CSE 431
Computer Architecture
Fall 2015

Chapter 5A: Exploiting the Memory Hierarchy: Main Memory

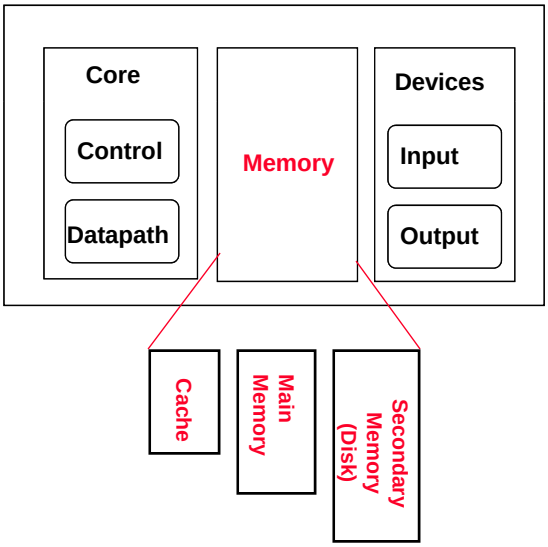
Mary Jane Irwin (www.cse.psu.edu/~mji)

[Adapted from *Computer Organization and Design, 5th Edition*, Patterson & Hennessy, © 2014, Morgan Kaufmann
With additional thanks/credits to Onur Mutlu, ECE/CMU]

Reminders

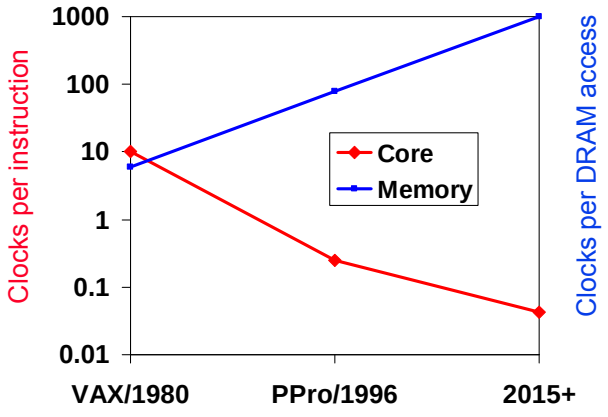
- This week
  - The memory hierarchy, DRAMs – P&H, 51.-5.2
  - Cache basics, improving cache performance – P&H, 5.3, 5.7
- Next week
  - Virtual memory hardware support (TLBs) – P&H 5.6-5.8
  - VLIW datapaths – P&H 4.10
- Reminders
  - Quiz 2 dropbox closes midnight Sept 22nd
  - HW3 dropbox closes midnight Oct 1st
  - Quiz 3 will open Sept 23rd (and will close midnight Oct 4th)
  - First evening midterm exam scheduled
    - Tuesday, October 6th, 20:15 to 22:15, Location 22 Deike
    - No conflict exam !!

Review: Major Components of a Computer



The "Memory Wall"

- Core vs DRAM speed disparity continues to grow



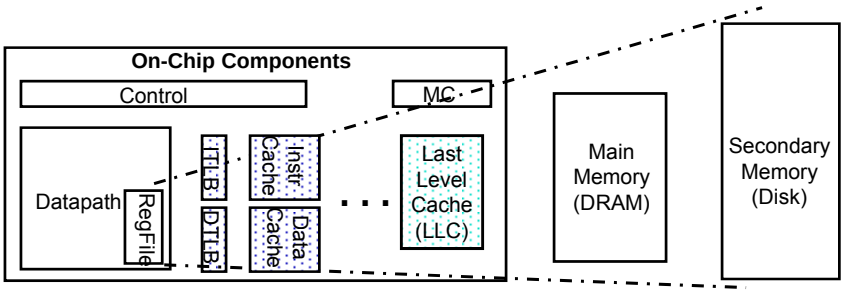
- Good memory hierarchy (cache) design is increasingly **important** to overall performance

The Memory Hierarchy Goal

- Fact: Large memories are slow and fast memories are small
- Fact: On-chip (with the cores) memories are much faster to respond than off-chip memories (even when they are the same size) because of much faster interconnect
- How do we create a memory that gives the illusion of being large, cheap and fast (most of the time)?
  - With hierarchy
  - With parallelism

A Typical Memory Hierarchy

- Take advantage of the principle of locality to present the user with as much memory as is available in the *cheapest* technology at the speed offered by the *fastest* technology



Speed (%cycles):	½'s	1's	10's	100's	10,000's
Size (bytes):	100's	10K's	M's	G's	T's
Cost/byte:	highest				lowest

The Memory Hierarchy: Why Does it Work?

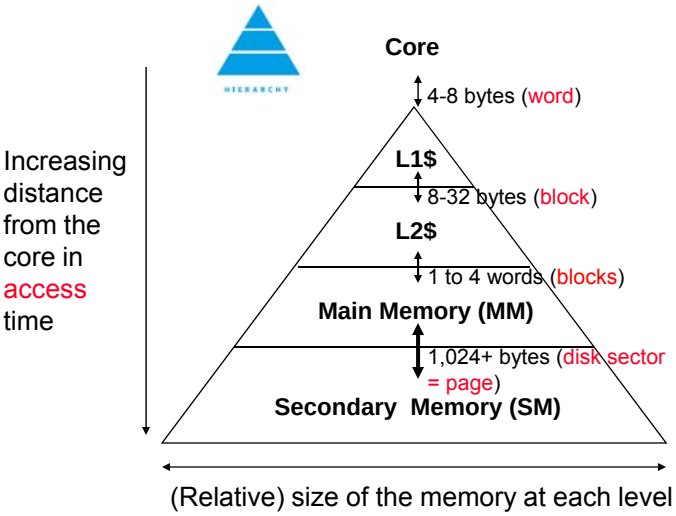
- Temporal Locality (locality in time)
  - If a memory location is referenced then it will tend to be referenced again soon

  ⇒ Keep most recently accessed data items closer to the core

- Spatial Locality (locality in space)
  - If a memory location is referenced, the locations with nearby addresses will tend to be referenced soon

  ⇒ Move blocks consisting of contiguous words closer to the core

Characteristics of the Memory Hierarchy



Inclusive caches – what is in L1$ is a subset of what is in L2$

Noninclusive caches – what is in L1$ is not a subset of what is in L2$

The Memory Hierarchy: Terminology

Hit Time << Miss Penalty

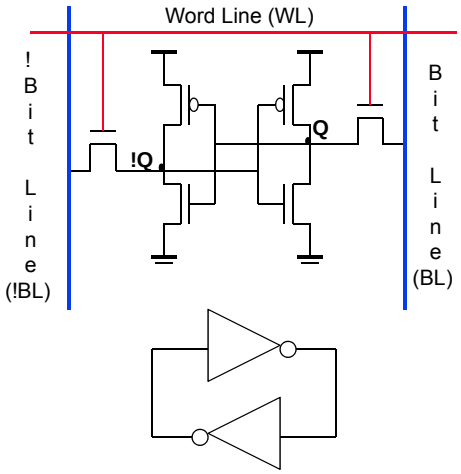
- Block (or line): the minimum unit of information that is present (or not) in a level of the memory hierarchy
- Hit Rate: the fraction of memory accesses found in a level of the memory hierarchy
  - Hit Time: Time to access that level which consists of
    Time to access the block + Time to determine hit/miss
- Miss Rate: the fraction of memory accesses *not* found in a level of the memory hierarchy ⇒ 1 - (Hit Rate)
  - Miss Penalty: Time to replace a block in that level with the corresponding block from a lower level which consists of
    Time to determine that there is a miss + Time to access that block in the lower level + Time to transmit that block to the level that experienced the miss + Time to insert the block in that level + Time to pass the block to the requestor

Memory Hierarchy Technologies

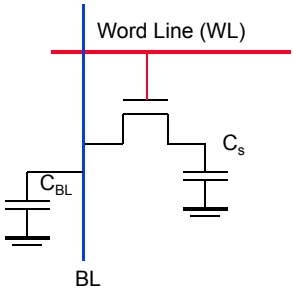
- Caches use *SRAM* for speed and because its technology is compatibility with the core's technology
  - Fast (typical access times of 100 psec to 2 nsec)
  - Lower density (6 transistor cells), higher power, expensive ($500 to $1000 per GB in 2012)
  - Static: content will last "forever" (as long as power is left on)
- Main memory uses *DRAM* for size (density)
  - Slower (typical access times of 10 nsec to 70 nsec)
  - Higher density (1 transistor cells), lower power, cheaper ($10 to $20 per GB in 2012); not typically compatible with the core's technology
  - Dynamic: needs to be "refreshed" regularly (~ every 64 ms) or will "forget" its state
    - refresh consumes ~ 1% of the active cycles of the DRAM

RAM Bit Cells

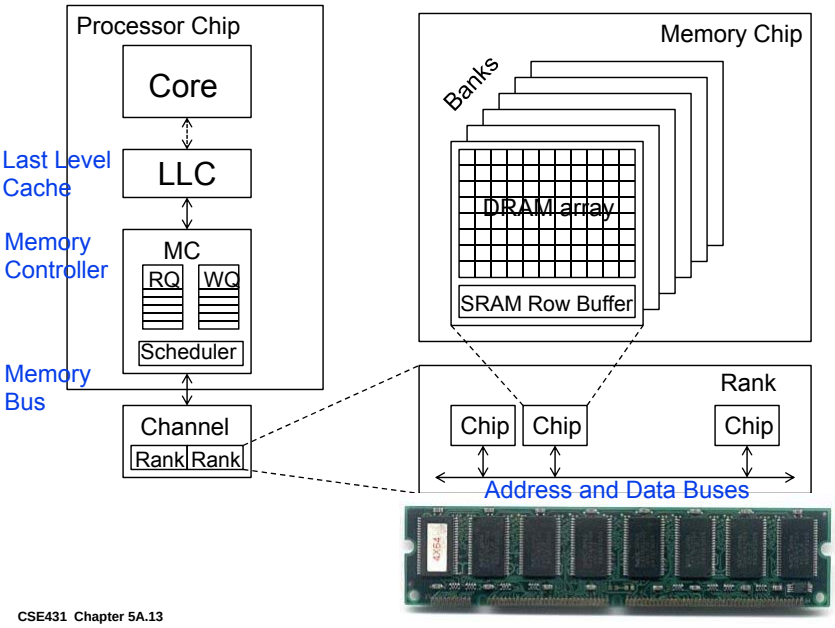
- 6-T SRAM Cell



- 1-T DRAM Cell
  - Read is destructive (so must refresh after read)
  - Requires sense amp (SA) to "read" the BL

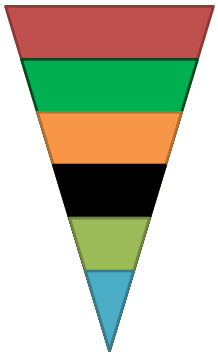


From Core to Main Memory DRAM

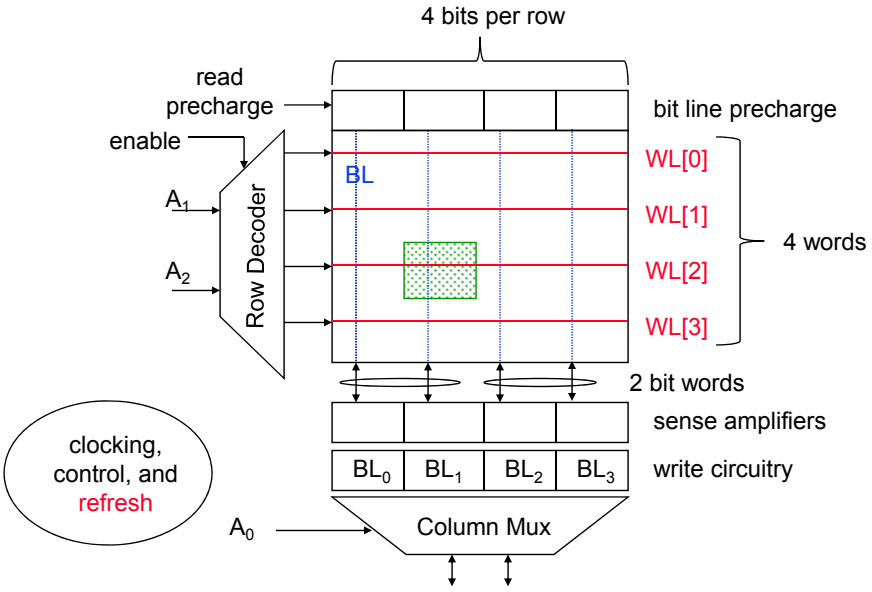


DRAM Subsystem Organization

- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column



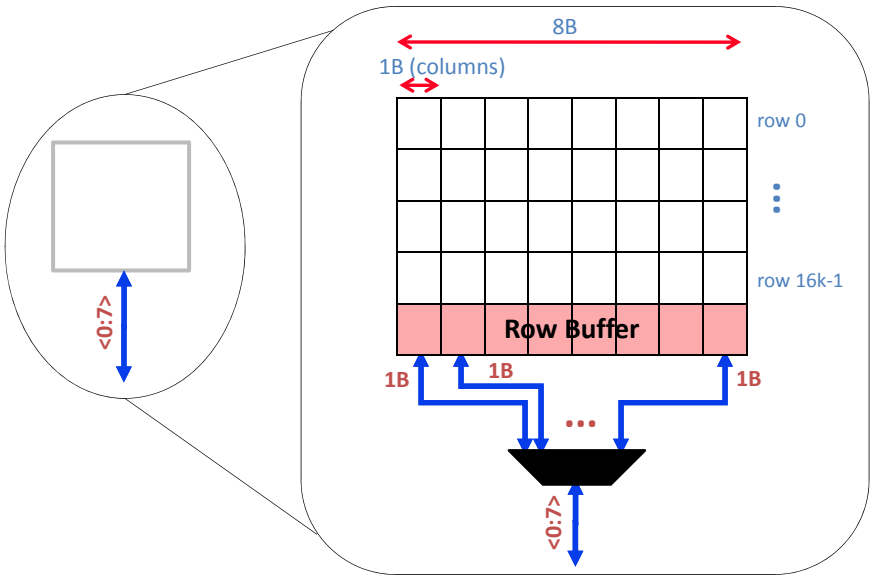
2D 4x4 DRAM Array



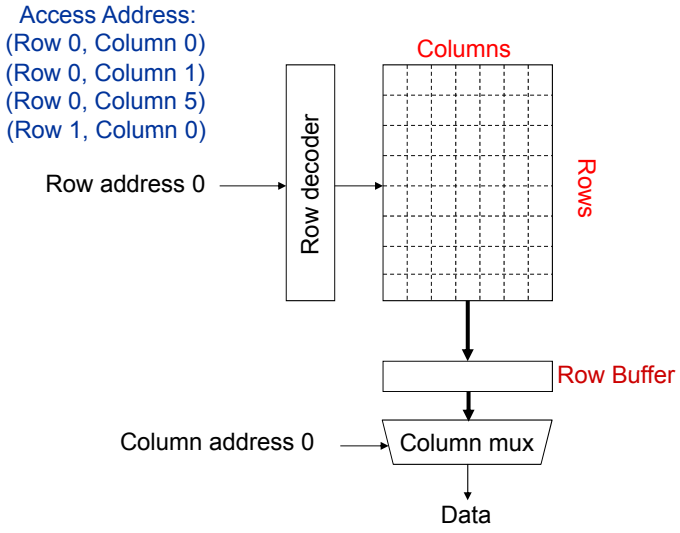
DRAM Row/Column Access

- A DRAM Bank is a 2D array of cells: rows x columns
  - A "DRAM row" is also called a "DRAM page"
- The row that is read out of the DRAM array is stored in an SRAM Row Buffer
- Each address is split into a <row,column> pair
  - *RAS* or *Row Access Strobe* triggering the row decoder
  - *CAS* or *Column Access Strobe* triggering the column mux
- Access to a "closed row"
  - Activate (ACT) command opens row (places it into the Row Buffer)
  - Read/write command reads/writes a column in the Row Buffer
  - Precharge command (PRE) closes the row and prepares the DRAM bank for the next access
- Accesses to an "open row"
  - No need for activate (ACT) command

Breaking Down a Bank



DRAM Bank Operation



Bank Access Timing

- So a typical memory bank access consists of

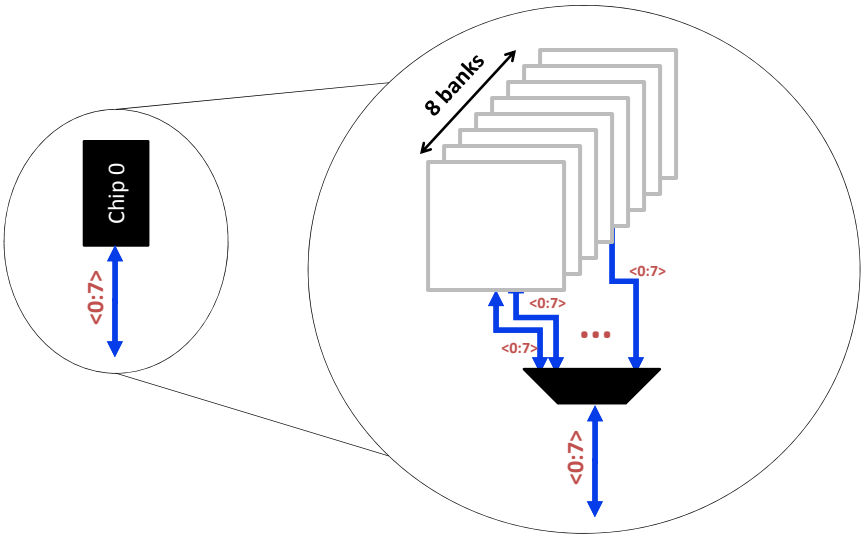
1. Opening the bank (ACT) if not already open by precharging (PRE) it
   10 units same bank
   2 units if different bank same chip

   and copying the requested row's bank data to its bank's Row Buffer (RAS)
   1 unit

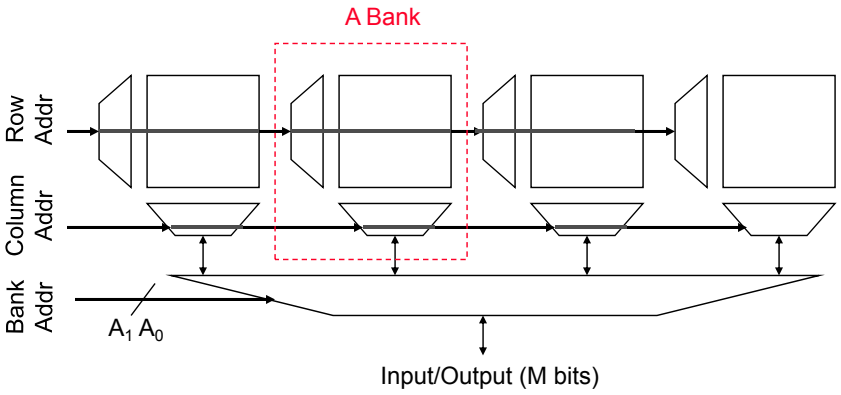
2. Issuing read (RD) and/or write (WR) which reads/writes from/to the Row Buffer (CAS)
   1 unit

3. And closing the bank by writing from the Row Buffer back to the DRAM
   1 unit

Breaking Down a Chip



Banked (or 3D) DRAM Chip

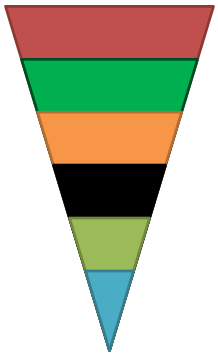


Advantages:

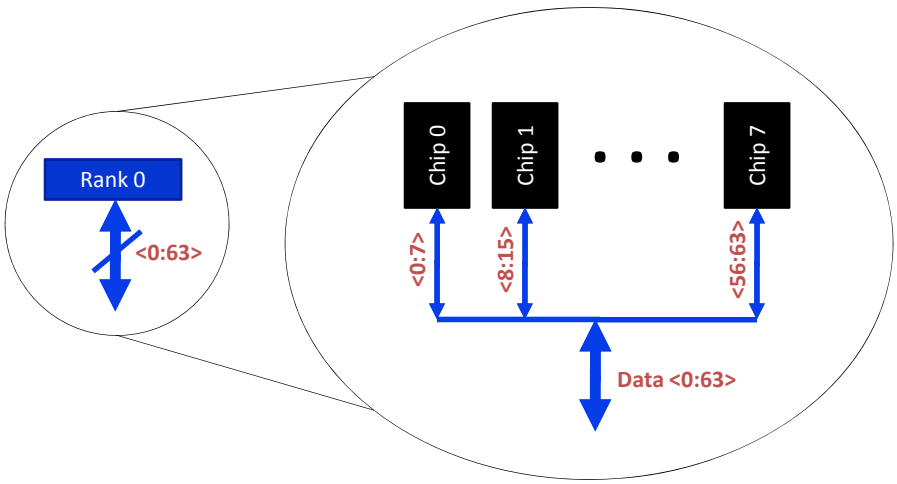
1. Shorter word and bit lines so faster access
2. Bank addr activates only 1 bank saving power

DRAM Subsystem Organization

- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column



Breaking Down a Rank

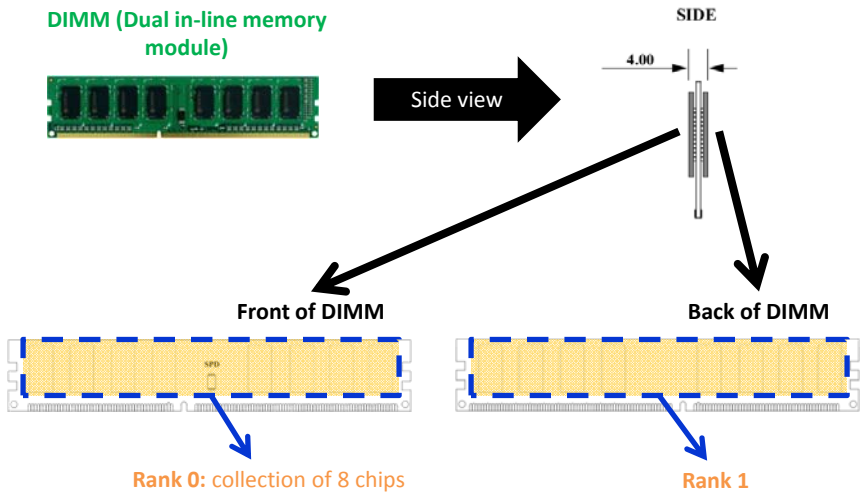


DIMM vs Rank

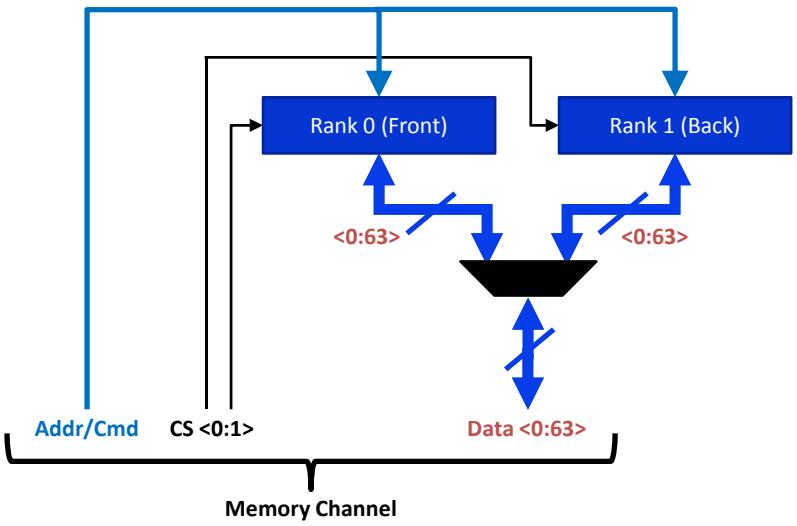
- Sometimes memory modules (DIMMs – Dual Inline Memory Modules) are designed with two or more independent sets of DRAM chips connected to the same address and data buses
  - Each such set is a **rank**
  - Since all ranks share the same buses, only one rank may be accessed at any given time; it is specified by activating the corresponding rank's chip select (CS) signal.

- DIMMs are currently being commonly manufactured with up to four ranks per module. Consumer DIMM vendors have recently begun to distinguish between single and dual ranked DIMMs.

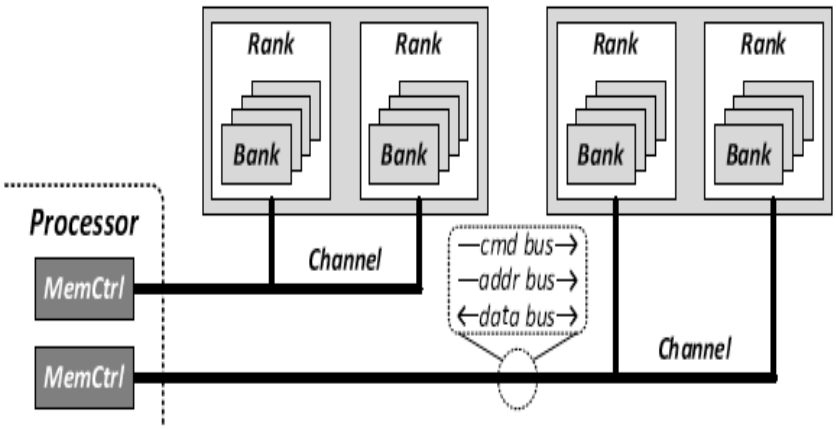
Breaking Down a DIMM



A DIMM with Two Ranks

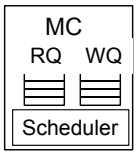


Generalized Memory Structure



Memory Controller (MC)

- The MC front-end buffers requests to the DRAM from the LLC (so independent of DRAM type)
  - Decodes the address into rank, bank, row, column
  - Schedules requests to DRAM to maximize memory bandwidth by
    1. Exploiting Row Buffer locality as much as possible
    2. Reordering bursts to minimize bank conflicts
    2. Prioritizing reads over writes (a core is waiting for the read data !)
    3. Preferring read after read and write after write to maximize read/write efficiency
  - Sends responses from the DRAM back to the LLC

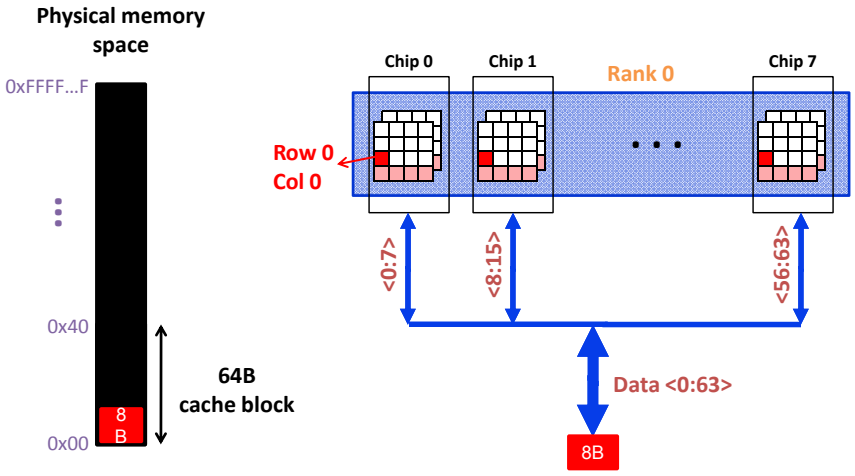


- The MC back-end interface to the target DRAM memory (so is dependent on DRAM type)

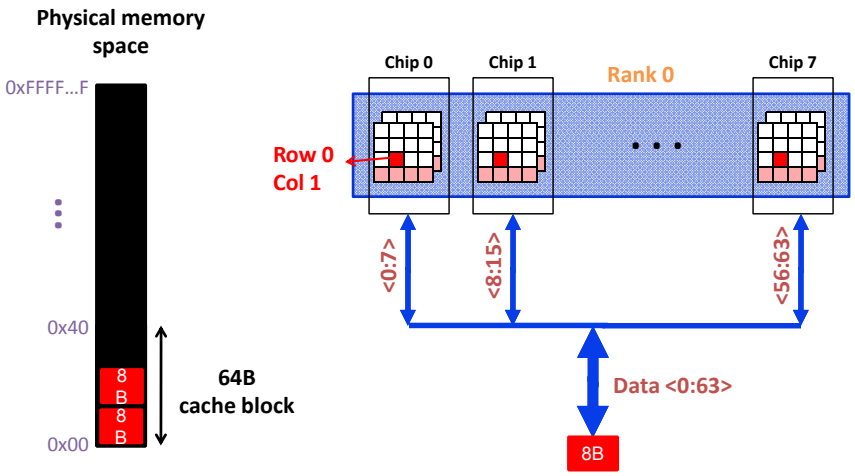
Timing Comparisons

1 Bank, 8 Blocks, 1 Word/Blocks		1 Bank, 2 Blocks, 4 Word/Blocks	
Row 0, Col 0		R0, Col 0	
R1, C0		R0, C1	
R2, C0		R0, C2	
R3, C0		R0, C3	
R4, C0		R1, C0	
R5, C0		R1, C1	
R6, C0		R1, C2	
R7, C0		R1, C3	
Total			

Example: Transferring a cache block



Example: Transferring a cache block



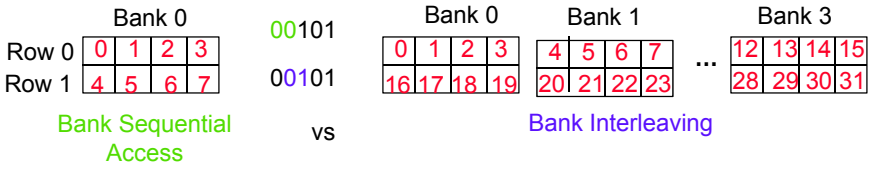
A 64B cache block takes 8 I/O cycles to transfer.
During the process, 8 columns are read sequentially

Latency Components

- Core → controller transfer time
- Controller latency
  - Queuing & scheduling delay at the controller
  - Converting access to basic memory commands
- Controller → DRAM transfer time
- DRAM bank latency
  - Simple CAS if row is "open" (in the Row Buffer) OR
  - RAS + CAS if the array is already precharged OR
  - PRE + RAS + CAS (worst case)
- DRAM → Controller transfer time
- Controller → Core transfer time

Multiple Banks (Interleaving) *and* Channels

- Multiple banks
  - Enables concurrent DRAM accesses
  - Bits in address determine which bank an address resides in



- Multiple independent channels or multiple memory controllers each with their own channel serve the same purpose
  - But they are even better because they have separate data buses and thus increased bus bandwidth
- Enabling more concurrency requires reducing
  - Bank conflicts and channel conflicts

More Timing Comparisons

2 Banks, 1 Block/Bank, 4 Word/Blocks		2 Chan, 1 Bank/Chan, 1 Block/Bank, 4 Word/Blocks	
B0, R0, C0		Ch0, R0, C0	
B0, R0, C1		Ch0, R0, C1	
B0, R0, C2		Ch0, R0, C2	
B0, R0, C3		Ch0, R0, C3	
B1, R0, C0		Ch1, R0, C0	
B1, R0, C1		Ch1, R0, C1	
B1, R0, C2		Ch1, R0, C2	
B1, R0, C3		Ch1, R0, C3	
Total			

DRAM Memory System Summary

- Its important to match the cache characteristics
  - caches access one block at a time (usually more than one word)

with the DRAM characteristics

  - use DRAMs that support fast multiple word accesses, preferably ones that match the block size of the cache

with the 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

- And to have a well-tuned, on-chip Memory Controller (or several in the case of multicores)

Reminders

- Next lecture
  - Cache basics, improving cache performance – P&H, 5.3, 5.7
- Next week
  - Virtual memory hardware support (TLBs) – P&H 5.6-5.8
  - VLIW datapaths – P&H 4.10
- Reminders
  - Quiz 2 dropbox closes midnight Sept 22nd
  - HW3 dropbox closes midnight Oct 1st
  - Quiz 3 will open Sept 23rd (and will close midnight Oct 4th)
  - First evening midterm exam scheduled
    - Tuesday, October 6th, 20:15 to 22:15, Location 22 Deike
    - No conflict exam !!