CSE 530
Fundamentals of Computer Architecture

Spring 2021

Main Memory Architectures

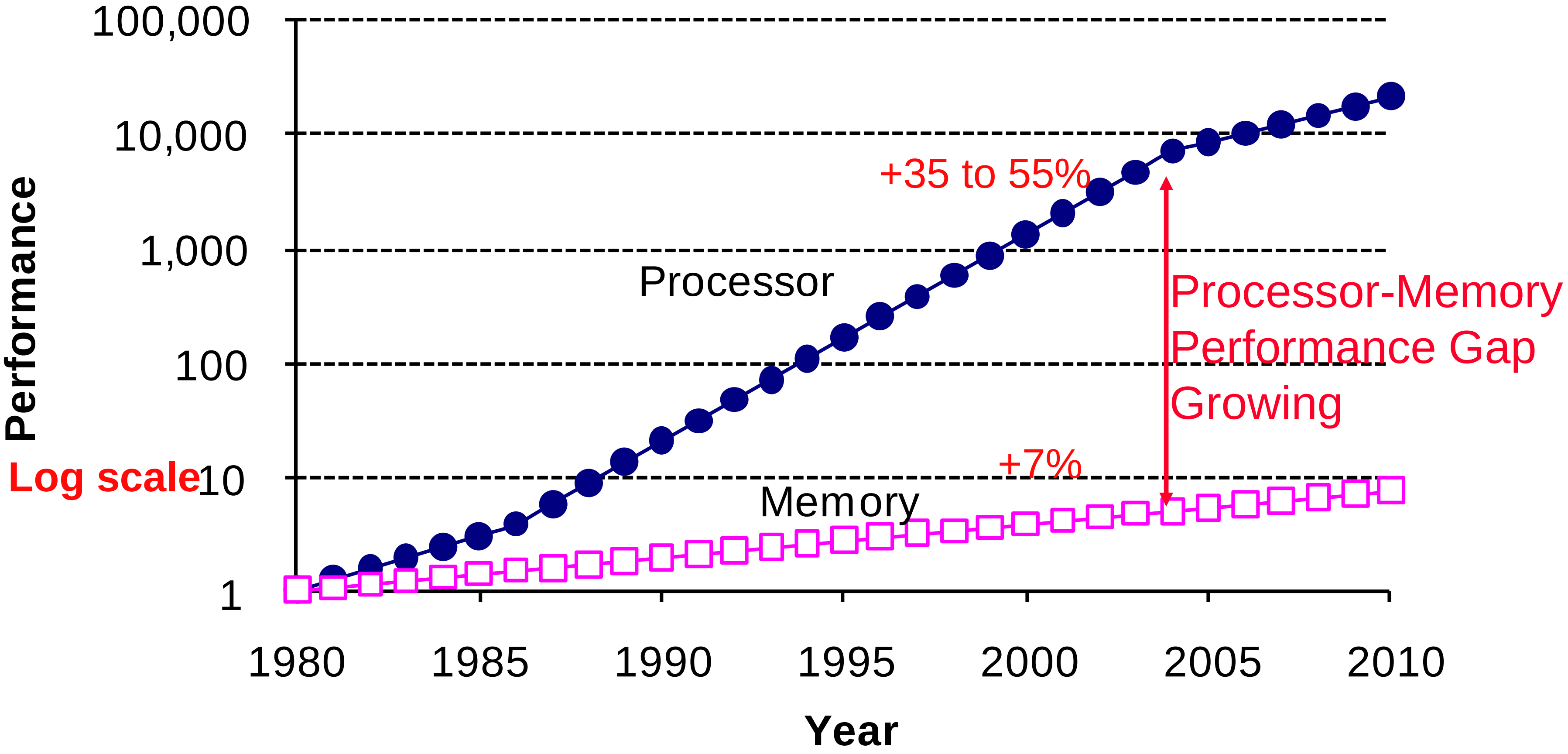
John (Jack) Sampson (cse.psu.edu/~sampson)

Course material on CANVAS

[Adapted in part from slides by Mary Jane Irwin, V. Narayanan, Amir Roth, Milo Martin, Onur Mutlu, Rajeev Balasubramonian, and others]

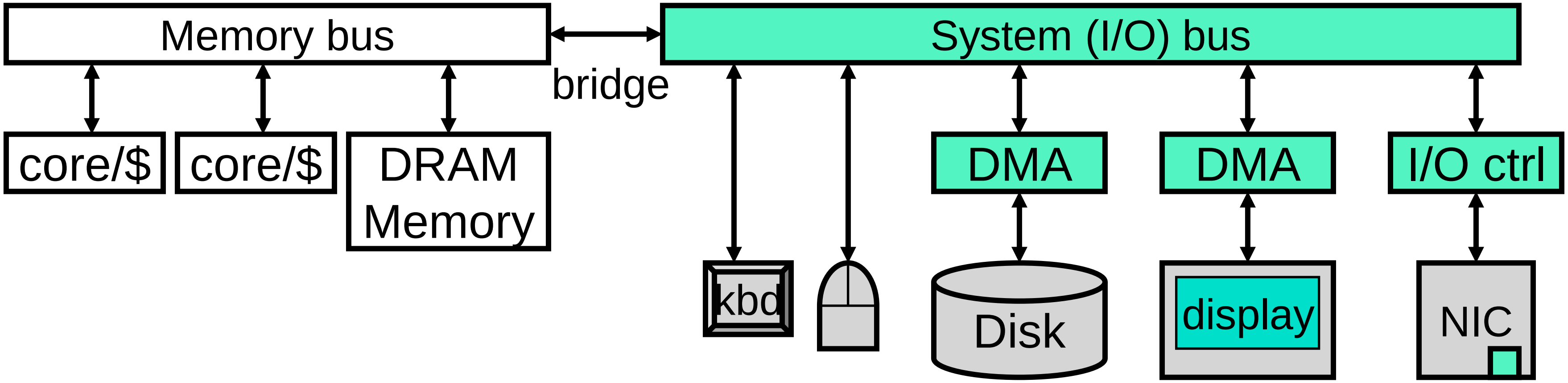
Recall: The memory “wall”

- Processors are getting faster faster than memories are getting faster



Completing the memory hierarchy

- Cores, caches and **main memory**
  - Connected by the **memory bus** (aka northbridge, ideally on-chip with the cores and caches – **is on-chip** for modern processors)



- I/O peripherals: storage, input, display, network, …
  - With separate or built-in DMA
  - Connected by the **system bus** (aka southbridge) which is connected to memory bus

Memory hierarchy design goal

- Its important to match the cache characteristics
  - Remember, caches want information provided to them one **block** at a time (and a block is usually more than one word)

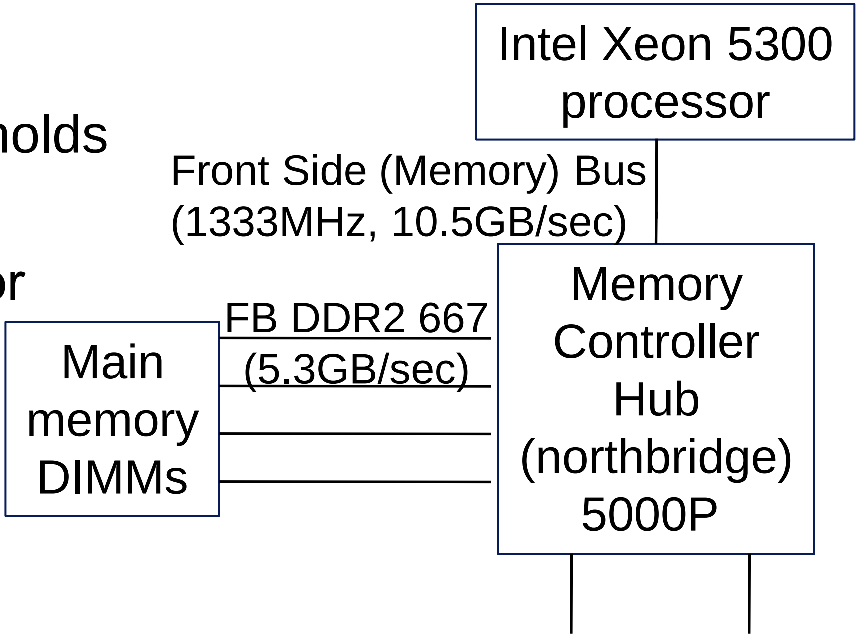
  with the main memory characteristics
  - use DRAMs that support fast multiple word accesses, preferably ones that **match the block size** of the cache

  with the controller/memory-bus characteristics
  - make sure the memory-bus can support the DRAM access rates and patterns
  - with the goal of increasing the Memory-Bus_to_Cache **bandwidth**

- Amdahl’s rule of thumb – memory capacity should grow linearly with processor speed to keep a balanced system

DRAM packaging - DIMMs

- Dual In-line Memory Modules
  - Small printed circuit board that holds DRAMs with a 64-bit datapath
  - Each contain eight* “x4” (by 4) or “x8”(by 8) DRAM parts



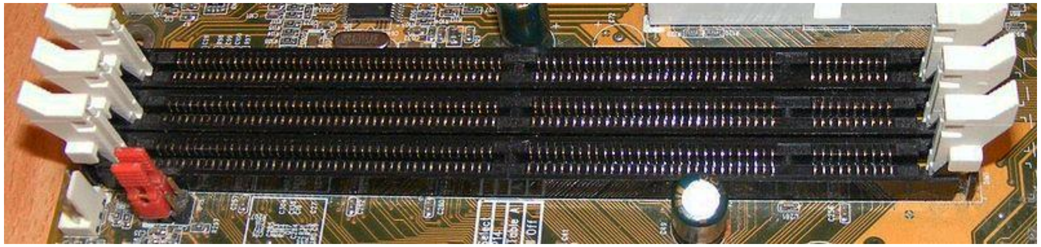
Intel Xeon 5300 processor

Front Side (Memory) Bus (1333MHz, 10.5GB/sec)

Memory Controller Hub (northbridge) 5000P

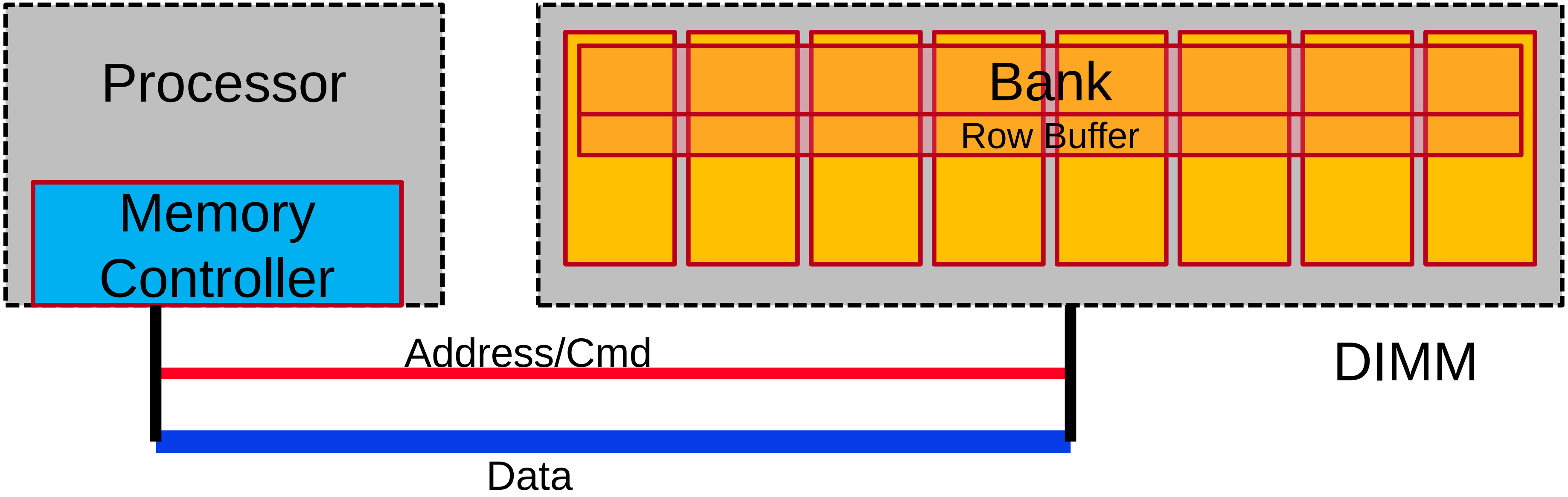
FB DDR2 667 (5.3GB/sec)

Main memory DIMMs



* or more, for ECC

Intra-DRAM hierarchy



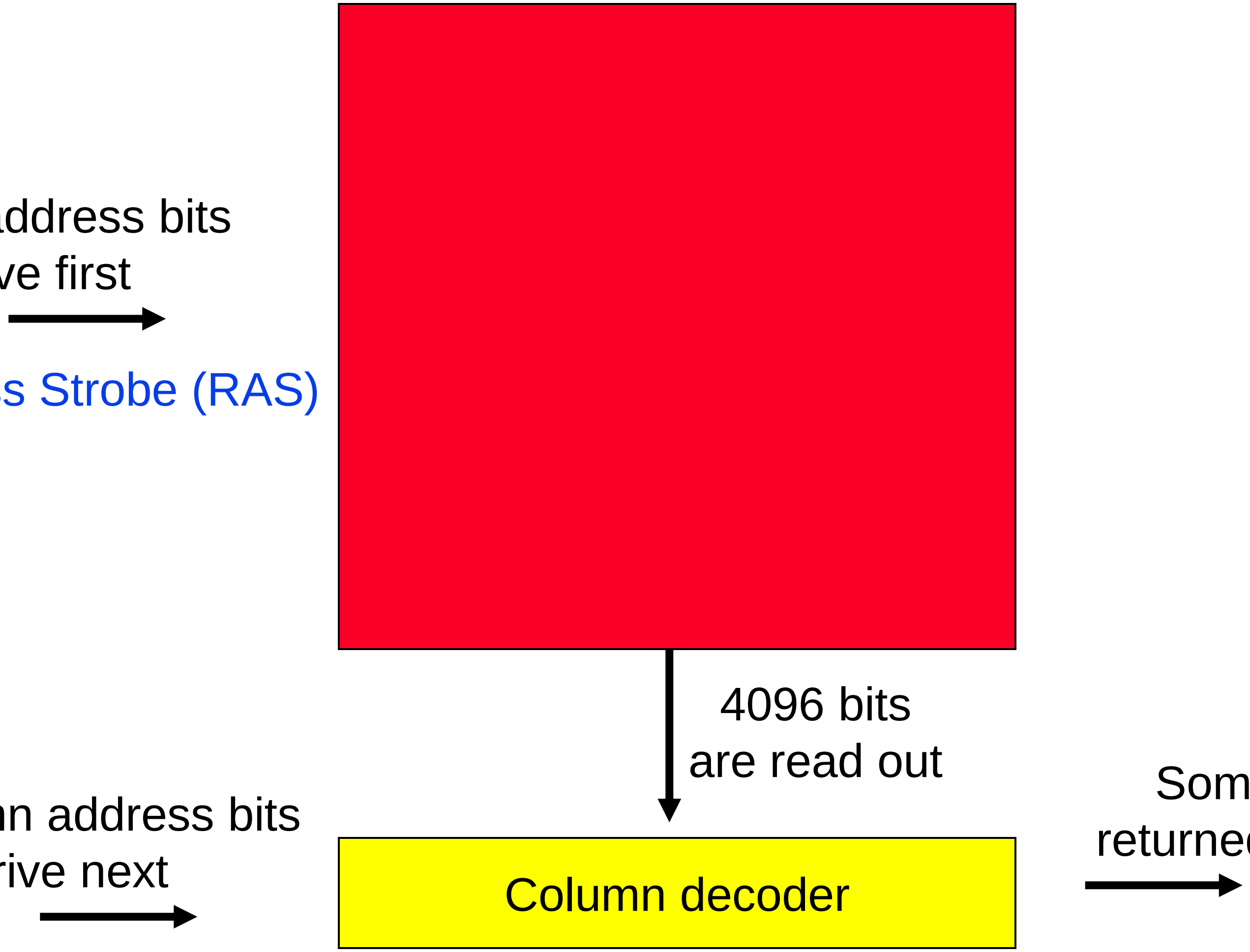
- DIMM: a PCB with DRAM chips on the back and front
- Rank: a collection of DRAM chips that work together to respond to a request and keep the data bus full
  - A 64-bit data bus will need 8 x8 DRAM chips or 4 x16 DRAM chips or..
- Bank: a subset of a rank that is busy during one request
  - DDR4 and later standards support “Bank Groups” between Rank and Bank
- Row buffer: the last row (say, 8 KB) read from a bank, acts like a cache

Main memory DRAMs

- DRAM addresses are divided into 2 halves (row and column) – so think of the memory as a 2D matrix
  - *RAS* or *Row Access Strobe* that triggers the row decoder
  - *CAS* or *Column Access Strobe* that triggers the column decoder
  - DRAM cells need to be refreshed periodically (~8 ms, <5% time)
- **Latency**: Time to access one word
  - *Access Time*: time between request and when word is read or written
    - read access and write access times can be different
    - row access time largely determines latency, data transfer time (CAS) largely determines memory bandwidth
  - *Cycle Time*: time between successive (read or write) requests
  - Usually cycle time > access time
- **Bandwidth**: How much data can be supplied per unit time
  - width of the data channel * the rate at which it can be used

DRAM Array Access

16Mb DRAM array = 4096 x 4096 array of bits



12 row address bits
arrive first

Row Access Strobe (RAS)

4096 bits
are read out

12 column address bits
arrive next

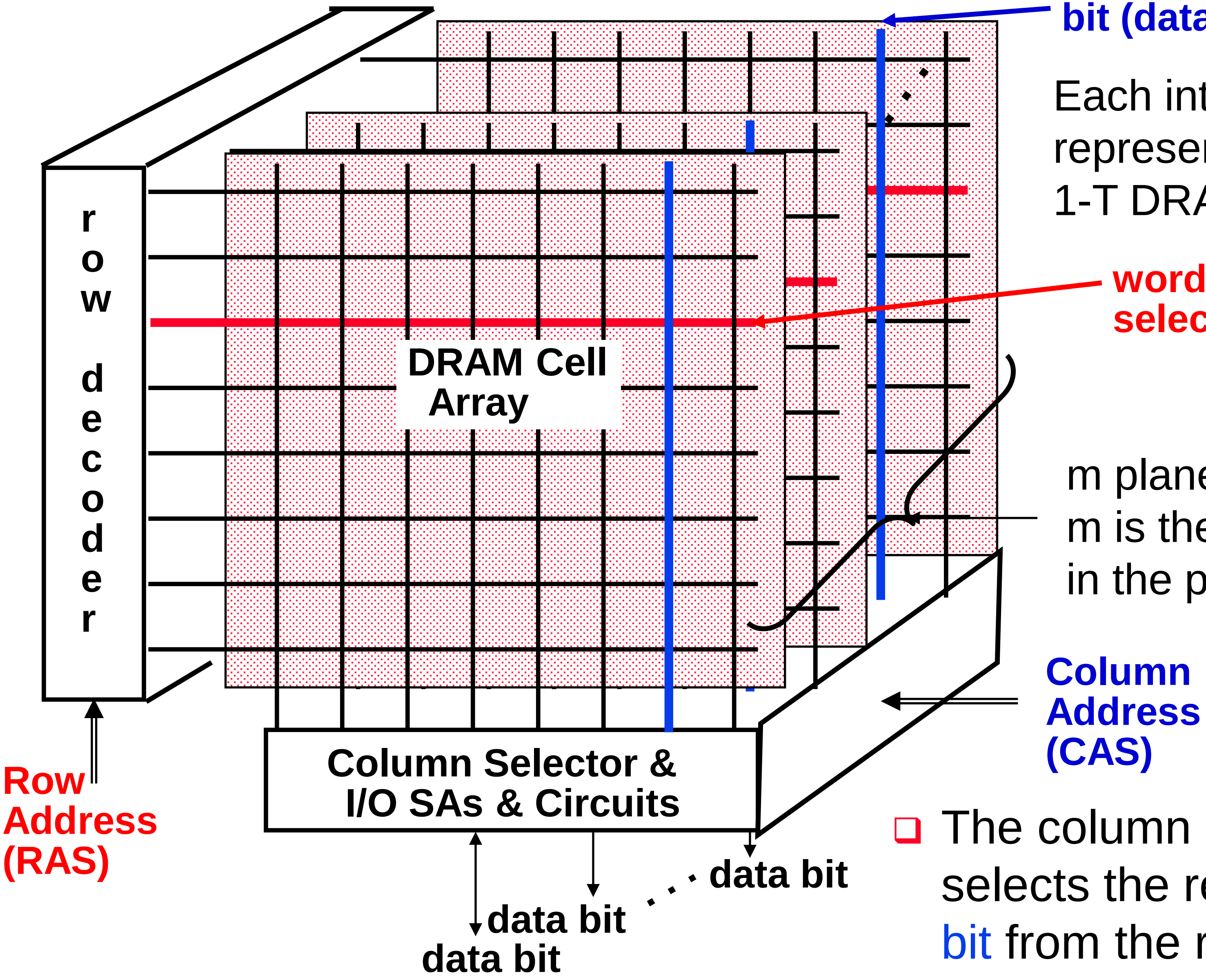
Column decoder

Some bits
returned to CPU

Column Access Strobe (CAS)

Row Buffer

Classical DRAM organization (~square planes)



Each intersection represents a 1-T DRAM cell

m planes (where m is the # of bits in the part)

- The column address selects the requested bit from the row in each plane

Organizing a Rank

- DIMM, rank, bank, array → form a hierarchy in the storage organization
- Because of electrical constraints, only a few DIMMs can be attached to a bus
- One DIMM can have 1-4 ranks
- For energy efficiency, use wide-output DRAM chips – better to activate only 4 x16 chips per request than 16 x4 chips
- For high capacity, use narrow-output DRAM chips – since the ranks on a channel are limited, capacity per rank is boosted by having 16 x4 2Gb chips than 4 x16 2Gb chips

Organizing Banks and Arrays

- A rank is split into many banks (4-16) to boost parallelism within a rank

- Ranks and banks offer memory-level parallelism

- A bank is made up of multiple arrays (subarrays, tiles, mats)

- To maximize density, arrays within a bank are made large → rows are wide → row buffers are wide (8KB read for a 64B request, called overfetch)

- Each array provides a single bit to the output pin in a cycle (for high density)

Aside: DRAMs and DIMMS (DDR nomenclature)

- http://en.wikipedia.org/wiki/DDR*_SDRAM

Standard	I/O bus (MHz)	M transf's / sec	DRAM Name	MB/sec/ DIMM	DIMM Name
DDR	133	266	DDR-266	2128	PC2100
DDR	200	400	DDR-400	3200	PC3200
DDR2	333	667	DDR2-667	5336	PC5300
DDR2	400	800	DDR2-800	6400	PC6400
DDR3	533	1066	DDR3-1066	8533	PC8500
DDR3	667	1333	DDR3-1333	10667	PC10600
DDR3	800	1600	DDR3-1600	12800	PC12800
DDR3	933	1866	DDR3-1866	14933	PC14900
DDR3	1066	2133	DDR3-2133	17066	PC17000

x 2

x 8

DDR4

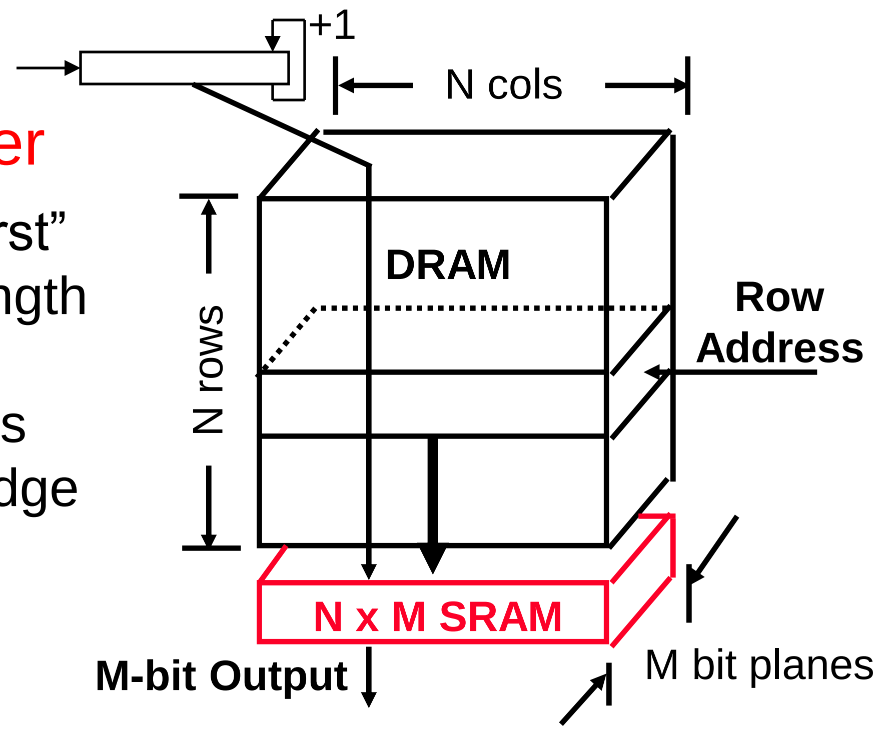
Standard name	Memory clock (MHz)	I/O bus clock (MHz)	Data rate (MT/s)	Module name	Peak trans-fer rate (MB/s)	Timings CL-tRCD-tRP	CAS latency (ns)
DDR4-1600J*	200	800	1600	PC4-12800	12800	10-10-10	12.5
DDR4-1600K	200	800	1600	PC4-12800	12800	11-11-11	13.75
DDR4-1600L	200	800	1600	PC4-12800	12800	12-12-12	15
DDR4-3200W	400	1600	3200	PC4-25600	25600	20-20-20	12.50
DDR4-3200AA	400	1600	3200	PC4-25600	25600	22-22-22	13.75
DDR4-3200AC	400	1600	3200	PC4-25600	25600	24-24-24	15

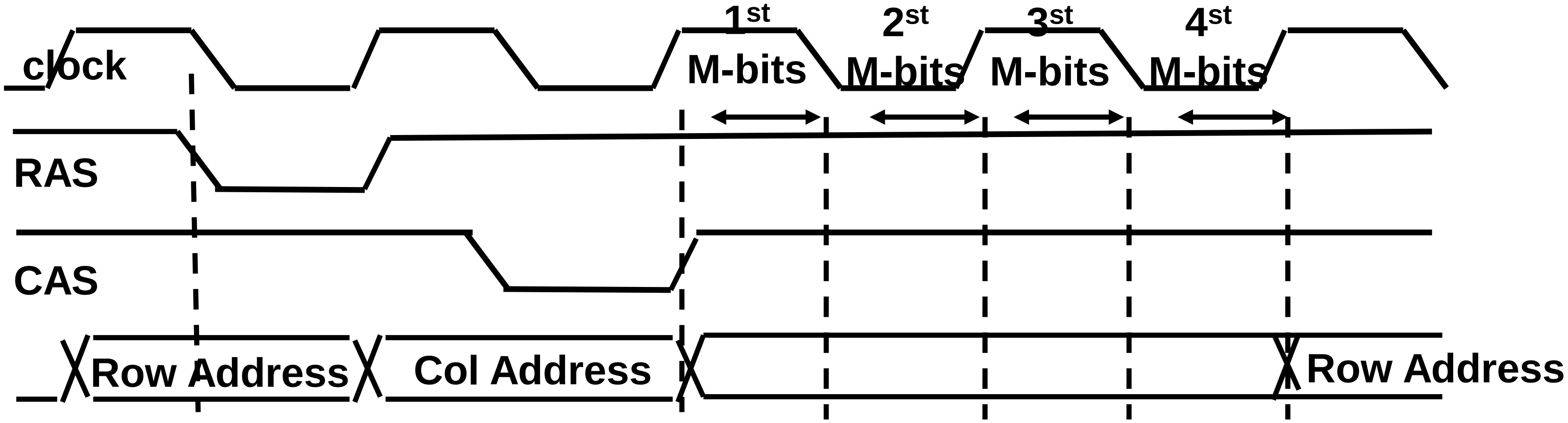
Synchronous DRAMs (SDRAMs)

- Synchronous DRAMs can transfer a burst of data from a series of sequential addresses that are in the same row
- For words in the same burst, don't have to provide the complete (row and column) addresses
  - Specify the starting (row+column) address and the burst length (burst must all be in the same DRAM row). The row is accessed from the DRAM and loaded into an SRAM row buffer.
  - Data words in the burst are then accessed from that row buffer under control of a clock signal.
- DDR SDRAMs (Double Data Rate SDRAMs)
  - Transfers burst data on *both* the rising and falling edge of the clock (so twice fast)
  - Improves bandwidth, not latency (to first word in the row)
- Had DDR2 (400 MT/s), now have DDR3 (800 MT/s) and, DDR4 (2133 MT/s)

Synchronous DRAM (SDRAM) Operation

- After RAS loads a row into the SRAM row buffer
  - Input CAS as the starting “burst” address along with a burst length to read a burst of data from a series of sequential addresses within that row on the clock edge





http://en.wikipedia.org/wiki/DDR_SDRAM

Row Buffers

- Each bank may have only a single (or few) row buffer(s)

- Row buffers act as a cache within DRAM
  - Row buffer hit: ~20 ns access time (must only move data from row buffer to pins)
  - Empty row buffer access: ~40 ns (must first read arrays, then move data from row buffer to pins)
  - Row buffer conflict: ~60 ns (must first precharge the bitlines, then read new row, then move data to pins)

- In addition, must wait in the queue (tens of nano-seconds) and incur address/cmd/data transfer delays (~10 ns)

Open/Closed Page Policies

- If an access stream has locality, a row buffer is kept open
  - Row buffer hits are cheap (open-page policy)
  - Row buffer miss is a bank conflict and expensive because precharge is on the critical path

- If an access stream has little locality, bitlines are precharged immediately after access (close-page policy)
  - Nearly every access is a row buffer miss
  - The precharge is usually not on the critical path

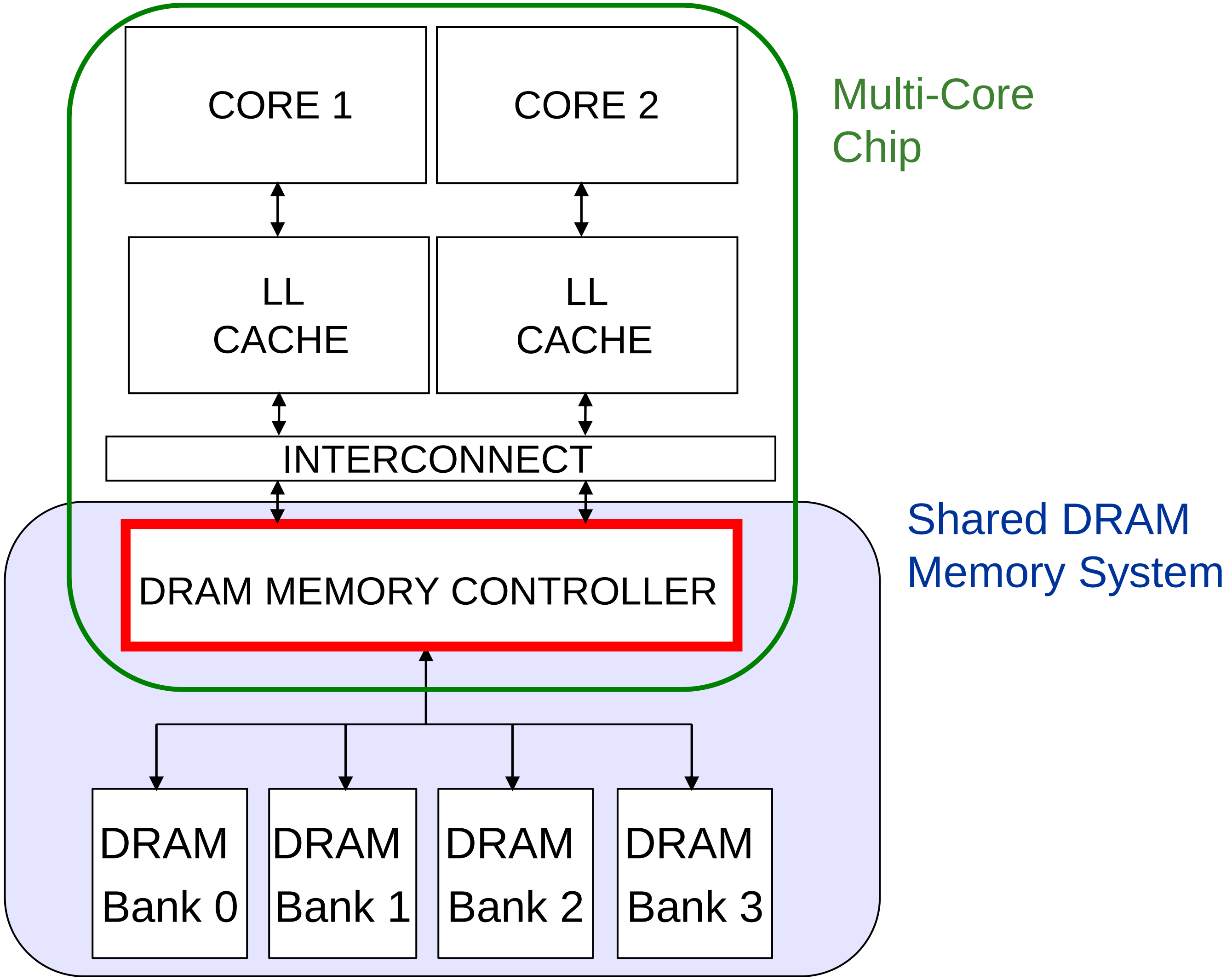
- Modern memory controller policies lie somewhere between these two extremes (usually proprietary)

Aside: Need for error correction in DRAMs

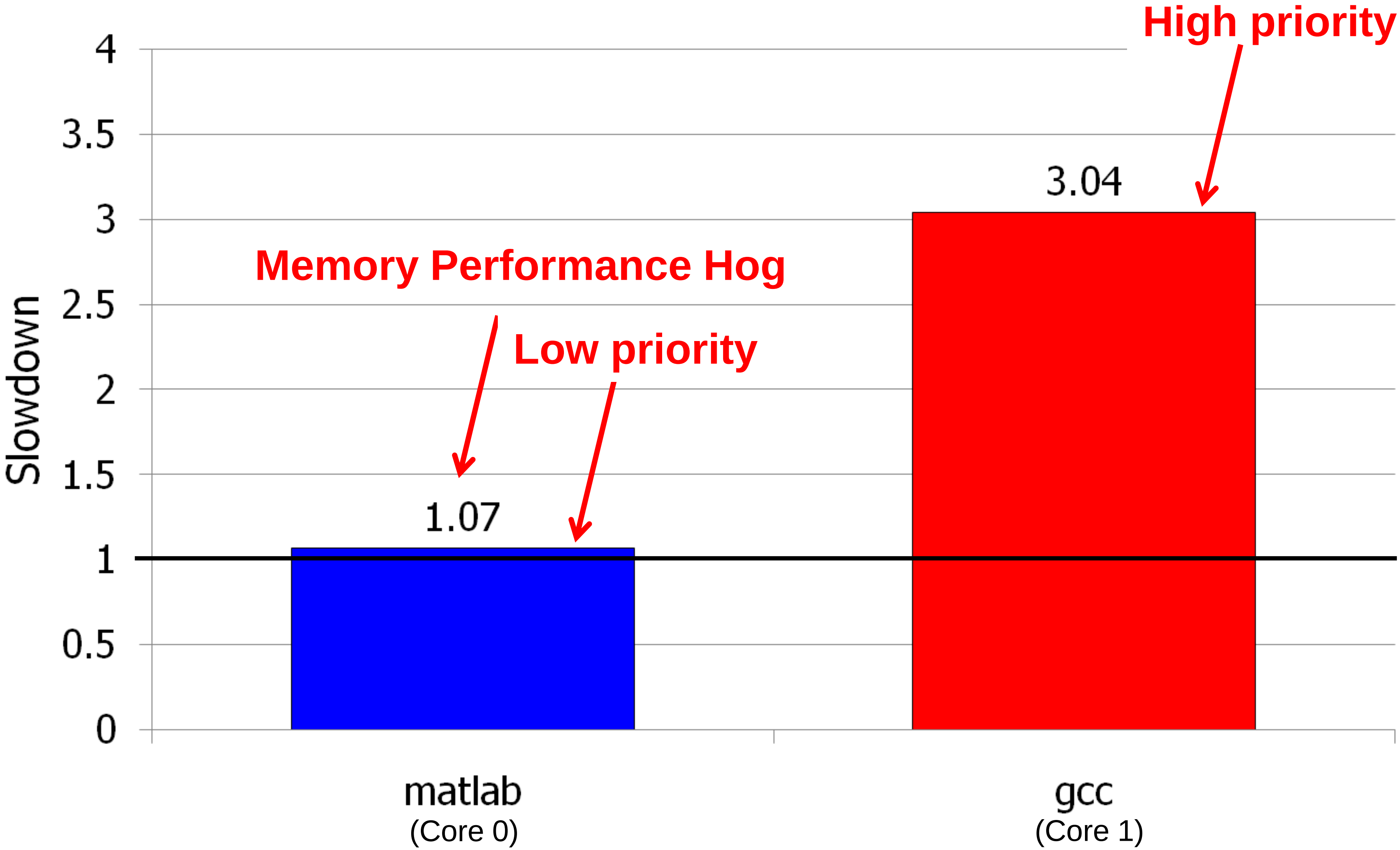
- Failures/time *proportional* to number of bits!
  - Also as DRAM cells scale down, they are more vulnerable

- Basic idea: add redundancy through parity bits
  - Common configuration: Random error correction
    - Parity (single error detect) – only takes one bit per word
    - SECDED (single error correct, double error detect) – e.g., for 64 data bits need 8 "parity" bits (11% overhead)
  - Really want to handle failures of physical components as well
    - Organization is multiple DRAMs/DIMM, multiple DIMMs
    - Want to recover from failed DRAM and failed DIMM!
    - "Chip kill" handle failures width of single DRAM chip

What about multicores ?



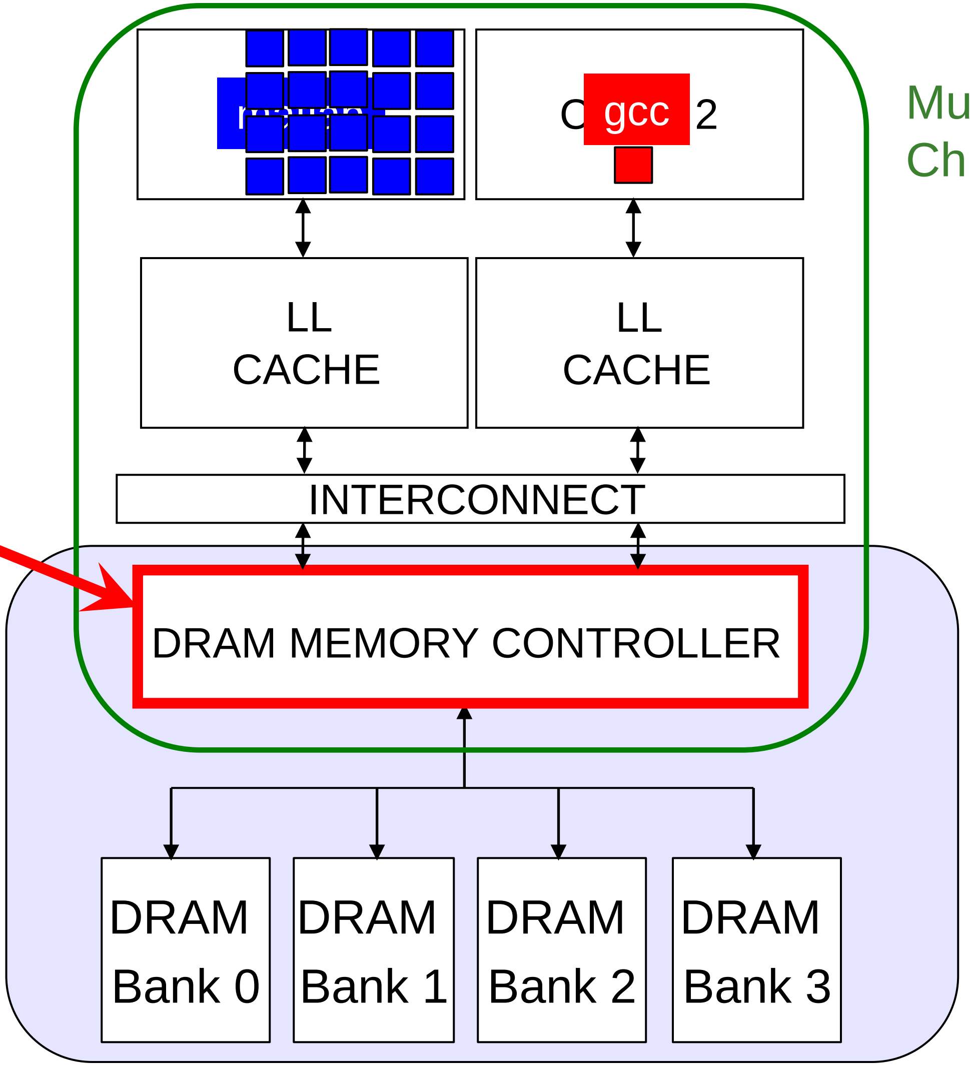
Unexpected slowdowns



Moscibroda and Mutlu, "Memory performance attacks: Denial of memory service in multi-core systems," USENIX Security 2007.

Slide by Onur Mutlu

Why? Uncontrolled memory interference



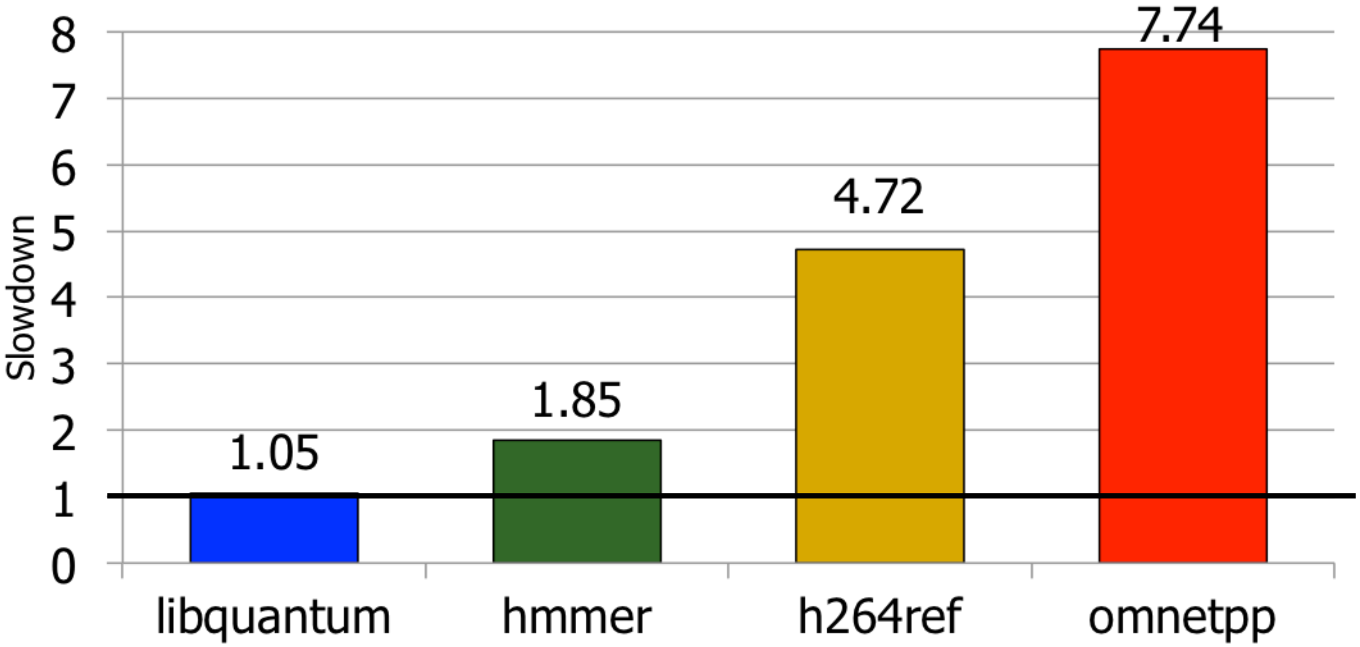
unfairness

Multi-Core Chip

Shared DRAM Memory System

Slide by Onur Mutlu

The more the cores, the greater the problem



- ❑ Vulnerable to denial of service [Usenix Security'07]
- ❑ Unable to enforce priorities or SLAs [MICRO'07,'10,'11, ISCA'08'11, ASPLOS'10]
- ❑ Low system performance [MICRO'07,'10,'11, ISCA'08,'11, HPCA'10, ASPLOS'10]

Uncontrollable, unpredictable system

Slide by Onur Mutlu