18-742:
Computer Architecture & Systems

3D-Stacked Memory Architectures for Multi-Core Processors

Prof. Phillip Gibbons

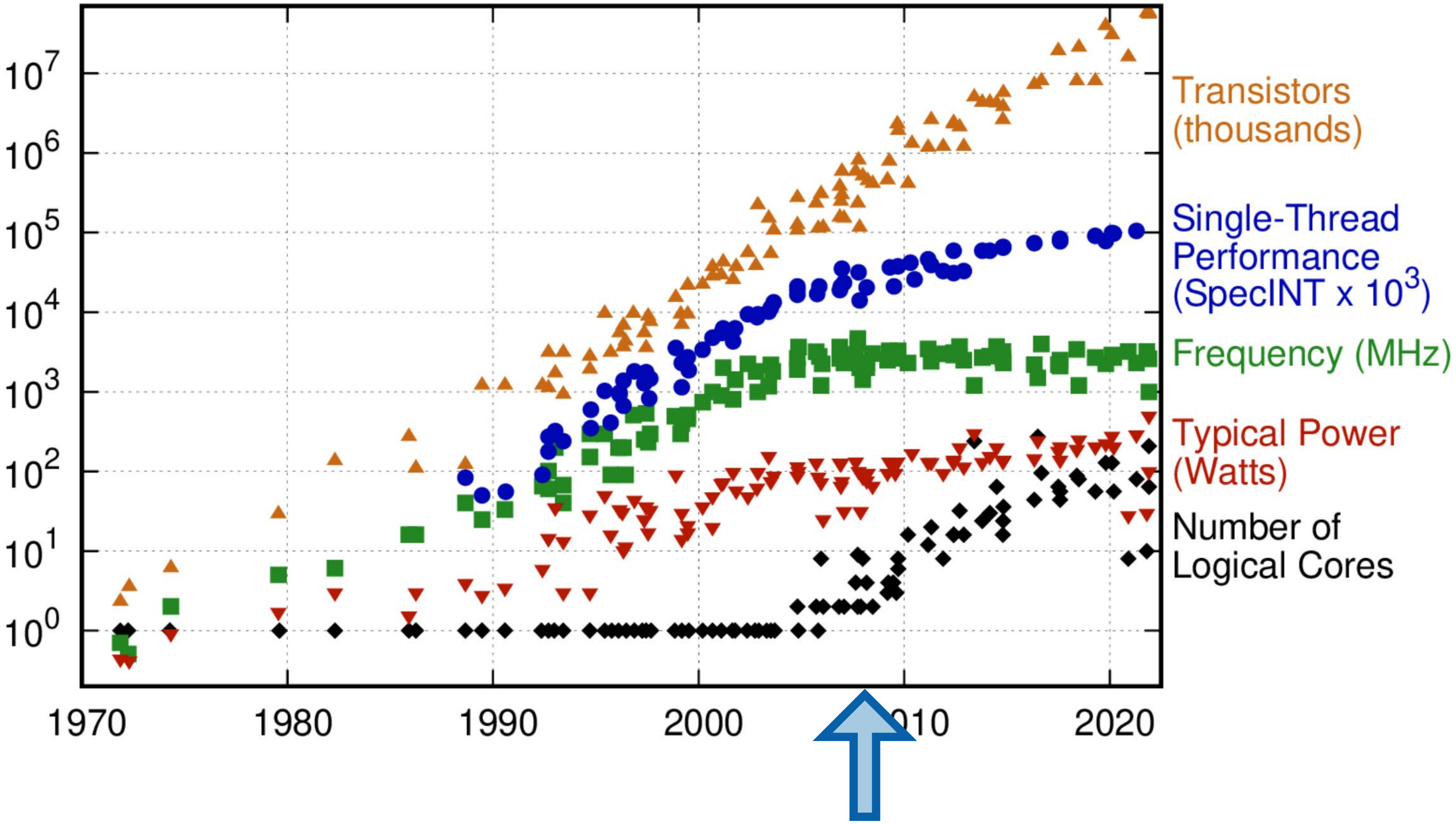
Spring 2025, Lecture 14

“3D-Stacked Memory Architectures for Multi-Core Processors”

Gabriel Loh 2008

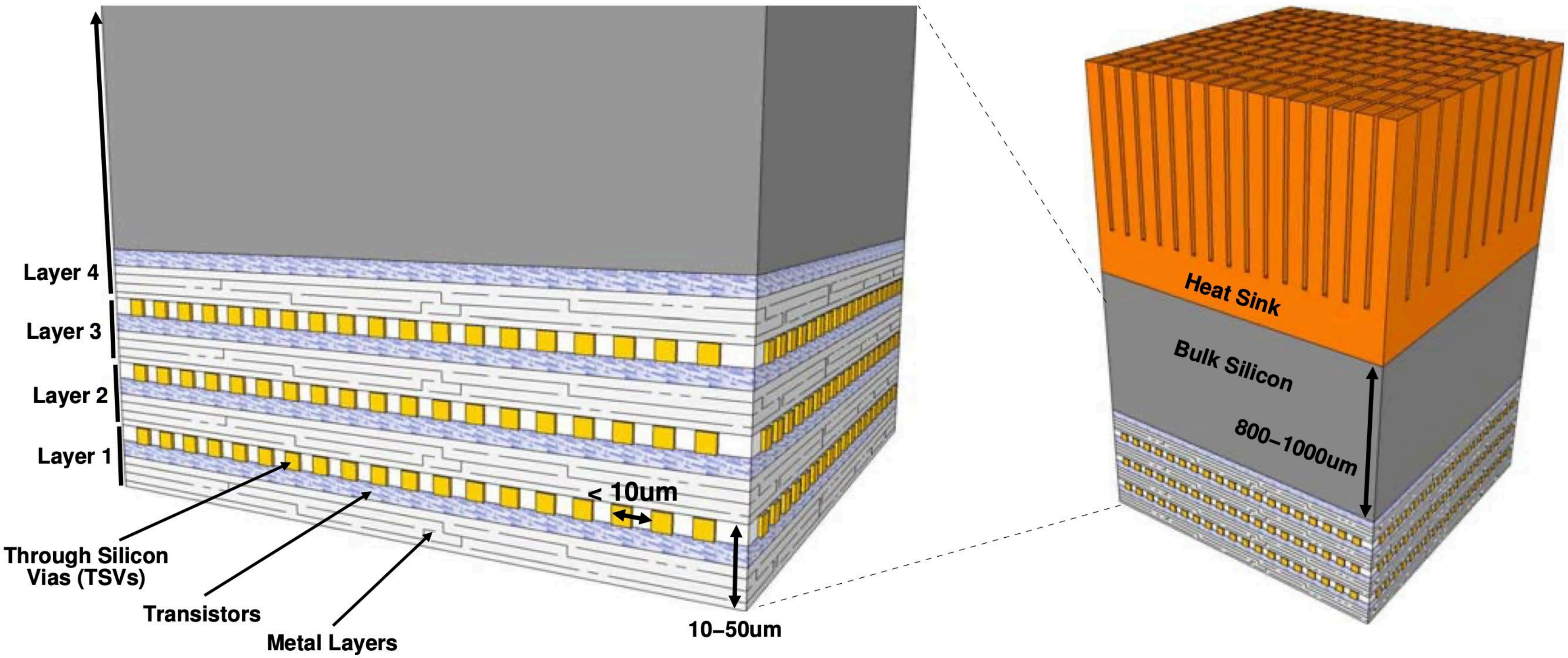
- **Gabe: Georgia Tech prof, now Senior Fellow@AMD**
  - ACM SIGARCH's Maurice Wilkes Award (2018)
  - ACM & IEEE Fellow

50 Years of Microprocessor Trend Data

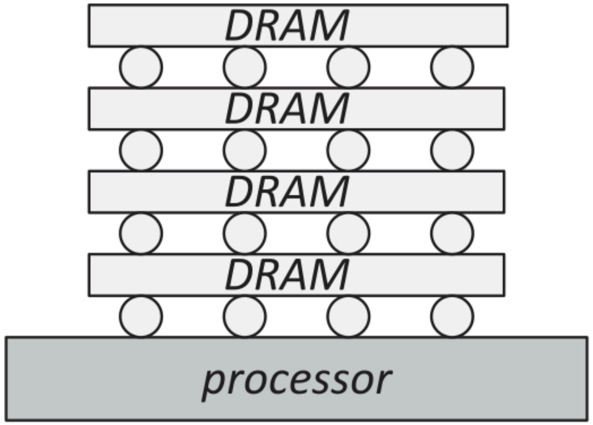


We are here

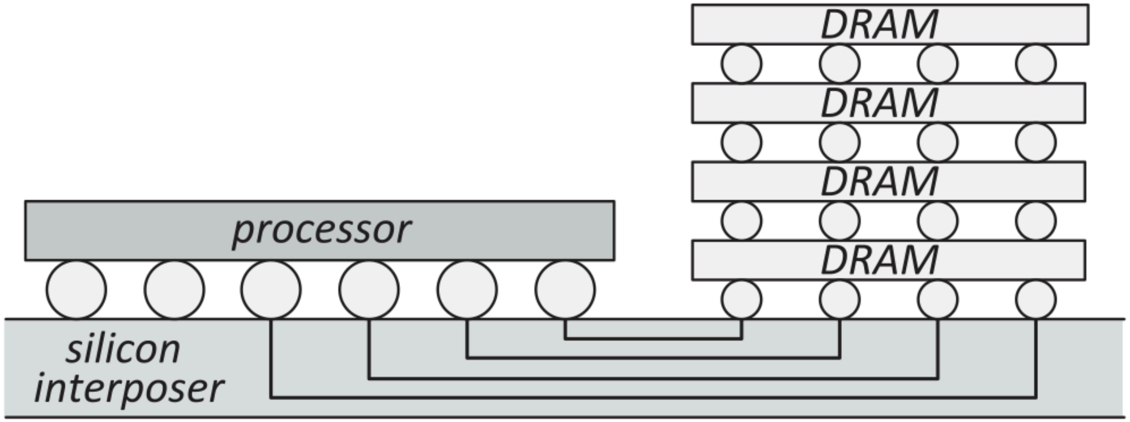
3D DRAM for Achieving Higher Bandwidth



Stacking Topology: 3D vs. 2.5D

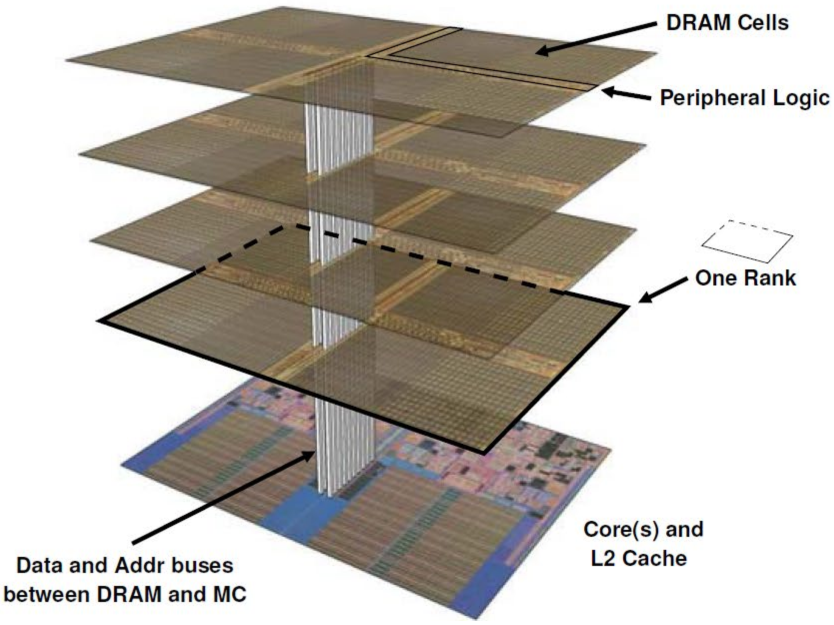


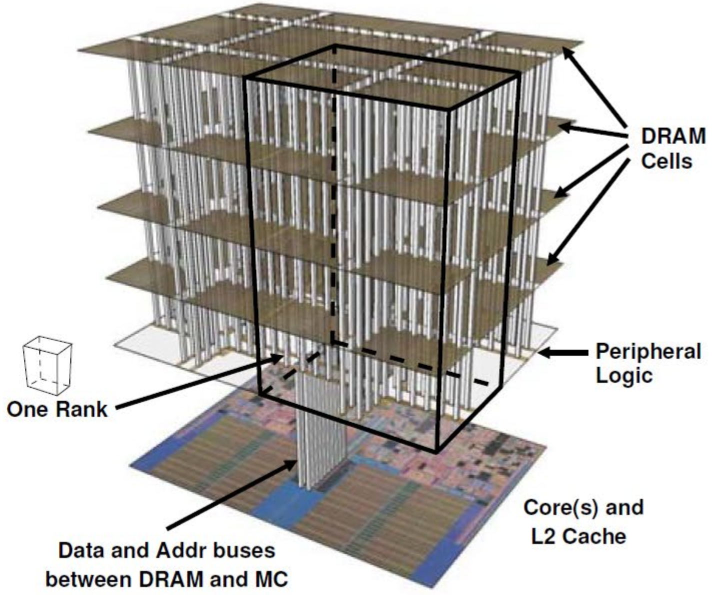
(a) Placing on top of processor (3D)



(b) Connecting with a silicon interposer (2.5D)

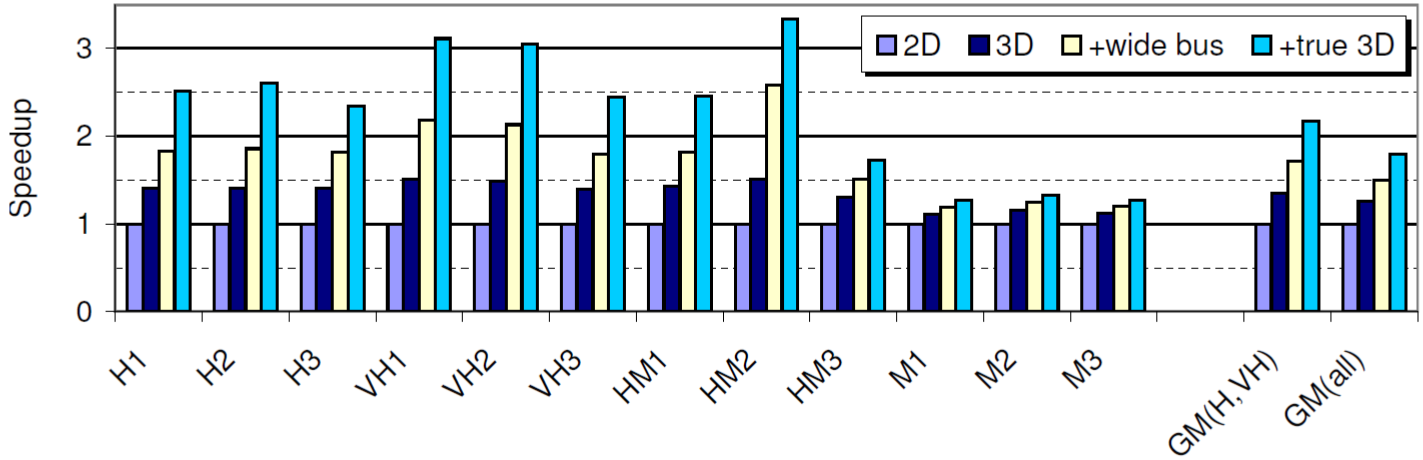
Rank Topology: 3D vs. True-3D





The Paper's Contribution

- **True-3D provides much higher performance than 3D**



- **To better utilize bandwidth, we need to enlarge L2 cache's MSHR**
  - But MSHR is fully-associative (not scalable)

- **This paper: Direct-mapped MSHR per MC + Vector Bloom filter**

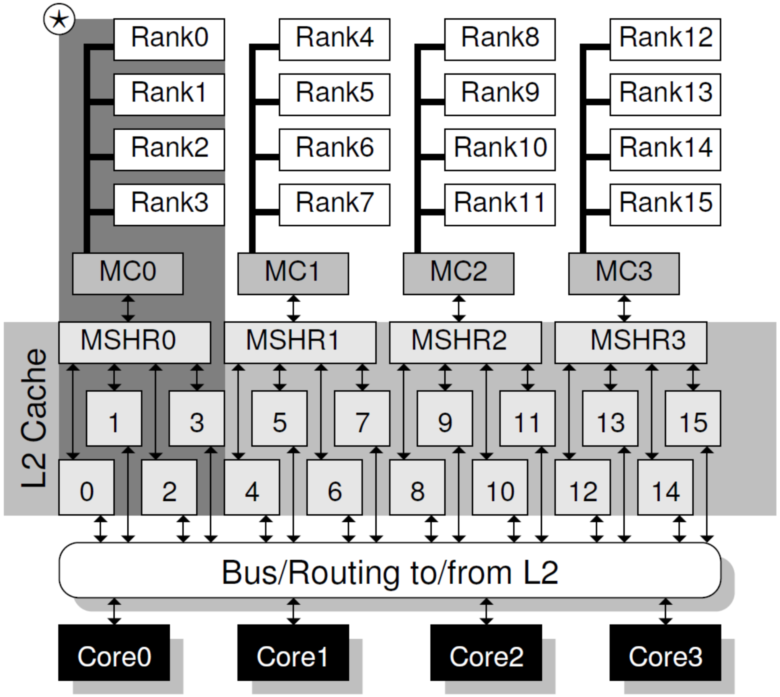
Discussion: Summary Question #1

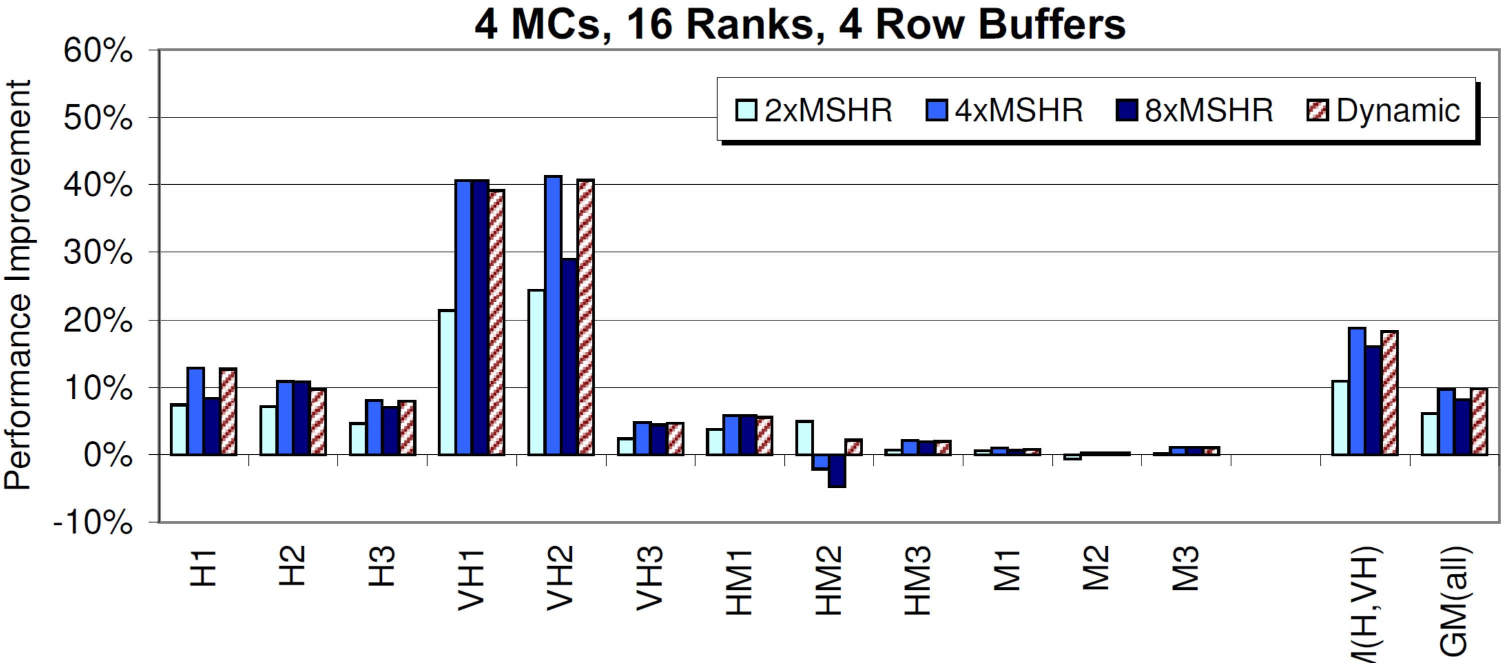
What Did the Paper Get Right?

State the 3 most important things the paper says.

These could be some combination of the motivations, observations, interesting parts of the design, or clever parts of the implementation.

Scalable Miss Handler Architecture





(b)

Discussion: Summary Question #2

What Did the Paper Get Wrong?

Describe the paper's single most glaring deficiency.

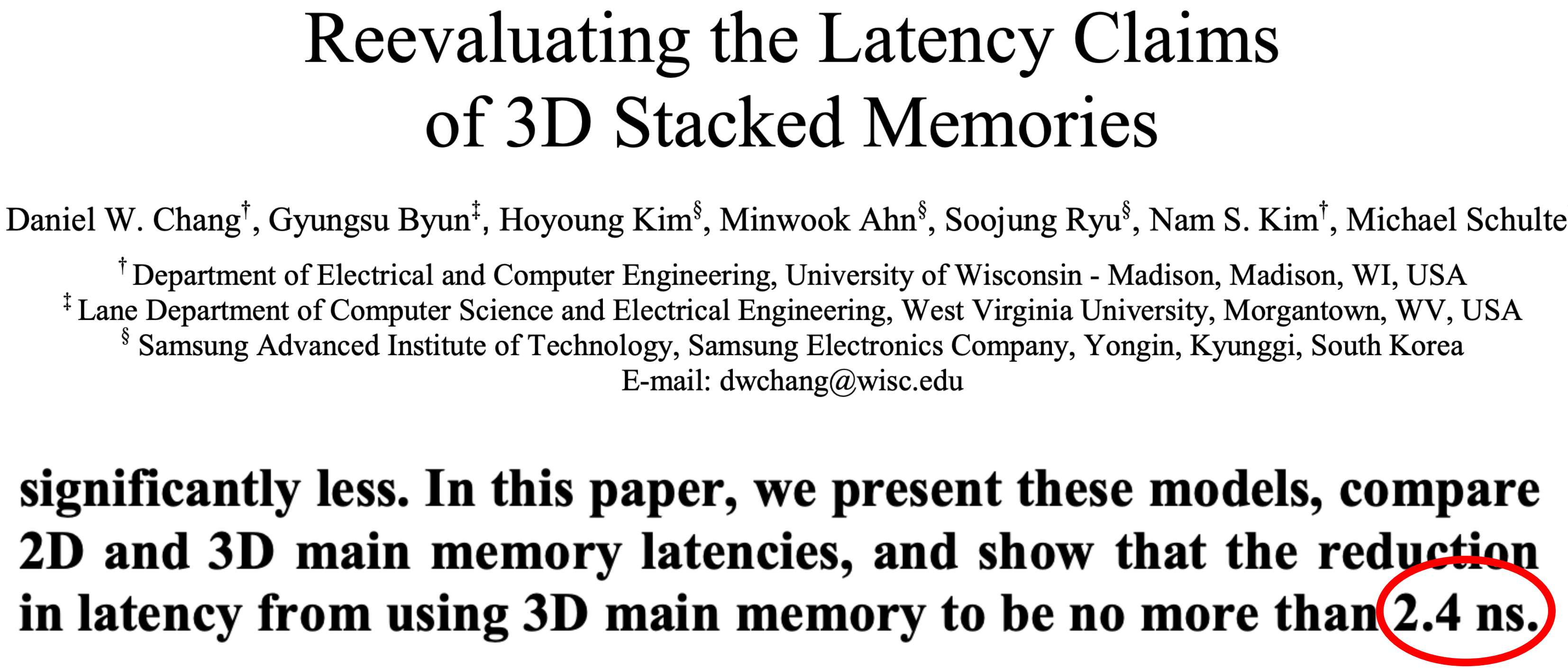
Every paper has some fault. Perhaps an experiment was poorly designed or the main idea had a narrow scope or applicability.

Benefits of 3D DRAM

- **Much Higher bandwidth**

- **Much Lower latency?**
  - No!

[ASPDAC'13]



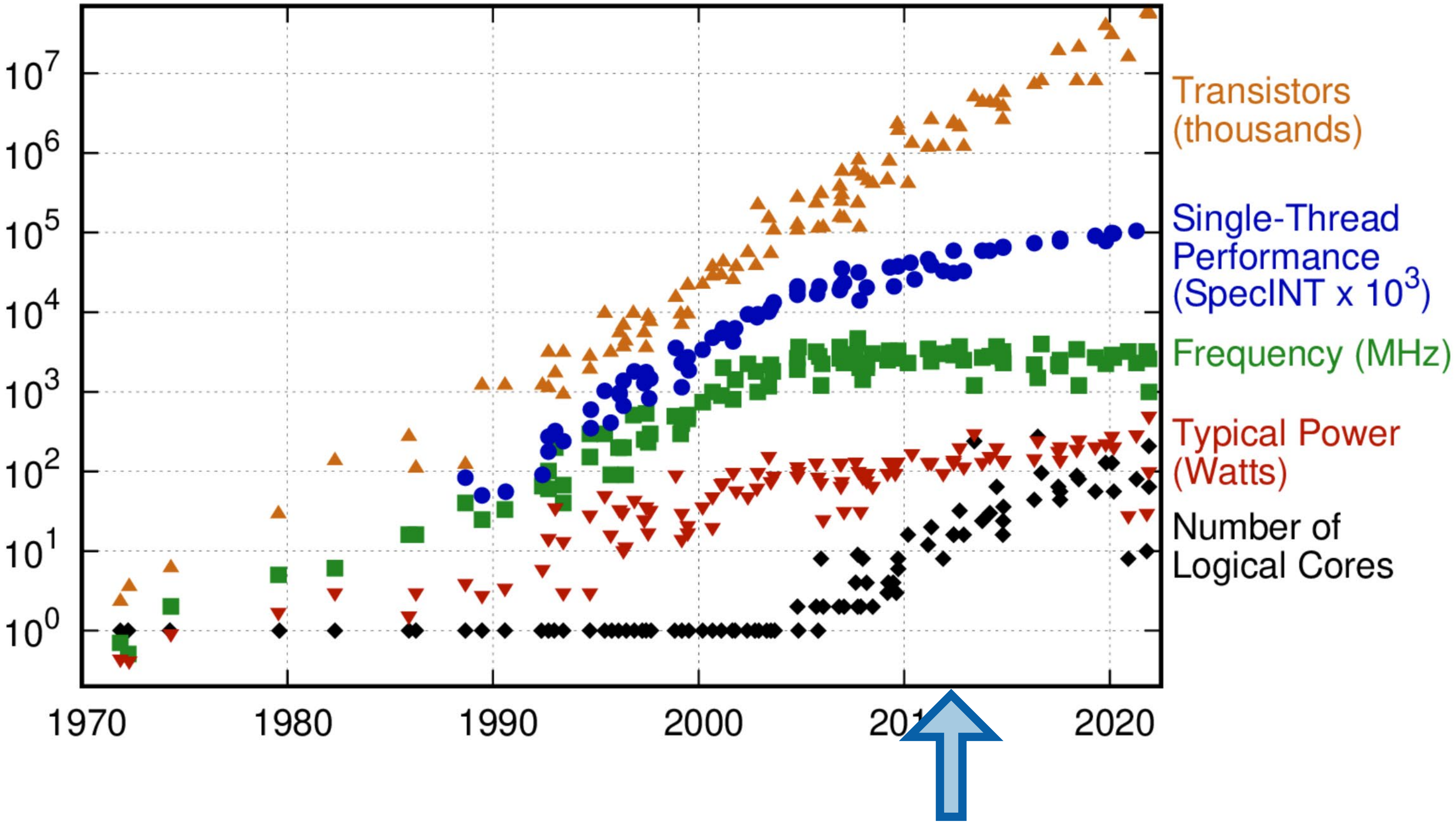
Reevaluating the Latency Claims
of 3D Stacked Memories

Daniel W. Chang[†], Gyungsu Byun[‡], Hoyoung Kim[§], Minwook Ahn[§], Soojung Ryu[§], Nam S. Kim[†], Michael Schulte[†]

[†] Department of Electrical and Computer Engineering, University of Wisconsin - Madison, Madison, WI, USA
[‡] Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV, USA
[§] Samsung Advanced Institute of Technology, Samsung Electronics Company, Yongin, Kyunggi, South Korea
E-mail: dwchang@wisc.edu

significantly less. In this paper, we present these models, compare 2D and 3D main memory latencies, and show that the reduction in latency from using 3D main memory to be no more than 2.4 ns.

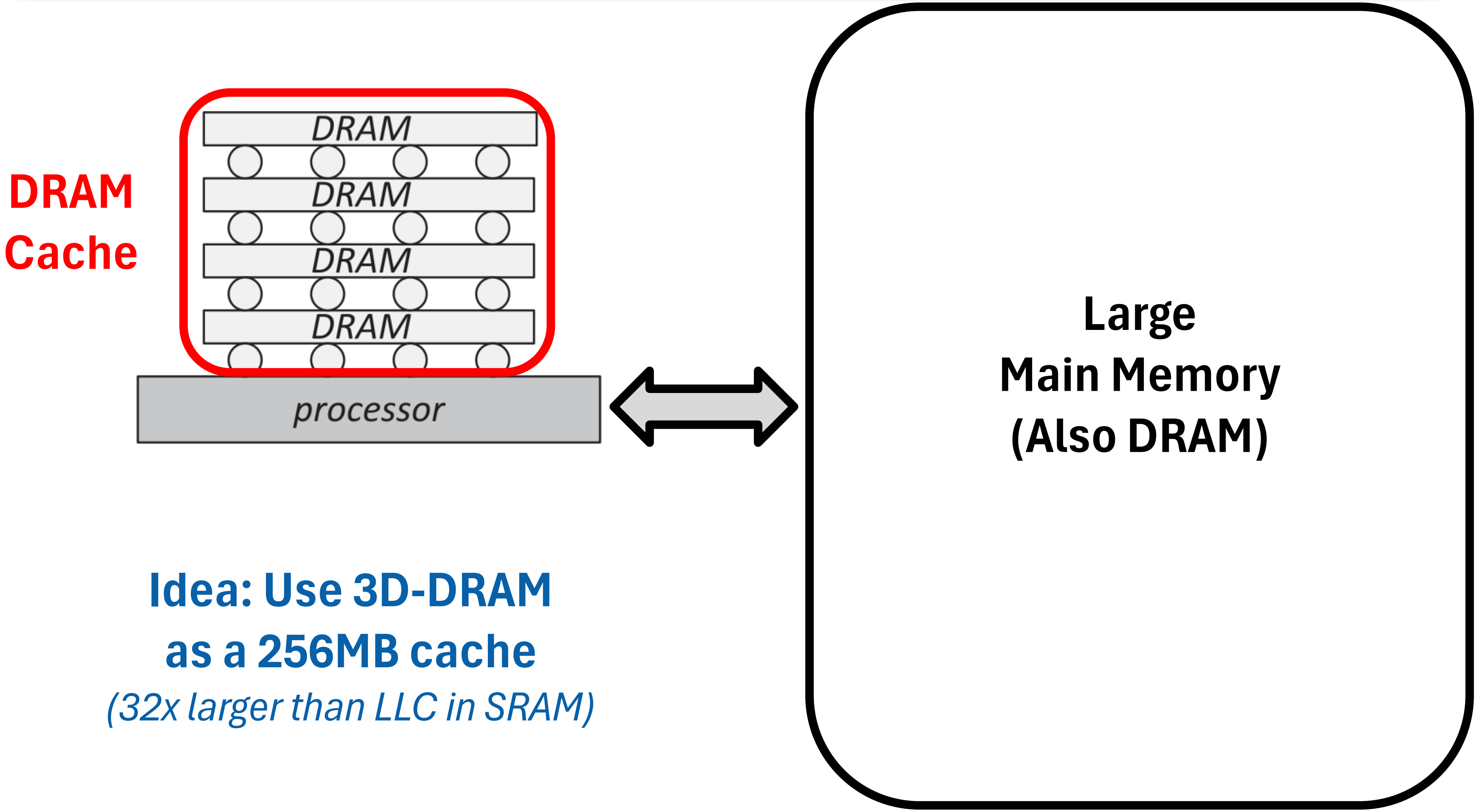
50 Years of Microprocessor Trend Data



We move to here

“Fundamental Latency Trade-offs in Architecting DRAM Caches”

Moinuddin Qureshi, Gabriel Loh 2012

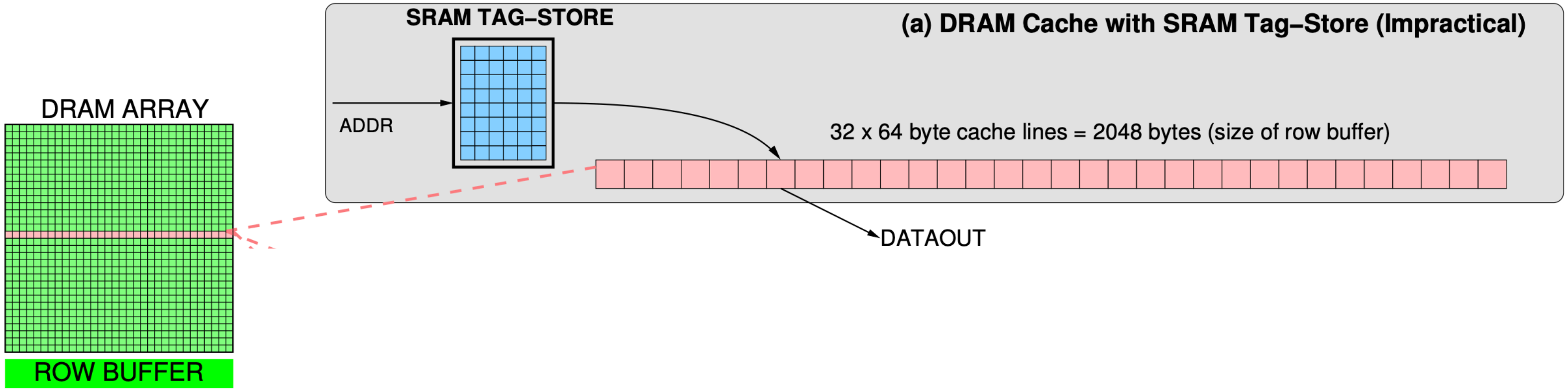


Idea: Use 3D-DRAM as a 256MB cache

(32x larger than LLC in SRAM)

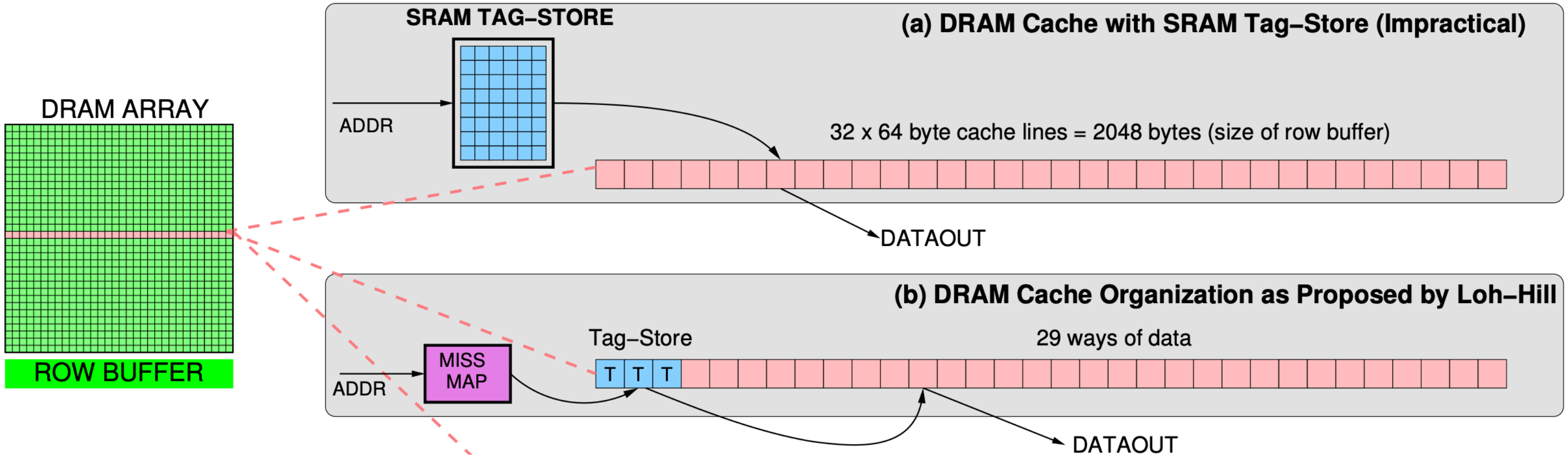
The “Tag” Problem

- **Huge cache → Huge tag array (one tag per cache line)**
  - Where to keep tags? ☹


SRAM TAG–STORE
(a) DRAM Cache with SRAM Tag–Store (Impractical)
DRAM ARRAY
ADDR
32 x 64 byte cache lines = 2048 bytes (size of row buffer)
DATAOUT
ROW BUFFER

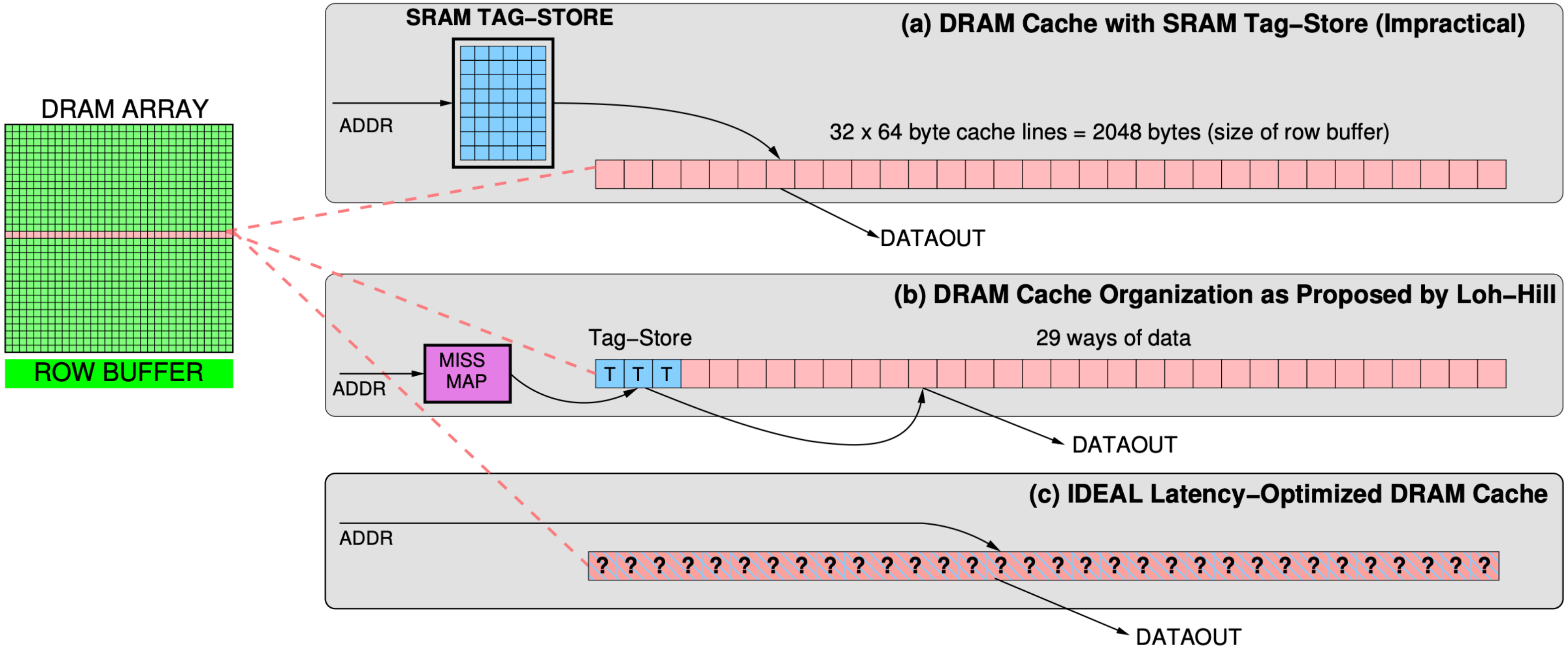
The “Tag” Problem

- Huge cache → Huge tag array
  - Where to keep tags? ☹


DRAM ARRAY
ROW BUFFER
SRAM TAG–STORE
(a) DRAM Cache with SRAM Tag–Store (Impractical)
ADDR
32 x 64 byte cache lines = 2048 bytes (size of row buffer)
DATAOUT
(b) DRAM Cache Organization as Proposed by Loh–Hill
Tag–Store
29 ways of data
MISS MAP
ADDR
T
T
T
DATAOUT

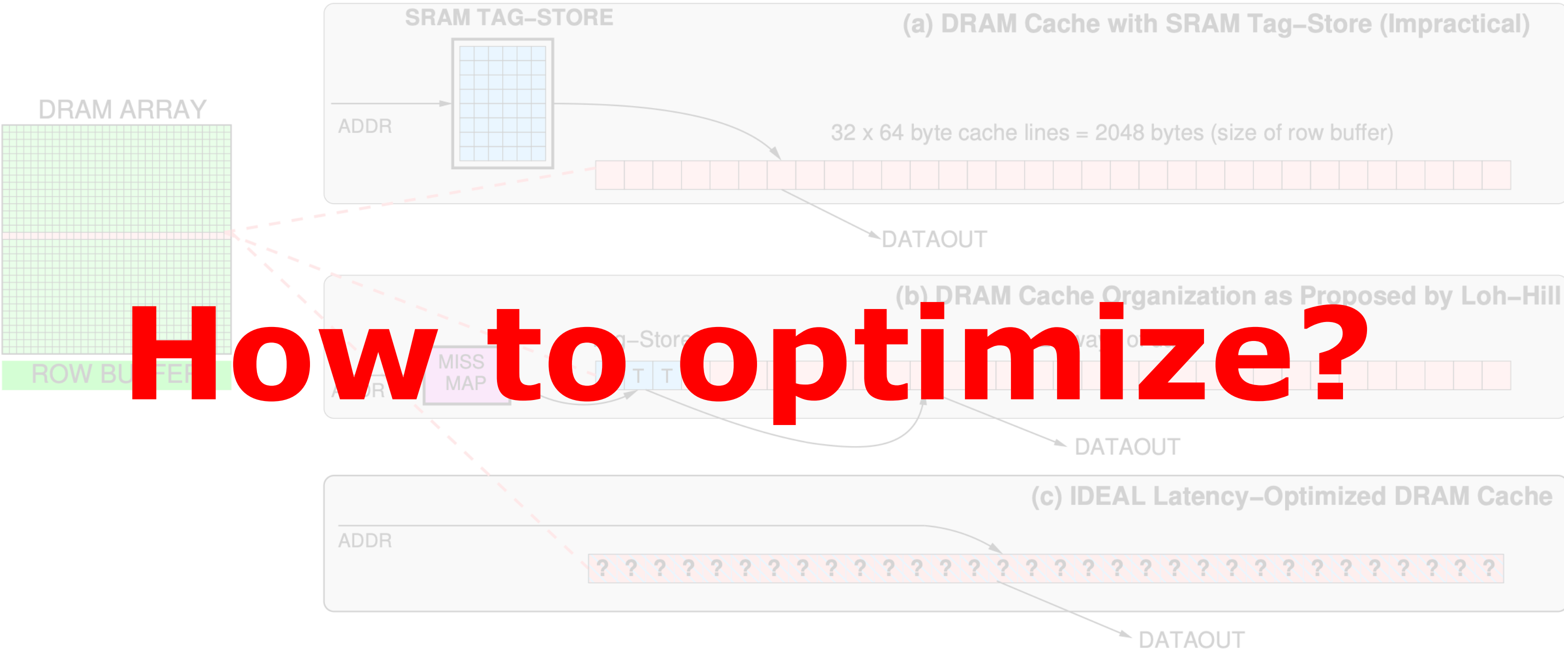
The “Tag” Problem

- **Huge cache → Huge tag array**
  - Where to keep tags? ☹


DRAM ARRAY
ROW BUFFER
SRAM TAG–STORE
(a) DRAM Cache with SRAM Tag–Store (Impractical)
ADDR
32 x 64 byte cache lines = 2048 bytes (size of row buffer)
DATAOUT
(b) DRAM Cache Organization as Proposed by Loh–Hill
Tag–Store
29 ways of data
MISS MAP
ADDR
T T T
DATAOUT
(c) IDEAL Latency–Optimized DRAM Cache
ADDR
? ?
DATAOUT

The “Tag” Problem

- Huge cache → Huge tag array
  - Where to keep tags? ☹



SRAM TAG–STORE
(a) DRAM Cache with SRAM Tag–Store (Impractical)
ADDR
32 x 64 byte cache lines = 2048 bytes (size of row buffer)
DATAOUT
DRAM ARRAY
ROW BUFFER
(b) DRAM Cache Organization as Proposed by Loh–Hill
MISS MAP
ADDR
T T
DATAOUT
(c) IDEAL Latency–Optimized DRAM Cache
ADDR
? ?
DATAOUT

How to optimize?

A Different World!

- **Let's assume we have two types of caches, fast and slow**
  - – Fast Cache's access latency = 0.1 of DDR latency
  - – Slow Cache's access latency = 0.5 of DDR latency

- **Now consider an optimization named "A" that**
  - + Increases hit ratio of the cache from 50% to 70%
  - – Increases its access latency by 1.4x

A Different World!

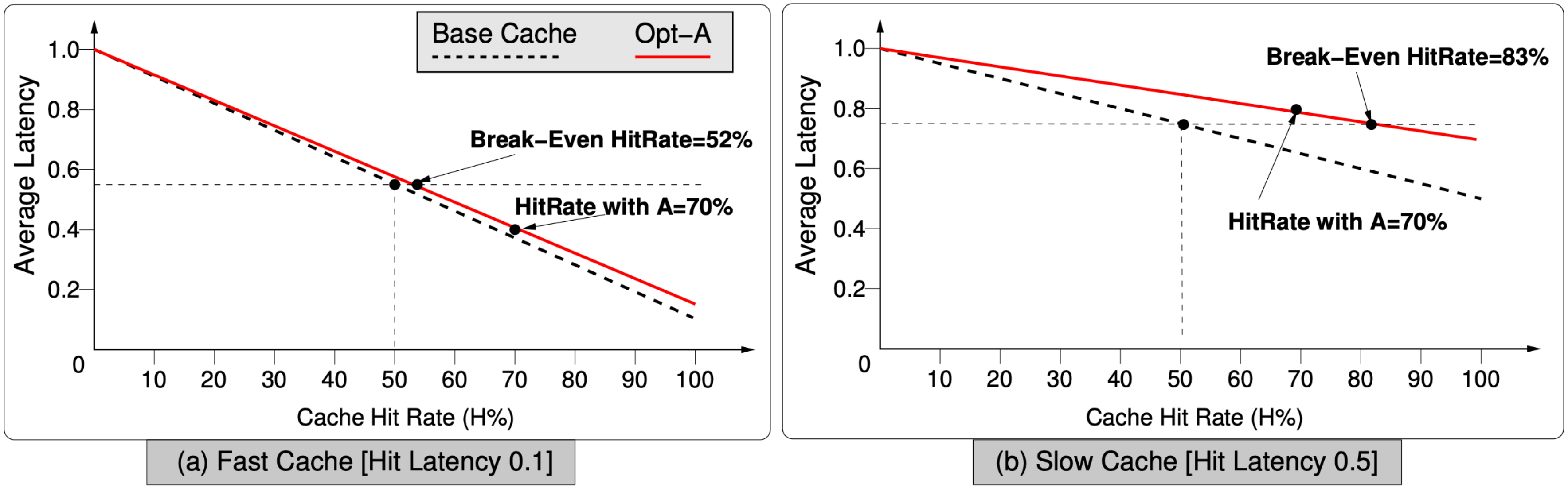


Figure 1: Effectiveness of cache optimizations depend on cache hit latency. Option A increases hit latency by 1.4x and hit-rate from 50% to 70%. (a) For a fast cache, A is highly effective at reducing average latency from 0.55 to 0.4 (b) For a slow cache, A increases average latency from 0.75 to 0.79.

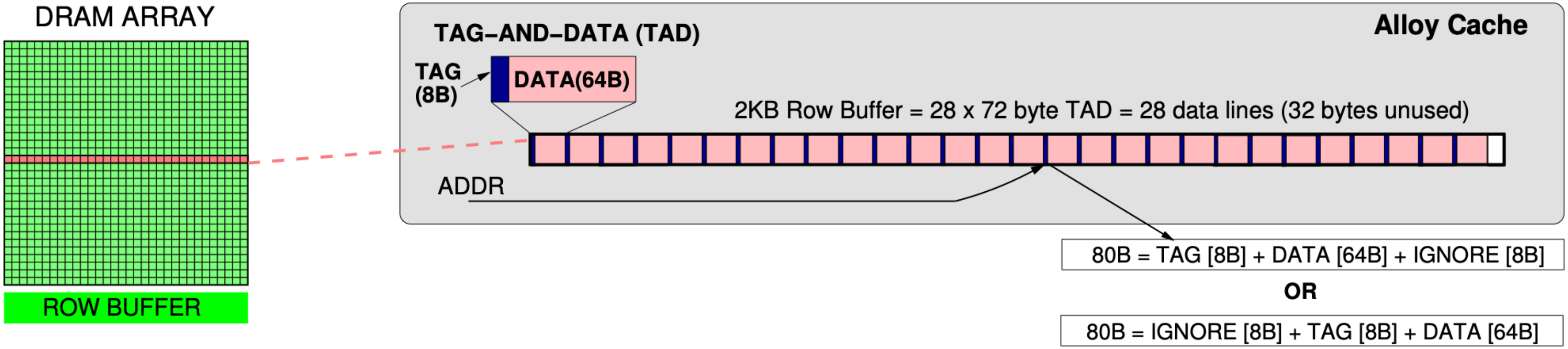
Lesson:

A highly effective optimization for a fast cache may be a bad idea for a slow cache.

Alloy Cache

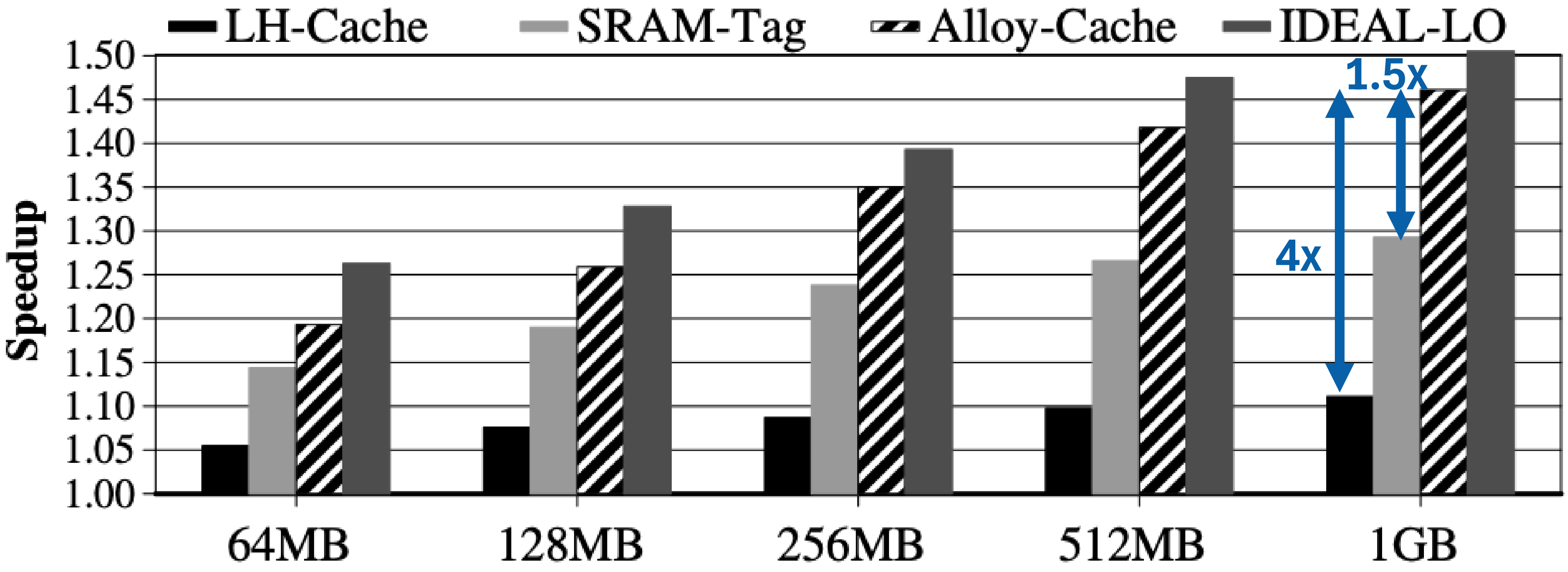
- **DRAM cache is a slow cache: Optimize for Hit Latency**

- **Go with direct-mapped caching!**
  - + Single tag. Speculatively return data with the tag
  - + Data locality → Row buffer locality
  - + No replacement bookkeeping on hits (or misses)


DRAM ARRAY
ROW BUFFER
TAG–AND–DATA (TAD)
Alloy Cache
TAG
(8B)
DATA(64B)
2KB Row Buffer = 28 x 72 byte TAD = 28 data lines (32 bytes unused)
ADDR
80B = TAG [8B] + DATA [64B] + IGNORE [8B]
OR
80B = IGNORE [8B] + TAG [8B] + DATA [64B]

Performance of Alloy Cache

On 8 SPEC benchmarks that would benefit most from perfect caching

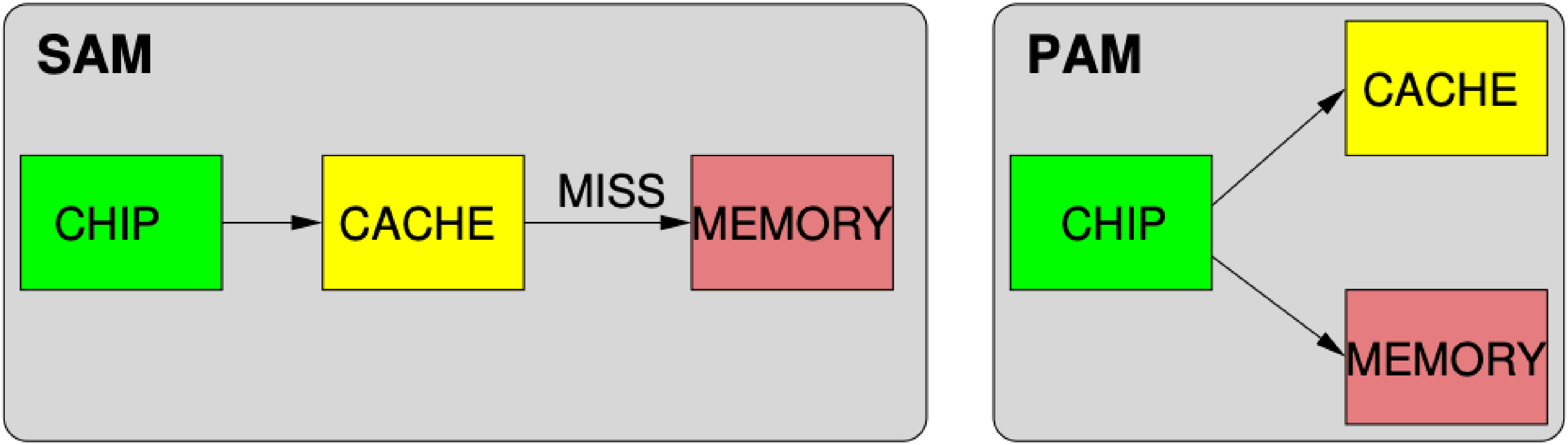


DRAM Cache Size

Note

Storage overhead at 1GB: SRAM-Tag 96MB vs. Alloy-Cache 1KB

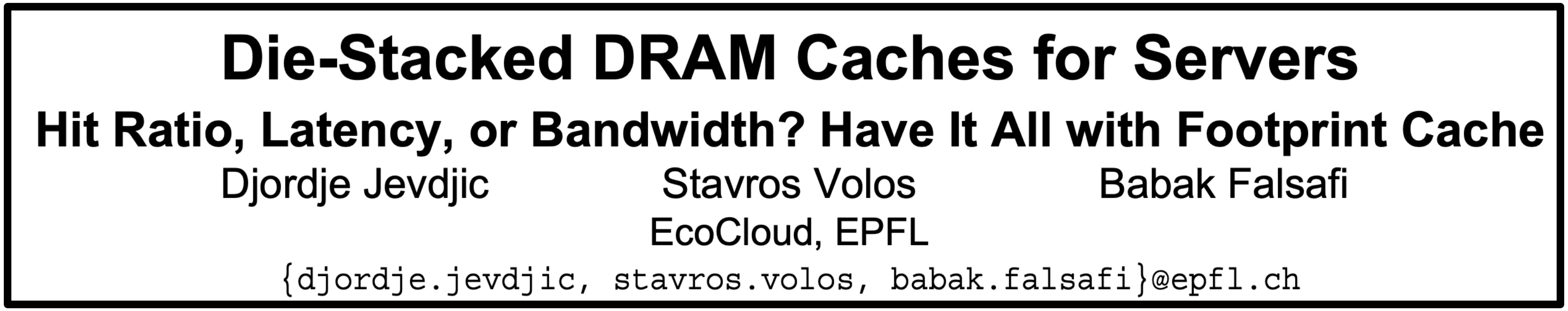
Serial v. Parallel Access Modes



- **PAM: Reduce miss latency by speculatively fetching from Memory**
  – But wastes memory bandwidth on cache hit

- **Alloy-Cache uses a simple (1 cycle) predictor to choose SAM or PAM**
  + Achieves close to PAM's miss latency and SAM's bandwidth usage

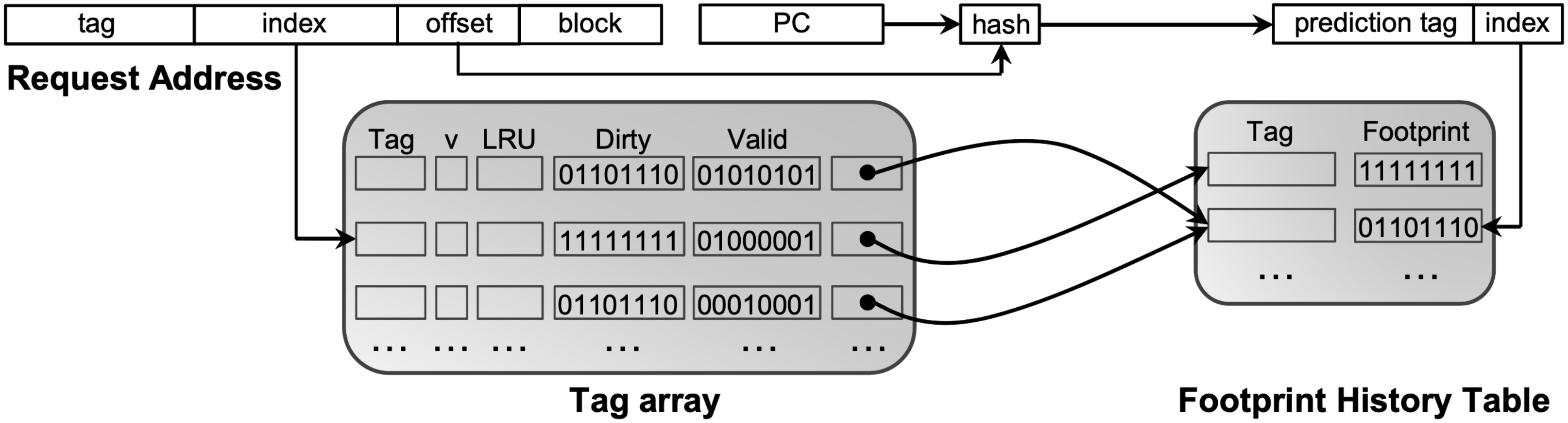
Page-based DRAM Caches



Die-Stacked DRAM Caches for Servers

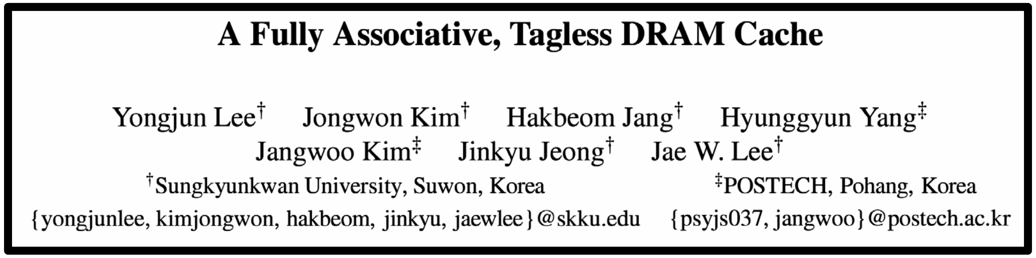
Hit Ratio, Latency, or Bandwidth? Have It All with Footprint Cache

Djordje Jevdjic   Stavros Volos   Babak Falsafi
EcoCloud, EPFL
{djordje.jevdjic, stavros.volos, babak.falsafi}@epfl.ch



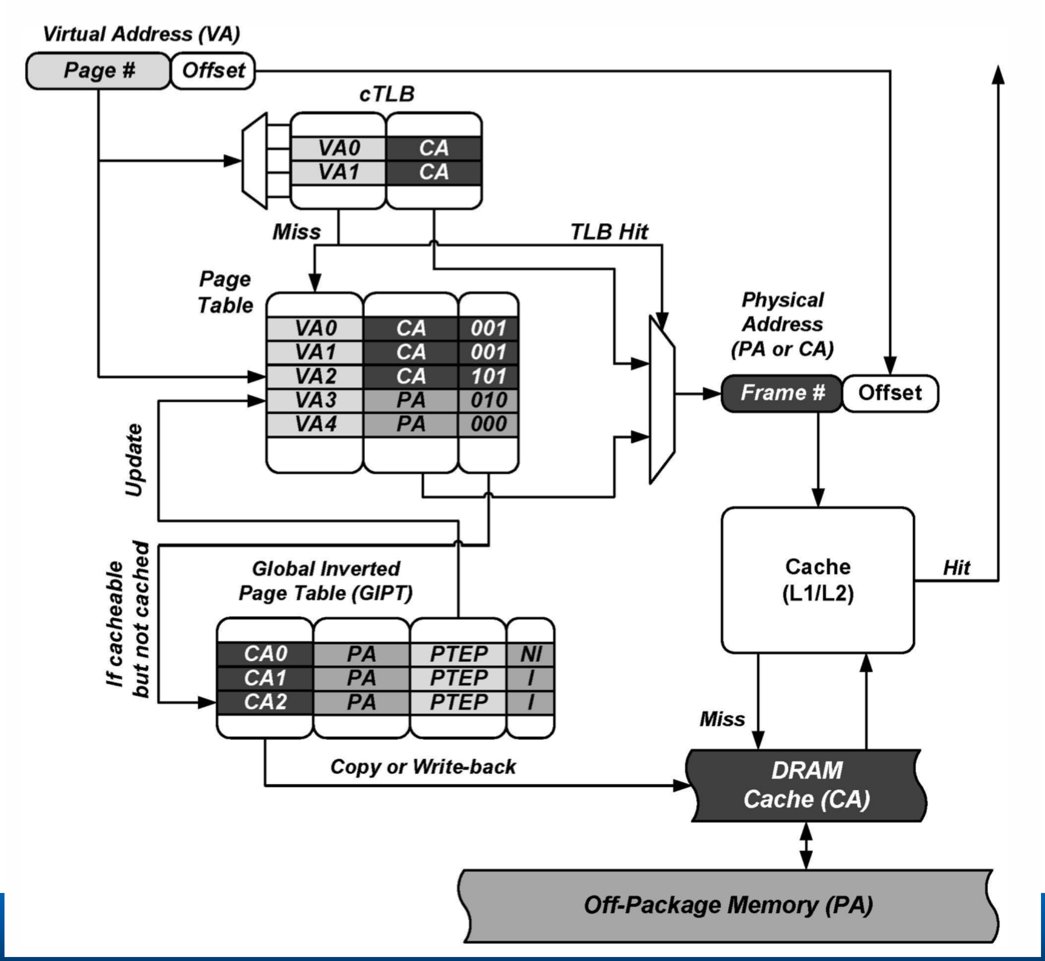
[ISCA'13]

Page-table-based DRAM Caches



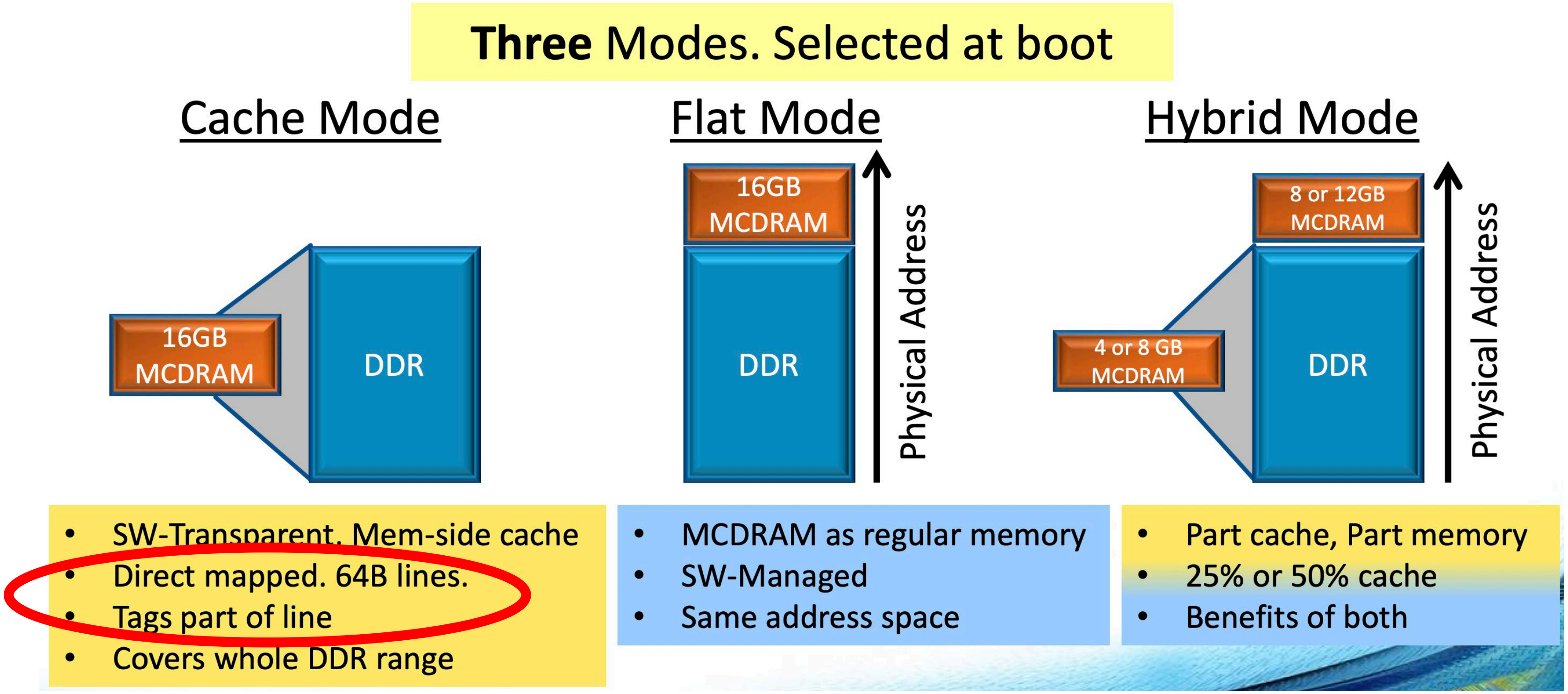
A Fully Associative, Tagless DRAM Cache

Yongjun Lee[†] Jongwon Kim[†] Hakbeom Jang[†] Hyunggyun Yang[‡]
Jangwoo Kim[‡] Jinkyu Jeong[†] Jae W. Lee[†]
[†]Sungkyunkwan University, Suwon, Korea [‡]POSTECH, Pohang, Korea
{yongjunlee, kimjongwon, hakbeom, jinkyu, jaewlee}@skku.edu {psyjs037, jangwoo}@postech.ac.kr



[ISCA'15]

Intel’s Knights Landing (2016)



Three Modes. Selected at boot

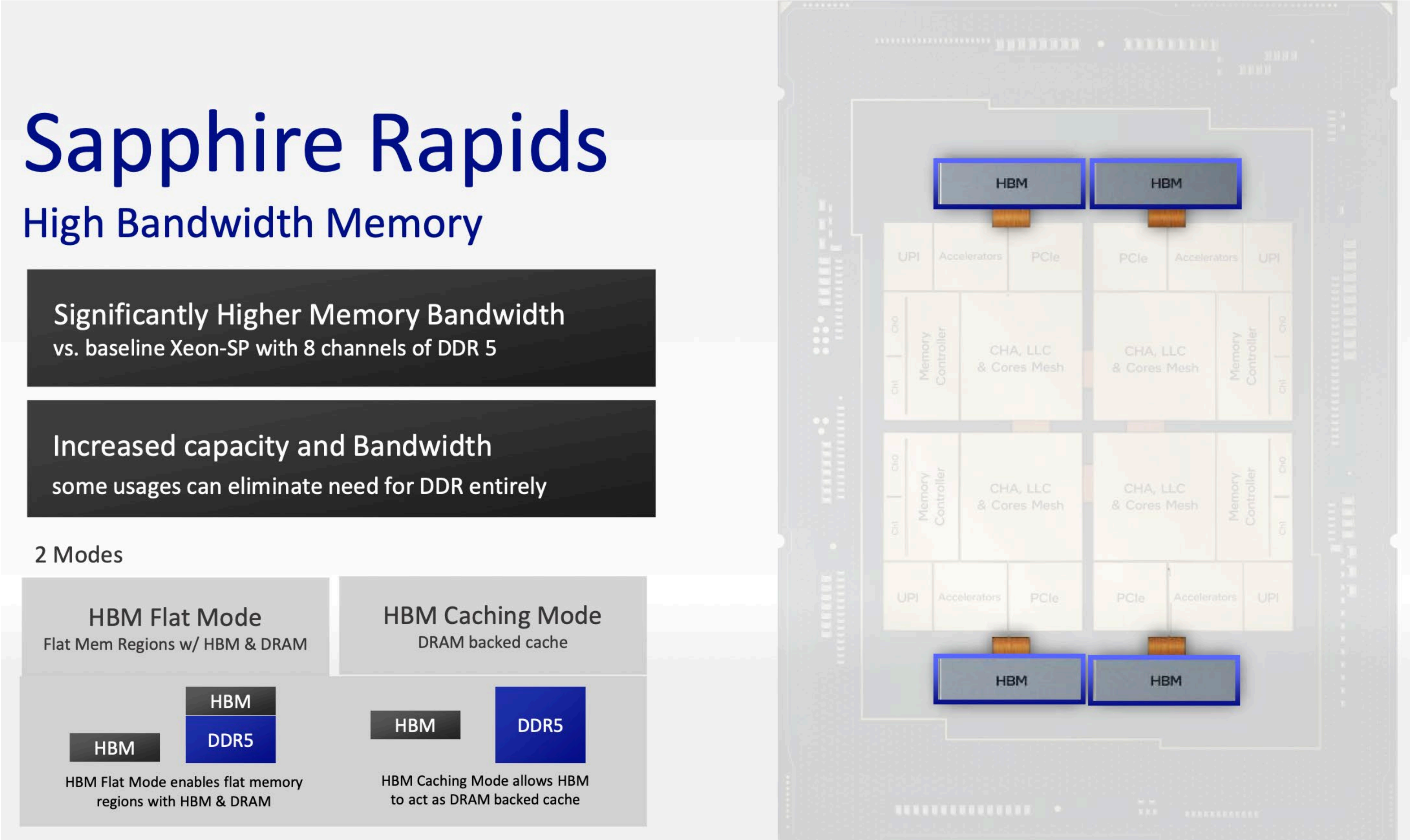
- SW-Transparent, Mem-side cache
- Direct mapped. 64B lines.
- Tags part of line
- Covers whole DDR range

- MCDRAM as regular memory
- SW-Managed
- Same address space

- Part cache, Part memory
- 25% or 50% cache
- Benefits of both

3D DRAM Now

- **Intel's Sapphire Rapids (2023): 64GB 3D DRAM**



Sapphire Rapids

High Bandwidth Memory

Significantly Higher Memory Bandwidth
vs. baseline Xeon-SP with 8 channels of DDR 5

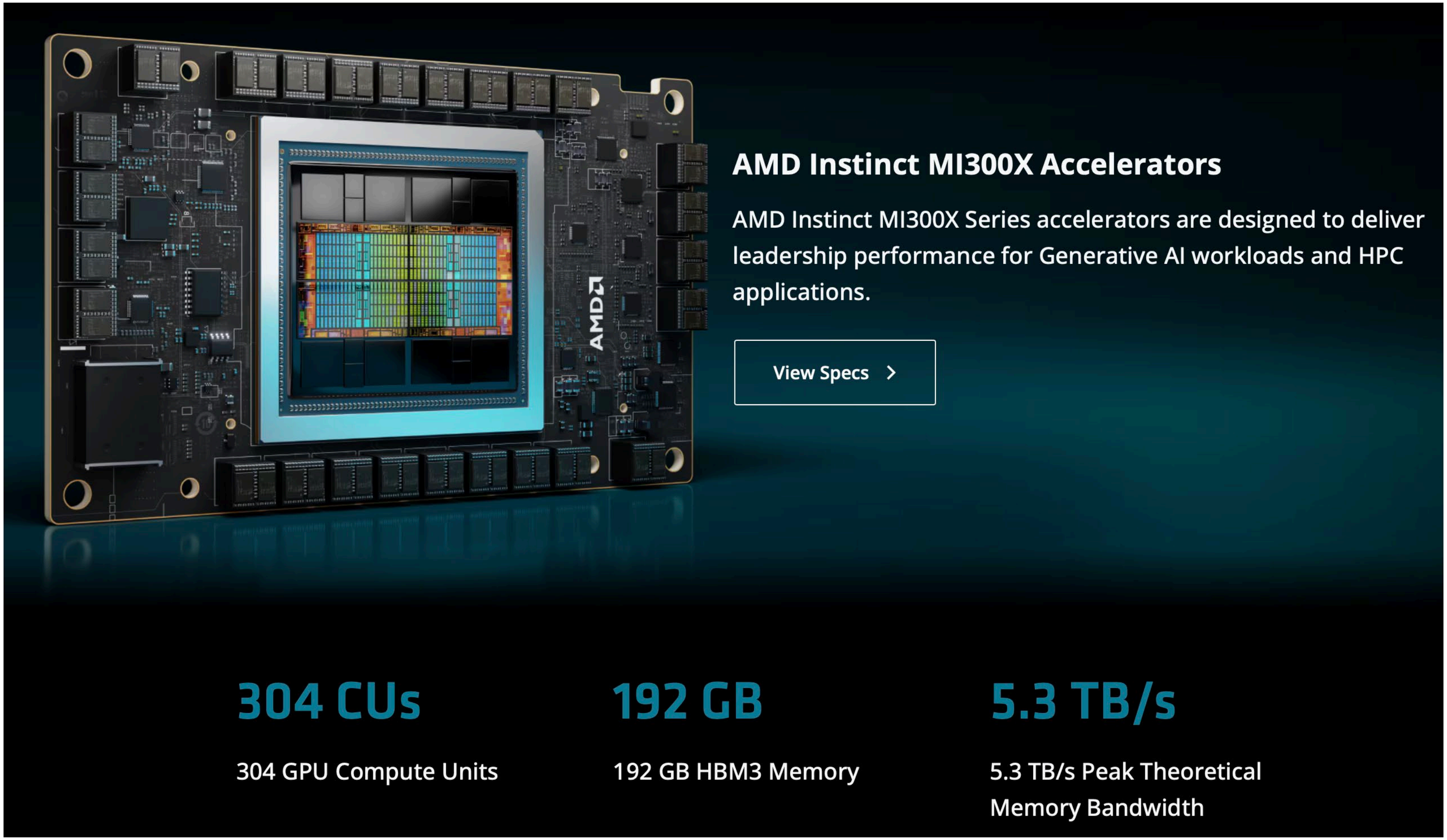
Increased capacity and Bandwidth
some usages can eliminate need for DDR entirely

2 Modes

HBM Flat Mode	HBM Caching Mode
Flat Mem Regions w/ HBM & DRAM	DRAM backed cache
HBM Flat Mode enables flat memory regions with HBM & DRAM	HBM Caching Mode allows HBM to act as DRAM backed cache

3D DRAM Now

- AMD’s Instinct MI300X



AMD Instinct MI300X Accelerators

AMD Instinct MI300X Series accelerators are designed to deliver leadership performance for Generative AI workloads and HPC applications.

304 CUs

304 GPU Compute Units

192 GB

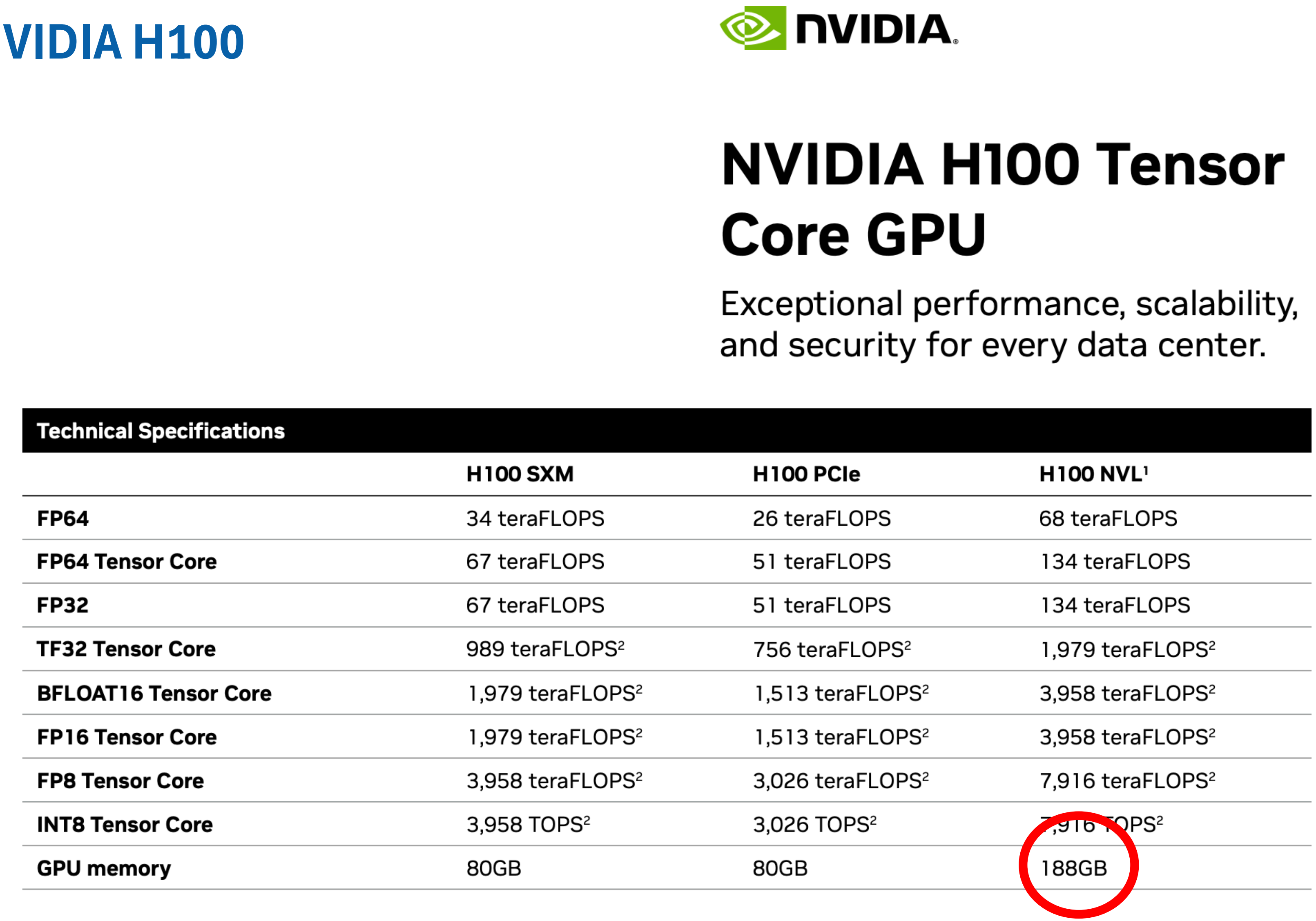
192 GB HBM3 Memory

5.3 TB/s

5.3 TB/s Peak Theoretical Memory Bandwidth

3D DRAM Now

- NVIDIA H100



NVIDIA

NVIDIA H100 Tensor Core GPU

Exceptional performance, scalability, and security for every data center.

Technical Specifications

	H100 SXM	H100 PCIe	H100 NVL[1]
FP64	34 teraFLOPS	26 teraFLOPS	68 teraFLOPS
FP64 Tensor Core	67 teraFLOPS	51 teraFLOPS	134 teraFLOPS
FP32	67 teraFLOPS	51 teraFLOPS	134 teraFLOPS
TF32 Tensor Core	989 teraFLOPS[2]	756 teraFLOPS[2]	1,979 teraFLOPS[2]
BFLOAT16 Tensor Core	1,979 teraFLOPS[2]	1,513 teraFLOPS[2]	3,958 teraFLOPS[2]
FP16 Tensor Core	1,979 teraFLOPS[2]	1,513 teraFLOPS[2]	3,958 teraFLOPS[2]
FP8 Tensor Core	3,958 teraFLOPS[2]	3,026 teraFLOPS[2]	7,916 teraFLOPS[2]
INT8 Tensor Core	3,958 TOPS[2]	3,026 TOPS[2]	7,916 TOPS[2]
GPU memory	80GB	80GB	188GB

To Read for Monday

"Designing Vertical Processors in Monolithic 3D"
Bhargava Gopireddy, Josep Torrellas 2019

Optional Further Reading:

"NOMAD: Enabling Non-blocking OS-managed DRAM Cache via Tag-Data Decoupling"
Youngin Kim, Hyeonjin Kim, William Song 2023