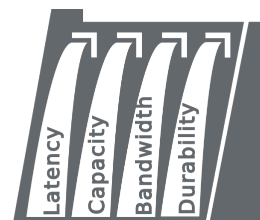
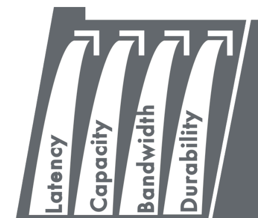
DAM: Differentiated Access Memory

Philip Levis and Caroline Trippel*
DAM/MemoryDAX Second Summer Retreat
June 22, 2026



DAM



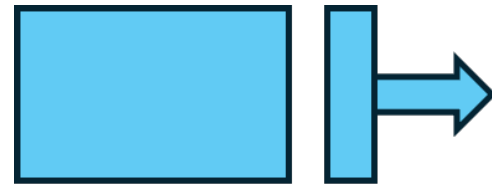
MemoryDAX

In One Slide

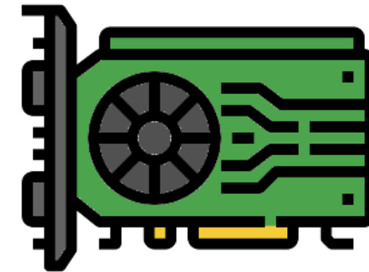
- Year 2 of a 7 year research project to overcome current computing limitations through heterogeneous memories
- Memory is the limiting factor in computing today
  - Significant performance, energy, and cost improvements require transformative changes to memory
  - Memory needs to specialize: differentiated access memories (DAM)
- Differentiated access memories raise many open research questions
  - Which application patterns can leverage differentiation?
  - How will software use the memories (explicit vs. implicit placement)?
  - Which memories and tradeoffs (energy, density, latency, retention, endurance, etc.)?
  - How will these memories be packaged and composed in larger systems?

Application Needs Vary Greatly

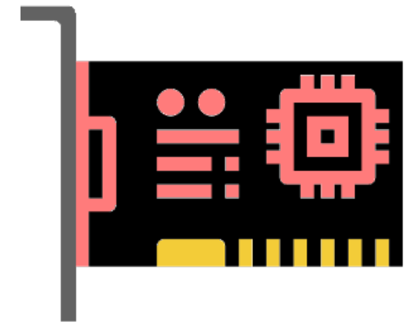
Data Analytics



Append-Mostly Databases



Machine Learning Accelerator



High-Speed Networking

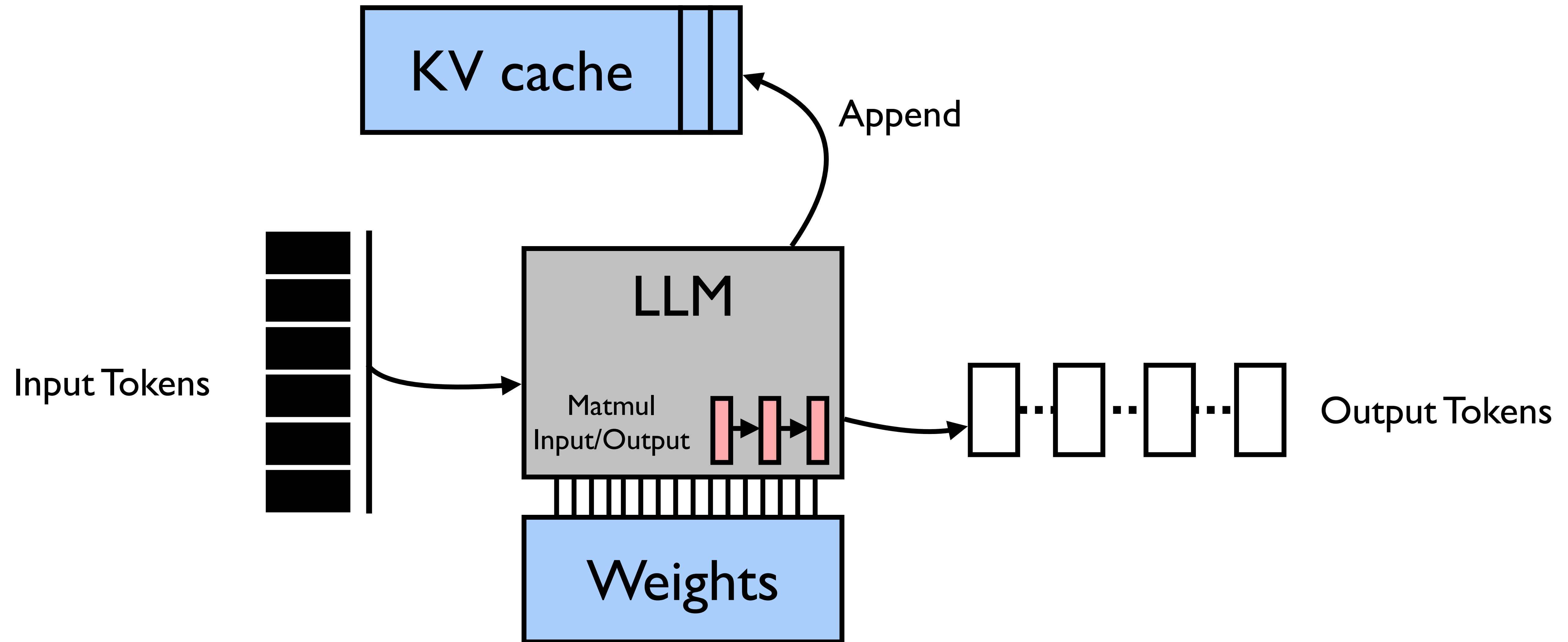
Data Analytics	Append-Mostly Databases	Machine Learning Accelerator	High-Speed Networking
Streams of data Write-once, read-once Filters (scans) Joins (random access)	Write once Mostly append Read many times Scans Random access	Blocked operations Sparse accesses Read multiple times Write many times Read/write Throughput (training) Latency (inference)	Ultra-low latency Header processing Packet-oriented Read once Write once

Actually Many Kinds of Memory...

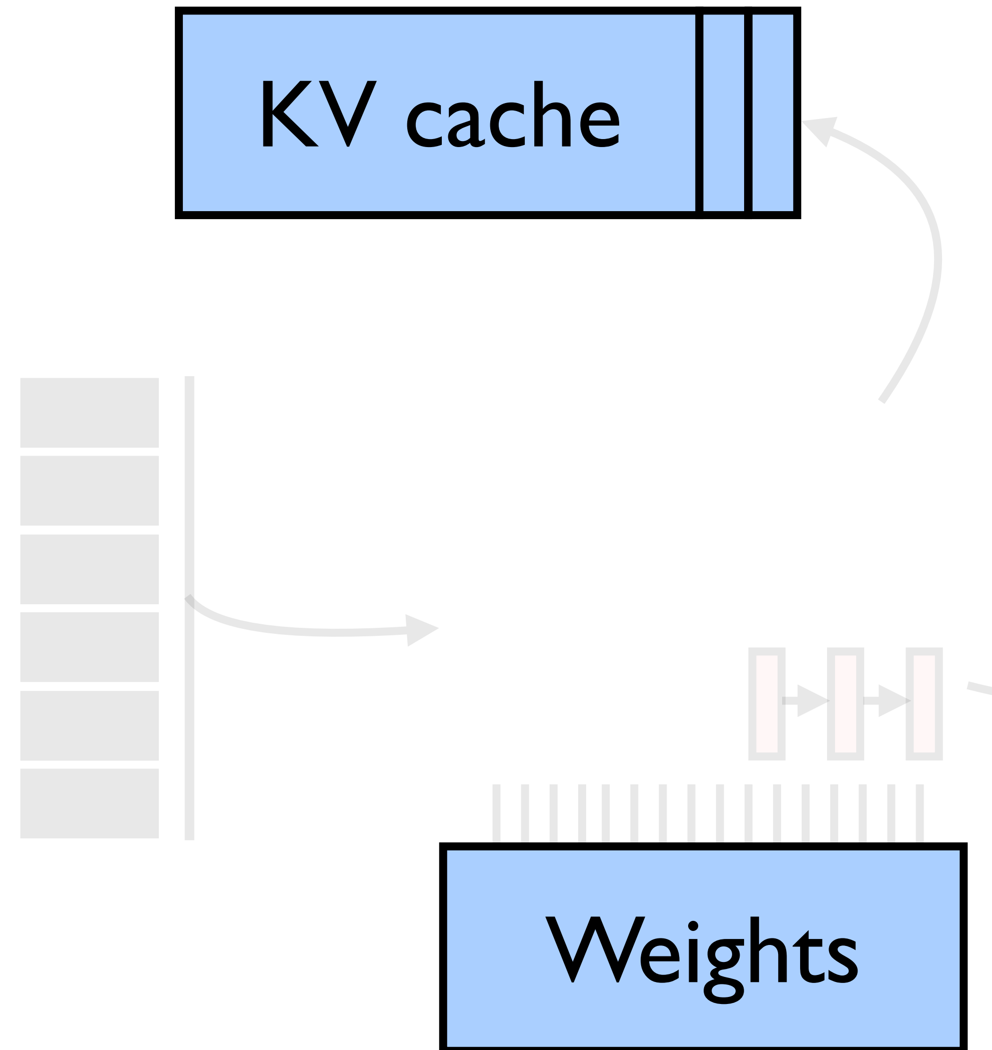
DRAM
SRAM
MRAM
RRAM
FRAM
PCM
Flash
GC
HGC
FeFet
OS-OS

	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

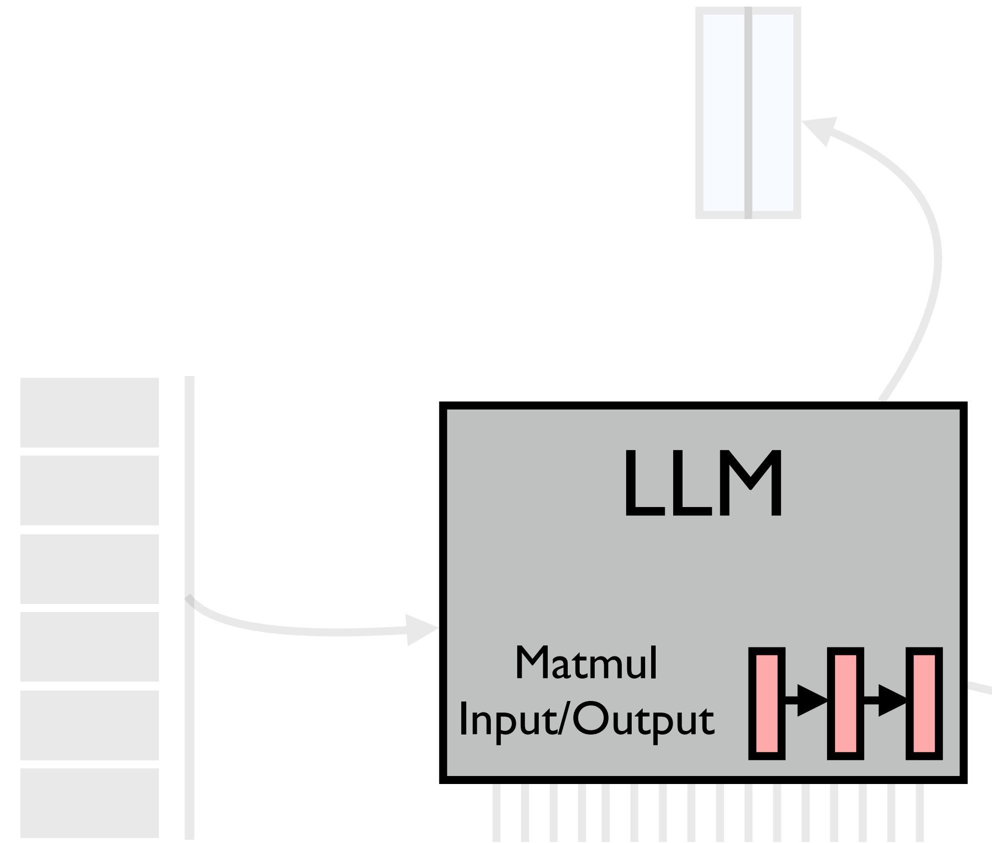


Write Rarely, Read Often



- 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

Outcome of Year One

- We can distill many common software patterns and memory types into two broad classes
- Application data
  - Written very rarely: read-only (code, inference model weights), read-mostly (network data stores, databases, KV caches), "cold" data,
  - Written and read very frequently: caches, scratchpads, inference activations, computational kernel scratchpads,
- Memory types
  - Non-volatile memories: expensive and limited writes, but higher density, lower read energy, and high retention
  - Dynamic memories: require refreshes if not accessed, but higher density, lower read and write energies

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

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	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: Flash, MRAM, RRAM, FeRAM, 3DXP
- On-die, in-package, or off-package with compute

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)
- Integrated on-die with compute

Year One Outcomes

- We can distill many common software patterns and memory types into two broad classes
- Long-term RAM (LtRAM) for written-rarely, long-lived data
- Short-term RAM (StRAM) for frequently accessed data

Year Two Progress

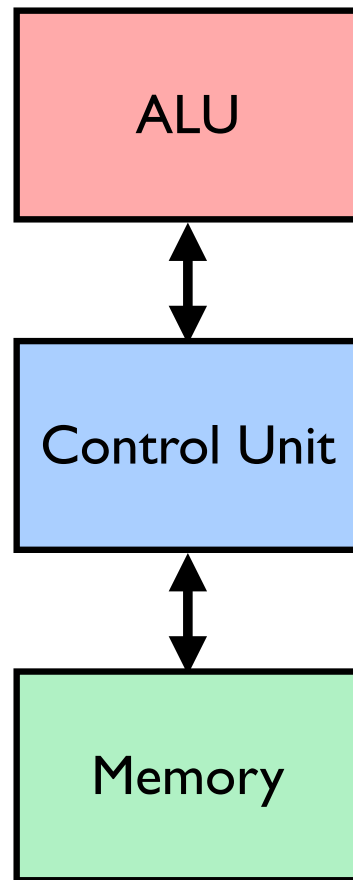
- Design and implementation of LtRAM and StRAM
- Building accelerators and systems incorporating StRAM and LtRAM
  - µVLA: 16nm chiplet SoP accelerator with StRAM and LtRAM
  - Server Long Term Memory: integrating LtRAM into a CPU server
- Engineering challenges in densely integrated memory and compute
  - 3D IC cooling: thermals are a critical concern
  - Retention tuning in StRAM: how short is "short"?
- Memory-centric applications and architecture
  - Memory fit: architectural support for OS memory management
  - Federation of Experts: increasing locality in large-scale inference
  - Flow translation table: matching addressing to application use cases

Year Two Graduates

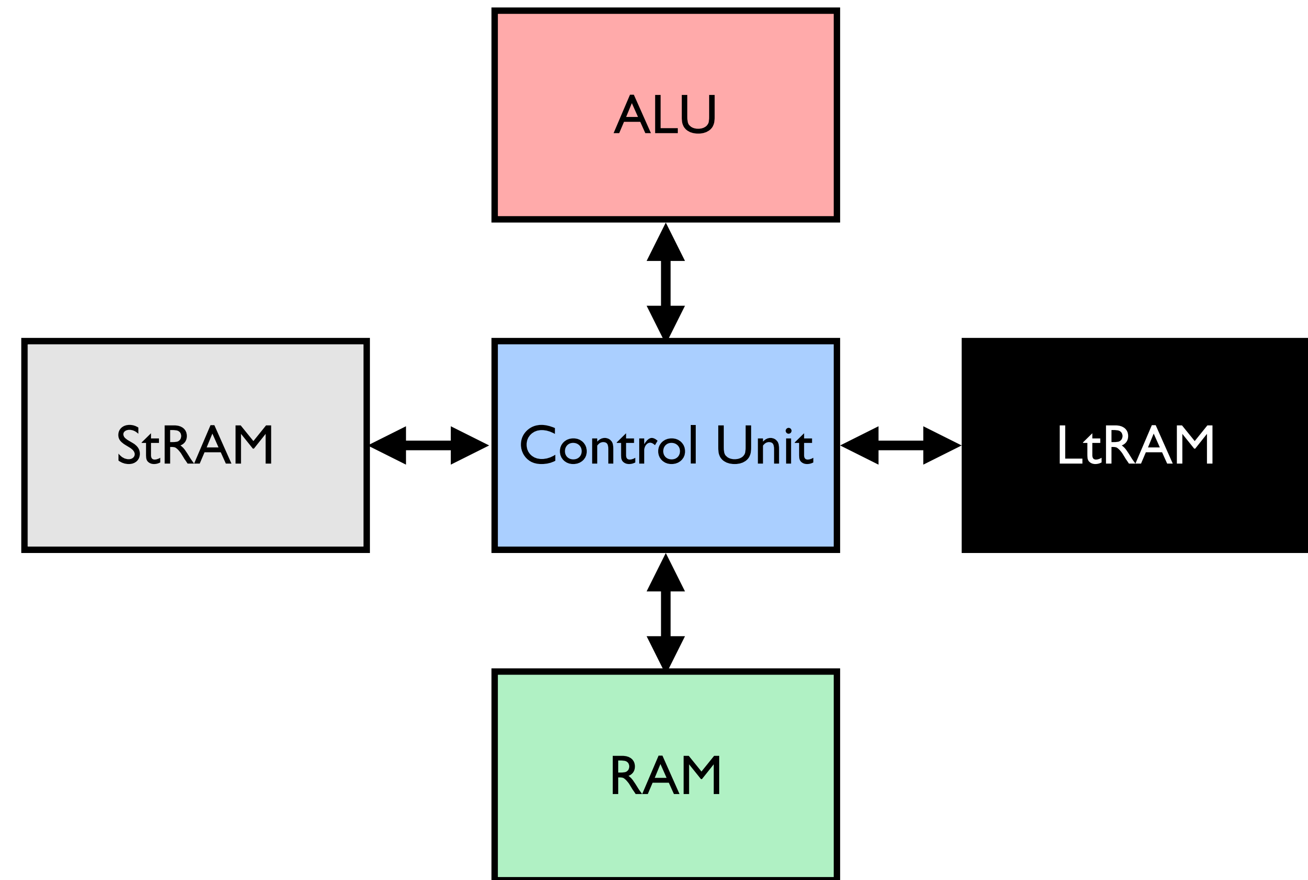
Shuhan Liu
Faculty
(deciding between UW, Yale, UPenn)

Agur Adams
Returned to active duty near D.C.
Leading next generation USMC
command and control network

Where We Are Going



von Neumann
Architecture



DAM
Architecture

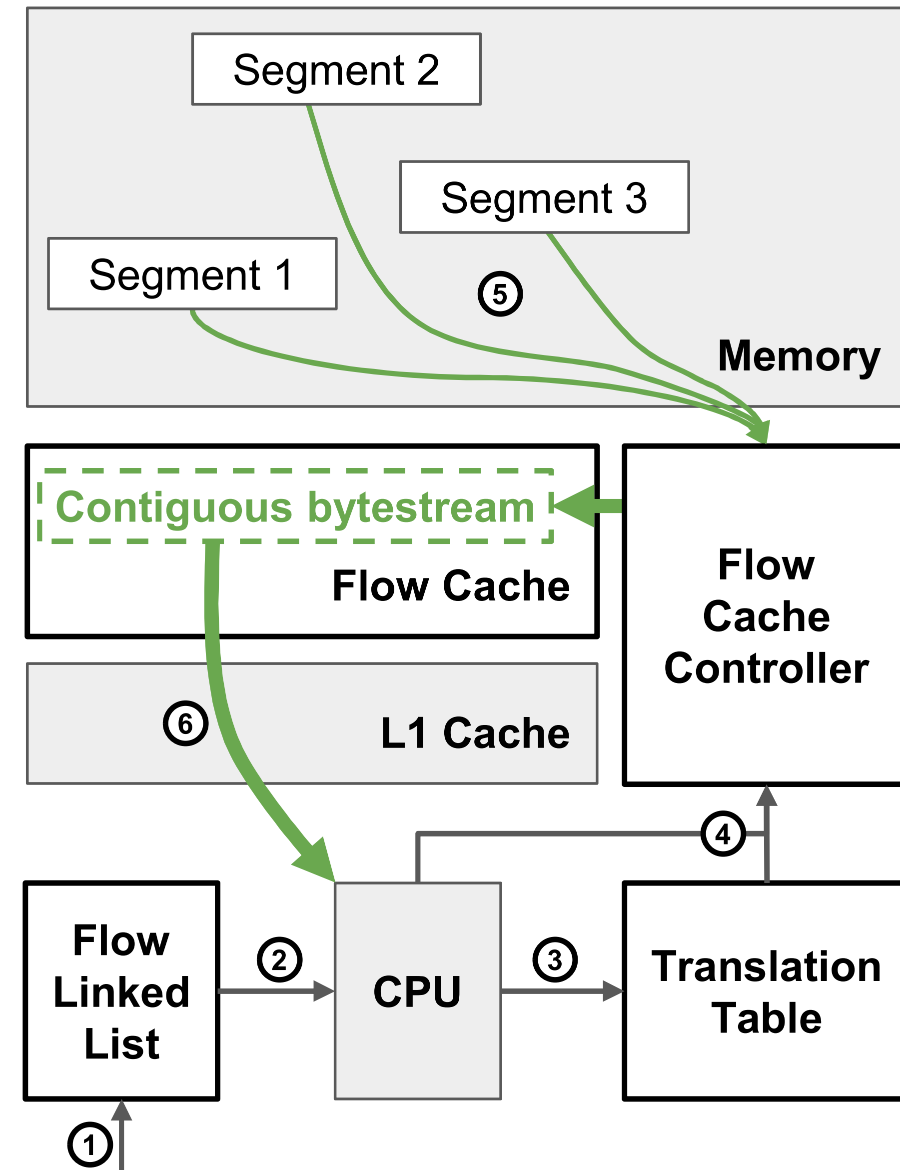
Today's Schedule

Time	Session
9:00	Project Review: Philip Levis
9:30	Packets are Not Pages: Flow-Based Addressing Conserves Memory Bandwidth - Colin Drewes
10:00	µVLA: Designing a Multi-Chiplet 16nm SoP with Heterogeneous Memory on Package (HMoP) to Enable Real-Time Physical AI on the Edge - Christian Kubicka
10:30	Break
10:45	Dipole engineering for retention tuning in oxide semiconductor gaincell memories - Fabia Farlin Athena
11:15	Federation of Experts: Communication Efficient Distributed Inference for Large Language Models - Shahir Abdurrahman
11:45	Lunch
12:45	Keynote: Bill Dally
13:45	Walk to Beach (discussing memory)
15:00	Ferroelectrics: Past, Present, and Future - H.-S. Philip Wong
15:45	The Future of Cooling 3D ICs - Dennis Rich
16:00	Server Long Term RAM - David Shim
16:30	Feedback and Wrap-Up - Everyone
17:00	Social Hour

Packets are Not Pages

Colin Drewes

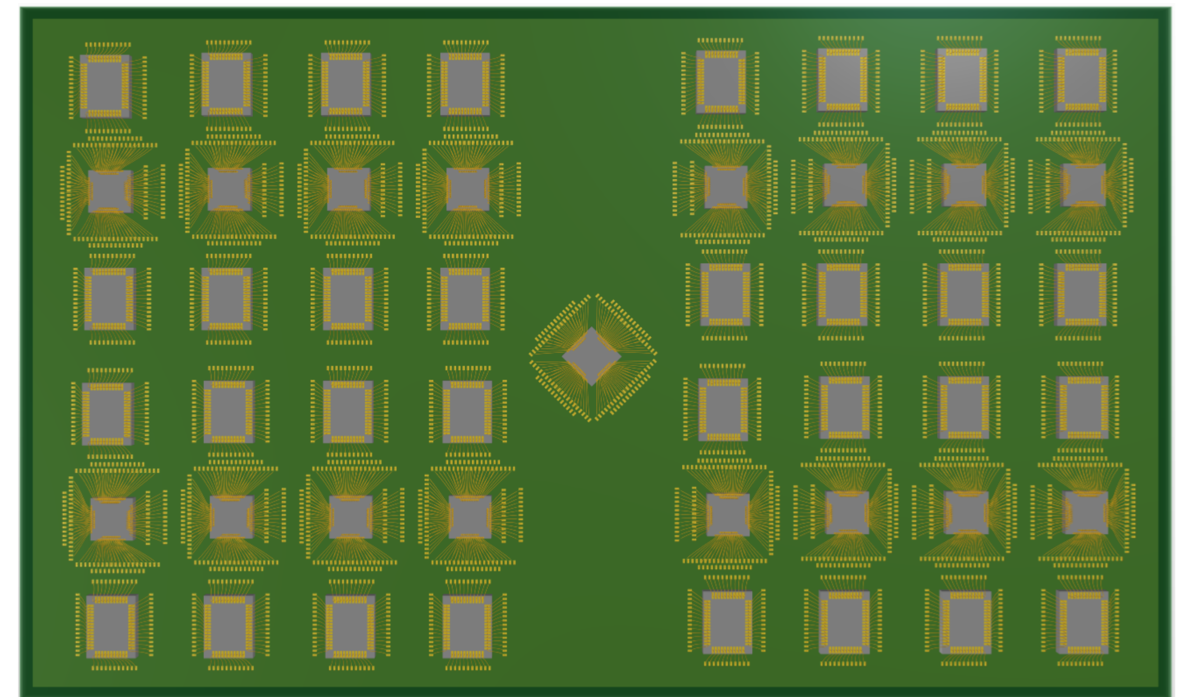
- Explores how processing addressing affects memory access patterns
- Software often needs to access memory sequentially (loops, prefetching)
- Network interface cards (NICs) have to handle scattered packets (descriptor rings)
- Introducing a new, packet-centric, addressing mode to a NIC reduces DRAM bandwidth use by 77-83%



μVLA: SoP with LtRAM and StRAM

Christian Kubicka

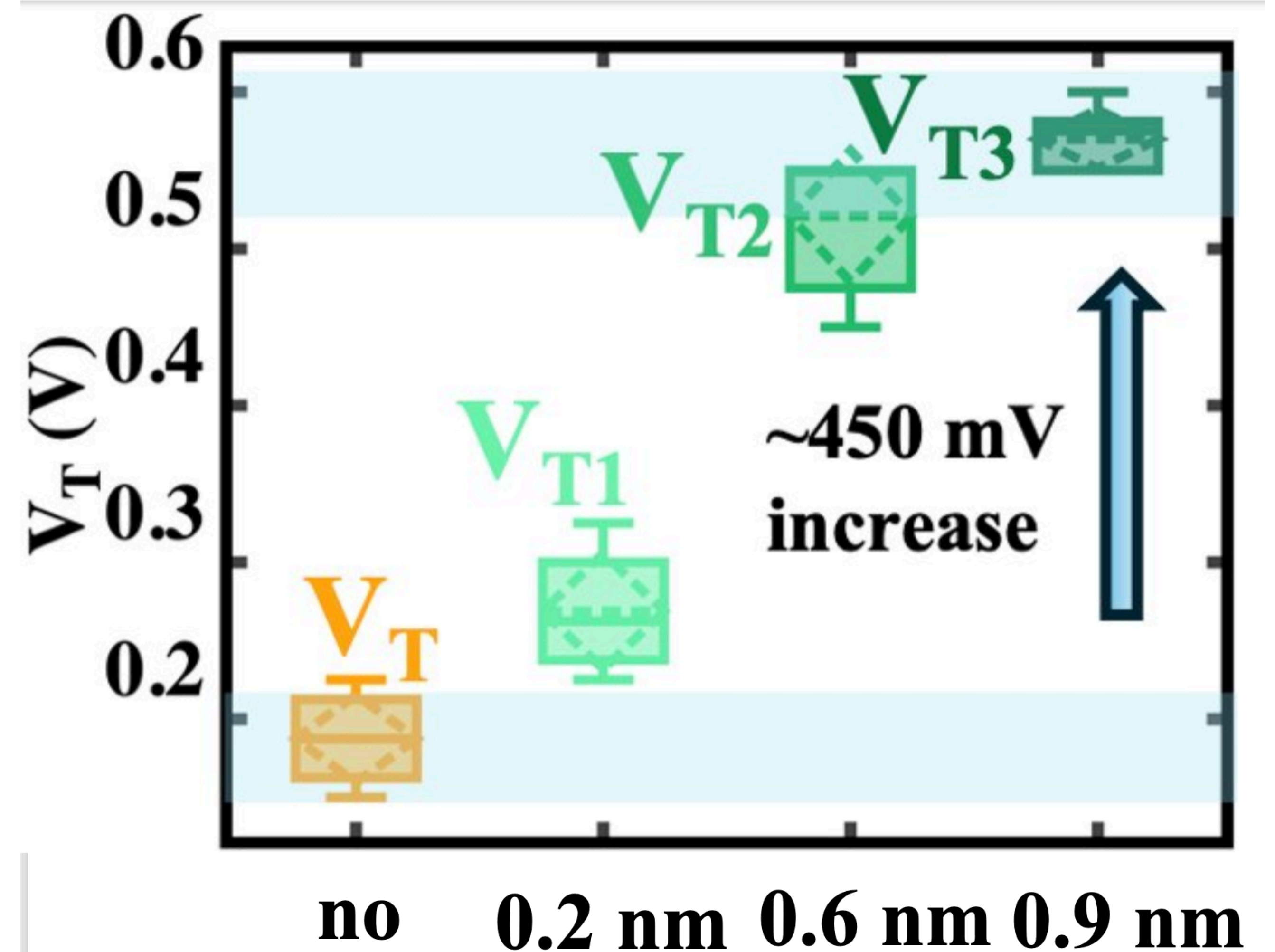
- Explores how LtRAM and StRAM can be integrated into a complete physical AI accelerator
- Multichiplet system-on-package (SoP) with on-die StRAM for compute and LtRAM chiplets
- Careful decomposition of AI workloads into computational and memory architecture



Dipole Engineering is OS Gain Cells

Fabia Darlin Athena

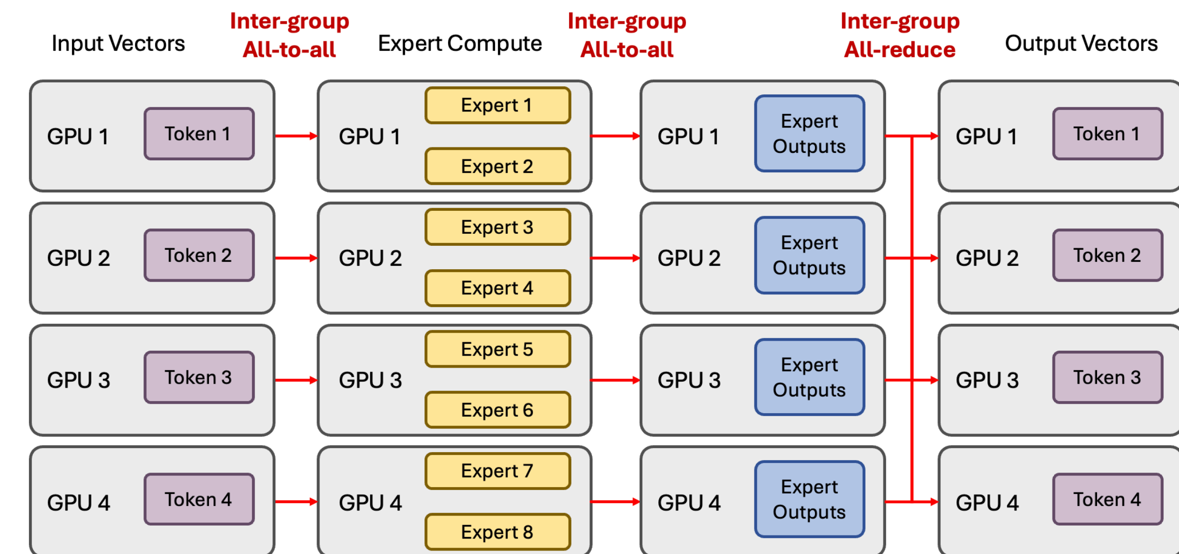
- Explores how to tune an interface dipole to control gain cell StRAM behavior
- Gain cells have variable retention
- Can tune the interface dipole to increase threshold voltage V_T
- Makes gain cell retention more robust across temperature
- Works across many technologies



Federation of Experts

Shahir Abdurrahman

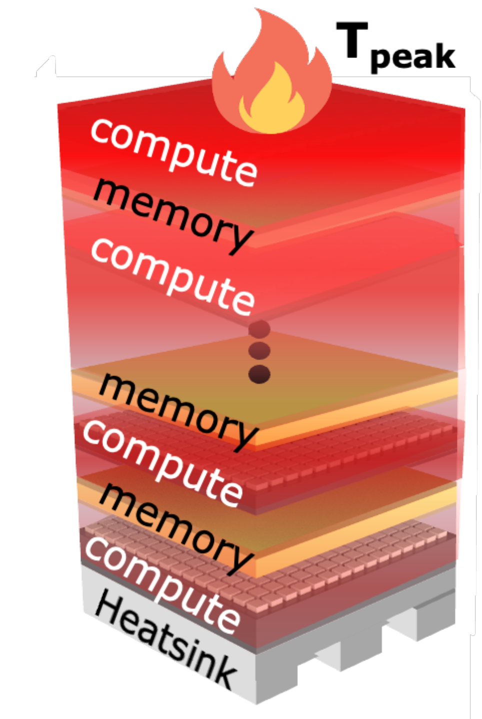
- Explores how to restructure large language models to increase memory locality
- Refactor mixture of experts so a token is routed within a smaller *group* of experts
  - k different subsets for top-k mixtures
  - Keeps token within a subset
  - Don't need all-to-all communication between all of the experts, just within a group
- Reduces time to first token by 66% and time between tokens 50%

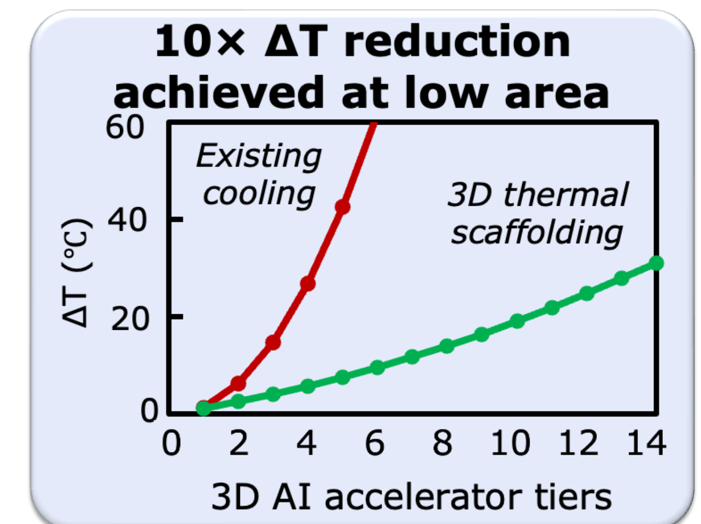


Cooling of 3D ICs

Dennis Rich

- Explores how to cool 3D ICs better through thermal scaffolding
- Dissipating heat quickly emerges as the limiting factor in dense 3D designs that tightly couple compute and memory
- Dense compute and memory you can't use is not cost effective
- Carefully introducing thermal scaffolding within dense 3D designs improves ΔT by 10x

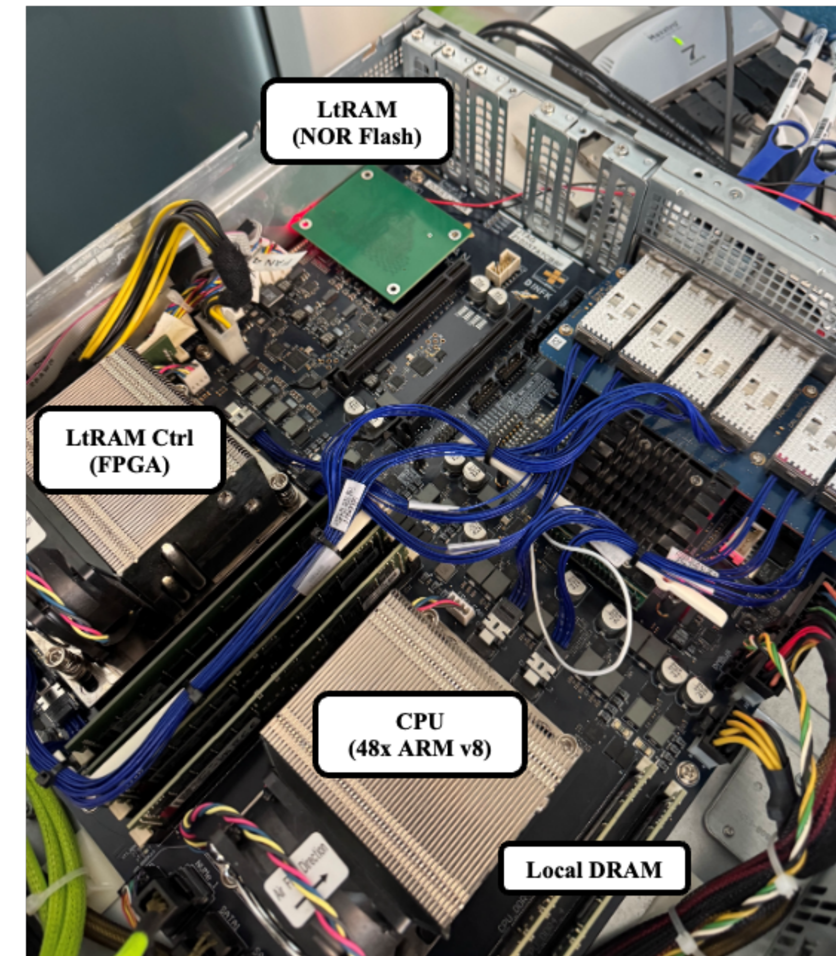




Server Long Term RAM

David Shim

- Explores how to reduce LtRAM latency by changing the CPU/memory interface to move responsibility into the operating systems kernel
- LtRAM pages can be read normally by software
- LtRAM pages are copy-on-write: use highly optimized code in OS to move into DRAM
- Operating system decides when to move unwritten pages into LtRAM, manages wear



Today's Schedule

Time	Session
9:00	Project Review: Philip Levis
9:30	Packets are Not Pages: Flow-Based Addressing Conserves Memory Bandwidth - Colin Drewes
10:00	μVLA: Designing a Multi-Chiplet 16nm SoP with Heterogeneous Memory on Package (HMoP) to Enable Real-Time Physical AI on the Edge - Christian Kubicka
10:30	Break
10:45	Dipole engineering for retention tuning in oxide semiconductor gaincell memories - Fabia Farlin Athena
11:15	Federation of Experts: Communication Efficient Distributed Inference for Large Language Models - Shahir Abdurrahman
11:45	Lunch
12:45	Keynote: Bill Dally
13:45	Walk to Beach (discussing memory)
15:00	Ferroelectrics: Past, Present, and Future - H.-S. Philip Wong
15:45	The Future of Cooling 3D ICs - Dennis Rich
16:00	Server Long Term RAM - David Shim
16:30	Feedback and Wrap-Up - Everyone
17:00	Social Hour

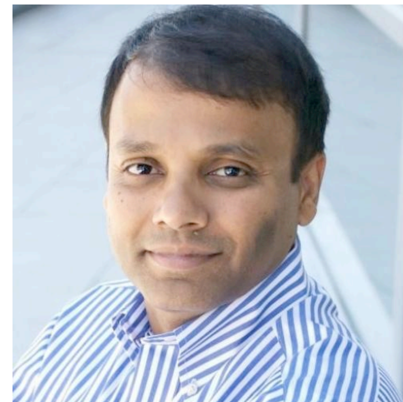
DAM/MemoryDAX Team

Philip Levis
Director

Caroline Trippel
Director

Shridhar Mukund
Affilate Coordinator

Christos Kozyrakis



Subhasish Mitra

Thierry Tambe

Keith Winstein

H.-S. Philip Wong

Mary Wootters

Thank You!