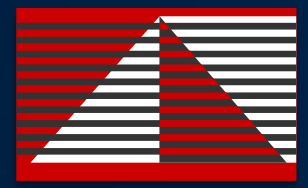


HiT: A Unified Sparsity-Adaptive Architecture for High-Throughput Matrix Multiplication

Tingting Xiang[1], Xiaochen Wang[2], Miao Yu[1], Trevor E. Carlson[1]

[1]*National University of Singapore* [2]*Zhejiang University*

International Symposium on Computer Architecture 2026 • Presentation Session 10D

Outline

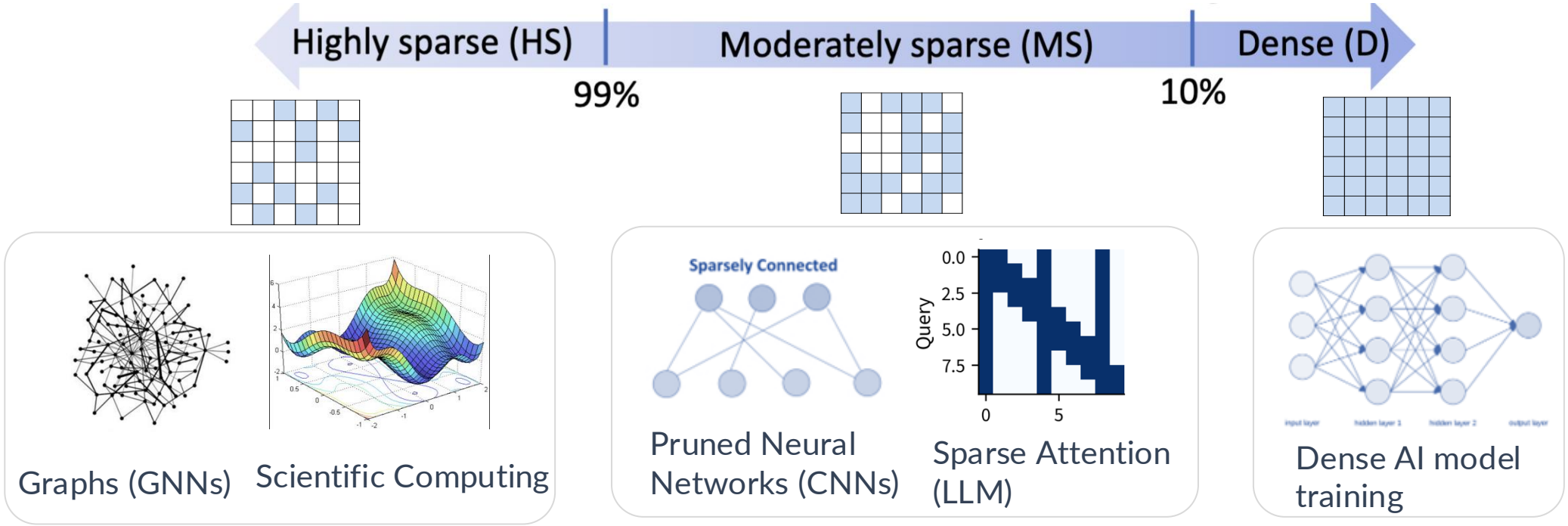
01 **Motivation**
Background & Prior Works

02 **HiT Overview**
Overview & Key Contributions

03 **The Design**
HiT's novel dataflow & Sparsity Handling Components

04 **Evaluation**
Setup & Evaluation results

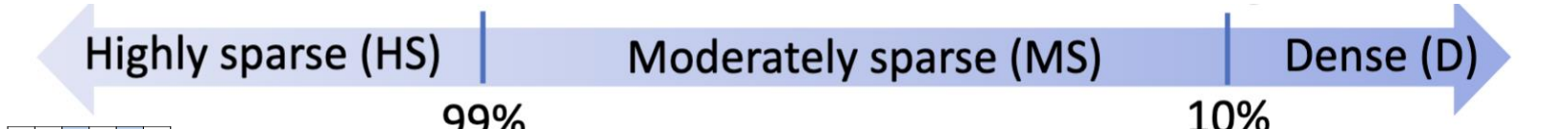
Modern workloads are diverse and span a broad sparsity spectrum



- ✓ Sparsity enables **reduced arithmetic operations** and **memory footprint.**
- ⚠ **Irregular memory access, load imbalance, identifying and processing only the non-zero computations.**

Images from online sources.

Motivation : The Sparsity Challenge

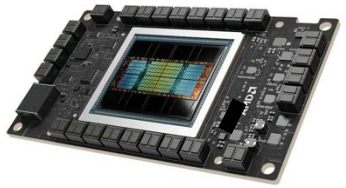


- GNNs
- Scientific Simulation

- Pruned CNNs
- ReLu Function
- Sparse Attention

- Dense AI model training

All run on the same accelerator platform



Time for a unified accelerator that can support the full sparsity spectrum efficiently

Motivation : Limitations of Prior Works

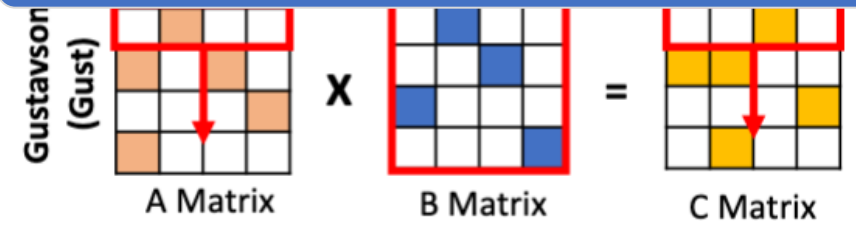
- Prior works typically focus on a specific sparsity, inefficient in workloads outside its target
- State-of-the-art unified accelerator Trapezoid
  - achieves **only 3.125% of peak throughput** on highly sparse x highly sparse workloads
  - a severe **bottleneck** for graph analytics and scientific computing

Efficiently Supports	Highly Sparse	Moderately Sparse	Dense
SpArch (HPCA'20)	✓	✗	✗
SIGMA (HPCA'20)	✗	✓	✗
Trapezoid (ISCA'24)	~	✓	✓
HiT	✓	✓	✓

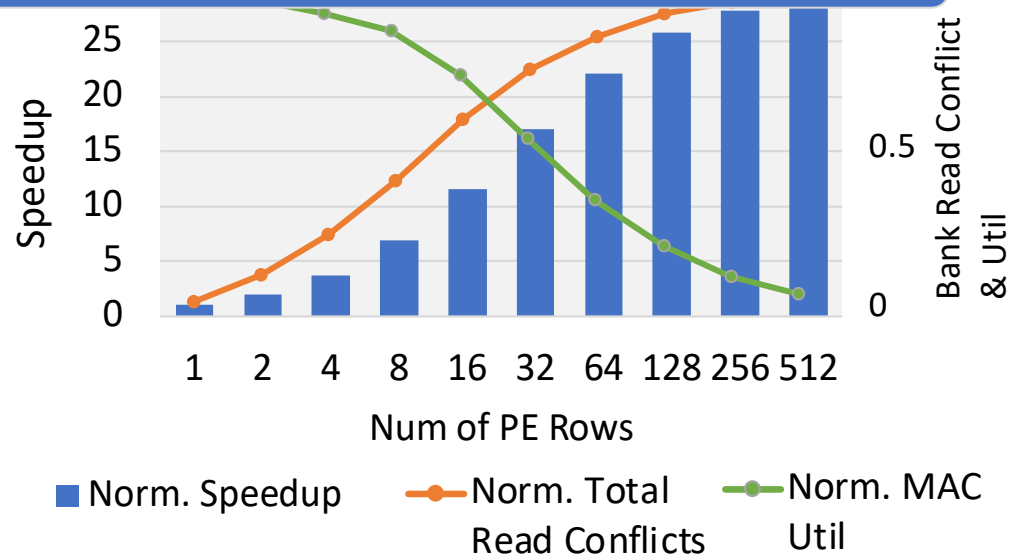
Motivation: Limitations of Gustavson Dataflow in HS Workloads

- Trapezoid implements Gustavson-based dataflow for Highly Sparse (HS) computations
- Gustavson is effective at a **small scale**, with a small number of MACs and limited parallelism.
- The fundamental issue:
  - ✕ **irregular memory accesses to Matrix B** cause **bank conflicts** that intensify with parallelism
  - ✕ scaling challenge, **quadratic area and power overhead** due to crossbar growth — an unsustainable solution

The memory subsystem, not compute, becomes the dominant bottleneck.



+ Regular access of A and C Matrix
+ No data matching needed

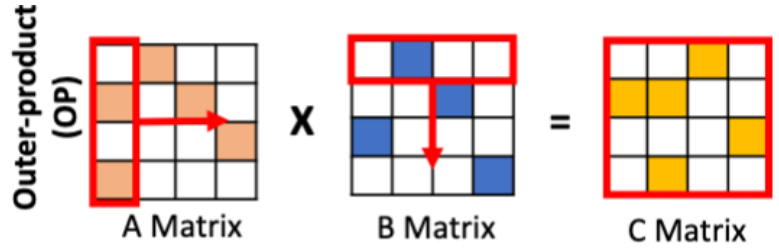


Motivation: Rethinking the Outer-product

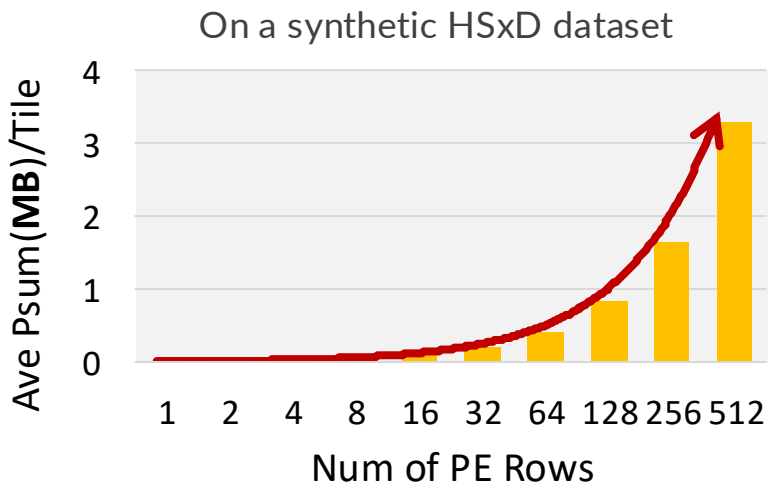
- Outer-product eliminates irregular B Matrix access
  - No memory bank conflict
  - No complex memory subsystem

→ ☑ Scalable

- It generates **a large number of partial sums**



+ Regular access of A and B Matrix
+ No data matching needed



A new outer-product based dataflow that ***allows regular memory access*** *and* ***more regular partial sums accumulation.***

HiT Overview and Key Contributions

HiT is a unified sparsity-adaptive architecture

- delivers **consistently high throughput across the entire sparsity spectrum**

Novel Dataflows

HSparse and **Msparse** that deliver higher MAC utilization and enable regular memory access & on-chip accumulation.

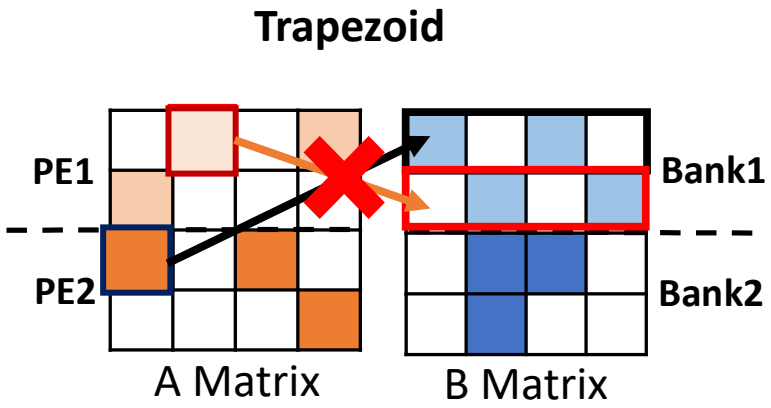
Output Sparsity Exploitation

Increases buffer utilization and improves compute density and throughput.

Unified Architecture

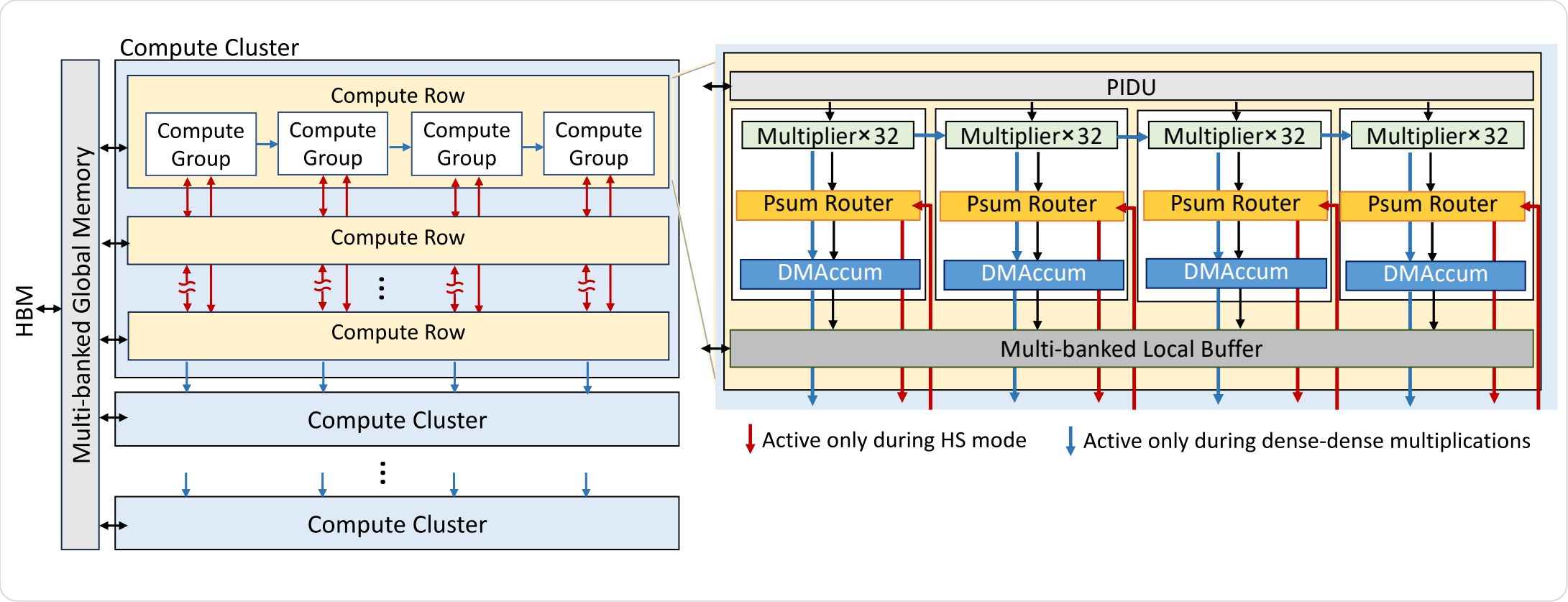
Single architecture that supports 3 dataflows with novel **Parallel Intersection & Distribution Unit** and **DMAccum**.

HiT Overview and Key Contributions



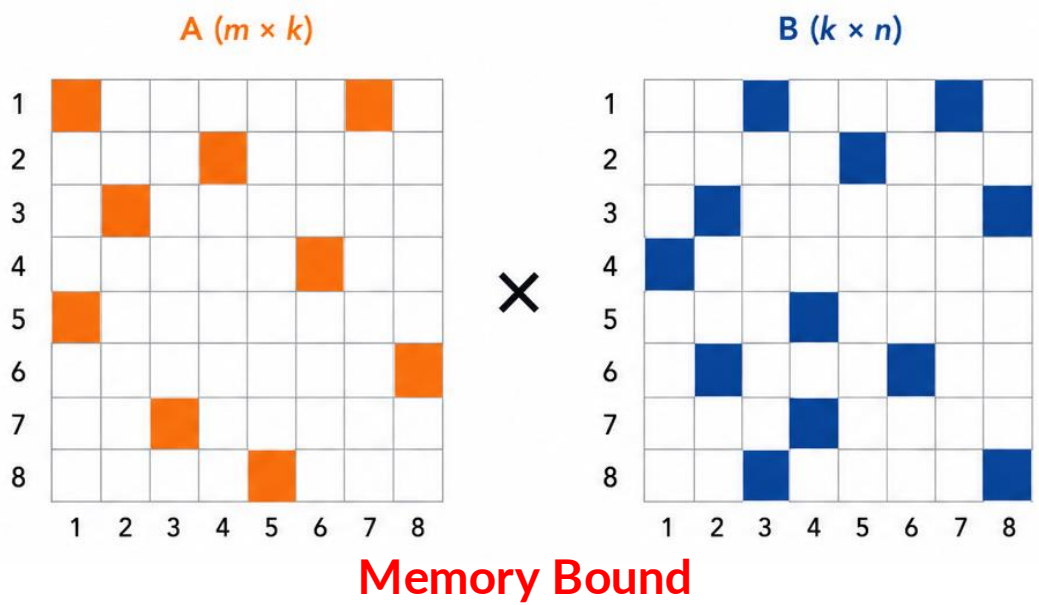
*Trapezoid completes in 5 steps, HiT completes in **3** steps*

HiT Architecture Overview



4 Compute Clusters, each with 32 Compute Rows
128 Multipliers and Accumulators per Compute Row
Peak Throughput: 32 TFLOPS

HiT: HSparse Motivation



Memory Bound

Low arithmetic intensity

Compute resources have a low utilization, reducing throughput.

Limited data reuse

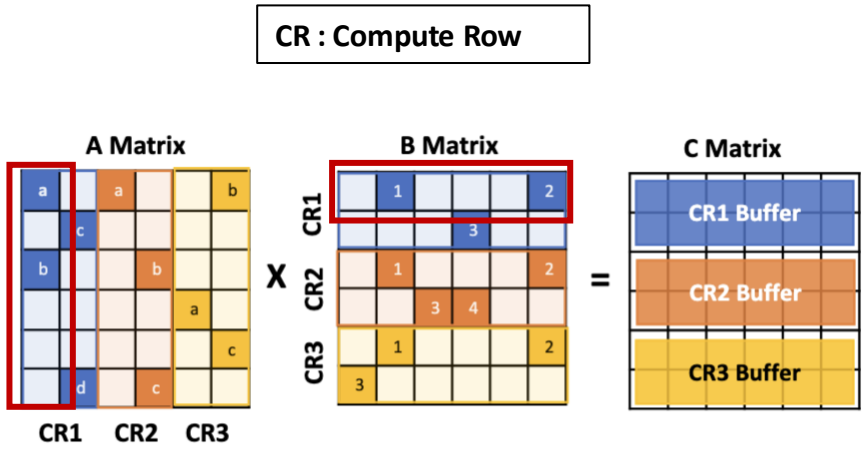
Performance degrades due to constant data movement.

BUT the absolute amount of computation remains substantial.

Increase the number of effective computations done in parallel

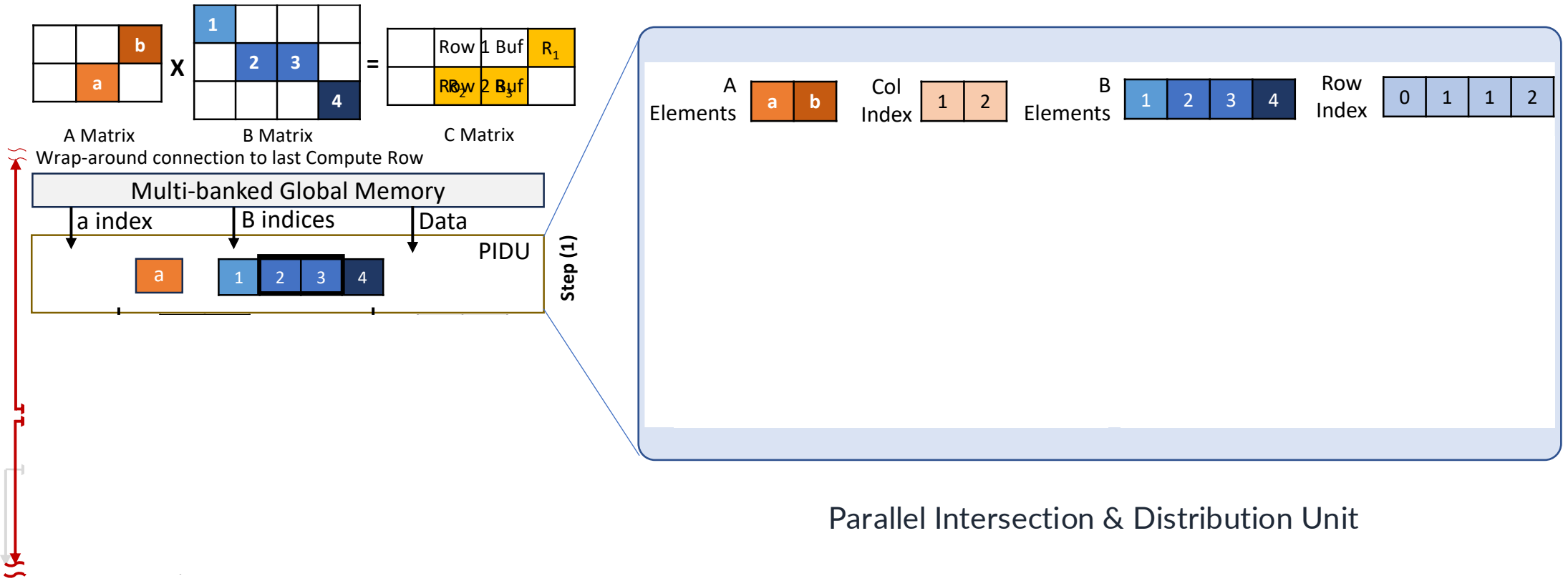
HiT: HSparse Dataflow

- Increase the matching rate by intersecting multiple columns of Matrix A with multiple rows of Matrix B.
- Supports highly sparse workloads: HS x HS, HS x MS, HS x D.



- Column-to-row intersection always guarantees a match
- Regular access of B Matrix

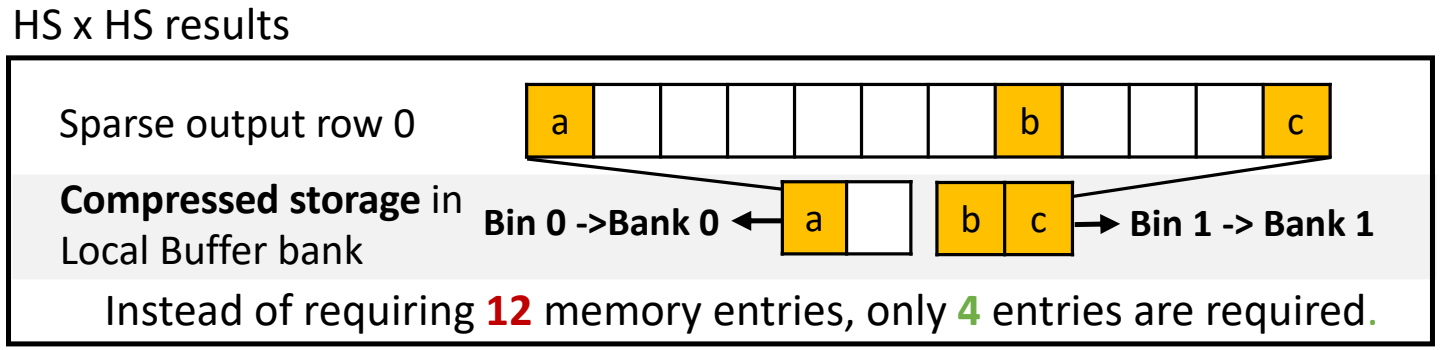
HiT: HSparse Execution in Compute Row



Parallel Intersection & Distribution Unit

HiT: HSparse On-chip Accumulation

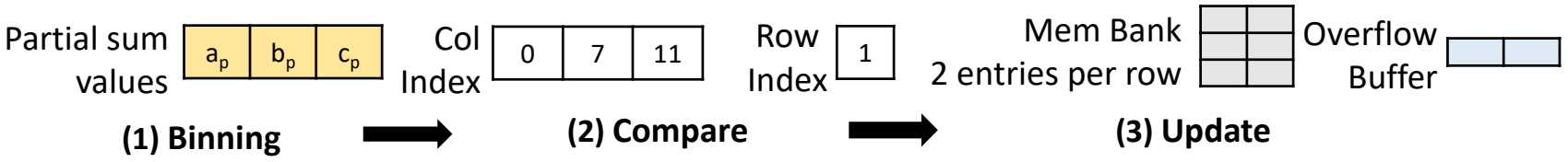
- In HS x HS computations, output is HS.
- To increase buffer utilization & to work on a larger input tile size, we introduce Dual-mode Accumulator



BUT, how do we update the partial sums ?

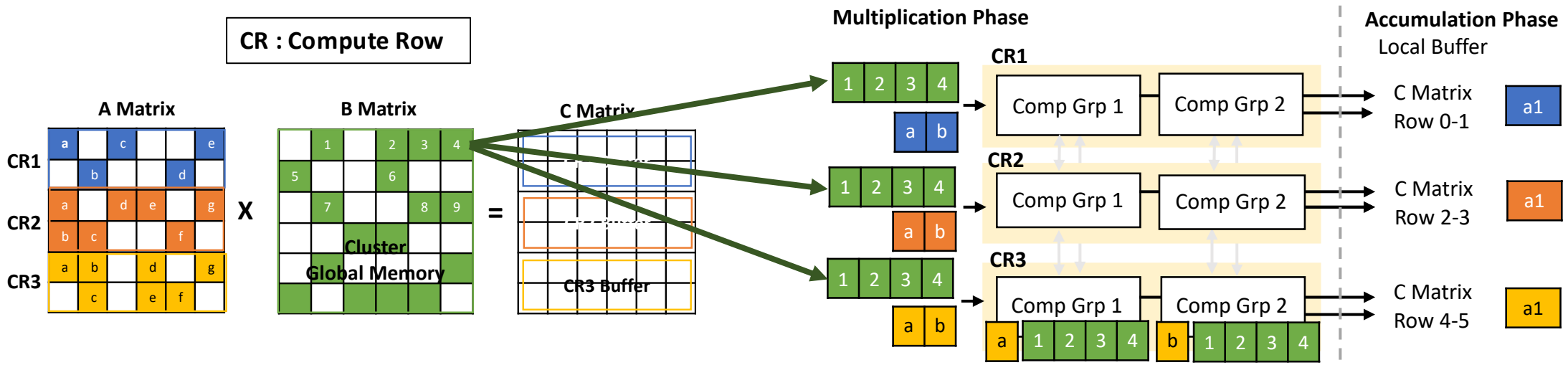
HiT: HSparse On-chip Accumulation

- Binning-compare-update procedure



HiT: MSparse Dataflow

- Broadcast Matrix B data to all Computer Rows in the same Cluster, Ring Network **disabled**.
- Supports moderately sparse workloads: MS x MS & MS x D.

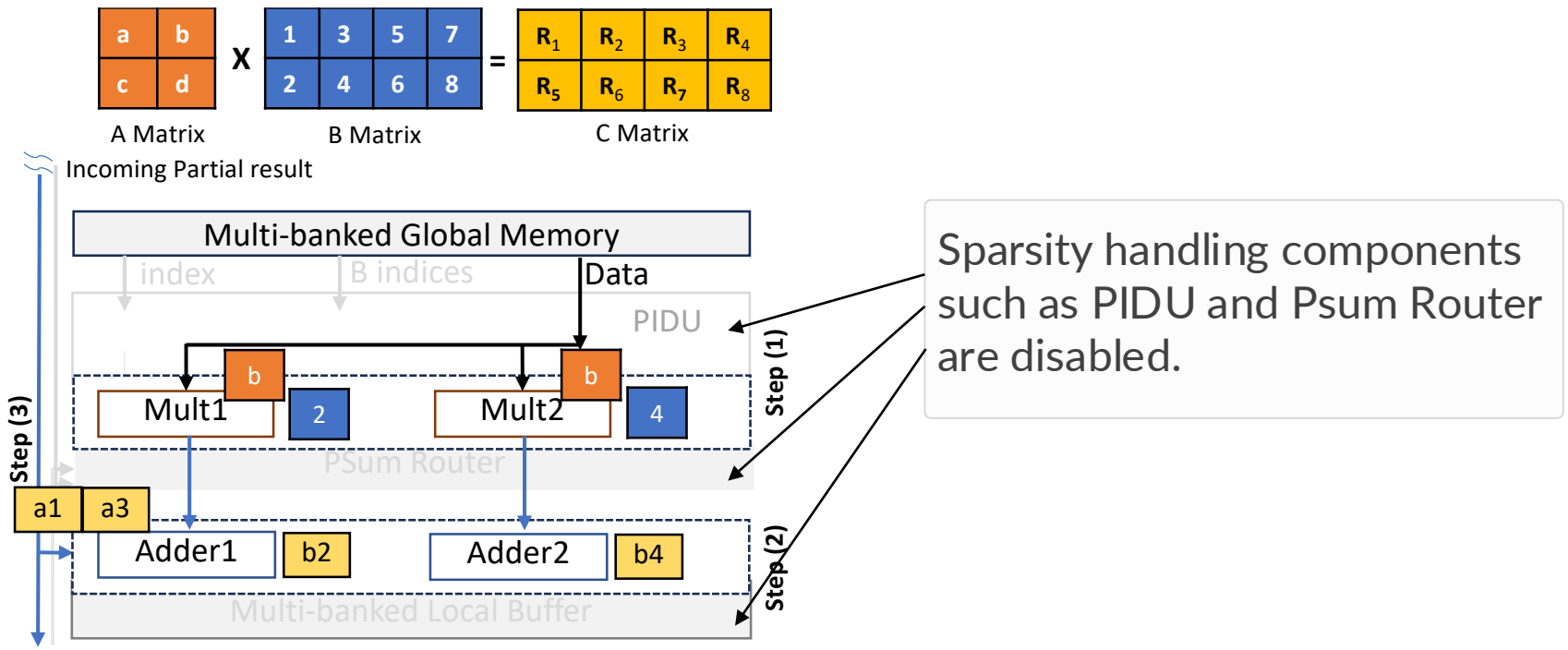


1. Workloads Processed in parallel. Same data from B matrix is send to all CRs.

2. Partial sums accumulated in the same Buffer as CR.

HiT: Inner-product Dataflow

- Operates as a systolic array.
- Supports D x D.



Evaluation: Setup

- Synthesized with 22-nm technology node @ 1GHz.
- Power analysis is performed with Synopsys Prime Power.

TABLE I: Dataset Details

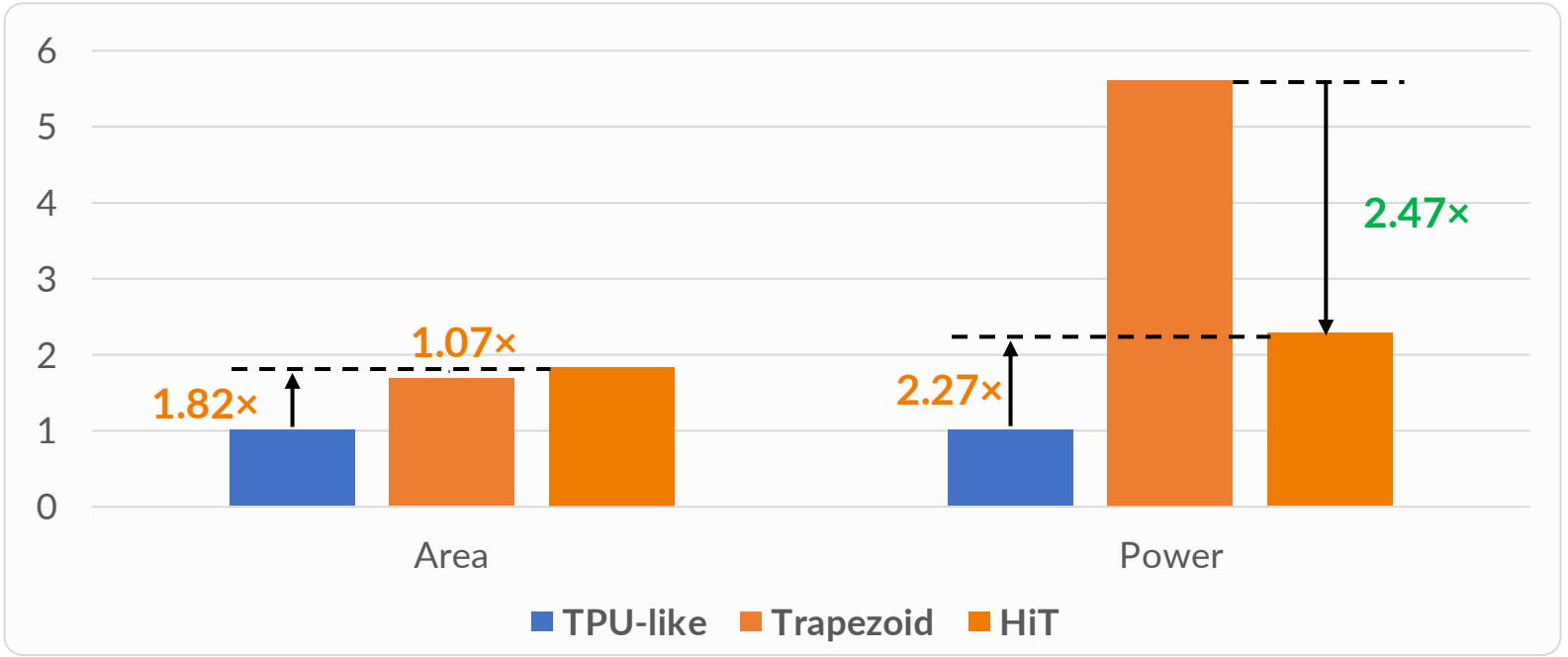
	Name	Dimension		Density	
HS	p2p-Gnutella24	26518 x 26518		9.3e-5	
	ca-CondMat	23133 x 23133		3.5e-4	
	opt1	15449 x15449		8.1e-3	
	cage12	130228 x 130228		1.2e-4	
	poisson3Da	13514 x 13514		1.9e-3	
	msc10848	10848 x10848		1.0e-2	
	Name	**Dimension**		**Density**	
		Act.	**W**	**Act.**	**W**
MS	Resnet50-0.4	1024x14x14	1x1 , 256	0.46	0.37
		512x28x28	1x1, 128	0.65	0.52
	Vgg16-0.4	512x28x28	3x3 , 512	0.23	0.38
MS / D	Llama2-7b	1024x11008	11008x4096	1	0.20 / 1
		1024x4096	4096 x11008	1	0.40 / 1
		1024x11008	11008x4096	1	0.60 / 1

TABLE II: HiT Design Specification

Hardware Component		Specs
Compute Row	Multiplier	4 x 32, FP32
	PIDU	4 x 64 comparator & shift registers
	PSum Router	4 x 6 ports (3 inputs & 3 outputs) ring buffer size: 6 (up/down incoming), 4 (multiplier in), 6 (out to DMAccum)
	DMAccum	4 x (32 FP32 adders & 512 comparators)
	Local Buffer	11.4KB, 8 banks, register file
	Compute Cluster	32 x Compute row
	Global Memory	16MB, 512 banks SRAM, 64B datawidth
	Overall	4 x Compute cluster @ 1 GHz 17.4MB memory, HBM @ 2TB/s

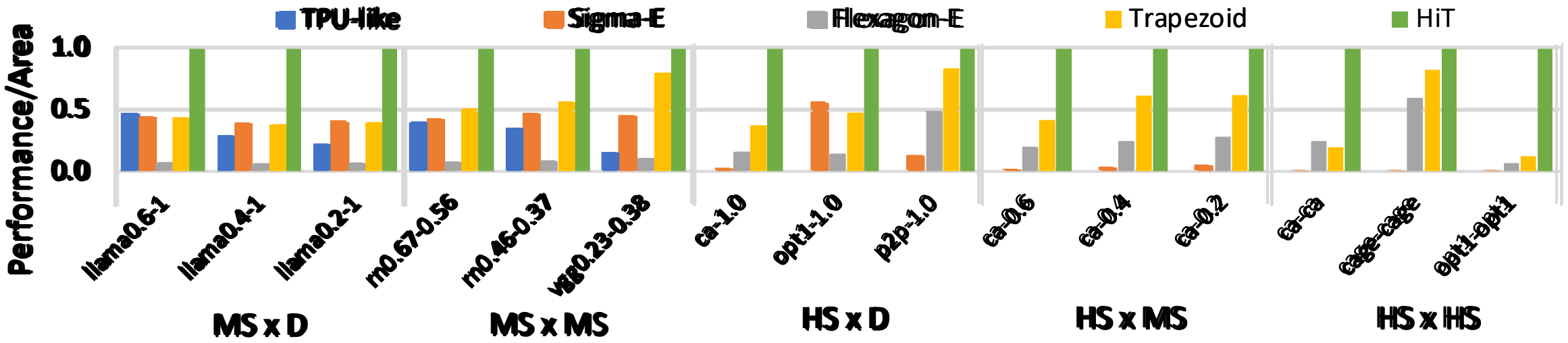
Evaluation: Area and Power Results

	TPU-like	Trapezoid		HiT
	22nm	16/15nm	22nm*	22nm
Area (mm^2)	91.4	81.9 (16nm)	154.8	**166.3**
Power (W)	49.9	191 (15nm)	280.1	**113.5**



*estimated result

Evaluation: Representative Dataset Geomean Area Efficiency



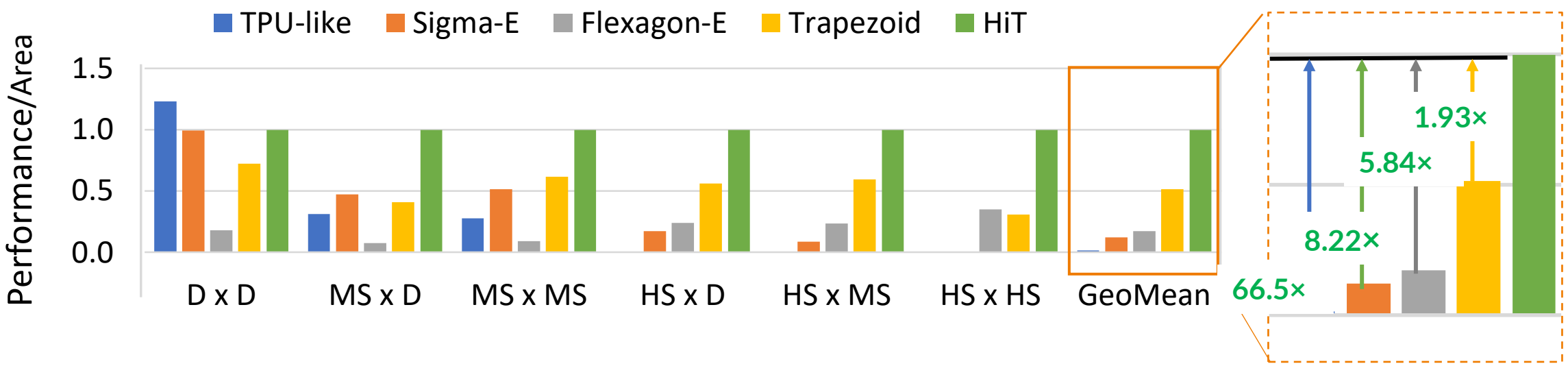
MS workloads

1.99× to 11.9× over MS specialized baselines, Sigma-E, Flexagon-E, and Trapezoid.

HS workloads

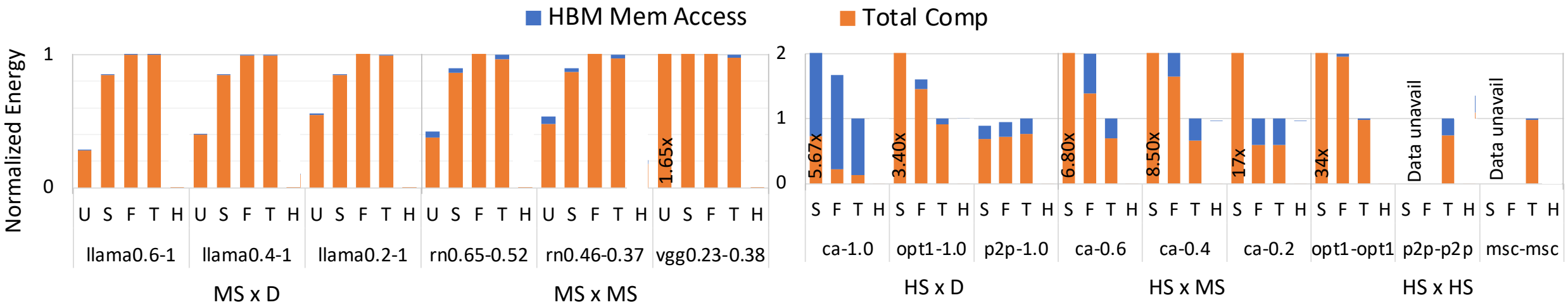
2.33× to 46.9×, over Sigma-E, Flexagon-E & Trapezoid.

Evaluation: Overall Geomean Area Efficiency

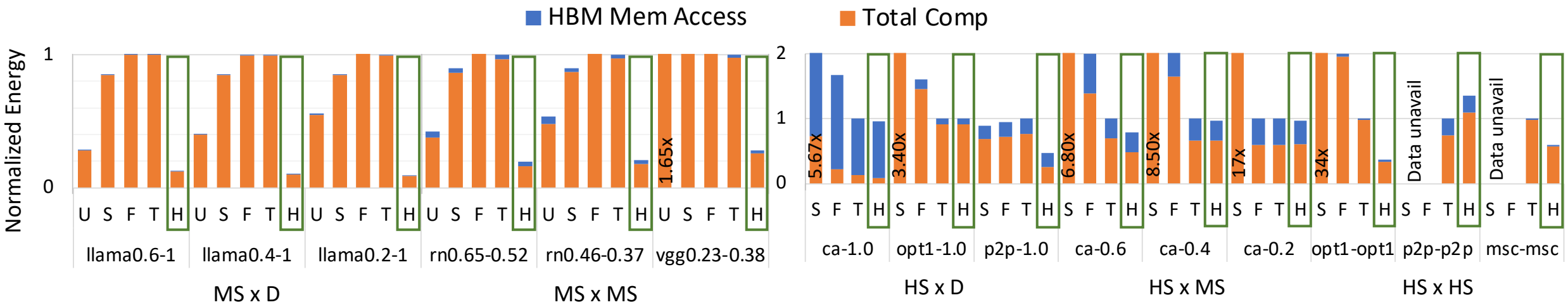


- HiT operates as a systolic array for D×D workloads, its larger area leads to **1.23×** lower performance/area than TPU-like.

Evaluation: Representative Dataset Geomean Energy Consumption



Evaluation: Representative Dataset Geomean Energy Consumption



MS workloads

Reduces energy by **3.01× - 5.79×** over TPU-like, Sigma-E, Flexagon-E and Trapezoid.

HS workloads

1.27× - 8.86× lower energy than Sigma-E and Flexagon-E, and Trapezoid.

Thank You! 😊

Questions & Answers

HiT: A Unified Sparsity-Adaptive Architecture for High-Throughput Matrix Multiplication

Novel Dataflows

HSparse and **Msparse** that deliver higher MAC utilization, enable regular memory access, and allow on-chip accumulation of partial sums.

Output Sparsity Exploitation

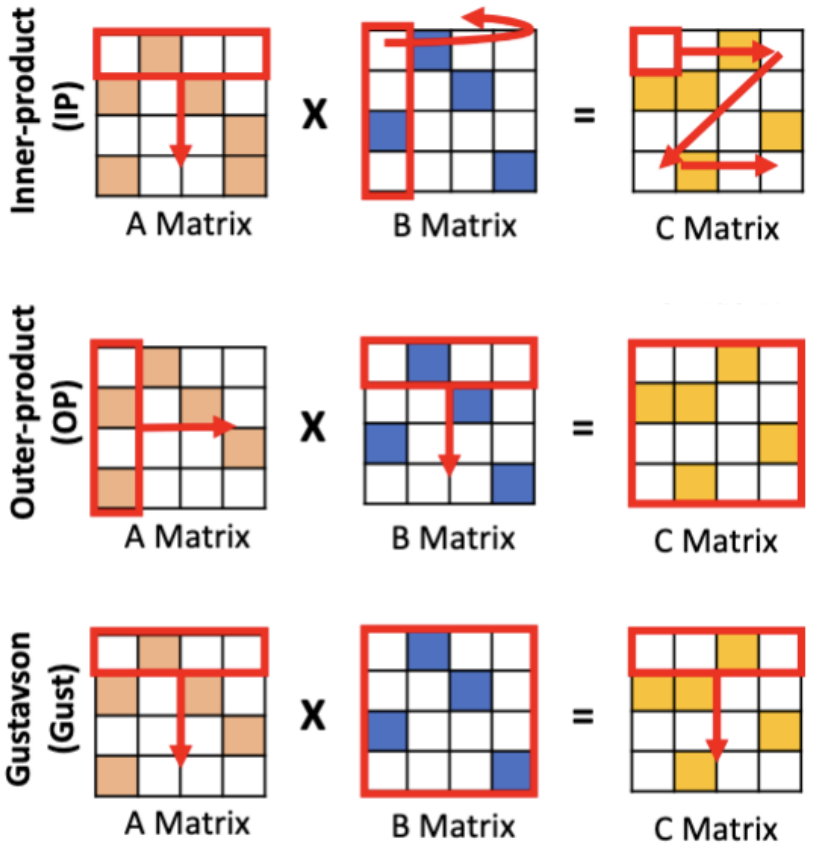
Increases buffer utilization and enables processing of larger tiles with more non-zero elements, improving compute density and throughput.

Unified Architecture

Single architecture that supports our novel dataflows and inner-product with novel **Parallel Intersection & Distribution Unit** and **DMAccum**.

1.93× - 66.5× higher performance/area and **1.27× - 8.86×** lower energy than prior works.

Conventional Dataflows



Inner-product

Computes one element of C at a time, multiplying a row of A with a column of B

Outer-product

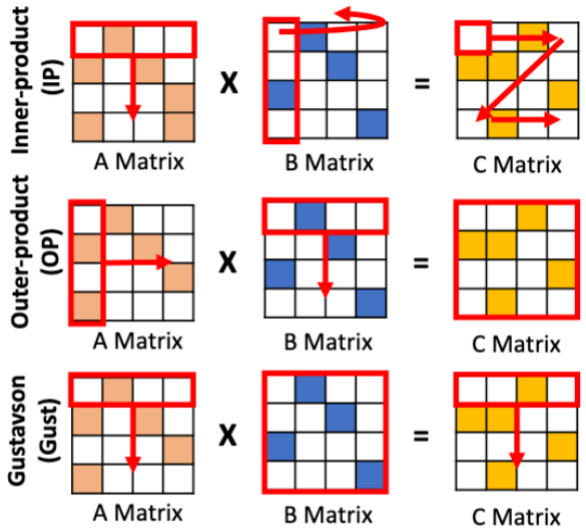
Multiplies each element of a column of A with the corresponding row of B, updating a row of C

Gustavson

Computes an entire row of partial outputs at a time, with each element of A

Conventional Dataflow

Dataflow selection is **critical**. It directly determines **memory access patterns**, **parallelism** & **intermediate data movement**.



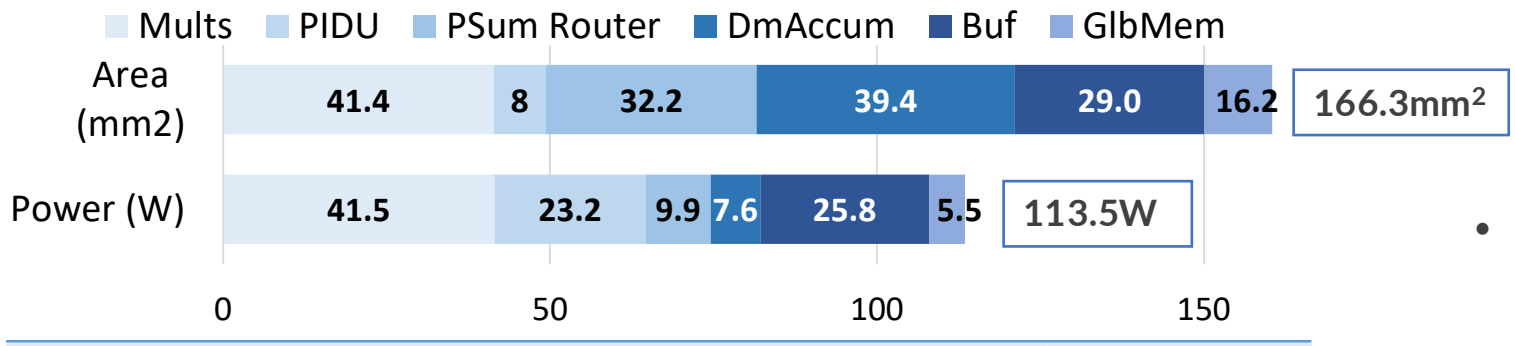
Inner-product: implemented in dense accelerators (TPU)

Outer-product: implemented in MS and HS

Gustavson: implemented in MS and HS, a common choice in HS

		IP	OP	Gust
Data reuse	A	High	High	High
	B	Low	High	Moderate
	C	High	Low	Moderate
Ineffectual intersection		Yes	No	No

Power and Area Breakdown



	TPU-like	Trapezoid		**HiT**
	22nm	16/15nm	22nm*	22nm
Area (mm^2)	91.4	81.9 (16nm)	154.8	166.3
Power (W)	49.92	191 (15nm)	280.1	**113.5**

- Mults & DMAccum results in 50% of the total area
- Mults and Local Buffer consumes 59% of the total power

Evaluation: Effective MAC Utilization

- Percentage of effective MACs / (total number of MACs x Latency)
- 2.35× higher utilization in HS workloads compared with Trapezoid

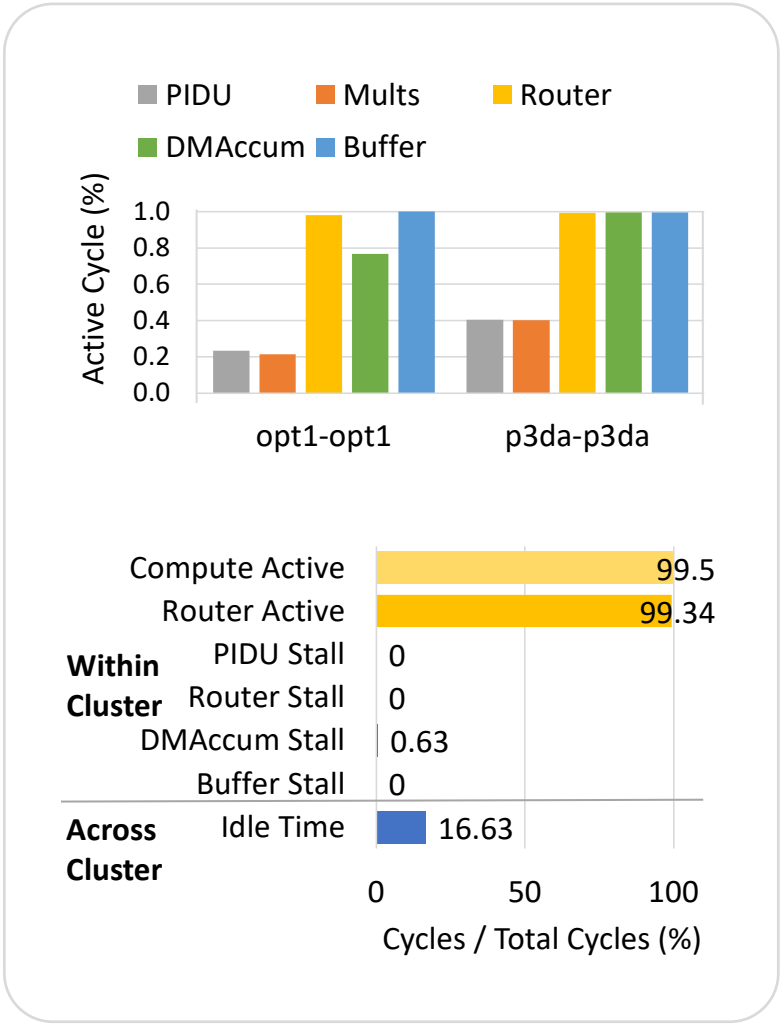
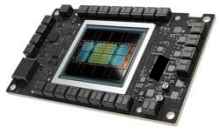


IMAGE SOURCES

https://www.clipartmax.com/middle/m2i8G6K9d3d3b1d3_road-map-png/

Source: www.clipartmax.com/



https://www.datacenterknowledge.com/data-center-chips/data-center-chips-in-2024-top-trends-and-releases

Source: www.datacenterknowledge.com