Tomorrow's Memory Systems

(2017 Edition)

Bruce Jacob

Keystone Professor
University of Maryland

Bruce Jacob

University of Maryland

Talk Outline

Bandwidth] **DRAM - HBM/HMC***

Capacity
Low Power
Nonvolatility
] **Flash, 3DXP, RRAM, PCM, etc - NVMM***

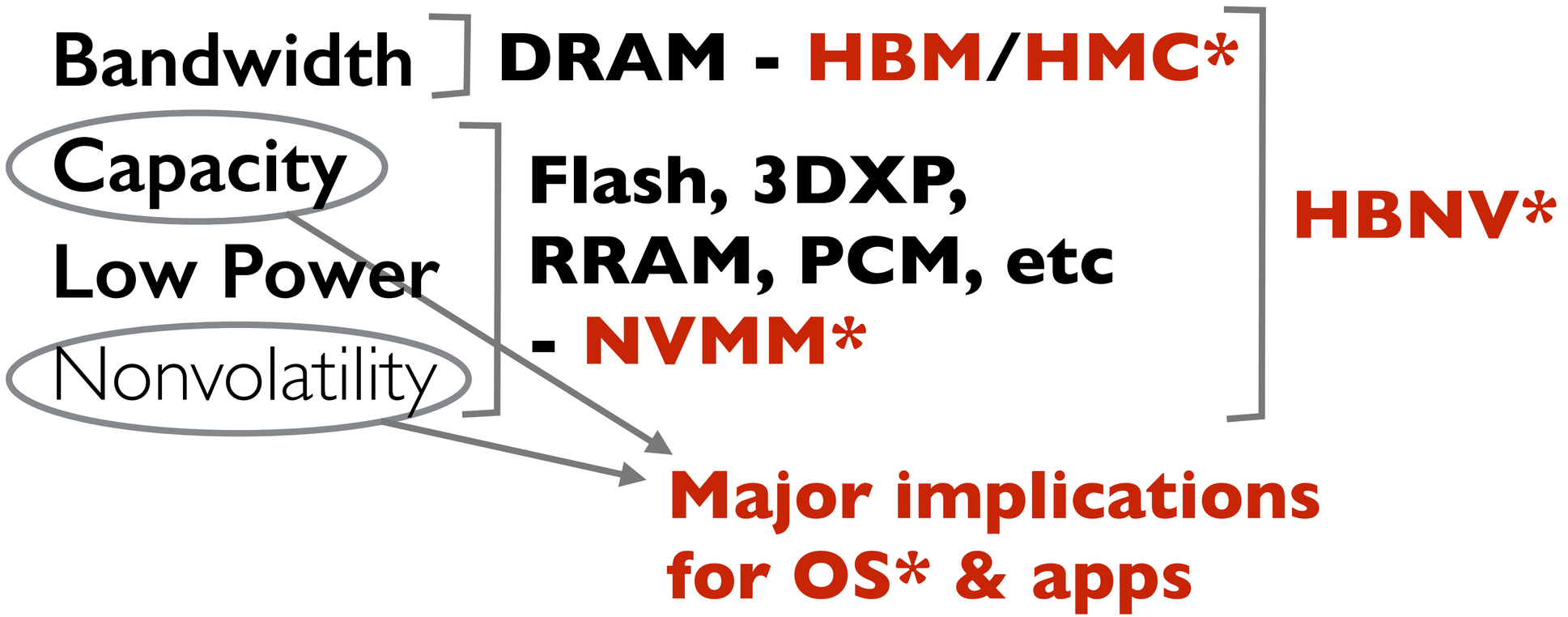
[**HBNV***]

* Things we did and/or are doing now (I'll cover in talk)

Bruce Jacob

University of Maryland

Talk Outline



* Things we did and/or are doing now (I'll cover in talk)

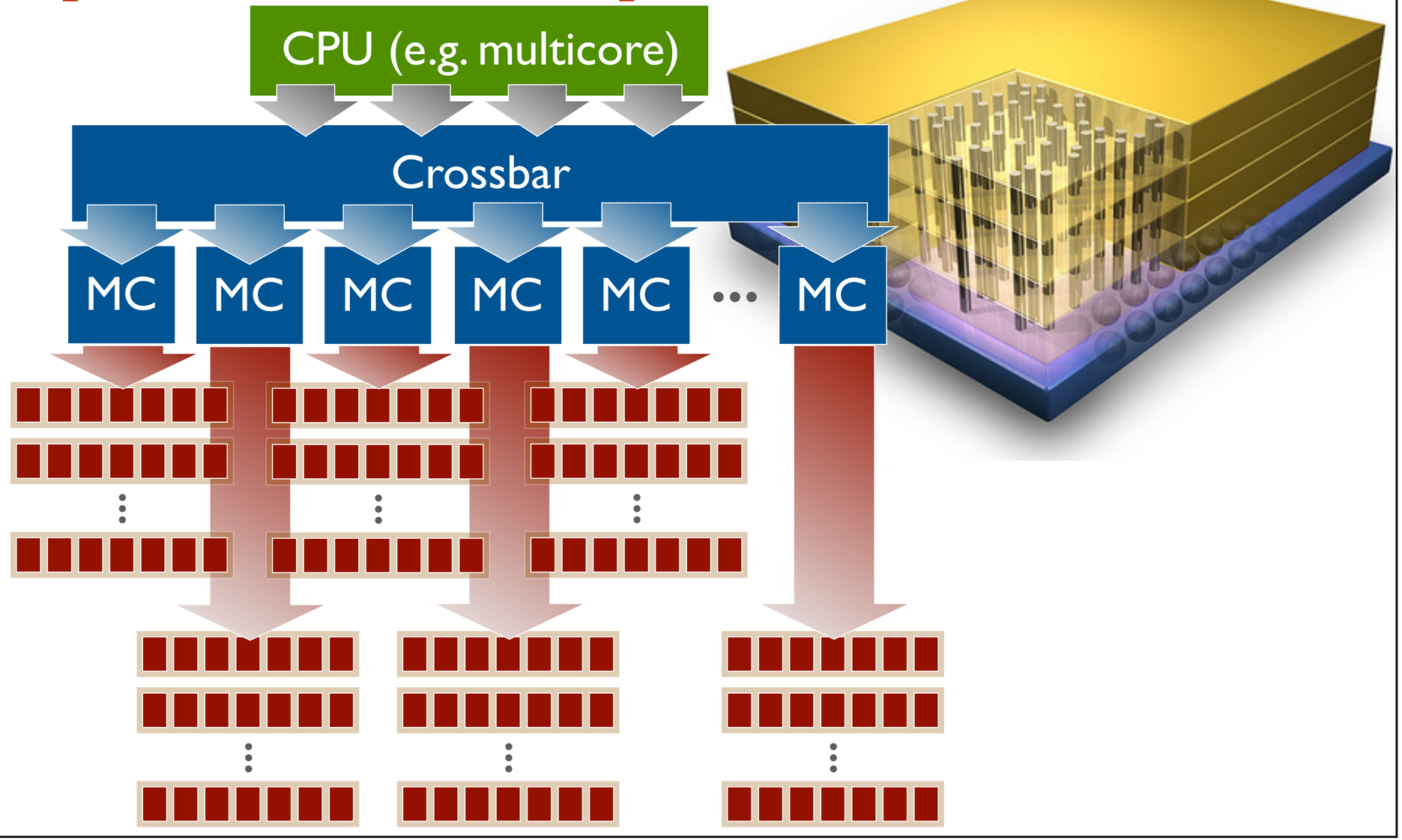
Hybrid Memory Cube

Off-chip: high speed SerDes and generic protocol

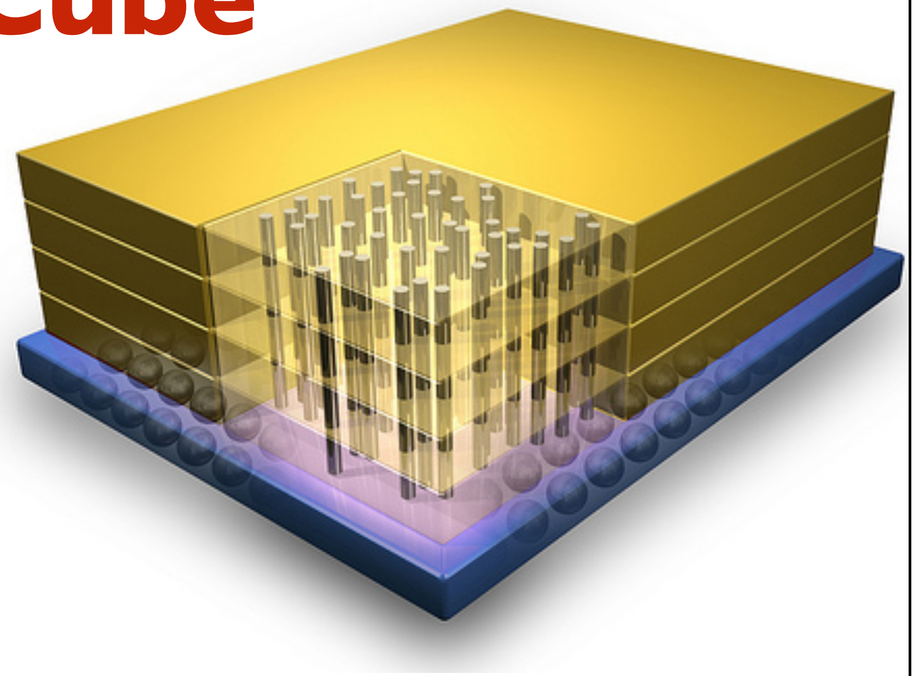
4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc'y = 16 x 8 x 2..8 (256–1024)



Hybrid Memory Cube



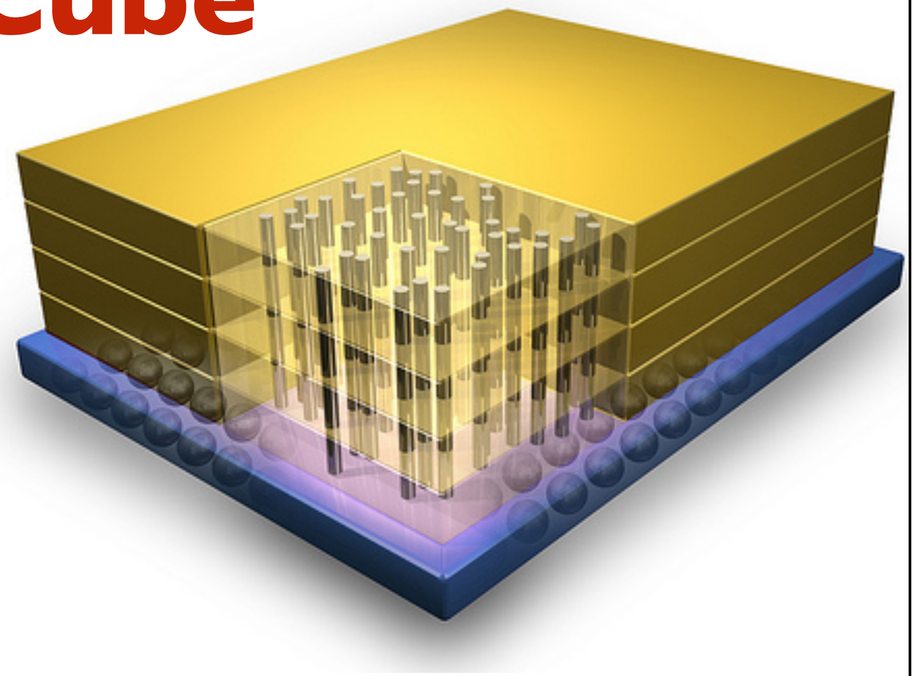
Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc'y = 16 x 8 x 2..8 (256–1024)

Hybrid Memory Cube

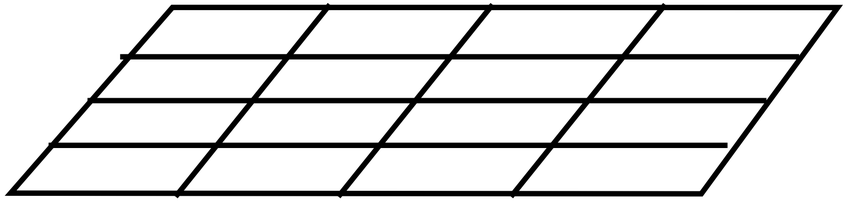


Off-chip: high speed SerDes and generic protocol

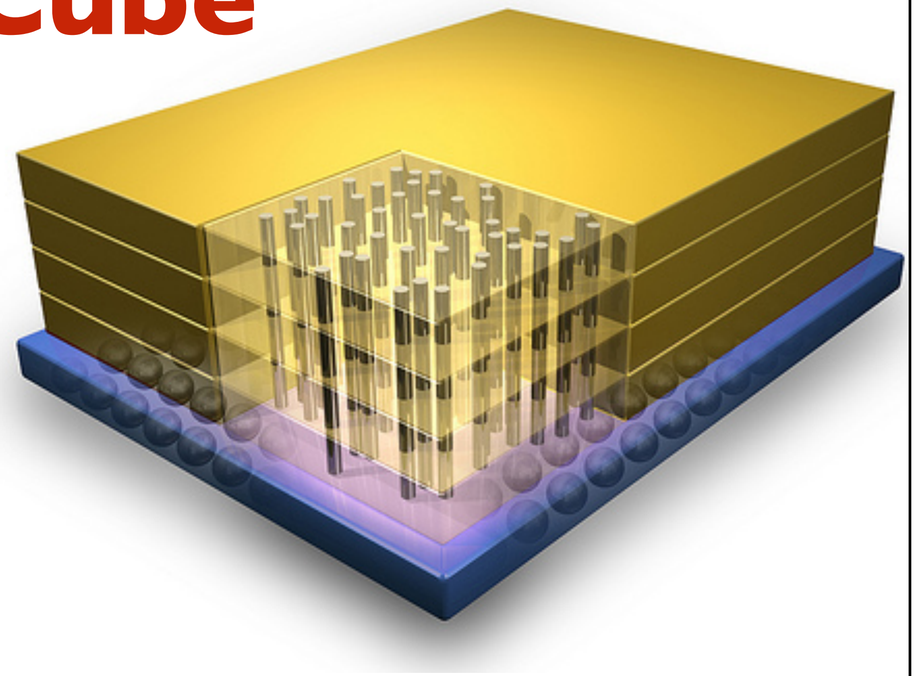
4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc'y = 16 x 8 x 2..8 (256–1024)



Hybrid Memory Cube

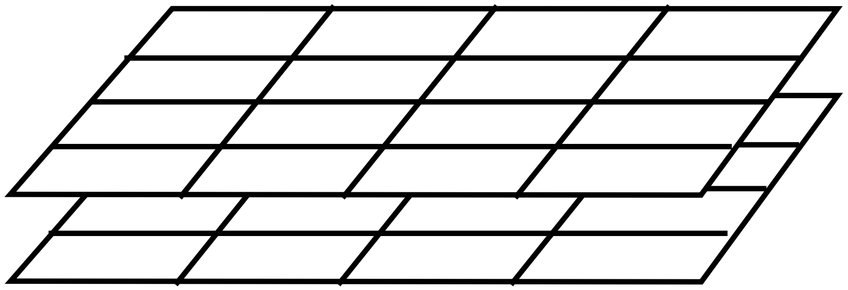


Off-chip: high speed SerDes and generic protocol

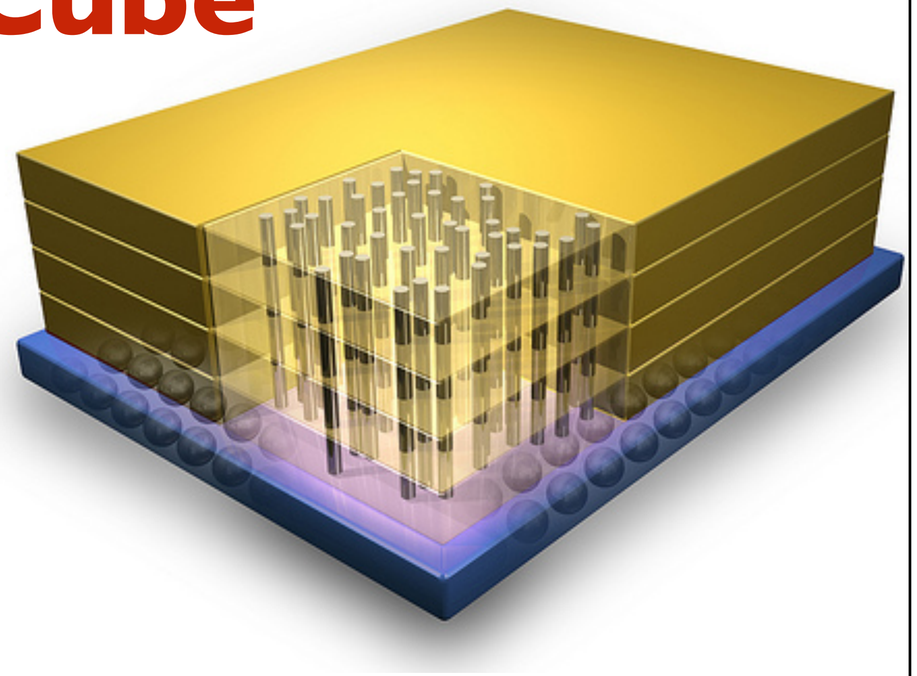
4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc'y = 16 x 8 x 2..8 (256–1024)



Hybrid Memory Cube

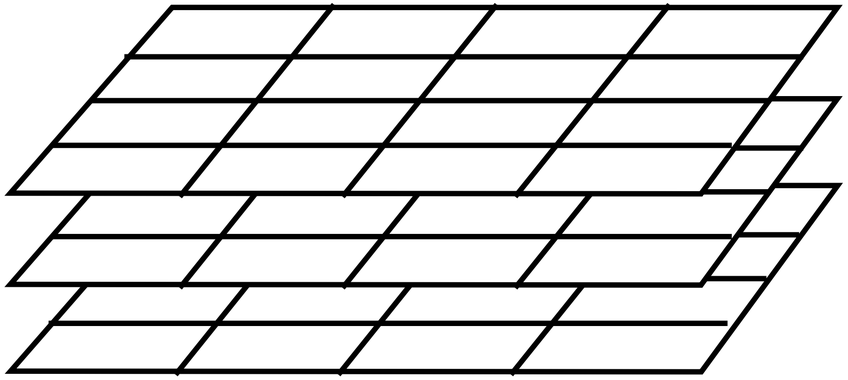


Off-chip: high speed SerDes and generic protocol

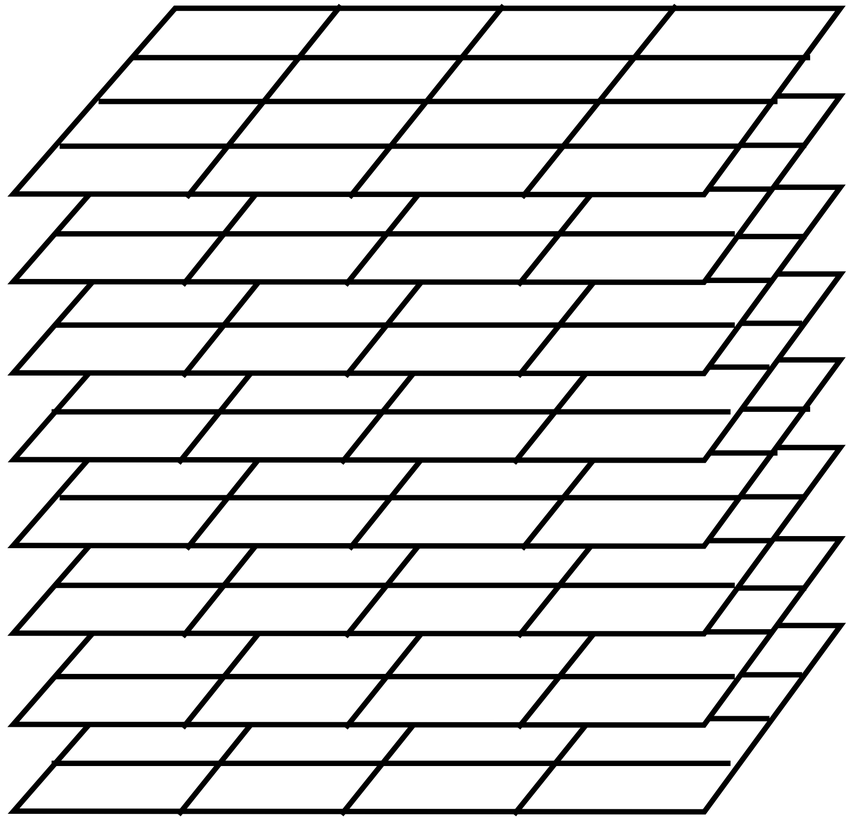
4 I/O Ports, up to 80 GB/s each

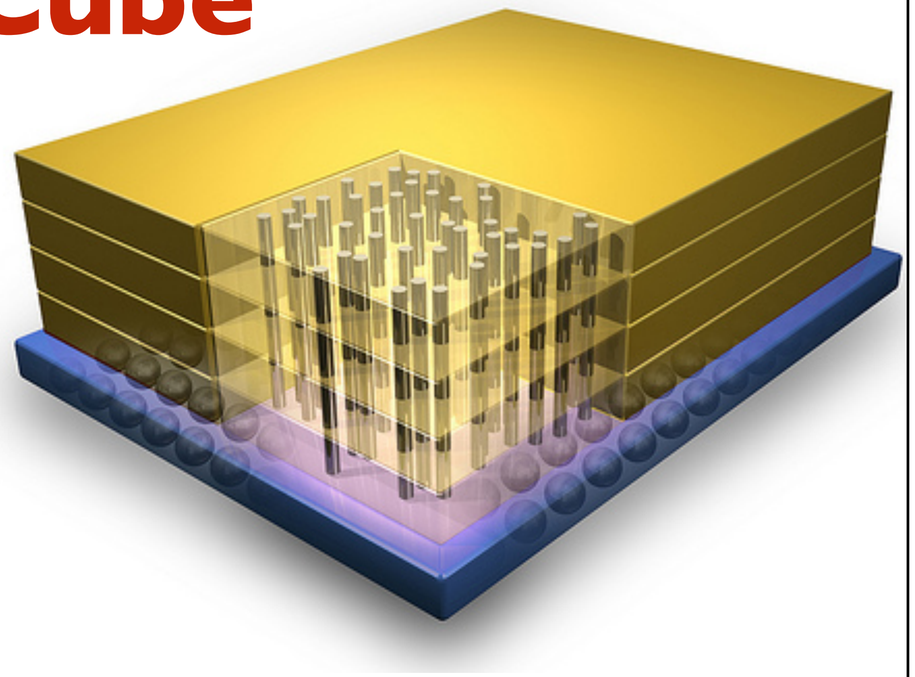
Next gen is **160** GB/s per (640 total)

Total conc'y = **16 x 8 x 2..8** (256–1024)



Hybrid Memory Cube





Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc'y = 16 x 8 x 2..8 (256–1024)

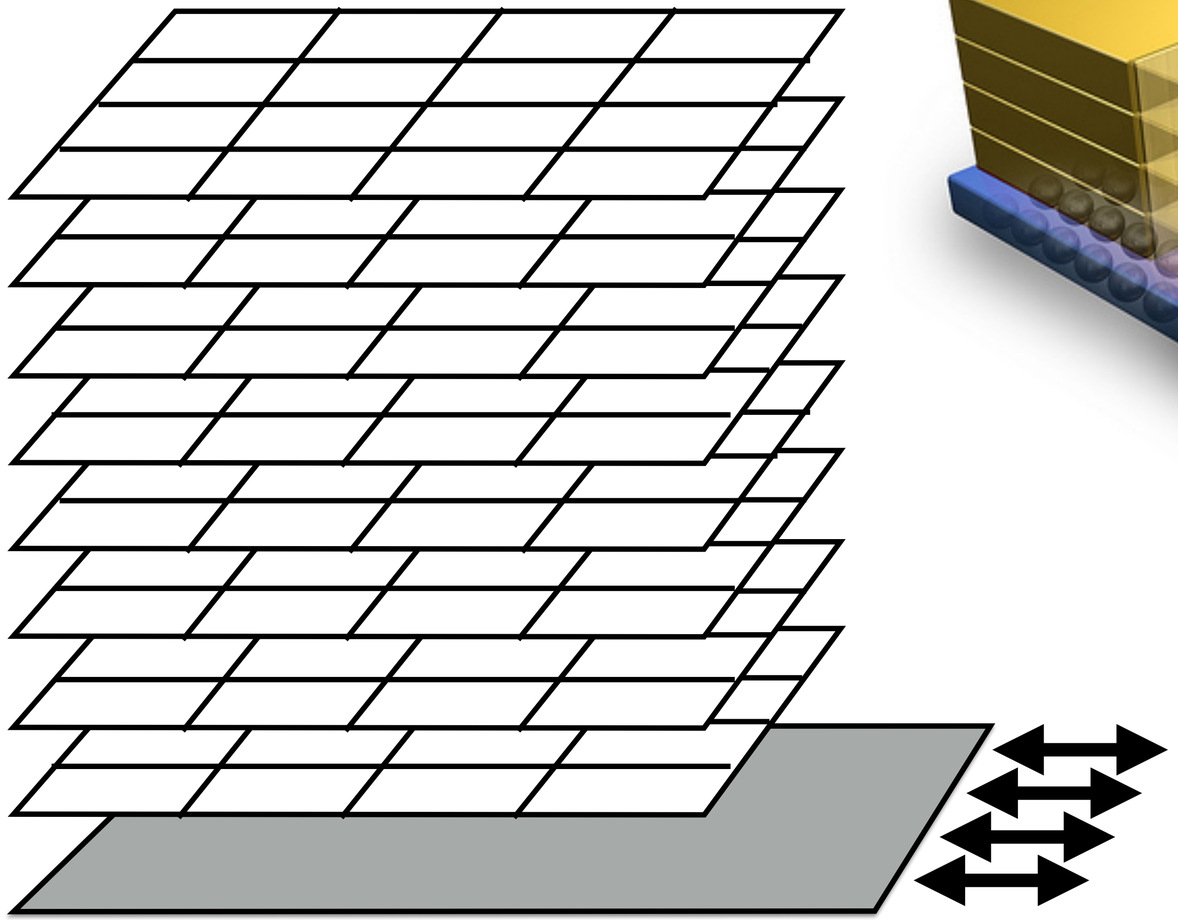
Hybrid Memory Cube

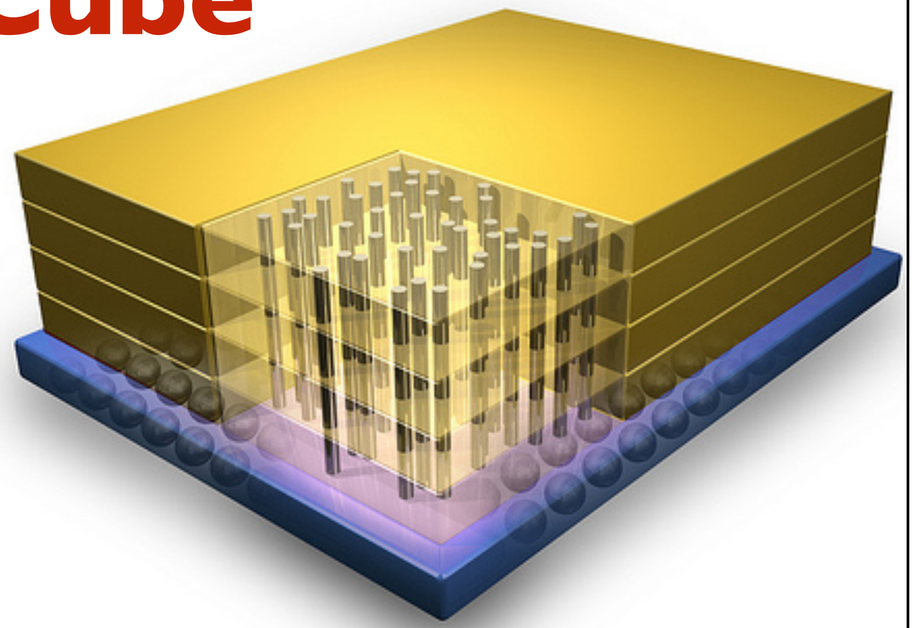
Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is **160** GB/s per (640 total)

Total conc'y = **16 x 8 x 2..8** (256–1024)





Logic Base
(I/O & CTL)

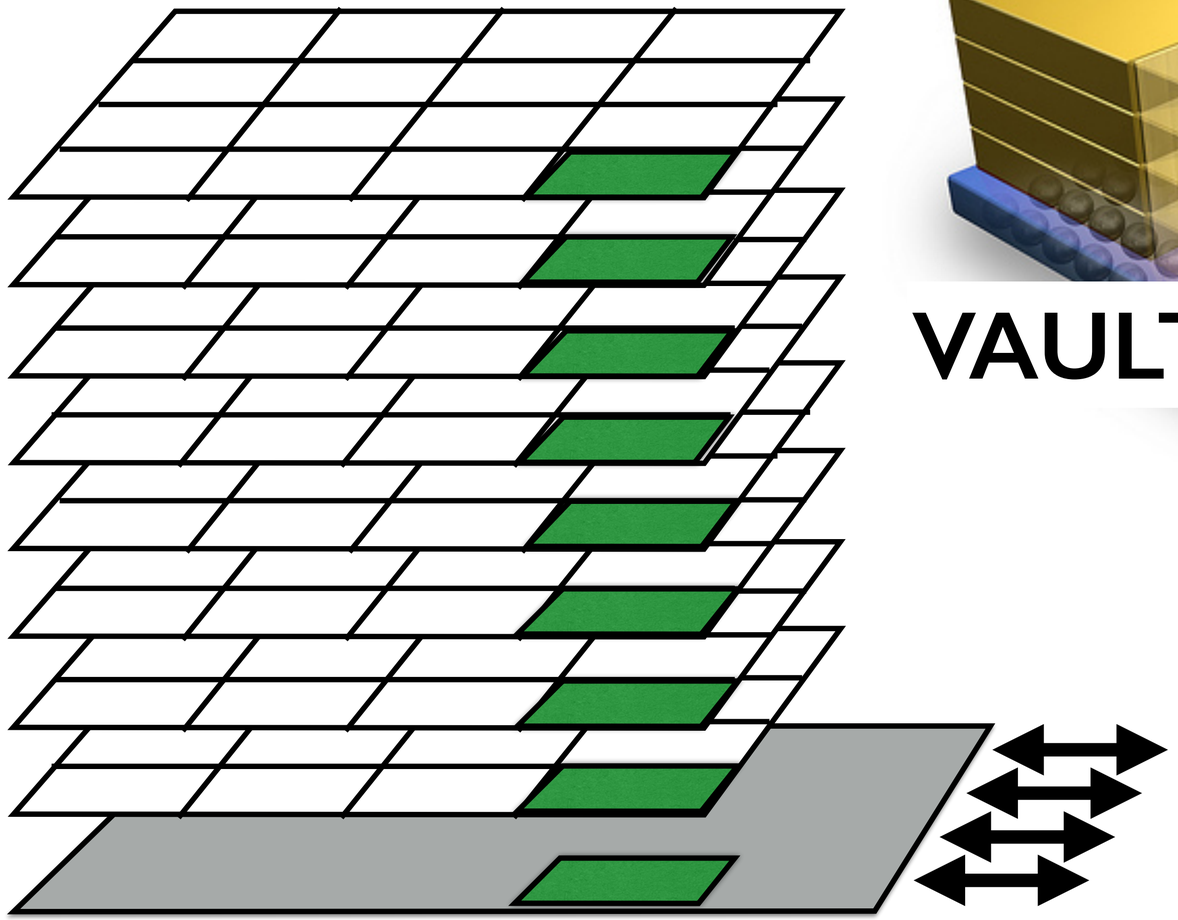
Hybrid Memory Cube

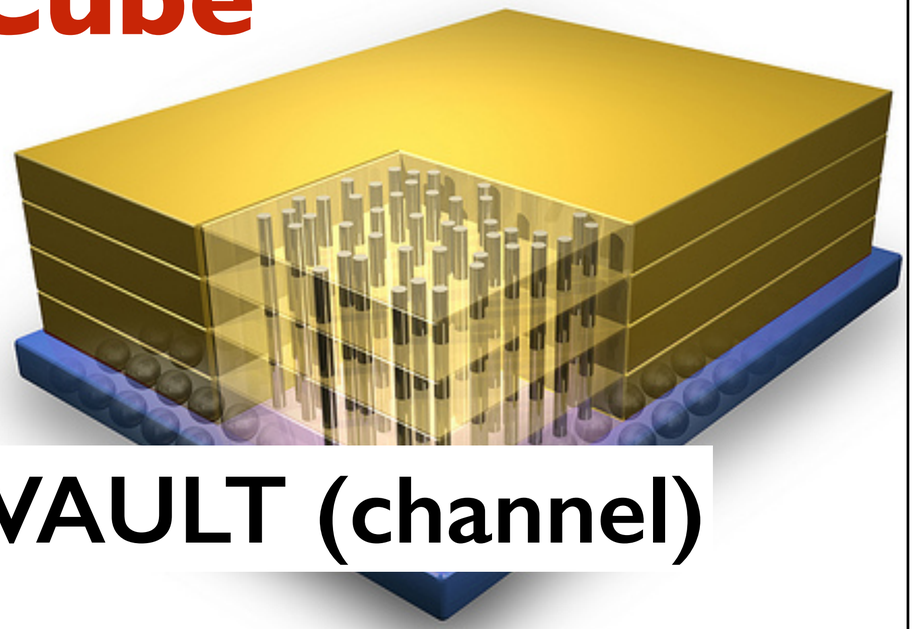
Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is **160** GB/s per (640 total)

Total conc'y = **16 x 8 x 2..8** (256–1024)





VAULT (channel)

Logic Base
(I/O & CTL)

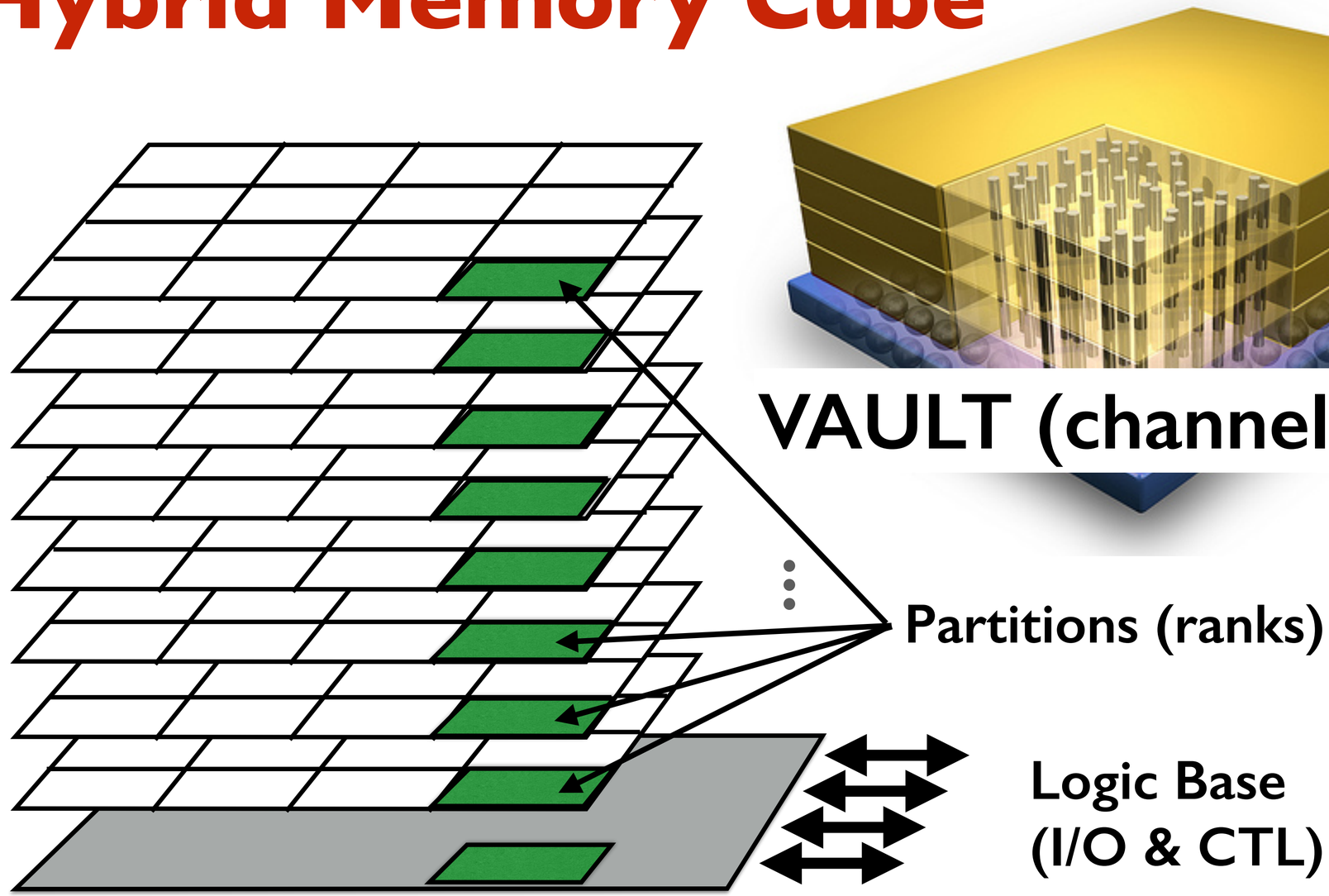
Hybrid Memory Cube

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

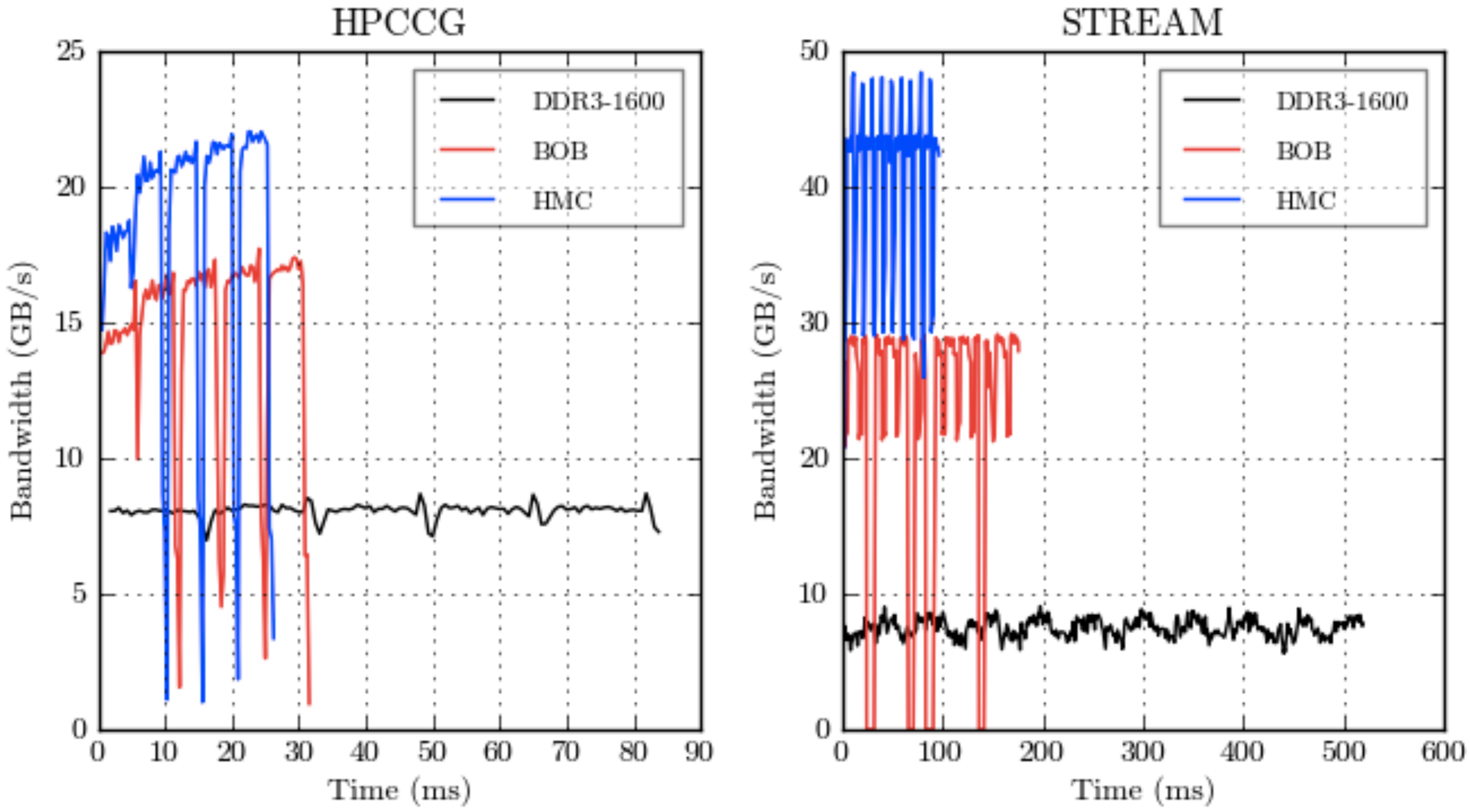
Next gen is **160** GB/s per (640 total)

Total conc'y = **16 x 8 x 2..8** (256–1024)



HMC Performance

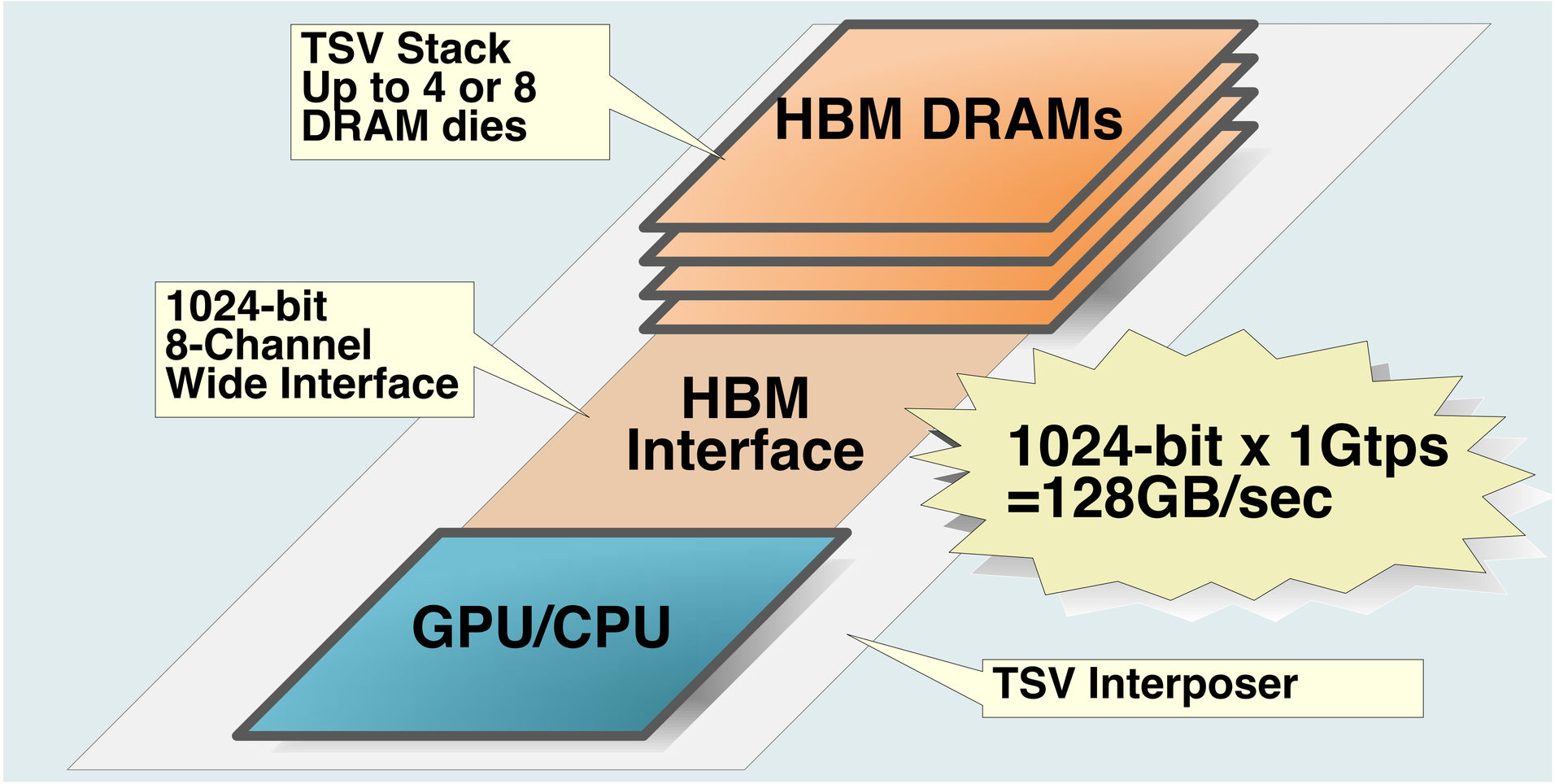
Execution can be several *times* faster than DDR3-1600



Source: Rosenfeld Ph.D. Thesis, U. Maryland 2014

High Bandwidth Memory

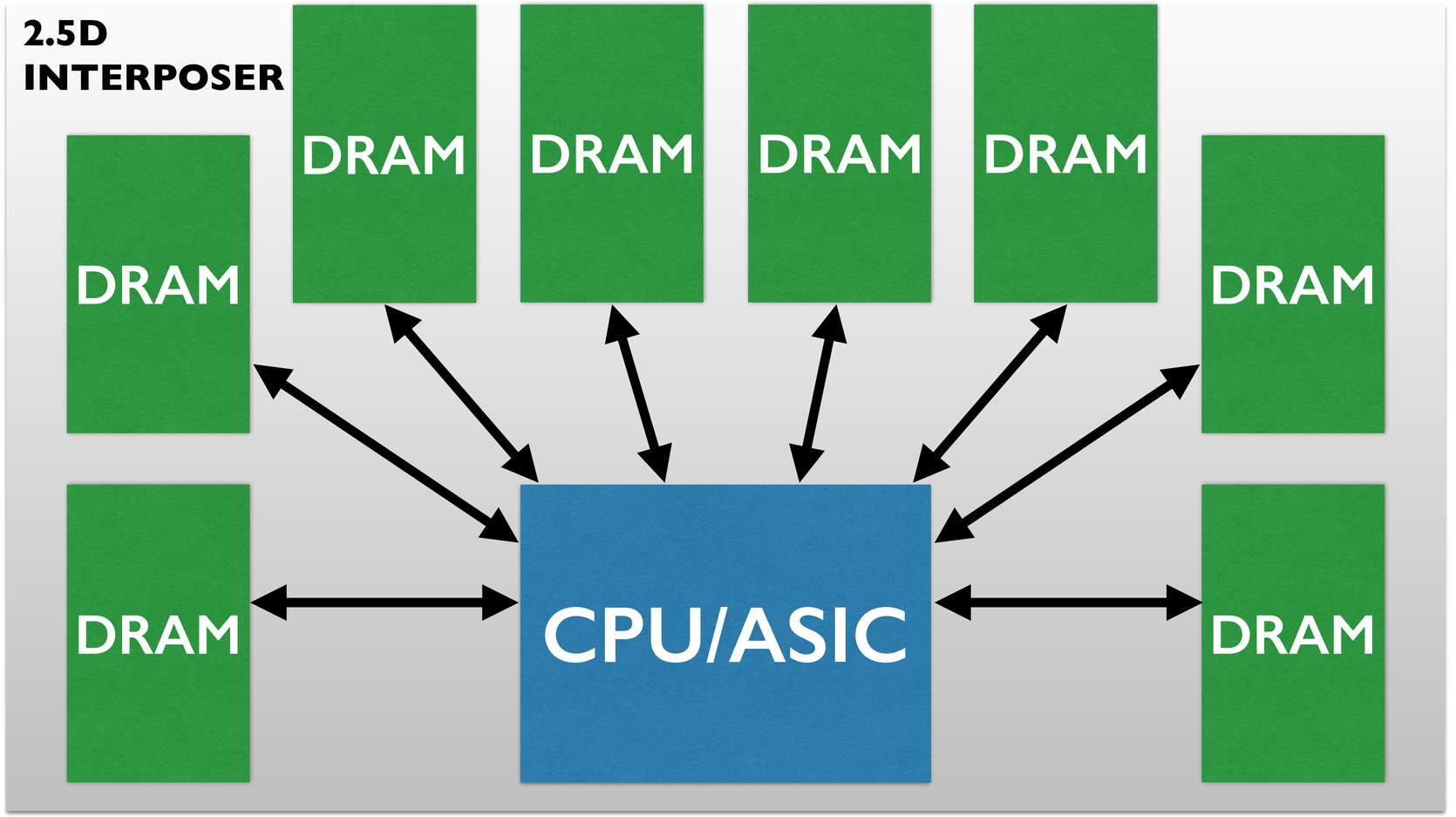
Uses a simple '2.5D' instead of full 3D stacking



Bruce Jacob

University of
Maryland

High Bandwidth Memory

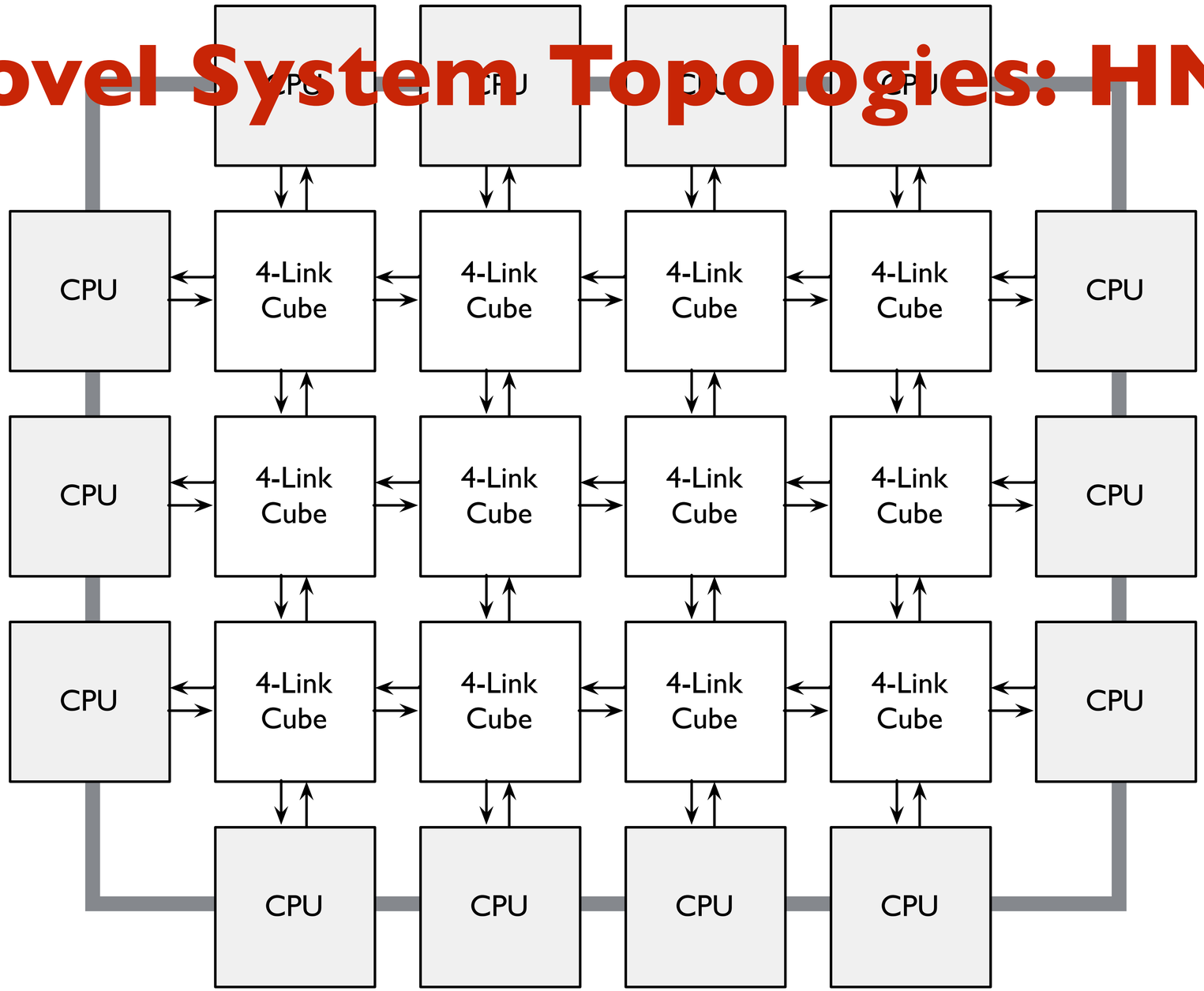


Each Link is 128 Bits Wide: *1024 Total*

Bruce Jacob

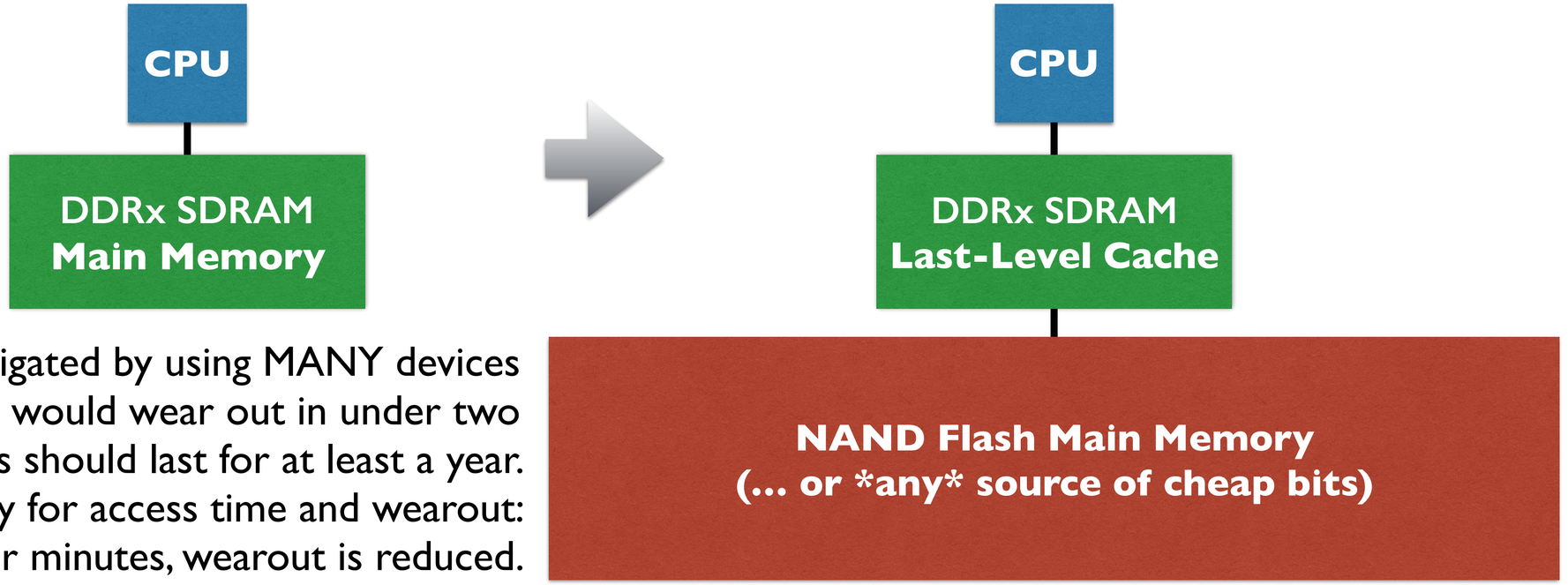
University of
Maryland

Novel System Topologies: HMC



Non-Volatile Main Memory

	Cost for 10 GB	Size of 10 GB	Power for 10 GB	Power per GB/s
Off-Chip SRAM	**$1,000**	**1 bucket**	**0.1–1 W**	**0.1 W**
DDR4 SDRAM	**$100**	**1 DIMM**	**1 W**	**0.1 W**
NAND Flash	**$10**	**<1 chip**	**0**	**0.1 W (?)**
3D XPoint	**$40**	**<1 chip**	**0**	**0.1 W (?)**

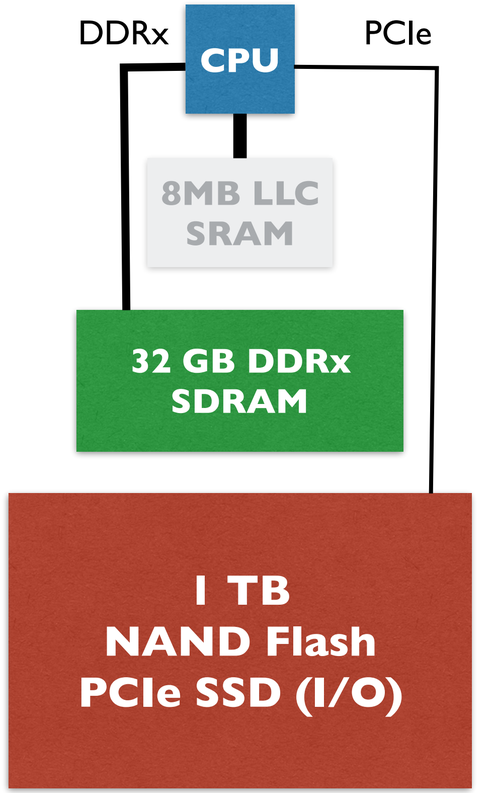


Note: wear-out mitigated by using MANY devices (thousands). A single device would wear out in under two days; therefore, 1000 devices should last for at least a year. Next, you can trade off longevity for access time and wearout: if the data need only last hours or minutes, wearout is reduced.

Bruce Jacob

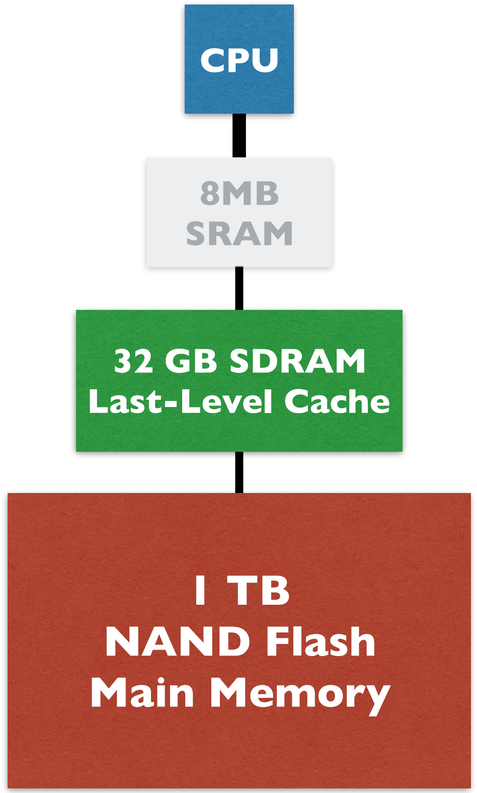
University of
Maryland

A Tale of 3 Memory Systems



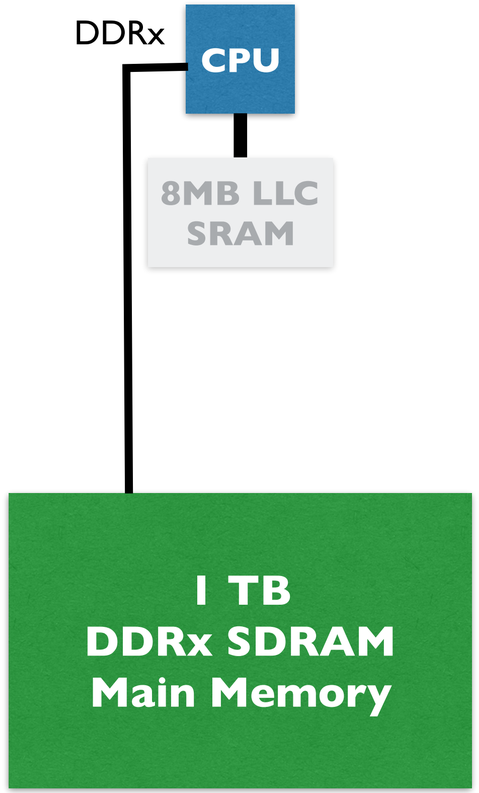
SSD

$500 – 10W



NVMM

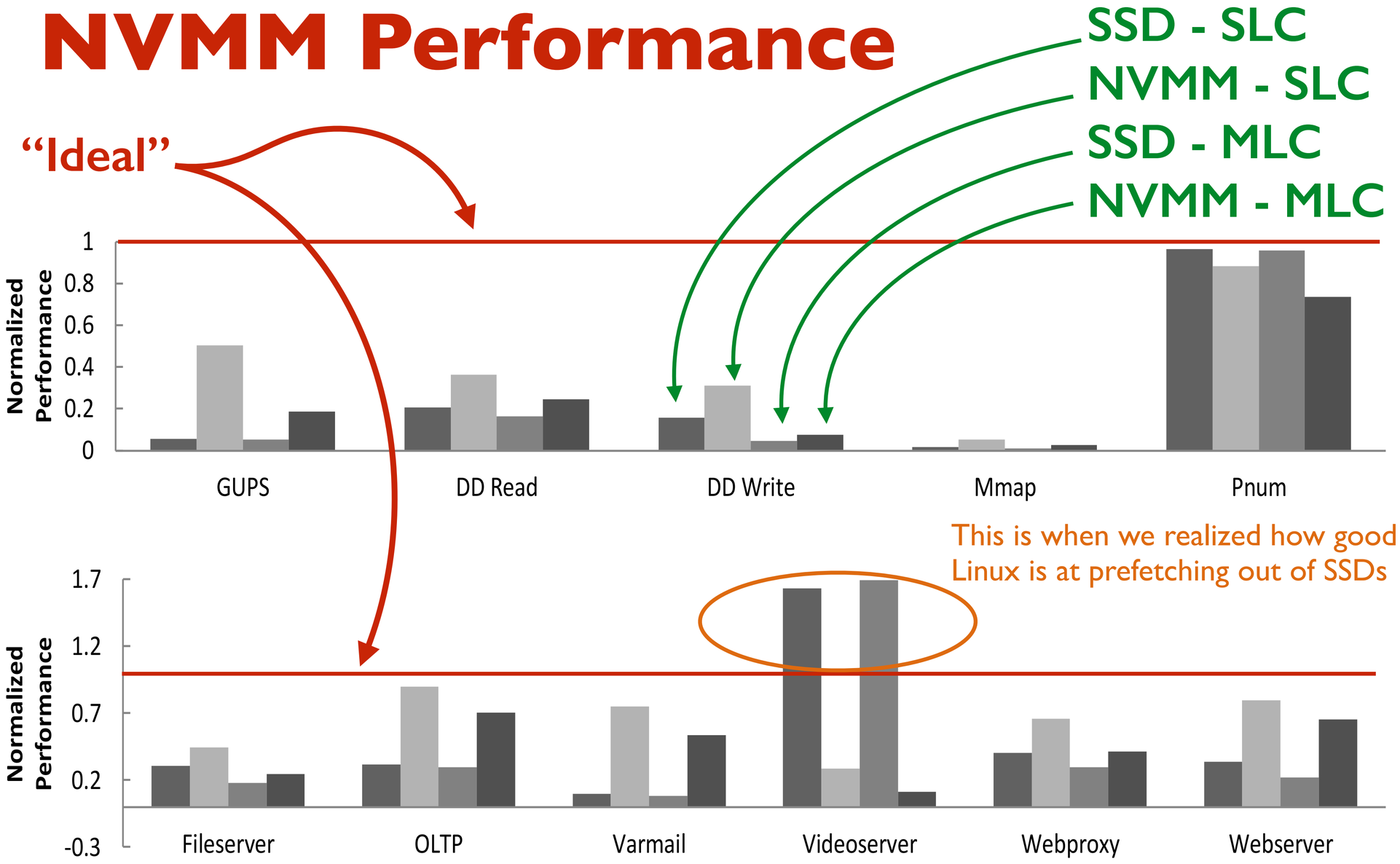
$500 – 10s of W



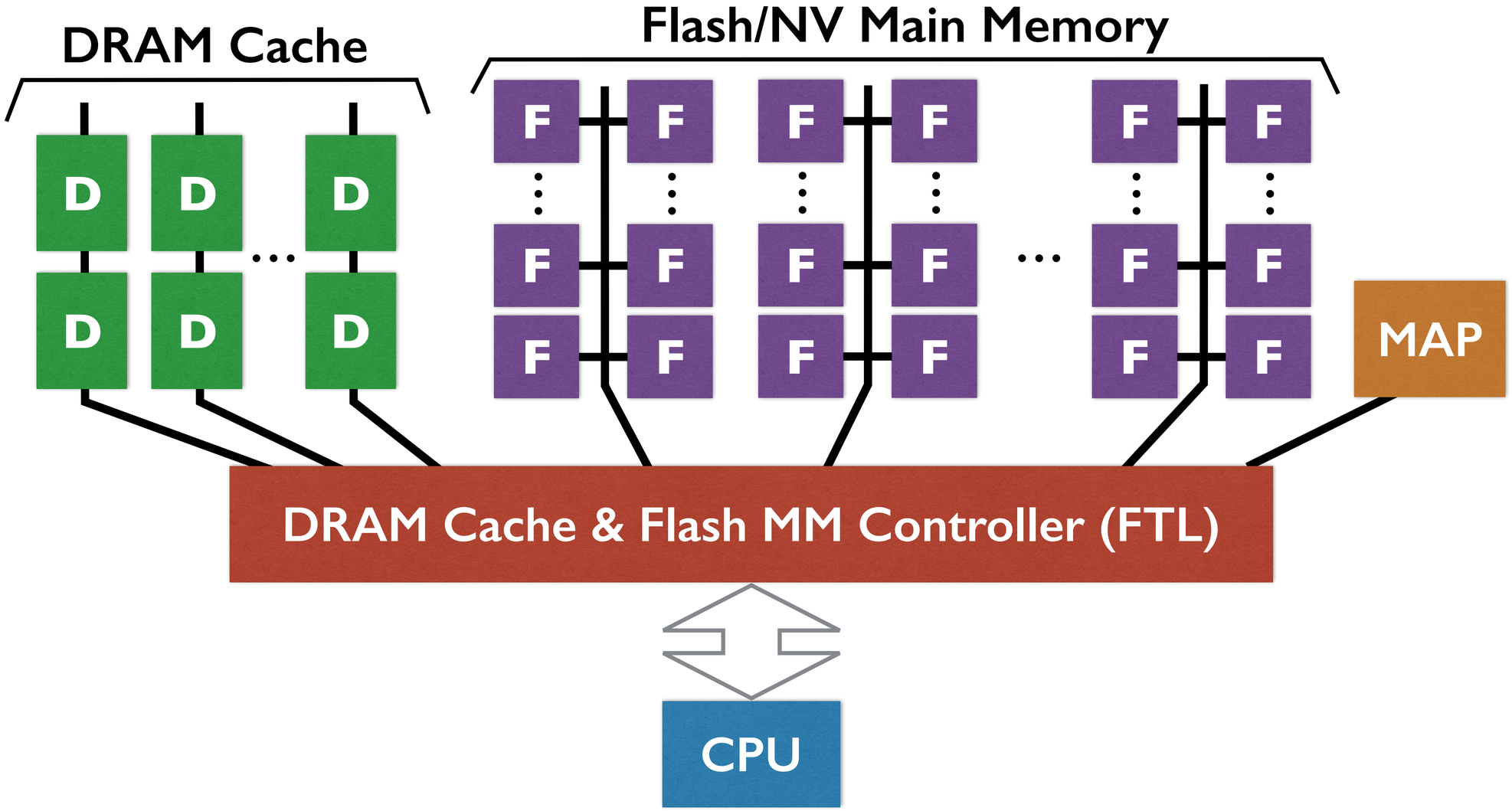
Ideal

$10,000 – 100W

NVMM Performance



Yeah, it's a lot of engineering



Bruce Jacob

University of Maryland

High Bandwidth Non Volatiles

The problem: You want 1TB @ 320 GB/s

Pure DRAM	Pure NAND Flash
64 HMCs	400 ONFI-4 flash chips*
1TB	300 TB — :O
20,000 GB/s — :O	320 GB/s*
100 W static power	0 W static power
128-byte granularity	16,000-byte granularity

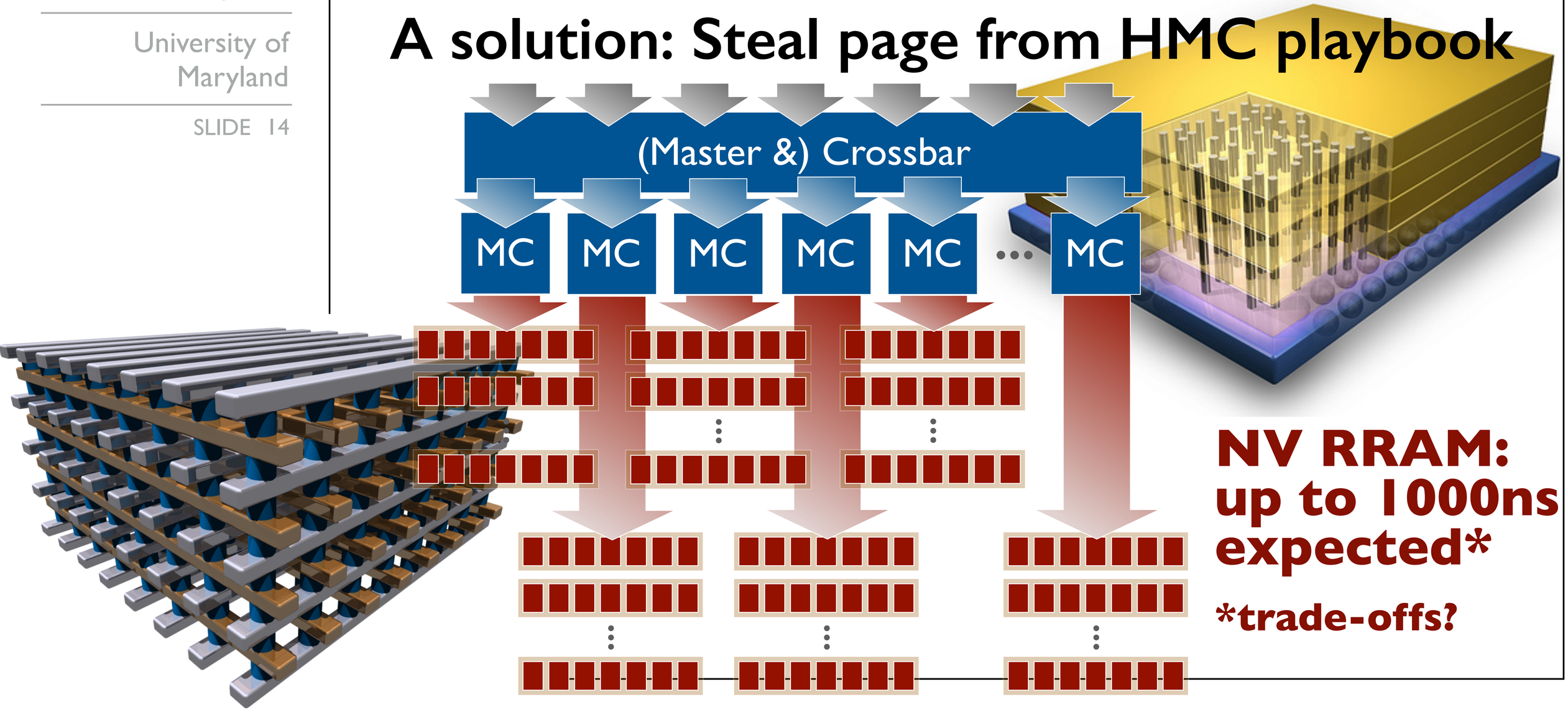
* on a 3200-pin parallel bus

Bruce Jacob

University of Maryland

High Bandwidth Non Volatiles

A solution: Steal page from HMC playbook



NV RRAM: up to 1000ns expected*

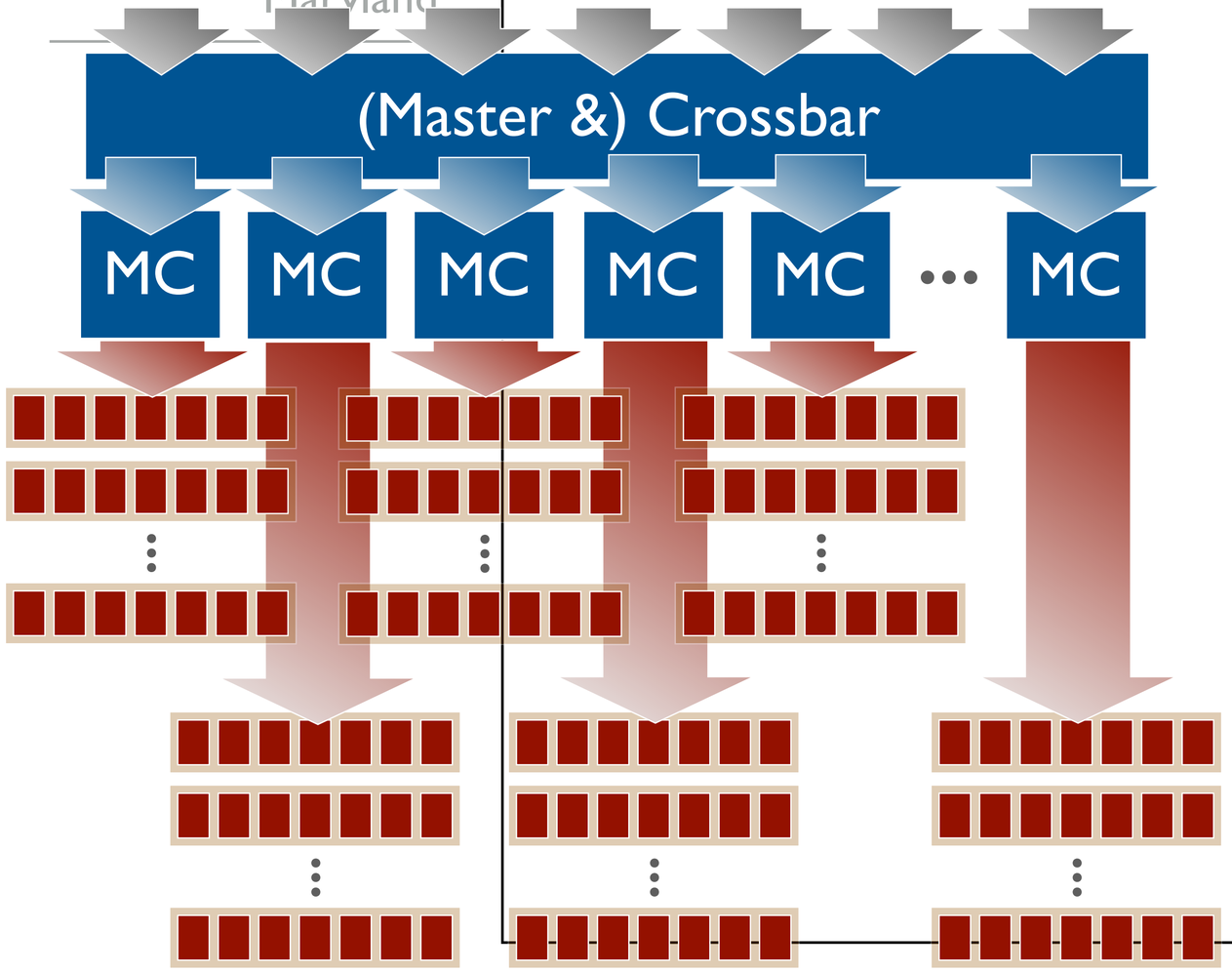
***trade-offs?**

Bruce Jacob

University of Maryland

High Bandwidth Non Volatiles

First-order concurrency requirements:



$$\frac{\text{bytes}}{\text{sec}} \bullet \frac{\text{sec}}{\text{access}} \bullet \frac{\text{access}}{\text{byte}}$$

$$\frac{320\ \text{GB}}{\text{sec}} \bullet \frac{1000\ \text{ns}}{\text{access}} \bullet \frac{\text{access}}{32\ \text{B}} = 10\text{K}$$

$$\frac{320\ \text{GB}}{\text{sec}} \bullet \frac{1000\ \text{ns}}{\text{access}} \bullet \frac{\text{access}}{256\ \text{B}} = 1250$$

$$\frac{160\ \text{GB}}{\text{sec}} \bullet \frac{500\ \text{ns}}{\text{access}} \bullet \frac{\text{access}}{128\ \text{B}} = 625$$

Bruce Jacob

University of
Maryland

Implications for Software

Compared to DRAM: 5x performance hit for a 100–1000x increase in **capacity**

- ➡ 10–100 TB main memory for 1-U server (*really* large data sets become realistic)
- ➡ Probably need lots of cores … sharing?

Nonvolatility opens up many questions:

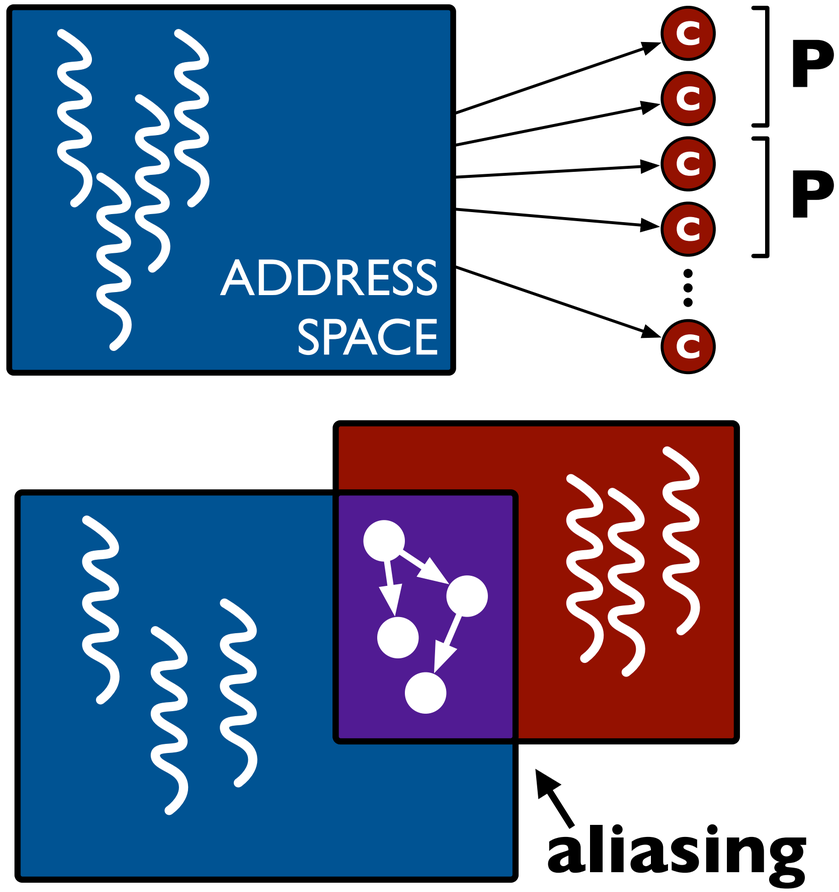
- ➡ Redesign VM+FS subsystems
- ➡ Journaled main memory (e.g. thru flash)
- ➡ Persistent objects (Mneme, POMS, etc)

Bruce Jacob

University of Maryland

Capacity Issues

Sharing & Coherence



Large **capacity** implies manycore

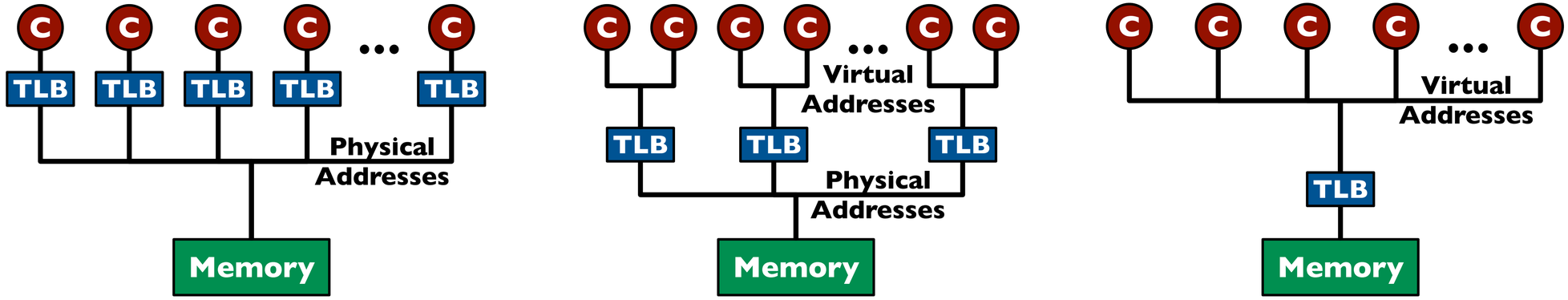
Multiple threads must map easily to different cores, regardless of hardware resources

Shared data resources **MAY** include pointers (as opposed to non-dynamic naming — easy)

Note: **persistence** implies same object name, not data location

Capacity Issues

Sharing & Coherence — Translation Point



Note: All support heterogeneous processes, shared memory, 0-based addressing, address-space protection, etc.

← Benefits:

- Larger effective TLB size
- Better performance (??)

Benefits: →

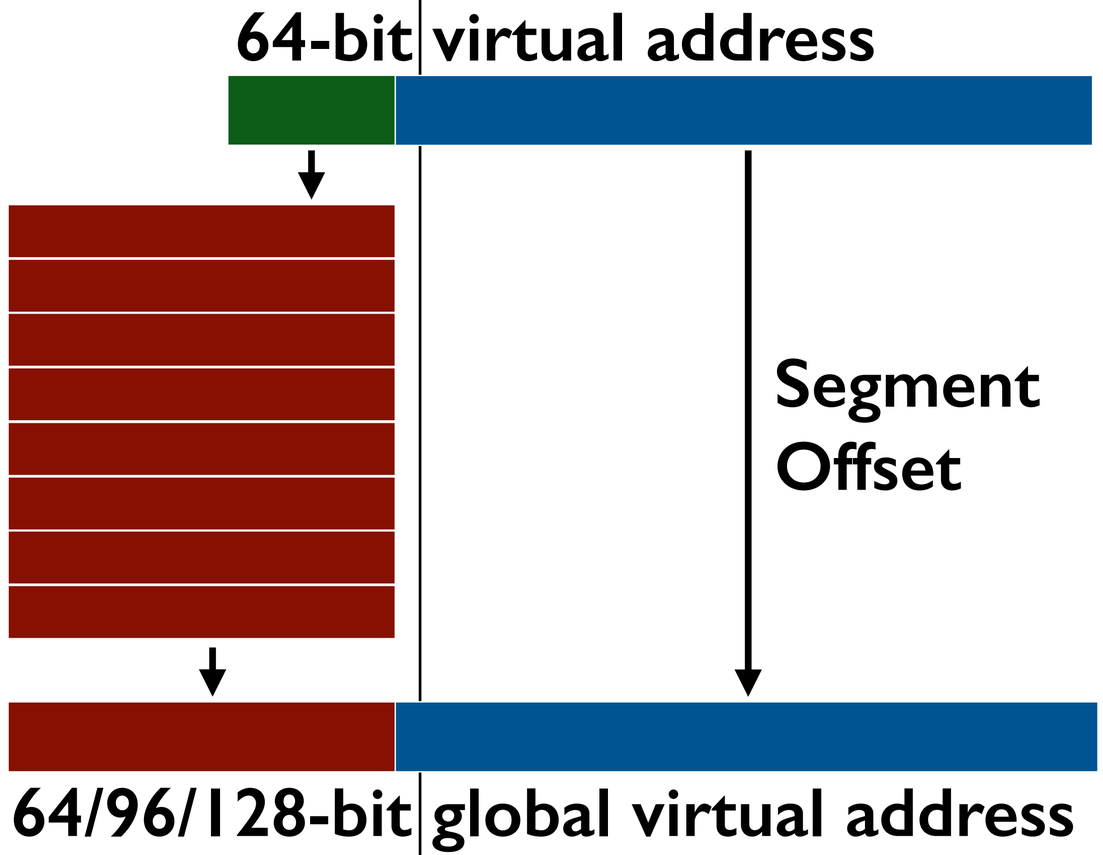
- Simpler coherence (less/no shootdown)
- **Lower power**

Bruce Jacob

University of Maryland

Capacity Issues

An argument for IBM 801-style segments



Back to the Goals:

- ✓ Supports simple mapping of threads/processes to cores
- ✓ Supports 0-based code & data
- ✓ Supports simple sharing at the segment level
- ✓ Allows different protections at segment level (**A** = WO, **B** = RO)

The Big Important Question (?):

Can I have BOTH 0-based address spaces AND shared pointers?

Nonvolatility Issues

Unified VM+FS Subsystems

➡ *Motivating example: OSF/1*

➡ *Possible directions:*

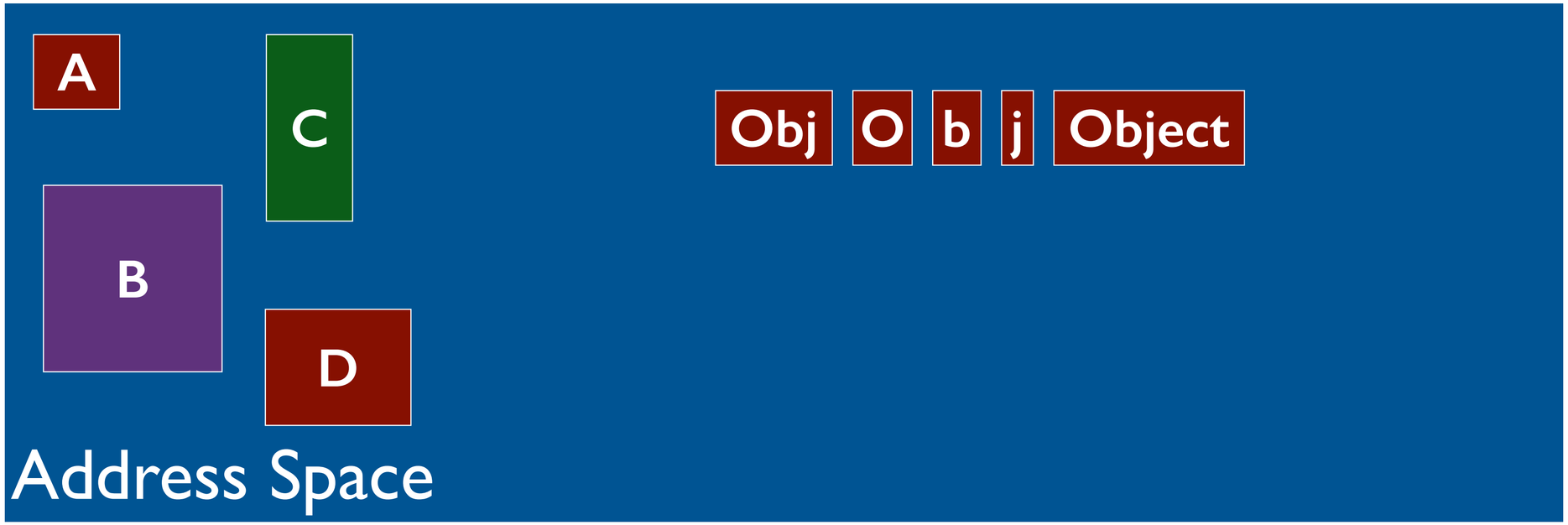
- Persistent objects (e.g. Mneme, POMS) [failed only due to reliance on disk]
- Named regions

➡ *By default, data in process address space temporary, garbage-collected at exit(); **permanentify** function bypasses this*

Nonvolatility Issues

Unified VM+FS Subsystems

➡ **Persistent Objects** (arguably more elegant)

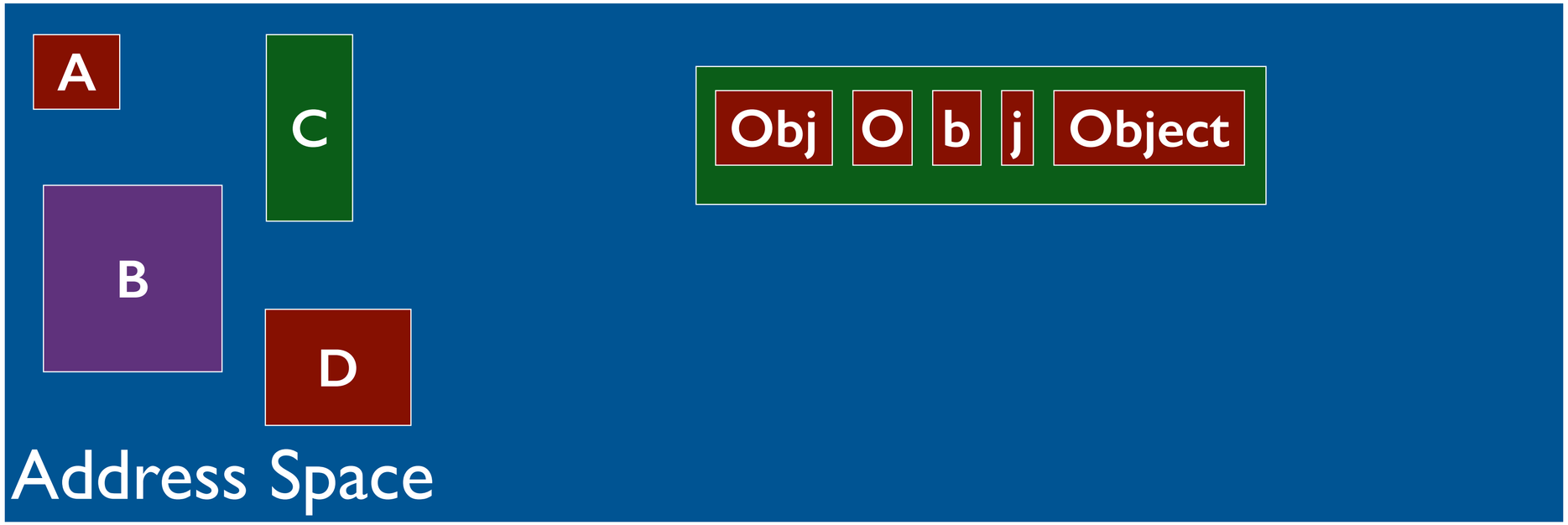


➡ **Access via Object references**

Nonvolatility Issues

Unified VM+FS Subsystems

➡ **Persistent Objects** (arguably more elegant)



➡ **Access via Object references**

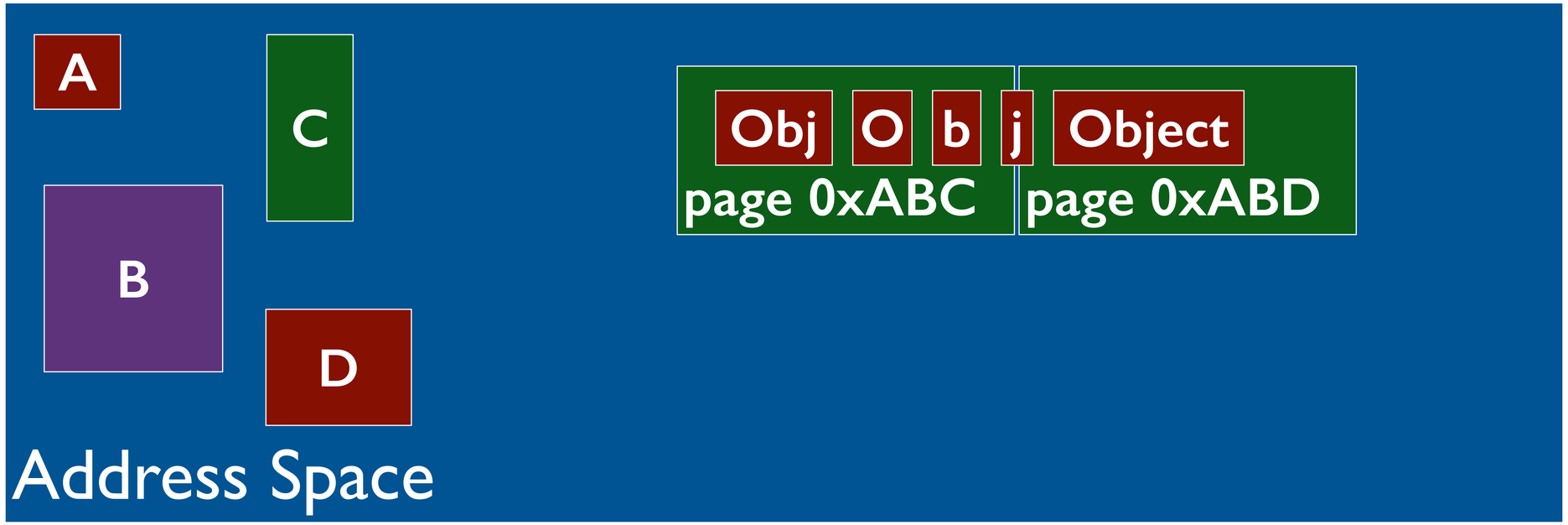
Bruce Jacob

University of Maryland

Nonvolatility Issues

Unified VM+FS Subsystems

➡ **Named Regions** (arguably far simpler)

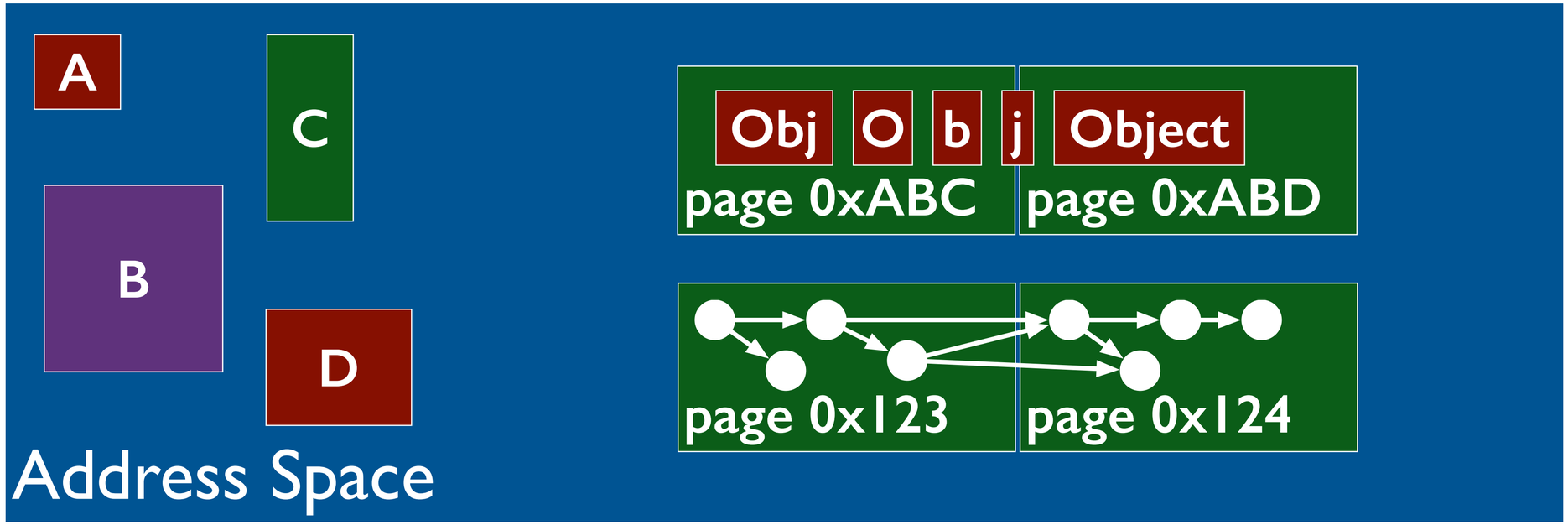


➡ **Access via 0xABC/D or "stringname"**

Nonvolatility Issues

Unified VM+FS Subsystems

➡ **Named Regions** (arguably far simpler)

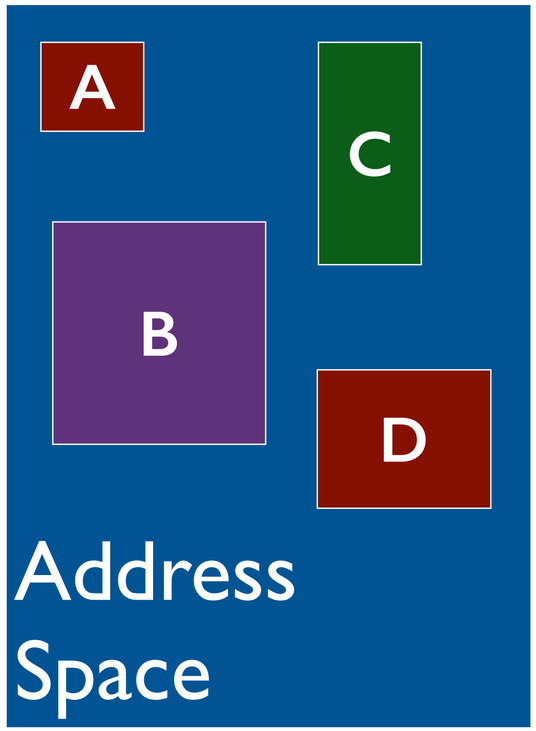


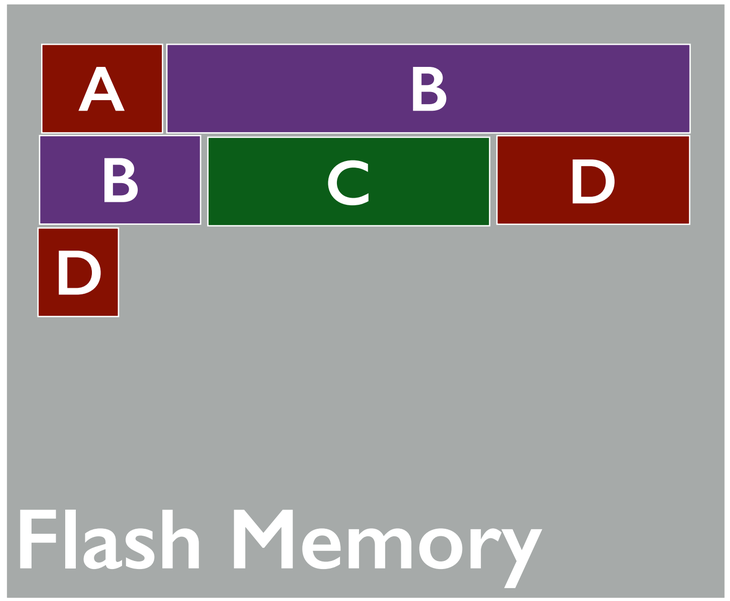
➡ **Access via 0xABC/D or "stringname"**

Nonvolatility Issues

Journaled Main Memory (built-in checkpoint)

➡ *Here's the way flash works:*





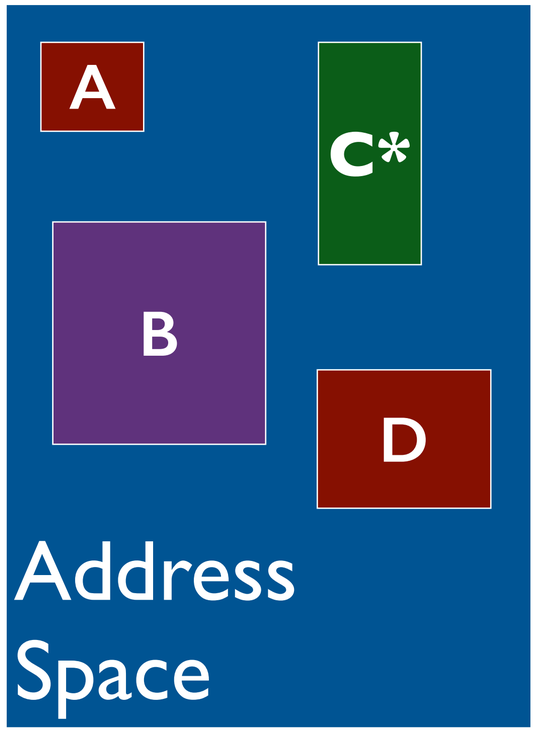
A: 0–2
B: 3–15,16–19
C: 20–26
D: 27–32,33–34

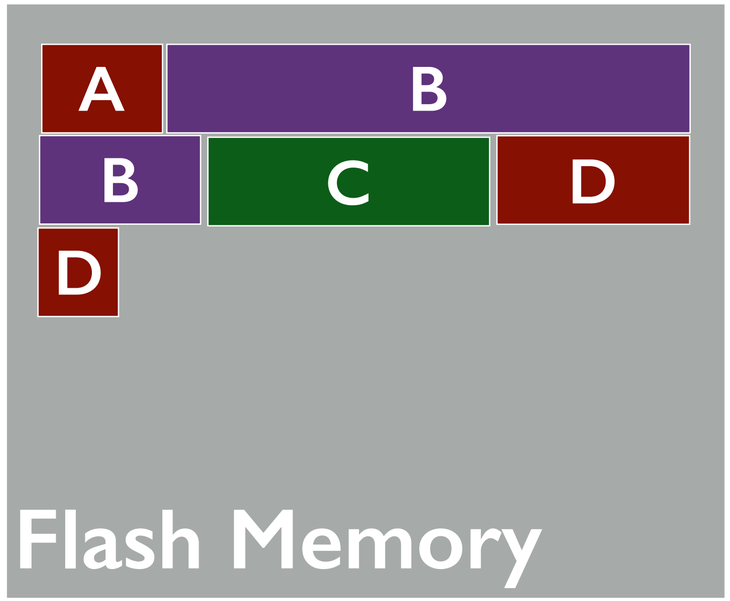
FTL

Nonvolatility Issues

Journaled Main Memory (built-in checkpoint)

➡ *Here's the way flash works:*





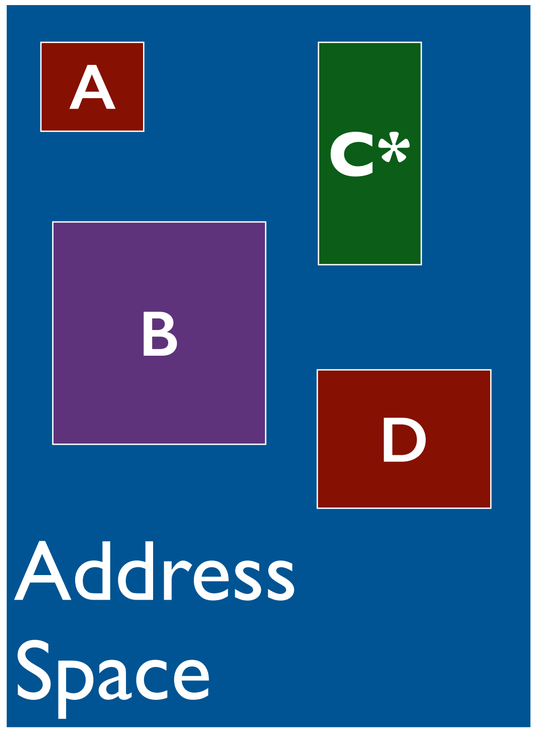
A: 0–2
B: 3–15,16–19
C: 20–26
D: 27–32,33–34

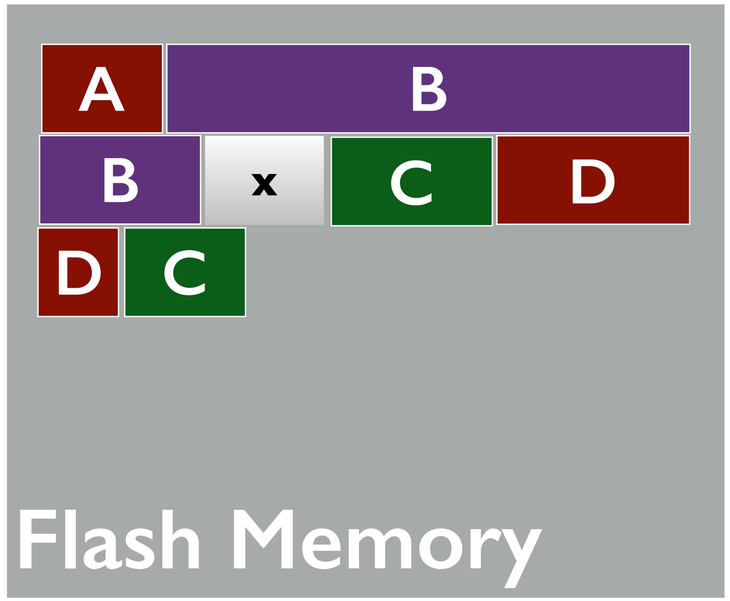
FTL

Nonvolatility Issues

Journaled Main Memory (built-in checkpoint)

➡ *Here's the way flash works:*





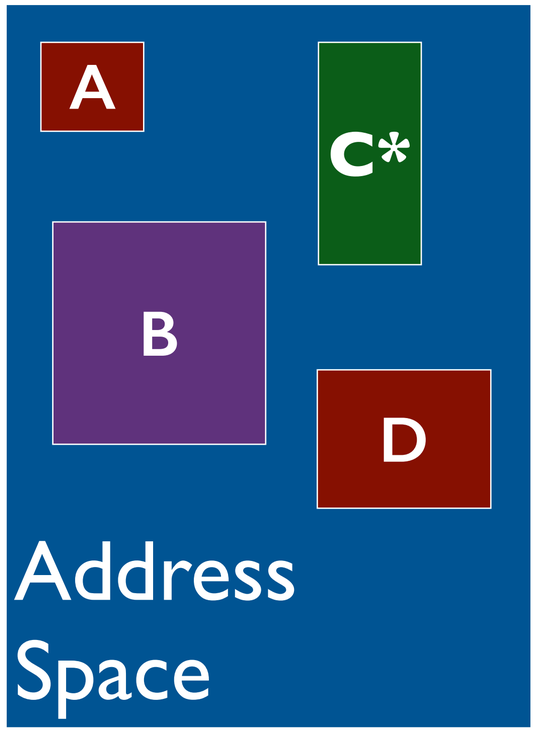
A: 0–2
B: 3–15,16–19
C: 20–26
D: 27–32,33–34

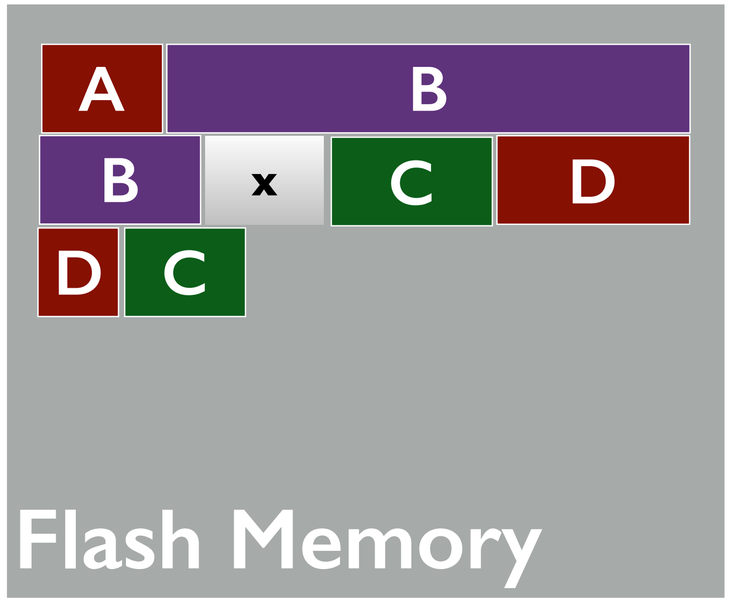
FTL

Nonvolatility Issues

Journaled Main Memory (built-in checkpoint)

➡ *Here's the way flash works:*





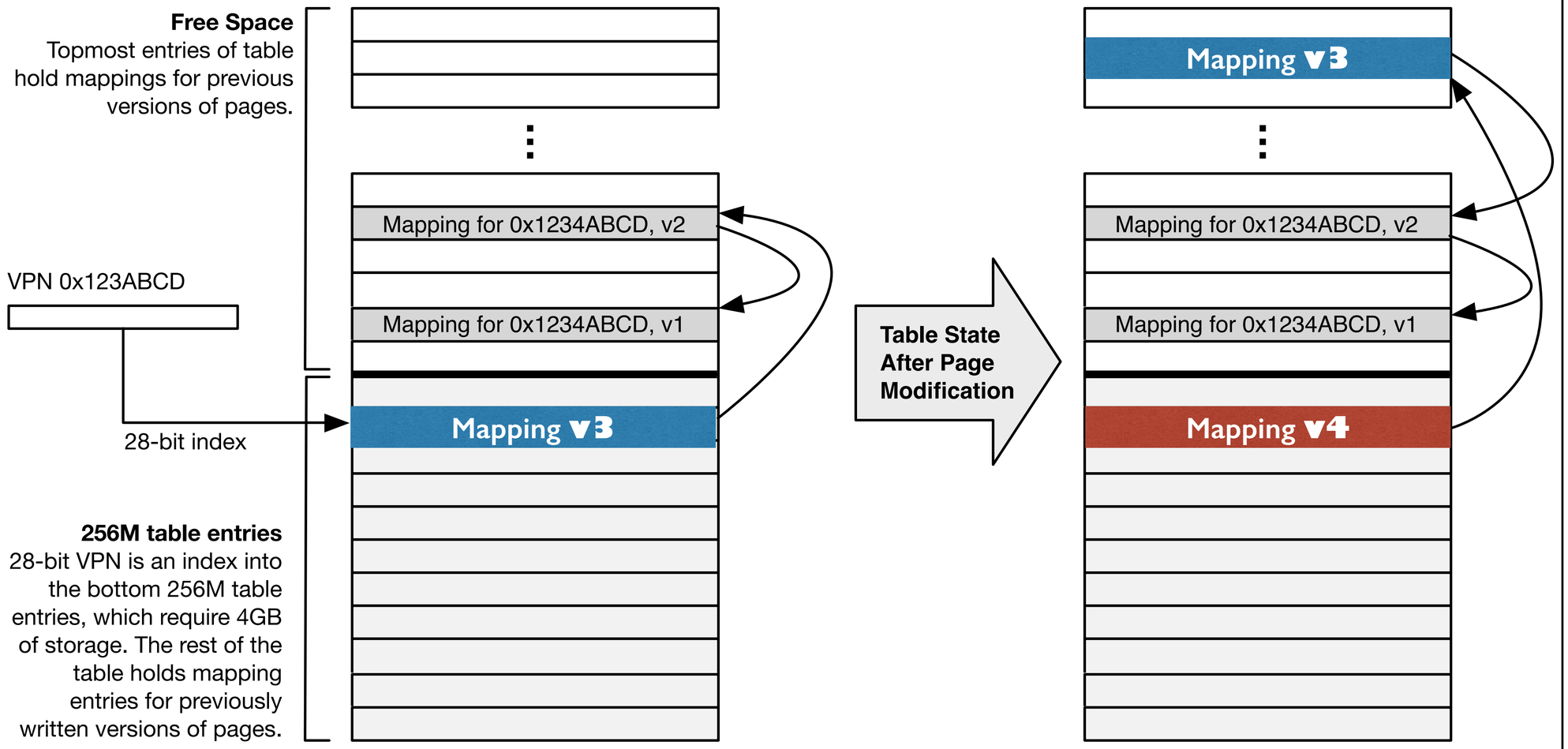
A: 0–2
B: 3–15,16–19
C: 35–37,23–26
D: 27–32,33–34

GC: 20–22

FTL

Nonvolatility Issues

Journaled Main Memory (built-in checkpoint)



Recap

Next-gen hardware (memory):

- Main memory: 10–100TB in 1-U server
  => support for really big data sets
  => BUT need LOTS of cores to drive it
- Power ~ today, cost: bandwidth
  (right now, BW does not come free)
- Performance w/ flash is acceptable;
  far less engineering is required w/ 3DXP

Bruce Jacob

University of
Maryland

Recap

Next-gen software (OS):

- Combined VM+FS subsystems
- Journaled main memory
- Persistent Object Store work from 80s
- No more "which is client" questions
- Simpler design, fewer potential bugs
- Built-in checkpoint/restart
- VM arguably a way better abstraction to distribute than the FS

Recap

Bottom Line (impact on SW):

- Great time for graph algorithms, data mining, deep learning
  => even distributed implementations
- Great time for novel approaches to application development (e.g., use of NVRAM, novel programming models, distributed/parallel programming via shared memory, etc.)
- Great time for systems research

Bruce Jacob

University of Maryland

Shameless Plug

www.memsys.io

Washington DC
October 2–5, 2017

Call For Papers www.memsys.io Call for Papers

MEMSYS Europe 2017

The International Symposium on Memory Systems ❖ 21–23 June 2017, Frankfurt am Main

Important Dates

Submission: 10 March*, 2017
Notification: 14 April, 2017
Camera-Ready: 28 April, 2017

Memory-device manufacturing, memory-architecture design, and the use of memory technologies by application software all profoundly impact today's and tomorrow's computing systems, in terms of their performance, function, reliability, predictability, power dissipation, and cost. Existing memory technologies are seen as limiting in terms of power, capacity, and bandwidth. Emerging memory technologies he potential to overcome both technology and design related tions to answer the requirements of many different applications. oal is to bring together researchers, practitioners, and others sted in this exciting and rapidly evolving field, to update each on the latest state-of-the-art, exchange ideas, and discuss future nges. *Visit memsys.io for more information.*

ference Schedule and Venue

naugural event will be held at the Mövenpick Hotel, with an ng reception & poster session Wednesday evening, followed by ll days of technical presentations on Thursday & Friday and an ls Banquet Thursday evening.

Bruce Jacob, U. Maryland
Kathy Smiley, Memory Systems

Eduard Ayguade, BSC and UPC
Luca Benini, U. Bologna/ETH Zürich
Angelos Bilas, FORTH
Damian Borth, DFKI
Koen De Bosschere, Ghent U.
Stephan Diestelhorst, ARM
David Donofrio, Berkeley Lab
Wendy Elsasser, ARM
Phil Emma, IBM
Paraskevas Evripidou, U. Cyprus
Babak Falsafi, EPFL
Paolo Faraboschi, Hewlett Packard
Dietmar Fey, U. Erlangen
Bastien Giraud, CEA Leti
Said Hamdioui, TU Delft
Ahmed Hemani, KTH Stockholm
Thuc Hoang, NNSA
Aamer Jaleel, NVIDIA
Toni Juan, Metempsy
Matthias Jung, U. Kaiserslautern
Thomas Kuhn, Fraunhofer IESE
Sally McKee, Chalmers
Thomas Mikolajick, U. Tech. Dresden
Onur Mutlu, ETH Zürich
Petar Radojkovic, BSC
Juri Schmidt, U. Heidelberg
Christian Schulze, DFKI
Georgios Sirakoulis, U. Thrace
Per Stenström, Chalmers
Ronald Tetzlaff, U. Tech. Dresden
Pedro Trancoso, U. Cyprus
Norbert Wehn, U. Kaiserslautern
Christian Weis, U. Kaiserslautern
Kenneth Wright, Rambus

Tracks and Topics

Tracks on the following topics are being organized and will be presented over the 2-day conference:

- Memory-centric programming models, programming languages, and compiler optimization
- Difficulties integrating different memory types into the software stack
- Memristors, other nonvolatile memories, and compute-in-memory technologies
- Emerging memory technologies, their controllers, and novel uses
- Memory systems, IP, SoC, controllers in automotive applications
- Interference at the memory level across datacenter applications
- Issues in the design and operation of large-memory machines
- In-memory databases and NoSQL stores
- Memory limitations in AI/ML applications and architectures
- Post-CMOS scaling efforts and memory technologies to support them, including cryogenic, neural, and heterogeneous memories

This CFP seeks papers and talks on these and other related topics.

Submissions and Presentations

Our primary goal is to showcase interesting ideas that will spark conversation between disparate groups—to get applications people, operating systems people, system architecture people, interconnect people and circuits people to talk to each other. We accept extended abstracts, position papers, and/or full research papers, and each accepted submission is given a 20-minute presentation time slot. **All accepted papers will be published in the ACM Digital Library.**

Bruce Jacob

University of Maryland

Thank You!

Bruce Jacob

blj@umd.edu
www.ece.umd.edu/~blj

Bruce Jacob

University of
Maryland

Backup Slides

HMC Die
1Gb

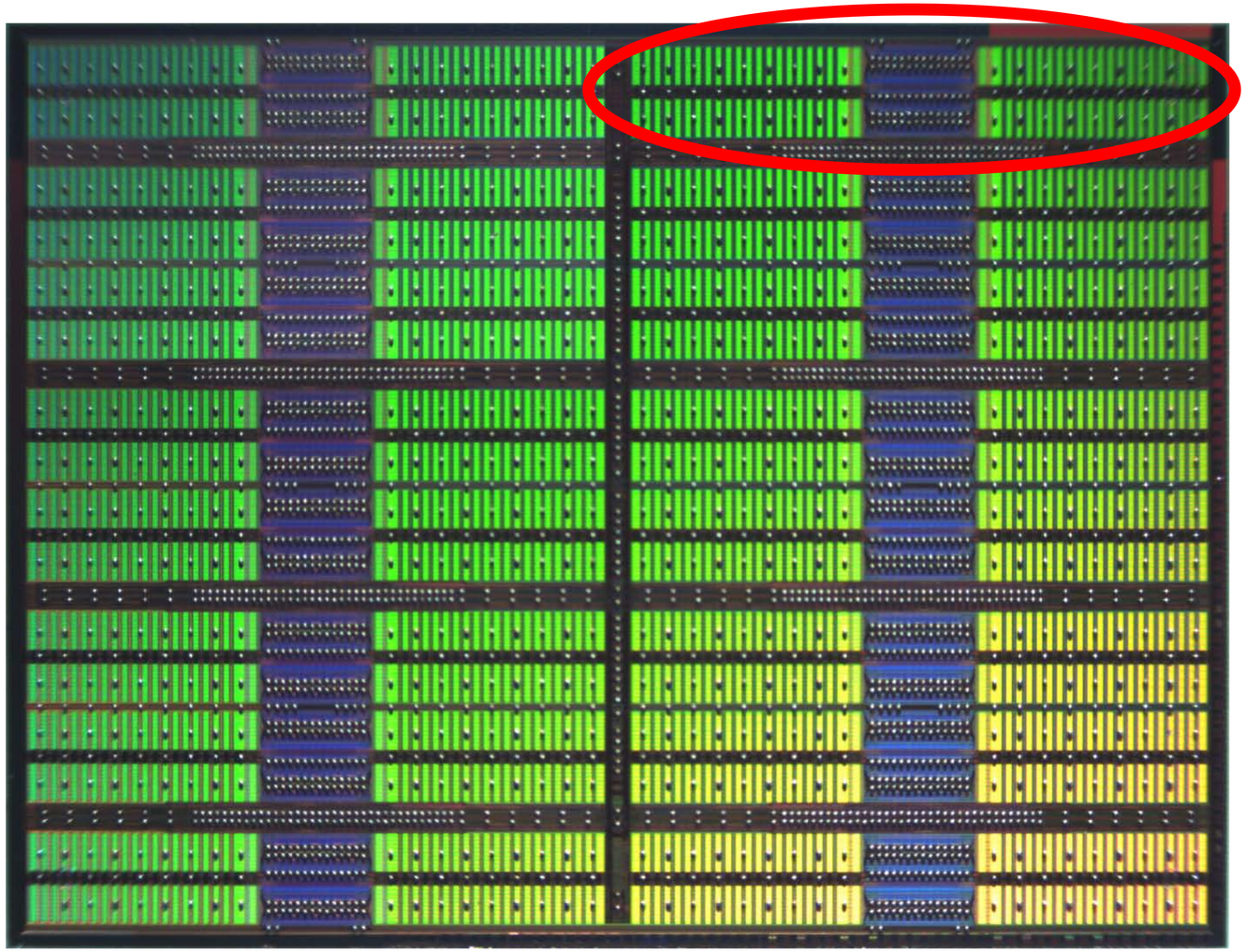
Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is **160** GB/s per (640 total)

Total conc'y = **16 x 8 x 2..8** (256–1024)

Partition, with internal banks



Source: Micron

Bruce Jacob

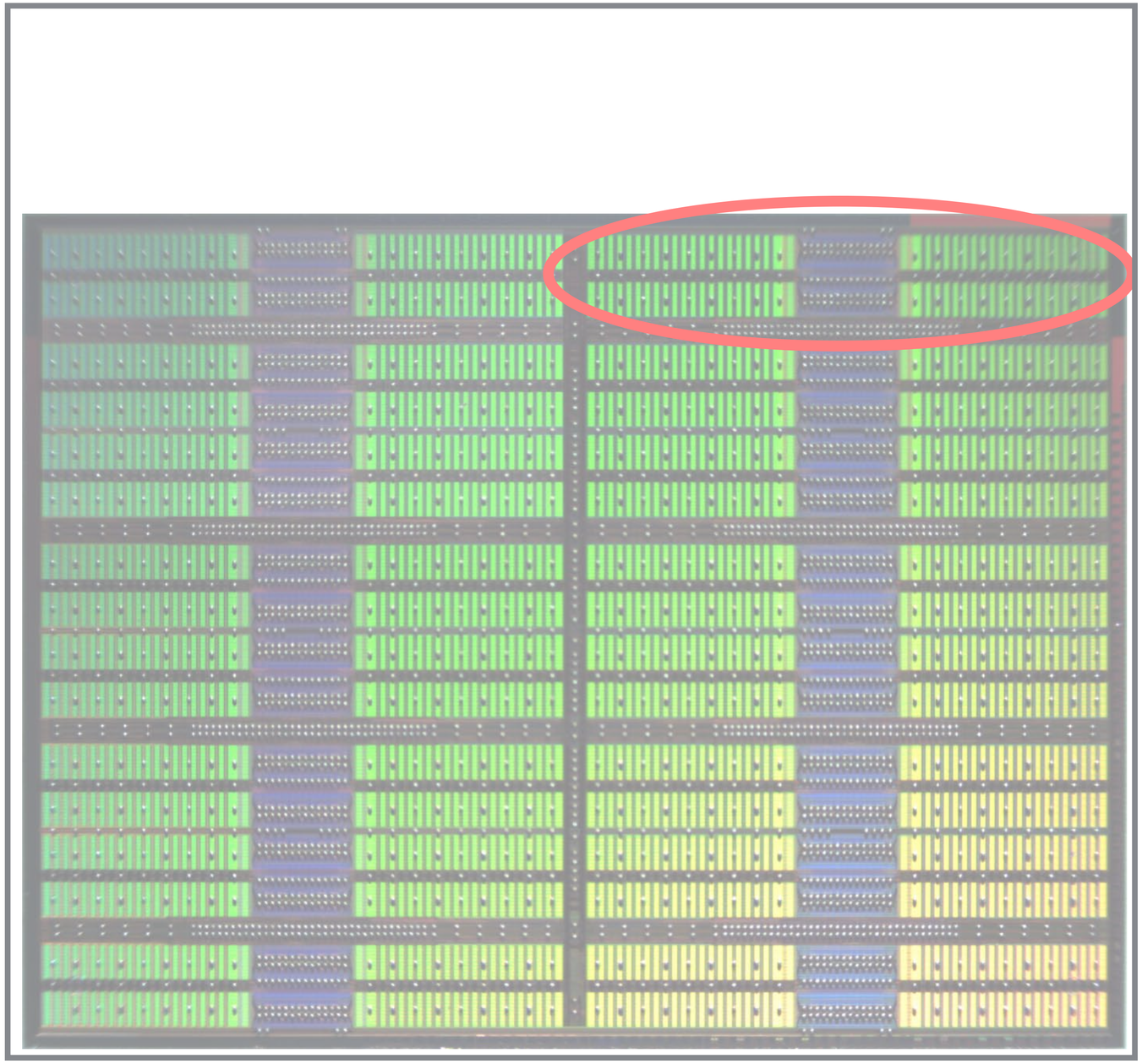
Logic Die

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is **160** GB/s per (640 total)

Total conc'y = **16 x 8 x 2..8** (256–1024)



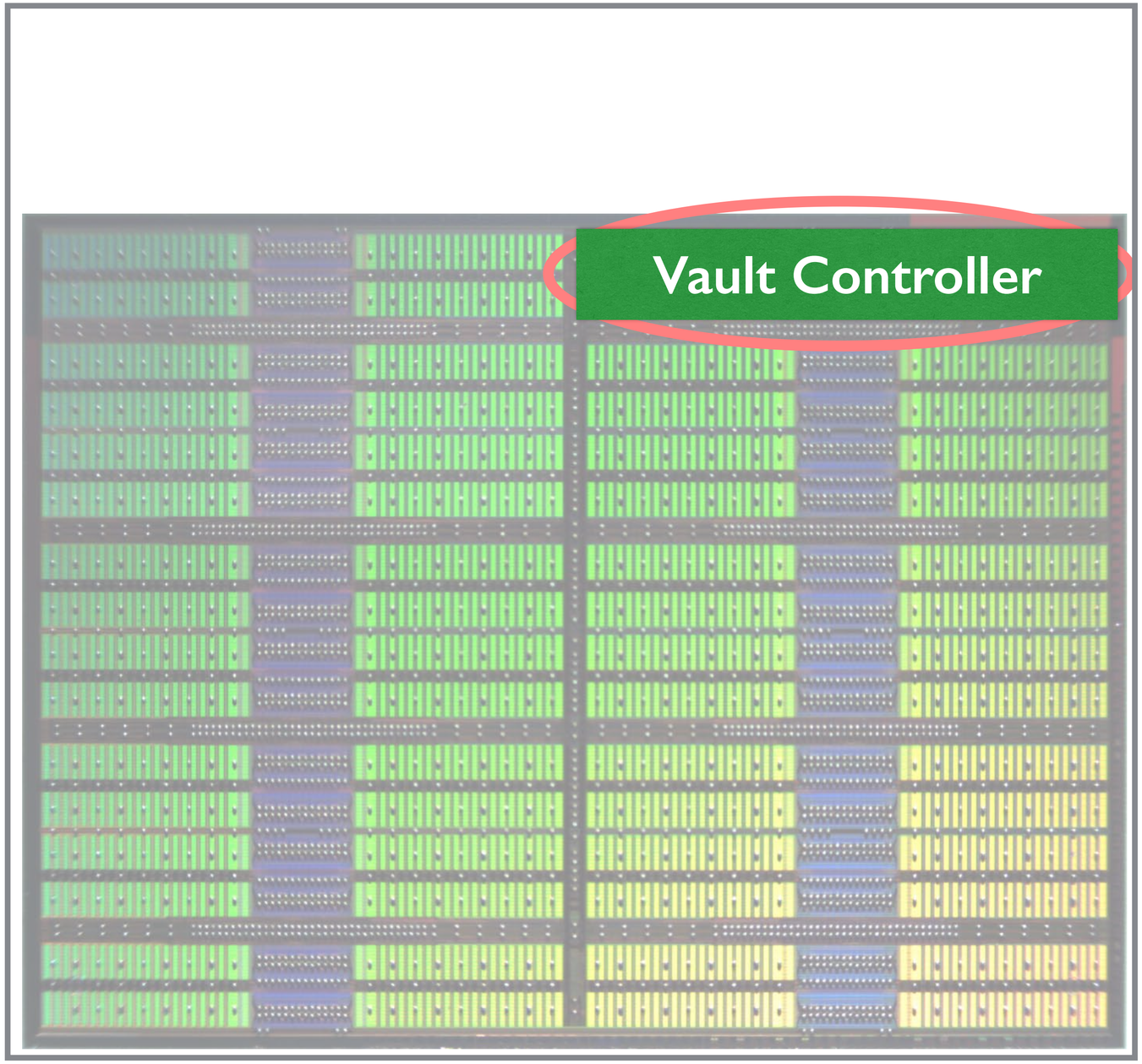
Logic Die

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is **160** GB/s per (640 total)

Total conc'y = **16 x 8 x 2..8** (256–1024)



Bruce Jacob

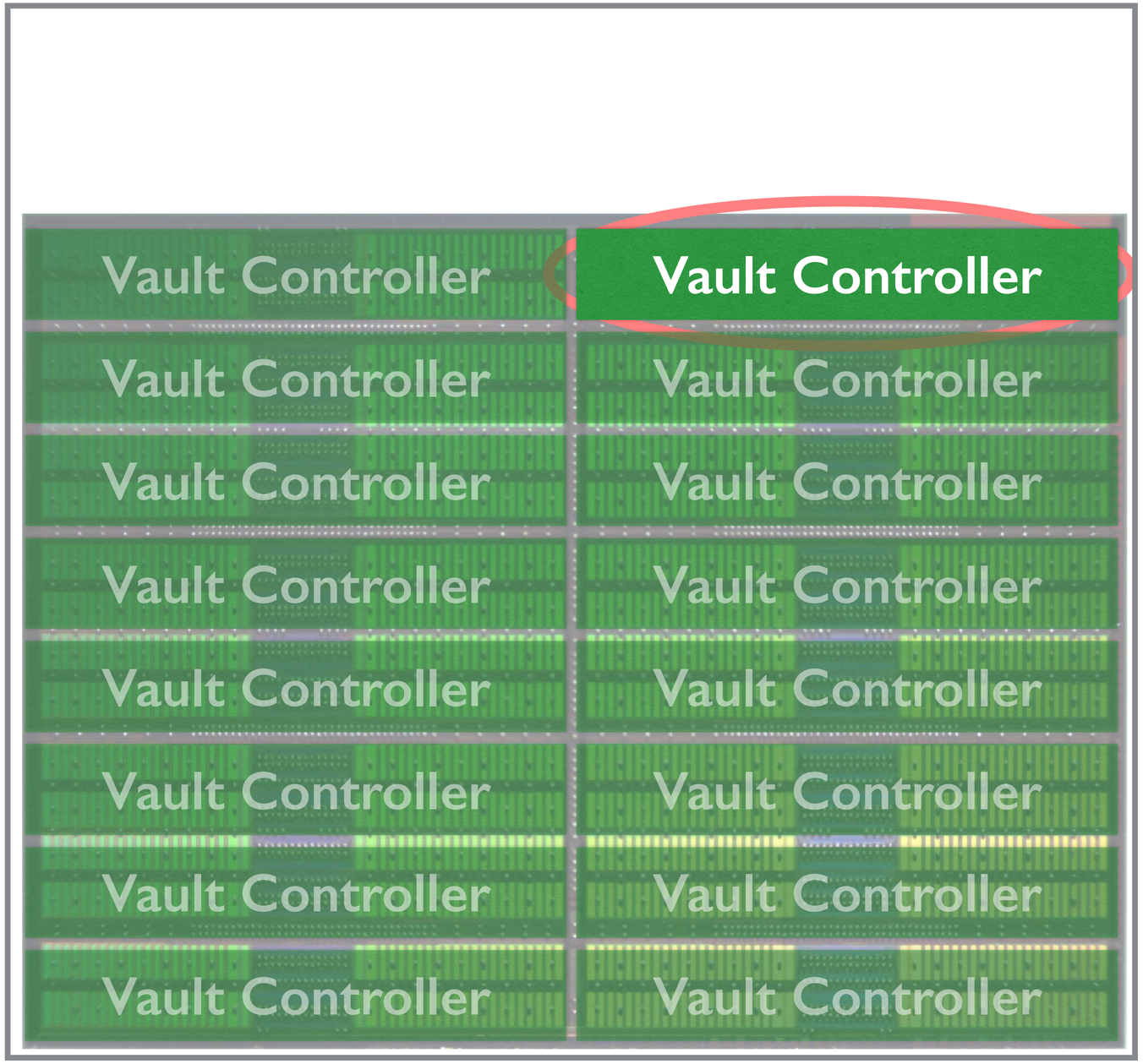
Logic Die

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc'y = 16 x 8 x 2..8 (256–1024)



Bruce Jacob

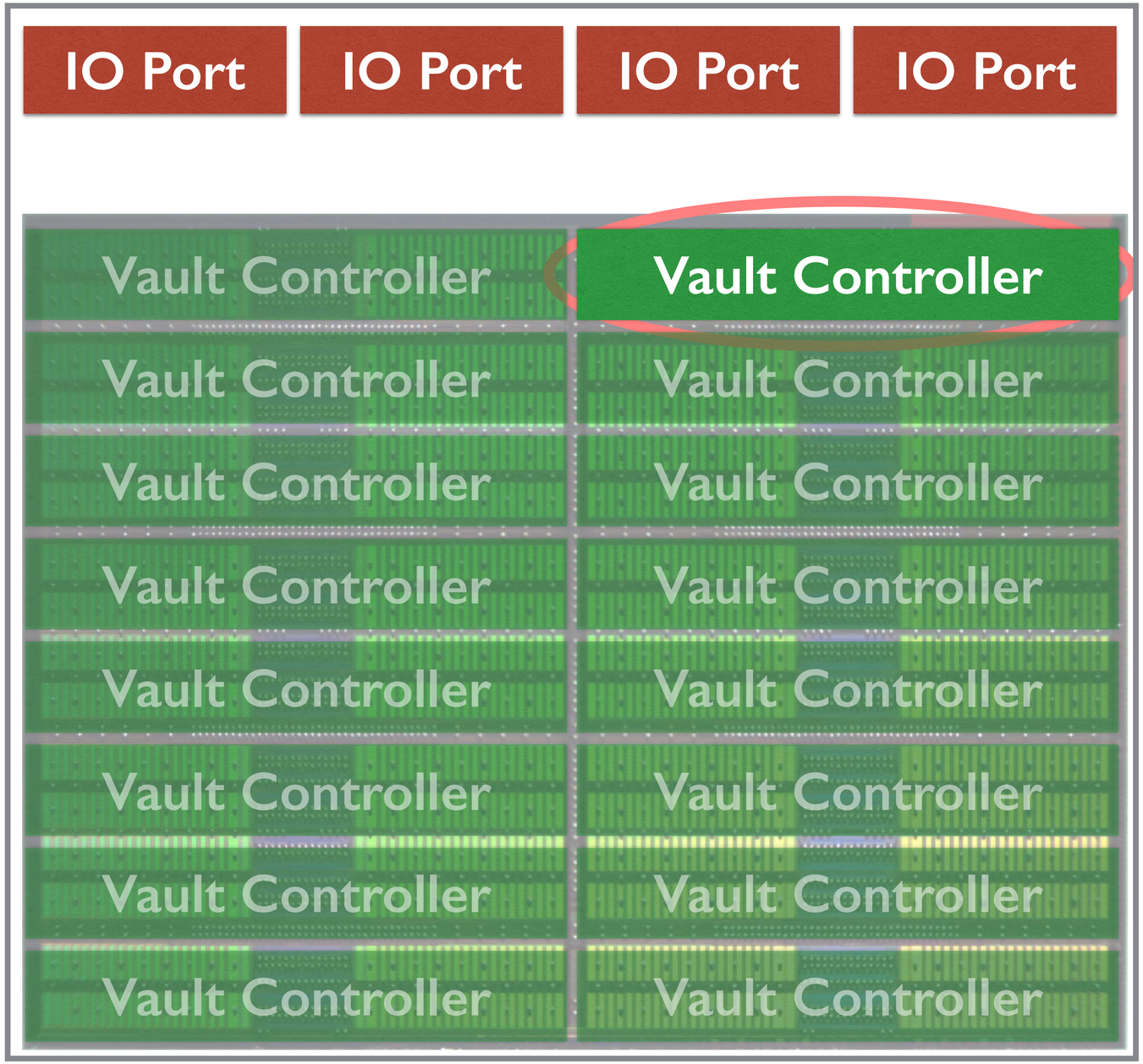
Logic Die

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is **160** GB/s per (640 total)

Total conc'y = **16 x 8 x 2..8** (256–1024)



Logic Die

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is **160** GB/s per (640 total)

Total conc'y = **16 x 8 x 2..8** (256–1024)