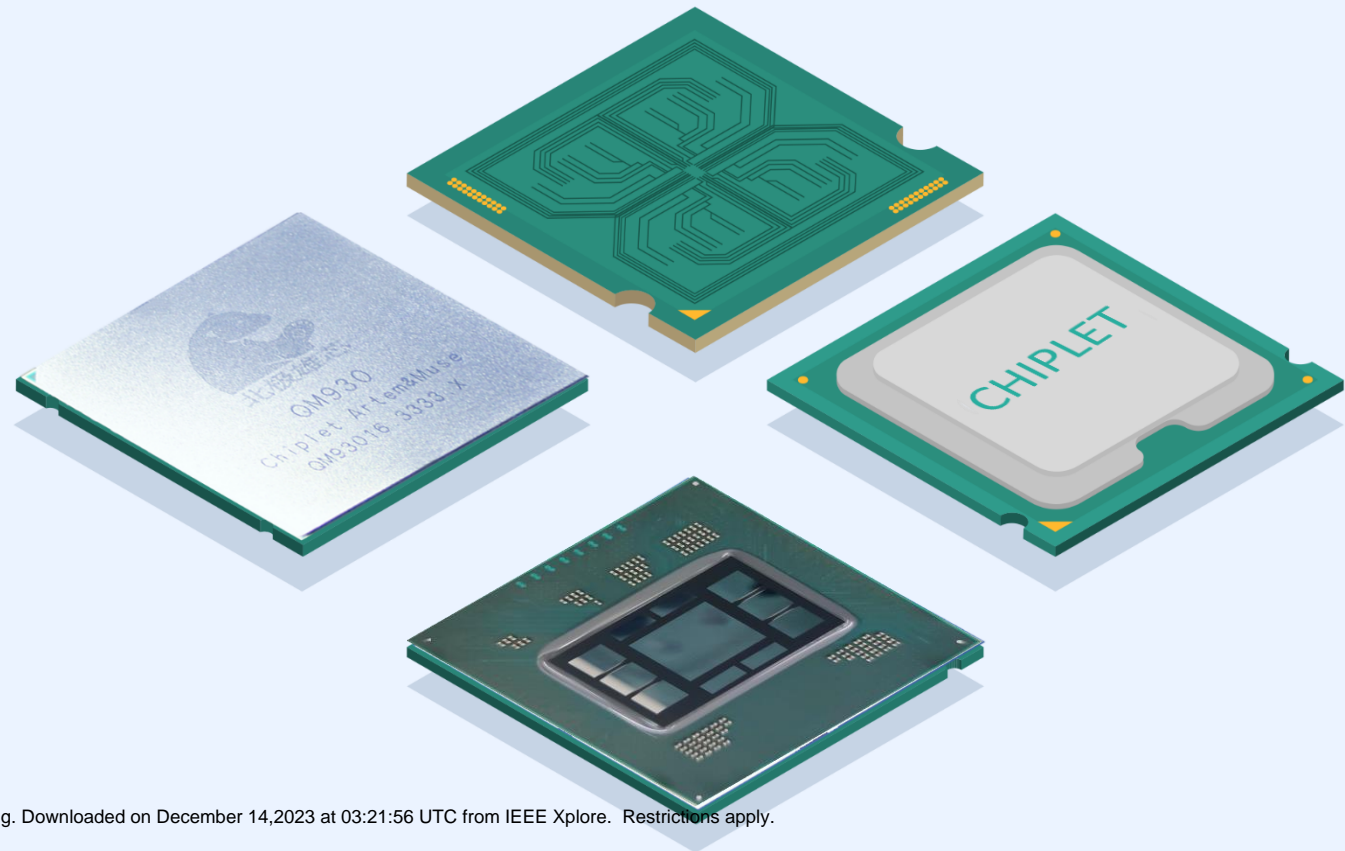# A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Zhanhong Tan[1], Yifu Wu[2], Yannian Zhang[2],
Haobing Shi[2], Wuke Zhang[2], Kaisheng Ma[1]

[1]Tsinghua University,
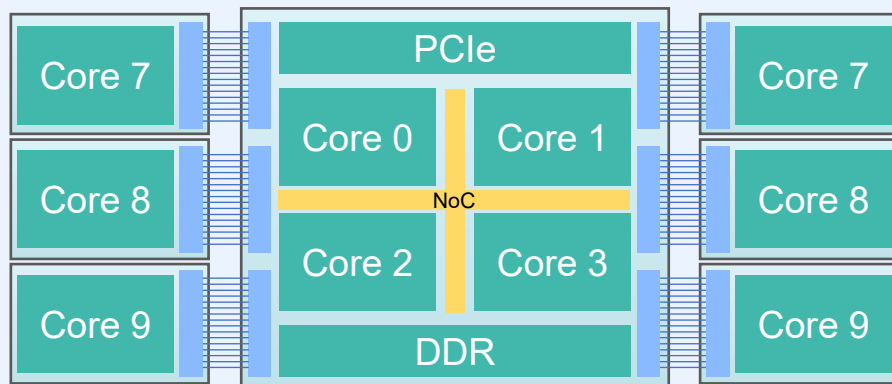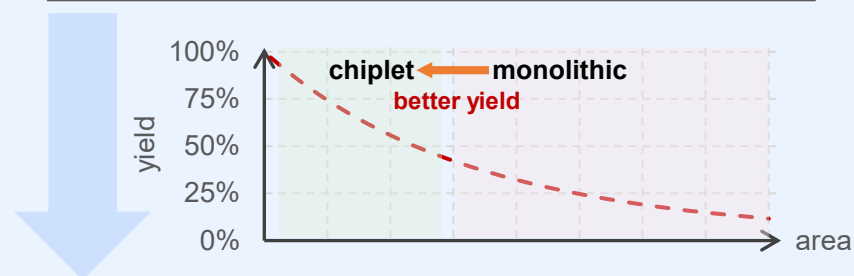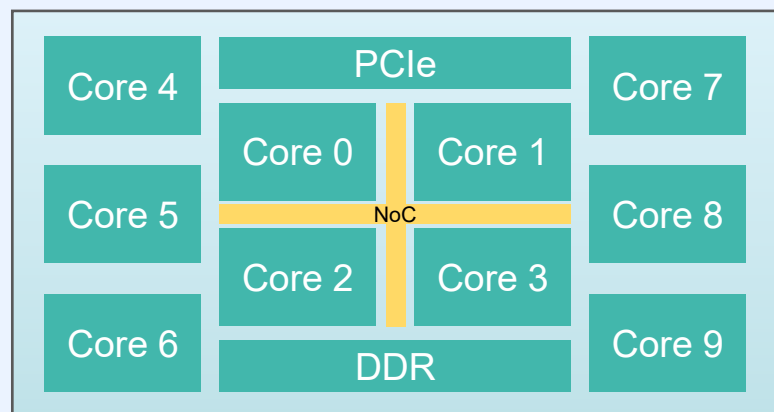[2]Polar Bear Tech

Abstract

With the slowdown of Moore's law, the scenario diversity of specialized computing, and the rapid development of application algorithms, an efficient chip design requires modularization, flexibility, and scalability. In this study, we propose a Chiplet-based deep learning accelerator prototype that contains one HUB Chiplet and six extended SIDE Chiplets integrated on an RDL layer for the 2.5D package. The SIDE and the HUB contain one and four AI cores, respectively.

Given that our Chiplet-system targets diverse scenarios via scalable connected SIDE Chiplets, we need to handle three challenges: a) devise a flexible architecture design supporting diverse shapes, b) search for a workload mapping with low die-to-die communication, and c) adopt a high-bandwidth die-to-die interface to maintain efficient data transfer.

This study proposes a flexible neural core (FNC) featuring dynamic bit-width computing and flexible parallelism. Next, we use a hierarchy-based mapping scheme to decouple different parallelism levels and help analyze the communication. A 12Gbps D2D interface is introduced to achieve 192Gb/s bandwidth per D2D port with 1.04pJ/bit efficiency and 55μm bump pitch.

The proposed seven-Chiplet accelerator achieves a peak performance of **10/20/40 TOPS for INT16/8/4**. When enabling 0~6 SIDE Chiplets, the system power ranges from 4.5W to 12W. The power efficiency of the FNC is **2.02TOPS/W** while that of the overall system is **1.67TOPS/W**.

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Background and Challenges

■ Decouple a monolithic SoC into Chiplets
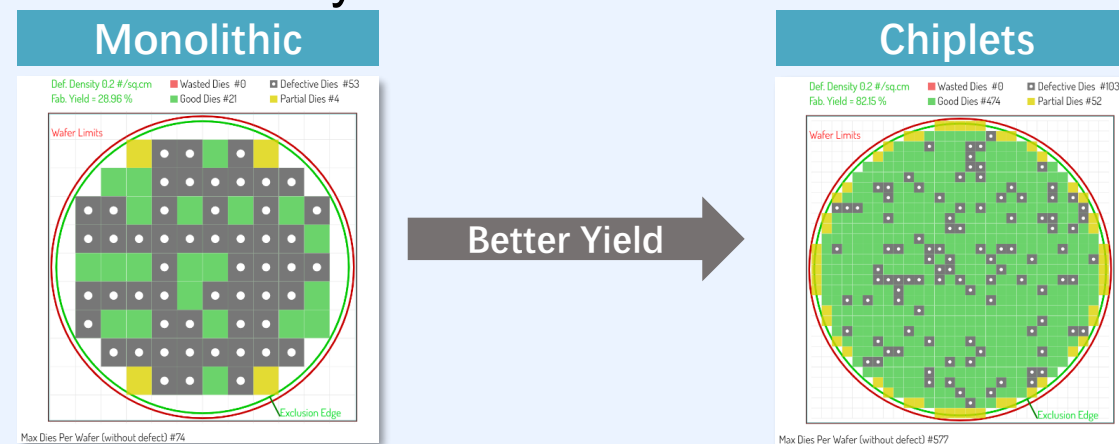
- Better die yield

- Scalability for diverse scenarios

$$M \times \text{Chiplet1} + N \times \text{Chiplet2} + K \times \text{Chiplet3}$$

- Rapid development pace to deliver new products
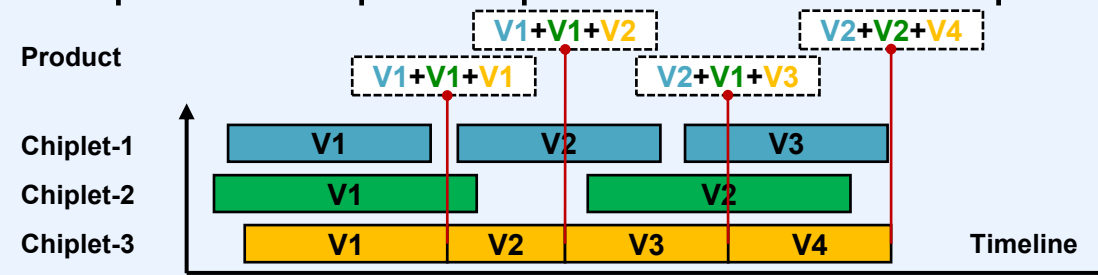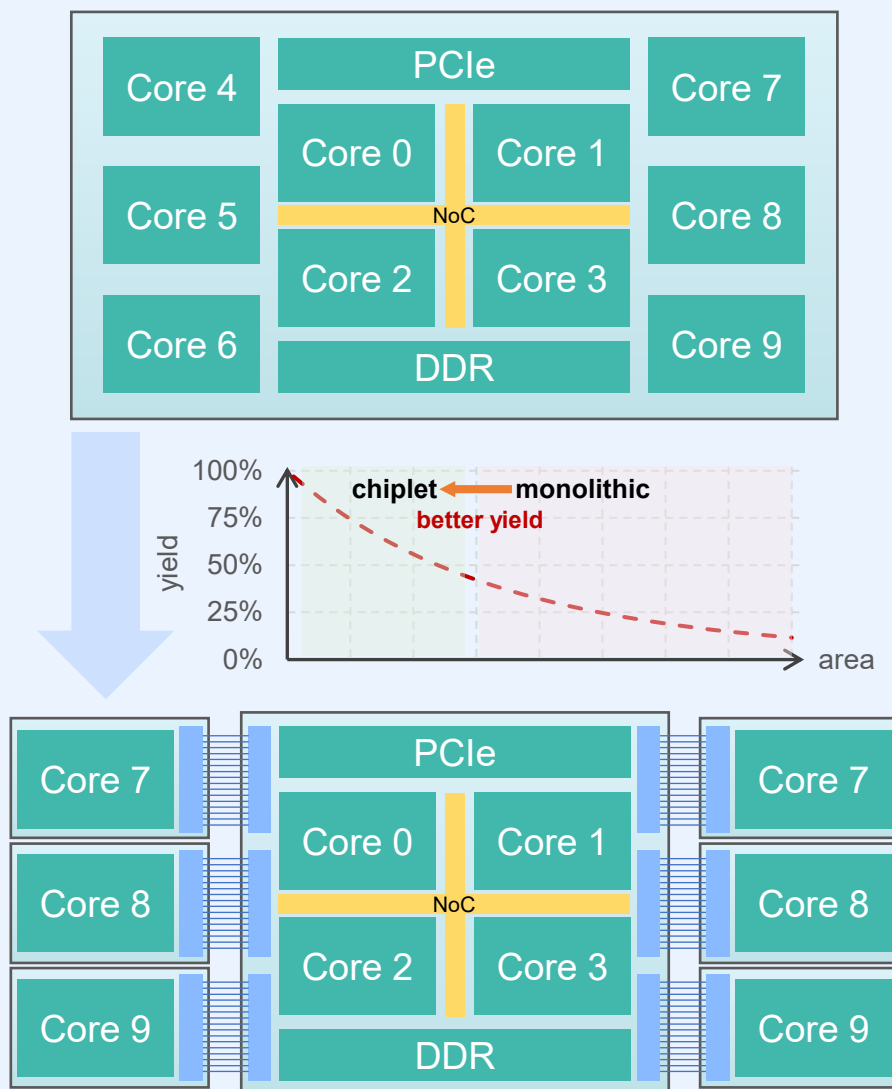
A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Background and Challenges

■ **Decouple a monolithic SoC into Chiplets**
- Better die yield
- Scalability for diverse scenarios
- Rapid development pace to deliver new products

■ **Challenges**
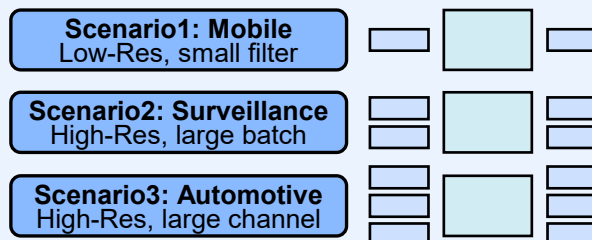
Challenge-1
Flexible architecture design supporting diverse shapes
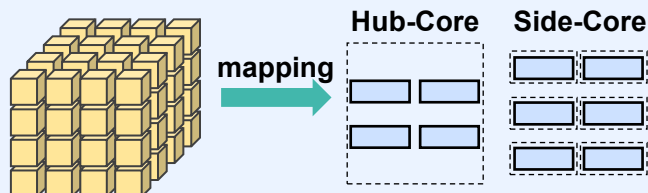
Challenge-2
Efficient workload mapping to optimize Die-to-Die communication
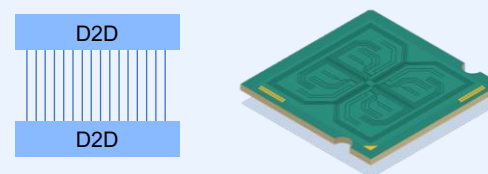
Challenge-3
High-bandwidth D2D and high-density package

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Overall Architecture

Model Description

Workload Partition

Mapping Strategy
Layer1: package-level:
spatial: (1:2 P-type)
temporal: (C-Priority,
C(4)→H(2)→W(2))
Layer1: chiplet-level:
spatial: (C-type)
temporal: ......

Lower-IR Optimization

Data Layout | Instructions

Mapping workflow

host PC

HUB Chiplet

PCIe 3.0 x8 (EP)

FNC | FNC

CLRU | D2D

CLRU | D2D

CLRU | D2D

FNC | FNC

CLRU | D2D

Top RISC-V Ctrl.

CLRU | D2D

Sys. mem | DMA

CLRU | D2D

GDDR6 (32bit, 16Gbps)

external memory access

D2D | FNC

D2D | FNC

D2D | FNC

··· ×6

D2D | FNC

Side Chiplet

■ **Flexible Neural Core (FNC)**
- Reconfigurable architecture for the shape diversity

■ **Mapping dataflow**
- Die-to-Die communication-aware workload generator

■ **Interconnection**
- High-bandwidth Die-to-Die based on 2.5D package
- Efficient chiplet routing unit (CLRU)

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Flexible Neural Core

- **Flexible Interconnect**
  - Arbitrary tile-based workload assignment to 8 cores via a configurable interconnect fabric
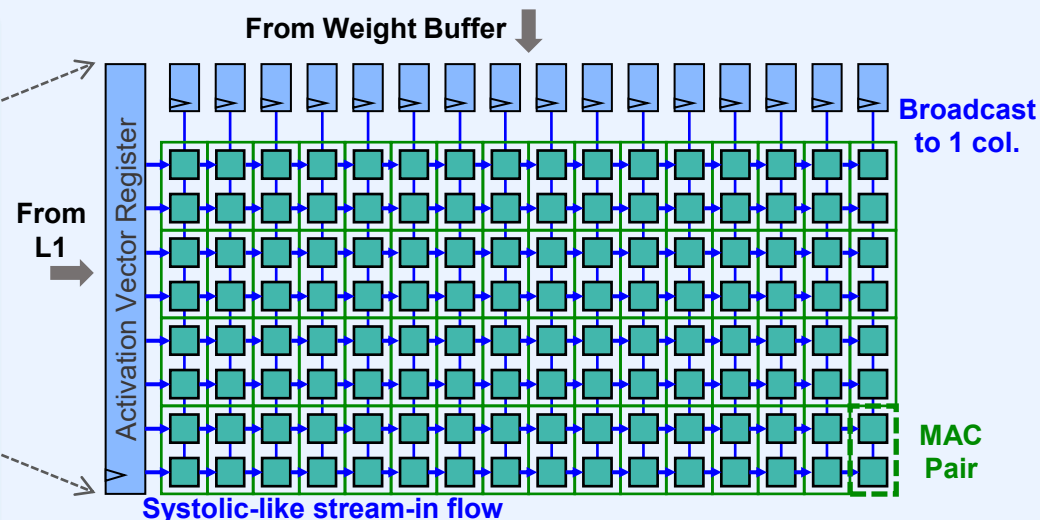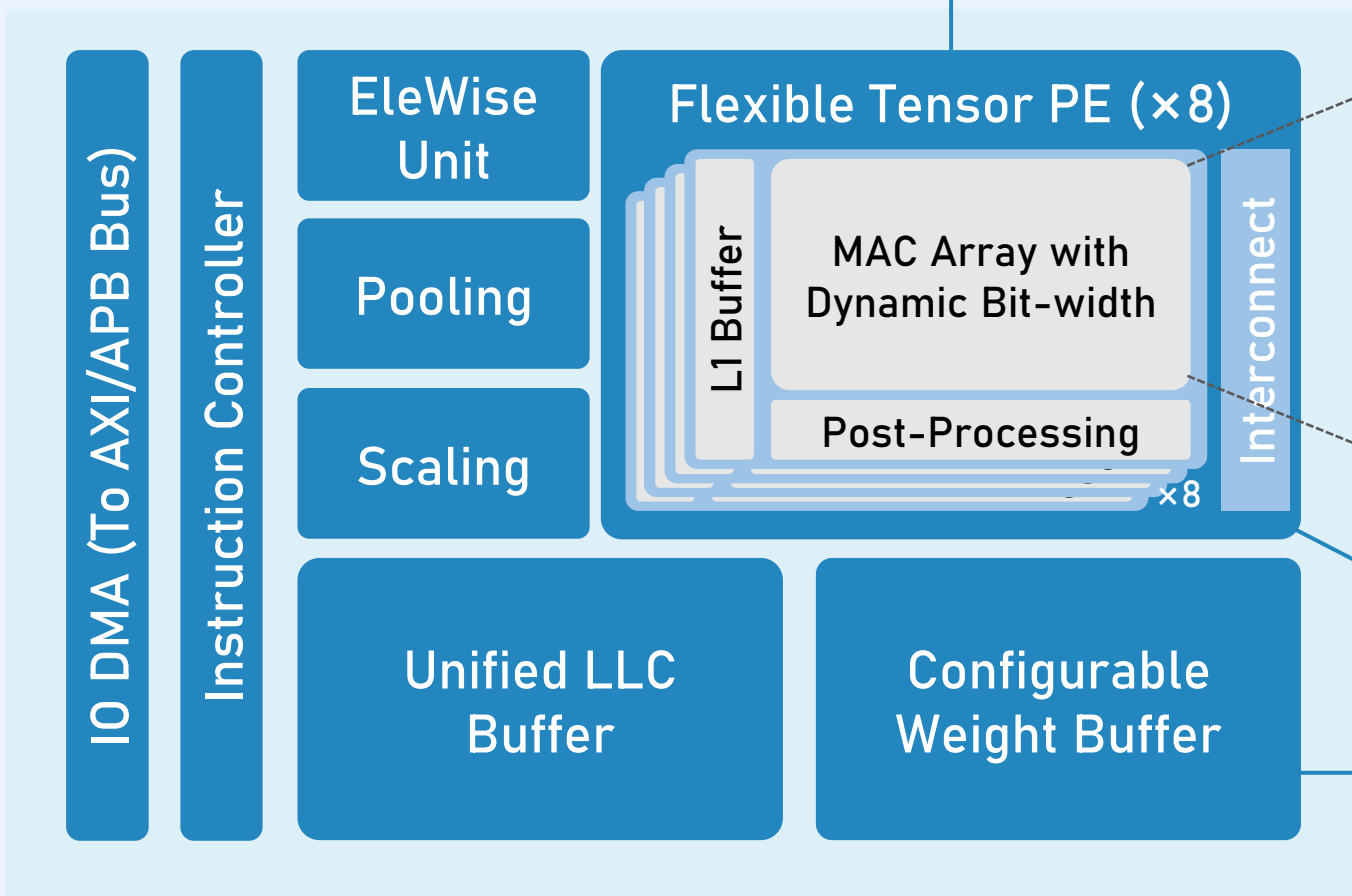
- **MAC array with Dynamic Bit-width**
  - Support INT4/8/16 with full bandwidth utilization of a tensor core
- **Configurable Weight Buffer**
  - Configured for NUMA/UMA mode for arbitrary workload assignment

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Flexible Neural Core

■ **The MAC Pair Supporting for Dynamic Bit-width**
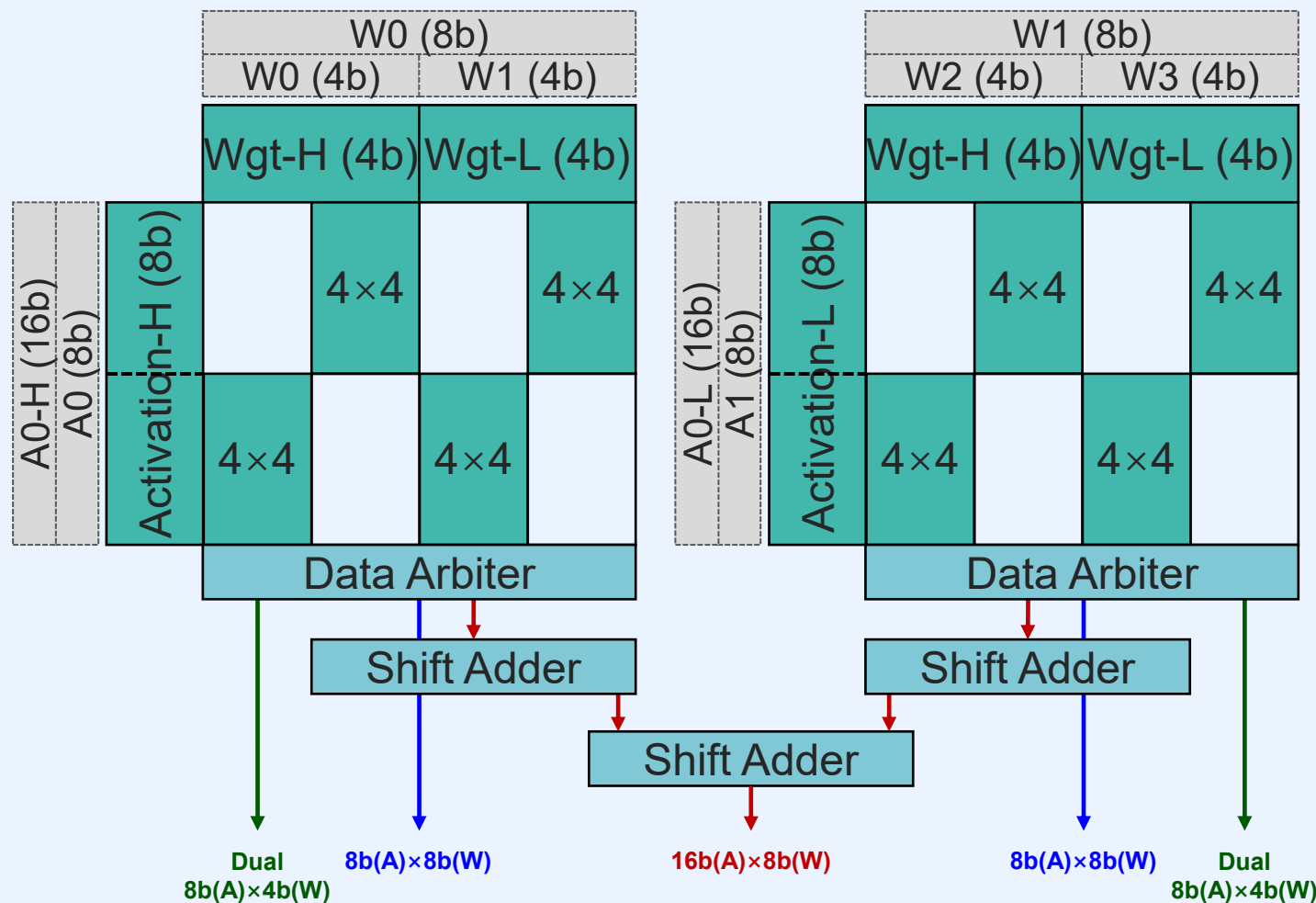
- **Support three quantization modes**
  - ➢ 8b-Acitvation × 4b-Weight
  - ➢ 8b-Acitvation × 8b-Weight
  - ➢ 16b-Acitvation × 8b-Weight

- **Each INT-8 MAC-Pair has eight 4×4 multipliers for mode reuse**

- **In three modes, the bandwidth and compute resources of one MAC-pair are fully utilized**

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Flexible Neural Core

■ Flex-Interconnect and Configurable Weight Buffer

Private Local Buffer

PE   PE

☹ Duplication for shared data

☺ Low access latency

→

Buffer Pool

► Reg   ► Reg

PE   PE

☺ Unified into a pool to share data

☺ Weight-stationary in each PE to overlap weight-loading in local registers

- **Support diverse eight-PE compositions**
  - ➤ 8-tile mode: share weights across 8 PEs for independent output in height/width
  - ➤ 4-tile mode: share weights across 4 PEs and 2 4-PE groups process 2 chunks of output channels
  - ➤ 2-tile mode: 4 2-PE groups for 4 chunks of output channels
  - ➤ 1-tile mode: 8 PEs for 8 chunks of output channels

1. UMA Mode, each bank for 1 core

Bank 0 | Bank 1 | Bank 2 | Bank 3 | Bank 4 | Bank 5 | Bank 6 | Bank 7

PE 0 | PE 1 | PE 2 | PE 3 | PE 4 | PE 5 | PE 6 | PE 7
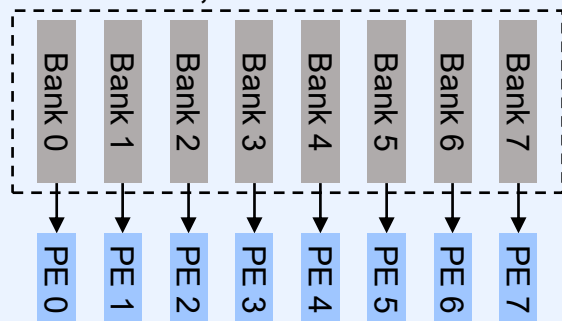
2. Dual-NUMA, each bank for 2 cores

Bank 0 | Bank 1 | Bank 2 | Bank 3 | Bank 4 | Bank 5 | Bank 6 | Bank 7
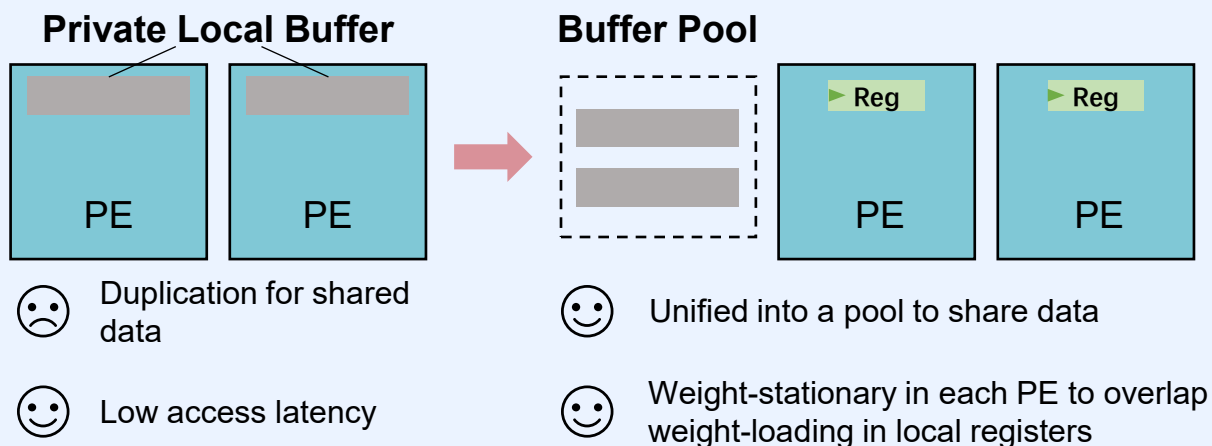
PE 0 | PE 1 | PE 2 | PE 3 | PE 4 | PE 5 | PE 6 | PE 7
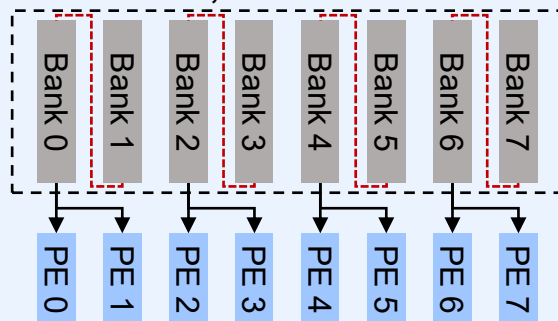
3. Quad-NUMA, each bank for 4 cores

Bank 0 | Bank 1 | Bank 2 | Bank 3 | Bank 4 | Bank 5 | Bank 6 | Bank 7

PE 0 | PE 1 | PE 2 | PE 3 | PE 4 | PE 5 | PE 6 | PE 7
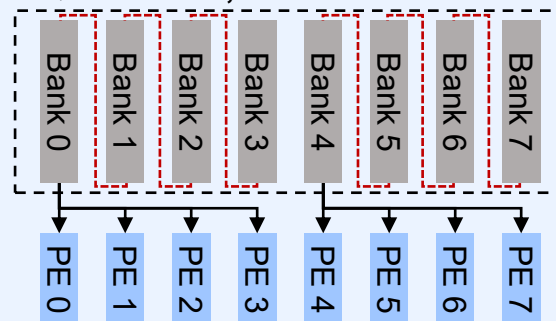
4. Full-NUMA, each bank for 8 cores

Bank 0 | Bank 1 | Bank 2 | Bank 3 | Bank 4 | Bank 5 | Bank 6 | Bank 7

PE 0 | PE 1 | PE 2 | PE 3 | PE 4 | PE 5 | PE 6 | PE 7
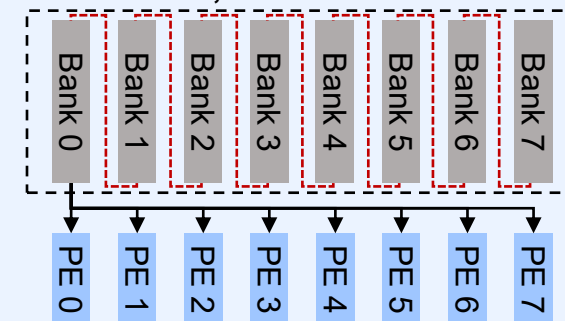
Each bank has 16 sub-banks for 16 columns of MACs

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

PolarBearTech

Dynamic Workload Parallelism

Low Data Dependency Partitioning

Chiplets @Package
(10 cores in 7 Chiplets)

Medium Data Dependency Partitioning

PEs @FNC (Core)
(8 Flexible Tensor PEs)

High Data Dependency Partitioning

MACs @PE
(8×16 MAC array)

Hierarchy-1

$(HO_t, WO_t, CO_t) =$
  temporal(spatial(HO, WO, CO))

(output cube)
$HO×WO×CO$

1 execution for a FNC
$HO_t×WO_t×CO_t$

For FNC-0
For FNC-1

Hierarchy-2

$(HO_c, WO_c, CO_c) =$
  temporal(spatial(HO_t, WO_t, CO_t))

For PE-0 For PE-1
For PE-2 For PE-3

(output cube)
$HO_t×WO_t×CO_t$

1 execution for a PE
$HO_c×WO_c×CO_c$

Hierarchy-3

Primitive($HO_p=1, WO_p=1, CO_p=16$) =
  temporal(spatial(HO_c, WO_c, CO_c))

CO=0 CO=1 ... CO=15
W0 W1 W2
C=0 CI0 MAC MAC MAC
C=1 CI0 MAC MAC MAC
C=2 CI7 MAC MAC MAC
O0 O1 O15
PE Division

Weight-Stationary to reduce data access latency from the buffer pool

I1,0 I1,1 I1,2
W1,0 W1,0 W1,0
ws
PE Workload Loop

- **Critical Position for the "X" buffer:** the inner-most loop related to the index of the X-buffer data (decide the <u>data size</u> on-core)

- **Reuse Region for the "X" buffer:** indicate the <u>reuse efficiency</u> when caching the data in inner loops decided by *critical position*

- Search for an optimized loop range with the highest memory utilization (<u>the largest data size that can be buffered on-core</u>) and reuse efficiency for each buffer

```
// Package-Level for Chiplet Parallelism
(HO_t, WO_t, CO_t) = temporal(spatial(HO, WO, CO))

for c2 = [0 : C2):    high data access overhead
  for h2 = [0 : H2):    // H2 * HO_t = HO
   for w2 = [0 : W2):    // W2 * WO_t = WO
    for c2 = [0 : C2):   // C2 * CO_t = CO

// Chiplet-Level for PE Parallelism
(HO_c, WO_c, CO_c) = temporal(spatial(HO_t, WO_t, CO_t))

for c1 = [0 : C1):    low data access overhead
  for h1 = [0 : H1):    // H1 * HO_c = HO_t
   for w1 = [0 : W1):    // W1 * WO_c = WO_t
    for c1 = [0 : C1):   // C1 *  8 = CO_t
```
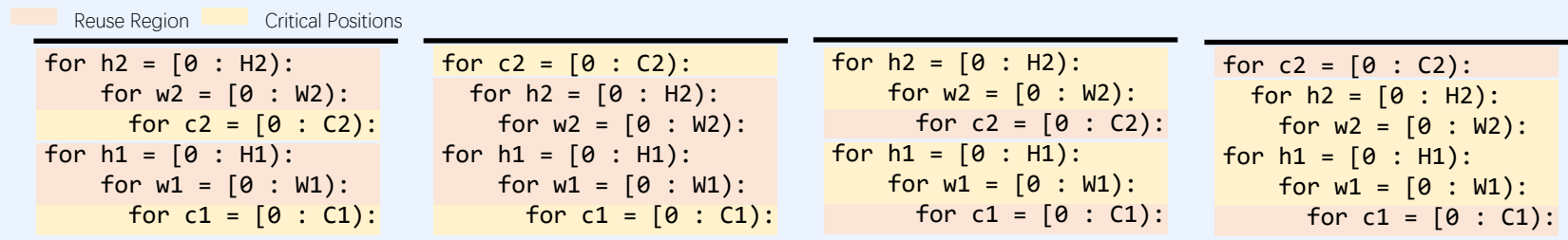
Loop order in the temporal primitive

The overhead bias helps to search for a low D2D communication mapping

Reuse Region   Critical Positions

```
for h2 = [0 : H2):
  for w2 = [0 : W2):
   for c2 = [0 : C2):
for h1 = [0 : H1):
  for w1 = [0 : W1):
   for c1 = [0 : C1):
```
Example-1 for the W-Buf analysis

```
for c2 = [0 : C2):
  for h2 = [0 : H2):
   for w2 = [0 : W2):
for h1 = [0 : H1):
  for w1 = [0 : W1):
   for c1 = [0 : C1):
```
Example-2 for the W-Buf analysis

```
for h2 = [0 : H2):
  for w2 = [0 : W2):
   for c2 = [0 : C2):
for h1 = [0 : H1):
  for w1 = [0 : W1):
   for c1 = [0 : C1):
```
Example-3 for the L1-Buf analysis

```
for c2 = [0 : C2):
  for h2 = [0 : H2):
   for w2 = [0 : W2):
for h1 = [0 : H1):
  for w1 = [0 : W1):
   for c1 = [0 : C1):
```
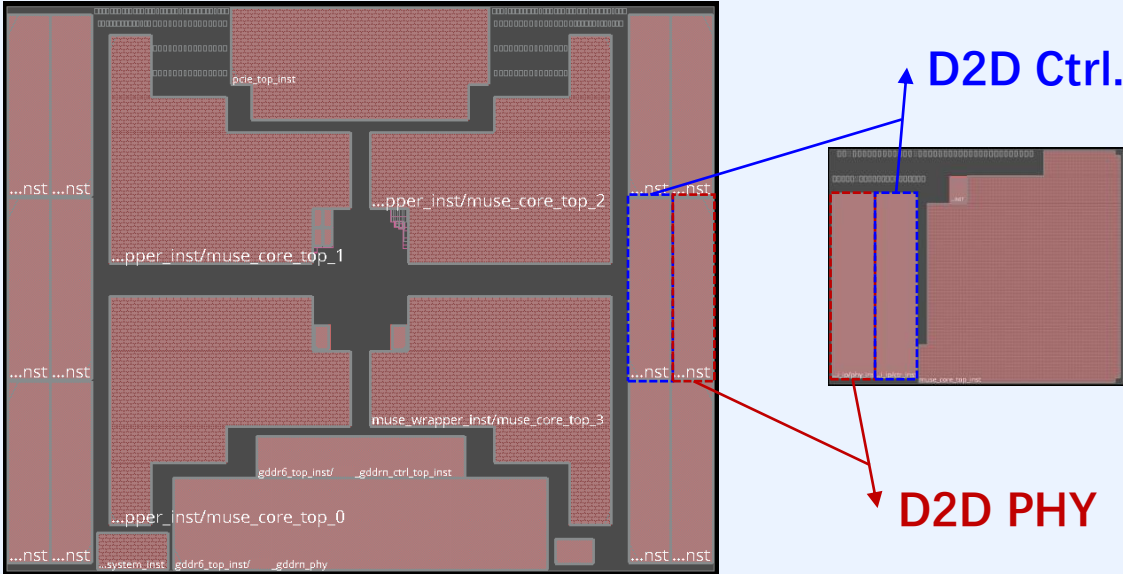Example-4 for the L1-Buf analysis

Notation: *HO, WO, CO: height, width, and channel of the output tensor; X_t: the tile for a Chiplet; X_c: the sub-tile for a PE*

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Chiplet Interconnection and Package

■ High-Bandwidth D2D Interface

D2D Ctrl.

D2D PHY

Bandwidth per D2D	RX: 192Gb/s TX: 192Gb/s	RX(TX) Lane	2(2)
		Data width per lane	8bit
		Data Rate	12Gbps
Bump Pitch	55μm	Package	2.5D
Area	2.2×0.5mm	Power	1.04pJ/bit

■ Chiplet Router Unit (CLRU)
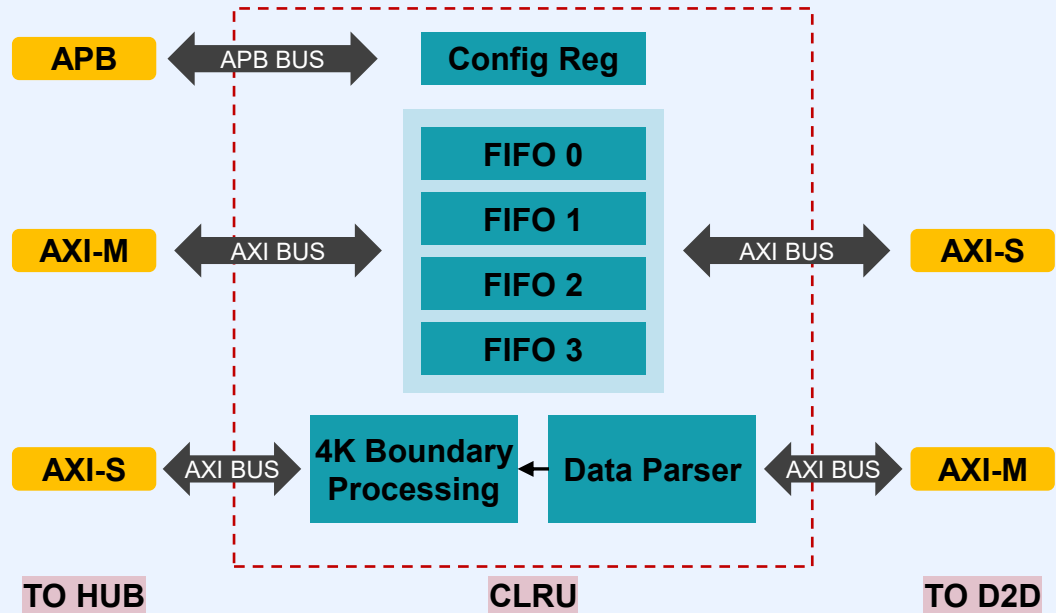
- Four FIFO queues to deal with burst transfer

- Data parser: support the data request from another Chiplet (access memory / other CLRU)

  ➤ *The head packages indicate the transfer mode*

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Chiplet Interconnection and Package

Substrate

33mm

33mm

RDL Layer

Side Chiplet

PCIe 3.0

D2D Region

HUB Chiplet

D2D Region

GDDR6

11mm

19mm
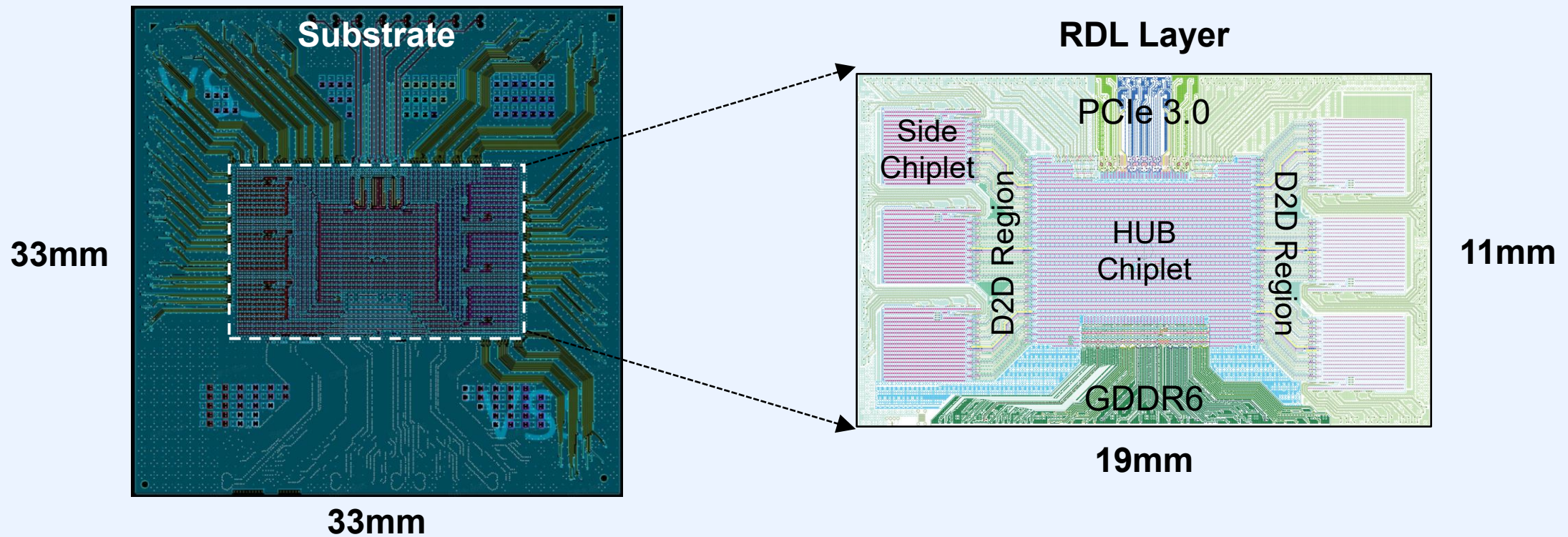
- **Non-conflict IO layout** in the HUB Chiplet to improve the fan-out efficiency

- 2.5D integration with a **high-density 65nm RDL** layer providing 55µm bump pitch

- The RDL layer contributes to a simpler 8-layer substrate of 3-2-3

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Software Stack

Full-Stack

Application Enablement

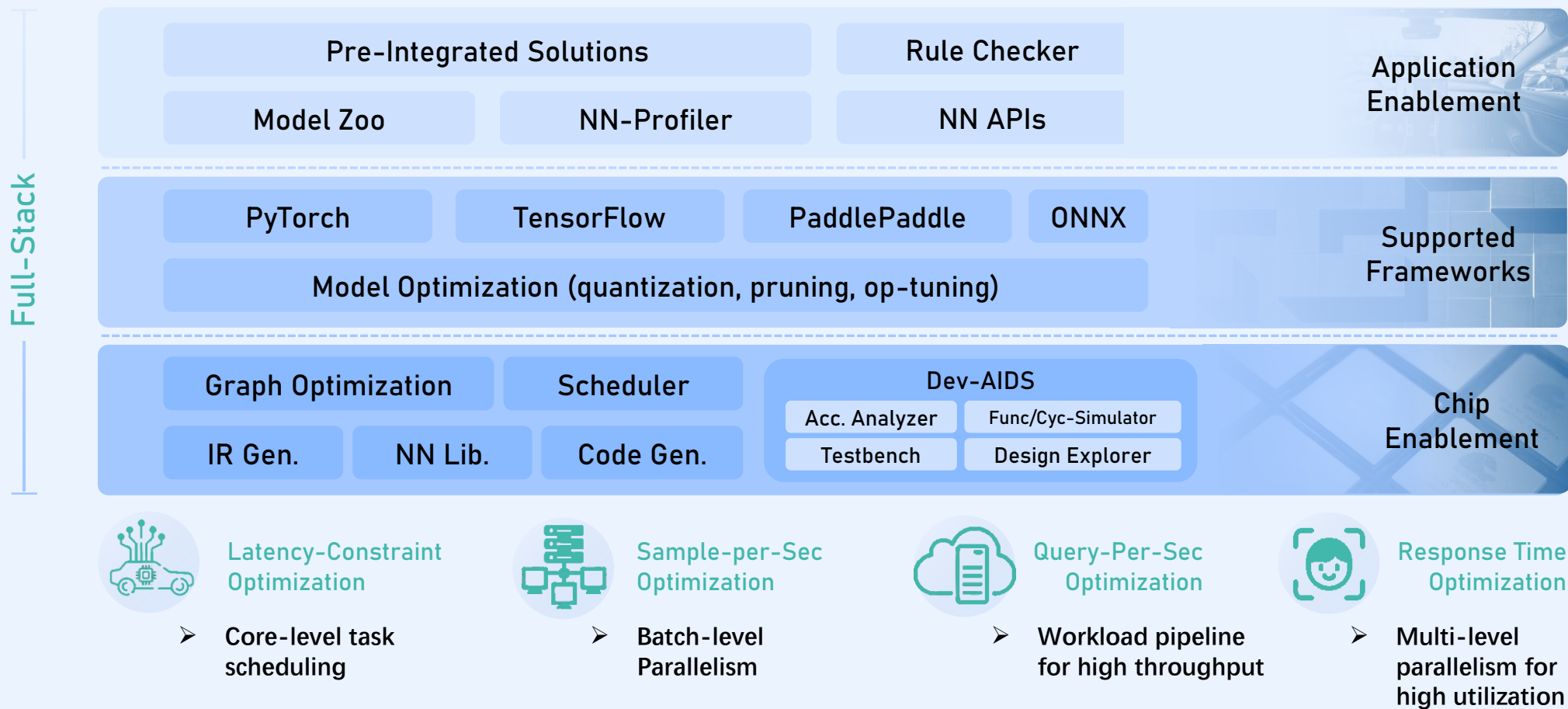| Pre-Integrated Solutions | | Rule Checker |
| Model Zoo | NN-Profiler | NN APIs |

Supported Frameworks

| PyTorch | TensorFlow | PaddlePaddle | ONNX |

Model Optimization (quantization, pruning, op-tuning)

Chip Enablement

Graph Optimization	Scheduler	Dev-AIDS
		Acc. Analyzer / Func/Cyc-Simulator
IR Gen.	NN Lib.	Code Gen.	Testbench / Design Explorer

Latency-Constraint Optimization
➢ Core-level task scheduling

Sample-per-Sec Optimization
➢ Batch-level Parallelism

Query-Per-Sec Optimization
➢ Workload pipeline for high throughput

Response Time Optimization
➢ Multi-level parallelism for high utilization

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

PolarBearTech

Evaluation

■ Evaluations on one Flexible Neural Core

Model	Utilization	Performance
DeepLabv3+		59.6ms, 16.8fps
SqueezeNet		1.5ms, 666.7fps
ResNet101		15.1ms, 66.2fps
ResNet50		8.7ms, 114.9fps
YOLOv3		119.3ms, 8.4fps
YOLOv5s		13.1ms, 76.3fps
YOLOv5n		6.2ms, 161.3fps
DrivableNet		89.7ms, 11.1fps

utilization: 0%, 30%, 60%, 90%
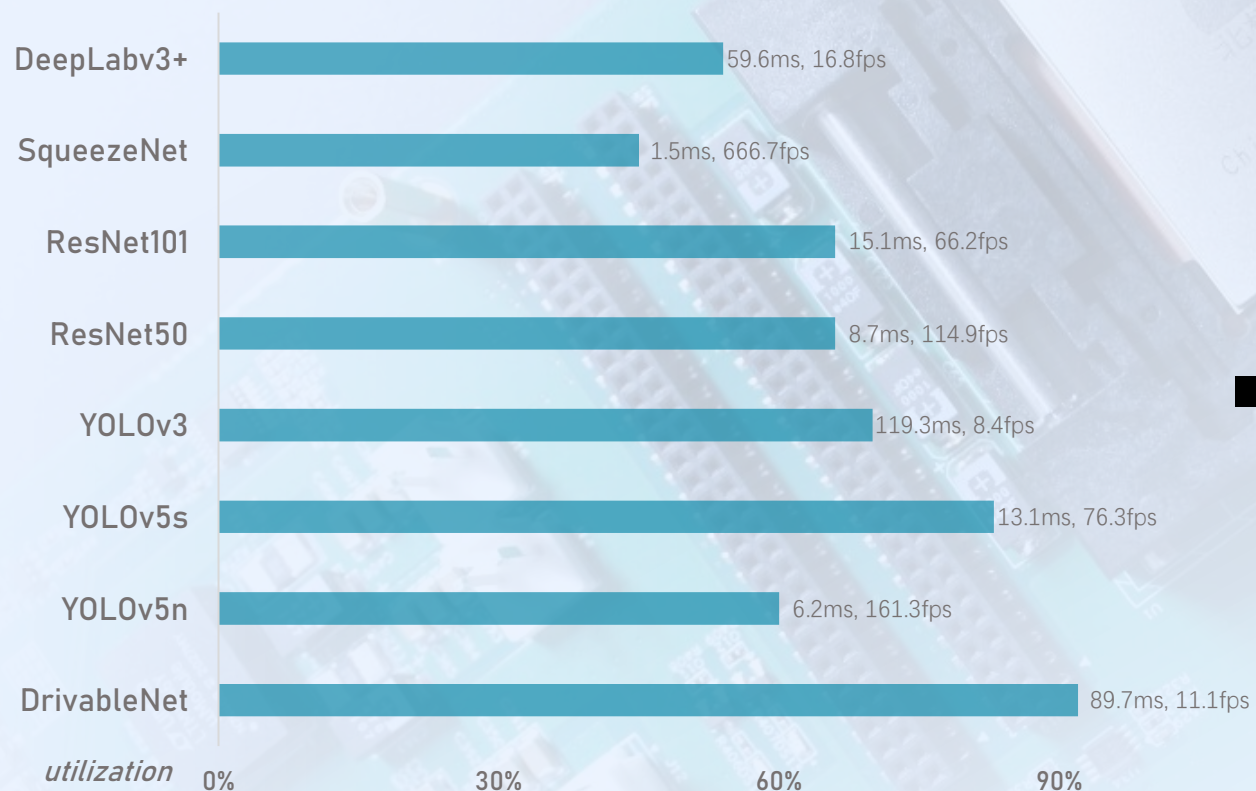
■ Evaluation on computing-bound workload

Speedup by ×9.95
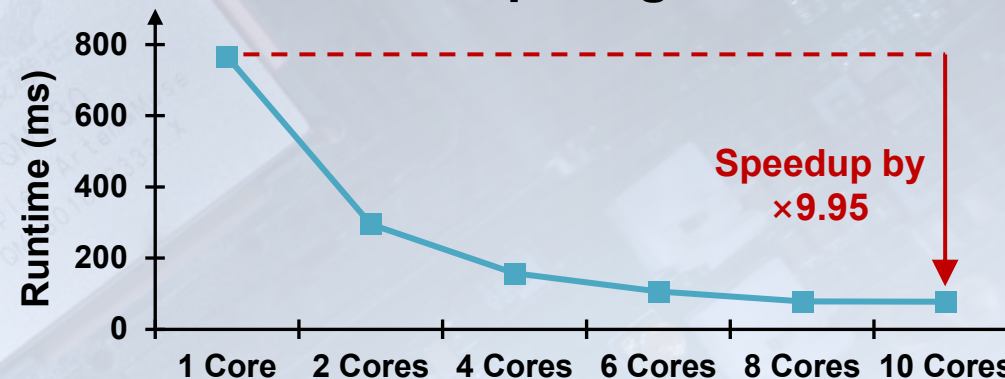
■ Evaluation on IO-bound workload

Speedup by ×6.25

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration
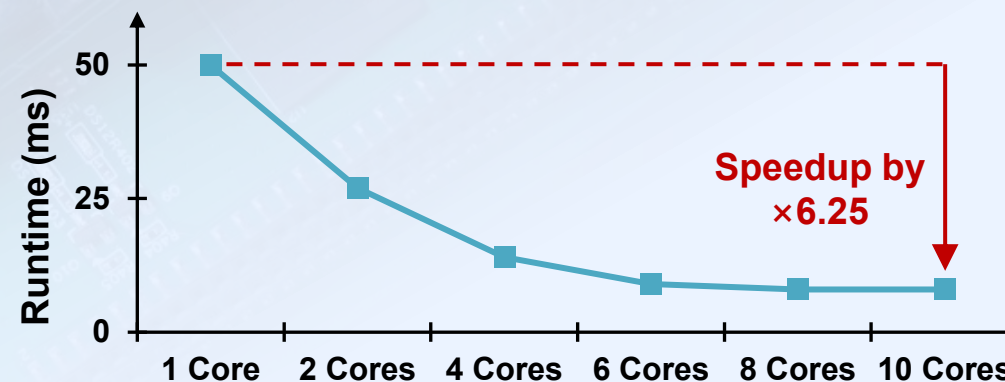
System Board and Demo

■ System PCIe-based Board

For the cooling module

2GB GDDR6

PolarBear QM930EVB

AI Chip

Host Communication for offloading workloads
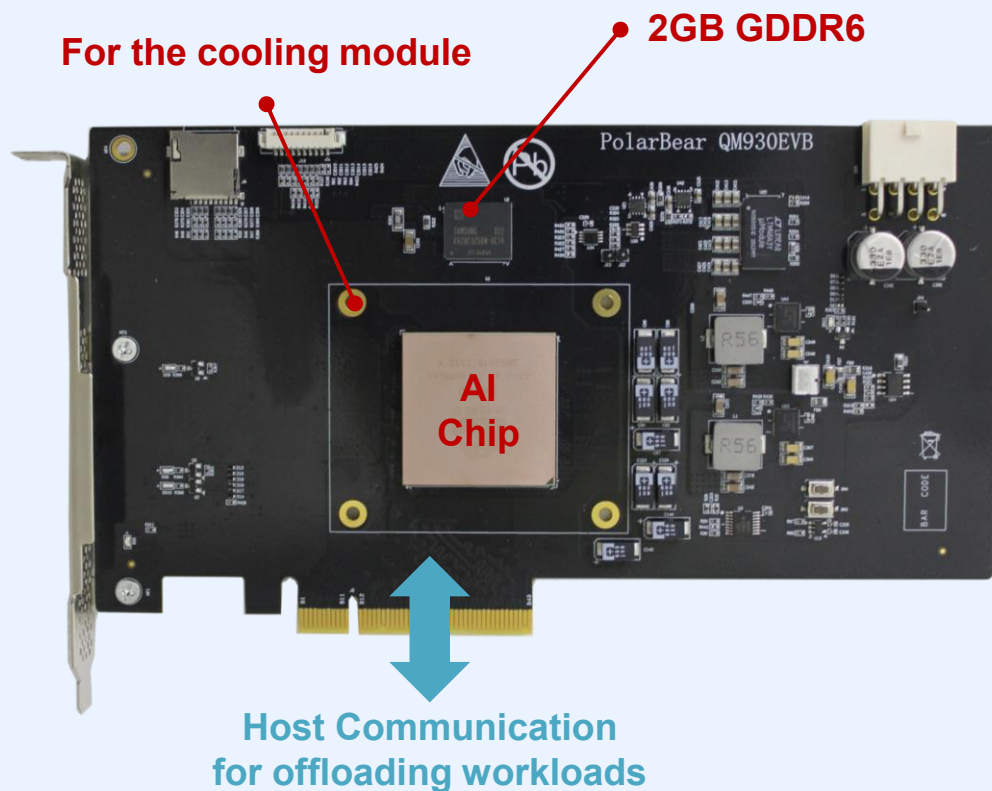
■ Demo for running concurrent 4 models

Model 1: VGG-16   **Model 2: YOLO v5s**   **Model 3: ResNet50**

Model 4: DeepLab v3+

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Chip Summary

HUB Chiplet

6.8mm

8.5mm

SIDE Chiplet

2.8mm

3.5mm

RDL Layer for 2.5D Package
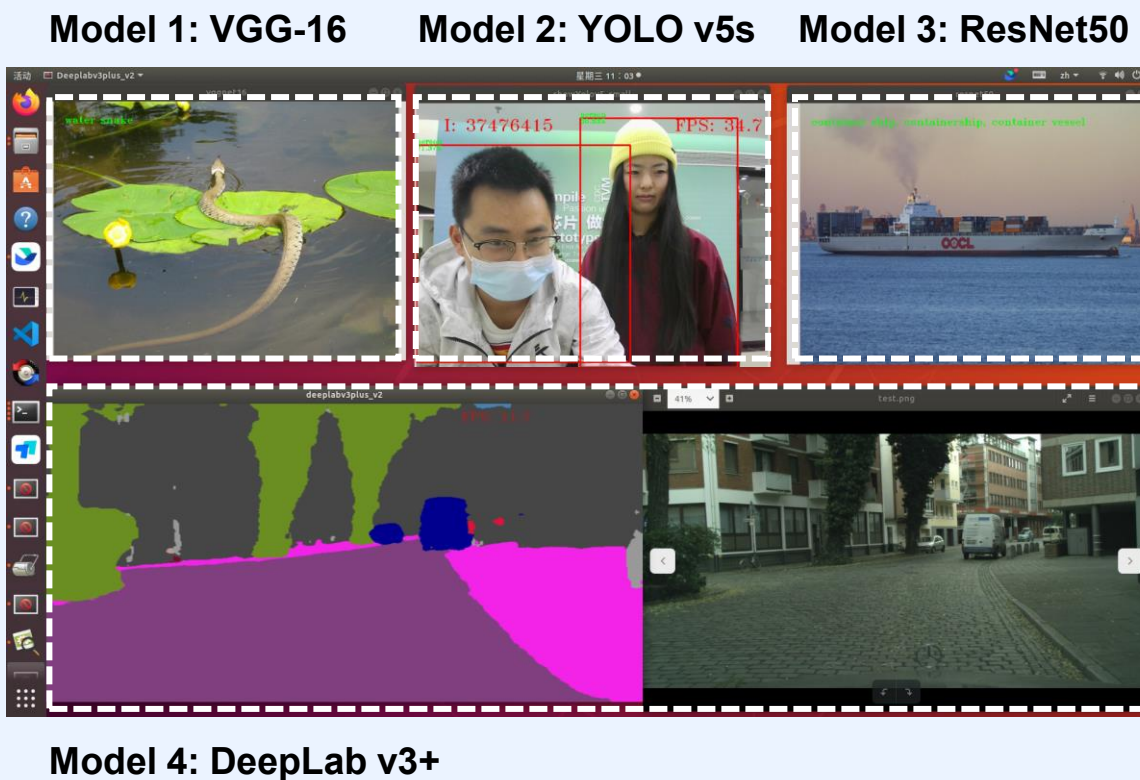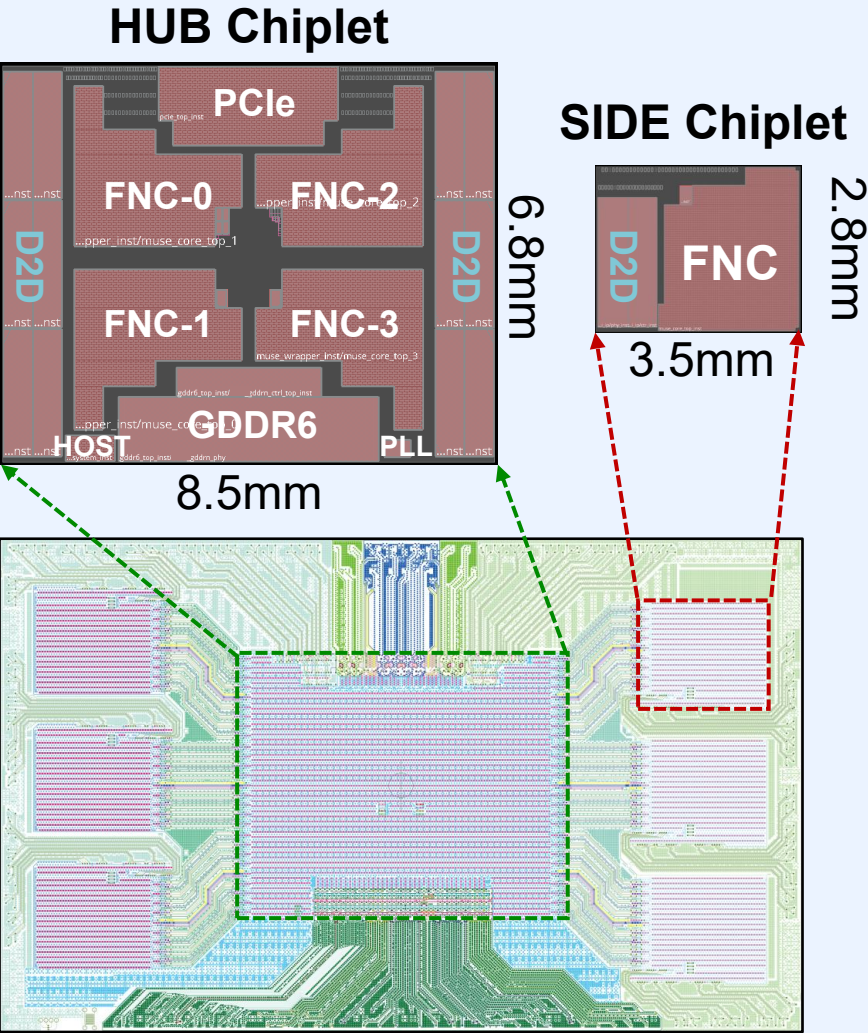
Items		Specifications
Technology		CMOS 12nm
Die Area	HUB Chiplet	8.5mm × 6.8mm = 57.8mm²
	SIDE Chiplet	3.5mm × 2.8mm = 9.8mm²
Supply Voltage		0.8V ~ 1.2V
Frequency		100MHz – 1GHz
Peak Performance	INT4	40TOPS (8b-A × 4b-W)
	INT8	20TOPS (8b-A × 8b-W)
	INT16	10TOPS (16b-A × 8b-W)
NPU Core Efficiency		2.02TOPS/W
Power		4.5W ~ 12W
D2D Bandwidth		6×24GB/s for TX/RX
External Memory Bandwidth		64GB/s (GDDR6)
Bump Pitch for 2.5D Pkg		55μm

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Comparison with Prior Multi-Chiplet Accelerator Works

	Simba (NVIDIA)	CHIMERA (Stanford)	NetFlex (A*STAR)	Ours
Year	2019	2021	2022	**2023**
Technology	16nm	40nm	22nm	**12nm**
Area	6mm²	29.2mm²	11.1mm²	**HUB: 57.8mm²** **Side: 9.8mm²**
Memory Size	752KB SRAM	0.5MB SRAM 2MB RRAM	2492KB SRAM	**HUB: 1.7MB** **Side: 439KB**
Voltage	0.42V ~ 1.2V	1.1V	0.6V ~ 0.89V	**0.8V ~ 1.2V**
Frequency	161MHz – 2001MHz	200MHz	190.3 – 492.3MHz	**600MHz – 1.2GHz**
Power	30 – 4160mW	126mW	57.6 – 499.8mW	**Side: 0.72W** **Hub: 4.75W**
Performance (TOPS)	0.32 – 4.01 (INT8)	2.2 (INT8, FP16)	0.41 – 1.07 (INT16)	**Side Die: 1/2/4 for INT16/8/4, Hub Die: 4/8/16 for INT16/8/4, Total: 10/20/40 for INT16/8/4**
Package	Organic MCM	PCB	HD-FOWLP	**2.5D RDL**
D2D I/O	GRS	C2C Links	AIB	**12Gbps Parallel Interface**
I/O Energy	0.82 – 1.75pJ/b	77pJ/b	3.07pJ/b	**1.04pJ/bit**

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration

Thank You

tanzh@mails.tsinghua.edu.cn

A Scalable Multi-Chiplet Deep Learning Accelerator with Hub-Side 2.5D Heterogeneous Integration