Scaling DRAM Technology to Meet Future Demands: System Challenges and Opportunities

Steven C. Woo, Rambus Inc.
Wendy Elsasser, Rambus Inc.
Robert Palmer, Rambus Inc.
Taeksang Song, Samsung Electronics

ISCA 2026 Tutorial
Raleigh, North Carolina, USA
June 27, 2026

Today's Presenters

Steven Woo is a Fellow and Distinguished Inventor at Rambus Inc., where he leads research in Rambus Labs on advanced memory systems for accelerators and computing infrastructure, and manages a team of senior architects. Since joining Rambus, Steve has worked in various roles leading architecture, technology, and performance analysis efforts, and in marketing and product planning roles leading strategy and customer programs. He has more than 30 years of experience working on advanced memory systems and holds more than 100 US and international patents. Steve received his PhD and MS degrees in Electrical Engineering from Stanford University, and Master of Engineering and BS Engineering degrees from Harvey Mudd College.

Wendy Elsasser is a Technical Director of Research Science at Rambus. She works in the Rambus Labs R&D division investigating future system architectures and developing innovative solutions to address the impact on the memory sub-systems. She has over 25 years of experience in industry, starting with semi-custom micro-controller design, test, and implementation. Over the last 20 years, her focus has been on memory sub-systems, primarily external DRAM. Her experience includes DRAM controller architecture, design, and validation as well as active contributions to consortiums and standards bodies. Specifically, she was a leader in the Gen-Z consortium and JEDEC, helping to define future memory interfaces and DRAM standards. Her work has resulted in 15 patents.

Robert Palmer is a Senior Technical Director of Research Science at Rambus Inc., leading research in future memory module and memory buffer chip architectures. He has over 25 years of industry experience developing silicon IP and communication ICs, spanning the design of high-speed wireline transceiver circuits, serial link and memory controller PHY microarchitectures, and DDR and CXL memory buffer chip architectures and microarchitectures. In addition to his tenure at Rambus, Robert has held positions at Velio Communications and Nvidia Research. He holds over 60 US and international patents.

Taeksang Song is a Corporate Vice President at Samsung Electronics where he is leading a team dedicated to pioneering cutting-edge technologies including CAMM, MRDIMM, CXL memory expanders, fabric attached memory solutions and processing near memory to meet the evolving demands of next-generation data-centric AI architectures. He has 20 years' professional experience in memory and sub-system architecture, interconnect protocols, system-on-chip design and collaborating with CSPs to enable heterogeneous computing infrastructure. Prior to joining Samsung Electronics, he worked at Rambus Inc., Micron Technology and SK hynix in lead architect roles for the emerging memory controllers and systems. Taeksang received his PhD from KAIST, South Korea, in 2006. He has authored and co-authored over 20 technical papers and holds over 50 U.S. patents.

Other Contributors to this Tutorial

- Our thanks to others at Rambus who helped make this tutorial possible
  - Brent Haukness
  - Michael R. Miller
  - Thomas Vogelsang
  - Lidia Warnes
  - The greater Rambus Labs Team

- These slides will be posted to the tutorial web page shortly after the conference
  https://www.rambus.com/rambus-dram-tutorial/

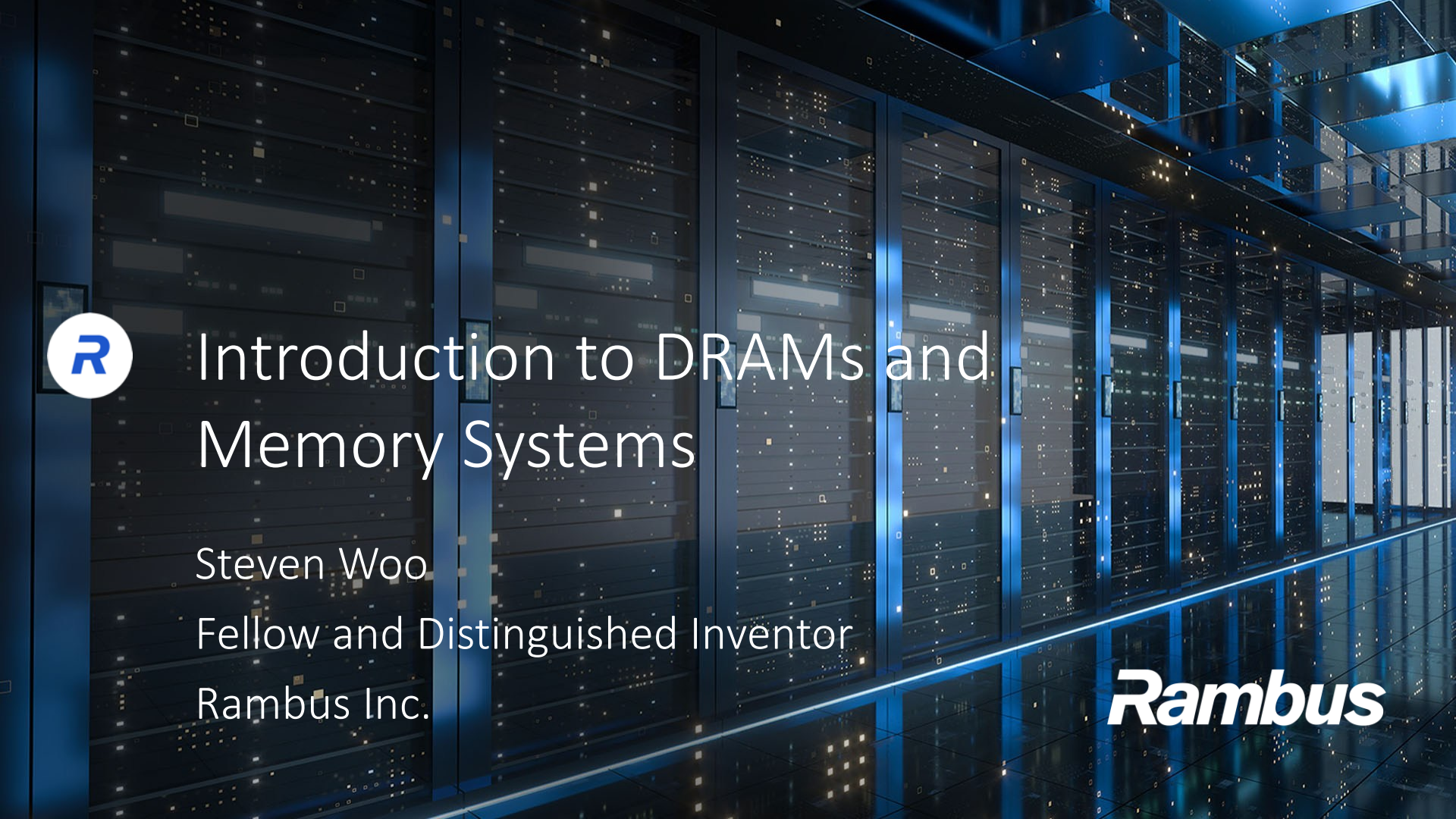
- We welcome feedback, please send thoughts on the material and additional areas to cover to swoo@rambus.com.

Tutorial Outline (Morning)

8:00	Introduction to DRAMs and Memory Systems	Steve
8:30	History and Markets	Steve
9:00	DRAM Architecture	Taeksang
10:00	Coffee Break	
10:30	Market Demands and Scaling Needs: Why One Size Doesn't Fit All	Wendy
11:00	Memory Controllers	Wendy
11:30	Enabling Communication with DRAMs: Modules, PHYs, and Buffer Chips (part 1)	Robert
12:00	Lunch Break	

Tutorial Outline (Afternoon)

1:30	Enabling Communication with DRAMs: Modules, PHYs, and Buffer Chips (part 2)	Robert
2:30	Future Memory Solutions	Taeksang
3:30	Coffee Break	
4:00	RAS – Reliability, Availability, and Serviceability	Wendy
4:30	Challenges for the Future	Steve
5:00	Conclusions	Steve
5:15	End	



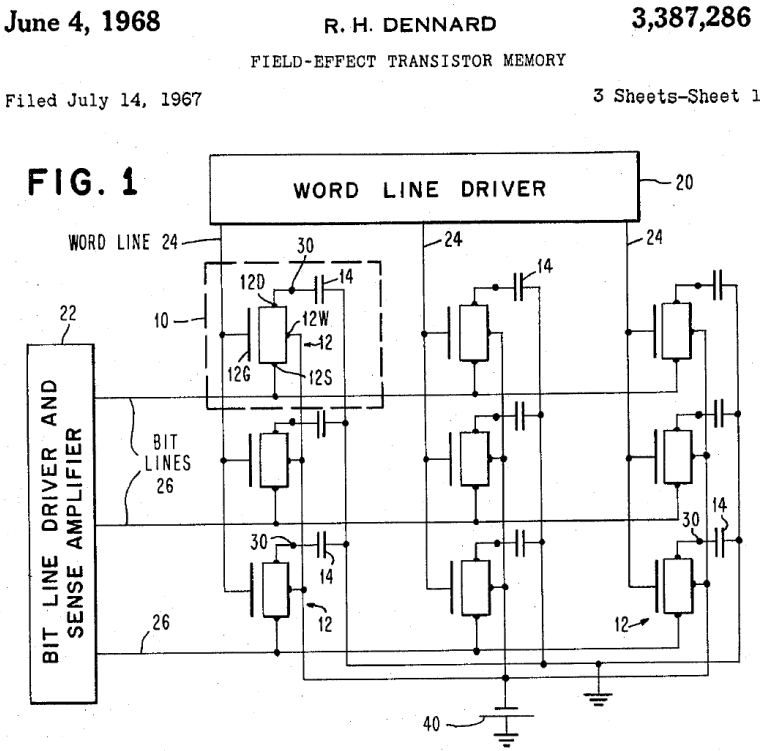
Introduction to DRAMs and Memory Systems

Steven Woo
Fellow and Distinguished Inventor
Rambus Inc.

Rambus

The Fundamental DRAM Building Block: The 1T1C Bit Cell

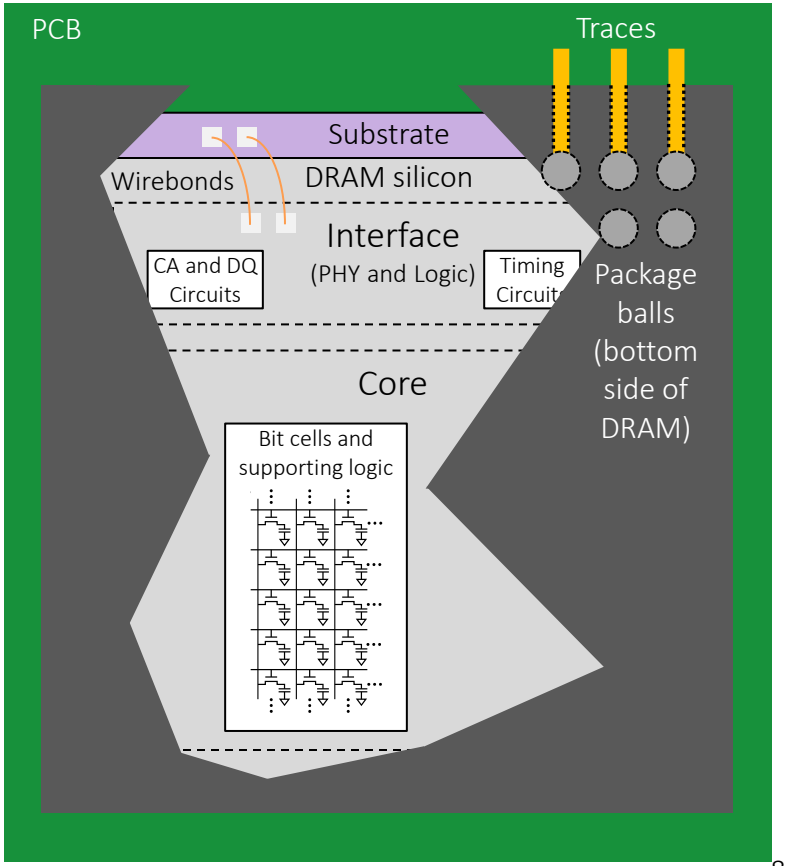
- *Field Effect Transistor Memory* first described in a patent filed by Robert Dennard in 1967
  - Charge on capacitor represents “0” or “1”
  - Access transistor to read and write a bit cell
- DRAM: Dynamic Random Access Memory
  - Dynamic: Bit cells are volatile (lose charge over time), must be *refreshed.* When power is removed, data is lost.
  - Random Access: Access any location without needing to sequentially search (like tape), latency to access bit cells is (more or less) uniform
- Bit cells aggregated into dense storage arrays that share resources, enabling high capacity



DRAM Interface and Core

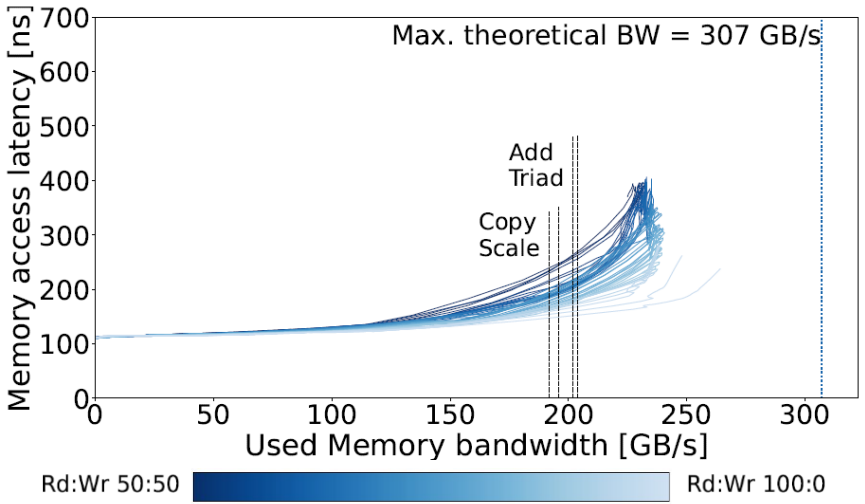
- DRAM Silicon
  - Interface: Circuits that interface to the host
  - Core: Bit cells and supporting logic
- Substrate
  - Connects DRAM pads to package balls (for example, with wirebond packaging)
  - Redistributes signals from silicon to package balls
- PCB
  - Connect DRAM package balls to other chips

DRAM interface and packaging are key differentiators between DRAM types



Memory System Overview: Theory and Practice

Latency Under Load Measurements
Intel Sapphire Rapids Xeon Platinum
56 Cores, 8 DDR5-4800 DIMMs (307GB/s Peak Bandwidth)



Source: P. Esmaili-Dokht et al., "A Mess of Memory System Benchmarking, Simulation and Application Profiling," 2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO), Austin, TX, USA, 2024, pp. 136-152, doi: 10.1109/MICRO61859.2024.00020.

Key Points

- Latency Under Load curves are a standard tool for characterizing memory systems
- Most of the latency isn't from the DRAM, DRAM produces data in ~18-38ns

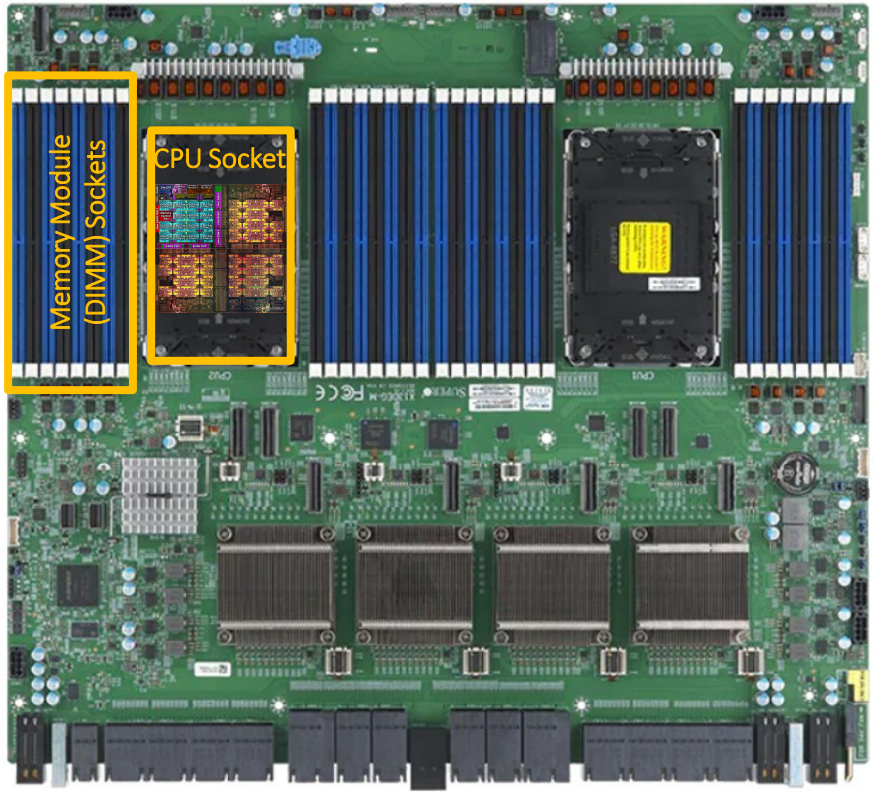
  What is the source of this extra latency?
- Demand for more bandwidth, but achieving high bandwidth utilization is getting harder

  What makes higher utilizations harder?
- Latency varies across a wide (e.g., ~4X) range

  What causes this wide variation in latency?

This tutorial will help answer these questions

Servers and DIMMs



64GB DDR5-4800 DIMM

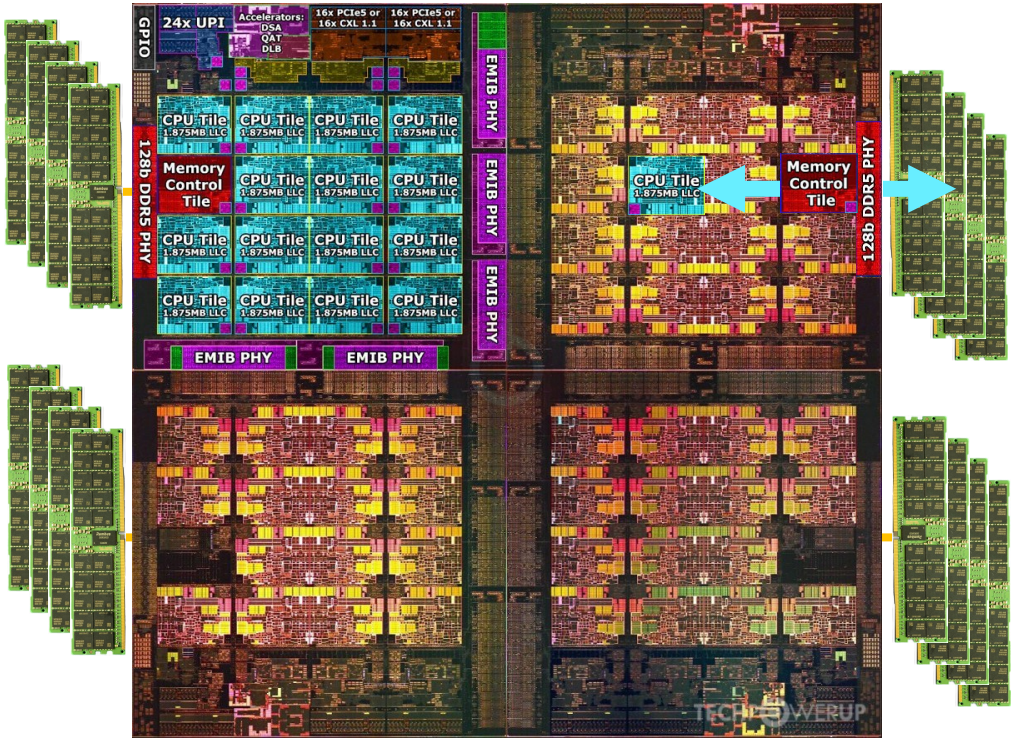
32GB DDR5-4800 DIMM

- DIMM modules
  - Serviceable: Replace in field
  - Upgradable: Add capacity and/or faster data rate
  - DIMM sockets make signal integrity more challenging
- DIMMs can have different capacities, same data rate
  - Manage cost while maintaining bandwidth to processor
  - Many applications depend heavily on bandwidth

Example Server Memory System with DDR5

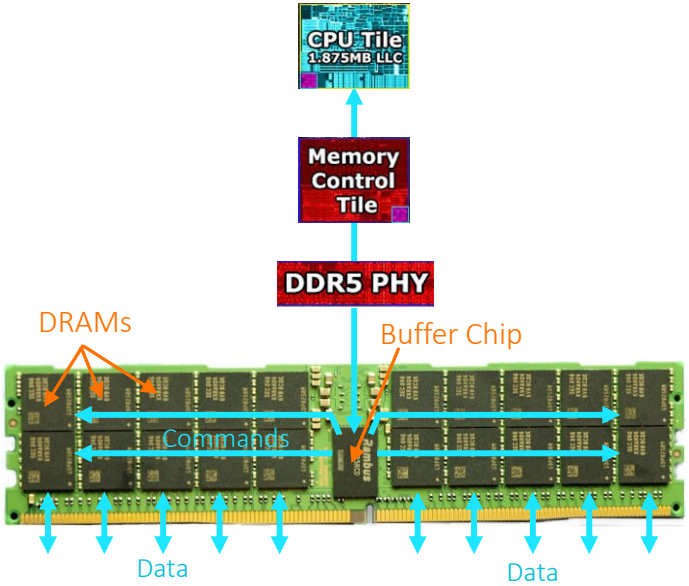
Intel Sapphire Rapids Xeon Platinum

56 Cores, 8 DDR5-4800 DIMMs (307GB/s Peak Bandwidth)



Source: https://www.techpowerup.com/cpu-specs/xeon-platinum-8480.c2958

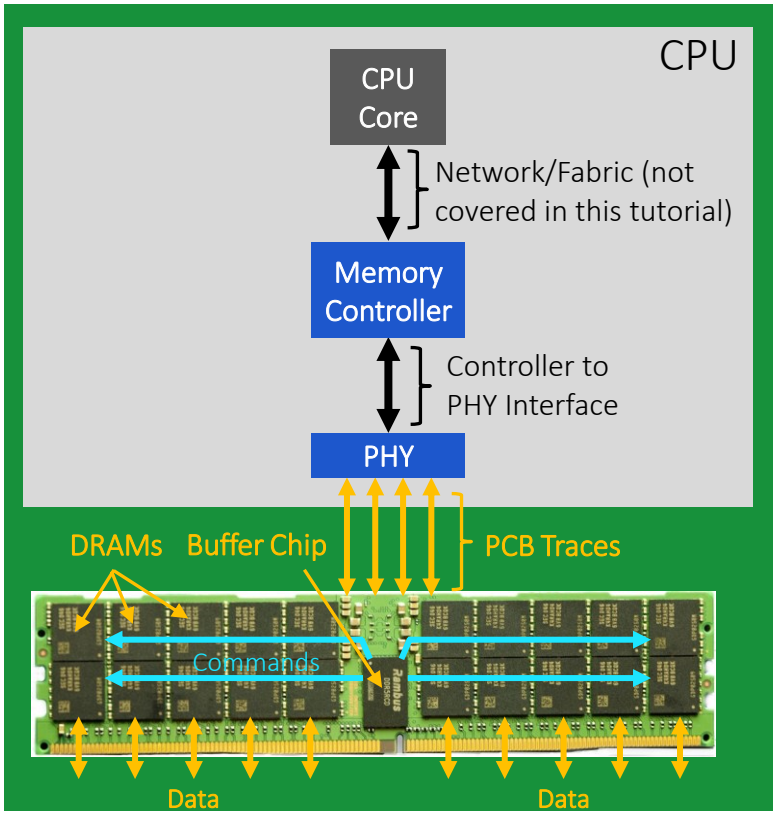
How CPUs Connect to DRAMs



Silicon between CPU cores and DRAMs is necessary for delivering high bandwidths, but adds to memory latency

Memory Systems are Much More than Just DRAMs

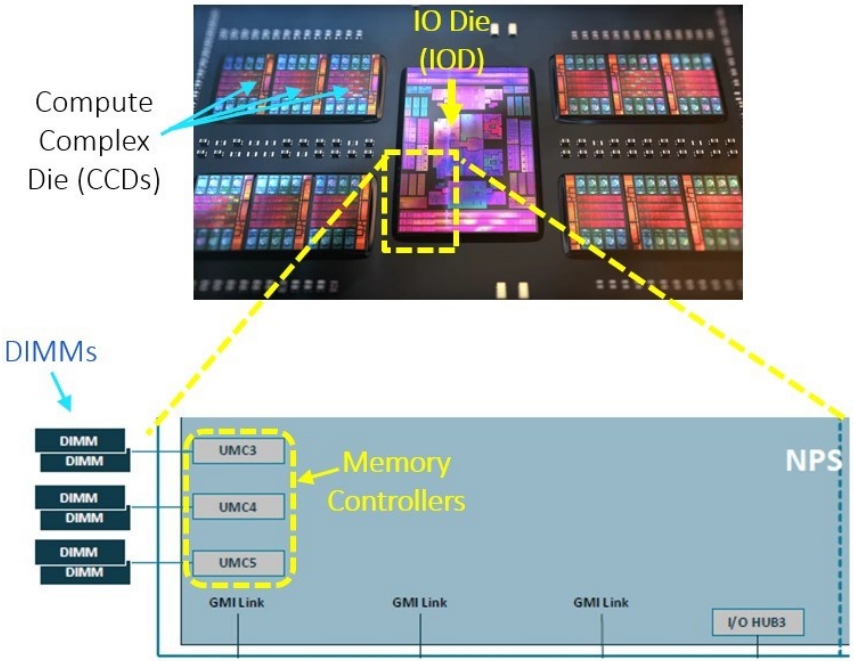
Example Server Memory System



- Lots of silicon and wiring between CPU and DRAMs
- Memory Controller (Wendy)
  - Turns CPU requests (e.g., Load/Store) into DRAM-specific commands, obeys timing constraints
  - Schedules transactions to achieve performance goals
- PHY (Robert)
  - Implement analog circuits (e.g., transceivers) that drive signals across the wires, consume lots of power
- Traces
  - Carry analog signals for commands and data
- Buffer Chips (Robert)
  - Used on some modules to improve signal integrity, reduce pin counts, achieve higher data rates
- DRAM (Taeksang)
  - Holds the data manipulated by the program

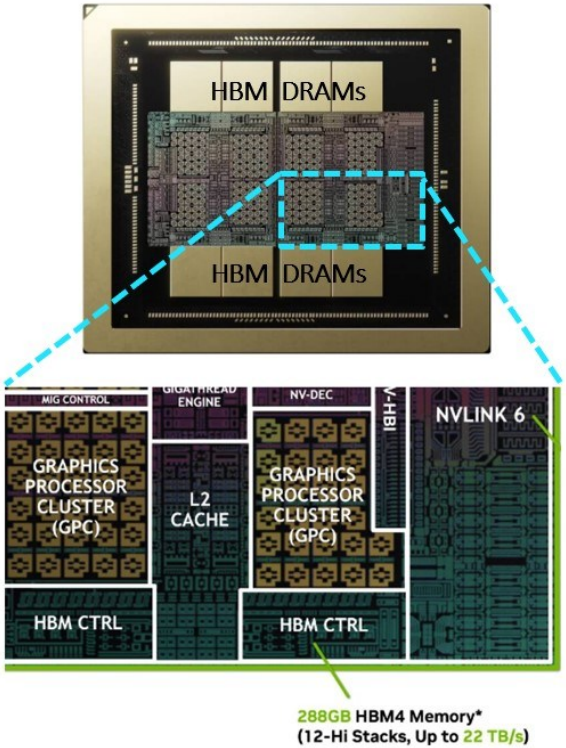
Memory Systems Similar Across Markets

AMD Epyc 9004 Chiplet Architecture with separate IO Die



Sources: https://www.techspot.com/news/96633-amd-announces-epyc-9004-up-96-cores-around.html
https://docs.amd.com/api/khub/documents/7JjmsYQvADa5z2Z~i9D_JA/content.

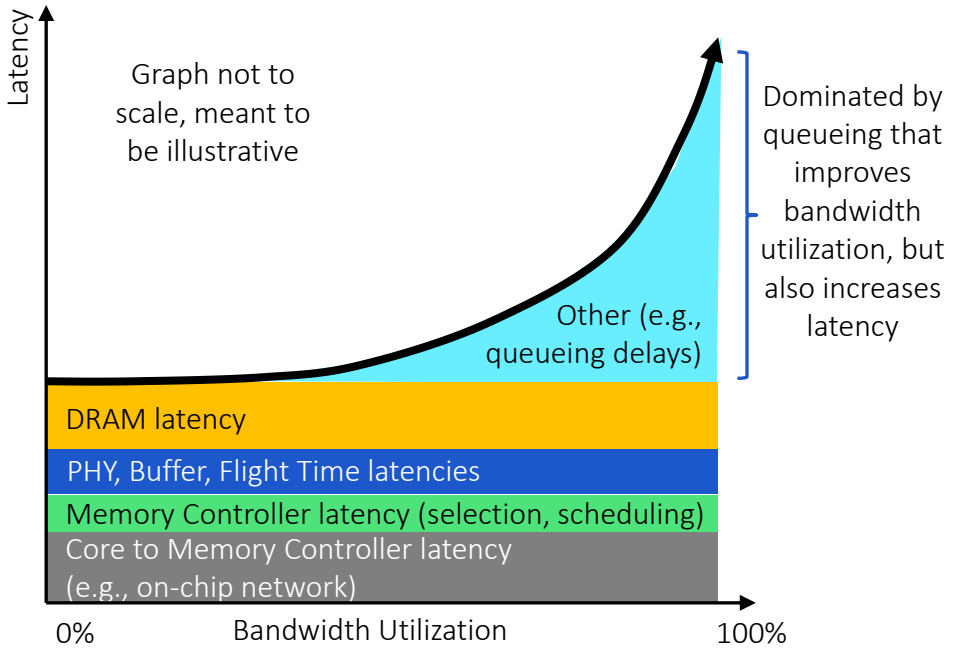
Nvidia Rubin GPU



Sources: https://newsletter.semianalysis.com/p/vera-rubin-extreme-co-design-an-evolution
https://developer.nvidia.com/blog/inside-nvidia-groq-3-lpx-the-low-latency-inference-accelerator-for-the-nvidia-vera-rubin-platform/

Revisiting Latency Under Load

- Complex path from cores to DRAMs, silicon functions are necessary but increase latency
- Many transactions in flight under high bandwidth utilization, requires queueing and reordering that increase latency
- Application performance impacted by both memory latency and bandwidth
- Memory controllers are complex and critical, schedule dynamic transaction streams to meet performance goals



Some Important DRAM and Memory System Characteristics

Latency

- Time from request issued by host or controller until data is returned to host or controller

Bandwidth

- The amount of data that can be moved in and out of the DRAM per second

Latency Under Load

- Characterization of how latency changes as a function of bandwidth delivered

Fill Frequency

- The ratio $\frac{Bandwidth}{Capacity}$, which defines how quickly the capacity of the device can be read/written

Power

- The amount of power used when transferring data, when idle, and in low-power modes

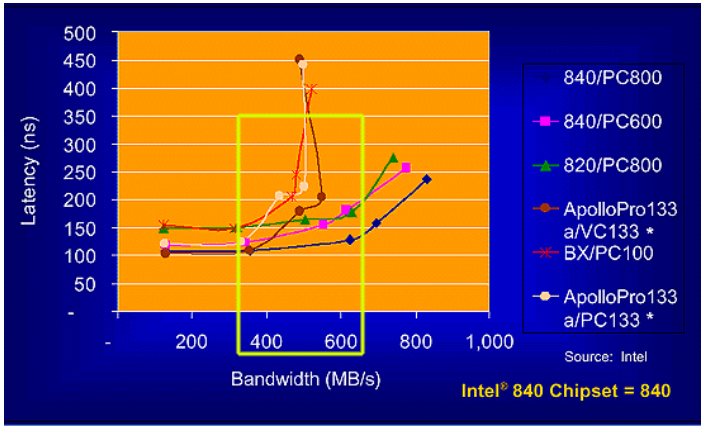
RAS/ECC/Reliability (Wendy)

- Resilience to circuit/DRAM failures, disturb effects (e.g., RowHammer), Silent Data Corruption (SDC) => increasingly important for future DRAMs and systems

Cost/bit and system cost

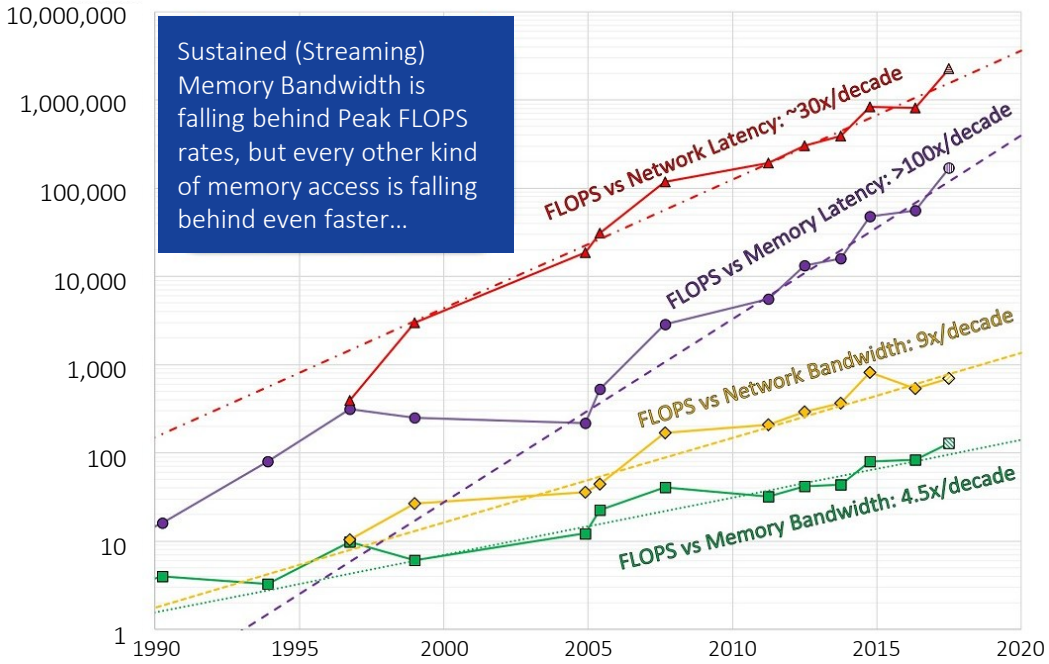
- Cost/bit of DRAM, system cost (2.5D packaging, interposers, power components, buffer chips)

Example Latency Under Load Curves



Source: AnandTech, http://www.anandtech.com/cpu/showdoc.html?i=1245&p=4

Memory and Networks Limiting System Performance



Source: hpcwire.com/2016/11/07/mccalpin-traces-hpc-system-balance-trends

- Compute continues to improve much faster than memory and networks
- Memory, network latency and bandwidth falling behind processor performance at an alarming rate
- Each year, memories and networks become even larger bottlenecks
- Several techniques developed to tolerate/mitigate these trends, but for how much longer?

New techniques and domain-specific silicon needed to address these issues

Reliability a Growing Concern

Categorization of Unexpected Interruptions During 54-day Period of Llama 3 405B Pre-training

Component	Category	Interruption Count	% of Interruptions
Faulty GPU	GPU	148	30.1%
GPU HBM3 Memory	GPU	72	17.2%
Software Bug	Dependency	54	12.9%
Network Switch/Cable	Network	35	8.4%
Host Maintenance	Unplanned Maintenance	32	7.6%
GPU SRAM Memory	GPU	19	4.5%
GPU System Processor	GPU	17	4.1%
NIC	Host	7	1.7%
NCCL Watchdog Timeouts	Unknown	7	1.7%
Silent Data Corruption	GPU	6	1.4%
GPU Thermal Interface + Sensor	GPU	6	1.4%
SSD	Host	3	0.7%
Power Supply	Host	3	0.7%
Server Chassis	Host	2	0.5%
IO Expansion Board	Host	2	0.5%
Dependency	Dependency	2	0.5%
CPU	Host	2	0.5%
System Memory	Host	2	0.5%

- Data from a 54-day period using 16K GPUs
- 419 unplanned interruptions
- ~78% of these interruptions confirmed as hardware issues, including silent data corruption
- GPU memory was the second highest cause of interruptions
- Driving the need for more automated detection and correction with techniques like on-die and system ECC

Source: A Grattafiori, et al., "The Llama 3 Herd of Models," https://arxiv.org/abs/2407.21783

History and Markets

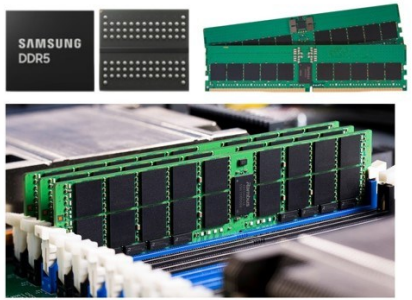
Steven Woo
Fellow and Distinguished Inventor
Rambus Inc.

Rambus

Different DRAMs for Different Markets

DRAM interfaces and packaging differ based on how DRAMs are used in the system

Compute

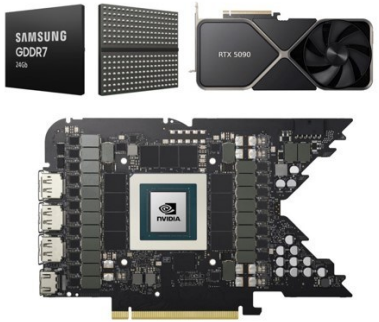


- DRAMs packaged on modules (DIMMs, CAMMs)
- Multiple DRAMs service one Read and Write transaction
- Can tolerate failure of 1 DRAM

Mobile

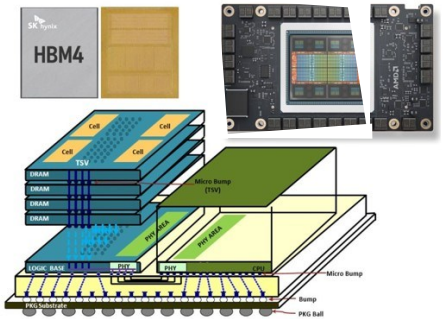
- Low active and idle power with special low power modes
- Fast wake up and power down supports bursty system activity
- Low profile supports stacking with processor die

Graphics



- Highest data rates, challenging signal integrity
- High fill frequency
- Standard manufacturing with DRAM soldered to PCB

AI/HPC

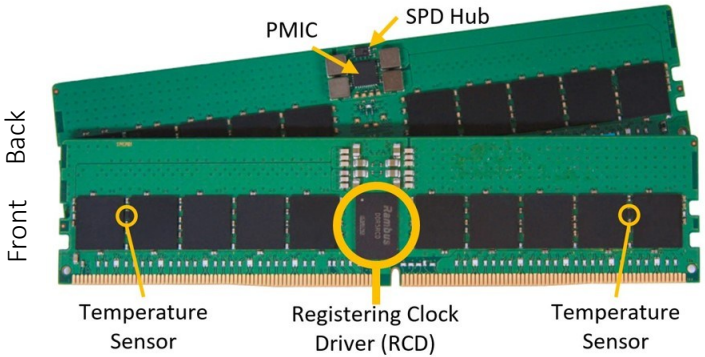


- Highest bandwidth DRAM using die stacking and base die
- Highest number of DQs
- Packaging uses an interposer to support high IO count

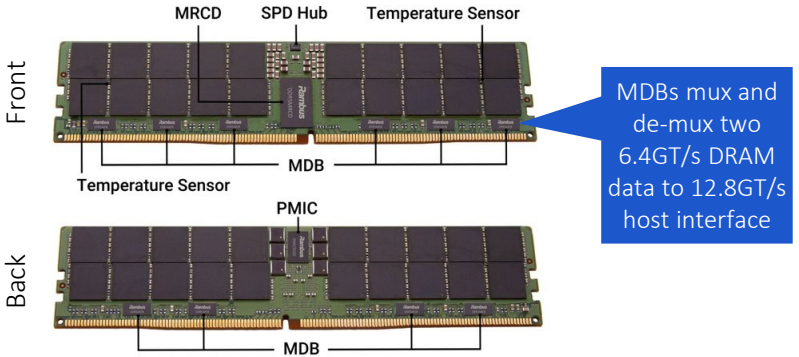
Several types of DRAMs, all use the 1T1C bit cell to store data

Compute Memory: DDR

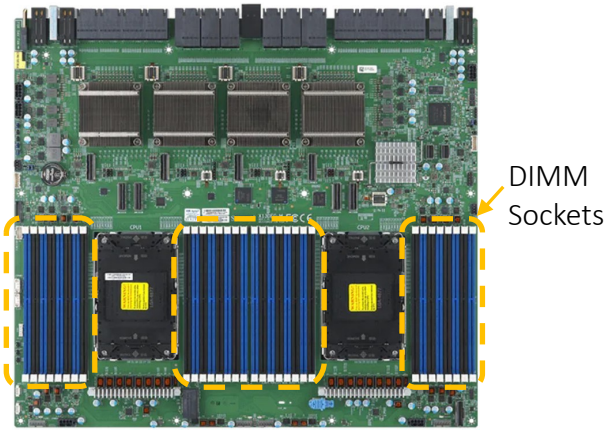
RDIMM



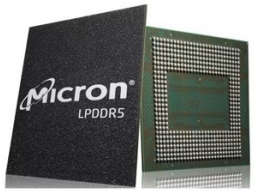
MRDIMM

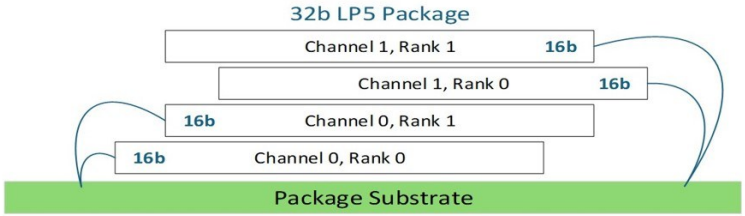


- Servers drive DDR roadmap today, was PCs
- DRAMs packaged onto DIMM Modules
- DIMMs connect to the processor through a DIMM socket => flexible, replaceable
- Transactions handled by a rank (group) of DRAMs
- High bandwidth, capacity from aggregating many DRAMs
- Can tolerate failure of 1 DRAM with Chipkill ECC



Mobile Memory: LPDDR





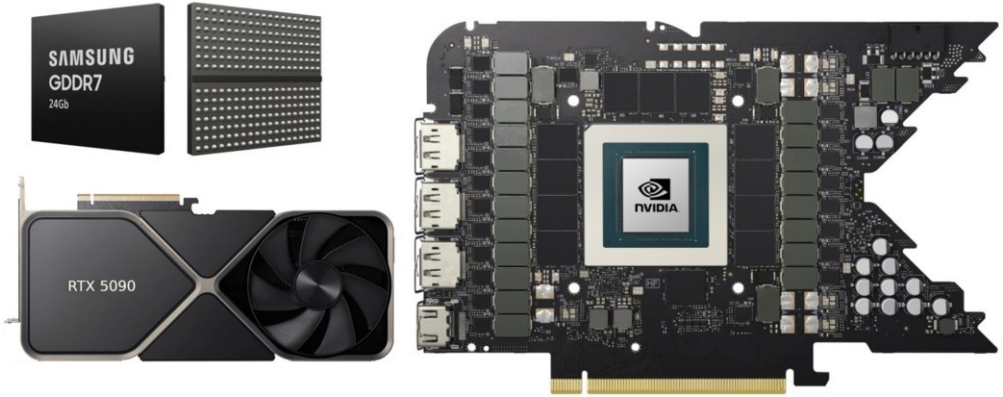
- Low active and idle power with special low power modes
- Special low power modes support system sleep modes
- Fast wake up/power down supports bursty memory system activity, fast transitions between active and sleep modes
- Low profile supports stacking with processor die

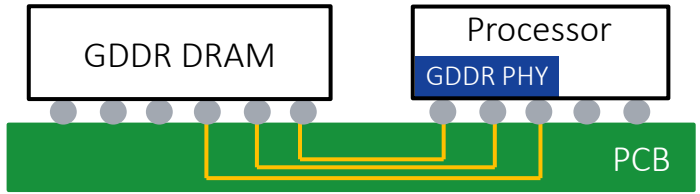
Palm Treo Pro with DDR (2008)

iPhone 4G with LPDDR2 (2010)

Early smartphones used binned DDR DRAMs, growing volumes justified development LPDDR

Graphics Memory: GDDR

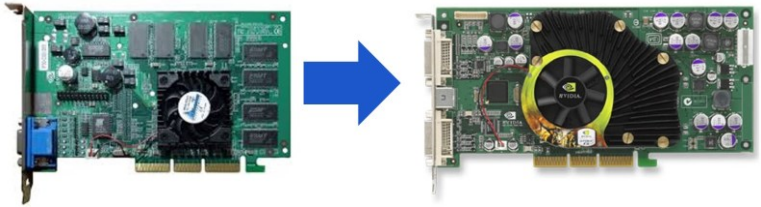




- Highest data rates, challenging signal integrity
- GDDR7: 4 8b channels = 32 DQs, up to 48Gbps (PAM3), up to 192GB/s per DRAM
- High fill frequency, good for graphics, some AI & HPC apps
- Clamshell mode offers 2X Capacity at same bandwidth
- Standard manufacturing with DRAM soldered to PCB
- Good tradeoff between DRAM cost, system cost, manufacturing complexity, and performance

nVidia GPU with DDR (2000)

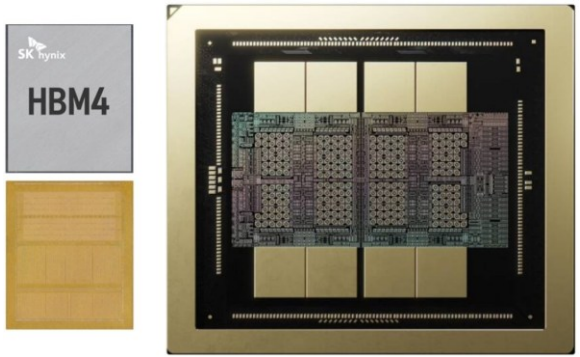
nVidia GPU with GDDR2 (2004)

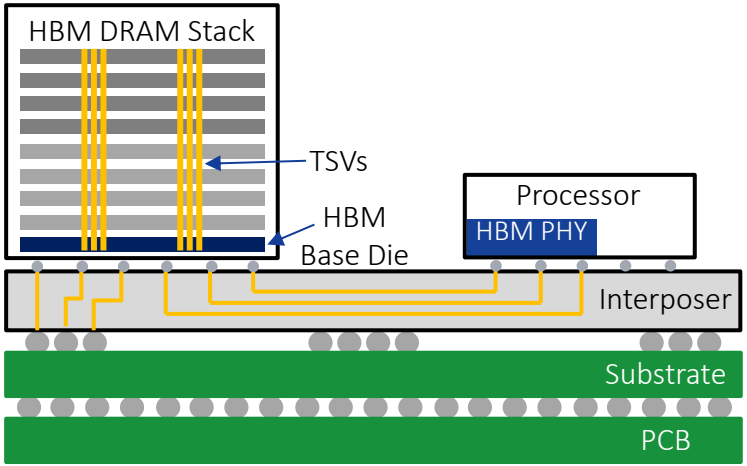


Early GPUs used binned DDR DRAMs, growing volumes justified development GDDR

AI/HPC Memory: HBM

Nvidia Rubin GPU





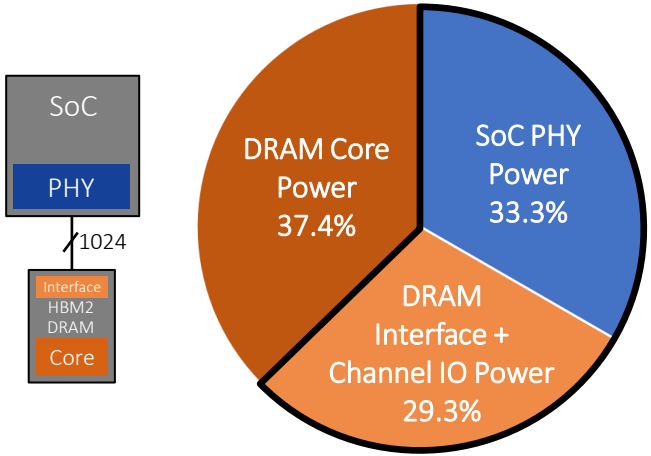
- Many channels, high DQ count
- HBM4e: 32 64b channels = 2048 DQs, up to 16Gbps data rates
- Highest bandwidth of any discrete DRAM (up to 4.1TB/s with HBM4e, high total power but extremely good power efficiency), high fill frequency
- Base die provides connectivity between DRAM die in stack and interposer
- Silicon interposer for fine-pitch connections between processor and DRAM
- More difficult and costlier manufacturing

Power is a Growing Problem in the Data Center

- Tremendous demand for memory bandwidth, high-performance memory standards evolving rapidly (GDDR6->GDDR6X->GDDR7, HBM3/3e->HBM4/4e)
- SoC power budgets: 40%+ used for high-speed memory interfaces, can be hundreds of watts and is getting worse
- HBM very power-efficient, but large portion of power budget spent moving data
- Stacking can help reduce data movement power, provides area interconnect for more IOs
- Stacking introduces new challenges: thermals, scaling capacity and bandwidth, power efficiency

HBM2 Memory System Power

PHY + DRAM Power at 2Gbps, Streaming Reads

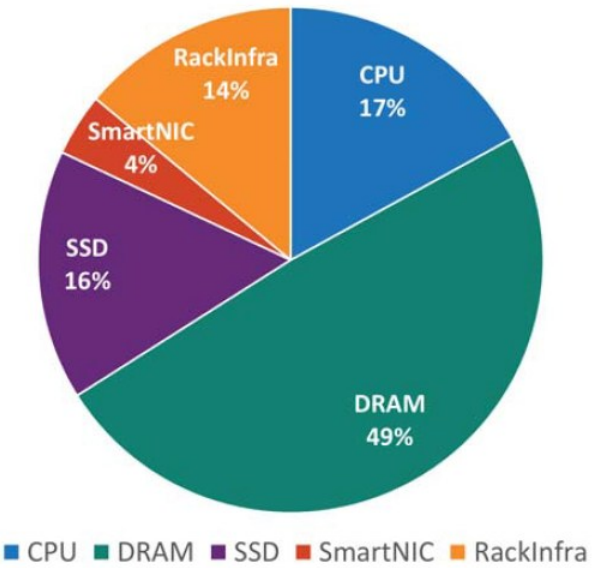


Data Movement: 62.6% of Power

Source: Rambus Inc.

Memory Can Be a Major Fraction of System Cost

Compute Node Component Costs



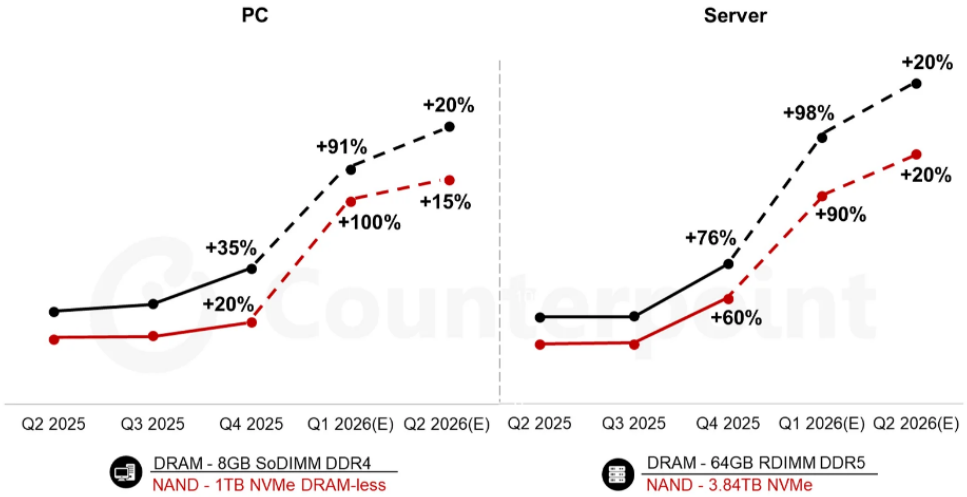
Source: Decadal Plan for Semiconductors, Semiconductor Research Consortium (SRC), January 2021

- Memory is major portion of server cost
- At data center scale, underutilized resources have big TCO impact
- Data center workloads becoming more diverse as new use cases evolve
- Want to compose infrastructure as needed, adapt to workload needs
- CPUs, memory, storage lifecycles different => replace separately to improve TCO

CXL, composability offer the potential to improve TCO

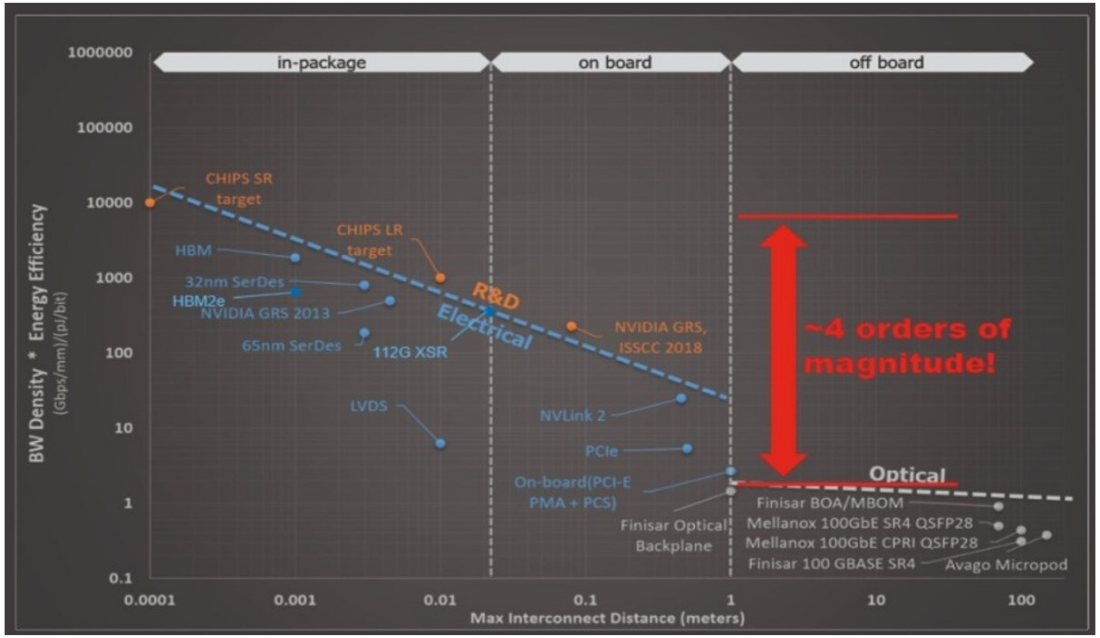
DRAM Prices are Sensity to Supply/Demand Balance

- Today's DRAM shortage causing big imbalance, in Q1 2026 server DIMM prices estimated to rise 300% in mid-2026 vs mid-2025
- Some markets "de-specing" DRAM capacity (but keeping same bandwidth) to reduce cost but keep system performance high



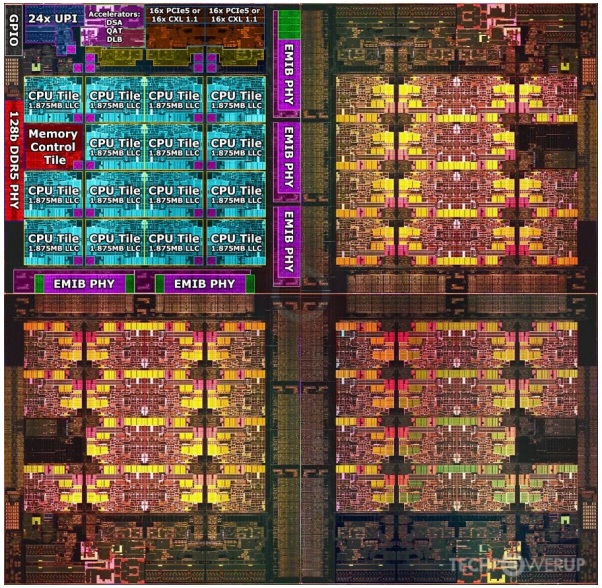
Source: Counterpoint, https://counterpointresearch.com/en/insights/Memory-Prices-Surge-Up-to-90-From-Q4-2025

Distance (Reach) is Power, Shoreline Critical



Source: G. Keeler, DARPA ERI 2019

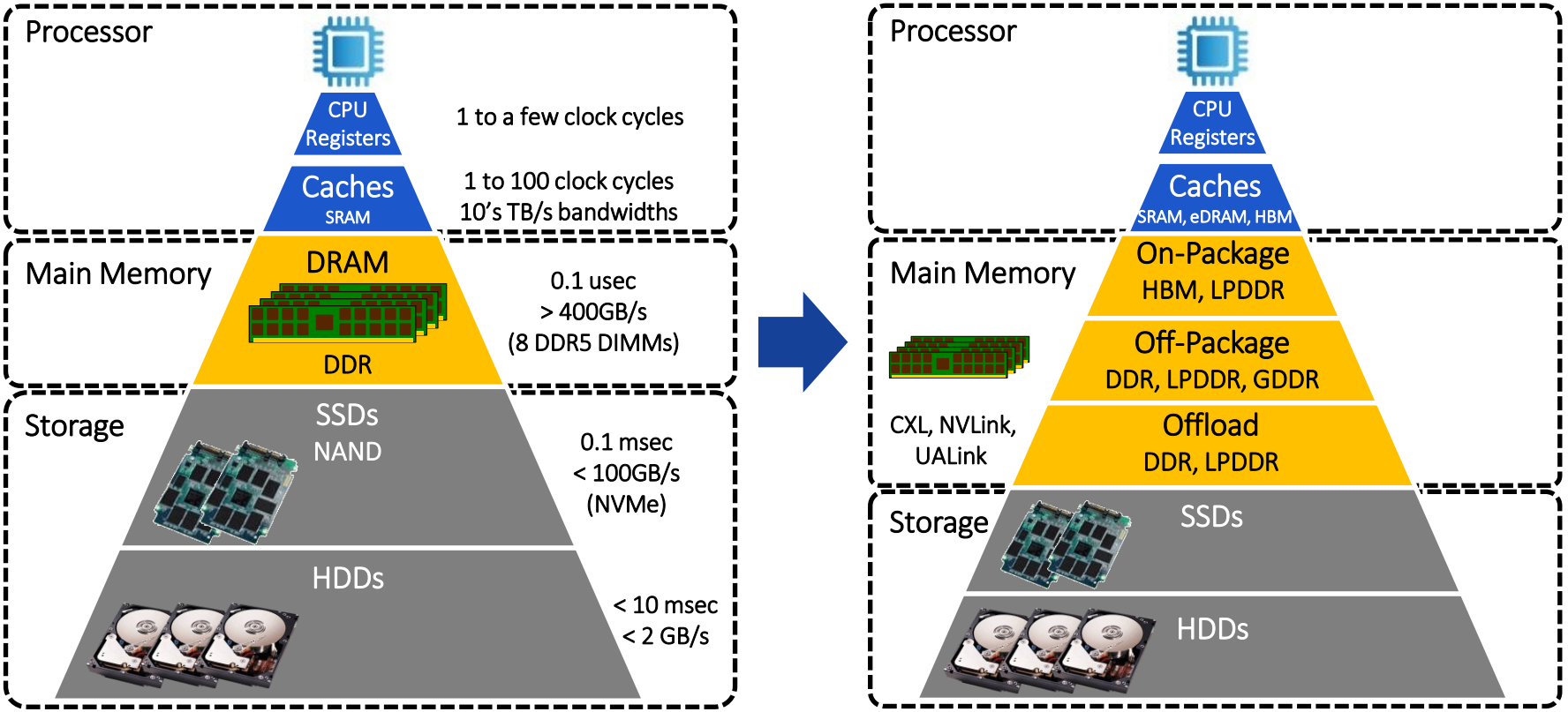
Intel Xeon Max 9470



Source: https://www.techpowerup.com/cpu-specs/xeon-max-9470.c3085

- Primary concerns: power and host shoreline (mm of die area for host PHY)
- Distance is critical determinant of IO power
- Good figure of merit: $Bandwidth\ Density \times Energy\ Efficiency$ [Gbps/mm]/[pJ/bit]

The Evolving Compute Memory Hierarchy



The memory hierarchy is becoming more sophisticated as application needs evolve

SAMSUNG

DRAM Architecture:
Cell, Array, Data Path and Interface

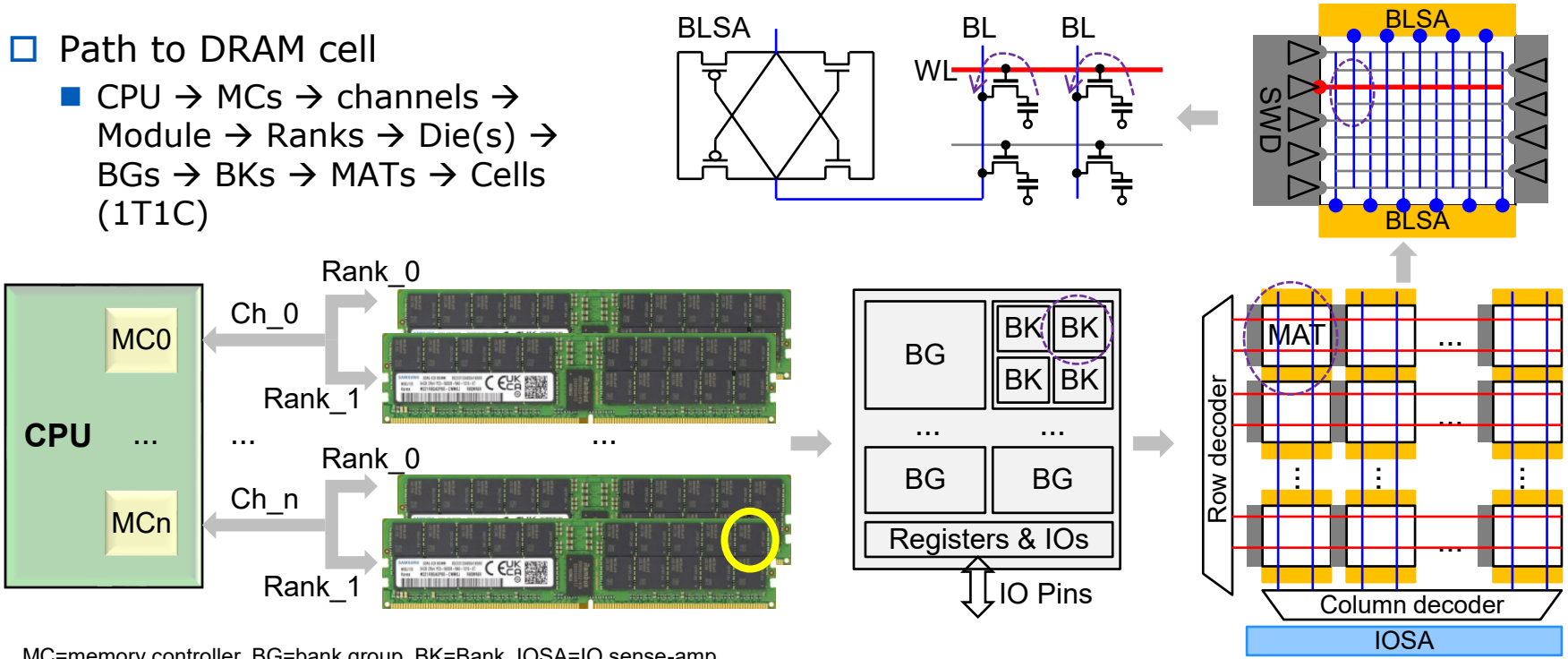
Taeksang Song, Ph.D.
Corporate VP of SoC & DRAM Solution Engineering,
Samsung Electronics

Outline

- Introduction
- DRAM Core & Peri Architecture
  - Cell Operation & Timing Parameters
  - Sensing Margin – Leakage & Offset
- DRAM Cell Scaling
  - Vertical Stacked (VS) DRAM & Cell-over-Peri
  - RowHammer and RowPress in VS DRAM
- DRAM Interface
- Summary

DRAM Architecture

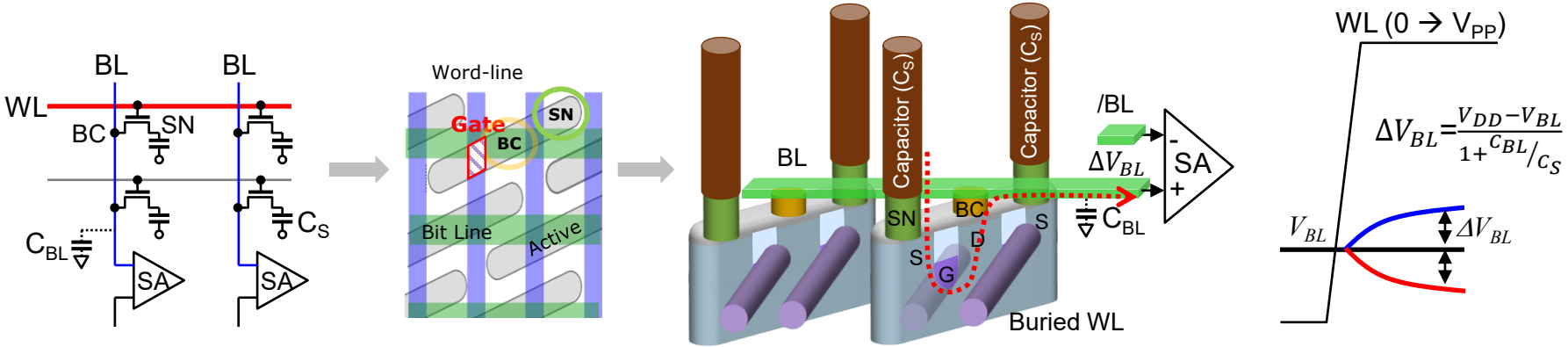
- Path to DRAM cell
  - CPU → MCs → channels → Module → Ranks → Die(s) → BGs → BKs → MATs → Cells (1T1C)



MC=memory controller, BG=bank group, BK=Bank, IOSA=IO sense-amp
BLSA=Bit line sense amp, WL = word line, BL = Bit line, SWD = Sub-WL Driver

Cell & Core Operation

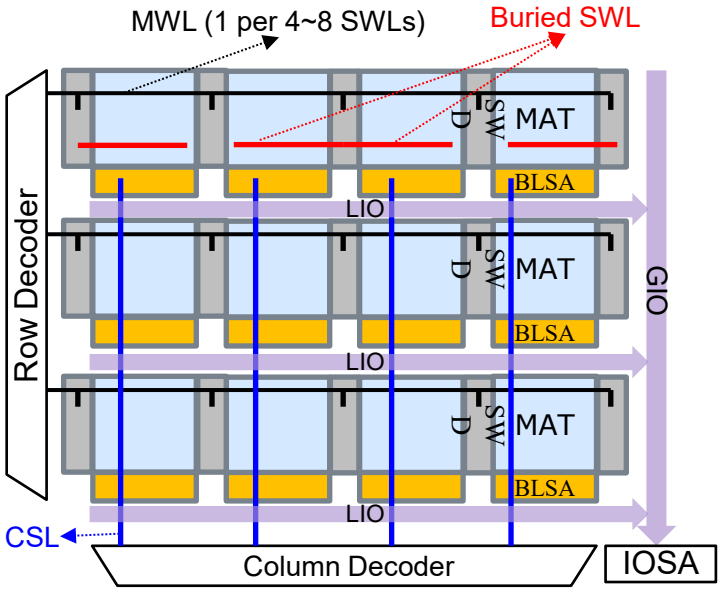
- MAT
  - Unit block which has cell-array (e.g., >1K WL * >1K BL), sub-WL driver & BLSA
  - Determines cell efficiency (cell area / chip area, ~50%) & sensing margin
    - Large array mitigates SWL driver & BLSA area overhead → high capacity, low cost
    - But, reduce sensing margin due to larger parasitic capacitance (C_{BL})



BC=BL contact, SN=Storage Node, C_S=Storage capacitance, C_{BL}=BL parasitic capacitance

Bank Configuration

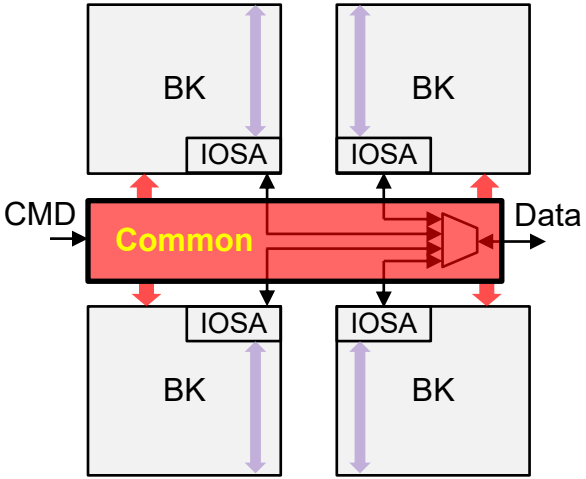
- Bank
  - Array of MATs + control circuits
  - Can activate only one row/page at a time, but can activate row/page independently and separately from other banks
- Configuration
  - Row decoder
    - Activate only one page (=one row of cells)
    - 1KB~2KB cells are activated and sensed by BLSA
  - Column decoder
    - CSLs connect the selected BLSA to LIO (e.g., only 128 out of 1KB → 1:32 mux selection)
  - LSA & IOSA
    - Latches to drive long interconnect lines



CSL=column selection line, MWL=Main WL, LIO=Local IO
GIO=Global IO, LSA=Local Sense-Amp, IOSA=IO Sense-Amp

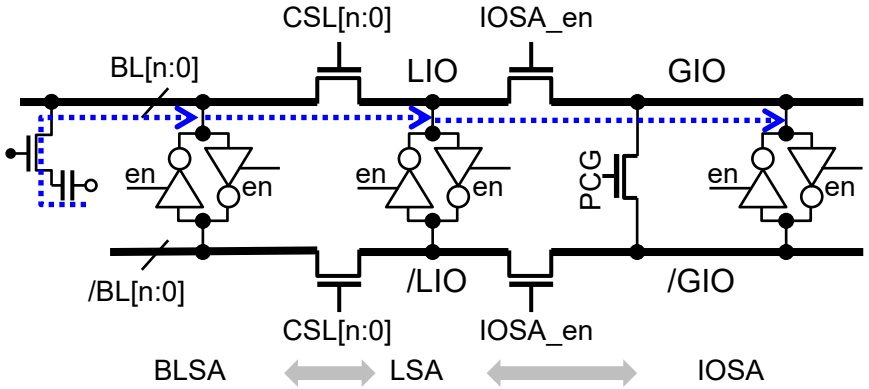
Bank Group Configuration

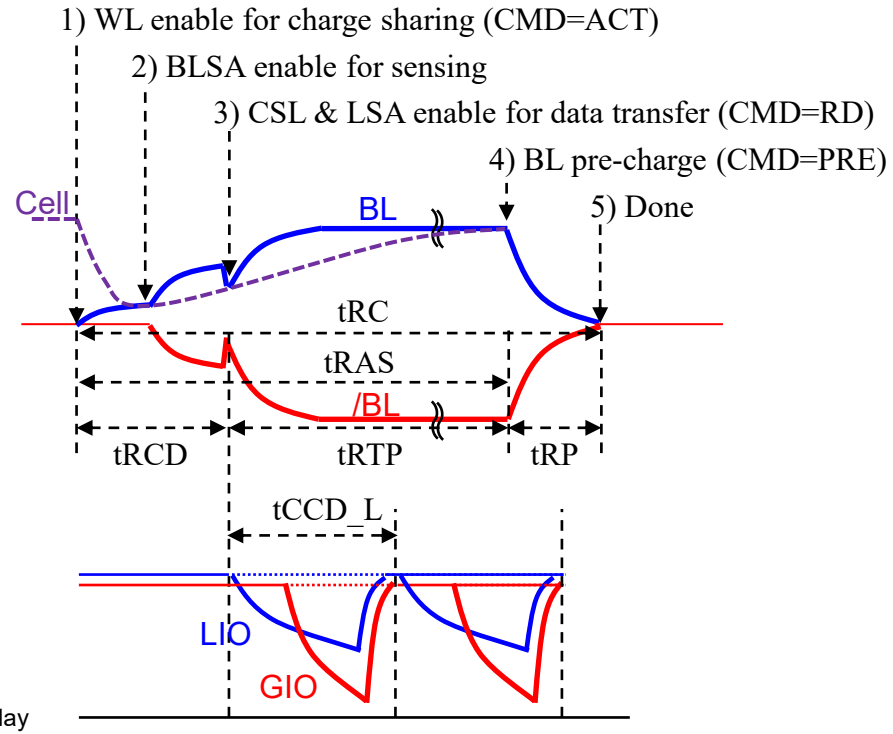
- Bank Group
  - A unit that operates independently
  - Share common blocks
    - Timing control
    - Data & command lines
- Restrictions
  - tCCD_L
    - Because of the shared common blocks, while read or writing data from one bank (tCCD_L), other banks within the same BG can't be read or written



Core Timing Diagram

- Row access
  - Same Bank: one per tRC ← shared BLSA
- Column access within same BG
  - tCCD_L ← shared data-path

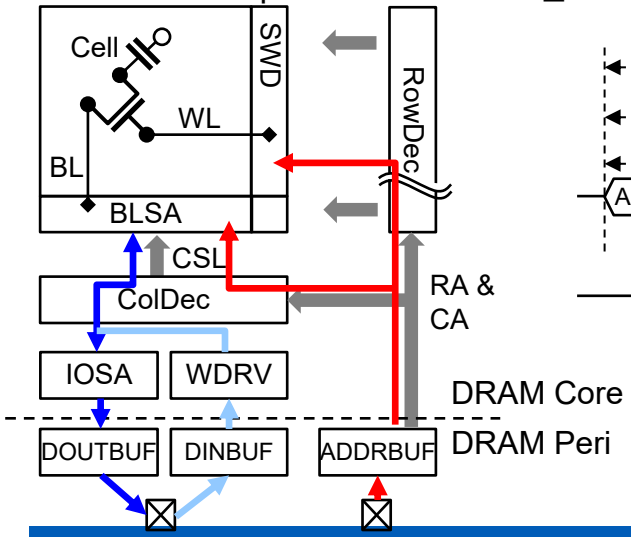


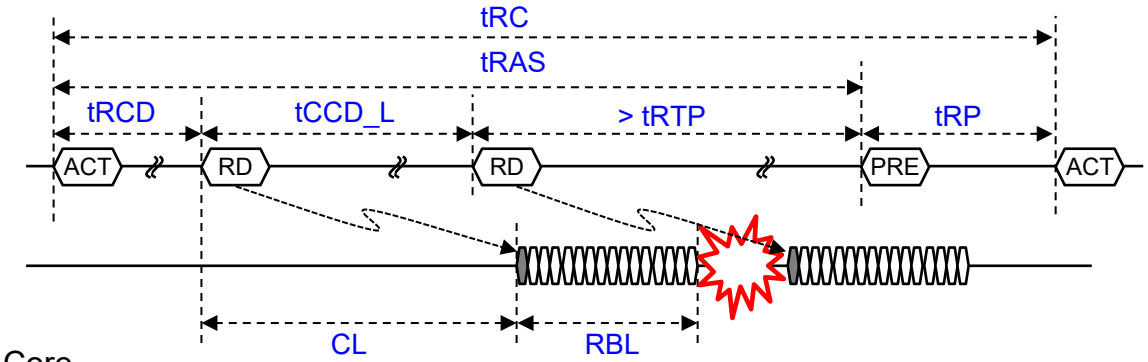


tRC= row cycle time, tRAS=Row Active to Precharge Delay, tRCD=Row-to-Column Delay
tRTP=Read to Precharge Delay, tRP=Row Precharge, tCCD=Column-to-Colunm Delay

DRAM Peripheral Timing Diagram

- DRAM peripheral
  - Serializes READ data from the core, and sends to the Host → Read Burst
    - No DRAM core timing scaling → Burst Length (BL) keeps increasing due to higher clock speed
- DQ bus utilization from a bank
  - One read data (RBL) from one row activation (tRC) → <10% utilization
  - Multiple reads at tCCD_L interval from the activated row → <50% utilization @ DDR5 8000

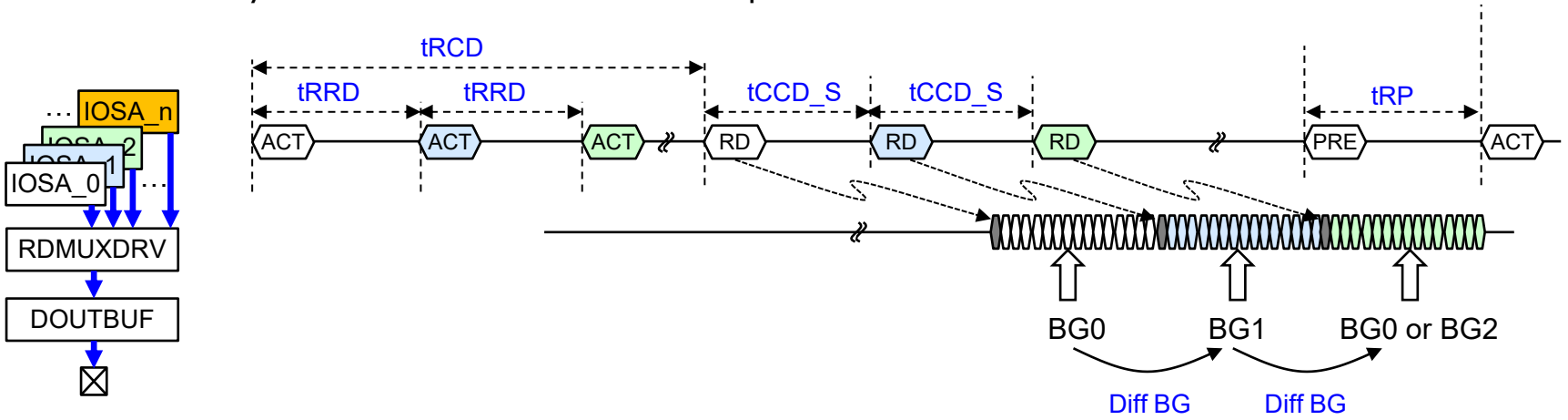




WDRV=Write driver, CL=CAS Latency, RBL=Read Burst Length

Bank-Level Parallelism (BLP)

- Multiple banks can be activated and handle different memory requests at the same time
- Timing constraints due to shared resource
  - Same BG: allows data access at tCCD_L interval ← shared lines before RD_MUX
  - Same chip: allows data access at tCCD_S=RBL interval ← shared DQ & external bus
- Can fully utilize when DRAM has multiple BGs and Banks

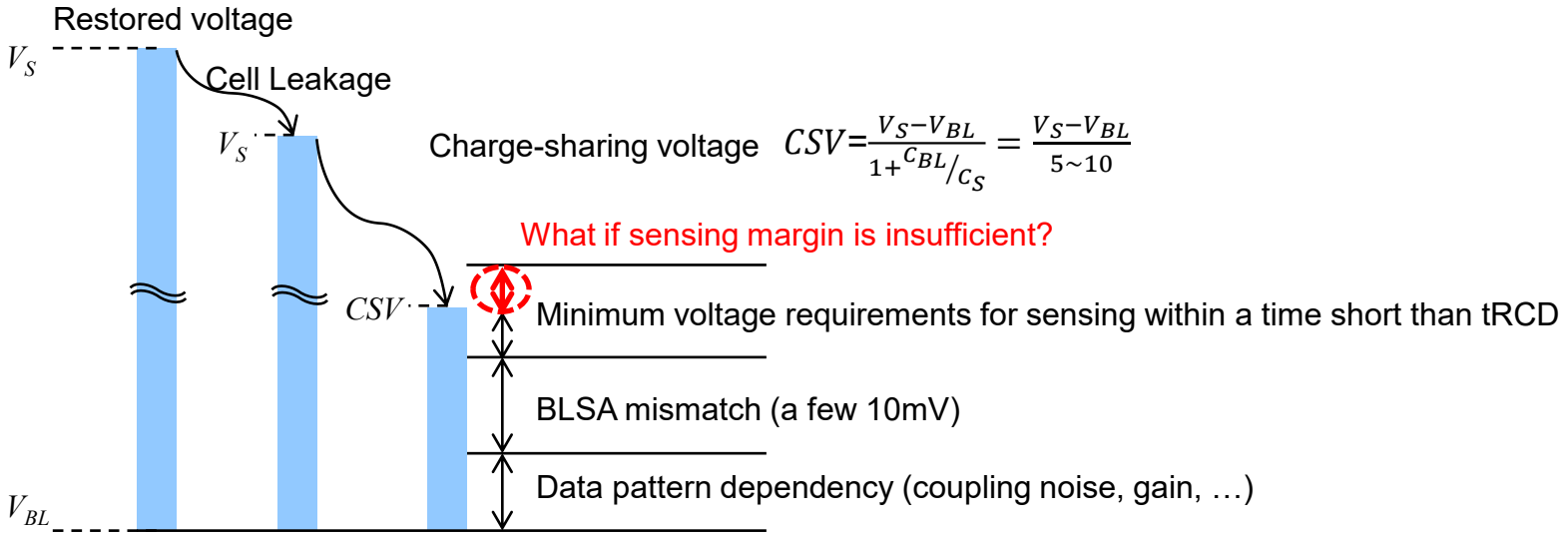


Core & Data Path Timing Parameters

Parameter		Description	Note
tRCD	Row to Column Delay	Time between activating a row and issuing a RD/WR command	Time for offset cancellation → charge sharing → developing BL & /BL
tRAS	Row Active	Time a row must stay open after activation	BLSA fully develops BLs to restore data back to cell
tRP	Row Precharge	Time to close a row	Make BL & /BL initial voltage level (V_{BL})
tRC	Row Cycle	Minimum time between row activations in the same bank	tRC=tRAS+tRP
tRTP	Read to Precharge Delay	Time between RD command to Precharge command	Guarantee data-path operation Precharge can be issues only when both tRTP and tRAS are satisfied
tCCD	Column-to-Column Delay	Time between two column access commands	tCCD_L: same bank group tCCD_S: different bank group
tRRD	Row-to-Row Delay	Minimum time between activating one row in a bank and another row in a different bank	tRRD_L: same bank group tRRD_S: different bank group
CL	CAS Latency	# of clock cycles between RD command and RD-data	

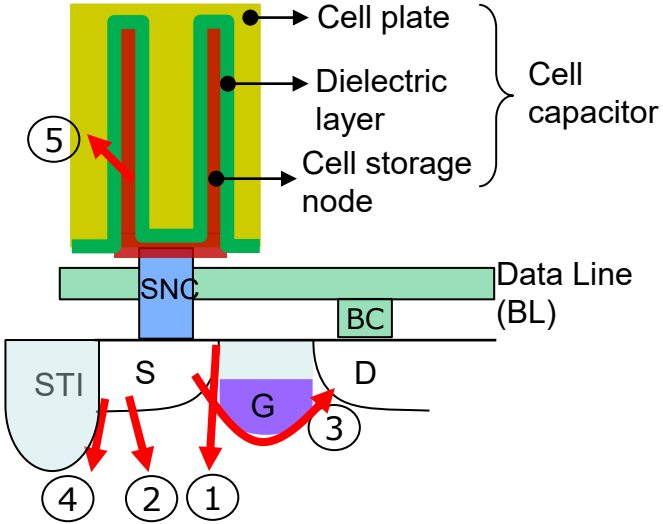
Sensing Margin - Overview

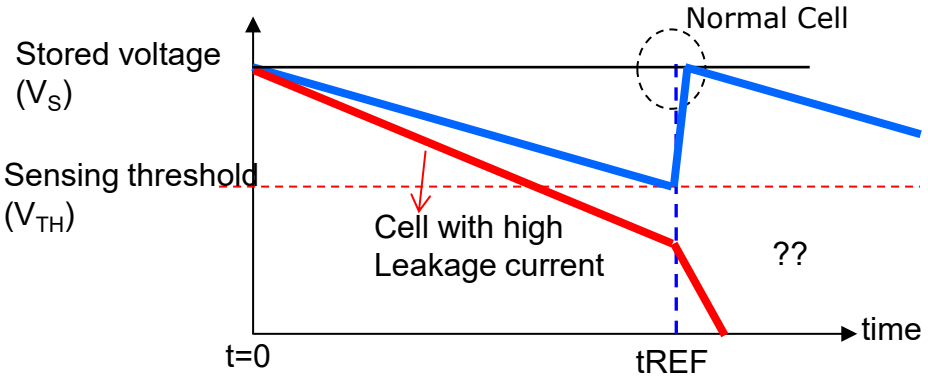
- What factors determine sensing margin
  - Charge Sharing voltage ← cell leakage, C_{BL}/C_S ratio, stored charge (V_{DD} & V_{BL})
  - Sensing margin ← BLSA mismatch, noise



Sensing Margin – DRAM Refresh Operation

- Charge leakage current
  - 1) GIDL, 2) Junction, 3) off-state, 4) STI, 5) dielectric leakage
- Periodic DRAM refresh
  - For data retention, DRAM must be refreshed at regular interval due to leakage current

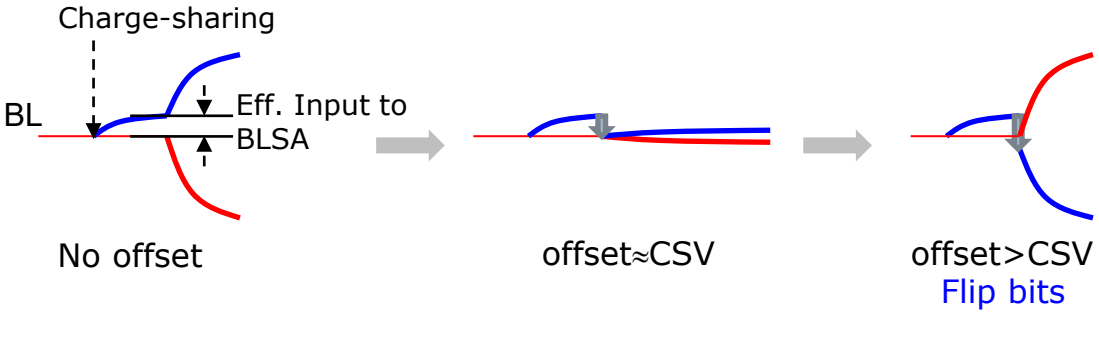


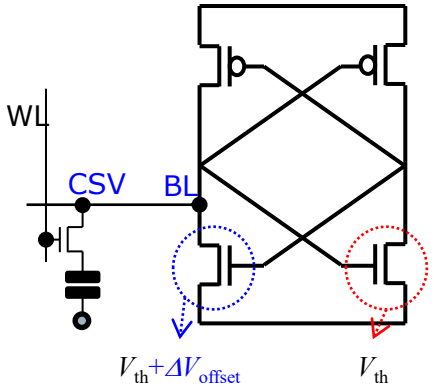


$$t_{REF} = \frac{C_S}{I_{Leak\_Total}} \cdot \left\{ V_S - \left(1 + \frac{C_{BL}}{C_S} \right) \cdot V_{TH} \right\}$$

Sensing Margin – BLSA Offset

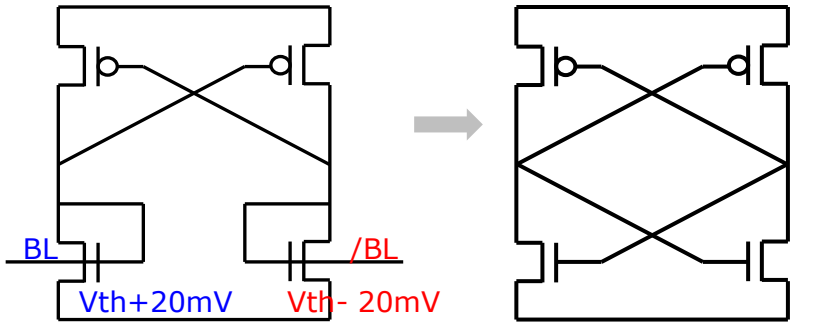
- Sources of BLSA mismatch
  - Geometric (W/L), Random dopant fluctuation, Cox
  - As technology scales down, mismatch increases seriously
- BLSA offset
  - A few 10mV
  - Makes BLSA develop time long or flips data



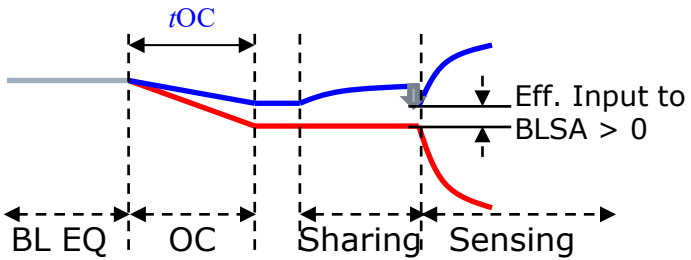


Sensing Margin – BLSA Offset Cancellation

- Additional step for offset cancellation (OC)
  - BL & /BL equalization → offset cancellation → charge-sharing → develop
- tOC is critical
  - Too short → no cancellation.
  - Too long → overcompensation

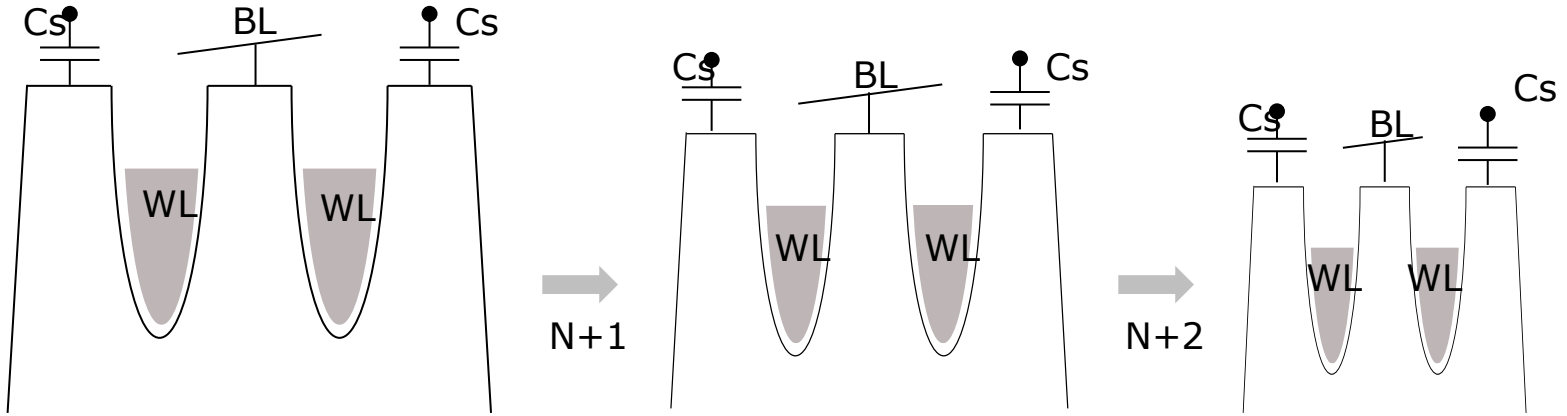


<Offset Cancel> <Sensing>



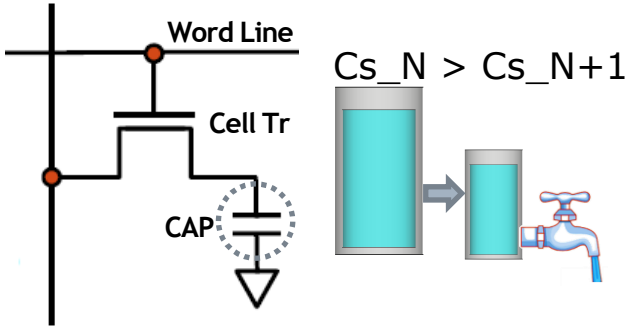
Sensing Margin – DRAM Scaling

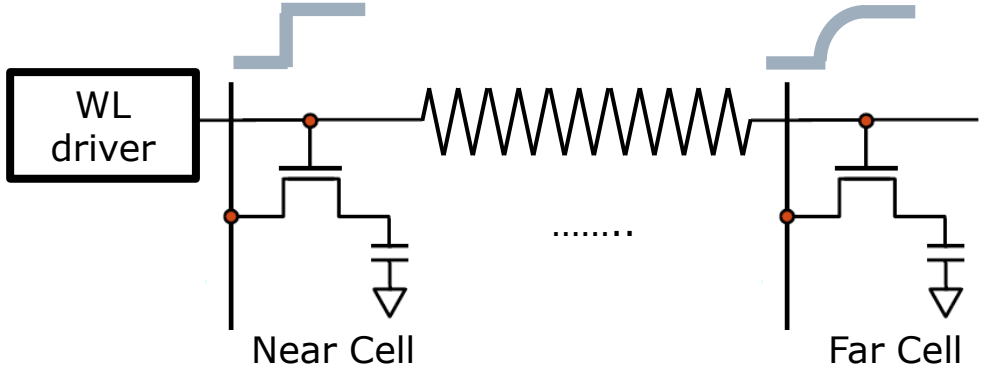
- Liner Scaling of DRAM Cell Transistor
  - Bit density has been increased by lateral shrink while keeping the shape of transistor.
  - Consequently, BL and SN got closer, linewidth of BL, WL decreased, capacitor size reduced.



Sensing Margin – Scaling Challenges

- The capacitance of stacked capacitor is approaching to the retention time limit
- WL resistance is approaching RC timing limit to guarantee read/write timing.
- As cell to cell distance gets closer, repeated access of one row can cause failure in adjacent row



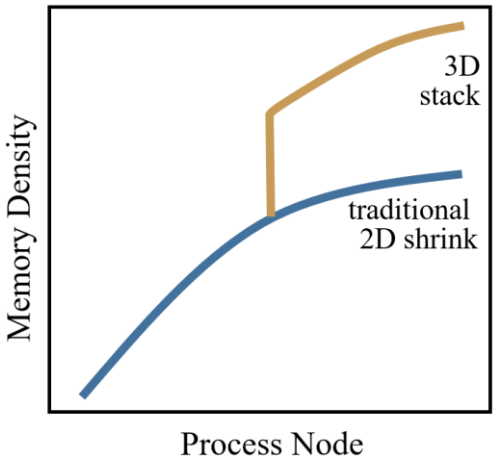


$$R_{WL} = \rho_{WL} \times L / H \cdot W$$

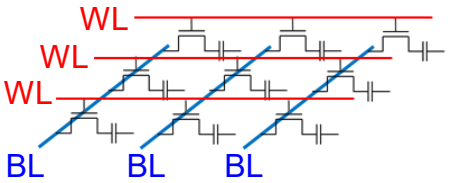
Scaling – Vertically Stacked (VS) DRAM

SAMSUNG

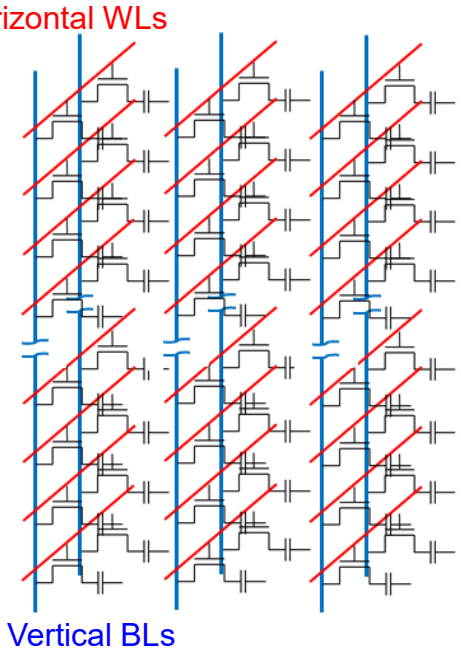
- Vertically stacked 3D DRAM can relax the dimension requirement via z-dimension



2D →
3D Stacking



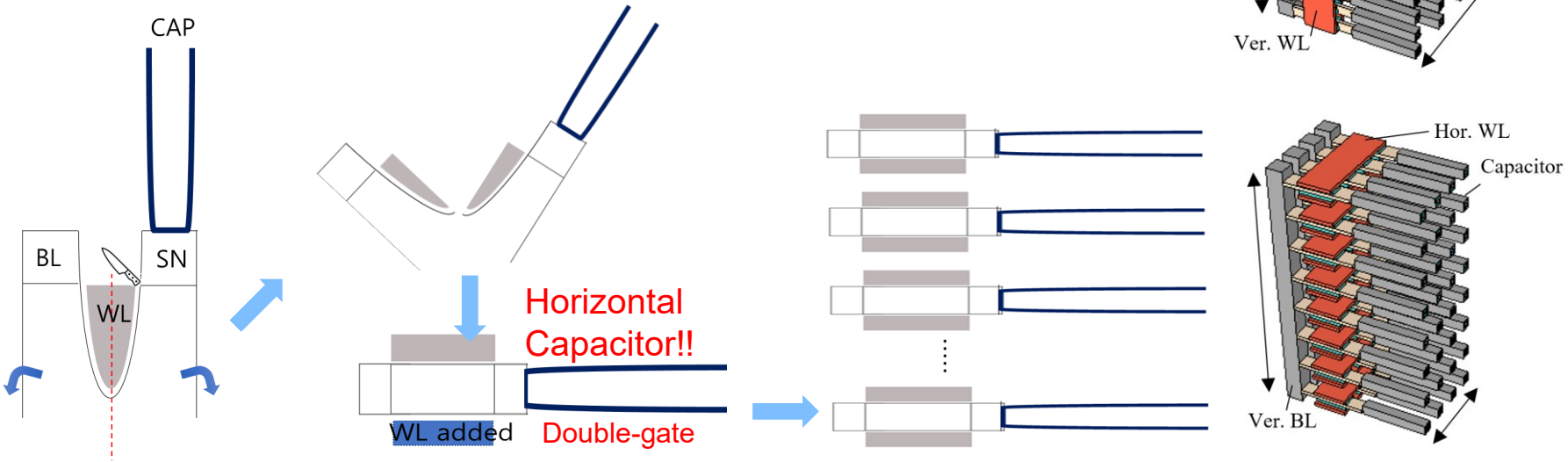
2D planar array



3D cubic block

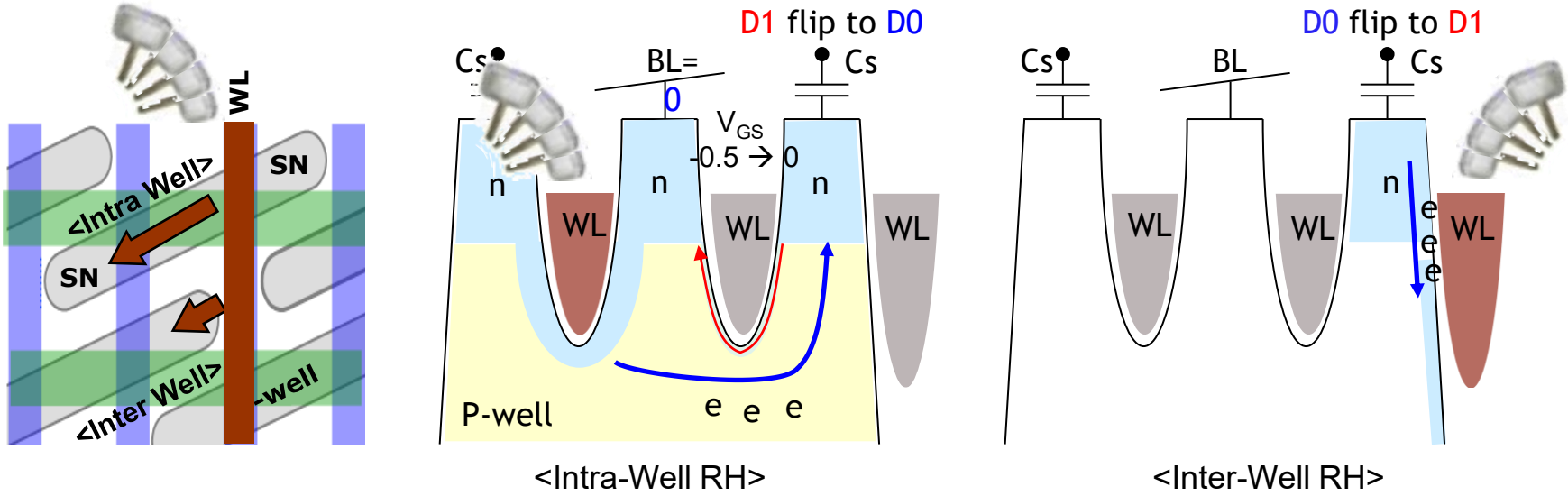
Scaling – VS DRAM Concepts

- Structure
  - Horizontal capacitor
  - (horizontal BL, vertical WL) or (vertical BL, horizontal WL)



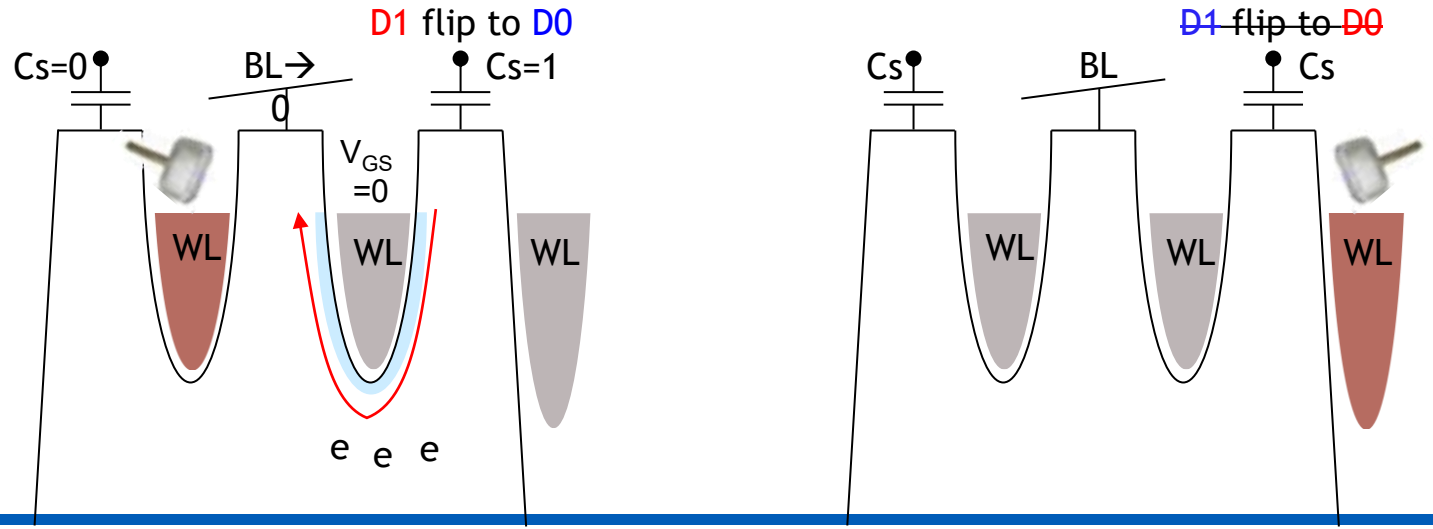
Row Hammer – Planar DRAM

- Intra-well: Electrons flow into victim cell & leakage causing D1→D0 failure
- Inter-well: Electrons flow out of victim cell, causing D0→D1 flip



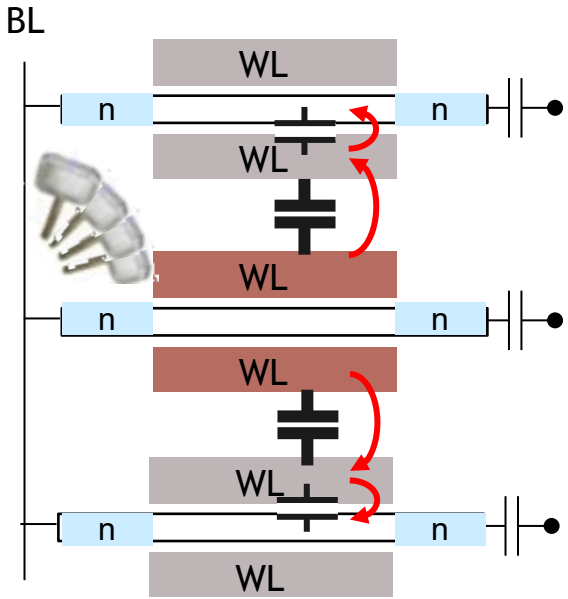
Row Press – Planar DRAM

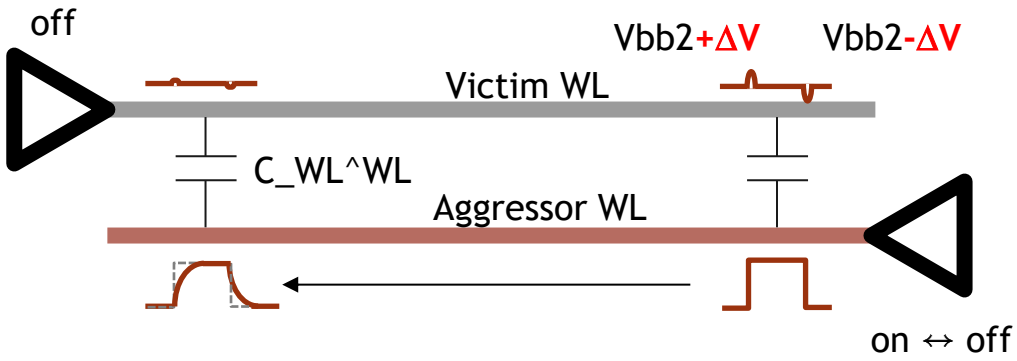
- Long-time activation of one row can cause bit-flip
- WL voltage of aggressor reduces Vt of victim cell, resulting in D1 failure.



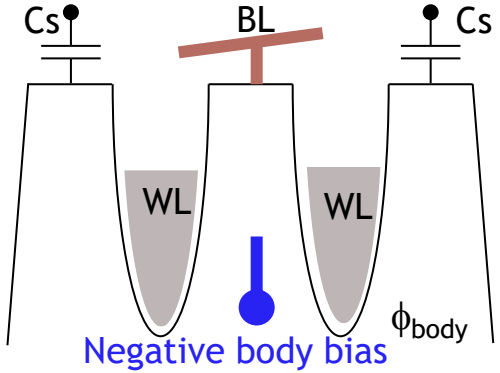
Row-Hammer & Row-Press in VS DRAM

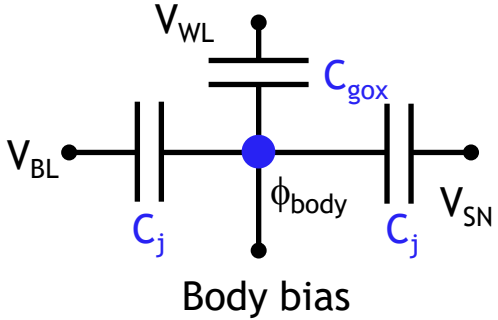
- WL to WL capacitive coupling causes bit flip as well.
- Silicon channel is fully surrounded by WL metal, no Row-Press occurs in VS



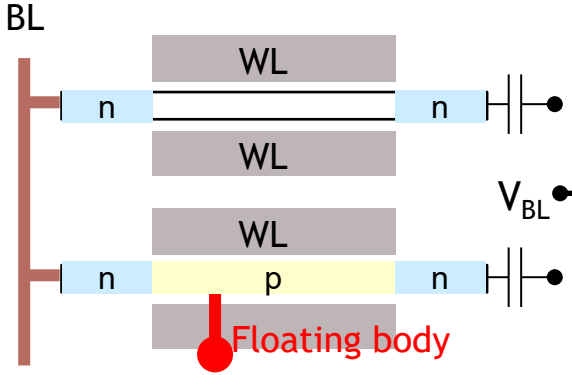


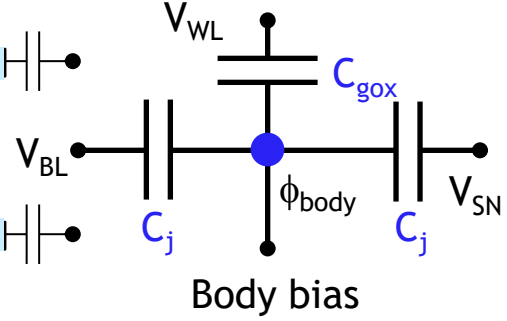
Column (BL) Hammer or Column Press





- ☑ GIDL generates holes in the body.
  - BL=1, WL=off-state
- ☑ Channel potential is fixed by body bias.
- ☑ Hole is swept out by negative body.

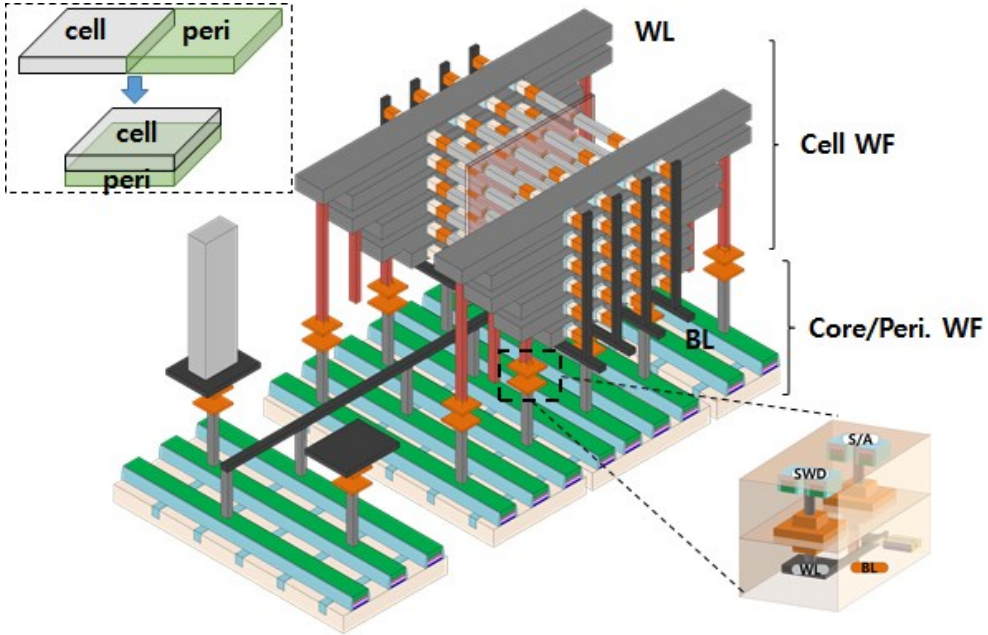




- ☑ GIDL generates holes in the body.
- ☑ Channel potential is floating.
- ☑ Hole due to GIDL raise body potential → trigger NPN bipolar action.

Scaling – Cell Over Peri Architecture

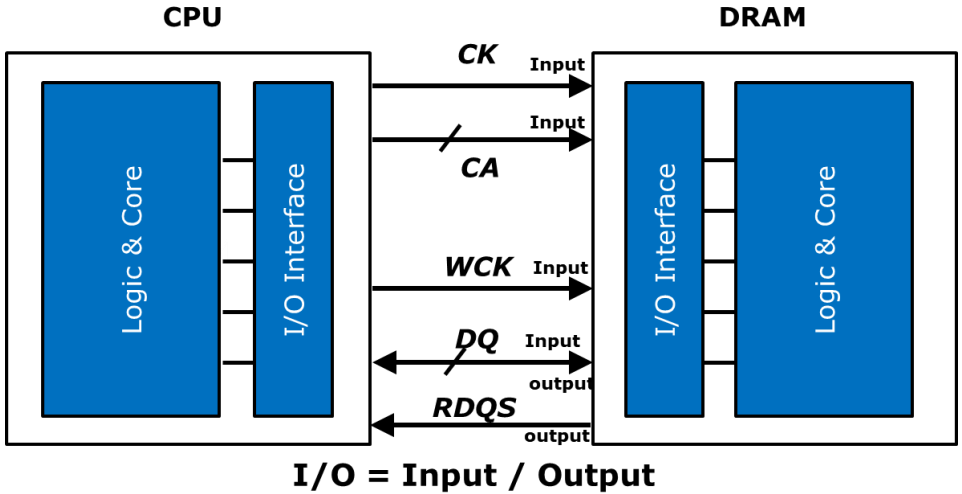
- Cell WF and Core-Per WF are separately fabricated and WoW bonded



Outline

- Introduction – DRAM
- DRAM Core & Peri Architecture
- DRAM Cell Scaling
- DRAM Interface
  - DRAM IO
  - Clocking
  - IO Training
  - Emerging DRAM IOs
- Summary

What's Memory I/O?



❖ **Pin Description**

✓ CK : Clock for command & address

✓ CA : Command and Address

✓ WCK : Clock for data

✓ DQ : Data transfer

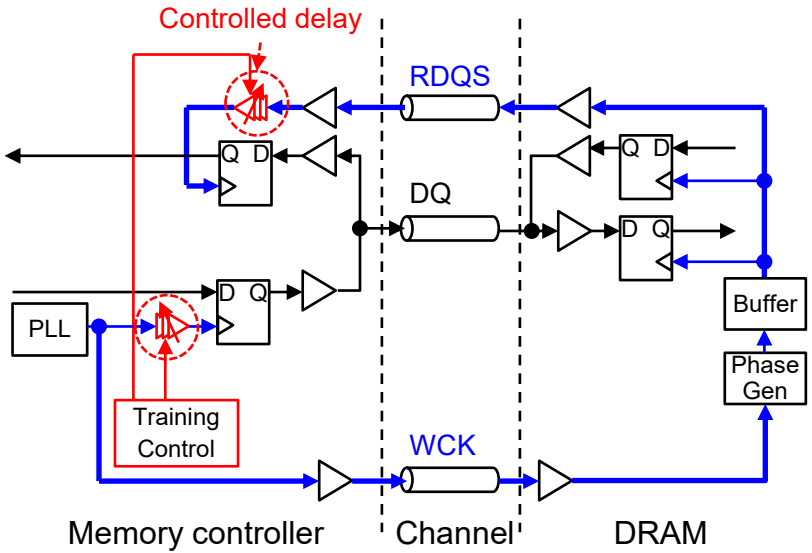
✓ RDQS : Read strobe clock

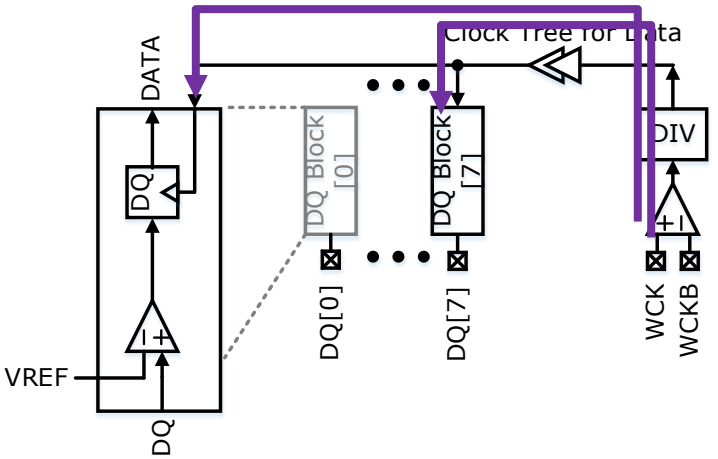
** CA and DQ show parallel IO configurations.

- Definition : Input/Output Pins and Relevant Circuitry (= I/O Interface)
  - Bi-directional DQ pins : Data in for Write, Data out for Read
  - Differential clocks (CK, WCK, RDQS) vs. Single-ended data (DQ and CA)
  - Multiple DQs (or CAs) are synchronized by WCK (or CK).

DRAM Clocking & Strobe

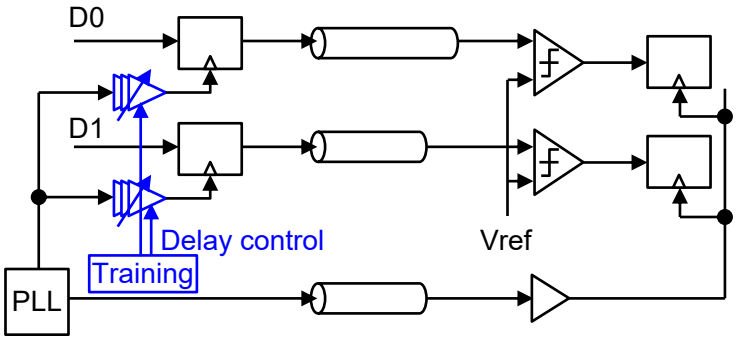
- Source synchronous un-matched clocking
  - Because of repeaters only for WCK, different timing relationship between DQs and WCK at the receiver input → requires training & complex timing control, but less jitter by DRAM

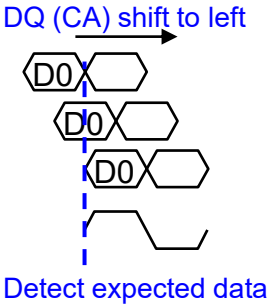


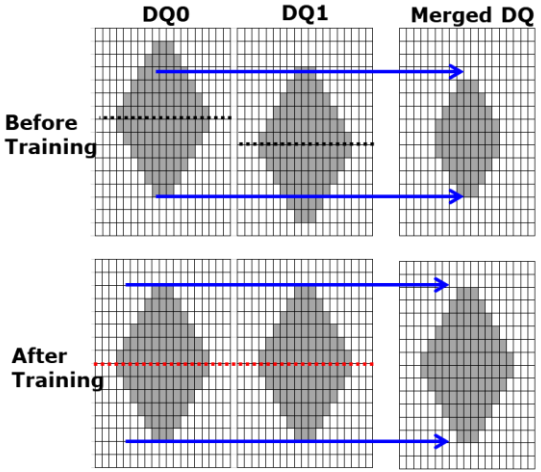


Training Method

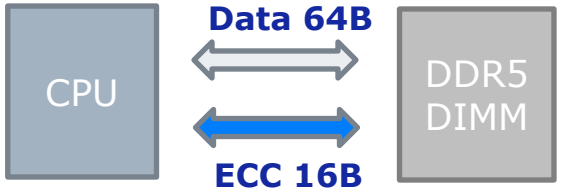
- Match the delay of DQ (CA) to that of DQS (CLK) at receiver input → Add the delay to DQ (CA) timing in order to compensate on-chip & channel mismatch
- Allows SoC to search the eye voltage and timing center per each DQ & CA

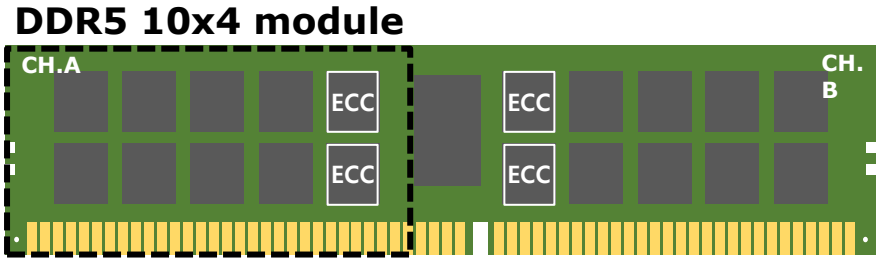






Link & Data Protection





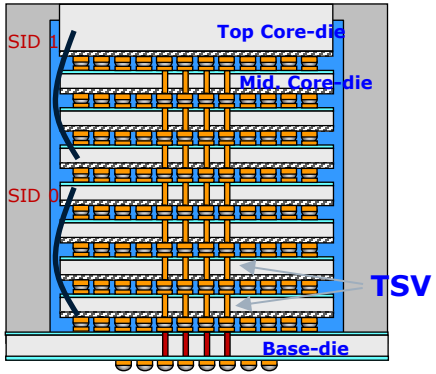
Access Granularity : Data 64B + ECC 16B

- System level ECC
  - 10x4 DDR5 RDIMM has dedicated 2 DRAM chips to store system ECC.
  - The system ECC protects system from both DRAM component errors (Cell & Core) and IO interface errors simultaneously.
  - The 16B ECC parity bits are able to cover any failure in a single memory chip

Emerging DRAM IO – HBM

SAMSUNG

HBM (8H) Architecture

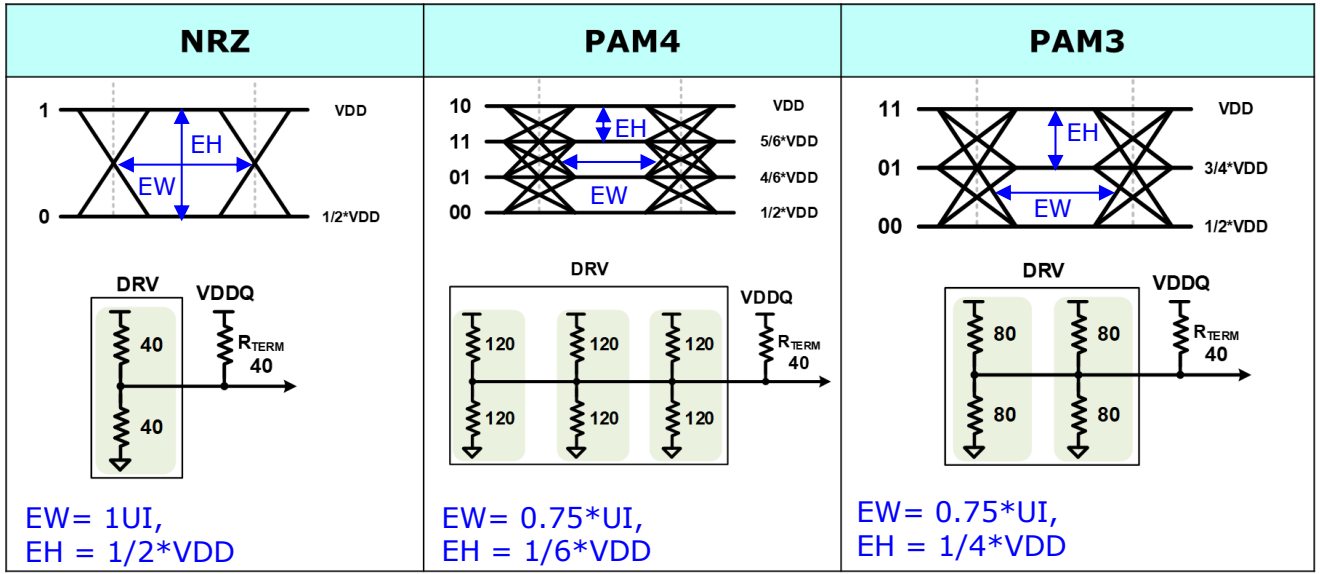


- To Maximize memory bandwidth, design constraints in more than a thousand of IO interface :
  - ODT is not allowed due to too much static power consumption
    - C_{IO} must be minimized for SI.
  - Base die without DRAM cell supports the requirements as buffers.
    - Reduced C_{IO}
    - Better Tr. Performance
    - Simpler IO design
    - IO shared between core dies.
  - Due to above constraints, IO width is increased rather than bit rate even in HBM4/4e.

Mode		HBM3e	HBM4/4e
Pin Speed		8.0Gbps~10Gbps	**8.0Gbps~16Gbps**
# of DQ		1024	**2048**
Max. BW		1.0TB	2.0TB
Power	VDDQ	1.1V	0.7~0.9V
	VDDQL	0.4V	0.4V
	VDDC	1.1V	1.0~1.05V
	VPPE	1.8V	1.8V

Emerging DRAM IO - GDDR7 PAM3

- GDDR7 is the first standard DRAM adopting PAM signaling.



* Assumption : TX impedance = GPU termination = 40 Ohm

Summary

- To meet the industry demands, DRAM will continue to evolve and scale in terms of both speed and capacity

- DRAM Architecture
  - Share resources to maximize cell efficiency → mitigate timing penalty caused by shared resources by using multiple banks/BGs
  - 3D-stacked DRAM and noise/offset cancellation for sensing margin

- DRAM IO Techniques
  - Source synchronous and un-matched clock for high-speed signaling
  - Training and Link Protection to extend IO speed
  - Two tracks to increase data bandwidth of DRAM:
    - Increase the number of pins like HBM, or the per-pin speed with multi-level signaling like PAM3 in GDDR7

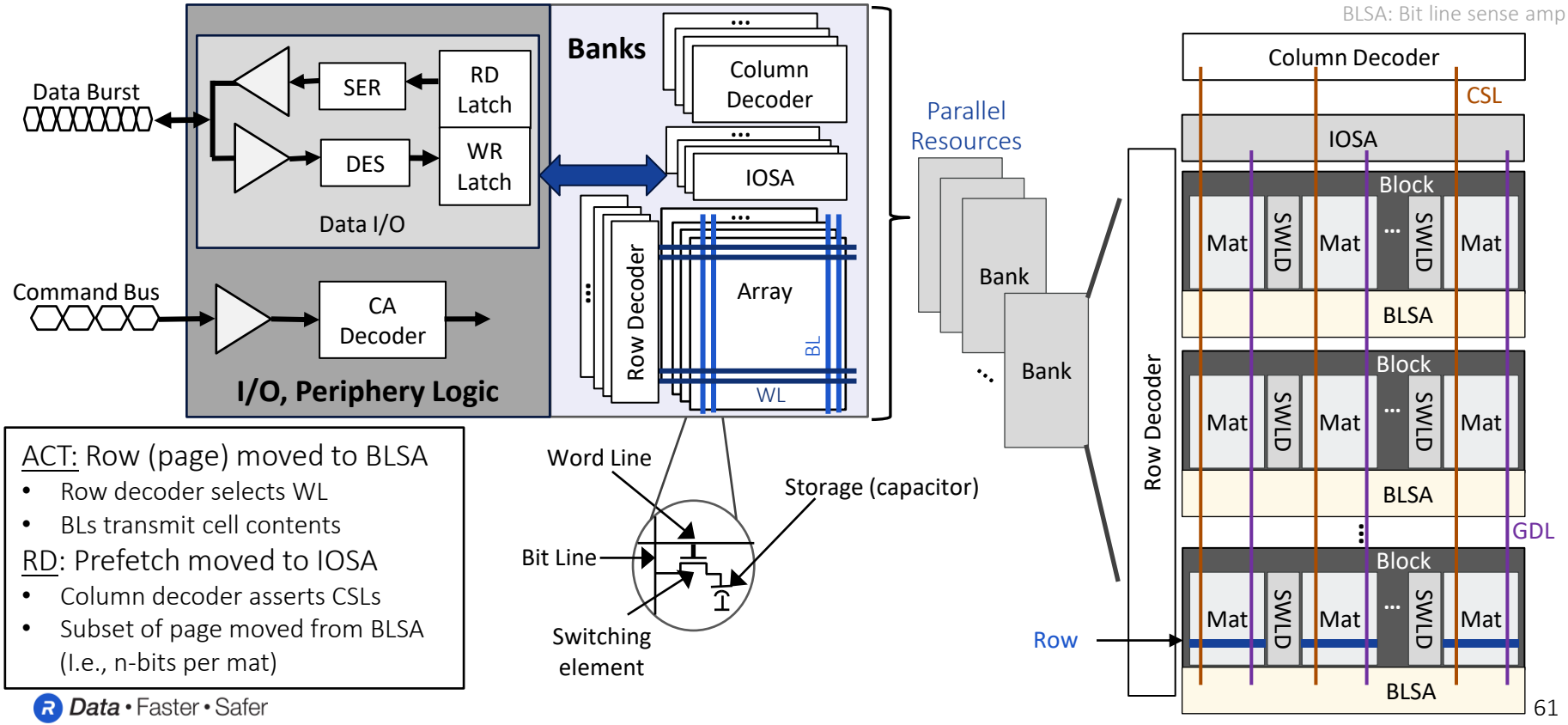
Market Demands and Scaling Needs: Why one size doesn't fit all

Wendy Elsasser
Technical Director
Rambus Inc.

Rambus

At the Core, All DRAMs Work the Same Way

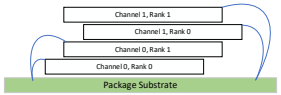
WL: Word line
SWLD: Sub-wordline driver
CSL: Column select line
BL: Bit line
GDL: Global data line
IOSA: IO sense amp
BLSA: Bit line sense amp



ACT: Row (page) moved to BLSA
- Row decoder selects WL
- BLs transmit cell contents

RD: Prefetch moved to IOSA
- Column decoder asserts CSLs
- Subset of page moved from BLSA (I.e., n-bits per mat)

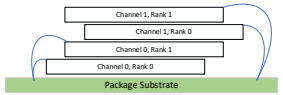
Mobile



LPDDR Package

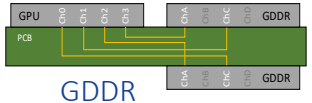
- Low cost
- Low power
- Small form factor with short reach
- Soldered down or PoP
- High-capacity package
- Multi-die, multi-channel package (1-2 dies/ CH)

Auto



LPDDR Package

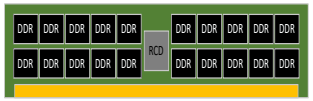
OR



GDDR

- Safety critical (ASIL req.)
- High error detection
- Predictive timing
- Device longevity
- Increasing BW requirements

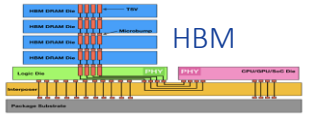
Traditional Cloud



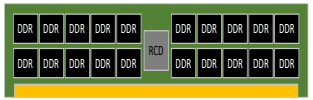
DDR DIMM

- High aggregate capacity with multi-die channel
- Commodity devices
- High RAS
- Field replaceable (serviceability)

AI Inference (Data Center)

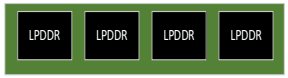


+



DDR DIMM

OR



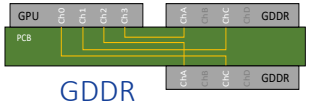
LPDDR SOCAMM

Local Memory

- Extreme BW
- Energy efficient
- High availability

Offload Memory for additional capacity

Graphics & Gaming



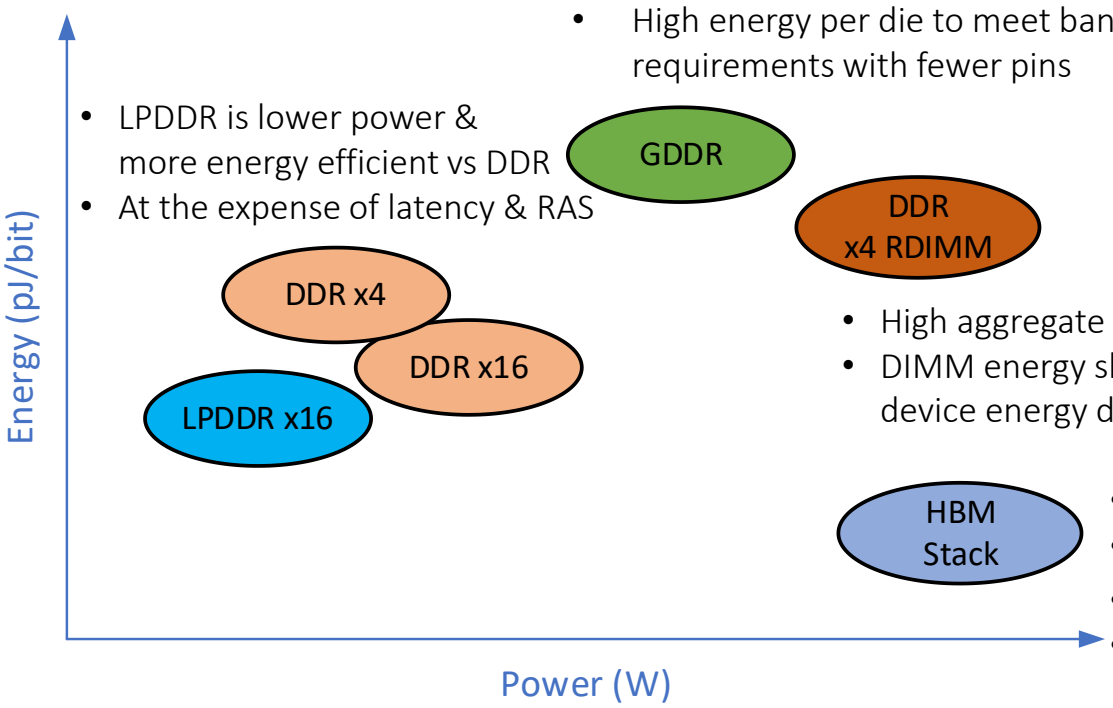
GDDR

- High bandwidth
- Moderate cost (lower than HBM)
- Clamshell mode to ↑ capacity

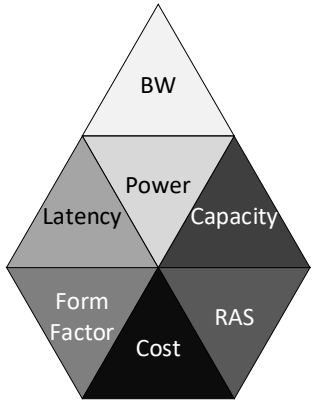
*Sample market representation
*Markets are constantly evolving
*This only represents a subset of markets

Markets Drive Tradeoff Decisions

Energy and Power Example



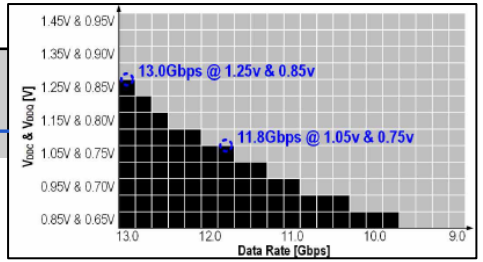
- High energy per die to meet bandwidth requirements with fewer pins
- LPDDR is lower power & more energy efficient vs DDR
- At the expense of latency & RAS
- High aggregate power
- DIMM energy slightly higher than device energy due to multi-rank + RCD
- Low energy per bit ($^{pJ}/_{bit}$)
- Low power per bandwidth ($^{Watt}/_{GB/s}$)
- High aggregate power (W)
- **Low energy doesn't always equal low power**



Multiple tradeoffs to consider

Low Device Power Doesn't Come for Free

Technique[1]	Reduces	Implication
Lower voltage (including DVFS)	Dynamic power	↓ Voltage & noise margin Impacts max data rate
New VDD rails I.e., VDD2C & VDD2D in LP6	Core & periphery power (better voltage domain opt)	↑ Design complexity & PMIC overhead
Short reach, lower complexity I/O architecture	I/O power & clock interface power	↓ Reach, topology flexibility & capacity scaling headroom
No on-device DLL	Clock generation & tracking power	↑ Timing uncertainty & PVT variation sensitivity
High-V_t transistors	Leakage, standby power	↑ Latency
Architectural changes, e.g., • Smaller mats/subarrays	Activation, sensing & precharge power	↑ Area & Complexity (limited use)
No column over fetch	Wasted internal activity	↓ Array re-use, on-die ECC



HBM4[3]
Voltage vs Data Rate

LP5 vs DDR5[2]:
- ~40% higher tRL
- ~25% higher tRC

DDR x4 over fetch of 2X

[1] Not an exhaustive list; [2] Multiple factors result in higher latency; high-Vt is one component,
[3] "15.6 A 36GB 3.3TB/s HBM4 DRAM with Per-Channel TSV RDQS Auto Calibration and Fully-Programmable MBIST," Sunghwan Joo et all, ISSCC 2026

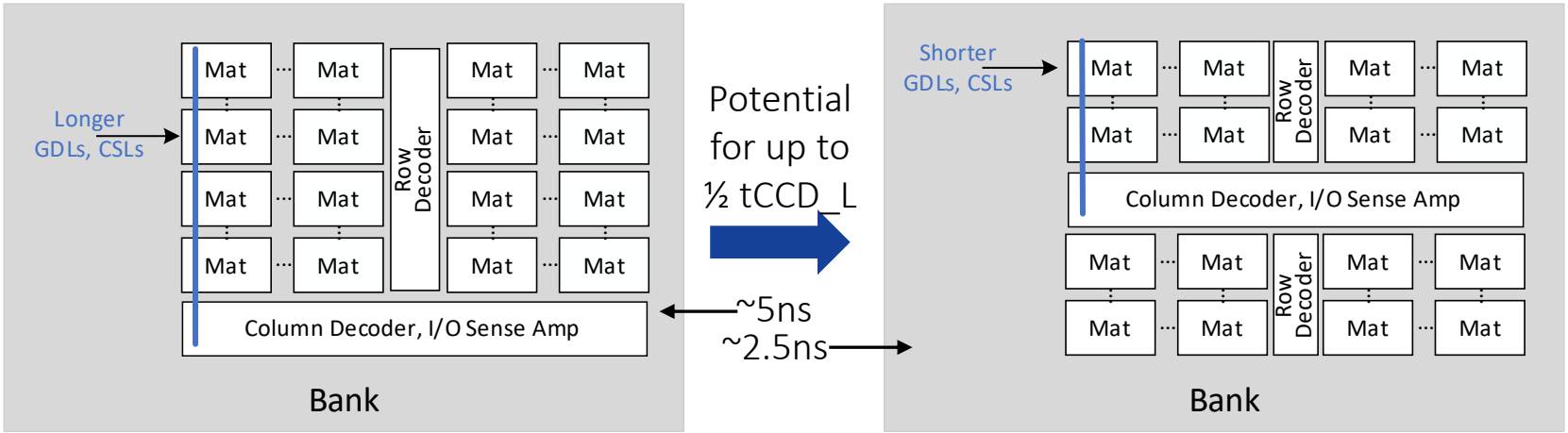
Shorter Internal Paths Improve Core Speed but Raise Cost

DDR - LPDDR

- IO sense amp at the end of a bank
- Reduced overhead for lower cost
- Shared logic within the bank group (BG) for timing control, decoder, etc.

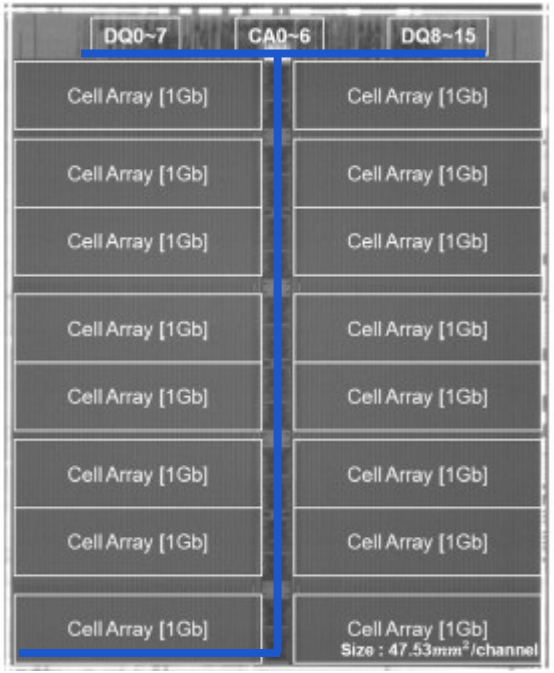
GDDR - HBM

- IO sense amp in the middle of a bank
  - Shorter path from BLSA to IOSA
- Reduced tCCD_L, faster core frequency
- Higher cost (duplicated logic, routing)



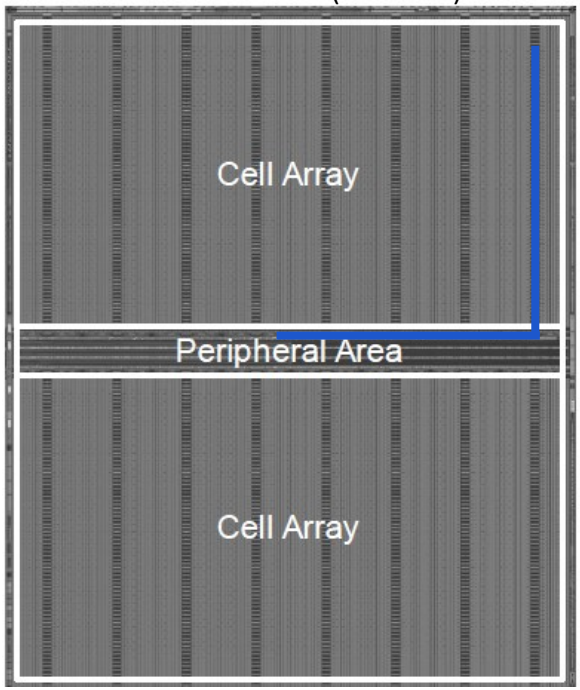
Packaging and Pad Placement Impact Data Latency

Edge Pads (LPDDR5X[1])



Longer internal DQ path

Center Pads (DDR5[2])



Shorter internal DQ path

Edge pads

- ✚ Enables low-cost multi-die packages with wire bonding technology
- ▬ Higher internal DQ latency

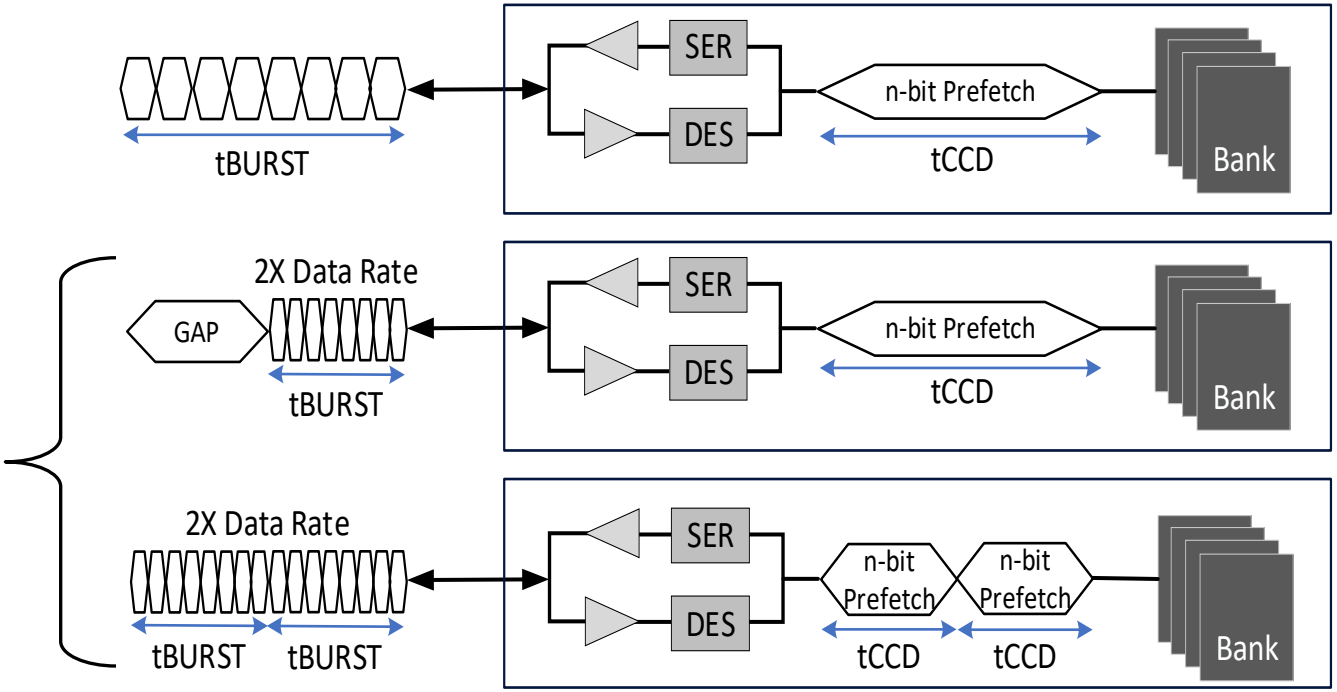
Center pads

- ✚ Lower internal DQ latency
- ▬ More expensive multi-die packaging with TSVs (3DS)

[1] A 16Gbit 9.5Gb/s/pin LPDDR5X SDRAM w/ Low-Power Schemes Exploiting Dynamic Voltage Frequency Scaling …, ISSCC 2022 28-3

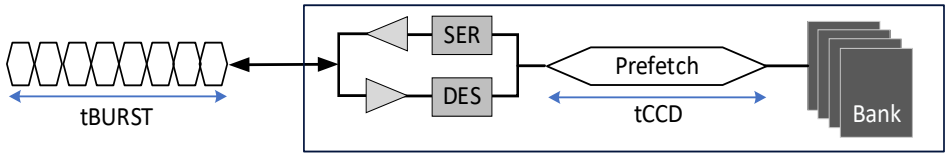
[2] A 1.1V 6.4Gb/s/pin DDR5 SDRAM with a Highly Accurate Duty Corrector and NBTI-Tolerant DLL, ISSCC 2023 28-7

Bandwidth Scaling Needs More Than Faster I/O

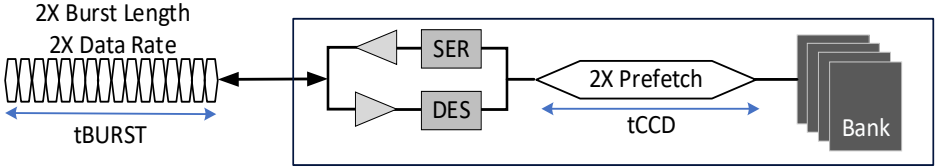


Shorter tBURST
- Harder to hide bank timing overheads
- More likely to be command/address bandwidth limited

Additional Knobs are Used to Scale Bandwidth

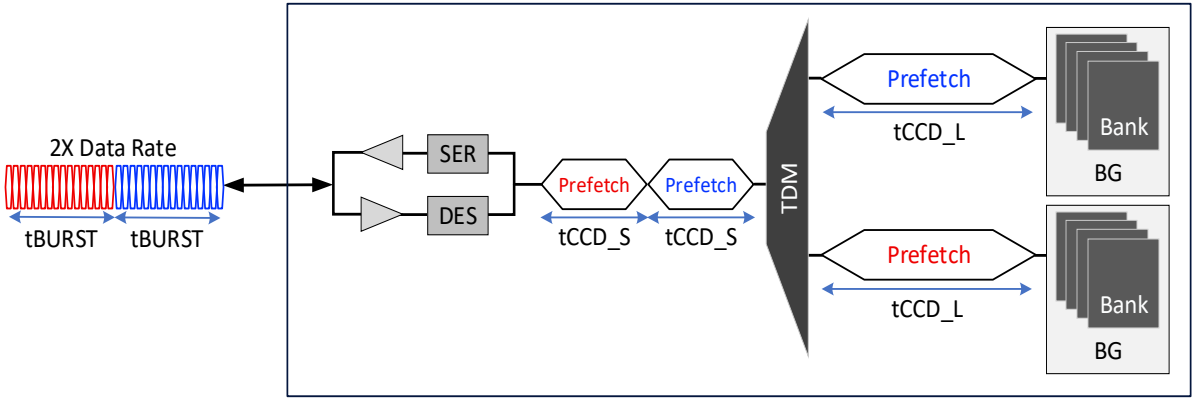


Same current generation baseline



Double the prefetch
Double the burst length
(2X access granularity)

DDR1: BL2, 2n prefetch
DDR2: BL4, 4n prefetch
DDR3: BL8, 8n prefetch

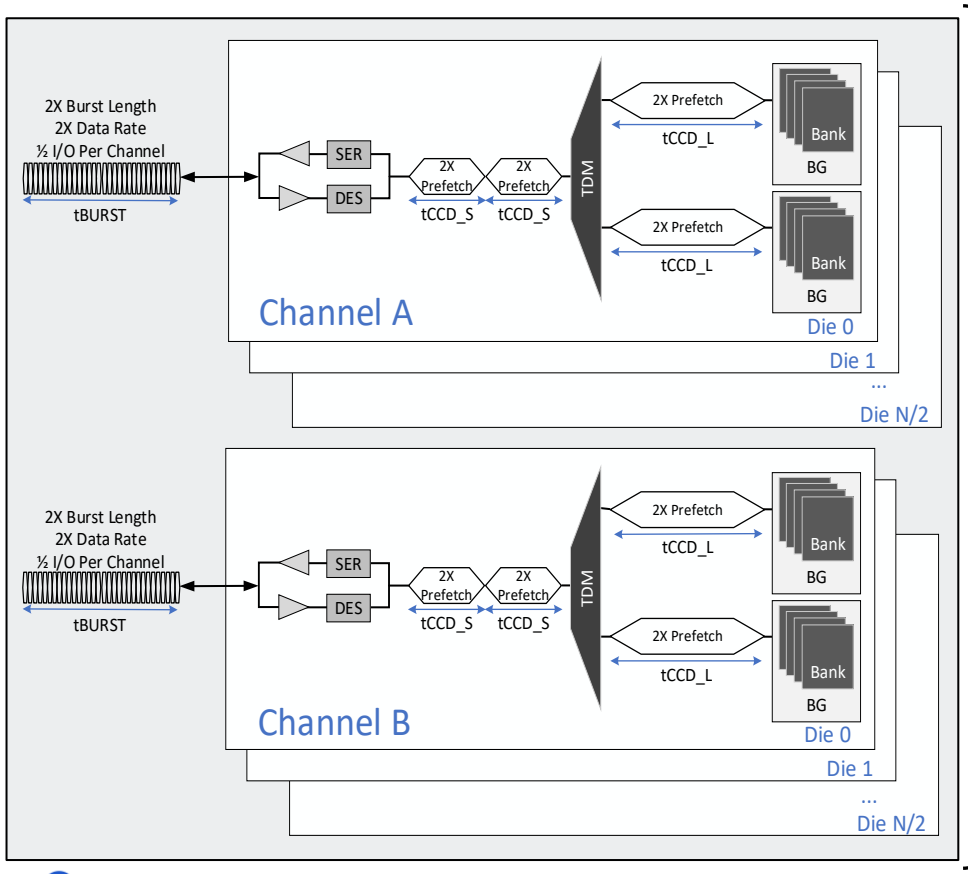


Leverage Bank Group Architecture

Time delay multiplexing between BGs to effectively double the core frequency

DDR4:
BL8, 8n prefetch
→ 64B / burst (x64b DIMM)

DDR5 Continues to Scale w/o Increasing Access Granularity



DDR4 Baseline

- BL8, 8n Prefetch
- One 64b channel per DIMM

➢ 64B per burst from N devices per channel

Changes for DDR5

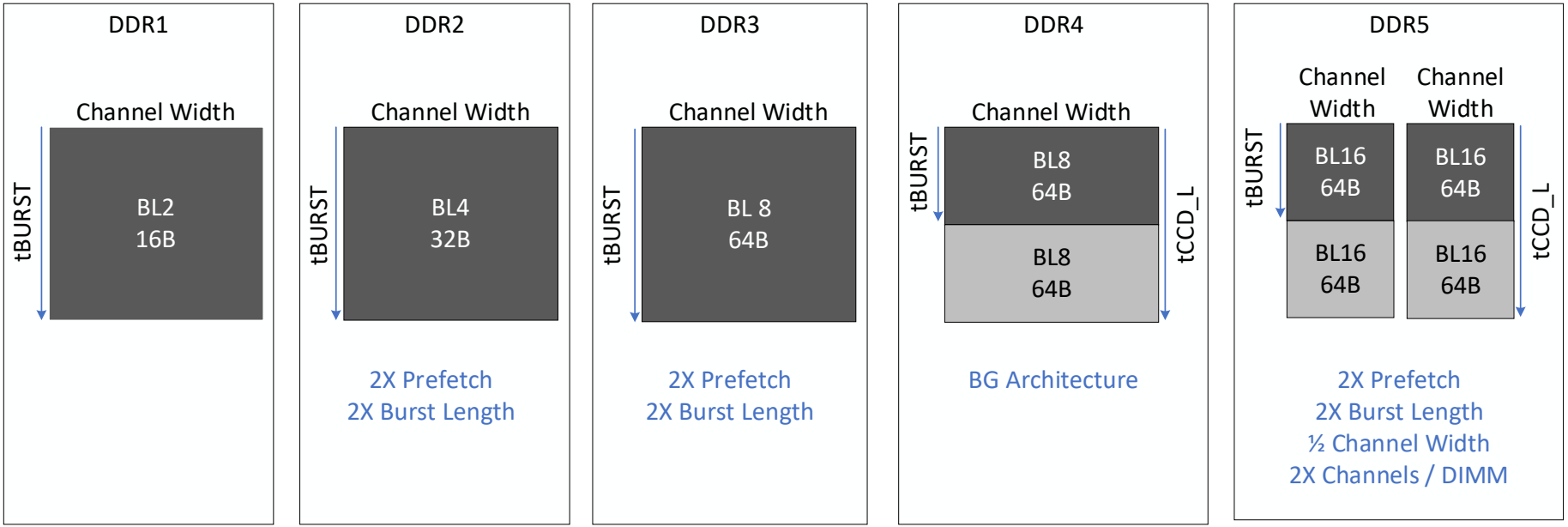
- Double the prefetch to 16n
- Double the burst length to BL16
- ½ channel width
  - 32b channel
  - 2 channels per DIMM

➢ 64B per burst from N/2 devices per channel

➢ Fewer data devices increase RAS overheads

Balancing Prefetch, Core Frequency and Channel Width

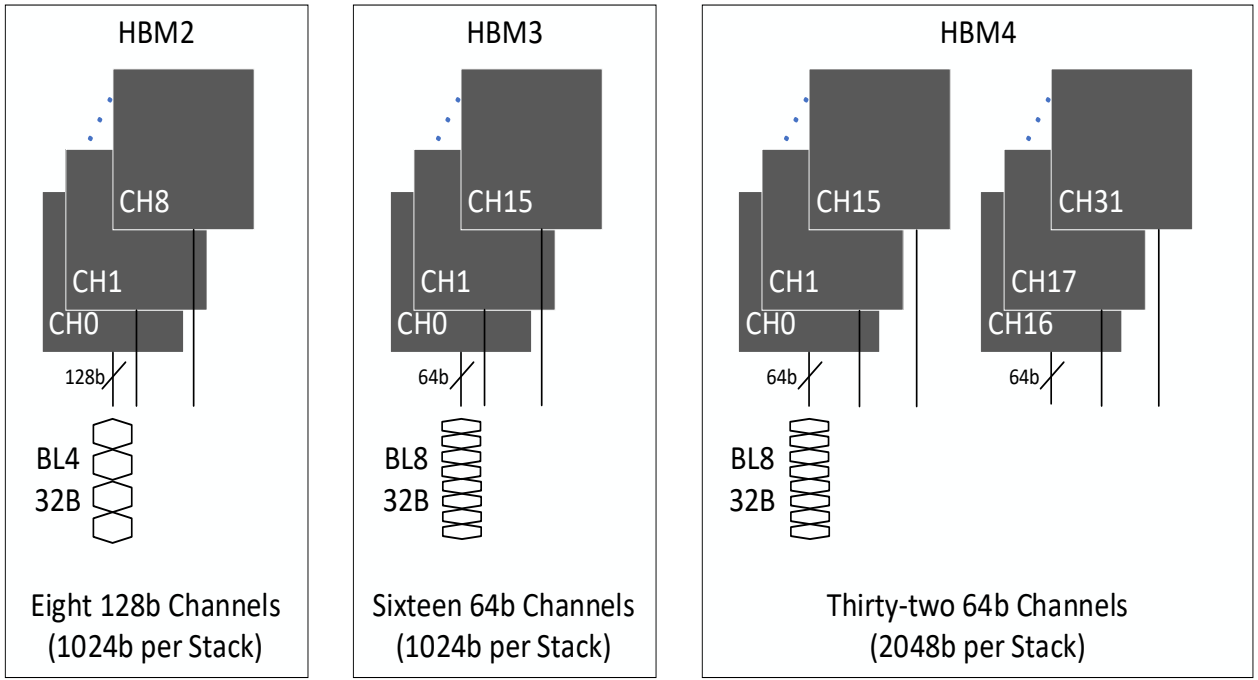
DDRx Evolution



- ↑ SDDC RAS overhead
- What's next?

*tBURST = tCCD (tCCD_S); value depends on data rate

HBM Scales with More Channels



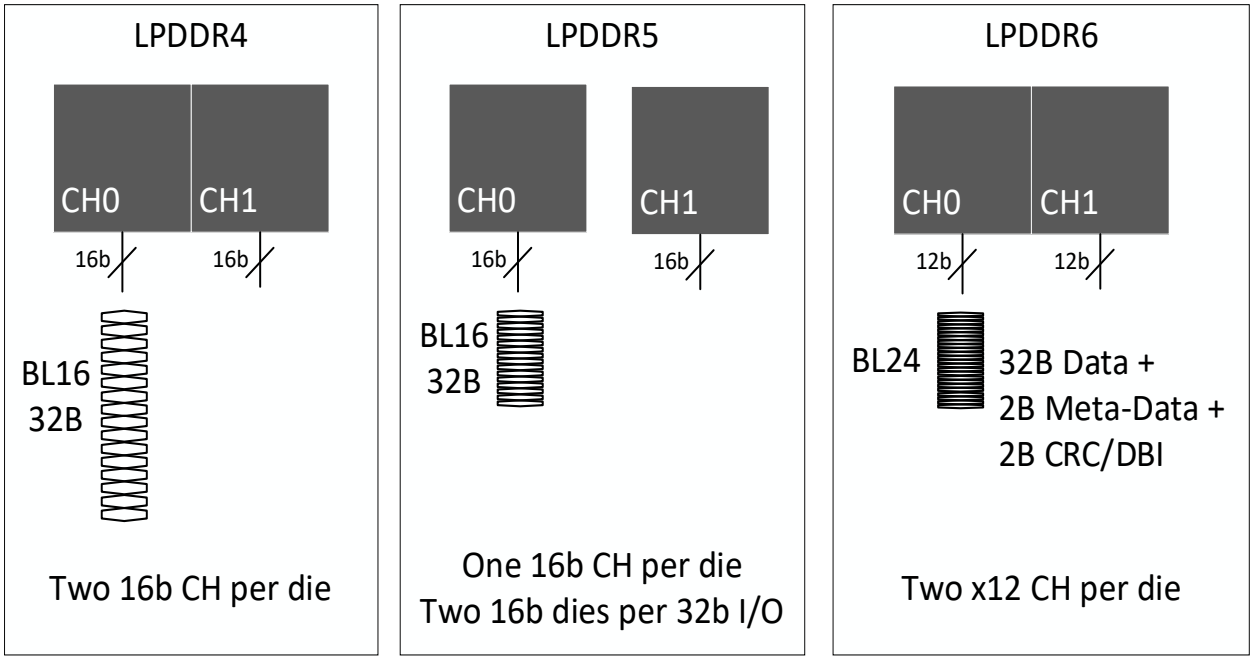
2X Burst Length
½ Channel Width
2X Channels

2X Channels

- Prefetch unchanged to maintain 32B granularity
- Increased number of channels instead

- Total number of DQs doubled with HBM4
  - Minimizes data rate increase at the expense of pins
- Aggressive bandwidth targets to meet AI req.

LPDDR Shifts to non-Log2 Data Widths and Bursts



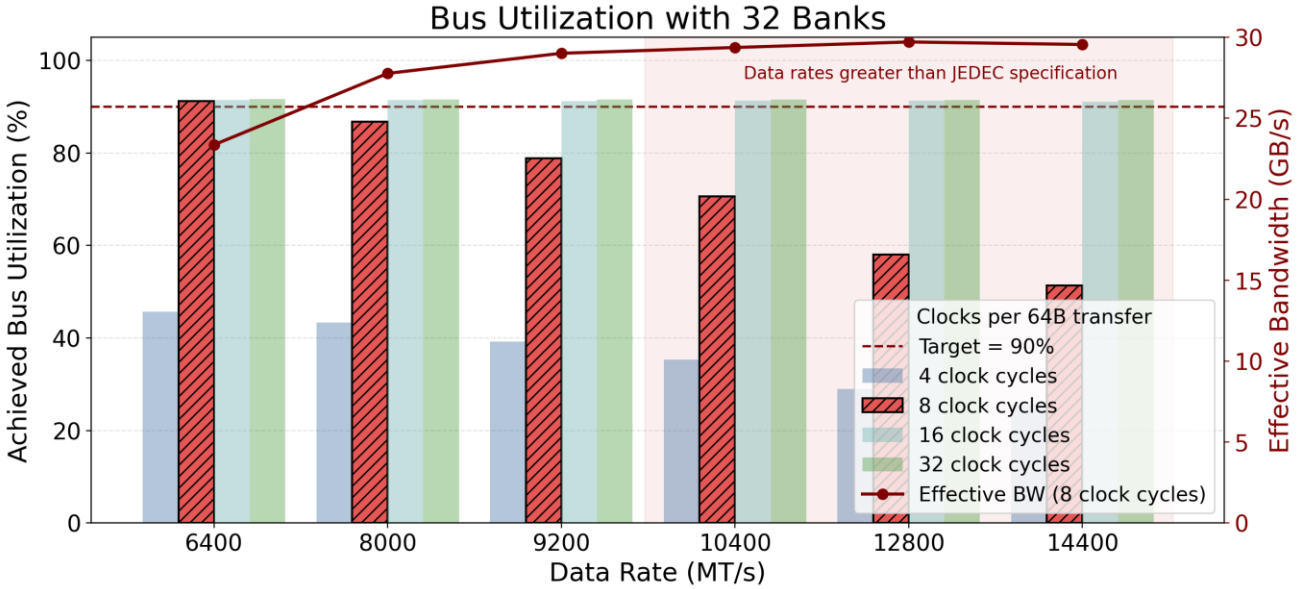
- Prefetch unchanged to maintain 32B granularity
- LP6 introduces non-log2 burst and data width
  - 288b packet
  - Includes 256b data with optional 16b meta-data and 16b CRC/DBI
- LP6 balances pin count and frequency increase to achieve required BW

BG Architecture

1.5X Burst Length
1.5X I/O per die

*LP5 die can be configured in x8 mode with a 16B burst per die
*LP6 die can be configured in 1-CH, x12 mode with a 32B burst per die

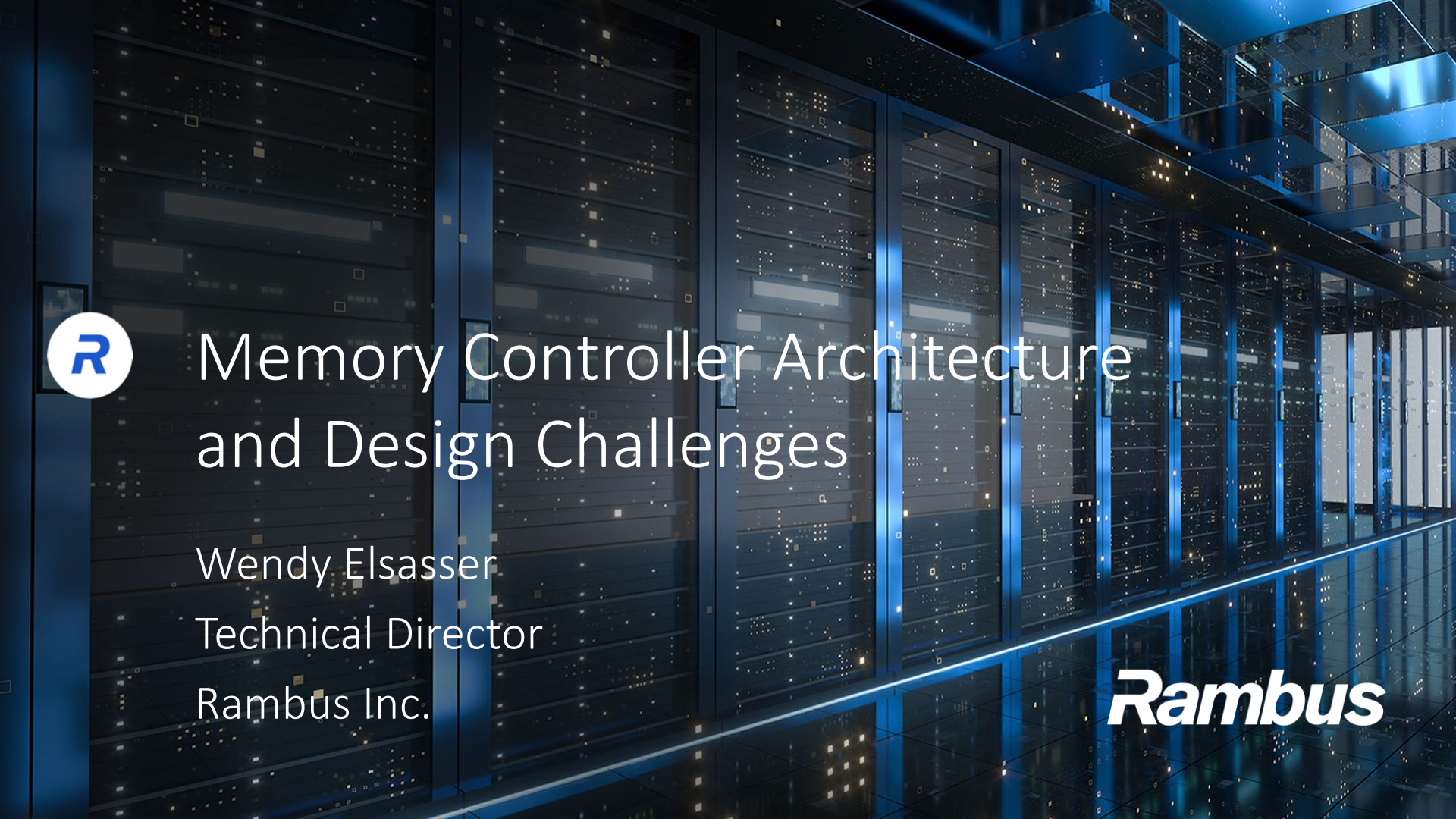
Pushing Data Rate Alone Provides Diminishing Returns



- gem5 simulation
- DDR5 1-Rank
- Random 64B traffic for all burst lengths
- Read only traffic
- 64-deep queue

- Initial DDR5 specification target of 6400 MT/s extended to 9200 MT/s
- Analog timing values (I.e., tCCD_L) do not scale, limiting effective bandwidth

New DRAM generations follow when increasing data rates outrun current architectures

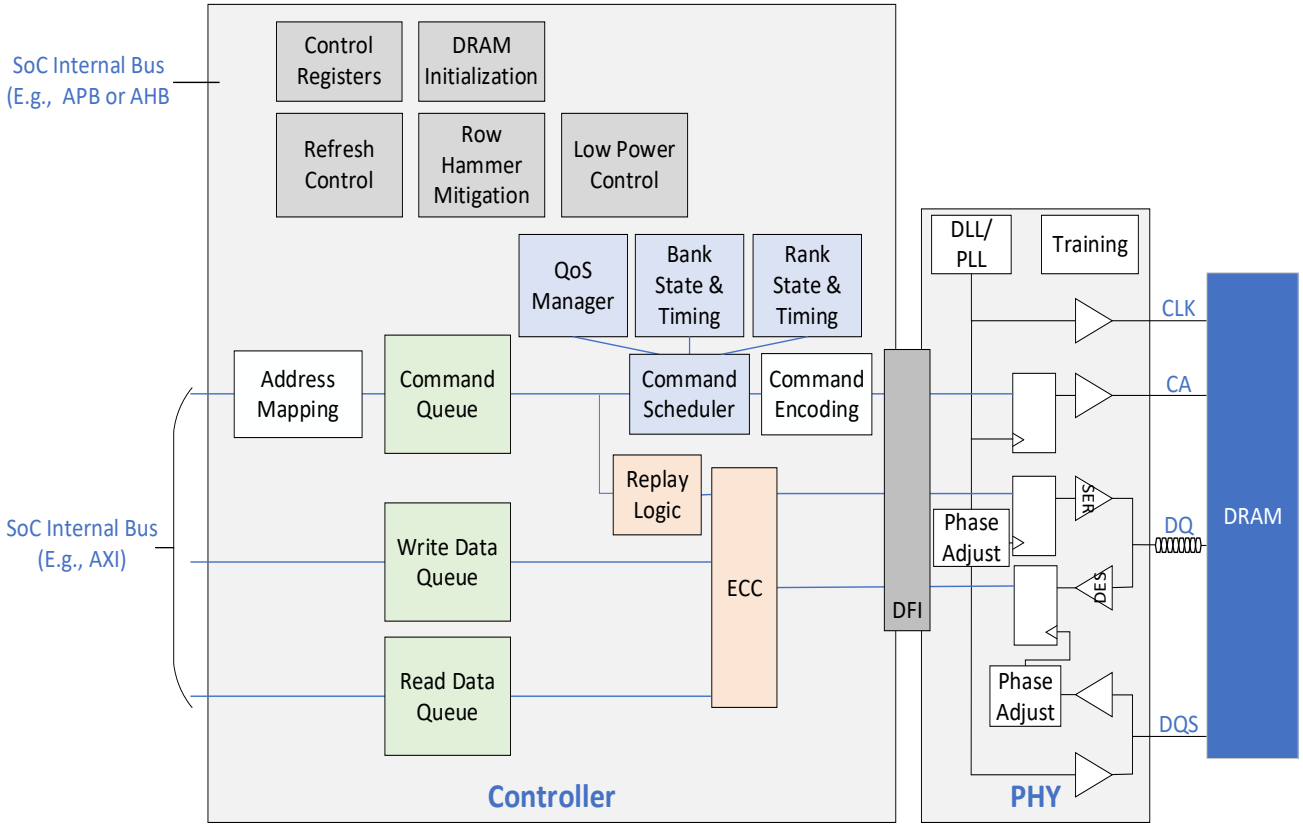


Memory Controller Architecture and Design Challenges

Wendy Elsasser
Technical Director
Rambus Inc.

Rambus

Controllers Orchestrate (most) Everything the DRAM Does

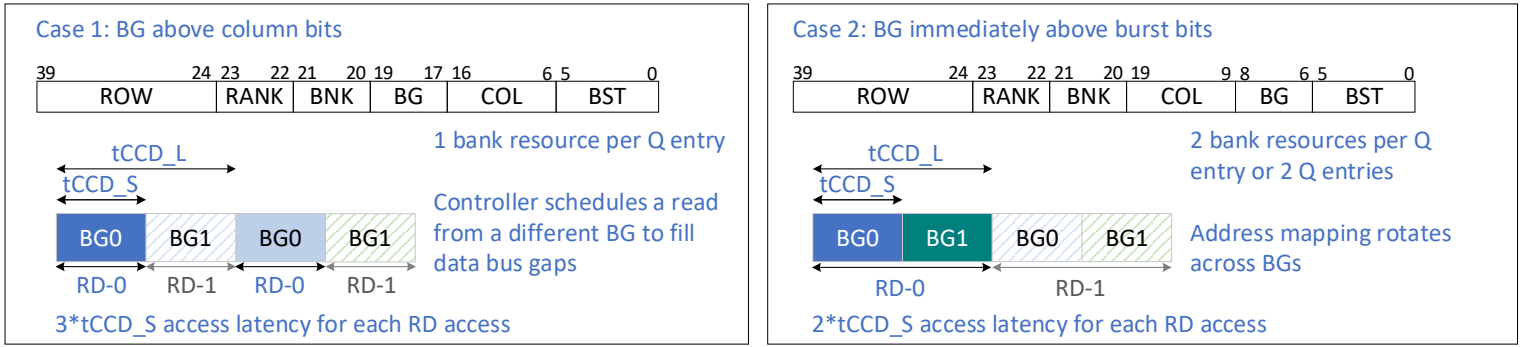


Controller Functions

- Address mapping
- Track bank / rank state, open rows and timing eligibility
- Manage and schedule memory requests
- Support required RAS functions
- Coordinate DRAM management operations
- Direct DRAM low-power state entry/exit

Address Mapping Shapes DRAM Locality and Parallelism

- Controller maps host physical address to DRAM address
  - Host Physical Address → Rank, Bank Group, Bank, Row, Col
- Mapping determines if accesses preserve row locality or rotate across banks / BGs
- Reduce same-bank, different-row conflicts (limit row-miss overheads)
  - I.e., Mapping row bits to higher order PA bits is a common choice to preserve locality
- Along with scheduler, minimize tCCD_L penalty by rotating across bank groups



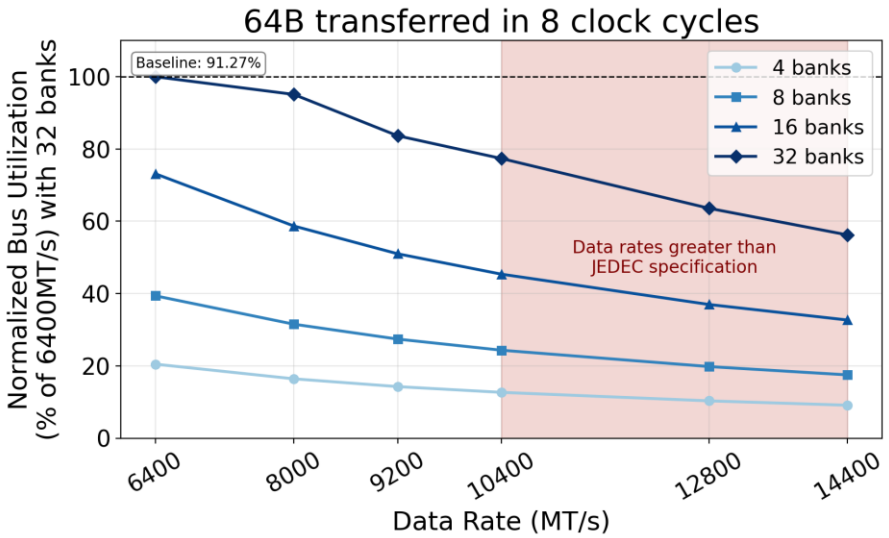
Example: 128B read request split into two DRAM bursts

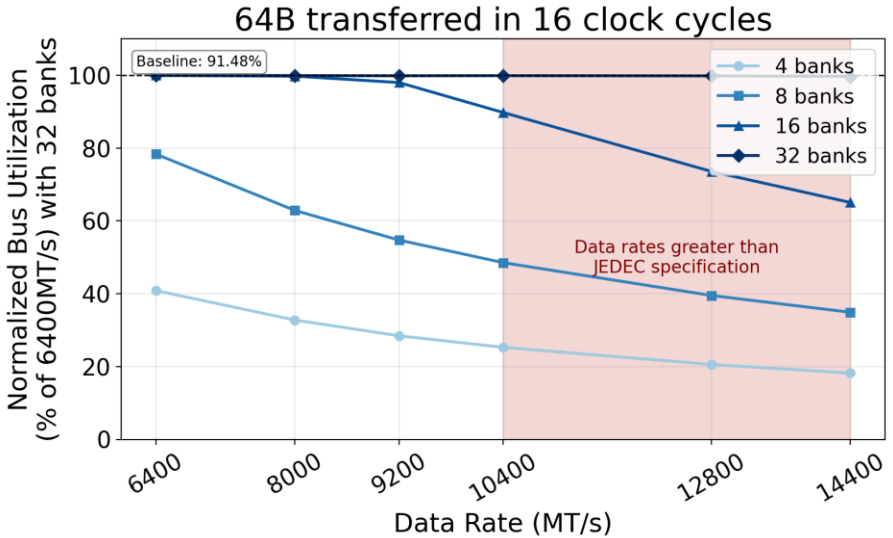
DRAM Schedulers Continuously Balance Tradeoffs

Schedulers balance locality, parallelism, bus efficiency, timing legality, fairness and QoS

Scheduler decision	Optimization goal	Tradeoff / challenge
Which request should win? (row-hit, oldest, high-priority)	Improve throughput, reduce loaded latency or meet QoS	May starve row-conflicting or lower-priority requests (increased tail latency)
Issue reads or drain writes?	Reduce read latency while amortizing RW turnaround penalties	Queue buildup if writes delayed too long Large write batches increase read latency
Keep row open or precharge?	Exploit row-buffer locality; Reduce ACT/PRE overhead	Wrong policy can increase future conflicts Locality / latency benefit vs power
Which bank/rank should issue next?	Exploit bank/rank parallelism and hide timing gaps	Requires detailed timing/state tracking and sufficient request diversity
Prepare a bank operation early?	Reduce visible ACT/PRE latency	Limited lookahead, uncertain future locality and cross-coupled constraints
Should QoS override locality? (or fairness)	Meet latency and bandwidth guarantees	Strong QoS constraints reduce opportunistic scheduling freedom

Bus Utilization Degrades with Low Bank Parallelism





For random traffic, independent bank resources needed scales roughly with $\frac{tRC}{MAX[tCCD_S, tRRD_S]}$

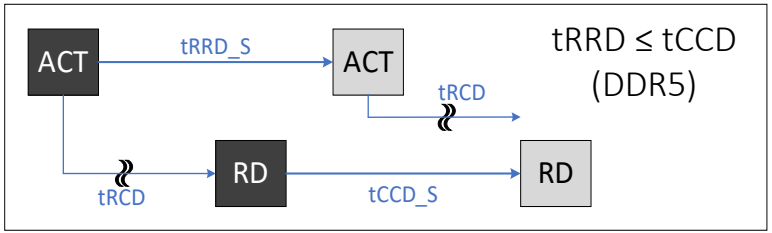
Adequate bank count and scheduling with bank-rotation is crucial with random traffic

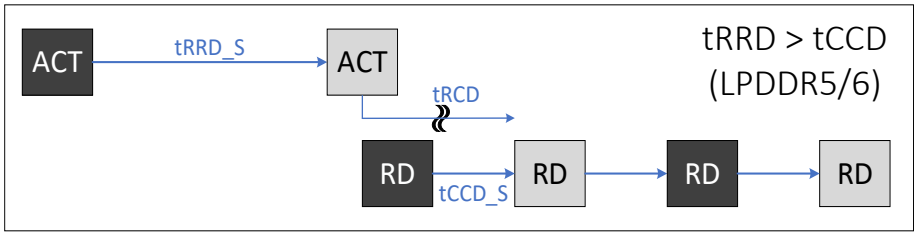
*gem5 simulation data for DDR5 with random 64B read traffic

Beyond Rotation: Why Random Traffic Is Hard to Schedule

Throughput can be ACT-limited

- If tRRD > tCCD, one burst per ACT cannot sustain full bandwidth
- High utilization thus requires multiple column bursts per ACT



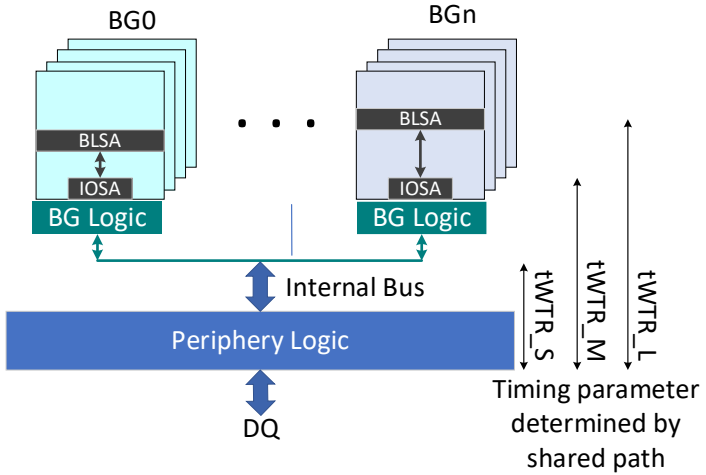


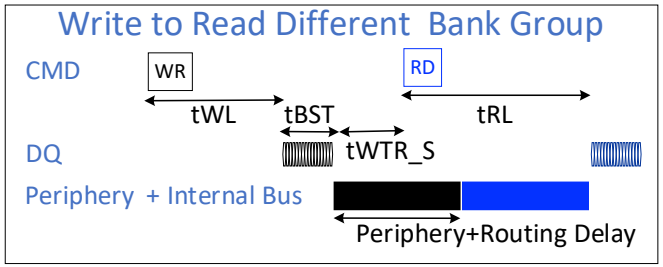
Command scheduling gets harder

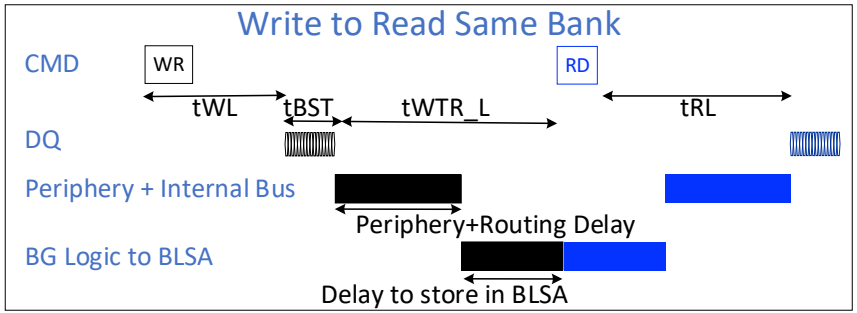
	tBURST	Act	Pre	RW
DDR5	8	2	1	2
LP6	6	4	2	2
HBM4	2	1.5	0.5	1

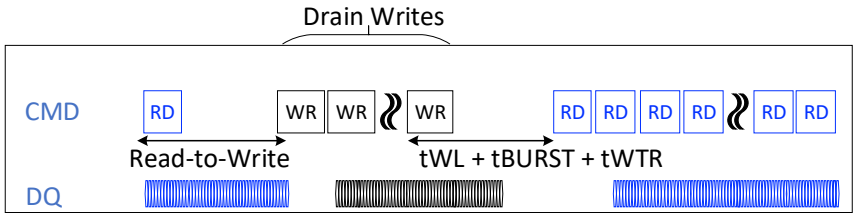
- Minimal open command slots
  - Auto-precharge can help
- Long command occupancy reduces command scheduling flexibility
- HBM separates row / column buses, but still shares command resources across 2 PCs

Read/Write Switching Creates Large Efficiency Gaps



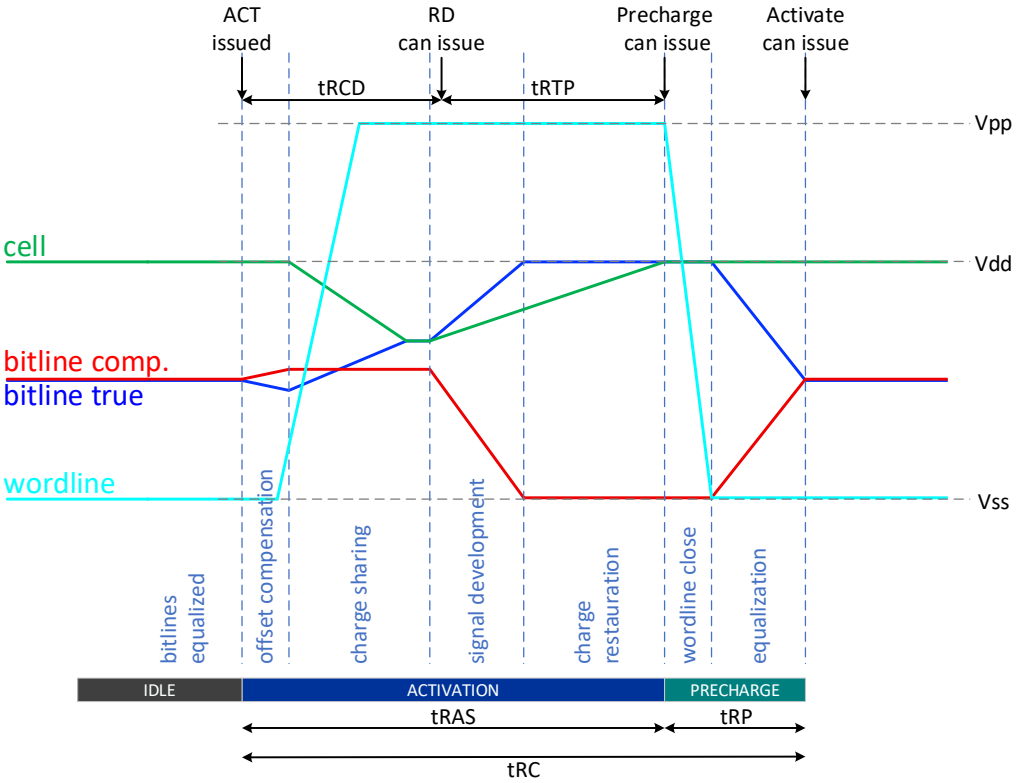






- Controller issues groups of reads and writes to amortize the penalty
  - Reads have higher priority
  - Switch to writes (different rank optimally) after reads complete or a threshold of writes enqueued
  - Improves efficiency but can hurt read latency

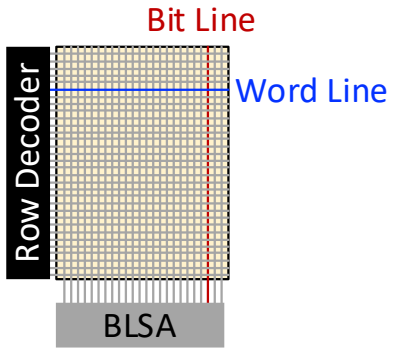
Bank Timing Constraint Review



*Precharge further delayed by tWR after burst completes with writes

*Read command decode overlaps final stages of signal development

- Bit line length impacts delays (subarray architecture)

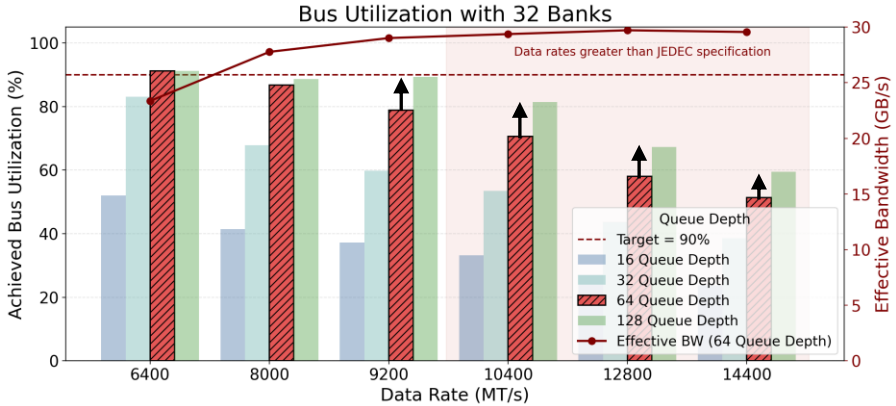


  - Longer lines → Higher Cap → Larger delay
  - All timing values impacted
- Delays vary across technologies but have not significantly changed within a technology

Data Rate Scaling Increases Scheduling Pressure

Fixed delays consume more clock cycles as data rates rise

Parameter	Delay (ns)	Delay (tCK)		
tRC	48	78	154	221
tWR	30	48	96	138
Write-to-Read(_S)	~17-20	28	61	92
Write-to-Read(_L)	~25-27.5	40	85	127
DDR5 Data Rate (GT/s)		3.2	6.4	9.2



Why controller pressure rises

- More clocks per delay calls for deeper reordering
- Increasing RD/WR commands in flight
  - E.g., 10 reads can issue in a tRL window (DDR5-9200)
- Large DRAM bus turnaround clock cycles motivate larger grouping of reads and writes

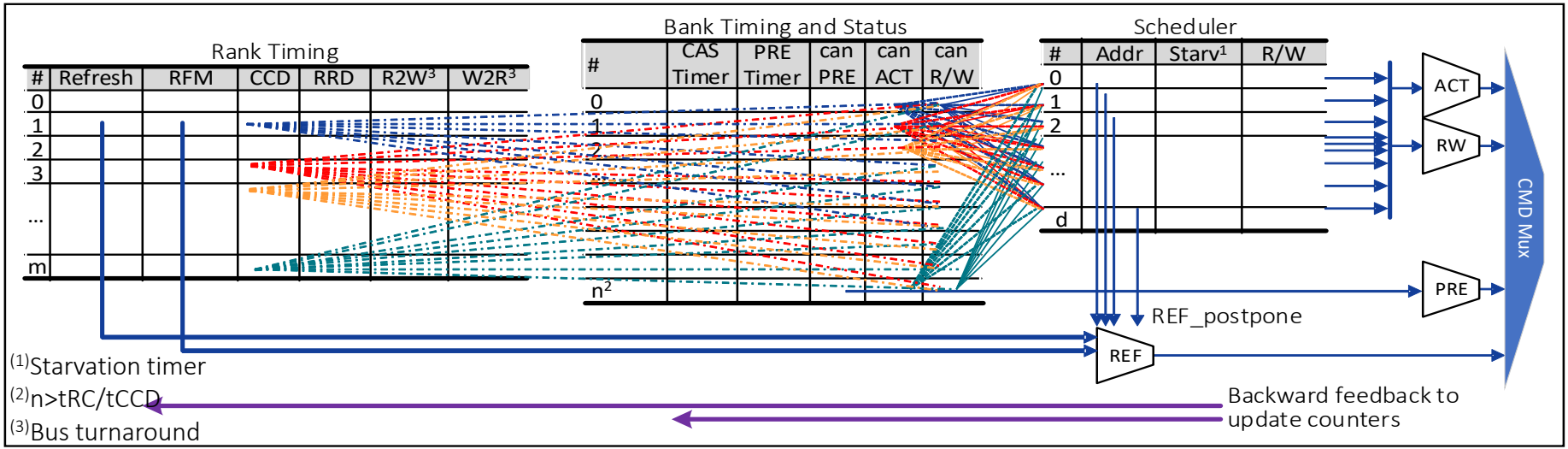
Q depth helps only if bank diversity is sufficient

- For 6400 simulations, ~64 entries are sufficient
- Larger Qs provide modest benefit at higher rates
  - 32 banks cannot fully cover tRC
- **Higher data rates need both deeper lookahead and sufficient independent bank resources**
- **Deeper Qs negatively impacts latency under load**

*gem5 simulation data for DDR5 with random 64B read traffic

Memory Controller Complexity is Increasing

- Many timing constraints must be tracked simultaneously
- Almost All-to-All connection between
  - Rank logic for channel CMD timing: CCD/RRD/R2W/W2R… and “same-bank refresh”+”RFM”
  - Bank logic for per-bank timing: tRCD/tRAS/tWR
  - Scheduler w/ starvation timer and QoS prioritization
- Bottleneck for timing closure – function of queue depth, number of banks, number of ranks
  - Can run controller at lower frequency (i.e., 4:1 clock ratio) but controller will need to make multiple decisions per clock



Enabling Communication with DRAMs: Modules, PHYs, and Buffer Chips

Robert Palmer
Senior Technical Director
Rambus Inc.

Rambus

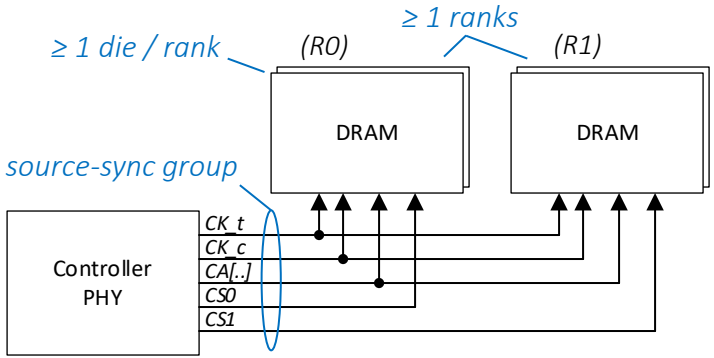
Outline

- DRAM memory channel overview and examples
- Data transfers
  - Command to data burst synchronization
  - Source-sync timing alignment
  - Read data realignment
  - Rank timing sets and switching
  - Bus turnaround
- Interconnect and signaling
- Capacity scaling
  - RDIMM example
- Bandwidth scaling at capacity
  - MRDIMM example
- Summary

Command / Address Bus

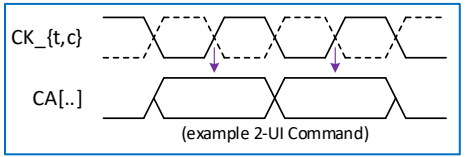
Command / Address (CA) Bus

- Unidirectional, source-sync to CK
- Multi-die signal loading is common (multi-die channels, ranks)
- Lower BW vs. data bus – fewer pins/ch, often lower transfer rate
- Commands often multi-cycle
- May implement other functions (e.g. parity, error alert)

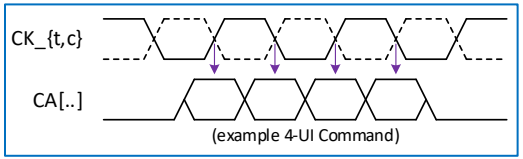


Technology	Interface Unit	Pin Counts			CA Timing	CA/DQ Per-Pin BW
		CA	CS	CK		
DDR5	Channel (x32)	14	1 / rank	2	SDR_CK	0.5
DDR5, w/ RCD	Channel (x32)	7			DDR_CK	1
LPDDR5(X)	Channel (x16)	7			DDR_CK	0.25
LPDDR6	Subchannel (x12)	4			DDR_CK	0.5
GDDR7	Channel (x11)	5	-	2 (WCK)	SDR_WCK	0.32
HBM4e	Channel (x64)	18	-	2	DDR_CK	0.5

SDR CA



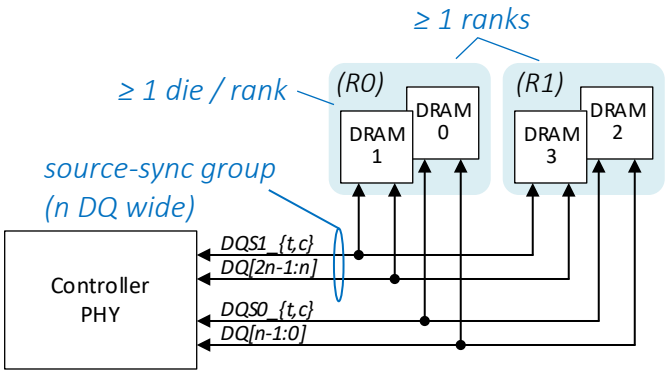
DDR CA



Data Bus

Data (DQ) Bus

- Bi-directional data pins, source-sync to DDR data strobe/clock
  - grouping based on DRAM device width (e.g. x4 → 4 DQ per DQS)
- Data strobe/clock may be bi-directional or unidirectional
- Physical ranks introduce multi-die/package signal loading
- May implement other functions (e.g. DBI, DM, CRC, ECC, ...)



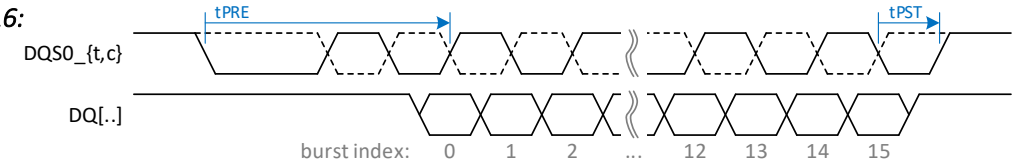
Technology	Burst Length	Interface Unit	Access Granularity [B]	Pin Counts DQ	DQS_{t,c}	WCK_{t,c}	RCK_{t,c}	WDQS_{t,c}	RDQS_{t,c}	Max Speed Bin [GT/s]	Max Payload BW [GB/s]
DDR5	16	Channel: 8x4	64	32	16	-	-	-	-	9.2 (1)	36.8
		Channel: 4x8			8	-	-	-	-		
LPDDR5(X)	16	Channel: 2x8	32	16	-	4	-	-	4	10.7 (1)	21.4
		Channel: 1x16			-	4	-	-	4		
LPDDR6	24	Subchannel: 1x12	32	12	-	2	-	-	2	14.4 (1)	19.2
GDDR7	16	Channel: 1x11	32	11	-	2	2	-	-	24 (1)	48.0
HBM4e	8	Channel: 2x32 (PCH)	2 x 32	64	-	-	-	4	4	16 (2)	128.0

multiply by #ch for total interface BW

(1) *per published JEDEC specification*

(2) *per product announcements*

Example DDR5 BL16:

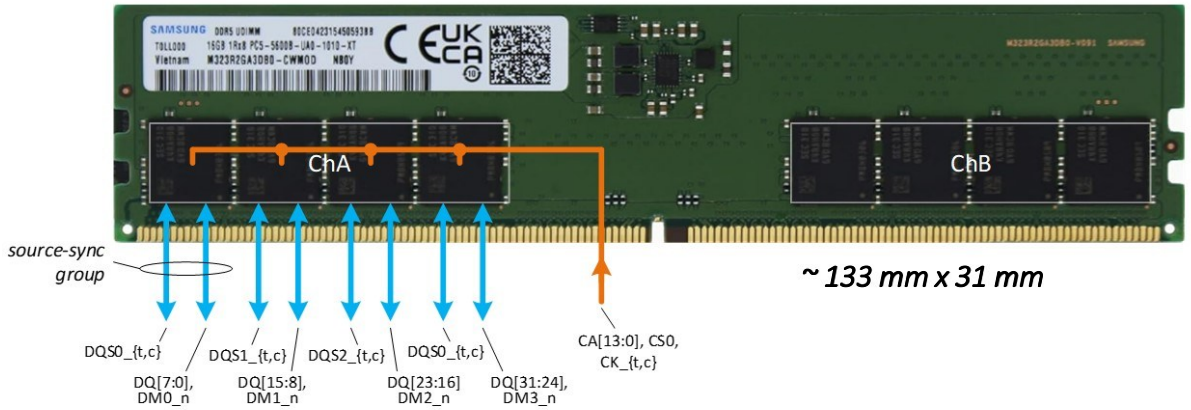


tPRE and tPST are a function of data rate and channel characteristics, e.g. :
tPRE ≈ 4 tCK, tPST ≈ 1.5 tCK at 8800 MT/s

Example DDR5 UDIMM and RDIMM Channel Interfaces

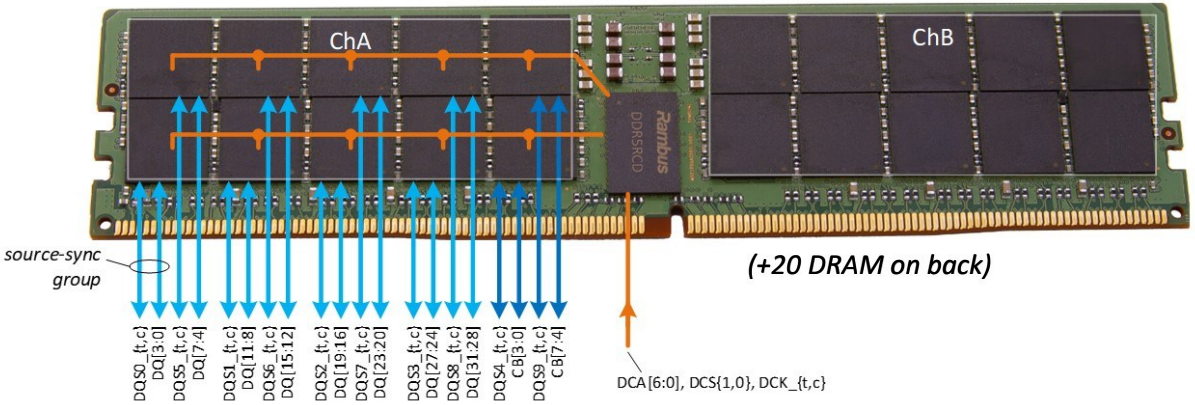
1Rx8 UDIMM

- Two channels (A,B)
  - 32 DQ per channel
  - Four x8 DRAM per rank * 1 rank / ch
  - 8 DQ per DQS_{t,c}
- Unbuffered CA/CS/CK
  - Fly-by routing to 4 DRAM

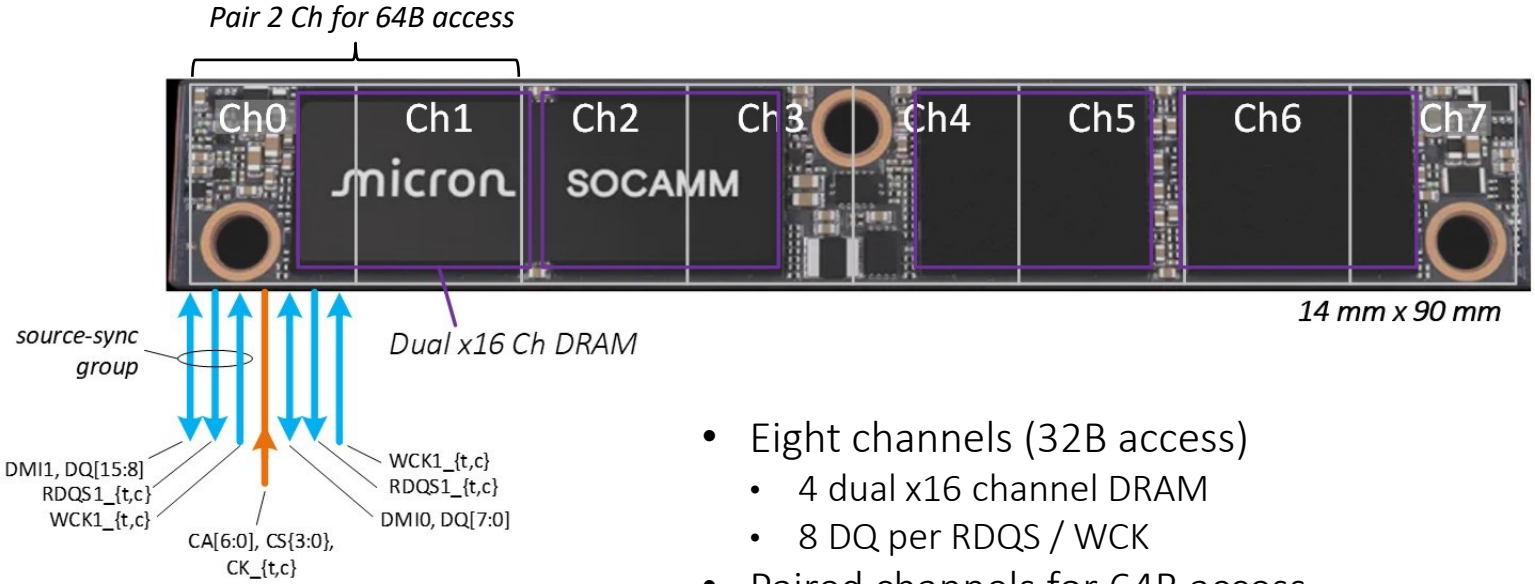


2Rx4 EC8 RDIMM

- Two channels (A,B)
  - 32 DQ + 8 CB per channel
  - (8 + 2) x4 DRAM per rank * 2 rank / ch
  - R0: front, R1: back
  - 4 DQ per DQS_{t,c}
- RCD buffers CA/CS/CK
  - More on this later



Example LPDDR5X SOCAMM Channel Interface



SOCAMM – small outline compression attached memory module

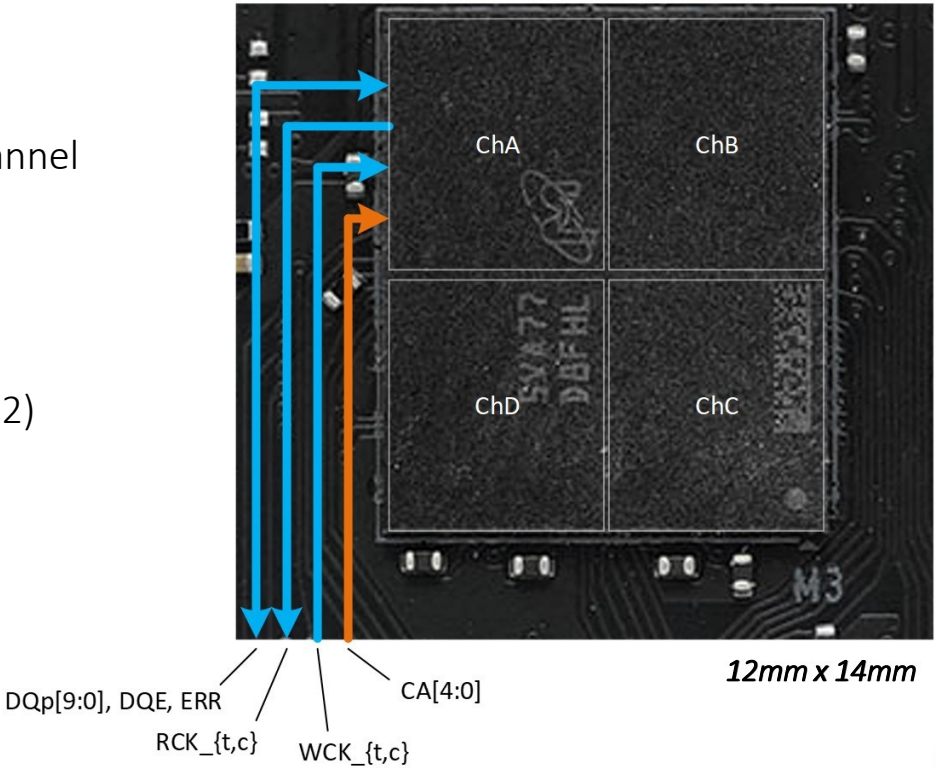
- Eight channels (32B access)
  - 4 dual x16 channel DRAM
  - 8 DQ per RDQS / WCK
- Paired channels for 64B access
  - Two CA bus copies from host – no fly-by routing needed
- Two x16 channels per DRAM package
  - Wire bonded DRAM die stacks
  - Capacity: byte mode and multiple ranks

Example GDDR7 Channel Interface

- 4-channel DRAM device
  - 11 DQ per channel
  - One write and read clock (WCK, RCK) per channel
  - High bandwidth density with 32B access
- Signal modulation formats:
  - CA → NRZ
  - DQ → PAM3 (x11 ch, BL16) / NRZ (x8 ch, BL32)
    - DQ NRZ available at lower data rates (≤ 16Gbps)
- Solder-down (no standard module)
- Single-rank, point-to-point

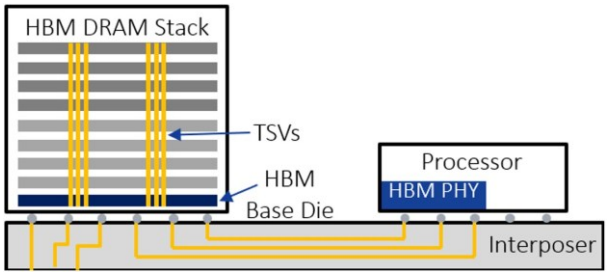
GDDR7 DRAM, Nvidia RTX-5060

https://cdn.wccftech.com/wp-content/uploads/2026/02/RTX-5060-Micron-GDDR7.jpg

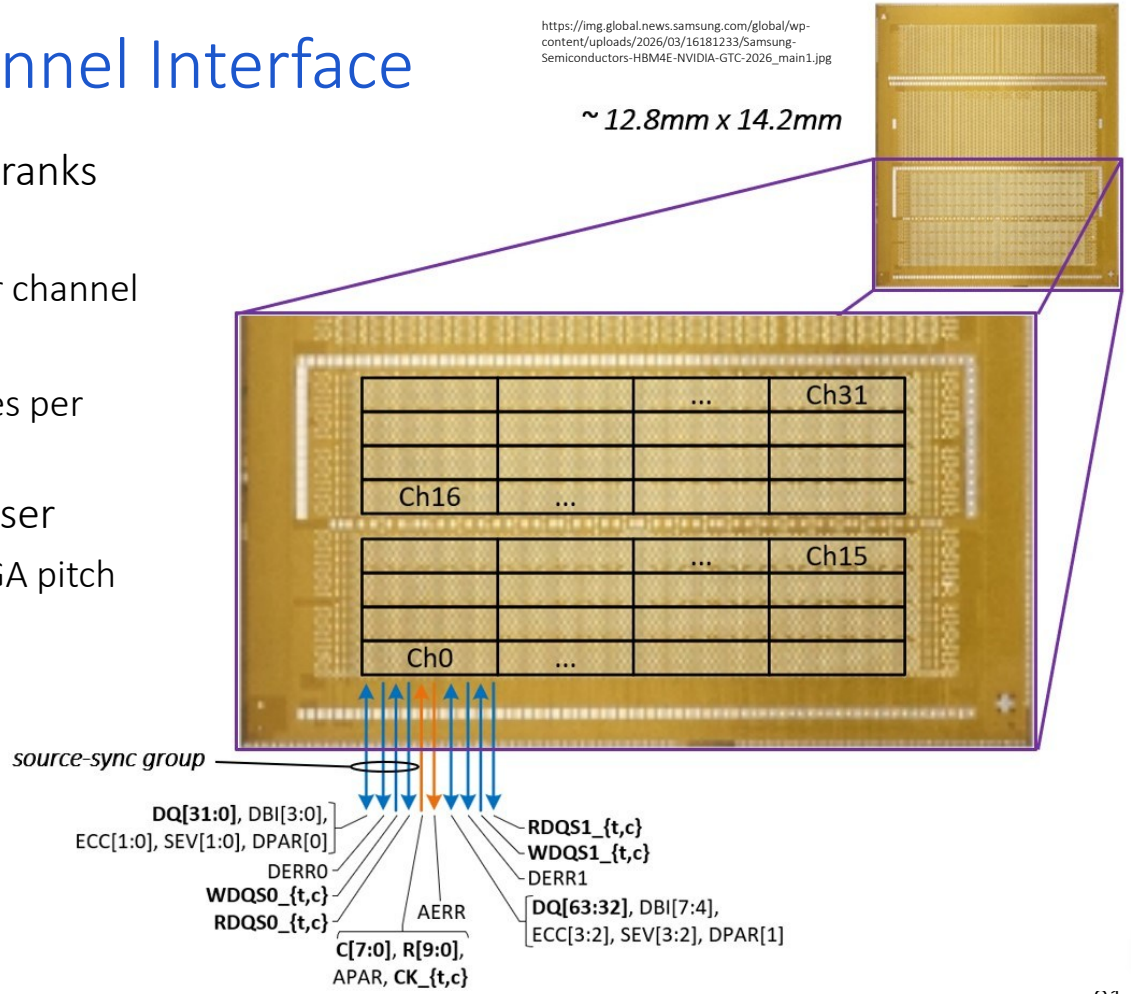


Example HBM4(e) Channel Interface

- Base die aggregates channels and ranks
  - 32 x64 channels
  - Two x32 pseudo-channels (PCH) per channel
  - One CA bus per channel (shared)
  - Unidirectional write and read strobes per PCH
- 2.5D integration w/ silicon interposer
  - 65um min µbump pitch; ~0.1x FCBGA pitch



https://img.global.news.samsung.com/global/wp-content/uploads/2026/03/16181233/Samsung-Semiconductors-HBM4E-NVIDIA-GTC-2026_main1.jpg

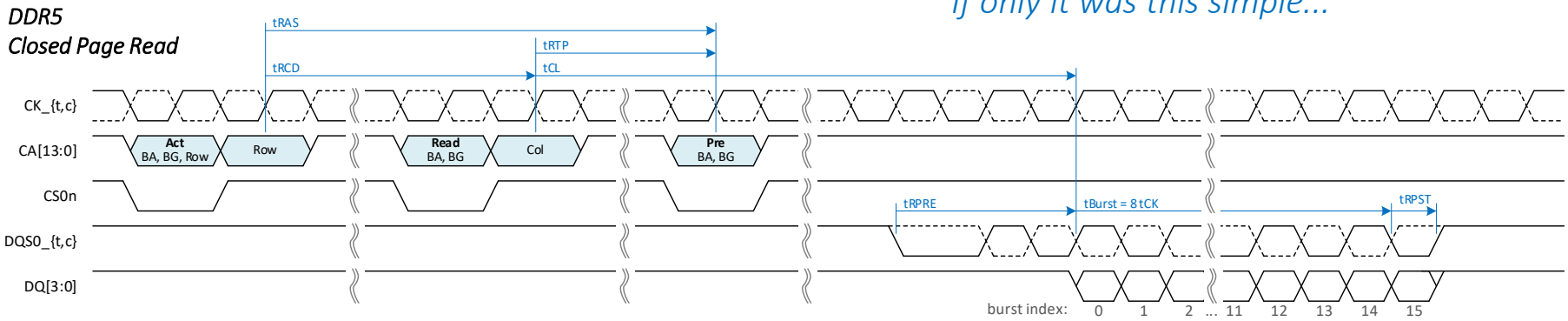


Outline

- DRAM channel overview and examples
- Data transfers
  - Command to data burst synchronization
  - Source-sync timing alignment
  - Read data realignment
  - Ranks timing sets and switching
  - Bus turnaround
- Interconnect examples
- Capacity scaling
  - RDIMM example
- Bandwidth scaling at capacity
  - MRDIMM example
- Summary

Why Is This Complicated?

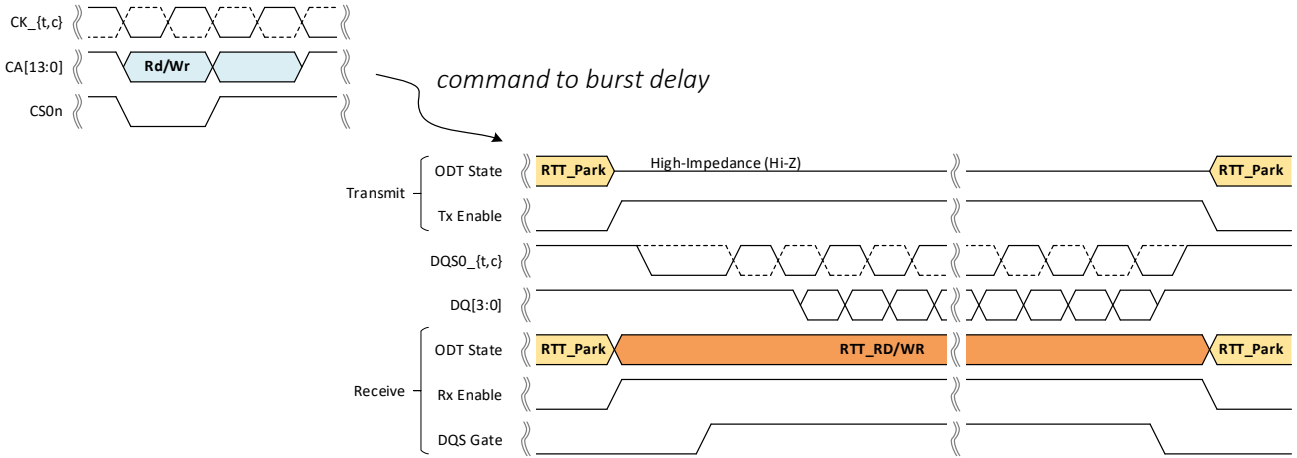
If only it was this simple...



- Devices are non-ideal
  - Silicon process variation
  - Circuit voltage and temperature sensitivity
- The physical channel is non-ideal
  - Material properties and manufacturing variation
  - Signal transmission: delay, loss & dispersion
  - Other signal and power integrity impairments (reflections, crosstalk, SSO/PSIJ)
- Channel composition increases complexity
  - Multi-device ranks and multi-rank channels
  - Aggregated sets of source-synchronous links
  - Multiple module types
- *The DRAM spec is a great place to start but doesn't tell you everything*
- *The host PHY (and buffer chips) must work with the DRAM to compensate system and device non-idealities*

Command to Data Burst Synchronization

- Host and DRAM PHY time their data bus controls relative to the associated Rd/Wr command
  - Tx and Rx enables – enable the appropriate IO circuits for read or write
  - (R)DQS gating – enable internal strobe paths to propagate a clean strobe for use in data sampling and sequential logic
  - On-die-termination (ODT) control – apply the proper signal termination impedance for proper data reception (SI)

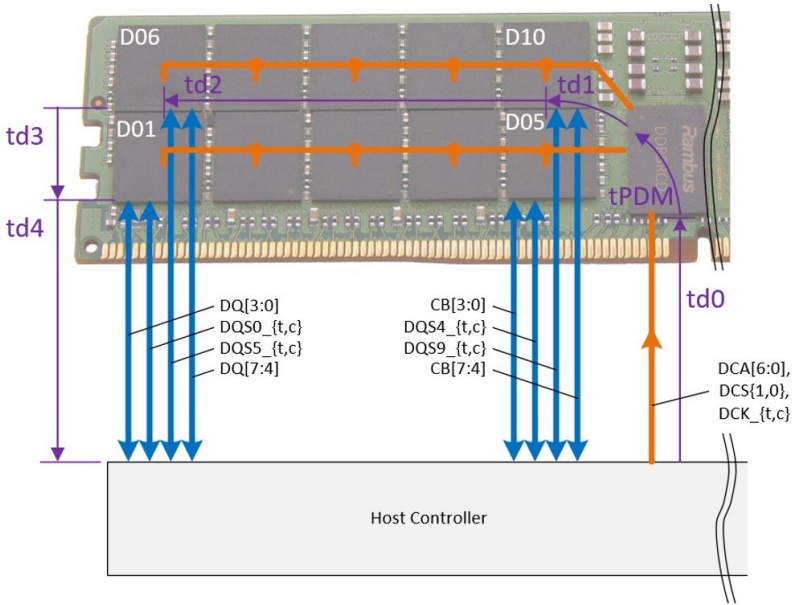


- Preceding example sequences show idealized command to data burst latency
- In practice, the interface must accommodate various timing offsets and skews

  → *DDR5 RDIMM provides a useful case study*

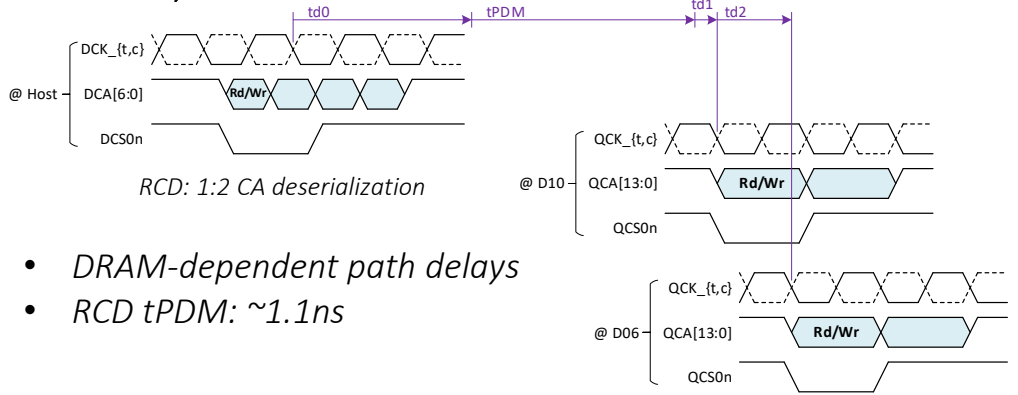
Example Command and Data Path Delays: DDR5 RDIMM

Impact of fly-by routing, unmatched data routes



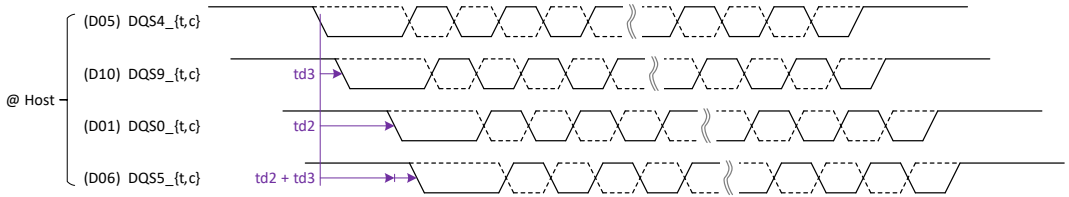
- *Routing delays: ~ 5 to 7 ps/mm*
- *100mm → ~5 tbit @ 8 Gbps*

Command Delays



- *DRAM-dependent path delays*
- *RCD tPDM: ~1.1ns*

Relative Strobe Burst Delays – Read Example

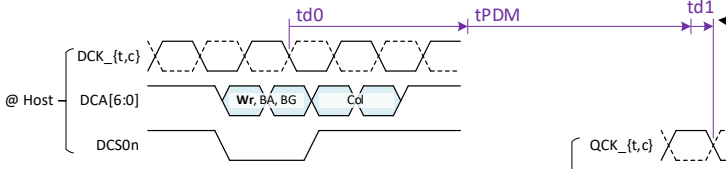


- *DRAM-dependent path delays produce relative delay offsets in strobe/data bursts*

Command to Data Burst Synchronization – Write Example

- Write command to burst timing offset is established by **write leveling**
  - Host PHY adjusts write burst delay relative to ideal CWL, aligns strobe preamble to internal DRAM reference pulse
  - Training is performed for each rank
  - Rank-specific write leveling delay is applied to host transmit controls (Tx enable, ODT, DQS phase)

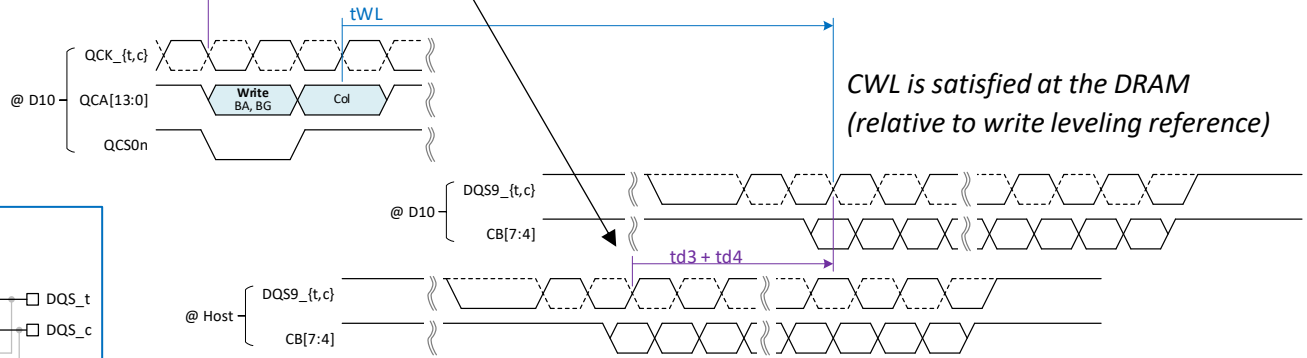
DDR5 RDIMM Example:



After write leveling:

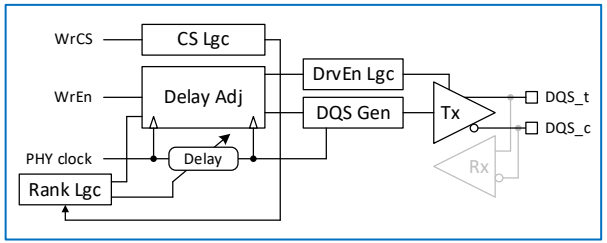
Host has delayed write burst to offset command path delays...

... and advanced write burst to offset data path delays



CWL is satisfied at the DRAM (relative to write leveling reference)

Example host PHY write DQS path

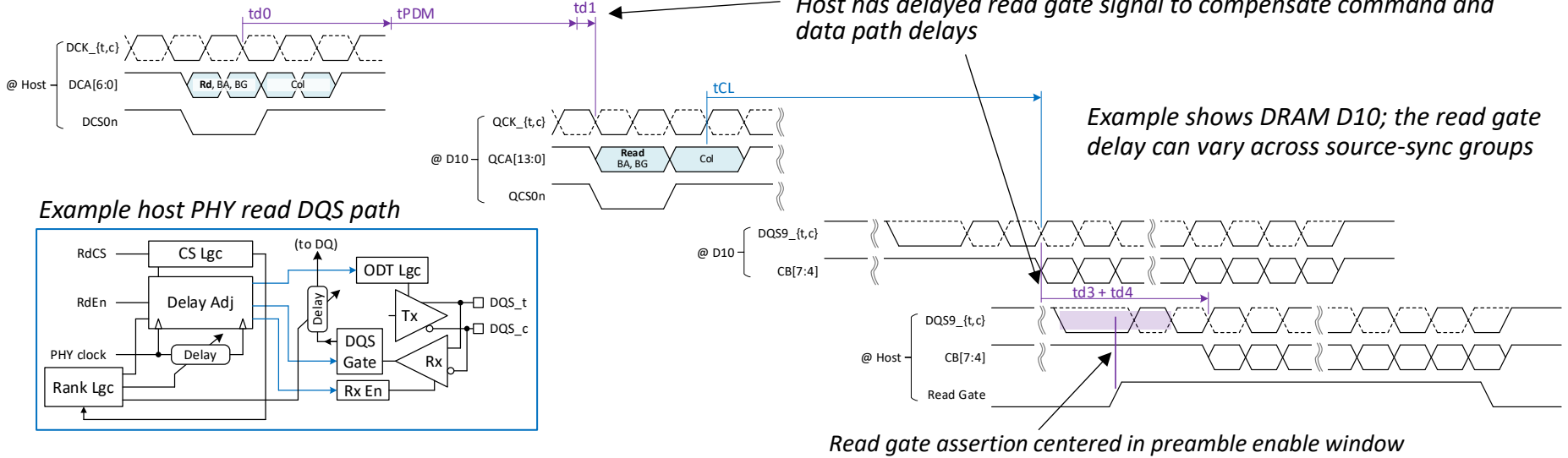


Example shows D10; write leveling delay varies across source-sync groups

Command to Data Burst Synchronization – Read Example

- Read command to burst timing offset is established by **read gate training** (a.k.a. read leveling)
  - Host PHY adjusts read gate enable relative to CL, centering it in the read strobe preamble enable window
  - Training is performed for each rank
  - Rank-specific read gate enable delay is applied to host PHY receiver controls (Rx enable, ODT, gate enable)

DDR5 RDIMM Example:



After read gate training:

Host has delayed read gate signal to compensate command and data path delays

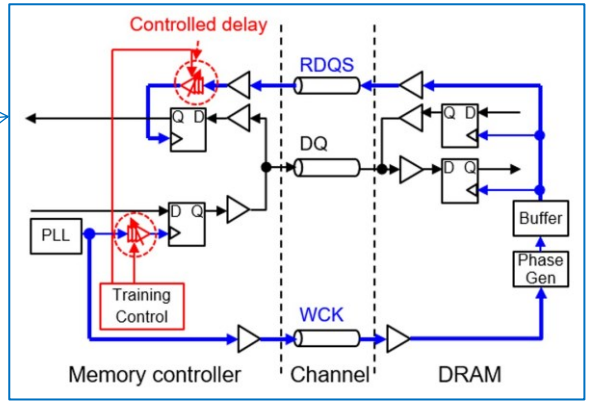
Example shows DRAM D10; the read gate delay can vary across source-sync groups

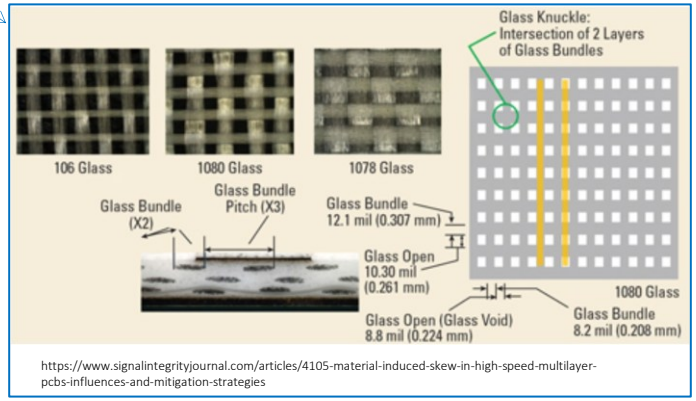
Read gate assertion centered in preamble enable window

Example host PHY read DQS path

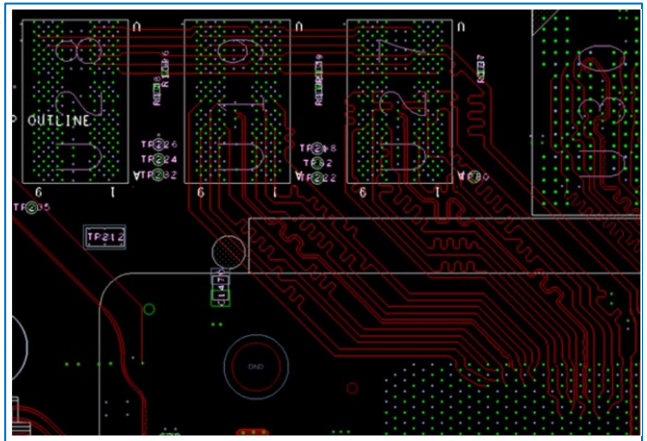
Source-Sync Timing Alignment

- Real channels have timing skew within source-sync groups
- Unmatched DQ and WCK (DQS) receive paths
  - Example: LPDDR5X tWCK2DQI_HF = 250 to 600ps (multiple t_{bit})
- Non-ideal design and manufacturing
  - Interconnect skews: ~10s of ps within a source-sync group
    - Package and board routing skews
    - Propagation velocity variation: e.g. dielectric non-uniformity
  - Manufacturing variation in PHY circuits: design-dependent, < ~10ps



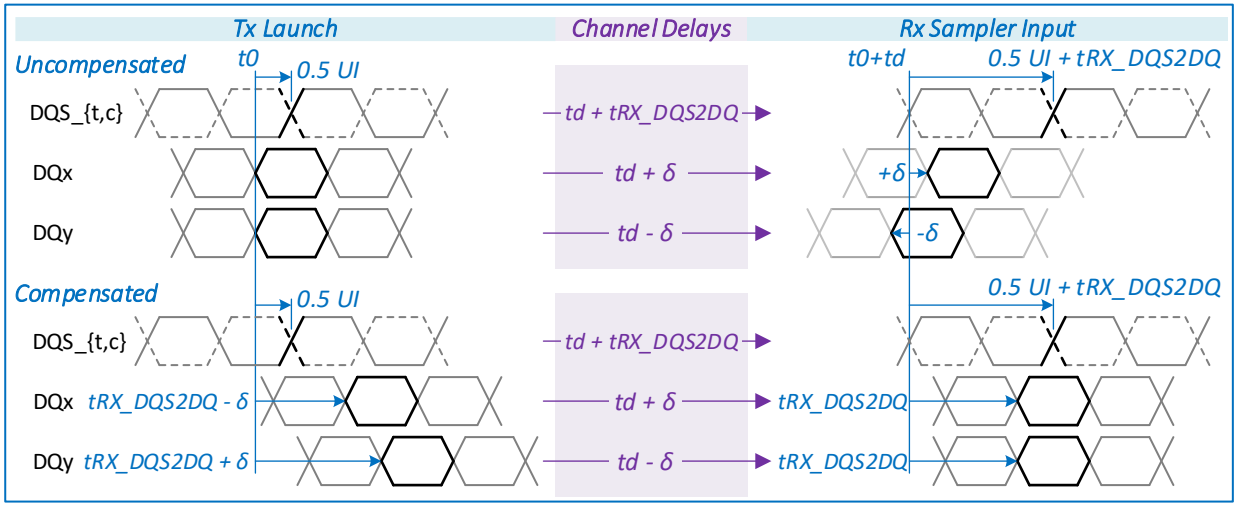


https://www.signalintegrityjournal.com/articles/4105-material-induced-skew-in-high-speed-multilayer-pcbs-influences-and-mitigation-strategies

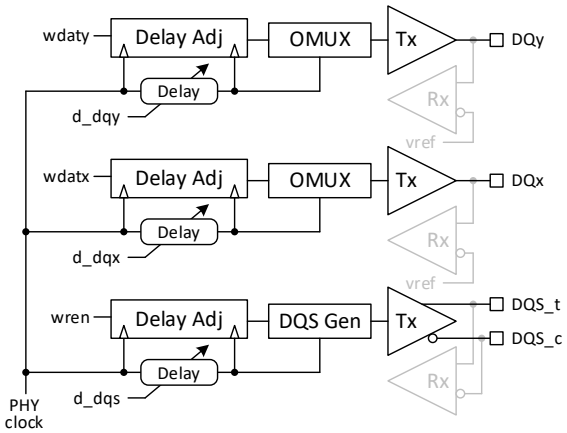


https://docs.altera.com/r/docs/772538/25.1.1/external-memory-interfaces-agilextm-7-m-series-fpga-ip-user-guide/example-of-a-ddr5-layout-on-an-altera-fpga-platform-board

Source-Sync Timing Alignment – Write Example



Example host PHY implementation

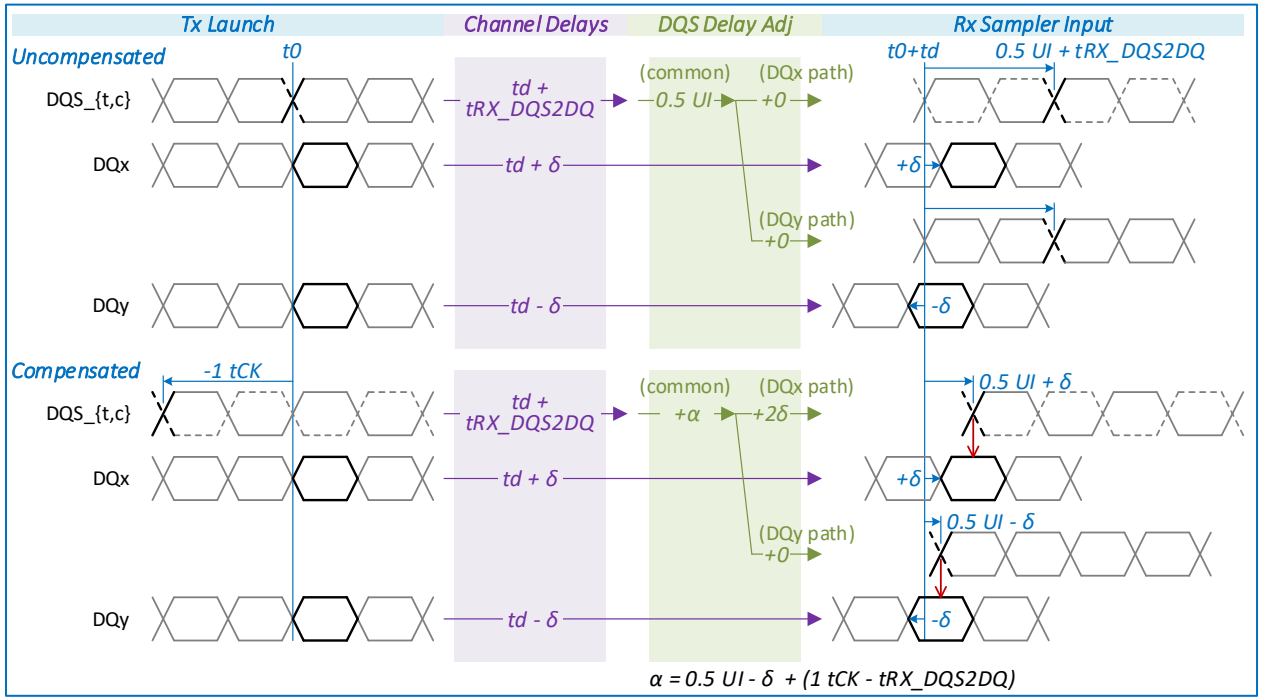


Transmit-side compensation of channel skews to center source-sync lanes on timing reference

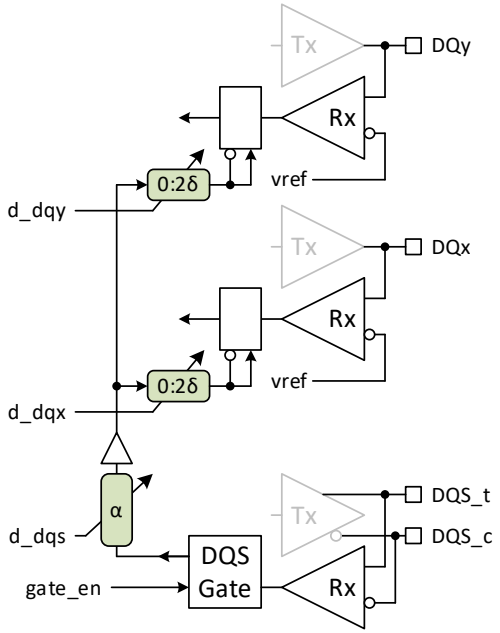
- CA/CS vs. CK
- DQ vs. DQS/WCK
- Typically applied at host (or in buffer chip), not in DRAM
- Optimized during write training (timing sweep)

Source-Sync Timing Alignment – Read Example

- Receive-side skew compensation used to center DQ lanes on timing reference
- Delay adjustments are applied in the host internal strobe path – both common and per-bit adjustments

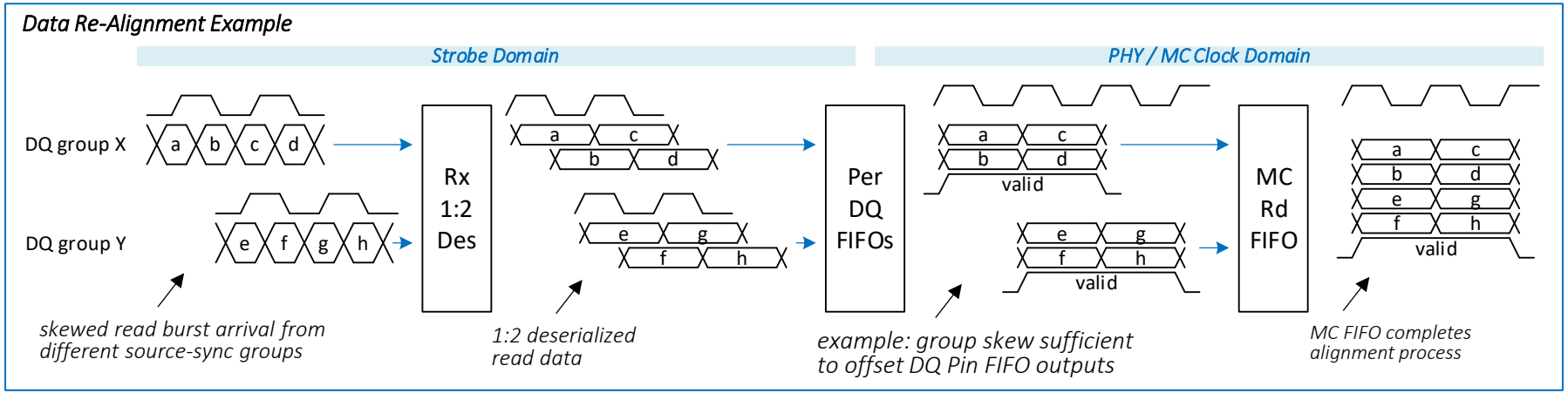


Example host PHY implementation



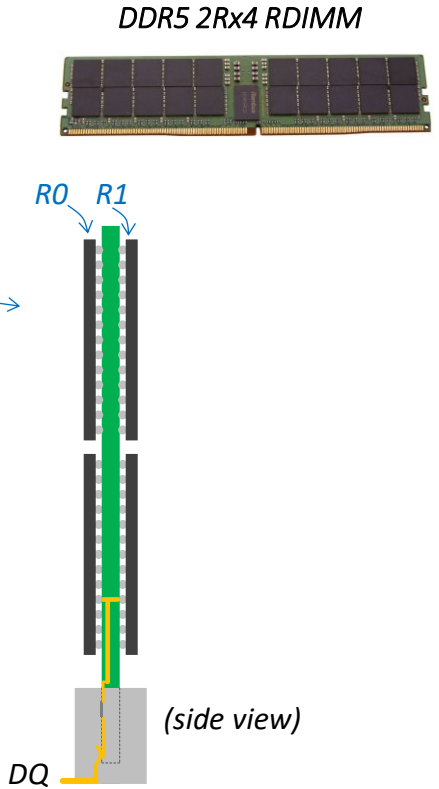
Host PHY/MC Read Data Alignment

- Read data across source-synchronous data groups is aligned in several steps:
  - Host PHY receives and partially deserializes data for each group (e.g. 1:2 w/ rising/falling strobe edges)
  - Host PHY retimes data for each group into a common MC/PHY interface clock domain
    - Per-DQ FIFO: CDC, absorbs untracked timing drift; group-dependent read gate training can produce cycle offsets
    - Additional deserialization can be done here
  - MC read data FIFO completes data alignment
    - Removes any n-cycle skews at the interface across source-sync data groups



Ranks and Associated Timing Sets

- A rank refers to a group of one or more DRAM devices operating in lockstep to form a complete data interface
- A “physical rank” is electrically observable on the bus
  - e.g. DDR5 2Rx4 RDIMM package ranks (front/back devices)
- Different ranks typically exhibit different transaction timing – routing and/or device timing mismatch
- Host controller PHY:
  - Measures and stores rank-dependent timing during training
  - Dynamically applies rank-dependent timing sets based on target rank of given read or write transaction

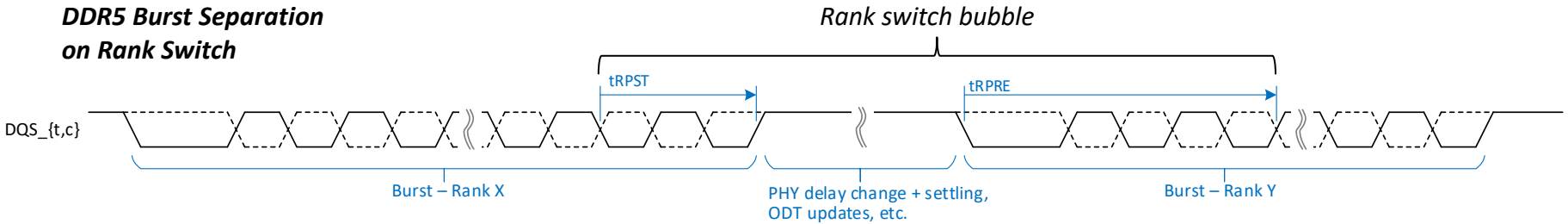


Rank Switching

- Several timing overheads associated with switching ranks
  - Load and settling times for change in active host PHY rank timing set
  - Strobe postamble of terminating burst, preamble of new burst
  - Rank-dependent write leveling / read gate training offsets
  - Timing precision of ODT control (host PHY and DRAM)

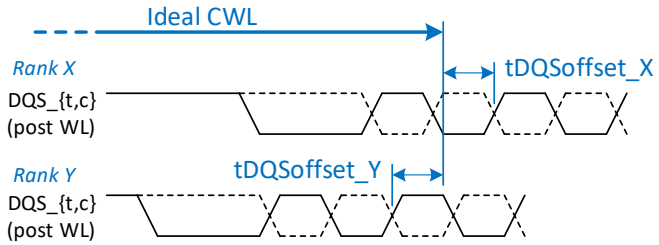
  } *These also affect bus turnaround*
- DRAM with logical ranks provide rank switch timing parameters
  - Examples: tCCD*_dlr (DDR5 3DS), tCCDR (HBM4)
- Command spacing must allow for required burst separation

DDR5 Burst Separation on Rank Switch

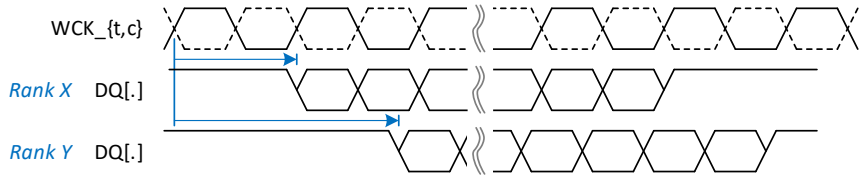


Some DRAM Characteristics Affecting Rank Switch

Mismatch in DRAM Internal Write Leveling Reference

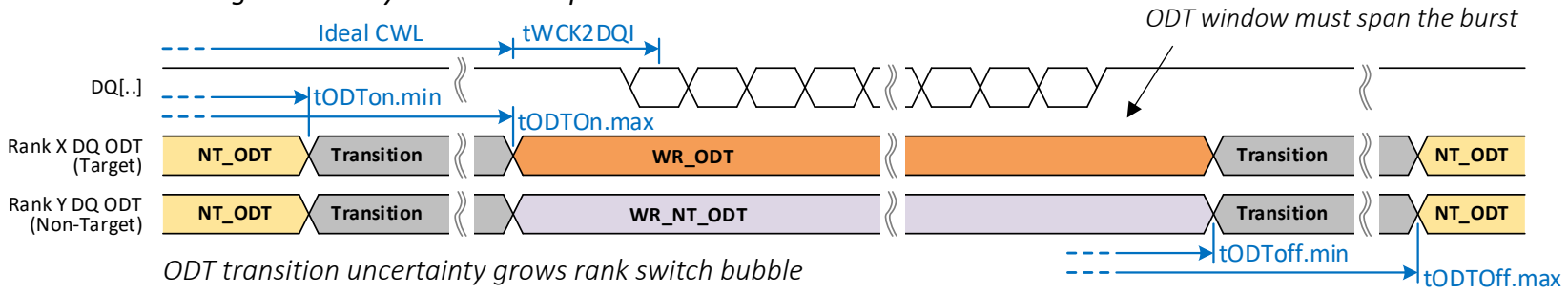


Internal WCK to DQ (and RDQS) Delay Variation



Grows burst timing variation – increases size of ODT windows (and ΔRdLev)

ODT Control Timing Uncertainty – Write Example

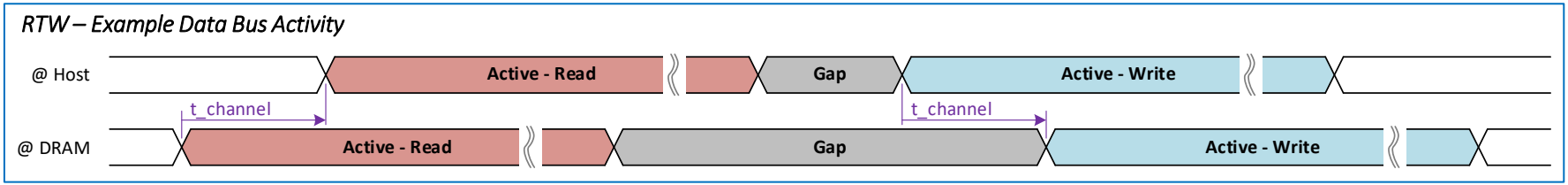


ODT transition uncertainty grows rank switch bubble

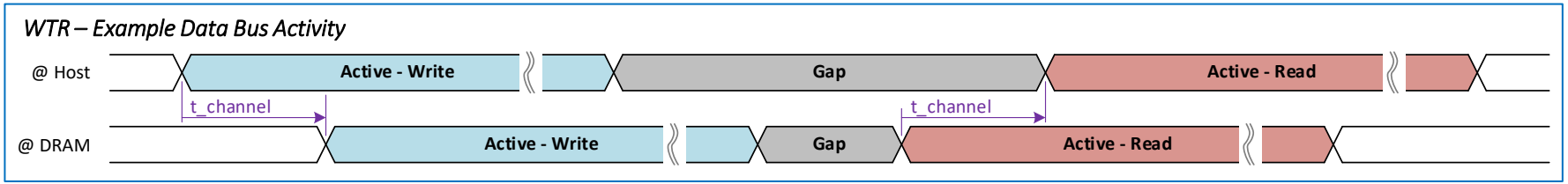
- Precision of burst placement and ODT control affects min burst and command separation

Routing Delays and Bus Turnarounds

- Read to Write
  - Read burst must traverse the channel interconnect before write burst can begin
  - Interconnect delay in both read and write directions expands the observed timing bubble at DRAM
  - The expanded gap at the DRAM translates to increased command spacing



- Write to Read
  - Write burst must traverse the channel interconnect before read burst can begin
  - In this case, interconnect delay expands the data bus utilization gap at host side
  - For same rank WTR, DRAM timing parameters can significantly impact the required gap – rank switch helps



Outline

- DRAM channel overview and examples
- Data transfers
  - Command to data burst synchronization
  - Source-sync timing alignment
  - Read data realignment
  - Ranks timing sets and switching
  - Bus turnaround
- Interconnect and signaling
- Capacity scaling
  - RDIMM example
- Bandwidth scaling at capacity
  - MRDIMM example

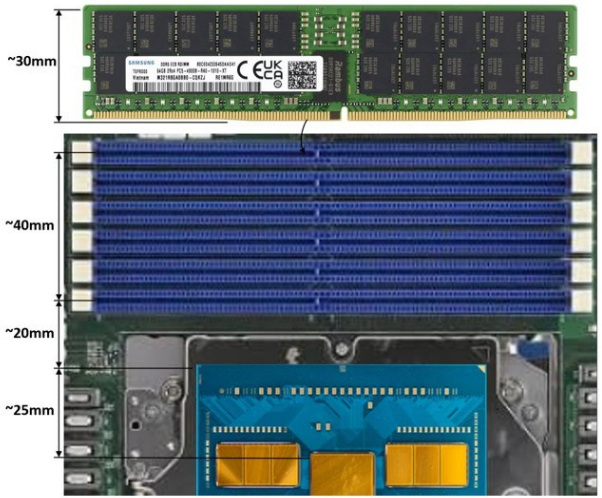
Real-World DRAM Interface Channels

Form factor drives interconnect technology selection, electrical characteristics, PHY requirements

Compute Server

- Organic FCBGA, FR4(+) PCB & module
- Module + connector; single/multi-drop
- Transmission line delay > t_bit

~100 mm reach

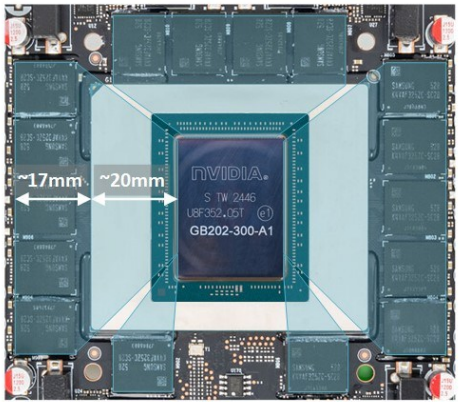


AMD Genoa (DDR5)

GPU Card

- Organic FCBGA, FR4(+) PCB
- Solder-down, point-to-point
- Transmission line delay > t_bit

~40 mm reach

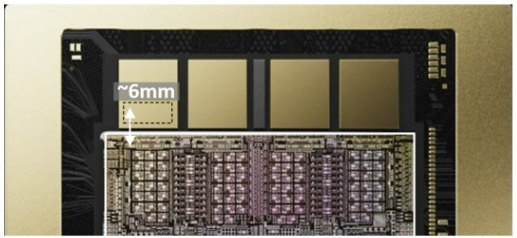


Nvidia RTX-5090 (GDDR7)

AI Engine

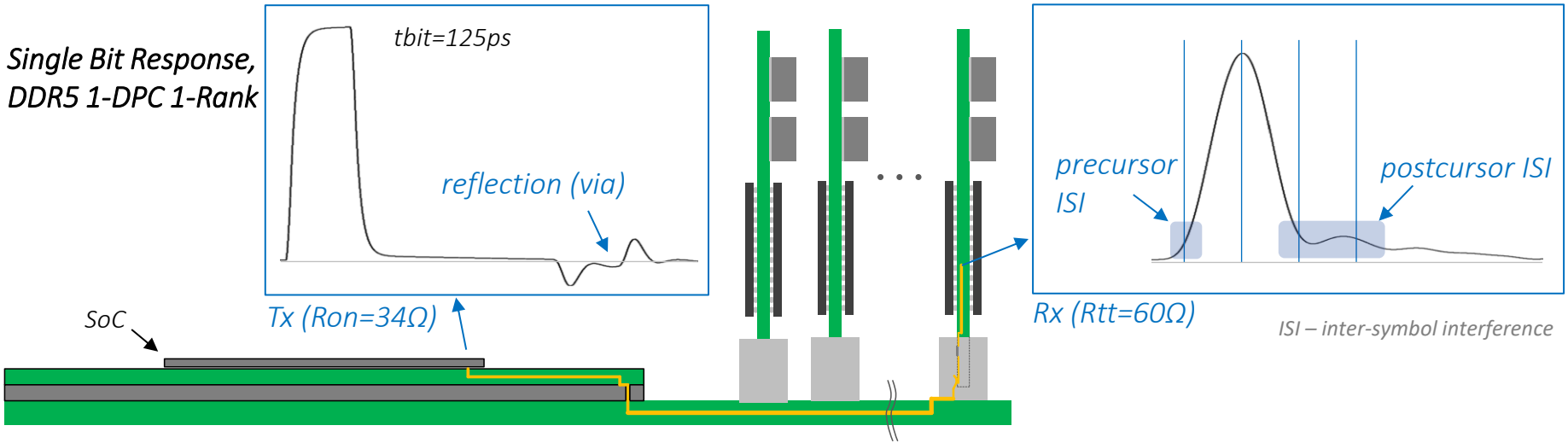
- Silicon Interposer
- Solder-down (µbump), point-to-point
- Transmission line delay ≲ t_bit

<10 mm reach



Nvidia B300 (HBM3e)

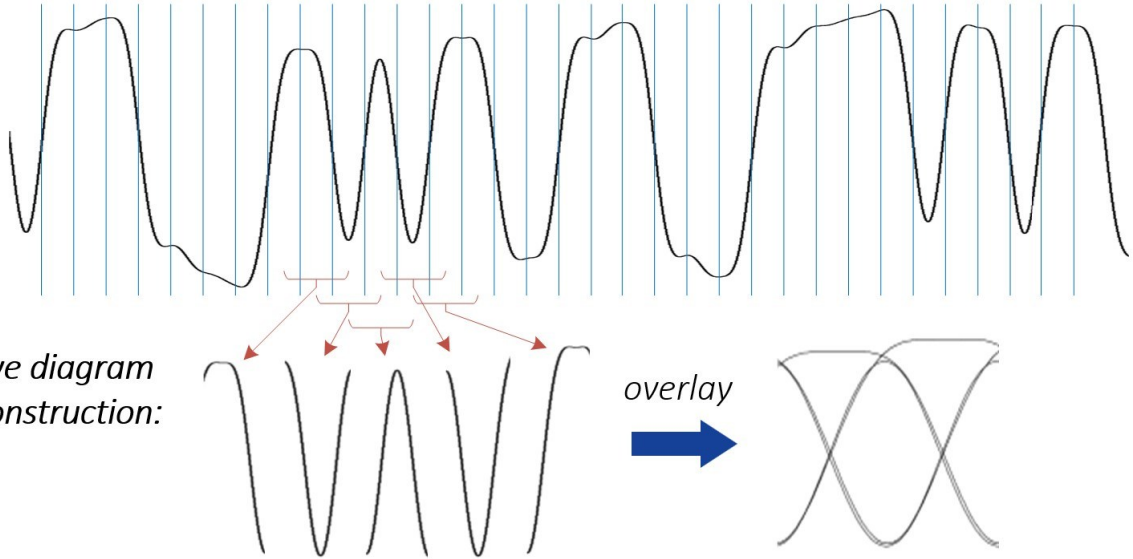
Channel Signal Impairments



- The conductors and dielectrics associated with FC-BGA/LGA packaging and motherboard/module printed wiring boards support lossy wave propagation
- The SBR above demonstrates the effects of channel loss, dispersion, and reflections
- Electromagnetic coupling among propagating signals generates crosstalk
- Host and DRAM IO implement various features to contend with these effects
  - e.g. controlled-impedance drivers and ODT, driver slew rate control, Tx and Rx equalization

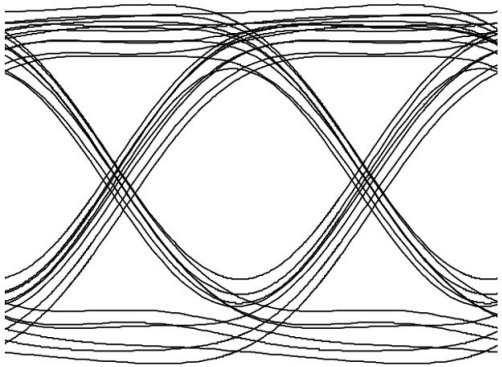
Example DQ Signal With Channel Effects

- Same DDR5 1Rx4 channel from SBR example, 8 Gbps
- Section of write DQ waveform at DRAM Rx pad (32 UI)



Five 2-UI segments centered on successive UI

Example eye diagram of 32 UI



Eye opening shrinks as more bits are measured – worst-case patterns are exercised, random voltage and timing noise accumulates

Equalization Techniques in DRAM and Host PHYs

*Rx: Continuous Time Linear Equalization (CTLE) **

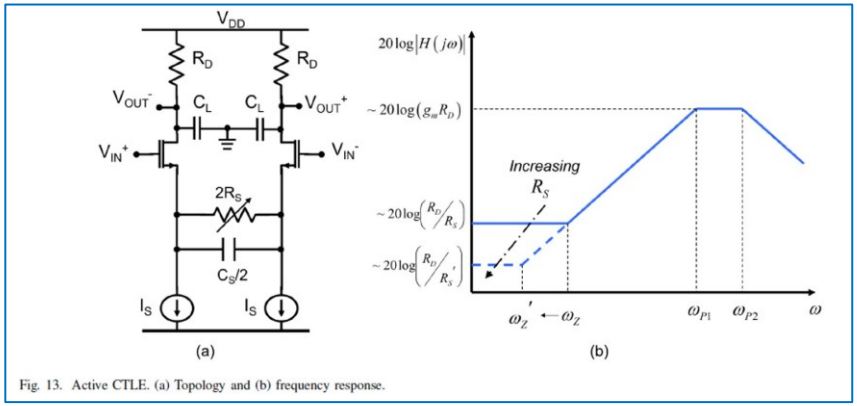
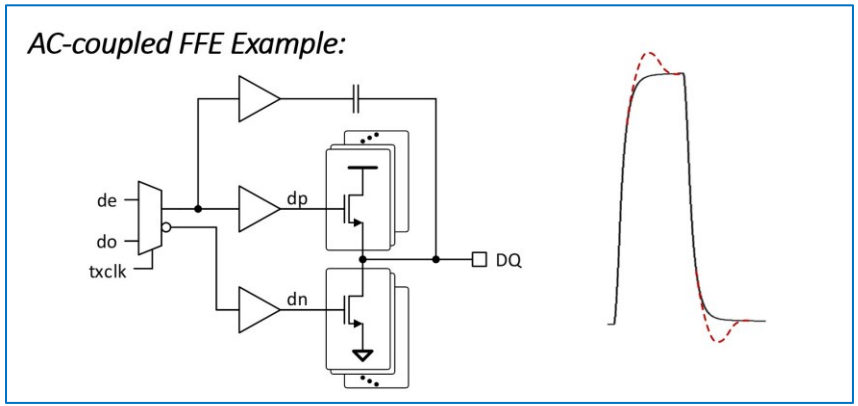


Fig. 13. Active CTLE. (a) Topology and (b) frequency response.

Tx: Feed Forward Equalization



*Rx: Decision Feedback Equalization (DFE) **

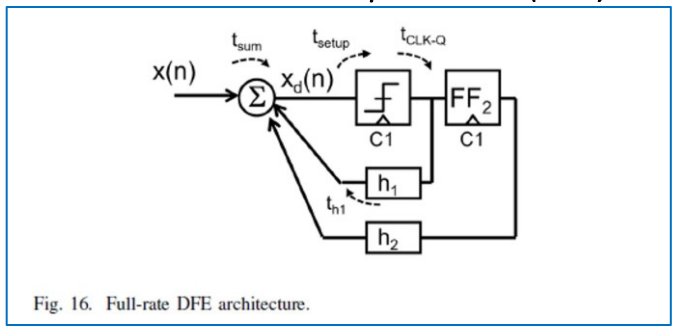
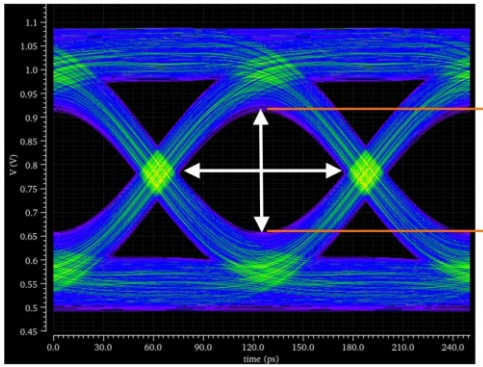


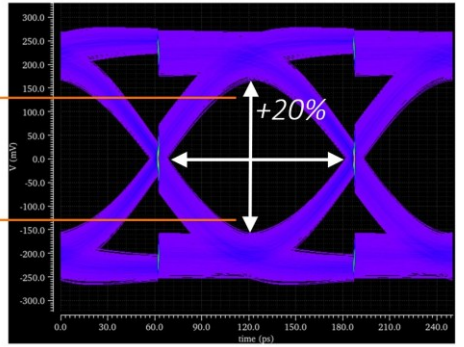
Fig. 16. Full-rate DFE architecture.

example

No Equalization



2-Tap DFE (Unity Gain Rx)



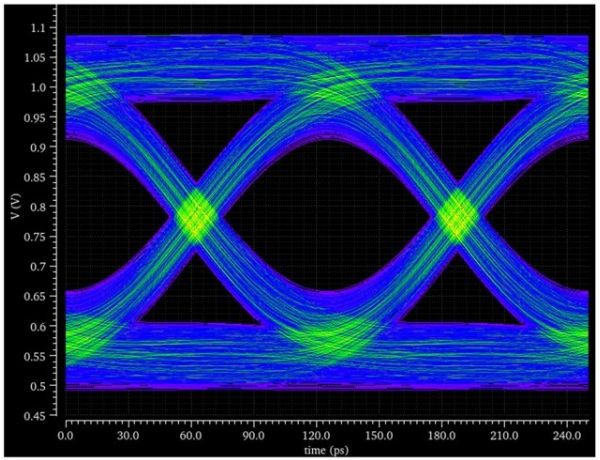
* Figures 13 and 16 from: Carusone, et al., "Modern Wireline Transceivers," in *IEEE Journal of Solid State Circuits*, vol. 61, no.2, pp. 395-422, Feb. 2026. [Open Access on IEEE Xplore]

Effect of Ranks and Non-Target Termination

DDR5 1Rx4 RDIMM – 8 Gbps Write

Ron = 34 Ω

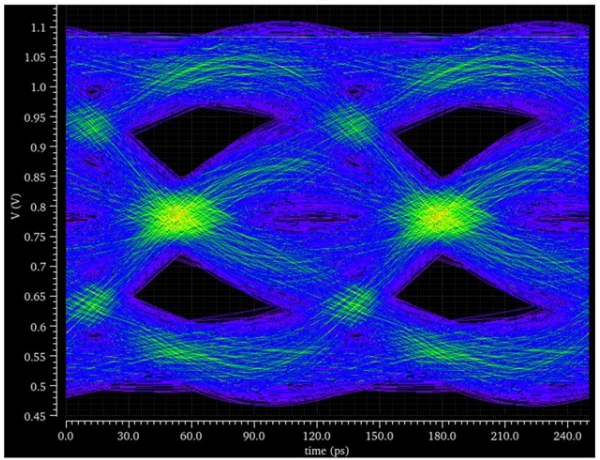
RTT_WR = 60 Ω



DDR5 2Rx4 RDIMM – 8 Gbps Write

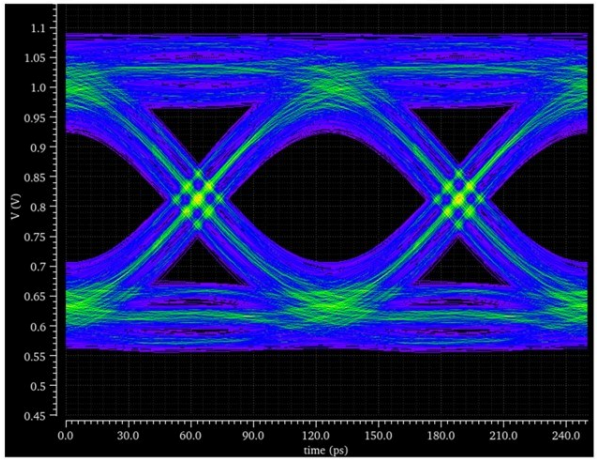
Ron = 34 Ω

RTT_WR = 60 Ω, RTT_NOM_WR = HiZ



Ron = 34 Ω

RTT_WR = 120 Ω, RTT_NOM_WR = 80 Ω



Note: simple channel model – no crosstalk or random voltage and timing noise included

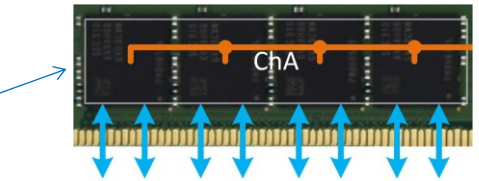
- Channels with multiple physical ranks require attention to target and non-target termination impedances

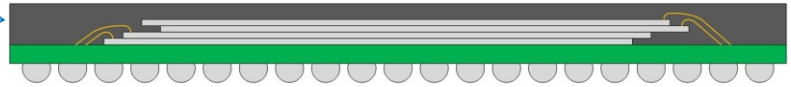
Outline

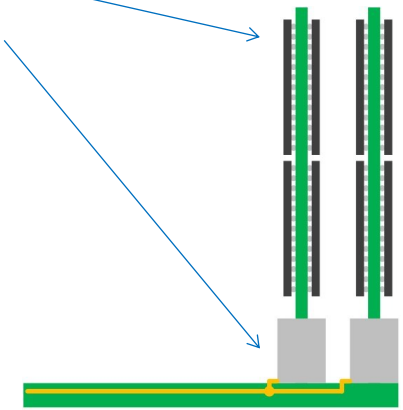
- DRAM channel examples
- Data transfers
  - Command to data burst synchronization
  - Source-sync timing alignment
  - Read data realignment
  - Ranks timing sets and switching
  - Bus turnaround
- Interconnect and signaling
- Capacity scaling
  - RDIMM example
- Bandwidth scaling at capacity
  - MRDIMM example
- Final Words

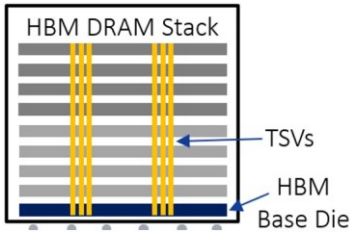
Capacity Scaling – Multi-Die Channels and Ranks

- Extending channel capacity beyond single die density
  - Multi-die ranks – spread the data across multiple die accessed in parallel
    - Multiple packaged devices per rank (e.g. DDR DIMM)
    - Multiple co-packaged die (e.g. LPDDR5 byte mode)
  - Multiple ranks per channel – access one of N ranks at a time
    - Internal device ranks: device stacking with wire bonding (e.g. LPDDR) or TSV (e.g. DDR 3DS, HBM)
    - Multi-rank modules: some combination of package ranks and possibly internal device ranks (e.g. 4R 3DS-2H RDIMM)
    - Multiple modules per channel: multiple modules sharing the same interface bus (e.g. 2-DPC)
- These topologies lead to increases in command/address and data bus loading *somewhere*
  - Unless the DRAM internally buffers the load, it is exposed to the host-DRAM interface
  - Interface loading can limit the practically achievable bandwidth (e.g. 2-DPC DDR5 RDIMM, multi-rank LPDDR5X SOCAMM)

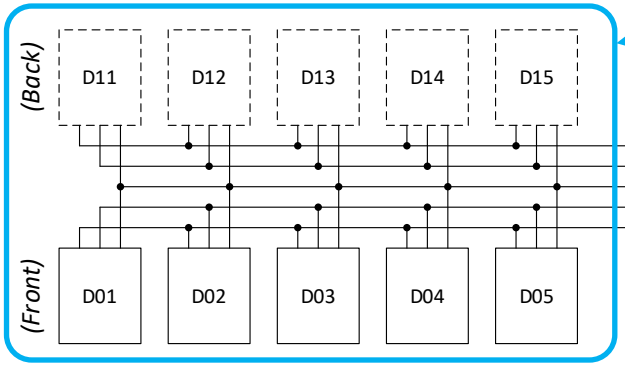


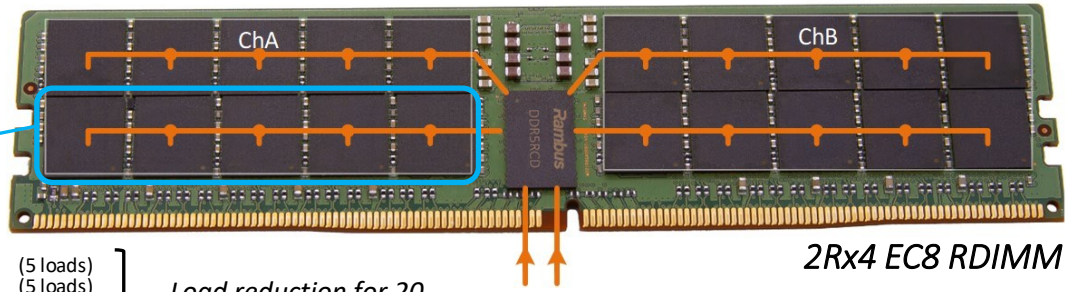






DDR5 RCD Overview



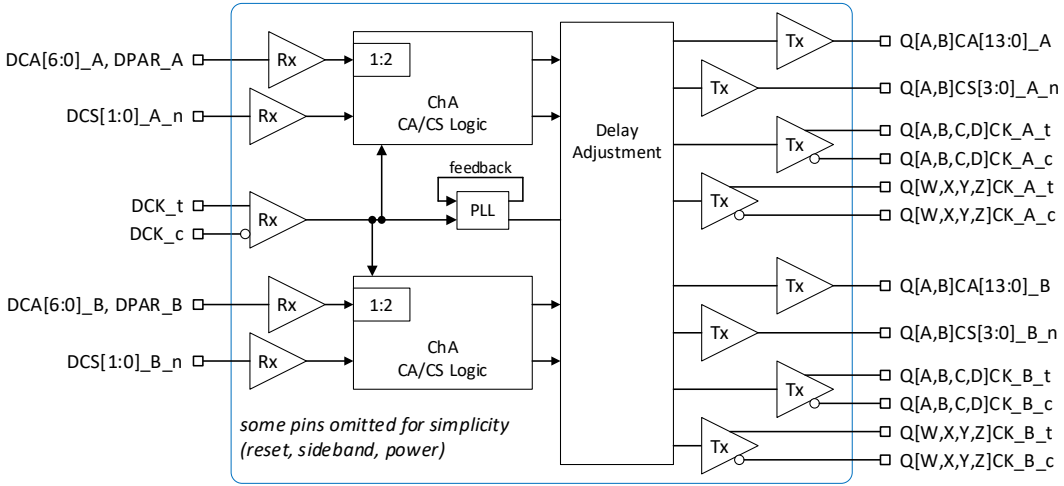


2Rx4 EC8 RDIMM

Load reduction for 20 DRAM per channel

Registering Clock Driver

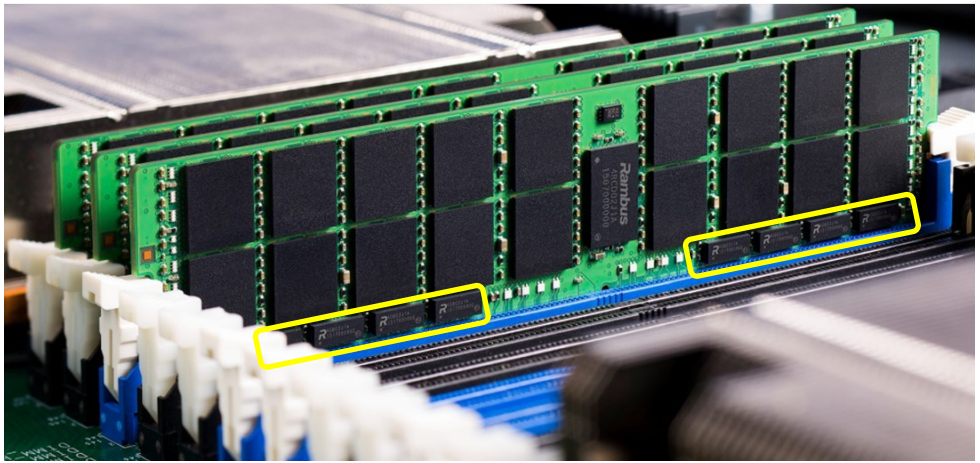
- Services two channels (A,B)
- Retimes CA/CS; implements 1:2 CA deserialization (host interface **pin reduction**)
- Regenerates low-jitter CK
- Multiple signal copies for **load reduction**:
  - two CA/CS copies per channel
  - up to 8 CK copies per channel
- Fixed baseline analog propagation delay; phase-stable host and DRAM interface timing
- DFE EQ on host CA/CS inputs
- Delay adjustability on CA/CS/CK outputs



Bandwidth Scaling at High Capacity

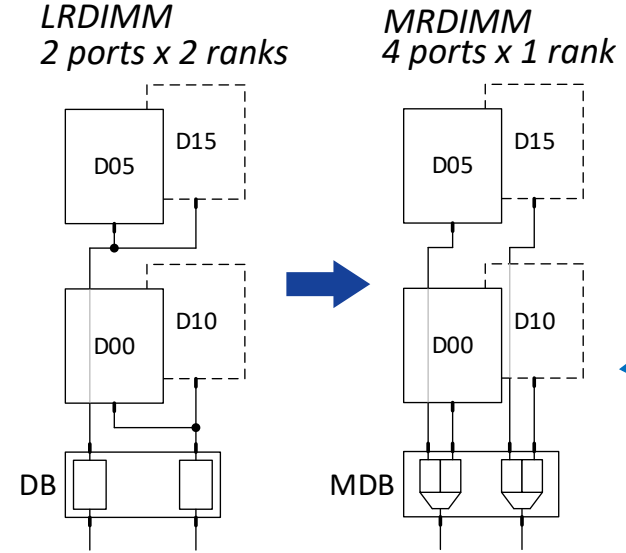
- Multiple DQ bus loads limit the maximum achievable data rate
  - Multiple physical ranks per module, multiple modules per channel
- Data buffer (DB) logic devices reduce host interface bus loading – enable bandwidth at high capacity
  - Retime DQ and re-generate DQS
  - Locate host interface at module edge

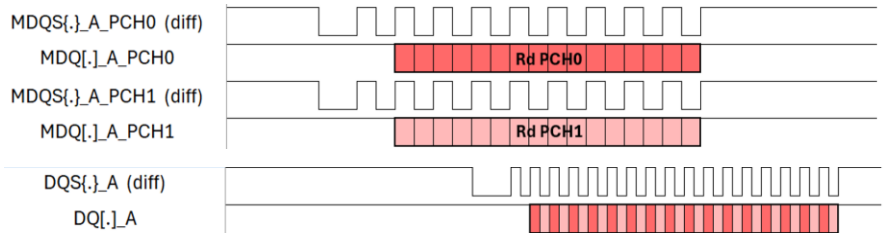
Example: DDR4 3-DPC LRDIMM – load isolation with data buffer

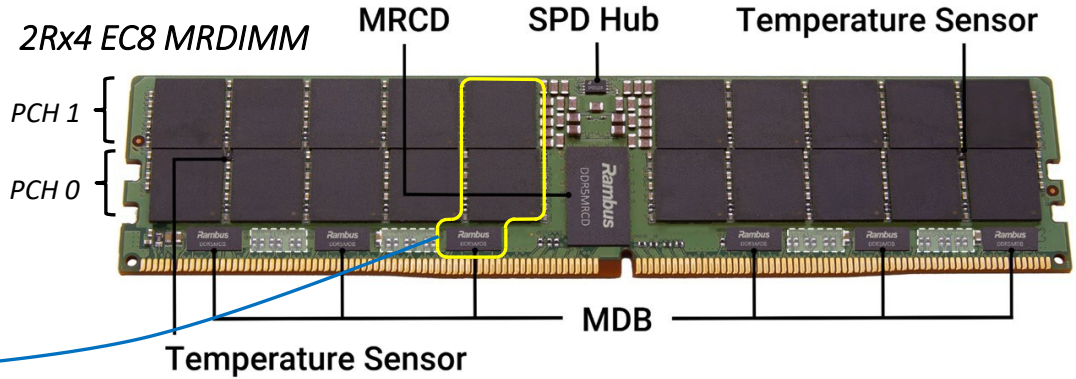


- DDR5: targeting ≤ 2 package ranks and ≤ 2-DPC also mitigates DQ bus loading
  - Bandwidth degradation still exists for 2-DPC RDIMM (e.g. 5200 vs. 6400 MT/s)
  - Disparity increases at higher rates → 1-DPC used to maximize bandwidth
- Data buffer used in DDR5 to enable even higher bandwidth modules → MRDIMM

Multiplexed Rank DIMM (MRDIMM) – Overview





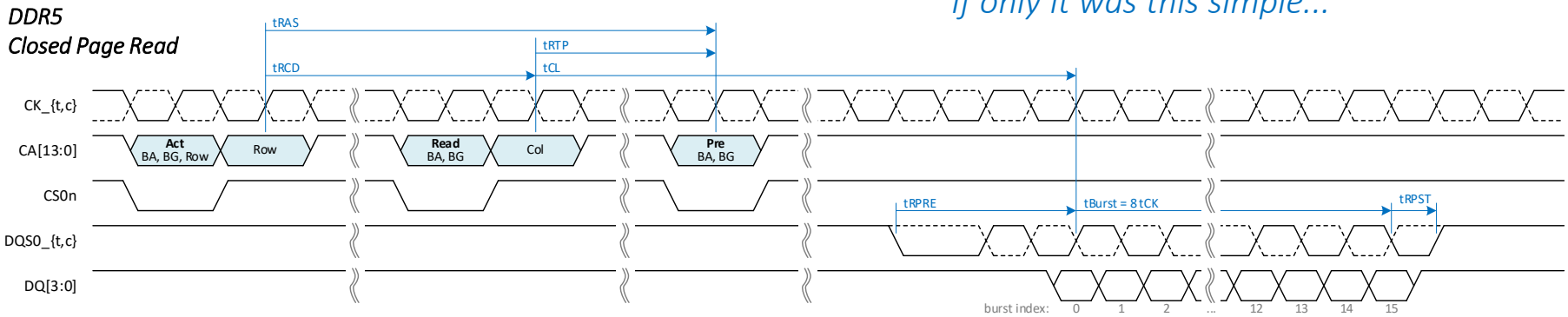


- Each x40 module channel operates as two interleaved PCH
- Two DRAM ranks are transformed into two one-rank PCHs
- MRCD: interleaved CA/CS from host; deinterleaved to PCH 0/1
- MDB: PCH bit-interleaved data to/from host; 2x the DRAM interface rate → doubles the module bandwidth w/ no changes to DRAM
- Gen 1 – 8.8 GT/s, Gen2 – 12.8 GT/s, Gen3 in progress
- 2-rank per PCH specified in 1U 3DS or 2U SDP configurations

PCH – pseudo-channel
1U/2U – 1 or 2 rack units (1U = 1.75 in)
SDP – single-die package

Why Is this Complicated?

If only it was this simple...



- Devices are non-ideal
  - Silicon process variation
  - Circuit voltage and temperature sensitivity
- The physical channel is non-ideal
  - Material properties and geometry limits
  - Signal transmission: delay, loss & dispersion
  - Other signal and power integrity impairments (reflections, crosstalk, SSO/PSIJ)
- Channel composition increases complexity
  - Multi-device ranks and multi-rank channels
  - Aggregated sets of source-synchronous links
  - Multiple module types
- *The DRAM spec is a great place to start but doesn't tell you everything.*
- *The host PHY (and buffer chips) must compensate system non-idealities to meet spec requirements at each DRAM device.*

Future Memory Solutions:
HBM, MRDIMM, CAMM, CXL & PIM/PNM

Taeksang Song, Ph.D.
Corporate VP of SoC & DRAM Solution Engineering,
Samsung Electronics

Outline

- Introduction
- Next Generation DRAM Module
  - High-Bandwidth Memory (HBM)
  - Multiplexed Rand DIMM (MRDIMM)
  - Small-Outline Compression-Attached Memory Module (SOCAMM)
  - CXL Memory Module (CMM)
- Compute-Capable Memory Solutions
  - Processing-In-Memory (PIM)
  - Processing-Near-Memory (PNM)
- Summary

Next-Gen Memory Solutions in Systems

Server

High Speed and Density
(Cloud/HPC)

CMM-D
* CXL Memory Module-DRAM

cHBM
* Custom High Bandwidth Memory

zHBM
* 3D-stacked High Bandwidth Memory

MRDIMM
* Multiplexed Rank DIMM

Client

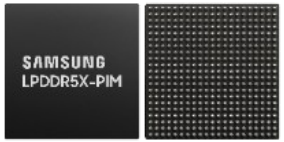
(Laptop, AI PC, High-end WS)

SOCAMM2
* Small Outline CAMM2

LPCAMM2
* LPDDR Compression Attached Memory Module

Mobile

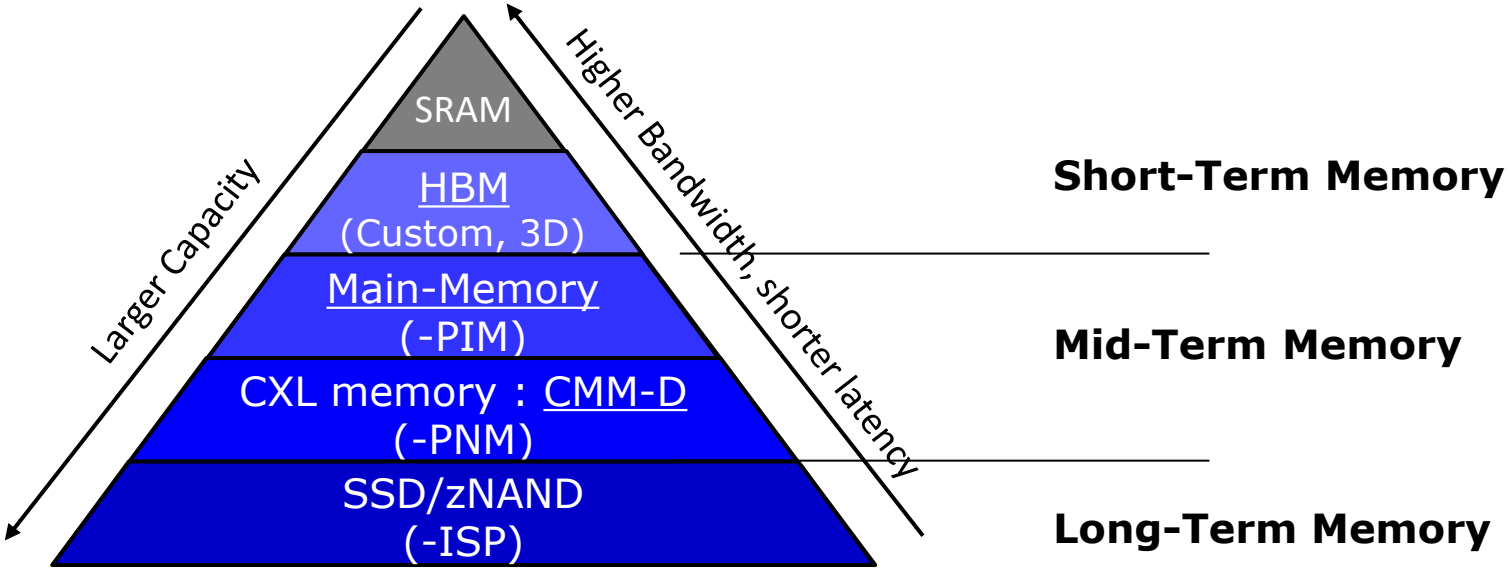
Low Power
(On-device AI)



LPDDR-PIM
*LPDDR Processing In Memory

Next-Gen Memory Solutions

- Memory hierarchy is getting more advanced and efficient

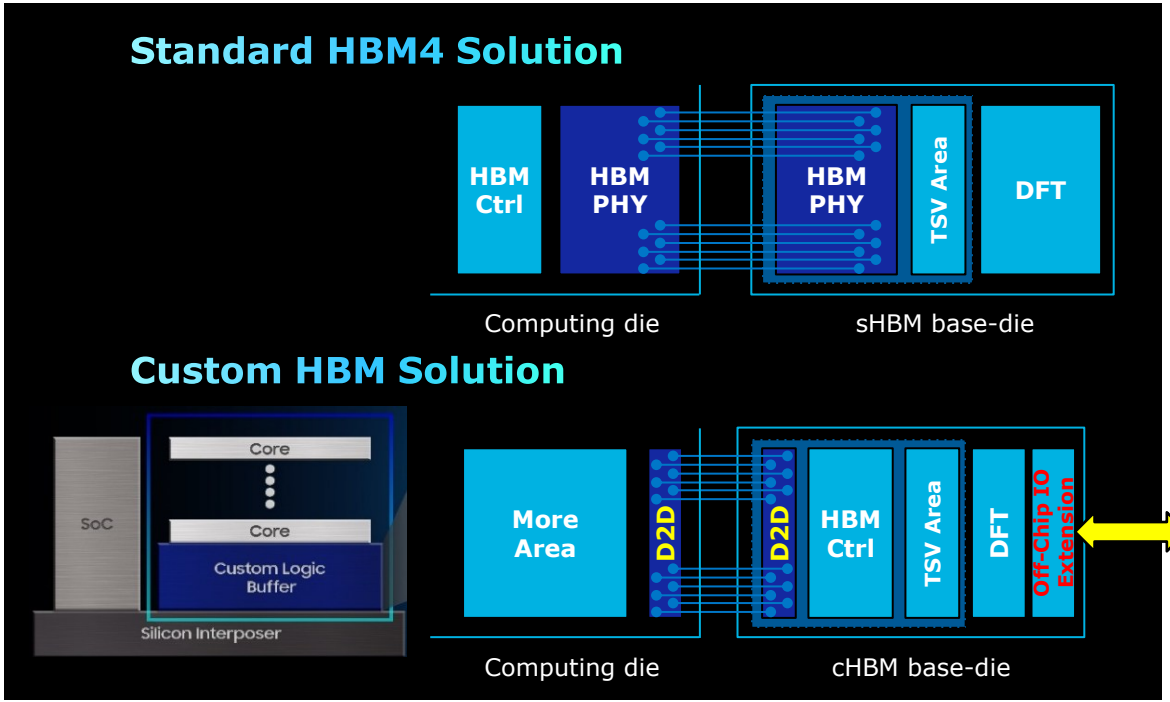


Key Technology for HBM

- SiP (System-in-Package) Structure using HBM
  - The first heterogeneous integrated DRAM: Buffer die + Multiple Core dies
  - Key technologies: TSV, u-bump, Si-interposer, 2.5D CoWoS, Power integrity, Thermal management

Custom HBM

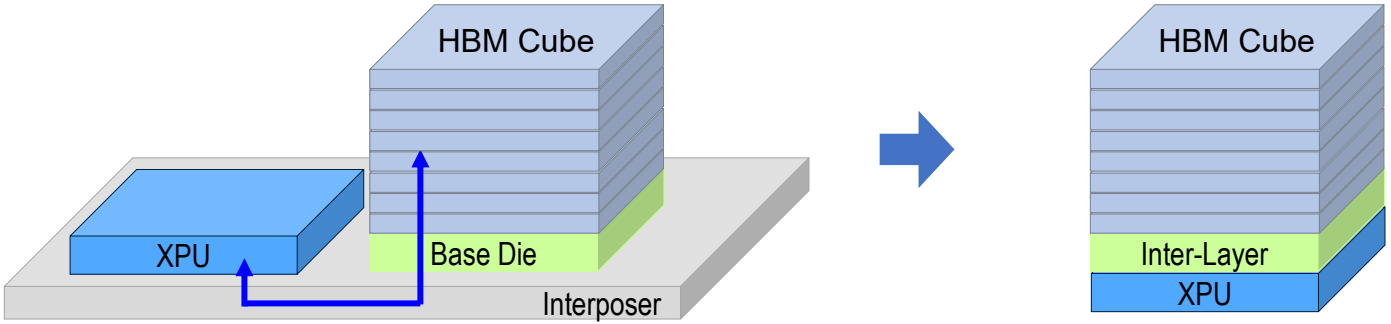
- Custom buffer die includes functions optimized for customers' individualized needs



- Custom logic
- Coherent interconnect to base-die
- External off-chip IO & DRAM extension

3D Stacked zHBM – Concept

- Extremely low-power and high-BW solution
  - Increase # of IOs by removing shoreline limitation
  - Improve power efficiency: distributed IO, short data-path, 2.5D PHY & SerDes



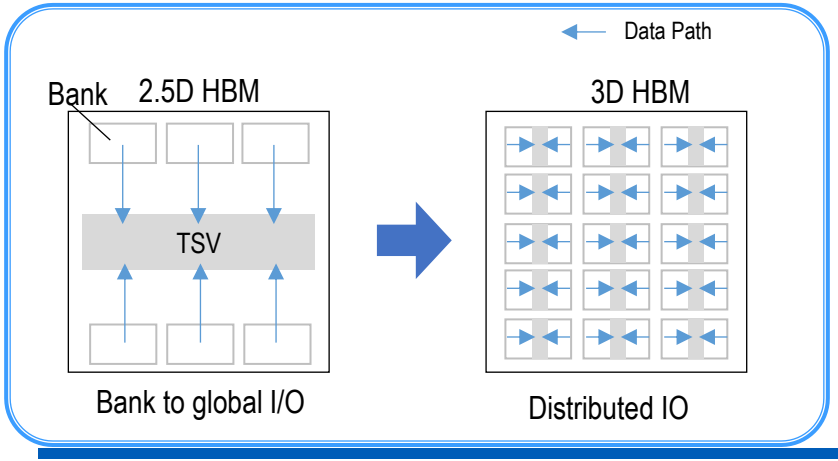
<2.5D Integration>

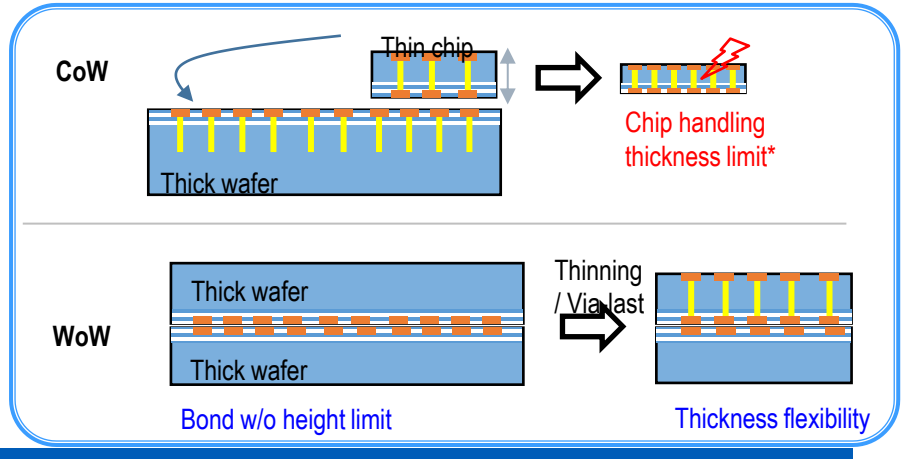
<3D Integration>

Interposer

3D Stacked zHBM – Key Enablers

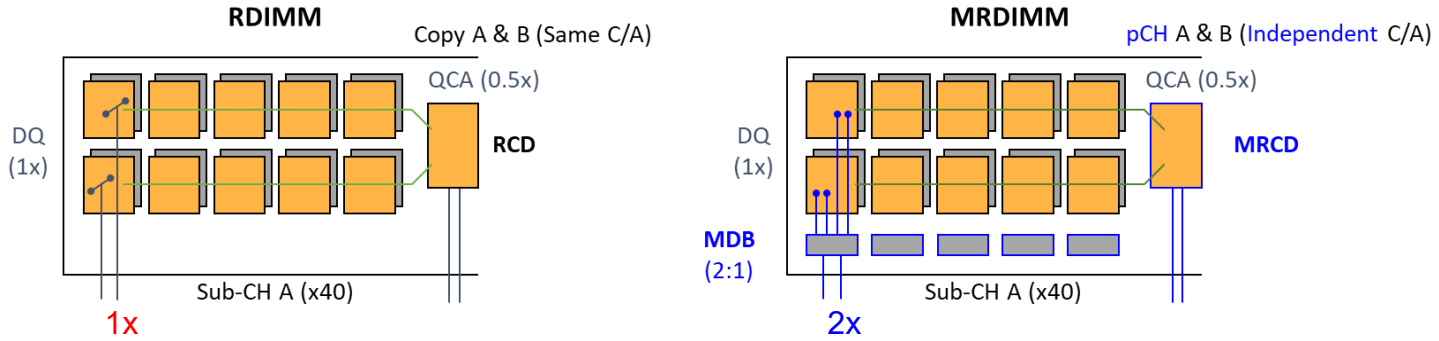
- Distributed IO & WoW Bonding
  - Wafer/die thickness determines minimum TSV pitch (due to TSV aspect ratio)
  - By using WoW bonding instead of CoW pick-and-place, TSV density can be >10x
- Challenges
  - 10x TSV density(=10x BW) increases power significantly → Memory arch.-innovation is essential
  - Data-placement to minimize xPU's data movement power
  - Addressing WoW yield loss through repair and redundant routing paths





DDR5 MRDIMM (Multiplexed Rank DIMM)

- Higher-BW memory module with commodity DRAMs
  - MDB (MRDIMM Data Buffer) communicates with the CPU at double speed by using a 2:1 MUX to transfer 1x DRAM data.



Item		RDIMM	MRDIMM
Host-side (Frontside)	Channel	Two x40 sub-channels	Two x40 sub-channels
	Data Rate	1x (same w/ DRAM)	**2x** of DRAM
DRAM-side (Backside)	Channel	Conventional	**Two Pseudo** channels
	Data Rate	1x	1x

MRDIMM Pros & Cons

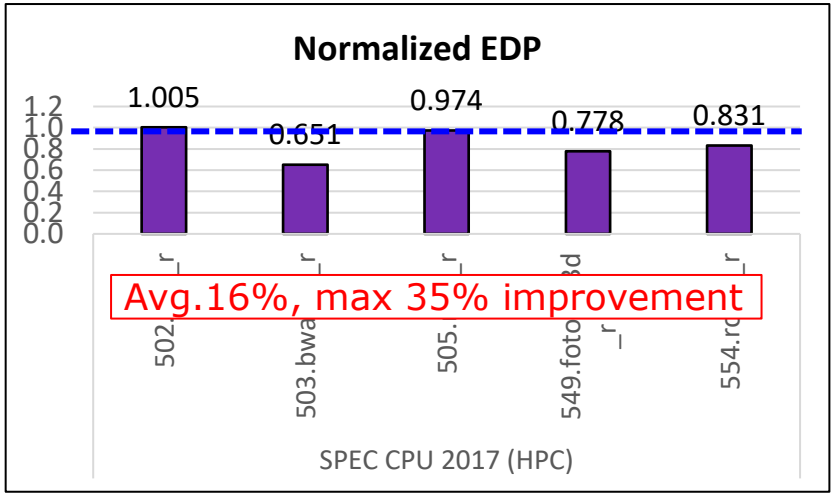
- Computation = Digesting Data
  - Higher core count CPU requires higher-BW and higher-capacity memory
    - Need to maintain constant BW/core & GB/core
    - AMD CPU core count: Rome 64 → Genoa 96 → Turin 192 → Venice 256 → …
  - 3D-Stacked RDIMM can provide 2x or 4x capacity, but no BW scaling
    - Should enhance both BW and capacity

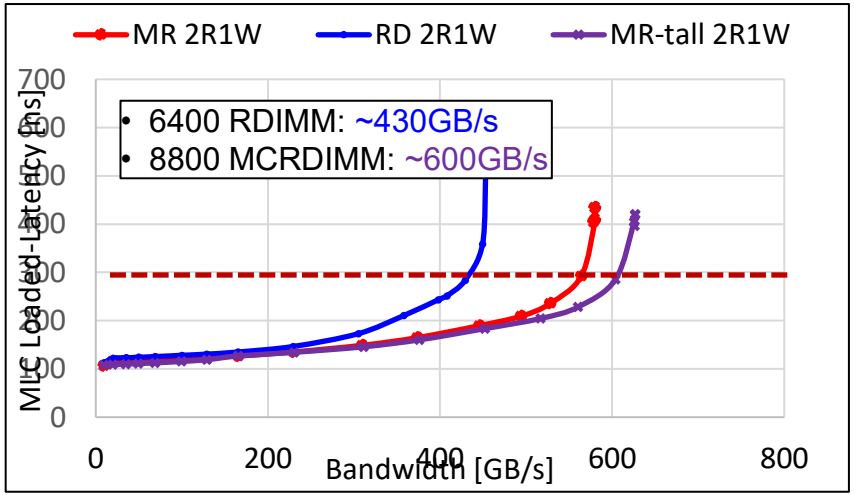
- MRDIMM is the solution for memory-intensive workloads
  - BW: 8.8Gbps (Gen1) → 12.8Gbps (Gen2) >16Gbps (Gen3)
  - Capacity: 2U Tall-DIMM has total 80-DRAMs (2x capacity)

- Concerns
  - Higher power consumption due to additional components (MDBs) + 2x DRAM operation
  - Higher cost than RDIMM

MRDIMM Performance Benefit

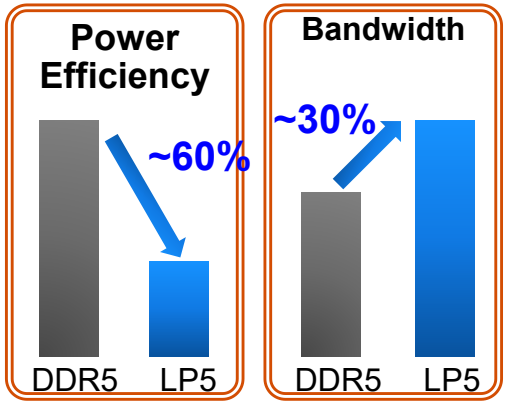
- System-level (CPU+MRDIMM) EDP (Energy-Delay-Product) BM
  - SPEC BM: MCRDIMM 8800 vs. RDIMM 6400 with Intel CPU
  - Despite higher power consumption of MRDIMM, shorter app runtime leads to overall EDP improvement
- Loaded Latency
  - 4-rank tall-MRDIMM can provide > 600GB/s (from 12-channel config) under 300ns latency

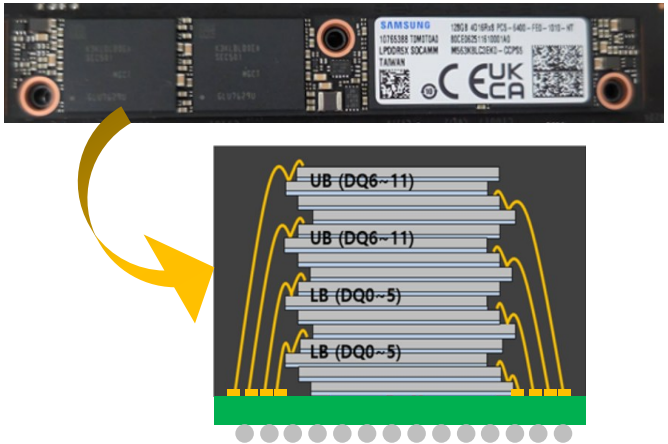




Why LPDDR-Based CAMM Solution

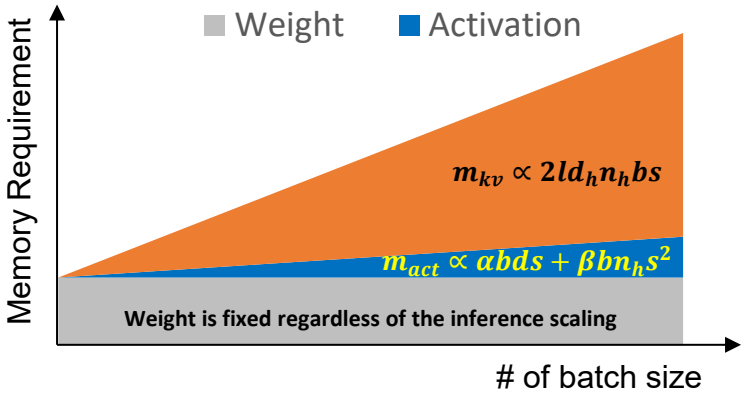
- Compression-Attached Memory Module with LPDDR DRAM for DC server
  - Lower power: No extra ECC die, no internal RMW operation, lower voltage IO, …
  - Higher capacity: Multi-die PKG without TSV process
  - Easy to build liquid cooling server
- Concerns
  - Reliability: no ECC die/package unlike DDR5 RDIMM
  - Per-pin speed degradation under heavy-load configuration

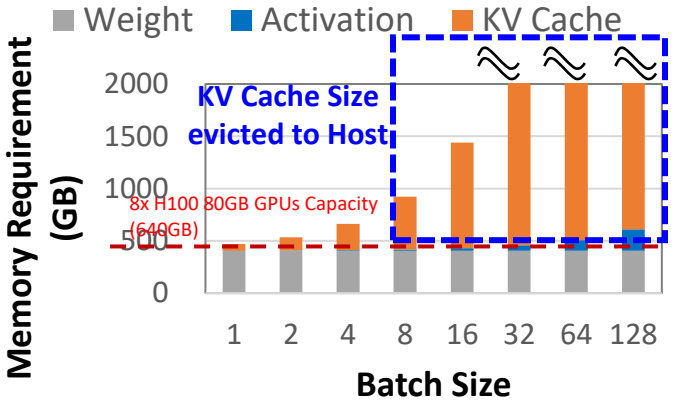




Memory Bottlenecks for AI Applications

- Example: LLM KV cache size
  - Weight Fixed, but KV cache is proportional to batch-size and context length and needed to be stored
  - The size of KV-cache with large batch can surpass the memory capacity of GPU devices





Scaling Memory Capacity and Bandwidth with CXL

- Memory capacity & BW expansion through serial/fabric-interconnect protocol
  - High-capacity: Multi-die stack PKG, Over-commitment (compression)
  - Reliability: Predictive Failure Analysis (PFA), Memory sparing/offlining
  - Low power solution: LPDDR-based media w/ RAIDDR ECC
  - High BW: PCIe Gen7
- Challenges: SW ecosystem to support overcommitment, sharing & Pooling

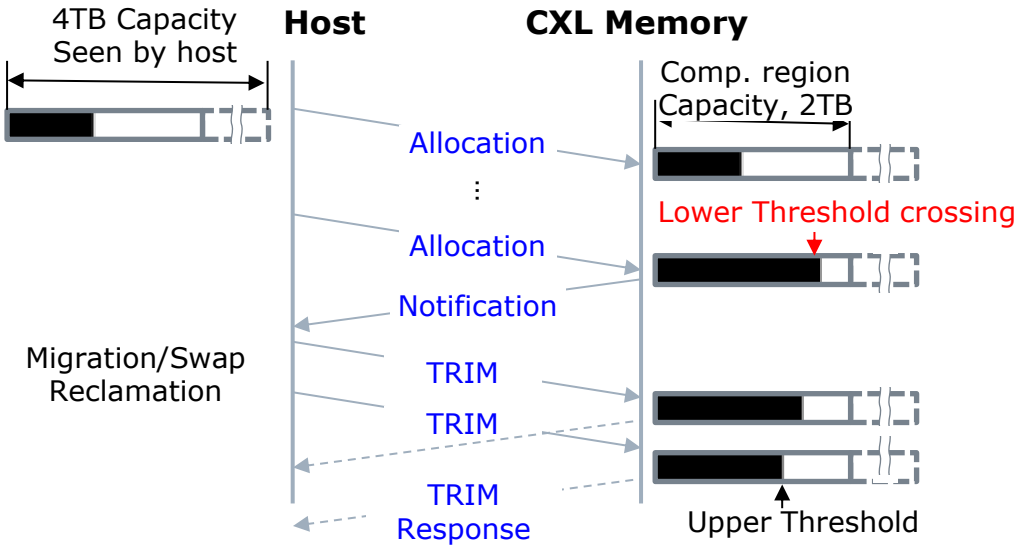
CMM-D (DRAM Expansion)

SW Requirements for Compression

- Challenges
  - To avoid page poisoning caused by device out-of-memory, the host must proactively monitor device capacity via host-device handshaking and take appropriate actions.

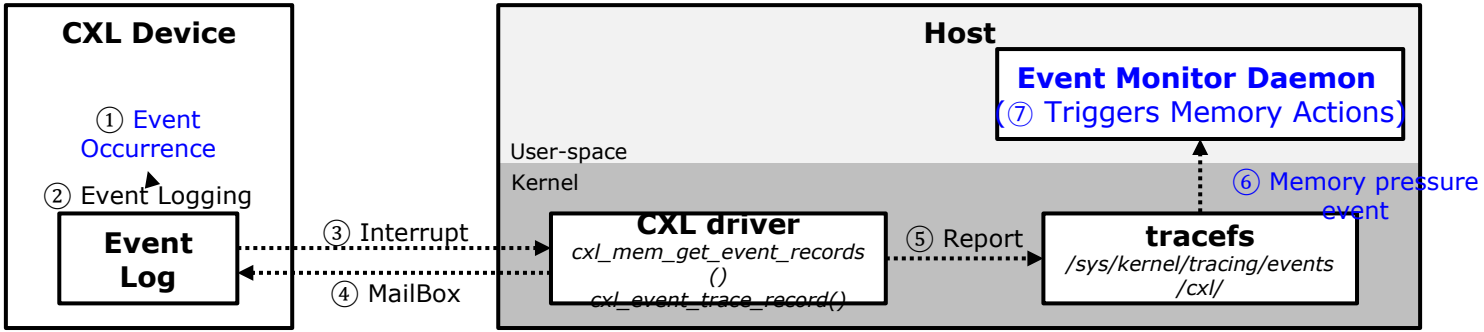
- Handshaking Interface
  - Memory pressure notification
  - Setting the pressure-level
  - TRIM command

- Host's actions to secure free memory space
  - Mitigation: reclaim device memory
  - Prevention: take proactive measures to avoid device OOM



Host-Device Memory Pressure Handling

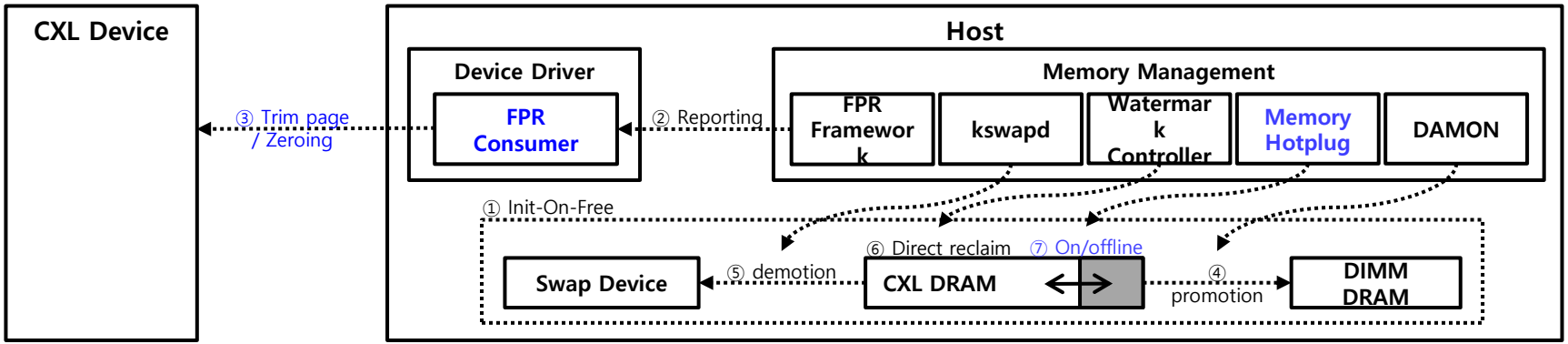
- **CXL Device operation flow**
  - ① CXL Device recognizes capacity threshold crossing, i.e., the free space is lower than user-defined threshold
  - ② Records the event and then sets Event Status register (Vendor Specific Event or Memory Module Event)
  - ③ Trigger an Interrupt to the host
- **Host operation flow**
  - ④ Read the Event Record from CXL device by sending the Get Event Records command via mailbox
  - ⑤ CXL Device Driver logs the event to tracefs1) (cf. /sys/kernel/tracing)
  - ⑥ Tracefs event monitor daemon catches the event, then ⑦ triggers memory management actions



1) Tracefs filesystem: the filesystem in the Linux kernel that provides user-space interface to kernel's event trace (v4.1~)

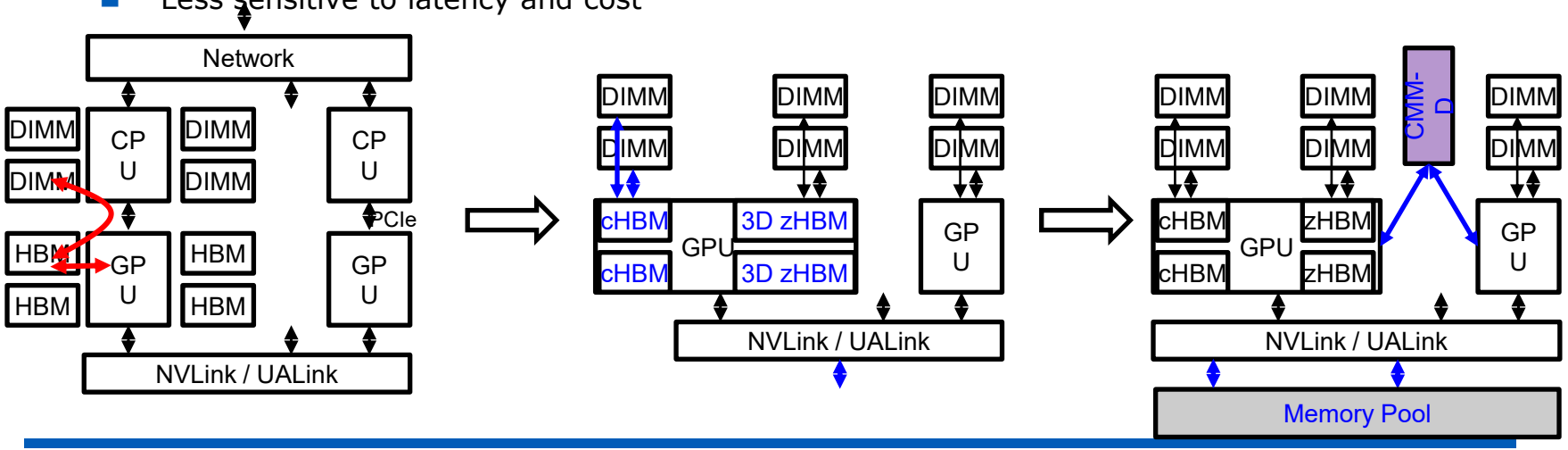
Host-side Memory Pressure Handling

- The monitor daemon mitigates pressure and prevents a OOM through following mechanisms.
- Mitigation for pressure on the CXL compressed region
  - Linux kernel init-on-free enabling – ① re-zero memory regions when heap memory is freed
  - Linux kernel free-page-reporting – ② report freed pages in the CXL compressed region, ③ trigger trim/zeroing
  - Proactive page promotion & demotion – ④ hot page promotion w/ DAMON, ⑤ cold page demotion w/ kswapd
  - Direct reclaim – ⑥ enforce direct reclaim by increasing the watermark of the CXL compressed NUMA node
  - Reclaim host-visible CXL memory – ⑦ prevent additional CXL memory allocation by the host.
    - Bare-metal – partially offline the CXL memory region
    - Hypervisor/VM or K8S environments – reclaim CXL memory resources from VMs or worker nodes



CXL & Fabric Memory in DC

- Megatrend in AI server is "Bring data closer to the compute"
  - 2.5D HBM → custom-HBM & 3D zHBM
    - Directly-connected DIMM to GPU → DIMM to cHBM data transfer without CPU
    - 3D stacked zHBM → only vertical direction data movement
  - Issue: limited capacity by brining all data closer to GPU
- CXL and fabric-attached memory for AI servers
  - Less sensitive to latency and cost

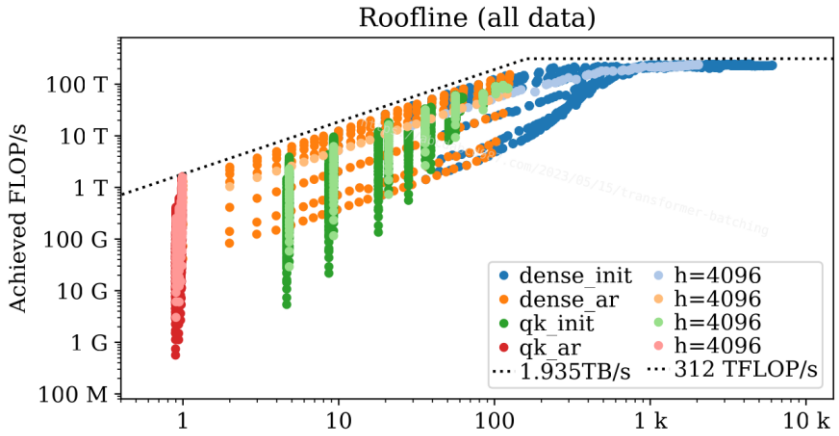


Outline

- Introduction
- Next Generation DRAM Module
  - High-Bandwidth Memory (HBM)
  - Multiplexed Rand DIMM (MRDIMM)
  - Compression-Attached Memory Module (CAMM)
  - CXL Memory Module (CMM)
- Compute-Capable Memory Solutions
  - Processing-In-Memory (PIM)
  - Processing-Near-Memory (PNM)
- Summary

Memory Bandwidth Limits Performance

- system memory bandwidth sets the upper limit for throughput in LLMs.
- Another Limitation of Von Neumann Architecture
  - DRAM consumes large energy to transfer data
- Minimize data movement by processing data in/near memory

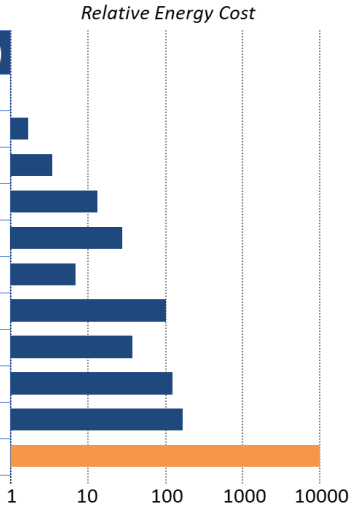


_init: summarization stage
_ar: generation stage (auto-regression stage)
dense: Linear layers (layers w/o self-attention)
qk: self-attention layers

Dissecting Batching Effects in GPT Inference (qun.ch)

Relative Energy Cost

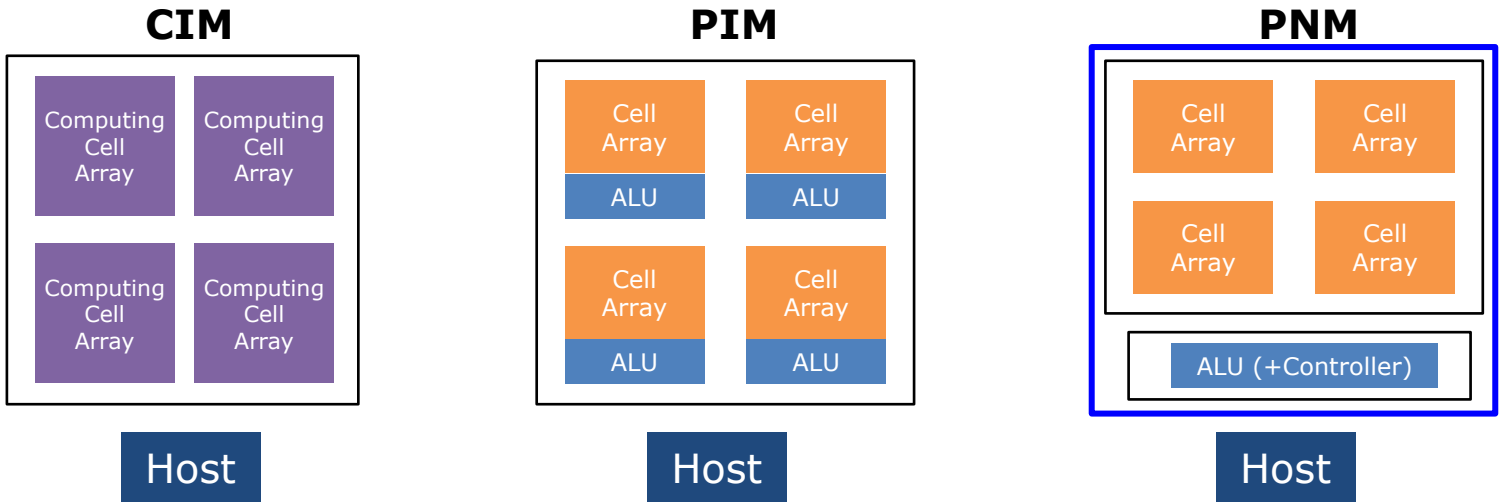
Operation	Energy (pJ)
8b Add	0.03
16b Add	0.05
32b Add	0.1
16b FP Add	0.4
32b FP Add	0.9
8b Mult	0.2
32b Mult	3.1
16b FP Mult	1.1
32b FP Mult	3.7
32b SRAM Read (8KB)	5
32b DRAM Read	640



Source : Computing's Energy Problem (and what we can do about it) (ISSCC'14)

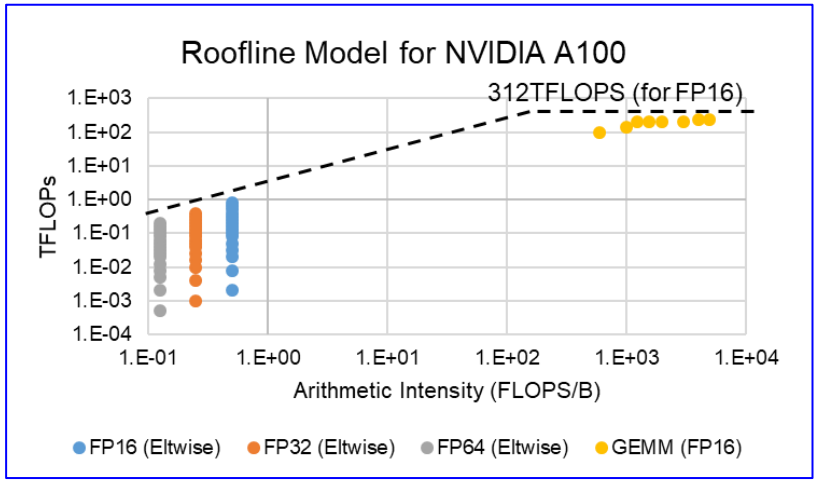
Intelligent Memory and Types

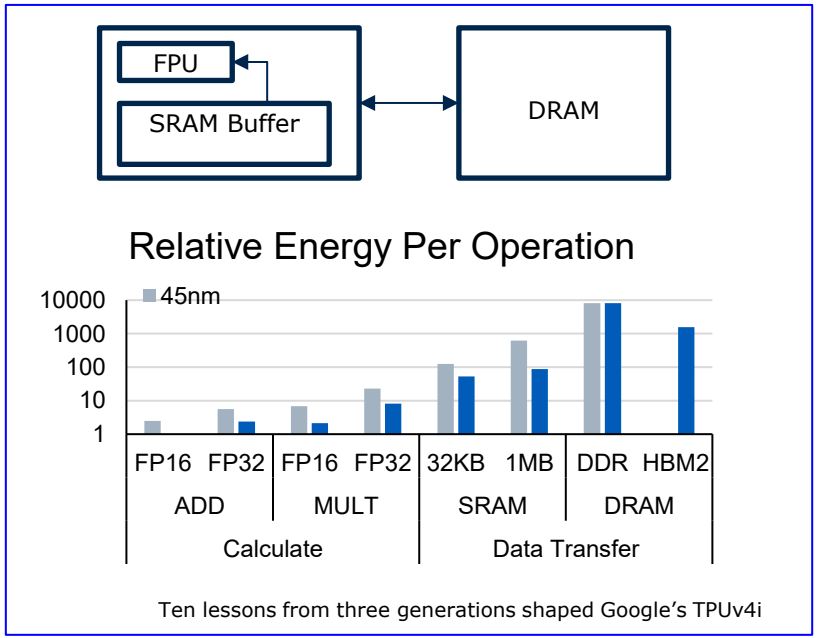
- Three distinct categories in this talk
  - CIM: use memory array as a processing unit
  - PIM: use embedded logic near memory array as a processing unit
  - PNM: use an additional chip for processing inside a memory package or a module



PIM: Renewed Interest

- ML workloads w/ growing model size need more frequent DRAM accesses, limiting performance and dominating energy consumption, which is not scaled (reduced) by enhanced process node



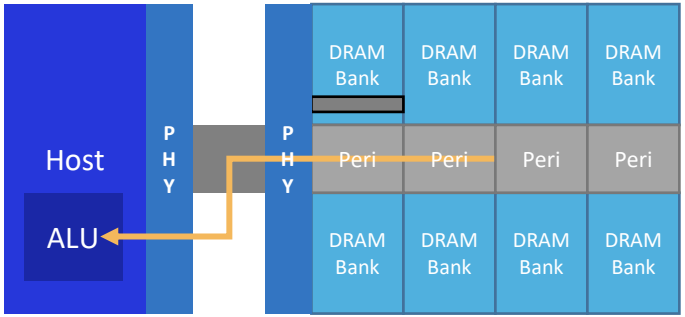


Ten lessons from three generations shaped Google's TPUv4i

Processing-in-Memory (PIM)

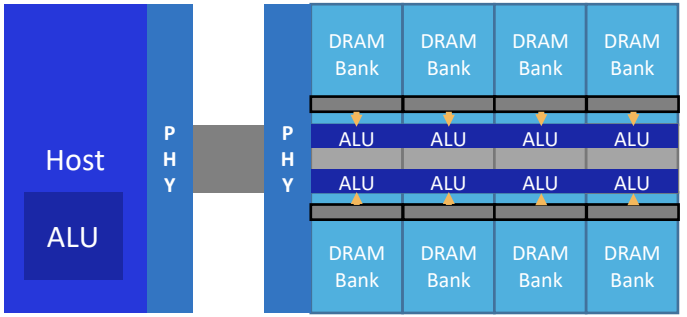
- Embedded arithmetic logic boosts bandwidth and energy efficiency
  - Key idea is utilizing bank-level parallelism
    - Host can access 1-bank (or BG) at a time ↔ PIM 8-BGs all-banks in parallel
  - Remove power consumption for data-path (IOSA-Peri-PHY)

Normal DRAM



Data PHY access for read operation (single bank)

PIM DRAM



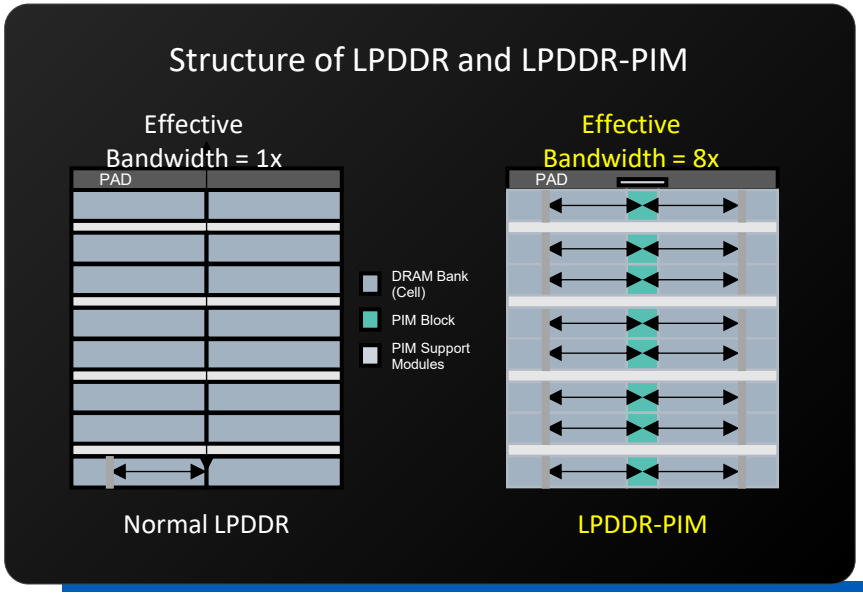
No data PHY access for PIM operation but all bank read

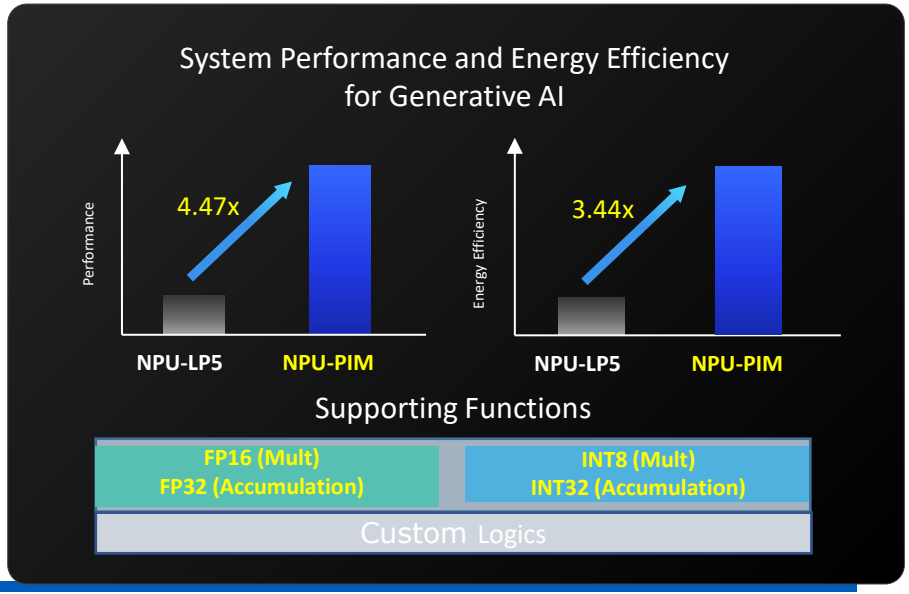
Samsung PIM Development Philosophy

- Be **memory device**
  - Must have an operating mode as a normal memory device
- Utilize **internal memory bandwidth** by parallelism of bank/rank
  - Need to embed processing unit inside memory die
- **Help** host processor to solve memory wall issue
  - PIM is not efficient for general processing
  - Host and DRAM-PIM have each role to improve whole system performance

LPDDR-PIM

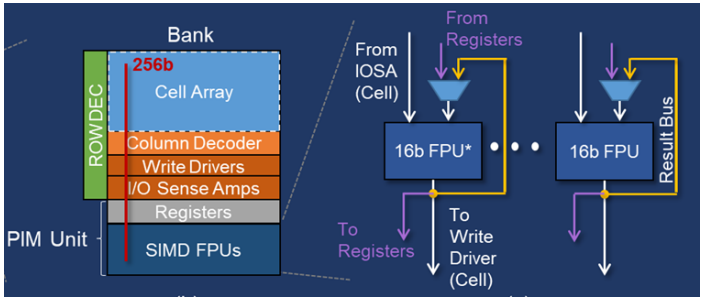
- LPDDR-PIM improves performance and energy efficiency of the system with in-DRAM processing
  - Performance: Utilizes up to 8x higher in-DRAM bandwidth by bank parallel operation
  - Energy Efficiency: Reduces data movement energy by utilizing in-DRAM processing unit

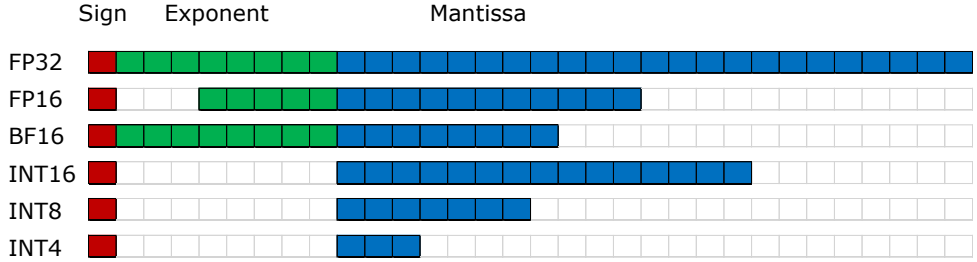




Challenges on On-Device AI w/ LPDDR-PIM

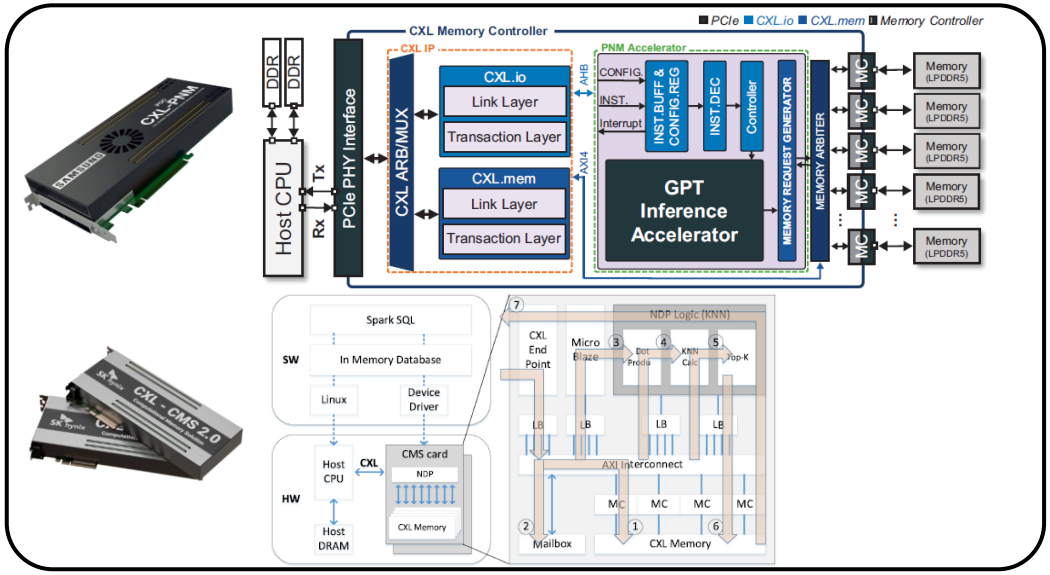
- Supporting various data format (e.g., INT4, INT8)
  - Low-precision data format is used because of model size issue (capacity issue).
  - PIM architecture should support ALUs covering multiple data formats.
- Thermal & Power issue
  - For mobile SoC, thermal stress should be effectively mitigated under limited power.



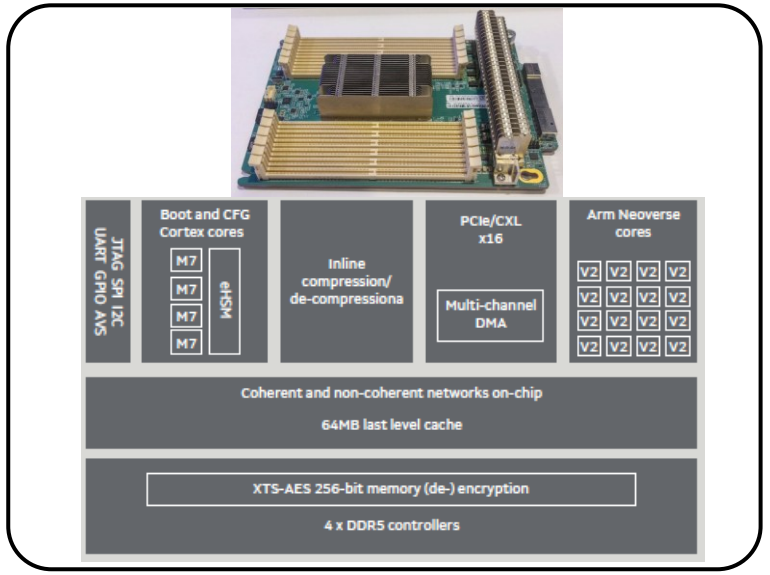


CXL PNM Approaches

- Initially developed as PoC using FPGAs, now being advanced via development on ASIC
  - FPGA-based application specific function acceleration: Samsung CXL-PNM ('23), SK Hynix CMS ('23)
  - ASIC-based general purpose application acceleration: Marvell Structera-A, XCENA MX1 ('26)

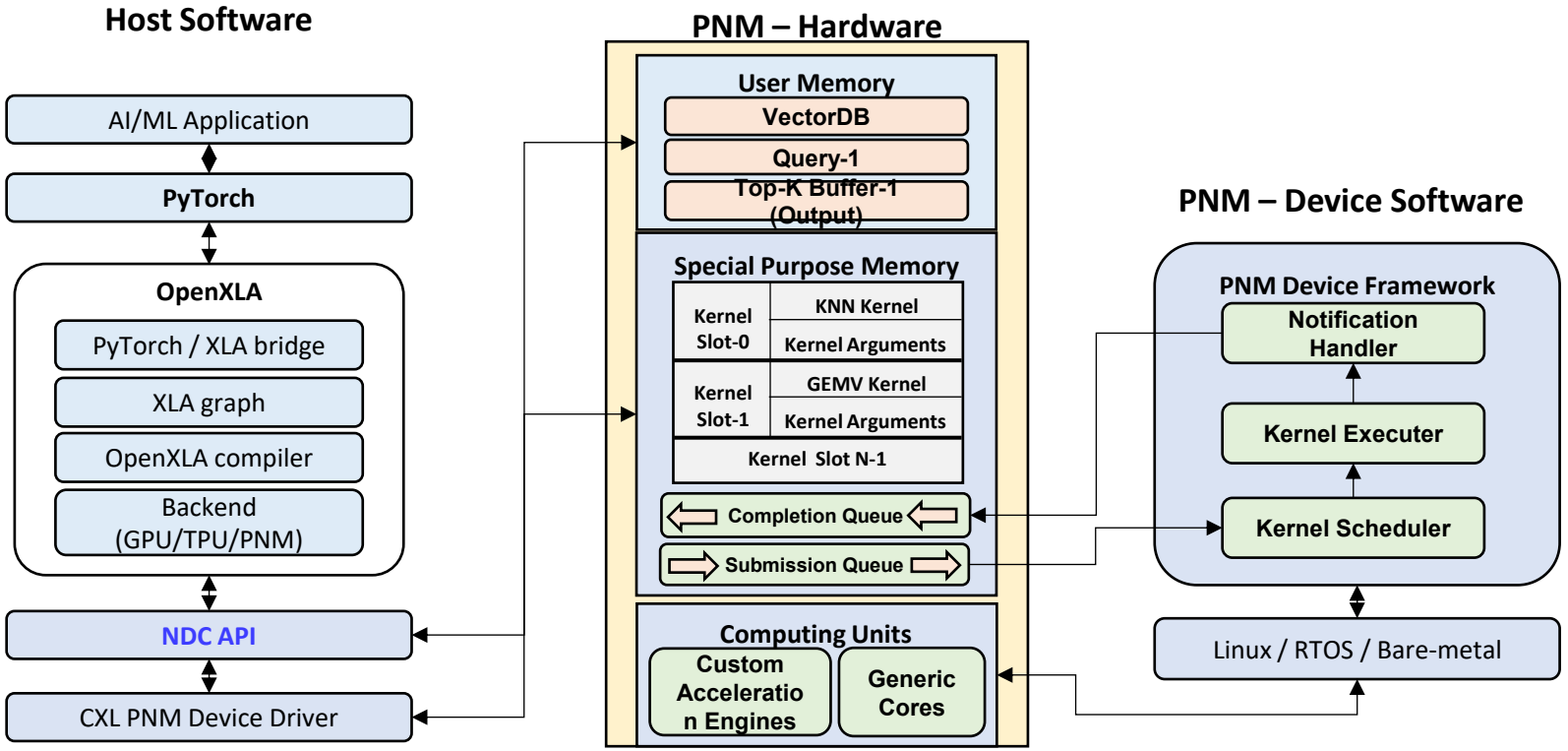


[FPGA-based CXL PNM: Samsung CXL-PNM (top), SK Hynix CMS (bottom)]



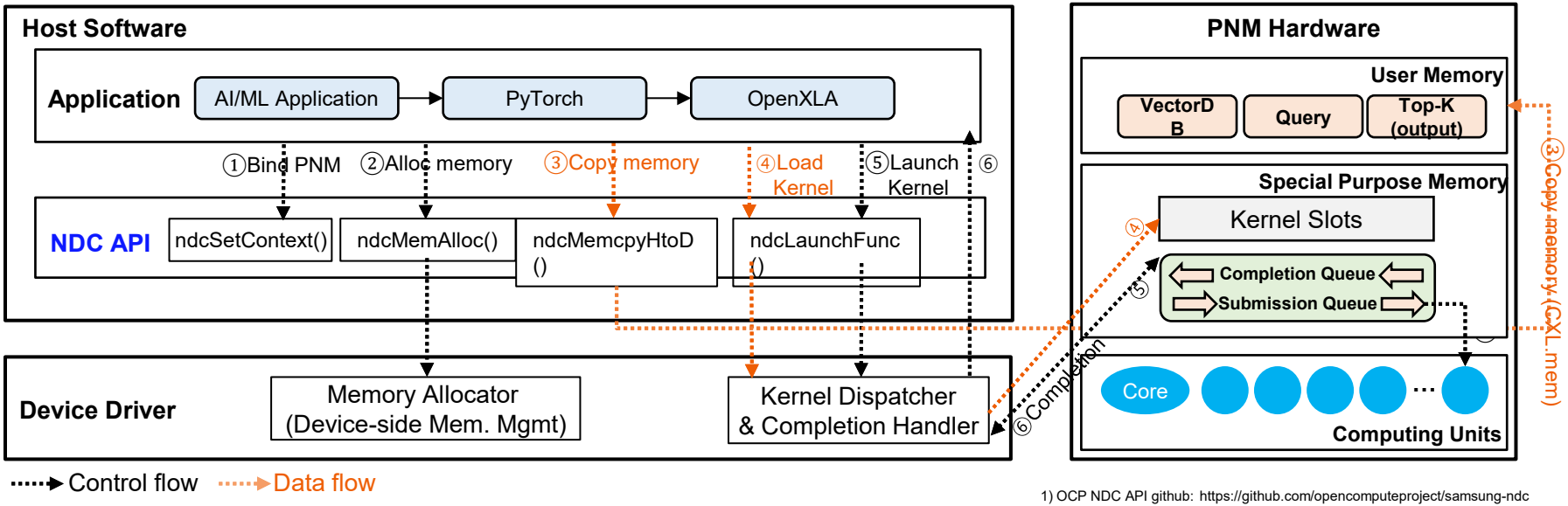
[ASIC-based CXL PNM (CMM-DC): Marvell Structera-A]

CXL PNM HW/SW Block Diagram



Vendor-agnostic PNM Library: NDC API

- NDC API provides task offloading interfaces for Near-Data-Computing devices
  - Bind PNM → Alloc memory → Copy memory → Load Kernel → Launch Kernel → Completion



1) OCP NDC API github: https://github.com/opencomputeproject/samsung-ndc

Challenges on Commercializing PIM & PNM

- Embedding processing unit inside DRAM chip or chip on memory module
  - PPA (Power, Performance, and Area) analysis
- Standardization for bigger market
  - Establishing JEDEC/OCP/CXL Specification
- System-level
  - Cache coherency & Memory request ordering
  - RAS (Reliability/Availability/Serviceability)
- Software-level
  - Supporting mature software stack
- Wide Target Application
  - Anything else except GEMV?

Summary

- Memory vendors provide Various Memory Solutions to meet requirements.
  - Customized HBM and 3D-stacked HBM
  - MRDIMM for high-BW and high-capacity
  - LP-based CAMM solution for lower-power, higher-capacity and BW
  - CXL memory module for capacity/BW expansion, pooling & sharing

- Processing capability in/near Memory enables higher bandwidth and energy efficiency.
  - CIM, PIM and PNM are meaningful and heavily studied in school and industry.
  - Still lots of challenges to be solved for commercializing.
  - Need strong collaboration between system, processor, memory and software.

RAS Techniques

Wendy Elsasser
Technical Director
Rambus Inc.

Rambus

Errors Are Not All Equal—and Some Are Invisible

- Errors beyond ECC coverage can alias, mis-correct, or escape detection
  - Detect error in wrong bit(s) and miss-correct
  - Do not detect any errors

  → Leads to SDC – host unaware of errors
- Host + On-die solutions in place
  - On-die ECC covers errors within 1 die
  - Host ECC covers errors across dies in a channel (high detection for low SDC)
- Transparency at host required
  - ECC is not complete without probity

DRAM Die	Single bit
	Bit-line, data line
	Data line control
	Column select line
	Sub-wordline arm
	Sub-wordline driver
	Main wordline (Row)
	Bank, Channel
	Full Die
	Command decode
Package	Bump failures Wire-bonding or TSV Shorts, opens
Module	Connectivity failures Thermal Issues

*Example list of faults, not exhaustive

Impacts multiple dies / ranks

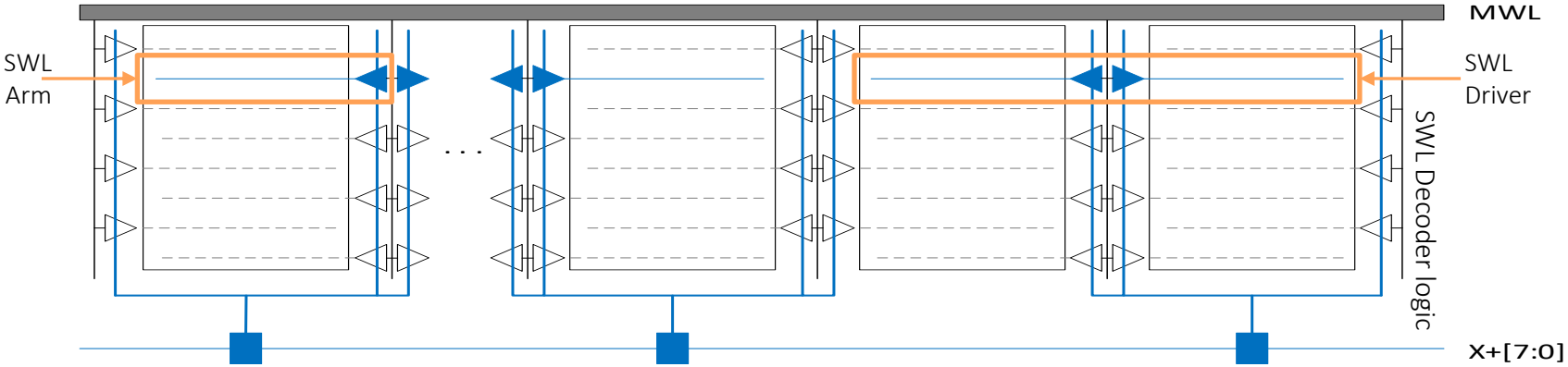
SDC: Silent Data Corruption

RAS Policies Map Error Types to Recovery Actions

Error Type	Policy / Potential Action
Single correctable error (CE)	Correct error Update relevant telemetry data Optionally scrub (demand scrub)
Detectable uncorrectable error (DUE)	Retry access If error persists, poison line Firmware or OS recovery if needed
Repeated errors in same DRAM rows	Remap rows / Trigger sparing
Repeated errors in same DRAM rank	Trigger sparing / maintenance alert
Repeated errors in same OS page	Retire/offline page
High temperature margin issue	Throttle traffic, derate timing

Fault Mechanisms Determine Error Blast Radius per Burst

- 1-MWL is connected to multiple SWLs (E.g., 8 SWLs)
- SWL decoder activates a single SWL
  - Decoder logic inputs are MWL and decoded X+ signals (derived from lower address bits)
  - Decoder logic located between mats with logic shared between 2 mats to minimize area



- ➢ SWL Arm error only affects data from 1 mat (I.e., 16b per burst)
- ➢ SWL Driver error affects data from 2 mats (I.e., 32b per burst)

  DDR x4 bursts are ½ the prefetch and could reduce SWD blast radius[1]

[1] "DRAM Fault Classification through Large-Scale Field Monitoring for Robust Memory RAS Management", Hoiju Chung, et all, MICRO 2025

Stronger ECC Coverage Increases Complexity and Overhead

- Hamming, BCH are lower coverage schemes
  - Detects random bit failures (SEC, SECDED, DECTED)
  - ($\log_2$(datawidth) + 2) parity bits required for SECDED (E.g., 8b for 64b data)
  - Protects against soft errors (radiation/noise), marginal cells, random isolated bit flips

Random Bit Errors

- Reed-Solomon provides higher coverage, at a cost
  - Symbol based code that requires more redundancy (higher overhead)
  - Protects against failures with larger blast radius (I.e., SWD, full chip)
  - With 2t ECC symbols, corrects 1t symbols in a codeword
  - ➢ SDDC, single die data correction, requires a 2 die overhead

Burst, Nibble, Byte, Device Errors

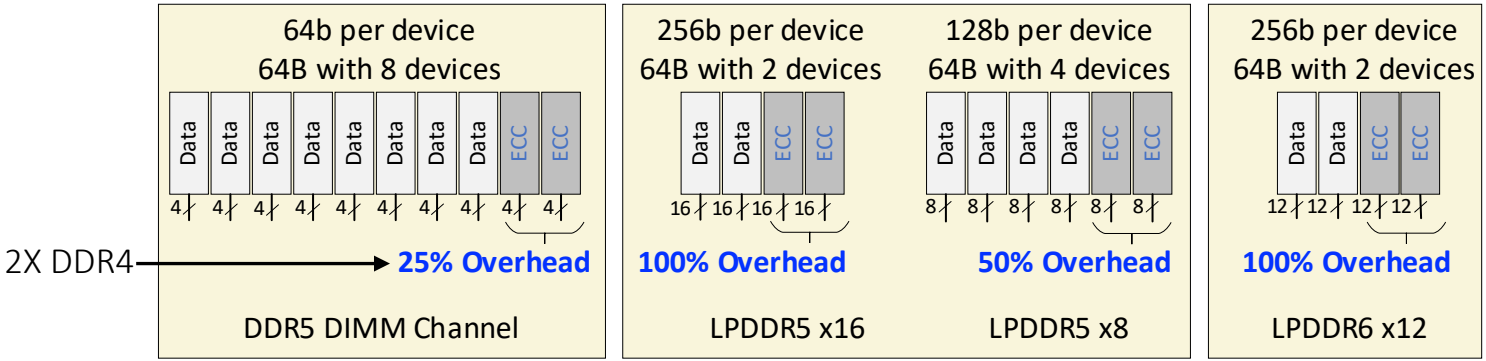
- Reed Solomon with erasure coding to increase coverage
  - With 2t ECC symbols, corrects 2t symbols when error locations are known
  - Constraint: #ECC symbols ≥ 1*Erasures + 2*Corrections

Known Symbol Errors

System Organization Has a Big Impact on ECC Overhead

DDR vs LPDDR

- Same fundamental bit cell and high-level array architecture
- Similar on-die ECC capability (SEC across 128b)
- Fundamental difference → Number devices per channel
  - Fewer bytes accessed per device (smaller prefetch) leads to more devices per channel with DDRx
  - Reduces overhead for symbol based ECC coding



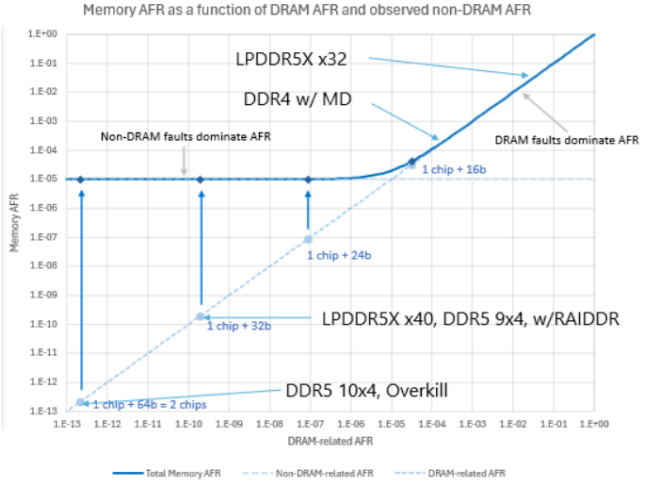
Overhead for Single-die Data Correction (SDDC) capability with Reed-Solomon Coding

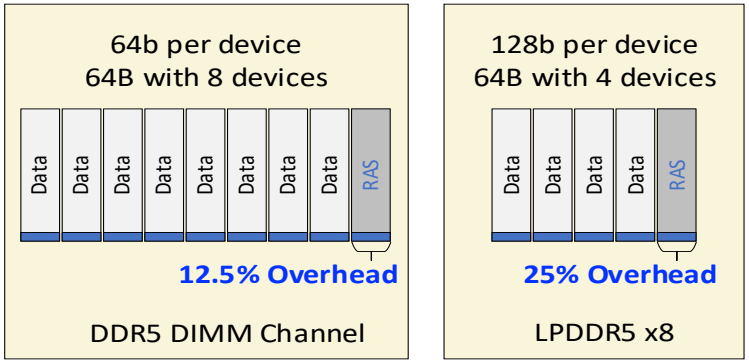
New Schemes Reduce System-Level Overheads

RAIDDR (RAID for Double Data Rate)

- Inspired by RAID techniques from storage
- Uses a mix of parity, CRC, and BCH (vs Reed-Solomon)
- Basic RAIDDR: Parity + CRC
  - Works with standard DIMMs
- Enhanced RAIDDR: Parity + CRC + BCH
  - Covers additional single bit failures (SDDC+1b)

Microsoft View on RAS Requirements[1]



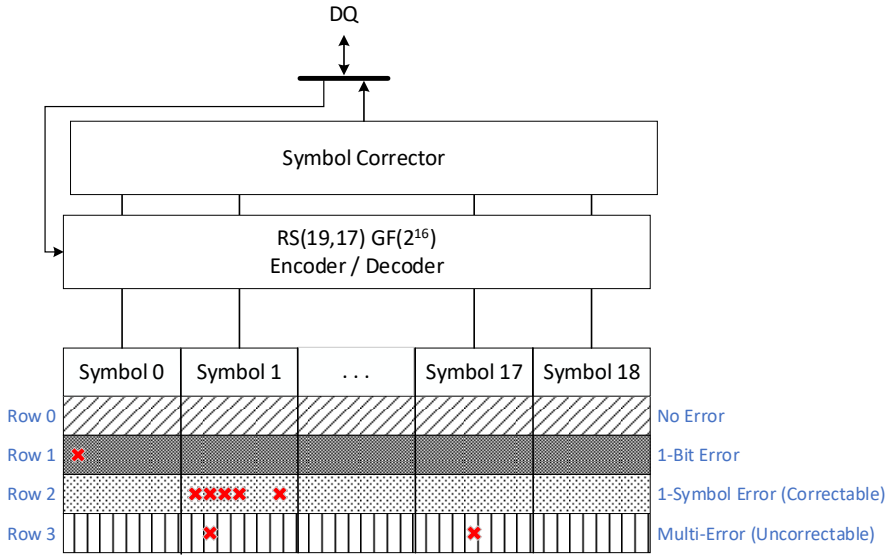


- ½ the overhead vs Reed-Solomon
- Enhanced RAIDDR leverages ~6% per die overhead
  - On-die ECC storage (or meta-data)
  - 1-die + 32b ECC bit required

[1]https://techcommunity.microsoft.com/blog/azureinfrastructureblog/raiddr-redefining-memory-reliability/4487951

HBM Improves on-Die ECC to Expand Fault Coverage

- DDRx & LPDDRx use SEC for on-die ECC
  - Covers single bit error correction
  - Fast and simple
- HBM has incorporated symbol-based coding for higher on-die ECC coverage
  - I.e., RS(19,17) GF(2^{16})
  - Corrects one 16-bit symbol
- Architecture determines fault coverage
  - I.e., Can correct sub-wordline failures if each SWL driver affects <= 16-bits in a burst
- Used with system level error detection
  - Leveraging **16-bit meta-data** accessed concurrently with data



ISSCC 2022 28.1, "A 192-Gb 12-High 896-GB/s HB3 DRAM with a TSV Auto-Calibration Scheme and Machine-Learning-Based Layout Organization"

RAS is More Than Just Reliability

Reliability

Detect (and correct) individual errors

- Error correcting codes
  - On-die and system level
  - SECDED, Reed-Solomon, …
- Scrubbing
  - Patrol and demand
- CA and data parity (i.e., CRC)
- Row hammer mitigation

Availability

Continue to operate

- Error logging and telemetry
- Sparing logic
- Page off-lining

(Sparing logic, Page off-lining: Firmware and/or OS)

- Lockstep mode
  - Larger ECC codeword that can tolerate more faults
- Memory mirroring
- Dynamic throttling/derating

Serviceability

Detect, isolate, report and repair faults efficiently

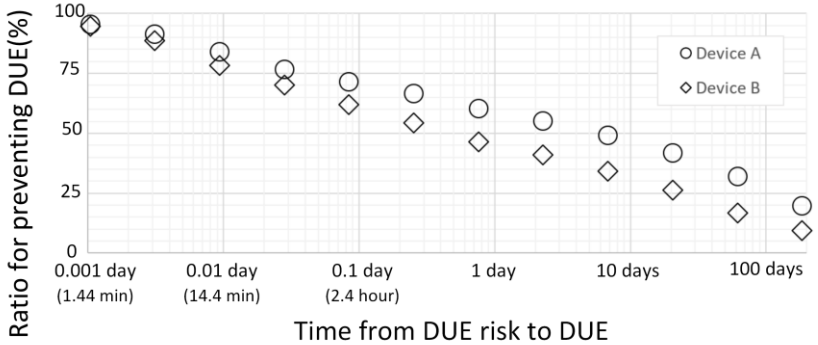
- Failure identification
  - Device, error type
  - Urgency; need to replace
- Predictive failure analysis
- Firmware & BIOS diagnostics
  - Identify issues on boot

DRAM RAS = Data Integrity + Uptime + Fast Recovery

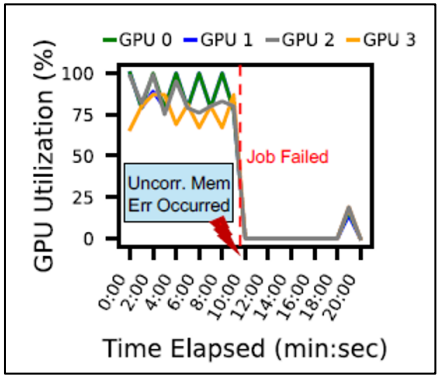
Availability is Increasing in Importance

CE: Correctable errors
UE: Uncorrectable error
DUE: Detectable, uncorrectable errors

- Shifting from “Make DRAM more reliable” to “Service survival despite DRAM faults”
- Availability is more than just hardware solutions
  - Includes telemetry + firmware policies + OS containment
- Uncorrectable errors are operationally costly
  - 17.2% of pre-training interruptions over 54-days of Llama-3 405B due to HBM3[1]
- DUE prevention depends on how quickly risk indicators trigger mitigation



[2] “DRAM Fault Classification through Large-Scale Field Monitoring for Robust Memory RAS Management”, Hoiju Chung et all, MICRO 2025



DRAM UEs are costly[1]

[1] “Story of Two GPUs: Characterizing the Resilience of Hopper H100 and Ampere A100 GPUs”, Shenkun Cui et all, SC 2025

Required DRAM RAS Depends on the Deployment Environment

Client, mobile

- Closed, single tenant systems
- Power conservation, thermal efficiency prioritized
- Lower RAS capability required

AI Data Center

- Errors result in accuracy loss
- Risk of cascading errors with multi-agent communication
- Checkpointing and redundancy for recovery
  (Uncorrectable errors will occur in long training)

Traditional Data Center

- Less tolerant to errors and sensitive to failure rates
- Must maintain continuous uptime for customer experience
- SDDC reliability along with availability & serviceability

Automotive & Aerospace

- Strong detection to mitigate SDC (along with correction)
- Functional safety integration
- Environmental derating and detailed diagnostics

Future Challenges

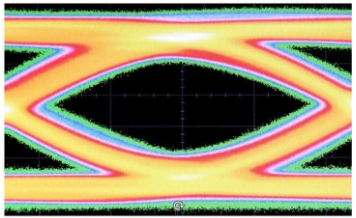
Steven Woo
Fellow and Distinguished Inventor
Rambus Inc.

Rambus

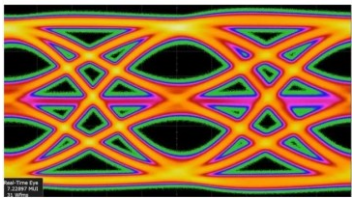
Signal Integrity and Reliability Becoming More Challenging

Multi-PAM Signaling

GDDR6



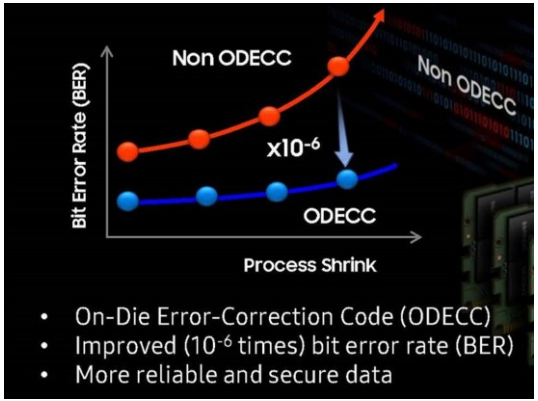
GDDR7



Sources: https://www.rambus.com/rambus-achieves-industry-leading-gddr6-performance-at-18-gbps/, https://semiengineering.com/gddr7-the-ideal-memory-solution-in-ai-inference/

- Lower frequency, multiple bits/symbol
- Reduces rate DRAM IOs need to run
- Increases bandwidth, tradeoff of signal integrity challenges

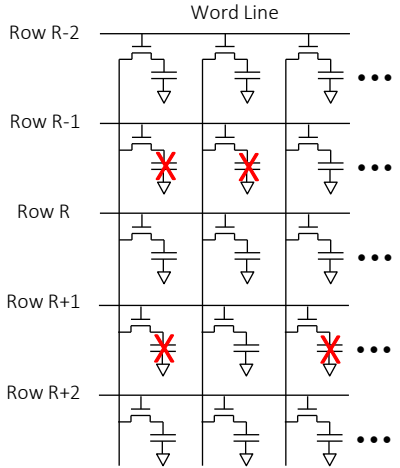
On-Die ECC



Source: "Samsung Teases 512 GB DDR5-7200 Modules," https://www.anandtech.com/show/16900/samsung-teases-512-gb-ddr5-7200-modules

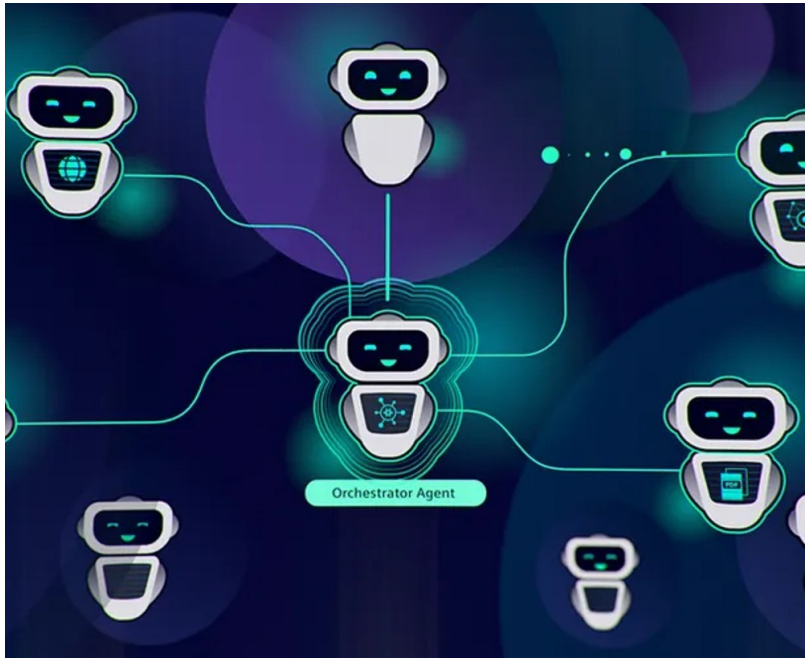
- DRAMs more error-prone
- On-die ECC improves reliability

RowHammer and RowPress



- Bit cells can flip due to activity in neighboring rows (disturb errors)
- Becoming more problematic, Hammer Counts dropping

Agentic AI: New Reliability and Availability Requirements

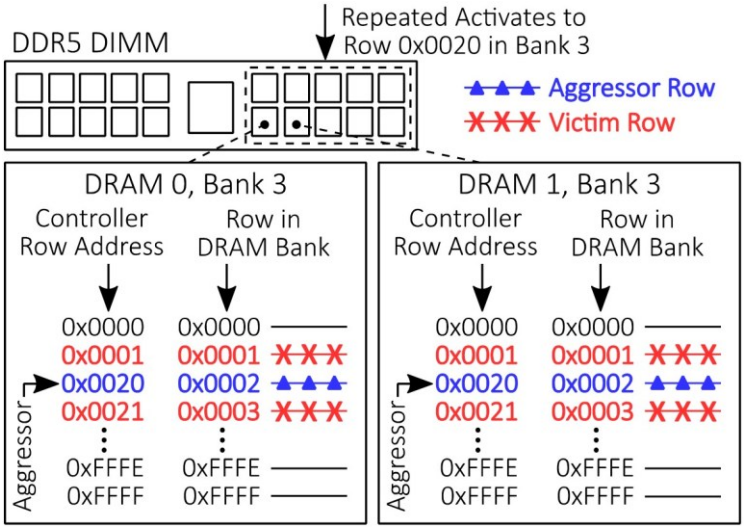


- Agentic AI: Multiple agents working together, some agents use results of other agents
- Agents call tools, do complex lookups
- Large jobs run on multiple cores/servers
- Need high reliability: Agents produce results for other agents, errors can propagate and produce bad results, waste resources
- Need high availability: Each disruption delays forward progress, focus on system resilience
- Telemetry growing in importance: Monitor system to gauge quality of results

DRAMs and memory systems are critical for AI, need to ensure reliability and resilience

Even with ECC, RowHammer/RowPress are Problems for Servers

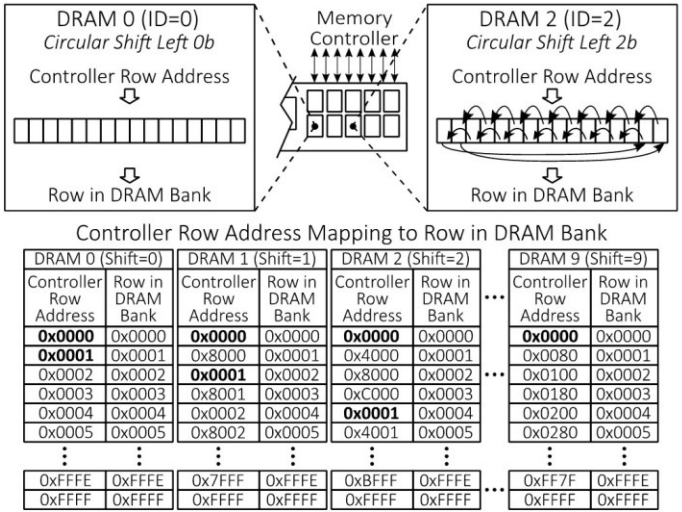
- DRAMs on DIMM map controller row addresses to rows in DRAM core in the same (non-public) way
- ECC can only correct a limited number of bit errors
  - Chipkill ECC can correct all errors from a single DRAM
- A RowHammer attack can flip many bits in multiple DRAMs, easily overwhelming ECC
  - Can result in uncorrectable errors or Silent Data Corruption (system can't tell data is corrupted)

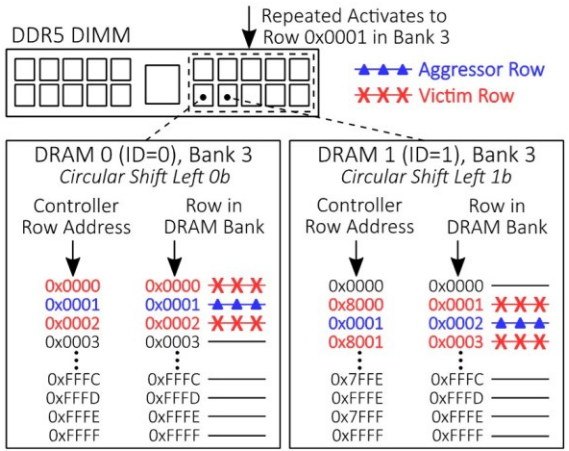


Fundamental problem: Row addresses can be neighbors in more than 1 DRAM, leading to many potential bit flips that can overwhelm ECC methods

Row Address Remapping with RAMPART

- Key observation: Can confine bit flips in victim row addresses to a single DRAM if controller row addresses are never neighbors in more than one DRAM
- Each DRAM maps addresses differently: incremental design change, no performance impact
- RAMPART: Row Address Map Permutation And Reassignment Technique

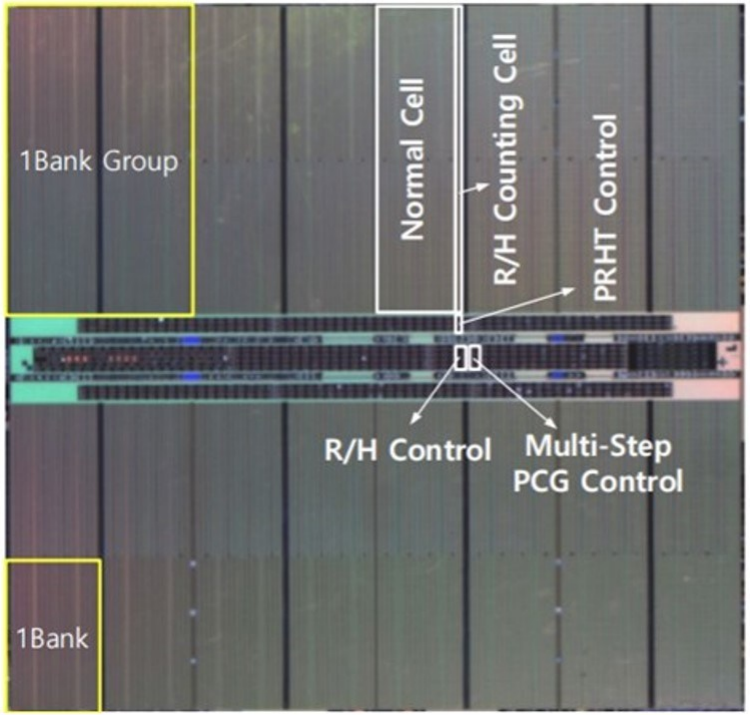




- Hammering controller row address 0x0001 flips bit in row addresses 0x0000 and 0x0002 in DRAM 0, and row addresses 0x8000 and 0x8001 in DRAM 1.
- A subsequent read to controller row address 0x0000 will only see errors from DRAM 0, all other DRAMs will be unaffected
- For any victim row address, RAMPART confines bit flips to a single DRAM

Confining all bit flips for any victim row address to a single DRAM allows Chipkill ECC to identify and correct errors from one successful RowHammer attack

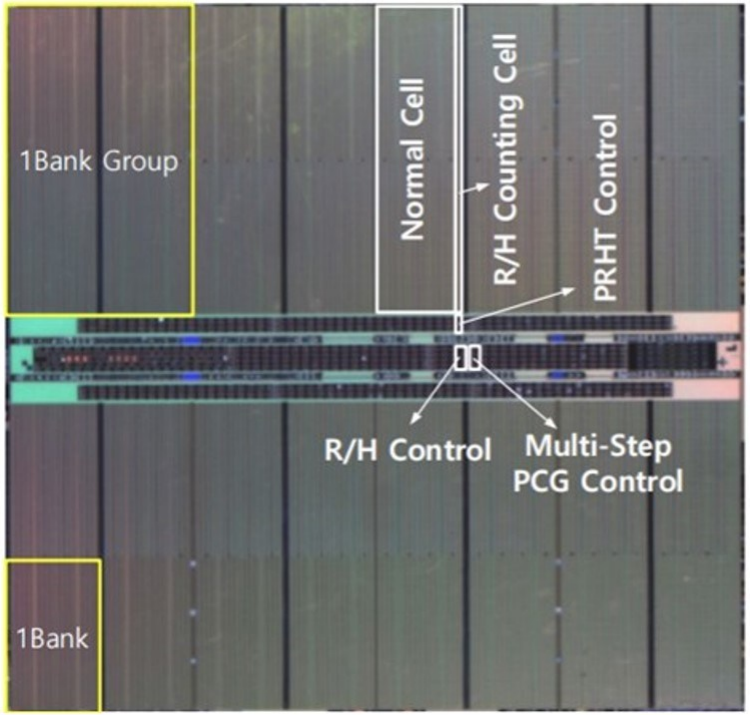
PRAC: Per-Row Activation Counters (1)



Source: W. Kim et al., "A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement," 2023 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2023, pp. 1-3, doi: 10.1109/ISSCC42615.2023.10067805.

- Many proposals for preventing RowHammer/RowPress
  - Controller methods
  - DRAM methods
- PRAC is industry approach to preventing RowHammer and RowPress bit flips, adds a counter per row
- Additional bits needed for counters, but small overhead per 1KB row
  - 16b PRAC counter is 16b adds 0.2% storage per 1KB page
  - DDR5 on-die ECC adds 64B per 1KB page (+6.25%)
- Counters incremented when pages are closed, value based on activate operation and duration page is open
- Increases length of row cycle time to increment and write back new counter value

PRAC: Per-Row Activation Counters (2)

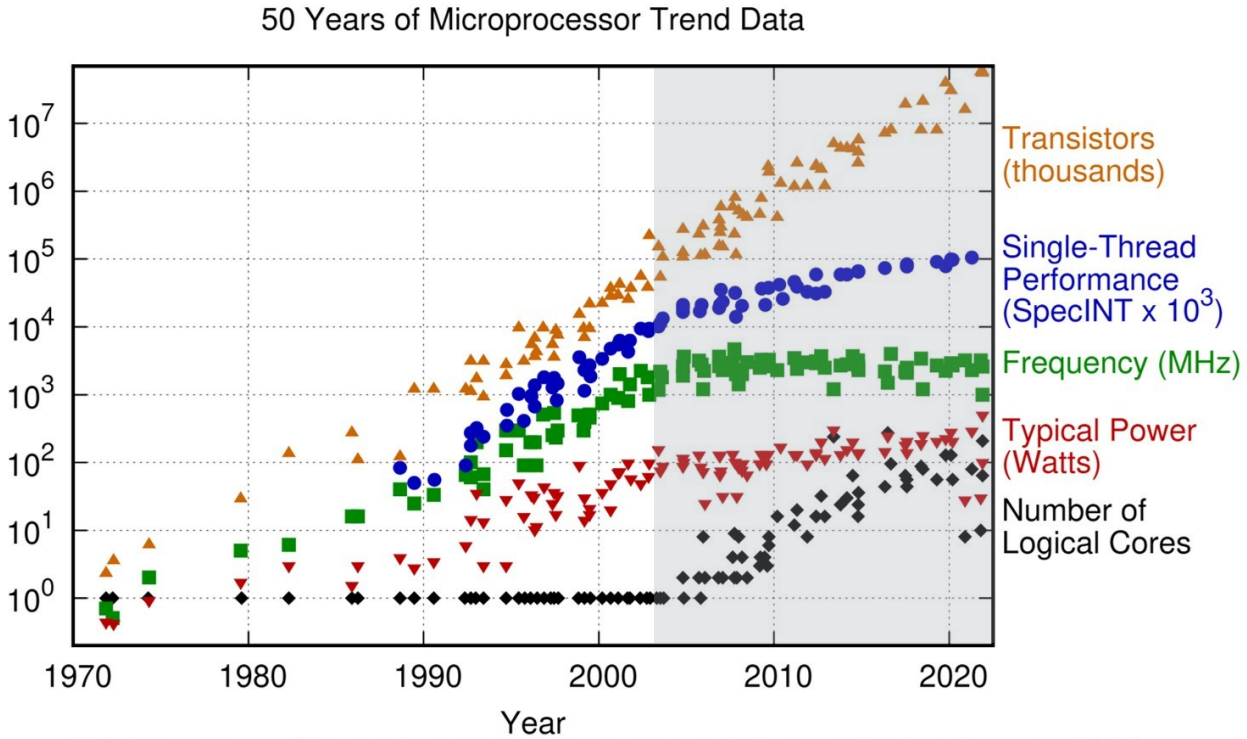


Source: W. Kim et al., "A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement," 2023 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2023, pp. 1-3, doi: 10.1109/ISSCC42615.2023.10067805.

- When PRAC count value exceeds a manufacturer-defined threshold
  - DRAM can refresh victims during Refresh operations
  - Controller alerted when PRAC counts get too high (e.g., too many rows have exceeded their thresholds), controller issues RFM operations until situation resolves
- Active research area, many papers at top architecture and security conferences, workshops like DRAMSec
  - DRAMSec 2026 Sunday June 28 (Full-day Workshop)
- Rambus research on RowHammer mitigation in servers

S. C. Woo, W. Elsasser, M. Hamburg, E. Linstadt, M. R. Miller, T. Song, and J. Tringali. 2024. RAMPART: RowHammer Mitigation and Repair for Server Memory Systems. In *Proceedings of the International Symposium on Memory Systems (MEMSYS '23)*. Association for Computing Machinery, New York, NY, USA, Article 4, 1–15. https://doi.org/10.1145/3631882.3631886

Computation Increasingly Power and Bandwidth-Limited

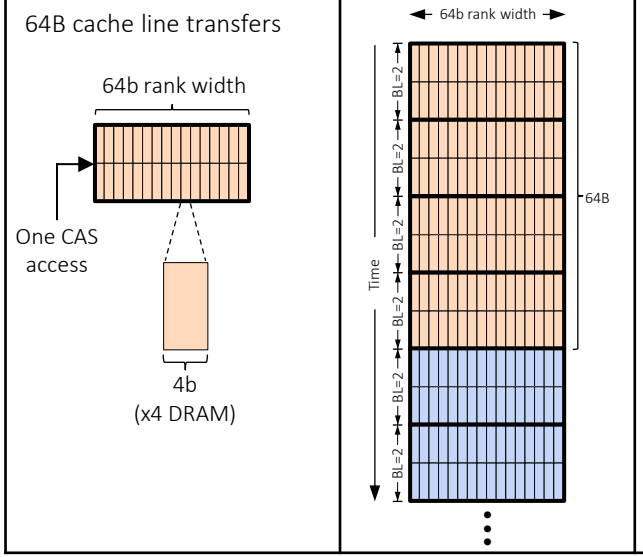


Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp

Source: Karl Rupp, "50 Years of Microprocessor Trend Data"
https://github.com/karlrupp/microprocessor-trend-data/tree/master/50yrs

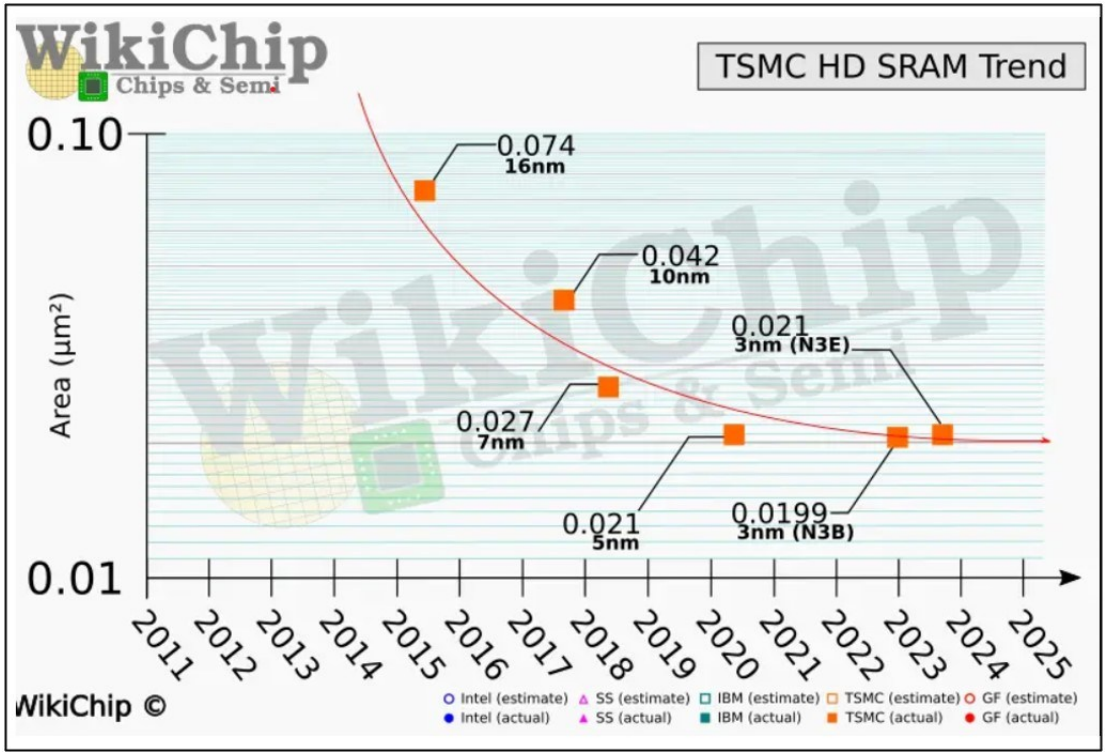
- Dennard scaling ended, chiplets enabling transistor growth
- Domain-specific silicon improving performance and energy-efficiency
- Parallelism increasing
- Memory system bandwidth under stress
- More liquid cooling in the future to manage heat

Keeping the DRAM Low Cost: DRAM Core Speed

	DDR		DDR2		DDR3		DDR4		DDR5	
Data Rate (Gbps)	200	400	400	800	800	1600	1600	3200	3200	6400
tCCD / tCCD_L (ns)	15	10	10	5	10	5	6.25	5	5	5
Core Frequency (MHz)	66.7	100	100	200	100	200	160	200	200	200
Prefetch (N)	2	2	4	4	8	8	8	8	16	16
DIMM Rank width (bits)	64/68/72	64/68/72	64/68/72	64/68/72	64/68/72	64/68/72	64/68/72	64/68/72	32/36/40	32/36/40
CAS granularity (B)	16	16	32	32	64	64	64	64	64	64
64B cache line transfers							• BL16 would produce 128B access, too large for 64B cache line => potentially wasted bandwidth • Introduced bank groups to allow core to stay at ≤200MHz		• BL16 together with smaller width provides 64B accesses • Bank groups to allow core to stay at ≤200MHz • Data transport time for 64B similar to DDR4, but can have multiple cache line transfers in parallel on the DIMM module	

Challenge: How to keep scaling data rate while keeping the core cost effective and CAS granularity useful to the CPU?

SRAM Scaling in Slowing, Can DRAM Help Fill the Gap?



Source: "IEDM 2022: Did We Just Witness The Death Of SRAM?," https://fuse.wikichip.org/news/7343/iedm-2022-did-we-just-witness-the-death-of-sram/

- Rising core counts, growing application footprints drive need for more SRAM
- SRAM scaling is slowing
- Opportunity for improved DRAM in the caching hierarchy?
- Lots of research on using DRAMs and modified DRAMs for caching
- UC Davis/Rambus research on modifying DRAM for caching

  M. Babaie, A. Akram, W. Elsasser, B. Haukness, M. R. Miller, T. Song, T. Vogelsang, S. C. Woo, J, Lowe-Power. "Efficient Caching with A Tag-enhanced DRAM," 2025 IEEE International Symposium on High Performance Computer Architecture (HPCA), Las Vegas, NV, USA, 2025, pp. 745-760, doi: 10.1109/HPCA61900.2025.00062

DRAM and Flash Compared with Emerging Memories

Key question: What performance (degradation) is acceptable at lower cost?

			Emerging Memory Candidates			
Characteristic	**DRAM**	**3D DRAM (est.)**	**STT-MRAM* (est.)**	**ReRAM* (est.)**	**Optane (est.)**	**3D NAND Flash**
Latency (RD/WR)	30ns/30ns	30ns/30ns	100ns/100ns	100ns/100ns	300ns/300ns	45µs/660µs
Endurance	>10^{16}	>10^{16}	10^6 – ~10^{10}	10^6 – ~10^9	10^6 – 10^9?	~10^4
Access Granularity	16B	16B	1B	1B	1B	4KB Read/Prog Block Erase
Management	None	None	Wear leveling	Wear leveling	Wear leveling	Block/Page Management, Wear leveling
Write Energy	++	++	-	-	- -	+
Cell Layers	1	>100	1	1	<=4	>300
bits/cell	1 bit	1 bit	1 bit	1 bit	1 bit	3-4 bits
Process Complexity/Cost	1	-	-	+	--	-
Relative cost/bit	1	<0.25	~5	1	0.6	0.015
Relative capacity	1	>4	~.2	1	2	50

* In production as embedded memory

For now, the future of DRAM is still DRAM


Summary and
Closing Remarks
Rambus

Summary and Closing Remarks (1)

- All DRAM is built from a 1T1C bit cell
  - Same building blocks: Cells, Arrays, Data Paths, and Interfaces
- Lots of silicon between processor cores and DRAMs
  - Memory controllers: Schedules transactions to meet performance goals, complexity is growing
  - PHYs: Drive analog signals at high speeds across wires, consume growing amounts of power
  - Buffer Chips: Improve signal integrity, reduce signal count, enable higher data rates
- Different tradeoffs made for each technology to meet the market demands
  - DDR: High capacity, high RAS
  - LPDDR: Low power, low energy per bit
  - GDDR: High fill frequency, high throughput
  - HBM: High fill frequency, high throughput, ultra-low energy per bit
- Communication between host and DRAM sounds easy, but is very complex in reality

Summary and Closing Remarks (2)

- Many scaling tradeoffs – there's no free lunch
  - Continuous re-evaluation as assumptions, constraints, and application needs change
- Host controller complexity is growing as systems evolve (area, power)
- RAS becoming more challenging, growing in importance
- System level requirements matter
  - Can't design each piece in isolation
  - Complex interplay between core timings and performance as DRAM technology scales
- Future memory solutions being developed to address emerging needs
  - MRDIMM, SOCAMM, CXL, and PIM/PNM
  - The future of DRAM is still DRAM

Many challenges ahead as DRAM technology scales, and the semiconductor industry is continuing to innovate to address them

We’re Interested in Your Feedback

- All feedback is welcome, please send your thoughts on the material and additional areas to address to swoo@rambus.com.


Thank you