DAM: Differentiated Access Memory

Philip Levis and Caroline Trippel

DAM/MemoryDAX First Research Retreat

July 1-2, 2025

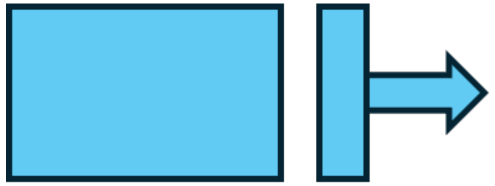
In One Slide

- Starting year 2 of a 7 year project to define the next generation of computer architectures, based on differentiated access memories
- Memory is the bottleneck today
  - Further gains in performance require transformative changes to memory
  - Memory needs to specialize: differentiated access memories (DAM)
- Differentiated access memories raise many open research questions
  - What are the application patterns that can leverage differentiation?
  - How will software use the memories (explicit vs. implicit placement)?
  - Which memories and their tradeoffs (energy, density, latency, retention, endurance, throughput, etc.)?
  - How will these memories be packaged and composed in larger systems?

Application Needs Vary Greatly

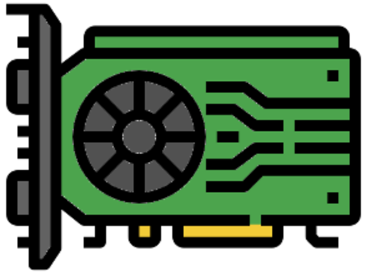
Data Analytics

Streams of data
Write-once, read-once
Filters (scans)
Joins (random access)



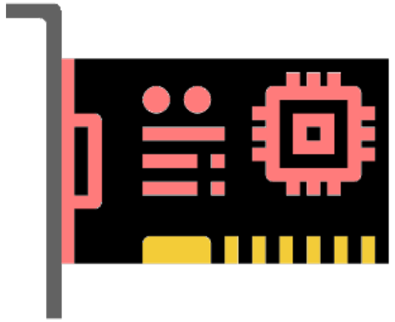
Append-Mostly Databases

Write once
Mostly append
Read many times
Scans
Random access



Machine Learning Accelerator

Blocked operations
Sparse accesses
Read multiple times
Write many times
Read/write
Throughput (training)
Latency (inference)



High-Speed Networking

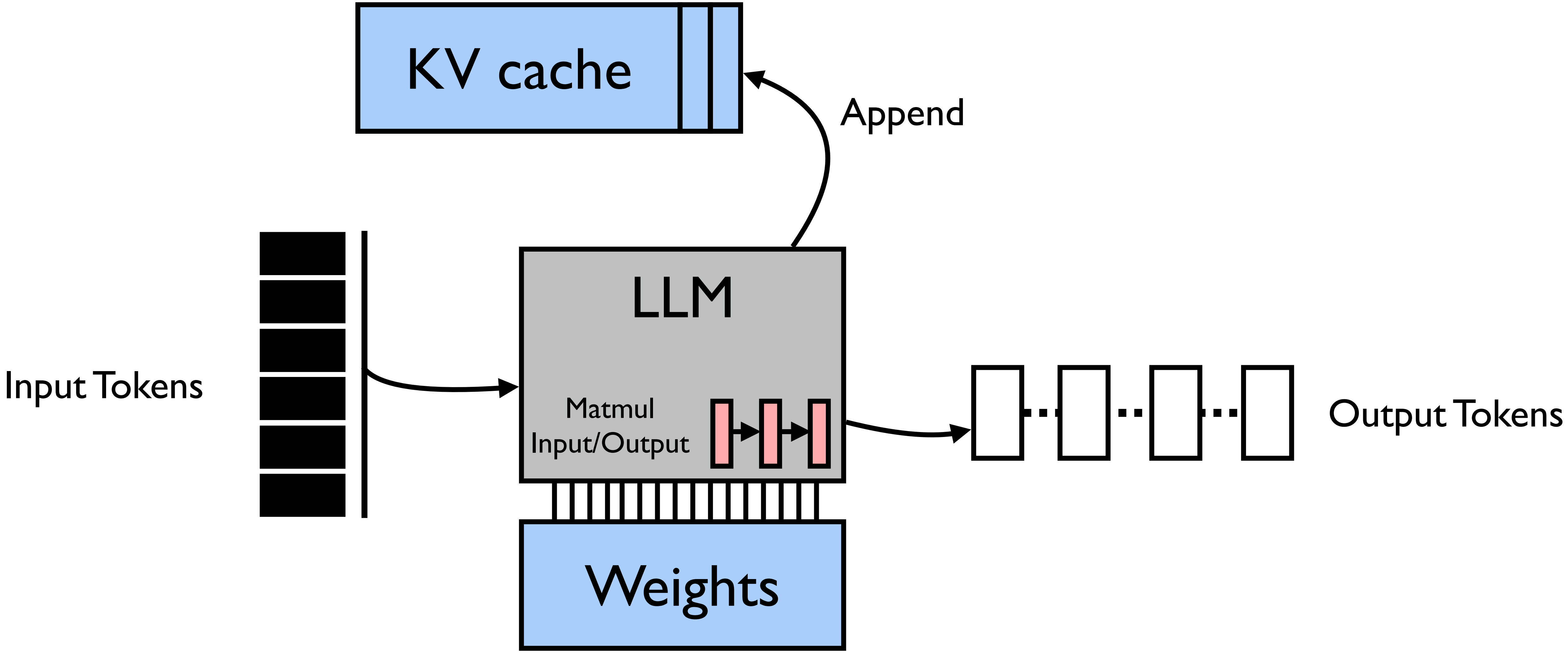
Ultra-low latency
Header processing
Packet-oriented
Read once
Write once

Actually Many Kinds of Memory...

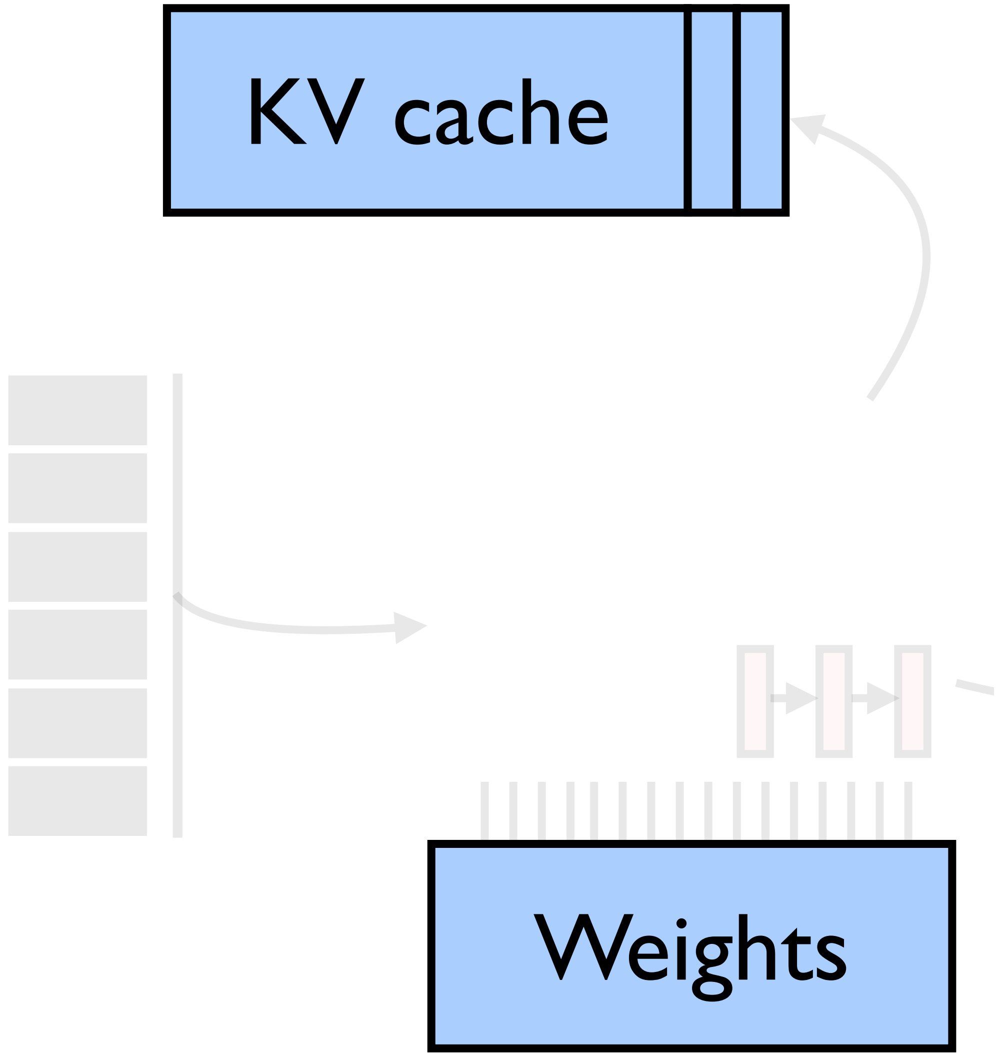
DRAM
SRAM
MRAM
RRAM
FRAM
PCM
Flash
GC
HGC
FeFet
OS-OS

	Energy/power (active)		Energy/ power (standby)	Access time, latency		endurance	retention	Density (capacity)	On-logic chip integration	
	read	Write		read	Write				One layer (possible)	Multiple layers for density
High	RRAM, MRAM, PCM, FeRAM,	RRAM, MRAM, PCM, Flash	DRAM	Flash	Flash	DRAM, SRAM, OS-OS GC, HGC	Flash, RRAM, MRAM, PCM, FeFET, FeRAM	Flash, FeFET	MRAM, PCM, RRAM, FeRAM,	FeFET, OS-OS GC
Medium	DRAM	DRAM, FeRAM	SRAM	RRAM, PCM, FeFET, FeRAM	RRAM, PCM, FeFET, FeRAM	FeRAM, MRAM	OS-OS GC, HGC	DRAM, FeRAM, OS-OS GC	DRAM	
Medium low	FeFET, OS-OS GC	FeFET	HGC, OS-OS GC	DRAM, MRAM, OS-OS GC	DRAM, OS-OS GC, HGC	PCM, RRAM	DRAM	HGC, MRAM, RRAM, PCM,		
low	SRAM, HGC	SRAM, HGC, OS-OS GC	RRAM, MRAM, PCM, FeFET, FeRAM, Flash	SRAM, HGC	SRAM	Flash, FeFET		SRAM	Flash	Flash, DRAM

Zooming In: LLM Inference

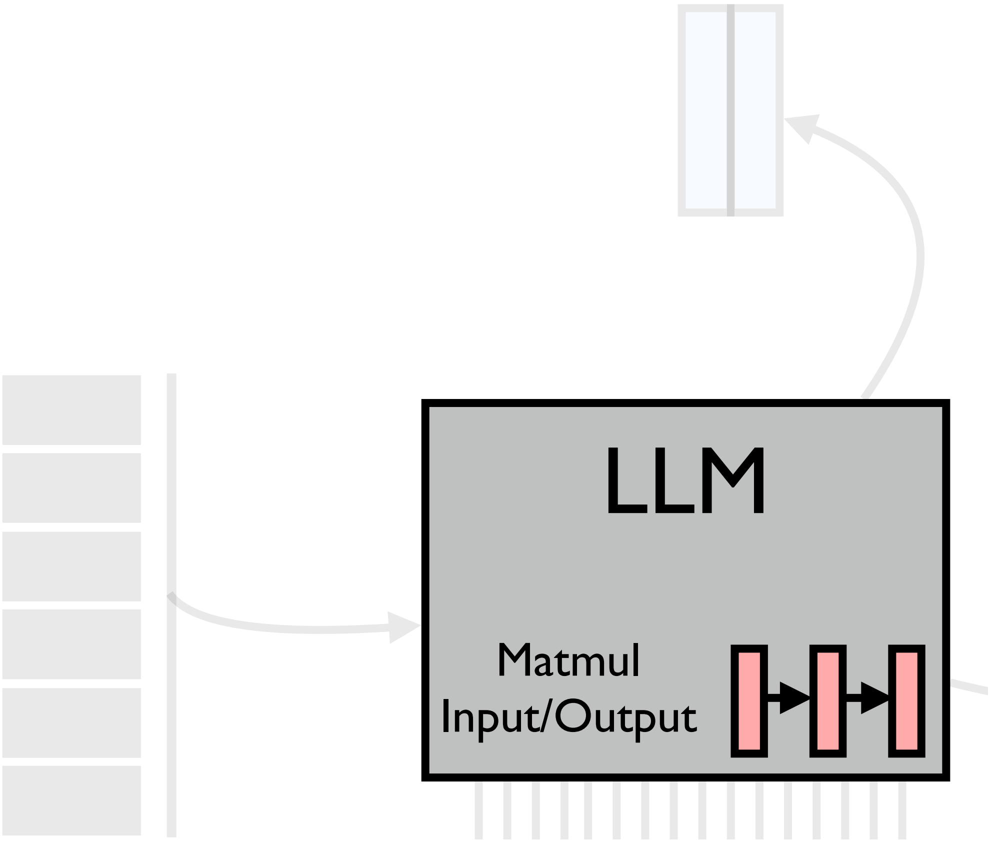


Write Rarely, Read Often



- Both the KV cache and model weights are read heavy
  - Read the cache N times for N output tokens, append cache entries
  - Write parameters on model load the model, read each time it executes

Write Once, Read Once



- Activations: inputs and outputs to matrix multiplications
  - Write the values once
  - Quickly read them once

Five Types of Memory

Structure
Benefits
Drawbacks
Uses

SRAM

	SRAM
Structure	6T
Benefits	Fast Easy to integrate Low static power
Drawbacks	Sparse
Uses	Fast read/write caches

DRAM

	SRAM	DRAM
Structure	6T	1T1C
Benefits	Fast Easy to integrate Low static power	Dense
Drawbacks	Sparse	Hard to integrate High power
Uses	Fast read/write caches	Large, random-access RW data

Block Flash

	SRAM	DRAM	Block Flash
Structure	6T	1T1C	1G
Benefits	Fast Easy to integrate Low static power	Dense	HUGE Capacity
Drawbacks	Sparse	Hard to integrate High power	No logic Low endurance Expensive, slow erases Block access Low bandwidth
Uses	Fast read/write caches	Large, random-access RW data	Large, read-mostly data

Long-term RAM (LtRAM)

	SRAM	DRAM	Block Flash	LtRAM (long-term RAM)
Structure	6T	1T1C	1G	FeRAM, MRAM, RRAM
Benefits	Fast Easy to integrate Low static power	Dense	HUGE Capacity	Dense Low Read Energy
Drawbacks	Sparse	Hard to integrate High power	No logic Low endurance Expensive, slow erases Block access Low bandwidth	Writes are slow and high energy Limited endurance
Uses	Fast read/write caches	Large, random-access RW data	Large, read-mostly data	Write rarely (static caches)

Short-term RAM (StRAM)

	SRAM	DRAM	Block Flash	LtRAM (long-term RAM)	StRAM (short-term RAM)
Structure	6T	1T1C	1G	FeRAM, MRAM, RRAM	Gain Cells (2T, 3T)
Benefits	Fast Easy to integrate Low static power	Dense	HUGE Capacity	Dense Low Read Energy	Dense Low Energy
Drawbacks	Sparse	Hard to integrate High power	No logic Low endurance Expensive, slow erases Block access Low bandwidth	Writes are slow and high energy Limited endurance	Active research Refresh power
Uses	Fast read/write caches	Large, random-access RW data	Large, read-mostly data	Write rarely (static caches)	Write-and-read

LtRAM: Long-term RAM

- Stores data for seconds to days
- High read:write ratio
- Lower read energy than DRAM, higher write energy
- Often non-volatile
- Server uses: copy-on-write memory caches, code pages, cold memory
- Inference uses: model weights, KV caches
- Example technologies: MRAM, RRAM, FeRAM, 3DXP

StRAM: Short-term RAM

- Stores data for microseconds to seconds
- Write:read ratio $\approx 1 : 1$
- Higher density than SRAM
- Lower write energy than SRAM, tunable retention
- Server uses: on-CPU caches, DMA memory, queues and buffers
- Inference uses: model activations, program variables
- Technology: gain-cell RAM (GCRAM)

Three Areas of Exploration

Inference Accelerators

- Performance critical
- Cooling and power infrastructure
- Dense compositions, many forms of parallelism

Mobile SoCs

- Energy-limited
- Cooling-limited
- Cost-sensitive
- Switch between many models/applications

Server CPUs

- Diverse applications
- Legacy applications
- Multi-tenancy
- Latency-optimized architectures

Three Areas of Exploration

Inference Accelerators

LtRAM

Model weights, KV caches

StRAM

Model activations

Mobile SoCs

LtRAM

Model weights, KV caches
Code and static resources

StRAM

Model activations
On-chip caches

Server CPUs

LtRAM

Code pages, read-only data
Cold memory/data

StRAM

On-chip caches
DMA buffers

Complete Systems

- Each system class has its own unique workloads, performance tradeoffs, and engineering constraints
- Designing requires a *whole system* approach
- Especially necessary for mobile and datacenter inference accelerators
  - Fast moving field
  - Much less understood design space

Open Questions in Devices and Circuits

- What memory technologies should LtRAM and StRAM use, how should they be tuned?
- Given the heterogeneity of memory and compute, how many voltage domains will be needed and what are the implications for voltage regulation?
- Given different scaling limits and voltage limits, when should memories be packaged versus manufactured simultaneously on-die with compute?

Device Types and Circuits

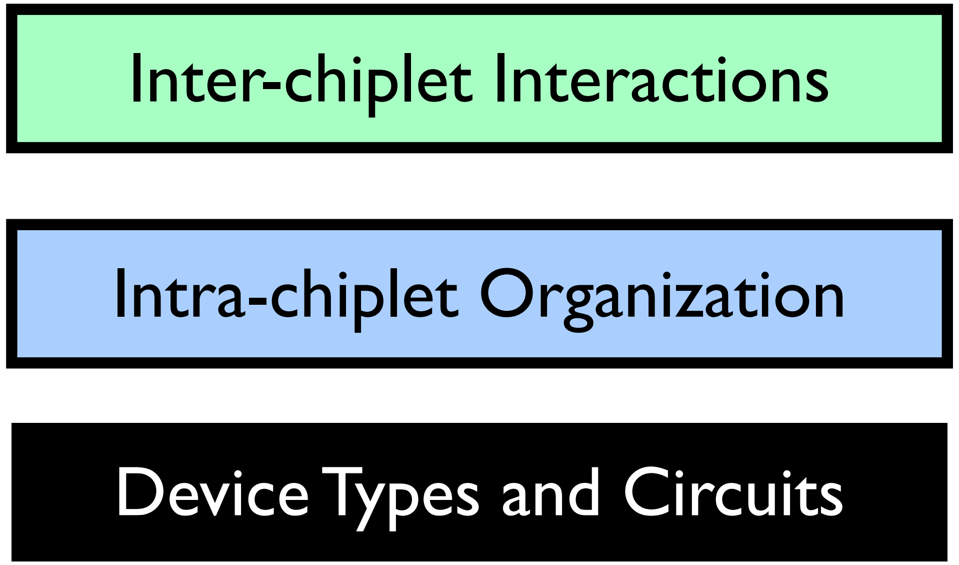
Intra-chiplet Organization Questions

- Given the complex access patterns of modern and future models, how should inference accelerators provision LtRAM and StRAM?
- For each memory technology, what capacity should be on-die with compute, integrated in 3D, or integrated in 2.5D?
- What are the thermal constraints for 3D integration and what circuit, packaging, architectural, or software techniques will alleviate them?

Intra-chiplet Organization

Device Types and Circuits

Inter-chiplet Interactions Questions



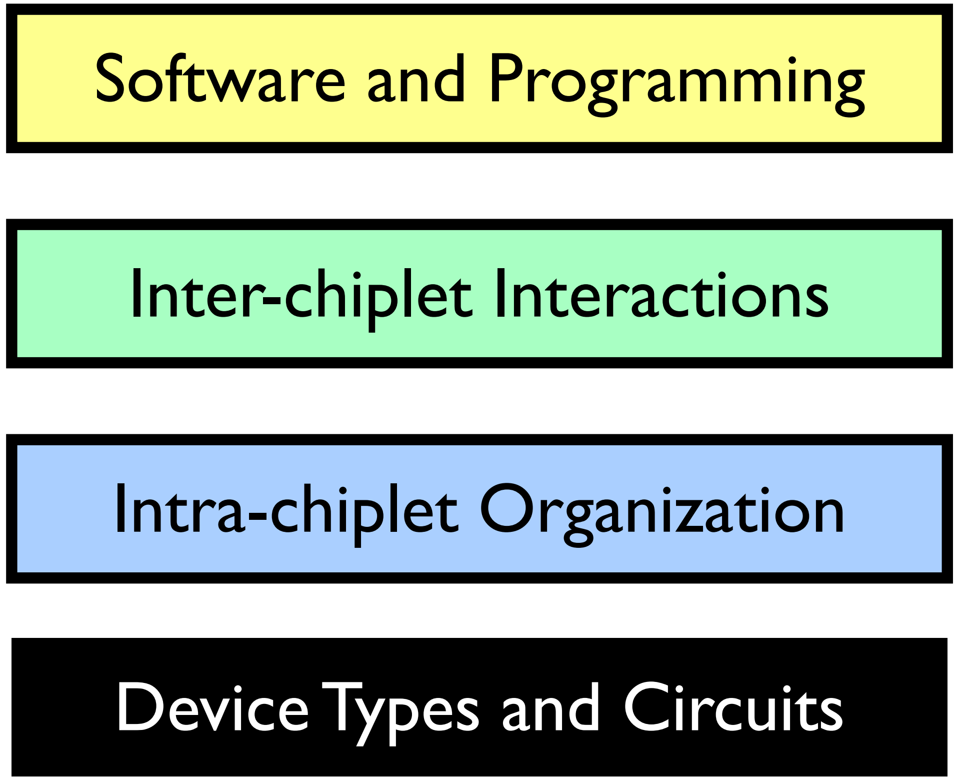
Inter-chiplet Interactions

Intra-chiplet Organization

Device Types and Circuits

- For each memory technology, what capacity should be on-die with compute, integrated in 3D, or integrated in 2.5D?
- What are the cost-benefit tradeoffs between using shared memory and message passing to coordinate inference computational elements and how should a system provision each?

Software and Programming Questions



Software and Programming

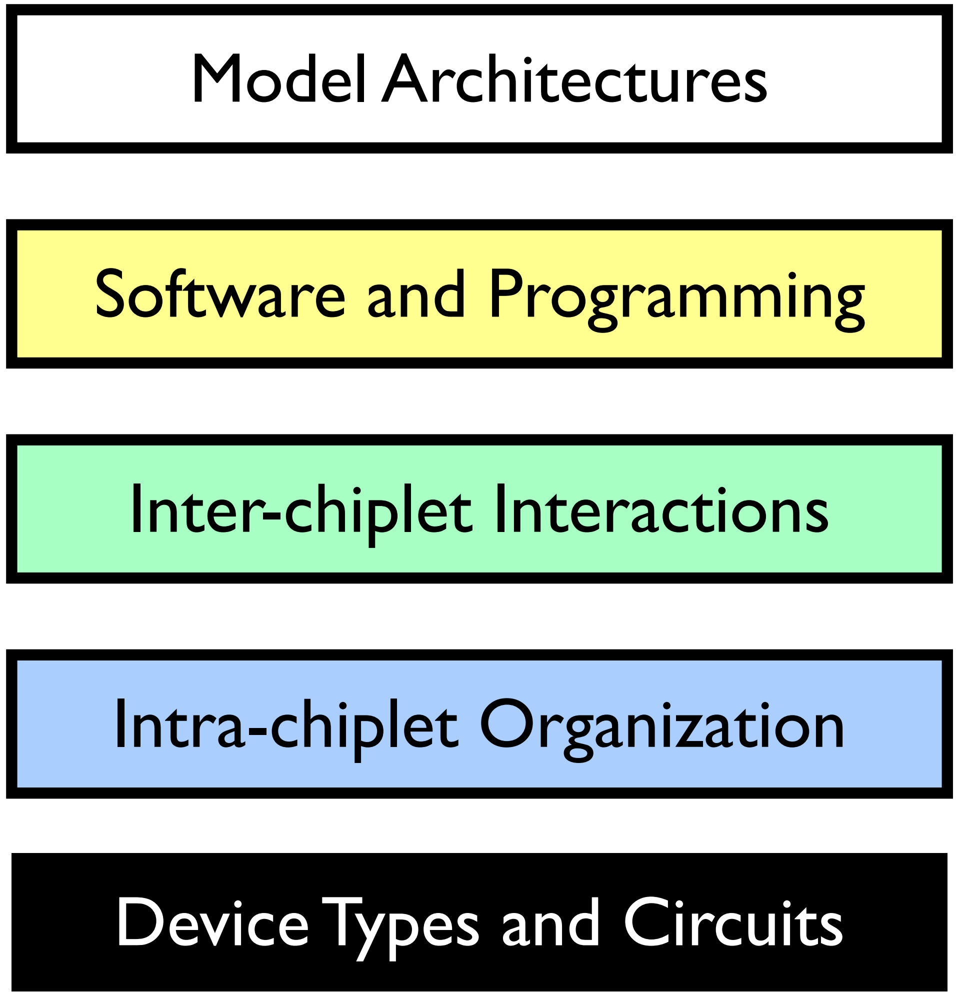
Inter-chiplet Interactions

Intra-chiplet Organization

Device Types and Circuits

- What are the cost-benefit tradeoffs between using shared memory and message passing to coordinate inference computational elements and how should a system provision each?
- When managing endurance and retention in inference workloads, which decisions should be left to hardware, which should be left to software, and what should the coherence protocols be across memories with different retentions?
- Under what circumstances should applications explicitly place data in a memory technology, and when should runtimes use virtual addressing to implicitly place it?
- How will operating systems decide to place pages when memory is no longer tiered and there are multiple dimensions of performance?

Model Architecture Questions



Model Architectures

Software and Programming

Inter-chiplet Interactions

Intra-chiplet Organization

Device Types and Circuits

- What memory use patterns in mobile inference would greatly benefit from new memories?
- Given the complex access patterns of modern and future mobile models, how should inference accelerators provision different memory technologies?

This Retreat

Tuesday

17:00	Registration
18:00	Welcome and Poster Lightning Talks
18:15	Poster Session
20:00	Adjourn

Wednesday

9:00	Project Review
	Monolithic 3-D Integration of Diverse Memories
	Memory Placement in Servers with fitd
10:30	Break
10:40	GainSight: Application-Guided Profiling
	Minions: Cost-efficient Collaboration Between Models
11:30	Lunch and Aquarium
14:00	Storage Class Memory is Dead
	Consistency-Directed Formal Verification
	AI-Boosted Chip Design
	OpenGCRAM: An Open-Source Gain Cell Compiler
	Illusion Scale-up for Massive AI/ML
15:40	Break
16:00	Student Panel
16:45	Closing

Thank You!

EMD Electronics

Google