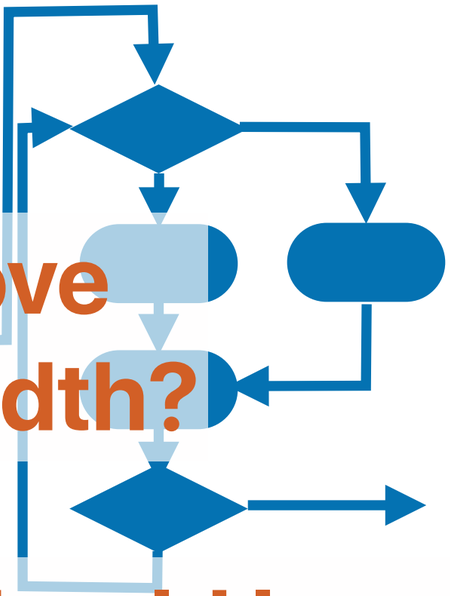
Hardware accelerators

Hung-Wei Tseng

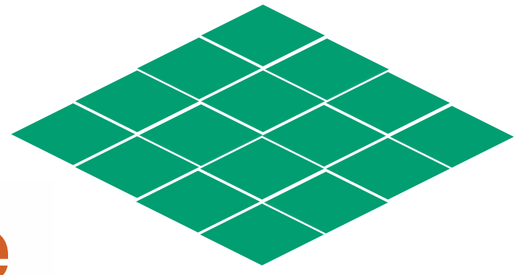
Recap: from the software perspective

Should I improve memory bandwidth?

Should I use NDP?

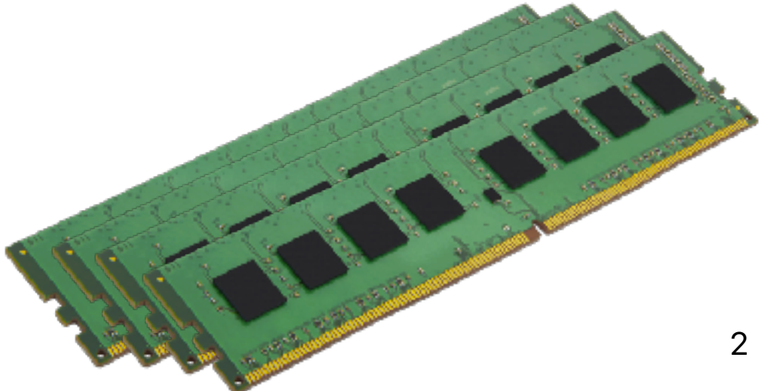


Should I use accelerators?



Source “data"

“Data” structures



Algorithms

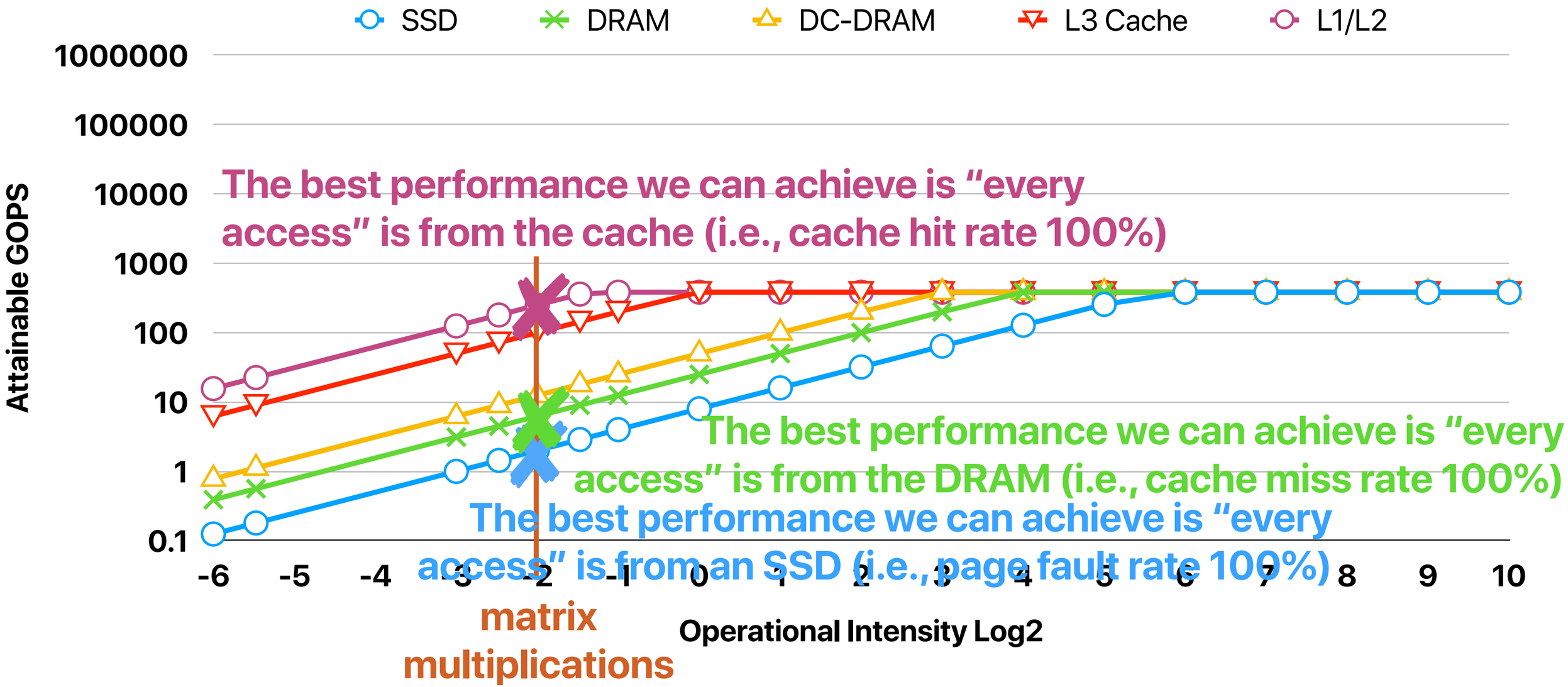
Result

Recap: the roofline model

- The speed of computation — determined by the "OPS" (operations per second) of the processing unit
- The speed of data supply — determined by the "effective bandwidth" of the data path
- At any point, the slower one determines the performance
  - $$\text{Attainable GFlops/sec} = min \left\{ \begin{array}{l} \text{Peak Floating-Point Performance} \\ \text{Peak Memory Bandwidth} \times \text{Operational Intensity} \end{array} \right.$$

Samuel Williams, Andrew Waterman, and David Patterson. 2009. Roofline: an insightful visual performance model for multicore architectures. Commun. ACM 52, 4 (April 2009), 65–76.

Recap: the roofline of a CPU-based system



Outline

- Raise the roof
  - GPU
  - Tensor processing units
  - Tensor cores and ray tracing units
  - FPGAs

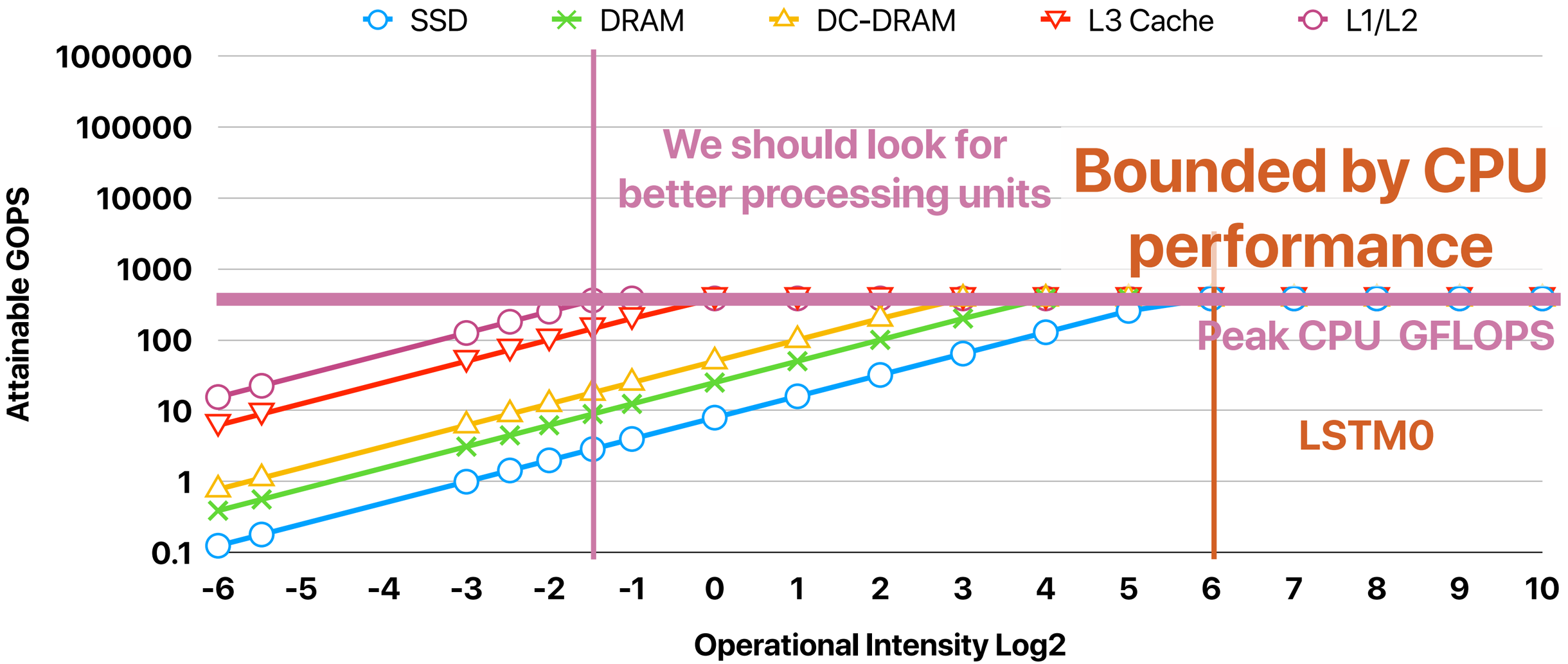
What's the operational intensity of modern workloads?

Table 1. Six DNN applications (two per DNN type). MLP stands for multi-layer perceptron, and LSTM stands for long short-term memory.

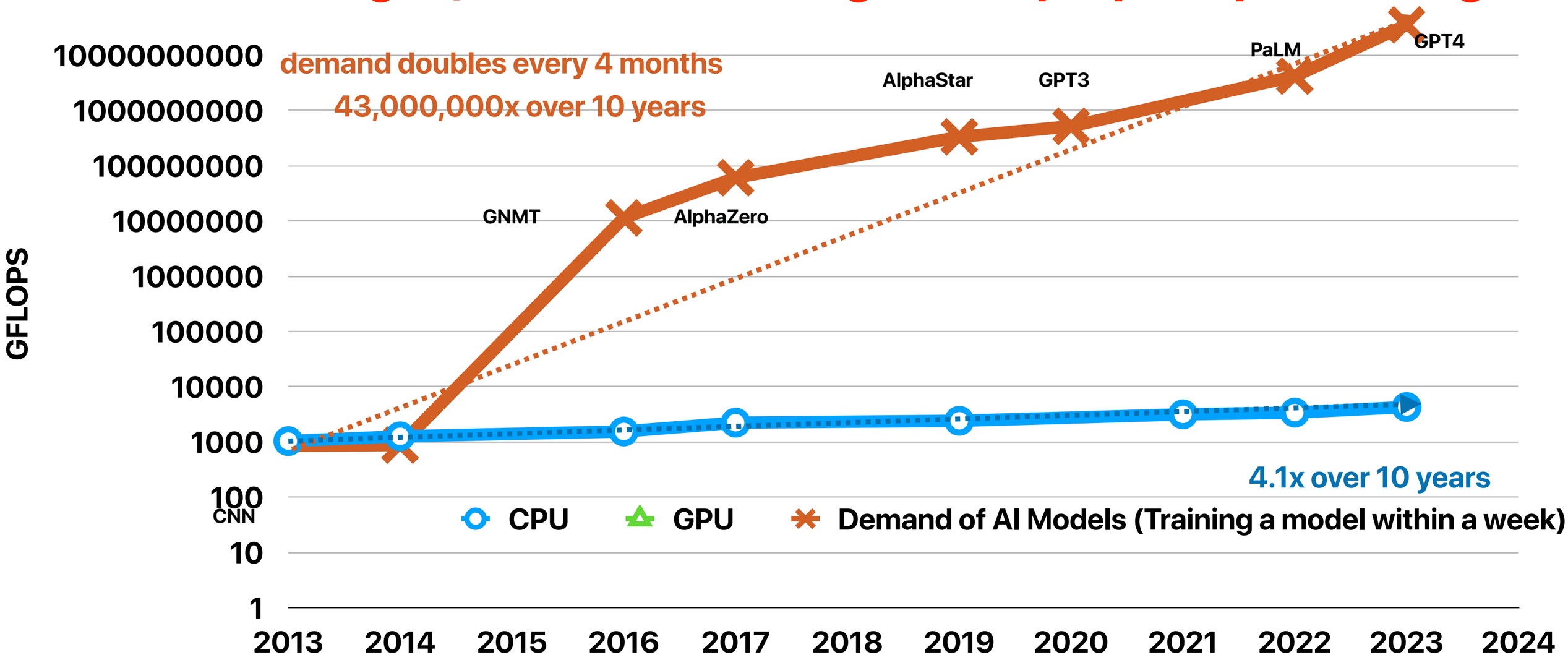
Name	Lines of Code	Weights	TPU Ops / Weight Byte	Operational Intensity Log2
MLP0	100	20 M	200	7.64385618977472
MLP1	1,000	5 M	168	7.39231742277876
LSTM0	1,000	52 M	64	6
LSTM1	1,500	34 M	96	6.58496250072116
CNN0	1,000	8 M	2,888	11.4958550268872
CNN1	1,000	100 M	1,750	10.7731392067197

N. Jouppi, C. Young, N. Patil and D. Patterson, "Motivation for and Evaluation of the First Tensor Processing Unit," in *IEEE Micro*, vol. 38, no. 3, pp. 10-19, May./Jun. 2018, doi: 10.1109/MM.2018.032271057.

Where is LSTM0 in the roofline?



Mis-matching AI/ML demand and general-purpose processing



https://ourworldindata.org/grapher/artificial-intelligence-training-computation

Dennardian Broken

- Given a scaling factor S

Parameter	Relation	Classical Scaling	Leakage Limited
Power Budget		1	1
Chip Size		1	1
Vdd (Supply Voltage)		1/S	1
Vt (Threshold Voltage)	1/S	1/S	1
tex (oxide thickness)		1/S	1/S
W, L (transistor dimensions)		1/S	1/S
Cgate (gate capacitance)	WL/tox	1/S	1/S
Isat (saturation current)	WVdd/tox	1/S	1
F (device frequency)	Isat/(CgateVdd)	S	S
D (Device/Area)	1/(WL)	S^2	S^2
p (device power)	IsatVdd	$1/S^2$	1
P (chip power)	Dp	1	S^2
U (utilization)	1/P	1	$1/S^2$

Recap: Power consumption to light on all transistors

Chip

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

=49W

Dennardian Scaling

Chip

0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

=50W

Dennardian Broken

Chip

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

On ~ 50W

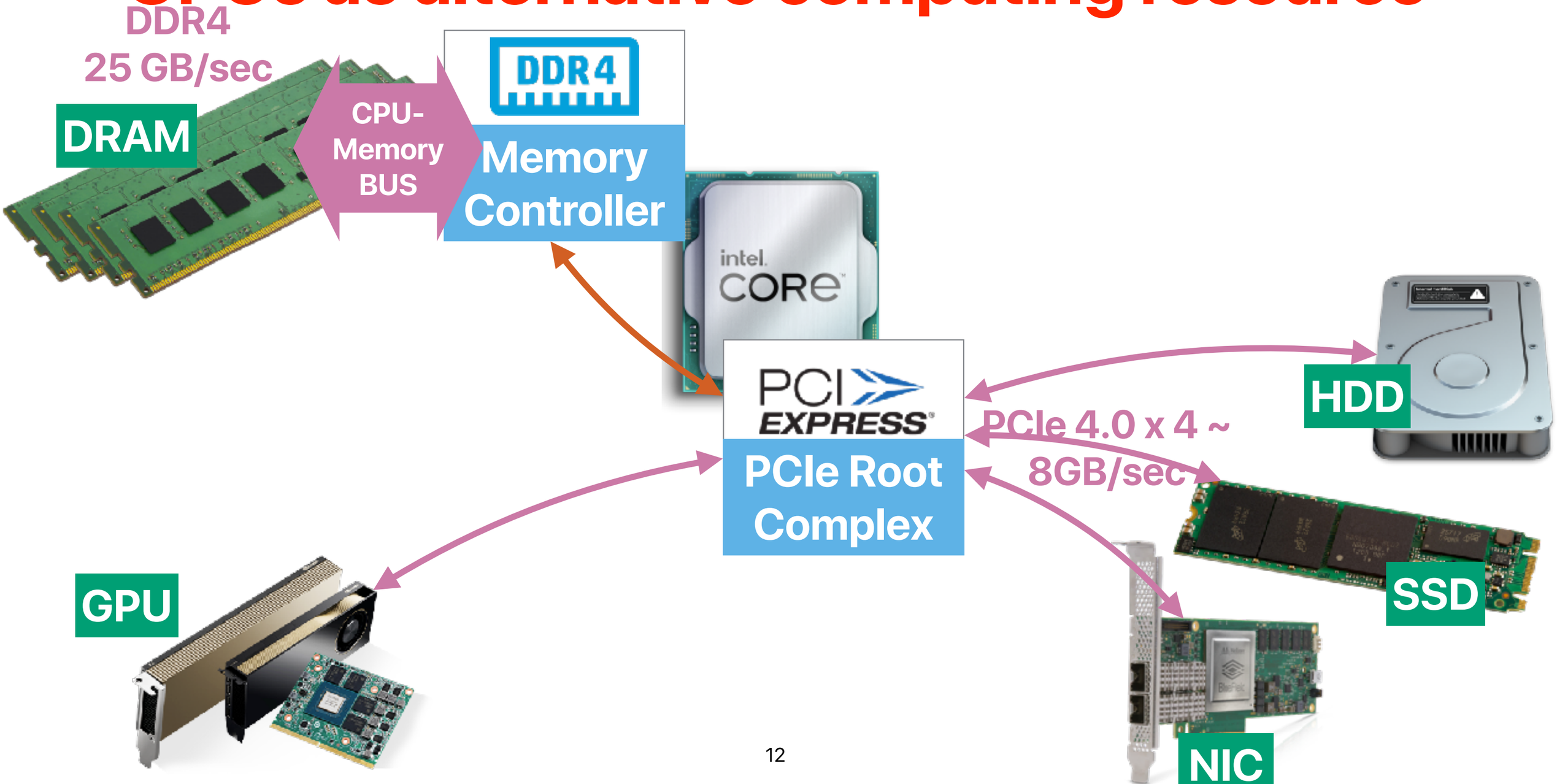
Off ~ 0W

Dark!

=100W!

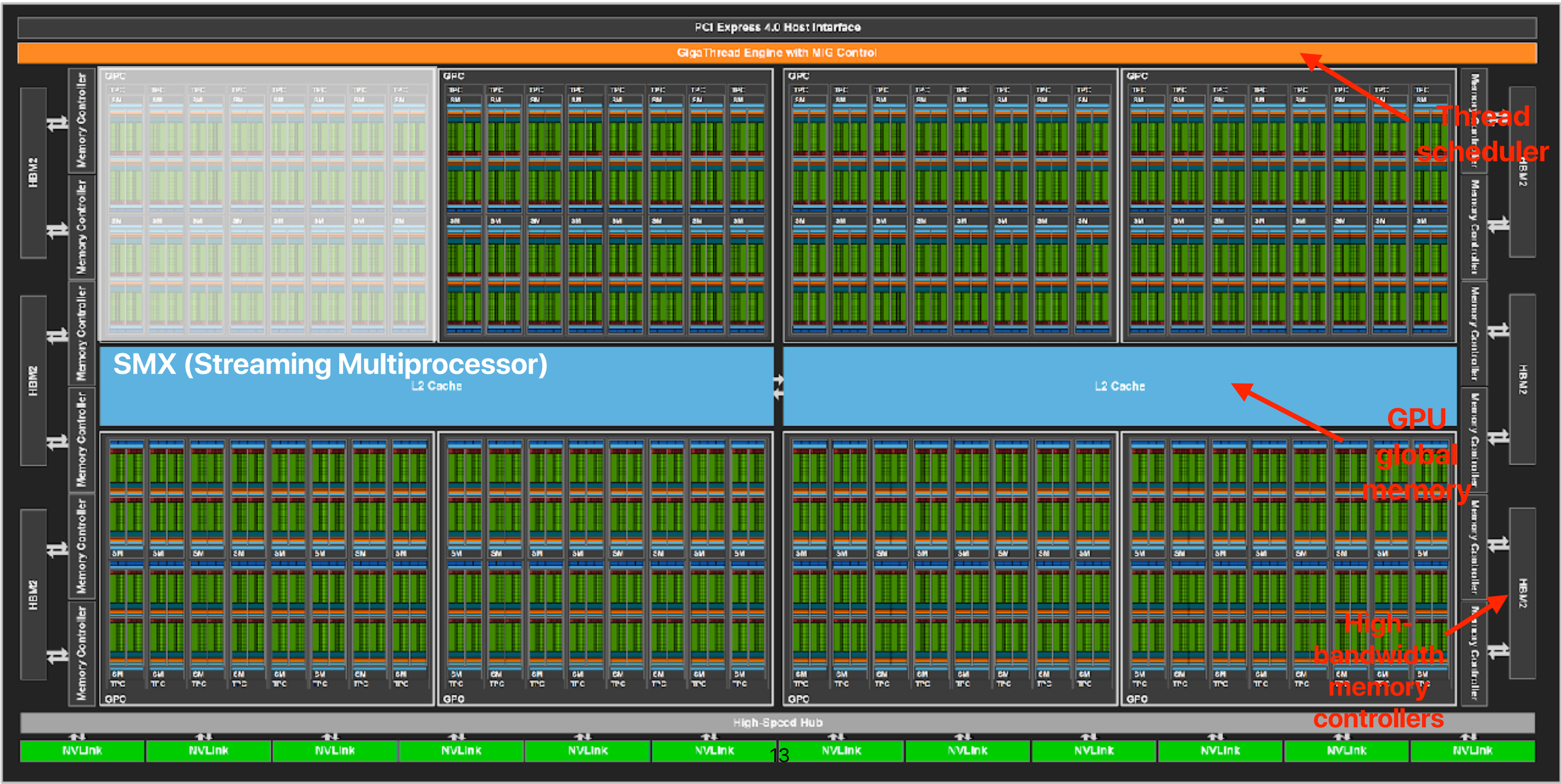
Alternative computing resources

GPUs as alternative computing resource



https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/

GPU Architecture



Inside an SM

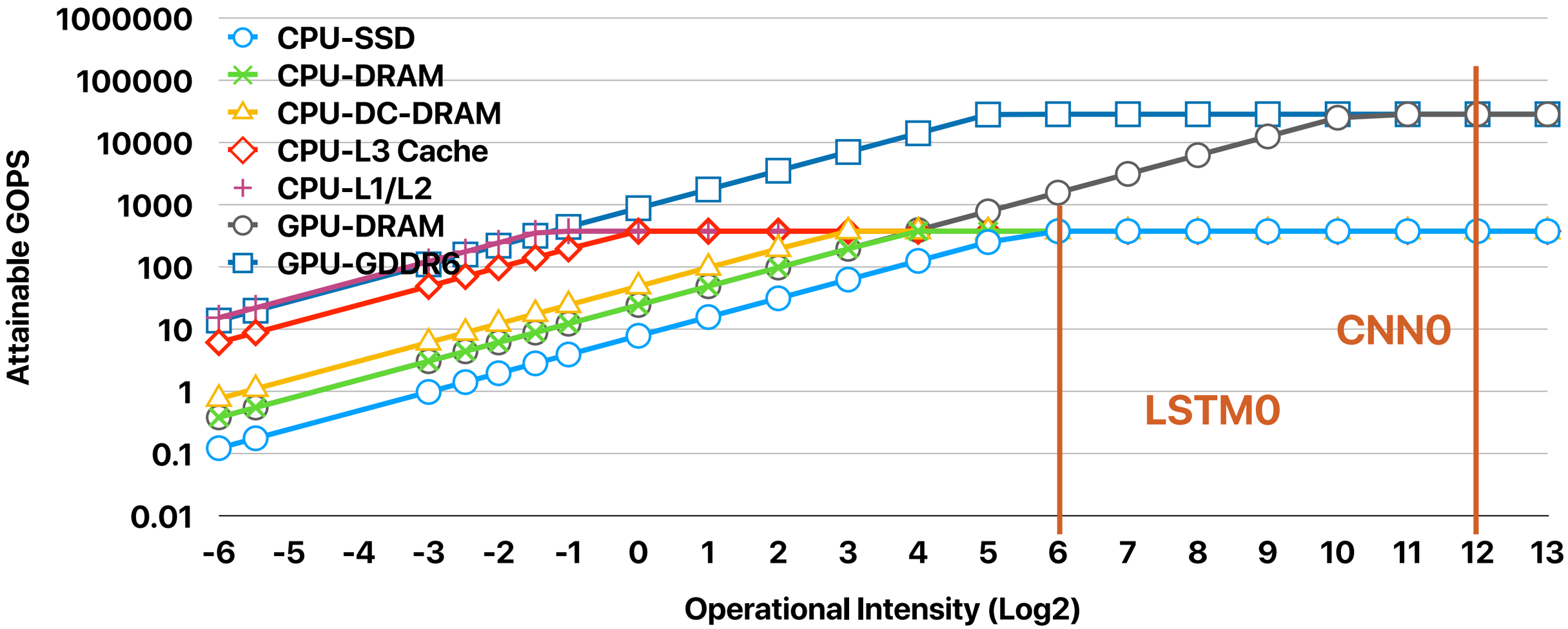
A total of 16*4 = 64 FP32 cores
A total of 16*4 = 64 INT32 cores
A total of 16*4 = 16 FP64 cores

- All of these can only perform the same operation at the same time, but each of these is named as a "thread" in CUDA
- You can only use either FP32, FP64, INT32 and "Tensor Cores" at the same time

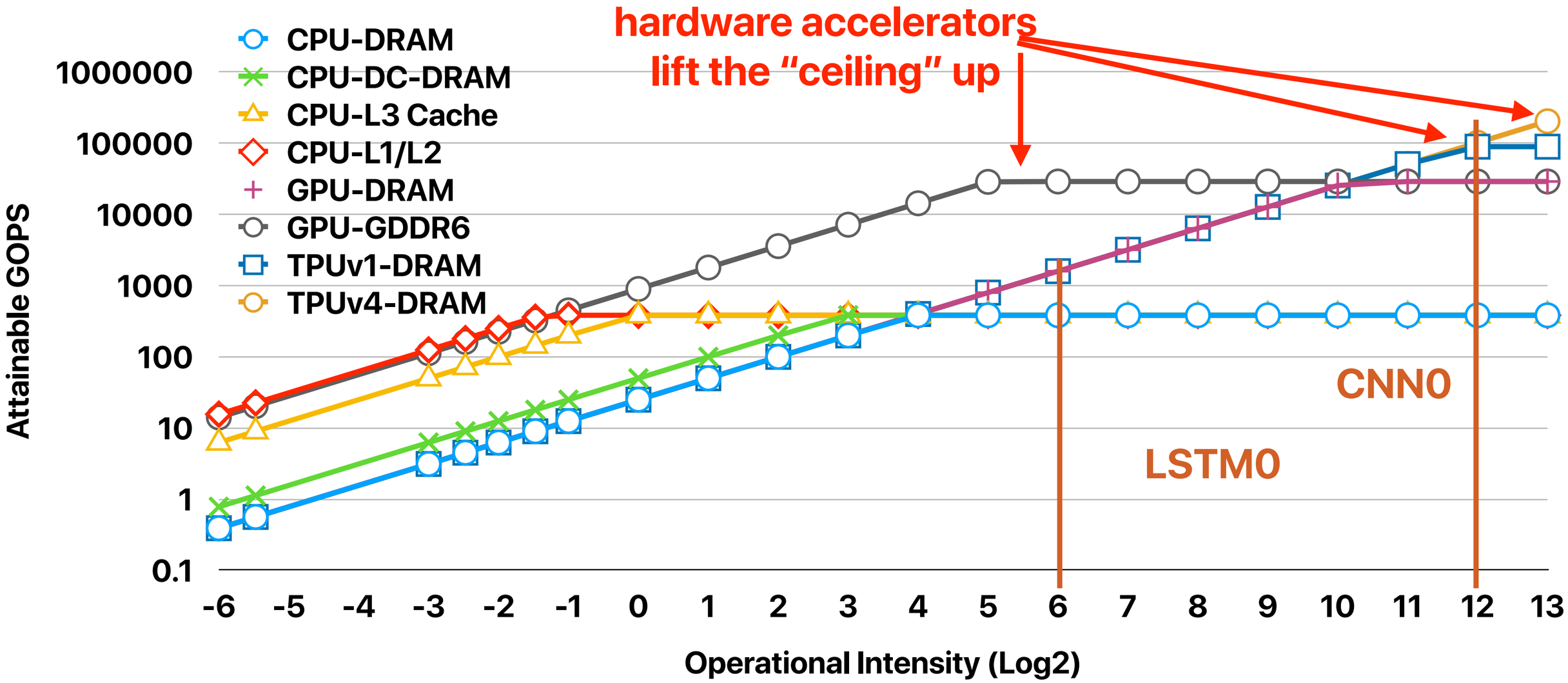
3:00

How do GPUs change the roofline model?

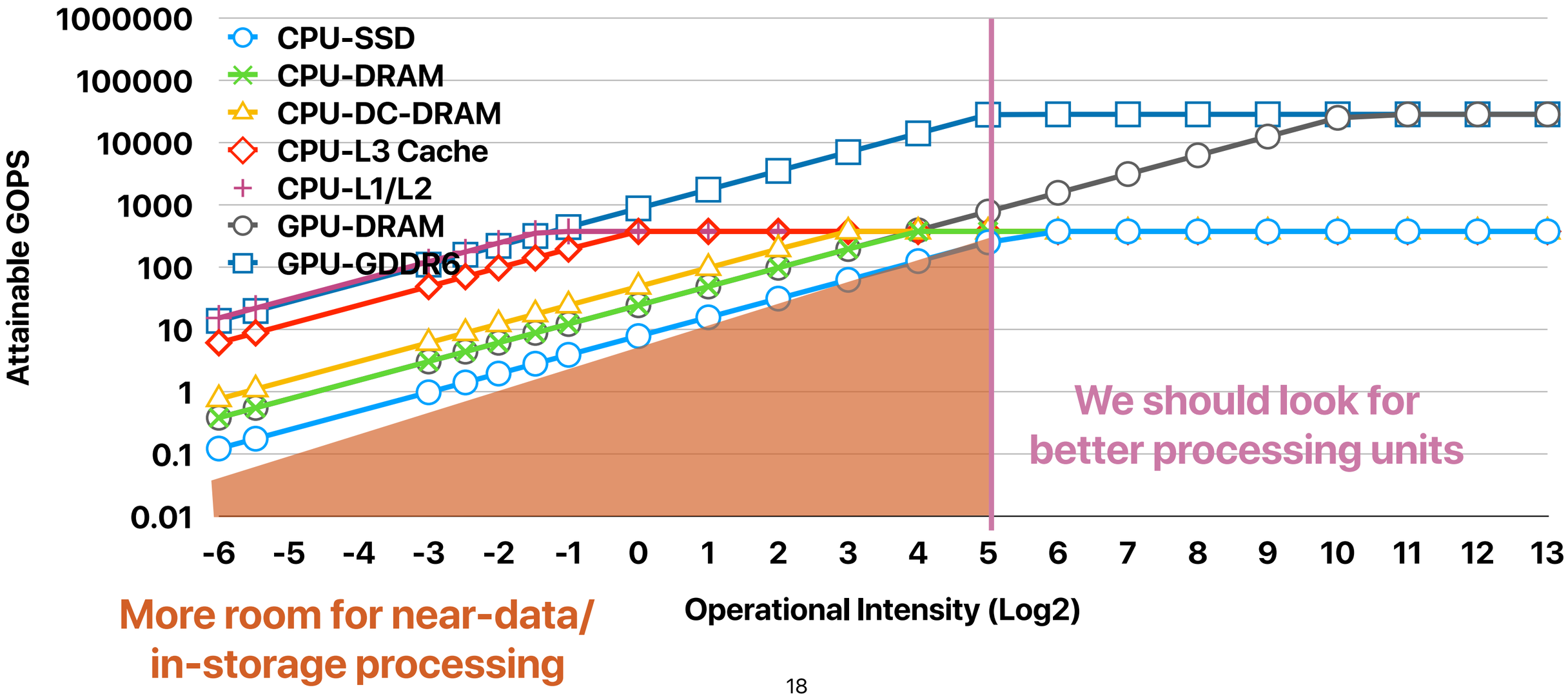
Placing GPUs in the roofline



Recap: the rooflines of modern systems



Placing GPUs in the roofline



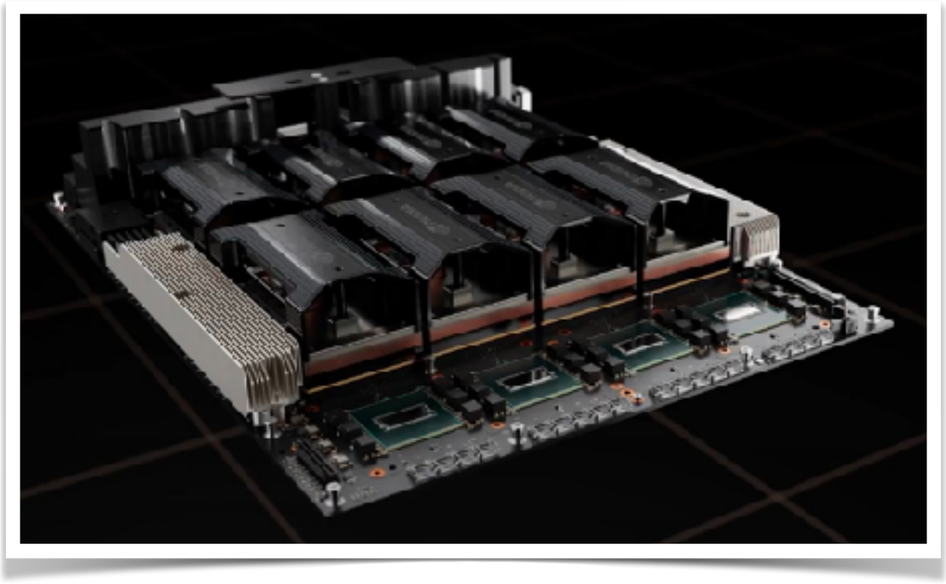
3:00

Why don’t we just put everything CPUs can’t do on GPUs?

Power/energy is the main issue

NVIDIA Accelerator Specification Comparison			
	H100	A100 (80GB)	V100
FP32 CUDA Cores	16896	6912	5120
Tensor Cores	528	432	640
Boost Clock	~1.78GHz (Not Finalized)	1.41GHz	1.53GHz
Memory Clock	4.8Gbps HBM3	3.2Gbps HBM2e	1.75Gbps HBM2
Memory Bus Width	5120-bit	5120-bit	4096-bit
Memory Bandwidth	3TB/sec	2TB/sec	900GB/sec
VRAM	80GB	80GB	16GB/32GB
FP32 Vector	60 TFLOPS	19.5 TFLOPS	15.7 TFLOPS
FP64 Vector	30 TFLOPS	9.7 TFLOPS (1/2 FP32 rate)	7.8 TFLOPS (1/2 FP32 rate)
INT8 Tensor	2000 TOPS	624 TOPS	N/A
FP16 Tensor	1000 TFLOPS	312 TFLOPS	125 TFLOPS
TF32 Tensor	500 TFLOPS	156 TFLOPS	N/A
FP64 Tensor	60 TFLOPS	19.5 TFLOPS	N/A
Interconnect	NVLink 4 18 Links (900GB/sec)	NVLink 3 12 Links (600GB/sec)	NVLink 2 6 Links (300GB/sec)
GPU	GH100 (814mm2)	GA100 (826mm2)	GV100 (815mm2)
Transistor Count	80B	54.2B	21.1B
TDP	700W	400W	300W/350W
Manufacturing Process	TSMC 4N	TSMC 7N	TSMC 12nm FFN
Interface	SXM5	SXM4	SXM2/SXM3
Architecture	Hopper	Ampere	Volta

https://www.workstationspecialist.com/product/nvidia-tesla-a100/



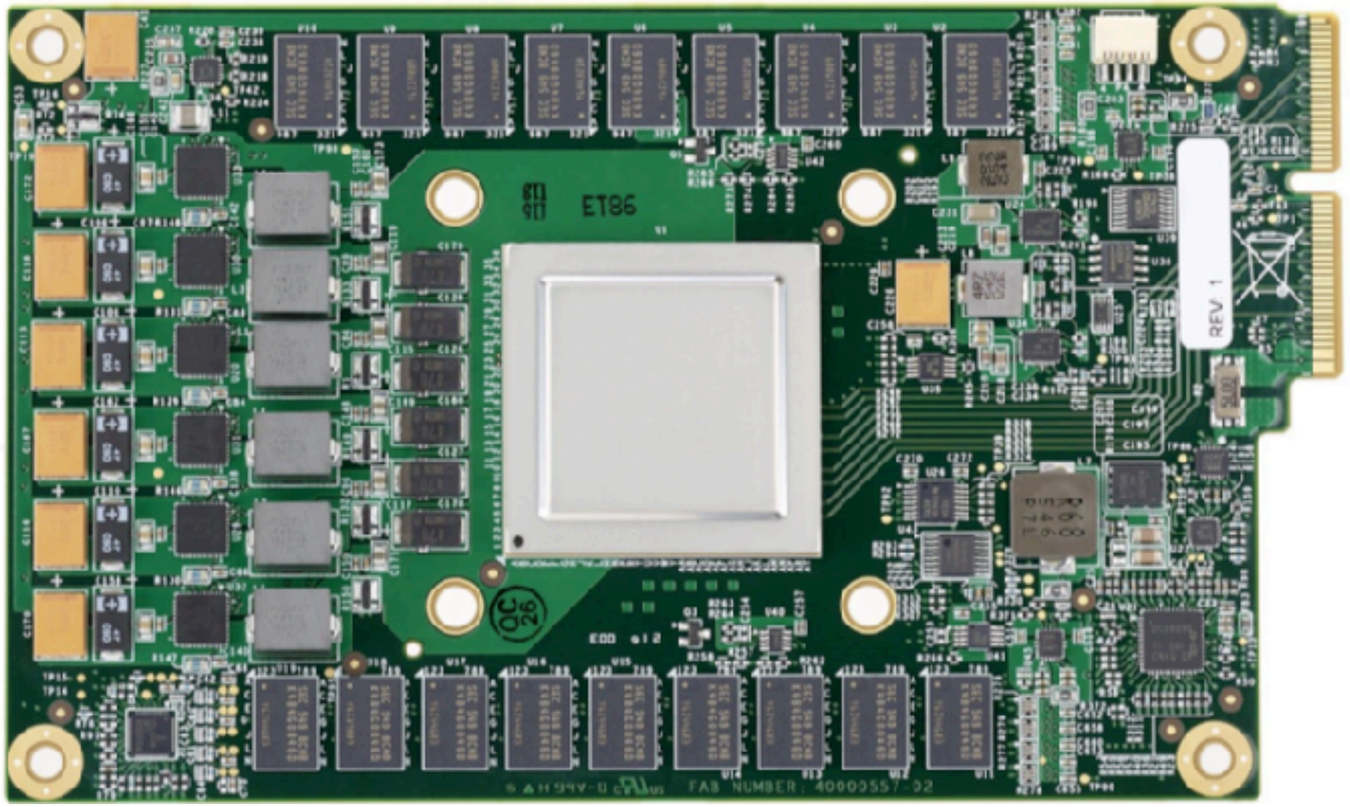
https://www.servethehome.com/wp-content/uploads/2022/03/NVIDIA-GTC-2022-H100-in-HGX-H100.jpg

Can we be fast but also power/ energy efficient?

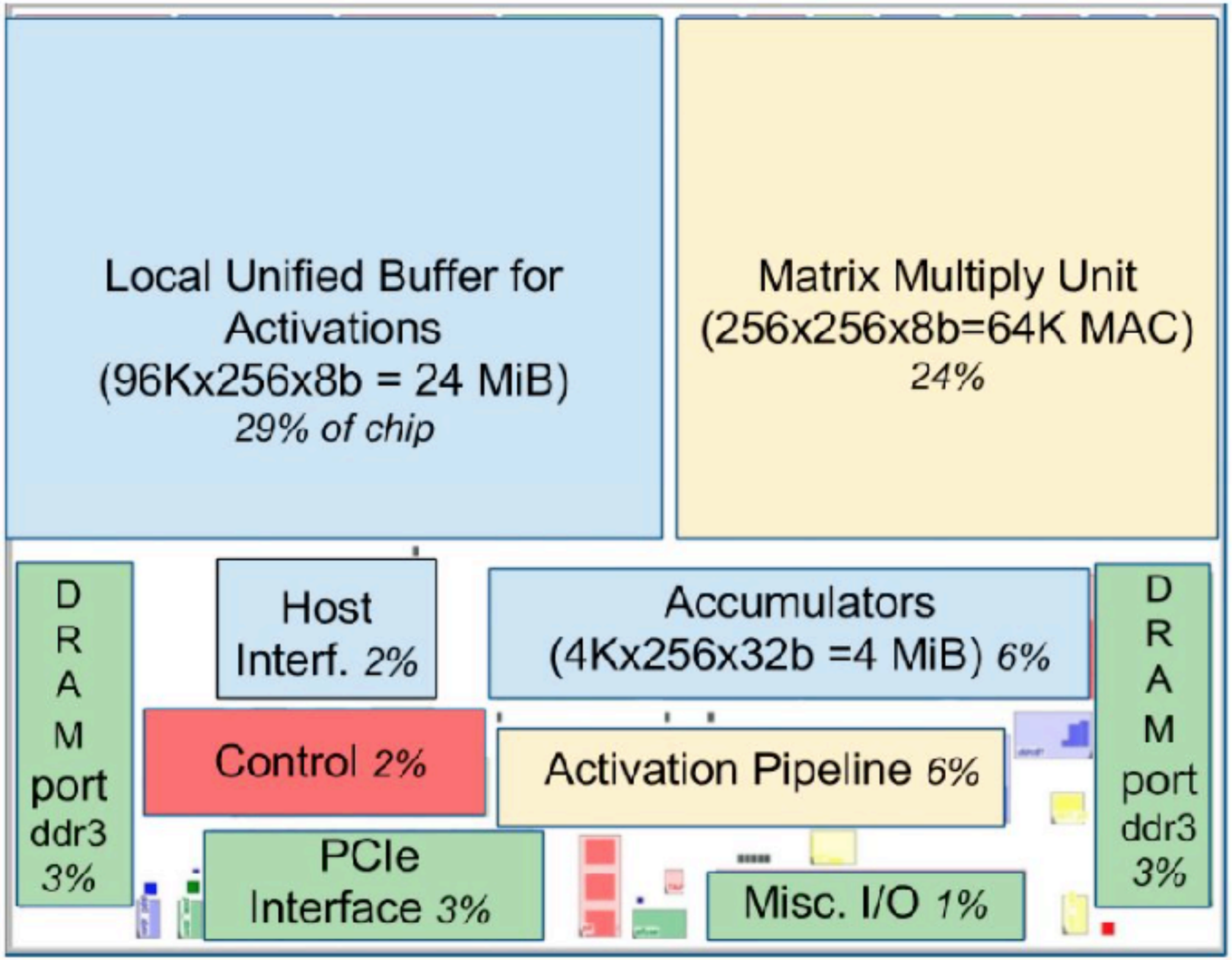
Example of a hardware accelerator — Tensor Processing Unit

- N. Jouppi, C. Young, N. Patil and D. Patterson. Motivation for and Evaluation of the First Tensor Processing Unit. In IEEE Micro, vol. 38, no. 3, pp. 10-19. 2018
- Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon. In-Datacenter Performance Analysis of a Tensor Processing Unit. In ISCA '17. 2017.

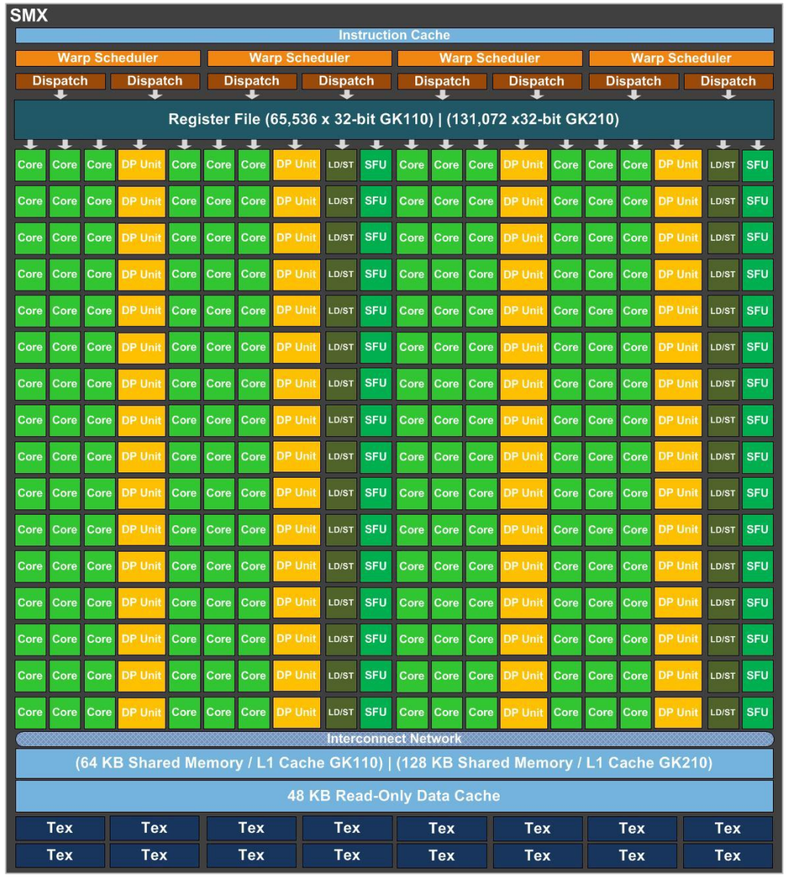
What the “first” generation of TPU looks like

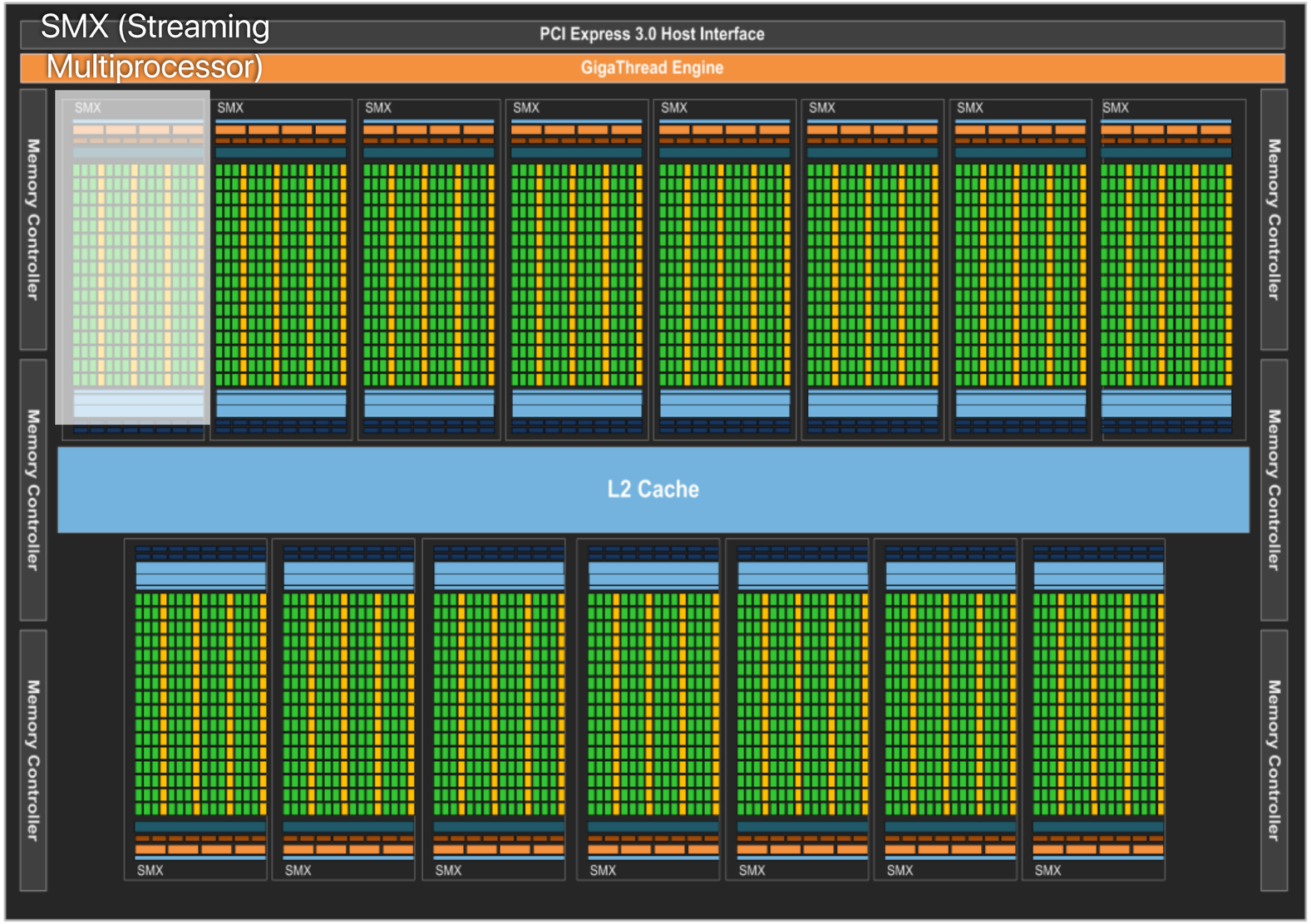


TPU Floorplan



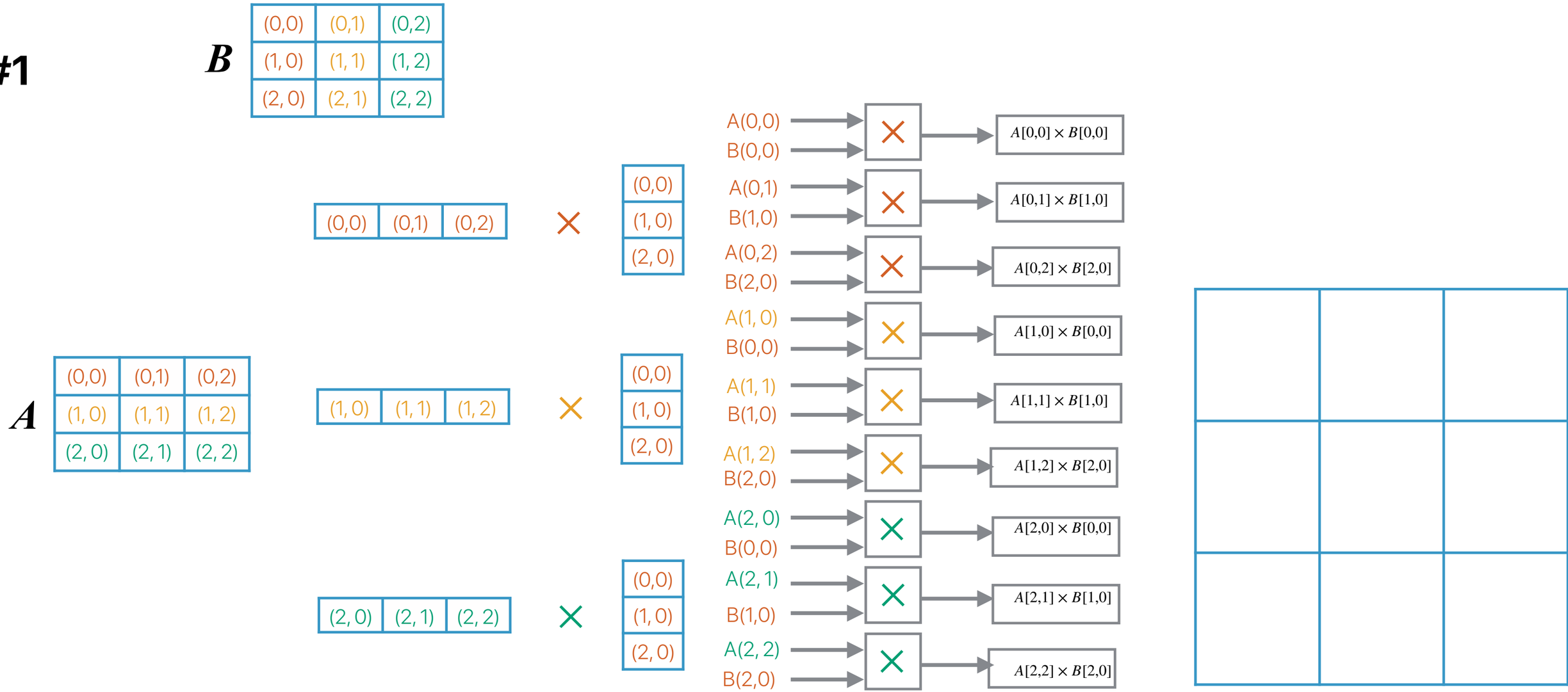
Vector processing model by GPUs





Vector processing for MM

#1



Vector processing for MM

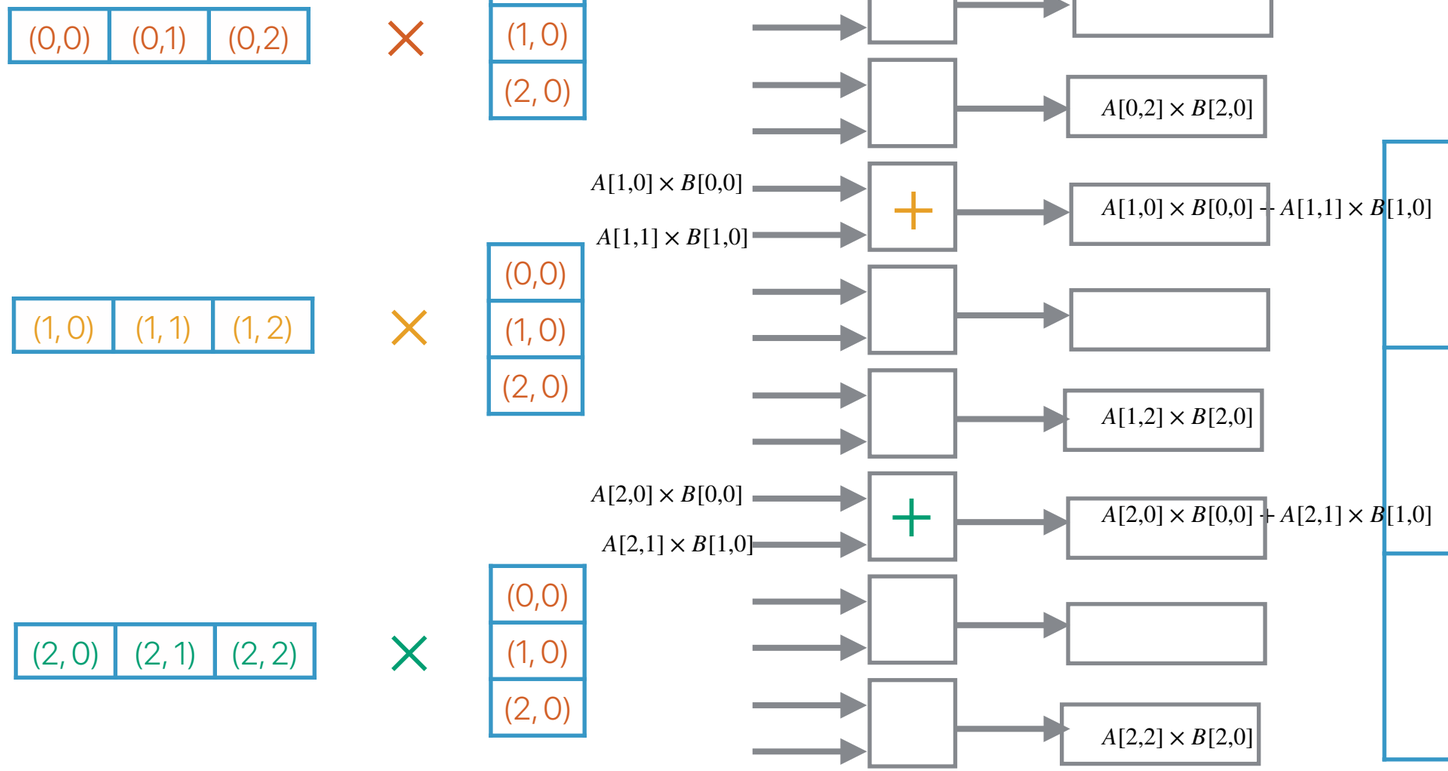
#2

B

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

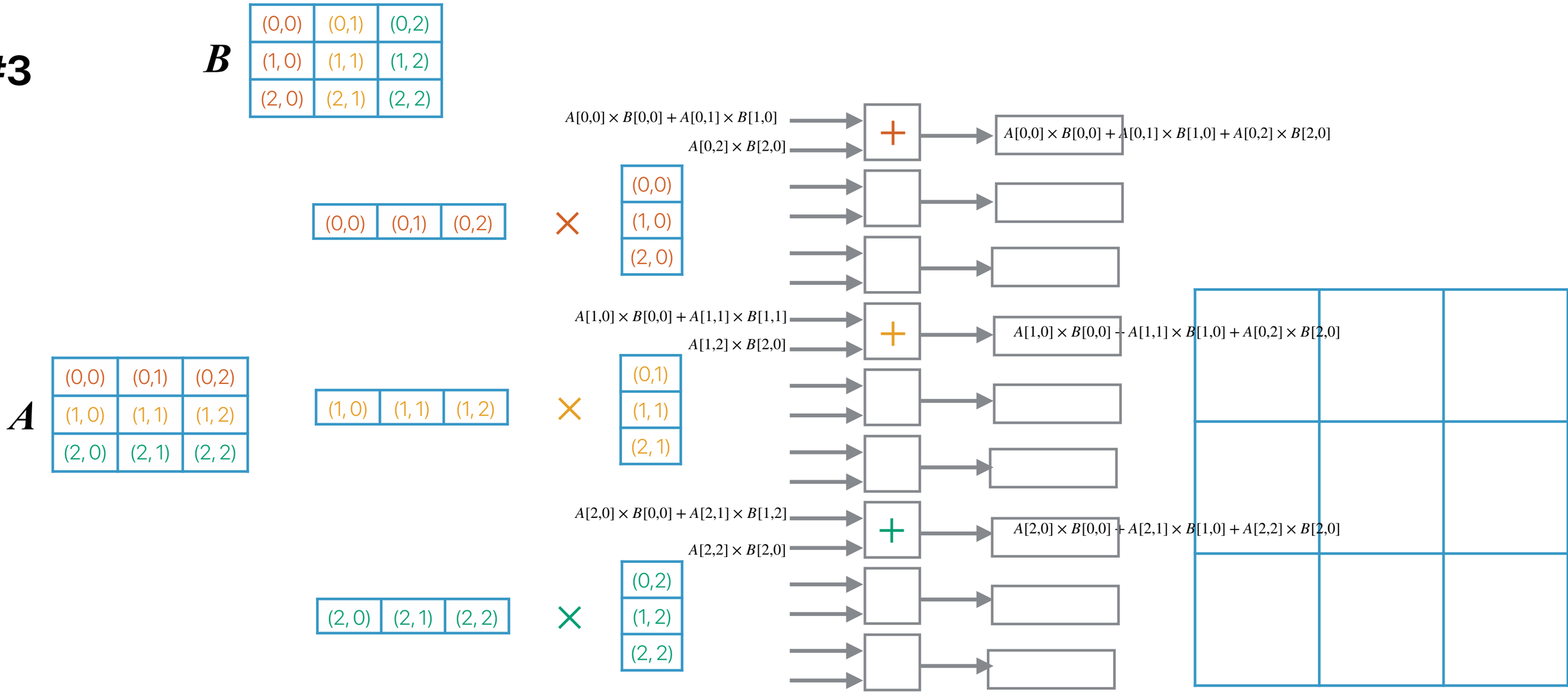
A

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)



Vector processing for MM

#3



Vector processing for MM

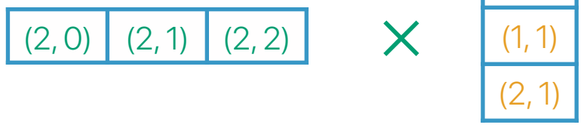
#4

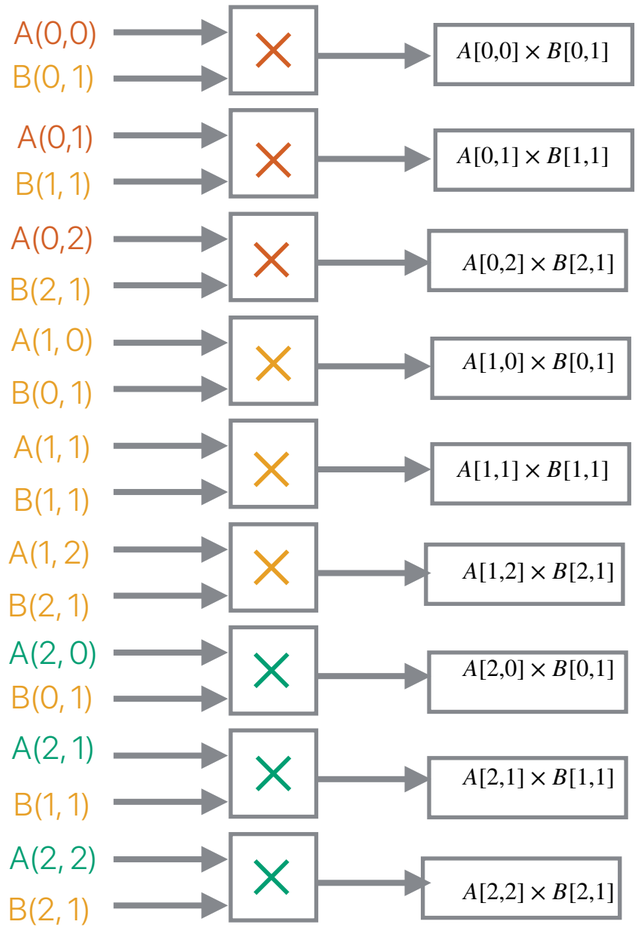
B

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

A

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

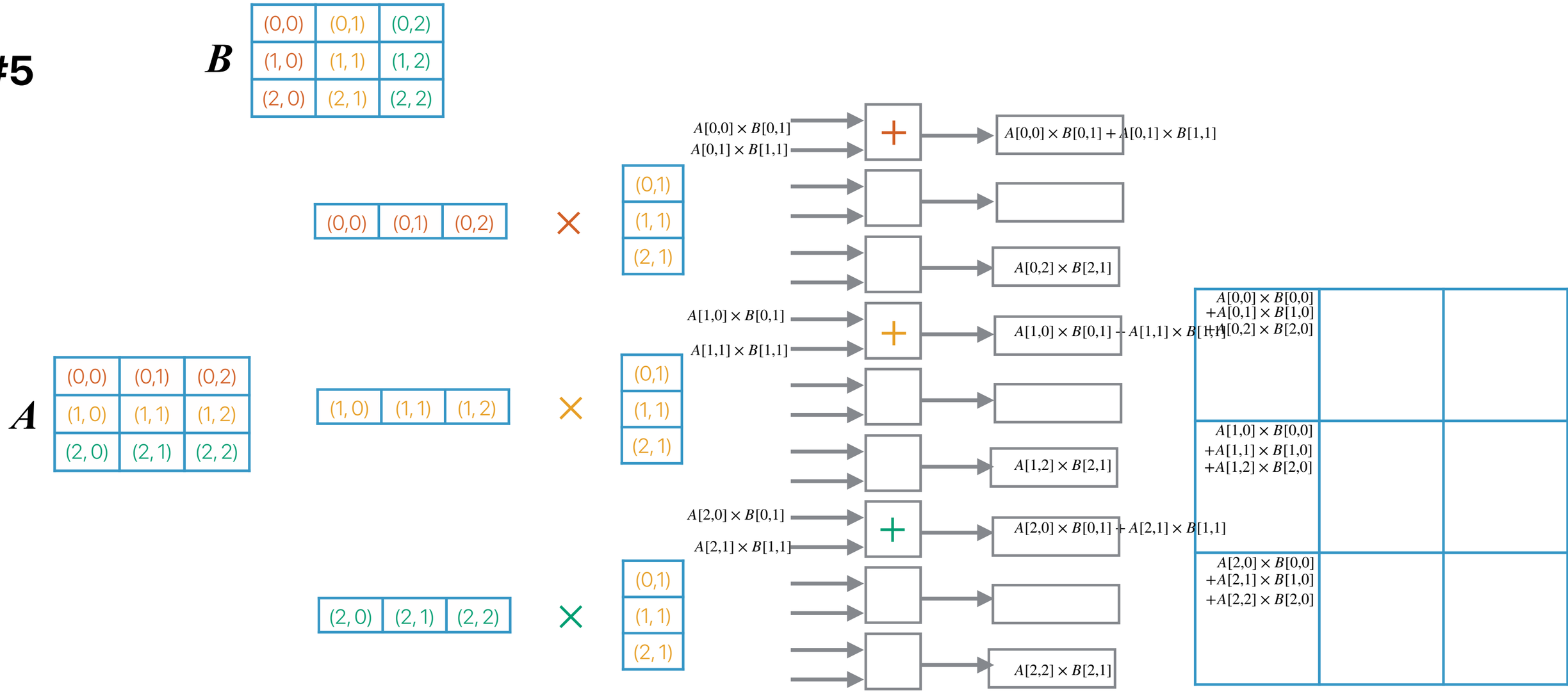




$A[0,0] \times B[0,0]$ $+A[0,1] \times B[1,0]$ $+A[0,2] \times B[2,0]$		
$A[1,0] \times B[0,0]$ $+A[1,1] \times B[1,0]$ $+A[1,2] \times B[2,0]$		
$A[2,0] \times B[0,0]$ $+A[2,1] \times B[1,0]$ $+A[2,2] \times B[2,0]$		

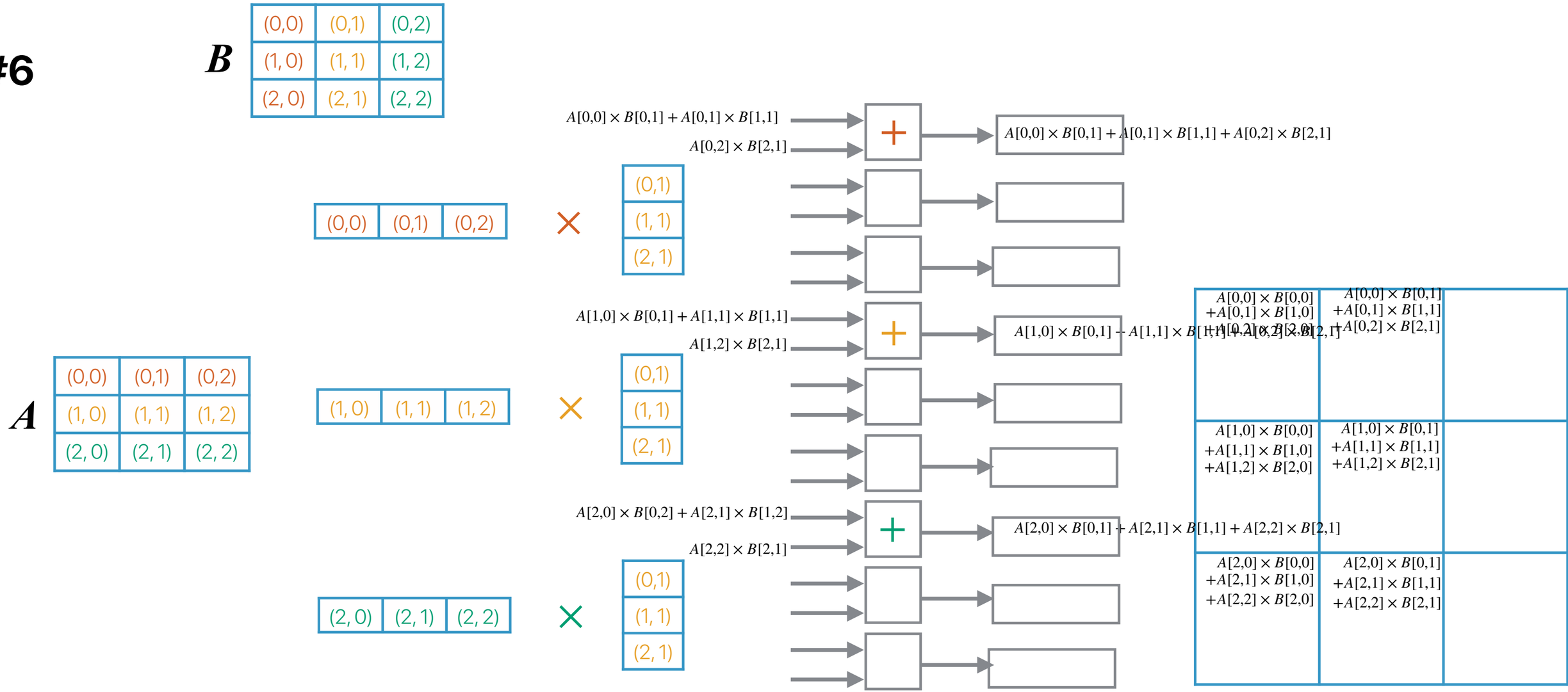
Vector processing for MM

#5



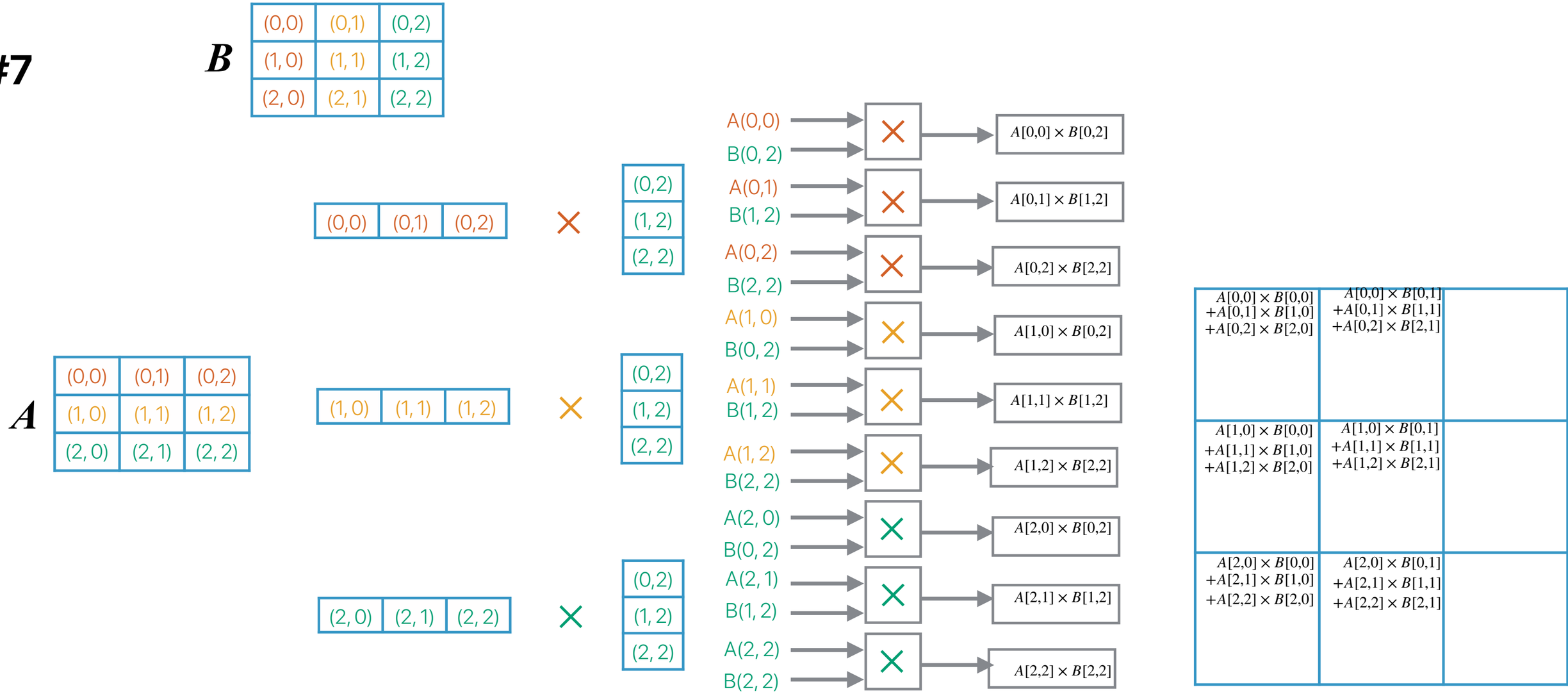
Vector processing for MM

#6



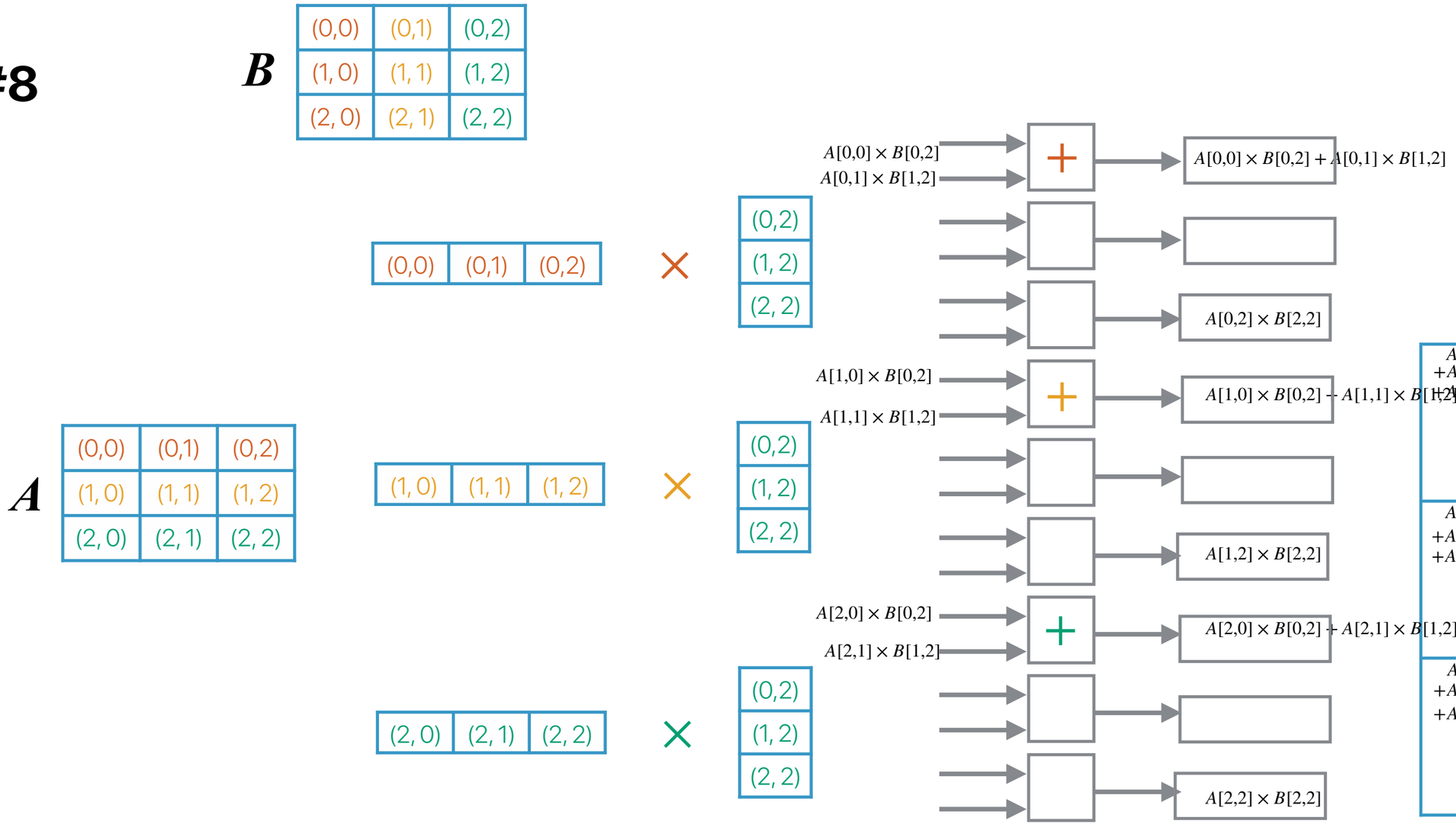
Vector processing for MM

#7



Vector processing for MM

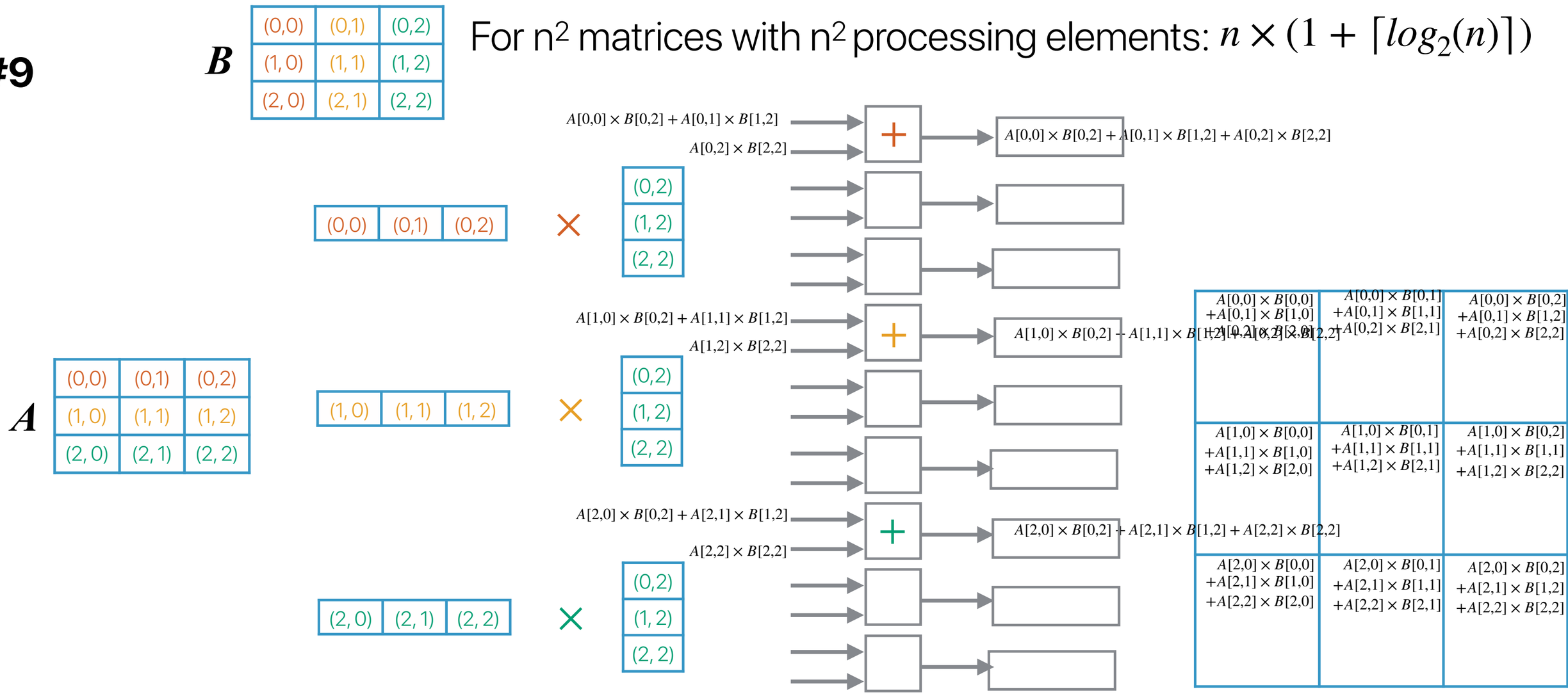
#8



Vector processing for MM

#9

For n² matrices with n² processing elements: $n \times (1 + \lceil log_2(n) \rceil)$



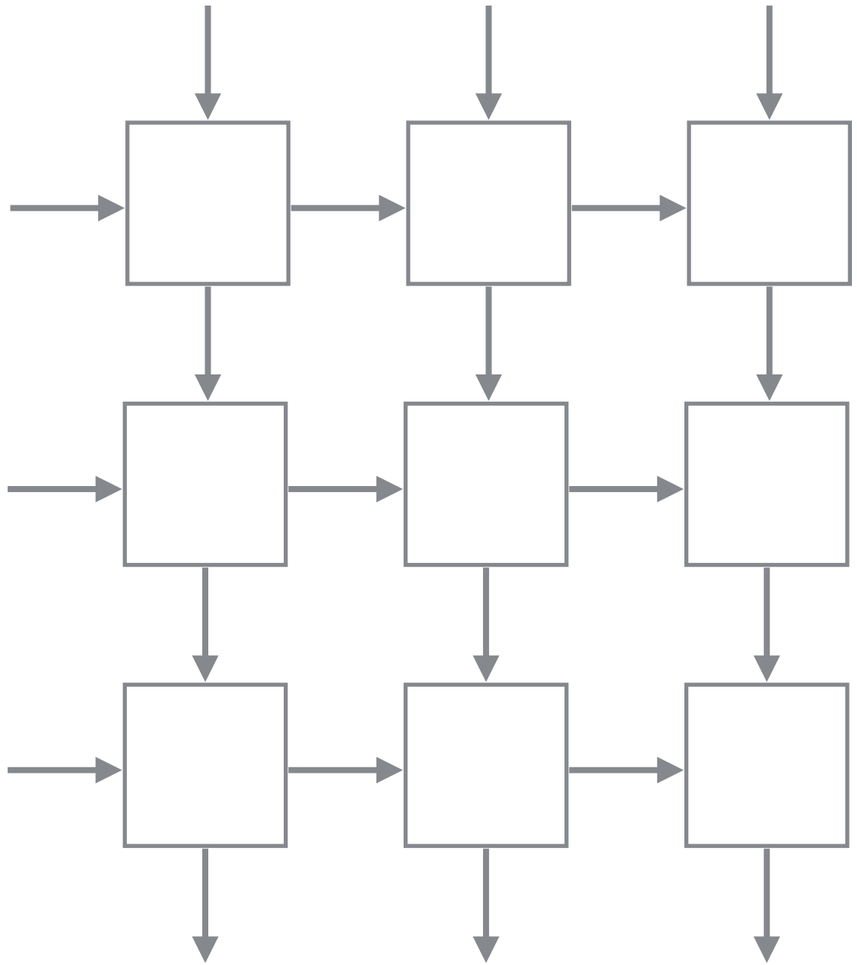
Systolic Arrays used by AI/ML Accelerators

B

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

A

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)



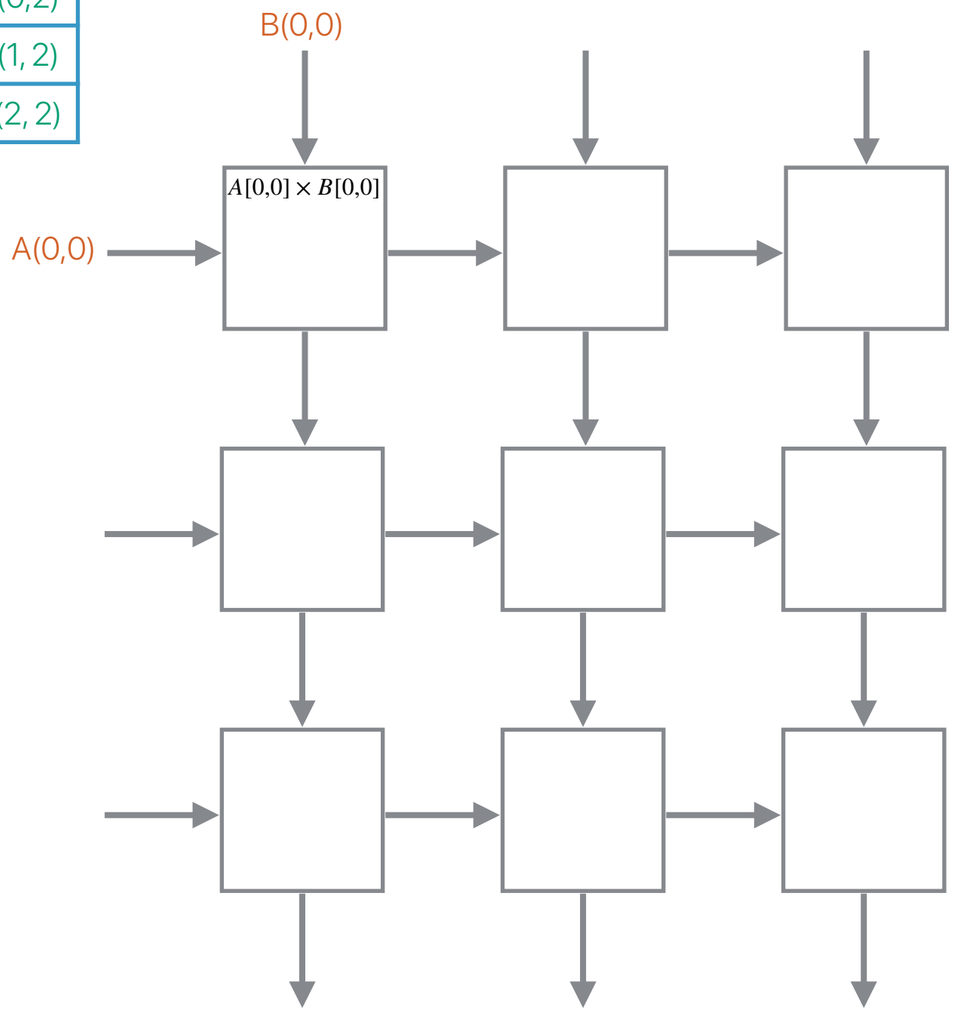
H.T. Kung, "Why systolic architectures?," in IEEE Computer, vol. 15, no. 1, pp. 37-46, Jan. 1982, doi: 10.1109/MC.1982.1653825.

Systolic Arrays used by AI/ML Accelerators

#1

B

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)



A

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

H.T. Kung, "Why systolic architectures?," in IEEE Computer, vol. 15, no. 1, pp. 37-46, Jan. 1982, doi: 10.1109/MC.1982.1653825.

Systolic Arrays used by AI/ML Accelerators

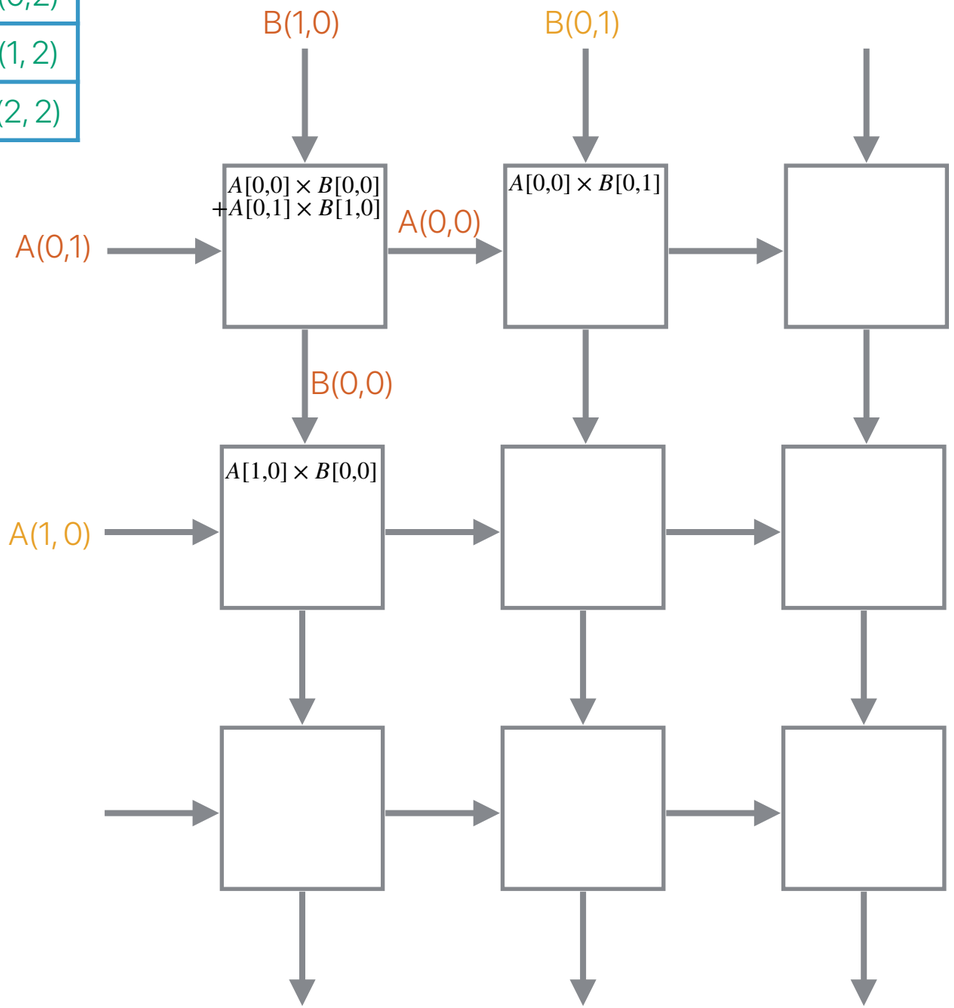
#2

B

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

A

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)



H.T. Kung, "Why systolic architectures?," in IEEE Computer, vol. 15, no. 1, pp. 37-46, Jan. 1982, doi: 10.1109/MC.1982.1653825.

Systolic Arrays used by AI/ML Accelerators

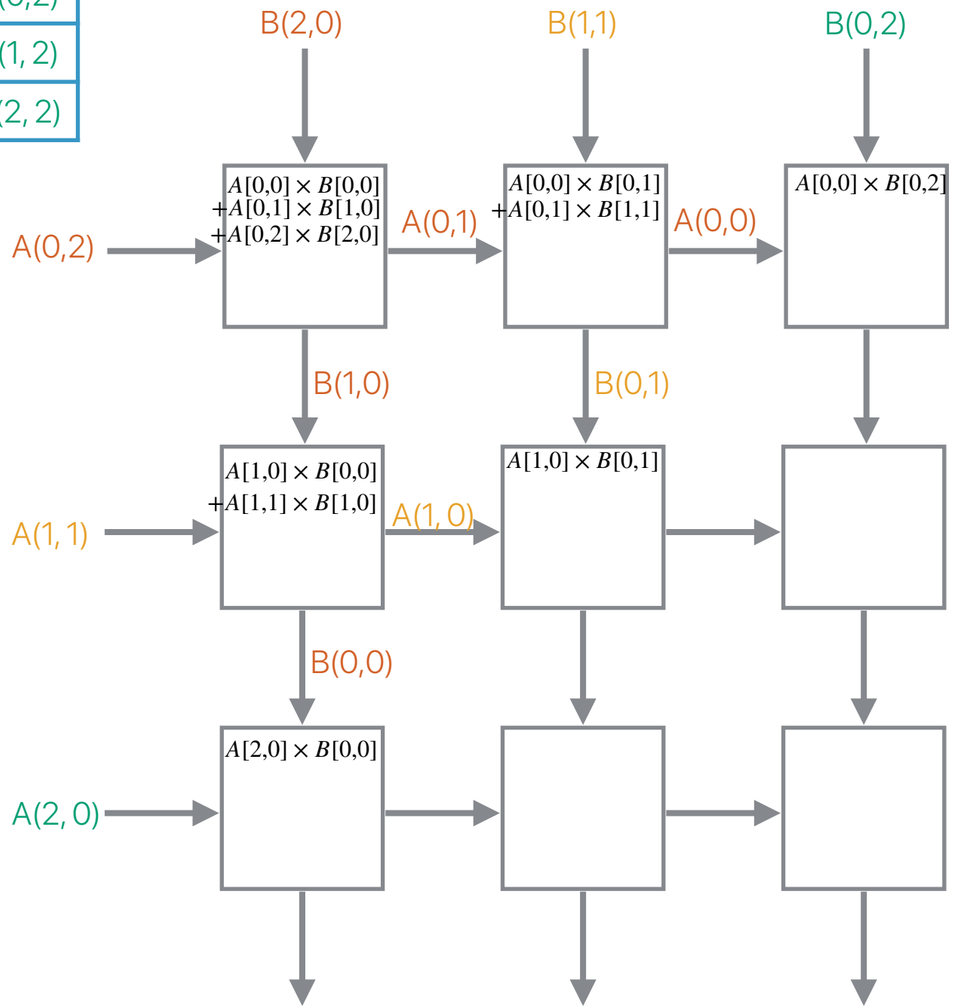
#3

B

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

A

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)



H.T. Kung, "Why systolic architectures?," in IEEE Computer, vol. 15, no. 1, pp. 37-46, Jan. 1982, doi: 10.1109/MC.1982.1653825.

Systolic Arrays used by AI/ML Accelerators

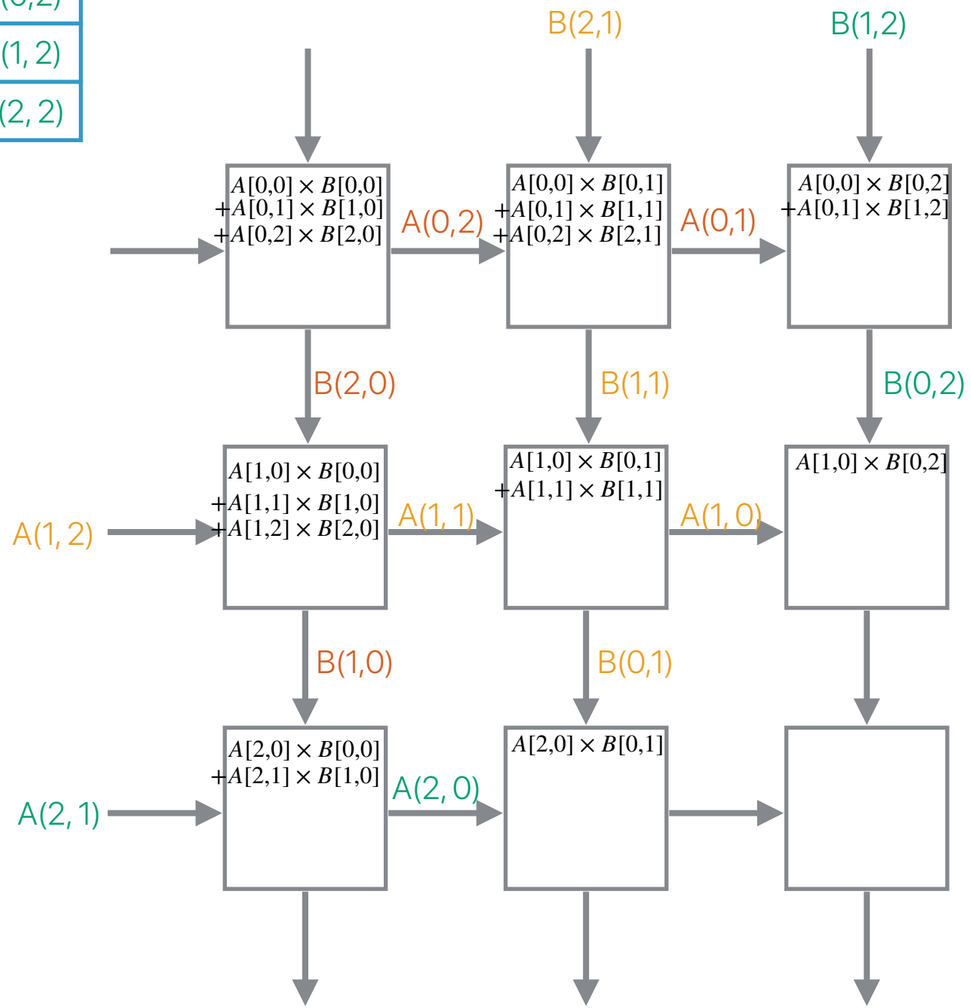
#4

B

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

A

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)



H.T. Kung, "Why systolic architectures?," in IEEE Computer, vol. 15, no. 1, pp. 37-46, Jan. 1982, doi: 10.1109/MC.1982.1653825.

Systolic Arrays used by AI/ML Accelerators

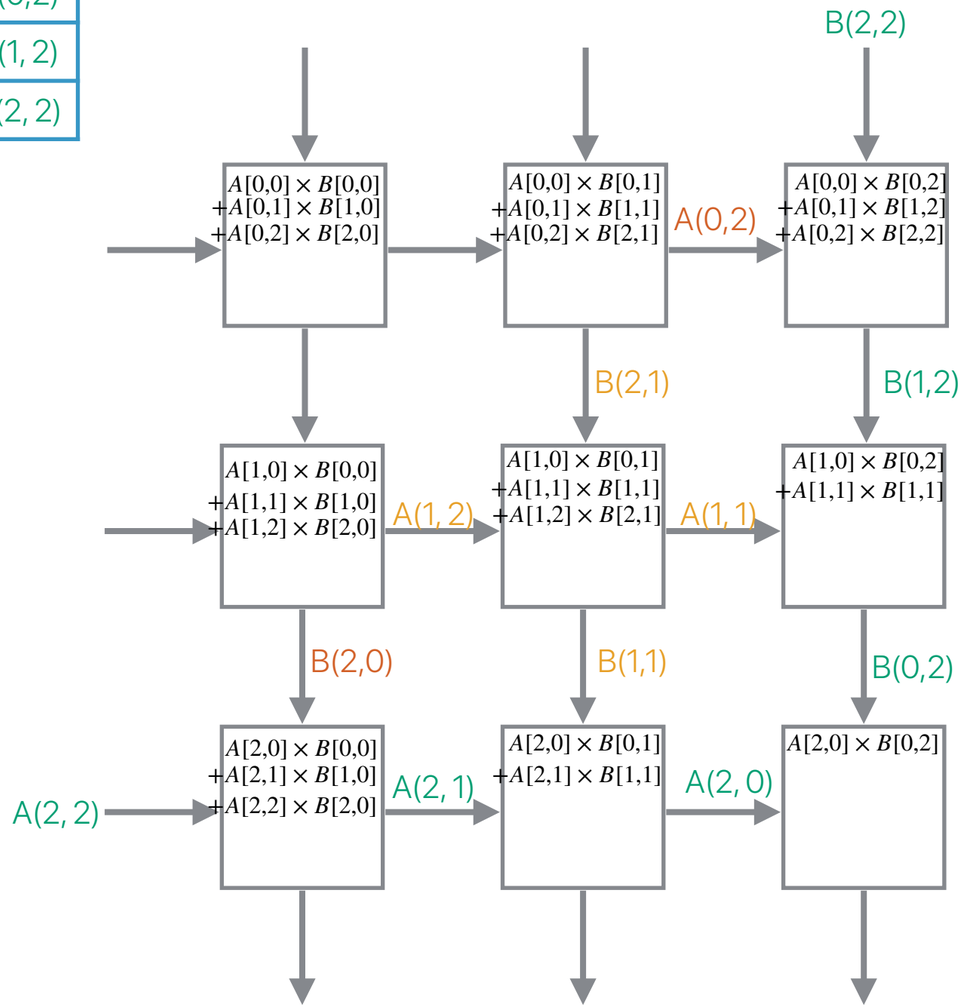
#5

B

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

A

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)



H.T. Kung, "Why systolic architectures?," in IEEE Computer, vol. 15, no. 1, pp. 37-46, Jan. 1982, doi: 10.1109/MC.1982.1653825.

Systolic Arrays used by AI/ML Accelerators

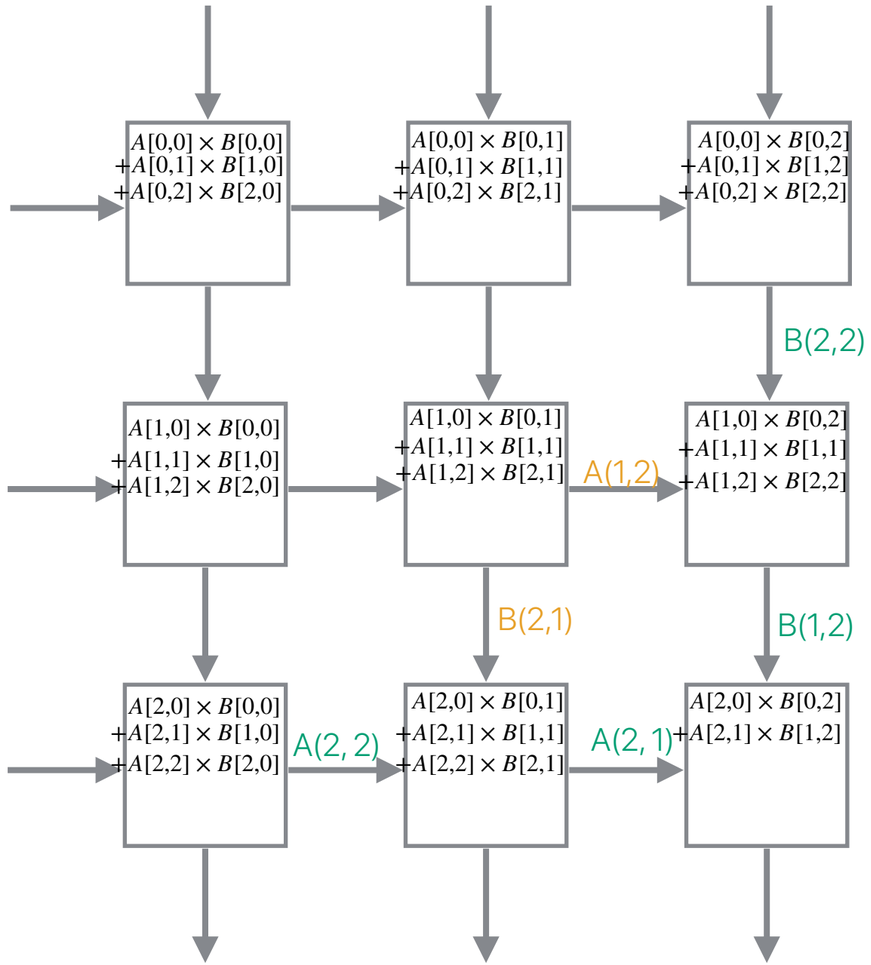
#6

B

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

A

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)



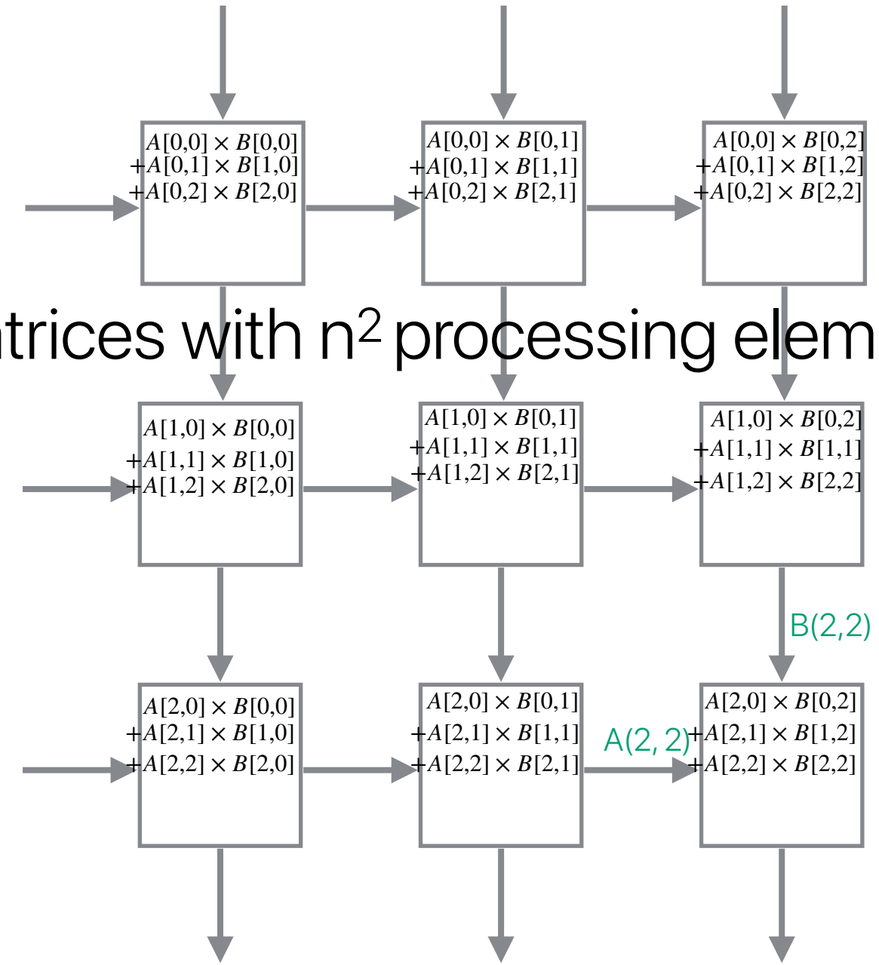
H.T. Kung, "Why systolic architectures?," in IEEE Computer, vol. 15, no. 1, pp. 37-46, Jan. 1982, doi: 10.1109/MC.1982.1653825.

Systolic Arrays used by AI/ML Accelerators

#7

B

(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)



For n² matrices with n² processing elements: $3 \times n - 2$

A

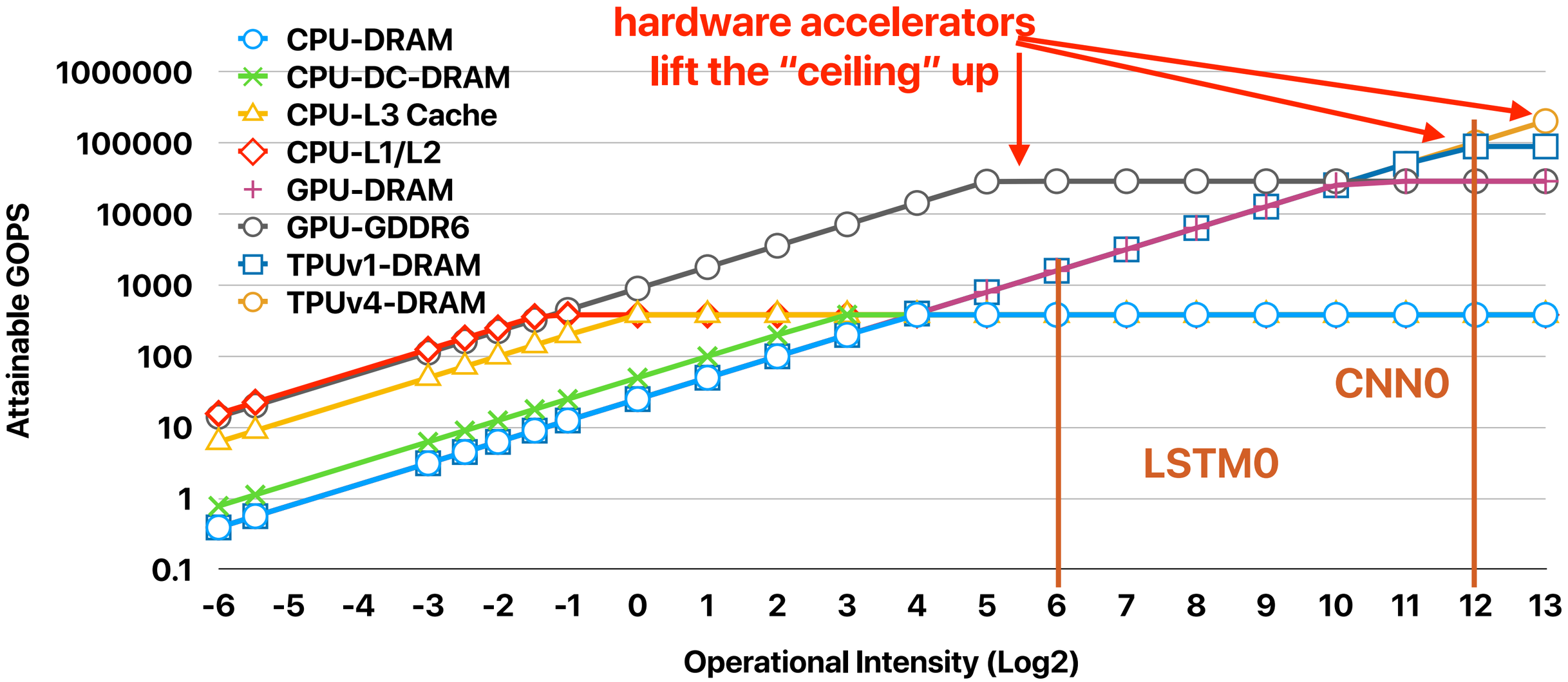
(0,0)	(0,1)	(0,2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

H.T. Kung, "Why systolic architectures?," in IEEE Computer, vol. 15, no. 1, pp. 37-46, Jan. 1982, doi: 10.1109/MC.1982.1653825.

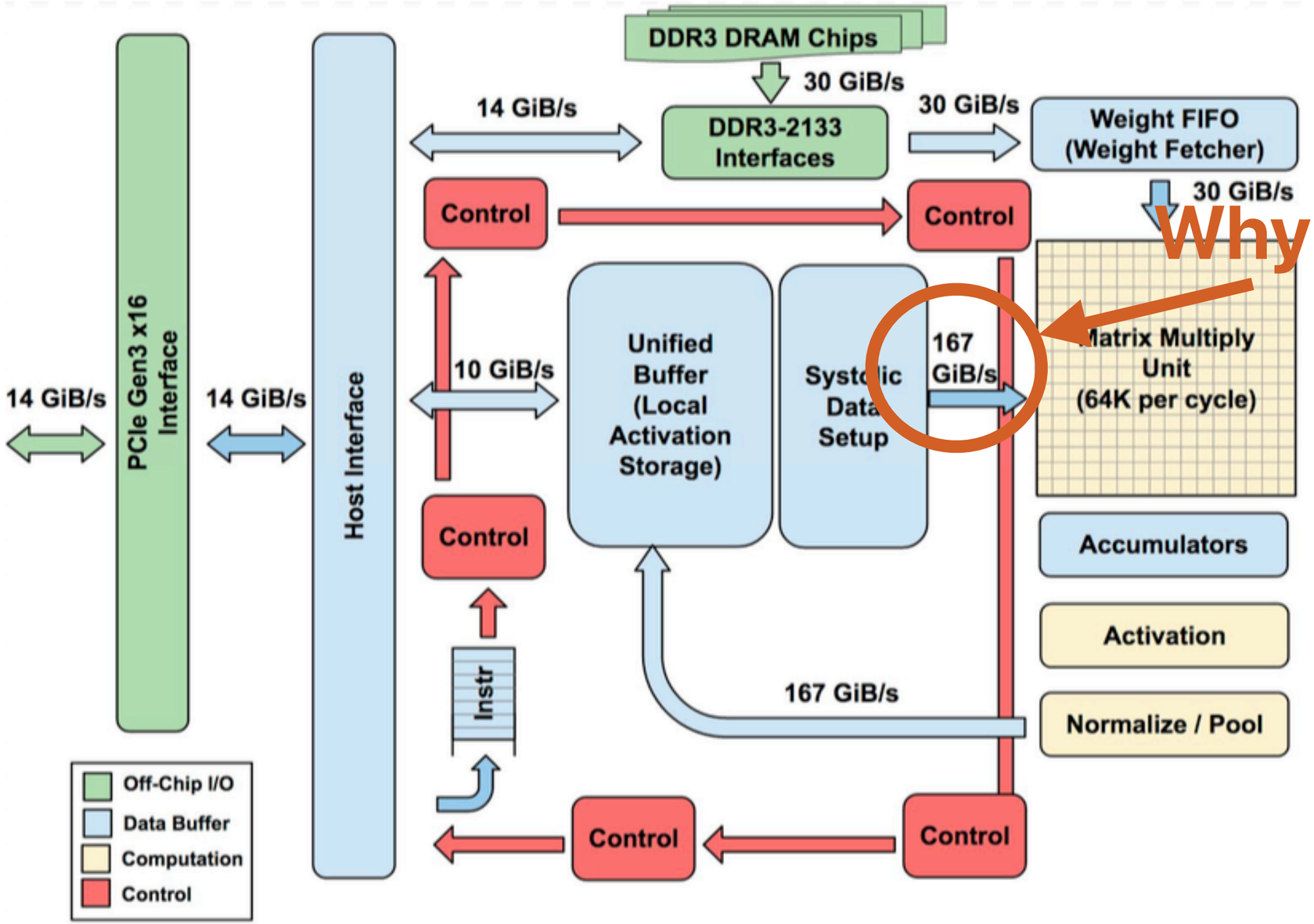
Why can accelerator help?

- Performance-wise $ET = IC \times CPI \times CT$
  - The accelerator itself reduces the amount of “instructions” used to implement a program
  - The accelerator’s logic is simpler than that of a general-purpose pipeline (potentially higher clock rate)
- Power-wise $P = \alpha C V^2 f$
  - Simpler logic reduces the waste of gates
- Energy-wise $Energy = P \times ET$
  - Energy is power times execution time
  - If you improve both, it’s better anyway

Placing TPUs in the roofline



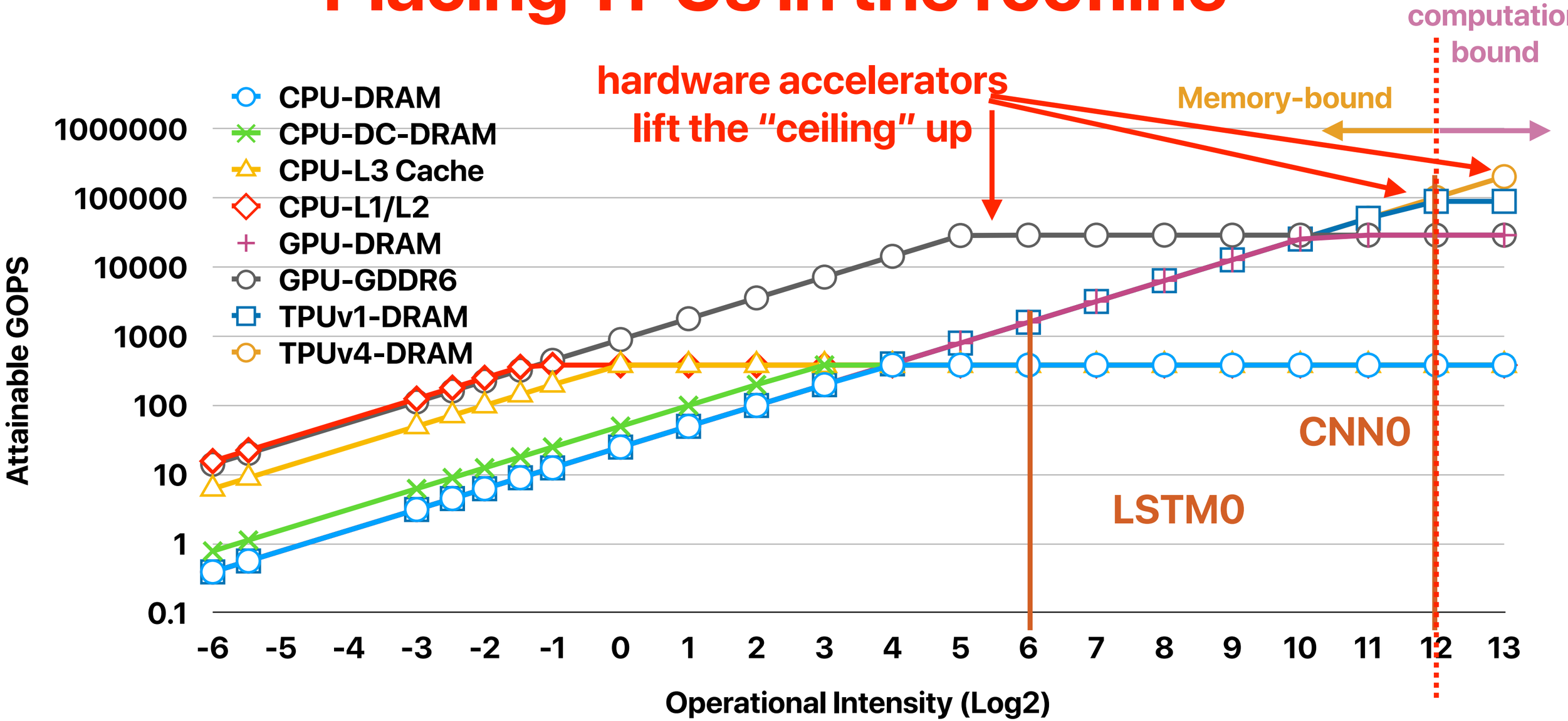
TPU Block diagram



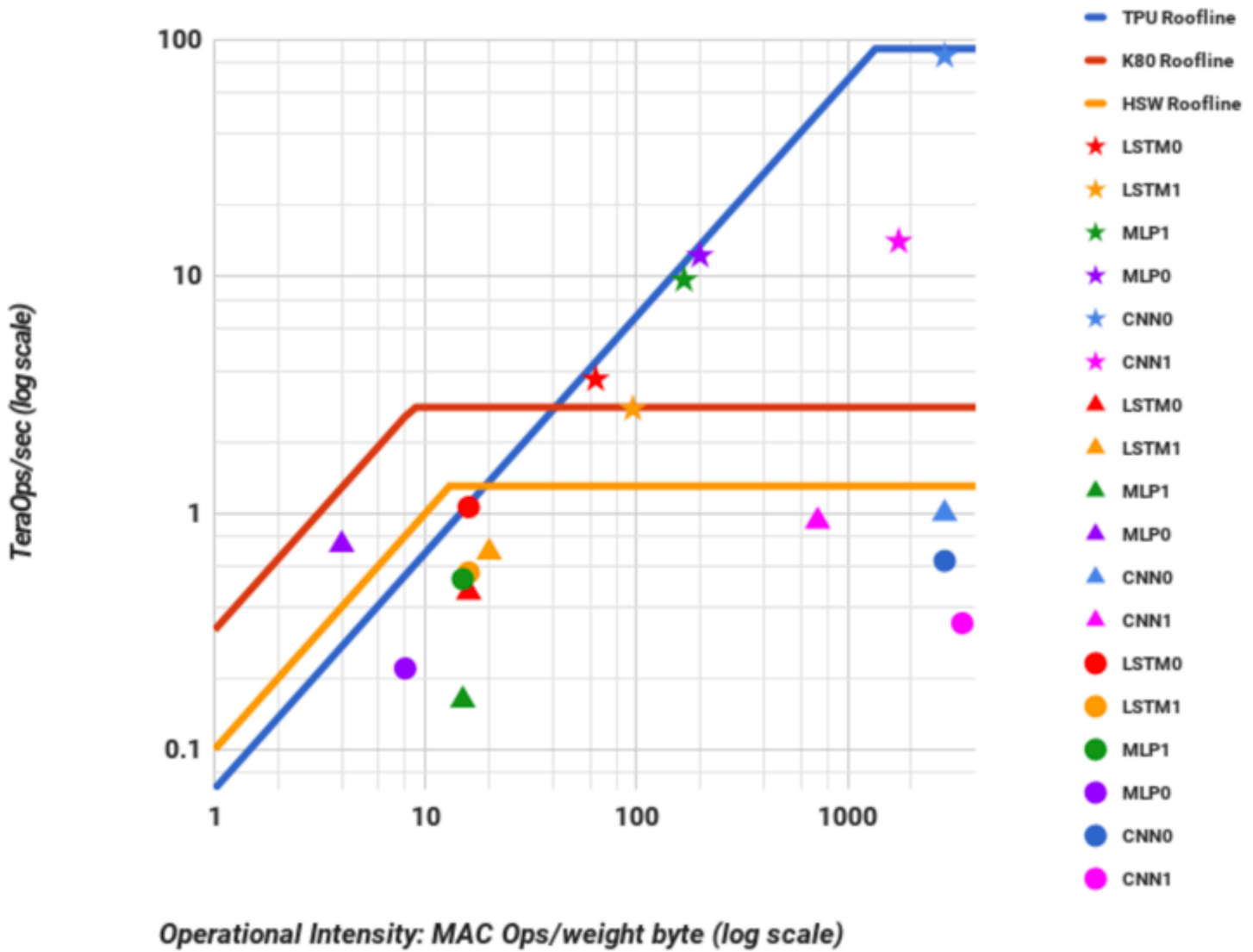
3:00

Why does TPU need large internal bandwidth?

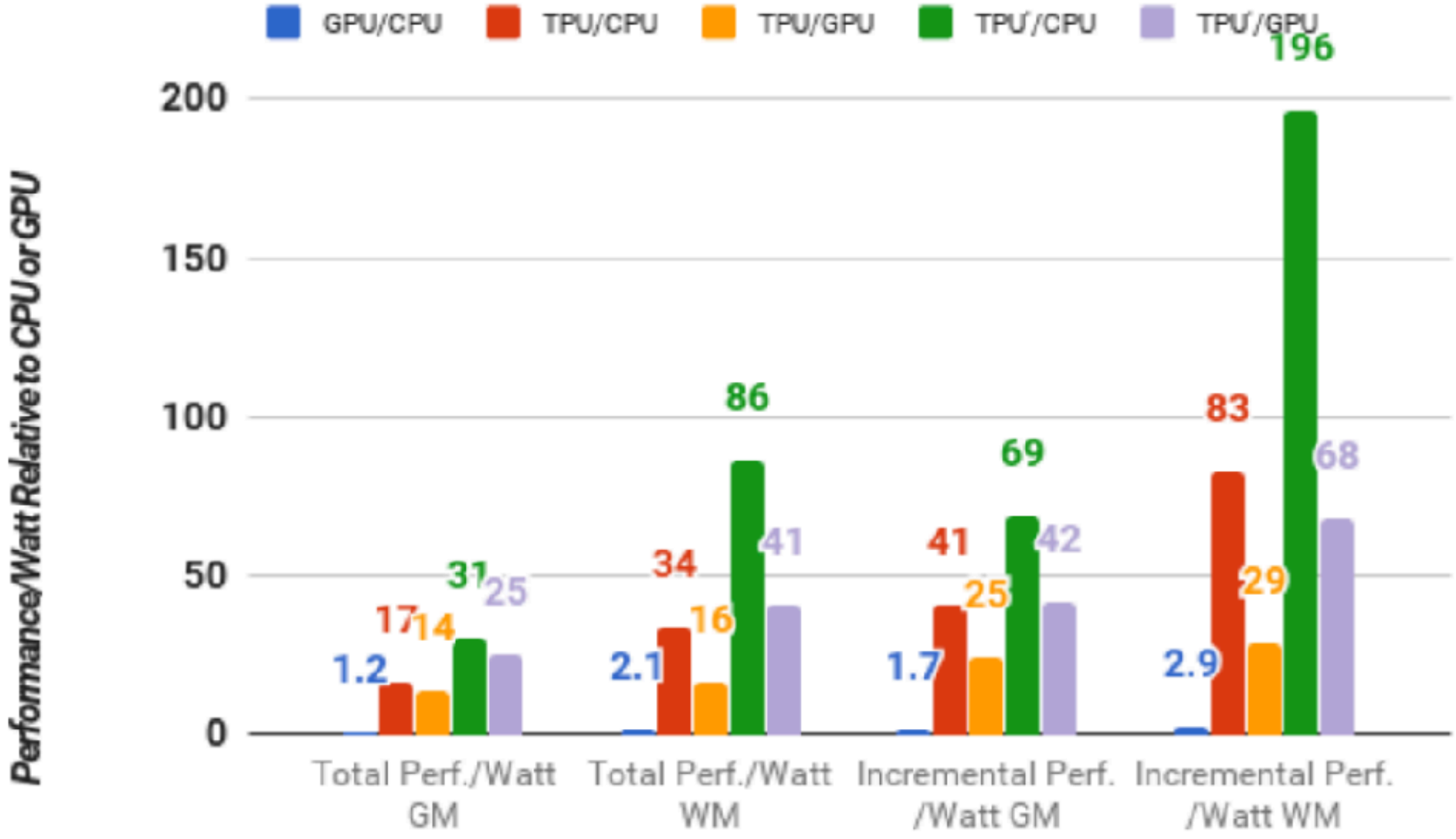
Placing TPUs in the roofline



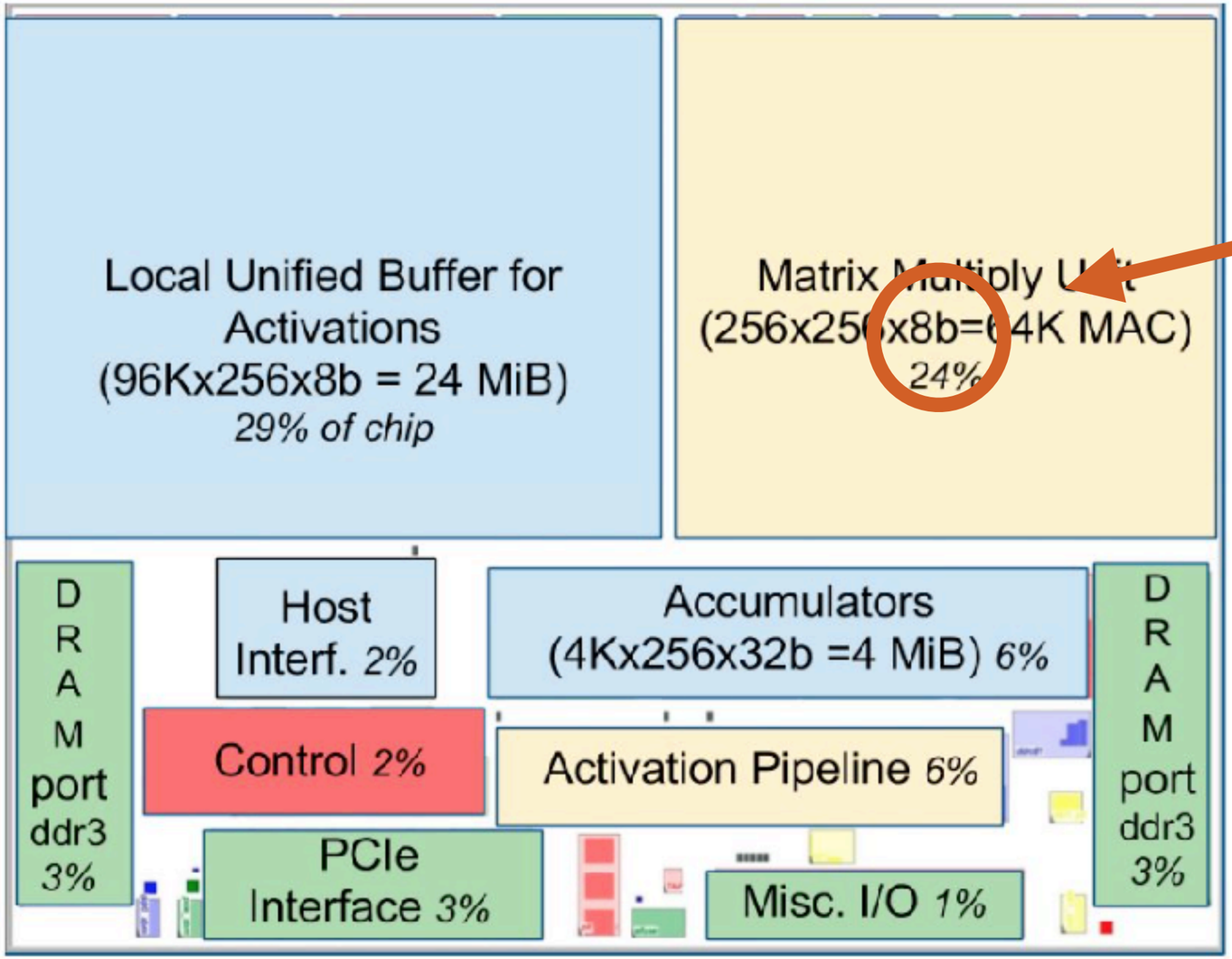
Most workloads surpass the roofline of GPUs



Energy efficiency is better! — reduced cost!



TPU Floorplan



Why 8-bit?

The benefit of application-specific design

- We only need to provide "just enough" hardware support rather than complete support
  - We can deliver higher throughput within the same area
    - Support 2x bits in precision 4x the area
  - Reduce the total power/energy consumption

Table 2. Benchmarked servers use Haswell CPUs, K80 GPUs, and TPUs. The GPU and TPU use the Haswell server as host. The TPU die is less than half the Haswell die size.

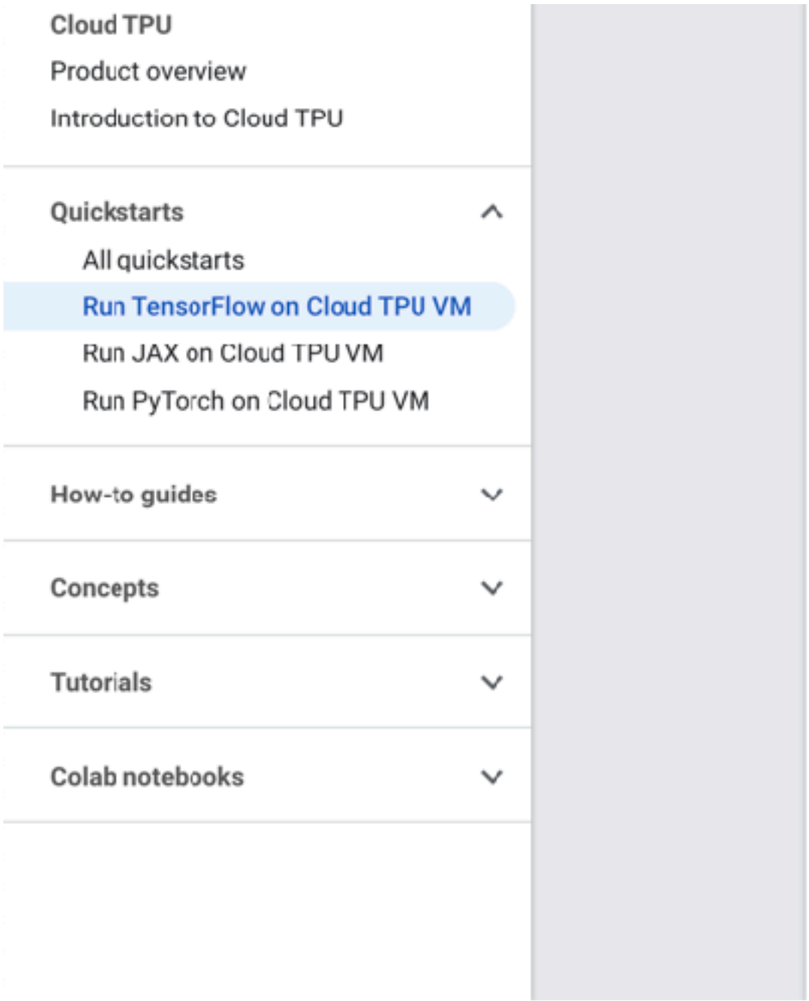
Model	mm^2	nm	MHz	Thermal Design Power (TDP) per Chip	Cores	TeraOps/s		Memory Gbyte/s	Chips/ Server	TDP per Server
						8 bit	FP32			
Haswell	662	22	2,300	145 W	18	2.6	1.3	51	2	504 W
Nvidia K80	561	28	560	150 W	13	--	2.8	160	8	1,838 W
TPU	< 331	28	700	75 W	1	92	--	34	4	861 W

From the TPU paper

Pitfall: Being Ignorant of Architecture History when Designing a DSA.

Ideas that didn't fly for general-purpose computing might be ideal for DSAs. For the TPU, three important architectural ideas date from the early 1980s: systolic arrays,[5] decoupled-access/execute,[4] and CISC instructions. The first reduced the area and power of the large matrix multiply unit, the second fetches weights concurrently during operation of the matrix multiply unit, and the third better utilizes the limited bandwidth of the PCIe bus for delivering instructions. History-aware architects could have a competitive edge in this DSA era.

TPU Programming model — domain specific languages



Note: For single TPUs, Slices, and Pods, pass your TPU name to `TPUClusterResolver()`.

```
import tensorflow as tf
print("Tensorflow version " + tf.__version__)

tpu = tf.distribute.cluster_resolver.TPUClusterResolver(tpu='your-tpu-name')  # TPU detection
print('Running on TPU ', tpu.cluster_spec().as_dict()['worker'])

tf.config.experimental_connect_to_cluster(tpu)
tf.tpu.experimental.initialize_tpu_system(tpu)
strategy = tf.distribute.experimental.TPUStrategy(tpu)

@tf.function
def add_fn(x,y):
    z = x + y
    return z

x = tf.constant(1.)
y = tf.constant(1.)
z = strategy.run(add_fn, args=(x,y))
print(z)
```

3:00

Do you think TPU is a good enough design for AI/ML applications?

Is TPU a good enough design?

10 lessons learned from 3 generations of Google TPUs

- Logic, wires, SRAM & DRAM improve unequally
- Leverage prior compiler optimizations
- Design for performance per TCO vs CapEx
- Backwards ML compatibility
- Inference DSAs need air cooling for global scale
- Some inference apps need floating point arithmetic
- Production inference normally needs multi-tenancy
- DNNs grow ~1.5x/year in memory and compute
- DNN workloads evolve with DNN breakthroughs
- Inference SLO limit is P99 latency, not batch size

N. P. Jouppi *et al.*, "Ten Lessons From Three Generations Shaped Google's TPUv4i : Industrial Product," *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, Valencia, Spain, 2021, pp. 1-14, doi: 10.1109/ISCA52012.2021.00010.

Roogle Project Meetings

- Make an appointment through the Google Calendar
- We're trying to make a company project theme
- Available resources
  - SmartSSD, Edge TPUs, CUDA GPUs, Tensor Cores, or something else
- Project ideas
  - Accelerating applications through AI/ML accelerators
  - Accelerating applications through intelligent storage devices
  - Accelerating applications through innovative parallel programming models that hardware accelerators enable
  - Anything related to what we discussed in this class!

Electrical
Computer Science
Engineering

277

つづく