1858: The Ocean Shrinks to a Channel



H.M.S. *Agamemnon* laying the first transatlantic telegraph cable, 1858.

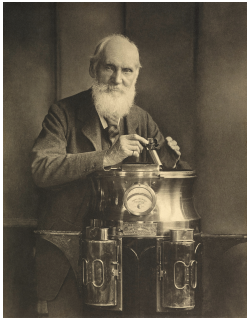
1858: The Ocean Shrinks to a Channel



H.M.S. *Agamemnon* laying the first transatlantic telegraph cable, 1858.

Queen Victoria's first message took **almost seventeen hours** to cross.
Three weeks later, **the cable was dead**.

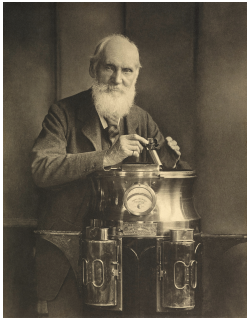
Two Ways to Fix a Failing Channel



William Thomson,

aboard the *Agamemnon*

Two Ways to Fix a Failing Channel

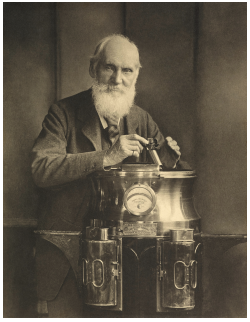


William Thomson,
aboard the *Agamemnon*

Whitehouse: force a stronger signal. The chief electrician drove the failing cable with induction coils at roughly 2,000 volts — and destroyed what remained of its insulation.

Thomson: measure, then re-engineer. His mirror galvanometer detected the faintest currents at low voltage; the re-engineered cable of 1866 remained in service for decades.

Two Ways to Fix a Failing Channel



William Thomson,
aboard the *Agamemnon*

Whitehouse: force a stronger signal. The chief electrician drove the failing cable with induction coils at roughly 2,000 volts — and destroyed what remained of its insulation.

Thomson: measure, then re-engineer. His mirror galvanometer detected the faintest currents at low voltage; the re-engineered cable of 1866 remained in service for decades.

A stronger transmitter destroyed the channel; measurement rebuilt it.
When the substrate is the limit, the substrate itself must be engineered.

The Memory Substrate of Tomorrow

Resilient DRAM for Secure and Intelligent Computing †

Keynote — 31st ACM Symposium on Access Control Models and Technologies (SACMAT '26)

Prashant J. Nair * — UBC and d-Matrix Inc.

Waterloo, ON, 9th July 2026

* Associate Professor at the University of British Columbia (UBC); lead architect of the 3D-memory architecture at d-Matrix.
† The 3D-DRAM accelerator work was done during sabbatical/leave from UBC.

The Quiet Substrate of Everything

~$100B

of DRAM per year — computing's largest single commodity

The Quiet Substrate of Everything

~$100B

of DRAM per year — computing's largest single commodity

100s of GB

stream through memory for *every token* a large model produces

The Quiet Substrate of Everything

~$100B

of DRAM per year — computing's largest single commodity

100s of GB

stream through memory for *every token* a large model produces

1 bit

flipped in DRAM has broken sandboxes, VMs, signing keys — and now *aligned models*

The Quiet Substrate of Everything

~$100B

of DRAM per year — computing's largest single commodity

100s of GB

stream through memory for *every token* a large model produces

1 bit

flipped in DRAM has broken sandboxes, VMs, signing keys — and now *aligned models*

Every access-control decision this community studies is, in the end,
a promise about bits held in memory.

Two Vantage Points, One Question

The university lab STAR Lab, UBC

A decade of memory mechanisms, from fault maps to in-DRAM defenses:

ArchShield [ISCA'13] → AVATAR [DSN'15] → XED [ISCA'16] → Hydra [ISCA'22] → PrISM [ISCA'26]

Two Vantage Points, One Question

The university lab STAR Lab, UBC

A decade of memory mechanisms, from fault maps to in-DRAM defenses:

ArchShield [ISCA'13] → AVATAR [DSN'15] → XED [ISCA'16] → Hydra [ISCA'22] → PrISM [ISCA'26]

The silicon floor d-Matrix

A 3D-DRAM accelerator for generative AI, built under production constraints:

105°C operation · yield and cost · 100 TB/s per card · Raptor [ISCA'26]

Two Vantage Points, One Question

The university lab STAR Lab, UBC

A decade of memory mechanisms, from fault maps to in-DRAM defenses:

ArchShield [ISCA'13] → AVATAR [DSN'15] → XED [ISCA'16] → Hydra [ISCA'22] → PrISM [ISCA'26]

The silicon floor d-Matrix

A 3D-DRAM accelerator for generative AI, built under production constraints:

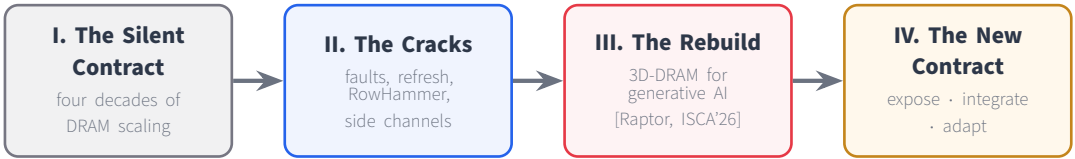
105°C operation · yield and cost · 100 TB/s per card · Raptor [ISCA'26]

Asked from both sides, it is the same question:
what should memory promise — and to whom?

For forty years, memory kept its promise so reliably
that we forgot anyone was keeping it.

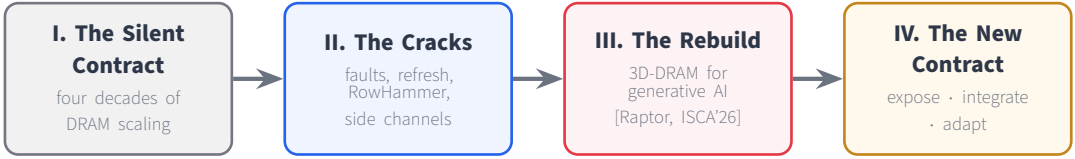
**This talk examines what happens when it fails —
and what we must build in its place.**

One Hour, Four Acts



Drawing on a decade of DRAM research at the STAR Lab (UBC) and d-Matrix.

One Hour, Four Acts



Drawing on a decade of DRAM research at the STAR Lab (UBC) and d-Matrix.

The thesis of this talk: resilience and security are not *features* — they are the **design constraints of the memory substrate of tomorrow.**

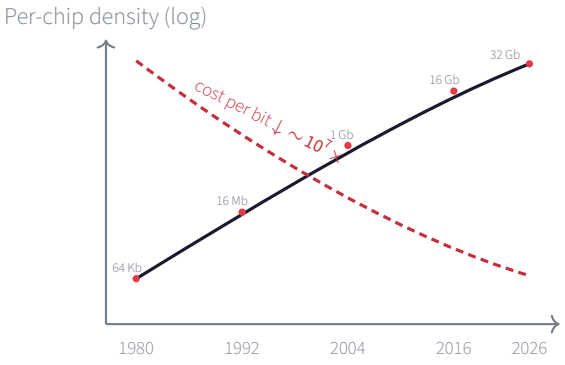
The Silent Contract

Four Decades of a Silent Contract



$\sim 10^6\times$ denser, $\sim 10^7\times$ cheaper per bit — the economic foundation of computing.

Four Decades of a Silent Contract



$\sim 10^6\times$ denser, $\sim 10^7\times$ cheaper per bit — the economic foundation of computing.

The contract software signed:

Passive. An array of bits — no policy, no intelligence, no opinions.

Reliable. Errors are rare; refresh and ECC quietly hide the rest.

Trusted. Data stays where you put it — correct, isolated, private.

For forty years, software never had to *think* about memory. **That was the contract.**

The Contract Is Cracking

Cells fail.

Scaling breeds weak bits: error rates approach 10^{-4} — far beyond sparing and SECDED.

ArchShield [ISCA'13]

Refresh encroaches.

Denser chips spend ever more time refreshing — and retention *changes at runtime*.

Refresh Pausing [HPCA'13] · AVATAR [DSN'15]

Neighbors interfere.

RowHammer: flip a bit you never touched. Thresholds fell 139 K → 4.8 K → below 1 K activations.

Kim, Nair, Qureshi [IEEE CAL'14] · Hydra [ISCA'22]

Abstractions leak.

ECC, trackers — even RowHammer *defenses* — become observable timing side channels.

PRACLeak [ISCA'25] · HuffDuff [ASPLOS'23]

The Contract Is Cracking

Cells fail.

Scaling breeds weak bits: error rates approach 10^{-4} — far beyond sparing and SECDED.

ArchShield [ISCA'13]

Refresh encroaches.

Denser chips spend ever more time refreshing — and retention *changes at runtime*.

Refresh Pausing [HPCA'13] · AVATAR [DSN'15]

Neighbors interfere.

RowHammer: flip a bit you never touched. Thresholds fell 139 K → 4.8 K → below 1 K activations.

Kim, Nair, Qureshi [IEEE CAL'14] · Hydra [ISCA'22]

Abstractions leak.

ECC, trackers — even RowHammer *defenses* — become observable timing side channels.

PRACLeak [ISCA'25] · HuffDuff [ASPLOS'23]

Density made memory cheap —
and made **every clause of the contract negotiable.**

When Memory Fails: Resilience

When Cells Fail: Resilience at High Error Rates

Scaling makes cells weak:

- Smaller geometries $\Rightarrow$ variation, leakage, weak bits
- Error rates climb toward 10^{-4} per cell
- Row/column sparing and SECDED become prohibitive at these rates

The old answer

Hide faults inside the device — and stop scaling when hiding gets too expensive.

When Cells Fail: Resilience at High Error Rates

Scaling makes cells weak:

- Smaller geometries ⇒ variation, leakage, weak bits
- Error rates climb toward 10^{-4} per cell
- Row/column sparing and SECDED become prohibitive at these rates

The old answer

Hide faults inside the device — and stop scaling when hiding gets too expensive.

ArchShield [Nair+, ISCA'13]

- Runtime testing ⇒ a **fault map**, exposed to the architecture
- Selective word-level replication in reserved DRAM
- Tolerates 10^{-4} error rates (**100**× beyond ECC) at <2% slowdown

Evaluating resilience at scale

FaultSim [TACO'15]: Monte-Carlo reliability simulation driven by real field-failure data — for 2D *and* 3D-stacked memories.

Don't hide faults inside the device — **expose them, and architect around them.**

The Refresh Wall

Every bit, every 64 ms:

- Refresh blocks reads ⇒ latency spikes
- Denser chips ⇒ more rows to sweep
- Penalty already worth ∼7% performance — and growing
- Worse: retention *changes at runtime* (VRT)

The Refresh Wall

Every bit, every 64 ms:

- Refresh blocks reads ⇒ latency spikes
- Denser chips ⇒ more rows to sweep
- Penalty already worth ∼7% performance — and growing
- Worse: retention *changes at runtime* (VRT)

Refresh Pausing [HPCA'13, TACO'14]
Pause refresh to serve pending reads: **+5.1%** at 8 Gb.

AVATAR [DSN'15 — DSN Test of Time Award]
VRT-aware *multirate* refresh: **62–72% fewer refreshes**.

Morphable ECC [DSN'15]
Stronger idle ECC ⇒ refresh every **1 s**, not 64 ms.

Refresh is not a constant of nature — it is a **policy**, and policies can adapt.
(Remember this at 105°C.)

Stacked Memory Fails Differently



one faulty TSV ⇒
rows and banks
across *all* dies

And the stack is soldered in place — *field replacement is impossible*.

Stacked Memory Fails Differently



one faulty TSV ⇒ rows and banks across *all* dies

And the stack is soldered in place — *field replacement is impossible.*

Citadel [MICRO'14, TACO'16]

TSV-Swap + tri-dimensional parity + dual-granularity sparing ⇒ **700**× the reliability of ChipKill-style codes — while keeping each cache line in *one* bank.

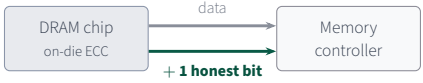
New geometry, new failure modes — and new mechanisms shaped to them. *Act III returns to exactly this geometry.*

Effective resilience is structure-specific — **mechanisms must follow the physical geometry of the device.**

Expose, Don’t Hide: The One-Honest-Bit Lesson

On-die ECC hides information

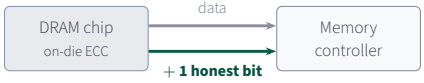
Vendors now correct errors *inside* the chip — invisible to the controller, its detections wasted.



Expose, Don’t Hide: The One-Honest-Bit Lesson

On-die ECC hides information

Vendors now correct errors *inside* the chip — invisible to the controller, its detections wasted.



XED [ISCA’16] — commercially adopted

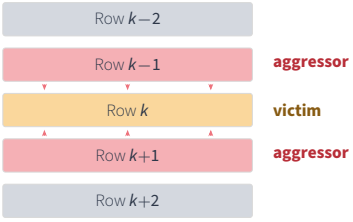
Expose on-die *detection* via a catch-word ⇒ Chipkill-level protection (**172**× SECDED) from a standard ECC-DIMM — 21% faster than Chipkill.

A single exposed detection bit yields **172**× **reliability.**
Information hidden behind a rigid interface is information lost.

When Memory Betrays: Security

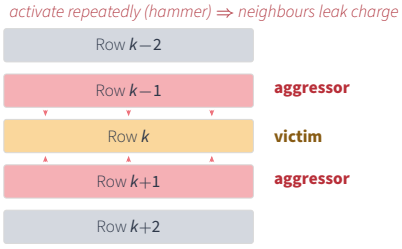
RowHammer: When Density Becomes a Weapon

activate repeatedly (hammer) ⇒ neighbours leak charge



Flip a bit you never touched — no software bug required.

RowHammer: When Density Becomes a Weapon



Flip a bit you never touched — no software bug required.

The threshold keeps collapsing:

2014	∼139 K activations
2020	∼4.8 K
Today	sub-1 K on some parts

A pure *reliability* artifact of density — weaponized into privilege escalation, sandbox escapes, and key theft.

The disturb threshold fell ∼**300**× in a decade.
Density turned a nuisance into **an attack surface**.

A Decade of Defense: Chasing a Falling Threshold

Our defense lineage — each generation answering the last one's weakness:

CAL'14	**CRA / PRA** — among the *first* mitigations: counters & probabilistic refresh
ASPLOS'22	**RRS** — randomized row swaps
MICRO'22	**AQUA** — quarantine aggressor rows
ISCA'22	**Hydra** — tracking that scales to low thresholds
HPCA'23	**Scale-SRS** — Juggernaut breaks swaps; a scalable fix (Best Paper)
HPCA'25	**QPRAC** — principled PRAC: secure at $T = 71$, 0.8% cost (Dist. Artifact)
ISCA'26	**PrISM** — the mitigation moves *inside* the DRAM die

A Decade of Defense: Chasing a Falling Threshold

Our defense lineage — each generation answering the last one's weakness:

CAL'14	**CRA / PRA** — among the *first* mitigations: counters & probabilistic refresh
ASPLOS'22	**RRS** — randomized row swaps
MICRO'22	**AQUA** — quarantine aggressor rows
ISCA'22	**Hydra** — tracking that scales to low thresholds
HPCA'23	**Scale-SRS** — Juggernaut breaks swaps; a scalable fix (Best Paper)
HPCA'25	**QPRAC** — principled PRAC: secure at $T = 71$, 0.8% cost (Dist. Artifact)
ISCA'26	**PrISM** — the mitigation moves *inside* the DRAM die

Defenses migrated from software, to the controller, **into the DRAM device itself**.
When a guarantee breaks, **its enforcement moves toward the physics**.

When Mitigations Backfire

The defense becomes the channel

PRAC-era DRAM keeps a counter *per row* and stalls the bus (ABO) when a row gets hot. Those stalls are **data-dependent**.

PRACLeak / TPRAC [ISCA'25]

"When Mitigations Backfire": mitigation-induced timing leaks **hundreds of bits/s** of a co-tenant's access pattern — and timing-safe PRAC (TPRAC) closes the channel.

When Mitigations Backfire

The defense becomes the channel

PRAC-era DRAM keeps a counter *per row* and stalls the bus (ABO) when a row gets hot. Those stalls are **data-dependent**.

PRACLeak / TPRAC [ISCA'25]

"When Mitigations Backfire": mitigation-induced timing leaks **hundreds of bits/s** of a co-tenant's access pattern — and timing-safe PRAC (TPRAC) closes the channel.

The defense becomes the weapon

An adversary can *deliberately* trigger mitigations to throttle a victim sharing the bank: a denial-of-service knob.

DAPPER [HPCA'25] · **PrIDE+RFM** [ISCA'26]

Performance-attack-resilient tracking; PrIDE+RFM cuts mitigation-abuse slowdown from **10.4% to 1.4%**.

Every mechanism you add is **attack surface**. Reliability, security, and performance must be verified *together* — or each inherits the other's vulnerabilities.

Reliability × Security: One Budget

Encrypted, integrity-protected memory needs *metadata* — counters, MACs, ECC.
They compete for the **same bits, banks, and bandwidth**.

Share the bits

SYNERGY [HPCA'18]: put the MAC *in* the ninth ECC chip ⇒ **+20%** performance *and* **185**× reliability.

Shrink the metadata

Morphable Counters [MICRO'18]: 128-ary encryption counters ⇒ **4**× denser, fewer re-encryptions.
DEUCE [ASPLOS'15]: **2**× NVM lifetime.

Cross-check the attacker

SafeGuard [HPCA'22]: the same MAC-in-ECC construction becomes an *integrity backstop* against RowHammer escapes in secure memory.

Reliability × Security: One Budget

Encrypted, integrity-protected memory needs *metadata* — counters, MACs, ECC.
They compete for the **same bits, banks, and bandwidth**.

Share the bits

SYNERGY [HPCA'18]: put the MAC *in* the ninth ECC chip ⇒ **+20%** performance *and* **185**× reliability.

Shrink the metadata

Morphable Counters [MICRO'18]: 128-ary encryption counters ⇒ **4**× denser, fewer re-encryptions.
DEUCE [ASPLOS'15]: **2**× NVM lifetime.

Cross-check the attacker

SafeGuard [HPCA'22]: the same MAC-in-ECC construction becomes an *integrity backstop* against RowHammer escapes in secure memory.

Treated separately, reliability and security each tax the system twice.
Co-designed, a single mechanism serves both.

The New Stakes: When Memory Breaks, Models Break

Bit-flips now flip *behavior*

PrisonBreak [2024]

Fewer than **25** targeted RowHammer bit-flips *jailbreak* an aligned billion-parameter LLM at runtime — no fine-tuning, no prompt.

HuffDuff [ASPLOS'23]

A DRAM *volume* side channel steals pruned-CNN architectures, collapsing the search space by $\sim\mathbf{10}^{\mathbf{94}}$.

The New Stakes: When Memory Breaks, Models Break

Bit-flips now flip *behavior*

PrisonBreak [2024]

Fewer than **25** targeted RowHammer bit-flips *jailbreak* an aligned billion-parameter LLM at runtime — no fine-tuning, no prompt.

HuffDuff [ASPLOS'23]

A DRAM *volume* side channel steals pruned-CNN architectures, collapsing the search space by $\sim\mathbf{10}^{\mathbf{94}}$.

And resilience must scale to models

Structural Coding [SC'23]

Error coding shaped to the *structure* of CNN inference: $\sim$**1000**$\times$ fewer silent data corruptions at $\sim$**3%** overhead.

The asset in memory is no longer just data — it is *weights*: the model's alignment, its IP, its correctness.

In the AI era, **memory integrity is model integrity**.

The asset has changed. The substrate has not — yet.

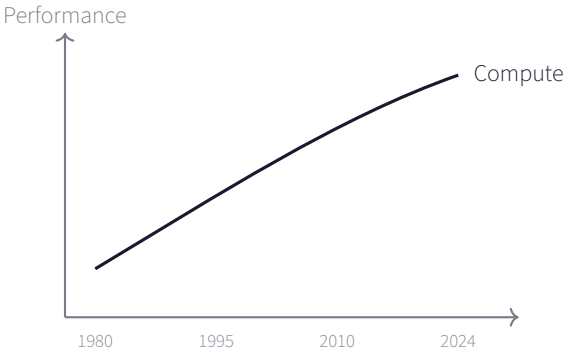
So memory can **fail**, and memory can **betray**.

Now it must also feed the most memory-bound workload computing has ever produced.

Act III: rebuilding the substrate for generative AI.

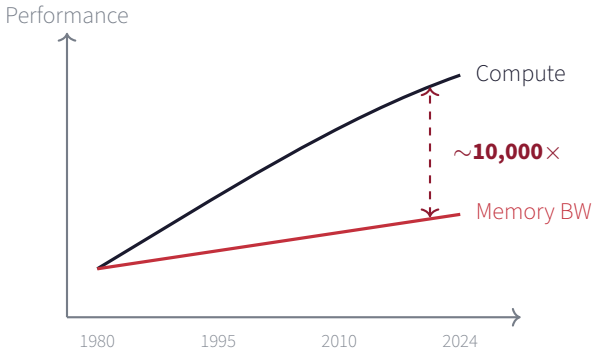
The Memory Wall Meets Generative AI

A Gap That Grew



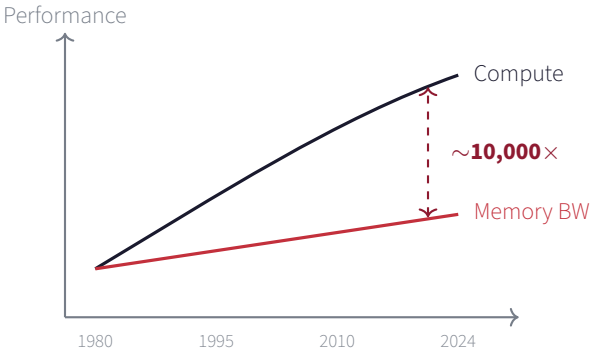
- Processor speed: ∼60%/year

A Gap That Grew



- Processor speed: ∼60%/year
- DRAM bandwidth: ∼7%/year
- Caches survived by exploiting *data reuse*

A Gap That Grew



- Autoregressive decode eliminates reuse.

Autoregressive decode reads every weight, every token.

⇒ **there is no reuse to exploit.**

What Changed in the Last Five Years

Before: Training & Classification

- Fixed input sizes, large batches
- High reuse across samples
- Compute-bound: attention scales as $O(L^2)$
- ✓ GPUs excel here

Notation: L = **context length** (number of tokens in the prompt + tokens generated so far).

What Changed in the Last Five Years

Before: Training & Classification

- Fixed input sizes, large batches
- High reuse across samples
- Compute-bound: $O(L^2)$
- ✓ GPUs excel

Now: Interactive Generation

- Variable context, single-token steps
- Read *all* weights *every* token
- Memory-bound: KV-cache $\Rightarrow O(L)$ per token
- ✗ GPUs starve for data

Batch processing $\rightarrow$ interactive serving:
the bottleneck moved from compute to **memory.**

Notation: L = **context length** (number of tokens in the prompt + tokens generated so far).

The Decode Bottleneck

Generative AI Inference: Two Phases

Prefill

Process all input tokens **in parallel**

High data reuse across tokens

✓ **Compute-bound**

Generative AI Inference: Two Phases

Prefill

Process all input tokens **in parallel**

High data reuse across tokens

✓ **Compute-bound**

then



Decode

Generate **one token** at a time

Read entire model + KV cache

✗ **Memory-bound**

Generative AI Inference: Two Phases

Prefill
Process all input tokens **in parallel**
High data reuse across tokens
✓ **Compute-bound**

then

Decode
Generate **one token** at a time
Read entire model + KV cache
✗ **Memory-bound**

10% | ~90% of wall-clock inference time is spent in decode

70 GB
model weights (INT8)
read every token

+ KV cache
0.16 MB/token/user
grows linearly with context

0 reuse
full read of weights
and cache, every token

How Memory-Bound Is Decode?

Arithmetic intensity (AI)
= compute per byte moved

Llama-70B decode:
140 GFLOP ÷ 75 GB = **AI = 1.9**

Balance point (AI*)
= peak compute ÷ memory BW
Below AI*: hardware idles.

	HBM	3D-DRAM
BW	18 TB/s	100 TB/s
AI*	278	50
Gap	146×	26×

More BW → lower AI* → faster

How Memory-Bound Is Decode?

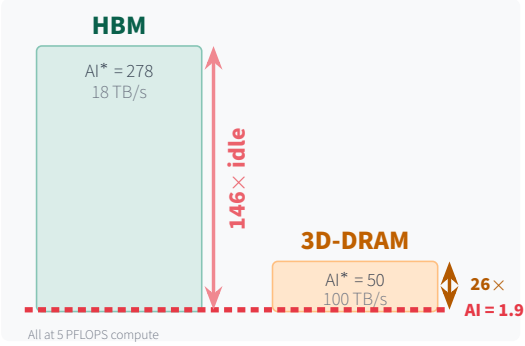
Arithmetic intensity (AI)
= compute per byte moved

Llama-70B decode:
140 GFLOP ÷ 75 GB = **AI = 1.9**

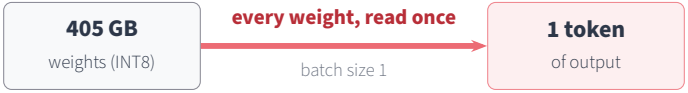
Balance point (AI*)
= peak compute ÷ memory BW
Below AI*: hardware idles.

	HBM	3D-DRAM
BW	18 TB/s	100 TB/s
AI*	278	50
Gap	146×	26×

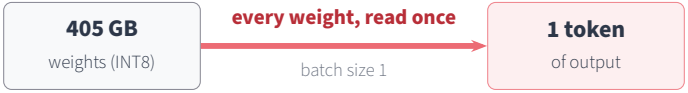
More BW → lower AI* → faster



What One Token Costs



What One Token Costs

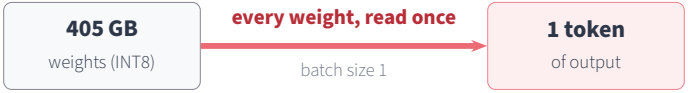


One user at a readable **50 tokens/s**:

$$405\ \text{GB} \times 50/\text{s} \approx \mathbf{20\ TB/s}$$

of weight traffic — *before* KV cache, *before* a second user.

What One Token Costs



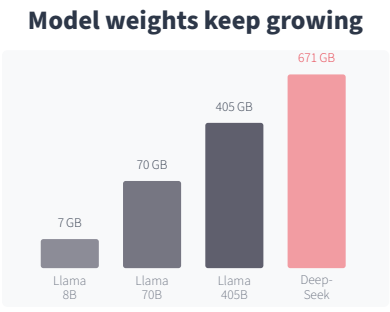
One user at a readable **50 tokens/s**:

$$405\ \text{GB} \times 50/\text{s} \approx \mathbf{20\ TB/s}$$

of weight traffic — *before* KV cache, *before* a second user.

Compute performs one operation per weight per token.
Memory must deliver every one of those weights.

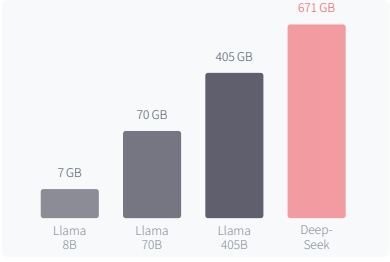
The Growing Data Problem



INT8 weights. Larger models need more cards.

The Growing Data Problem

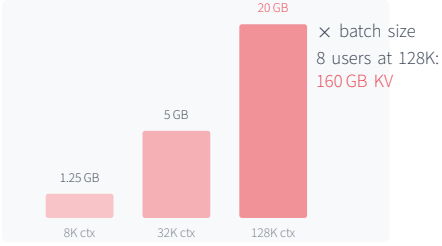
Model weights keep growing



INT8 weights. Larger models need more cards.

KV cache grows with context

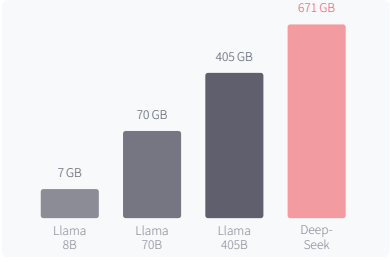
(Llama-3.1 70B, INT8, per user)



Longer context × more users ⇒ KV dominates — even with Keyformer [MLSys'24], constant-size KV [ICML'25], *weights still stream*.

The Growing Data Problem

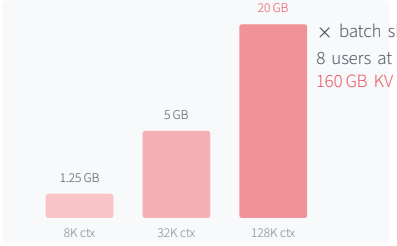
Model weights keep growing



INT8 weights. Larger models need more cards.

KV cache grows with context

(Llama-3.1 70B, INT8, per user)



Longer context × more users ⇒ KV dominates — even with Keyformer [MLSys'24], constant-size KV [ICML'25], *weights still stream*.

Weights set the bandwidth floor. KV cache sets the **capacity ceiling**. Both are growing.

Batching Has Hard Limits

Batching amortizes weight reads:

- Batch B users together
- Read weights once, reuse across B requests
- AI scales: $\mathrm{AI}_B \approx B \cdot \mathrm{AI}_1$

But KV cache scales *with* batch size:

- Each user carries its own KV cache
- Total memory: $\underbrace{70\,\mathrm{GB}}_{\text{weights}} + B \times \underbrace{\mathrm{KV}}_{\text{per user}}$
- At 128K, $B = 8$: $70 + 8 \times 20 = 230\,\mathrm{GB}$

Batching Has Hard Limits

Batching amortizes weight reads:

- Batch B users together
- Read weights once, reuse across B requests
- AI scales: $\mathrm{AI}_B \approx B \cdot \mathrm{AI}_1$

But KV cache scales *with* batch size:

- Each user carries its own KV cache
- Total memory: $\underbrace{70\,\mathrm{GB}}_{\text{weights}} + B \times \underbrace{\mathrm{KV}}_{\text{per user}}$
- At 128K, $B = 8$: $70 + 8 \times 20 = 230$ GB

Realistic Batch Sizes

Poisson arrivals simulated over 1 second:

Arrival rate	Max batch	Avg batch
110 req/s	1	1.0
500 req/s	5	1.2
1000 req/s	27	5.1

Even under heavy load, batch stays < 32.

Hard limits: KV-cache capacity, latency SLAs, bursty arrivals.

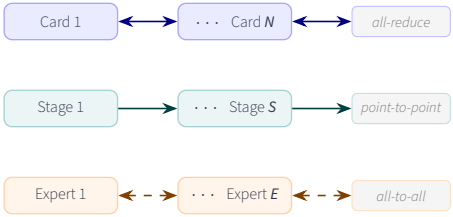
A **26× gap** (3D-DRAM) needs BS ≥ 26 to close $\Rightarrow$ but real batches average **∼5.**
The memory wall stands.

Multi-Card Scaling: The Parallelism Tax

Tensor Parallel
Split weight matrices

Pipeline Parallel
Split layers into stages

Expert Parallel
Route tokens to experts (MoE)

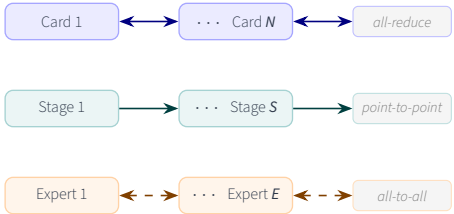


Multi-Card Scaling: The Parallelism Tax

Tensor Parallel
Split weight matrices

Pipeline Parallel
Split layers into stages

Expert Parallel
Route tokens to experts (MoE)



The vicious cycle:
Limited capacity ⇒ more cards
⇒ more collectives ⇒ exposed latency
⇒ **worse TPOT**

Llama-70B	Capacity	TP	PP	Total cards	Collectives/token
SRAM	4 GB/card	8	4	**32**	163
3D-DRAM	**32 GB/card**	**4**	**1**	**4**	**160**

8× fewer cards, no pipeline bubbles, and collective traverses within a switch.
The natural solution is not more cards ⇒ it is **more memory per card.**

The decode bottleneck is fundamental. Batching cannot solve it. Parallelism taxes it further.

The question becomes:

What memory does this workload demand?

Fast enough to feed the compute, and large enough to hold the model.

Today’s Memory Technologies

SRAM: Bandwidth Advantage

Why SRAM is fast:

- 6T cell with direct bit-line access
- On-die: no interposer, no package crossing
- Sub-nanosecond access latency

Specs (on-die, per card):

Bandwidth	**150 TB/s**
Capacity	**4 GB**
Latency	∼1 ns
I/O energy	∼0.5 pJ/bit

SRAM: Bandwidth Advantage

Why SRAM is fast:

- 6T cell with direct bit-line access
- On-die: no interposer, no package crossing
- Sub-nanosecond access latency

Specs (on-die, per card):

Bandwidth	**150 TB/s**
Capacity	**4 GB**
Latency	∼1 ns
I/O energy	∼0.5 pJ/bit

Why SRAM cannot scale:

- 6T cell is **100–150× larger** than DRAM 1T1C
- Leakage: ∼1 nW/bit (significant at GB scale)
- Practical limit: ∼**4 GB** per card
- Cost: ∼$1000/GB vs. ∼$10/GB for DRAM

Llama-70B on SRAM

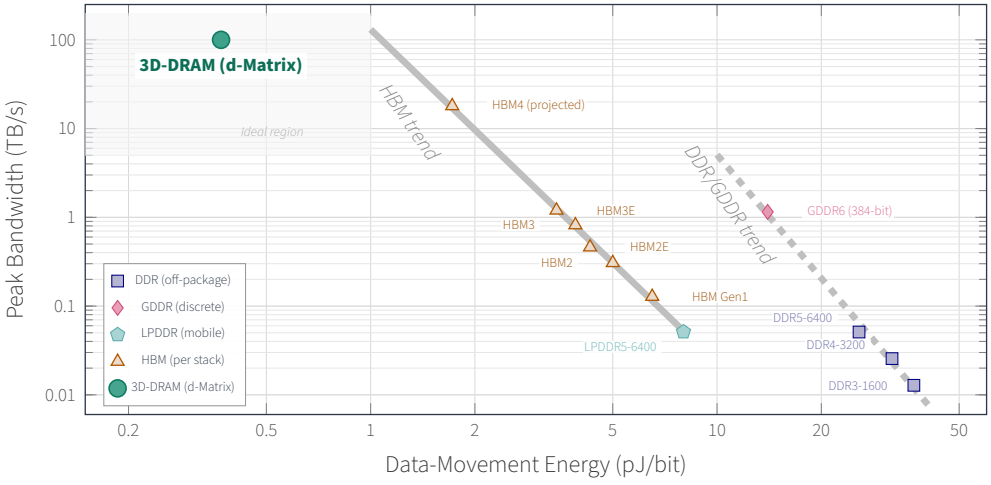
70 GB weights + KV cache ÷ 4 GB/card

⇒ TP = 8, PP = 4 = **32 cards**

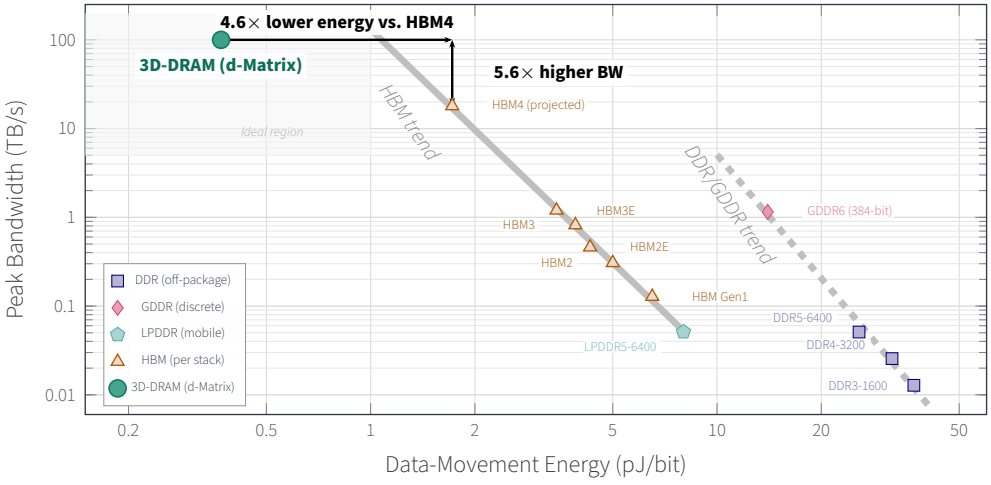
(163 collective ops every token)

Very fast memory ⇒ but **4 GB** cannot hold a 70B model.

DRAM Landscape: The Power-Normalized Bandwidth (PN) Inversion Law



DRAM Landscape: The Power-Normalized Bandwidth (PN) Inversion Law



HBM: Capacity Advantage

Why HBM has capacity:

- 1T1C DRAM cell (dense storage)
- 3D stacking: 8–12 dies per stack
- Wide bus: 1024-bit × 8 channels

Specs (HBM3/3e, per package):

Capacity	**192 GB**
Bandwidth	**18 TB/s** (6 stacks)
Latency	∼20–30 ns
I/O energy	∼2–3 pJ/bit

HBM: Capacity Advantage

Why HBM has capacity:

- 1T1C DRAM cell (dense storage)
- 3D stacking: 8–12 dies per stack
- Wide bus: 1024-bit × 8 channels

Specs (HBM3/3e, per package):

Capacity	**192 GB**
Bandwidth	**18 TB/s** (6 stacks)
Latency	∼20–30 ns
I/O energy	∼2–3 pJ/bit

Why HBM bandwidth is limited:

- 2.5D: memory *beside* logic on interposer
- Horizontal mm-scale routing
- Bump pitch: 55 μm limits I/O density
- PHY overhead at each stack interface

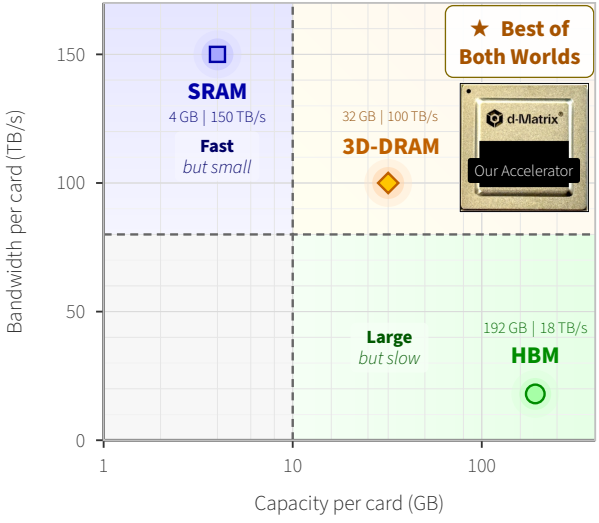
Llama-70B on HBM

75 GB read/token ÷ 18 TB/s = **4.2 ms**

That's 8–42% of a 10–50 ms TPOT budget spent *just* on memory reads.

Plenty of capacity ⇒ but **18 TB/s** leaves the accelerator starving.

The Fundamental Trade-off

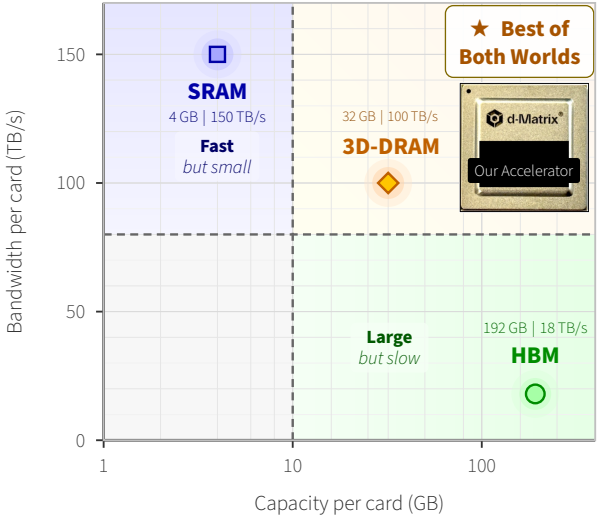


How do we reach the upper-right?

Stack memory *below* logic, face-to-face:

- Vertical μbumps (μm, not mm)
- $C \propto L$: shorter wires $\Rightarrow$ lower energy
- 36 μm pitch $\Rightarrow$ **700 conn/mm^2**
  (vs. 55 μm / $\sim$100 conn/mm^2 for HBM)
- More parallel wires, each at lower pJ/bit

The Fundamental Trade-off



How do we reach the upper-right?

Stack memory *below* logic, face-to-face:

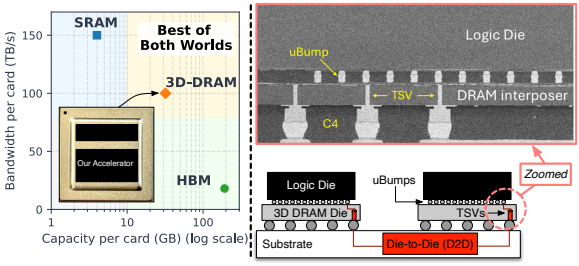
- Vertical μbumps (μm, not mm)
- $C \propto L$: shorter wires $\Rightarrow$ lower energy
- 36 μm pitch $\Rightarrow$ **700 conn/mm^2**
  (vs. 55 μm / $\sim$100 conn/mm^2 for HBM)
- More parallel wires, each at lower pJ/bit

The Opportunity

3D face-to-face stacking delivers **7$\times$ denser interconnect** and **6–8$\times$ lower I/O energy** than HBM $\Rightarrow$ enough to reach 100 TB/s at 32 GB per card.

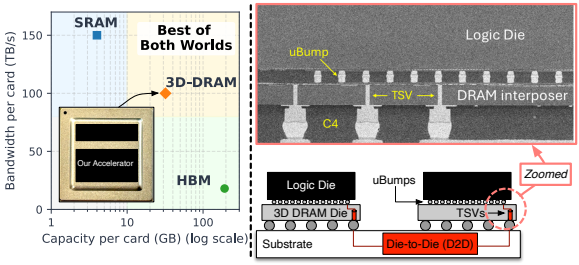
3D-DRAM: The Opportunity

First-of-a-Kind 3D-DRAM Accelerator for Gen-AI [Nair et al., ISCA 2026]



Logic die bonded face-to-face to 3D-DRAM die.
Vertical interconnects replace mm-scale HBM traces.

First-of-a-Kind 3D-DRAM Accelerator for Gen-AI [Nair et al., ISCA 2026]



Logic die bonded face-to-face to 3D-DRAM die.
Vertical interconnects replace mm-scale HBM traces.

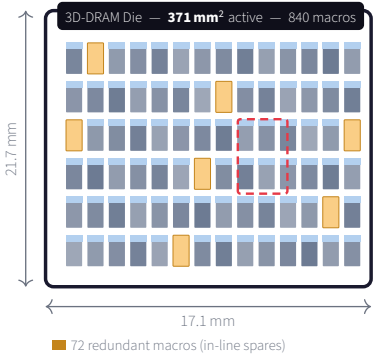
	HBM	3D-DRAM
Topology	2.5D (beside)	F2F (below)
I/O energy	2–3 pJ/bit	0.2–0.3 pJ/bit
BW / card	18 TB/s	100+ TB/s
Capacity	192 GB	32 GB

5.6× higher BW

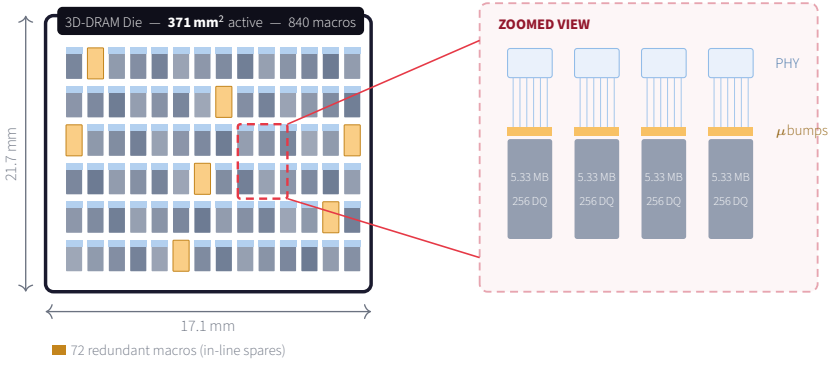
5× lower energy/bit

7× denser I/O

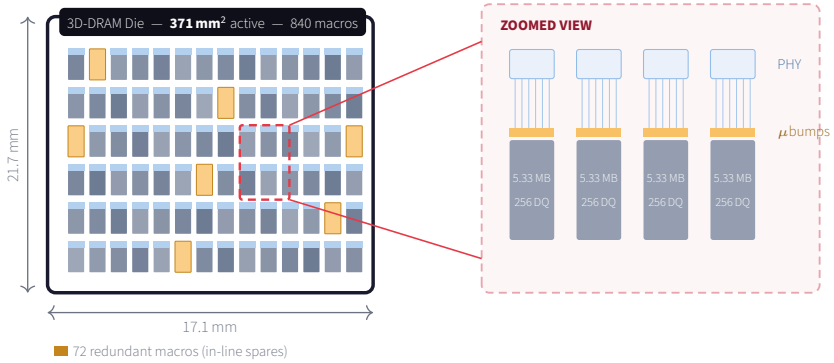
Raptor 3D-DRAM Die: 840 Macros



Raptor 3D-DRAM Die: 840 Macros



Raptor 3D-DRAM Die: 840 Macros



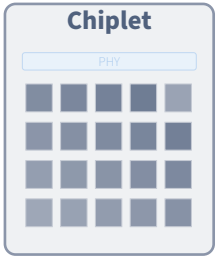
840
macros (768+72 sp.)

371 mm²
active area

4 GB
per chiplet

100 TB/s
per 8-chiplet card

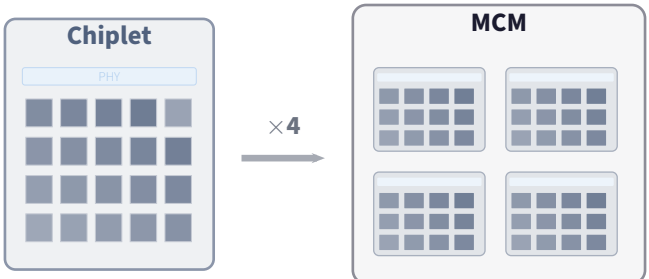
From Chiplet to Card



840 macros | 4 GB

15+ TB/s

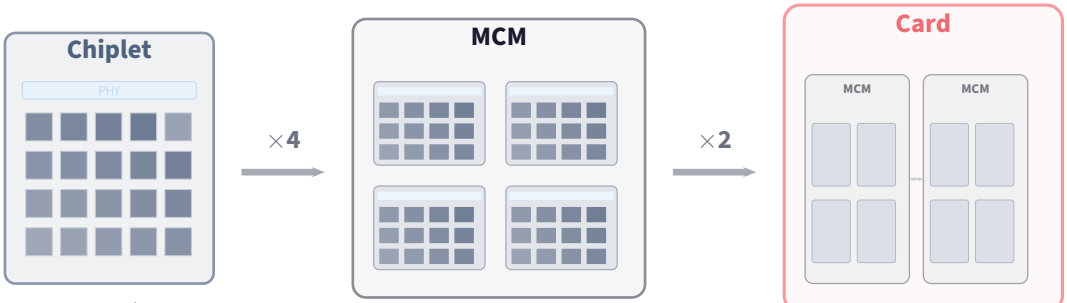
From Chiplet to Card



840 macros | 4 GB
15+ TB/s

4 chiplets | 16 GB
60+ TB/s

From Chiplet to Card



840 macros | 4 GB
15+ TB/s

4 chiplets | 16 GB
60+ TB/s

8 chiplets | 32 GB
100+ TB/s

From Concept to Silicon: Three Hard Problems

840 Banks, 18 Engineering Problems, 3 Challenges

eighteen distinct engineering problems, clustered into three challenges

Feeding the engines
bank mapping, overfetch, address interleaving …

Paying for the bits
I/O energy, toggling, no sideband pins …

Surviving the heat
105°C retention, faults, in-place repair …

840 Banks, 18 Engineering Problems, 3 Challenges

eighteen distinct engineering problems, clustered into three challenges

Feeding the engines
bank mapping, overfetch, address interleaving …

Paying for the bits
I/O energy, toggling, no sideband pins …

Surviving the heat
105°C retention, faults, in-place repair …

These three decide whether the technology is *viable* — and each one concludes with a concrete, quantitative result.

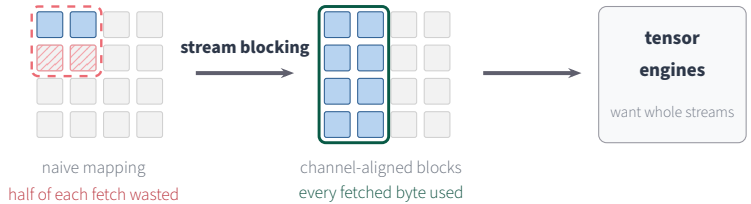
One slide per challenge: the problem, the mechanism, **the result**.

Challenge 1: Feeding the Engines

Problem. Tensor engines want long, contiguous streams — but the die offers **840 small banks**. Naive mappings fetch data the engines never use.

Challenge 1: Feeding the Engines

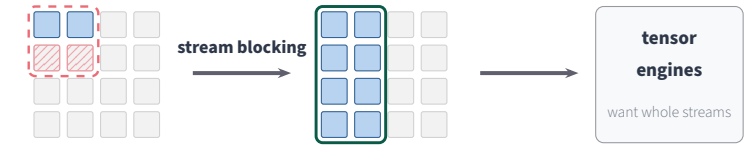
Problem. Tensor engines want long, contiguous streams — but the die offers **840 small banks**. Naive mappings fetch data the engines never use.



The mechanism: stream blocking. Group banks into channel-aligned blocks so access patterns map onto *whole* fetches — no partial reads, no wasted bytes.

Challenge 1: Feeding the Engines

Problem. Tensor engines want long, contiguous streams — but the die offers **840 small banks**. Naive mappings fetch data the engines never use.



The mechanism: stream blocking. Group banks into channel-aligned blocks so access patterns map onto *whole* fetches — no partial reads, no wasted bytes.

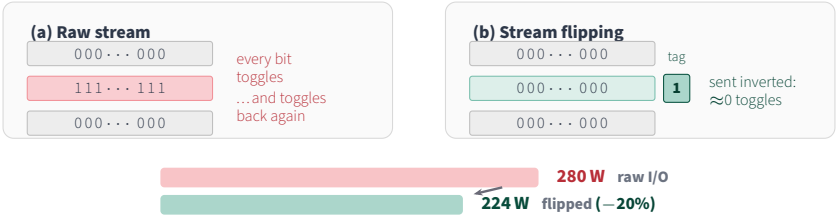
The result: the full 100 TB/s remains usable bandwidth.
Overfetch is eliminated by construction.

Challenge 2: The I/O Power Budget

Problem. 100 TB/s $\times$ 0.35 pJ/bit $=$ **280 W** of I/O alone — and the classic fix (DBI) needs a **sideband pin** this interface doesn't have.

Challenge 2: The I/O Power Budget

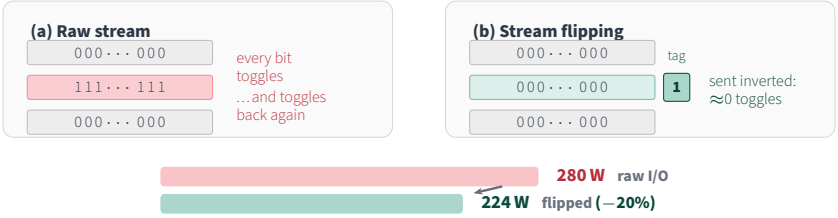
Problem. 100 TB/s × 0.35 pJ/bit = **280 W** of I/O alone — and the classic fix (DBI) needs a **sideband pin** this interface doesn't have.



The mechanism: stream flipping — pinless DBI. If most bits would toggle vs. the *previous* flit, send it inverted; the 1-bit tag is co-located with ECC metadata — no new pin, no PHY change.

Challenge 2: The I/O Power Budget

Problem. 100 TB/s × 0.35 pJ/bit = **280 W** of I/O alone — and the classic fix (DBI) needs a **sideband pin** this interface doesn't have.



The mechanism: stream flipping — pinless DBI. If most bits would toggle vs. the *previous* flit, send it inverted; the 1-bit tag is co-located with ECC metadata — no new pin, no PHY change.

The result: 20% of the I/O power reclaimed, for 0.8% overhead — a purely architectural mechanism, with no additional pins.

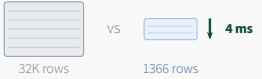
Challenge 3: Reliability at 105°C

Problem. Stacked on hot logic, DRAM runs at **105°C**: retention falls 32 ms $\rightarrow$ 4 ms (**8**$\times$ more refresh), and every fault must be repaired *in place*.

Challenge 3: Reliability at 105°C

Problem. Stacked on hot logic, DRAM runs at **105°C**: retention falls 32 ms → 4 ms (**8×** more refresh), and every fault must be repaired *in place*.

Small, deep banks



Shallow banks sweep refresh rapidly — an 8× rate at negligible cost.

Bank chaining



A faulty bank shifts one hop to a *local* spare — channels stay symmetric.

Refresh as a policy



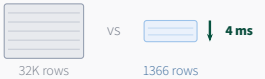
Temperature-aware refresh + ECC roots: AVATAR, Morphable ECC.

Corollary: the 4 ms sweep completes within the 8.8 ms disturbance window — **refresh itself mitigates RowHammer**.

Challenge 3: Reliability at 105°C

Problem. Stacked on hot logic, DRAM runs at **105°C**: retention falls 32 ms → 4 ms (**8×** more refresh), and every fault must be repaired *in place*.

Small, deep banks



Shallow banks sweep refresh rapidly — an 8× rate at negligible cost.

Bank chaining



A faulty bank shifts one hop to a *local* spare — channels stay symmetric.

Refresh as a policy



Temperature-aware refresh + ECC roots: AVATAR, Morphable ECC.

Corollary: the 4 ms sweep completes within the 8.8 ms disturbance window — **refresh itself mitigates RowHammer**.

The result: **<1.4% bandwidth lost** to 8× refresh, at 105°C — reliability and security at negligible cost.

Looking Forward

Roadmap: Hybrid Bonding

Current (F2F μBump):	
Bonding pitch	36 μm
I/O energy	**0.35 pJ/bit**
DRAM stack	1-high
Bandwidth	**100 TB/s**/card
Capacity	**32 GB**/card

Future (Hybrid Bonding):	
Bonding pitch	$<$2 μm
I/O energy	$\sim$**0.05 pJ/bit**
DRAM stack	8-high
Bandwidth	**800+ TB/s**/card
Capacity	**256 GB**/card

Today's Silicon

Channel Mapping + DBI +
thermal and redundancy features.
Production-validated at 105°C.

Roadmap: Hybrid Bonding

Current (F2F μBump):	
Bonding pitch	36 μm
I/O energy	**0.35 pJ/bit**
DRAM stack	1-high
Bandwidth	**100 TB/s**/card
Capacity	**32 GB**/card

Future (Hybrid Bonding):	
Bonding pitch	$<$2 μm
I/O energy	$\sim$**0.05 pJ/bit**
DRAM stack	8-high
Bandwidth	**800+ TB/s**/card
Capacity	**256 GB**/card

Today's Silicon

Channel Mapping + DBI +
thermal and redundancy features.
Production-validated at 105°C.

Projected Improvement

8$\times$ capacity, **8**$\times$ bandwidth
7$\times$ lower energy per bit
$\Rightarrow$ **56**$\times$ bandwidth-per-watt

Channel topology and architectural mechanisms $\Rightarrow$ **56**$\times$ **more bandwidth per watt.**

Raptor 3D-DRAM: Takeaways

The problem:

- Generative AI decode is fundamentally memory-bound
- SRAM: bandwidth without capacity
- HBM: capacity without bandwidth

Our contribution:

- 3D-DRAM closes the gap ⇒ bandwidth *and* capacity in one package
- Three co-designed solutions:
  stream blocking, stream flipping,
  topology-preserving redundancy
- Silicon-validated at 105°C

Results:

Metric	Value
TPS/card vs. HBM	**4.7**×
TPS/card vs. SRAM	**2.4**×
Responsiveness vs. HBM	**9.9**×
Bandwidth/card	**100 TB/s**
Capacity/card	**32 GB**
I/O energy	**0.35 pJ/bit**
Network sensitivity	Lower than SRAM
Roadmap (HB)	**56**× BW/W

Raptor 3D-DRAM: Takeaways

The problem:

- Generative AI decode is fundamentally memory-bound
- SRAM: bandwidth without capacity
- HBM: capacity without bandwidth

Our contribution:

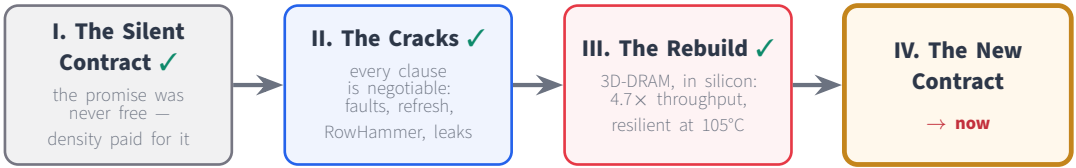
- 3D-DRAM closes the gap ⇒ bandwidth *and* capacity in one package
- Three co-designed solutions: stream blocking, stream flipping, topology-preserving redundancy
- Silicon-validated at 105°C

Results:

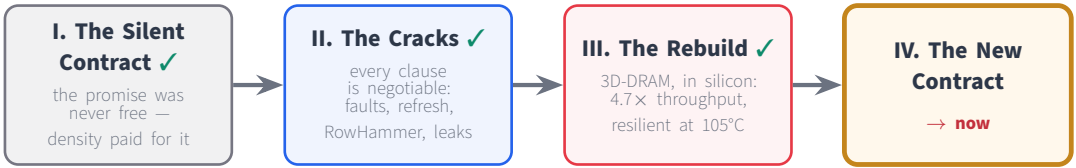
Metric	Value
TPS/card vs. HBM	**4.7**×
TPS/card vs. SRAM	**2.4**×
Responsiveness vs. HBM	**9.9**×
Bandwidth/card	**100 TB/s**
Capacity/card	**32 GB**
I/O energy	**0.35 pJ/bit**
Network sensitivity	Lower than SRAM
Roadmap (HB)	**56**× BW/W

The memory wall is a technology problem. **3D-DRAM is the solution.**

Three Acts Down, One to Go



Three Acts Down, One to Go

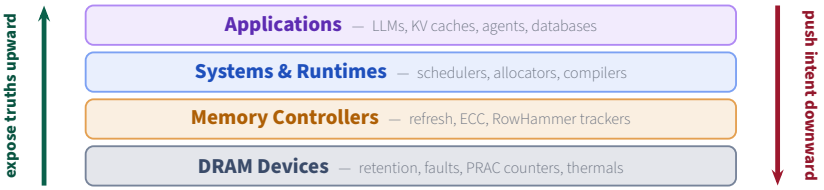


We broke the old promise, and we rebuilt the substrate.

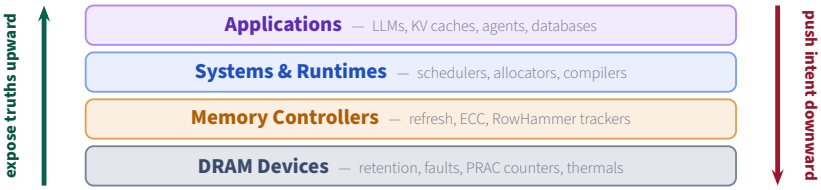
So what should memory promise now?

The New Contract

The New Contract: Devices, Architectures, Systems, Applications



The New Contract: Devices, Architectures, Systems, Applications



1. Expose

Surface device state: XED [ISCA'16], PRAC counters [HPCA'25].

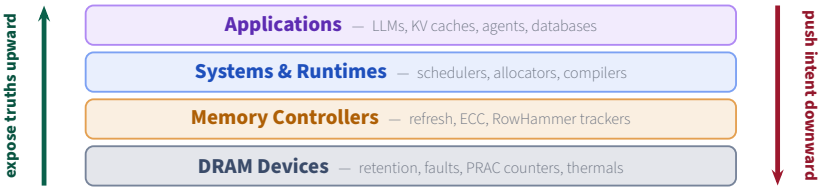
2. Integrate

One budget for reliability *and* security: SYNERGY, SafeGuard, Raptor's ECC+DBI.

3. Adapt

Mechanisms follow the workload: AVATAR, Keyformer, stream flipping.

The New Contract: Devices, Architectures, Systems, Applications



1. Expose

Surface device state: XED [ISCA'16], PRAC counters [HPCA'25].

2. Integrate

One budget for reliability *and* security: SYNERGY, SafeGuard, Raptor's ECC+DBI.

3. Adapt

Mechanisms follow the workload: AVATAR, Keyformer, stream flipping.

Memory's future is not a bigger passive array — it is an **active, resilient, application-aware substrate.**

What This Asks of This Room

If you build systems

Demand interfaces that expose device truth — activation counters, error telemetry, thermal state — and treat them as *first-class security signals*, not diagnostics.

What This Asks of This Room

If you build systems

Demand interfaces that expose device truth — activation counters, error telemetry, thermal state — and treat them as *first-class security signals*, not diagnostics.

If you build devices

Publish honest bits. One exposed detection bit bought **172**× reliability [XED]; exposed activation counters made principled RowHammer defense possible [PRAC].

What This Asks of This Room

If you build systems

Demand interfaces that expose device truth — activation counters, error telemetry, thermal state — and treat them as *first-class security signals*, not diagnostics.

If you build devices

Publish honest bits. One exposed detection bit bought **172**× reliability [XED]; exposed activation counters made principled RowHammer defense possible [PRAC].

If you do research

Work the seams between devices, controllers, systems, and models. Memory contracts are *policies* — specification, delegation, revocation — this community's core expertise.

What This Asks of This Room

If you build systems

Demand interfaces that expose device truth — activation counters, error telemetry, thermal state — and treat them as *first-class security signals*, not diagnostics.

If you build devices

Publish honest bits. One exposed detection bit bought **172**× reliability [XED]; exposed activation counters made principled RowHammer defense possible [PRAC].

If you do research

Work the seams between devices, controllers, systems, and models. Memory contracts are *policies* — specification, delegation, revocation — this community's core expertise.

Every access-control decision ends at a bit in DRAM.
Let's make that bit worth trusting.

Open Problems at the Seam

For architects

- **Verifiable contracts:** attest what a device *does* — its refresh, its mitigations — not just its firmware
- **Mitigations with proofs:** defenses that provably don't leak (the PRACLeak lesson)
- **Model-aware integrity:** error coding shaped to weights (Structural Coding is a start)

Open Problems at the Seam

For architects

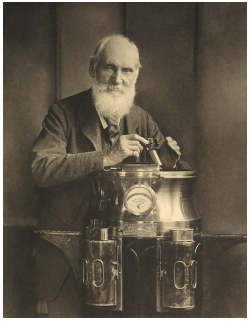
- **Verifiable contracts:** attest what a device *does* — its refresh, its mitigations — not just its firmware
- **Mitigations with proofs:** defenses that provably don't leak (the PRACLeak lesson)
- **Model-aware integrity:** error coding shaped to weights (Structural Coding is a start)

For this community

- **Self-managing memory leaks:** every adaptive policy is an observable — audit them
- **Pooled-memory trust:** CXL sharing — whose counters, whose refresh, whose keys?
- **Access control *for* memory state:** who may read an activation counter? Who may tune a refresh rate?

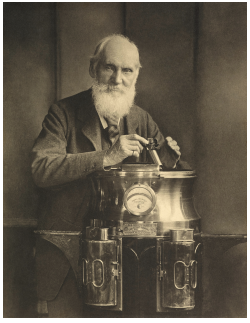
The next RowHammer is already sitting in a mechanism **no one has audited yet.**

The Man on the Agamemnon



He sailed with the failed cable of 1858 — and stayed for the fix: low voltages, honest measurement, and a substrate progressively engineered better. The **1866** cable worked.

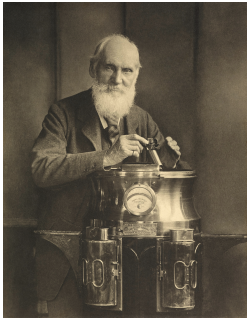
The Man on the Agamemnon



He sailed with the failed cable of 1858 — and stayed for the fix: low voltages, honest measurement, and a substrate progressively engineered better. The **1866** cable worked.

William Thomson was knighted for it, and is better known today by another name: **Lord Kelvin** — absolute temperature, thermodynamics, the science of signals.

The Man on the Agamemnon



He sailed with the failed cable of 1858 — and stayed for the fix: low voltages, honest measurement, and a substrate progressively engineered better. The **1866** cable worked.

William Thomson was knighted for it, and is better known today by another name: **Lord Kelvin** — absolute temperature, thermodynamics, the science of signals.

The Atlantic was not crossed by driving the cable harder.

It was crossed by measuring the substrate — and engineering it until it held.

The Memory Substrate of Tomorrow

Resilient DRAM for Secure and Intelligent Computing

Thank You