ISC'10

New Memory & Storage Hierarchies for HPC – Opportunities & Challenges

Memory Systems Then, Now, and To Come

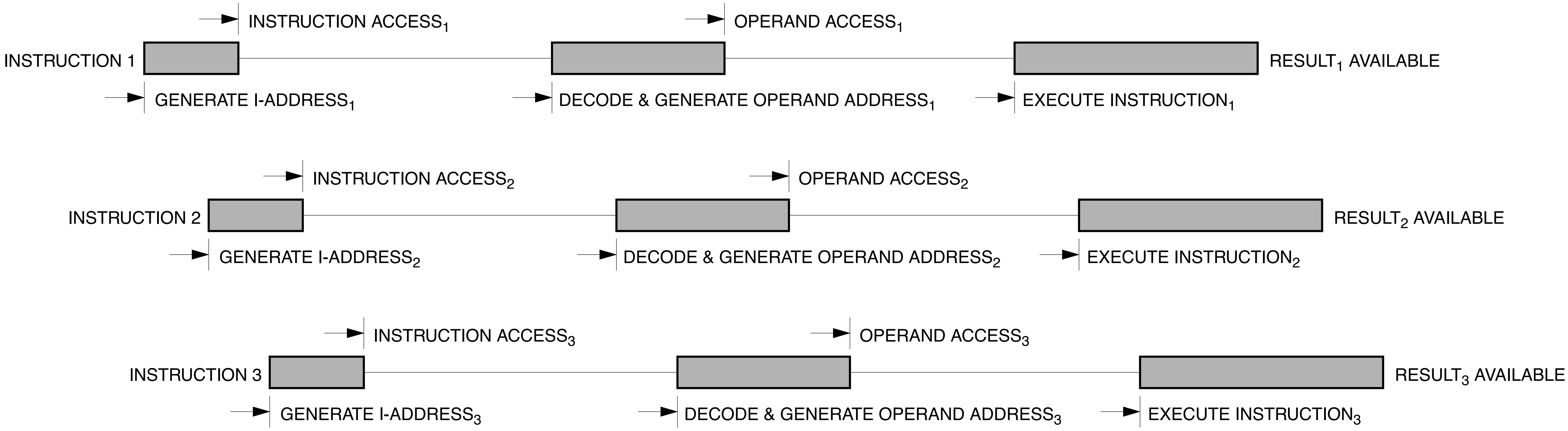
Prof.Dr. Bruce Jacob

Keystone Professor & Director of Computer Engineering Program

Electrical & Computer Engineering

University of Maryland at College Park

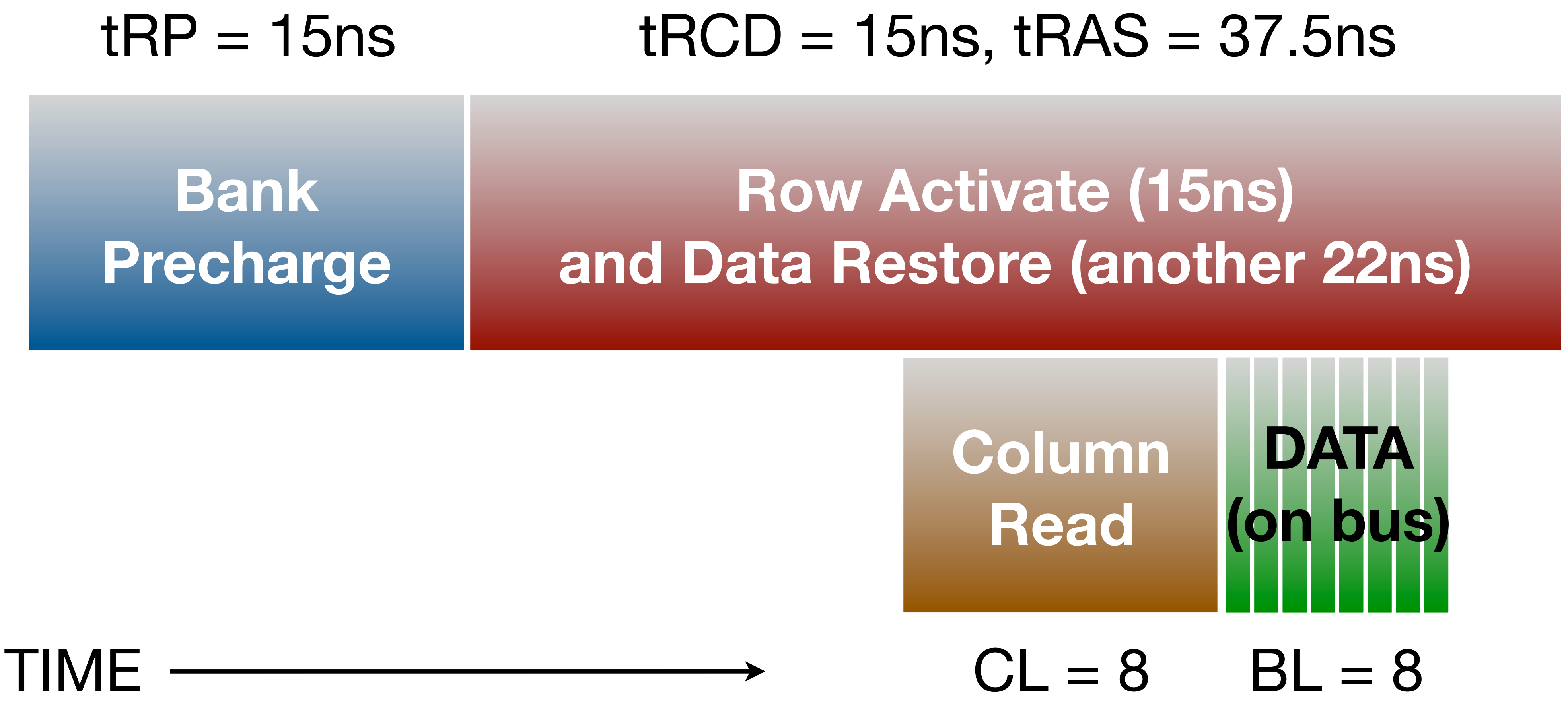
THEN



IBM 360/91 Fixed-Point Pipe

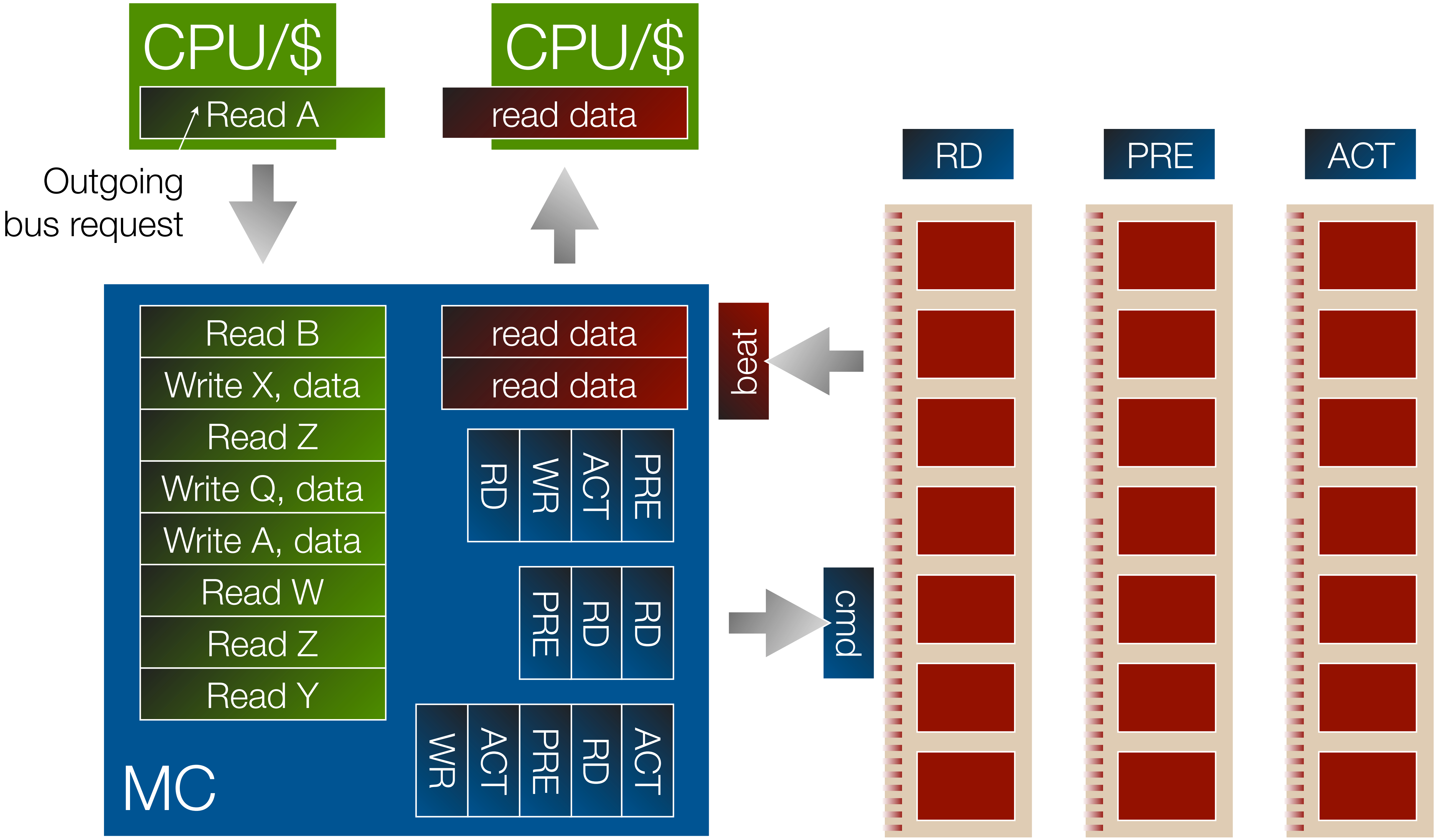
NOW

DRAM Read Timing


tRP = 15ns
tRCD = 15ns, tRAS = 37.5ns
Bank Precharge
Row Activate (15ns) and Data Restore (another 22ns)
Column Read
DATA (on bus)
TIME
CL = 8
BL = 8

Cost of access is high; requires **significant effort** to amortize this over the (increasingly short) payoff.

“Significant Effort” [deep pipes, reordering]

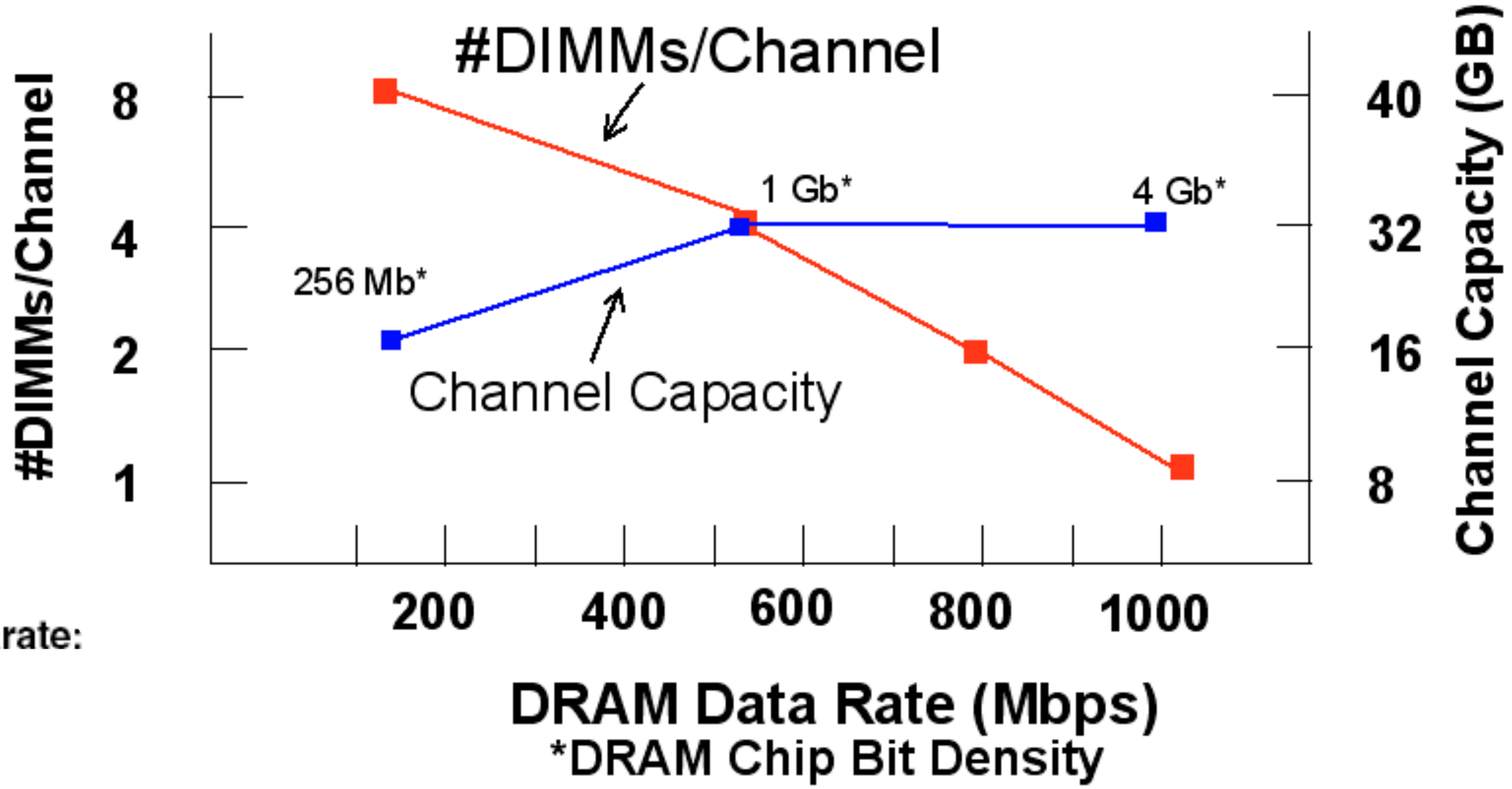


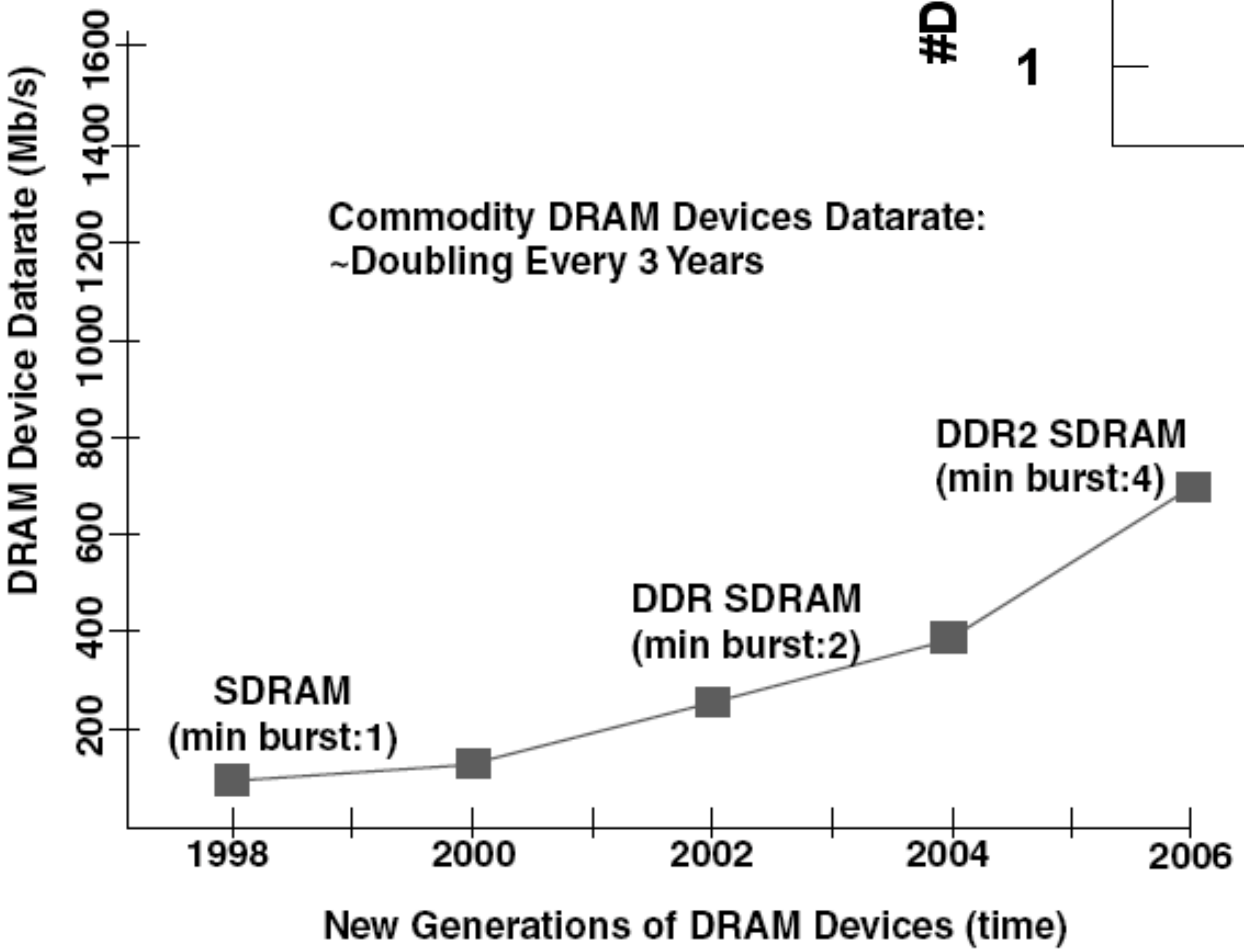
Consequence: Due to buffering & reordering at multiple levels, the **average** latency is typically much higher than the **minimum** latency

TO COME

Move from **concurrency via pipelining**
to **concurrency via parallelism**
(mirrors recent developments in CPU design)

Problem: Capacity



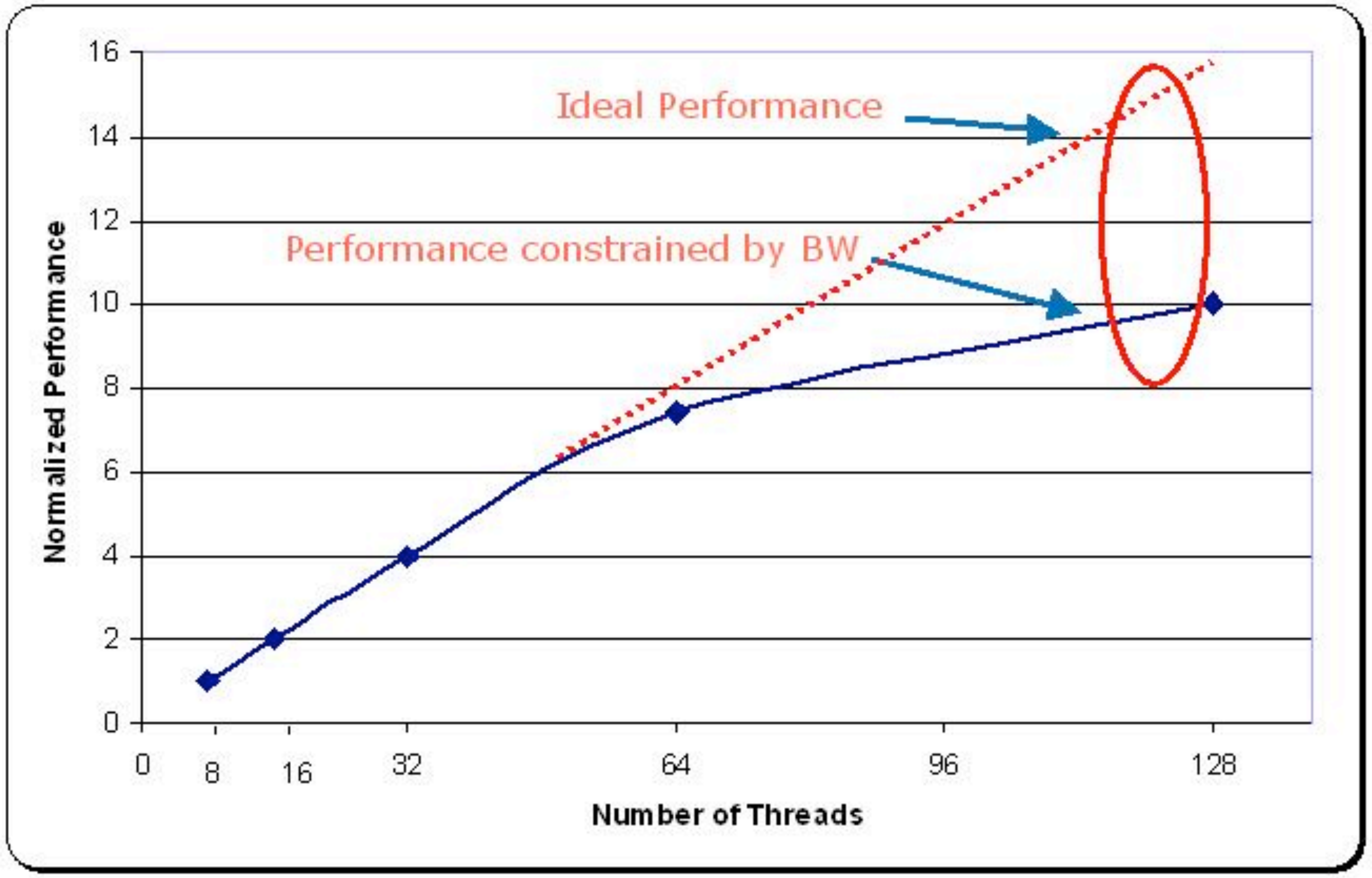


Problem: Bandwidth

- Like capacity, primarily a power and heat issue: can get more BW by adding busses, but they need to be narrow & thus fast. Fast = hot.

- Required BW per core is roughly 1 GB/s, and cores per chip is increasing

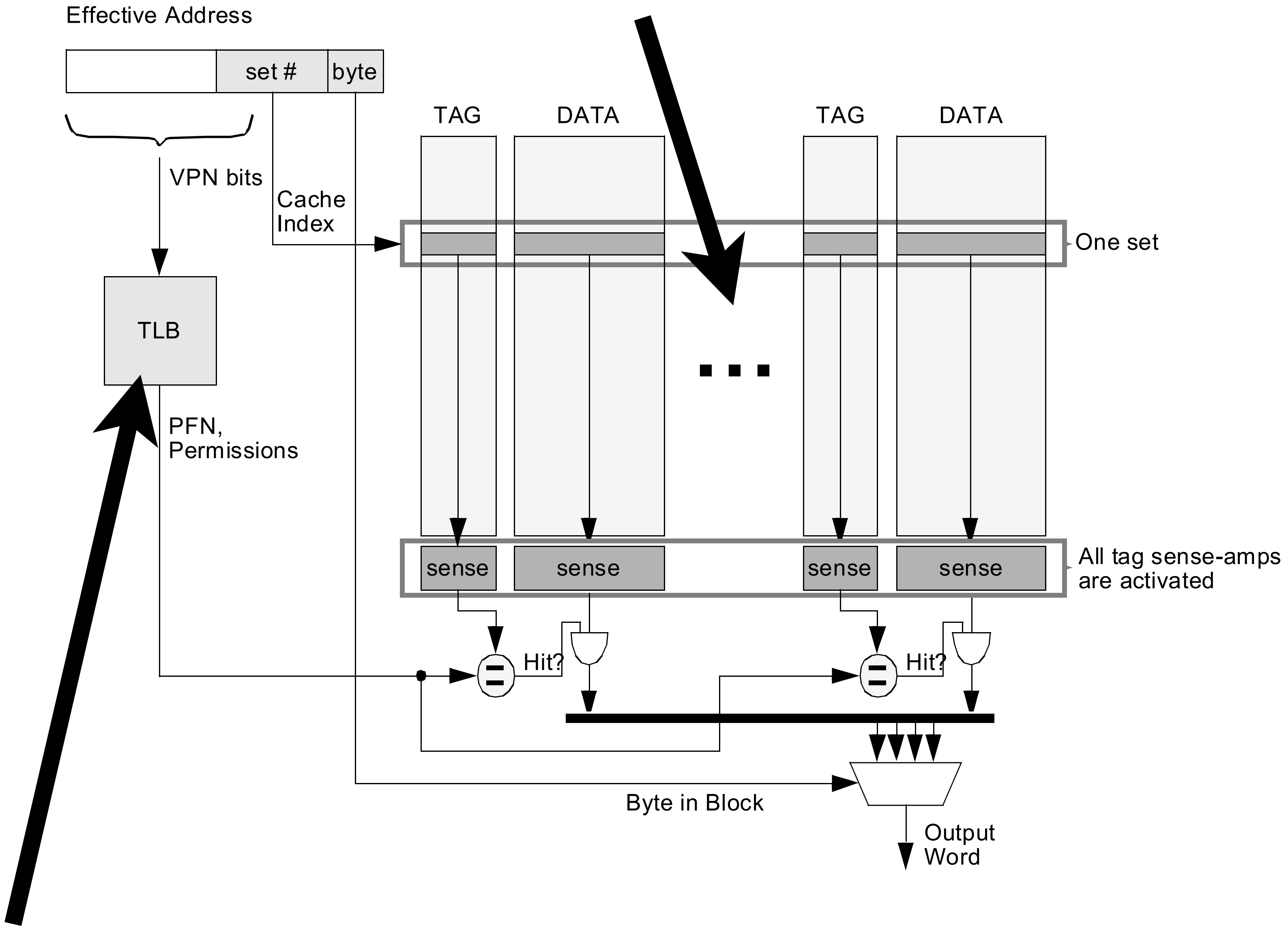
- Graph: Thread-based load (SPECjbb), memory set to 52GB/s sustained
  … cf. 32-core Sun Niagara: saturates at 25.6 GB/s



Problem: TLB Reach

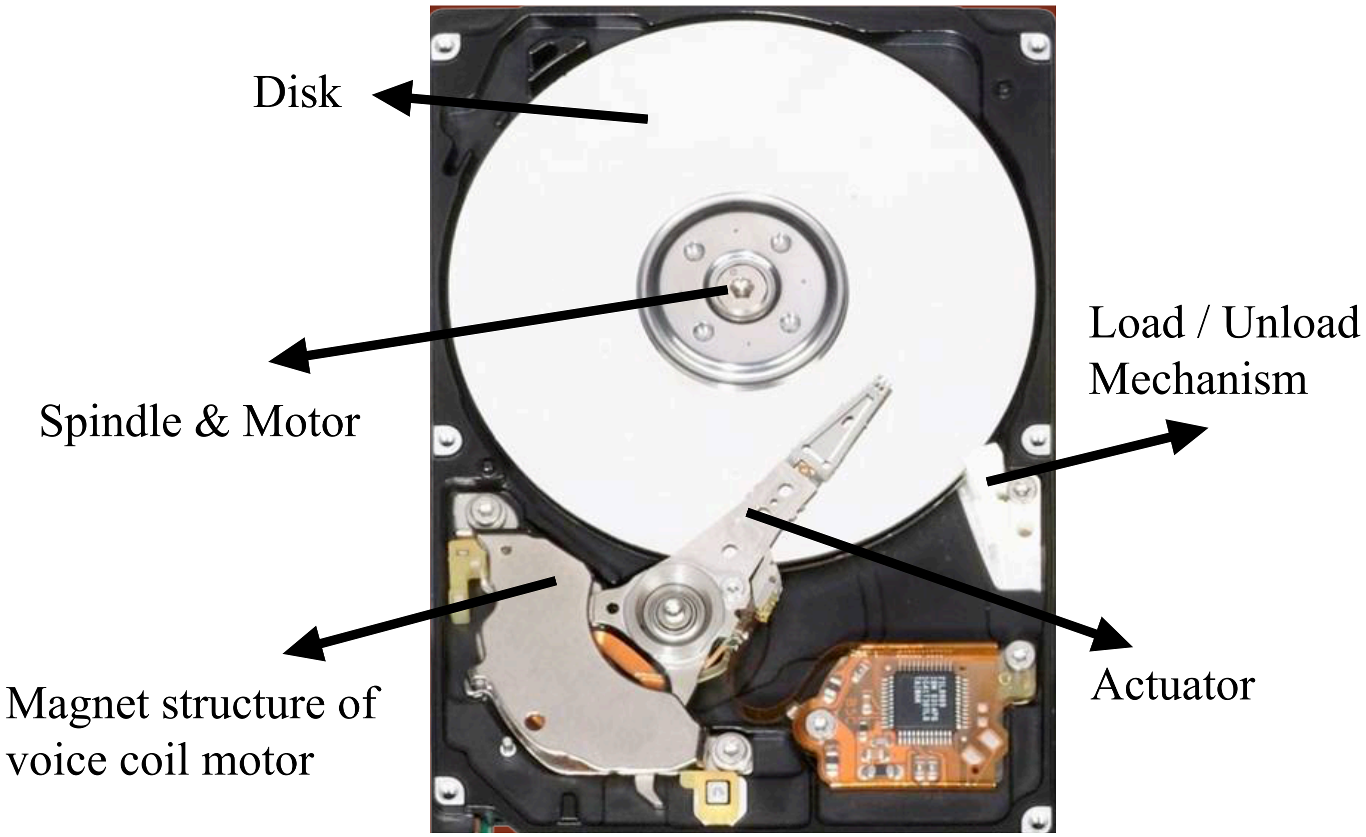
- Doesn’t scale at all (still small and not upgradeable)
- Currently accounts for **20+%** of system overhead
- Higher associativity (which offsets the TLB’s small size) can create a power issue
- The TLB’s “reach” is actually much worse than it looks, because of different access granularities

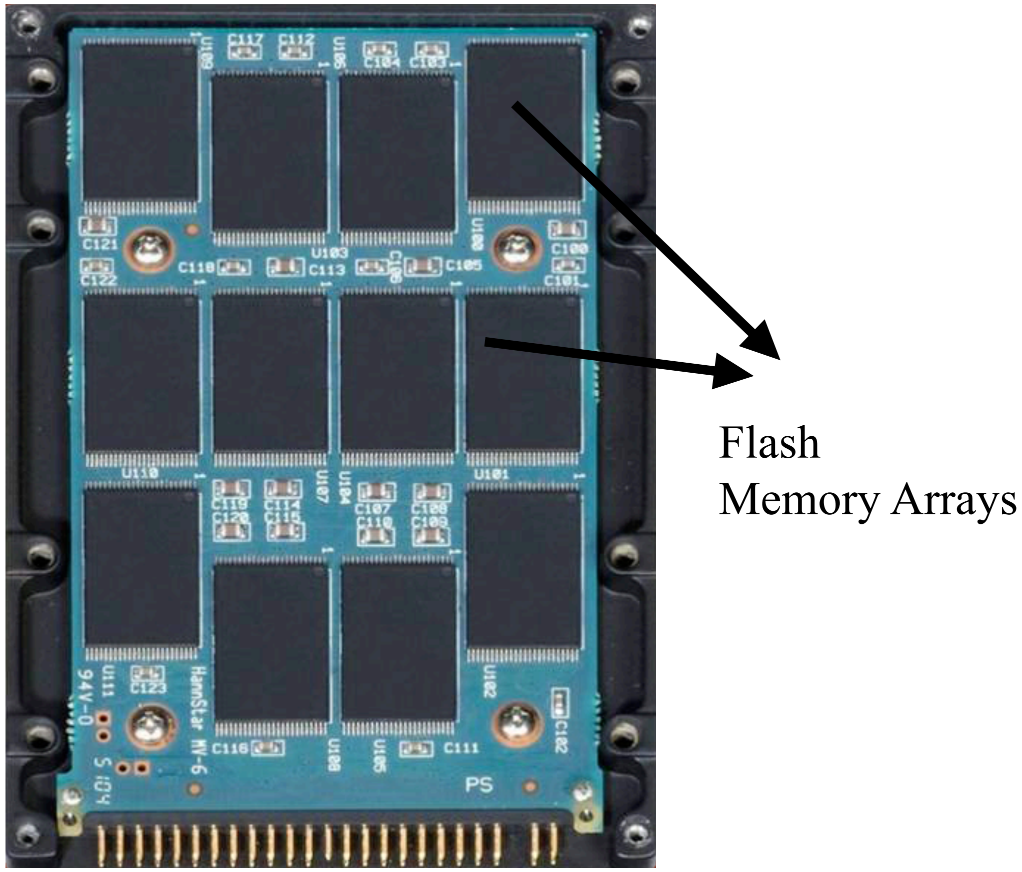
~10MB



Maps ~1MB

Trend: Disk, Flash, and other NV

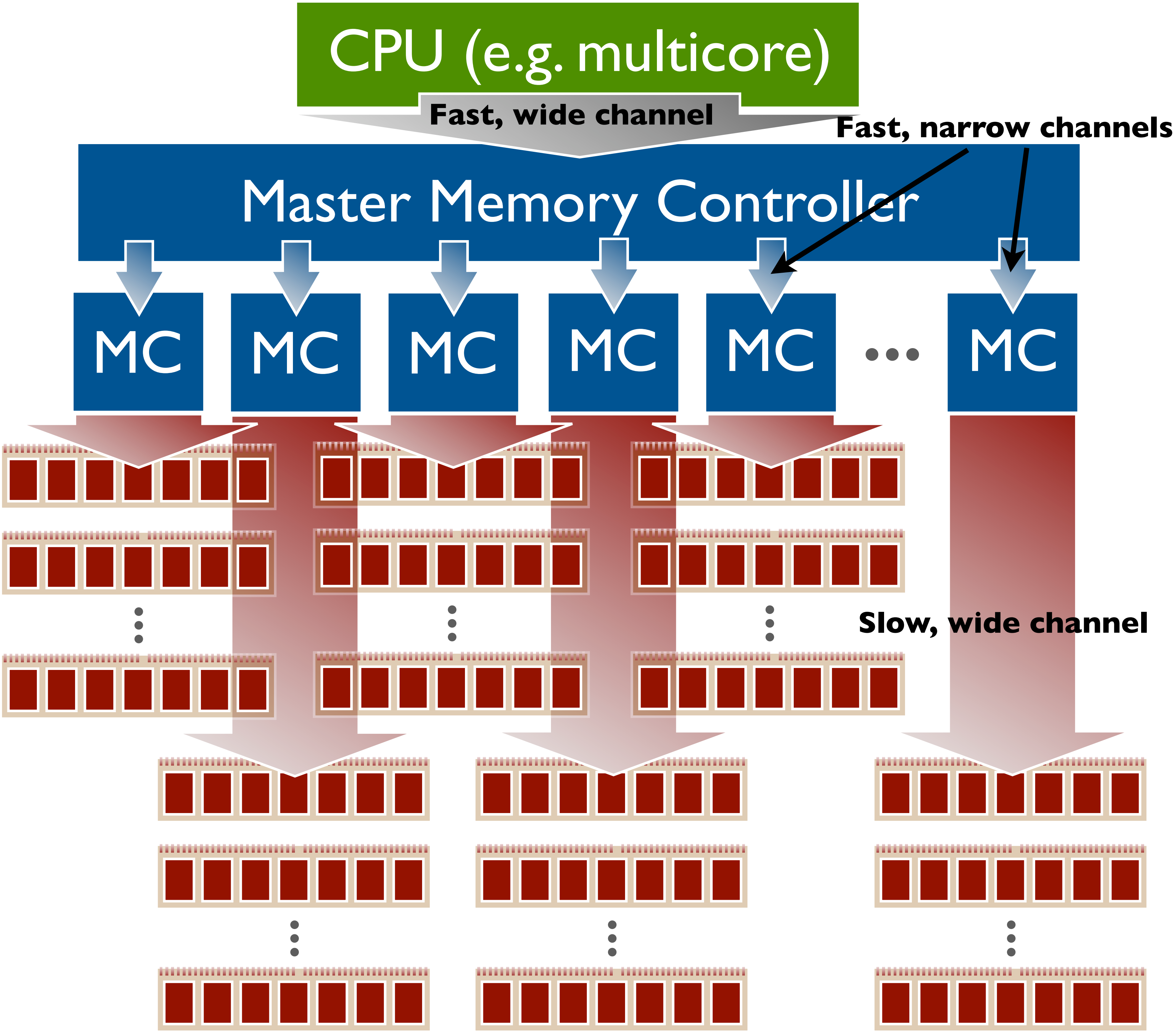




- Flash is currently eating Disk’s lunch
- PCM is expected to eat Flash’s lunch

Obvious Conclusions I

- Want capacity without sacrificing bandwidth
- **Need a new memory system architecture**
- This is coming (details will change, of course)



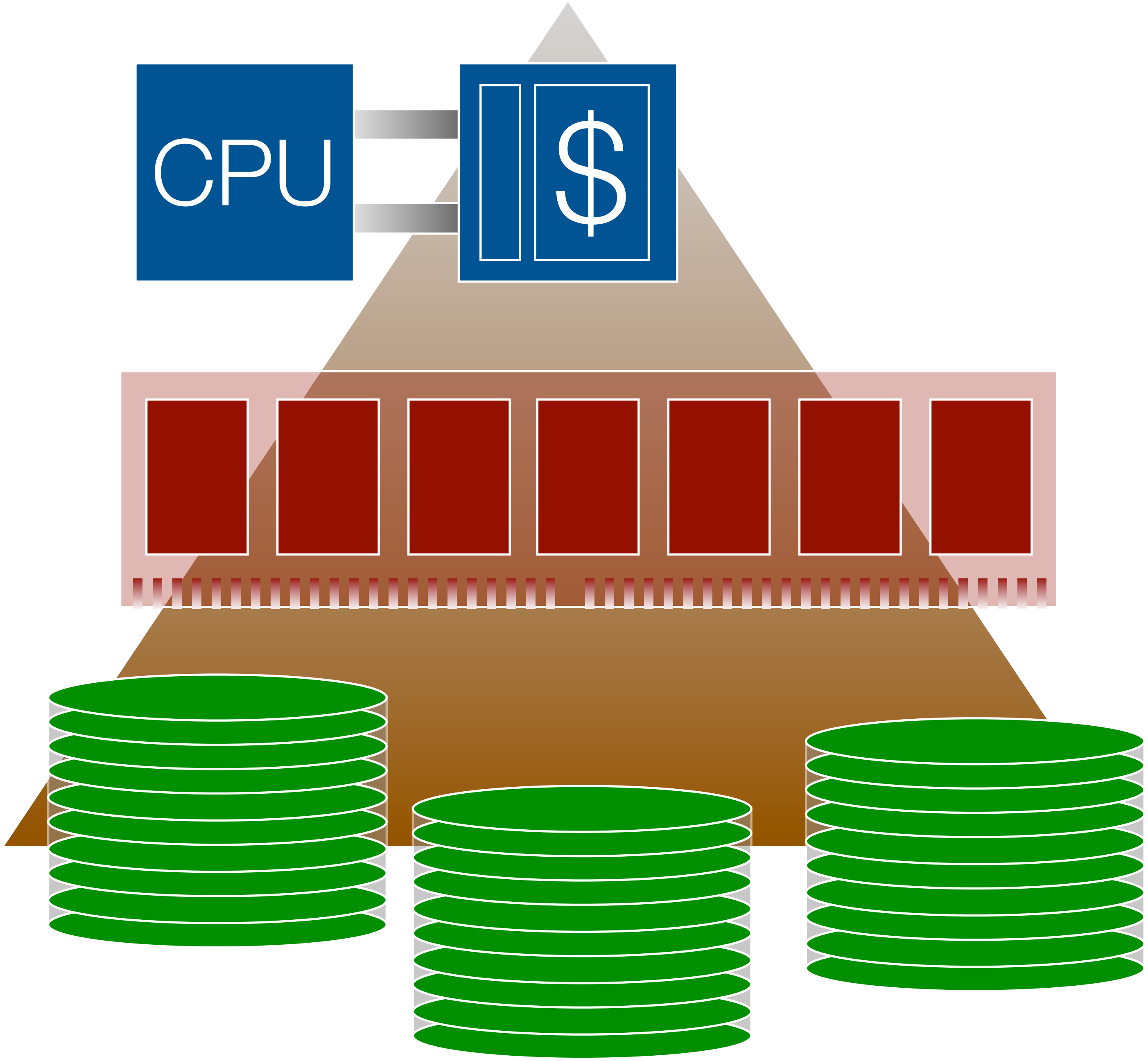
Obvious Conclusions II

- Flash/NV is inexpensive, is fast (rel. to disk), and has better capacity roadmap than DRAM

- **Make it a first-class citizen in the memory hierarchy**

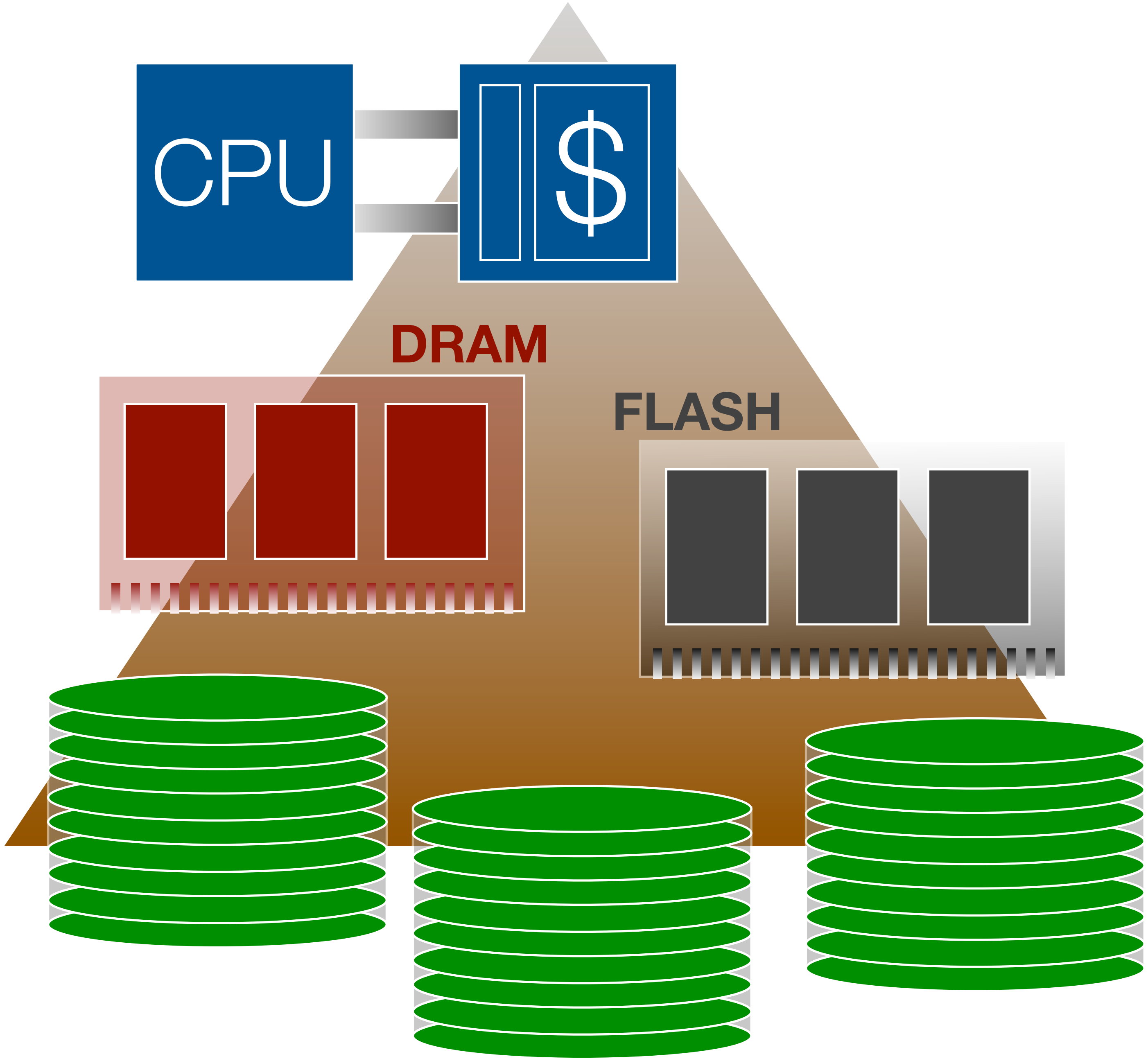
- Access it via load/store interface, use DRAM to buffer writes, software management

- Probably reduces capacity pressure on DRAM system



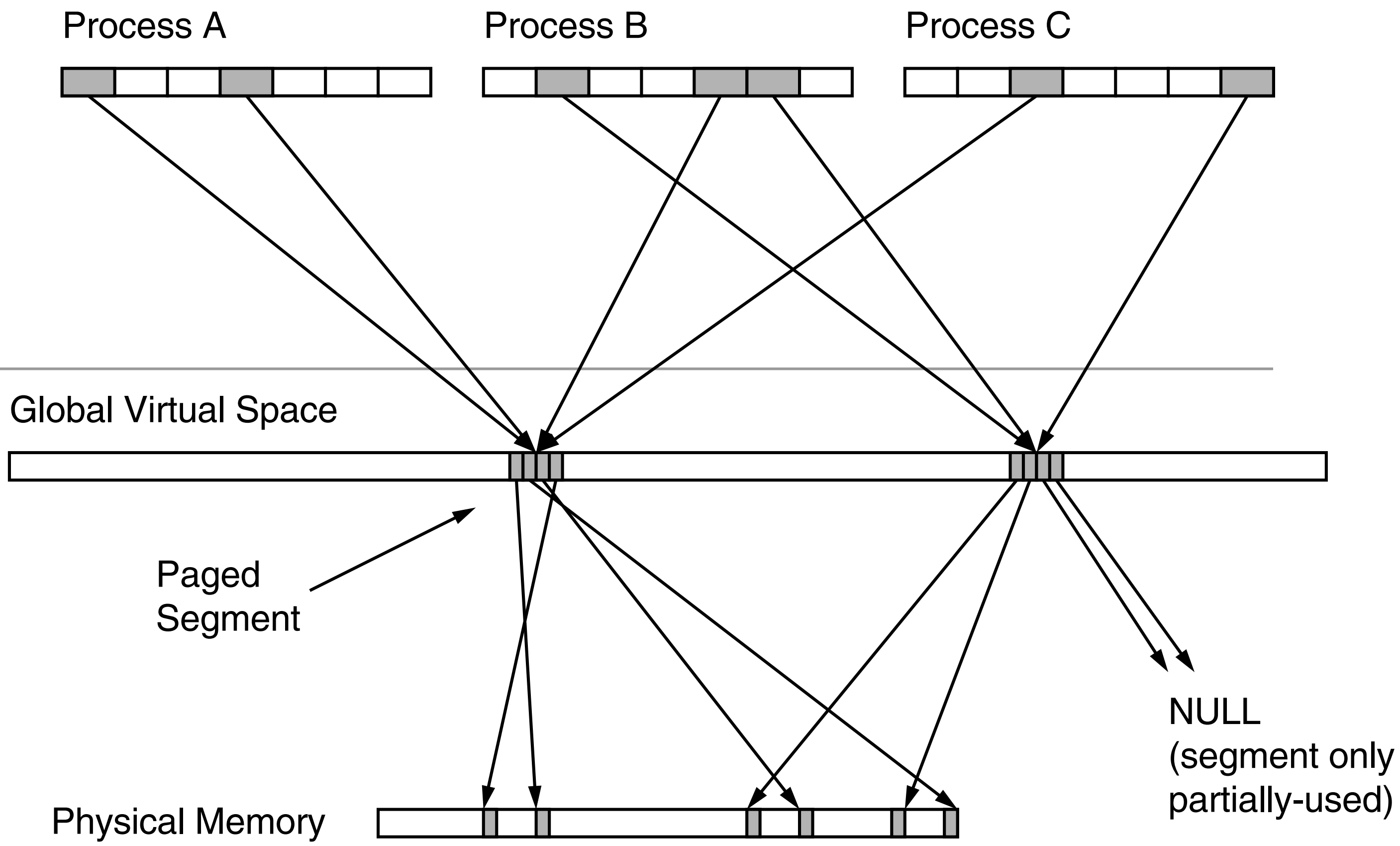
Obvious Conclusions II

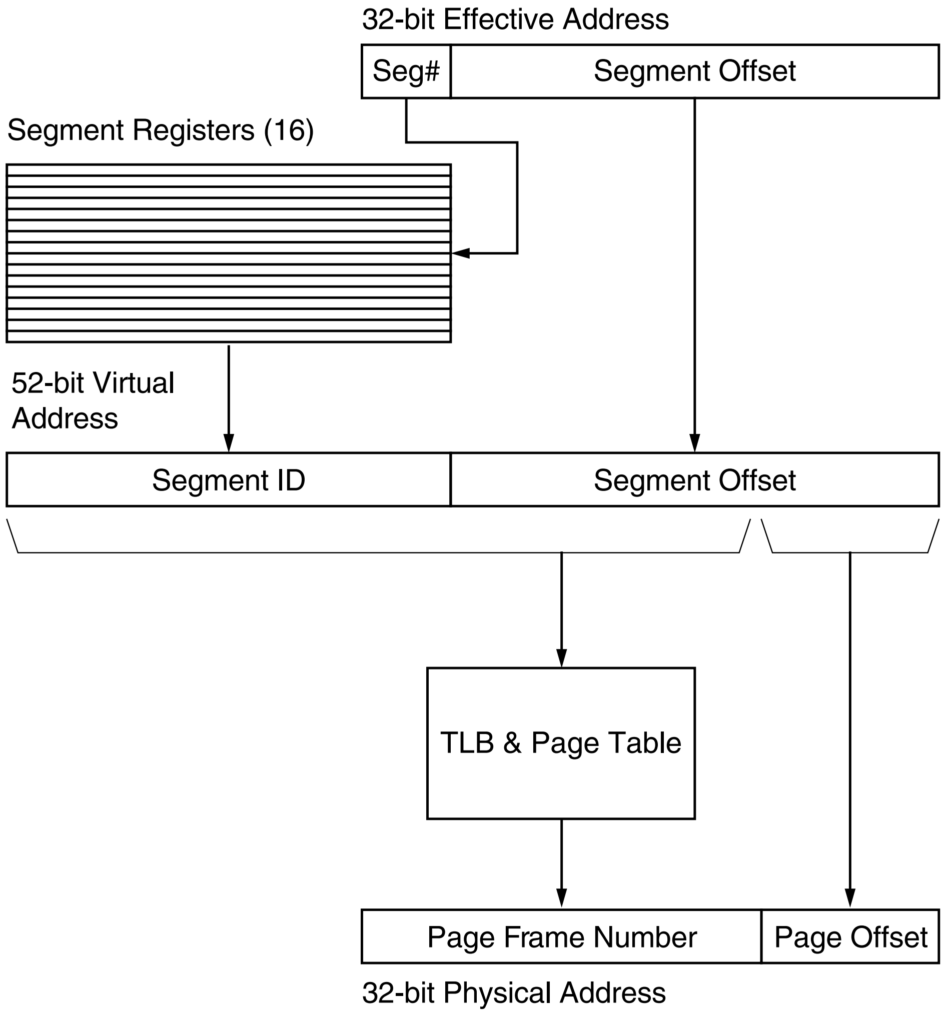
- Flash/NV is inexpensive, is fast (rel. to disk), and has better capacity roadmap than DRAM
- **Make it a first-class citizen in the memory hierarchy**
- Access it via load/store interface, use DRAM to buffer writes, software management
- Probably reduces capacity pressure on DRAM system



Obvious Conclusions III

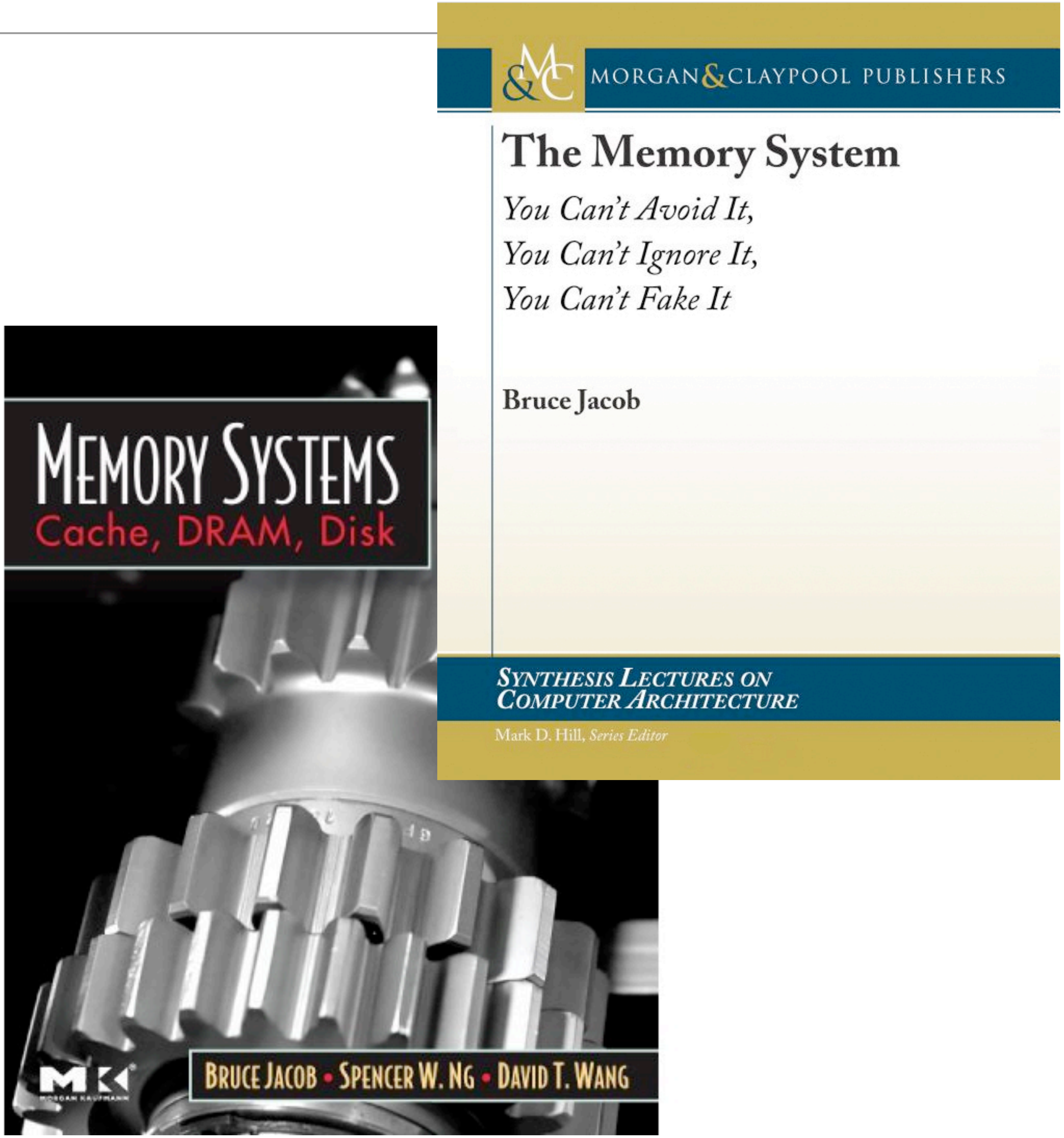
- Reduce translation overhead (both in performance & power)
- **Need an OS/arch redesign**
- Revisit superpages, multi-level TLBs
- Revisit SASOS concepts, *location of translation point/s* (i.e., **PGAS**)
- Arguably a good programming model for CMP





Acknowledgements & Shameless Plugs

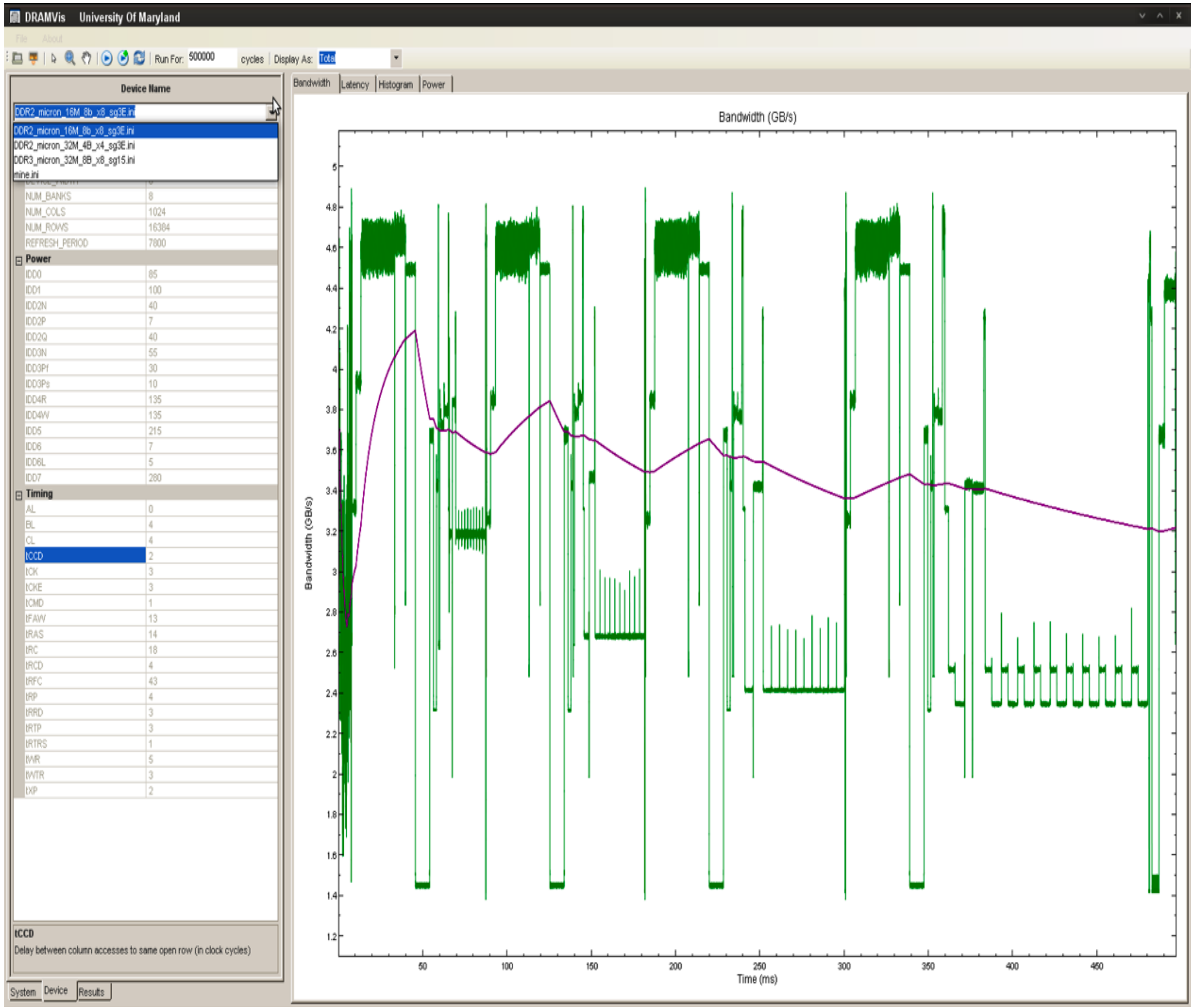
- Much of this has appeared previously in our books, papers, etc.
  - The Memory System (You Can't Avoid It; You Can't Ignore It; You Can't Fake It). B. Jacob, with contributions by S. Srinivasan and D. T. Wang. ISBN 978-1598295870. Morgan & Claypool Publishers: San Rafael CA, 2009.
  - Memory Systems: Cache, DRAM, Disk. B. Jacob, S. Ng, and D. Wang, with contributions by S. Rodriguez. ISBN 978-0123797513. Morgan Kaufmann: San Francisco CA, 2007.
- Support from Intel, DoD, DOE, Sandia National Lab, Micron, Cypress Semiconductor


MORGAN & CLAYPOOL PUBLISHERS
The Memory System
You Can't Avoid It,
You Can't Ignore It,
You Can't Fake It
Bruce Jacob
SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE
Mark D. Hill, Series Editor
MEMORY SYSTEMS
Cache, DRAM, Disk
MK MORGAN KAUFMANN
BRUCE JACOB • SPENCER W. NG • DAVID T. WANG

Acknowledgements & Shameless Plugs

- **DRAMsim** — the world's most accurate (hardware-validated) DRAM-system simulator:
    - "DRAMsim: A memory-system simulator." D. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel, and B. Jacob. *SIGARCH Computer Architecture News*, vol. 33, no. 4, pp. 100–107. September 2005.
- Version II now available at

  www.ece.umd.edu/dramsim



Run For: 500000 cycles | Display As: Total

Device Name: DDR2_micron_16M_8b_x8_sg3E.ini

DDR2_micron_16M_8b_x8_sg3E.ini
DDR2_micron_32M_4B_x4_sg3E.ini
DDR3_micron_32M_8B_x8_sg15.ini
mine.ini

NUM_BANKS	8
NUM_COLS	1024
NUM_ROWS	16384
REFRESH_PERIOD	7800
Power	
IDD0	85
IDD1	100
IDD2N	40
IDD2P	7
IDD2Q	40
IDD3N	55
IDD3Pf	30
IDD3Ps	10
IDD4R	135
IDD4W	135
IDD5	215
IDD6	7
IDD6L	5
IDD7	280
Timing	
AL	0
BL	4
CL	4
tCCD	2
tCK	3
tCKE	3
tCMD	1
tFAW	13
tRAS	14
tRC	18
tRCD	4
tRFC	43
tRP	4
tRRD	3
tRTP	3
tRTRS	1
tWR	5
tWTR	3
tXP	2

tCCD
Delay between column accesses to same open row (in clock cycles)

ETC.

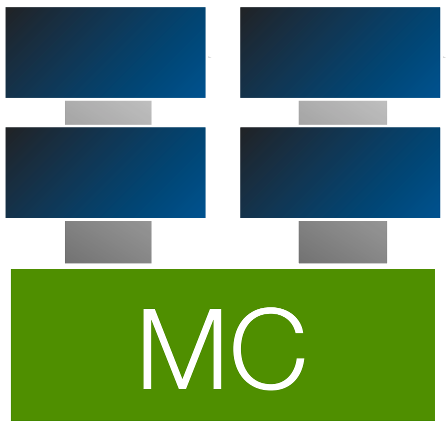
Problem: We don’t understand it very well

How it is represented

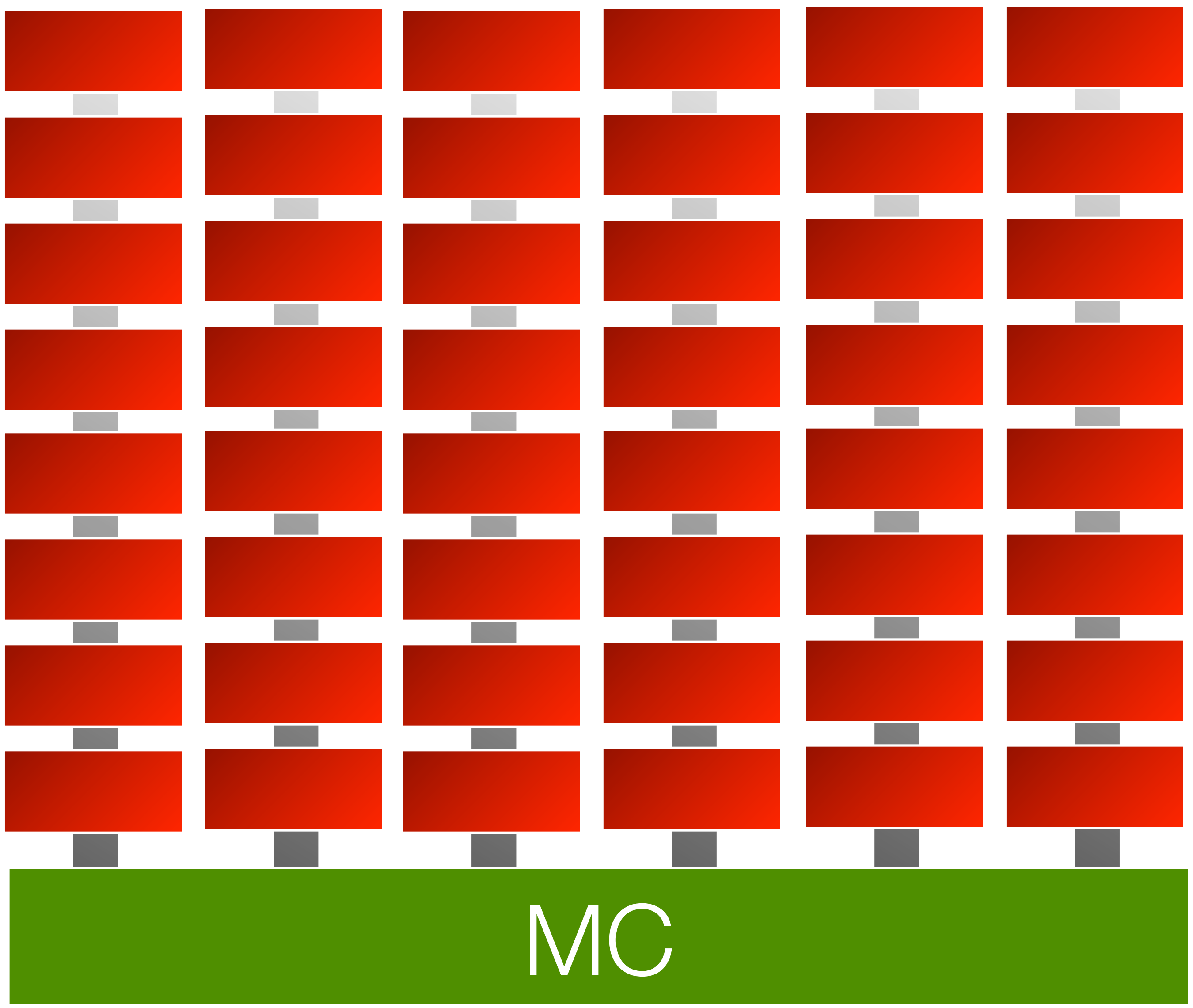
```
if (cache_miss(addr)) {
  cycle_count += DRAM_LATENCY;
}
```

even in simulators with “cycle accurate” memory systems—no lie

Problem: Capacity



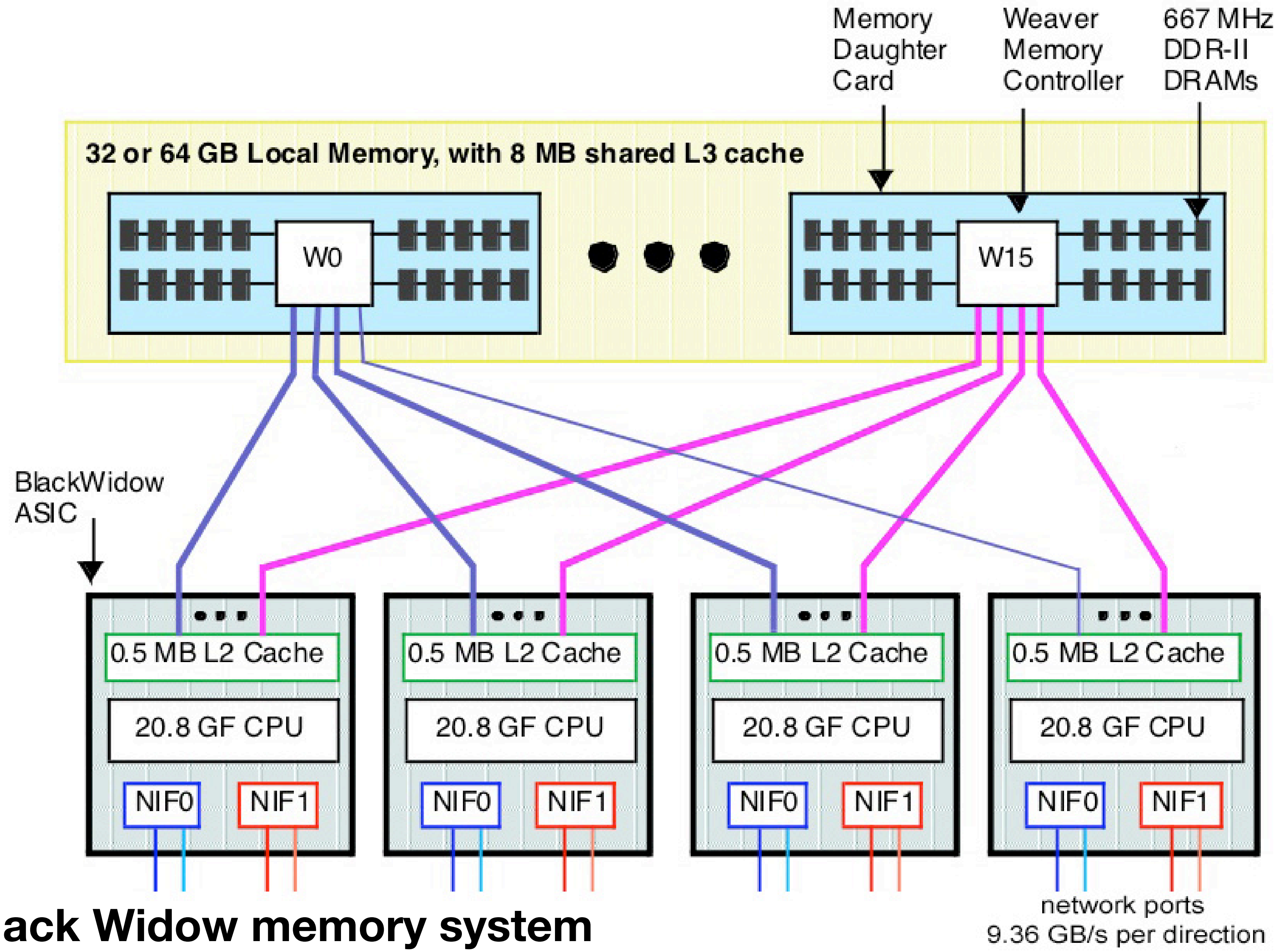
JEDEC DDRx
~10W/DIMM, ~20W total



FB-DIMM
~10W/DIMM, ~300W total

Problem: Bandwidth

Sometimes bandwidth is everything ...



Cray Black Widow memory system