Scaling DRAM Technology to Meet Future Demands: Challenges and Opportunities

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

ISCA 2025 Tutorial
Waseda University, Tokyo, Japan
June 22, 2025

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 25 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.

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

- Rambus
  - Brent Haukness
  - Michael R. Miller
  - Thomas Vogelsang
  - Lidia Warnes
  - The greater Rambus Labs Team

Tutorial Agenda

- Introduction: Markets and History
- DRAM Architecture: Cell, Array, Data Path, and Interface
- Tradeoffs that Motivate Different DRAM Architectures
- Power and Energy Comparison
- RAS Techniques, Overheads, and Tradeoffs
- Memory Controller Architecture and Design Challenges
- RowHammer and RowPress
- System Performance
- Future Memory Solutions: MRDIMM, SOCAMM, CXL, and PIM/PNM
- Future Challenges
- Summary and Closing Remarks

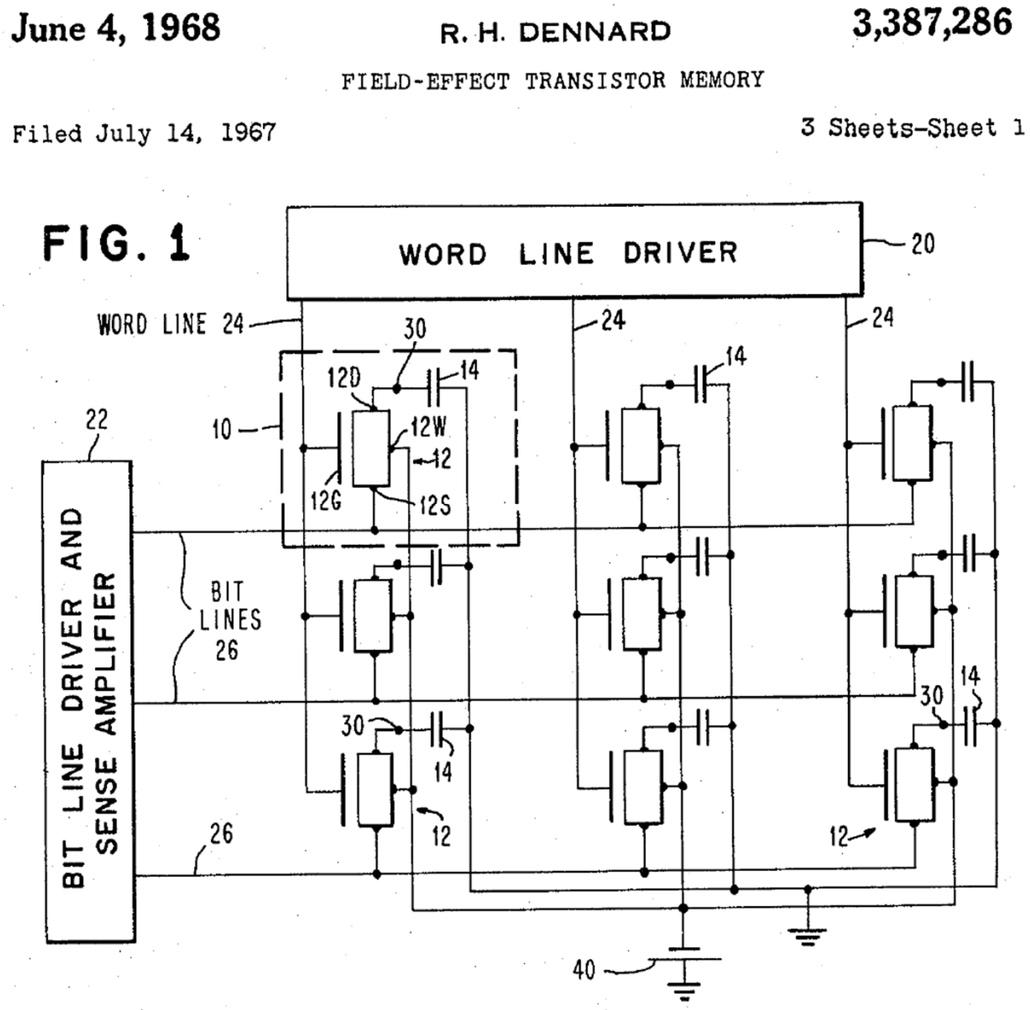
Introduction: Markets and History

Steven Woo
Fellow and Distinguished Inventor
Rambus Inc.

Rambus

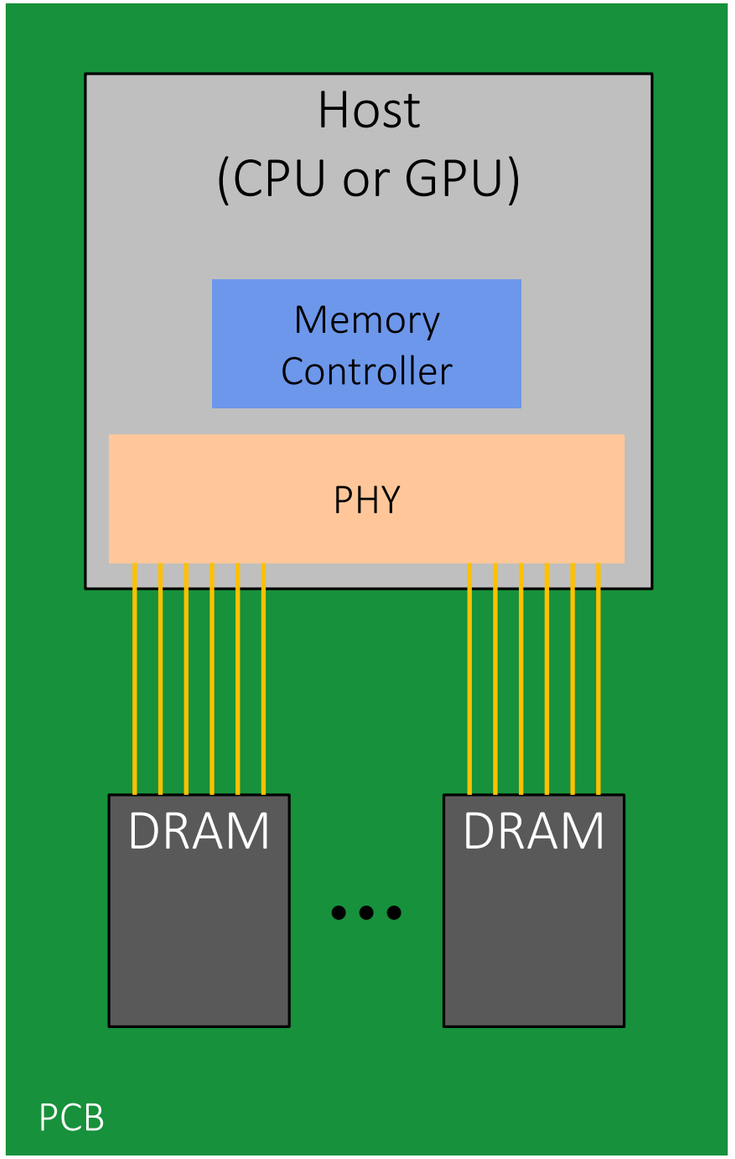
The Fundamental DRAM Building Block: The 1T1C Bit Cell

- *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



Memory Subsystem

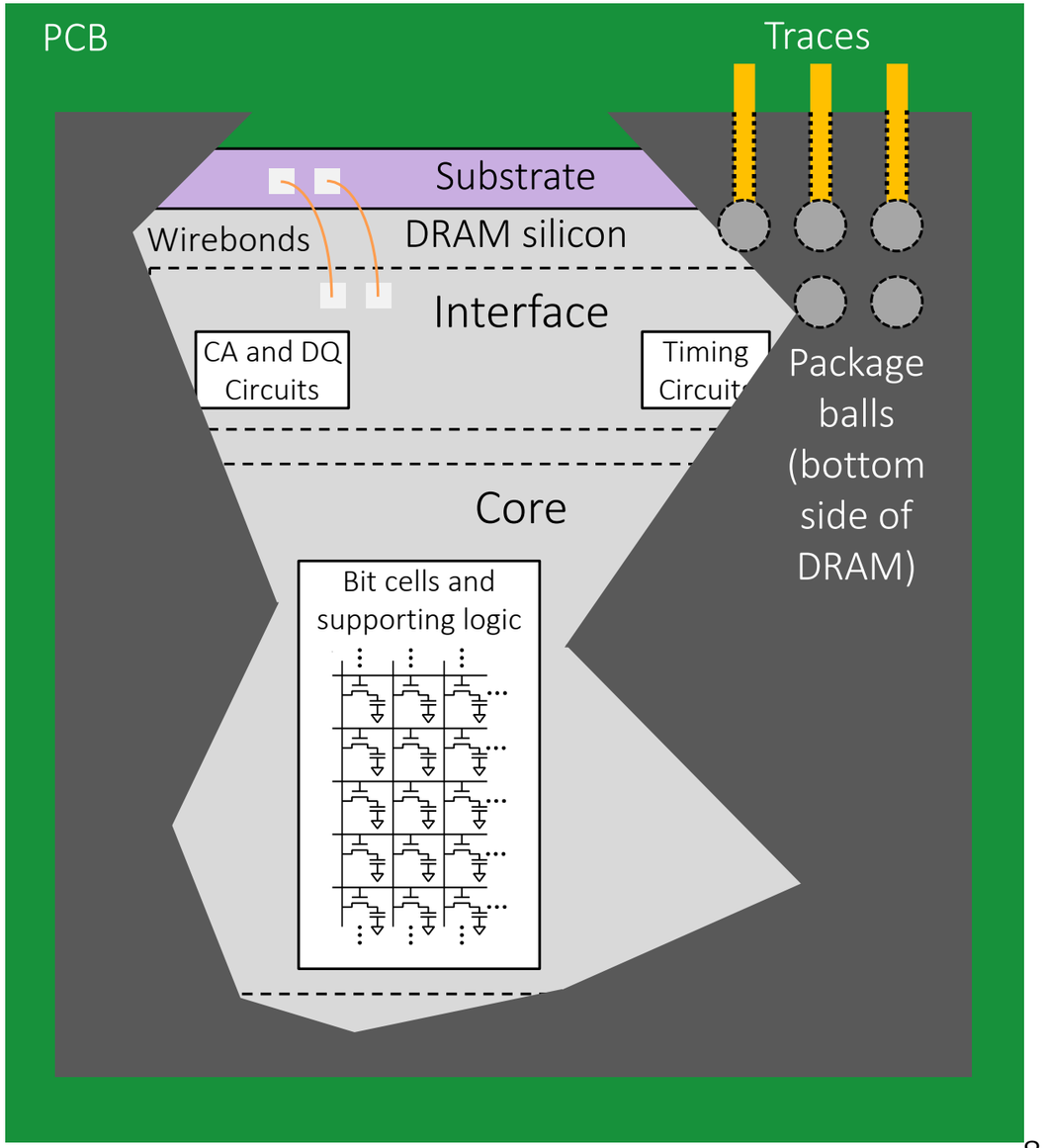
- Memory Controller
  - Accepts memory requests from the host (e.g., Loads and Stores)
  - Translates Load and Store requests into the DRAM-specific protocol required to read and write data to and from DRAMs
- PHY
  - Implements analog mixed-signal circuits (e.g., transceivers) that drive signals across the wires
  - Responsible for voltage and timing of signals (e.g., DLLs, PLLs)
- Traces
  - Carry signals that represent commands and data
- DRAM
  - Hold the data manipulated by the program



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



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

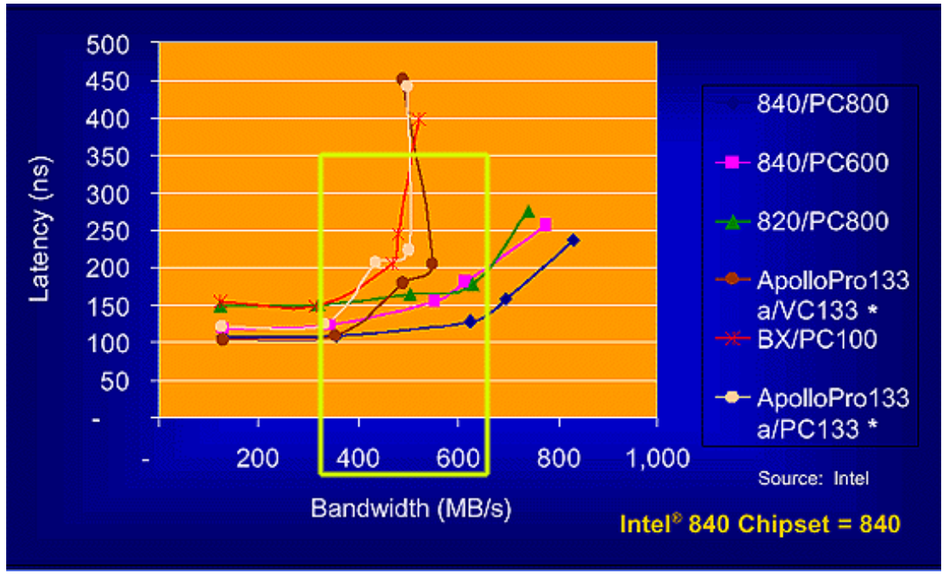
ECC/Reliability

- 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

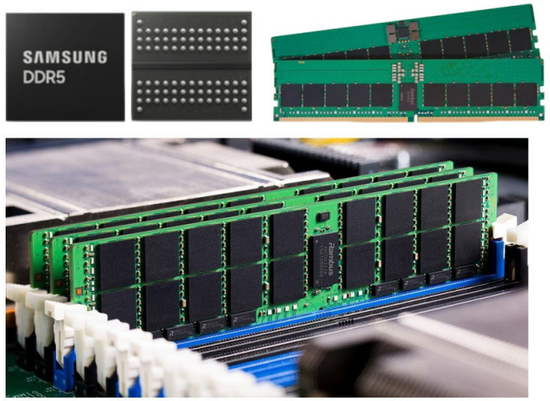


Source: AnandTech, http://www.anandtech.com/cpu/showdoc.html?i=1245&p=4

Different DRAMs for Different Markets

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

Compute

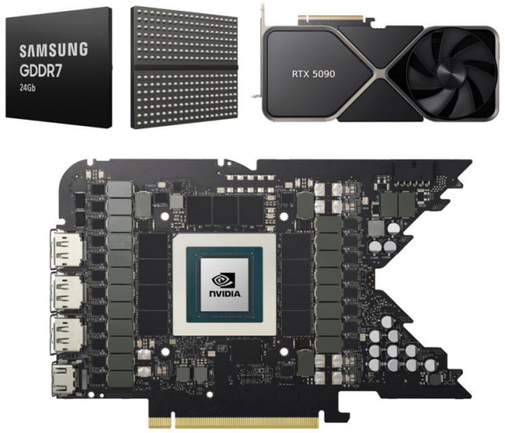


- DRAMs packaged on modules (DIMMs, CAMMs)
- Multiple DRAMs service one Read and Write transaction
- Can tolerate failure of 1 DRAM

Mobile

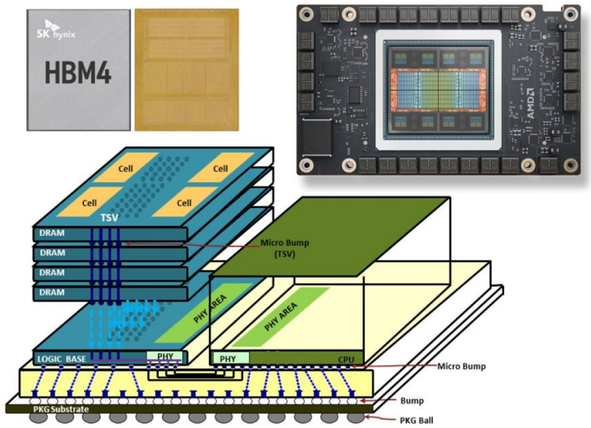
- Low active and idle power with special low power modes
- Fast wake up and power down supports bursty system activity
- Low profile supports stacking with processor die

Graphics



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

AI/HPC

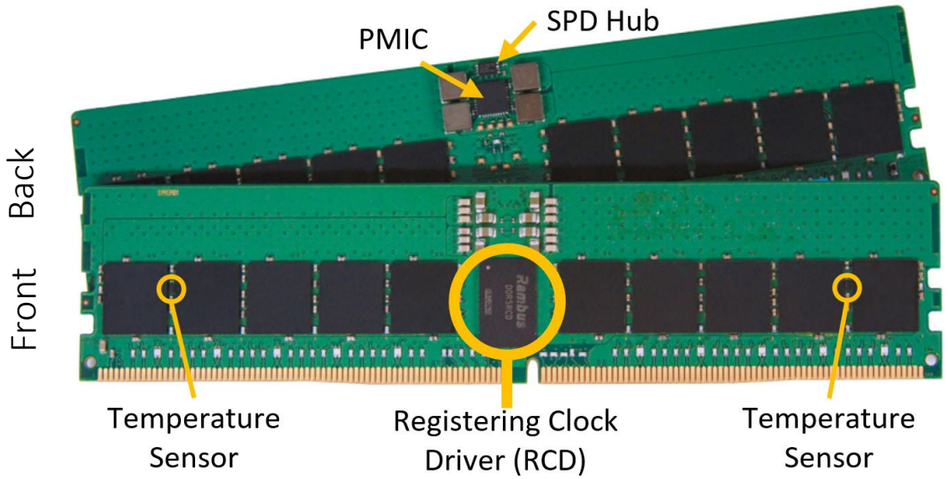


- Highest bandwidth DRAM using die stacking and base die
- Highest number of DQs
- Packaging uses an interposer to support high IO count

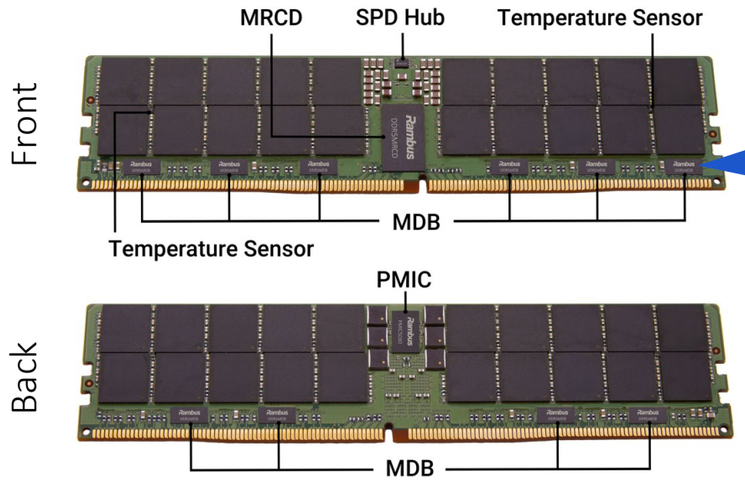
Several types of DRAMs, all use the 1T1C bit cell to store data

Compute Memory: DDR

RDIMM

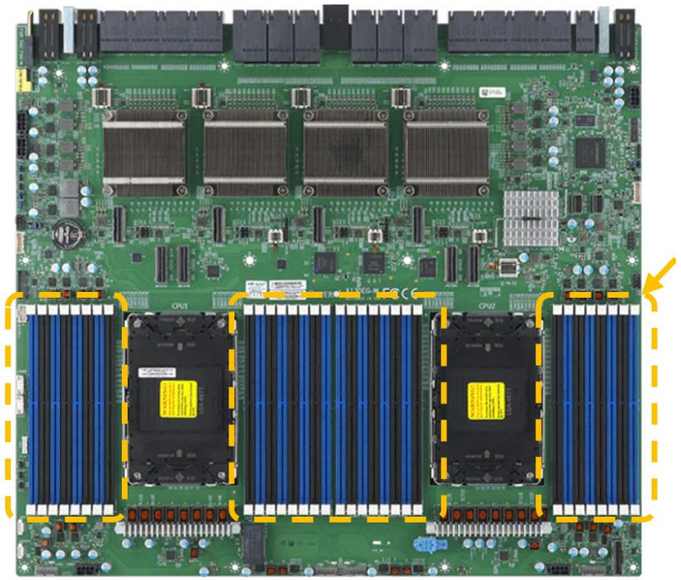


MRDIMM

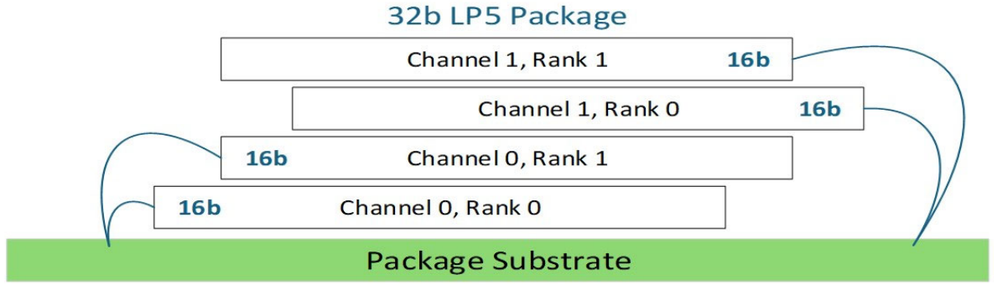


MDBs mux and de-mux two 6.4GT/s DRAM data to 12.8GT/s host interface

- 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

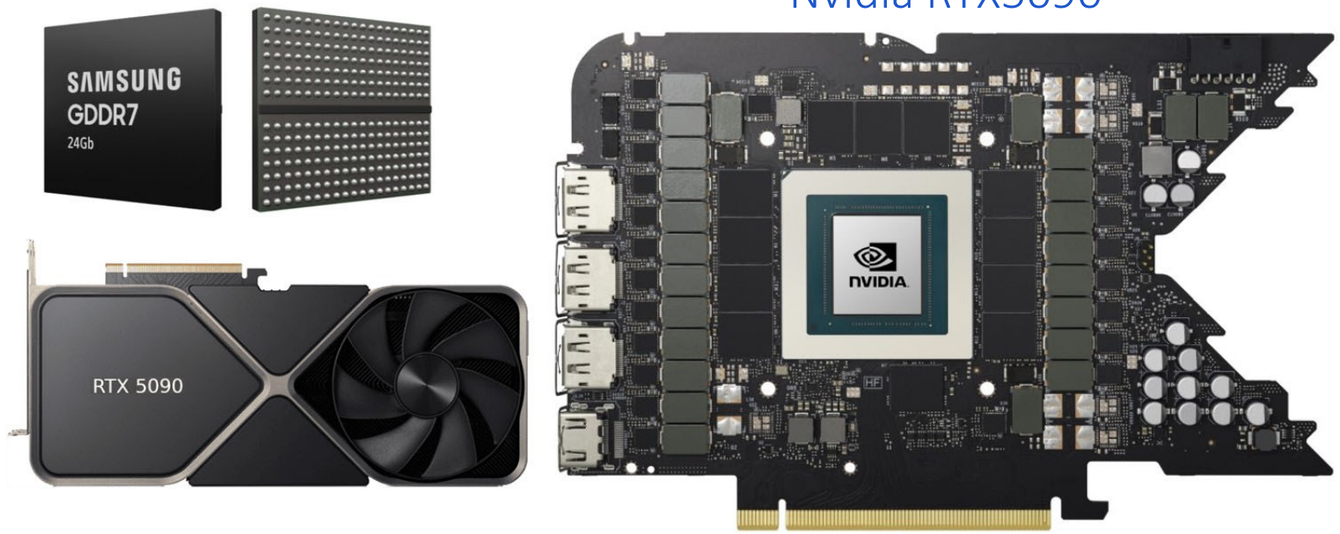
Palm Treo Pro with DDR (2008)

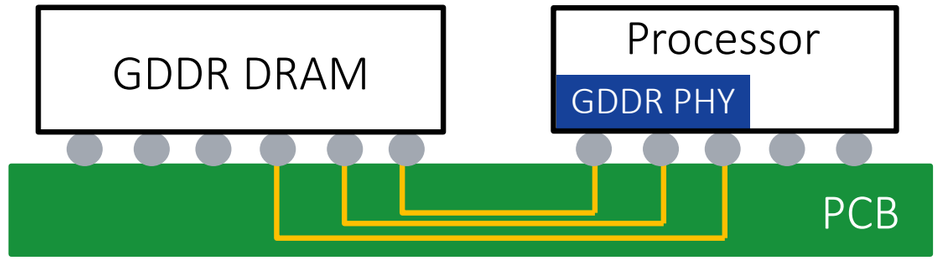
iPhone 4G with LPDDR2 (2010)

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

Graphics Memory: GDDR

Nvidia RTX5090

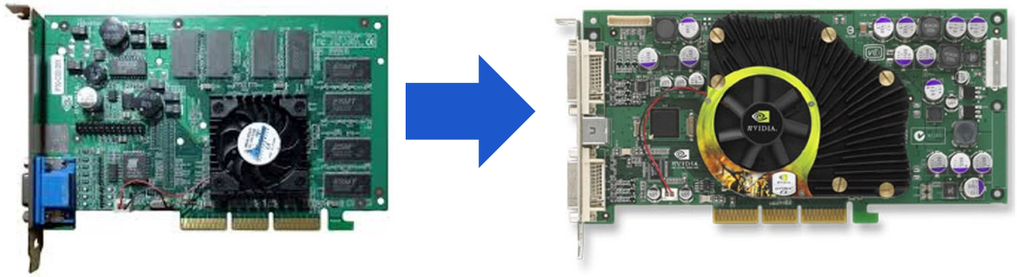




- Highest data rates, challenging signal integrity
- GDDR7: 4 8b channels = 32 DQs, up to 48Gbps (PAM3), up to 192GB/s per DRAM
- High fill frequency, good for graphics, some AI & HPC apps
- Clamshell mode offers 2X Capacity at same bandwidth
- Standard manufacturing with DRAM soldered to PCB
- Good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

nVidia GPU with DDR (2000)

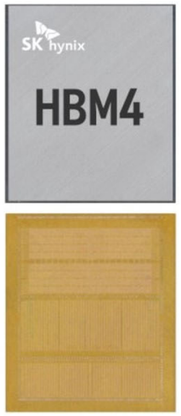
nVidia GPU with GDDR2 (2004)

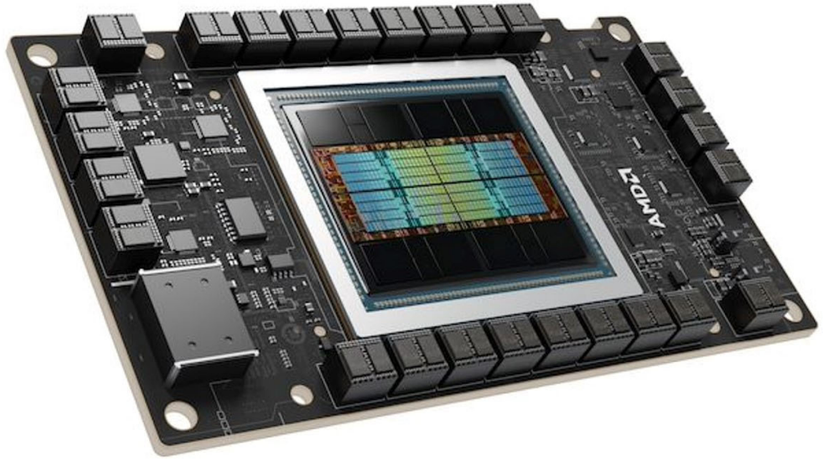


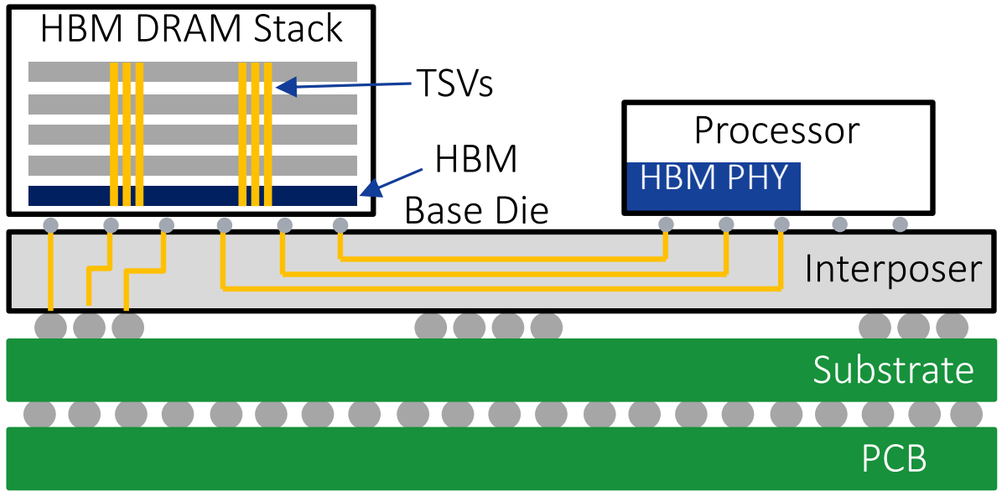
Early GPUs used binned DDR DRAMs, growing volumes justified development GDDR

AI/HPC Memory: HBM

AMD MI325X







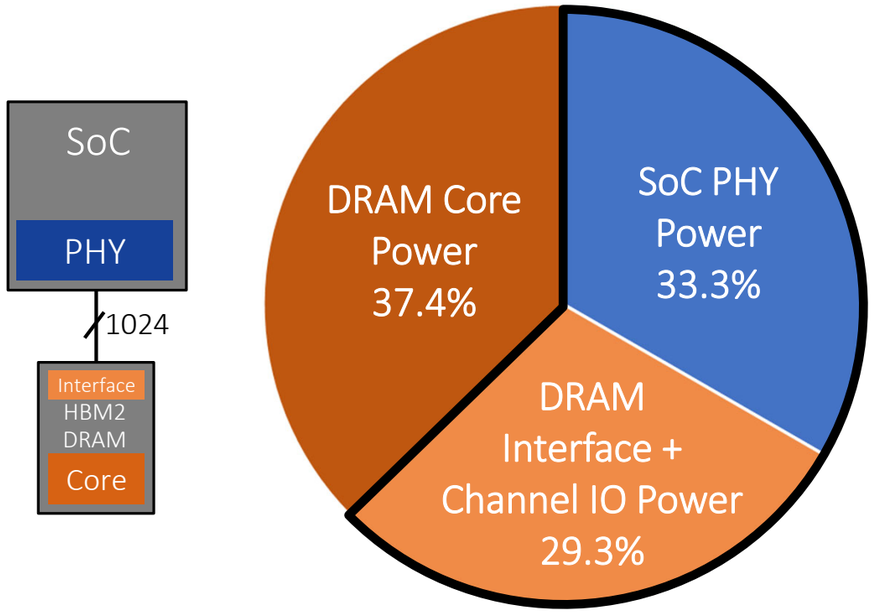
- Many channels, high DQ count
- HBM4: 32 64b channels = 2048 DQs, up to 10Gbps data rates
- Highest bandwidth of any discrete DRAM, high fill frequency (up to 2TB/s with HBM4, high total power but extremely good power efficiency)
- Base die provides connectivity between DRAM die in stack and interposer
- 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->HBM3E->HBM4)
- SoC power budgets: 40%+ used for high-speed memory interfaces (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

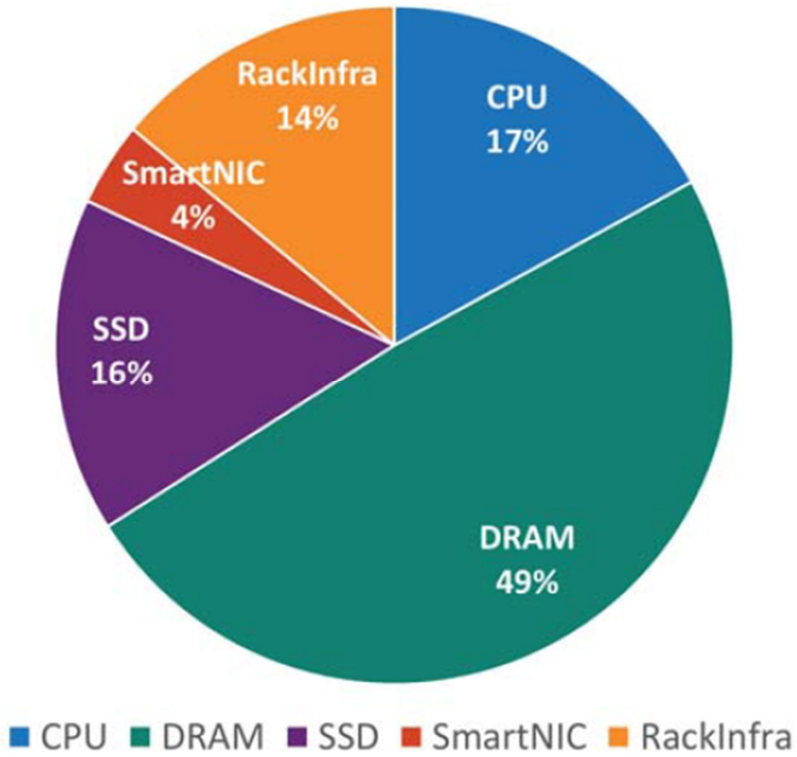


Data Movement: 62.6% of Power

Source: Rambus Inc.

Memory Can Be a Major Fraction of System Cost

Compute Node Component Costs

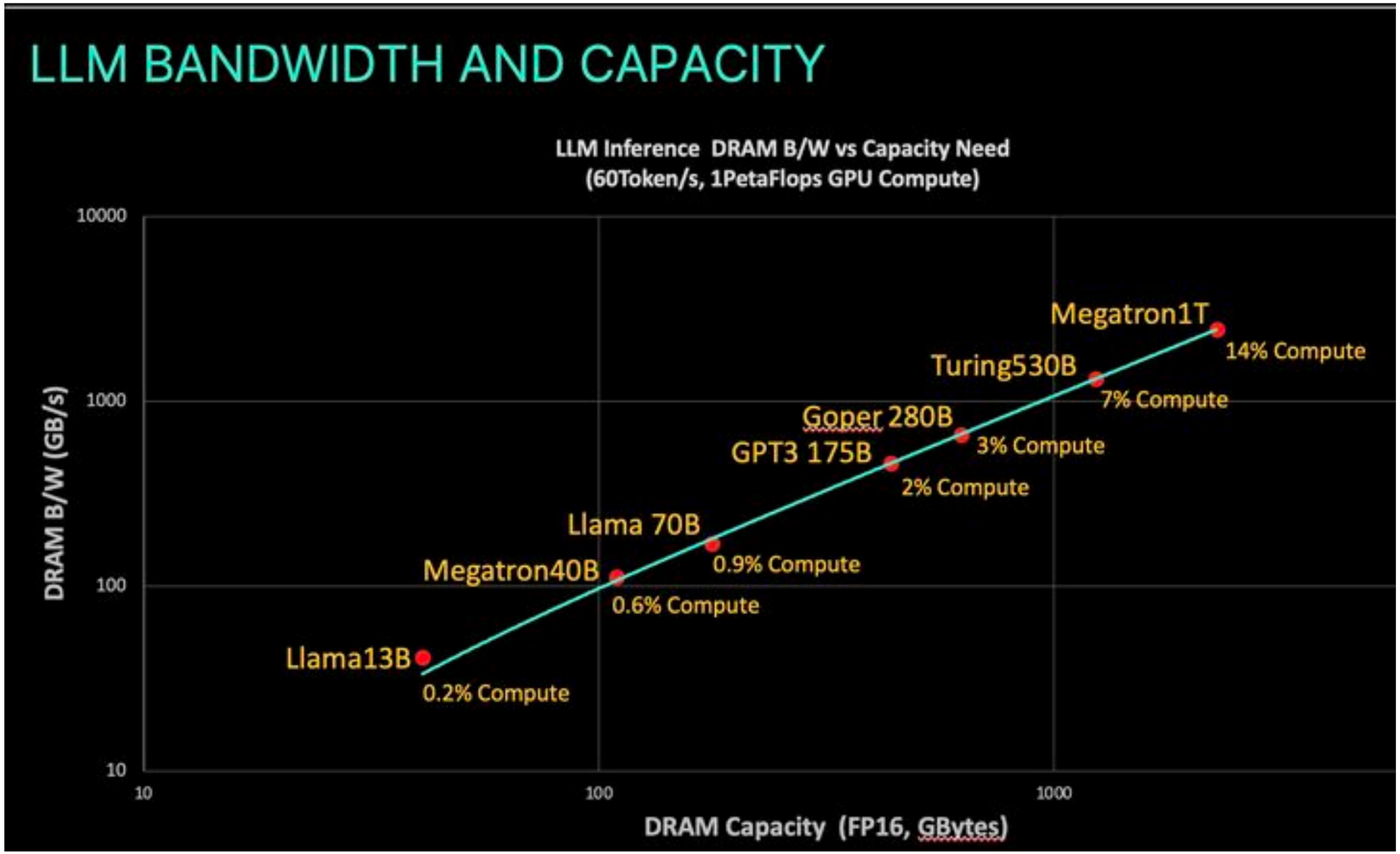


Source: Decadal Plan for Semiconductors, Semiconductor Research Consortium (SRC), January 2021

- 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

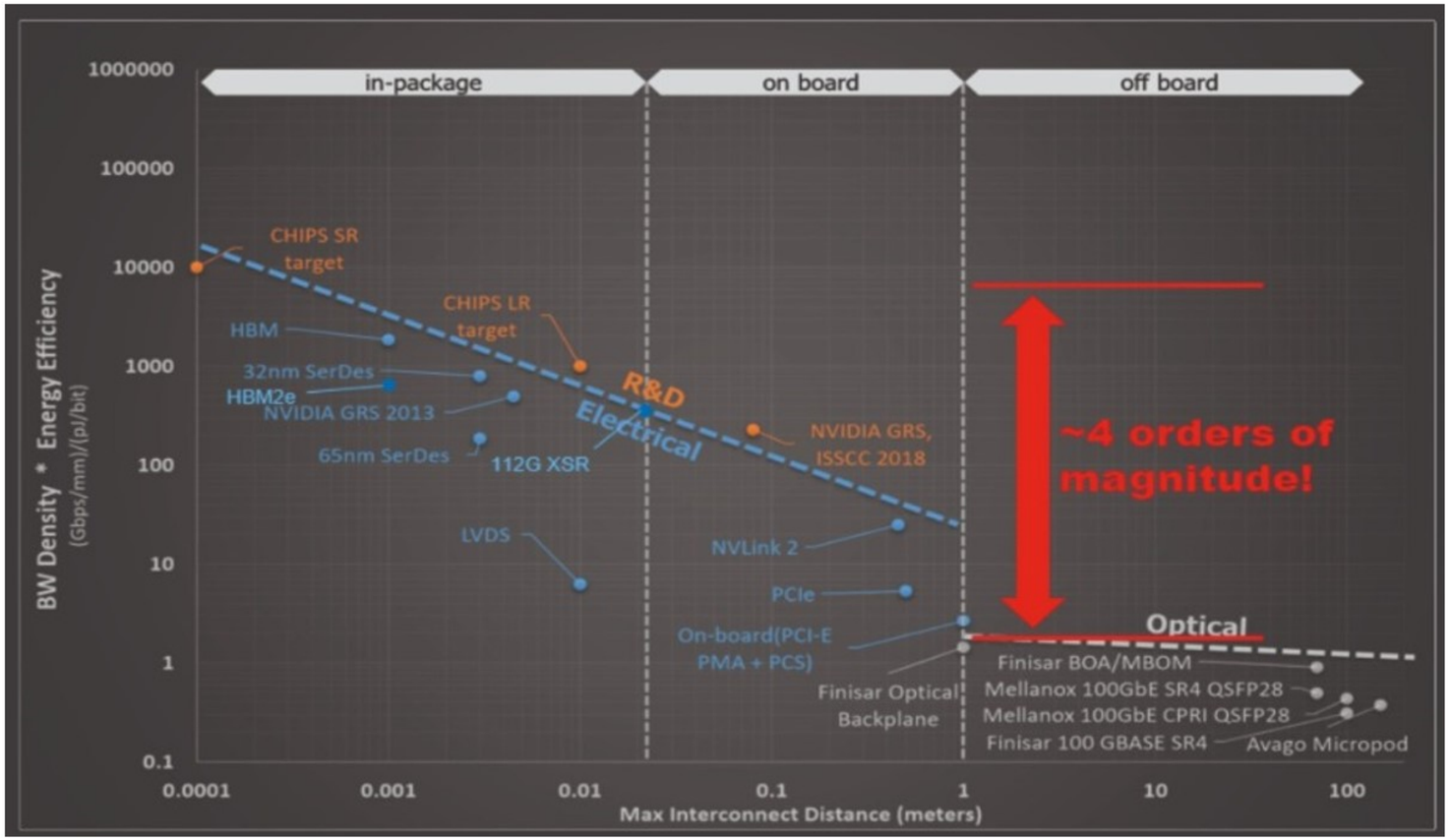
DRAM Bandwidth and Capacity are Key Resources



- As AI models increase in size, both bandwidth and capacity must scale
- Fill frequency critical in LLM inference, need to maintain/grow fill frequency

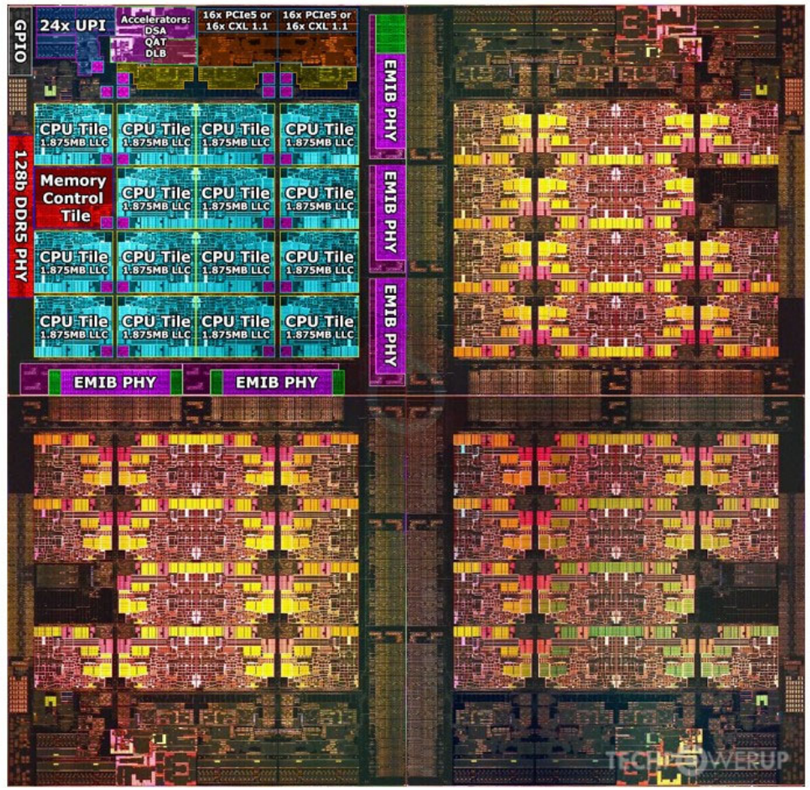
Source: Raja Koduri, https://x.com/RajaXg/status/1758938708601180577

Distance (Reach) is Power, Shoreline Critical



Source: G. Keeler, DARPA ERI 2019

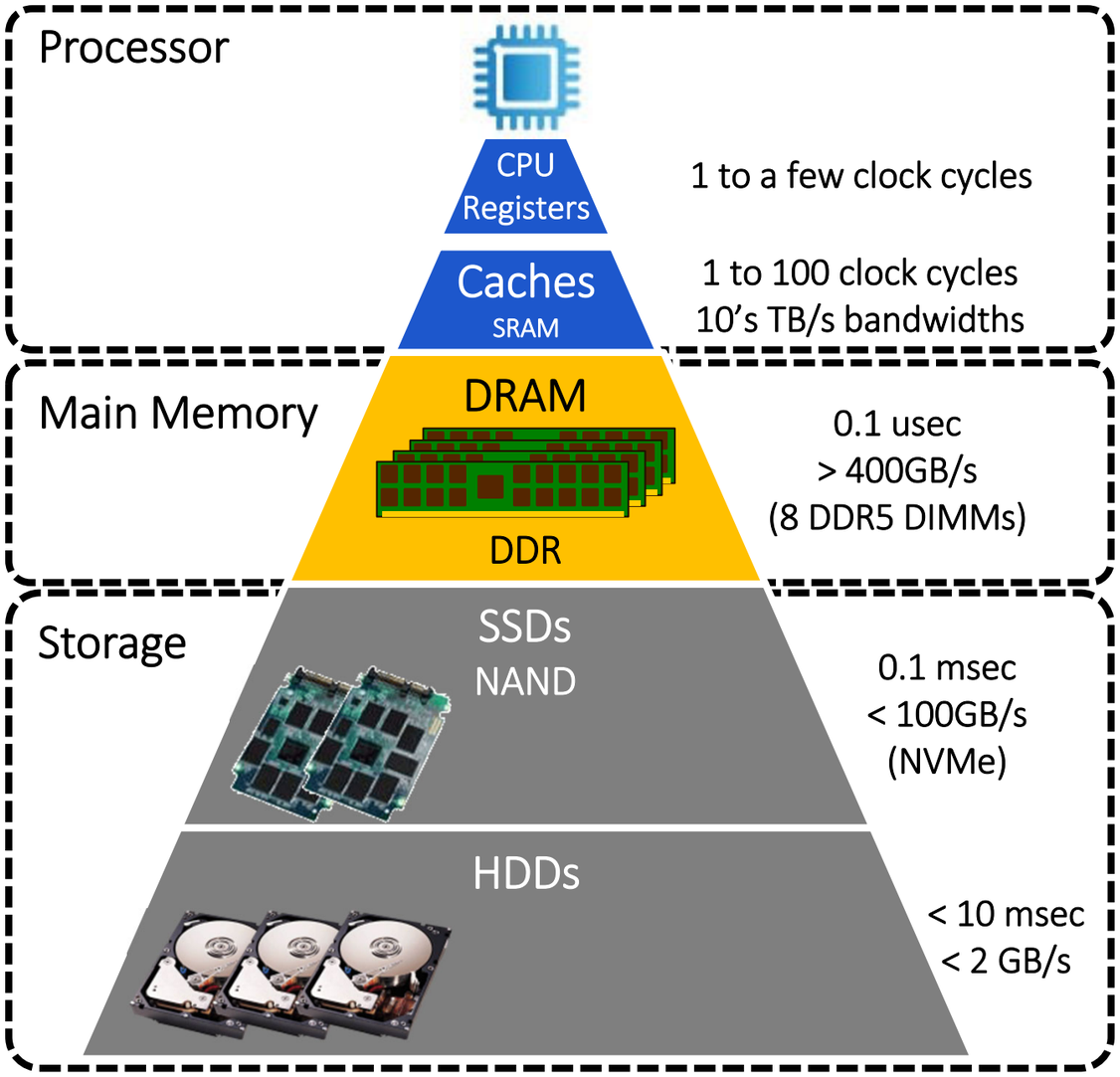
Intel Xeon Max 9470

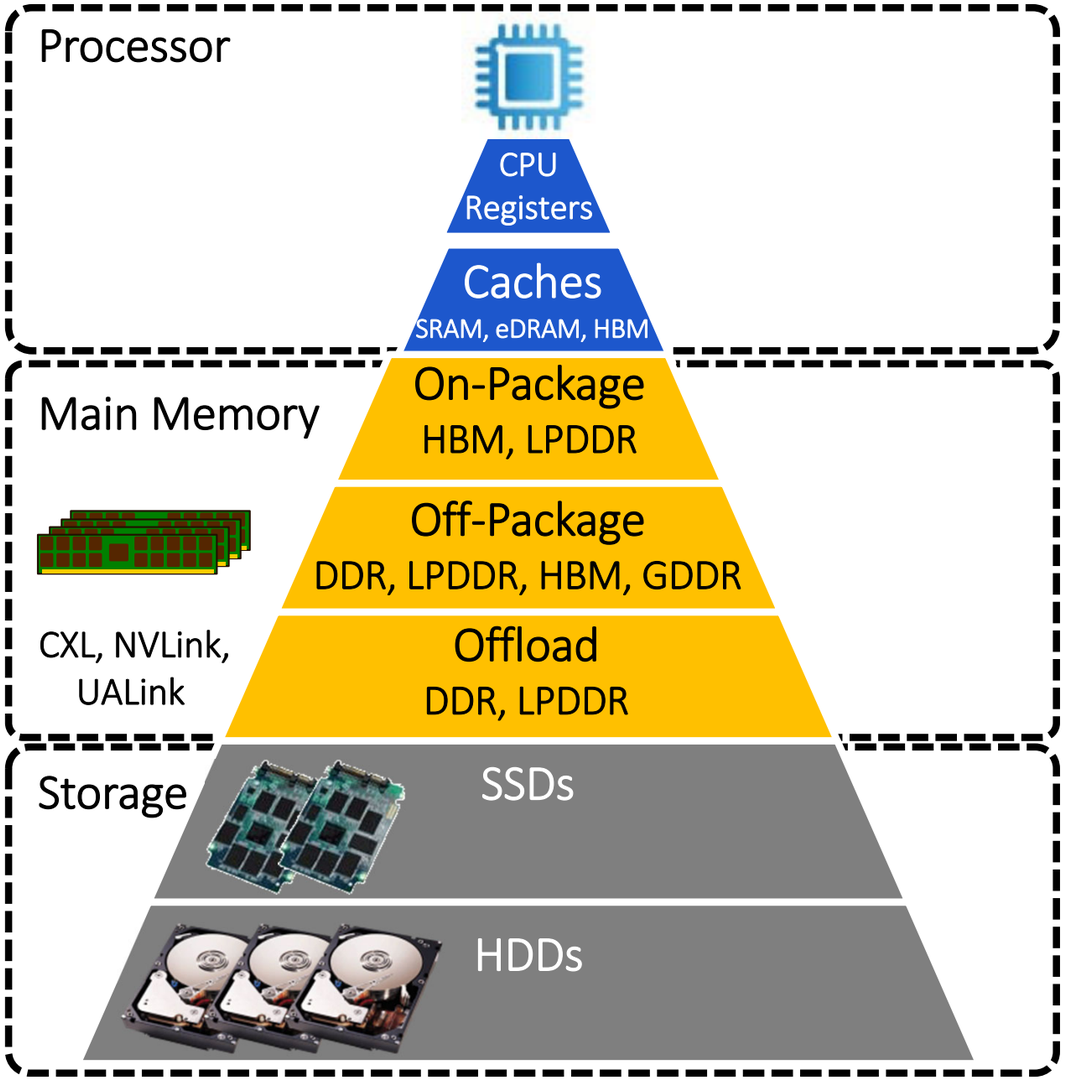


Source: https://www.techpowerup.com/cpu-specs/xeon-max-9470.c3085

- 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

DRAM Architecture:
Cell, Array, Data Path and Interface

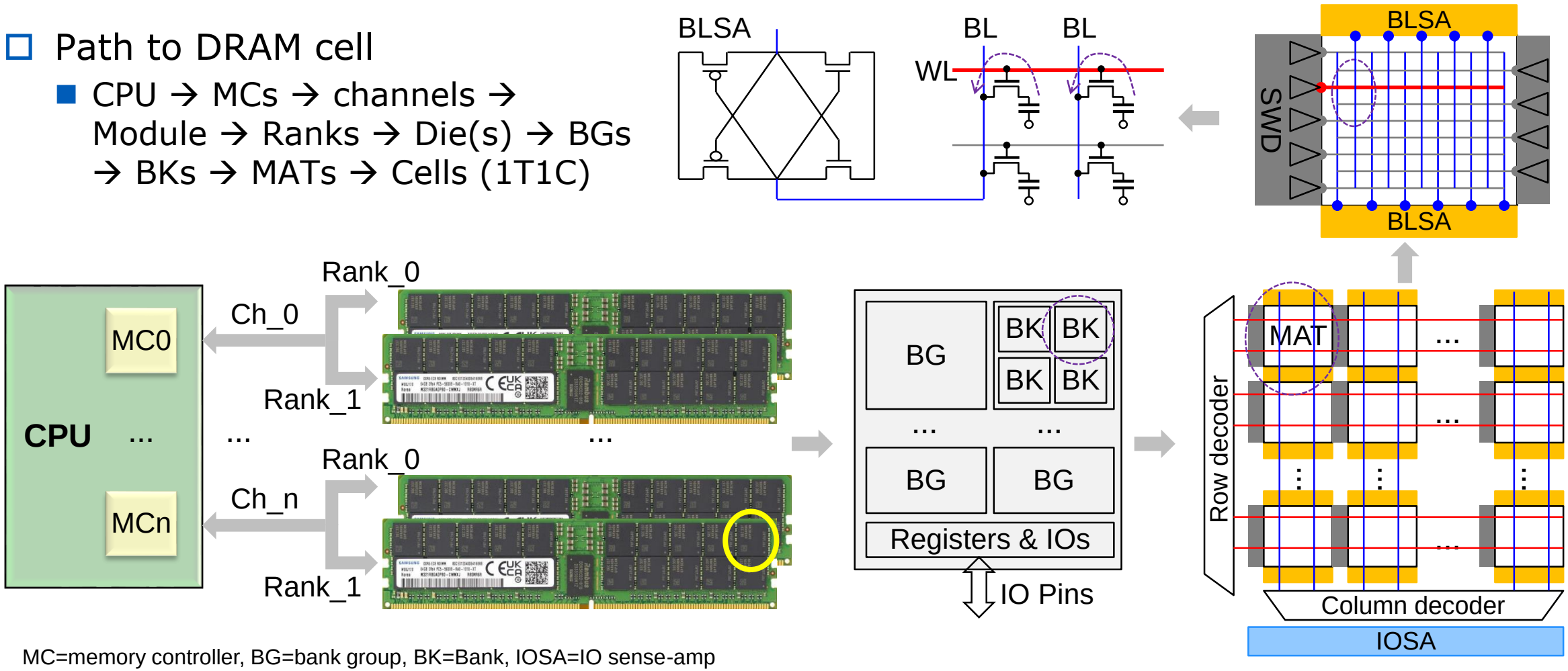
Taeksang Song, Ph.D.
Corporate VP of DRAM Solution Engineering,
Samsung Electronics

Outline

- Introduction
- DRAM Core & Peri Architecture
  - Cell Operation & Timing Parameters
  - Sensing Margin – Leakage & Offset
- DRAM Cell Scaling
  - Vertical Stacked DRAM & Cell-over-Peri
- DRAM Interface
- Summary

DRAM Architecture

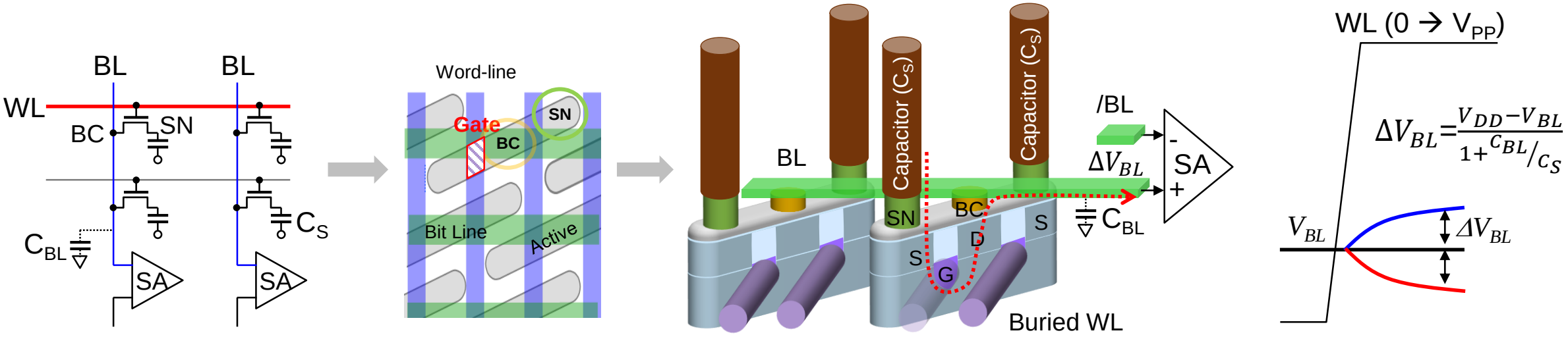
- Path to DRAM cell
  - CPU → MCs → channels → Module → Ranks → Die(s) → BGs → BKs → MATs → Cells (1T1C)



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

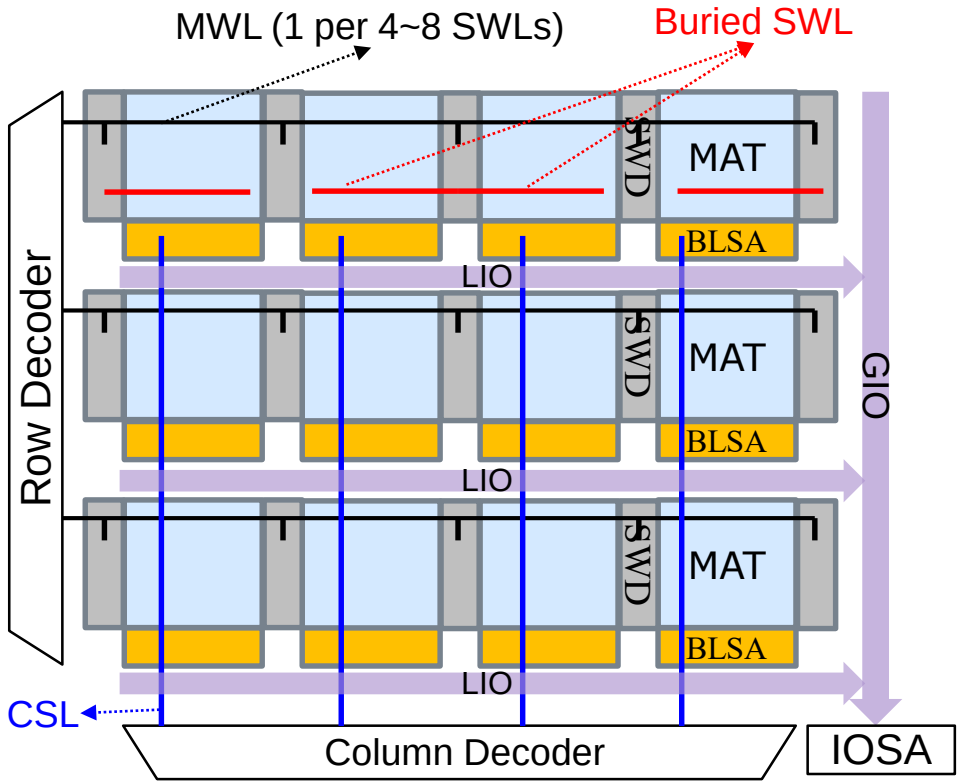
- MAT
  - Unit block which has cell-array (e.g., >1K WL * >1K BL), sub-WL driver & BLSA
  - Determines cell efficiency (cell area / chip area, ~50%) & sensing margin
    - Large array mitigates SWL driver & BLSA area overhead → high capacity, low cost
    - But, reduce sensing margin due to larger parasitic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



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

Bank Configuration

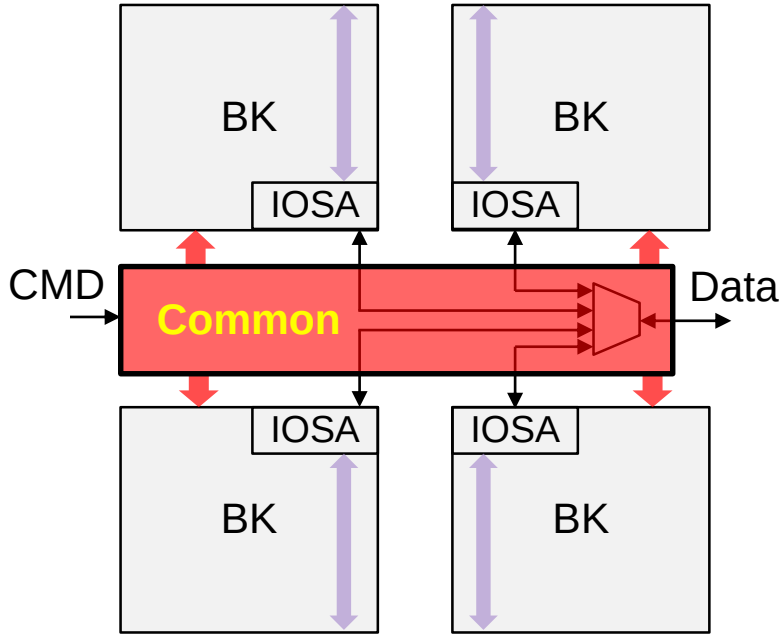
- Bank
  - Array of MATs + control circuits
  - Can activate only one row/page at a time, but can activate row/page independently and separately from other banks
- Configuration
  - Row decoder
    - Activate only one page (=one row of cells)
    - 1KB~2KB cells are activated and sensed by BLSA
  - Column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

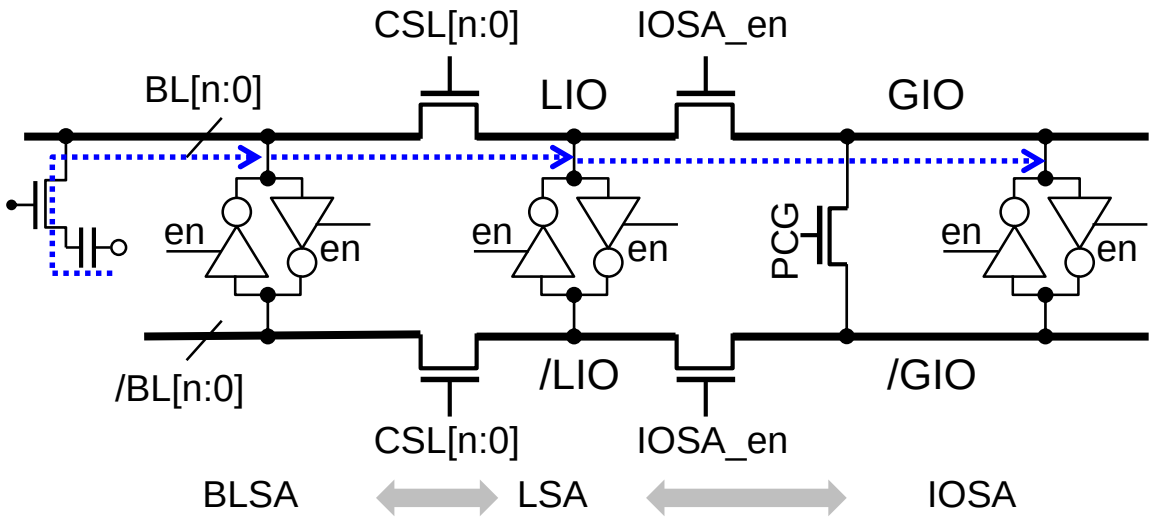
Bank Group Configuration

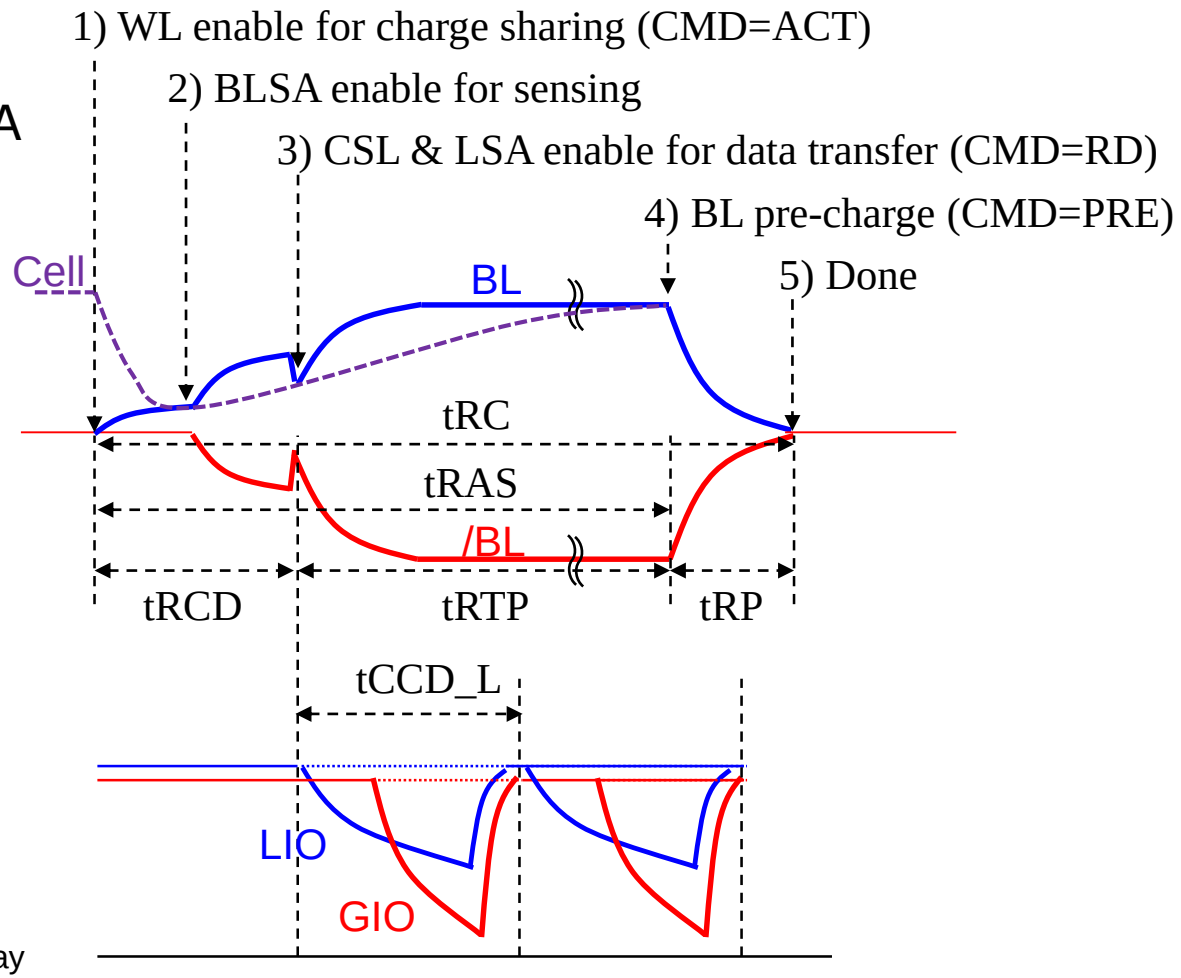
- Bank Group
  - A unit that operates independently
  - Share common blocks
    - Timing control
    - Data & command lines
- Restrictions
  - tCCD_L
    - Because of the shared common blocks, while read or writing data from one bank (tCCD_L), other banks within the same BG can't be read or written



Core Timing Diagram

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

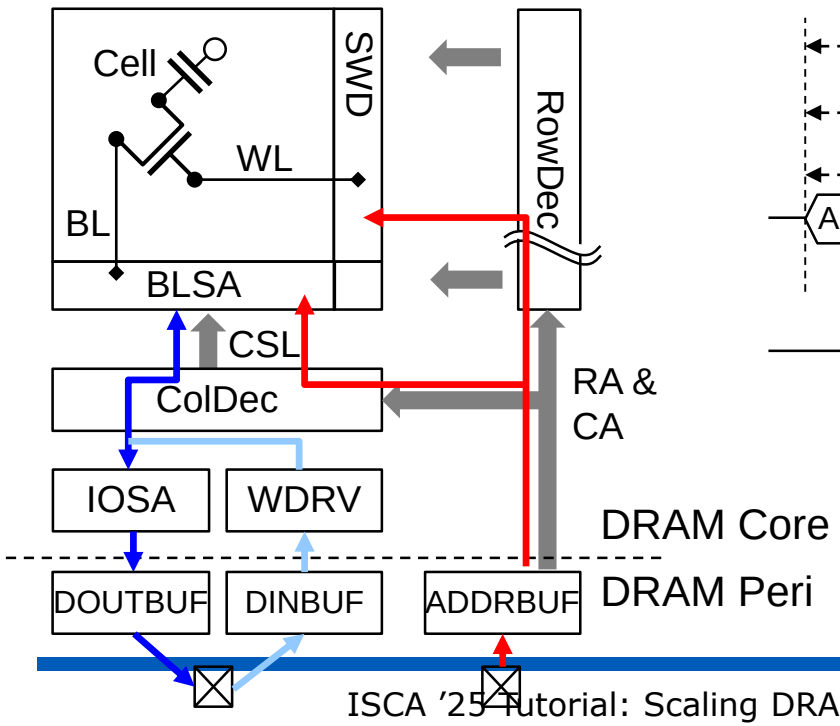


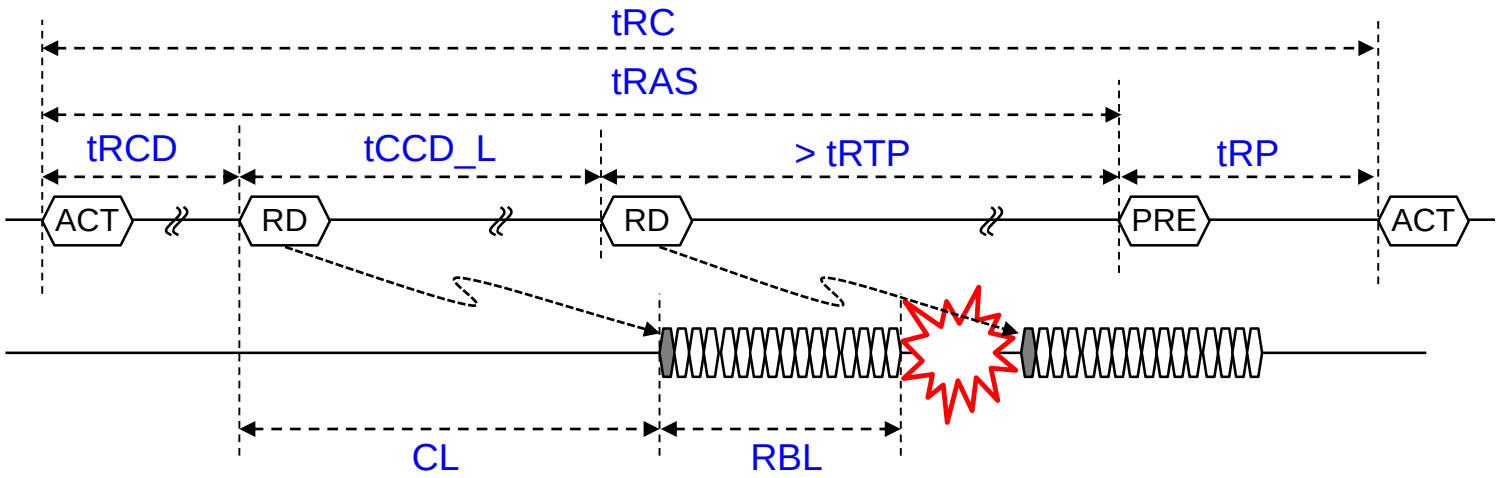


tRC= row cycle time, tRAS=Row Active to Precharge Delay, tRCD=Row-to-Column Delay
tRTP=Read to Precharge Delay, tRP=Row Precharge, tCCD=Column-to-Colunm Delay

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

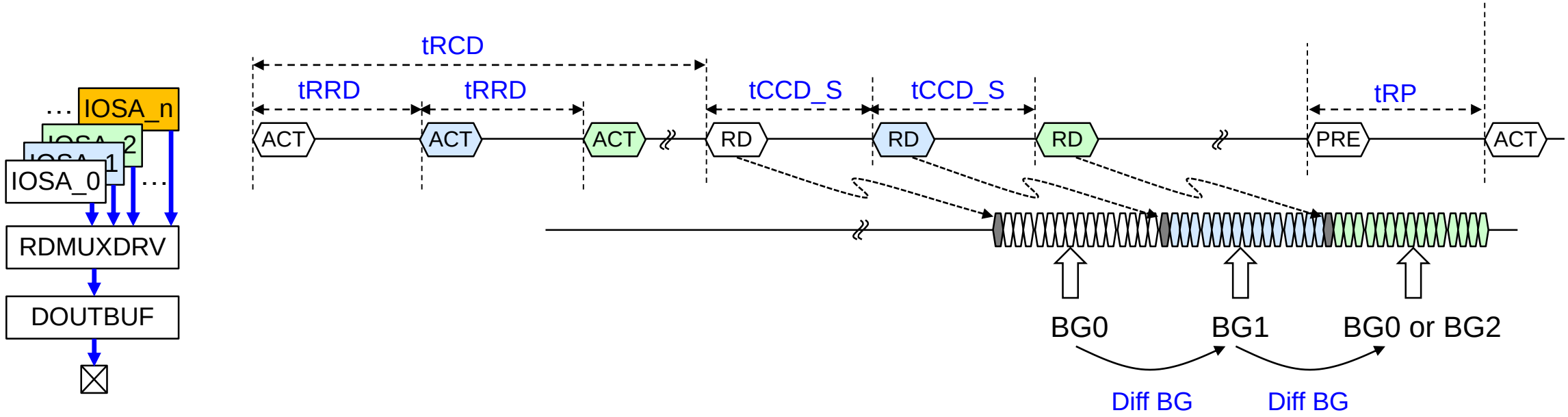




WDRV=Write driver, CL=CAS Latency, RBL=Read Burst Length

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

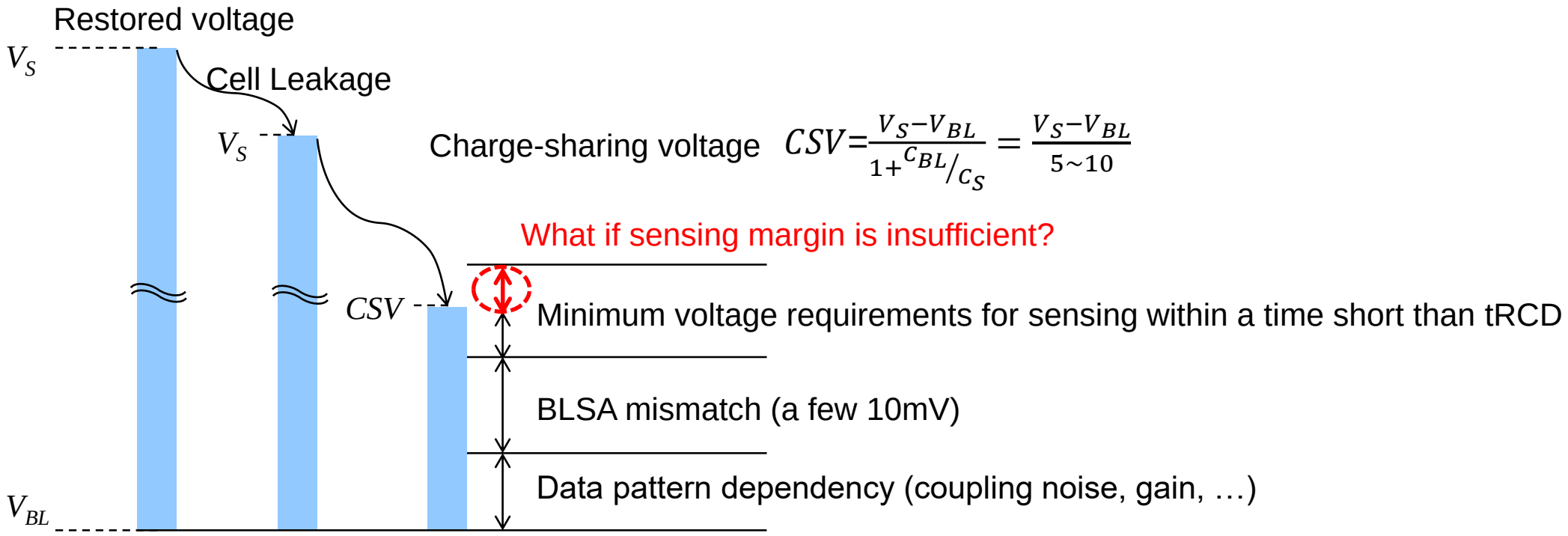


Core & Data Path Timing Parameters

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	

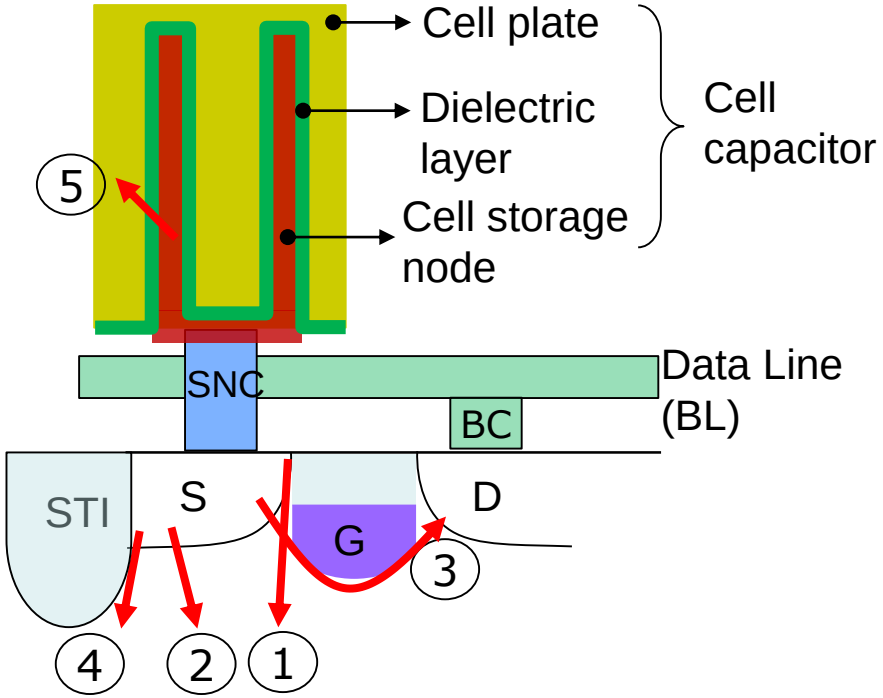
Sensing Margin - Overview

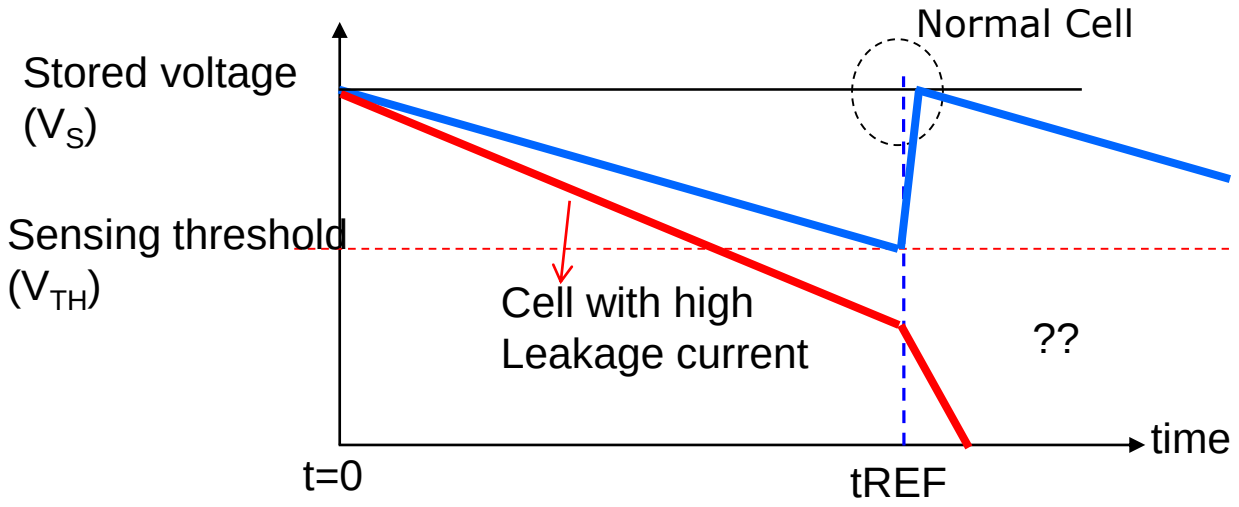
- What factors determine sensing margin
  - Charge Sharing voltage ← cell leakage, C_{BL}/C_S ratio, stored charge (V_{DD} & V_{BL})
  - Sensing margin ← BLSA mismatch, noise



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

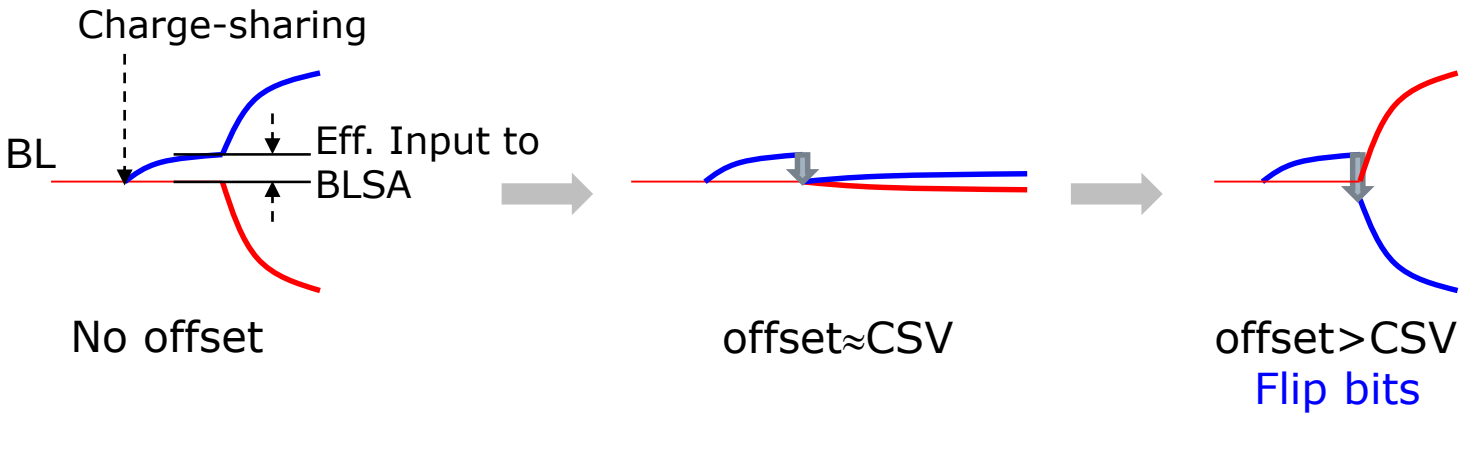


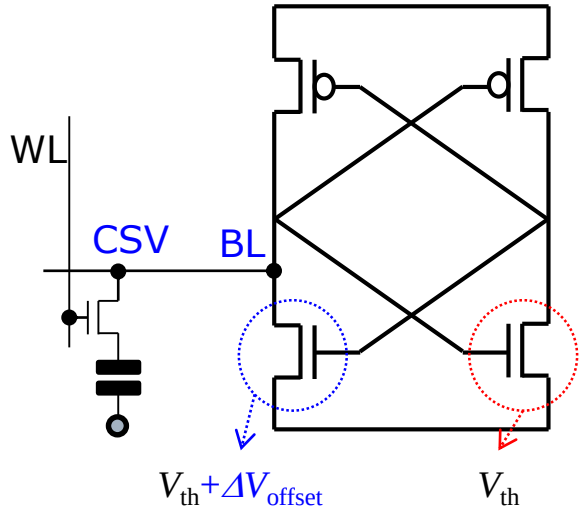


$$t_{REF} = \frac{C_S}{I_{Leak\_Total}} \cdot \left\{ V_S - \left(1 + \frac{C_{BL}}{C_S} \right) \cdot V_{TH} \right\}$$

Sensing Margin – BLSA Offset

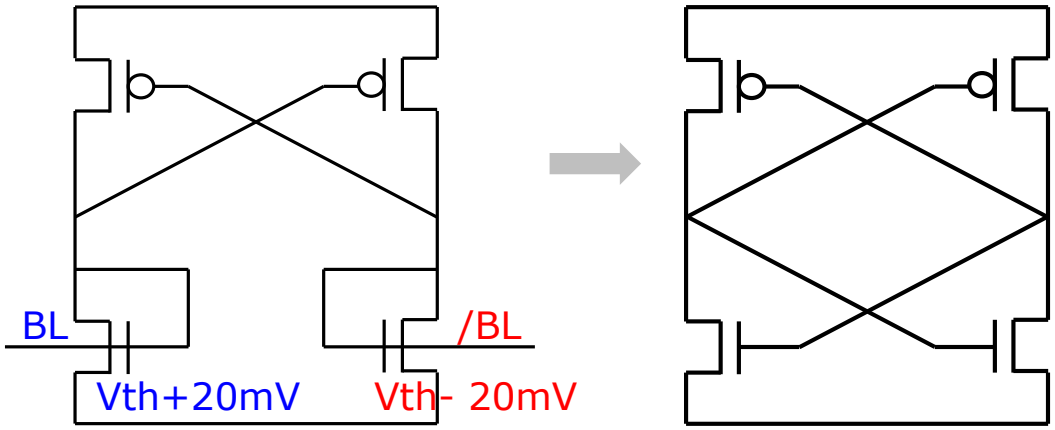
- Sources of BLSA mismatch
  - Geometric (W/L), Random dopant fluctuation, Cox
  - As technology scales down, mismatch increases seriously
- BLSA offset
  - A few 10mV
  - Makes BLSA develop time long or flips data



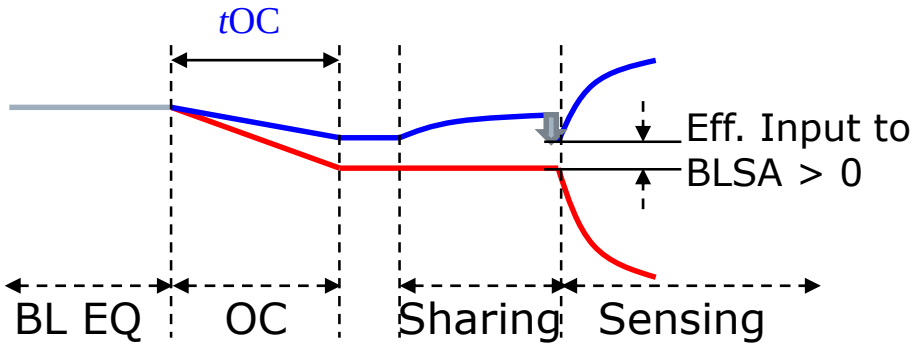


Sensing Margin – BLSA Offset Cancellation

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

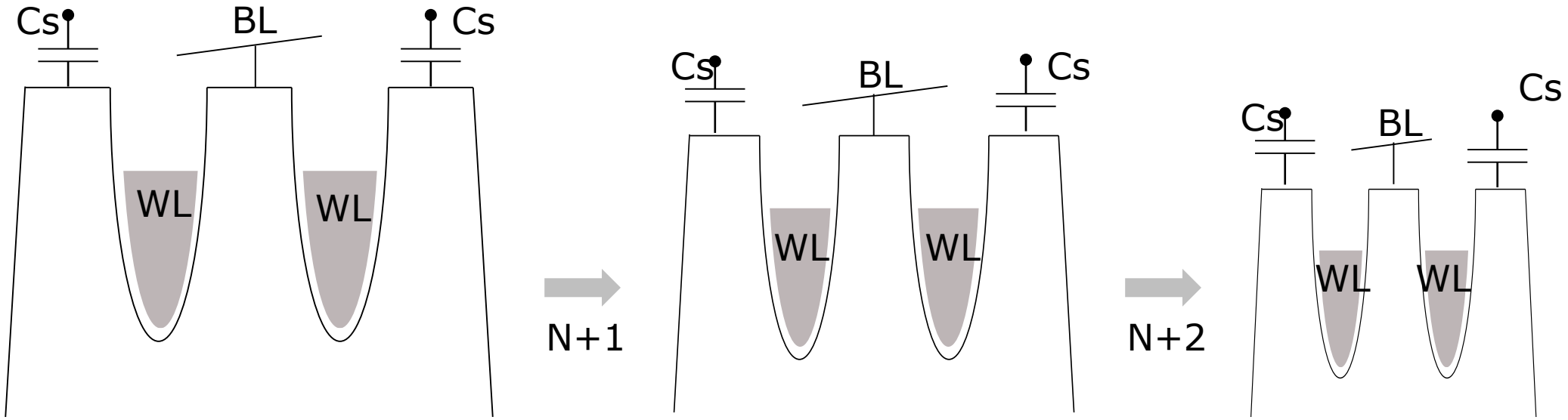


<Offset Cancel> <Sensing>



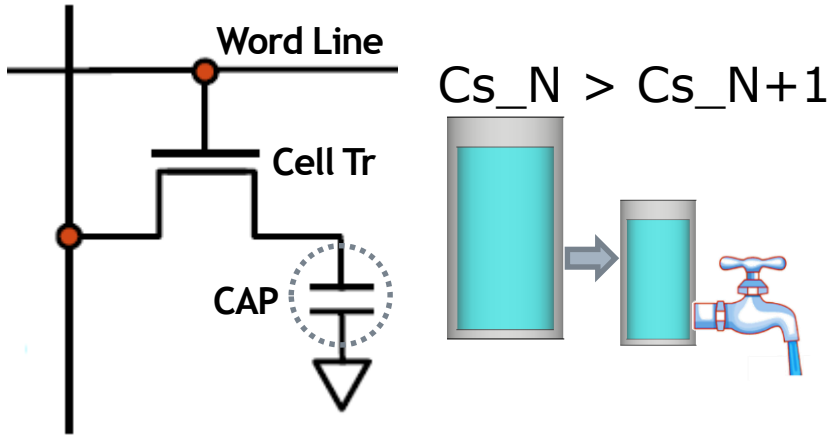
Sensing Margin – DRAM Scaling

- 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.

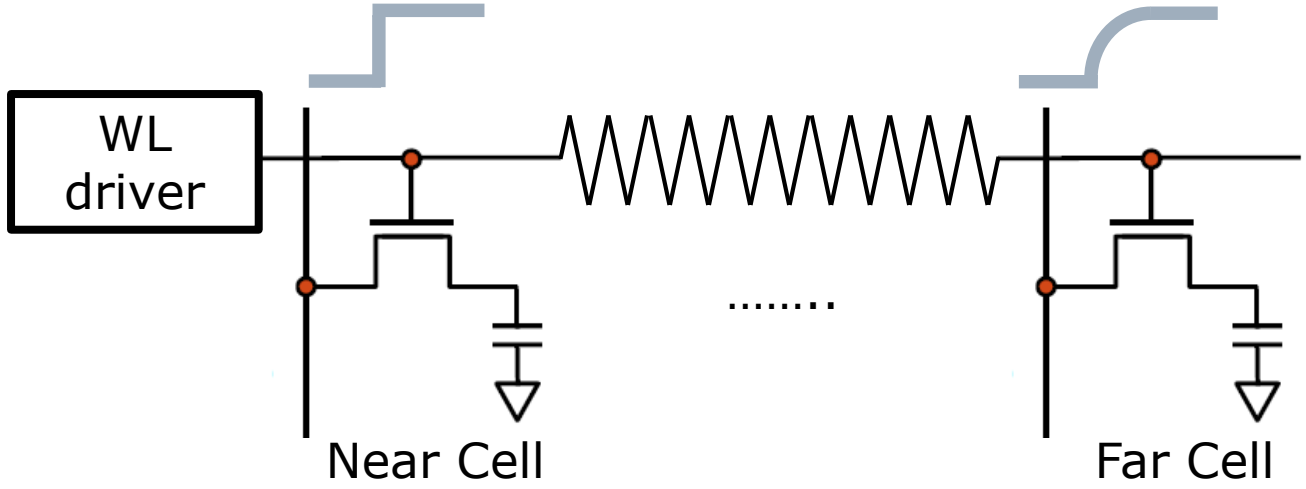


Sensing Margin – Scaling Challenges

- 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



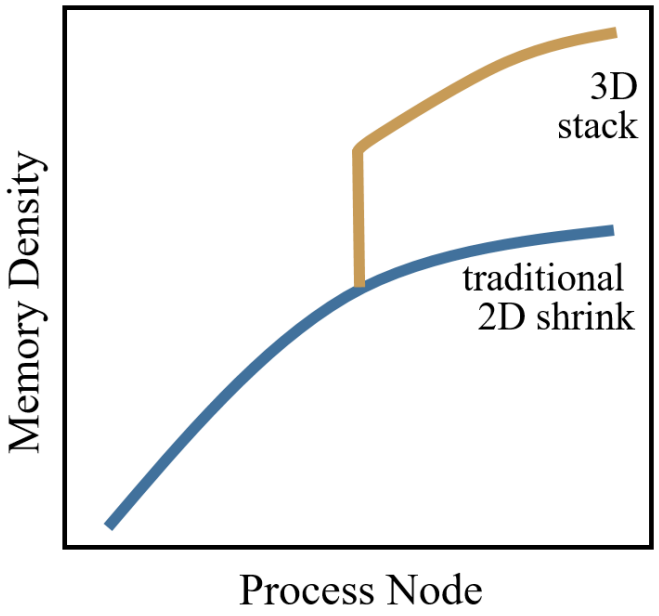
Cs_N > Cs_N+1



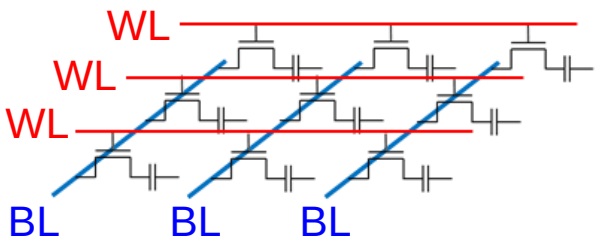
$$R_{WL} = \rho_{WL} \times L / H \cdot W$$

Scaling – Vertically Stacked (VS) DRAM

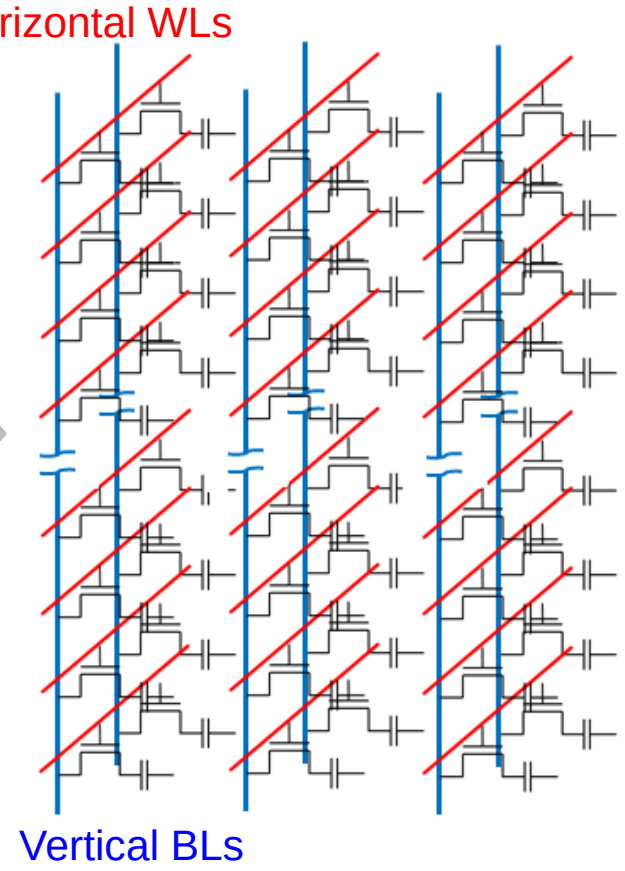
- Vertically stacked 3D DRAM can relax the dimension requirement via z-dimension



2D →
3D Stacking



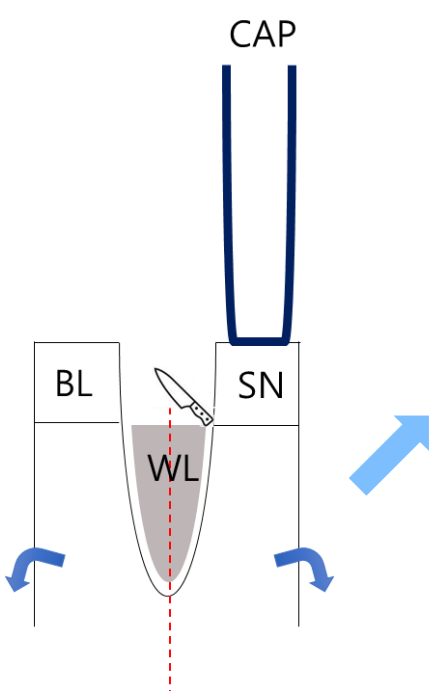
2D planar array

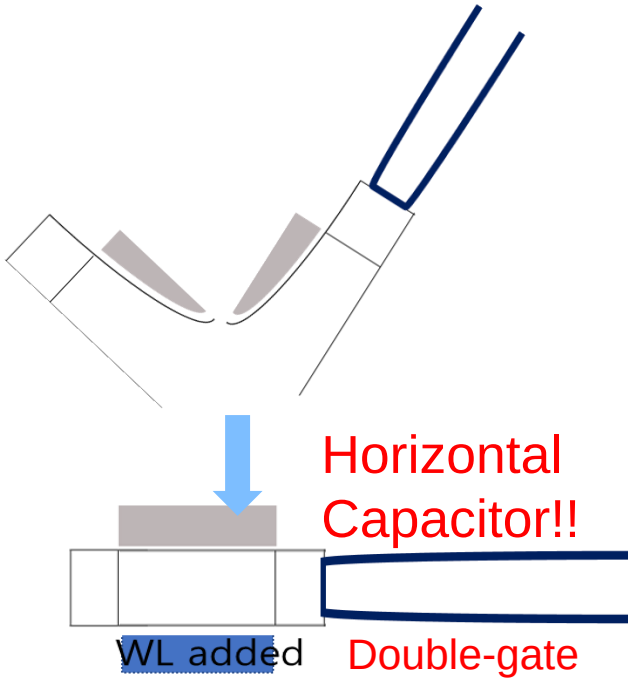


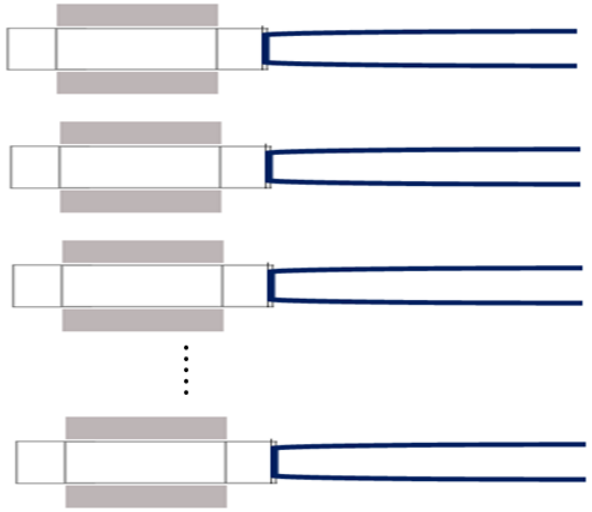
3D cubic block

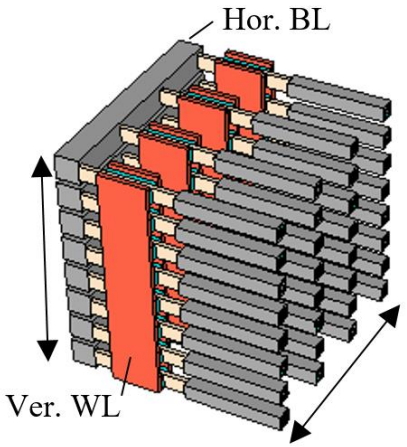
Scaling – VS DRAM Concepts

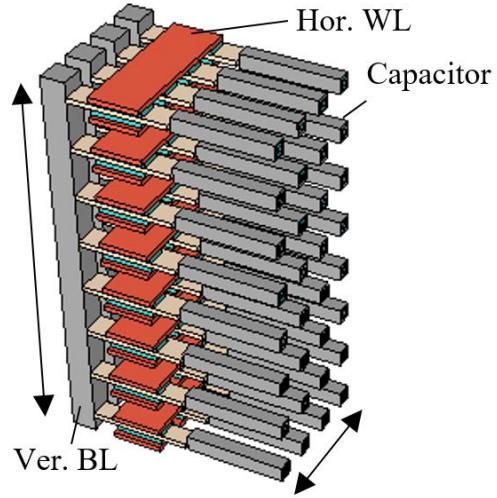
- Structure
  - Horizontal capacitor
  - (horizontal BL, vertical WL) or (vertical BL, horizontal WL)





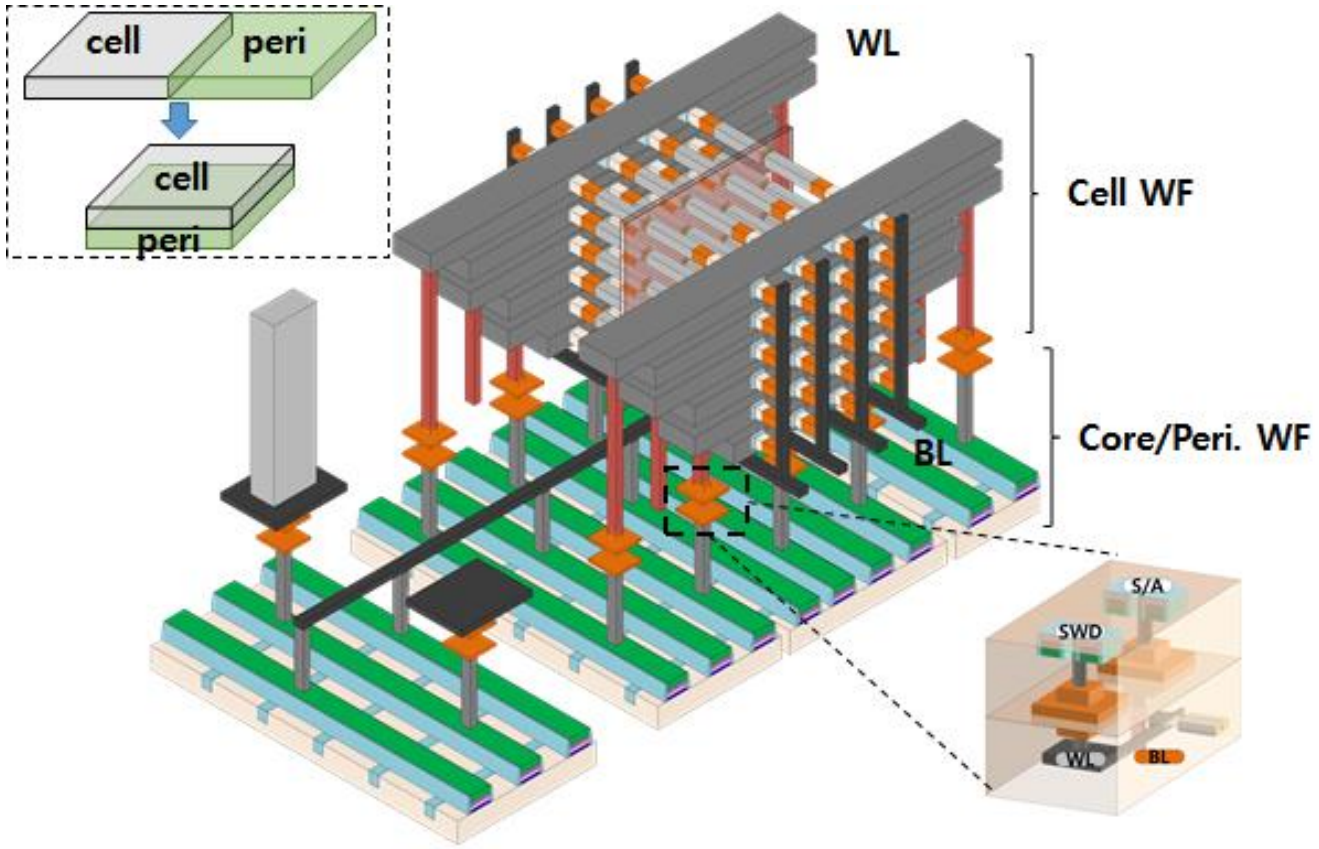






Scaling – Cell Over Peri Architecture

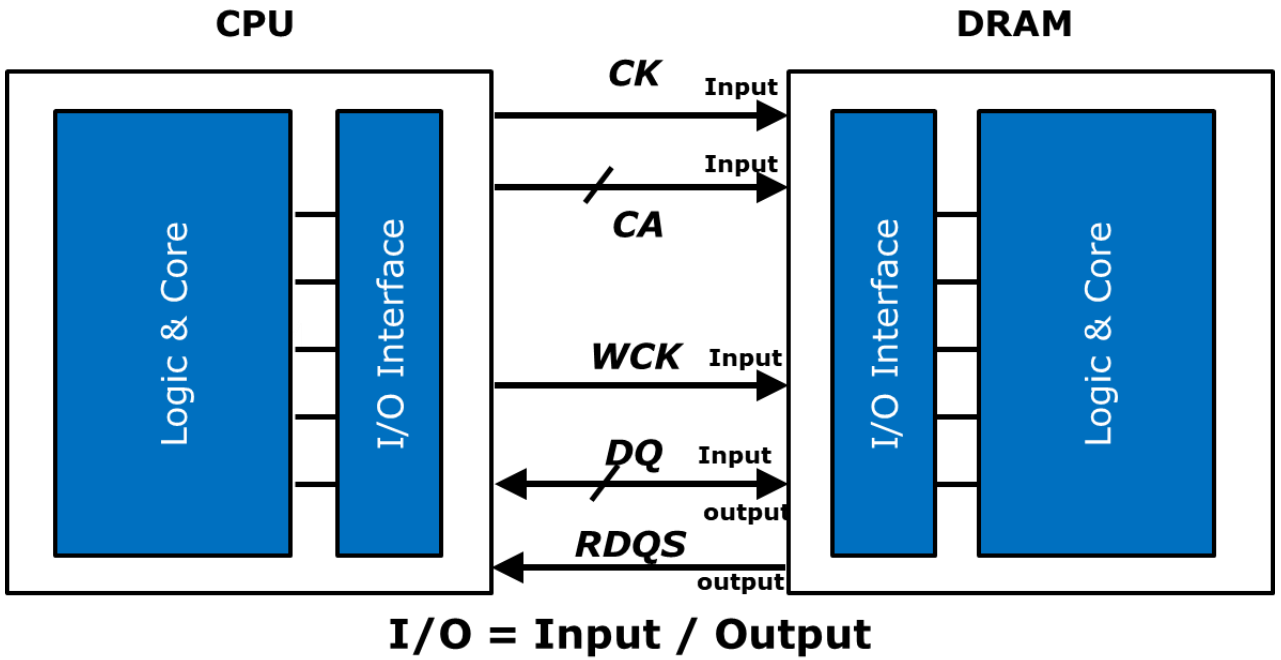
- Cell WF and Core-Per WF are separately fabricated and WoW bonded



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**

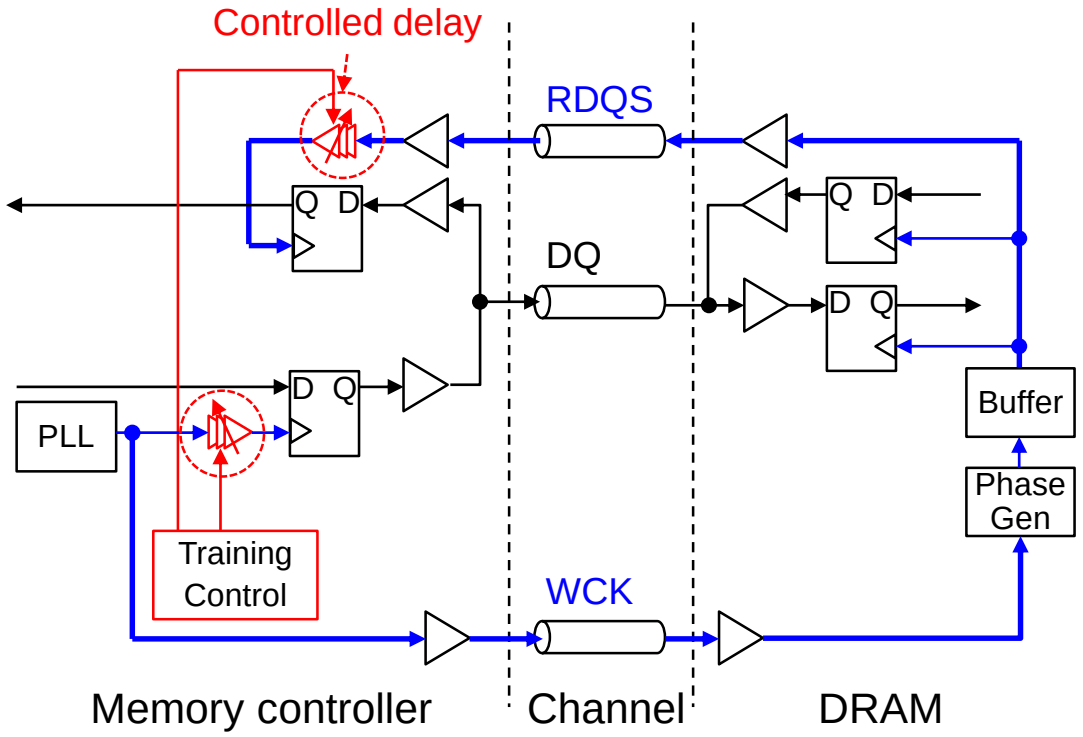
✓ CK : Clock for command & address
✓ CA : Command and Address
✓ WCK : Clock for data
✓ DQ : Data transfer
✓ RDQS : Read strobe clock

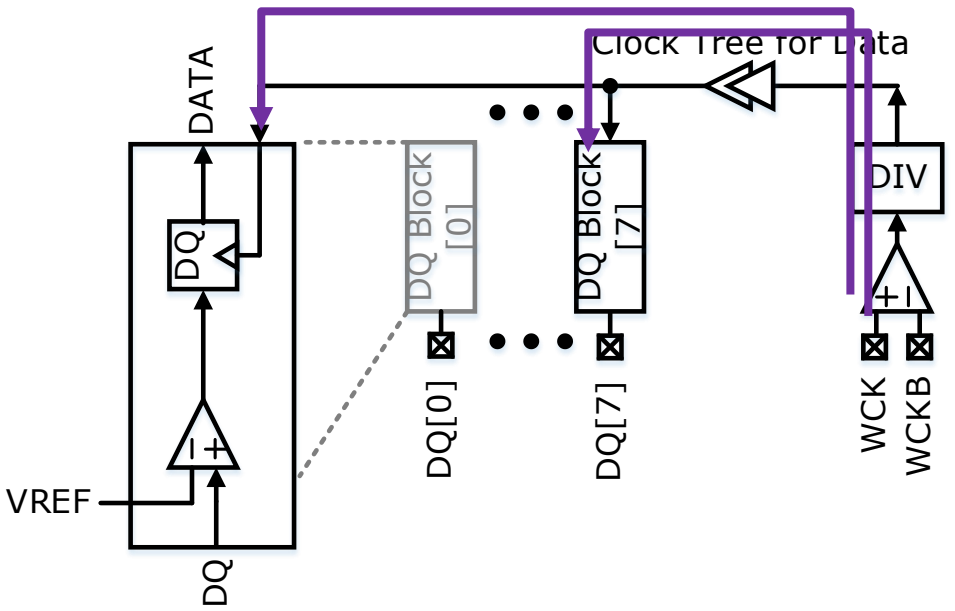
** CA and DQ show parallel IO configurations.

- 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).

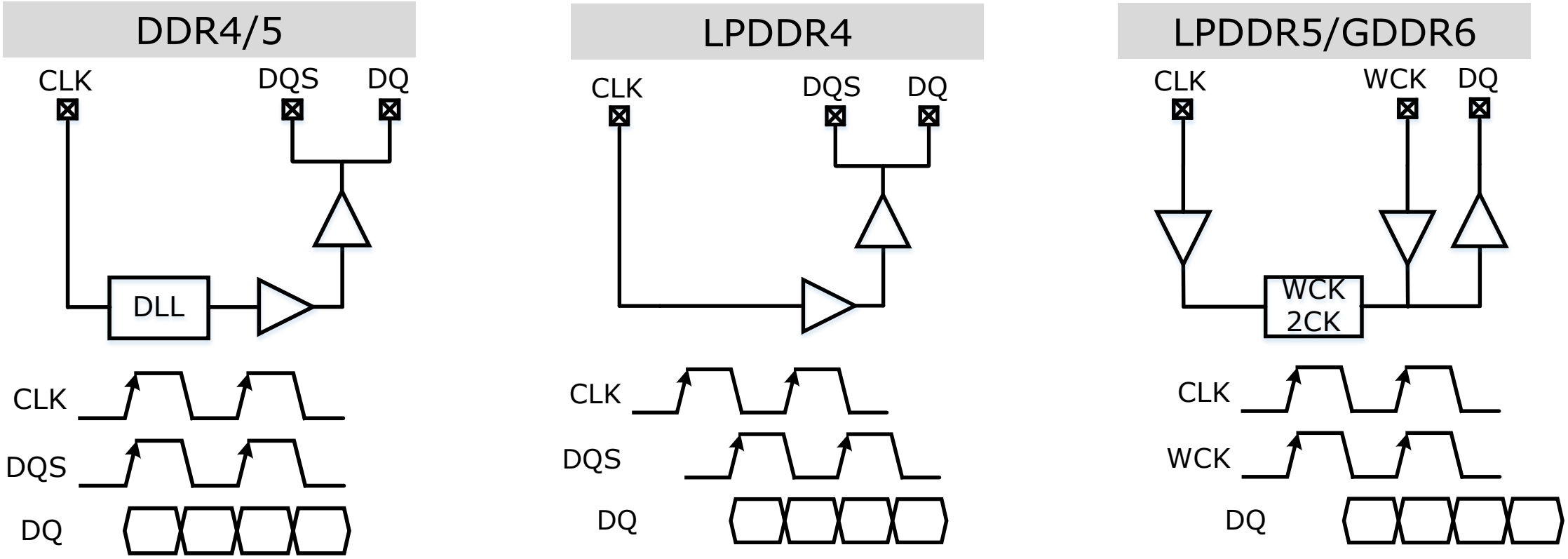
DRAM Clocking & Strobe

- 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





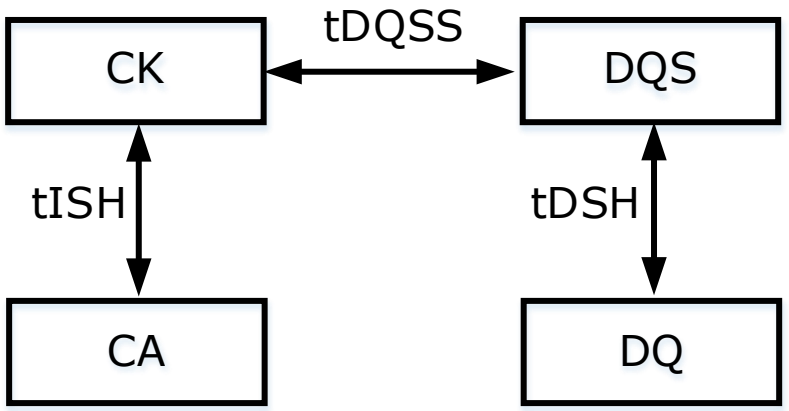
Tx Clocking



- Different timing relationship between CLK, DQS (WCK), and DQ for each applications
- DLL guarantees the aligned edge of DQS to CLK

DRAM IO Training

- The timing relationship between DRAM I/O pins are initially optimized through training operations (Mandatory for high-speed DRAMs).
- The timing constraints to guarantee the reliable operation of DRAM are defined in the specification.

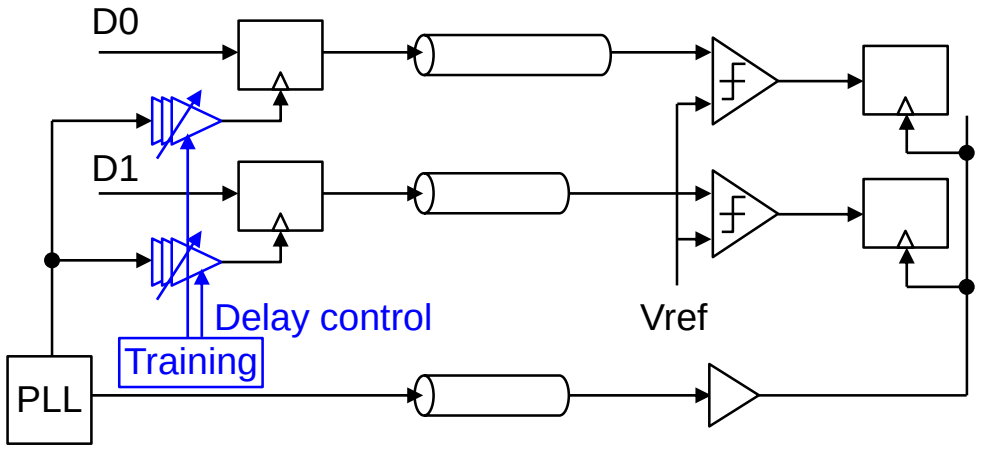


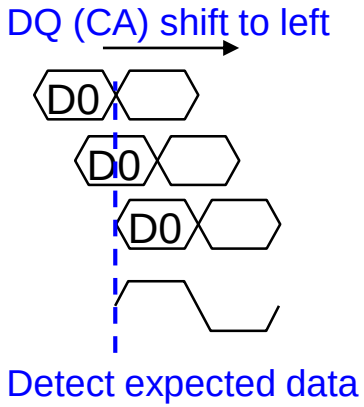
❖ SoC is required to maintain :

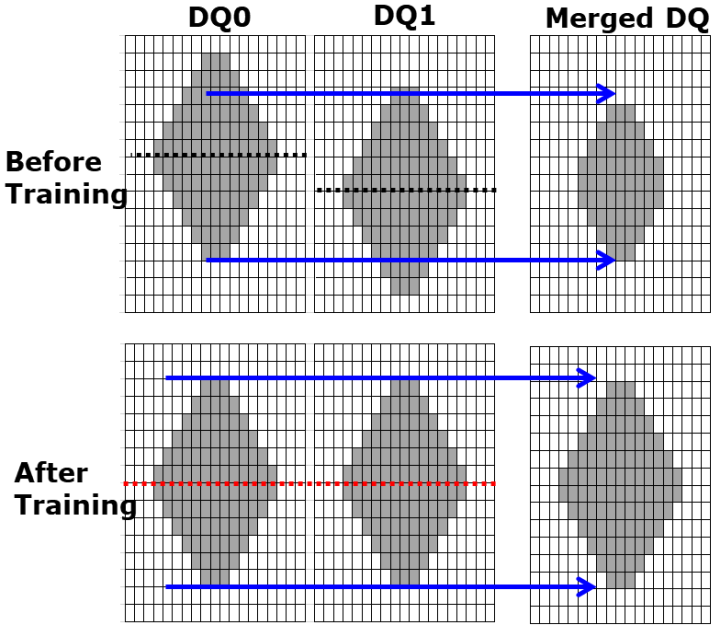
- tDQSS : CK-DQS → Write leveling
- tDSH : DQS-DQ (Write) → Write training
- tISH : CK-CA → CA calibration (Command Bus Training)
- tDQSQ : DQS-DQ (Read) → Read training

Training Method

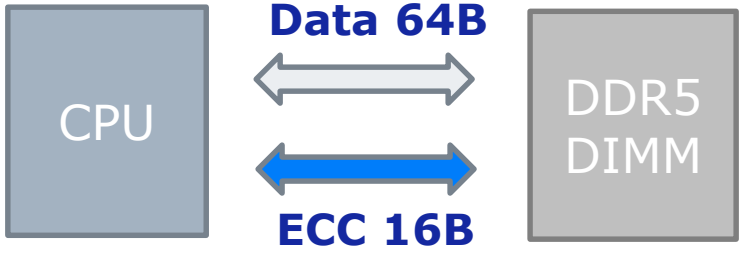
- Match the delay of DQ (CA) to that of DQS (CLK) at receiver input → Add the delay to DQ (CA) timing in order to compensate on-chip & channel mismatch
- Allows SoC to search the eye voltage and timing center per each DQ & CA



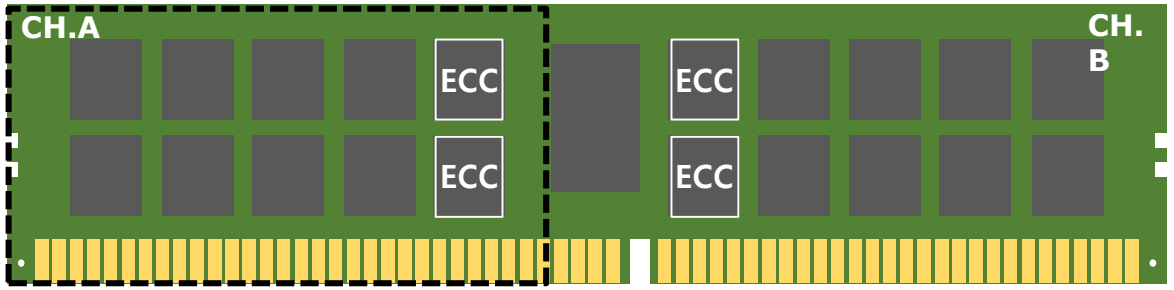




Link & Data Protection



DDR5 10x4 module



Access Granularity : Data 64B + ECC 16B

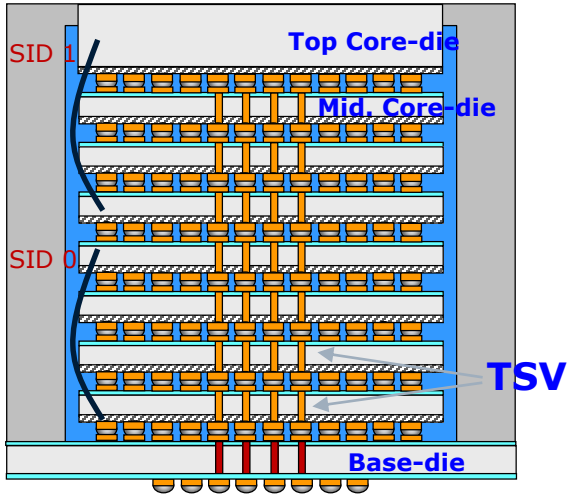
- System level ECC
  - 10x4 DDR5 RDIMM has dedicated 2 DRAM chips to store system ECC.
  - The system ECC protects system from both DRAM component errors (Cell & Core) and IO interface errors simultaneously.
  - The 16B ECC parity bits are able to cover any failure in a single memory chip

Emerging DRAM IO – HBM

SAMSUNG

HBM (8H) Architecture

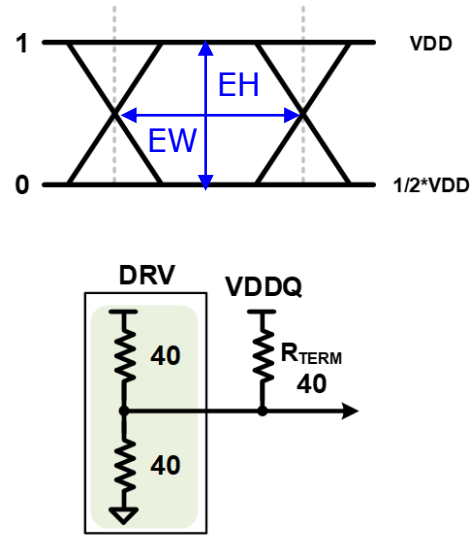
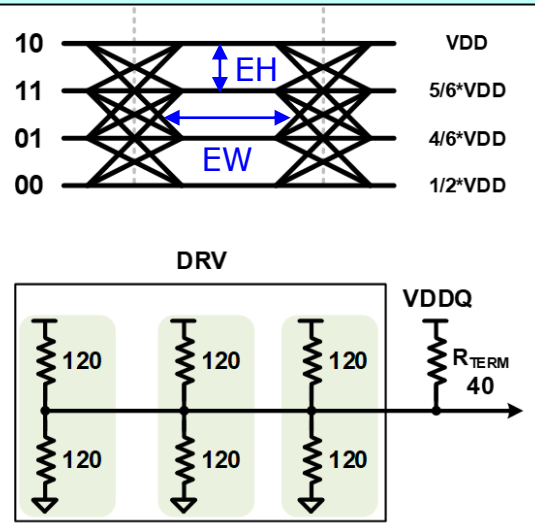
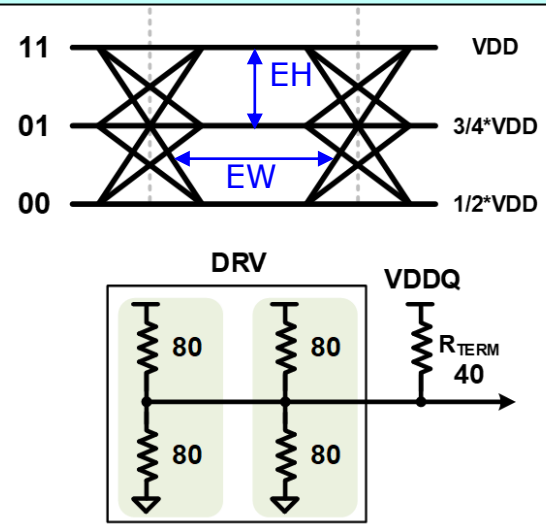
- To Maximize memory bandwidth, design constraints in more than a thousand of IO interface :
  - ODT is not allowed due to too much static power consumption
    - C_{IO} must be minimized for SI.
  - Base die without DRAM cell supports the requirements as buffers.
    - Reduced C_{IO}
    - Better Tr.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



Mode		HBM3e	HBM4
Pin Speed		8.0Gbps~10Gbps	**8.0Gbps~10Gbps**
# 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.

NRZ	PAM4	PAM3
		
EW= 1UI, EH = 1/2*VDD	EW= 0.75*UI, EH = 1/6*VDD	EW= 0.75*UI, EH = 1/4*VDD

* 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

Tradeoffs That Motivate Different DRAM Architectures

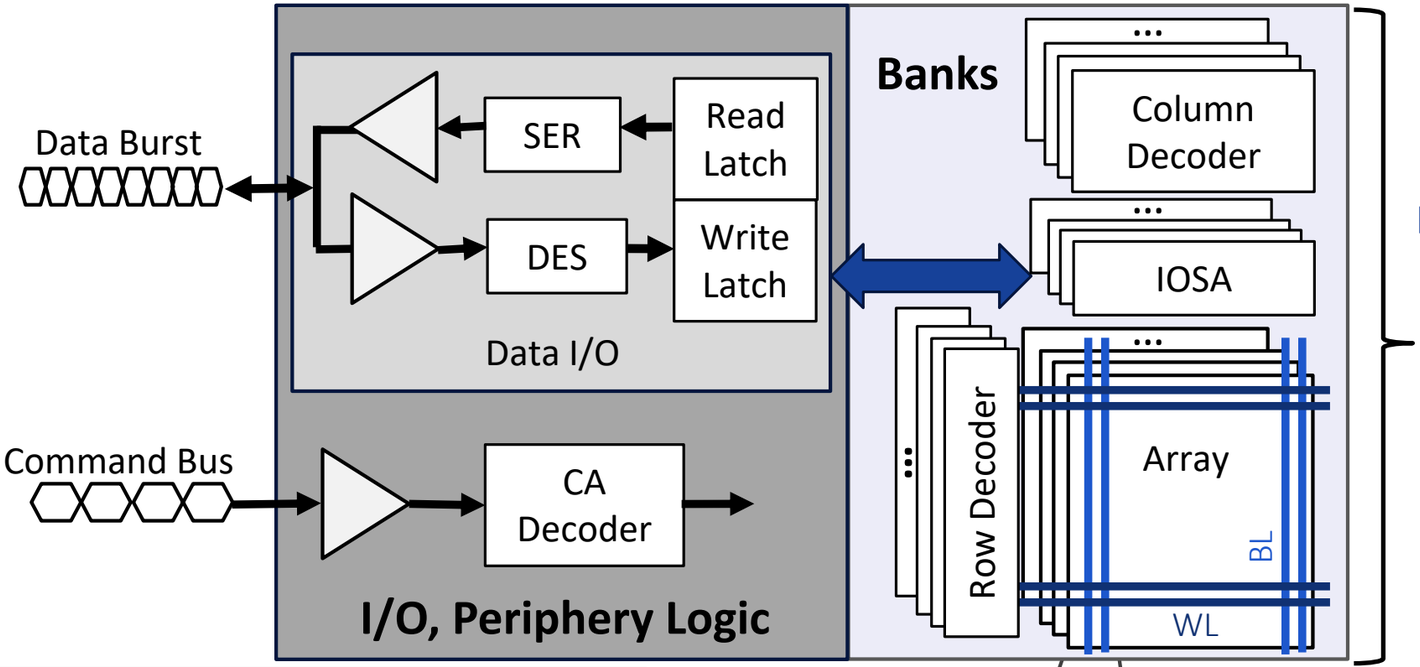
Wendy Elsasser
Technical Director
Rambus Inc.

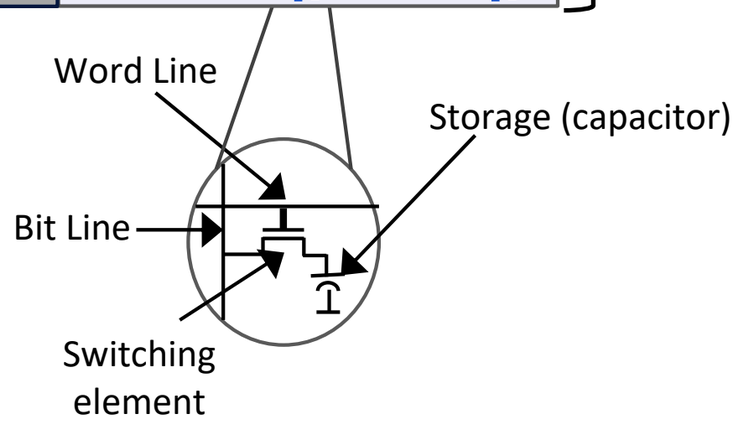
Rambus

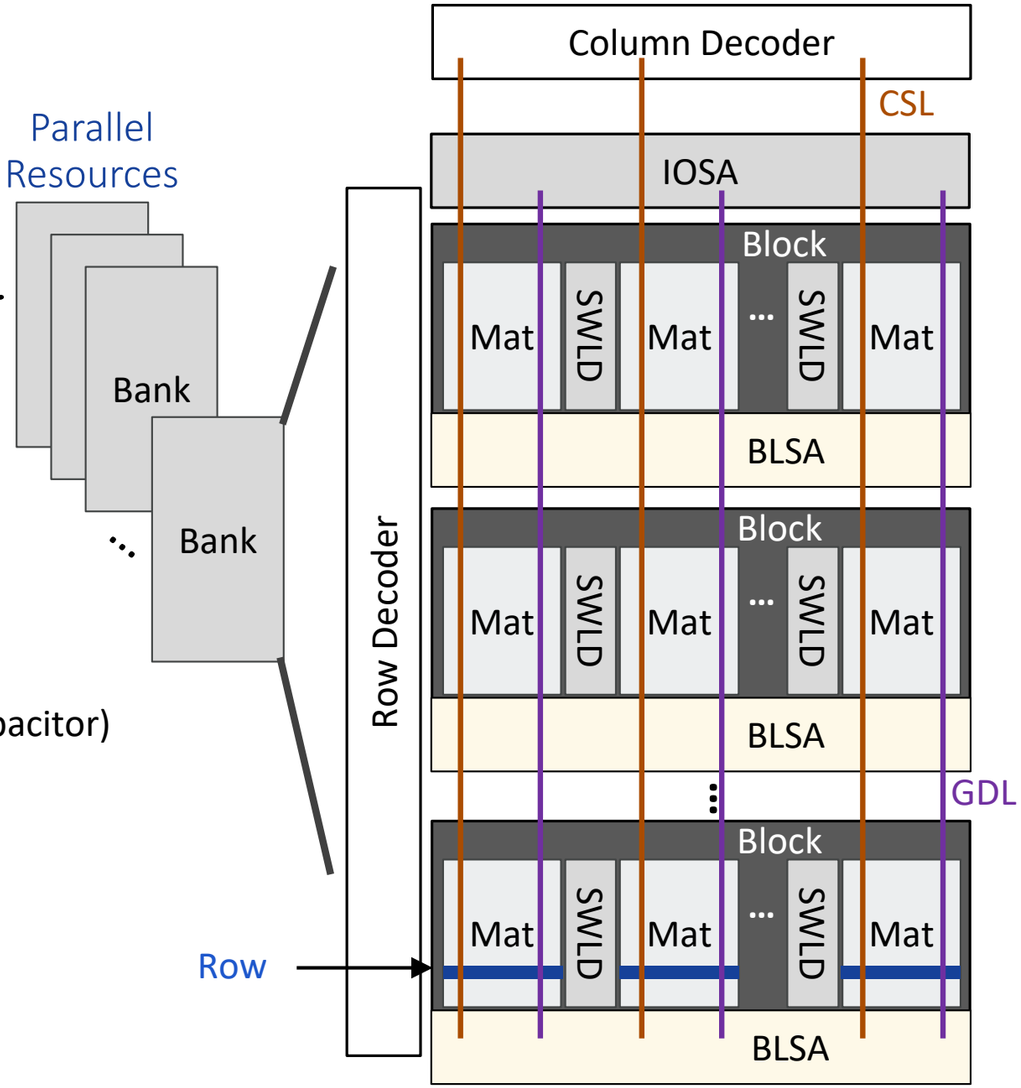
DRAM array structure

SER: Serializer logic
DES: De-serialzer logic
IOSA: IO sense amp
BLSA: Bit line sense amp

WL: Word line
SWLD: Sub-wordline driver
CSL: Column select line
BL: Bit line
GDL: Global data line







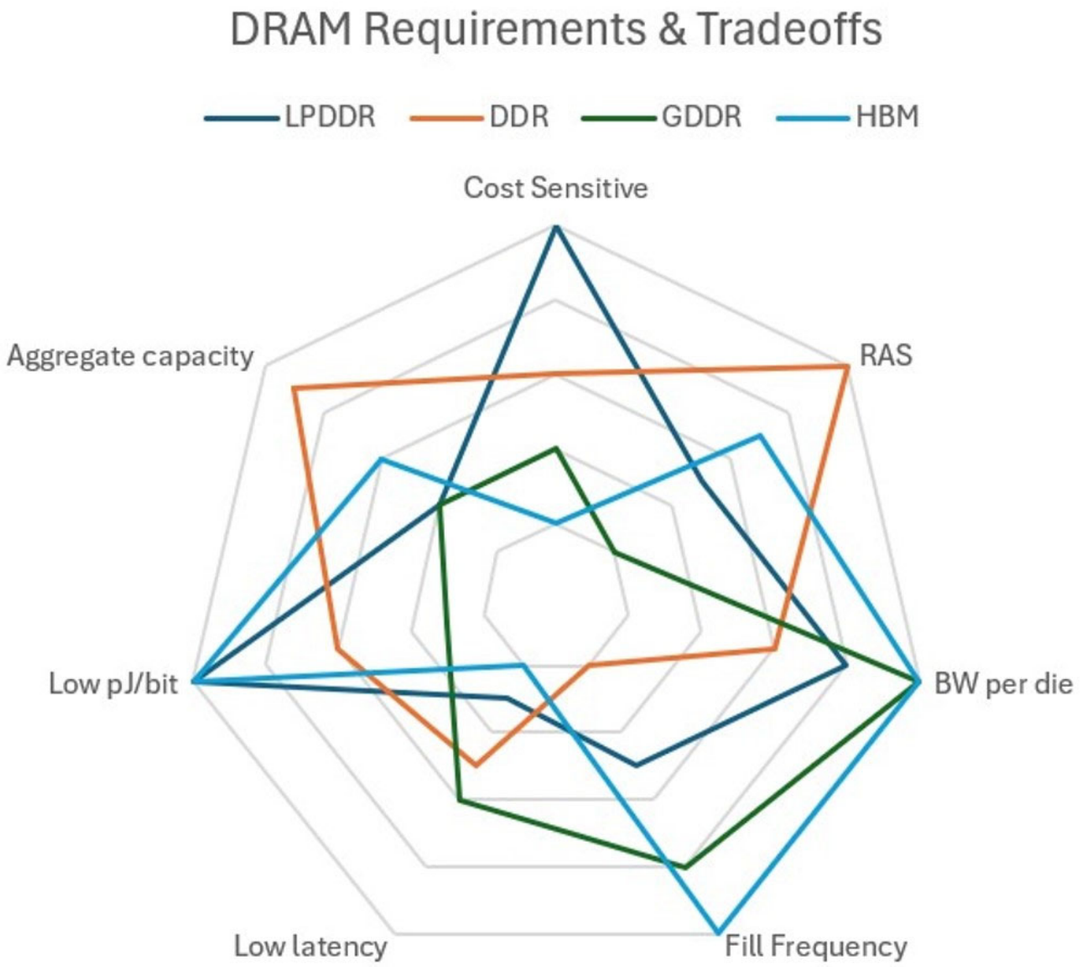
ACT: Row (page) moved to BLSA
- Row decoder selects WL
- BLs transmit cell contents

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

DRAM technology tradeoffs

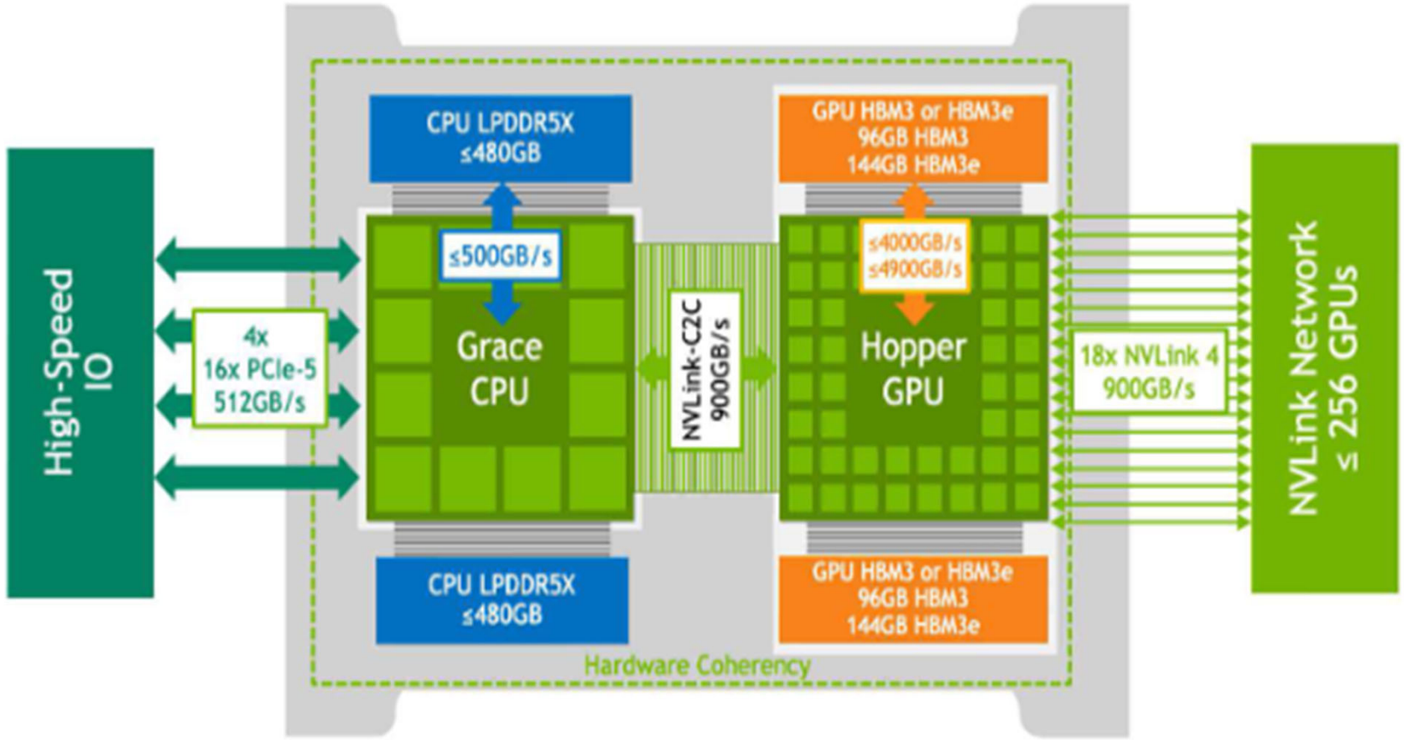
Multiple DRAM technologies serve different markets with varied tradeoffs

- LPDDR driven by mobile / client
  - Low cost, low power, energy efficient
  - Small package form factor
  - Use in other markets (auto, client, data center) drive configurability
- DDR driven by the data center (CPU memory)
  - High-capacity modules with high RAS
  - Lower cost (I.e., capacity/RAS) required for client but cannot sacrifice data center solution
- HBM and GDDR driven by AI and graphics
  - Throughput oriented
  - HBM provides high BW with low energy/bit
  - GDDR is a lower cost alternative with lower BW and higher energy/bit compared to HBM



Expanding and Overlapping Markets

- Mobile DRAM in the data center
- Limited use currently
- Low power is key advantage
- Capacity scaling is challenging
- Limited RAS capability
  (at reasonable overhead)



https://resources.nvidia.com/en-us-grace-cpu/grace-hopper-superchip?ncid=no-ncid

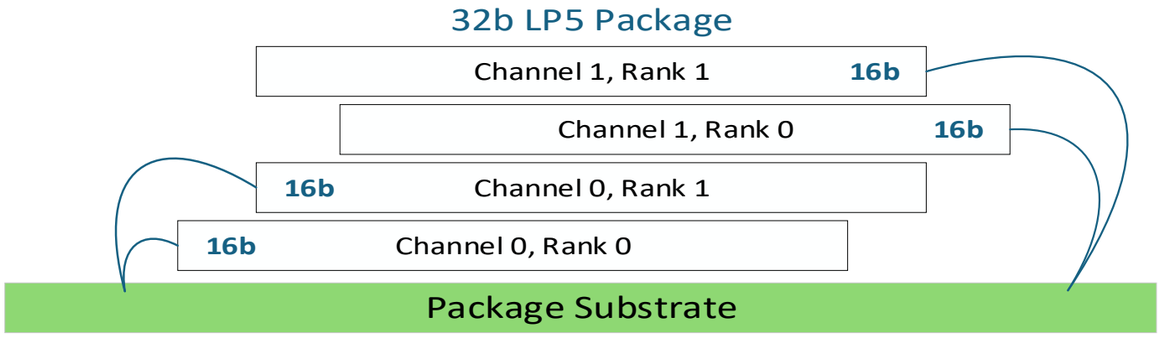
Requirements Drive Varied Commodity Solutions

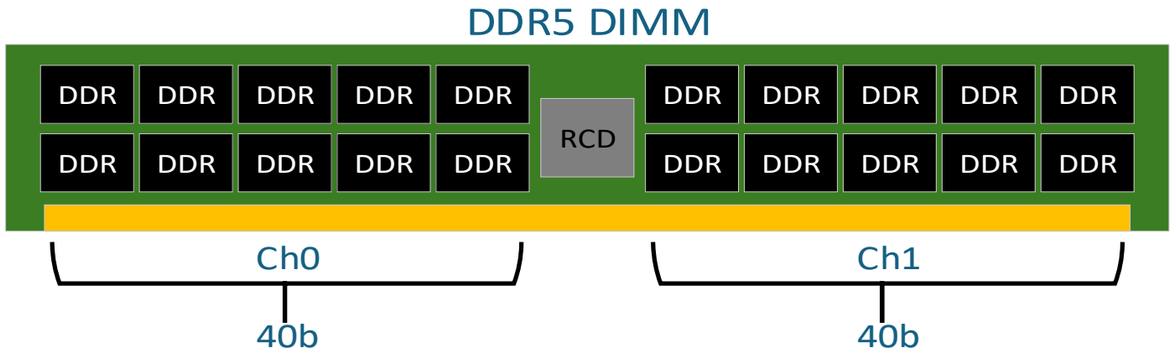
LPDDR Package

- Short reach, multi-die, multi-channel
  - Small form factor, soldered down or PoP
  - Lower signaling energy
  - Low cost with wire bonding
- 1-2 dies per rank (more DQs per die)
  - Lower CA loading
- Latency sacrificed for lower power
  - ~40% higher tRL, ~25% higher tRC (device latency)

DDR Data Center DIMM

- Multi-die channel
  - Many dies accessed concurrently for 64B (fewer DQs per die)
  - High RAS capability with low overhead
  - High aggregate capacity per channel
- Field replaceable unit
  - ‘Easy’ replacement for serviceability

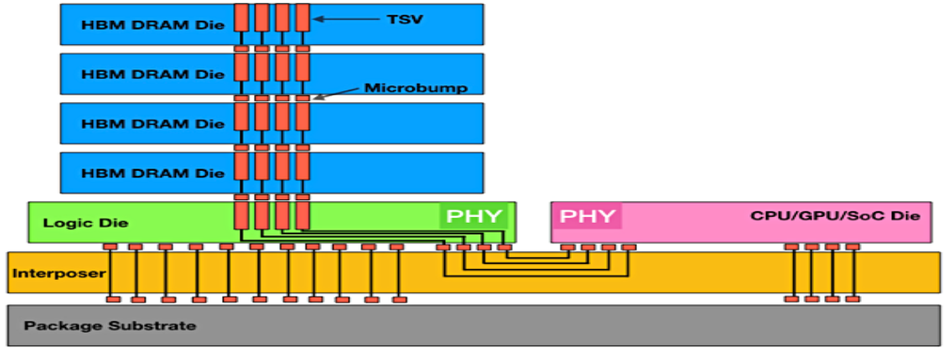




Requirements Drive Varied High-Throughput Solutions

HBM Stack

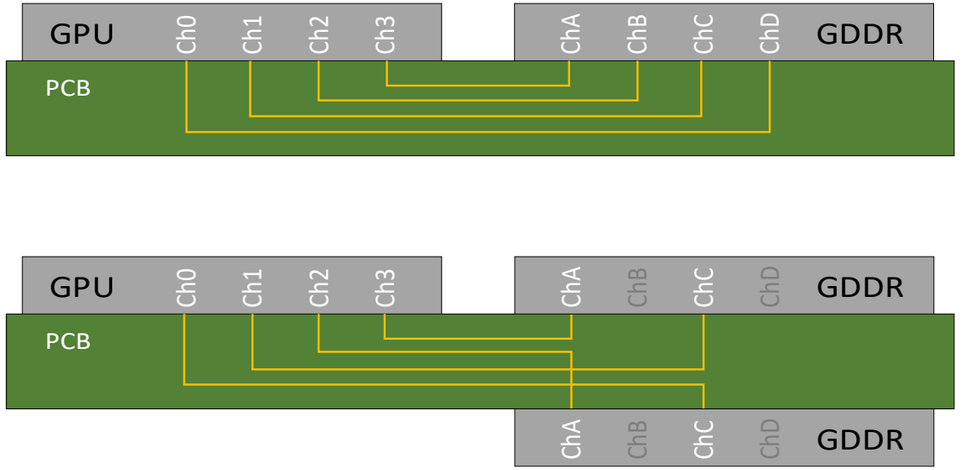
- Ultra short reach, tightly coupled
  - Wide, ultra-energy efficient (low pJ/bit)
  - Multi-die stacks w/ multiple channels per die
  - Logic base die (logic process) between host and DRAM with internal TSVs to DRAM dies
- Channels accessed independently
  - High aggregate BW &fill frequency
  - Subset of a die accessed per burst



https://semiwiki.com/forum/threads/hbm4-set-to-land-sooner-than-expected.20642/

GDDR Package

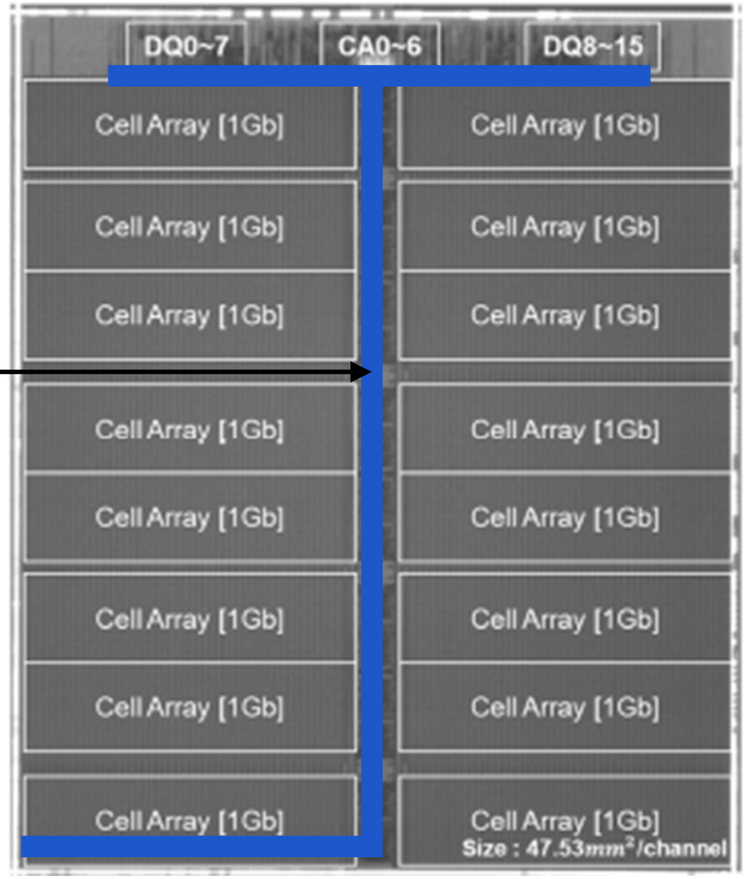
- Short reach, high-speed signaling
  - Narrow, high BW per pin (PAM3 w/ GDDR7)
  - At the expense of energy per bit
- Lower cost compared to HBM
- Clamshell mode for higher capacity
  - Versus higher stack (more dies) in HBM



Architectural tradeoffs

Optimized packaging versus data latency

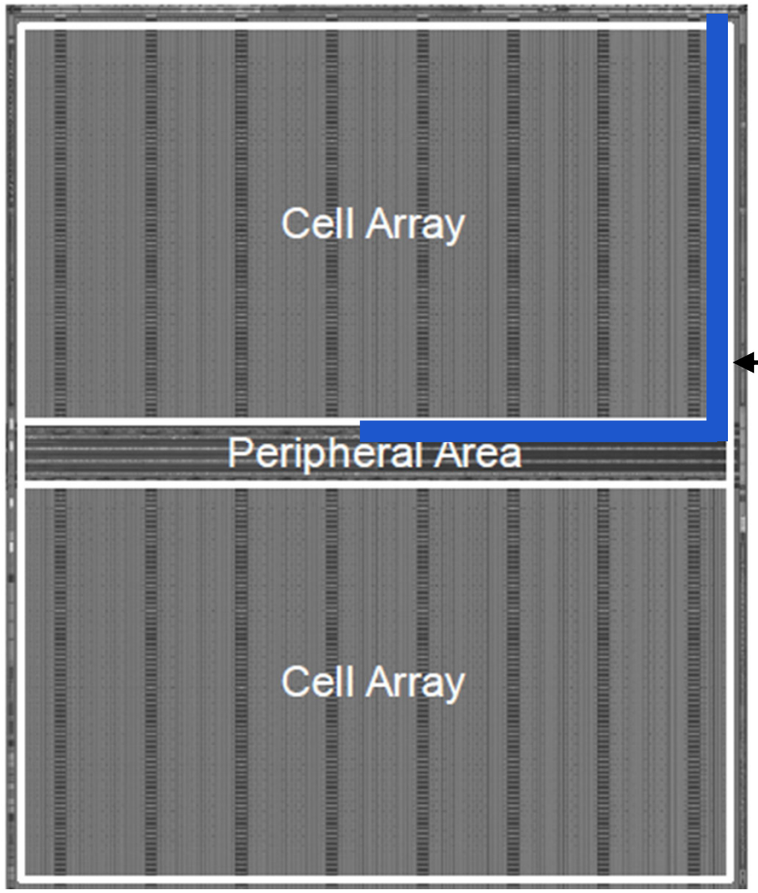
Edge Pads



Longer internal DQ path

ISSCC 2022 28.3, “A 16Gbit 9.5Gb/s/pin LPDDR5X SDRAM “

Center Pads



Shorter internal DQ path

ISSCC 2023 28.7, “A 1.1V 6.4Gb/s/pin DDR5 SDRAM “

Architectural tradeoffs

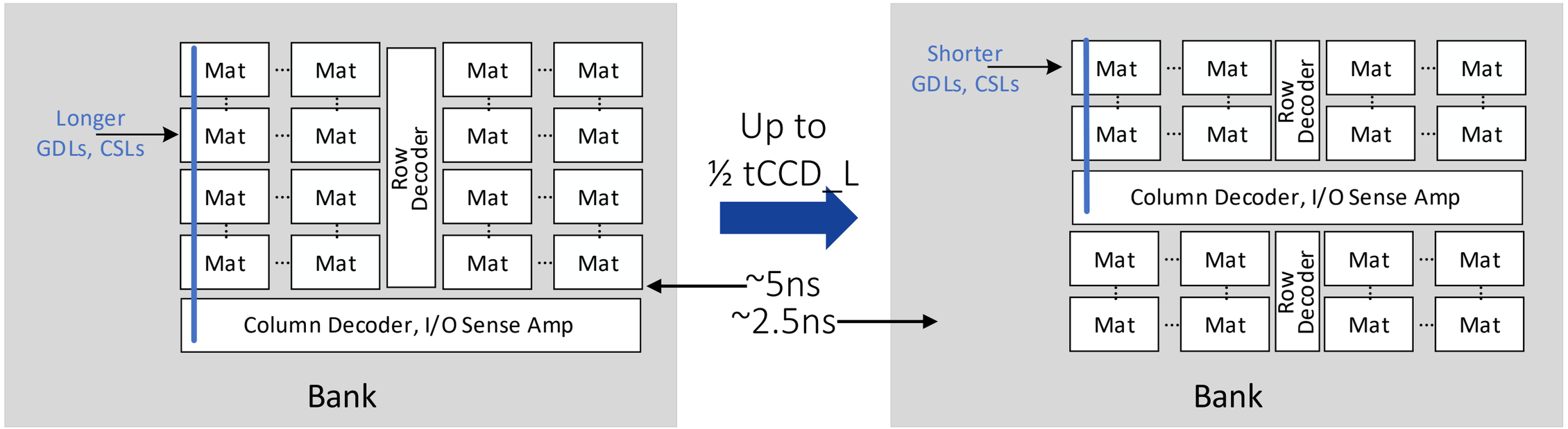
Cost versus Performance

DDR - LPDDR

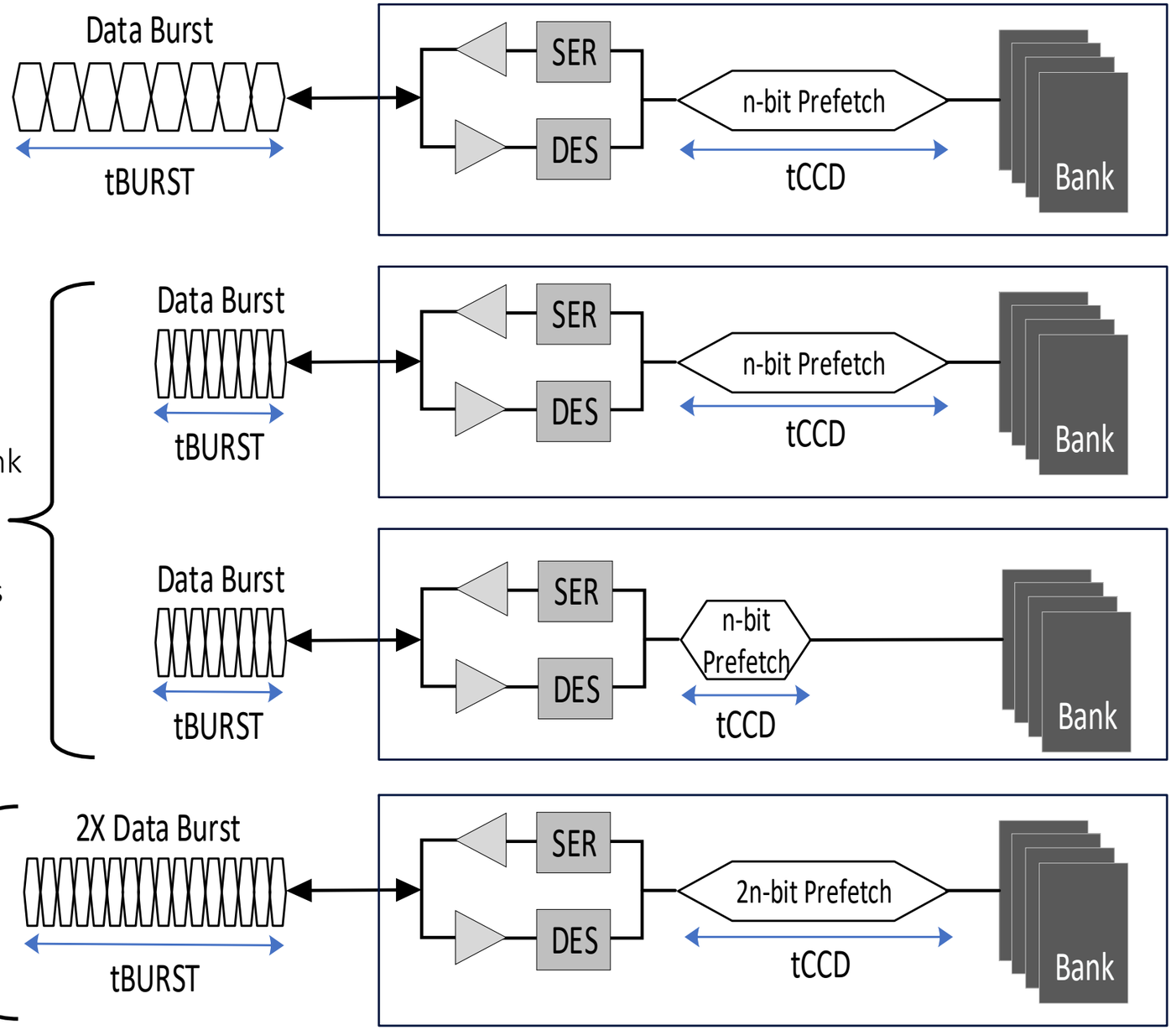
- IO sense amp at the end of a bank
- Reduced overhead for lower cost
- Shared logic within the bank group for timing control, decoder, etc.

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)

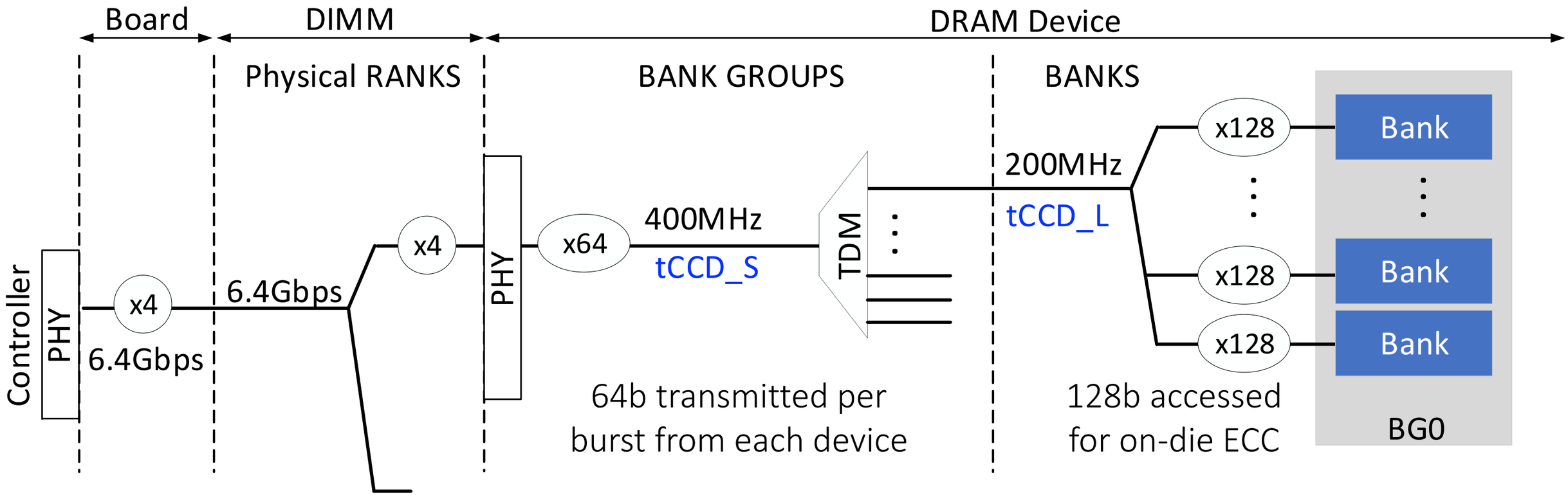


Bandwidth Scaling Tradeoffs



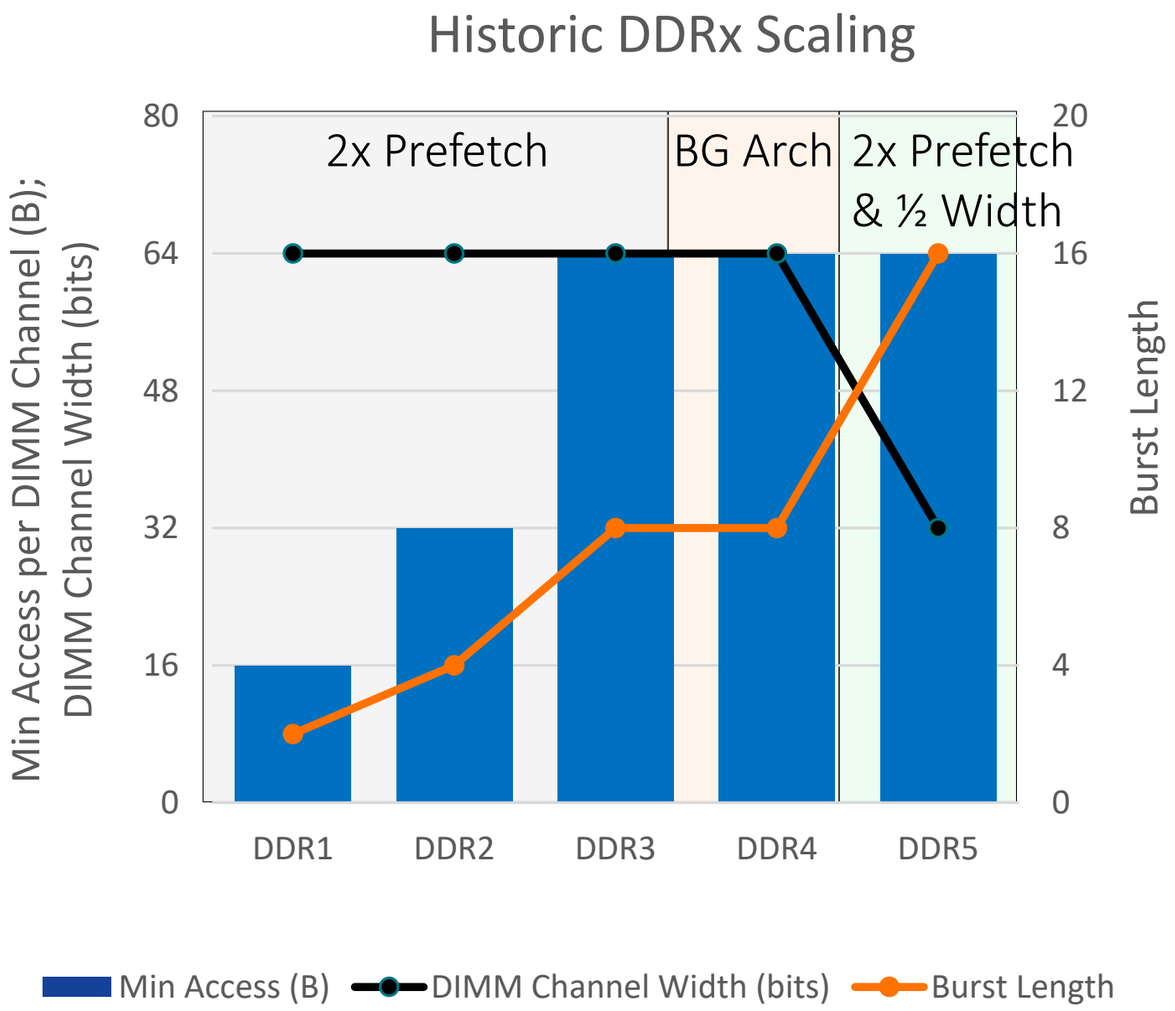
Bandwidth Scaling with Bank Group Architecture

x4 DDR5 DIMM example

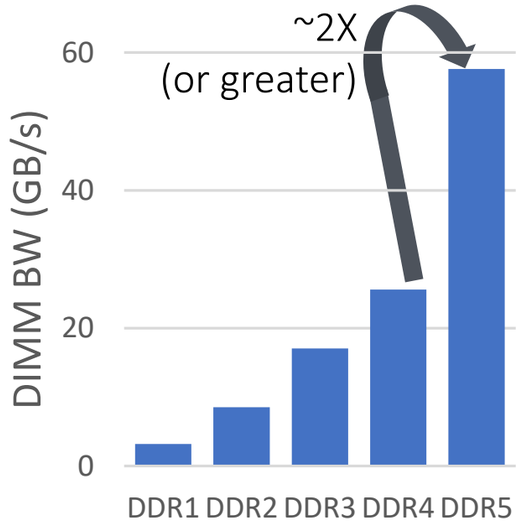


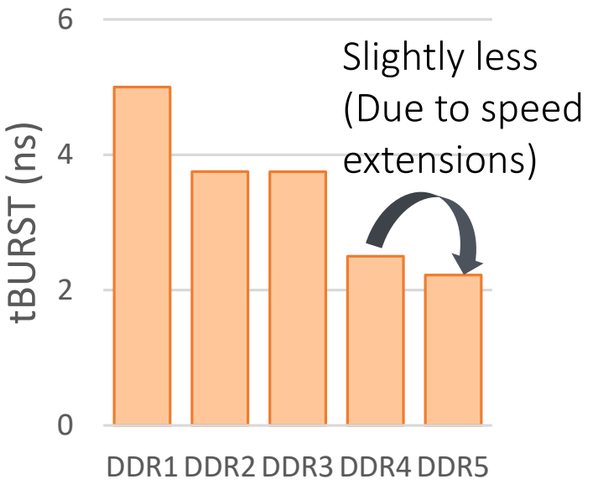
Time delay multiplexing between BGs to effectively double the core frequency
(Multiple ranks can also be accessed simultaneously)

DDRx Evolution



- Prefetch doubled each generation from DDR1 – DDR3 with 2x burst length
- To maintain 64B min access granularity
  - DDR4: 2x effective core frequency with BGs
  - DDR5: 2X prefetch but with ½ width DIMM channel (2 channels per DIMM)
- tBURST >= 2ns maintained





Challenges Pushing Technology Roadmaps

- Double the bandwidth
- Maintain 64B access granularity
- Do not increase core frequency
- Do not increase pins per DIMM

} Potential Constraints

Example Base DIMM

- 200MHz core, 6.4Gbps I/O
- 64B accessed from a bank every 5ns

→ Burst of 16 in 2.5ns + BG arch

Potential Options for Future DIMMs[1]

Opt1: Lower tBURST impacts scheduling complexity and/or requires multi-level BG architecture

Opt2: Narrow channel width challenging for RAS

Opt3: Others?

	Base	Opt1	Opt2	Opt3
Core Freq (MHz)	200	→	→	→
DIMM Prefetch (B)	64	→	→	→
Data Rate (Gbps)	6.4[2]	12.8	12.8	?
DIMM Channel (bits)	32	→	16	?
Channels / DIMM	2	→	4	?
DIMM BW (GB/s)	51.2	102.4	102.4	102.4
Burst Length	16	→	32	?
tBURST (ns)	2.5	1.25	2.5	?

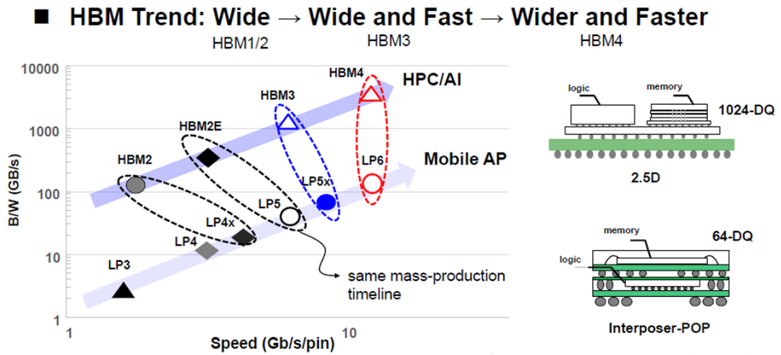
[1] Illustrating potential options & tradeoffs, final architecture TBD

[2] Initial max DDR5 BW target; spec has since been extended

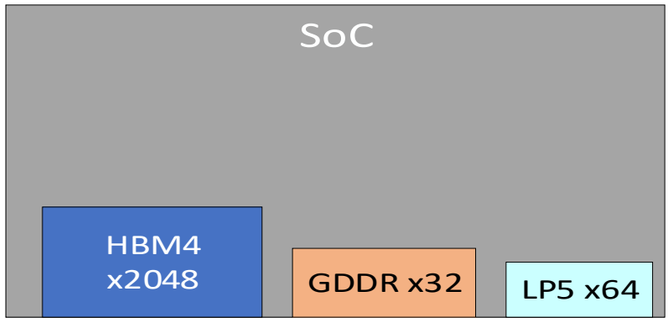
Additional Bandwidth Scaling Techniques

Wider interfaces

HBM, LPW (mobile)



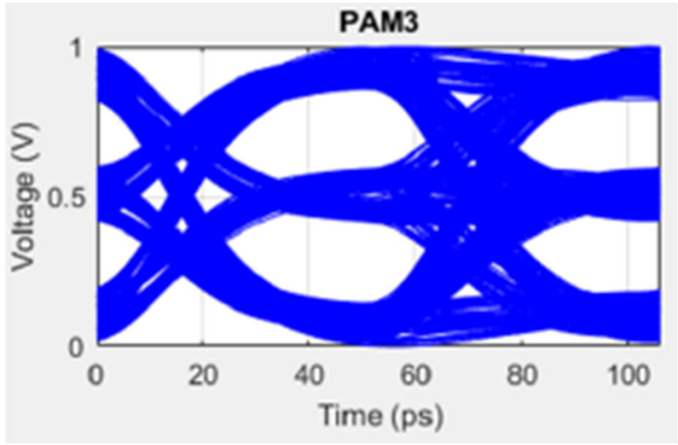
ISSCC 2023 28.4, "A 4ns 1.15TB/s HBM3 Interface"



Optimized bandwidth per shoreline (GB/s per mm)

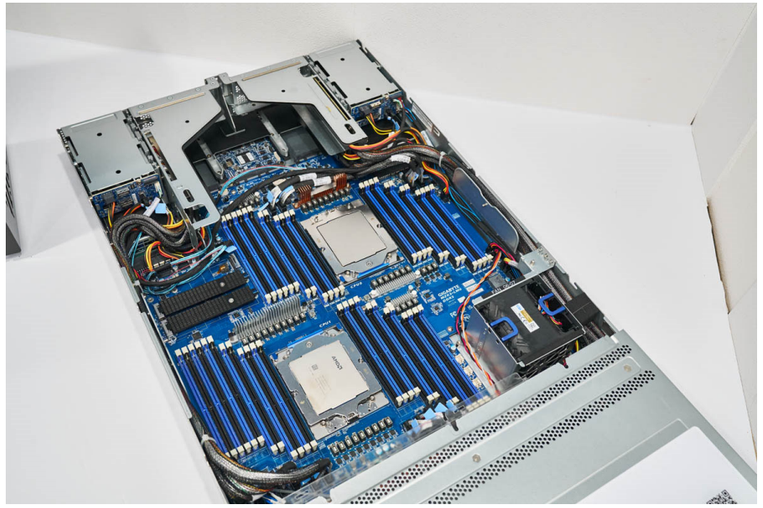
More complex signaling

- PAM3 I/O with GDDR
- More information transmitted per UI
- Data encoded into (-1, 0, +1) instead of (0,1) NRZ
- Higher voltage I/O



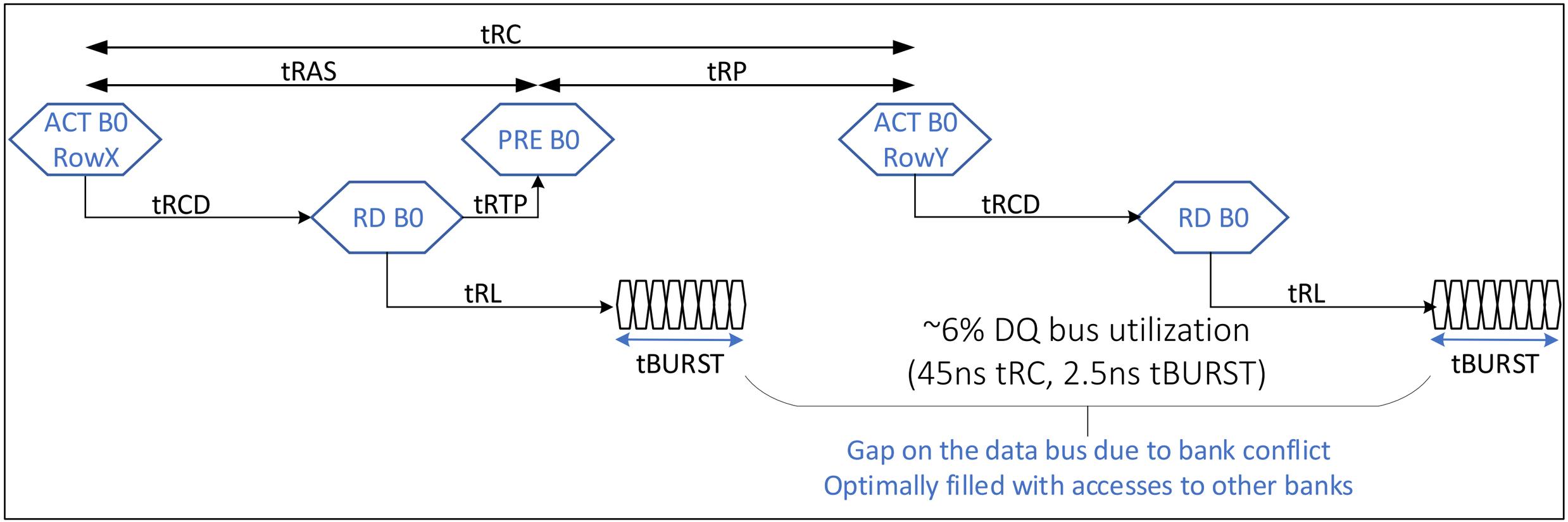
More channels

- More DIMMs or DRAM packages per SoC
- Higher BW and capacity
- SoC shoreline, PCB routing & thermal limited



https://www.servethehome.com/48-ddr5-memory-slots-twisting-to-fit-in-a-2u-server-gigabyte-r283-zk0-amd-epyc/

Other Considerations – Bank Timing



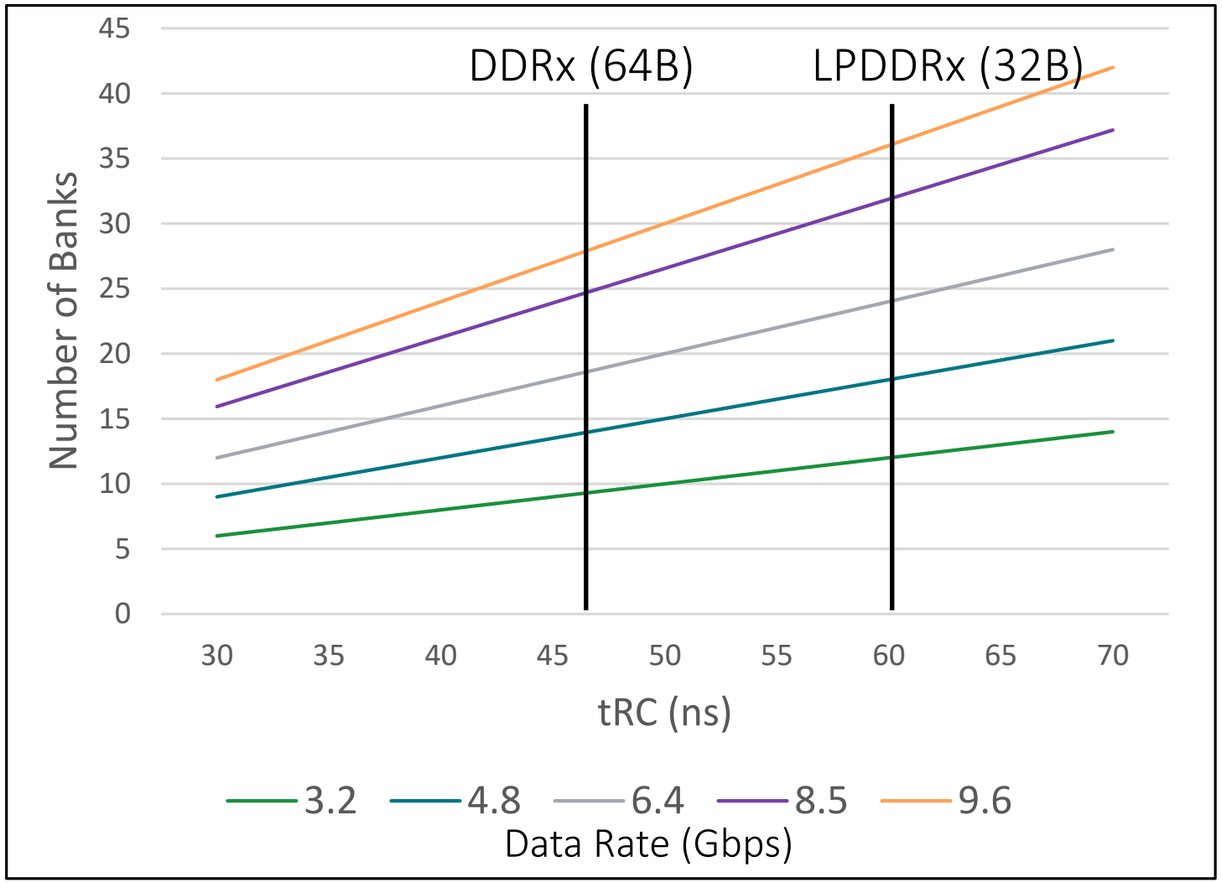
- Access different bank groups avoids tCCD_L gaps
- Access to banks within a single rank avoids rank-to-rank delay gaps
- Timing shown for case where tRTP + tRP can be met within tRC delay
- Write accesses need to also account for tWR, which will push out the precharge command

Bank Timing and Minimum Number of Banks

Impacts random access throughput and can limit achievable bandwidth

- Random traffic must leverage bank level parallelism to maintains I/O bus utilization
- DRAM timing parameters determine minimum number of banks required
  - Perfectly rotating across banks
  - tRC: ACT-to-ACT same banks
  - Min #Banks = tRC/tBURST
- Higher data rates reduce tBURST
  - More banks required to maintain throughput
- Purely random traffic requires more banks
  - Birthday paradox (bank conflicts w/in the queue)
  - Larger queue helps but increases host complexity
- Lower tRC beneficial for random traffic

Minimum #banks for random access (BL16) with perfect bank rotation based on data rate & tRC

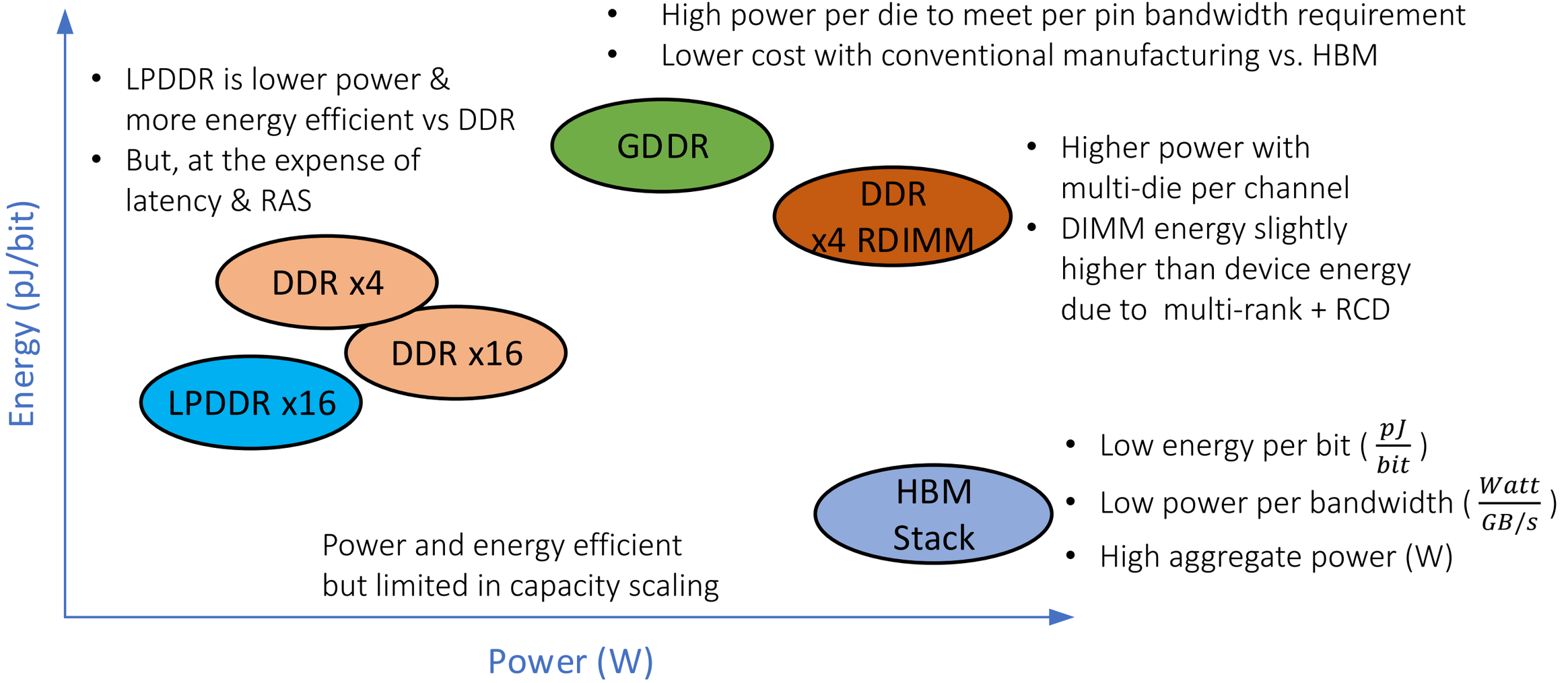


Power and Energy Comparison

Wendy Elsasser
Technical Director
Rambus Inc.

Rambus

What About Power and Energy?



- High power per die to meet per pin bandwidth requirement
- Lower cost with conventional manufacturing vs. HBM

- LPDDR is lower power & more energy efficient vs DDR
- But, at the expense of latency & RAS

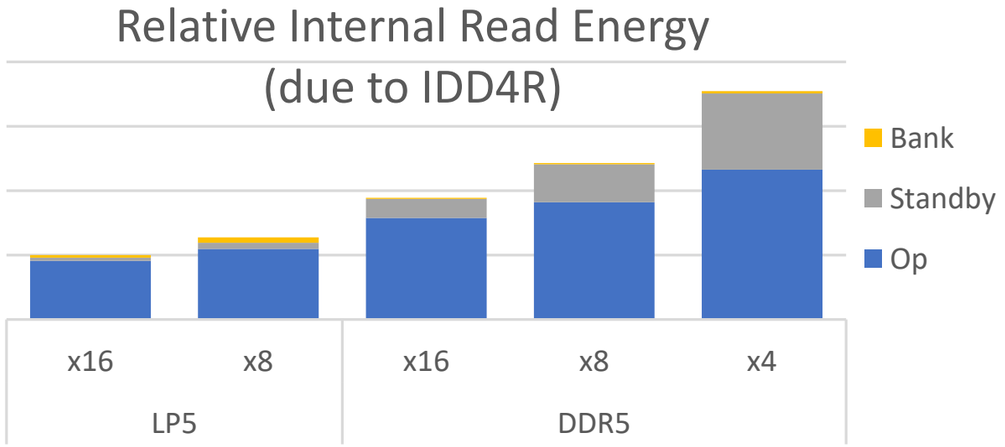
- Higher power with multi-die per channel
- DIMM energy slightly higher than device energy due to multi-rank + RCD

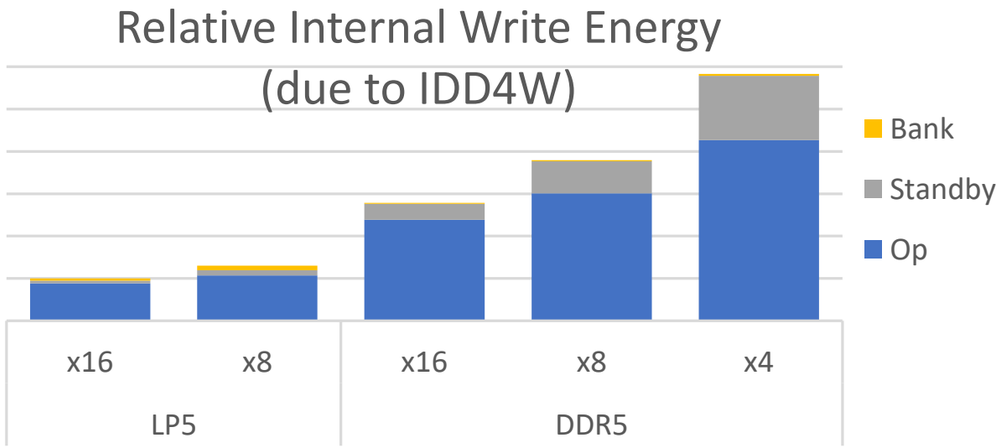
Power and energy efficient but limited in capacity scaling

- Low energy per bit ($\frac{pJ}{bit}$)
- Low power per bandwidth ($\frac{Watt}{GB/s}$)
- High aggregate power (W)

LPDDR vs DDR Device Energy

- DDR5 energy is significantly higher
  - Standby power is a large contributor
  - Gap increases with smaller width due to lower bandwidth per device
- Voltage accounts for a small %
  - 1.1V vs 1.05V VDD2 → ~5% decrease
- Other contributions
  - DDR x4 over prefetch
  - More complex Rx logic & equalization circuits
  - Clock delivery network (on-device DLL w/ DDR)
  - Smaller drivers & high threshold transistors
  - Process technology (gap may be shrinking)
  - IDD spec padding
- Channel energy is also a factor (not included in graph)
  - Higher energy for longer reach; lower LPDDR VDDQ





Lower energy at the expense of latency & RAS

- ~40% higher read latency
- ~25% higher tRC

} Device latency

- Higher RAS overhead with mobile devices

RAS Techniques, Overheads and Tradeoffs

Wendy Elsasser
Technical Director
Rambus Inc.

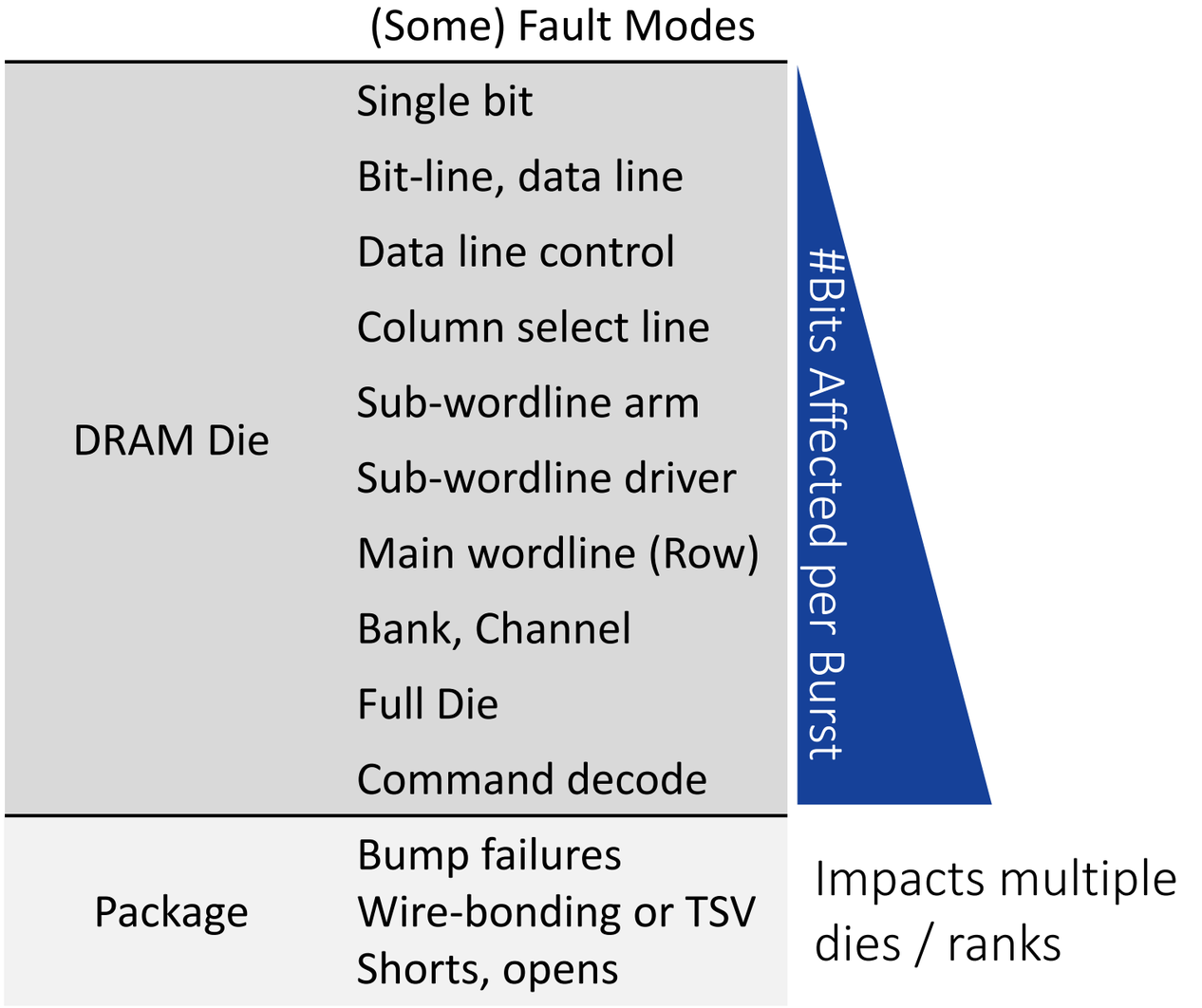
Rambus

How to Ensure High RAS Capability when Needed?

Reduce silent data corruptions (SDC)

- When errors exceed ECC coding capability, aliasing can occur
  - Detect error in wrong bit(s) & miss-correct
  - Do not detect any errors

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

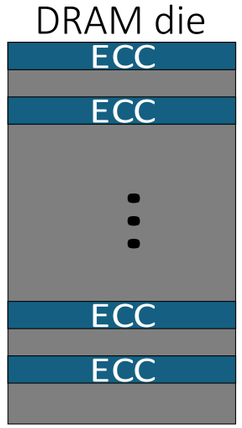


Attempts to Enhance RAS

LPDDR, GDDR: In line ECC

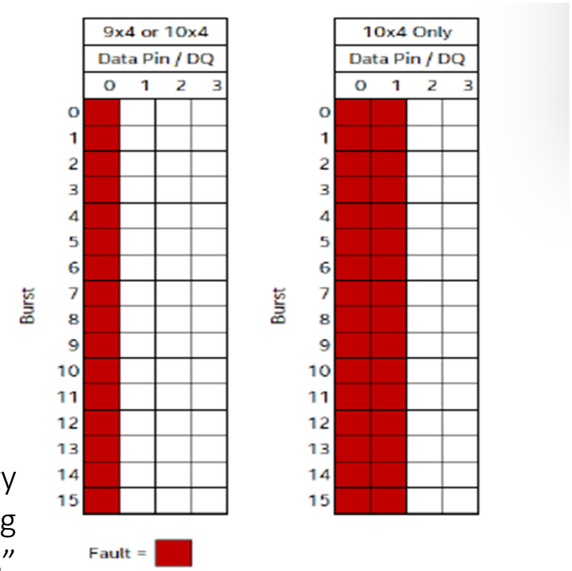
- ECC & data stored in the same DRAM
- ECC symbols transmitted in a separate burst
- Reduces data capacity
  - Amount depends on RAS capability
- Reduces data bandwidth
  - Can amortize loss with streaming accesses, with multiple data bursts accessing same ECC burst

    E.g., 12.5% loss with 64b ECC per 64B cacheline
  - 50% loss with random accesses
- ECC stored in different row (or bank) to bound errors to different fault domain



DDR: Bounded Fault

- Limits the number of failure patterns seen by the memory controller
- Enables the memory controller designer to align the error code symbols to maximize error correction coverage.
- No bounded fault currently defined fo LPDDR or GDDR

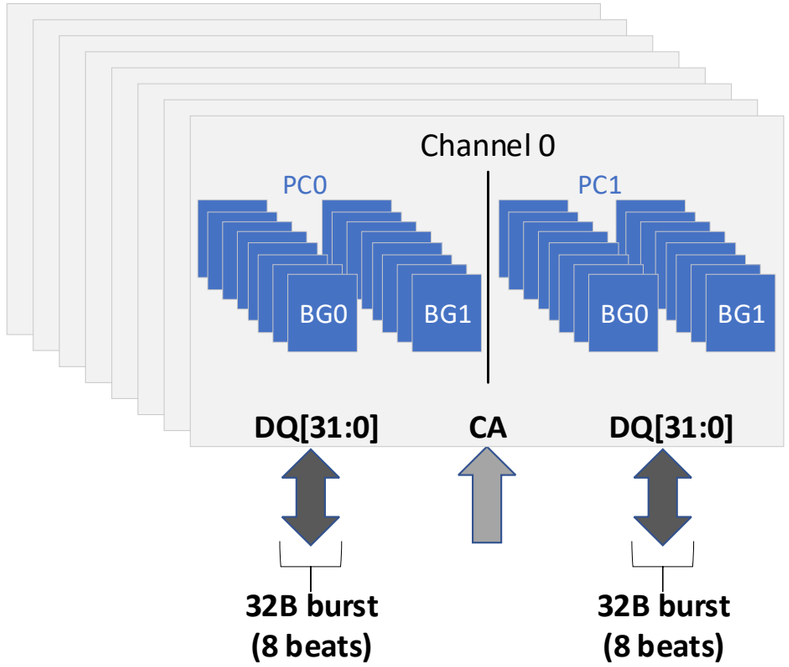


" Improving Memory Reliability by Bounding DRAM Faults"

Figure 2: Fault boundary for DDR5-BF x4 devices. In all 9x4 devices major sub-CL faults will impact data on one DQ. Major sub-CL faults in 10x4 devices can impact data on either one or two DQs.

RAS Overhead Depends on DRAM Architecture

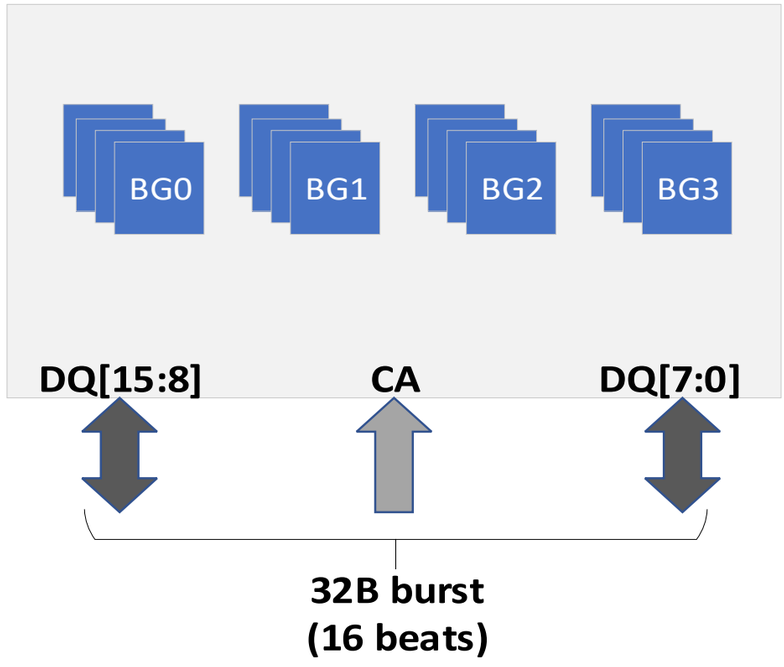
HBM die (x64 channel)



- Multiple channels per die
- 32B from a single PC

(4X) x4 DDR die

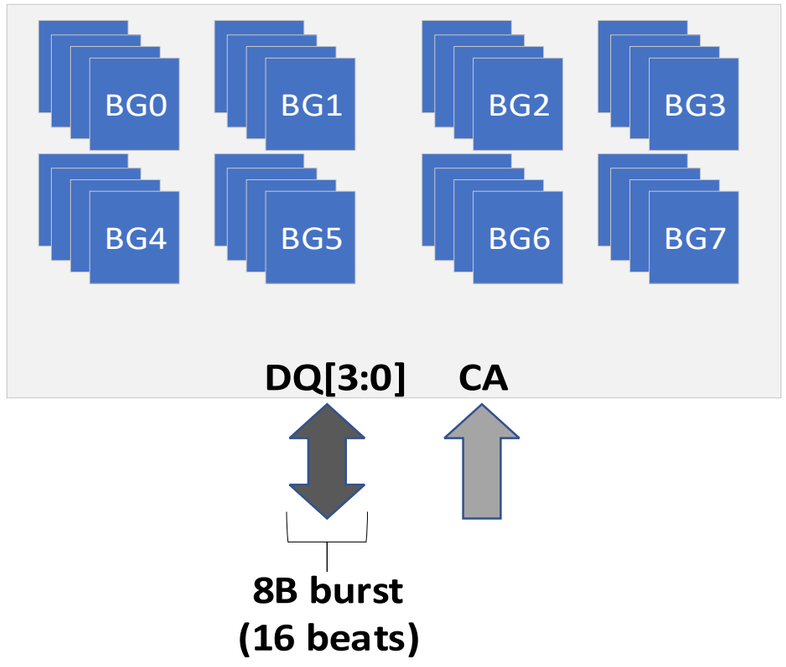
LPDDR5 x16 die



- Byte mode connects 8b DQ to host with 16B burst

(2X to 4X) x4 DDR

DDR5 x4 die

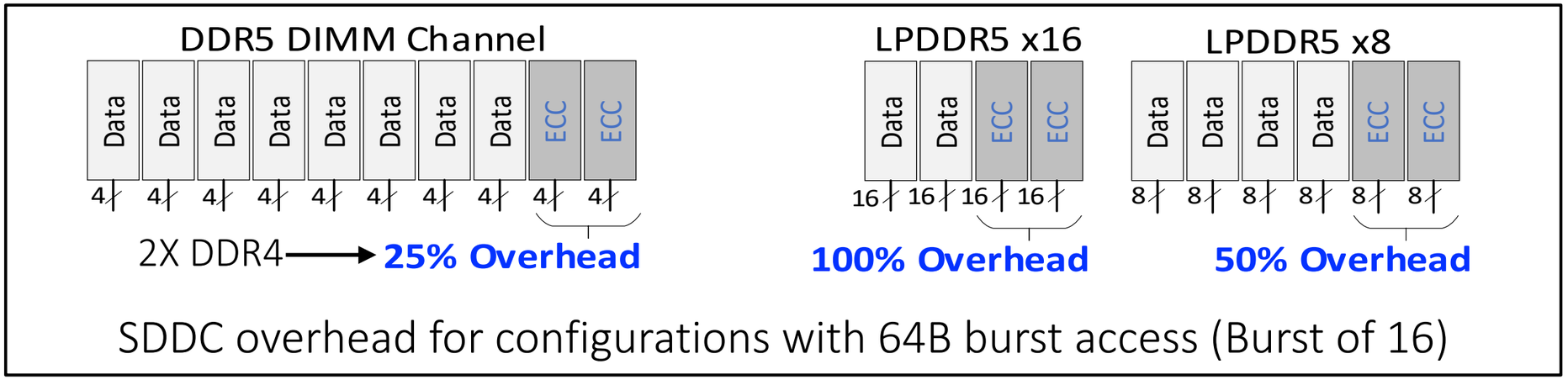


- x8 dies provide 16B burst (lower module capacity)

System RAS Coding Schemes

Tradeoff of coverage, complexity and overhead

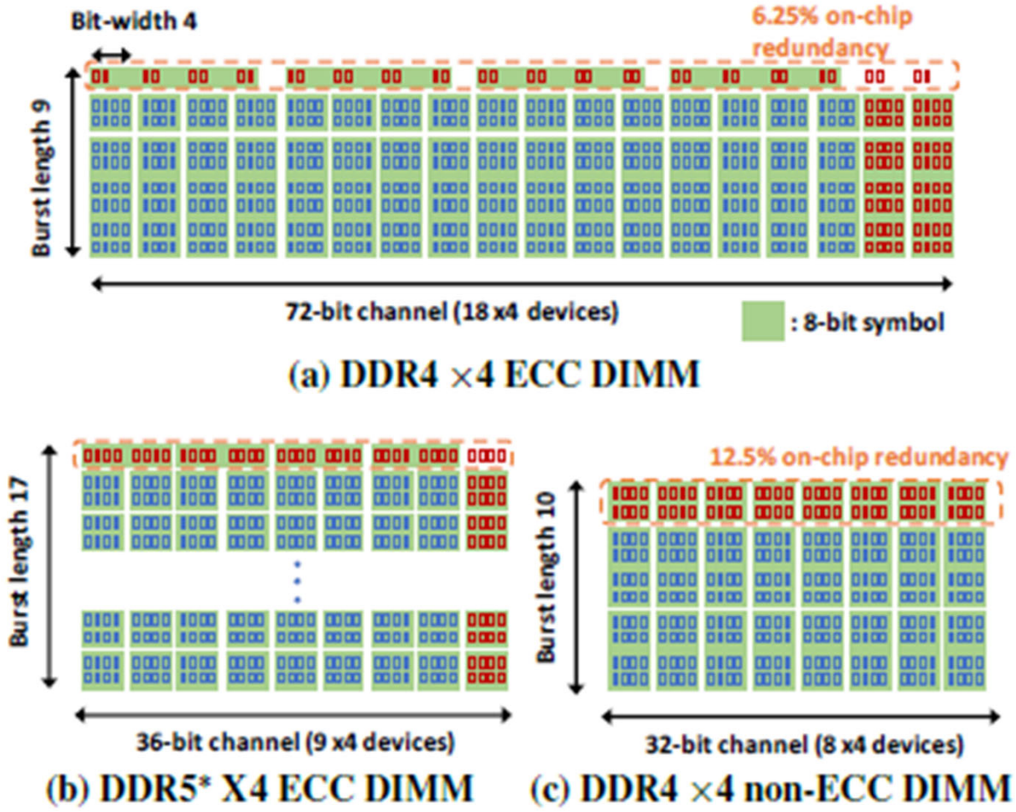
- Hamming, BCH are lower coverage schemes
  - Detects random bit failures (SEC, SECDED, DECTED)
  - ($\log_2$(datawidth) + 2) parity bits required for SECDED
- Reed-Solomon provides higher coverage
  - Symbol based code that requires more redundancy (higher overhead)
  - With 2t ECC symbols, can correct t symbol errors in a codeword
  - ➢ SDDC, single die data correction, requires a 2 die overhead (high cost – are there other options?)



SDDC overhead for configurations with 64B burst access (Burst of 16)

Lower Cost “SDDC-lite” Options (1 of 2)

Expose (and re-purpose) on-die ECC bits to host controller



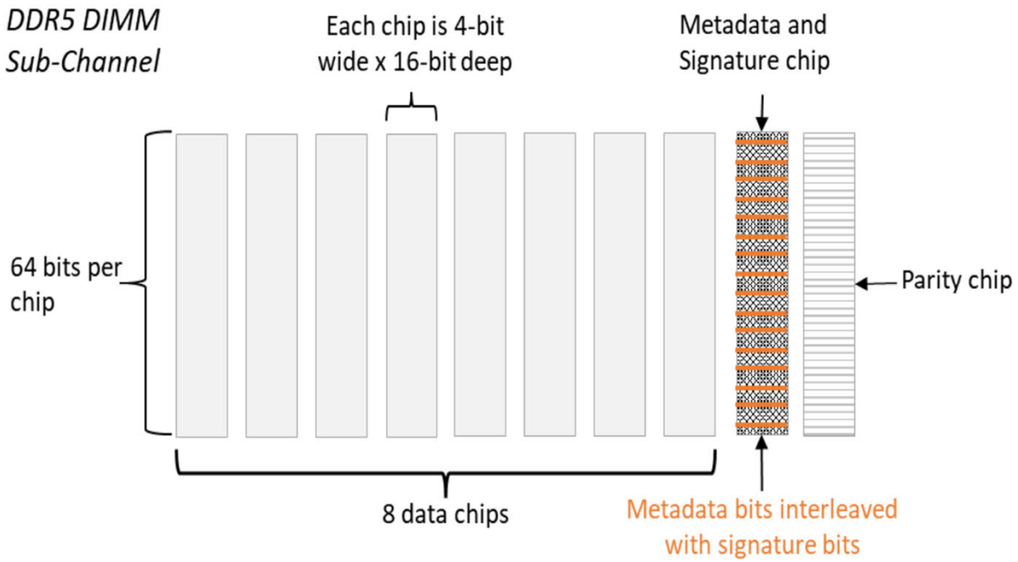
“DUO: Exposing On-Chip Redundancy to Rank-Level ECC for High Reliability”, HPCA 2018

Increased ECC coverage with lower overhead

- DUO – SDDC with 1 redundant device
  - 512b data + 100b redundancy (64b + 4b*9)
  - RS(76,64) GF(2^8) corrects six 8b symbols
  - Additional 4b for on-chip redundancy parity
- Burst erasure encoding extends capability
  - 1t symbols needed for 1 erasure
  - Brute force decoding search serially checks for failures (erasures) in each device
- Internal read-modify-write not needed
- No aliasing between system and on-die ECC algorithms
- Burst extended, impacting performance

Lower Cost “SDDC-lite” Options (2 of 2)

Parity + Signature for lower overhead RAS



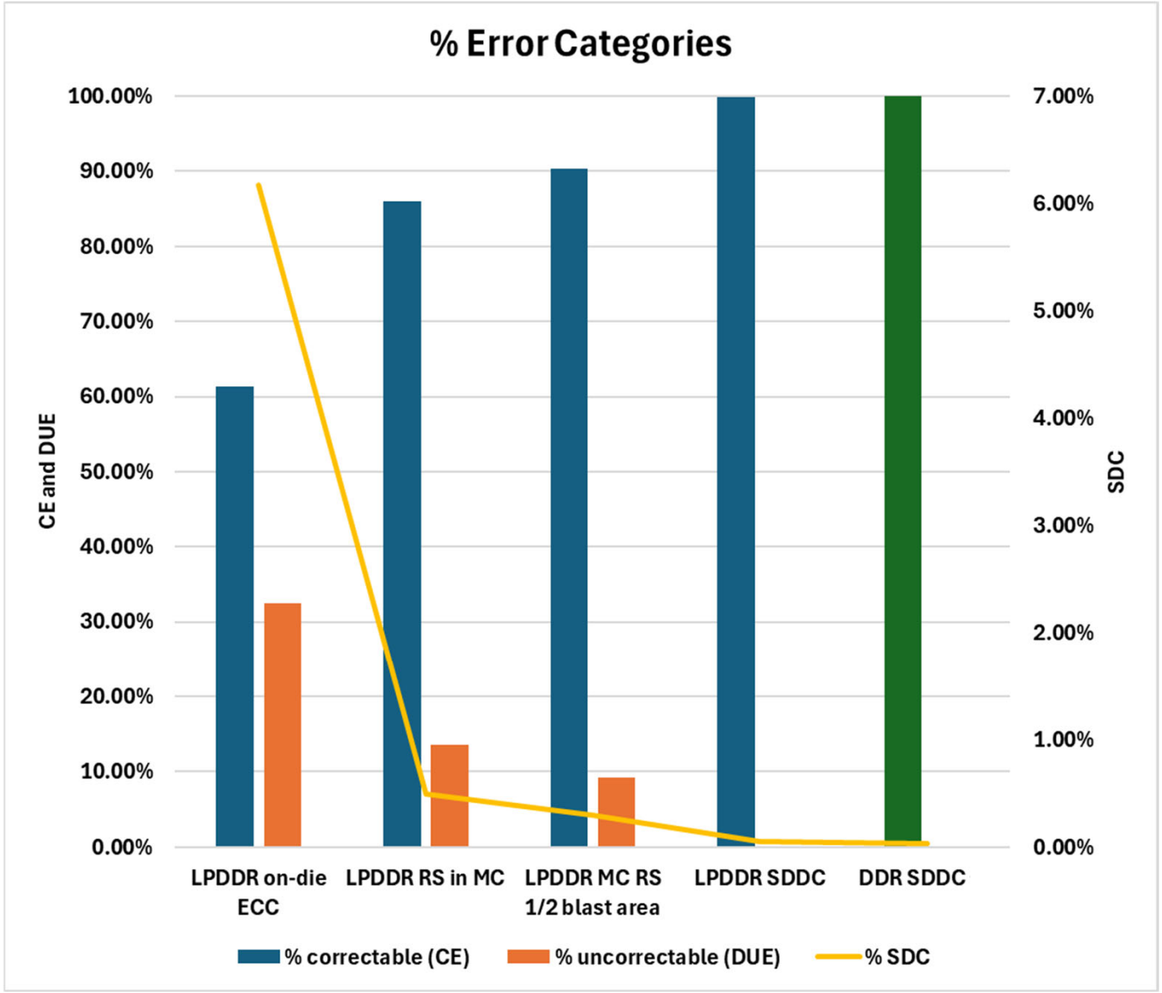
“Chip Guard ECC: An Efficient, Low Latency Method”, Tanj Bennett, 2023

- ECC logic should be small since it is duplicated across many channels
- Low latency is advantageous to reduce performance impact

- DDR5 x4 ECC DIMMs have 128-bits of ECC for 512-bits data (2 devices)
  - High (25%) overhead for SDDC
  - How can 128-bits be used more efficiently?
- Use parity to detect full chip failures
  - 64-bit parity across the devices & burst
  - Store 64-bits in one device
- Combine with a 48-bit ‘Signature’
  - Find the failing chip or detect a multi-chip failure
  - Construction is separable (split into parts)
- Extra bits leveraged for other functions
  - Poison, cache hints, capabilities (CHERI), …

System-level Error Correction Capabilities

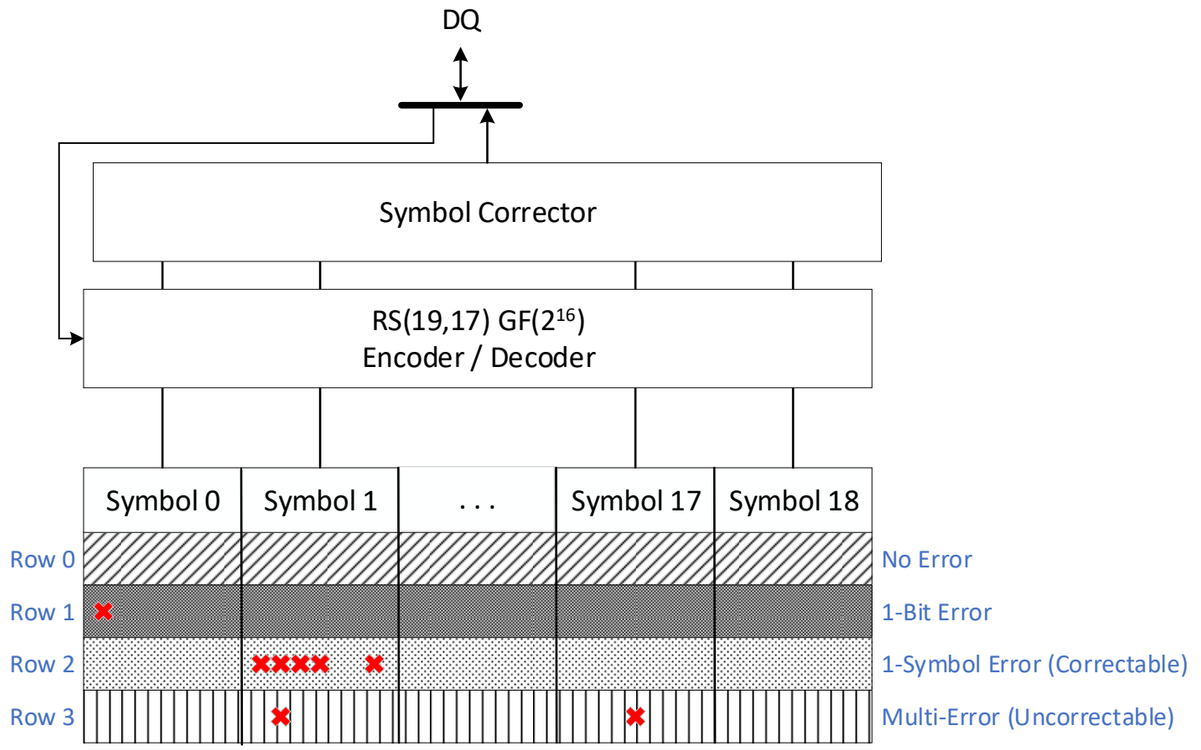
Error Correction	Memory Capacity Overhead	Max Error Correction Capability
On-die ECC	6.25%	1 bit per die
LPDDR Reed-Solomon in the memory controller using the on-die ECC bits	6.25%	16 bits per 64B
LPDDR Reed Solomon in the memory controller using on-die ECC, ½ blast area	6.25%	16 bits per 64B (the 16b covers more faults due to smaller blast radius)
LPDDR SDDC	50% to 100%	Die per rank
DDR SDDC	25%	Die per rank



Error rates based on the "A Systematic Study of DDR4 DRAM Faults in the Field". No similar field data exists to date for DDR5 or LPDDR.

Alternatively, Improve on-Die ECC Capability

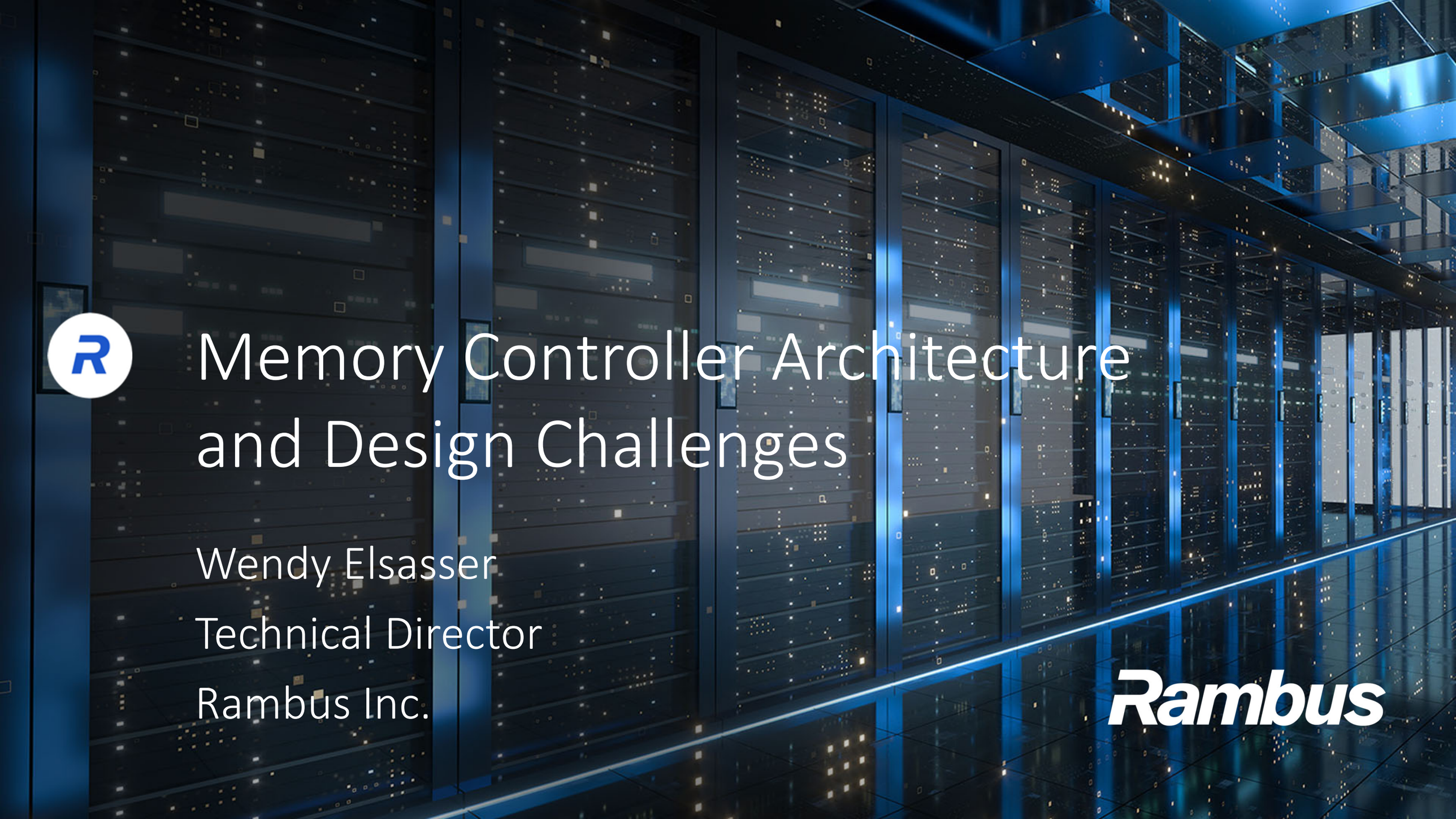
- DDRx & LPDDRx use SEC for on-die ECC
  - Covers single bit error correction
  - Fast and simple
- HBM has incorporated Reed Solomon coding for higher on-die ECC coverage
  - RS(19,17) GF(2^{16})
  - Corrects one 16-bit symbol
- Architecture determines fault coverage
  - I.e., Can correct sub-wordline 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 Recap

- Cost of traditional SDDC coverage has doubled from DDR4 to DDR5 (12.5% to 25%)
  - How to scale further for DDR6?
  - Are there viable, lower cost alternatives that make more efficient use of ECC bits
- RAS for LPDDR hasn't historically been a priority
  - With use in the data center this could be changing
- HBM has improved RAS capability with higher on-die ECC + meta-data
  - More optimal option for HBM with full burst accessed from a single channel (subset of a die)
  - Cannot simply replace when failures occur - failover or mapping out of bad regions required
- OS functions to handle errors triggered when acceptable thresholds are surpassed
  - Move data when feasible to failover or unused DRAM region
  - Map out erroneous page
  - Reconfigure page table for new address mapping
- Efficient and high RAS capability will continue to be an important topic

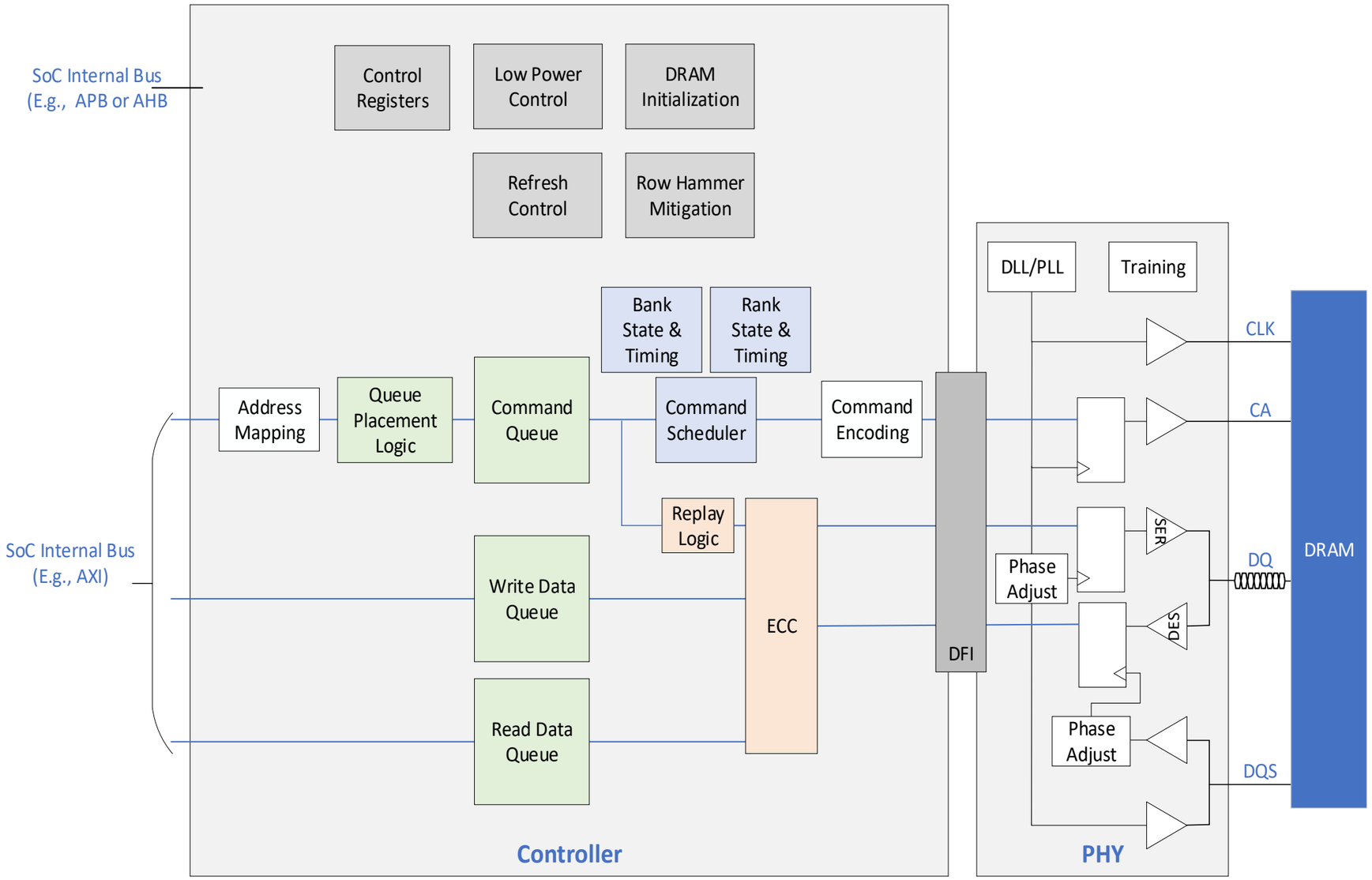


Memory Controller Architecture and Design Challenges

Wendy Elsasser
Technical Director
Rambus Inc.

Rambus

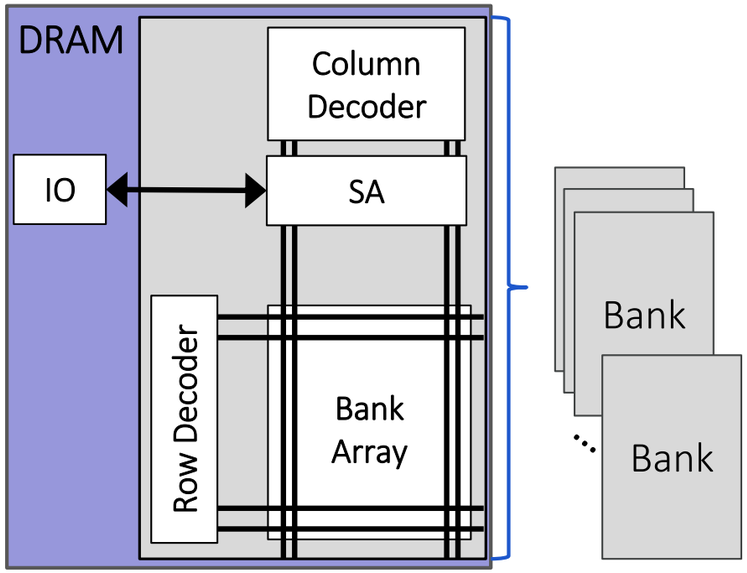
Host Controller & PHY



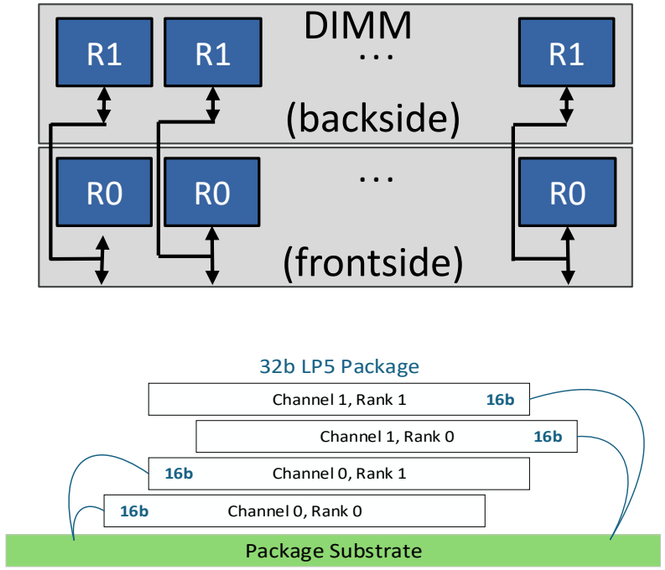
- Manages and schedules memory requests
  - Placement in queue
  - Out of order scheduler
- Maintains bank states (open, closed)
- Translates host physical address into DRAM protocol address
- Orchestrates DRAM management operations
  - Refresh
  - Row-hammer mitigation
- Times data transfer and receive (Tx, Rx)

Optimize Utilization across Parallel Resources

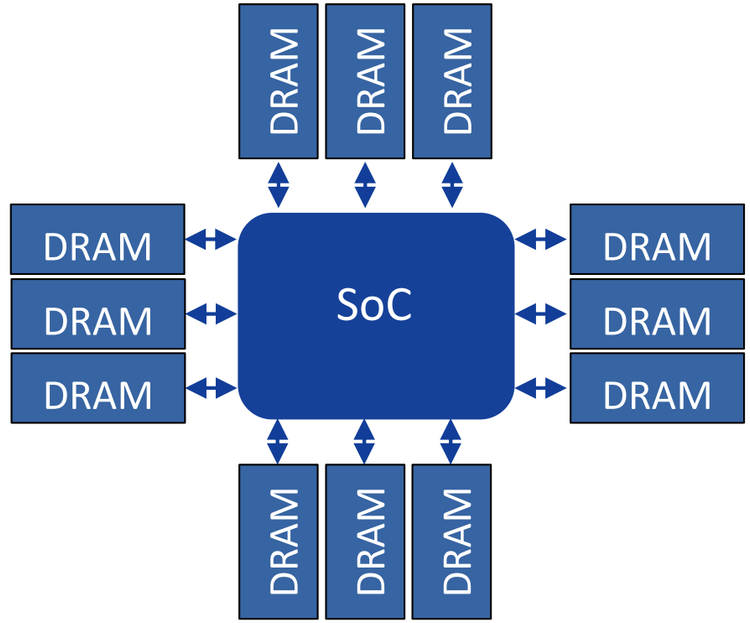
Bank Interleaving



Rank Interleaving



Channel Interleaving



- Higher BW per channel with random traffic
- Managed by physical address to DRAM Bank/Rank mapping in the host memory controller

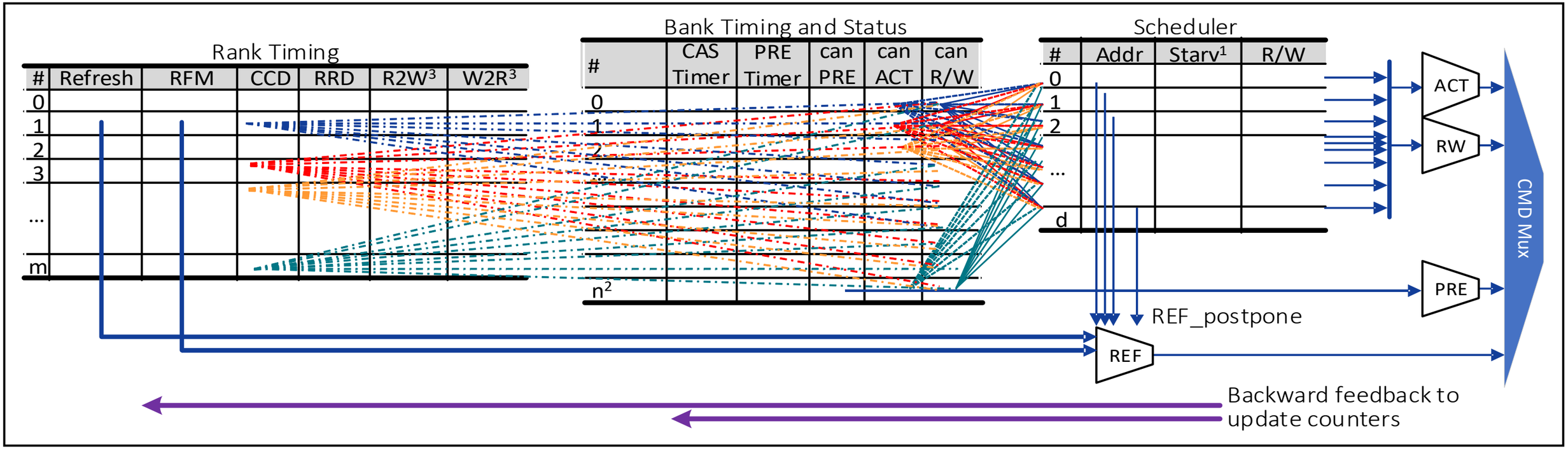
- Higher aggregate BW
- Managed by address mapping in SoC fabric (hashing, etc.)

SoC Interleaving across parallel resources for higher aggregate performance

Memory Controller Design Complexity

- Almost All-to-All connection between
  - Rank logic for channel CMD timing: CCD/RRD/R2W/W2R… & “same-bank refresh”+”RFM”
  - Bank logic for per-bankl 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

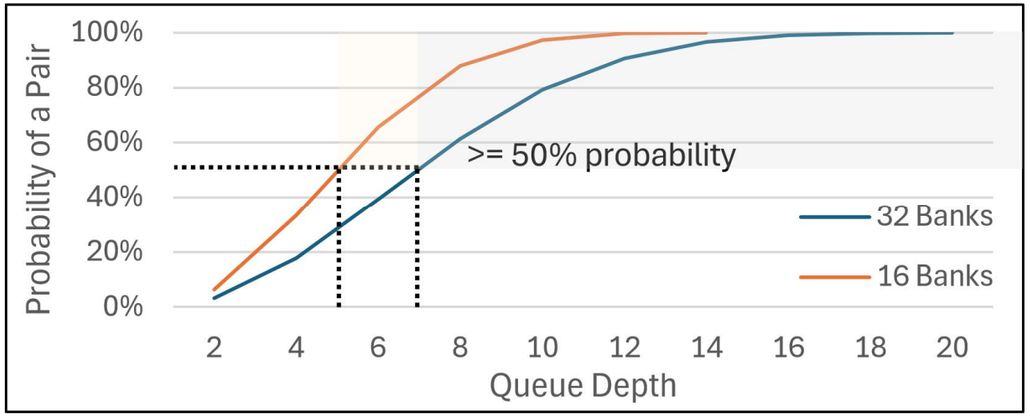
[1]Starvation timer
[2]n>tRC/tCCD
[3]Bus turnaround

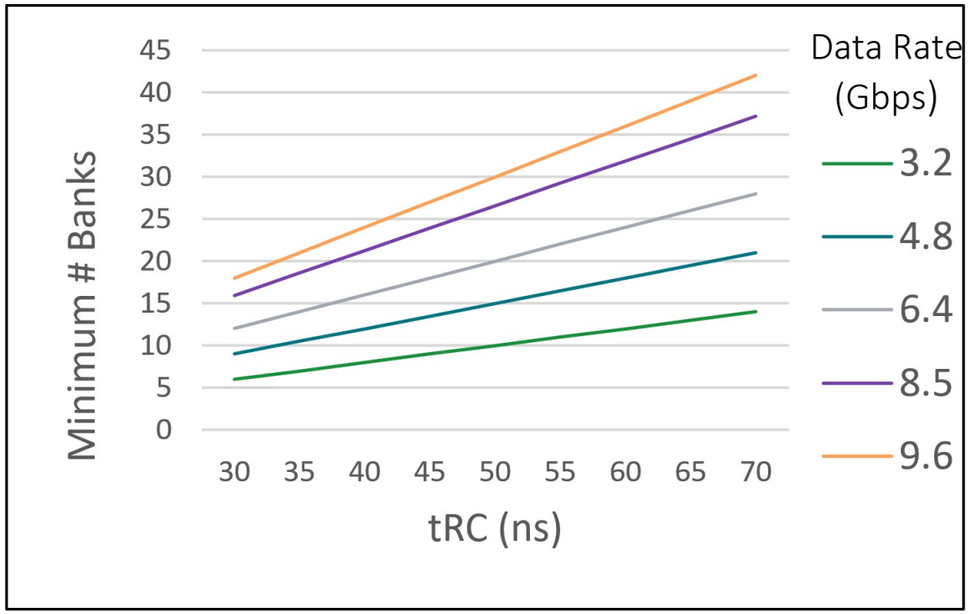


Queue Depth

Random access throughput & effective bandwidth versus complexity

- Higher data rates reduce tBURST
  - More banks required to maintain throughput
- Multiple commands in flight before initial R/W can be cleared from the queue
  - For tRL=16ns, tCCD_S=2ns → 8 Reads in flight
- Purely random traffic requires more banks
  - Birthday paradox probability of multiple commands to the same bank in the queue





- Host controller complexity is growing as systems evolve and grow (area, power)
- Scaling tradeoffs need to be evaluated across the system (DRAM + SoC)

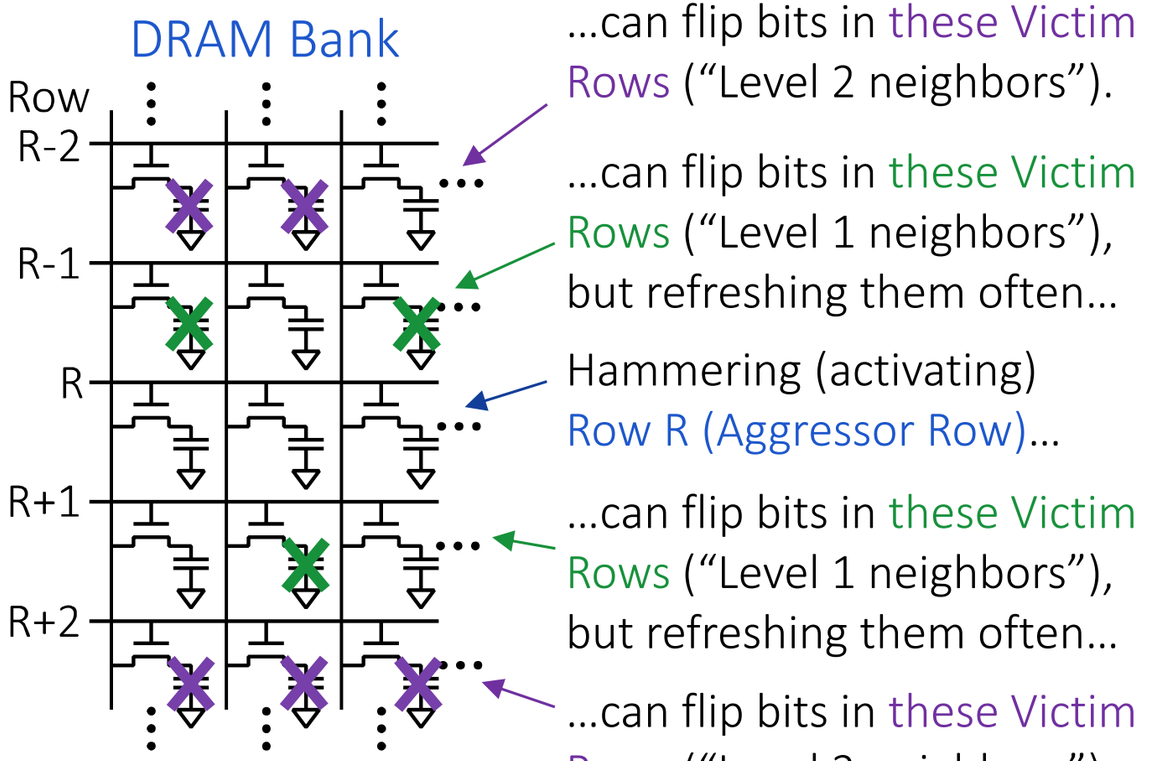
Security: RowHammer and RowPress

Steven Woo
Fellow and Distinguished Inventor
Rambus Inc.

Rambus

RowHammer is a Growing Concern

- Traditional attack: Repeated activates of a DRAM row cause neighboring row bit flips
- Half-Double attack: Refreshing victims stops bit flips, but hammers victim neighbors
- Hammer Count (HC): Activate count to flip bits in victim rows
  - HC has fallen more than a factor of 10 over the past decade [Orosa, et. al. 2021], continuing to fall at smaller process nodes
  - More neighboring rows can be affected
  - Extrapolating recent data, HC may reach 1K-3K
- RowPress: Leaving pages open after an Activate operation can accelerate bit flips

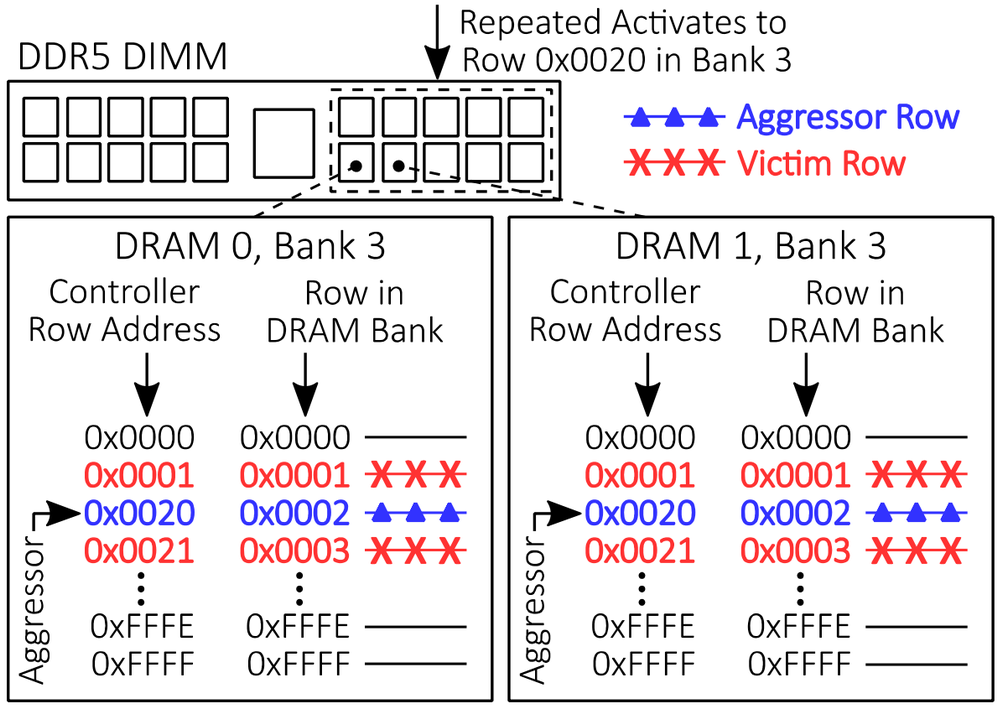


Traditional Attack: Flip bits in neighboring (victim) rows

Half-Double Attack: Victim Refreshes hammer Victim Row neighbors

Even with ECC, RowHammer/RowPress are Problems for Servers

- DRAMs on a DIMM map controller row addresses to rows in the DRAM core in the same way
  - Mapping isn't public, repaired rows are an exception
- ECC can correct a limited number of bit errors
- 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)

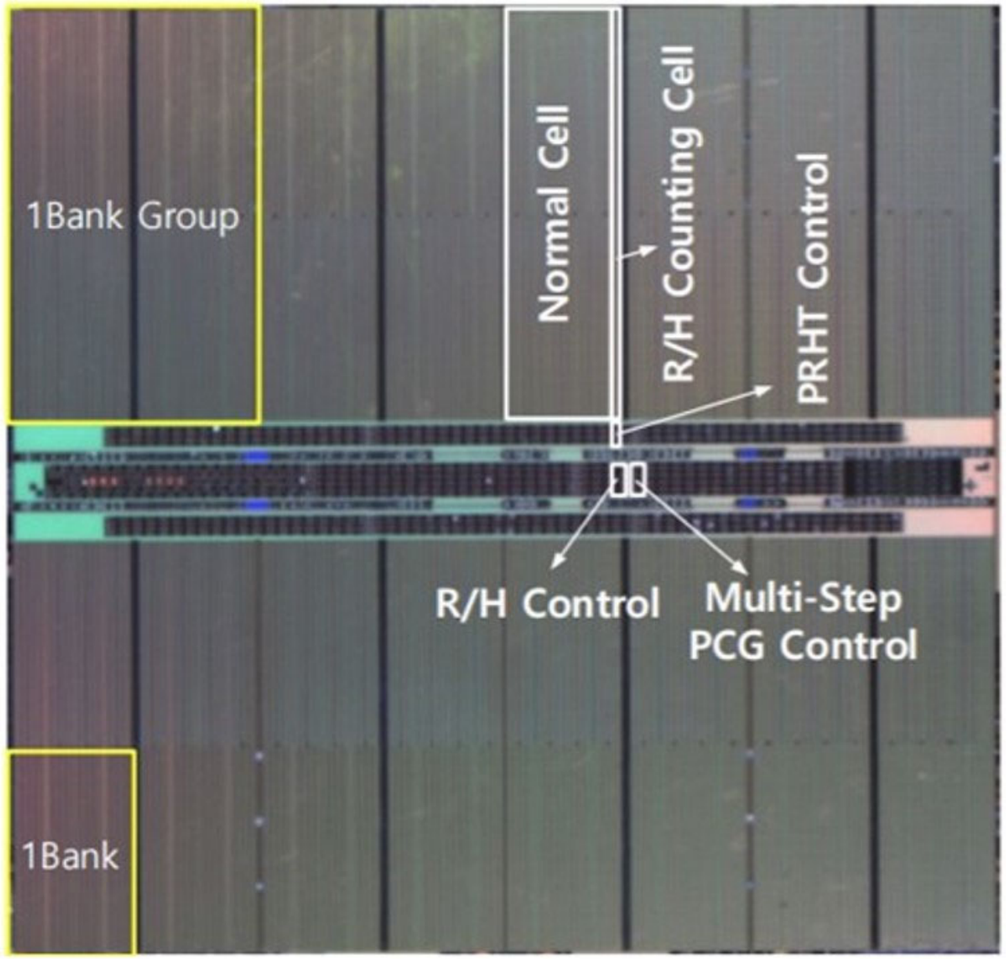


Fundamental problem: Row addresses can be neighbors in more than 1 DRAM, leading to many potential bit flips that can overwhelm ECC methods

Preventing RowHammer/RowPress Bit Flips

- Controller-based tracking methods
  - Track memory accesses, look for rows being activated frequently
  - Tell DRAM to proactively refresh potential victim rows when thresholds are reached
  - Challenges
    - Manufacturer-specific thresholds are not published
    - Storage for tracking increases as Hammer Counts fall, more channels are added to a system
    - Controller doesn't know which rows are neighbors in the DRAM
- DRAM-based tracking methods
  - Additional storage for tracking in the DRAM
  - DRAM can perform proactive victim refresh operations
  - DRAM can signal controller to stop transaction pipeline if too many rows reaching threshold
  - Challenges
    - More storage per DRAM for tracking
    - Potential performance impact (e.g., longer row cycle times) for tracking in DRAM

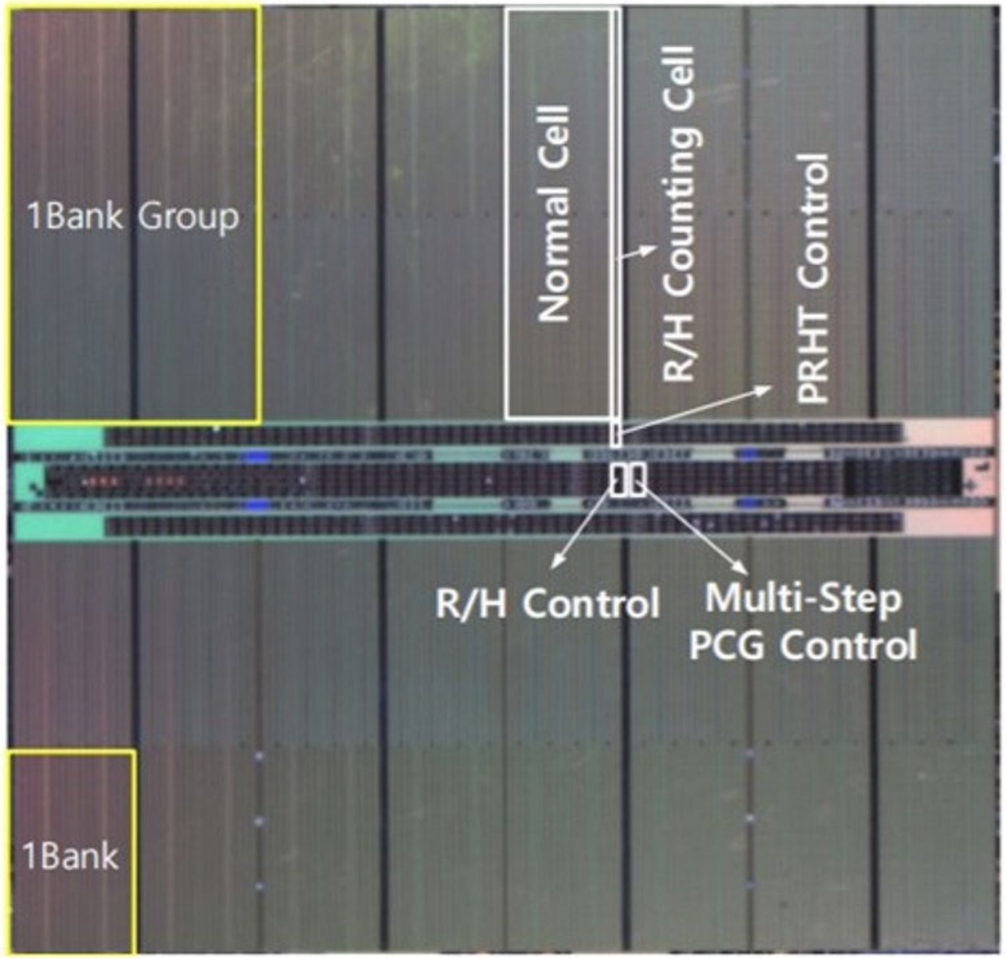
PRAC: Per-Row Activation Counters (1)



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.

- 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 (small overhead)
  - If PRAC counter is 16b, this is an additional overhead of 0.2% storage per 1KB page
  - Compare with DDR5 on-die ECC overhead of 64B per 1KB row (6.25% additional on-die ECC storage)
- Counters incremented when pages are closed, increment value based on activation 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 can be alerted via Alert pin 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
- 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

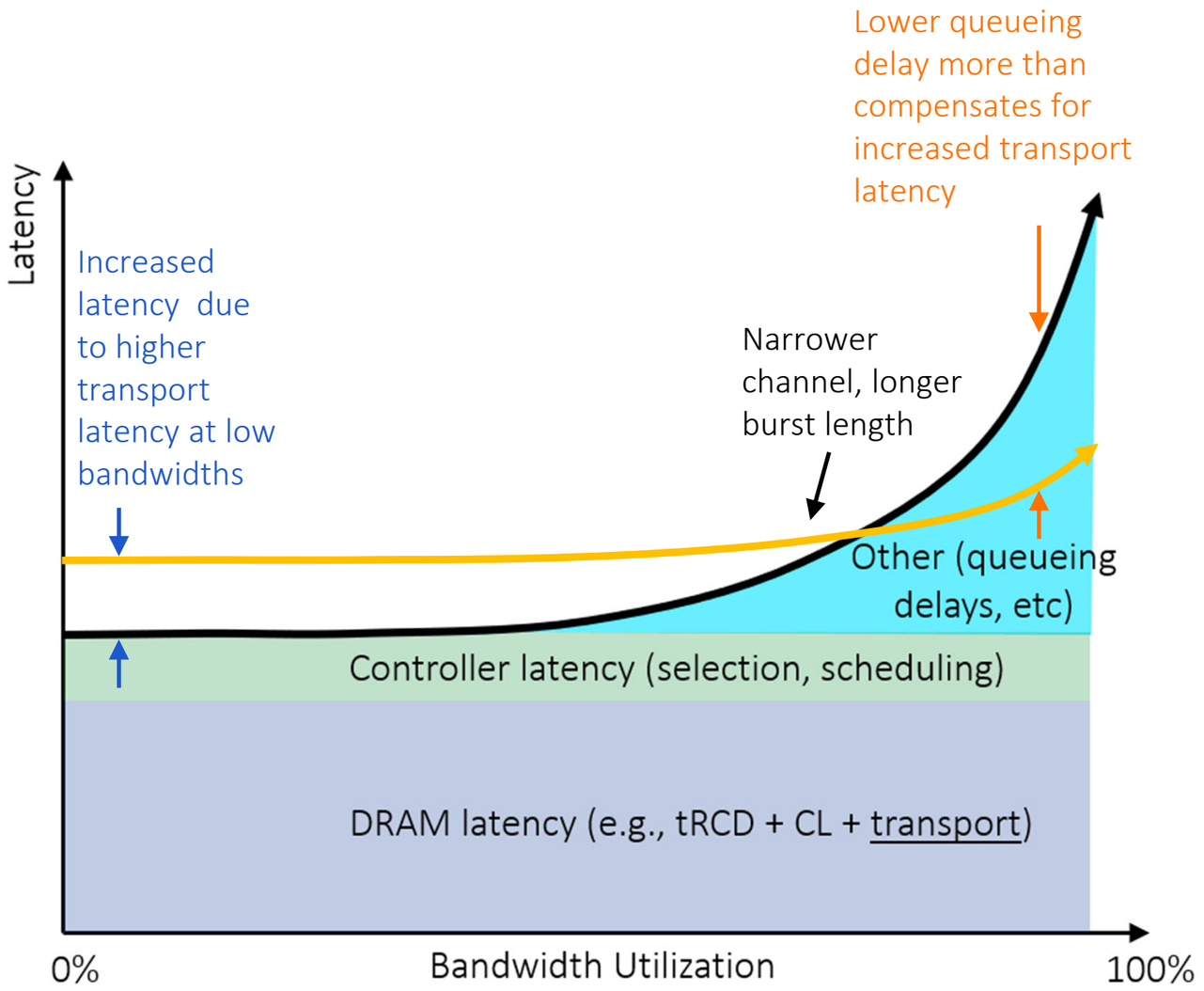
System Performance

Steven Woo
Fellow and Distinguished Inventor
Rambus Inc.

Rambus

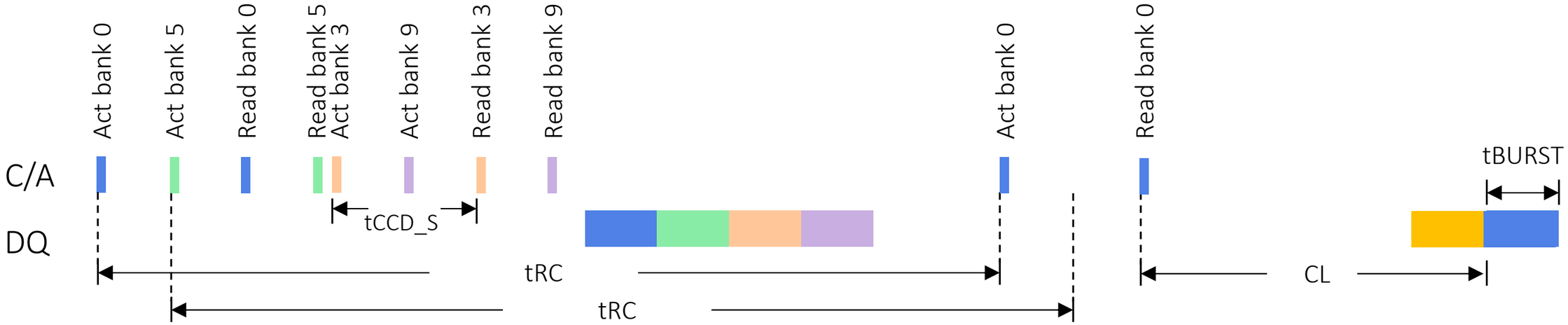
Latency Under Load

- Important for understanding memory system performance
- Often look at random workloads - random bank, row, and column addresses
- Affected by core timing parameters, number of banks, queue depths
- Narrower channels, longer burst lengths
  - Achieves lower latency at higher bandwidths due to less queueing (right side of curve)
  - But increases transport latency, which hurts at lower bandwidths (left side of curve)



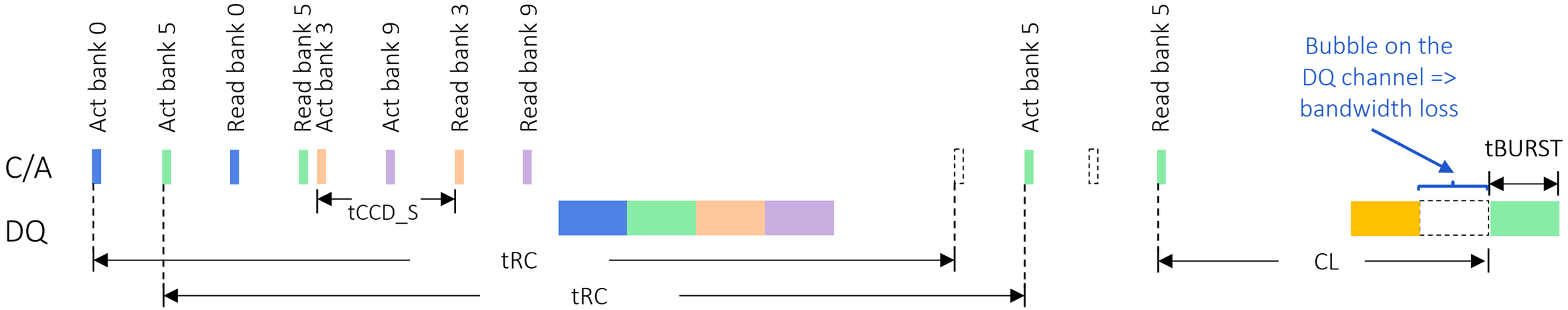
Latency Under Load curves are important for characterizing the behavior of memory systems

100% Read Transaction Pipeline (1)



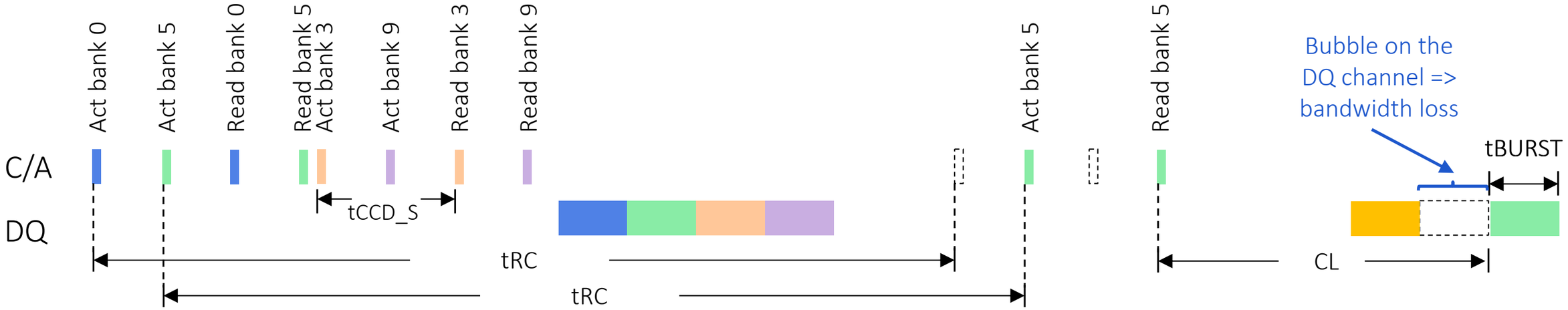
- Under random traffic, to keep the pipeline full (100% bandwidth) we need a sufficient number of banks equal to $\mathbf{Ceiling}\left[\frac{tRC}{tBURST}\right]$
  - For tRC=45ns, tBURST=2.5ns (similar to DDR5-6400), need at least $\mathbf{Ceiling}\left[\frac{45}{2.5}\right]$ = 18 banks
- If we don't have enough banks, wait until tRC finishes before activating another row in bank same bank
  - In the above example, must wait until tRC for bank 0 finishes before activating another row in this bank
- Having more banks is helpful for random traffic
- Having deeper transaction queues is helpful for random traffic

100% Read Transaction Pipeline (2)



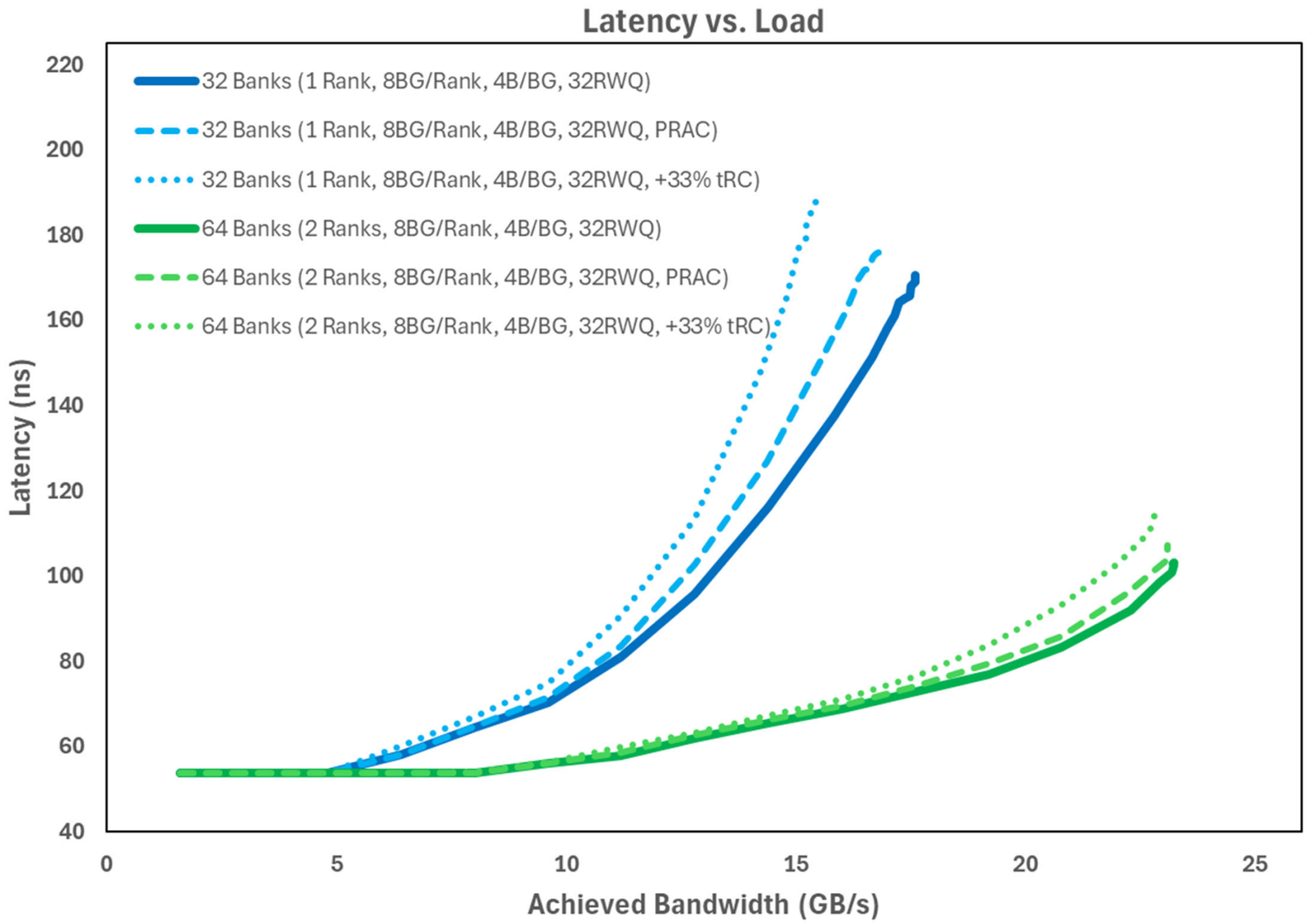
- Having more banks is helpful for random traffic
  - In a channel with 32 banks, the probability of the 18th transaction conflicting with one of the previous 17 transactions is $\left(\frac{17}{32}\right)$ = 53.1%
  - If the 18th transaction has a bank conflict with 2nd transaction (bank 5), need to wait until tRC is finished => bubble in the DQ channel, reducing bandwidth and performance
  - In a channel with 64 banks, the probability of the 18th transaction conflicting with one of the previous 17 transactions is $\left(\frac{17}{64}\right)$ = 26.5%

100% Read Transaction Pipeline (3)



- Having deeper transaction queues is helpful for random traffic
  - In a channel with 32 banks, if we have 17 transactions in flight and 2 queued transactions that we can choose between, the probability of the 2 queued transaction both conflicting with one of the previous 17 transactions is $\left(\frac{17}{32}\right)^2$ = 28.2% (compared to 53.1% with 1 queued transaction)
  - In a channel with 32 banks, if we have 17 transactions in flight and 8 queued transactions that we can choose between, the probability of the 8 queued transaction all conflicting with one of the previous 17 transactions is $\left(\frac{17}{64}\right)^8$ = 0.63% (compared to 53.1% with 1 queued transaction)

Impact of Bank Count and Row Cycle Times

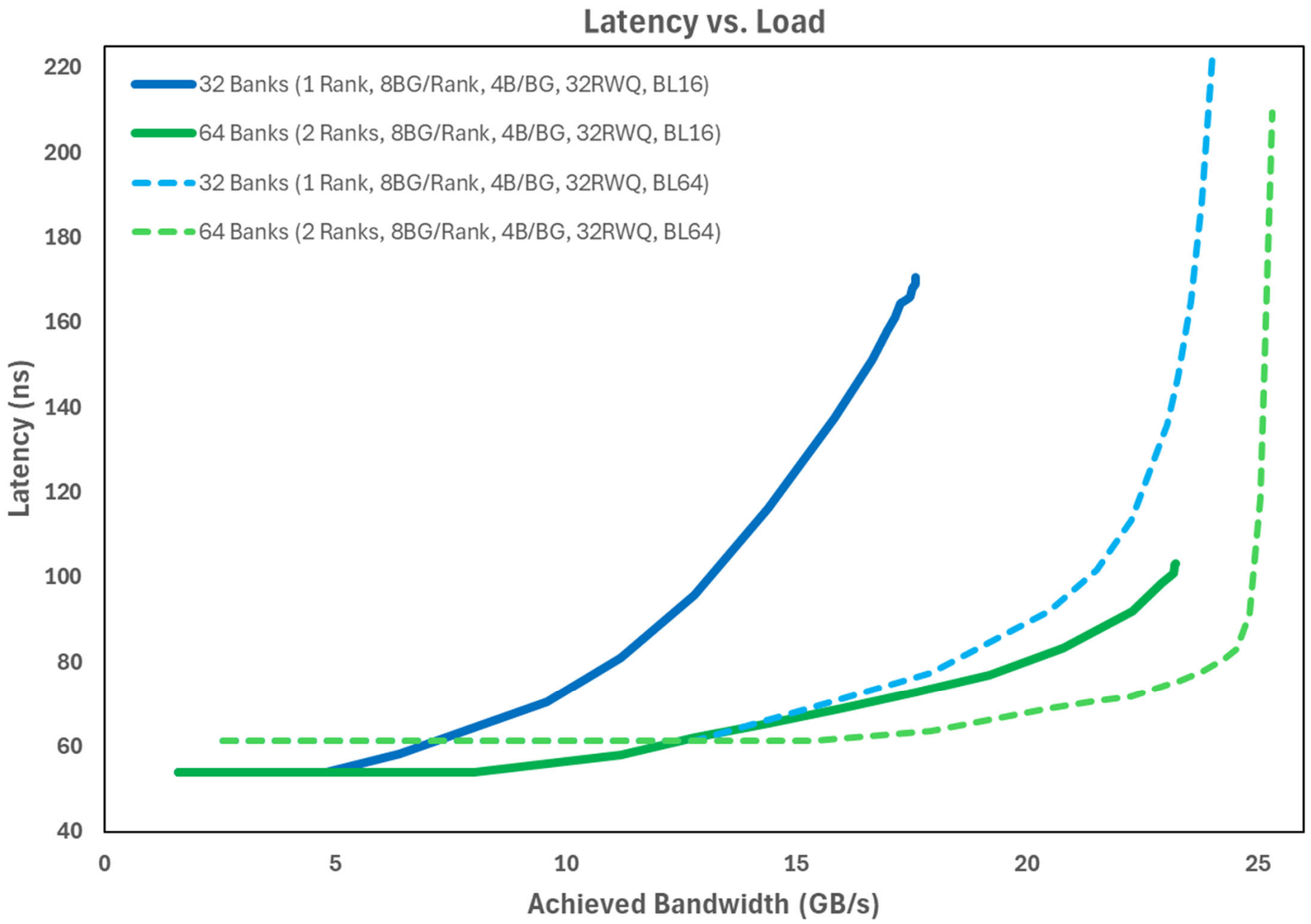


- Having more banks reduces latency under load
- Shorter row cycle times also reduce latency under load

DRAMSys Simulation Setup:

- DDR5-6400 core timings, 8 Bank Groups, 4 Banks per Bank Group
- 1 and 2 ranks
- Separate Read and Write queues (32 entries each)
- Random accesses
- 67% Reads, 33% Writes
- Closed page policy

Impact of Longer Burst Lengths with Narrower Channels

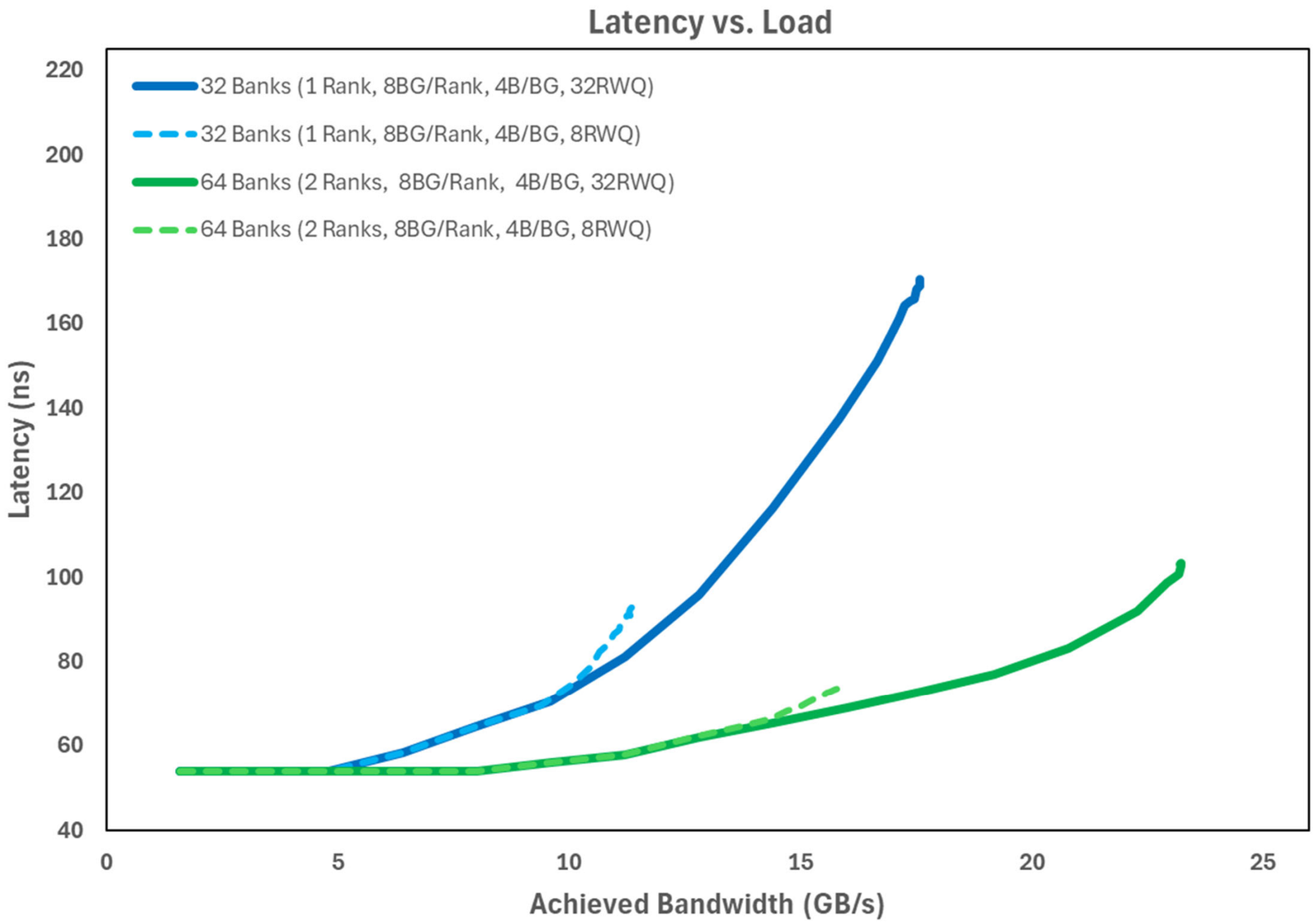


- Longer burst lengths reduce the number of banks required for full bandwidth under random traffic

DRAMSys Simulation Setup:

- DDR5-6400 core timings, 8 Bank Groups, 4 Banks per Bank Group
- 1 and 2 ranks
- Burst length 16 and 64 (narrower channels to achieve same granularity)
- Separate Read and Write queues (32 entries each)
- Random accesses
- 67% Reads, 33% Writes
- Closed page policy

Impact of Scheduling Queue Depth



- Deeper queues reduce bank conflicts and improve latency under load

DRAMSys Simulation Setup:

- DDR5-6400 core timings, 8 Bank Groups, 4 Banks per Bank Group
- 1 and 2 ranks
- Separate Read and Write queues (8 and 32 entries each)
- Random accesses
- 67% Reads, 33% Writes
- Closed page policy

Future Memory Solutions:
MRDIMM, SOCAMM, CXL & PIM/PNM

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

Outline

- Introduction
- Next Generation DRAM Module
  - Multiplexed Rank DIMM (MRDIMM)
  - LP Compression-Attached Memory Module (CAMM)
  - 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

MRDIMM

* Multiplexed Rank DIMM

Client

(Laptop, AI PC, High-end WS)

SOCAMM2

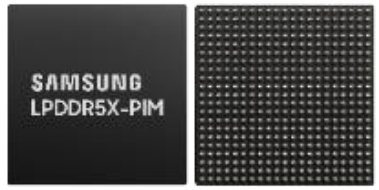
* Small Outline CAMM2

LPCAMM2

* LPDDR Compression
Attached Memory Module

Mobile

Low Power
(On-device AI)

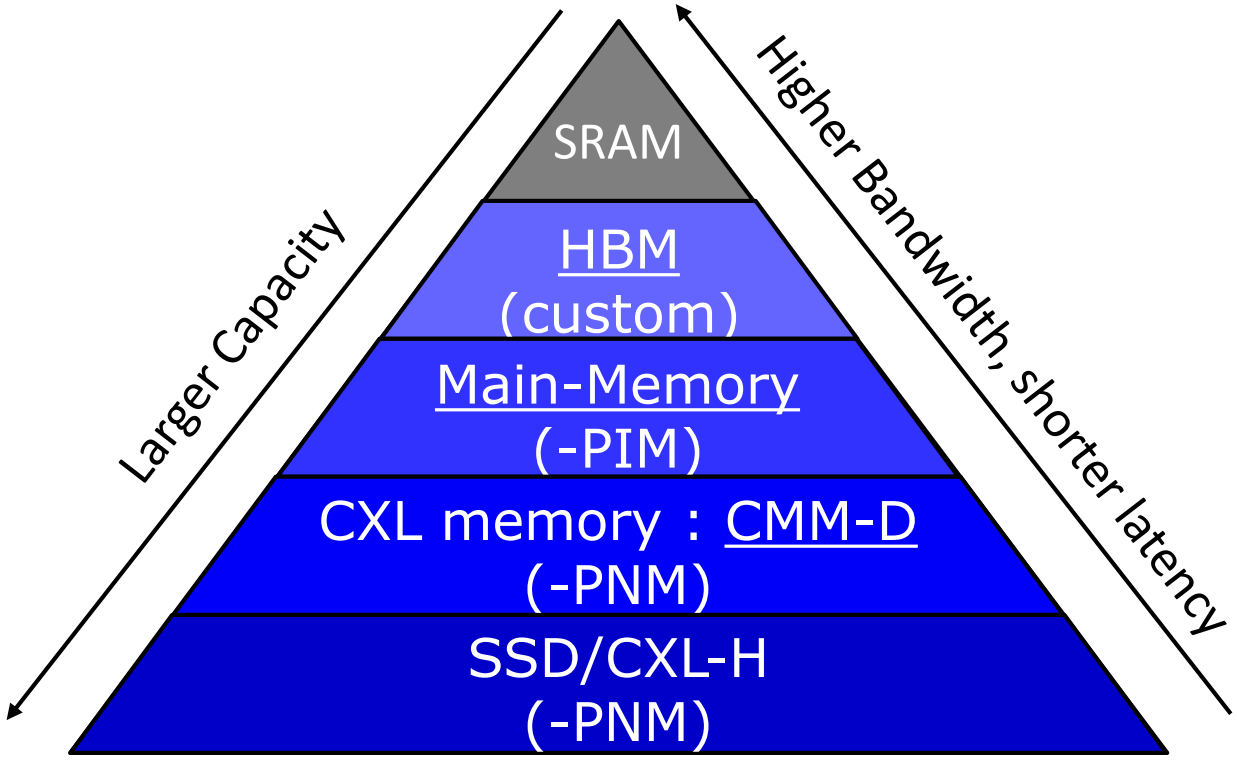


LPDDR-PIM

*LPDDR Processing In Memory

Next-Gen Memory Solutions

- Memory hierarchy is getting more advanced and efficient

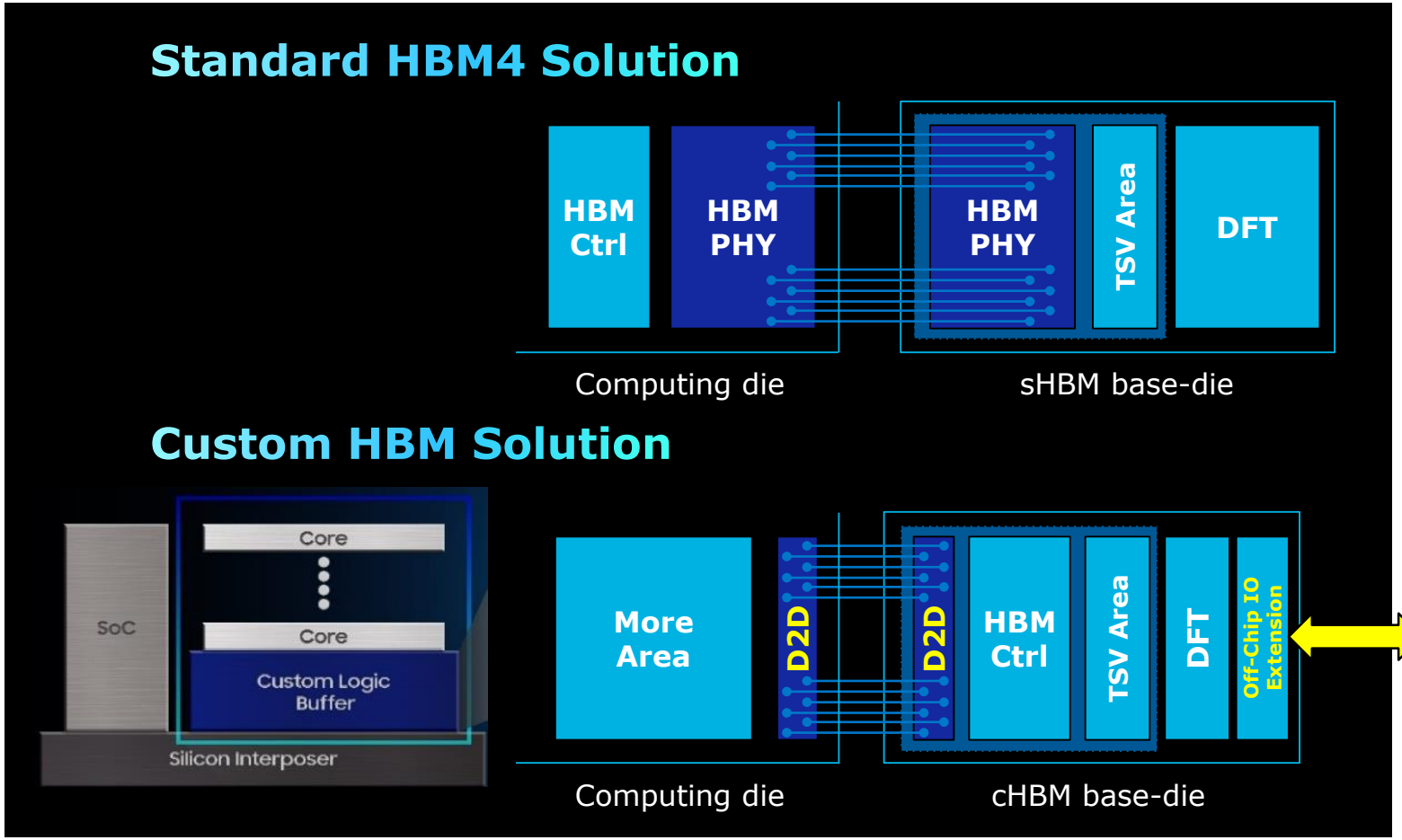


Key Technology for HBM

- 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

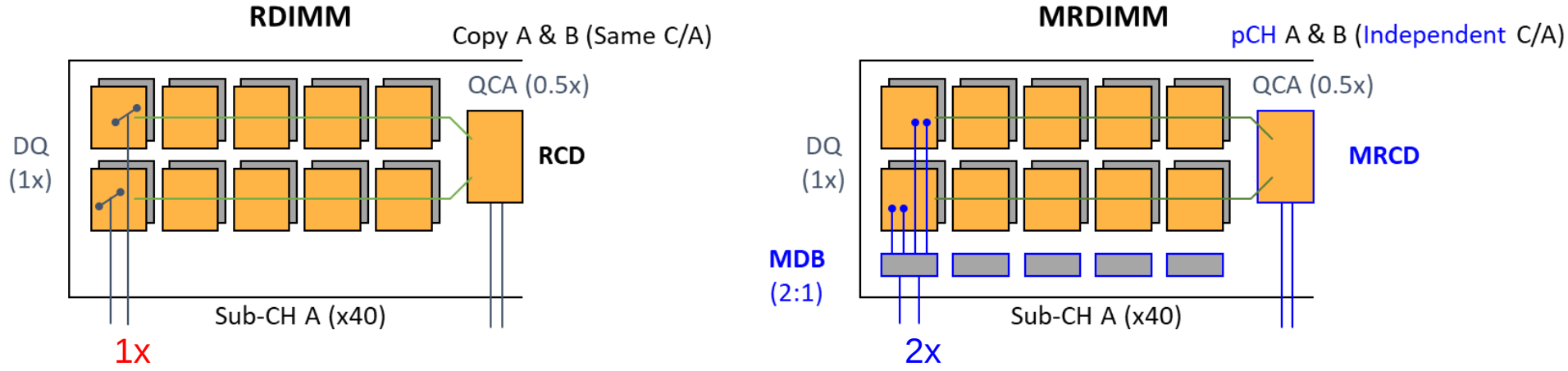
- Custom buffer die includes functions optimized for customers' individualized needs



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

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

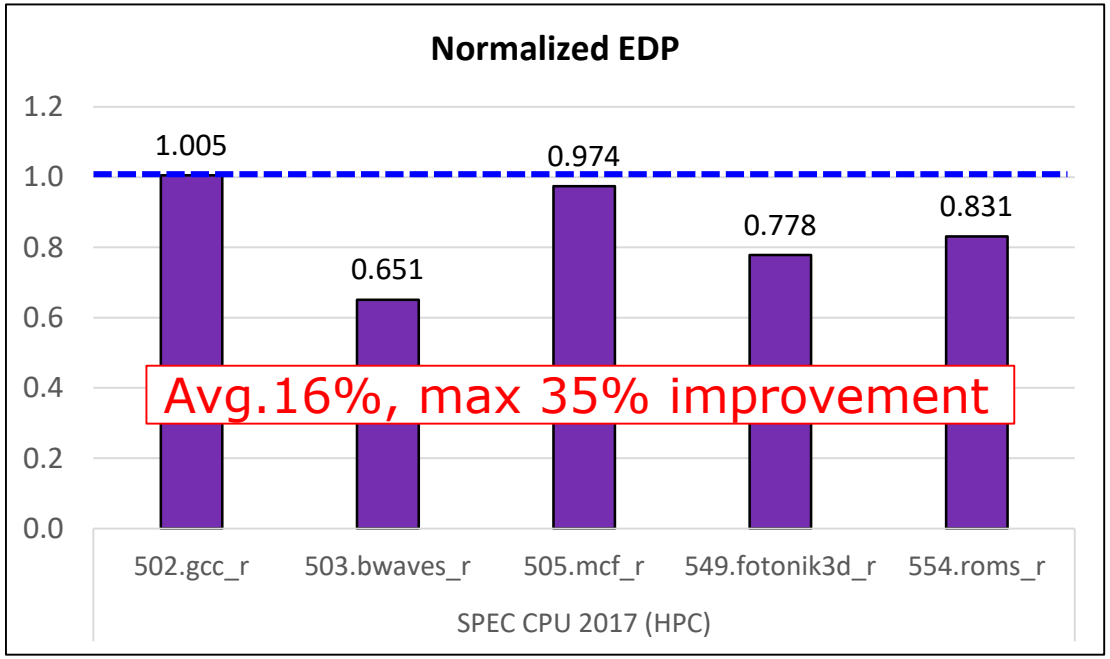
- Computation = Digesting Data
  - Higher core count CPU requires higher-BW and higher-capacity memory
    - Need to maintain constant BW/core & GB/core
    - CPU core count: 64 → 96 → 192 → 256 → …
  - 3D-Stacked RDIMM can provide 2x or 4x capacity, but no BW scaling
    - Should enhance both BW and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

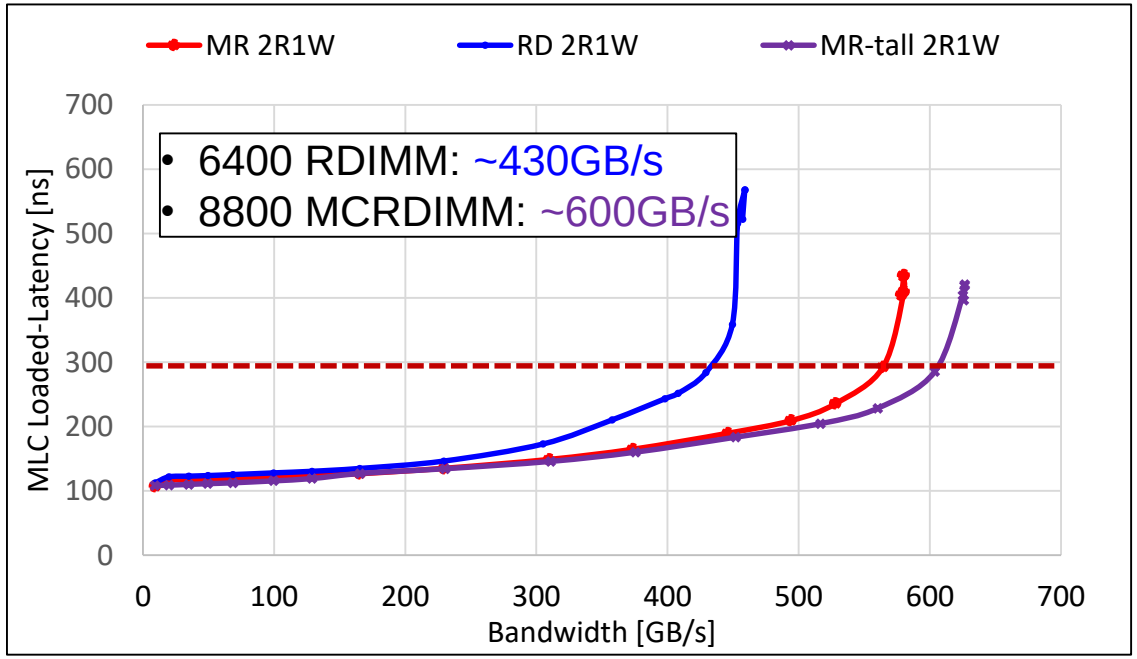
- MRDIMM is the solution for memory-intensive workloads
  - BW: 8.8Gbps (Gen1) → 12.8Gbps (Gen2) → 14.4Gbps (Gen3) → >16Gbps (Gen4)
  - Capacity: 2U Tall-DIMM has total 80-DRAMs (2x capacity)

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

MRDIMM Performance Benefit

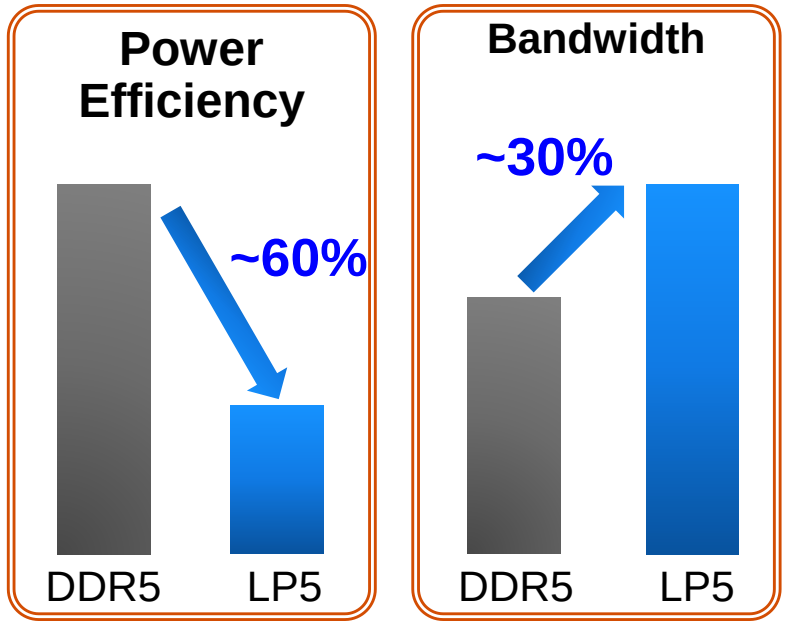
- System-level (CPU+MRDIMM) EDP (Energy-Delay-Product) BM
  - SPEC BM: MCRDIMM 8800 vs. RDIMM 6400 with Intel CPU
  - Despite higher power consumption of MRDIMM, shorter app runtime leads to overall EDP improvement
- Loaded Latency
  - 4-rank tall-MRDIMM can provide > 600GB/s (from 12-channel config)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

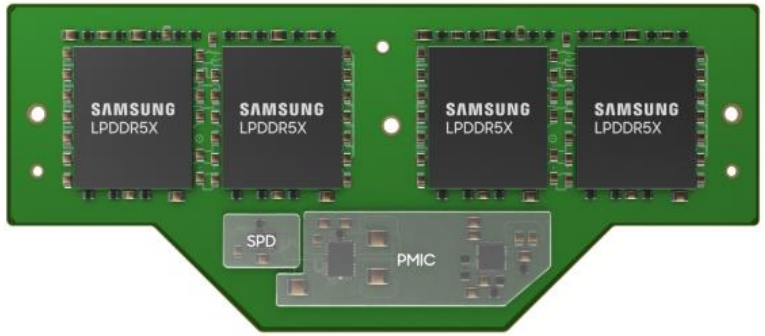


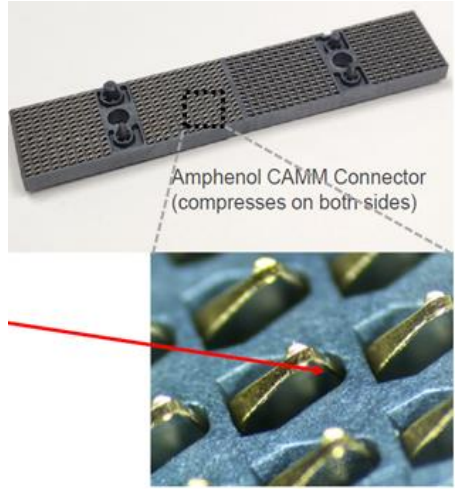


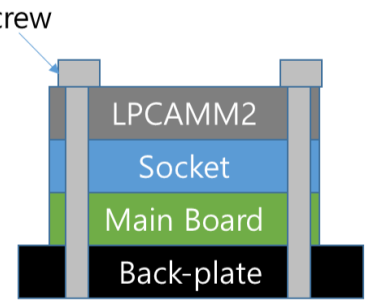
Why LPDDR-Based Module Solution

- LPDDR is now a module component for DC server, workstation and laptop
  - Lower power, Higher per-pin BW than DDR5, Multi-die PKG for higher capacity
- Concerns
  - Reliability (no ECC die/package unlike RDIMM)



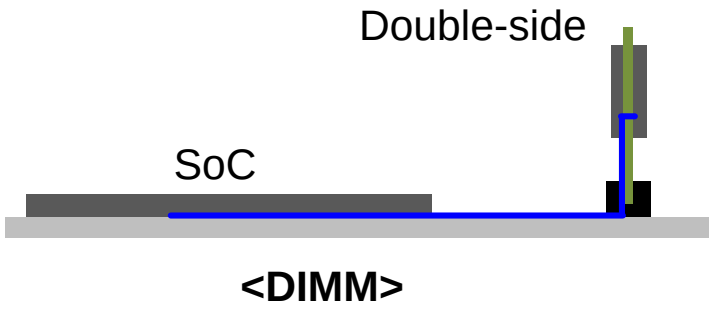


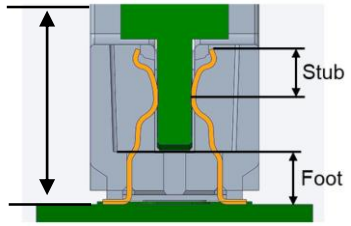


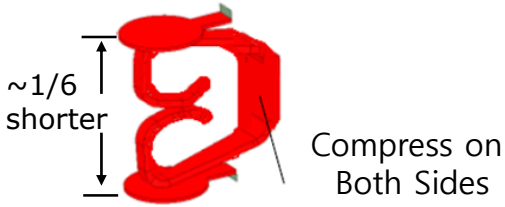


Compression Attached Memory Module

- **CAMM exhibits substantially enhanced SI performance**
  - Advantageous at speeds exceeding 10-Gbps

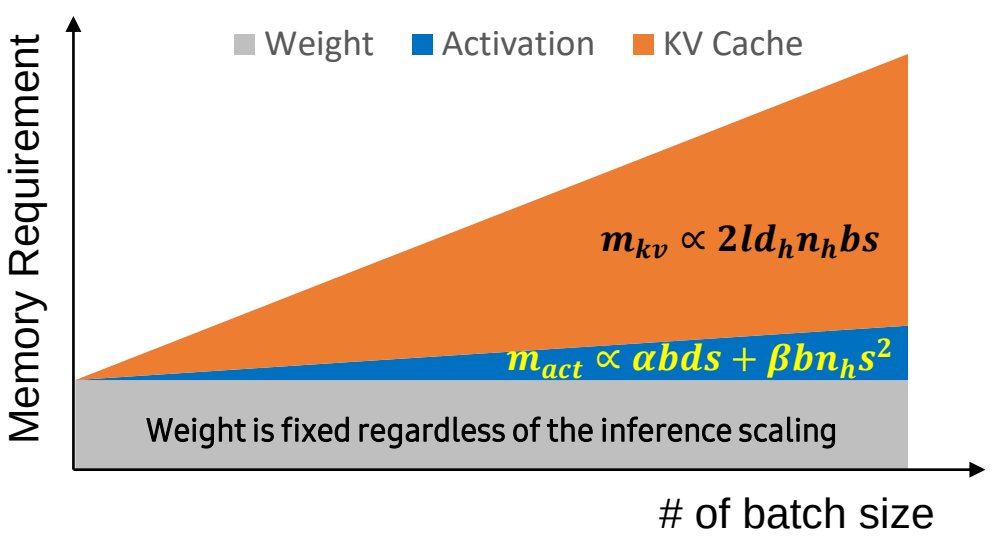


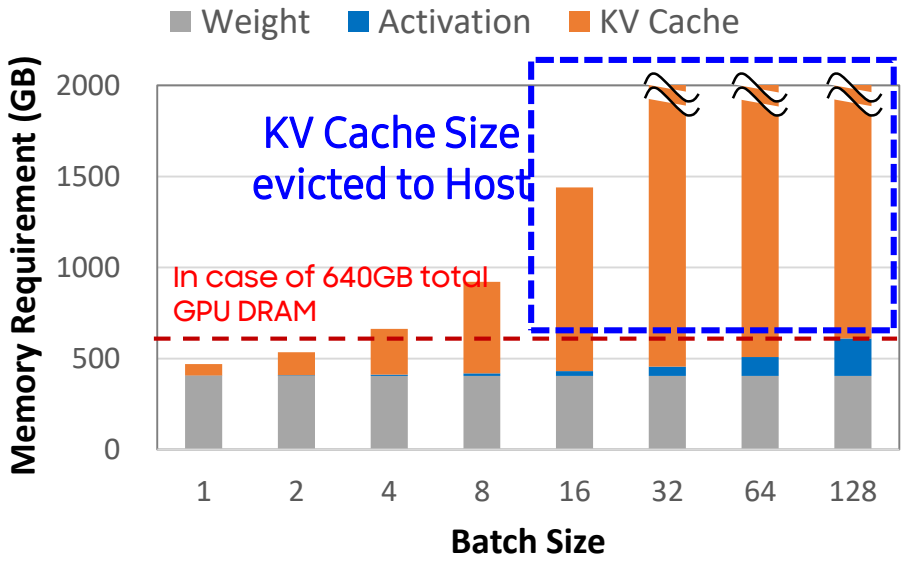




Memory Bottlenecks for AI Applications

- 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

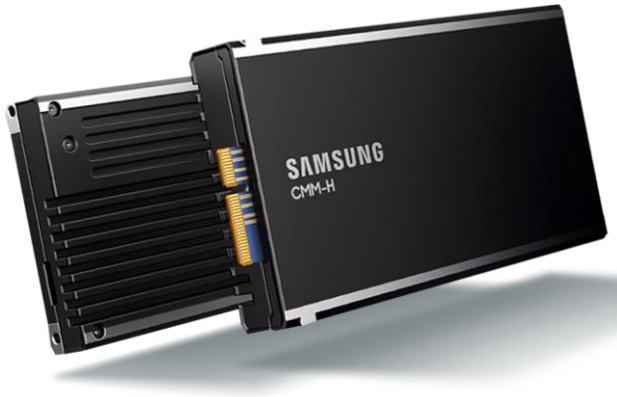




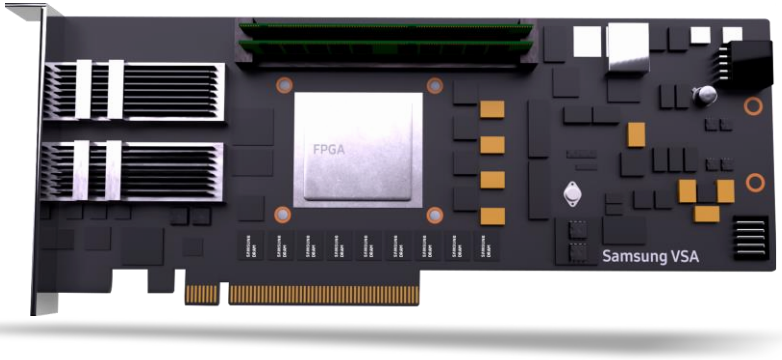
Scaling Memory Capacity and Bandwidth with CXL

- Memory capacity & BW expansion through serial/fabric-interconnect protocol

CMM-D
DRAM Expansion



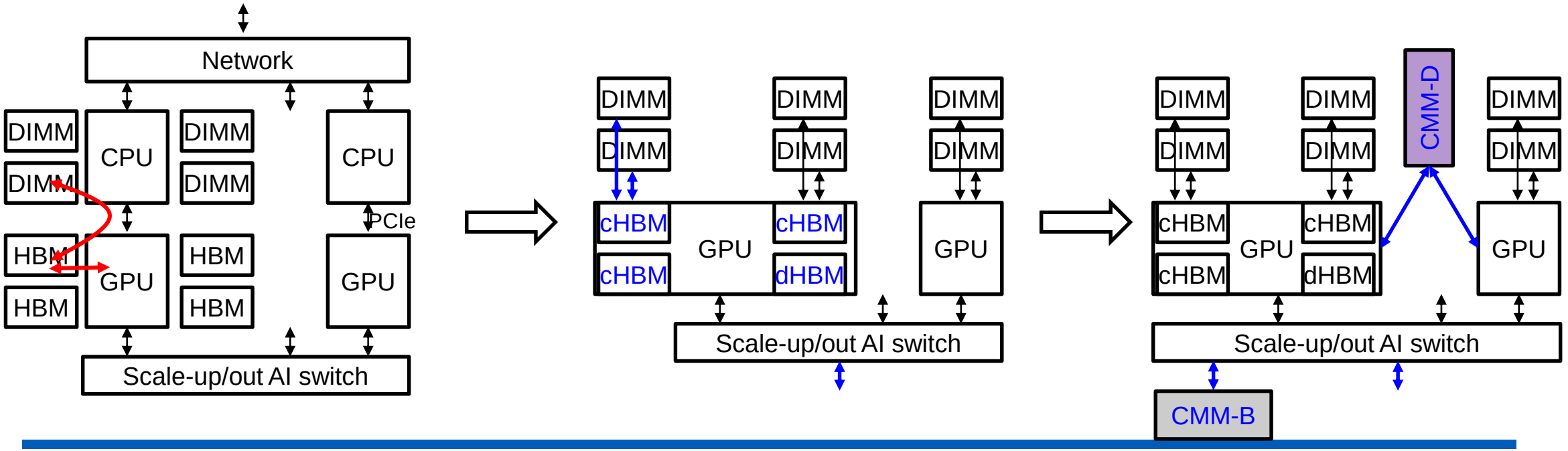
CMM-H PM
Persistent Memory



CMM-H TM
Tiered Memory

CXL & Fabric Memory in DC

- Megatrend in AI server is "Bring data closer to the compute"
  - 2.5D HBM → custom-HBM
    - Directly-connected DIMM to GPU
    - DIMM to cHBM data transfer without CPU → GPU IO-die has both cHBM and DIMM PHYs
  - Issue: limited capacity by brining all data closer to GPU
- CXL and fabric-attached memory for AI servers
  - Less sensitive to latency and cost

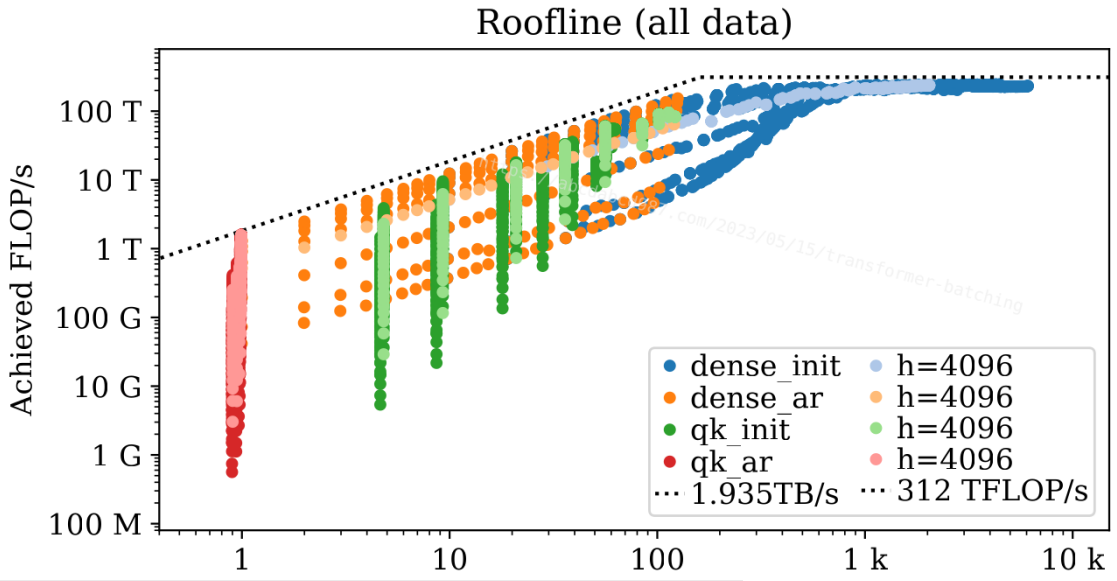


Outline

- Introduction
- Next Generation DRAM Module
  - Multiplexed Rank 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

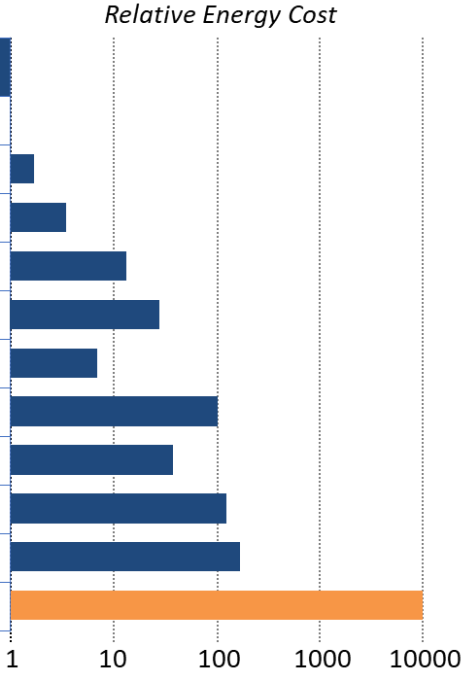


_init: summarization stage
_ar: generation stage (auto-regression stage)
dense: Linear layers (layers w/o self-attention)
qk: self-attention layers

[Dissecting Batching Effects in GPT Inference (gun.ch)]

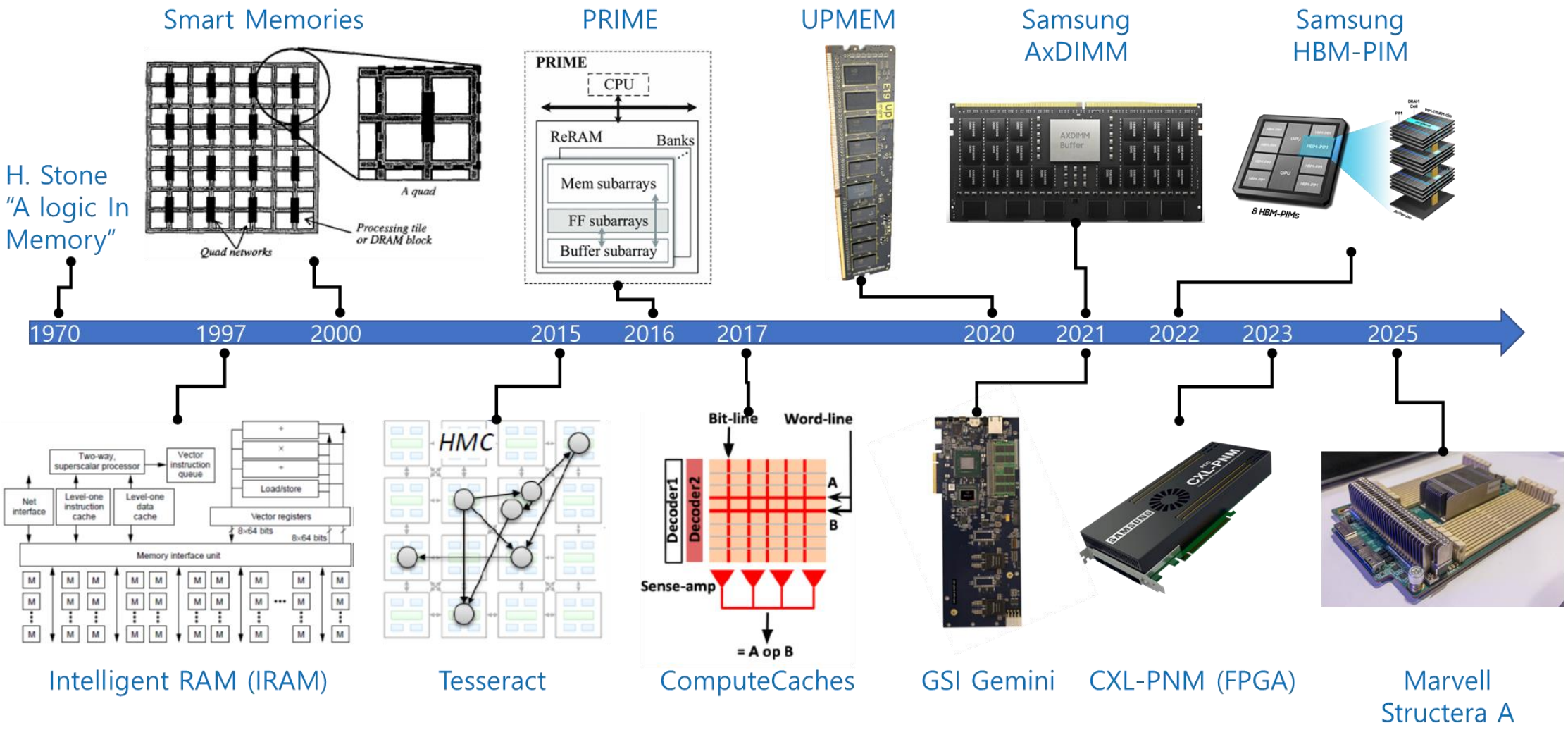
Relative Energy Cost

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



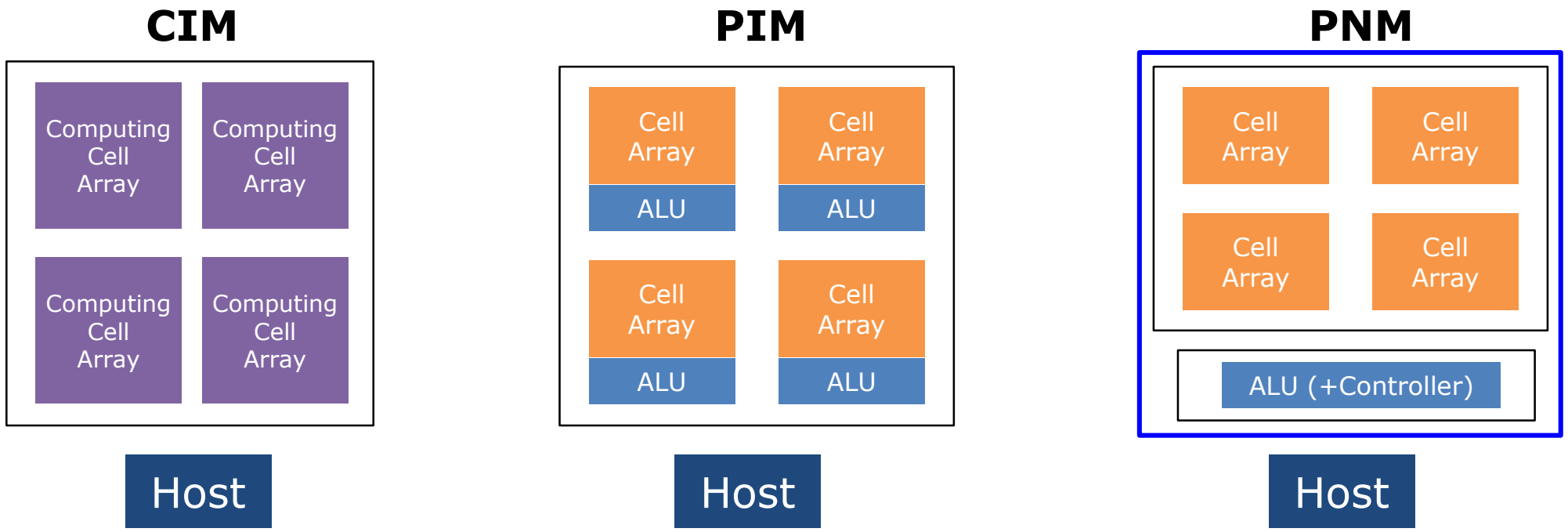
Source : Computing's Energy Problem (and what we can do about it) (ISSCC'14)

History of Processing in/near Memory



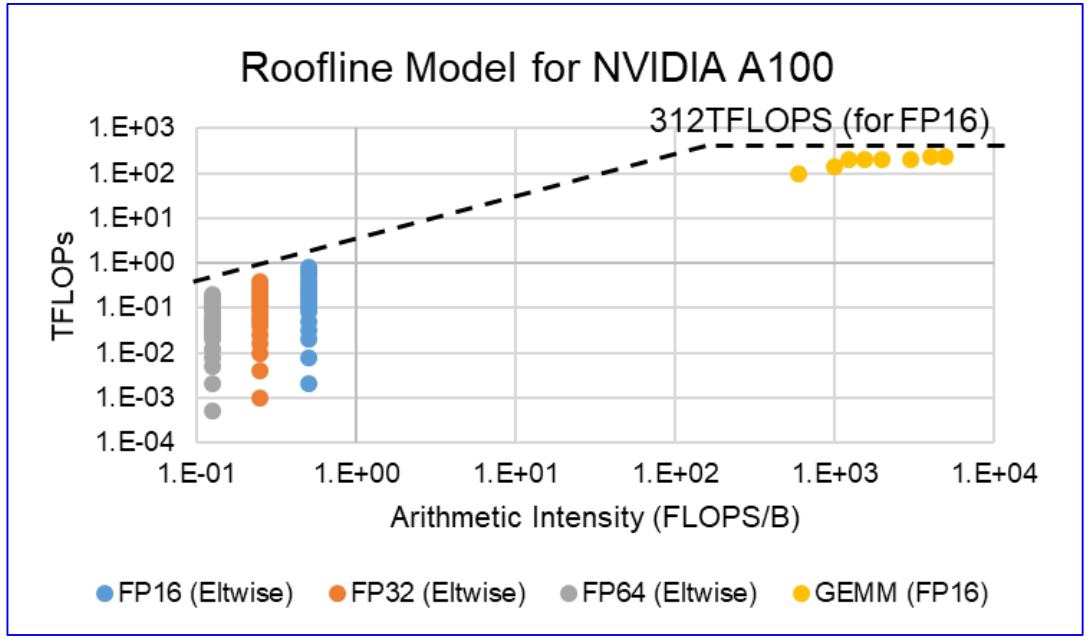
Intelligent Memory and Types

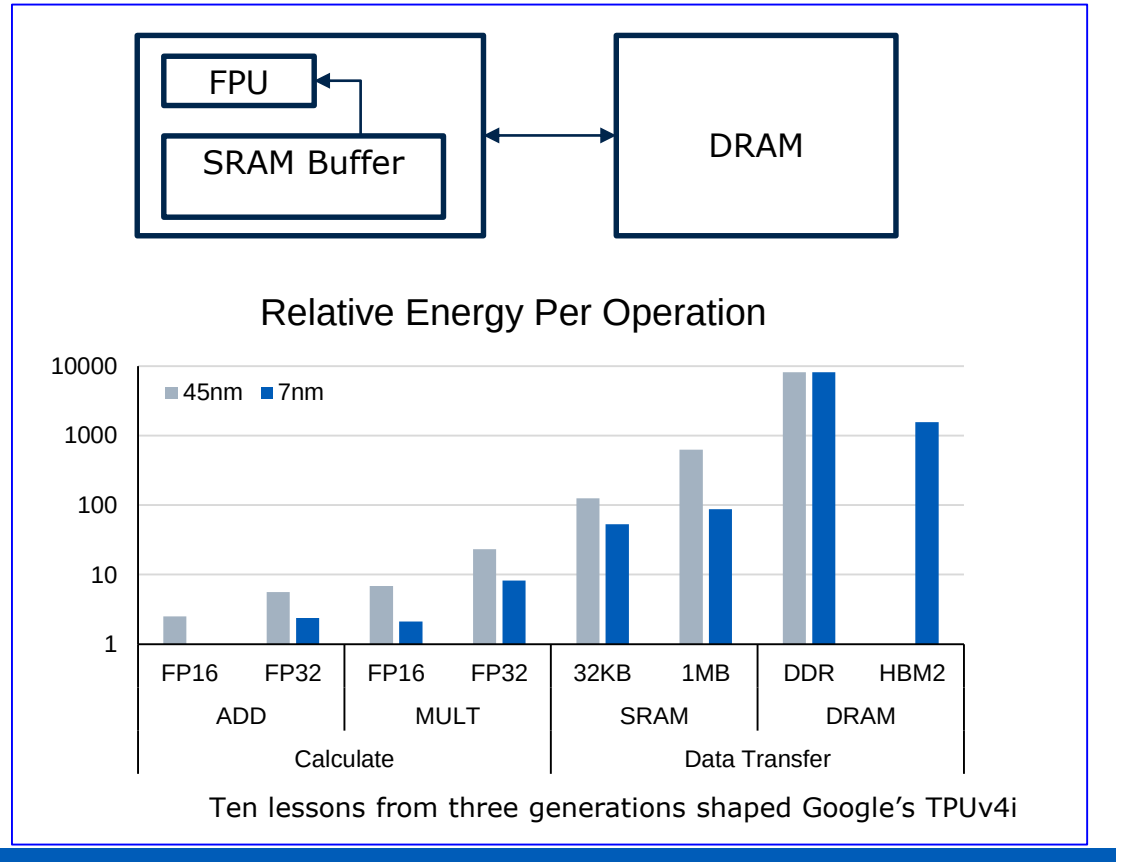
- Three distinct categories in this talk
  - CIM: use memory array as a processing unit
  - PIM: use embedded logic near memory array as a processing unit
  - PNM: use an additional chip for processing inside a memory package or a module



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

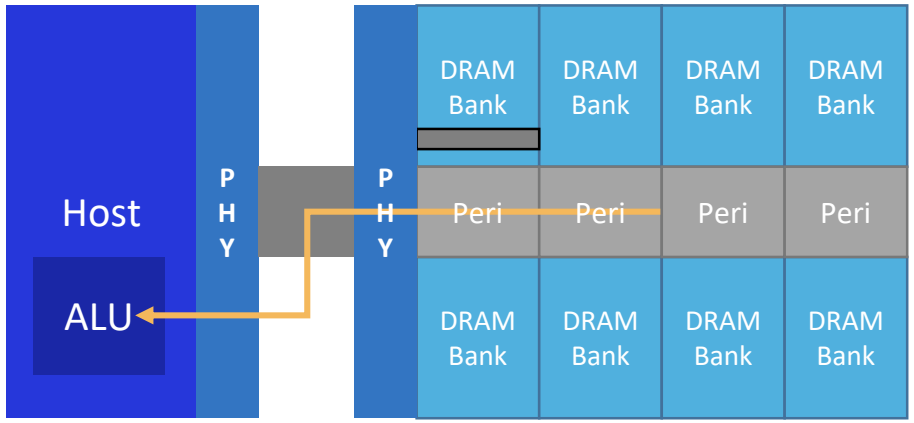




Processing-in-Memory (PIM)

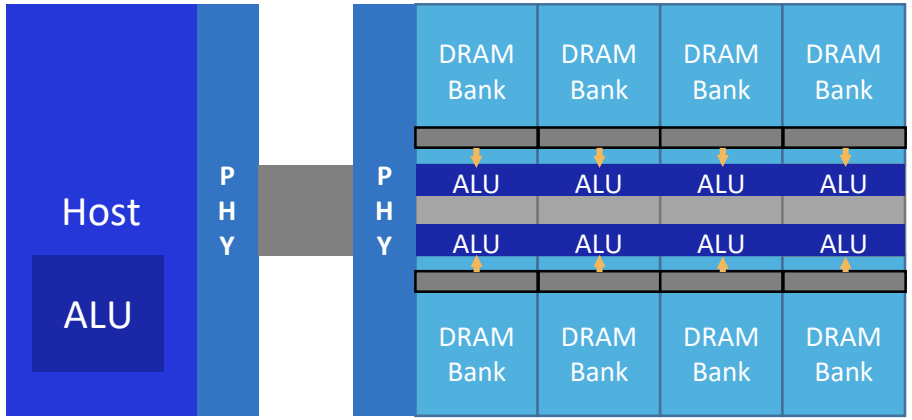
- Embedded arithmetic logic boosts bandwidth and energy efficiency
  - Key idea is utilizing bank-level parallelism
    - Host can access 1-bank (or BG) at a time ↔ PIM 8-BGs all-banks in parallel
  - Remove power consumption for data-path (IOSA-Peri-PHY)

Normal DRAM



Data PHY access for read operation (single bank)

PIM DRAM



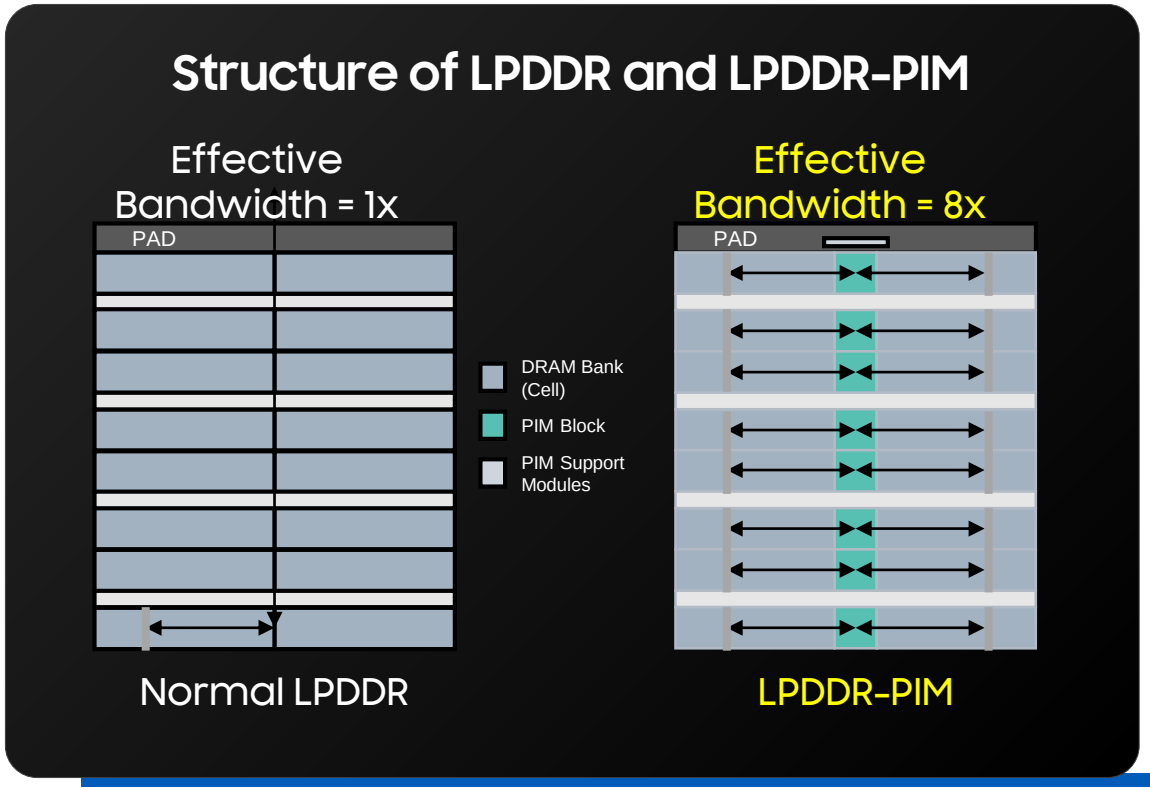
No data PHY access for PIM operation but all bank read

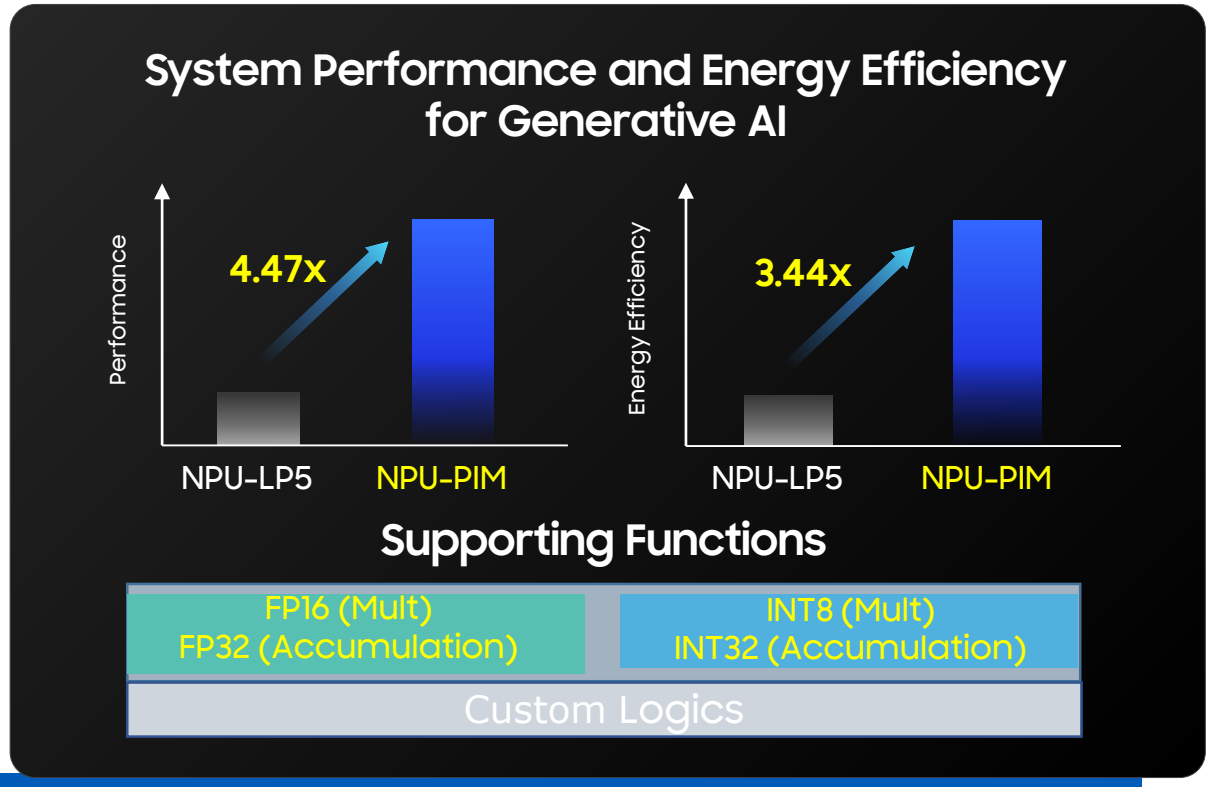
Samsung PIM Development Philosophy

- 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

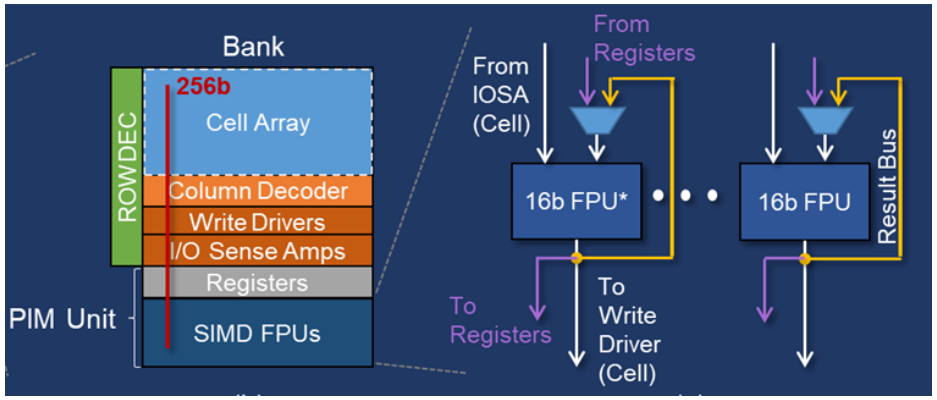
- LPDDR-PIM improves performance and energy efficiency of the system with in-DRAM processing
  - Performance: Utilizes up to 8x higher in-DRAM bandwidth by bank parallel operation
  - Energy Efficiency: Reduces data movement energy by utilizing in-DRAM processing unit

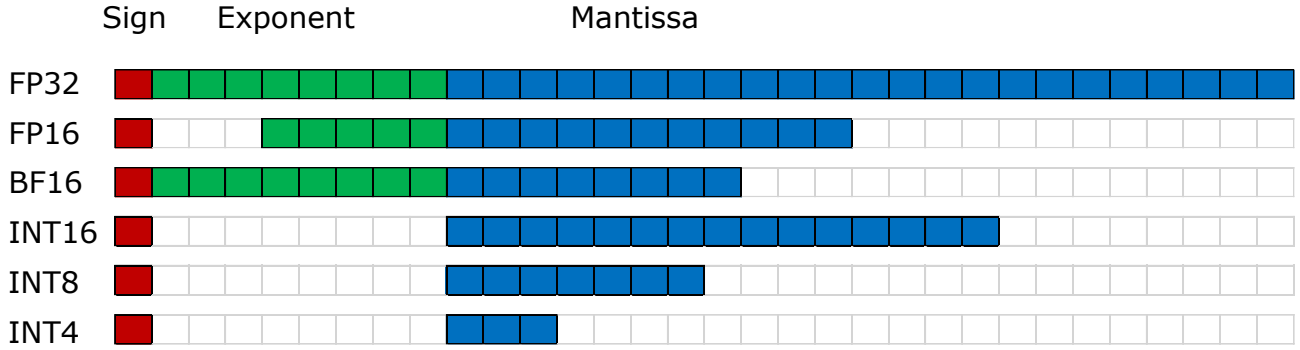




Challenges on On-Device AI w/ LPDDR-PIM

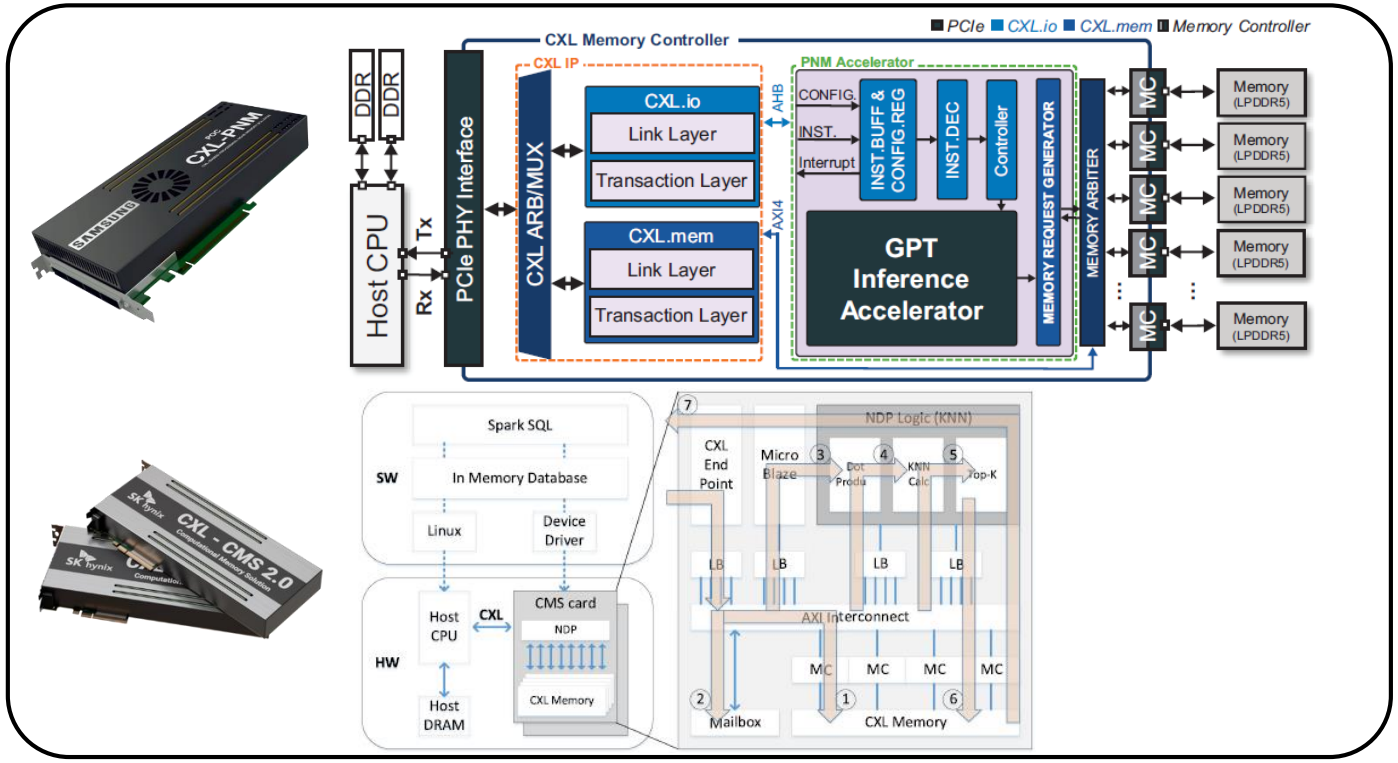
- Supporting various data format (e.g., INT4, INT8)
  - Low-precision data format is used because of model size issue (capacity issue).
  - PIM architecture should support ALUs covering multiple data formats.
- Thermal & Power issue
  - For mobile SoC, thermal stress should be effectively mitigated under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



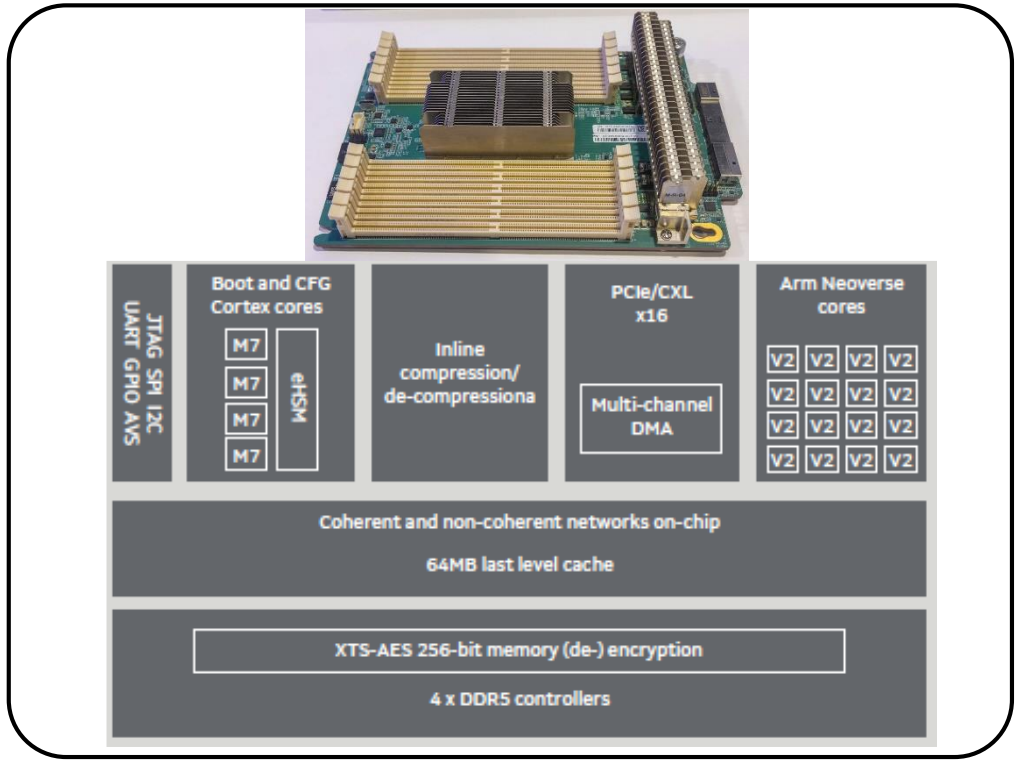


CXL PNM Approaches

- Initially developed as PoC using FPGAs, now being advanced via development on ASIC
  - FPGA-based application specific function acceleration: Samsung CXL-PNM[1] (‘23), SK Hynix CMS[2] (‘23)
  - ASIC-based general purpose application acceleration: Marvell Structera-A3) (‘25)



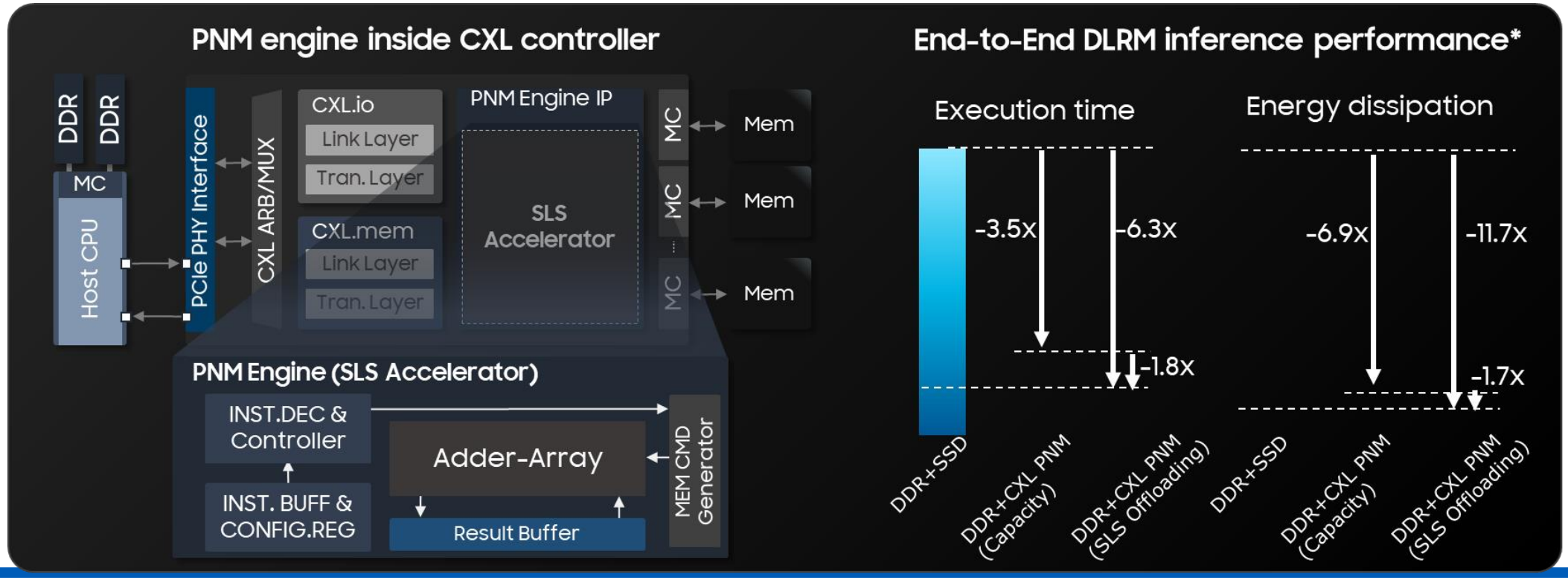
[FPGA-based CXL PNM: Samsung CXL-PNM (top), SK Hynix CMS (bottom)]



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

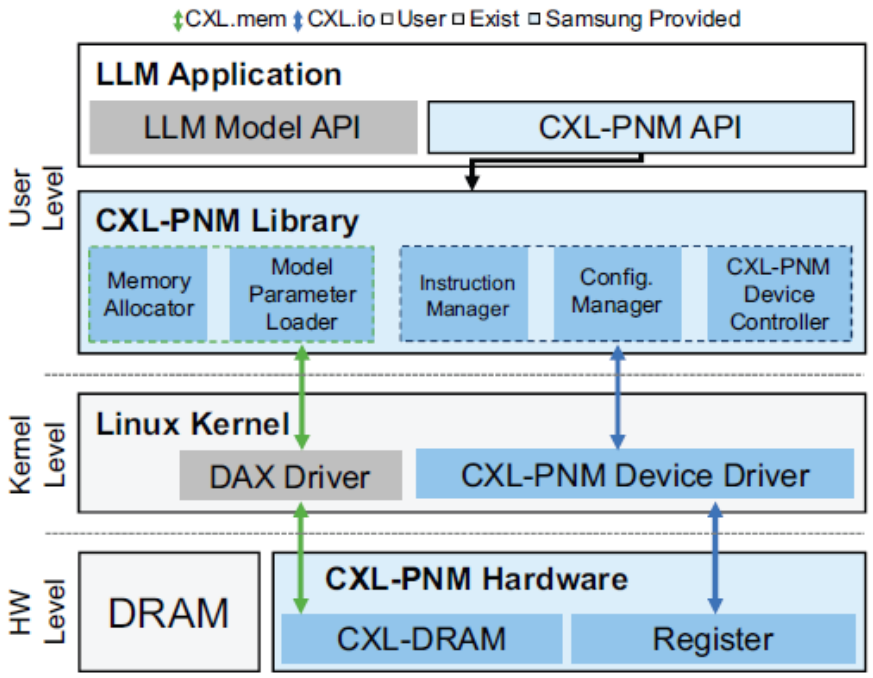
CXL PNM Usecase: DLRM Inference

- Offloading random memory access with little data reuse Sparse Length Sum (SLS) operation
  - By eliminating storage accesses, the execution time dramatically reduced by x3.5 (DDR+CXL PNM (Capacity))
  - With the computation capability, CXL-PNM achieves addition performance improvement by x1.8 (DDR+CXL PNM (SLS Offloading))



Programming Model for CXL PNM

- User application replaces standard Python-based AI/ML library into CXL PNM's at linking time and invokes CXL PNM APIs at runtime
- CXL PNM Python library provides memory (de)allocate and LLM function APIs
  - Supported LLM functions: LayerNorm, Conv1D, MaskedMM, Softmax, GELU, etc.
  - Software guarantees/controls data coherency and consistency between host CPU and PNM (e.g., CPU cache flush after PNM write)

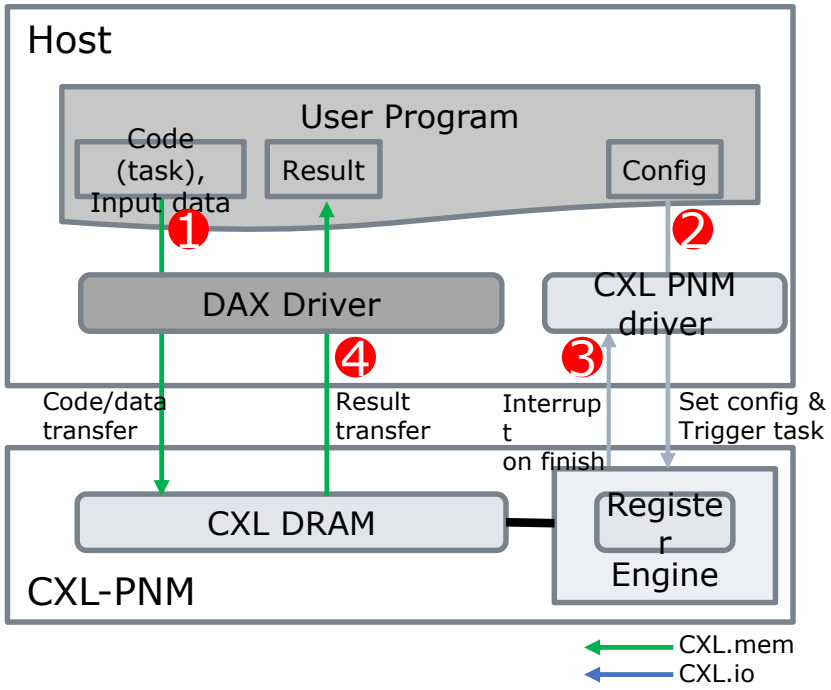


[CXL PNM software stack]

```
// CXL.io: PNM Code transfer
data = read_file("vpu.bin")
PNM_write_mem(data, offset)
...
// CXL.mem: input data transfer
data = read_file("layNorm_in")
PNM_write_mem(data, offset)
...
// CXL.io: set configurations
reg_update(NUM_IP_TOKENS)
reg_update(EMB_SIZE)
...
// CXL.io: Execute PNM
offload_start()

// Get result
PNM_read_mem(res, offset)
```

[Example Embedding Python code]



[CXL PNM Runtime Execution Flow]

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.
  - Custom HBM
  - MRDIMM for high-BW and high-capacity
  - LP-based CAMM solution for lower-power and higher BW
  - CXL memory module for capacity/BW expansion, pooling & sharing

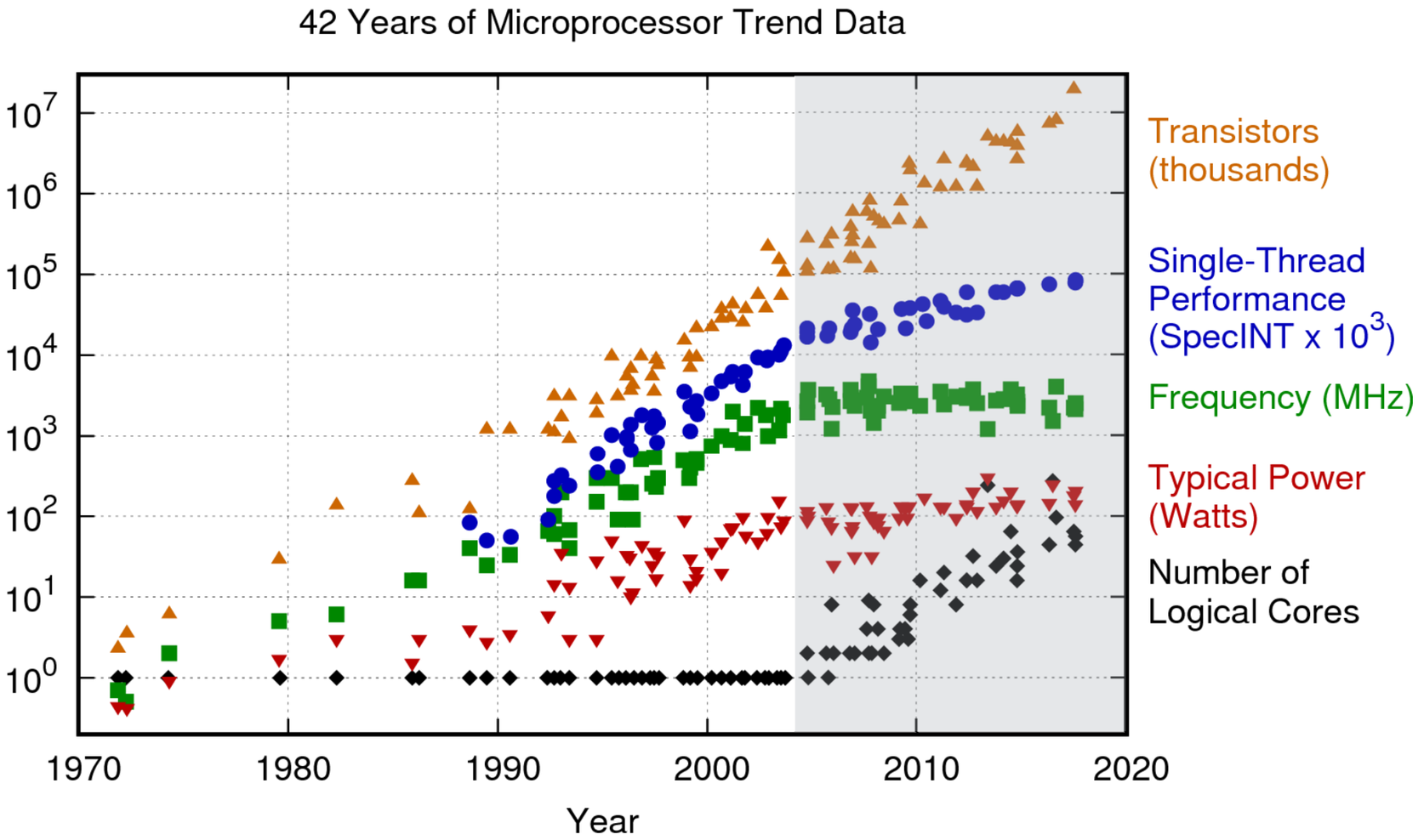
- Processing capability in/near Memory enables higher bandwidth and energy efficiency.
  - CIM, PIM and PNM are meaningful and heavily studied in school and industry.
  - Still lots of challenges to be solved for commercializing.
  - Need strong collaboration between system, processor, memory and software.

Future Challenges

Steven Woo
Fellow and Distinguished Inventor
Rambus Inc.

Rambus

Computation Increasingly Power and Bandwidth-Limited

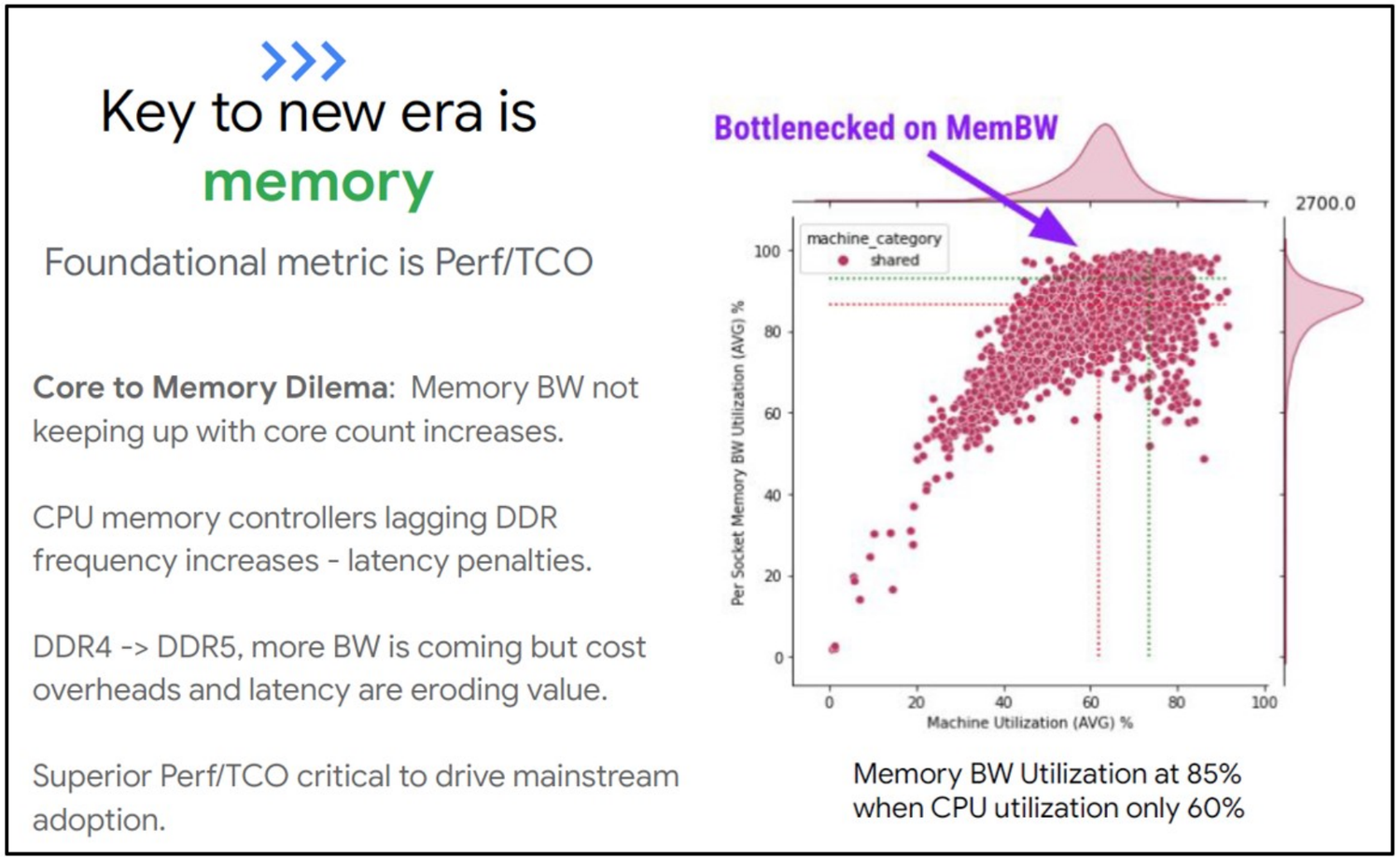


Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Source: Karl Rupp, "42 Years of Microprocessor Trend Data"
https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/

- Dennard scaling ended, Moore's Law slowing
- Domain-specific silicon improving performance and energy-efficiency
- Parallelism increasing from more cores, greater number of compute pipelines per core
- Memory system bandwidth under stress

Lack of Memory Bandwidth Can Bottleneck Systems



- In some cases, cores bottlenecked by memory bandwidth
- Can be exacerbated by core count increases
- Technologies like MRDIMM can improve memory system bandwidth in similar footprint

Source: "The Renaissance in Datacenter Design: Delivering Modern and Scalable Solutions," Tom Garvens, MemCon 2023

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

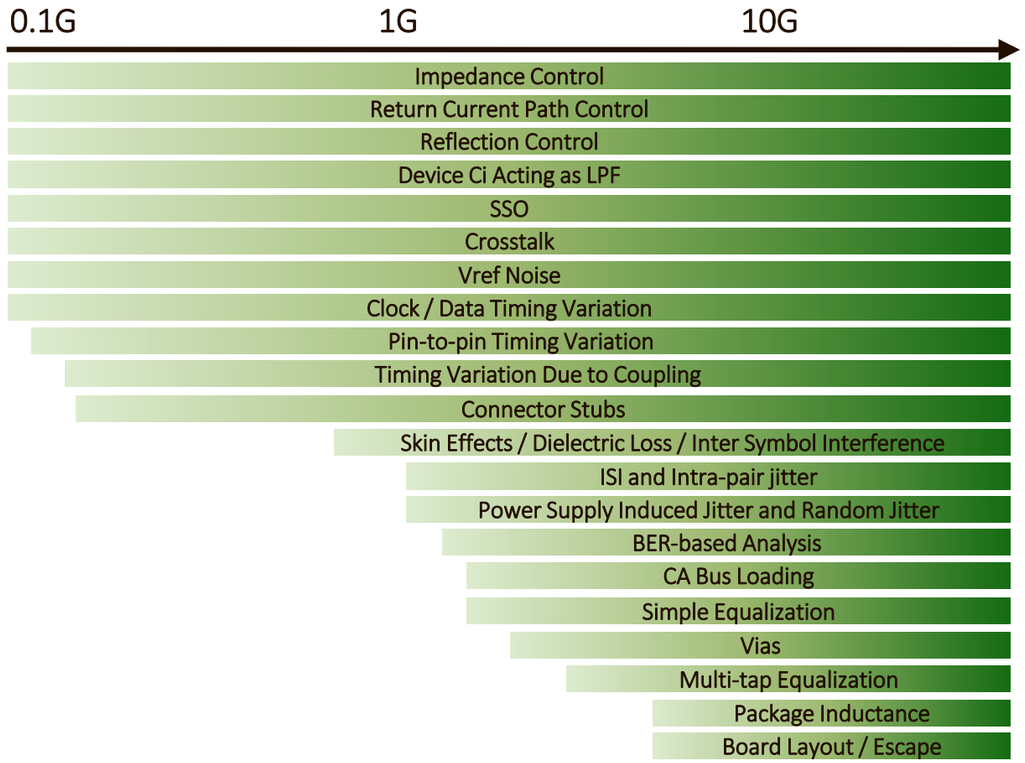
* In production as embedded memory

For now, the future of DRAM is still DRAM

Faster DRAM Data Rates: Signal Integrity More Challenging

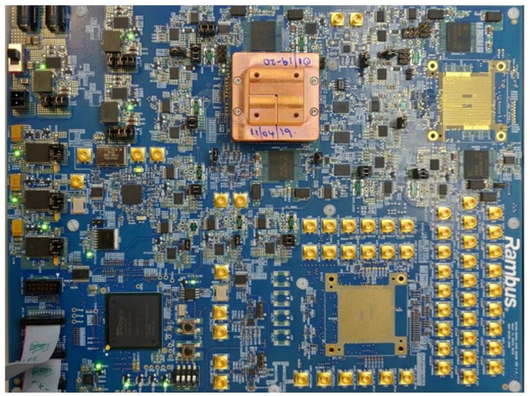
Growing number of effects must be accounted for as speeds increase

Partial List of Memory System Effects to Consider

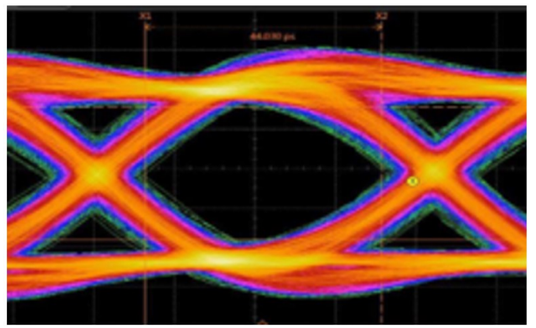


Ensuring good signal integrity requires accurate full system modeling from transmitter to receiver

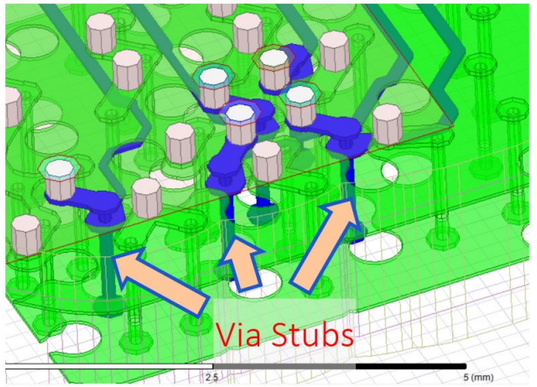
Rambus GDDR6 PHY Evaluation Board



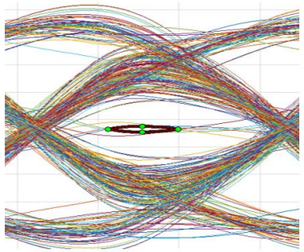
Rambus GDDR6 PHY 16Gbps Write Eye



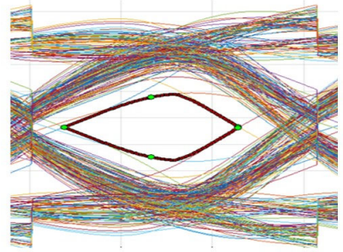
Example: Mitigating the effects of via stubs



Baseline Eye



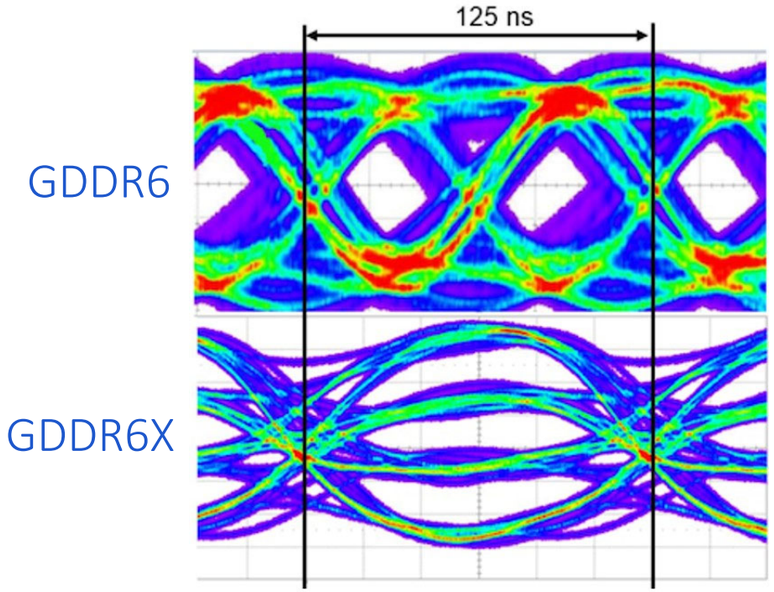
Via Back-drilling + 1 Tap DFE



Source: "Design with Confidence Using 16Gb/s GDDR6 Memory," Micron Tutorial at DesignCon 2019

Signal Integrity and Reliability Becoming More Challenging

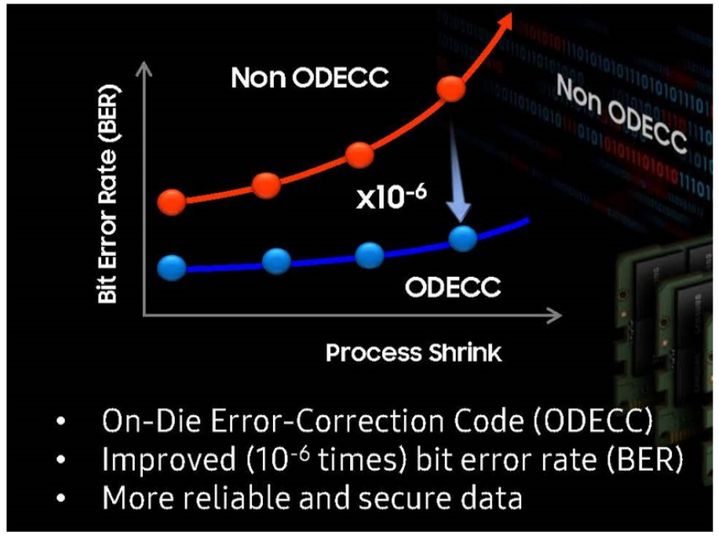
Multi-PAM Signaling



Source: "Micron Spills on GDDR6X: PAM4 Signaling For Higher Rates, Coming to NVIDIA's RTX 3090," https://www.anandtech.com/show/15978/micron-spills-on-gddr6x-pam4-signaling-for-higher-rates-coming-to-nvidias-rtx-3090

- Lower frequency, multiple bits/symbol
- Increases bandwidth, eases signal integrity challenges

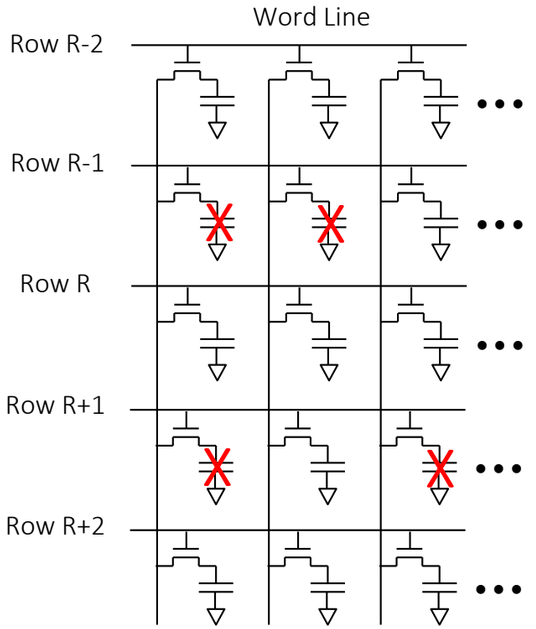
On-Die ECC



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

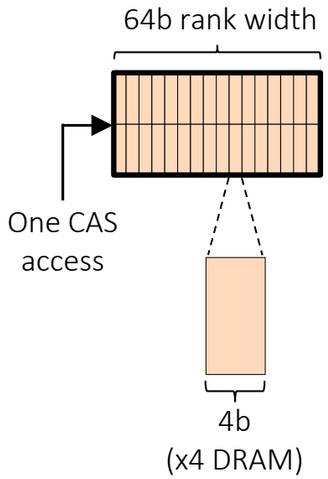


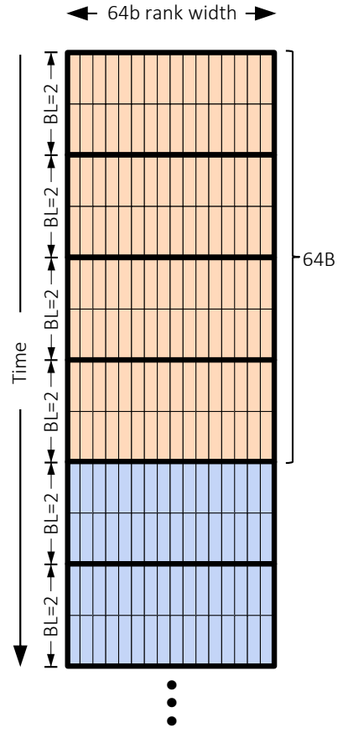
- Bit cells can flip due to activity in neighboring rows (disturb errors)
- Becoming more problematic

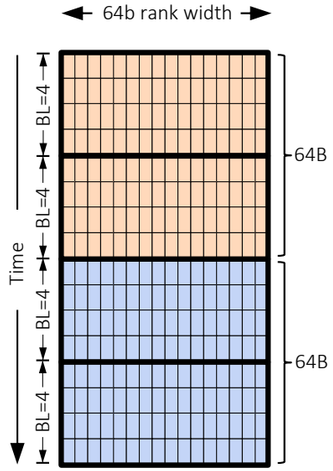
Keeping the DRAM Low Cost: DRAM Core Speed

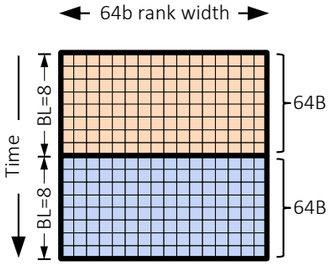
	DDR		DDR2		DDR3		DDR4		DDR5	
Data Rate (Gbps)	200	400	400	800	800	1600	1600	3200	3200	6400
tCCD / tCCD_L (ns)	15	10	10	5	10	5	6.25	5	5	5
Core Frequency (MHz)	66.7	100	100	200	100	200	160	200	200	200
Prefetch (N)	2	2	4	4	8	8	8	8	16	16
DIMM Rank width (bits)	64/68/72	64/68/72	64/68/72	64/68/72	64/68/72	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

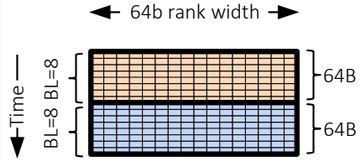
64B cache line transfers



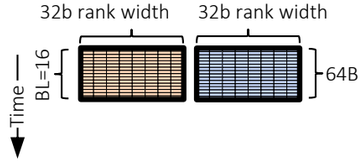








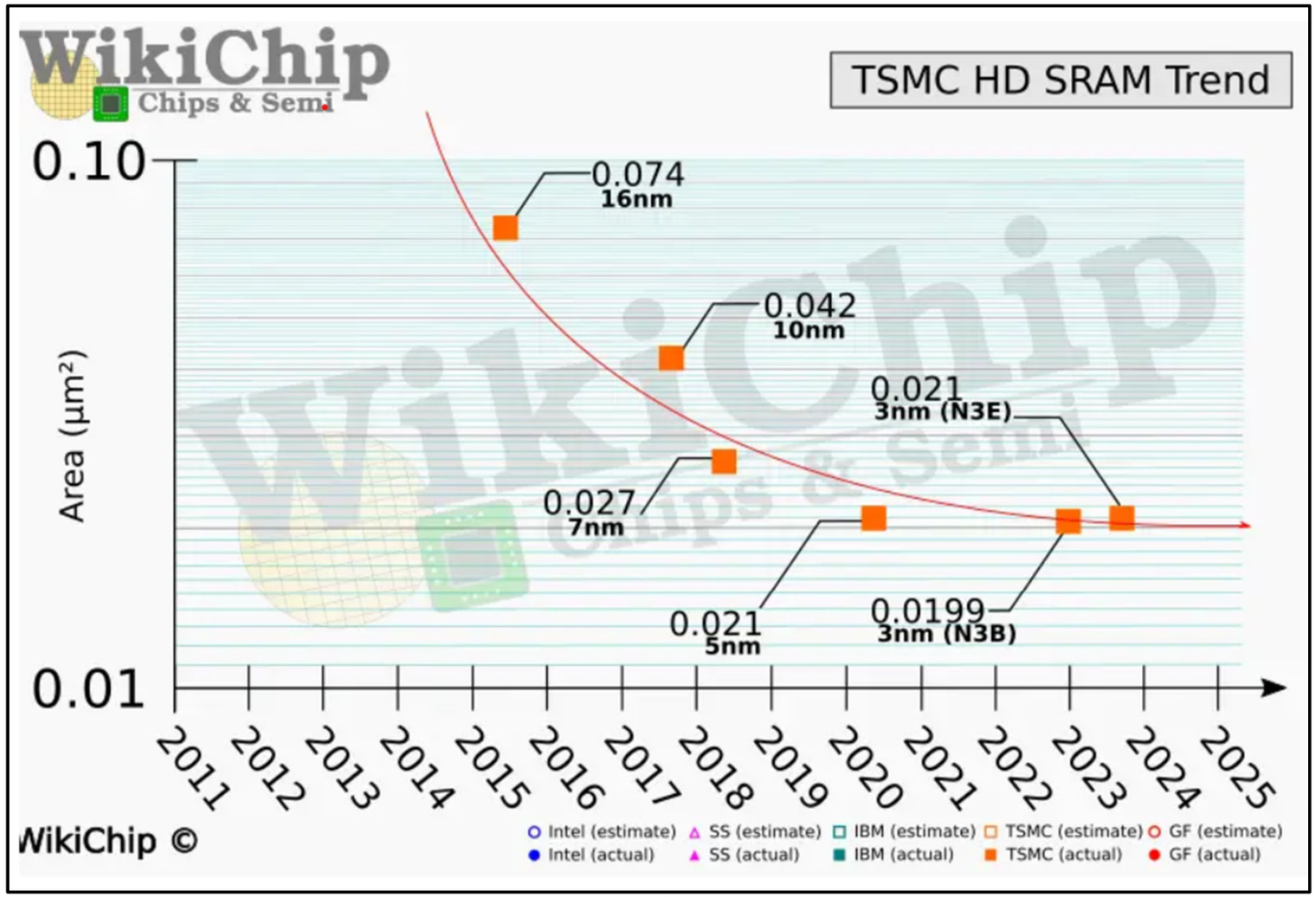
- BL16 would produce 128B access, too large for 64B cache line => potentially wasted bandwidth
- Introduced bank groups to allow core to stay at ≤200MHz



- 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

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
- 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
- The lines are starting blur in some cases
  - How to make DDR lower power or LPDDR channels support higher RAS capability?
- Many scaling tradeoffs – there's no free lunch
  - Continuous re-evaluation as assumptions, constraints, and application needs change

Summary and Closing Remarks (2)

- 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 continues to innovate and develop new solutions


Thank you