text{m}$	
TSV density	$(1\text{mm}/20\mu\text{m}) \times (1\text{mm}/20\mu\text{m}) = 2.5\text{K}/\text{mm}^2$	
Signal TSV Area	8 mm x 2 mm = <b>16 mm<sup>2</sup></b> (up to 40K TSVs)	8 mm x 1 mm = <b>8 mm<sup>2</sup></b> (up to 20K TSVs)
Datarate	24 Gbps	24 Gbps
<b>Bandwidth (GPU-HBM)</b>	<b>24 TB/s</b>	<b>12 TB/s</b>

# Architecture of High Bandwidth Flash (HBF)



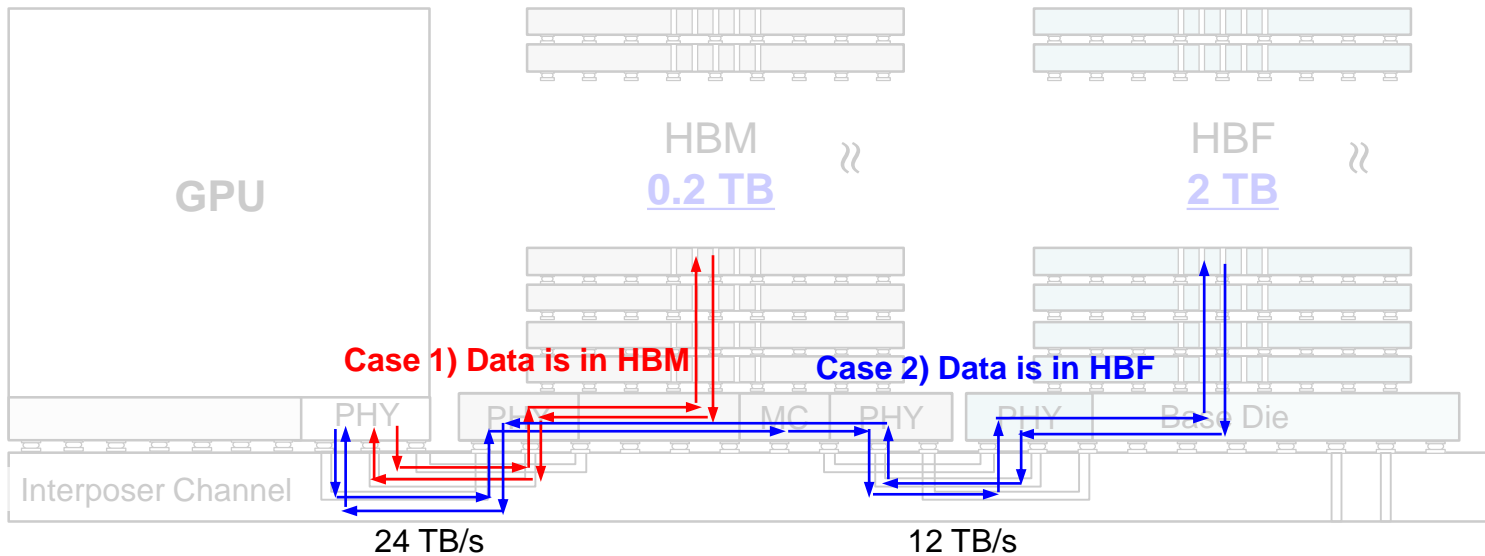
< High Bandwidth Flash (HBF) Architecture for Memory Intensive LLM Inference >

# Base Die Design of HBM-HBF-Storage Network Architecture



< Base die design of HBM-HBF-Storage Network Architecture >

# Data Flow Path for GPU-HBM-HBF Architecture



< Data flow path for GPU-HBM-HBF Architecture >

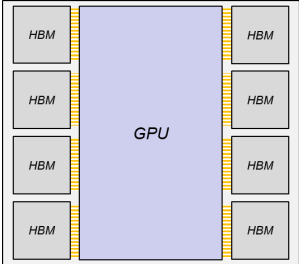
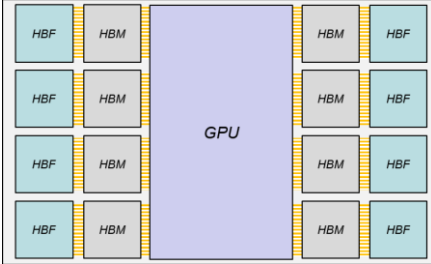
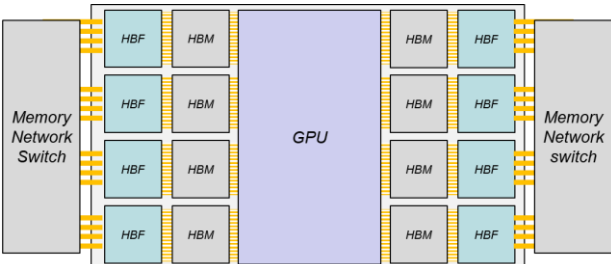
## Case 1) Data is in HBM

- ① GPU → ② HBM (read request issued) → ③ GPU-side Memory Controller (decodes & schedules) → ④ HBM DRAM access (local) → ⑤ Return data → ⑥ GPU via PHY (24 TB/s)

## Case 2) Data is in HBF

- ① GPU → ② HBM (read request issued) → ③ HBM-internal Memory Controller (detects HBF access) → ④ HBF Flash access (through H2F link @ 12 TB/s) → ⑤ Data returned to HBM buffer → ⑥ GPU via PHY (24 TB/s)

# Comparison of HBF-based Memory Architectures

		HBM-Only	HBM-HBF Architecture	HBM-HBF-Storage Network Architecture
Architecture (Top view)				
Total Memory Capacity		0.2 TB x 8 EA <b>= 1.6 TB</b>	1.6TB(HBM) + 16TB(HBF) <b>= 17.6 TB</b>	<b>17.6 TB + alpha (~100 TB)</b>
Total Bandwidth	GPU-HBM	✓ 24 TB/s x 8 ✓ <b>BW = 192 TB/s</b>		
	HBM-HBF	NA	✓ 12 TB/s x 8 ✓ <b>BW = 96 TB/s</b>	
	HBF-Storage	NA	NA	✓ 2 TB/s x 8 ✓ <b>BW = 16 TB/s</b>

< Comparison of HBF-based memory Architectures >

# Thank You!

## HBM

# Near-Memory-Computing(NMC)-HBM with High-Bandwidth-Flash(HBF) for Large Scale AI Inference

Hyuni Lee

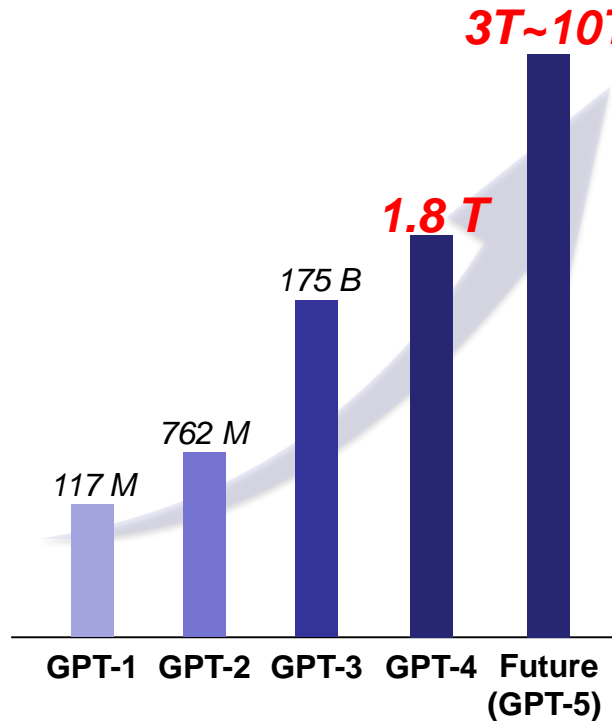
Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

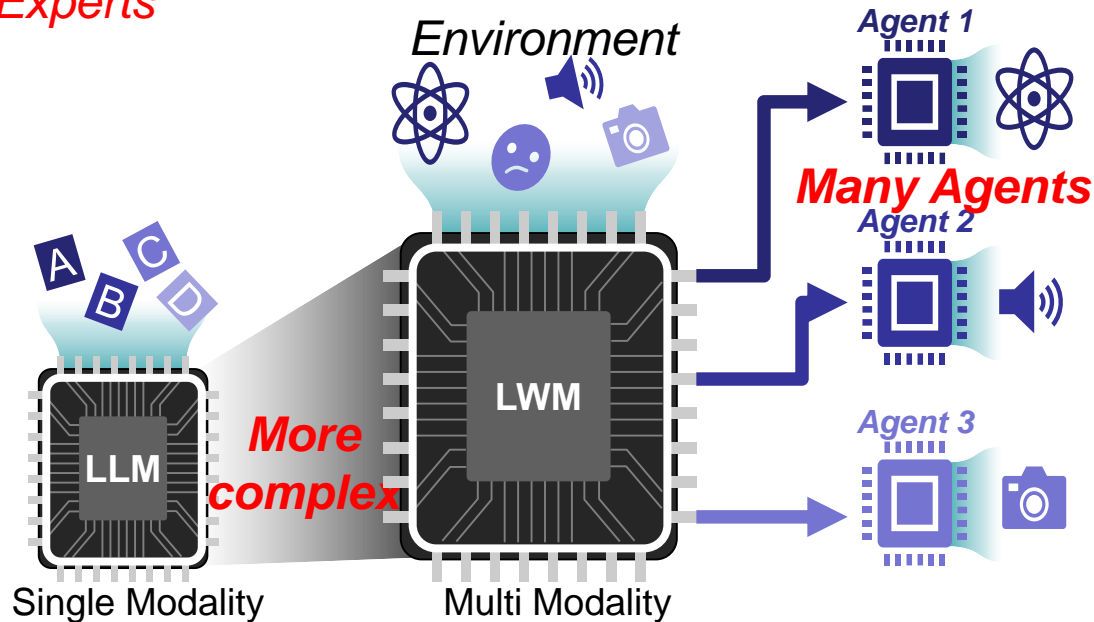
- I. Introduction : Why Memory Expansion needs and can be solved with Flash?
- II. Proposal of New Memory Expansion Scheme in Cascaded Hetero-Memory Architecture
- III. Advantages of the NMC-HBM with HBF (High Bandwidth Flash) Architecture



# Multi-Agent LWM(Large World Models) Toward AGI



< Model Size Trends by GPT Ver. >

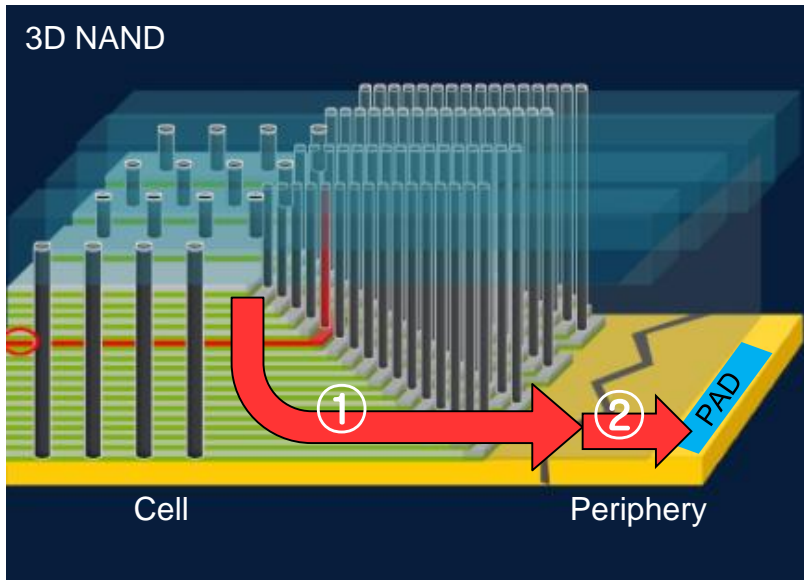


< LLM Toward LWM(Large World Model) >

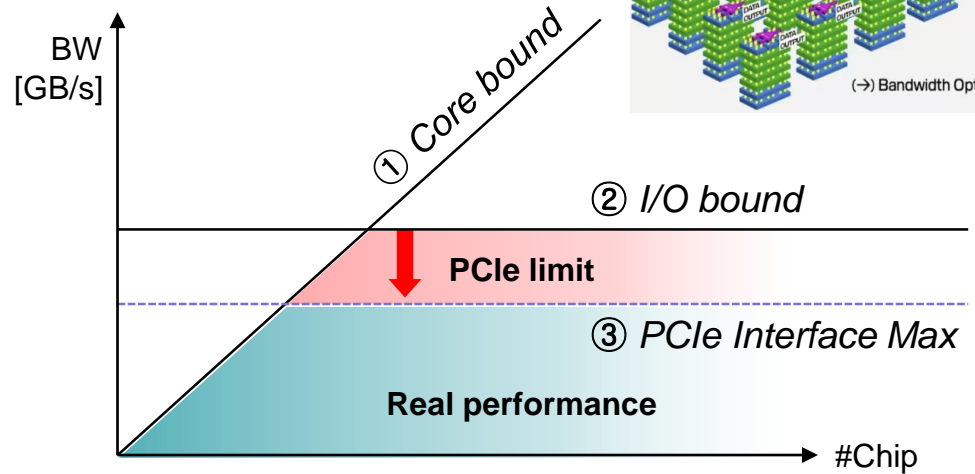
- LWM(Large World Models) like Google's Genie 2 are designed to understand the real world.
- Jensen Huang @CES2025 "Perception AI → Generative AI → Agentic AI → Physical AI"
- Agentic AI drives rapid post-training scaling, requiring multiple models for larger inference capacity.

➔ **Beyond bandwidth, memory scalability is essential for AGI with Agentic AI** (2024~2030yr)

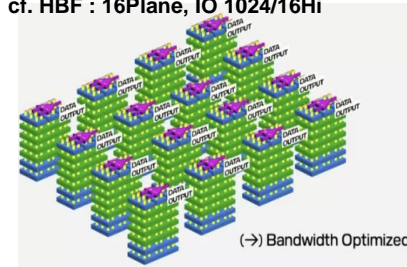
# Flash Bandwidth isn't so low in terms of sequential read



<3D-NAND Chip Architecture>



cf. HBF : 16Plane, IO 1024/16Hi



<NAND Bandwidth : Core vs. IO vs. I/F Bound>

- NVMe SSDs bandwidth consists of core-bound, I/O-bound components and PCIe interface
- Ex) Core-bound = 16 channels  $\times$  1 way  $\times$  16KB/page  $\times$  4 planes / 20 $\mu$ s (51 GB/s, SLC)
- Ex) I/O bound = 16 channel  $\times$  3.6 Gb/s  $\times$  #IO (8)  $\times$  80% (46 GB/s)
- PCIe spec. = Interface becomes the bottleneck (4Lane-based Gen5: 14 GB/s, Gen6: 28 GB/s)

➔ **Sequential Read mode in SSD NVMe offers BW potential far beyond**

**PCIe I/F limits**

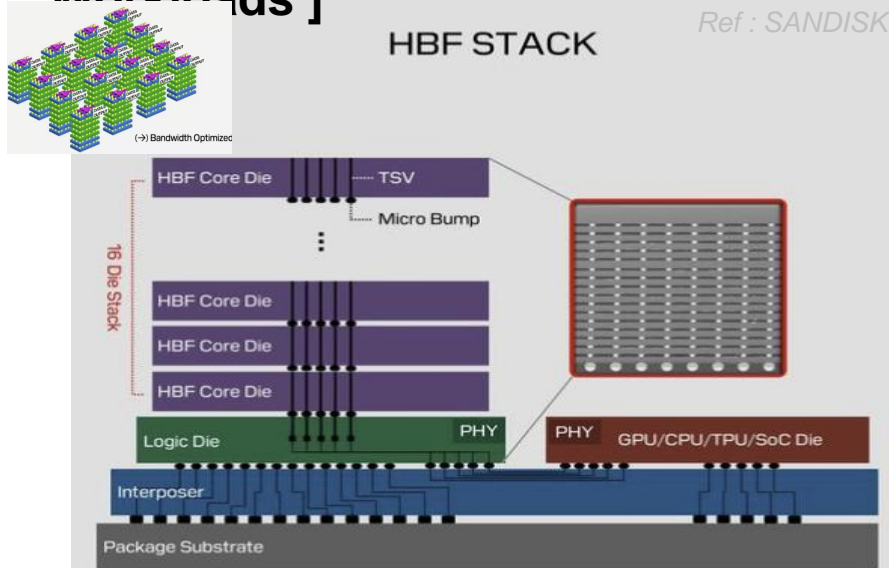
- Core-bound = 16 channel  $\times$  1way  $\times$  16KB/page  $\times$  16 planes / 5 $\mu$ s (800 GB/s, SLC ZNAND 수준) > 500GB/s @ GPT4 based
- I/O bound = 16 channel  $\times$  12 Gb/s  $\times$  #IO/ch (1024 = 8Byte/ch)  $\times$  70% (1536 GB/s)  $\rightarrow$  (core가 1/2배 shortage)
- PCIe spec. = Interface becomes the bottleneck (4Lane-based Gen5: 14 GB/s, Gen6: 28 GB/s  $\rightarrow$  Gen6 16Lane : 112 GB/s)

# Industry Sensing) HBF from SANDISK

## [ High Bandwidth Flash(HBF) Augmenting HBM Memory with NAND Flash for AI Inference

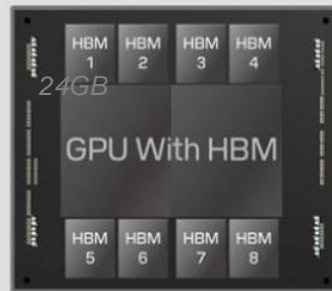
### Workloads ]

Ref : SANDISK

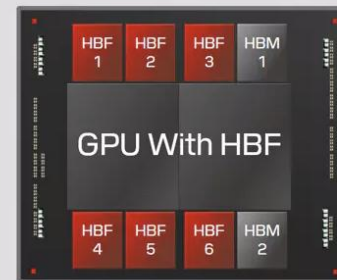


- 1) Match HBM Bandwidth  
: Deliver 16x Capacity at Similar Cost
- 2) Enabled by BiCS Technology  
: With CBA Wafer Bonding (*Cmos directly Bonded to Array*)
- 3) Proprietary Stacking Technology  
: Ultra-Low Die Warpage for 16H Stacking
- 4) Architecture Developed Over the Past Year  
: With Inputs From Major AI Players

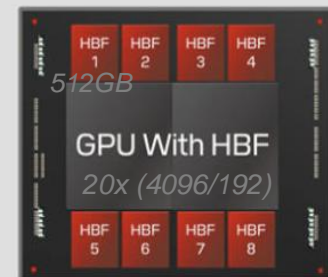
	GEN 1	GEN 2	GEN 3	
CAPACITY	1x	1.5x	2x	↗
READ BANDWIDTH	1x	1.45x	2x	⌚
ENERGY EFFICIENCY	1x	0.8x	0.64x	⚡



192GB Total Memory



3,120GB Total Memory

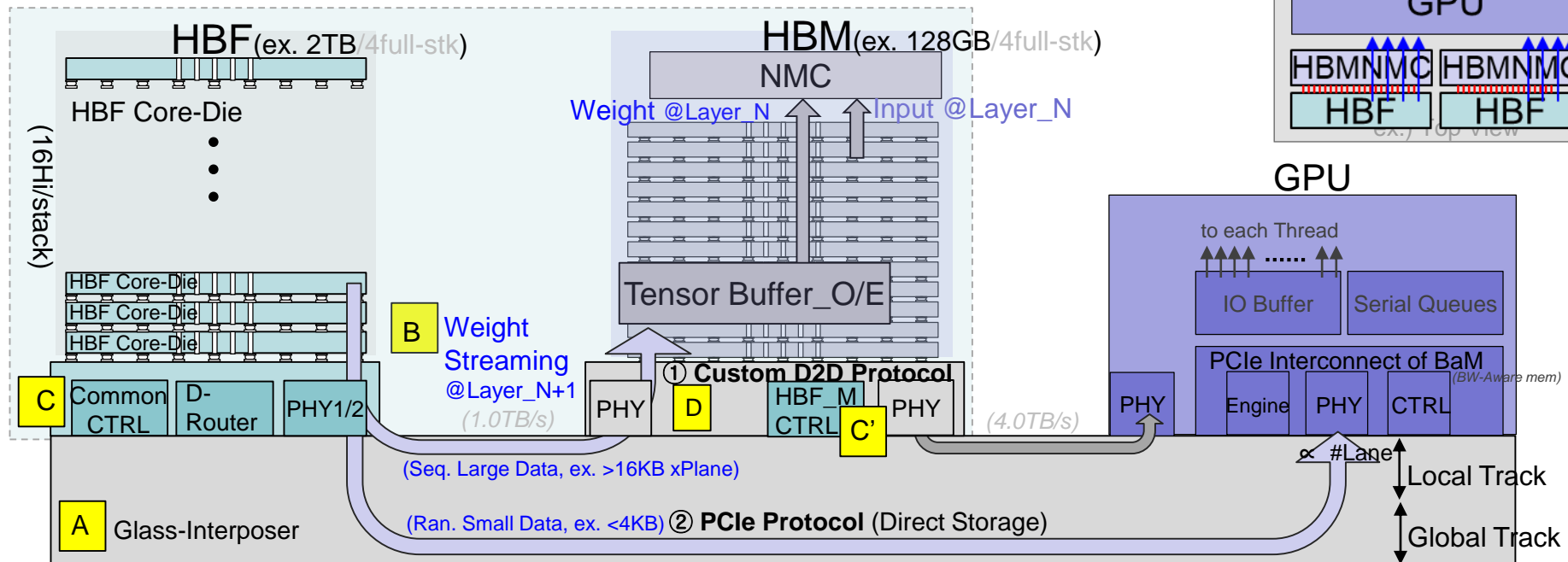


4,096GB Total Memory

256Gb/die x 16Hi/stack x 8HBF = 4TB

- I. Introduction : Why Memory Expansion needs and can be solved with Flash?
- II. Proposal of New Memory Expansion Scheme in Cascaded Hetero-Memory Architecture
- III. Advantages of the NMC-HBM with HBF (High Bandwidth Flash) Architecture

# A Proposed Cascaded Hetero-Memory PKG Architecture using High-Bandwidth-Flash(HBF)

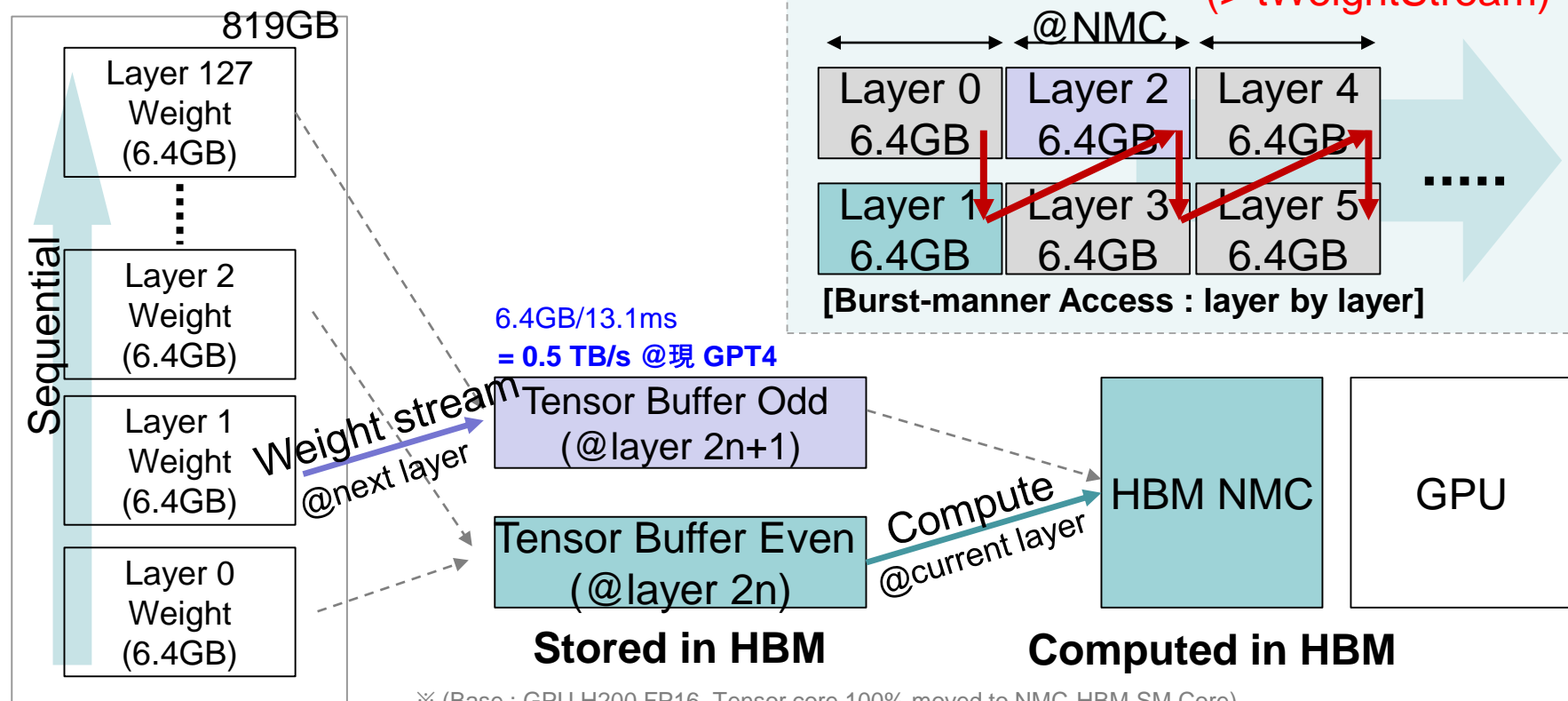


[ A proposed Cascaded Hetero-Memory Architecture : HBF stores Pre-trained fixed data (Read intensive), HBM stores Intermediate result data (R/W balanced) ]

- A ■ **Glass-Interposer** : allows Stack-up for global far track & Size-up for HBF area, compared to SI-Interposer
- B ■ **Weight Streaming to HBM-NMC** : pipelined via 'Tensor Buffer' (or directly in future) located at an allocated address within the HBM itself
- C ■ **HBF/HBM Base-die** : 1) Common CTRL : Flash Cell Reliability-related CTRL  
(ex. FTL, DRAM/SRAM Buffer, NAND Scheduler, ECC Engine, Wear-leveling/GC unit, PCIe Host I/F)  
2) HBF\_M Ctrl : layer transition-aware prefetch scheduling (ex. CMD Parser & Queue)  
3) Dual-protocol router (D-router) for both Custom D2D & PCIe  
& Conditional 2:1 Mux (only if. Full-Address sharing access type rather than Allocated-Address one)
- D ■ **On-package** : Enables beyond SSD PCIe Gen limit by providing large chunks of sequential weight data per layer for NMC via non-packetized custom D2D protocol

# HBF can achieve zero latency by **pipelining I/O and NMC** computation via a **pre-allocated** tensor buffer in HBM inner-address

“On-Package Compute-I/O Parallelism”

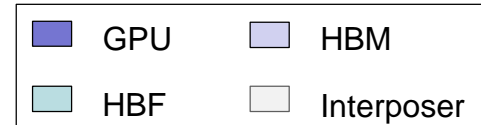
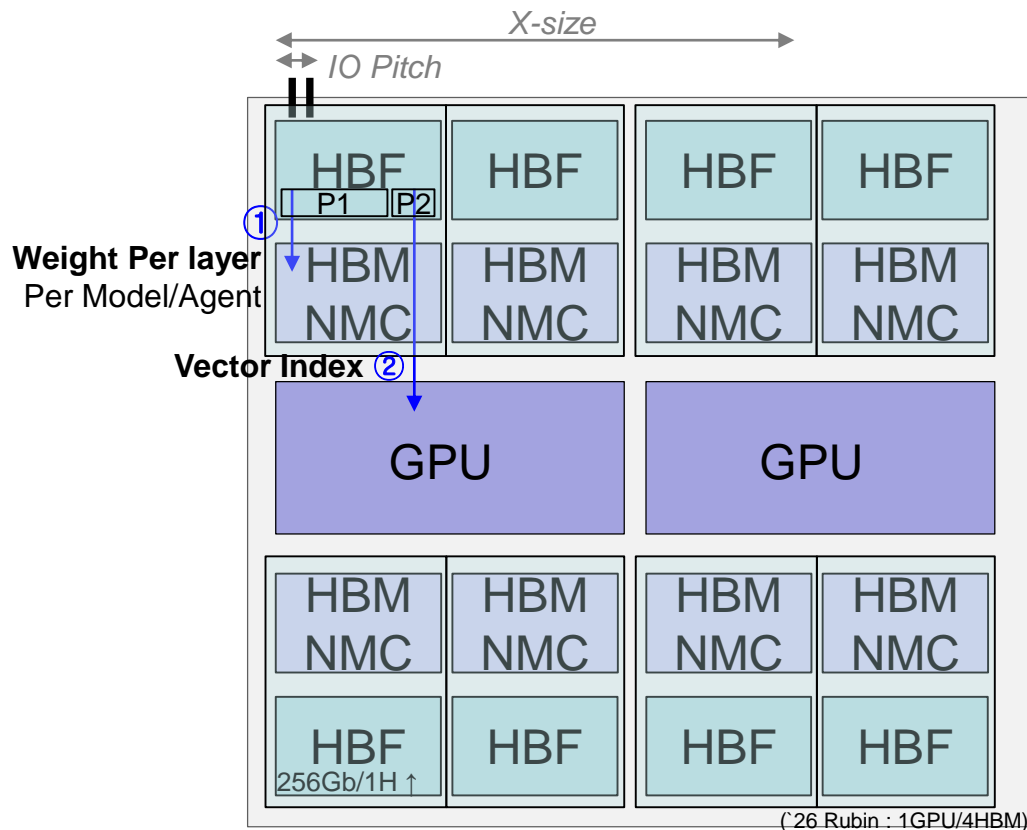


※ (Base : GPU H200 FP16, Tensor core 100% moved to NMC-HBM SM Core)

GPT4 Compute per Layer (26 TFLOPs) / H200 Tensor Core Max (1979 TFLOPs) = Min layer latency (13.1ms)

- Weight streaming for layer1 while near-memory-computing for layer0 **with prefetched weight beforehand**
- DMA **transfer between heterogenous memory within a package** without GPU
- Merit-1) Flash latency can be hidden, -2) Memory expansion can be feasible
- How) HBF : IO #1024/16H = 8 byte/die (x8 #IO 比 Conv) → Core bound ↑ w/ fast-sensing & more planes

# HBM Direct Storage with HBF : Top view



“HBM’s **BW** For **Active**  
model/agent among Total  
models/agents”

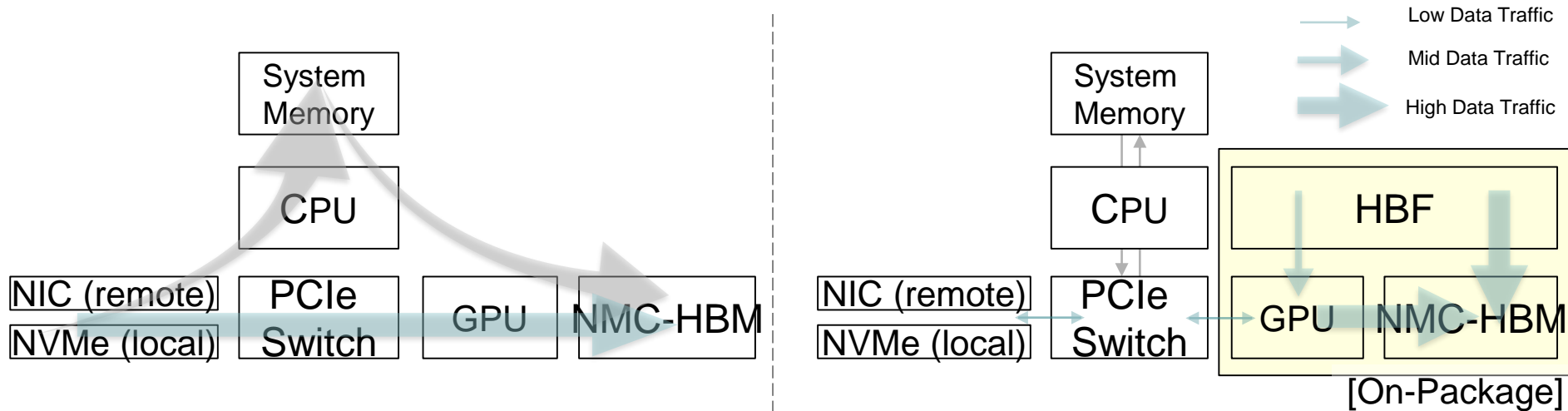
“HBF’s **GB** For **Total**  
models/agents”

- Application-①. Large ‘Weight per layer’ in Multi-Model/Multi-Agent Inference  
: Larger BW requires larger local cache capacity to avoid frequent remote fetches (prefetch efficiency ↑)
- Application-②. Low Latency ‘Vector Index’ for massive DB Indexing such as RAG during Inference  
: Using ‘Local Flash media’ as a GPU memory tier helps fully utilize GPU compute performance
- Future HBF : Can be even more than division PHY1/2 for Network Storage, HBF Self Link +etc

- I. Introduction : Why Memory Expansion needs and can be solved with Flash?
- II. Proposal of New Memory Expansion Scheme in Cascaded Hetero-Memory Architecture
- III. Advantages of the NMC-HBM with HBF (High Bandwidth Flash) Architecture



# Non-Volatile Memory Flash : Less Network Traffic



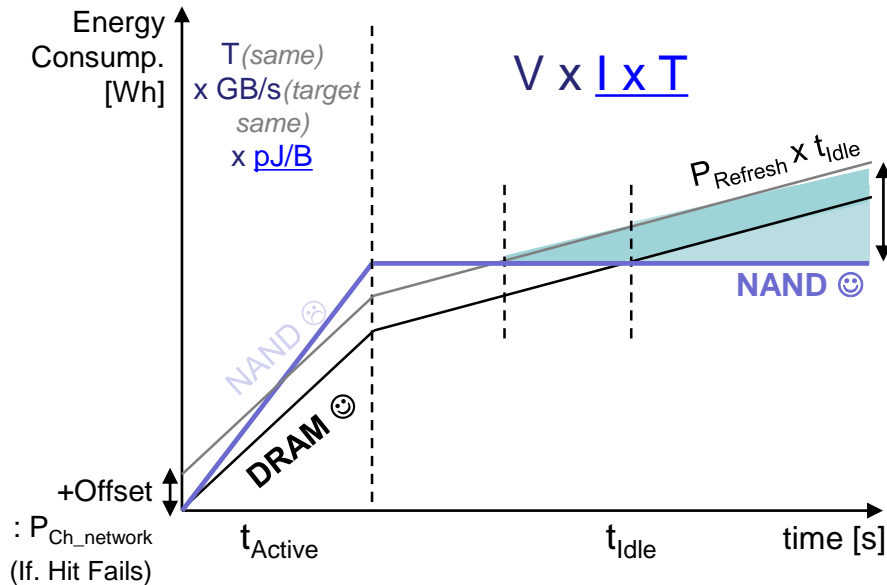
Without HBF :

Conventional systems must repeatedly fetch model weights or large activation feature maps from NVMe SSDs or network-attached storage whenever the GPU requires them.

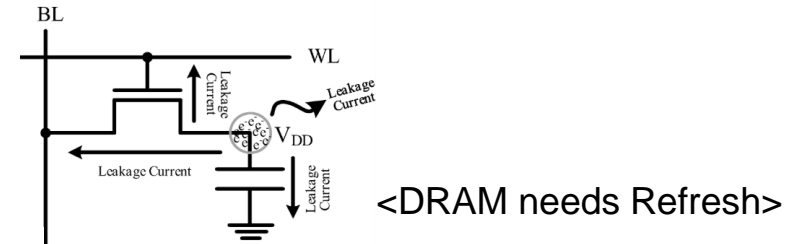
With HBF :

- 1) High Data Hit Ratio via On-Package Flash(*HBF*)
- 2) Network I/O Offloading → Significant reduction in network traffic during inference
- 3) Improved Prefetch Accuracy(*Layer-aware burst*)
- 4) Enhanced Compute-IO Parallelism → Improve Throughput/latency per token
- 5) Higher hit ratio leads to reduced system power consumption associated with network access

# Flash Power Consumption : Better in case of long idle time



< System power in DRAM vs. FLASH >



	DRAM	FLASH
Active read energy	5 pJ/b <small>*DDR4 6.4GB/s 64bit</small>	25 pJ/b
Self refresh current	148 mA <small>*DDR4 32Gb (163mW)</small>	0 mA ☺️

DRAM = DDR4 @6.4GB/s, 64bit  
 NAND = Toggle DDR @ 2.4 GB/s, 8bit

< Chip power efficiency : DRAM vs. FLASH>

- Chunk-based access to large-scale data improves amortized energy efficiency via block-wise I/O.
- NAND consumes no standby power for data retention, unlike DRAM.

➔ **NAND scales better for on-demand AI Inference with zero refresh overhead.**



# HBM CENTRIC

# Thank you!

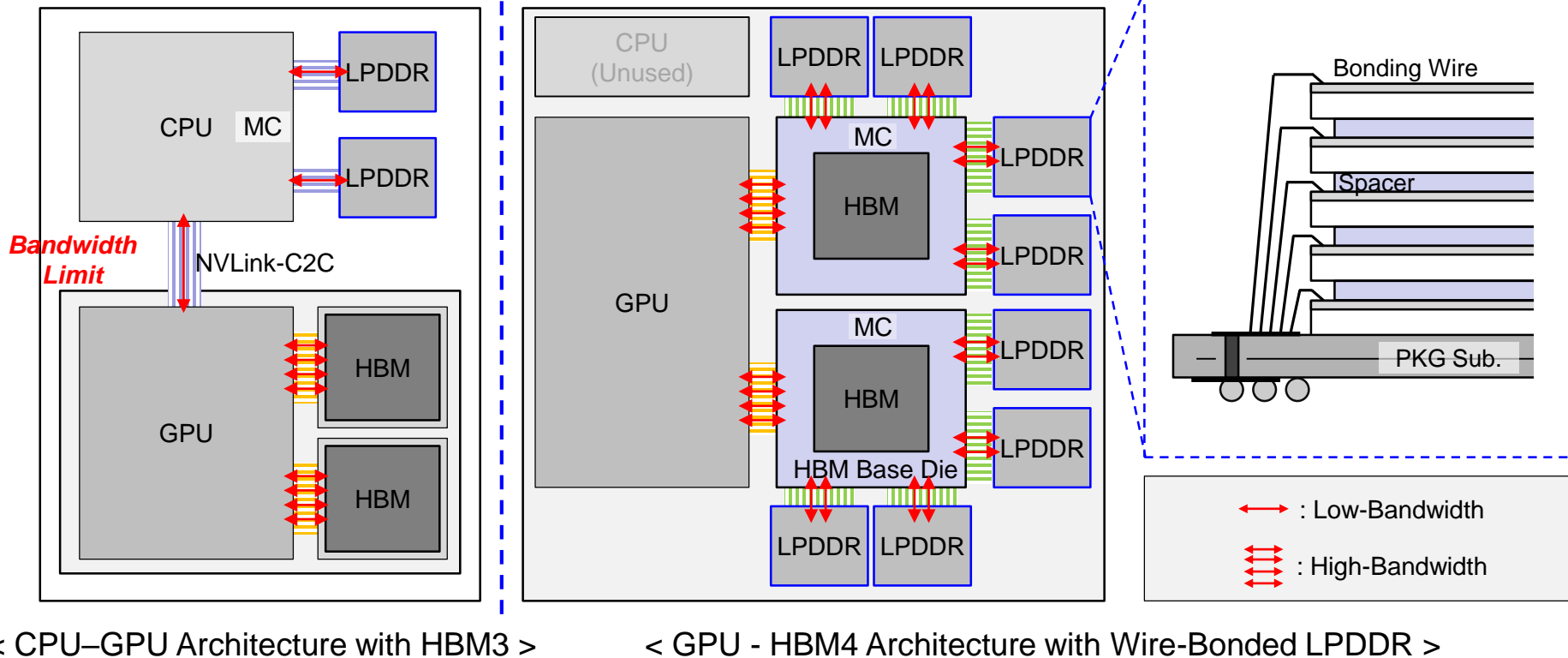
# HBM7 Architecture Integrated with High-Capacity 3D-Stacked LPDDR

Inyoung Choi

Advising Professor : Prof. Joungho Kim

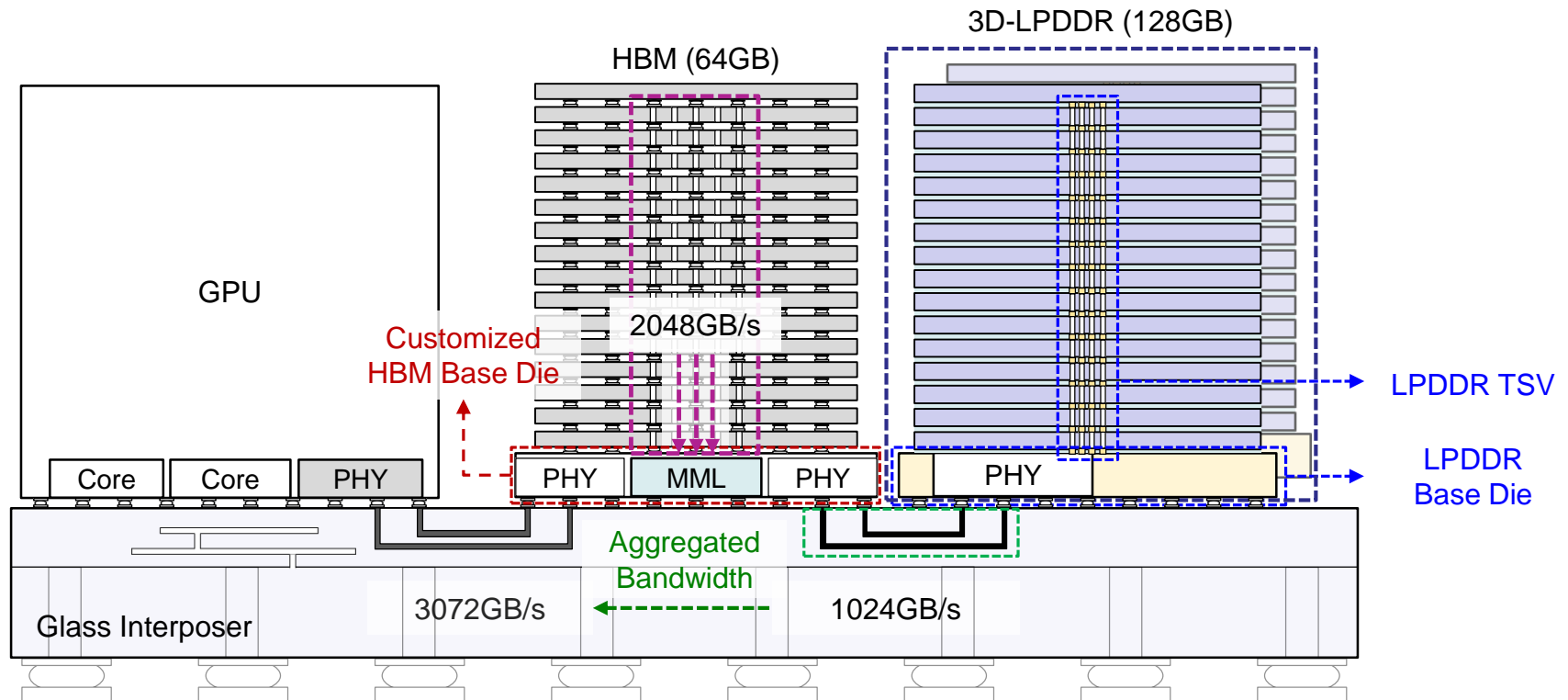
TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

# Limitations of the Previous HBM System Architectures Integrated with Wire-Bonded LPDDR [HBM3 & HBM4]



- HBM3 systems adopted NVLink-C2C instead of PCIe to achieve higher off-package bandwidth.
- Despite this, NVLink-based access still suffers from long physical distance and latency overhead with insufficient bandwidth for memory-bound workload.
- To address this, in HBM4 systems, LPDDR was integrated on silicon interposer, connected to HBM base die integrated with memory controller.
- However, LPDDR still uses wire bonding, which limits I/O density and achievable bandwidth.
- Long and parasitic wire paths degrade signal integrity, making them suboptimal for high-speed memory access.

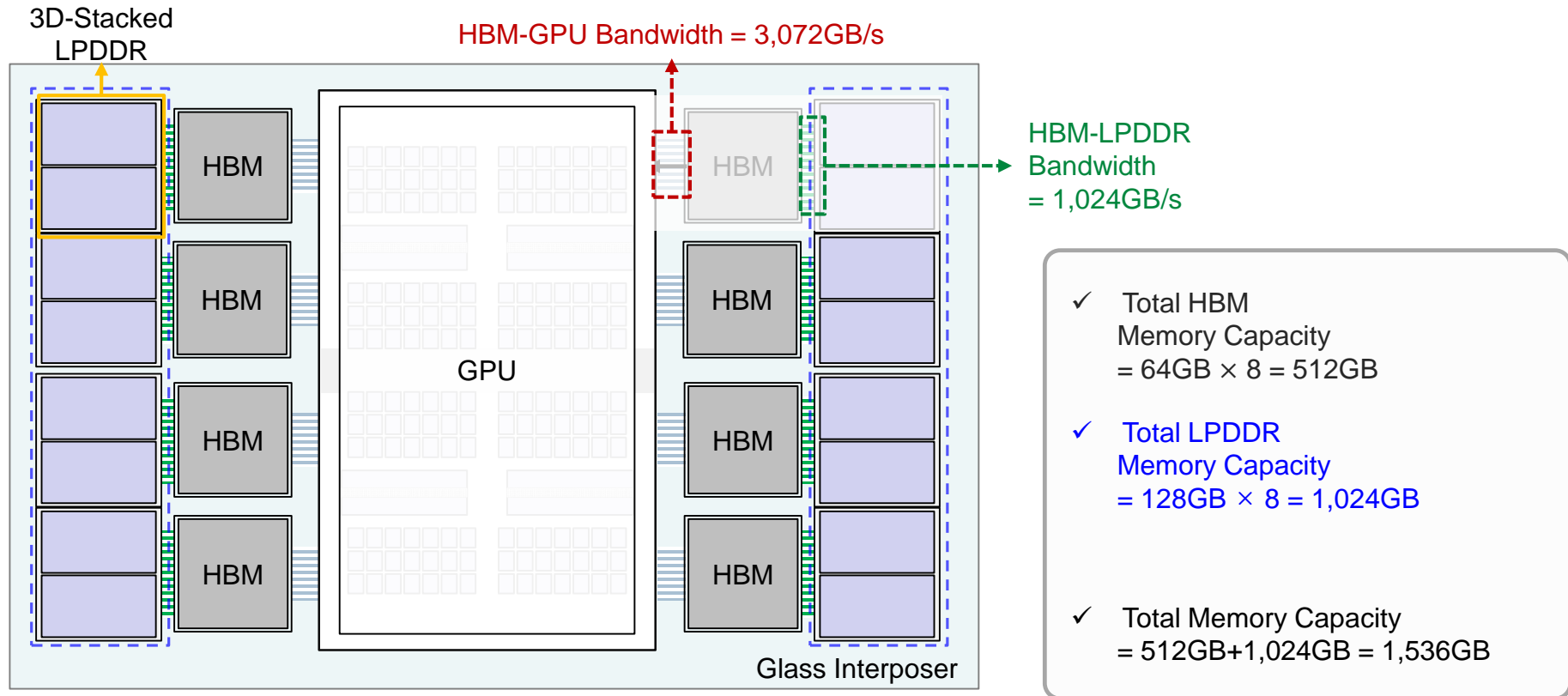
# Proposal of HBM7 Architecture Integrated with High-Capacity 3D-Stacked LPDDR on Glass Interposer



## < Proposed HBM7 Architecture Integrated with 3D-Stacked LPDDR on Glass Interposer >

- The proposed architecture integrates high-capacity 3D-stacked LPDDR alongside HBM stacks on a shared glass interposer .
- Each 3D-LPDDR stack is composed of 16 DRAM dies vertically connected through TSVs, enabling high-bandwidth and energy-efficient memory access.
- Two LPDDR stacks are mounted on a single base die, which interfaces with a customized HBM base die that includes integrated Memory Management Logic (MML) for unified memory access.

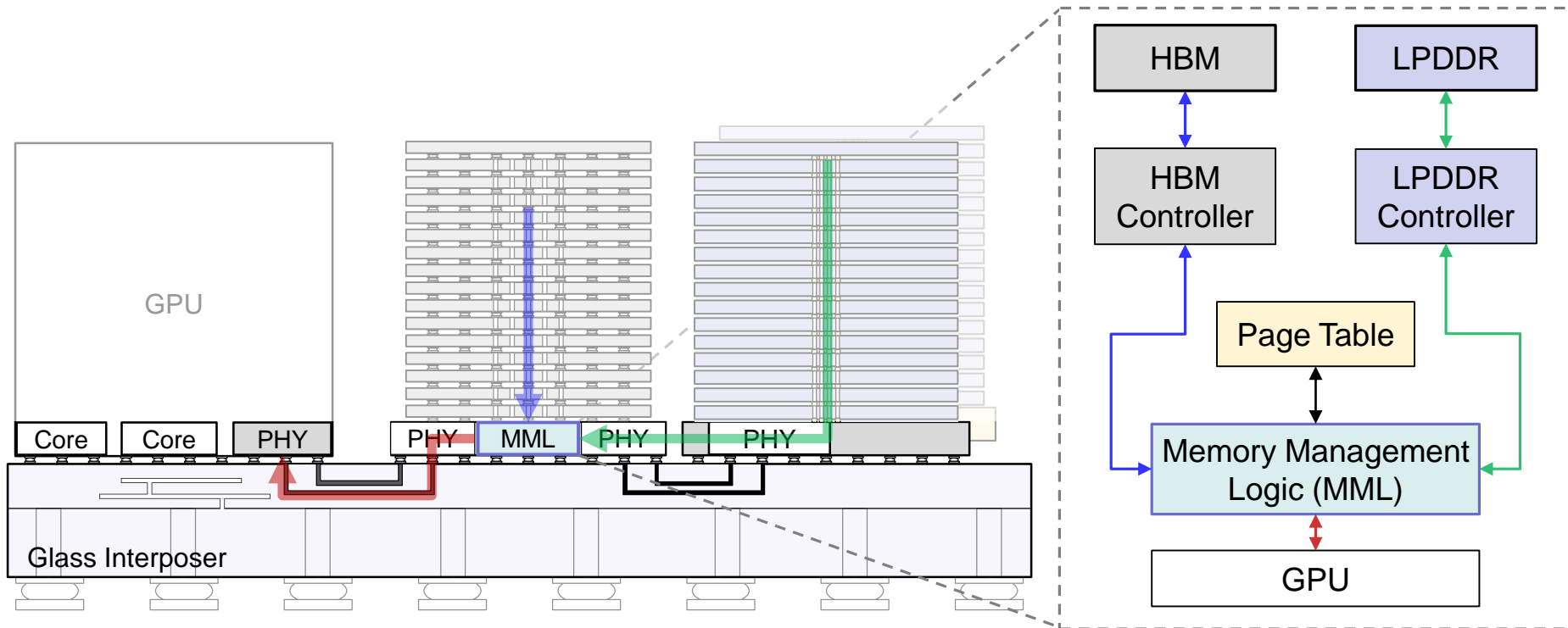
# High-Capacity GPU Memory System with 3D-Stacked LPDDR in HBM7-based Architecture



< Configuration Overview the Proposed HBM7 integrated with 3D-Stacked LPDDR >

- 8 memory modules are symmetrically integrated around the GPU on a glass interposer.
- 3D-stacked LPDDR modules are placed adjacent to the HBM, minimizing latency and reducing interconnect energy.
- The GPU-HBM interface supports up to 3,072GB/s bandwidth, which is the combined bandwidth from HBM and LPDDR aggregated through the MML, enabling data transfer to GPU without bottlenecks.

# Memory Management Logic-based Unified Memory Access in the Proposed Architecture

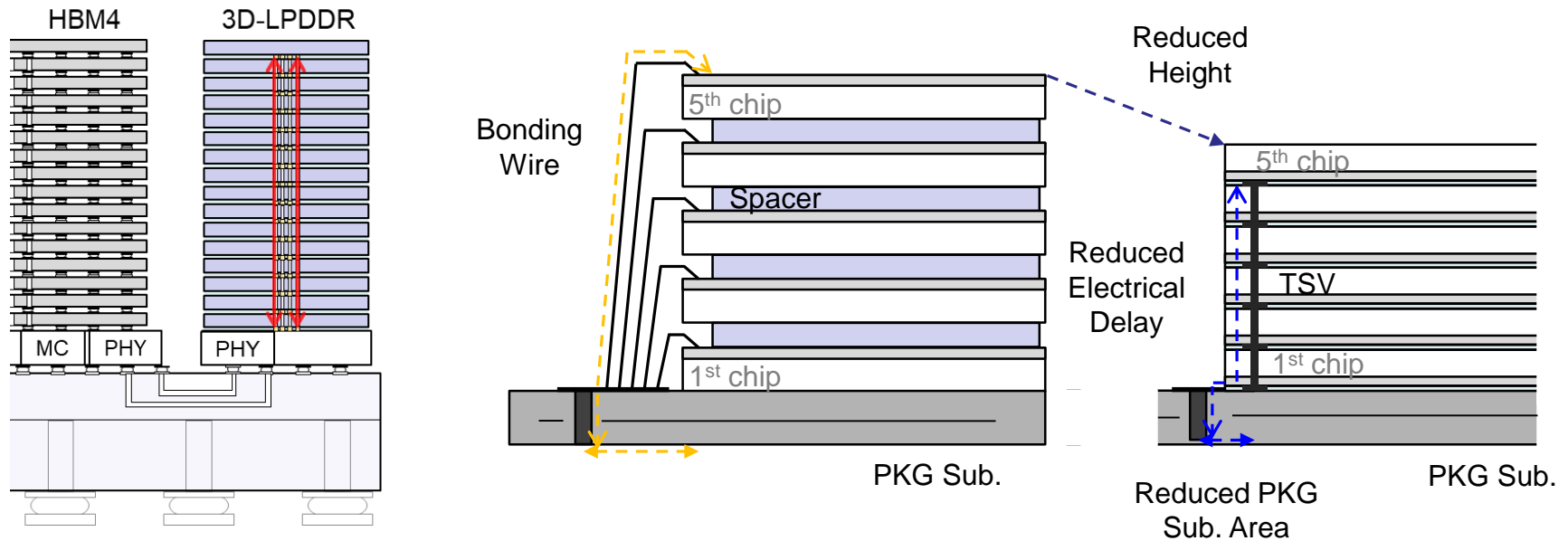


## < Memory Access Flow with Customized HBM Base Die integrated with MML >

- When GPU issues a memory access request, MML references the page table to translate the address and determine the appropriate memory interface.
- MML in the customized HBM base die aggregates memory capacity and bandwidth from both HBM and LPDDR stacks and provides unified memory access to GPU.
- This transparent abstraction relieves GPU from managing memory mapping and management operations, enabling a more scalable, HBM-centric memory architecture.



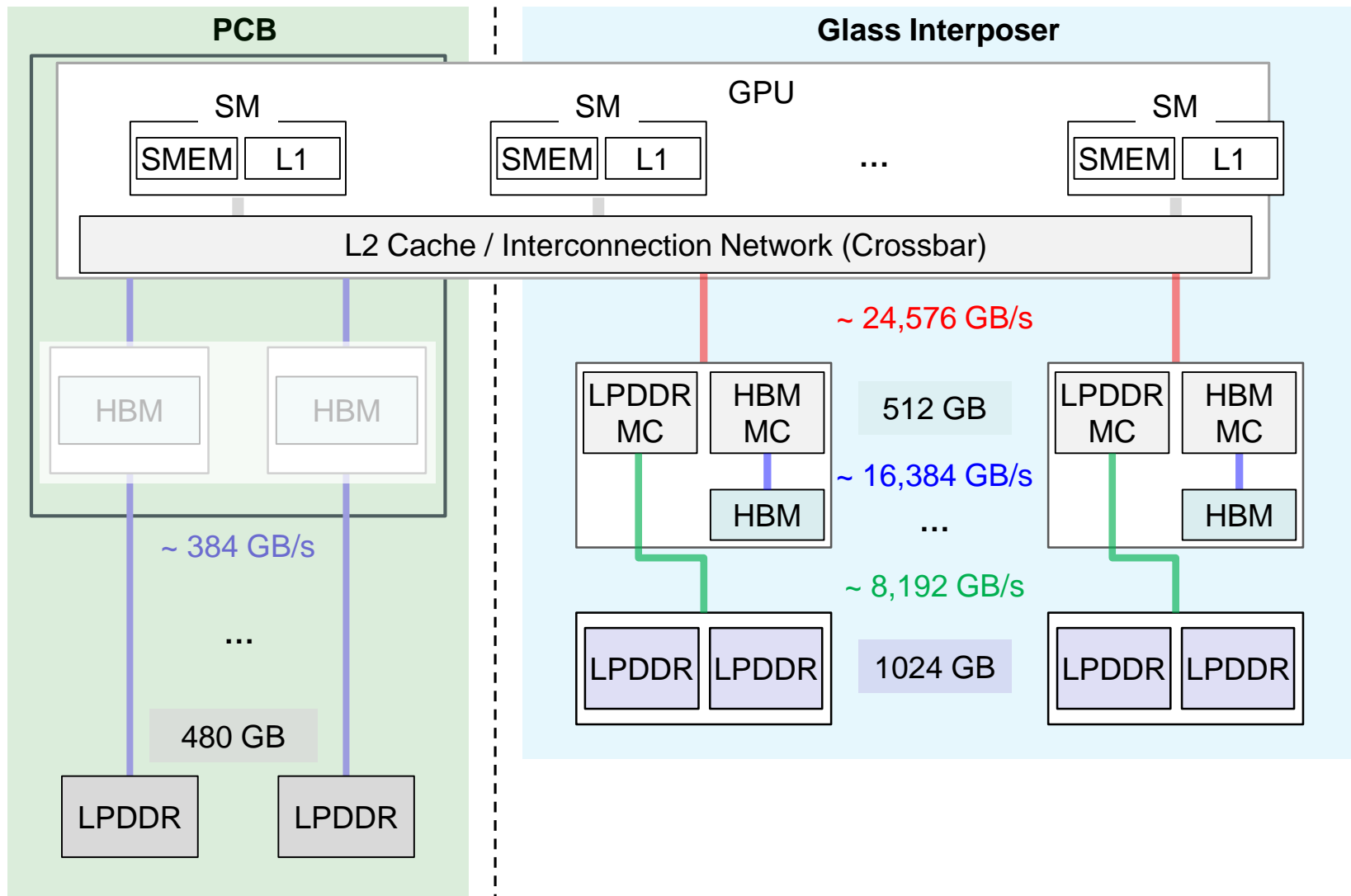
# Advantages of TSV-based 3D LPDDR Stacking based over Conventional Wire Bonding



## < Comparison of TSV and Wire Bonding Interconnection Methods in 3D-Stacked LPDDR >

- In the proposed 3D-LPDDR architecture, TSVs are used to vertically connect multiple DRAM dies.
- Compared to wire bonding, TSVs offer several key advantages:
  - ✓ Shorter interconnect length reduces inductance and energy per bit, improving timing margins.
  - ✓ Enhanced signal integrity with lower delay, crosstalk, impedance mismatch.
  - ✓ Reduced package height and form factor by eliminating bonding loops and spacers.
  - ✓ Higher I/O density within a smaller footprint, enabling greater bandwidth scalability.
- These advantages make TSVs the preferred interconnect choice for high-performance, energy-efficient 3D-stacked LPDDR systems.

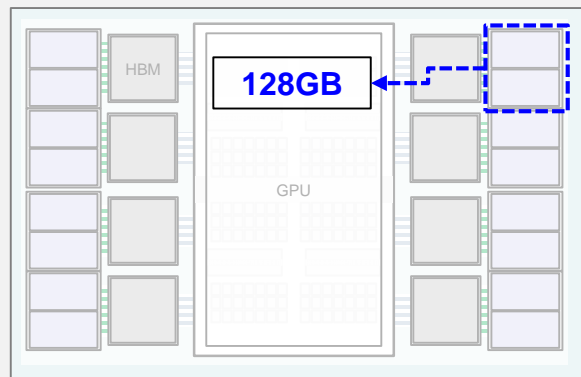
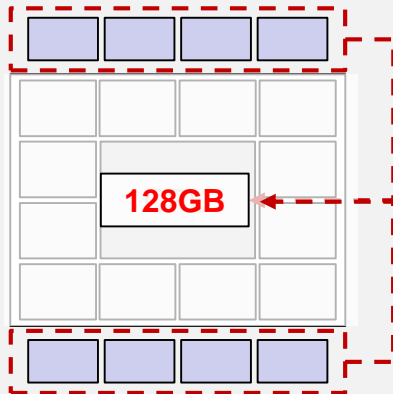
# Overall System Hierarchy of the Proposed HBM7 Architecture Integrated with 3D-Stacked LPDDR



< Overall System Block Diagrams of Conventional (HBM3) and Proposed Architecture >

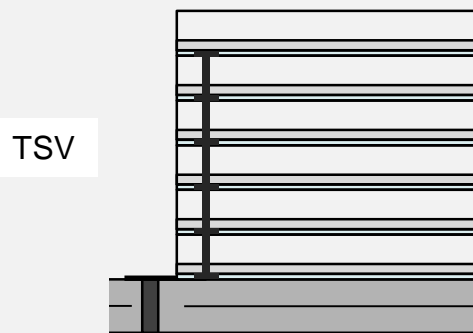
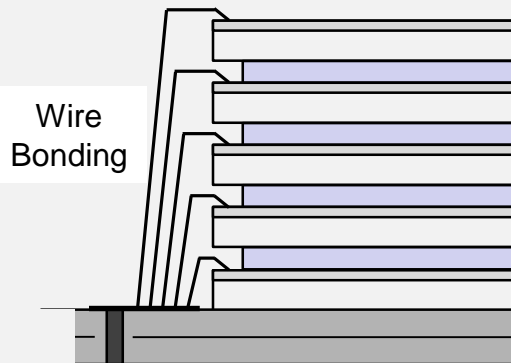
# Key Advantages of the Proposed HBM7 Architecture Integrated with 3D-Stacked LPDDR

High Memory Density in Compact Area



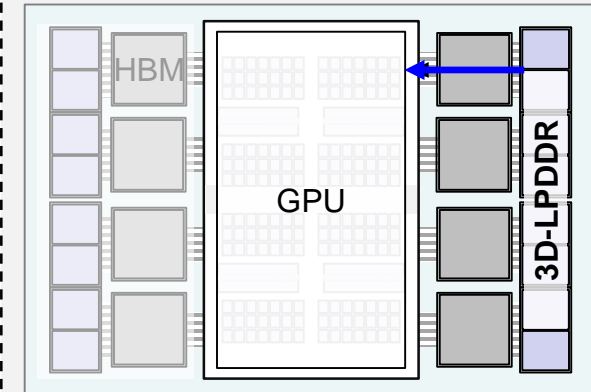
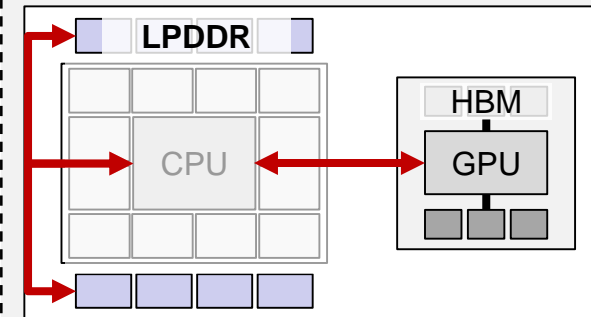
- ✓ High Tokens/sec
- ✓ Room for Other Logic

High TSV Bandwidth and Energy Efficiency



- ✓ Lower I/O Driver Power
- ✓ Smaller Delay

Reduced Off-Package Memory Access



- ✓ Lower Access Latency

< Key Advantages of the Proposed Architecture >

# Conclusion

- Conventional HBM3/HBM4-based memory systems integrated with wire-bonded LPDDR suffer from critical limitations, including long physical distance, limited bandwidth, and high access latency.
- To address these bottlenecks, a HBM7-based architecture was proposed, integrating high-capacity 3D-stacked LPDDR alongside GPU-HBM modules on a shared glass interposer.
- The proposed system employs TSV-based vertical stacking for LPDDR and connects it directly to a customized HBM base die, which incorporates memory management logic for unified access control.
- The proposed architecture enables scalable memory capacity and bandwidth with improved energy efficiency, making it highly suitable for memory-bound workloads such as large language model inference.

# Thank You!

## HBM

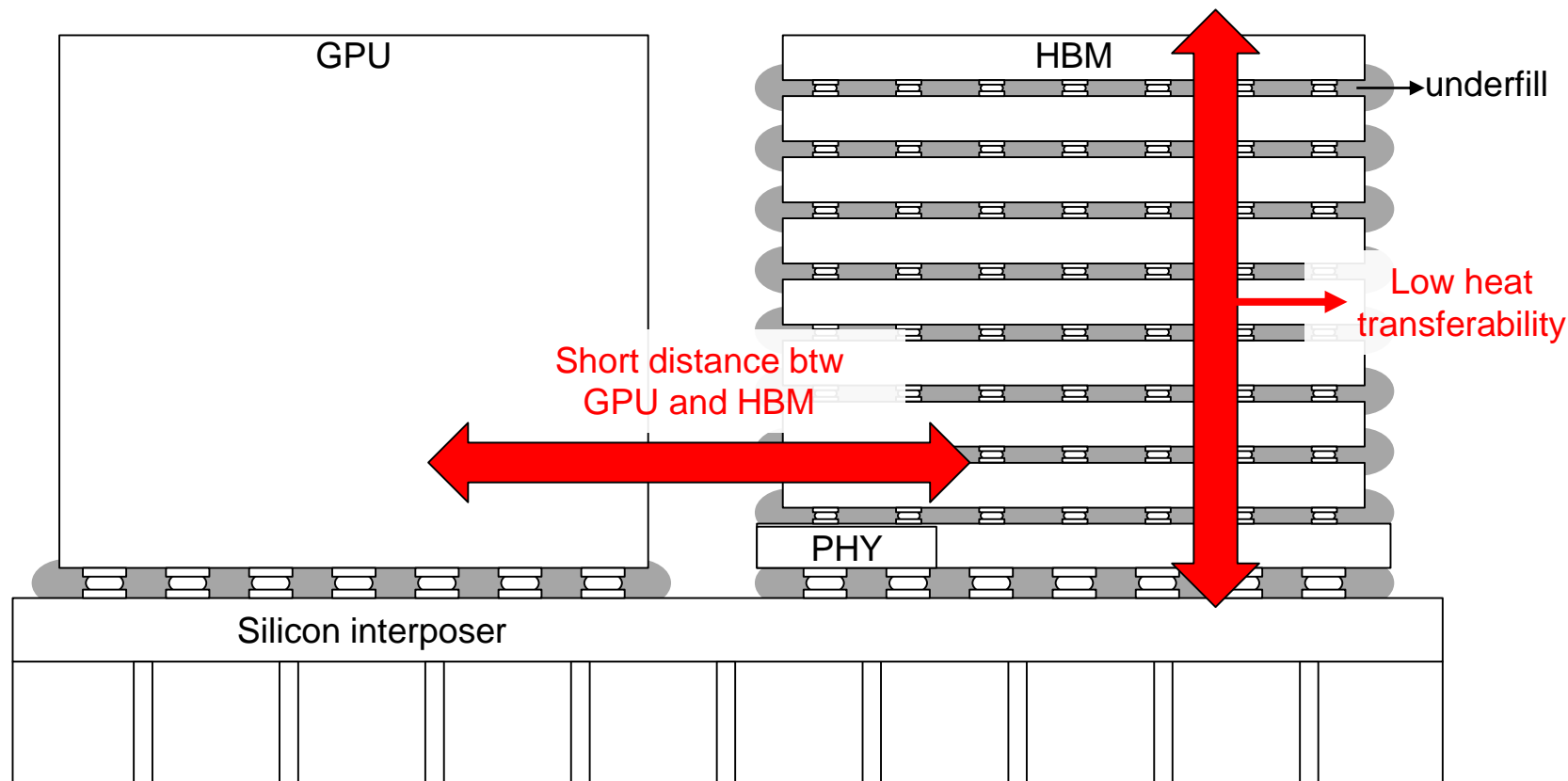
# Embedded Cooling Structure for HBM7 Architecture

Keeyoung Son

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

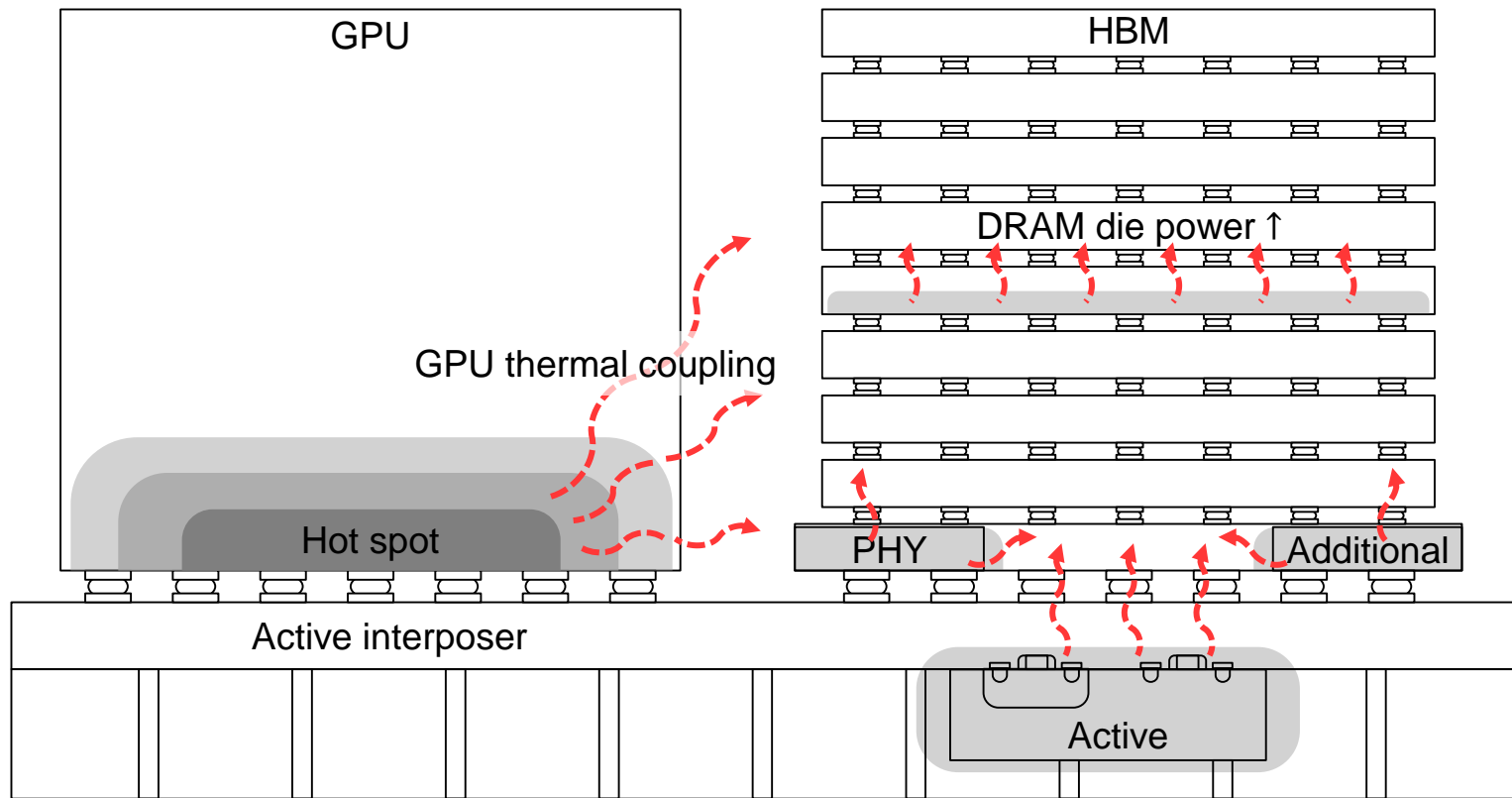
# Thermally Vulnerable Characteristics of HBM



## < Thermally vulnerable characteristics of HBM >

- HBM has thermally vulnerable characteristics due to silicon interposer and 3D structure.
- HBM is thermally vulnerable because it is located close to GPU, which consumes a lot of power, on silicon interposer.
- Furthermore, each HBM die is connected by underfill and microbumps, therefore HBM has low vertical heat transferability.

# Dominant Factors of Thermal Issues in Next-generation HBM Module

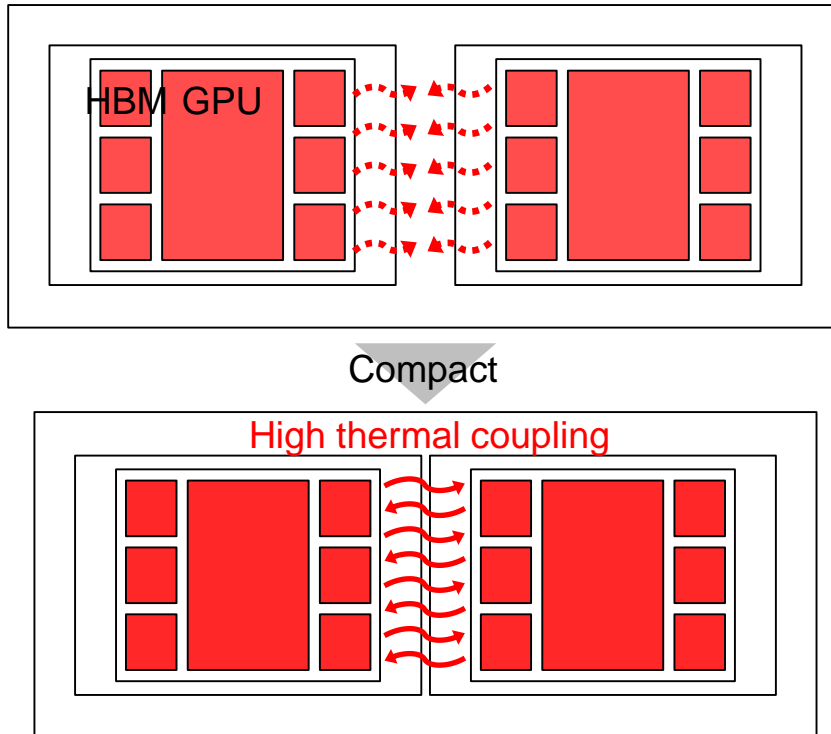


< Dominant factors of thermal issues in next-generation HBM module >

- Dominant factor of thermal issues in previous HBM is thermal coupling from GPU to HBM.
- Especially, in the case of HBM7, power consumption per each DRAM die cannot be ignored due to the increased data rate.
- Moreover, adapting additional function on HBM and advanced interposer generates additional power consumption.



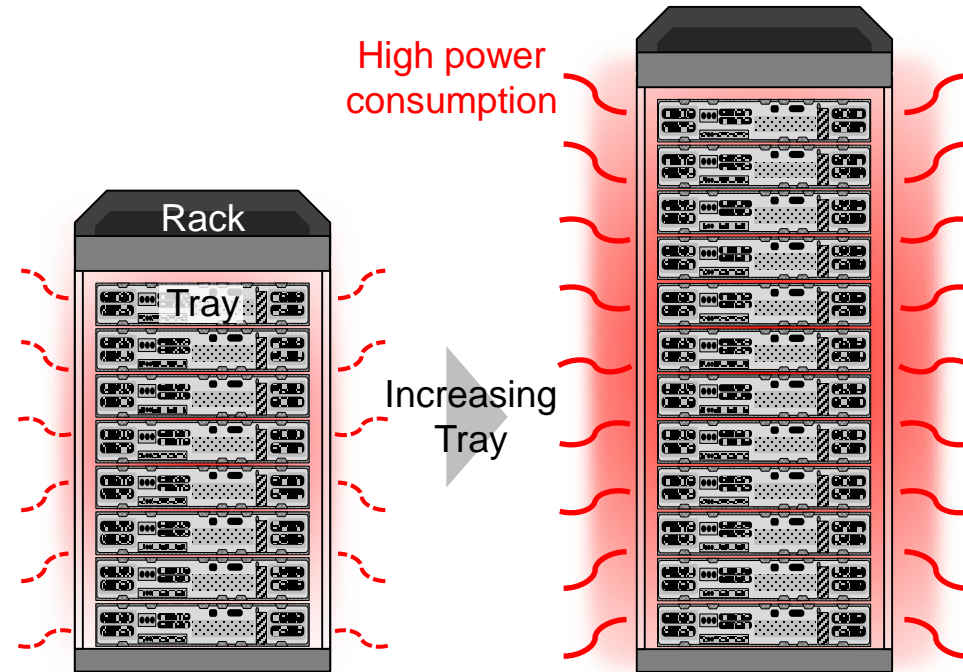
# Thermal Issues Arising from Enhanced Computing Density with Multi-GPU-HBM based AI Supercomputer



< Thermal issues from compact design of multi-GPU-HBM compute module >

$$(Thermal\ coupling) \propto \frac{1}{(Distance)}$$

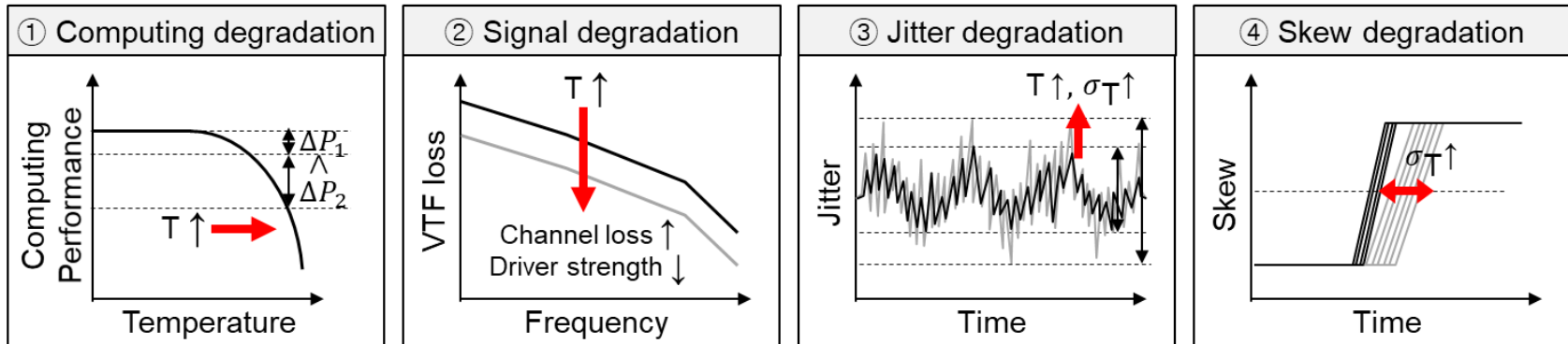
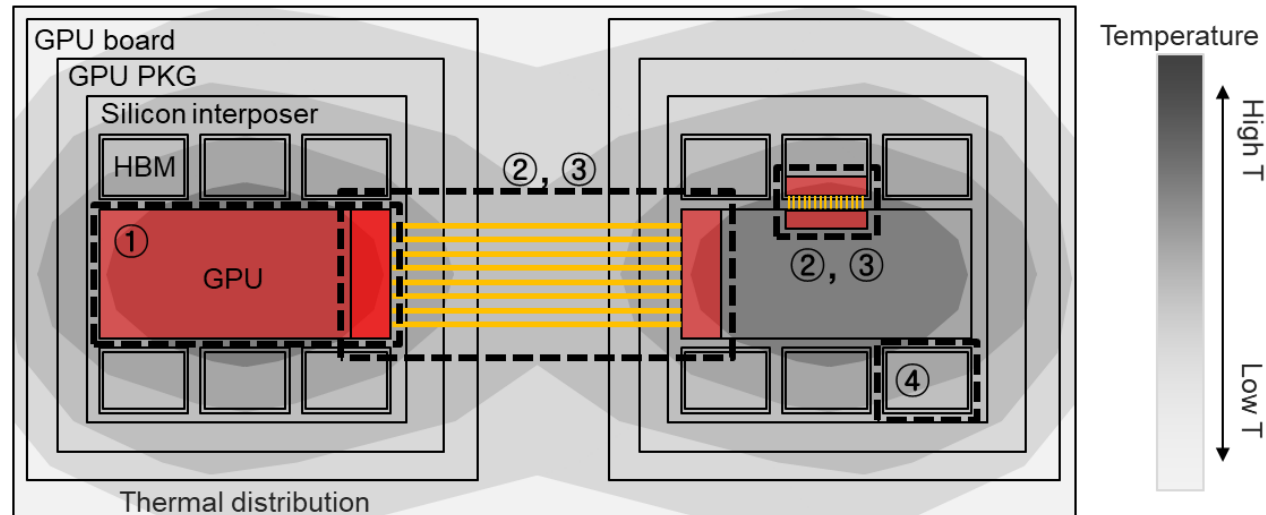
- Integrate each GPU-HBM closely for increase computing density, it increases the thermal coupling between each GPU-HBM which worsens thermal reliability of compute module.
- As increasing the number of compute nodes within AI supercomputer to increase computing density, it lowering the cooling performance of conventional rack-level cooling.



< Thermal issues from increasing number of compute tray in rack of AI supercomputer >

$$(Cooling\ performance) \propto \frac{(Cooling\ capacity)}{(Power\ consumption)}$$

# Necessity of Powerful and Uniform Cooling System for Multi-GPU-HBM based Computing Modules

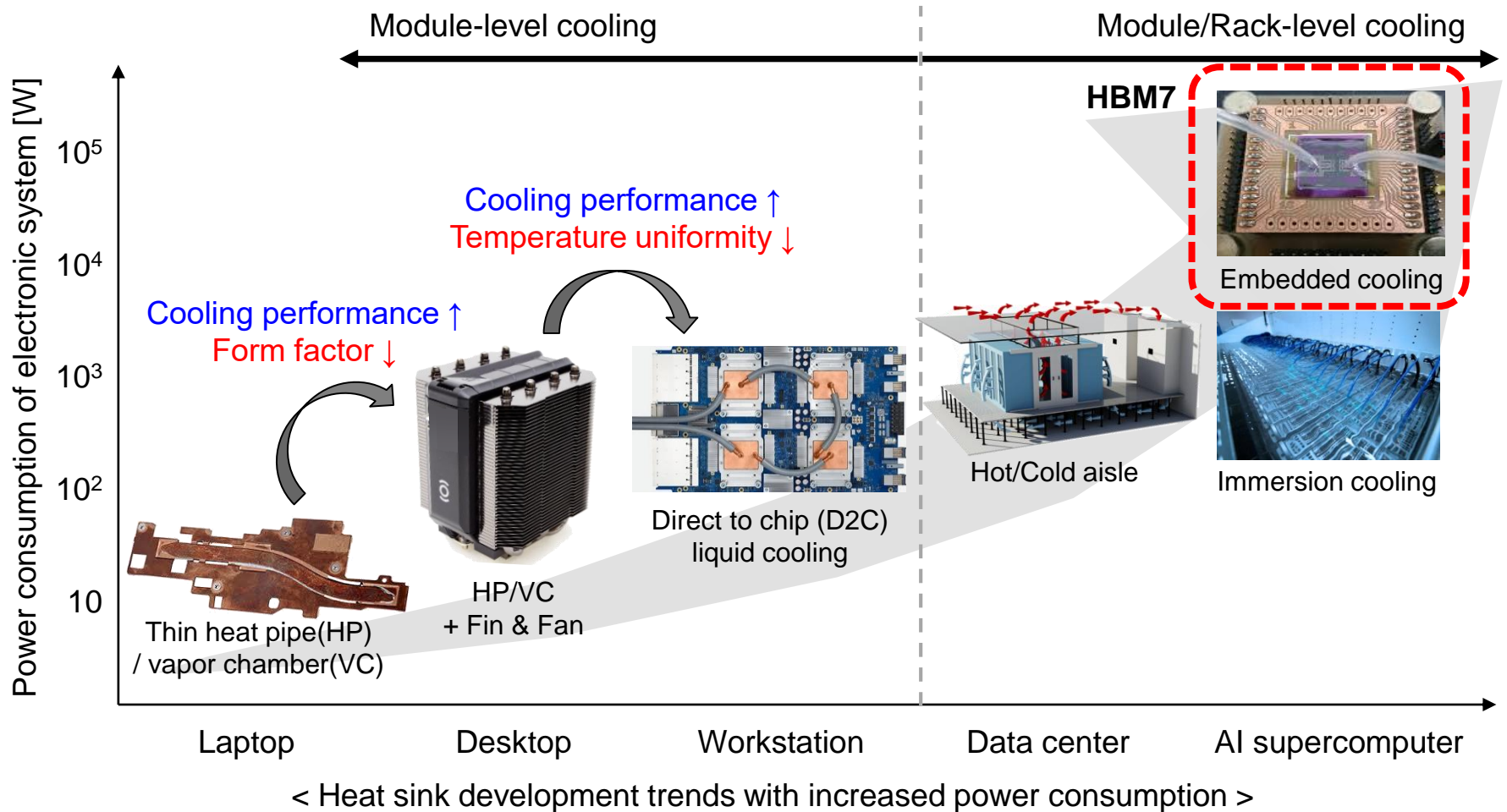


\* T = Temperature

< Temperature-dependent electrical issues on multi-GPU-HBM compute module >

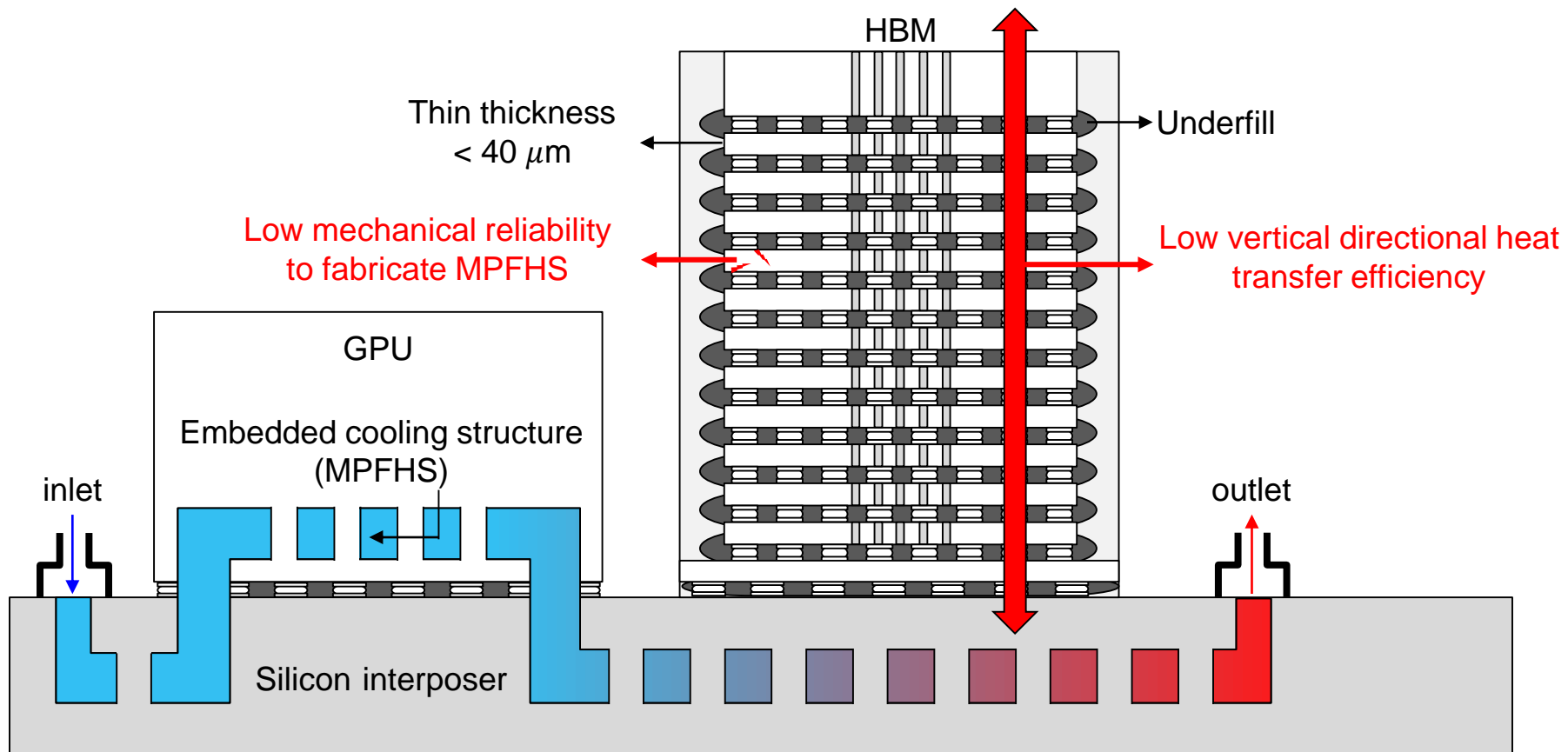
- For computing performance and signal integrity, multi-GPU-HBM compute module requires powerful and uniform cooling.

# Heat Sink Development with Increased Power Consumption



- As power consumption of system has increased, heat sink developed for high cooling performance.
- From HBM7, it requires embedded cooling structure (ECS) to guarantee its thermal reliability.

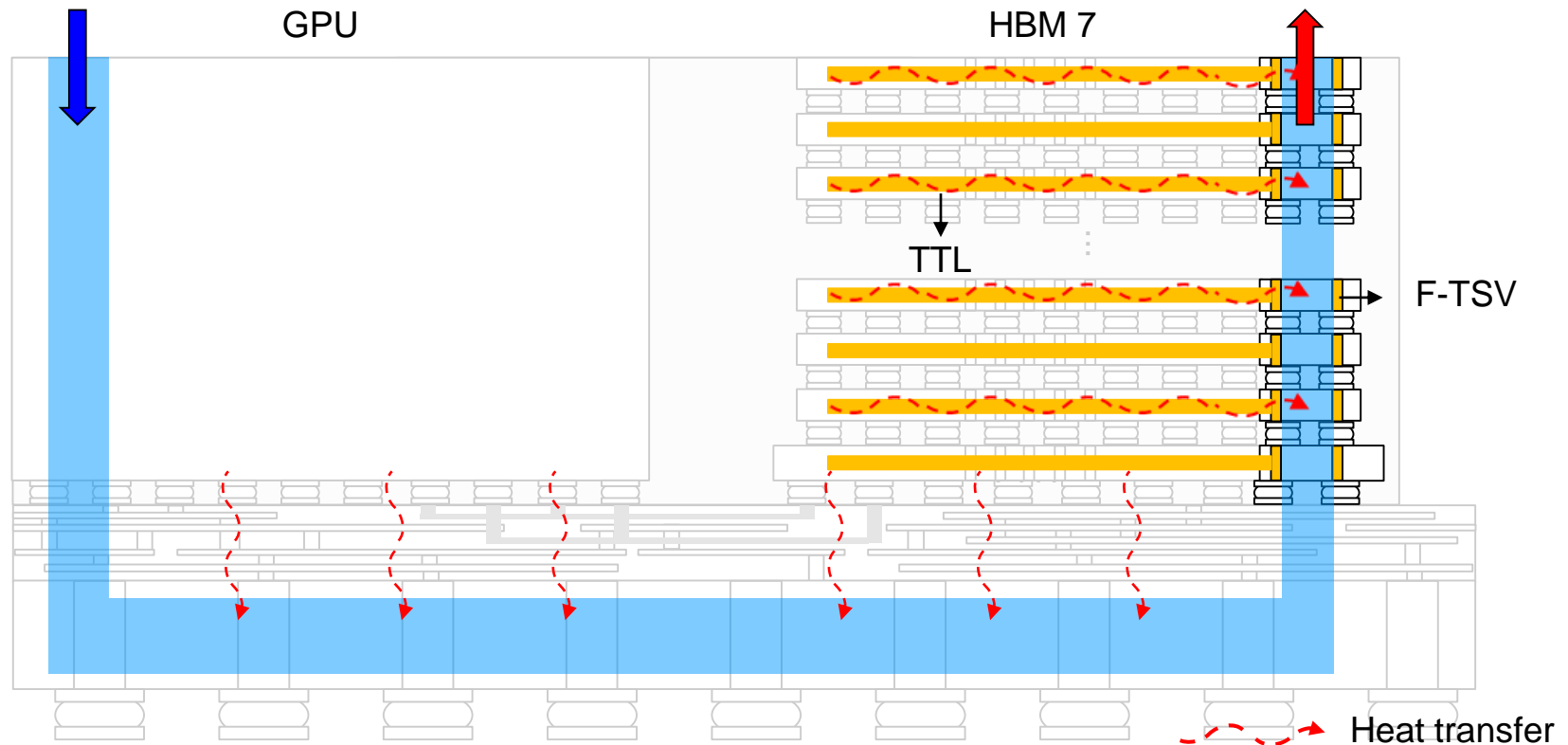
# Thermal Management Limitation of Conventional Embedded Cooling Structure for HBM7



< Thermal management limitation of conventional embedded cooling structure for HBM7 >

- Due to each HBM dies are too thin, low mechanical reliability caused difficult to adapt embedded cooling structure (ECS) based on the micropin-fin heat sink (MPFHS) structure directly.
- High stacked underfill degrades the vertical directional heat transferability from HBM to flowing fluids.

# Proposed Embedded Cooling Structure with Thermal Transmission Line (ECS-TTL) for HBM7

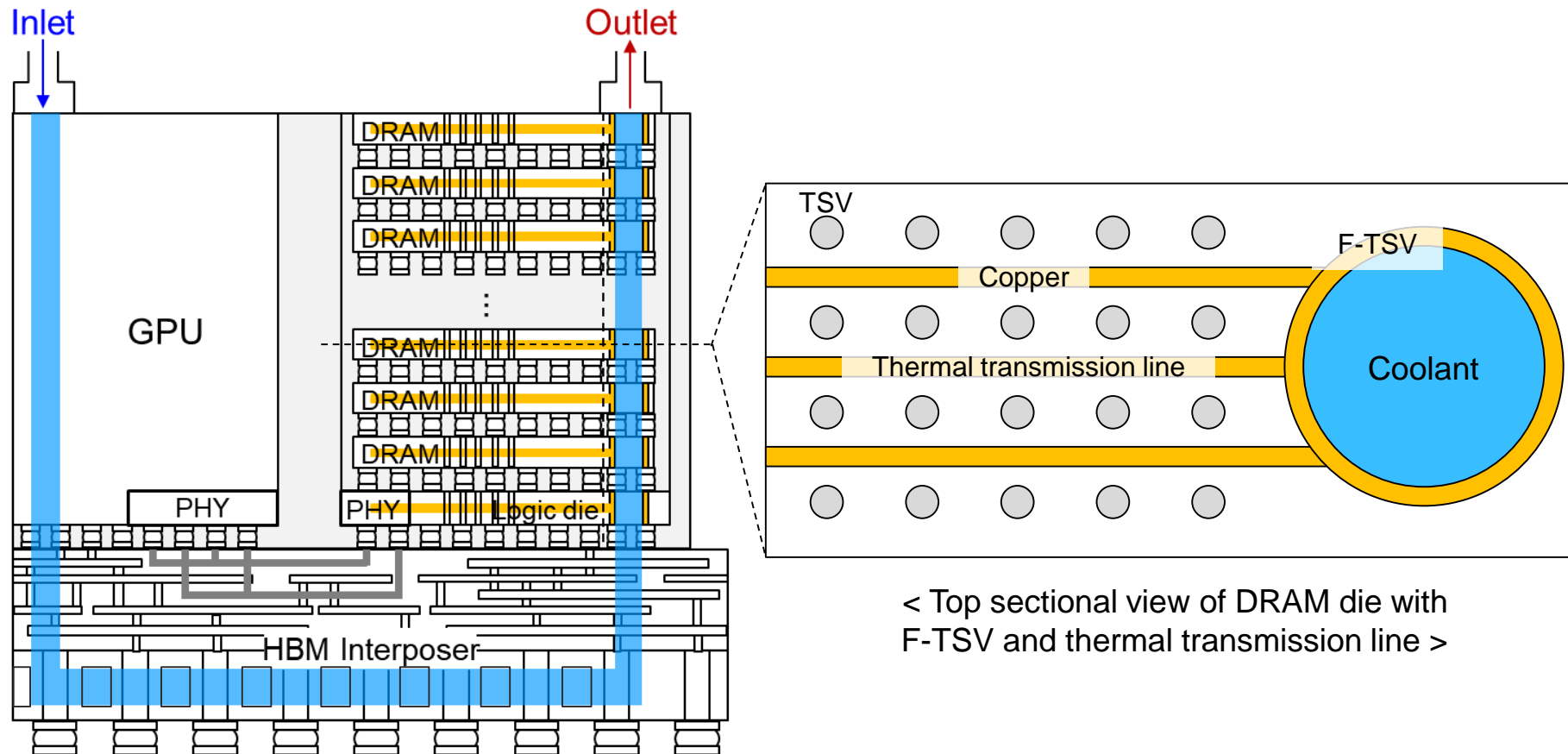


< Concept of the proposed embedded cooling structure with thermal transmission line for HBM7 >

- The proposed ECS-TTL can cool down HBM7 efficiently by circulates fluid through GPU to interposer and HBM.
- Additionally, the proposed TTL transfer internal heat of each HBM die to fluid flowing inside the proposed Fluidic-TSV (F-TSV).

# TTL and F-TSV based Embedded Cooling Structure for HBM Module

## – Thermal Transmission Line (TTL)

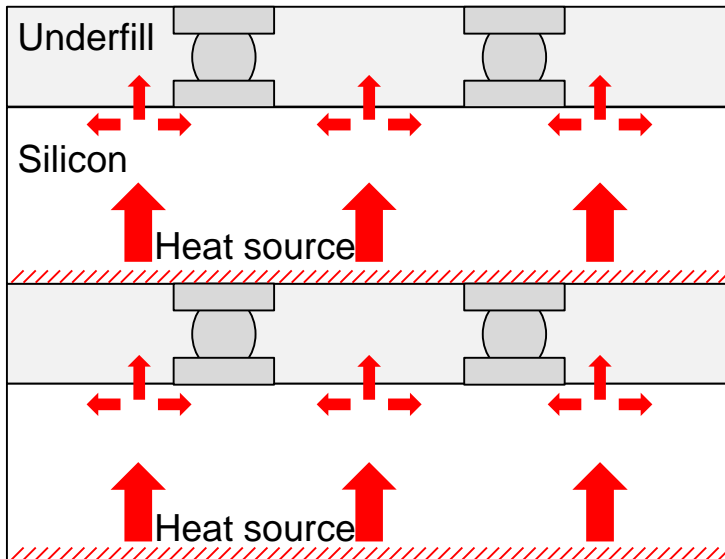


< Top sectional view of DRAM die with F-TSV and thermal transmission line >

- Since each of HBM dies is thin, liquid cooling inside each dies is difficult.
- Thermal transmission lines (TTL) are connected to copper coated around F-TSV for increasing horizontal directional heat transferability from overall die to fluid.
- Therefore, the proposed TTL reduces the temperature standard deviation of HBM.

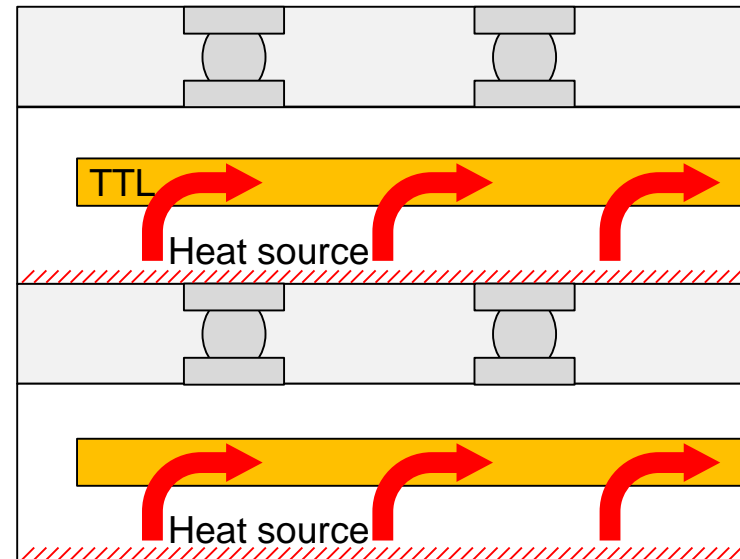
# Thermal Path of Stacked Die depending on TTL

→ : Heat Transfer



< Thermal path of stacked die without TTL >

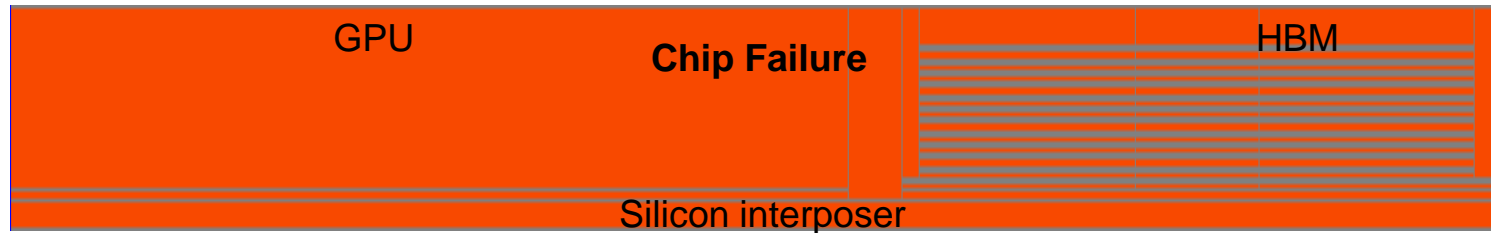
→ : Heat Transfer



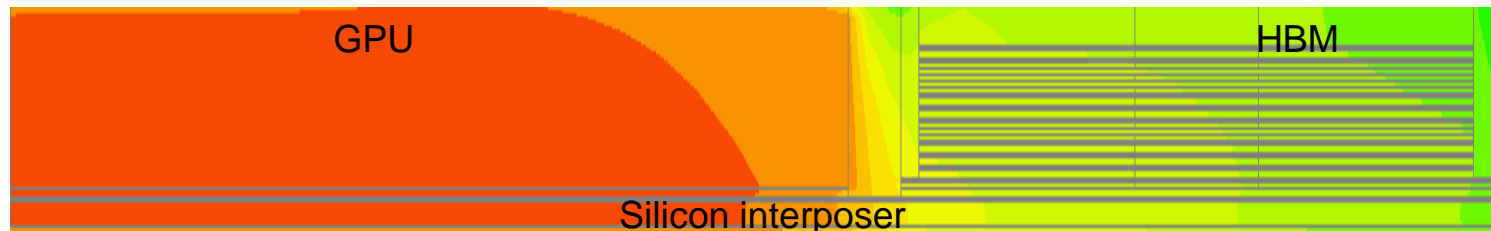
< Thermal path of stacked die with TTL >

- Thermal path of stacked die without TTL is shorted when heat transfers to underfill, because of low thermal conductivity of underfill.
- Therefore, underfill caused heat accumulation in HBM dies.
- However, TTL prevents heat accumulation of HBM dies by transfer internal heat of dies horizontally.

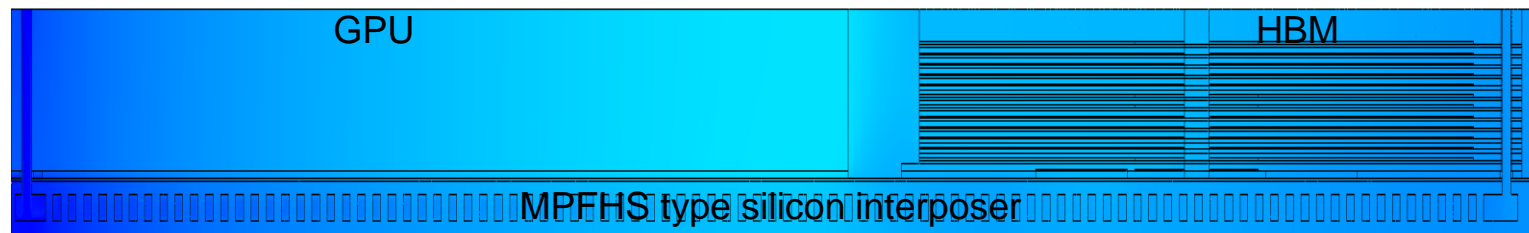
# Temperature Distribution Comparisons of HBM Module with Various Cooling Conditions



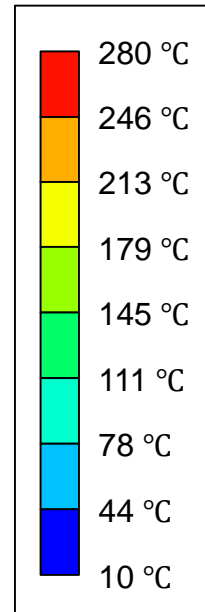
< Temperature distribution of HBM module without cooling structure >



< Temperature distribution of HBM module with fin and fan cooling structure >



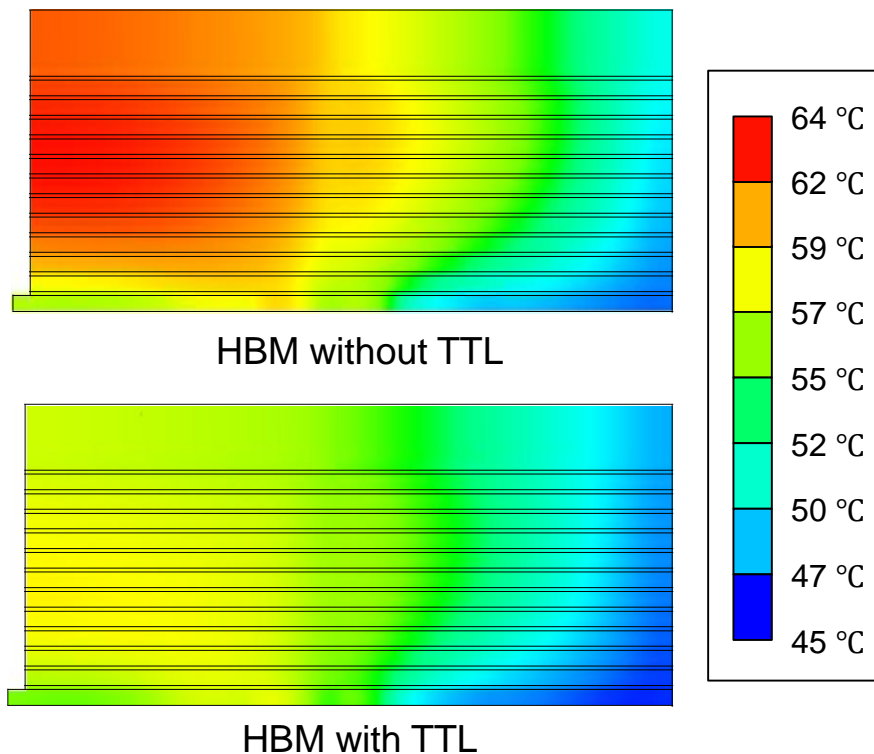
< Temperature distribution of HBM module with embedded cooling structure >



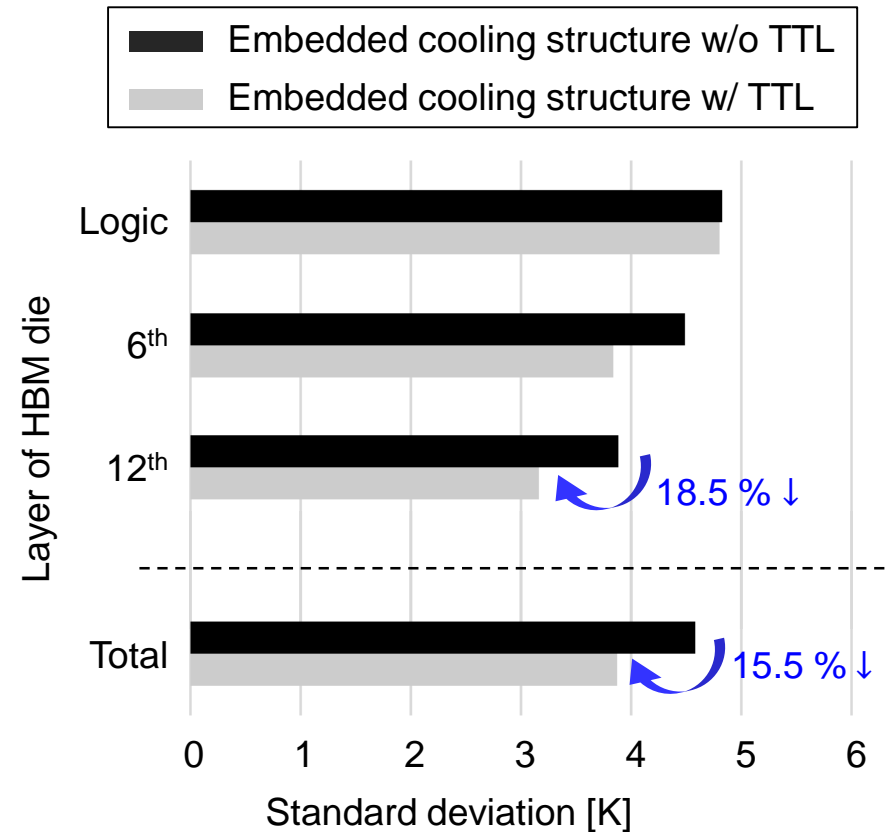
Junction Temperature [°C]	GPU	HBM
w/ fin and fan cooling structure	276.486 $\rightarrow$ 205.2 °C ↓	226.253 $\rightarrow$ 166.5 °C ↓
w/ embedded cooling structure (w/ TTL)	71.308 $\leftarrow$	59.719 $\leftarrow$



# TTL Effects on Standard Deviation of HBM Temperature Distribution



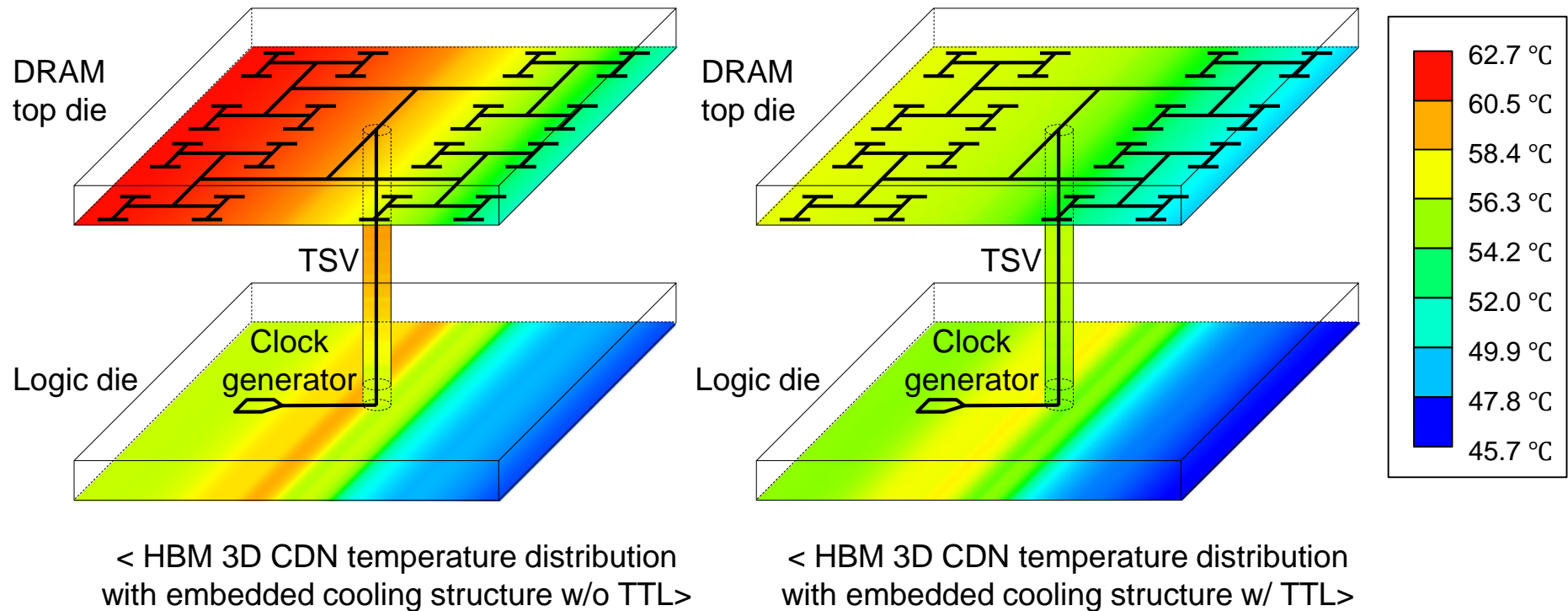
< Temperature distribution of HBM with embedded cooling structure >



< Temperature standard deviation of HBM >

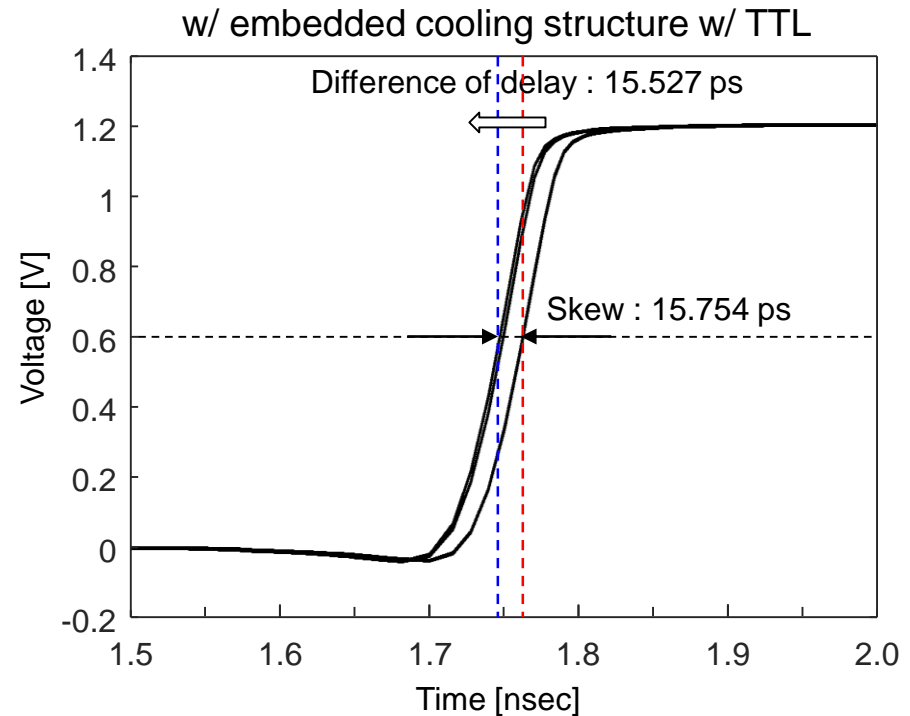
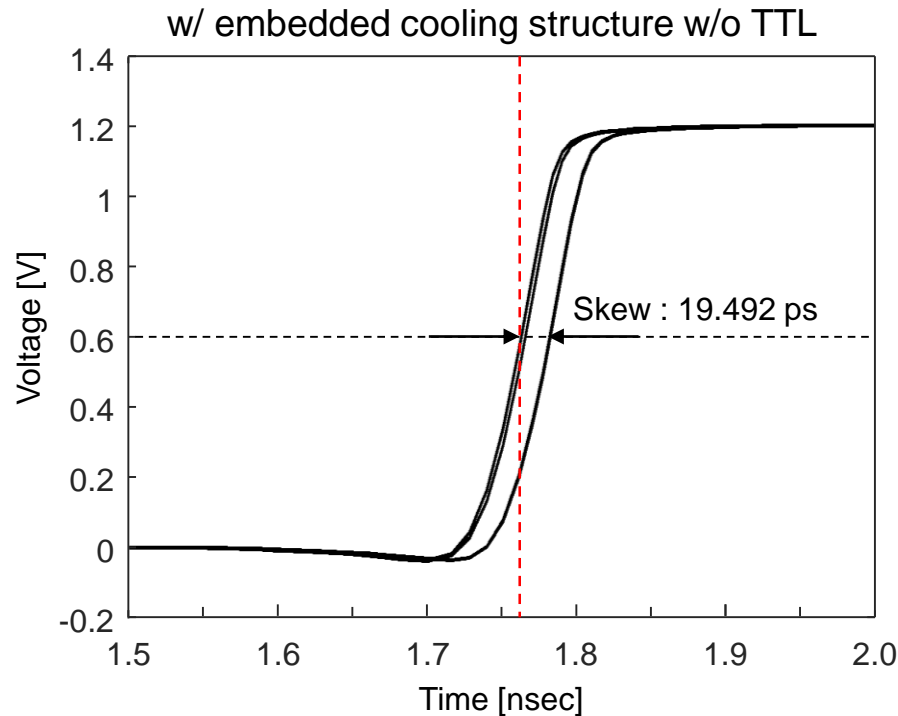
- Because TTL increases the horizontal directional heat transfer, the standard deviation of each HBM die's temperature distribution decreased.
- The improvement of standard deviation of temperature distribution of each die and the total HBM is up to 18.5% and 15.5%, respectively.

# HBM 3D CDN with Temperature Distributions of DRAM dies depending on TTL



- Temperature distribution causes variations of channel characteristics, repeater strength and etc.
- Analyzed the skew of HBM 3D CDN along logic die to DRAM top die because differences of standard deviation of temperature distributions at DRAM top die is largest.

# Temperature-dependent Skew and Delay Evaluation of HBM 3D CDN depending on TTL



< Temperature-dependent transient wave form of HBM 3D CDN with embedded cooling structure >

- Skew of HBM 3D CDN with TTL decreased as 19.18 % rather than without TTL.
- Furthermore, delay of HBM 3D CDN with TTL is smaller than without TTL, and the difference of delay is 15.53 ps.

# Thank You!

## HBM

# 3D Memory Expansion Architecture for HBM8 with Double-Sided Interposer and HBM-HBF-LPDDR Integration

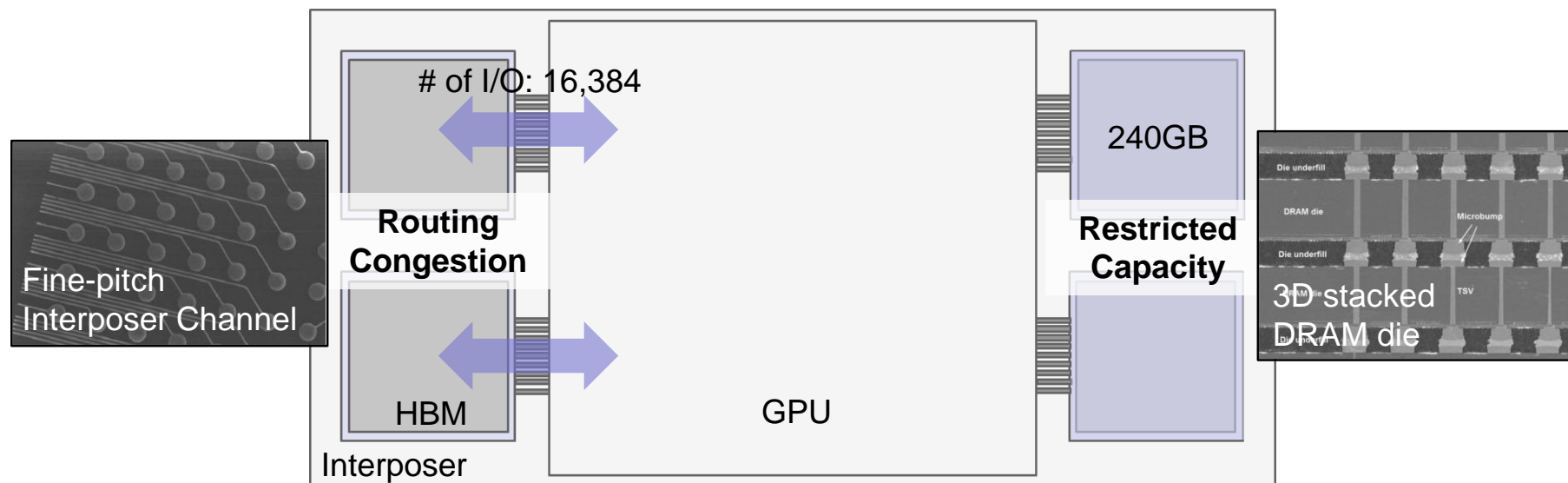
**Chaemin Yang**

Advising Professor : Prof. Jounggho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

June. 11<sup>th</sup>, 2025

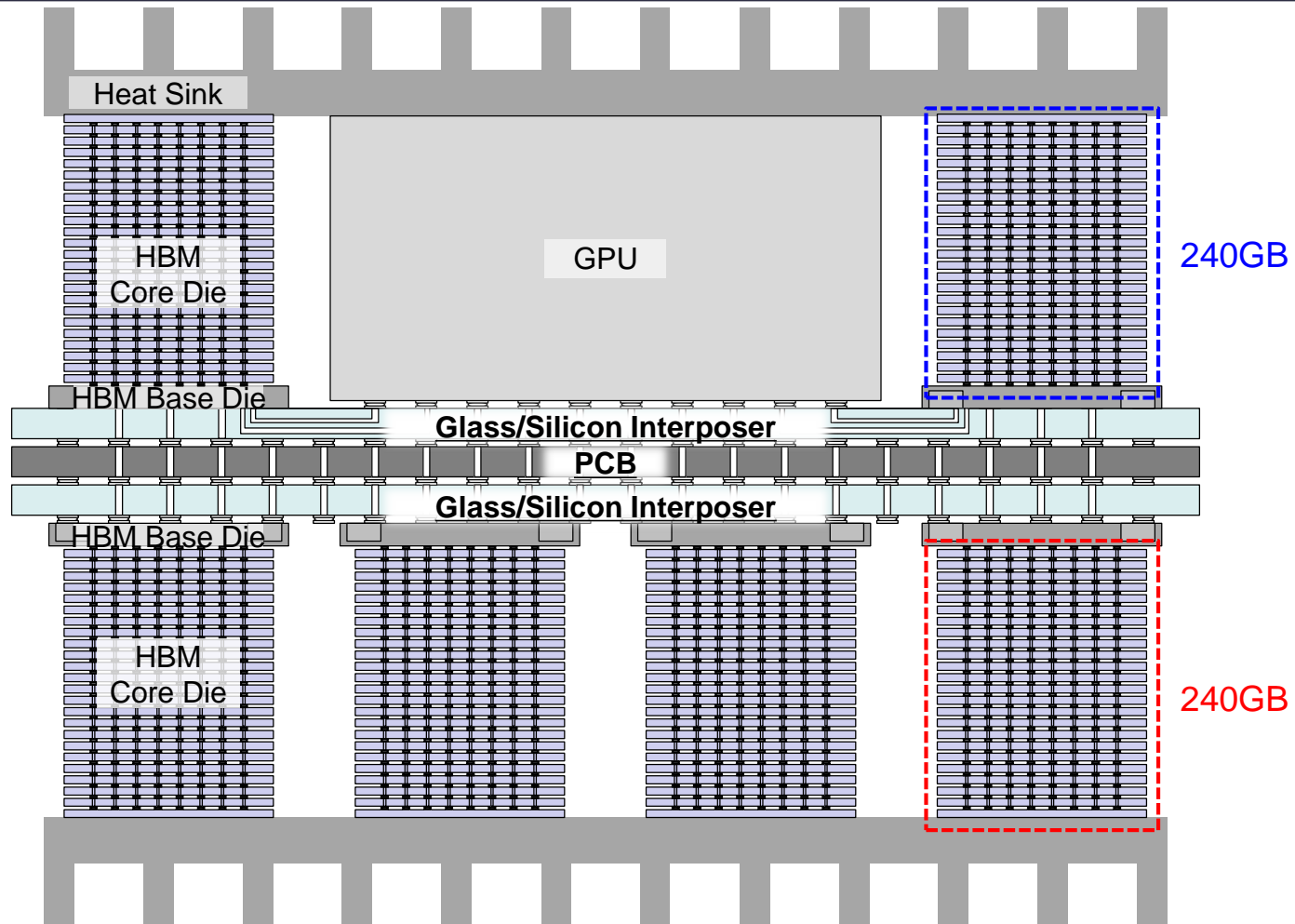
# Limitations of Previous HBM Generations and Motivation for 3D Expansion



## < Architectural Limit of Conventional 2.5D Interposer-Based GPU-HBM Integration >

- Horizontal interposer-based connections face routing congestion, limiting bandwidth scalability.
  - HBM-only stacking restricts flexible capacity expansion for heterogeneous memory demands.
  - Thermal and power inefficiencies arise in dense 2.5D packages
- These issues demand a new vertical memory integration strategy with modular capacity and thermal-aware design flexibility.

# 3D Memory Expansion Architecture for HBM8 with Double-Sided Interposer [1/3]: GPU-HBM-HBM

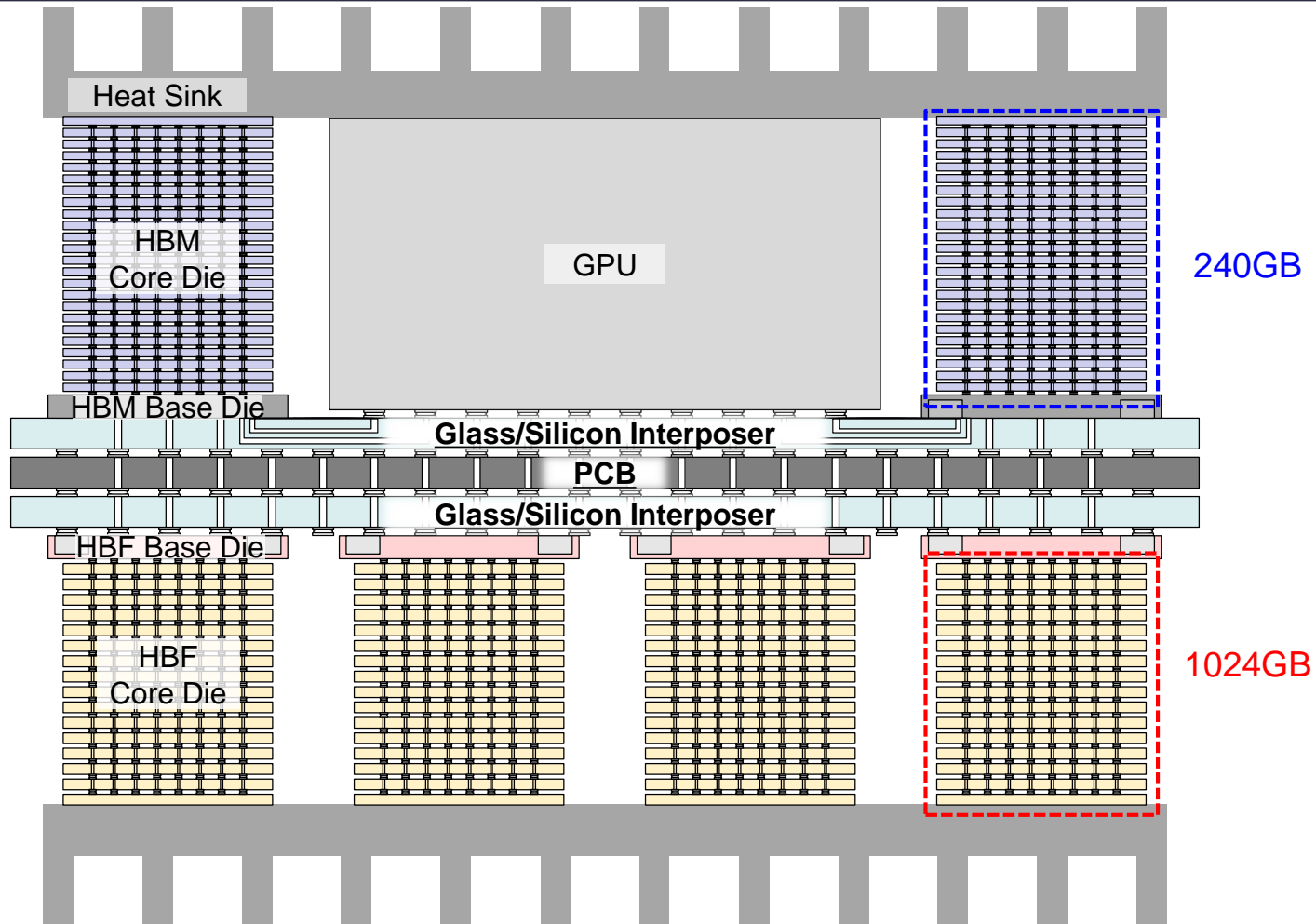


< 3D Memory Expansion Architecture for HBM8 with Double-Sided Interposer using HBM >

- This structure expands memory capacity by vertically stacking HBM dies on both sides while maintaining bandwidth symmetry.



# 3D Memory Expansion Architecture for HBM8 with Double-Sided Interposer [2/3]: GPU-HBM-HBF

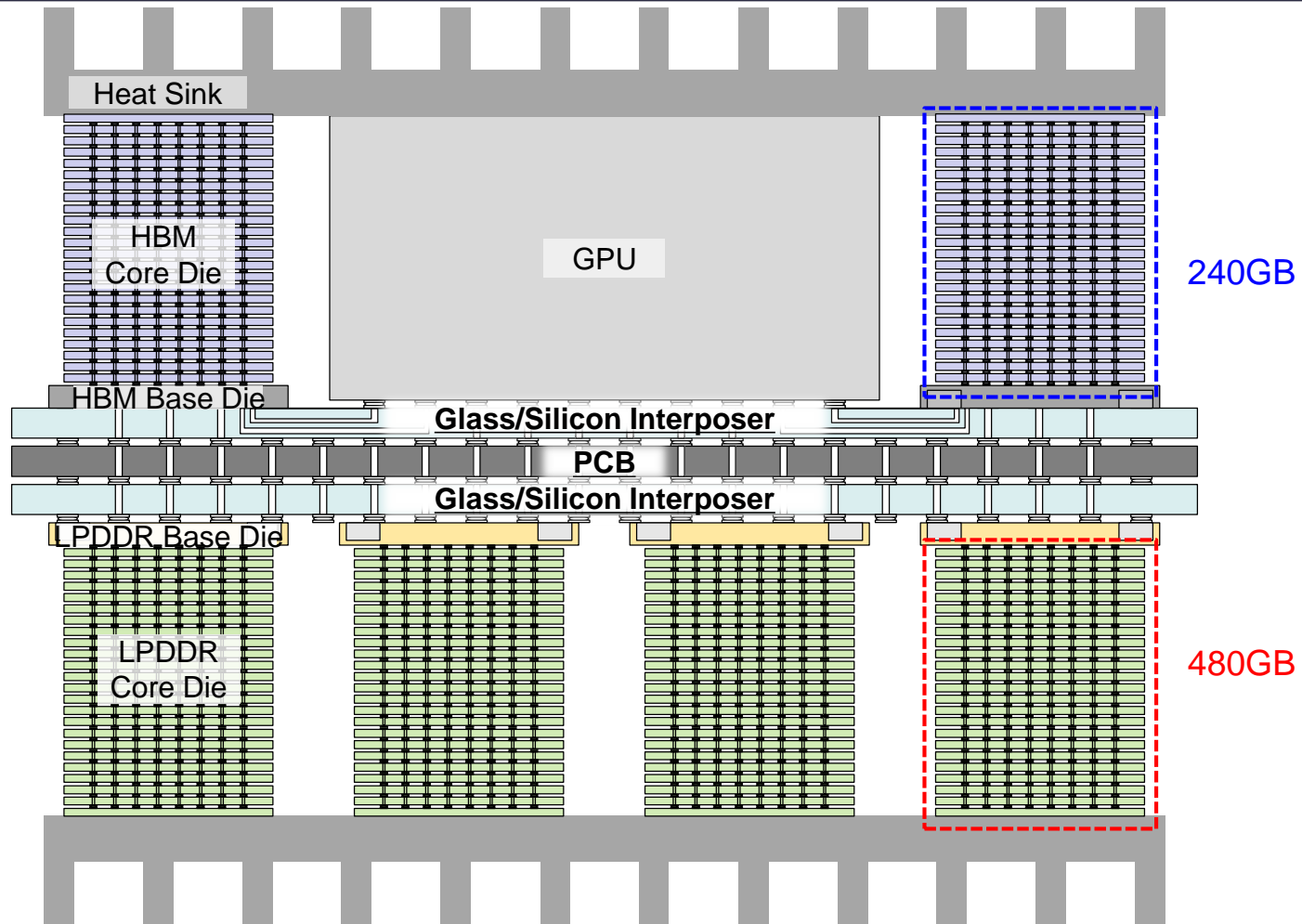


< 3D Memory Expansion Architecture for HBM8 with Double-Sided Interposer using HBF >

- By replacing one side with HBF, the system significantly increases total memory capacity while preserving uniform bandwidth.



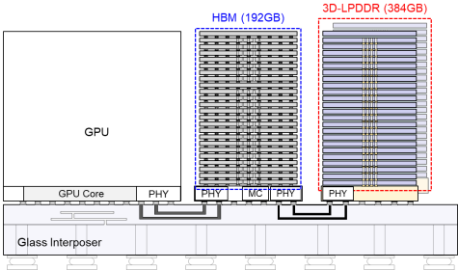
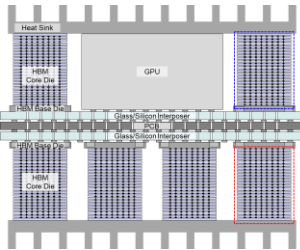
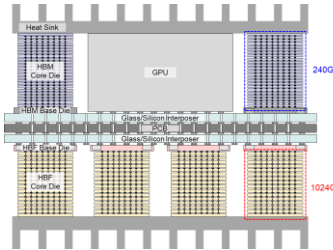
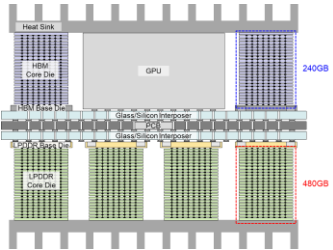
# 3D Memory Expansion Architecture for HBM8 with Double-Sided Interposer [3/3]: GPU-HBM-LPDDR



< 3D Memory Expansion Architecture for HBM8 with Double-Sided Interposer using LPDDR >

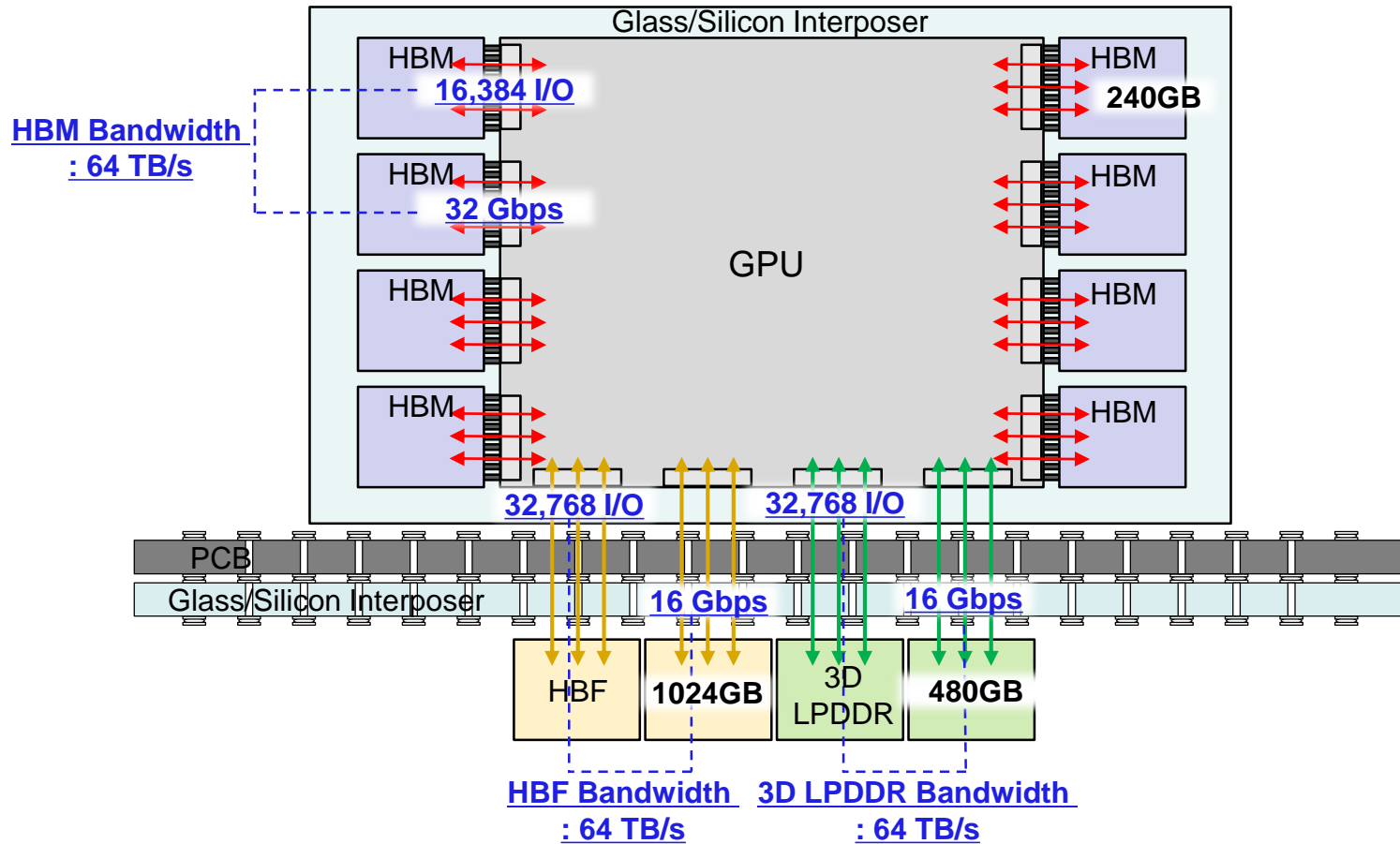
- Integrating 3D LPDDR enables cost-effective capacity scaling, offering a balance between bandwidth and power efficiency.

# Comparison of Key Features and Memory Capacities in HBM7 & HBM8 Architectures

	HBM7	HBM8		
	HBM7 with 3D-Stacked LPDDR	3D Memory Expansion with HBM	3D Memory Expansion with HBF	3D Memory Expansion with LPDDR
Architecture (Side View)				
Key Features	<ul style="list-style-type: none"> <li>2D Horizontal Memory Expansion</li> <li>Wider Interposer Footprint</li> <li>Energy Efficient</li> </ul>	<ul style="list-style-type: none"> <li>3D Vertical Expansion under Interposer</li> <li>Compact Package Size</li> </ul>		
		<ul style="list-style-type: none"> <li>GPU-HBM PHY Compatibility</li> </ul>	<ul style="list-style-type: none"> <li>Low Active Power</li> <li>Non-volatile</li> </ul>	<ul style="list-style-type: none"> <li>Energy Efficient</li> </ul>
Capacity per GPU	4608GB	5760GB	18304GB	9600GB

< Comparison of Key Features and Memory Capacities in HBM7 & HBM8 Architectures >

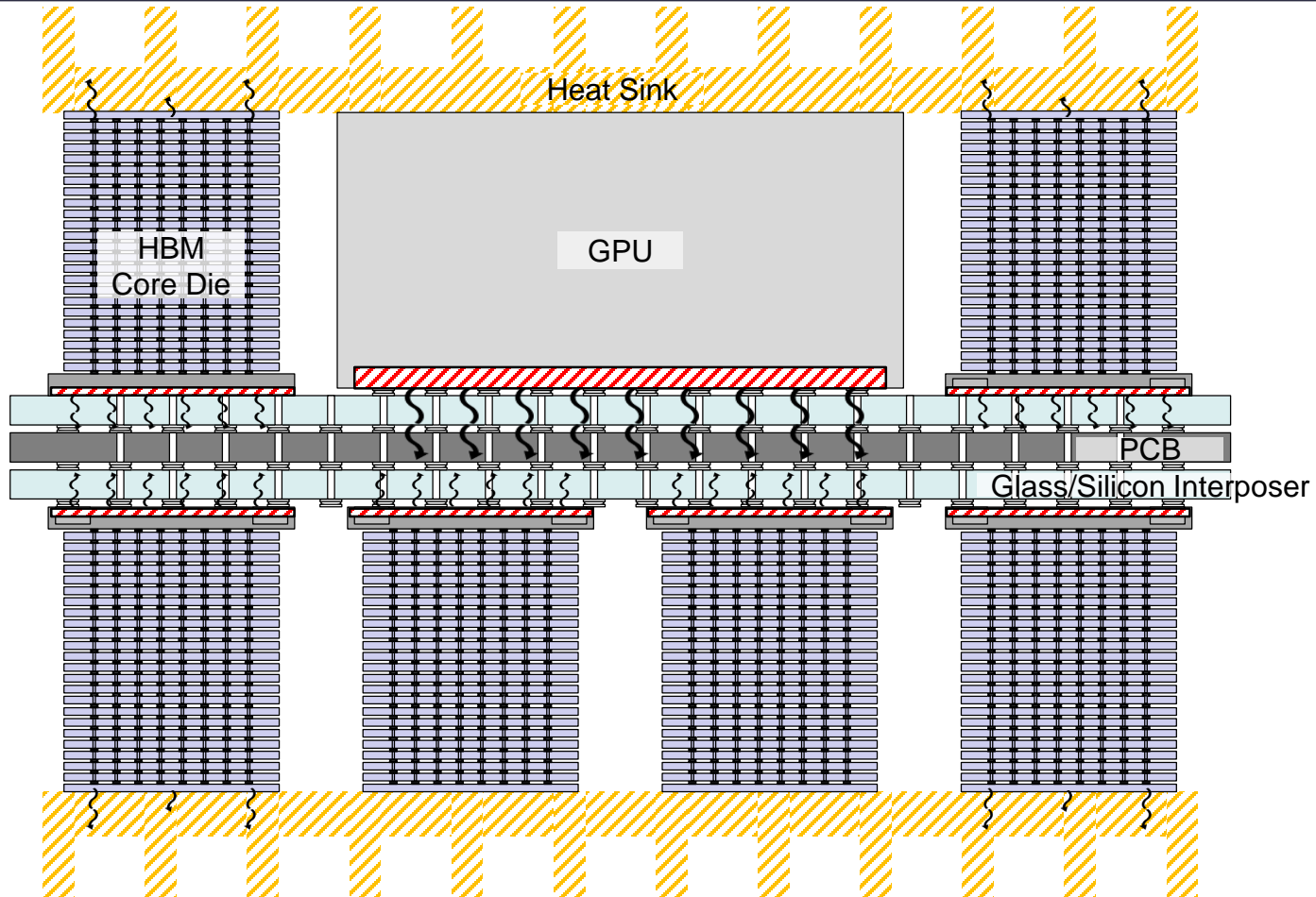
# Bandwidth-Matched 3D Memory Architecture with HBM, HBF, and LPDDR



## < Consistent Bandwidth Across Diverse Memory Types via I/O–Datarate Trade-Off >

- Flash-based memories like HBF and low-power DRAMs like LPDDR typically operate at lower data rates.
- However, by increasing the number of I/Os through TSV-based stacking, the system can match the **bandwidth to 64 TB/s for all memory types**.

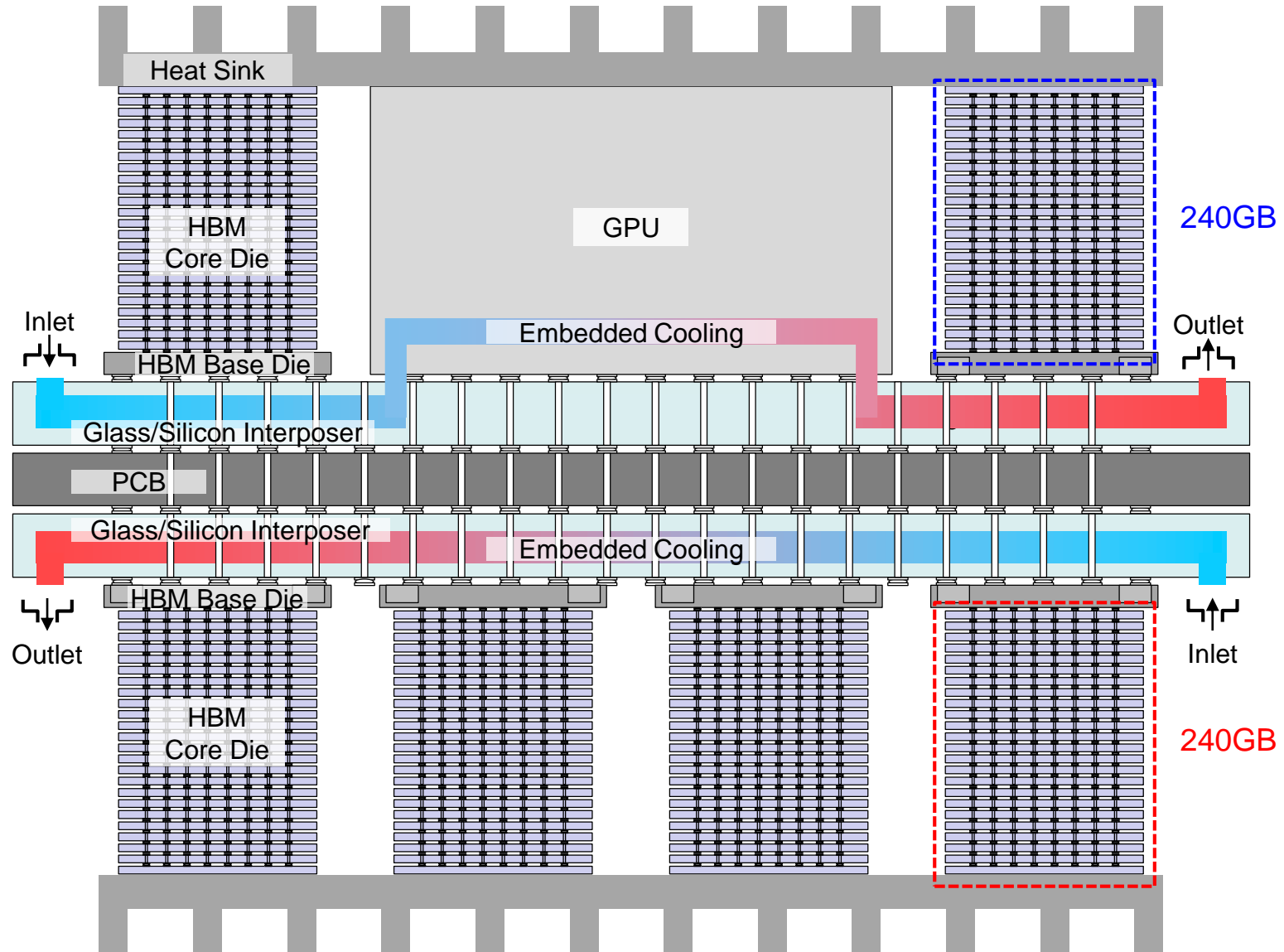
# Heat Accumulation Issues in 3D Memory Expansion Architecture



## < Thermal Congestion Across 3D Stacked Architectures >

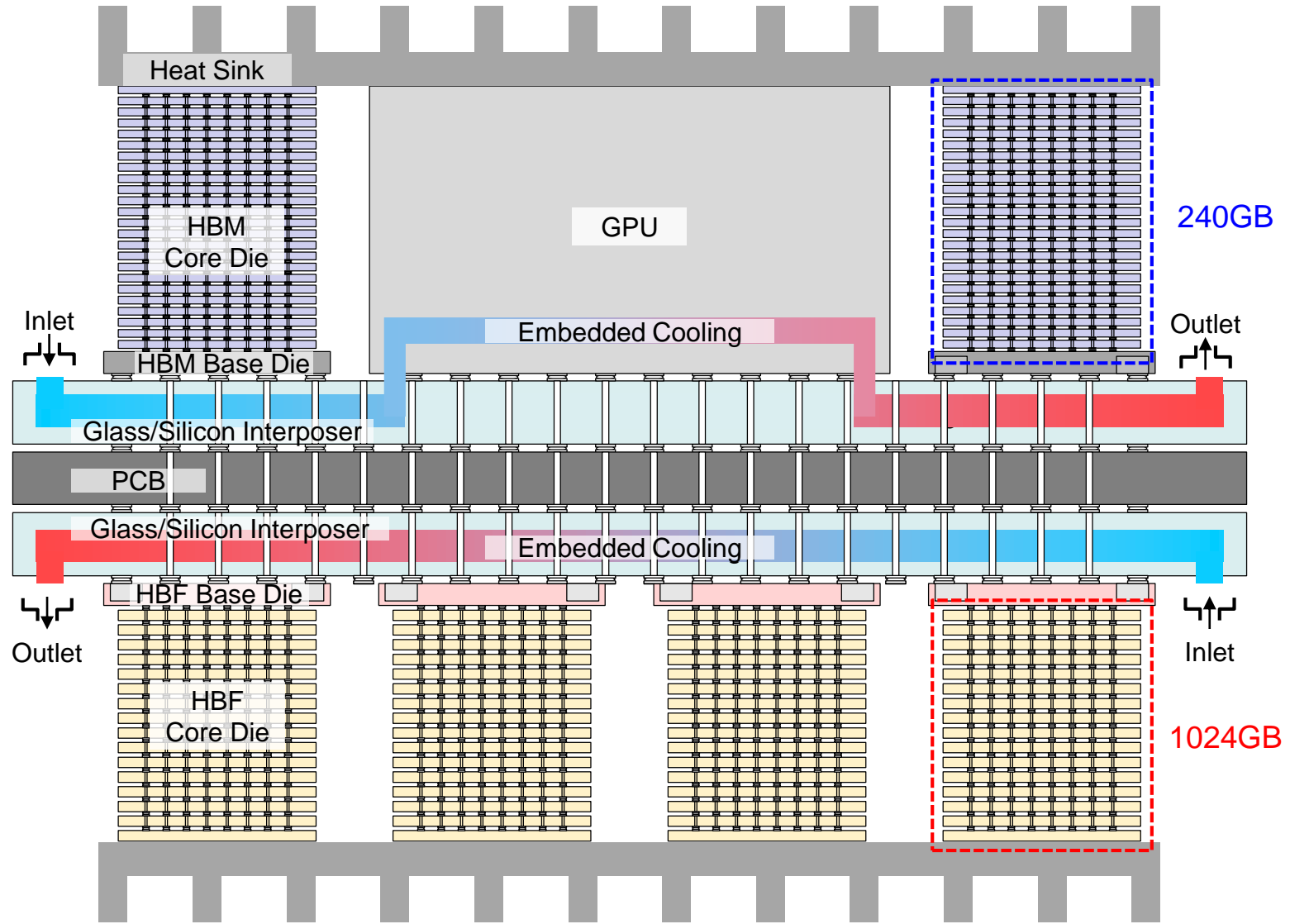
- Lack of bottom-side airflow and direct cooling interface
  - GPU die acts as a concentrated heat source, generating significant thermal load
- Thermal-aware structural design or embedded cooling becomes essential.

# 3D Memory Expansion Architecture with Embedded Cooling Structure for HBM8 with Double-Sided Interposer [1/3]: GPU-HBM-HBM



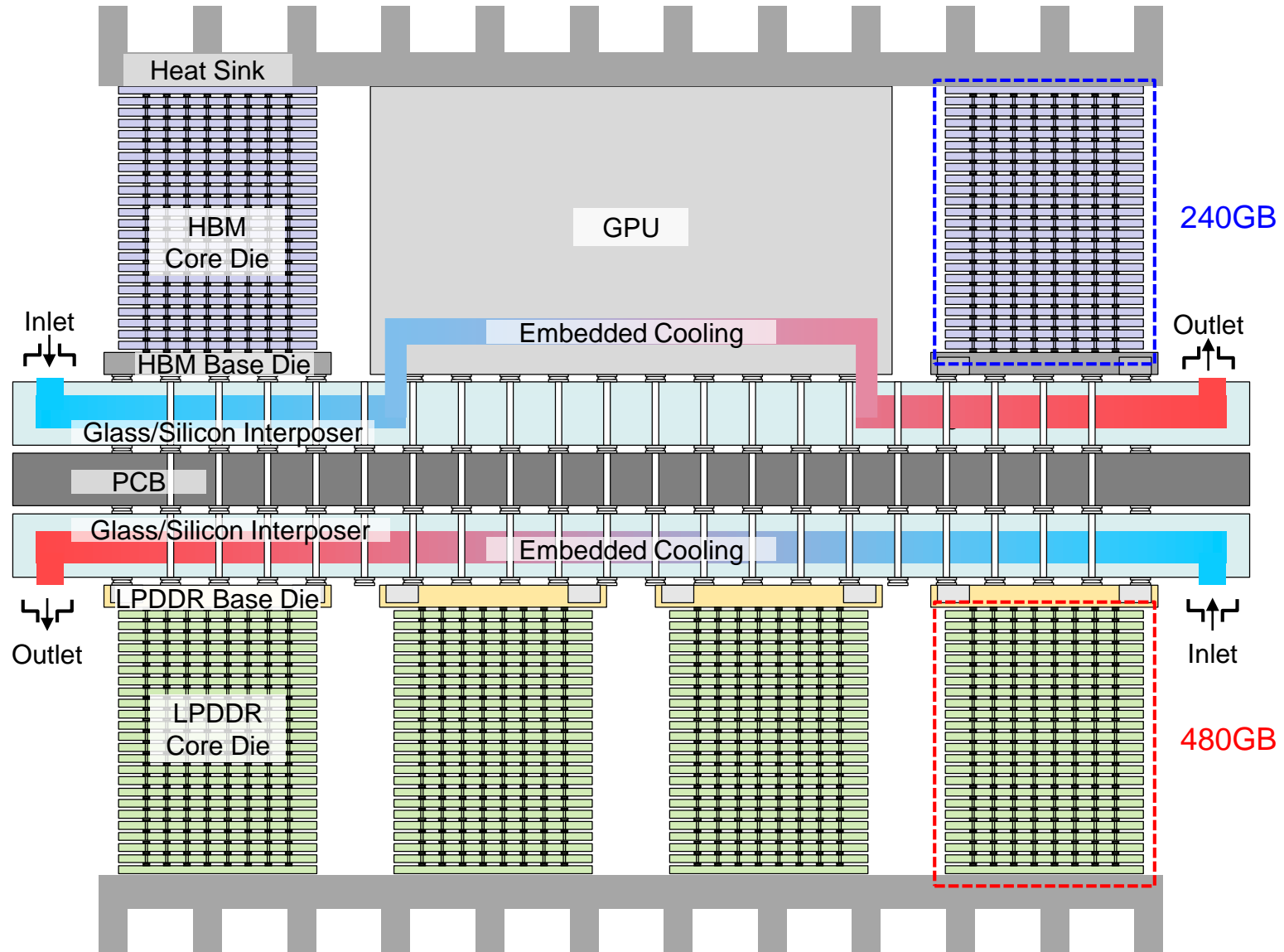
< 3D Memory Expansion Architecture with Embedded Cooling for HBM8 with Double-Sided Interposer using HBM >

# 3D Memory Expansion Architecture with Embedded Cooling Structure for HBM8 with Double-Sided Interposer [2/3]: GPU-HBM-HBF



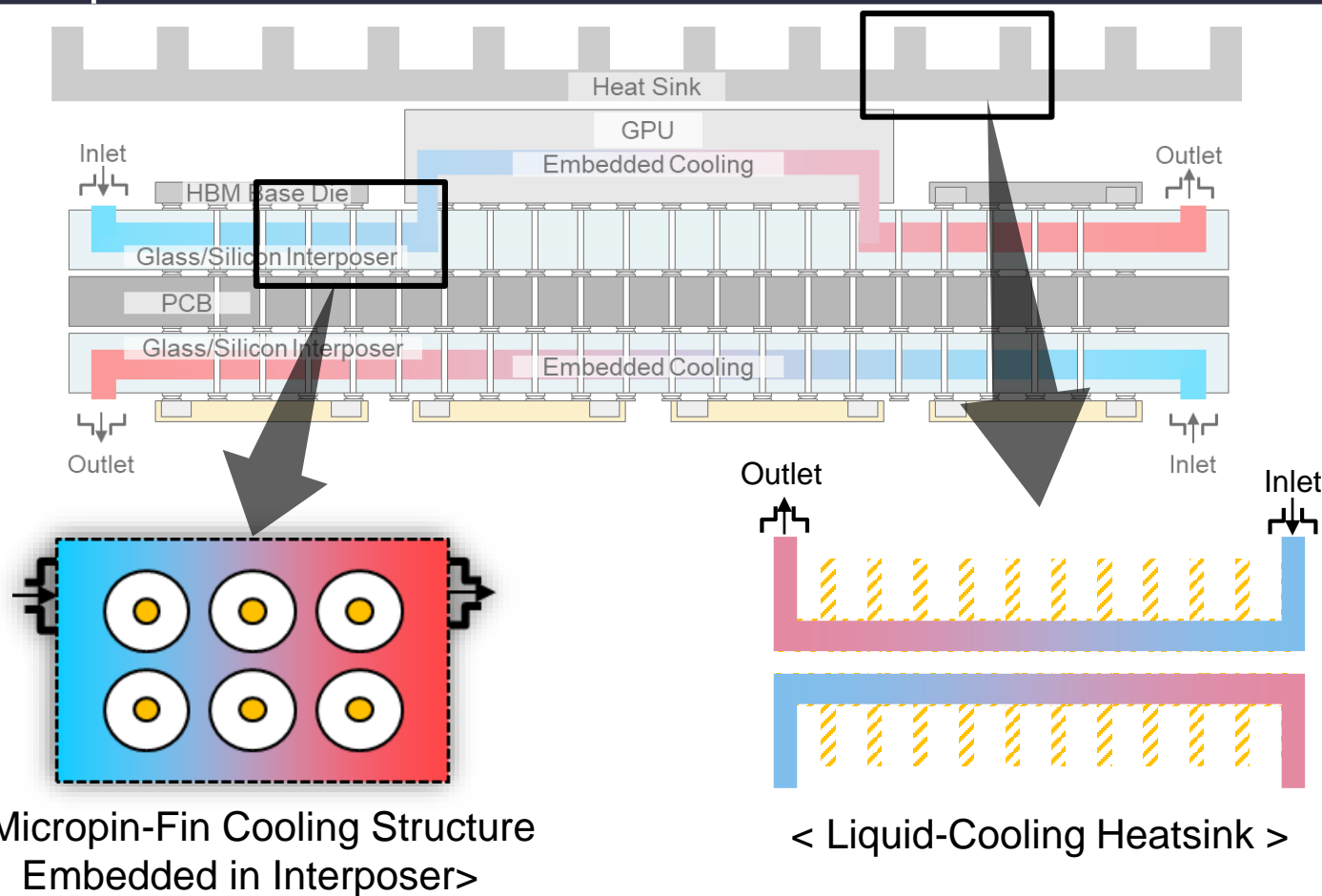
< 3D Memory Expansion Architecture with Embedded Cooling for HBM8 with Double-Sided Interposer using HBF >

# 3D Memory Expansion Architecture with Embedded Cooling Structure for HBM8 with Double-Sided Interposer [3/3]: GPU-HBM-LPDDR



< 3D Memory Expansion Architecture with Embedded Cooling for HBM8 with Double-Sided Interposer using LPDDR >

# Embedded Cooling Strategies for 3D Memory Integration with Double-Sided Interposer



- Micropin-fin structure is integrated around TSVs, enabling localized liquid flow and vertical heat extraction through the interposer.
  - Liquid-cooled heatsink is attached directly above the heat-intensive compute die.
- These two cooling layers are independently optimized but thermally coupled, enabling reliable heat dissipation from both the GPU and stacked memory dies.



- In the conventional 2.5D GPU-HBM systems, horizontal routing congestion and limited stack configuration prevent further bandwidth and capacity scaling.
- Therefore, we propose a vertically integrated 3D memory expansion architecture using a double-sided interposer.
- Our architecture enables modular stacking of HBM, HBF, and LPDDR with bandwidth-matching flexibility, and incorporates embedded cooling to ensure thermal reliability in dense 3D stacks.
- Proposed HBM8 will support both high bandwidth and large memory capacity, making it well-suited for future LLM workloads requiring scalable GPU-memory systems.



# HBM CENTRIC

# Thank you!

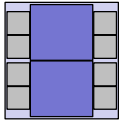
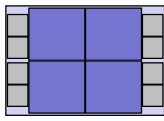
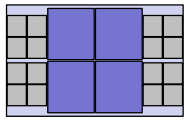
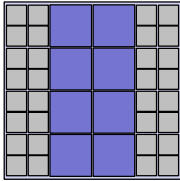
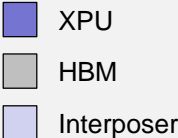
# AI Design Agent for 3D Placement and Routing Optimization for HBM8 using Reinforcement Learning considering Thermal-Signal Integrity

Hyunseo Uhm

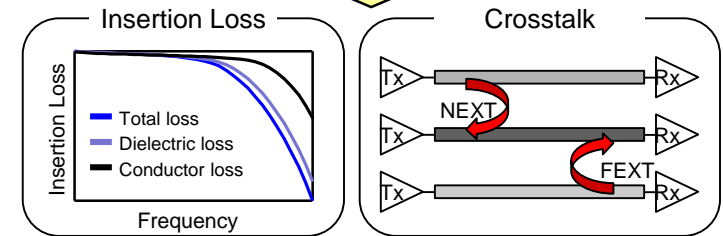
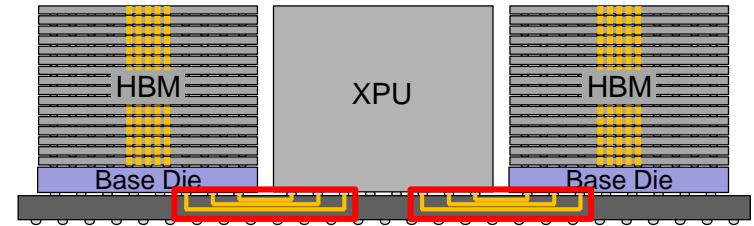
Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

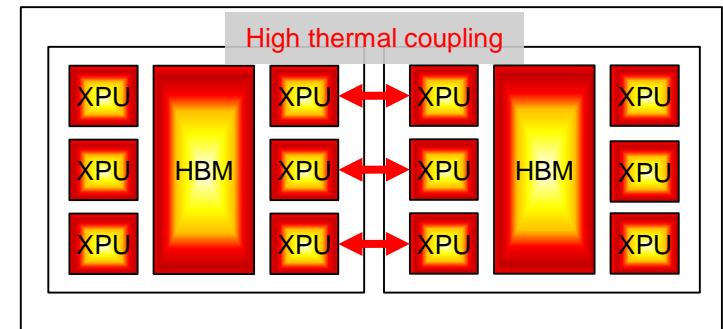
# Signal/Thermal Integrity Problems on Multiple HBM8-XPU Structure

GPU Architecture	Rubin (2026)	Feynman (2028)	Post Feynman (X*) (2030)	Next-Gen GPU (Z*) (2032)
GPU-HBM System	R200	F400	X400*	Z800*
Die Shot				
				
# of GPU Dies	×2	×4	×4	×8
# of HBM Stack	HBM4×8	HBM5×8	HBM6×16	HBM7×32
Total Bandwidth	16 / 32 TB/s	48 TB/s	128/256 TB/s	1,024 TB/s
Total HBM Capacity	288/384 GB	400/500 GB	1,536/1,920 GB	5,120/6,144 GB

< Requirement of Multiple HBMs >



< Signal Integrity Problems on Multiple HBM-XPU Structure >

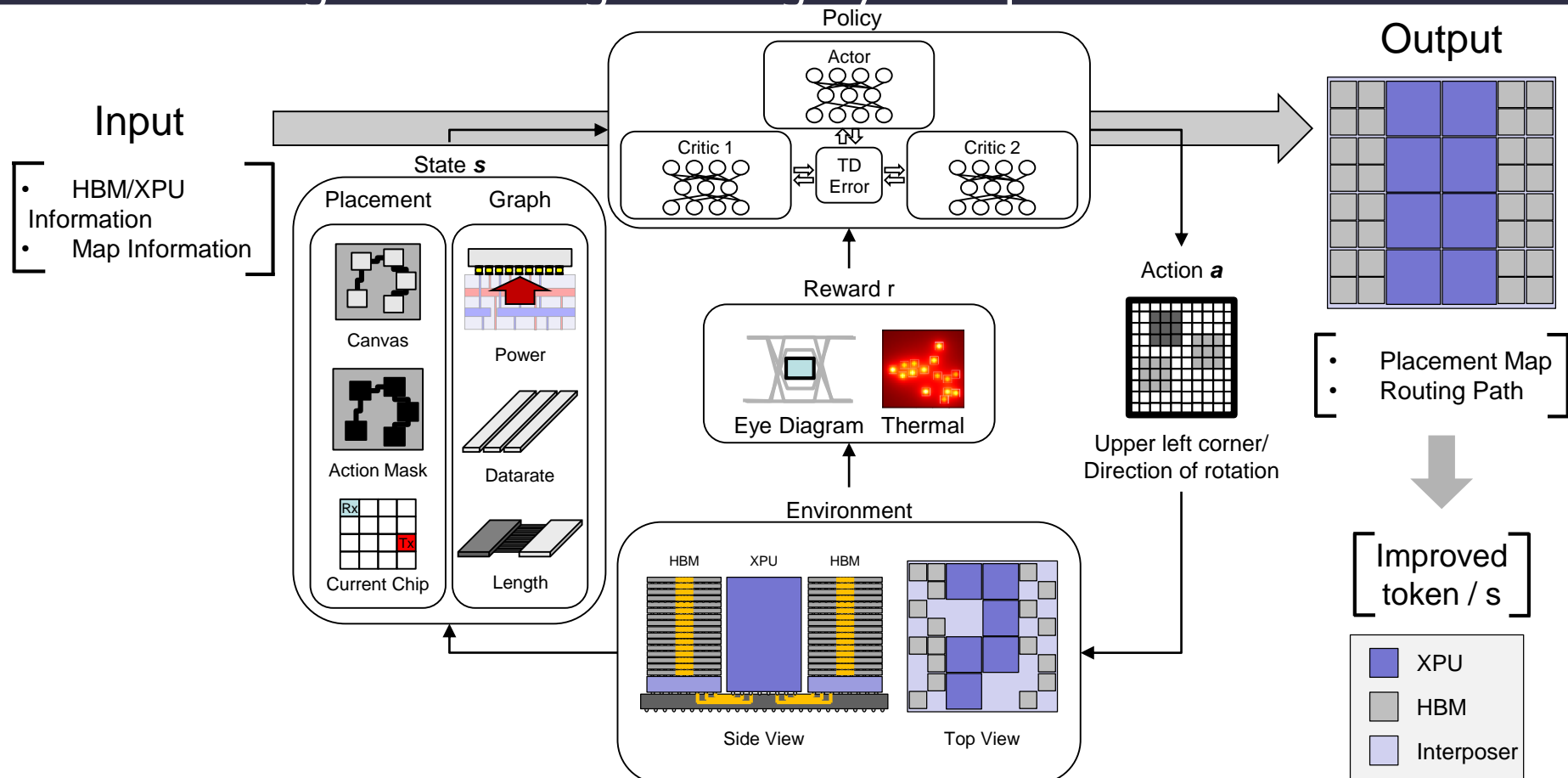


$$(Thermal\ coupling) \propto 1/(Distance)$$

< Thermal Coupling between HBM-XPU Modules >

- On HBM8, full-3D HBM-XPU architecture with stacked processor is required for higher bandwidth and computational power.
- Degradation of signal integrity and thermal coupling problem happens on multiple HBM-XPU architecture.

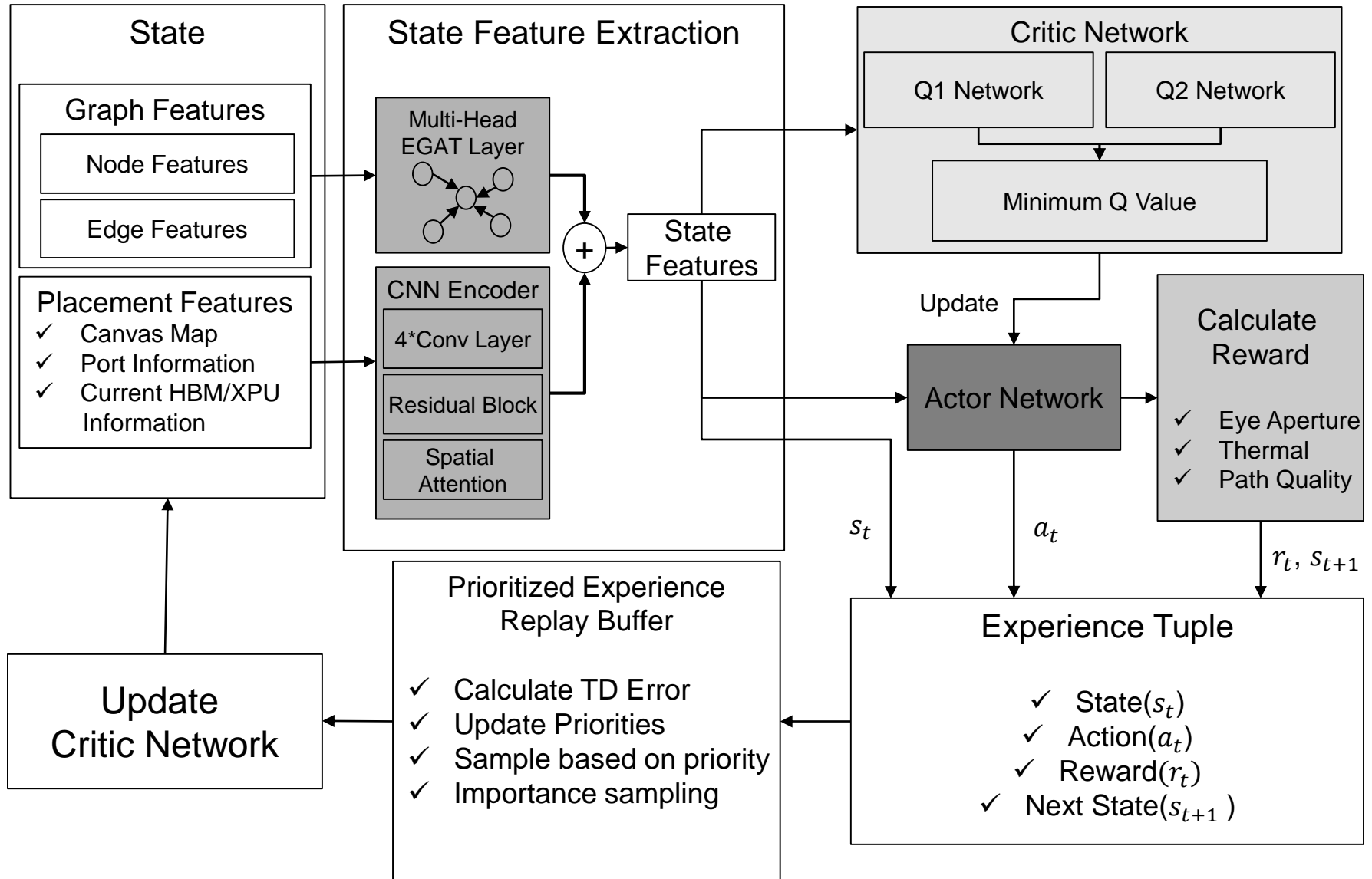
# Proposal of Multiple HBM-XPU Placement and Routing Agent Considering Thermal-Signal Integrity Co-Optimization



< Overview of the proposed Artificial Intelligence Multiple HBM-XPU Placement & Routing Design Agent Considering Thermal-Signal Integrity Co-Optimization >

- The input is a placement map and information of Multiple HBM/XPUs , and the output is the revised placement map and routing path.
- The goal is to obtain the optimal design parameters through an Soft-Actor Critic-based optimization methodology.

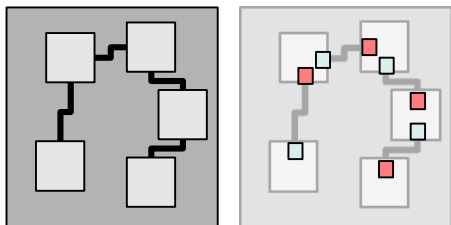
# Detailed Structure of Soft-Actor Critic Based Multiple HBM-XPU Floorplanning Algorithm



< Overview of Soft-Actor Critic Based Multiple HBM-XPU Floor-planning Algorithm >

# Markov Decision Process [1/3]: State

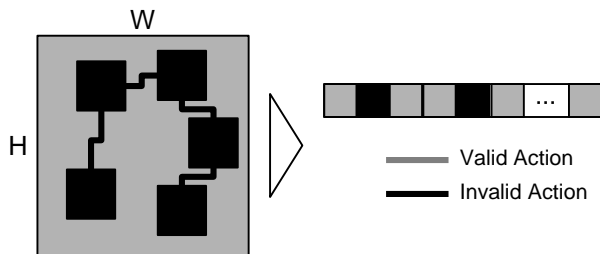
## Placement Features



### Canvas

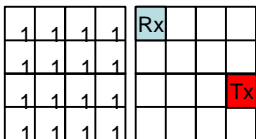
0<sup>th</sup> Floor: Placement Map

1<sup>st</sup> Floor: Port map



### Action Mask

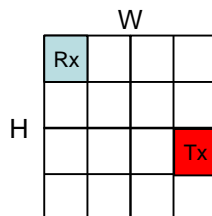
(1-dimensional array  
of size  $W \times H \times 4$  (Rotation Degrees))



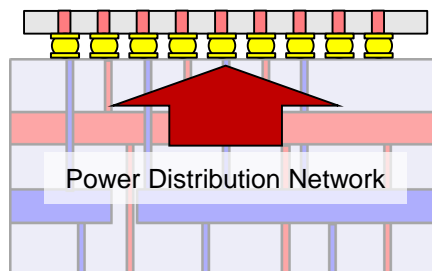
### Current Block Information

## Graph Features

### Node Features

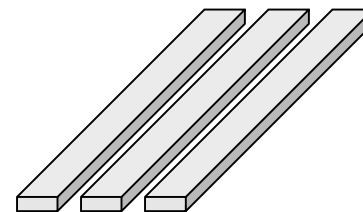


Width(W) / Height(H)

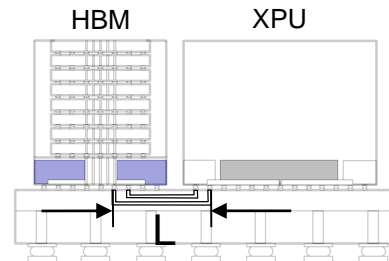


Assigned Power

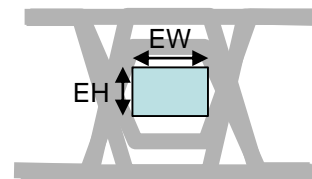
### Edge Features



Datarate



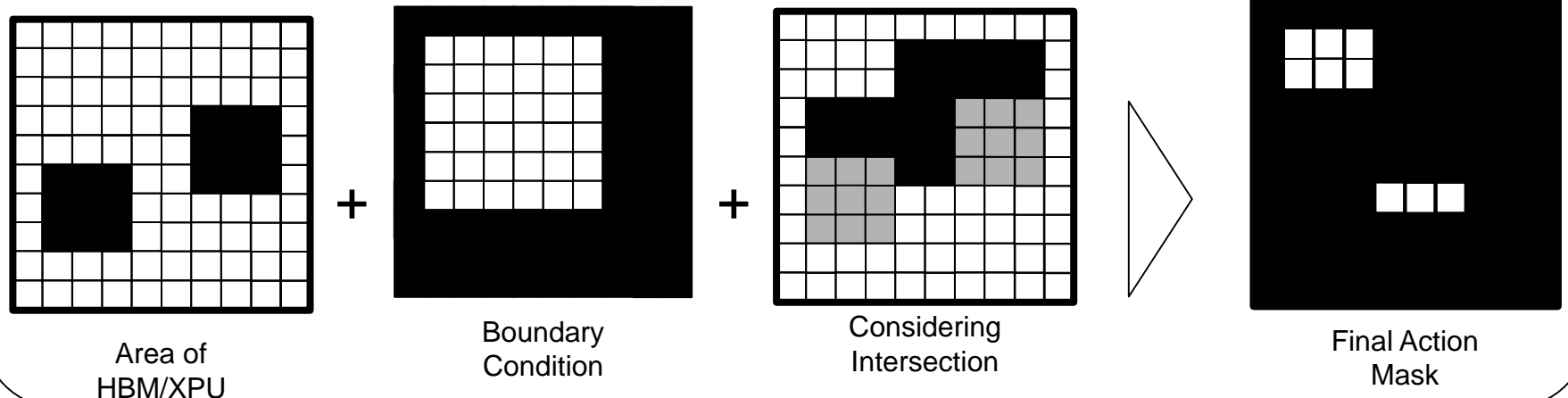
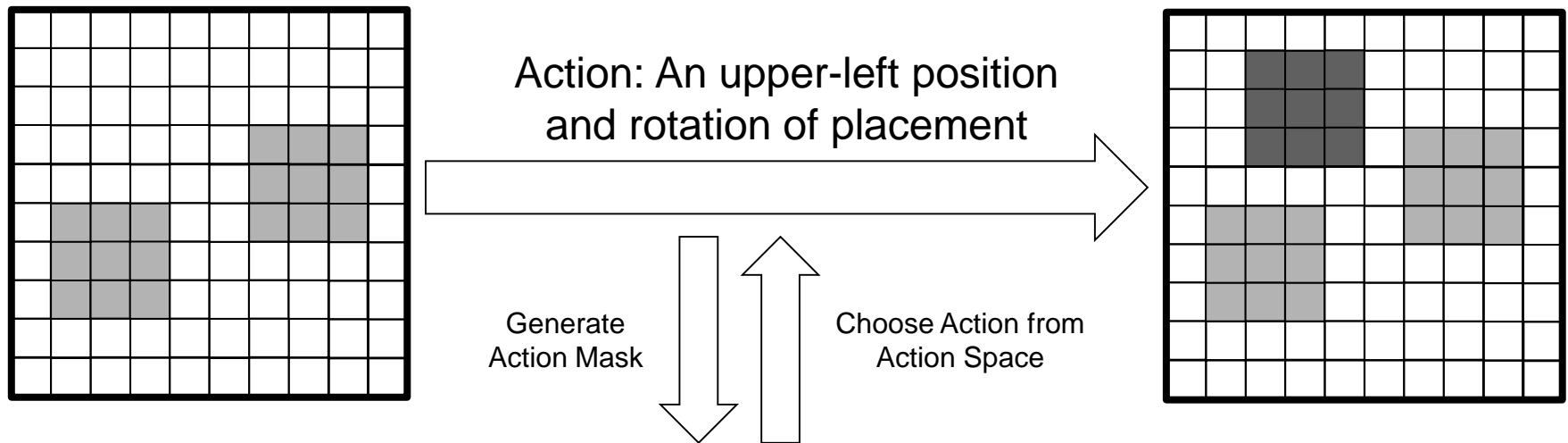
Length(L)



Eye Aperture( $EH \times EW$ )

< Information of Input State >

# Markov Decision Process [2/3]: Action

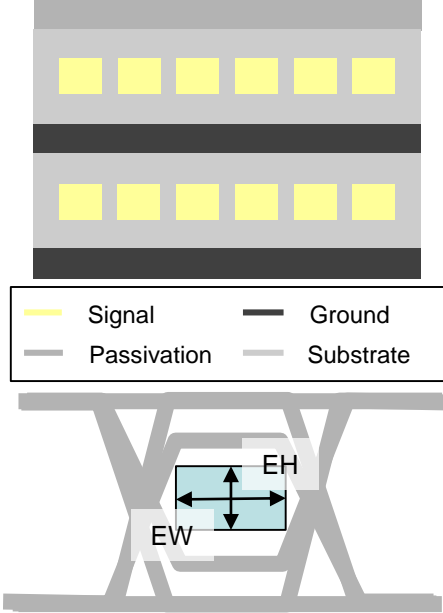
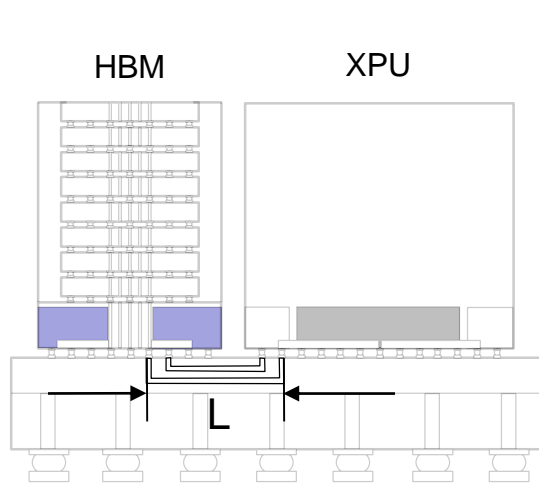
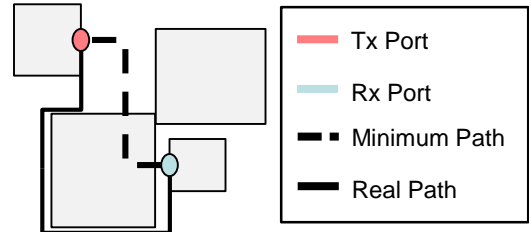


## < Placement Action and Action Mask Generation >

- The placement action is an upper-left position and rotation of placement.
- Action mask is generated considering HBM/XPU area, boundary condition, and intersection area.



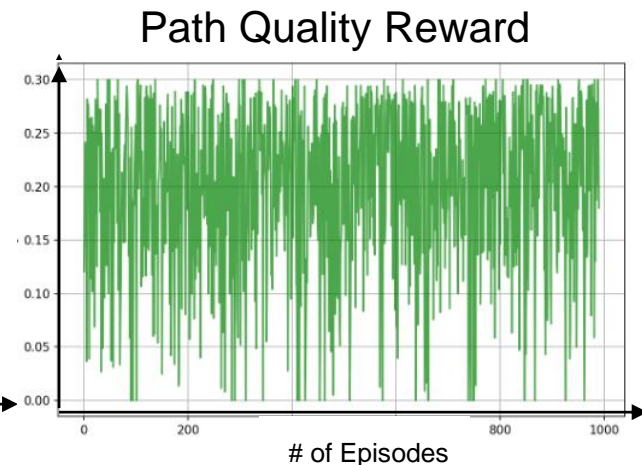
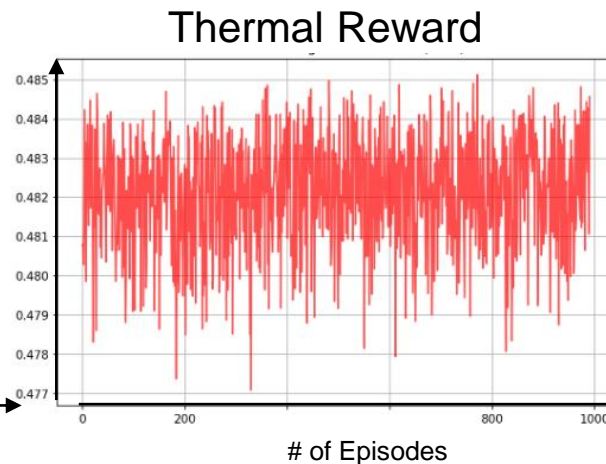
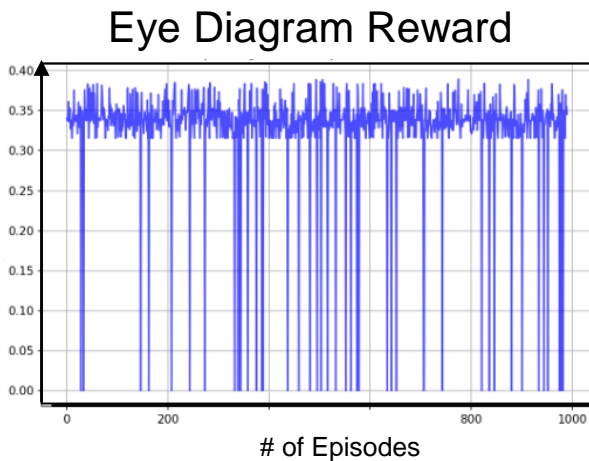
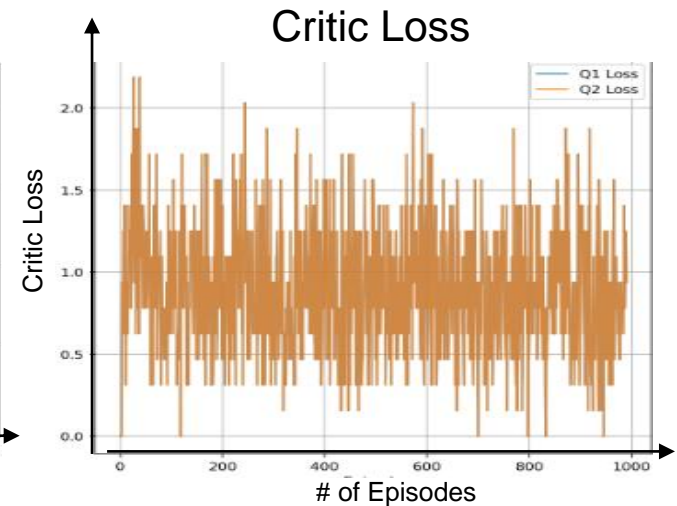
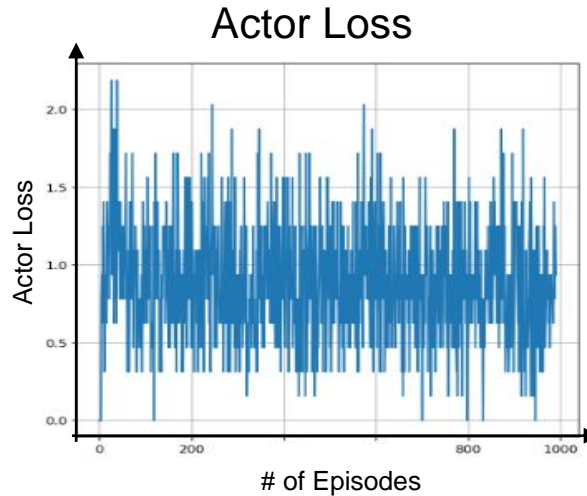
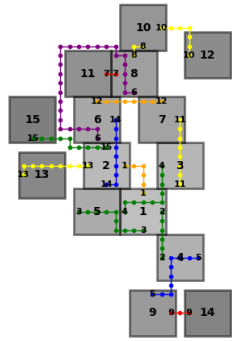
# Markov Decision Process [3/3]: Reward

Eye Diagram Reward	Thermal Reward	Path Quality
 <p> <span style="color: yellow;">■</span> Signal    <span style="color: black;">■</span> Ground  <span style="color: gray;">■</span> Passivation    <span style="color: lightgray;">■</span> Substrate         </p> <p> <math>EH</math>  <math>EW</math> </p> <p> <i>Eye Diagram Reward</i>  <math>= (EH \cdot EW)</math> </p>	 <p> <math>(Thermal\ coupling) \propto 1/(Distance)</math>  <i>Thermal Reward</i>  <math>= f(Max\ temperature\ of\ HBM-XPU\ Structure)</math> </p>	 <p> <span style="color: red;">—</span> Tx Port  <span style="color: cyan;">—</span> Rx Port  <span style="color: black;">- -</span> Minimum Path  <span style="color: black;">—</span> Real Path         </p> <p> <i>Path Quality</i>  <math>= f(\frac{Total\ Actual\ Length}{Total\ Minimum\ Length})</math> </p>

< Reward: SI Performance, TI Performance, Path Quality >

- Eye diagram reward is used as a signal integrity performance metric.
- Thermal reward is used as a function of maximum temperature of HBM-XPU structure, which is inversely proportional to the distance between HBM and XPU.
- Difference between theoretical minimum path and actual path is used as a path quality reward.

# Performance Verification of Proposed Method



## <Results and Metrics of Multiple HBM-XPU Placement and Routing Optimization>

- Reward is not increasing effectively, because of the wirelength constraints for calculating eye diagram reward.
- For both of the rewards, more sophisticated construction of reward is crucial.

# Conclusion

- On HBM8, full-3D HBM-XPU architecture with stacked processor is required for higher bandwidth and computational power.
- In order to archive signal integrity and thermal integrity, optimization of floor-planning and routing on HBM-XPU structure is crucial, and we proposed the Reinforcement Learning(RL) based floor-planning optimization method.
- Edge-Aware Graph Attention Network and Soft-Actor Critic is used in RL-Based Multiple HBM-XPU floor-planning agent.
- Eye diagram reward is used as a signal integrity performance metric.
- Thermal reward is used as a function of maximum temperature of HBM-XPU structure, which is inversely proportional to the distance between HBM and XPU.
- While extending to full-3D architecture, thermal issues can be a more critical issue, and signal characteristics vary significantly by layer, which can lead to the requirement of thermal-signal integrity co-optimization.

# Thank You!

## HBM

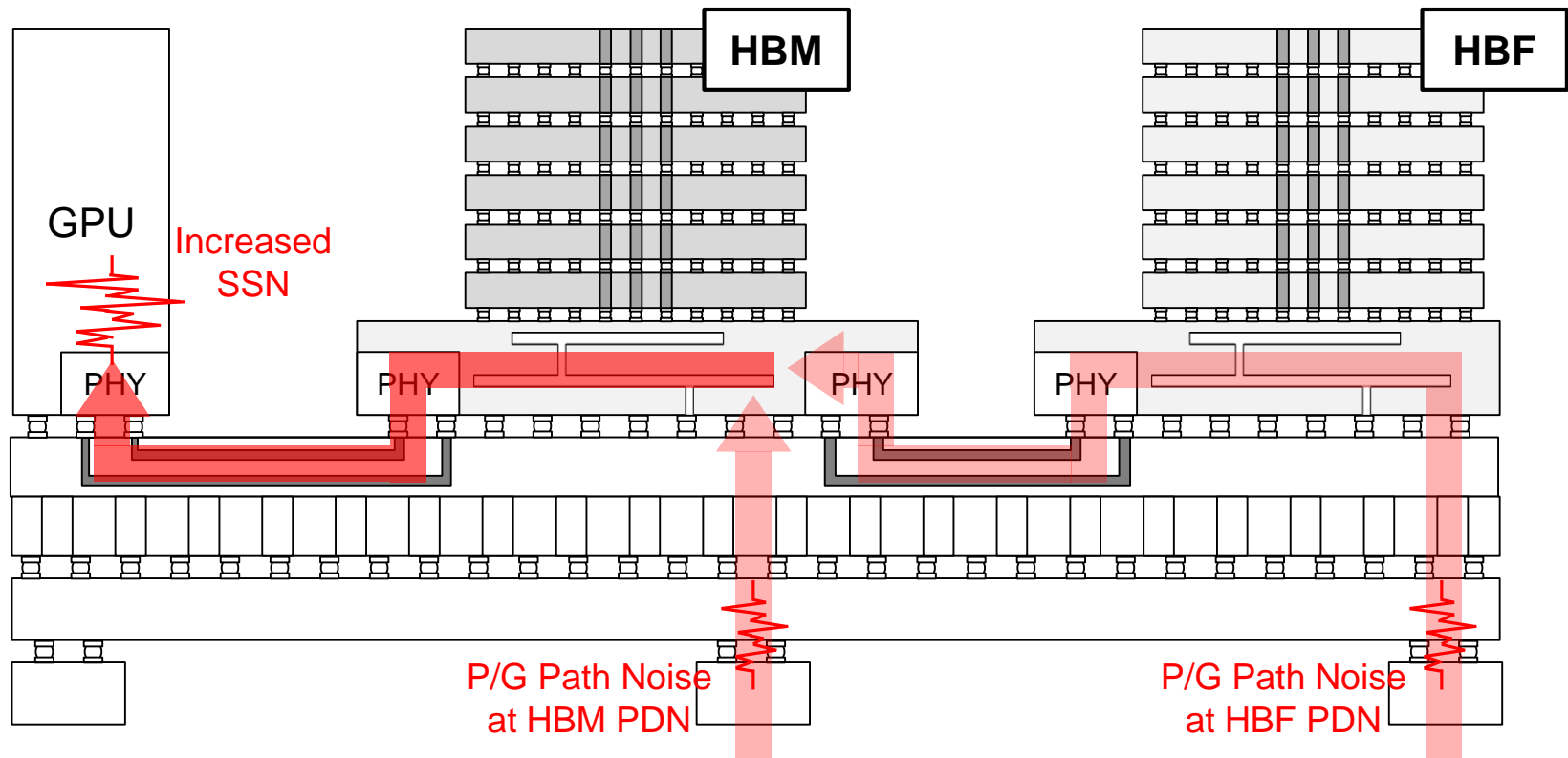
# LLM-aided Interactive Reinforcement Learning (IRL) with Mixture-of-Experts (MoE) Transformer for Power Supply Induced Jitter (PSIJ) Reduction in HBM7

Jaegeun Bae

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST

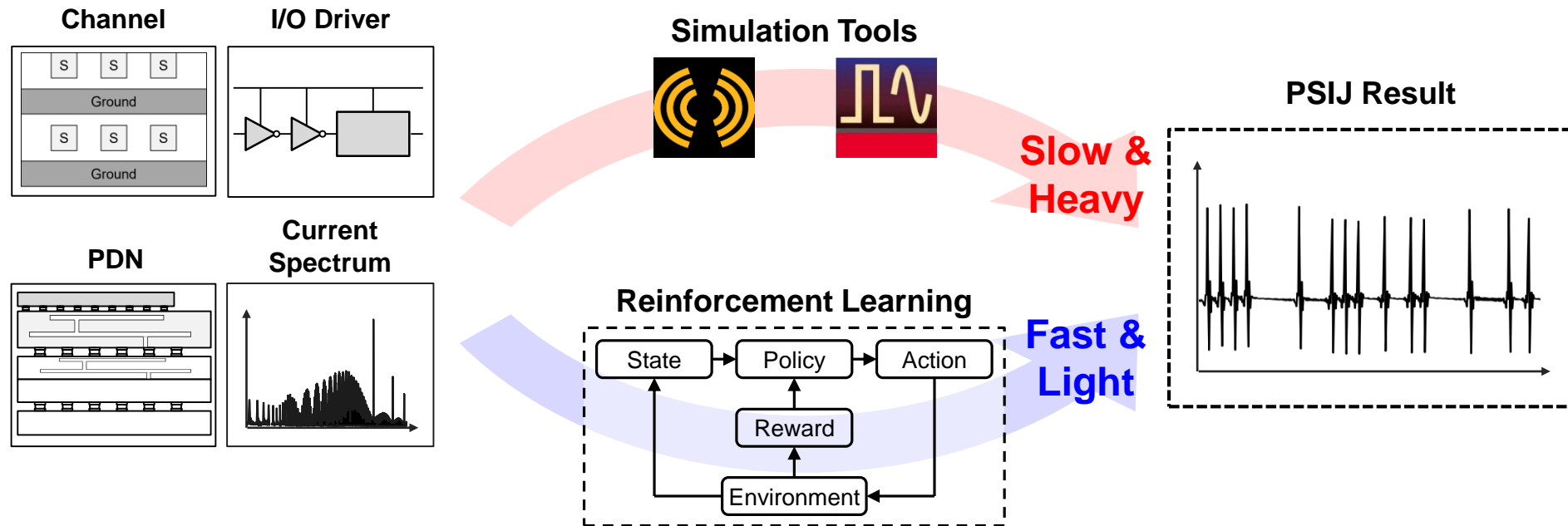
# HBM7: I/O Interface Optimization for PSIJ Reduction in HBM-HBF Structure



< SSN Increase at HBM-HBF Structure >

- HBM-HBF structure incorporates additional I/O ports, which results in increased simultaneous switching noise (SSN).
  - As SSN increases, power supply induced jitter (PSIJ) also increases, which significantly impairs signal integrity.
- The increased data rate of HBM7 (16 Gbps) requires a tighter jitter margin, which amplifies the impact of PSIJ.

# Reinforcement Learning based Method for PSIJ-aware Design Optimization

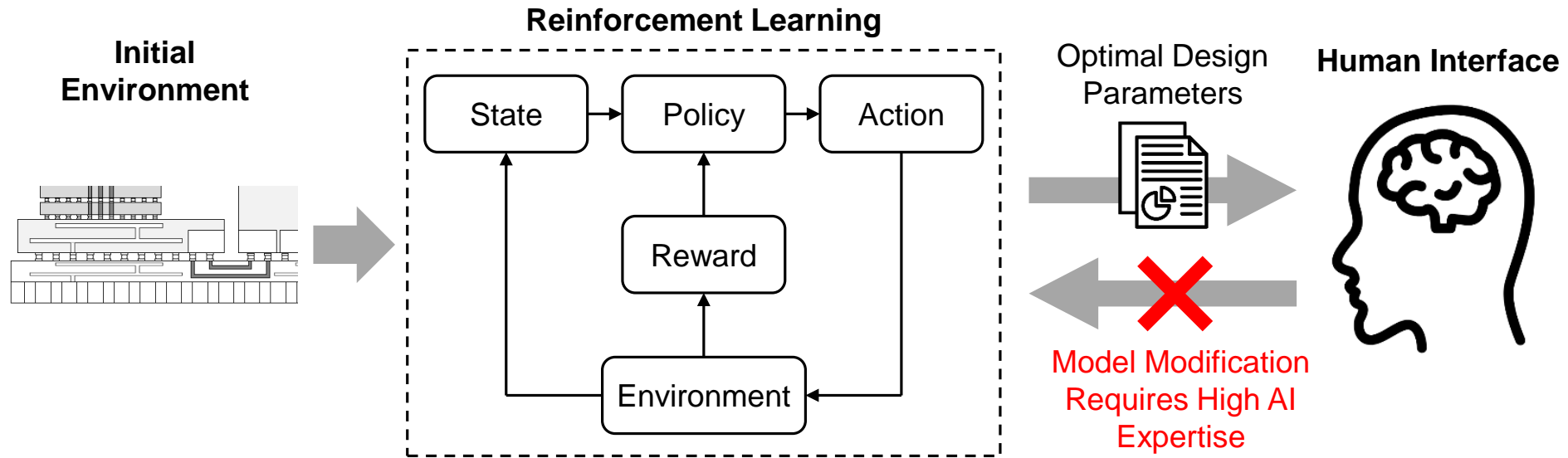


< PSIJ-aware Design Optimization Problem of HBM4 Base Die >

- Choosing Adequate design parameters (PDN, I/O Driver, Interposer Channel) plays an important role in improving PSIJ.
  - However, performing simulation for all the parameter requires **large computing cost**, as simulation goes through both HFSS and ADS model.
- By using **reinforcement learning**, optimal design parameters can be achieved without consuming too much time and computing power.



# Limitations of Hardware Design by Conventional Reinforcement Learning (RL)

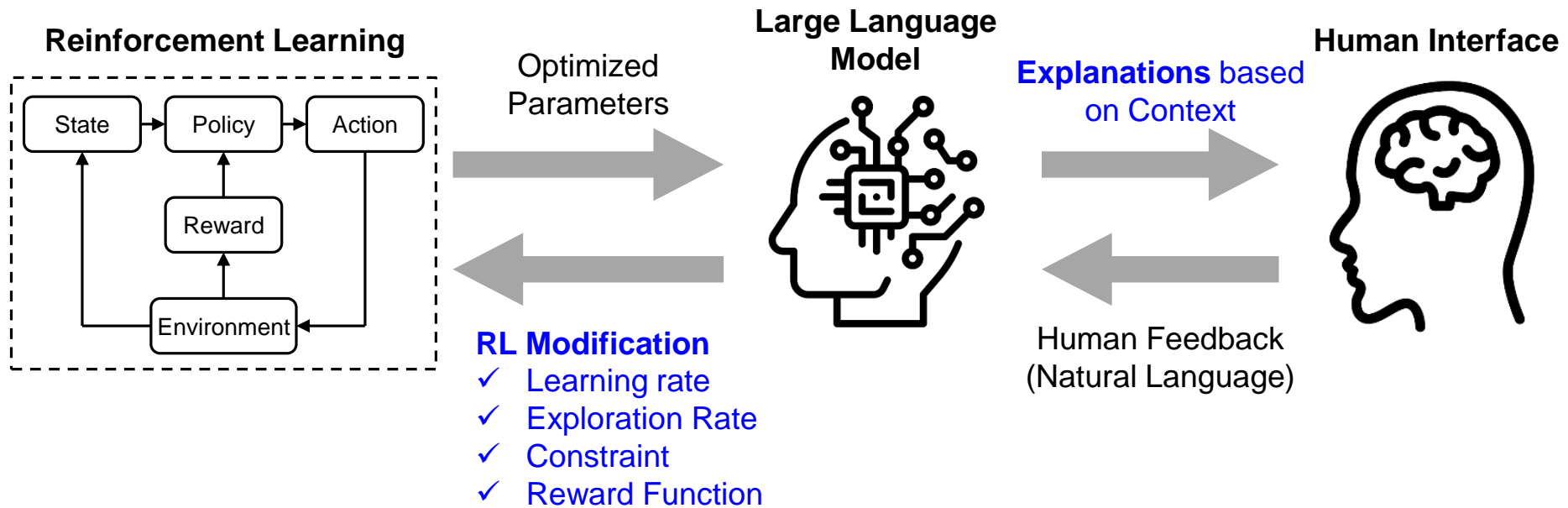


< Conventional Reinforcement Learning Architecture:  
Lacking Interaction Between Model and Human >

- Reinforcement Learning (RL) method has been in the spotlight for various hard-to-solve hardware design optimization problems.
  - However, lack of interpretability in RL result and inflexible RL reward causes severe inefficiency and waste of resource in RL debugging.
  - As hardware designs utilizing AI become increasingly complex and resource-intensive, **intimate human-RL interaction** has become necessary for efficient and convenient RL sessions.



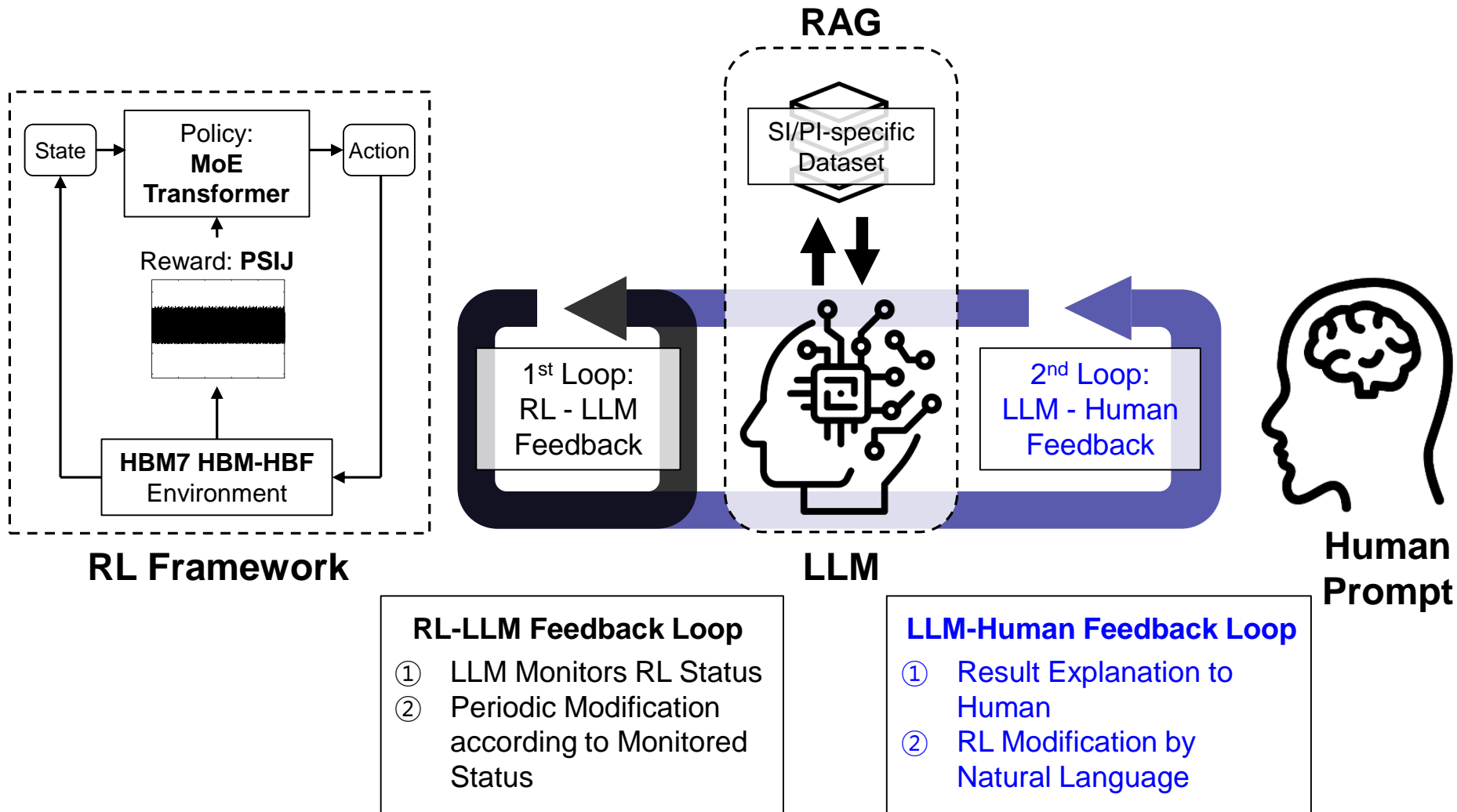
# Interactive Reinforcement Learning (IRL): Large Language Model (LLM)-aided RL



< Interactive Reinforcement Learning System Outline: Human and RL Linked by LLM >

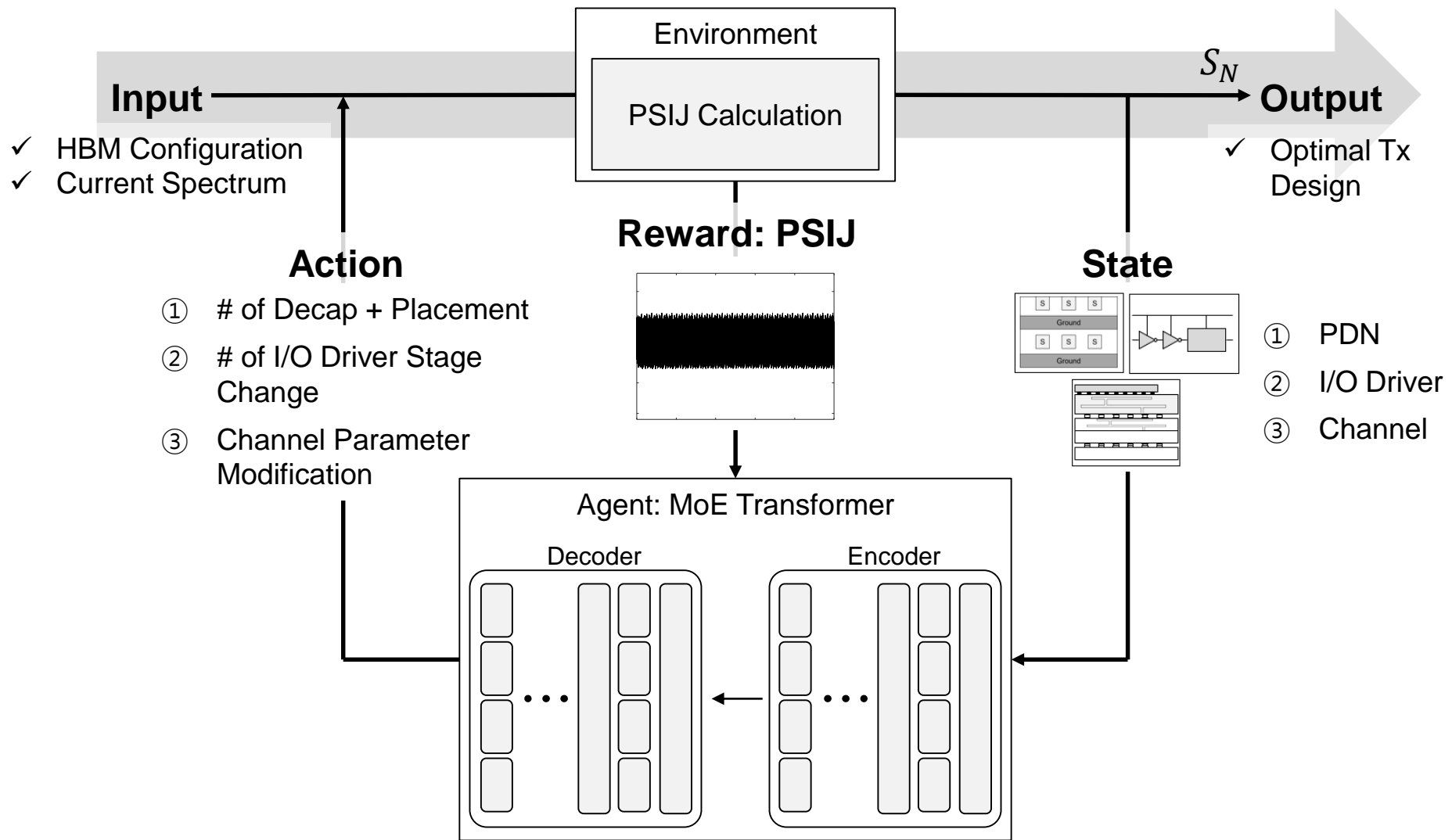
- Interactive Reinforcement Learning (IRL) introduces Large Language Model (LLM) to hardware design as bridge between human and RL.
  - **RL-Modification:** Human prompt is interpreted by LLM and is implemented to RL.
  - **Human-Friendly Interface:** RL results are analyzed and explained to human by LLM to human language interface.
- LLM can modify RL factors such as **learning rates**, **exploration rate**, and **action constraints**, and **reward function** itself.

# Proposal of LLM-aided IRL-based HBM Design Agent for PSIJ Reduction



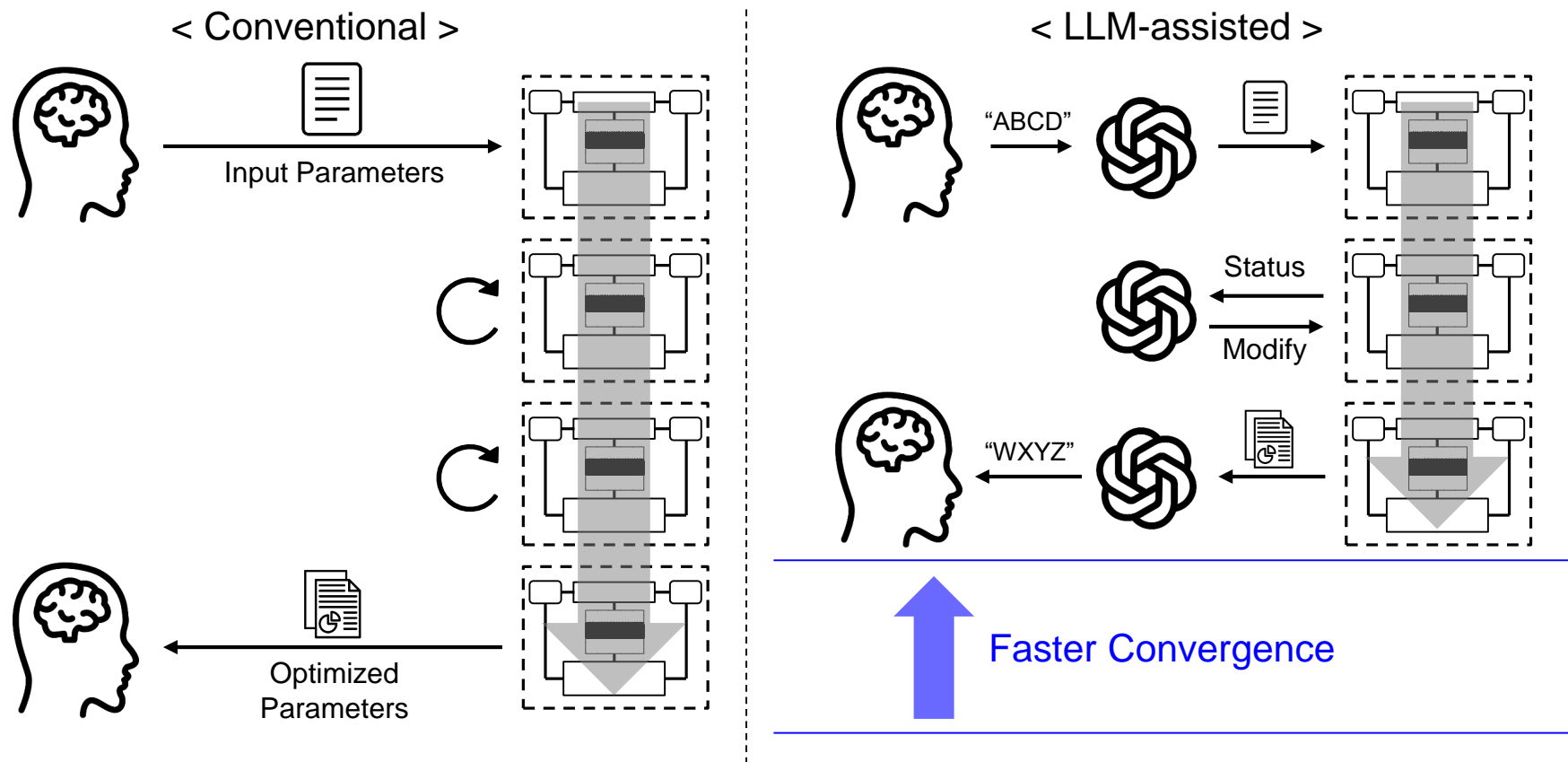
< LLM-aided IRL-based HBM Design Agent for PSIJ Reduction >

# Markov Decision Process (MDP) Formulation



< LLM-aided IRL-based HBM Design Agent with MoE Transformer for PSIJ Reduction >

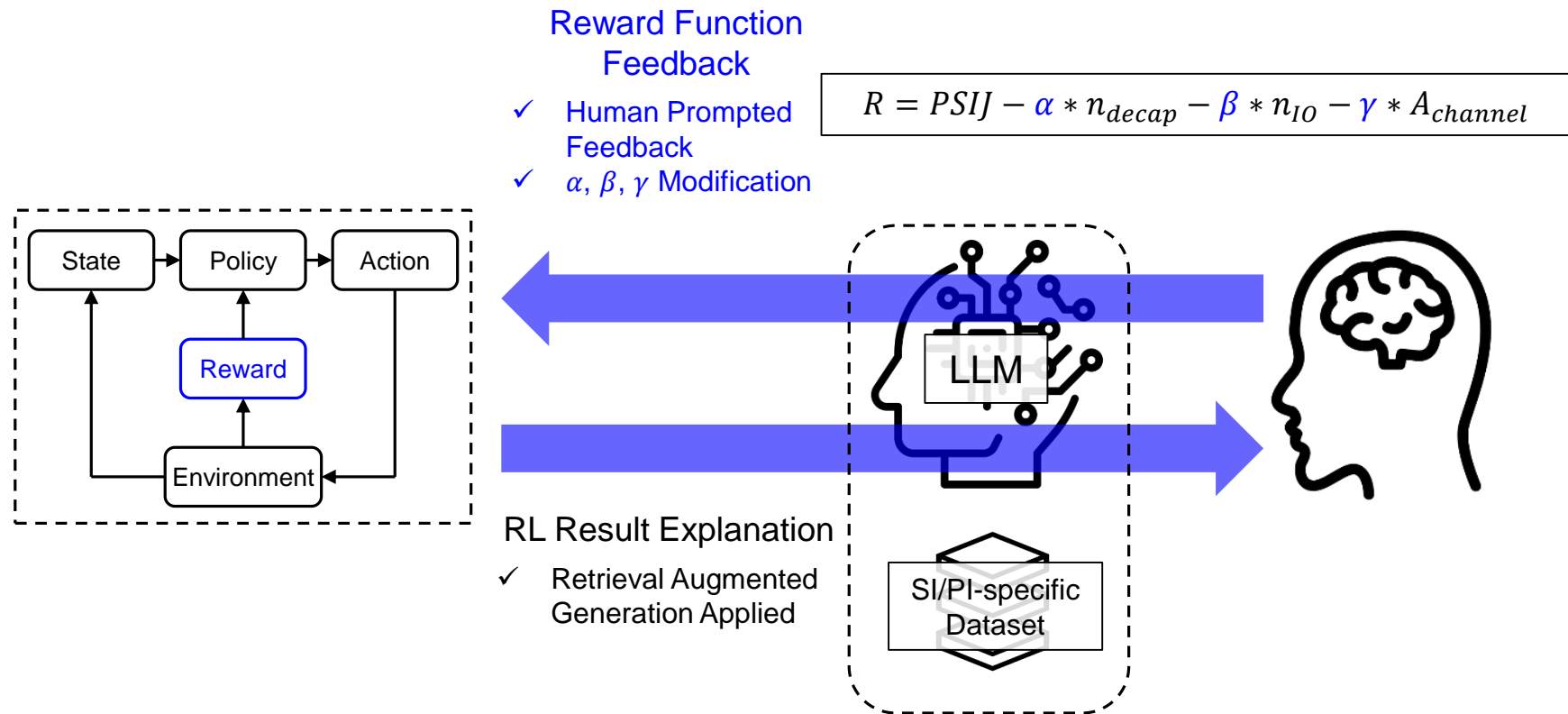
# LLM-assisted RL Optimization: Faster RL Convergence



< Comparison between Conventional RL and LLM-assisted RL >

- Unlike traditional RL architectures, LLM acts as an intermediary between humans and the RL agent, enabling real-time monitoring and improvement of the model.
  - LLM periodically adjusts the hyperparameters and reward function, enabling RL agent to converge more quickly while avoiding local minima.

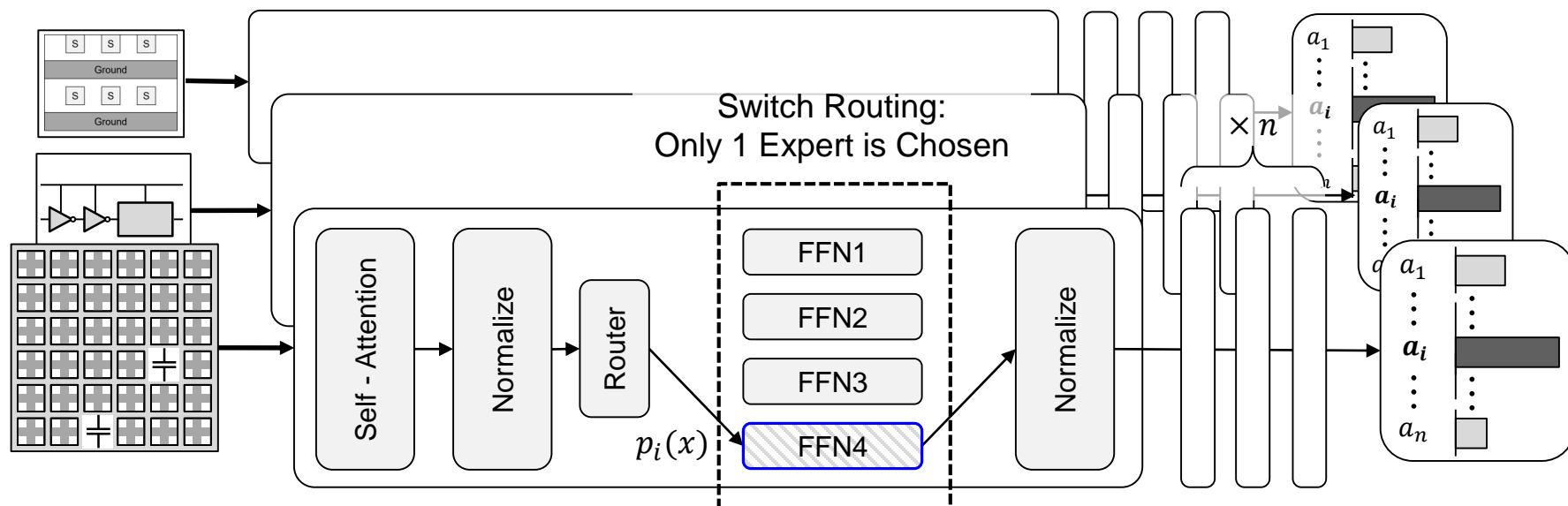
# LLM-based Reward Function Generation



## < LLM-Human Feedback: Reward Function Shaping >

- The reward function in RL can be easily modified through the natural language processing capabilities of the LLM.
  - Specifically, adjusting the reward function's coefficients allows influence on the priority of factors in RL agent.

# MoE Transformer Agent: High Sparsity, Fast Inference



\* Routing Probability:  $p_i(x) = \frac{e^{W_r * x_i}}{\sum_j e^{W_r * x_j}}$

Less Calculation, Fast Inference

< Mixture of Experts (MoE) Transformer-based Policy Network >

- Mixture of Experts (MoE) Transformer model is utilized for the policy network.
  - 1 expert is chosen via routing layer: routing probability is calculated by softmaxing weighted attention values.
- MoE Transformers improve computational sparsity by activating only a single expert, which leads to more efficient and faster inference.

# Thank You!

## HBM

# LLM-based HBM7 Design Agent using Interactive Reinforcement Learning (IRL) for Decoupling Capacitor Placement

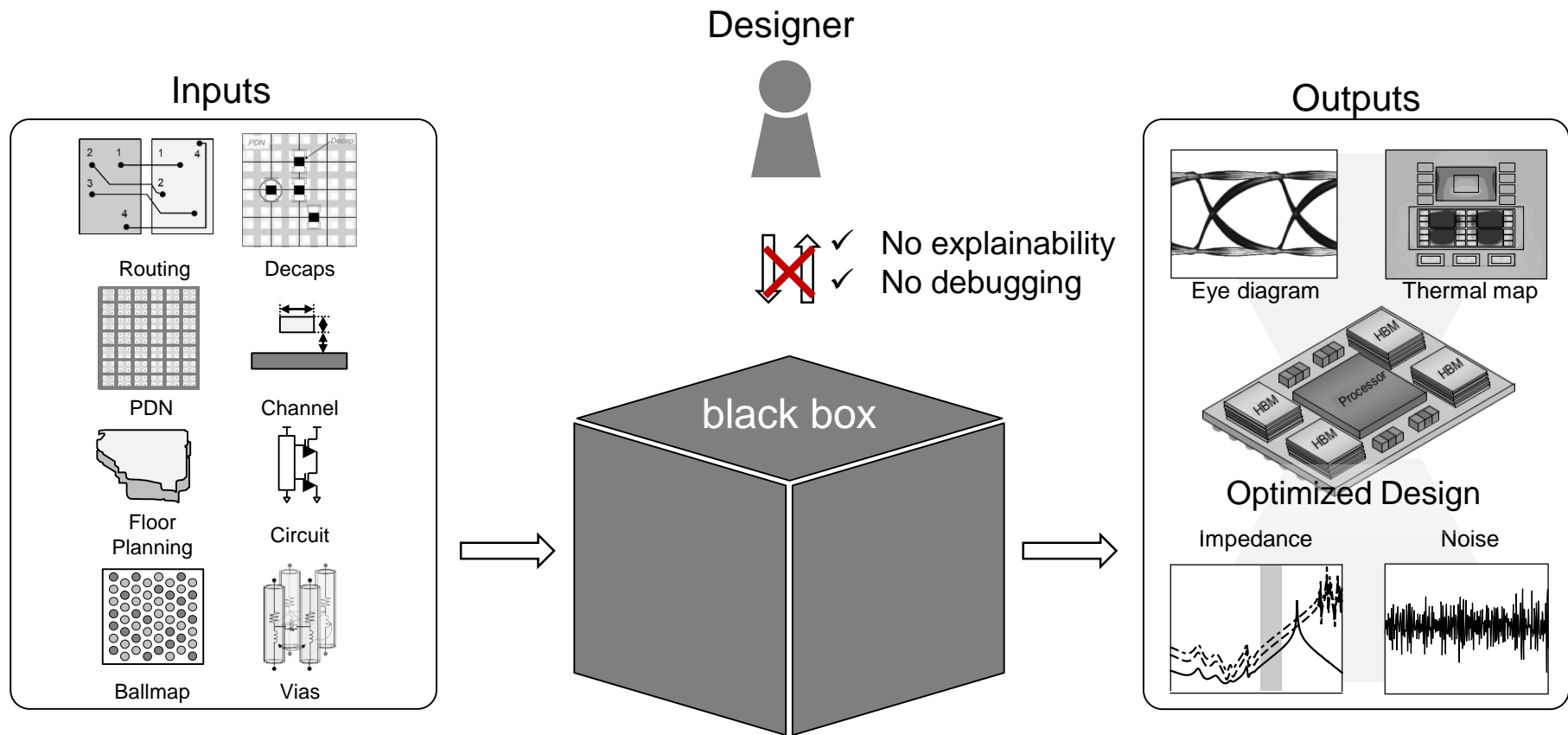
Keunwoo Kim

Advising Professor : Prof. Joungho Kim

TeraByte Interconnection and Package Laboratory  
School of Electrical Engineering  
KAIST



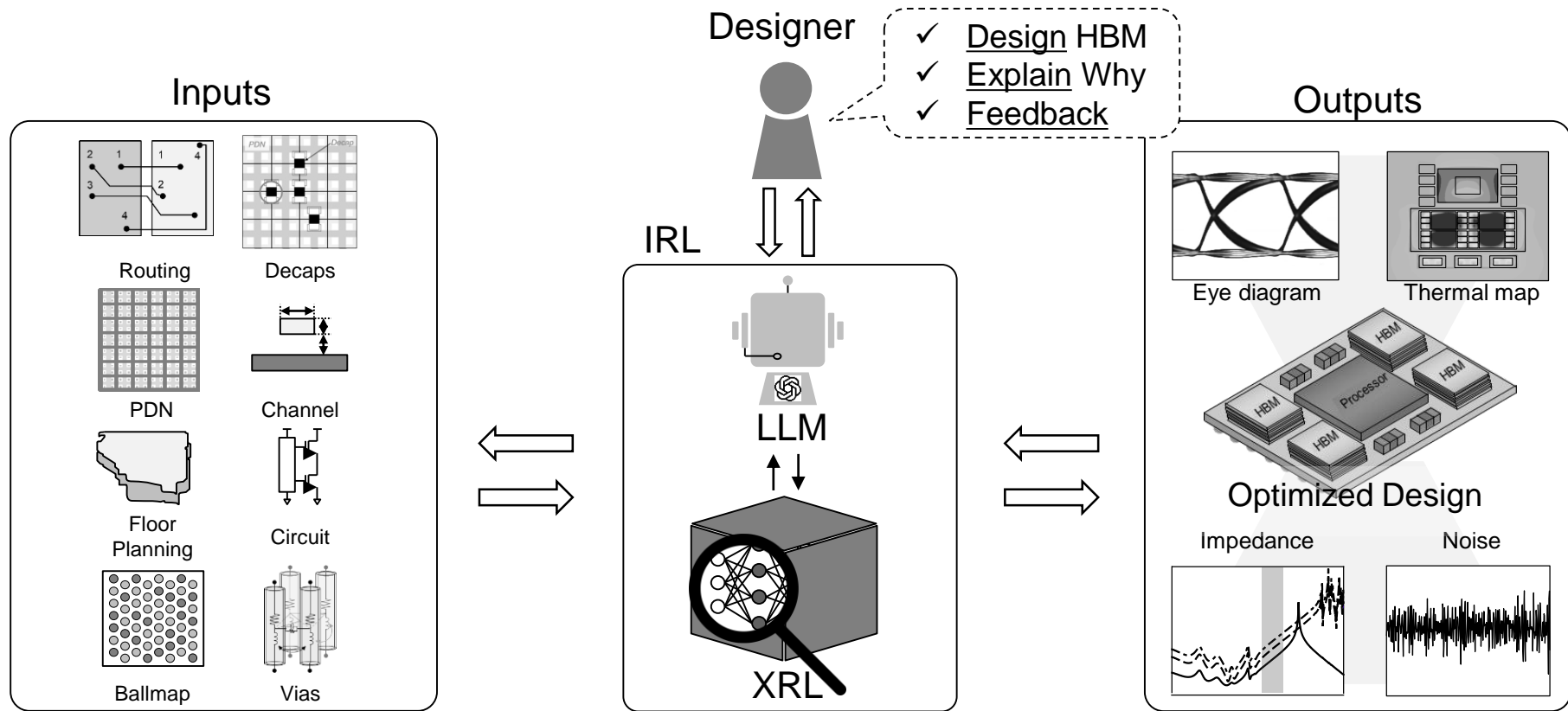
# Conventional Reinforcement Learning-based HBM Design



## < Conventional Reinforcement Learning-based HBM Design: Black box problem >

- Recent approaches have applied reinforcement learning (RL) to HBM design due to the complexity of the design space.
- However, conventional RL agents operate as black boxes, lacking explainability, transparency, and human interaction.
- This limits their usability in iterative, human-in-the-loop engineering workflows.

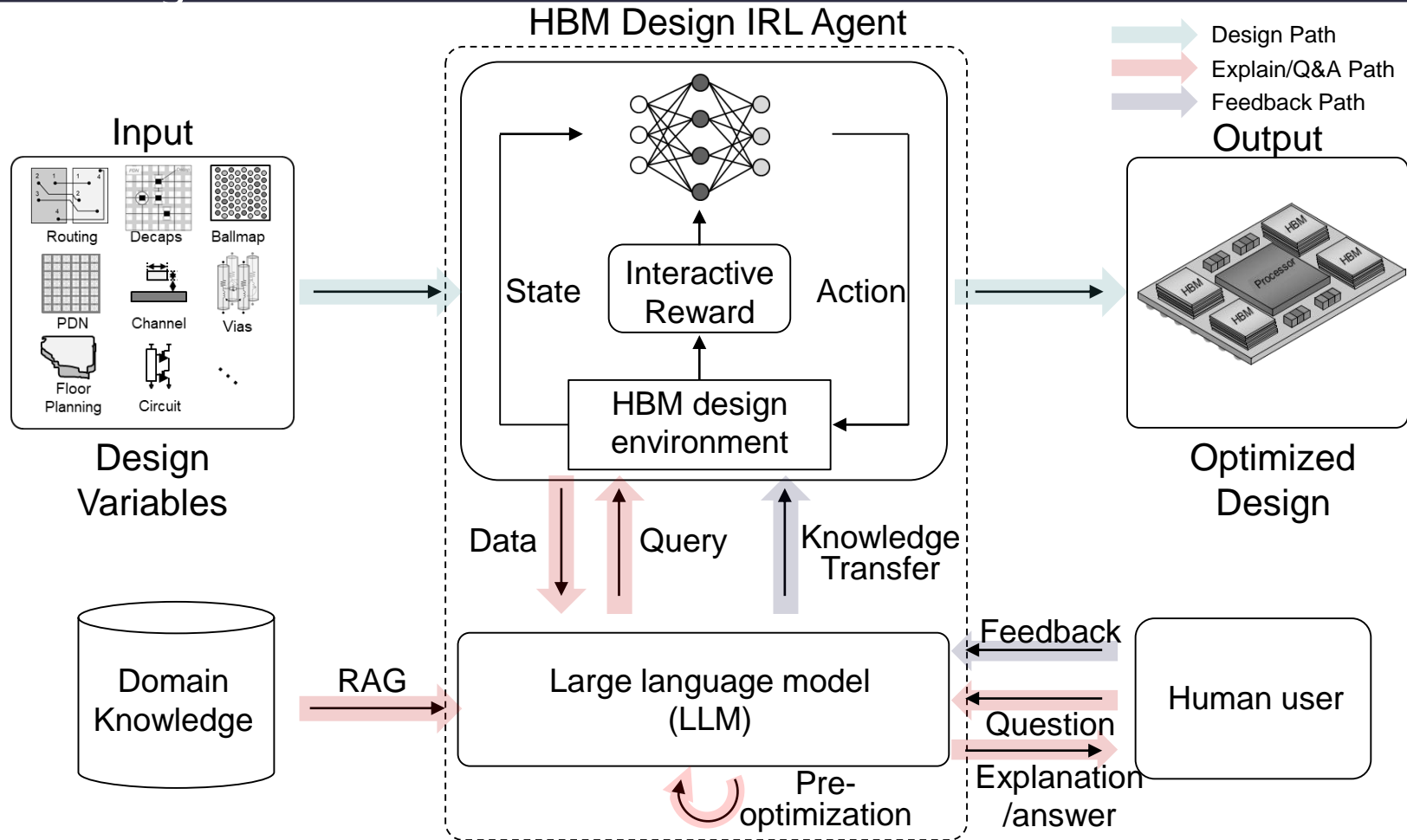
# Interactive Reinforcement Learning-based HBM Design



## < Interactive Reinforcement Learning(IRL)-based HBM Design >

- To overcome the limitations of conventional RL, an LLM-based interactive reinforcement learning (IRL) agent is proposed for HBM design.
- The IRL agent enables human designers to interact with the learning process, receive interpretable feedback, and iteratively refine design outcomes.
- This fosters a collaborative, explainable, and adaptive design workflow.

# LLM-based HBM Design Agent using Interactive Reinforcement Learning

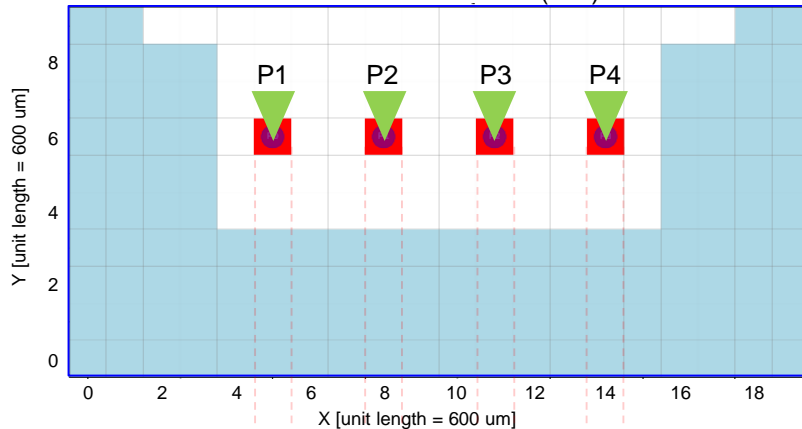


< LLM-based HBM Design Agent using Interactive Reinforcement Learning >

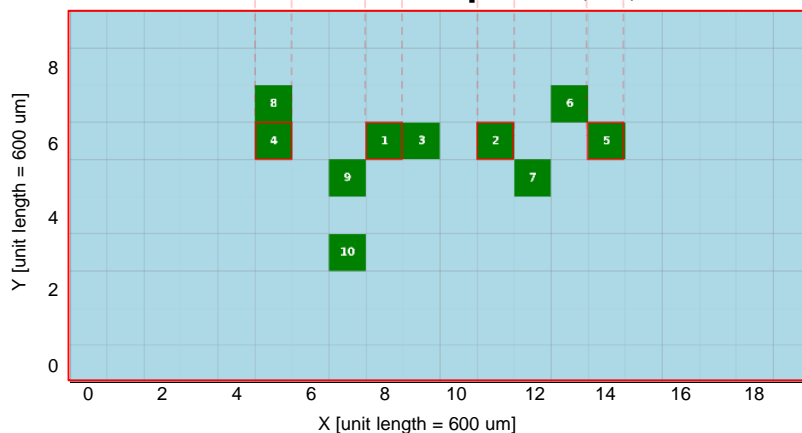
1. **Design** HBM
2. **Explain** Optimized HBM design
3. **Feedback** User's preference/review to IRL Agent

# Explanation using LLM-based Explanation Generator

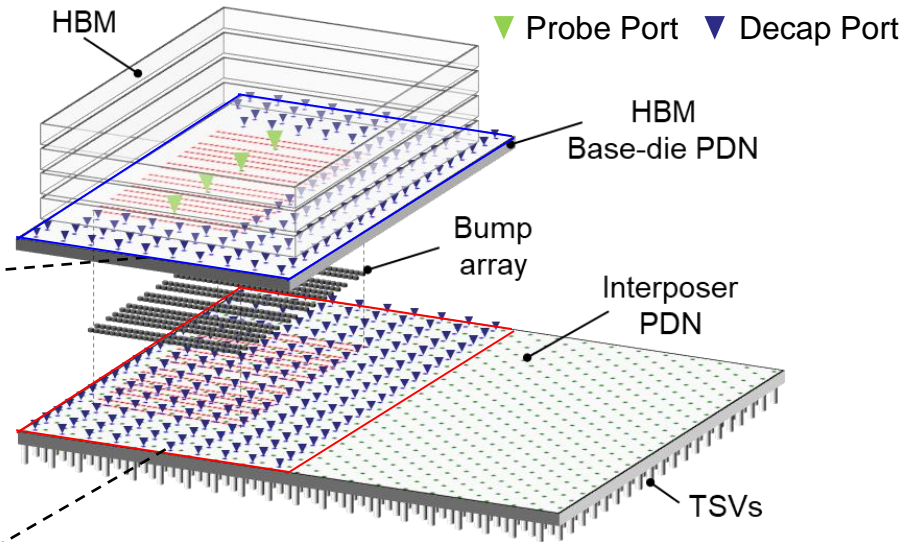
**HBM Base Die (z=1)**



**Silicon Interposer (z=0)**



< Decap assignment result by IRL agent >



< Sideview of HBM and GPU system >



**Q:** Why did you place decap 5 at there?



**A:** The placement of decap 5 at location (6, 14, 0) is expected to have a substantial impact on reducing self-impedance for Port 4



**Q:** What if decap 3 was placed at (6,10,0) instead?



**A:** Better for Transfer Impedance of Port 3 (+1.80), worse for Self Impedance of Port 2 (-28.38).

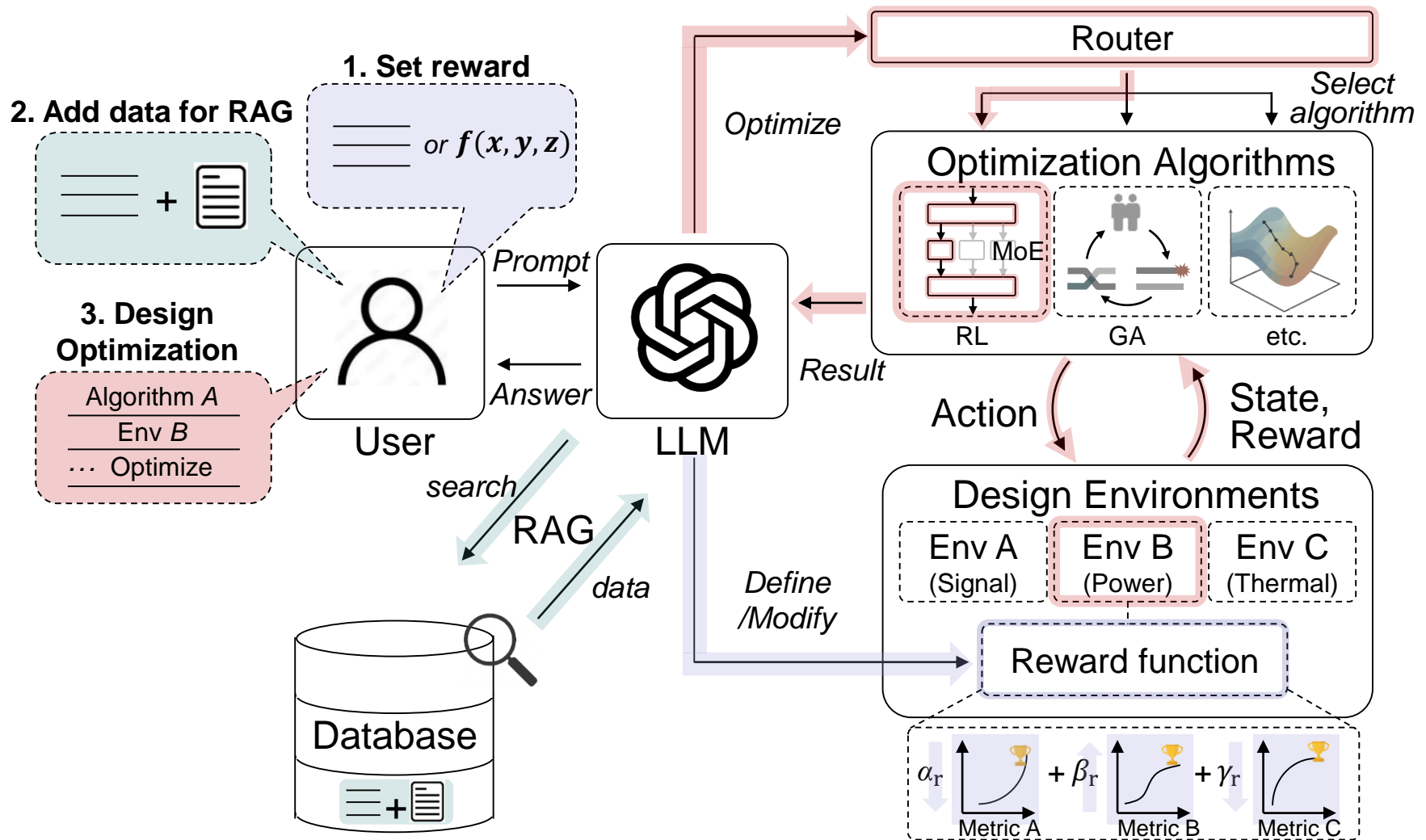
Answer was generated by Llama 3.1:8B

# Demo: Interaction with LLM-based HBM Design AI Agent

The screenshot shows a web browser window with the address bar displaying 'localhost:8501'. The page title is 'Decap Placement Optimization Assistant'. On the left sidebar, there is a logo for 'TERA AI Agent' featuring a stylized brain and circuitry. Below the logo, the 'Settings' section is visible, showing the 'Model Path (.pth file)' as './models/best\_decap\_model\_config\_82'. A green status bar indicates 'Model file found: best\_decap\_model\_config\_8206.pth' with a file size of '23.7 MB'. The 'Ollama Model' is set to 'llama3.1:latest'. Under 'Voice Settings', 'Voice input enabled' is checked. The main content area has a large heading 'Decap Placement Optimization Assistant' and an 'Input Section' with 'Voice Input' and 'Text Input' options. A text input field contains the prompt: 'Try: 'Show current state', 'Consider 10% more transfer impedance' (IRL), or 'Re-training is done?' (Status)...'. A 'Deploy' button is located in the top right corner.

< Demo: Interaction with LLM-based HBM Design AI Agent >

# Future: LLM-based Interactive Design Optimization Agent



< LLM-based Interactive Design Optimization Agent >



Thank you



# HBM CENTRIC

[keunwookim@kaist.ac.kr](mailto:keunwookim@kaist.ac.kr)