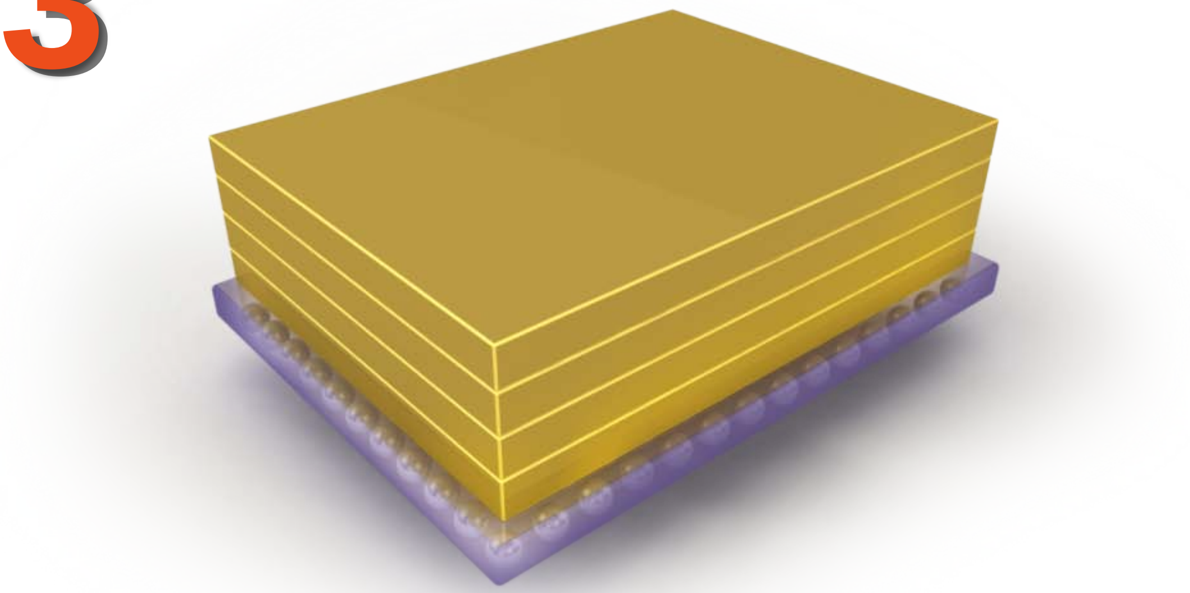
HOT CHIPS 23



Hybrid Memory Cube (HMC)

J. Thomas Pawlowski, Fellow

Chief Technologist, Architecture Development Group, Micron
jpawlowski@micron.com

Outline

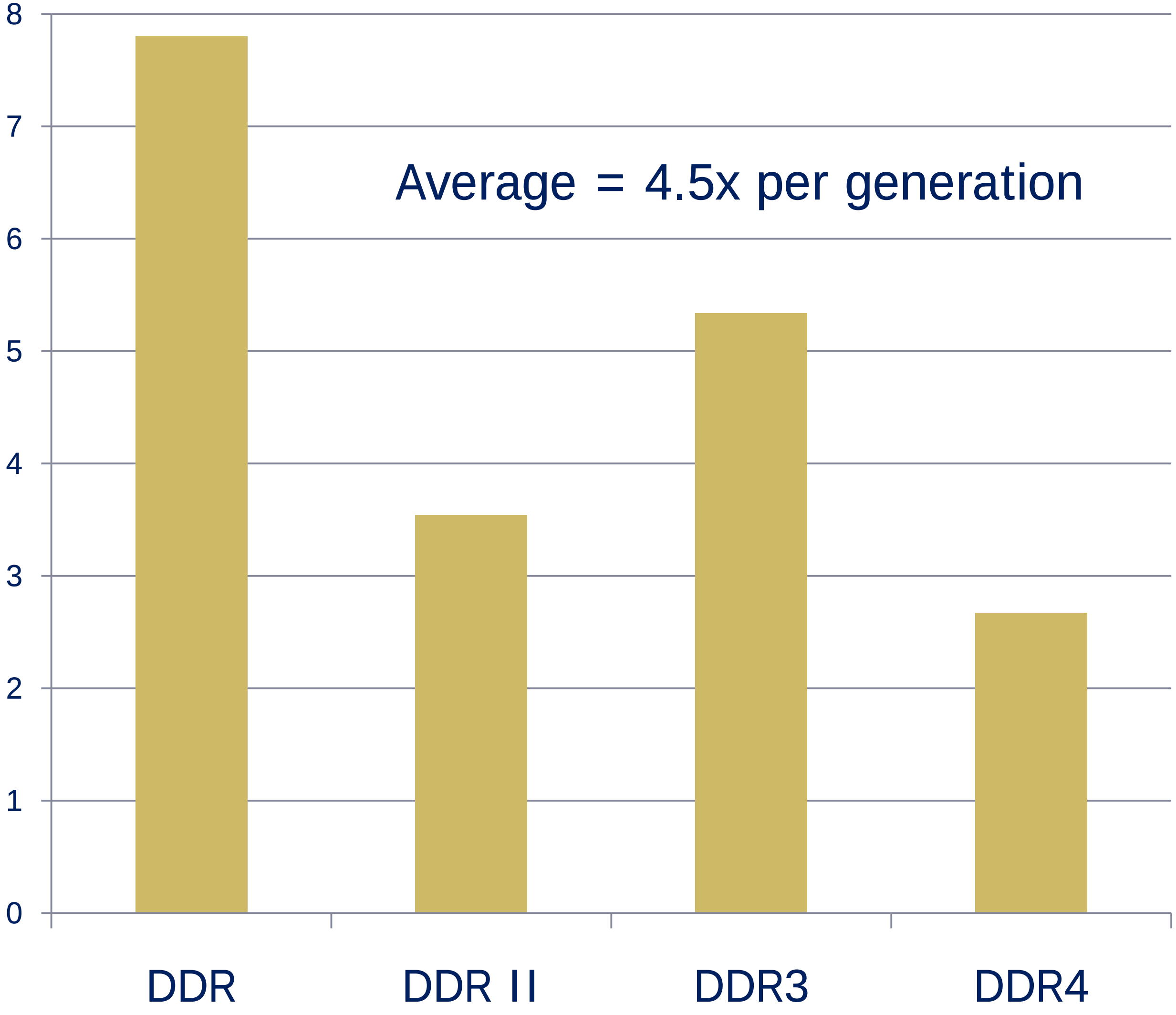
- Problems
- Goals
- HMC introduction
- New relationship between CPU and memory
- Internal architecture
- Performance
- Summary

The Problems

- Observed Problems:
  - Latency (classic “memory wall”)
  - Bandwidth related issues
  - Power / energy
  - Many-core, Multi-threaded CPUs generate higher random request rates
  - Memory capacity per unit footprint

- Future Problems:
  - Scalability of bandwidth, densities, request rates and lower latencies
- Essential Underlying Issue:
  Direct control of memory must give way to memory abstraction
  - Mitigate negative characteristics of next generation DRAM processes
  - Introduction of future DRAM replacement technologies

Historic Energy x Bandwidth Improvement Per DRAM Generation

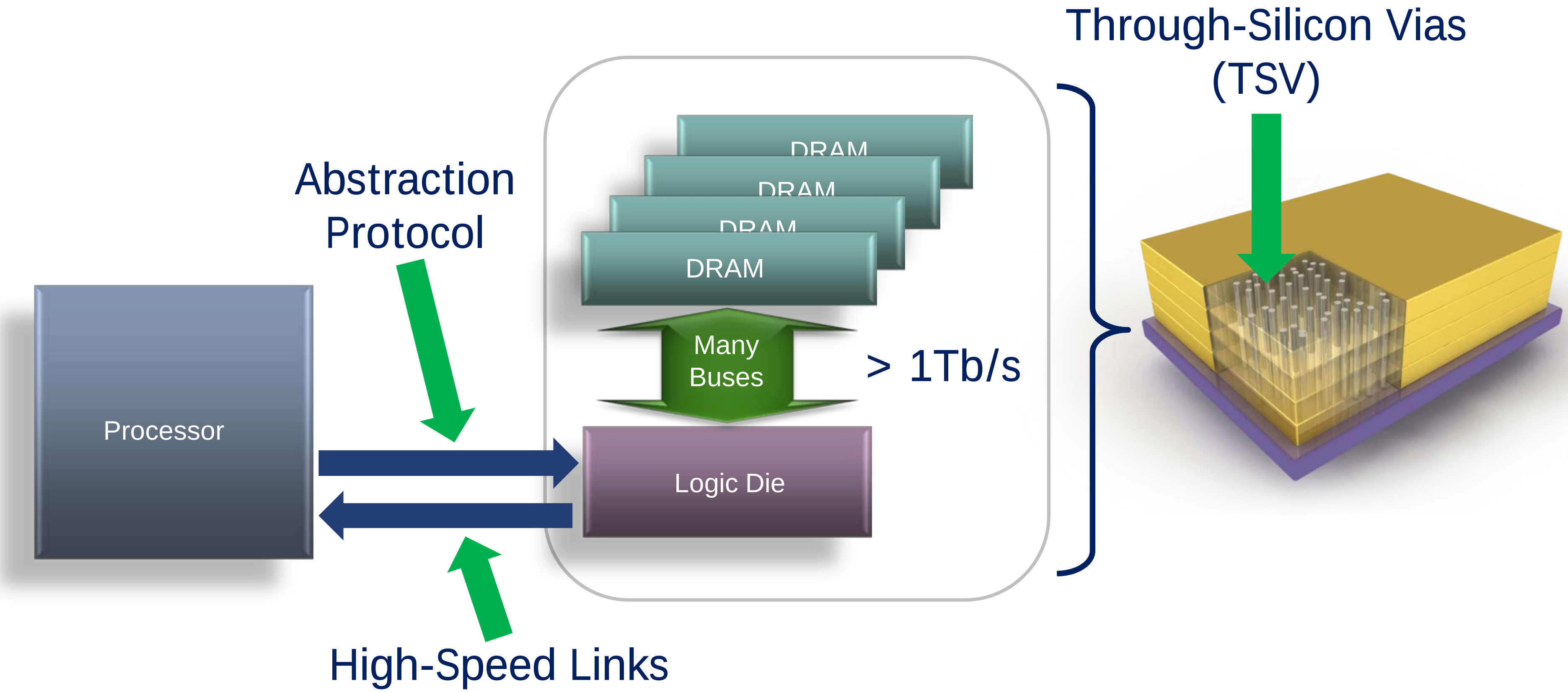


Average = 4.5x per generation

Hybrid Memory Cube Goals

1. Higher bandwidth
2. Higher signaling rate
3. Lower energy per useful unit of work done
4. Lower system latency
5. Increased request rate, for many-core: **CONCURRENCY !**
6. Higher memory packing density
7. Abstracted interface
    1. lighten CPU/DRAM interaction
    2. enable new DRAM management concepts
    3. manage future process scaling and future technology introductions
8. Scalability for higher future bandwidths and density footprint
9. Reduce customer and Micron time to market

Hybrid Memory Cube (HMC)



Through-Silicon Vias (TSV)

Abstraction Protocol

DRAM

DRAM

DRAM

DRAM

Many Buses

> 1Tb/s

Processor

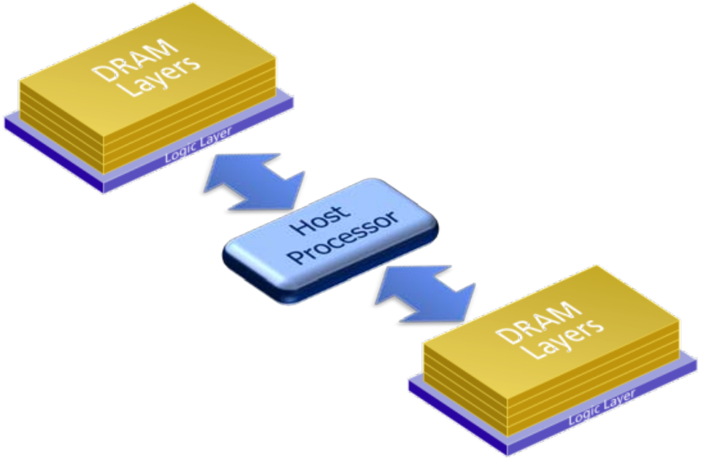
Logic Die

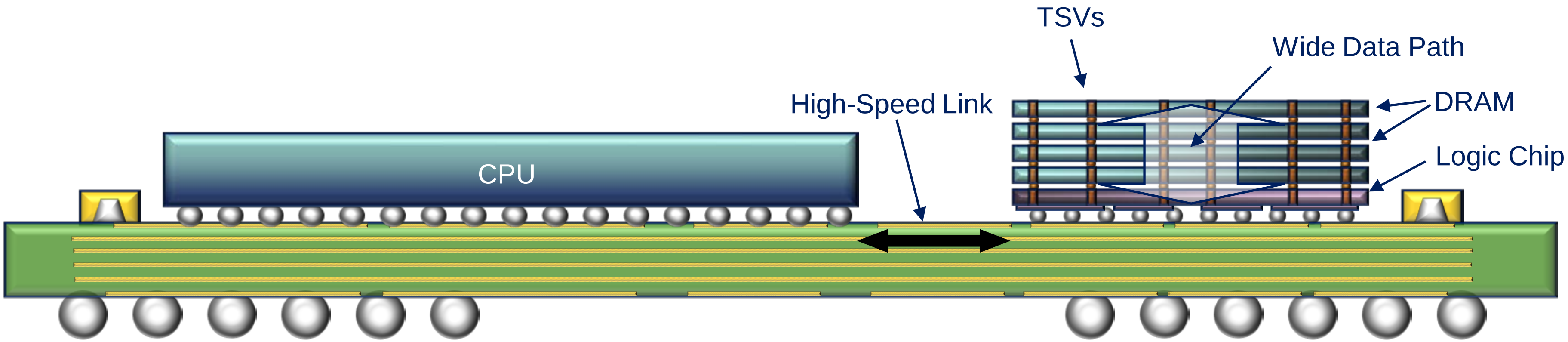
High-Speed Links

Notes: Tb/s = Terabits / second
HMC height is exaggerated

HMC Near Memory – MCM Configuration

- All links are between host CPU and HMC logic layer
- Maximum bandwidth per GB capacity



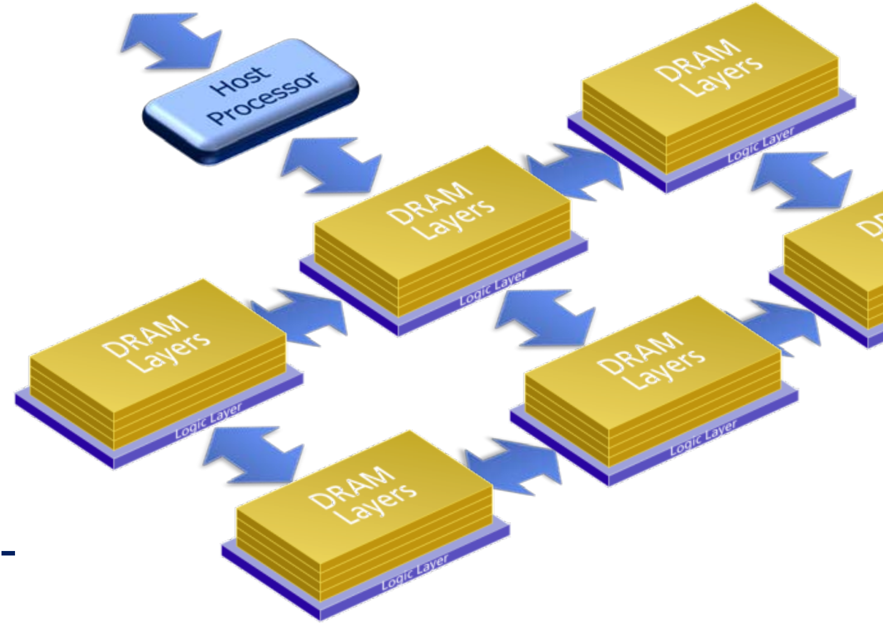


Notes: MCM = multi-chip module

Illustrative purposes only; height is exaggerated

HMC “Far” Memory

- Far memory
  - Some HMC links connect to host, some to other cubes
  - Serial links form networks of cubes
    - the memory = the network
  - Scalable to meet system requirements
  - Can be in module form or soldered-down
  - Can form a variety of topologies e.g., tree, ring, double-ring, mesh
- Future interfaces
  - Higher speed electrical (SERDES)
  - Optical
  - Whatever the most appropriate interface for the job

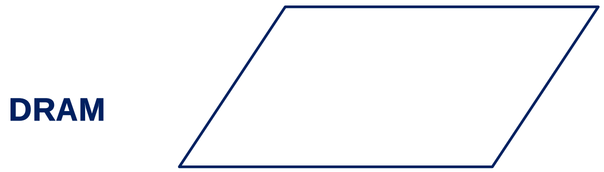


Processor – Memory Interaction

- Yesterday: multi-core CPU direct connection to DRAM-specific buses
  - Complex scheduler, deep queues, high reordering especially writes
  - Many DRAM timing parameters standardized across vendors
    Worst case "everything"
  - Result is conservative, evolutionary, uncreative, slow performance growth
- Now: direct connect to HMC logic chip via abstracted high-speed interface
  - No need for complex scheduler, just thin arbiter, shallow queues
  - Only the high-speed interface, protocol, form-factor might be standardized
    NO TIMING constraints, overrun is prevented
  - Great innovations can occur "under the hood"
  - Maximize performance growth
  - Logic layer flexibility allows HMC cubes to be designed for multiple platforms and applications without changing the high-volume DRAM
- HMC takes requests, delivers results in most advantageous order

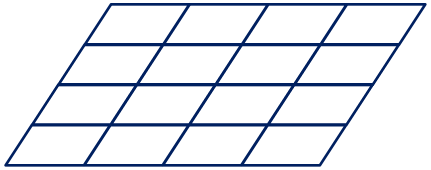
HMC Architecture

Start with a clean slate



HMC Architecture

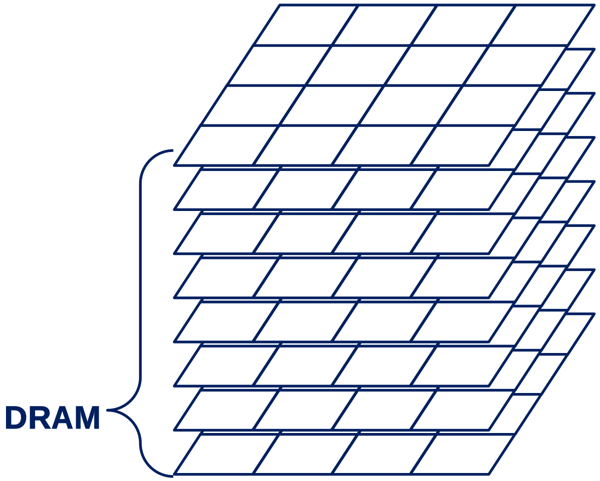
Re-partition the DRAM and strip away the common logic



DRAM

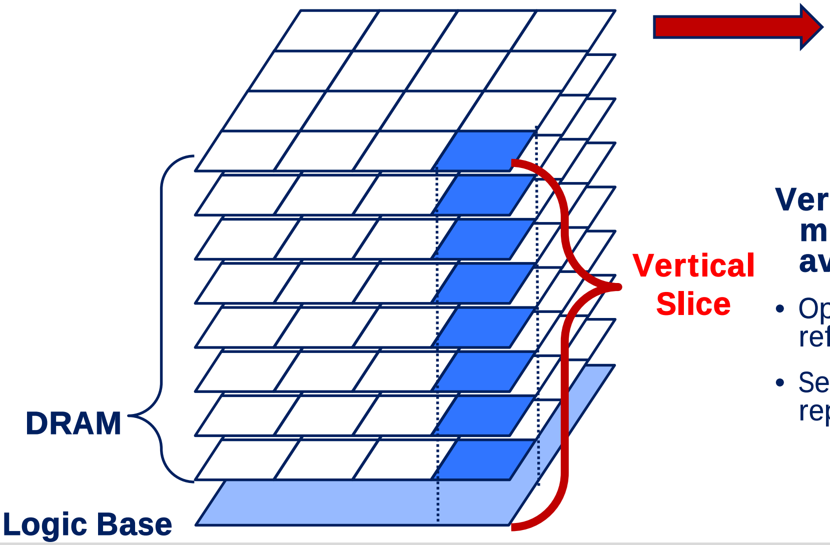
HMC Architecture

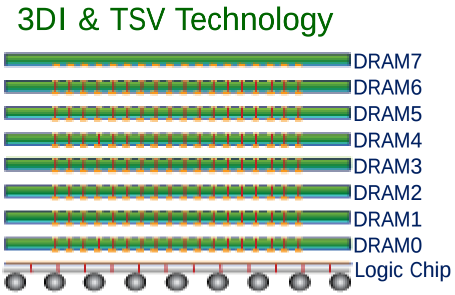
Stack multiple DRAMs



HMC Architecture

Re-insert common logic on to the Logic Base die

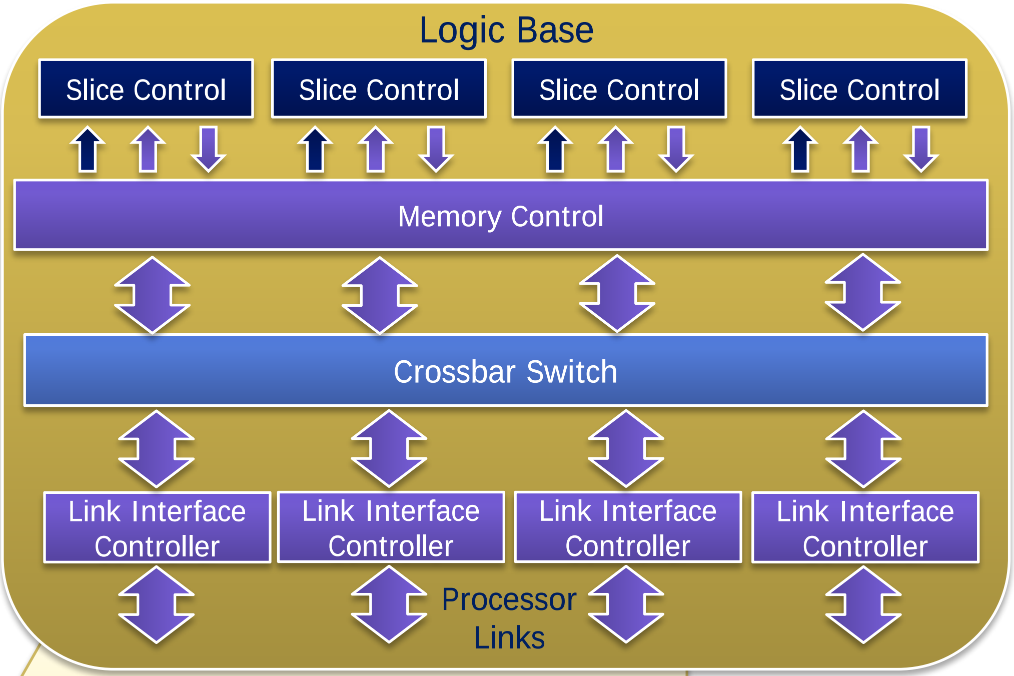




Vertical Slices are managed to maximize overall device availability

- Optimized management of energy and refresh
- Self test, error detection, correction, and repair in the logic base layer

HMC Architecture

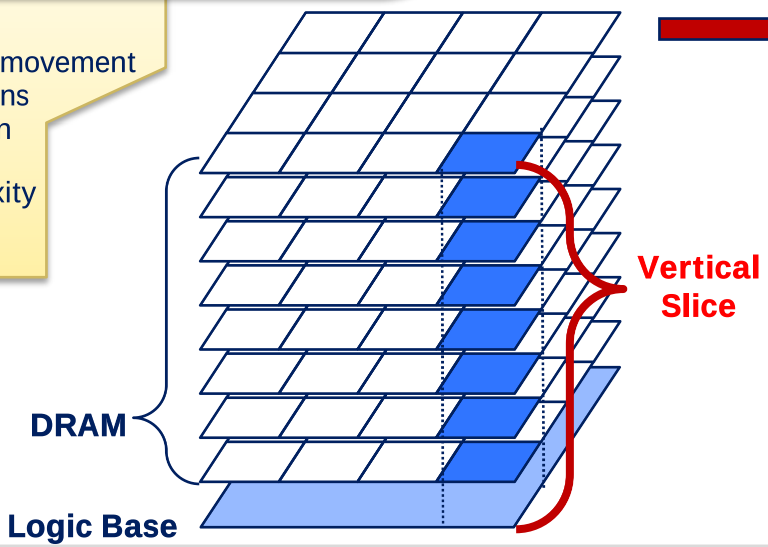


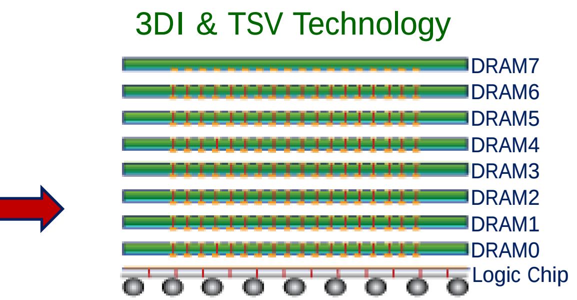
Add sophisticated switching and optimized memory control...

And now we have a whole new set of capabilities

Logic Base

- Wide, high-speed local bus for data movement
- Advanced memory controller functions
- DRAM control at memory rather than distant host controller
- Reduced memory controller complexity and increased efficiency





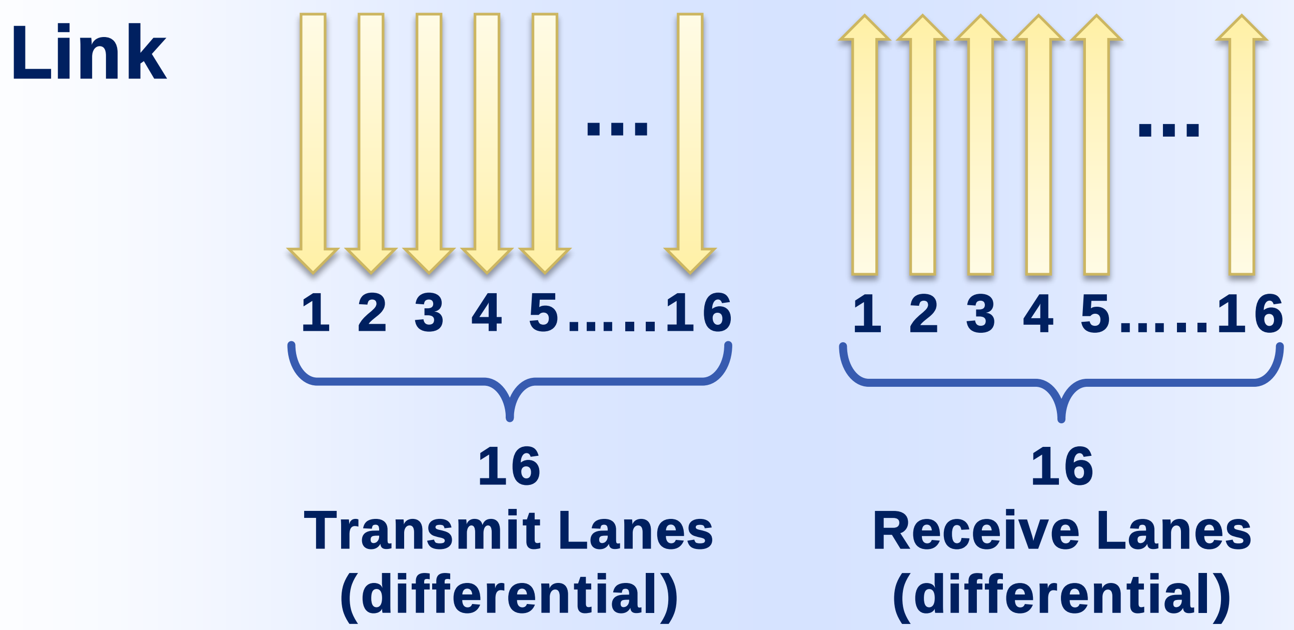
Vertical Slices are managed to maximize overall device availability

- Optimized management of energy and refresh
- Self test, error detection, correction, and repair in the logic base layer

Vastly More Responders

- Conventional DRAM DIMM example: 8 devices, 8 banks/device
  - Banks of all devices run in lock-step
  - One of 8 potential responders will answer a typical request
- Not only does HMC give excellent concurrency
  - HMC gen 1 example: 4 DRAMs * 16 slices * 2 banks = 128
  - One of 128 potential responders will answer a typical request
  - Double that to 256 if 8 DRAMs are in the stack
  - Vast improvement in response to random request stream
- Significant impact on system latency
  - DRAM tRC is lower by design
  - Lower queue delays and higher bank availability further shortens latency
  - Serial links slightly increase latency
  - Net effect is a substantial system latency reduction

Available Total Bandwidth



Link

1 2 3 4 5…..16

16
Transmit Lanes
(differential)

1 2 3 4 5…..16

16
Receive Lanes
(differential)

10 Gbps/Lane
32 Lanes = 4 bytes → **40 GBps per Link**

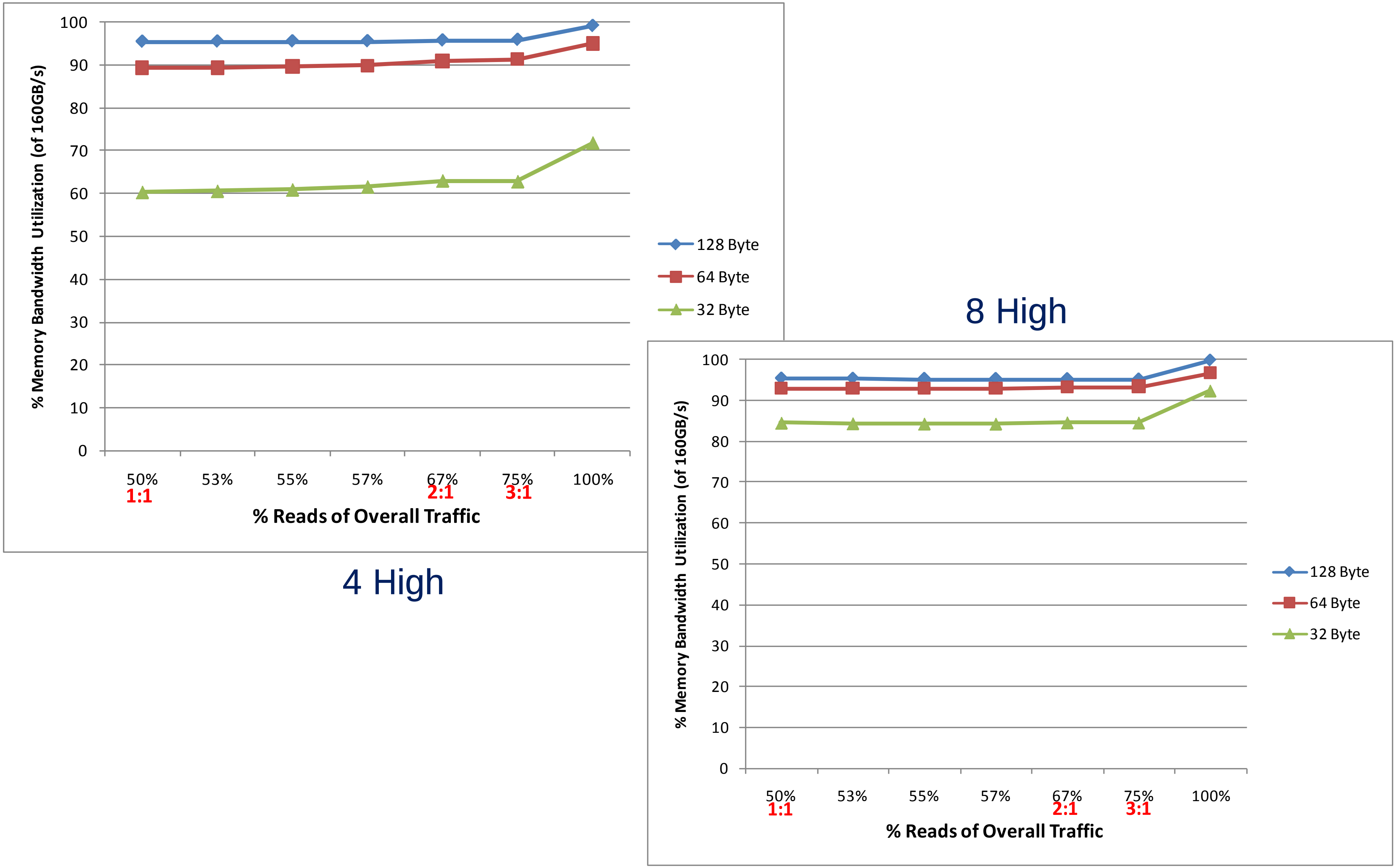
4 Links per Cube → **160 GBps per Cube**

8 Links per Cube → **320 GBps per Cube**

RAS Features

- Reliability, Availability, and Serviceability (RAS) features are built into HMC
- Systemic RAS features
  - Array repair
  - DRAM/Logic IO interface repair
- Internal ECC is utilized for detection and short term correction
  - Upon detection, repair is scheduled so that ECC is not a perpetual crutch for an identified issue and is free to cover future errors
- Link IO has means to detect communication errors
  - Worst case hard failure shuts down only one of many links

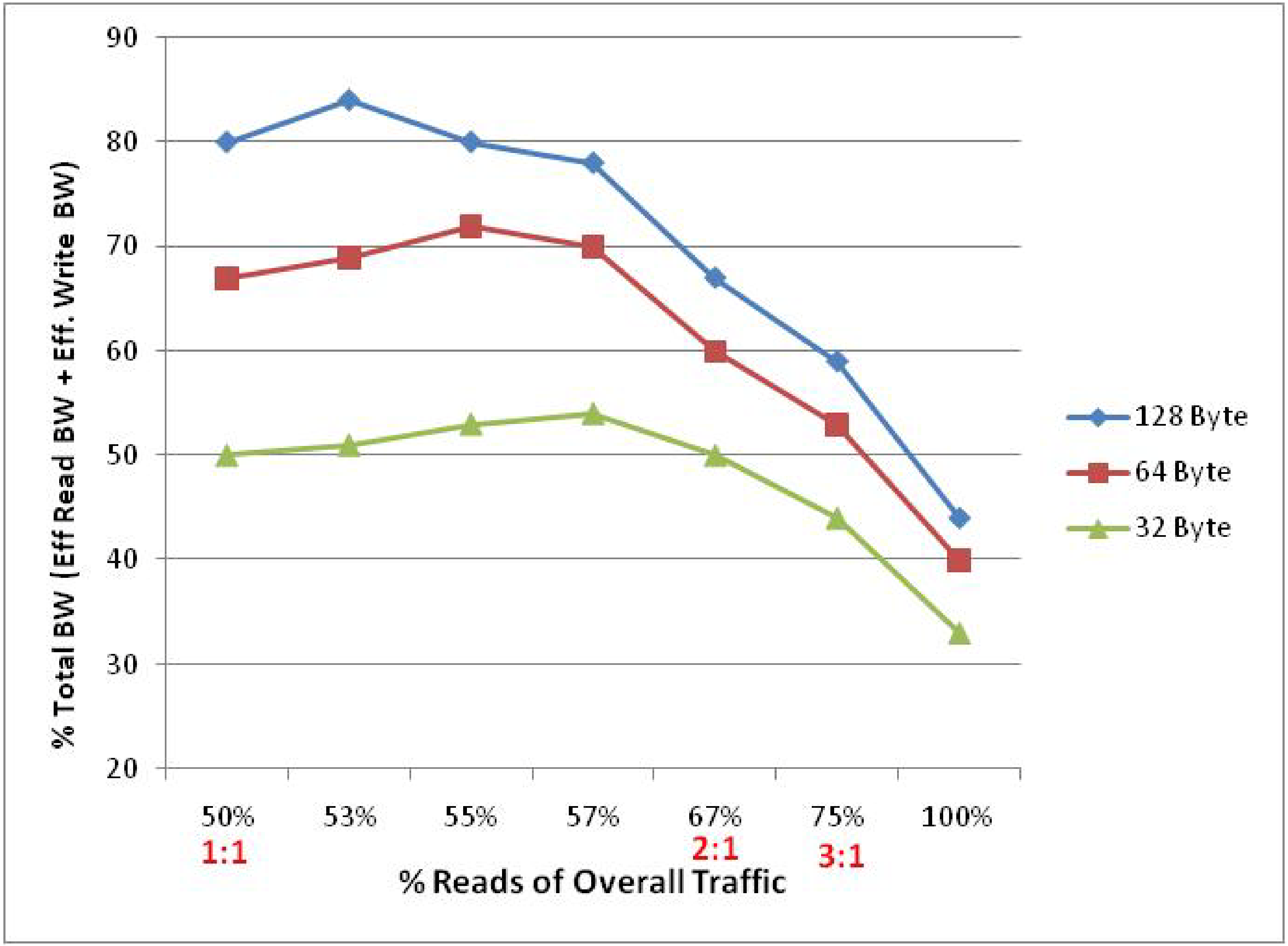
Internal DRAM Efficiency 4-8 High Stacks



4 High

8 High

Effective Read/Write External Link Efficiency



Other Unique System Capabilities

- Request rate capability
  - 3.2G operations per second (32 byte data transactions)
  - Limited by external data links
    - 2.3Gops at 128GB/s, 2.9Gops at 160GB/s
- Random request capability
  - Historically this gets worse as bandwidth is increased
  - DDR3 SDRAM is ~29% at BL8, DDR4 is worse, GDDR5 is worse yet
  - HMC achieves 75% of peak bandwidth (64B data transactions)
- Network of HMCs
  - E.g., double-ring or mesh of interconnected HMCs
  - Average latency grows but is a constant regardless of the individual HMC's depth in the mesh

HMC$_{Gen1}$: Technology Comparison

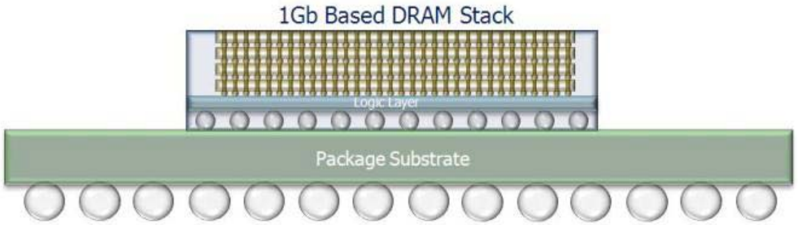
Generation 1 (4 + 1 memory configuration)

Technology	VDD	IDD	BW GB/s	Power (W)	mW/GB/s	pj/bit	real pJ/bit
SDRAM PC133 1GB Module	3.3	1.50	1.06	4.96	4664.97	583.12	762
DDR-333 1GB Module	2.5	2.19	2.66	5.48	2057.06	257.13	245
DDRII-667 2GB Module	1.8	2.88	5.34	5.18	971.51	121.44	139
DDR3-1333 2GB Module	1.5	3.68	10.66	5.52	517.63	64.70	52
DDR4-2667 4GB Module	1.2	5.50	21.34	6.60	309.34	38.67	39
HMC, 4 DRAM w/ Logic	1.2	9.23	128.00	11.08	86.53	10.82	13.7

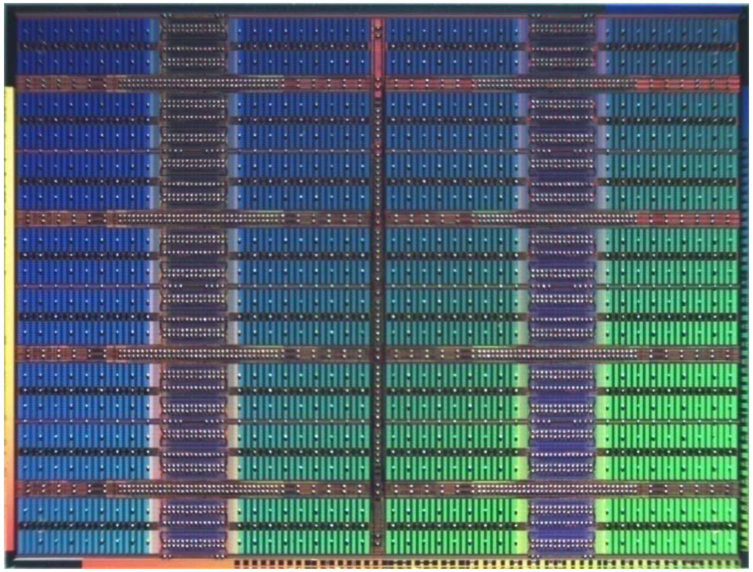
Simple calculation from IDD7 (SDRAM IDD4)

Real system, some with lower density modules

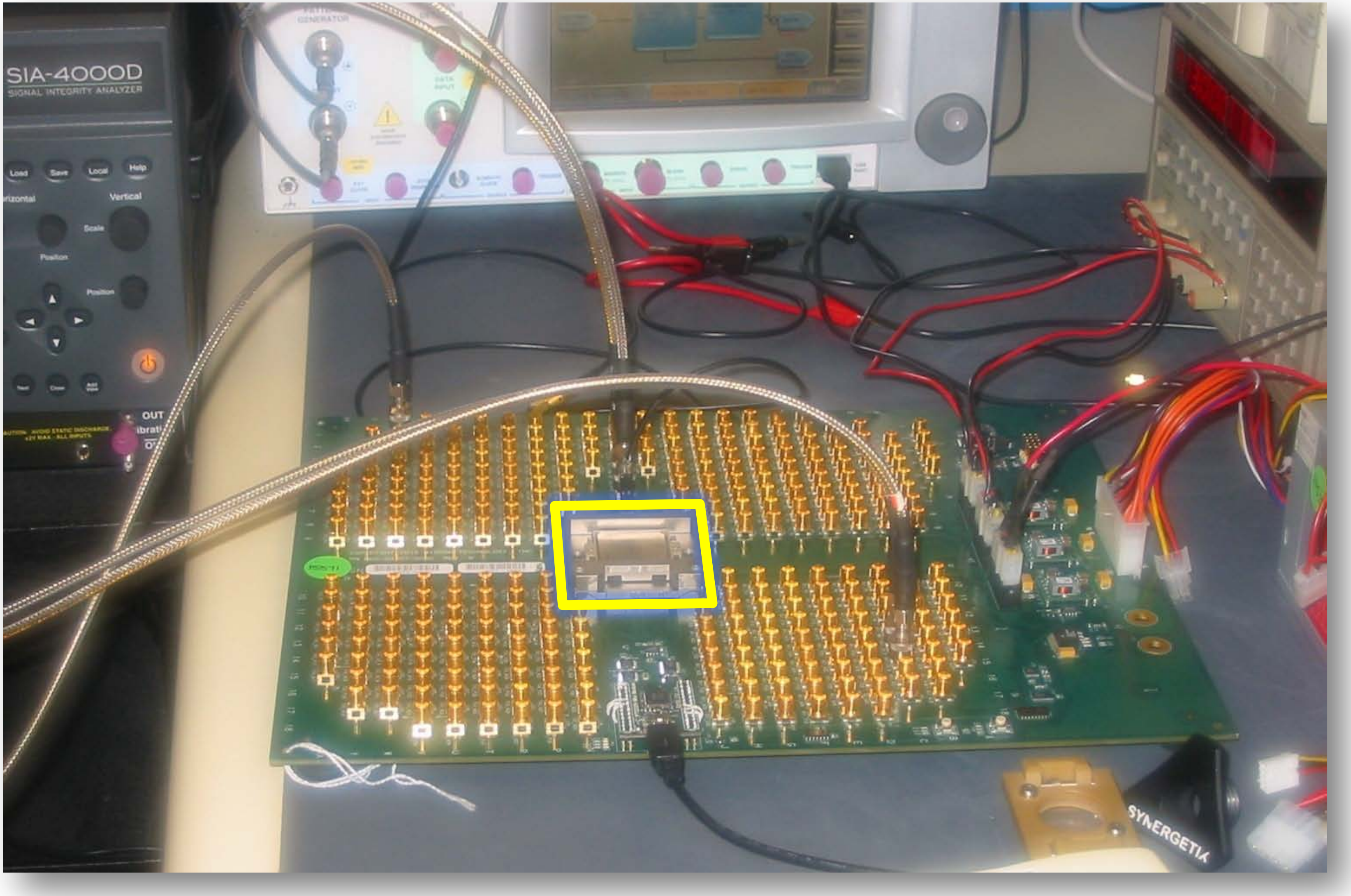
- 1Gb 50nm DRAM Array
- 90nm prototype logic
- 512MB total DRAM cube
- 128GB/s Bandwidth
- 27mm x 27mm prototype
- Functional demonstrations!
- Reduced host CPU energy



HMC Gen 1 DRAM



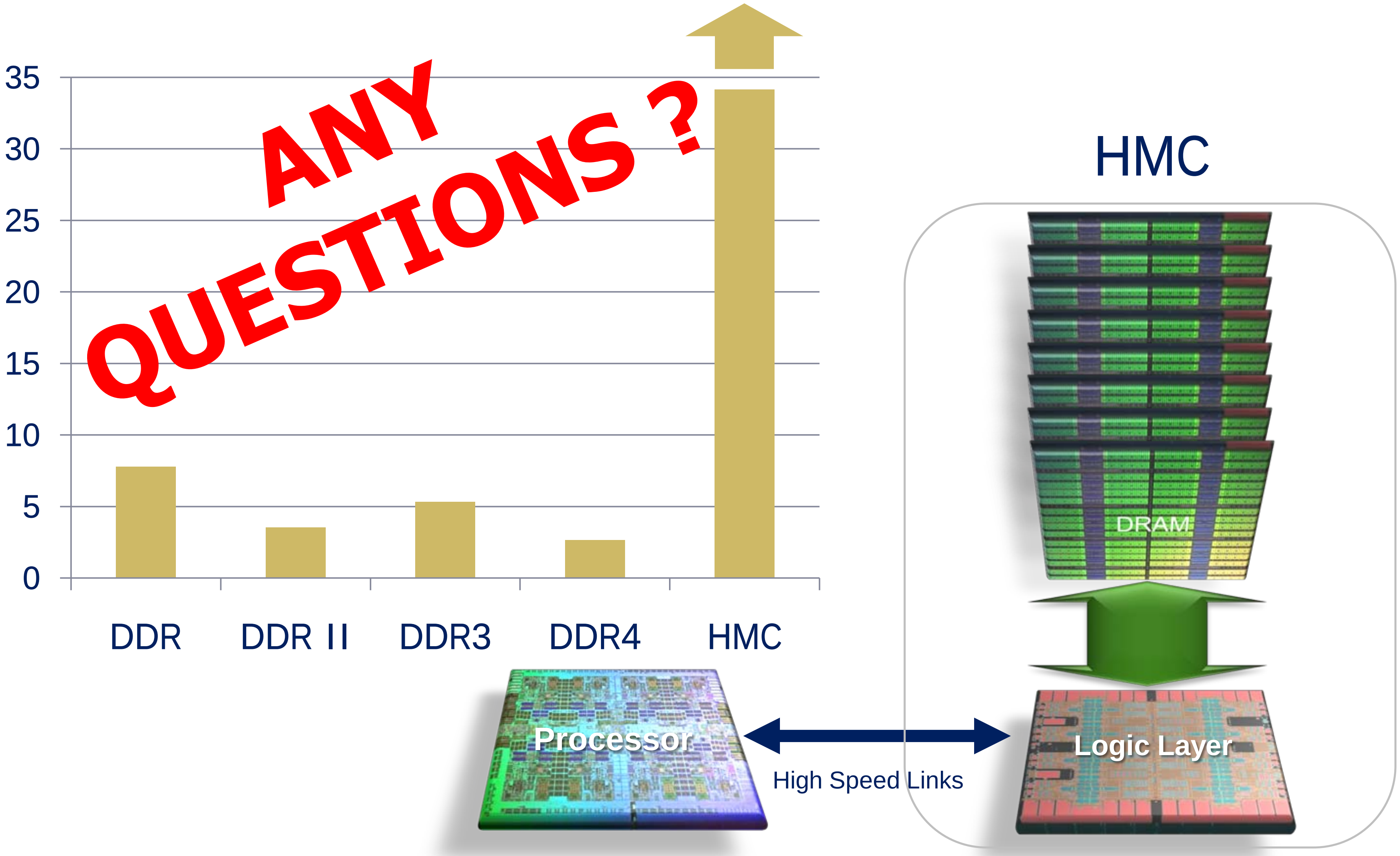
HMC Demonstration Platform



Summary

- Revolutionary DRAM performance improvement demonstrated by
  - Changing to abstracted high-speed buses
  - Employing 3D packaging using a hybrid of DRAM and logic technologies
  - Pulling in and improving the DRAM controller
  - Marrying DRAM and logic together using many TSVs
  - Completely rethinking DRAM architecture to exploit 3D
  - Managing component health for robust system solutions
- Result is >10x bandwidth, <1/3 energy, lower latency
- Request rates far beyond 2 billion operations per second
- Logic layer flexibility allows HMC to be tailor-made for multiple platforms and applications
- Scalable to ANY performance level. Imagine the possibilities.

Energy x Bandwidth Improvement Per DRAM Generation



ANY QUESTIONS ?