Computer Architecture: Main Memory (Part I)
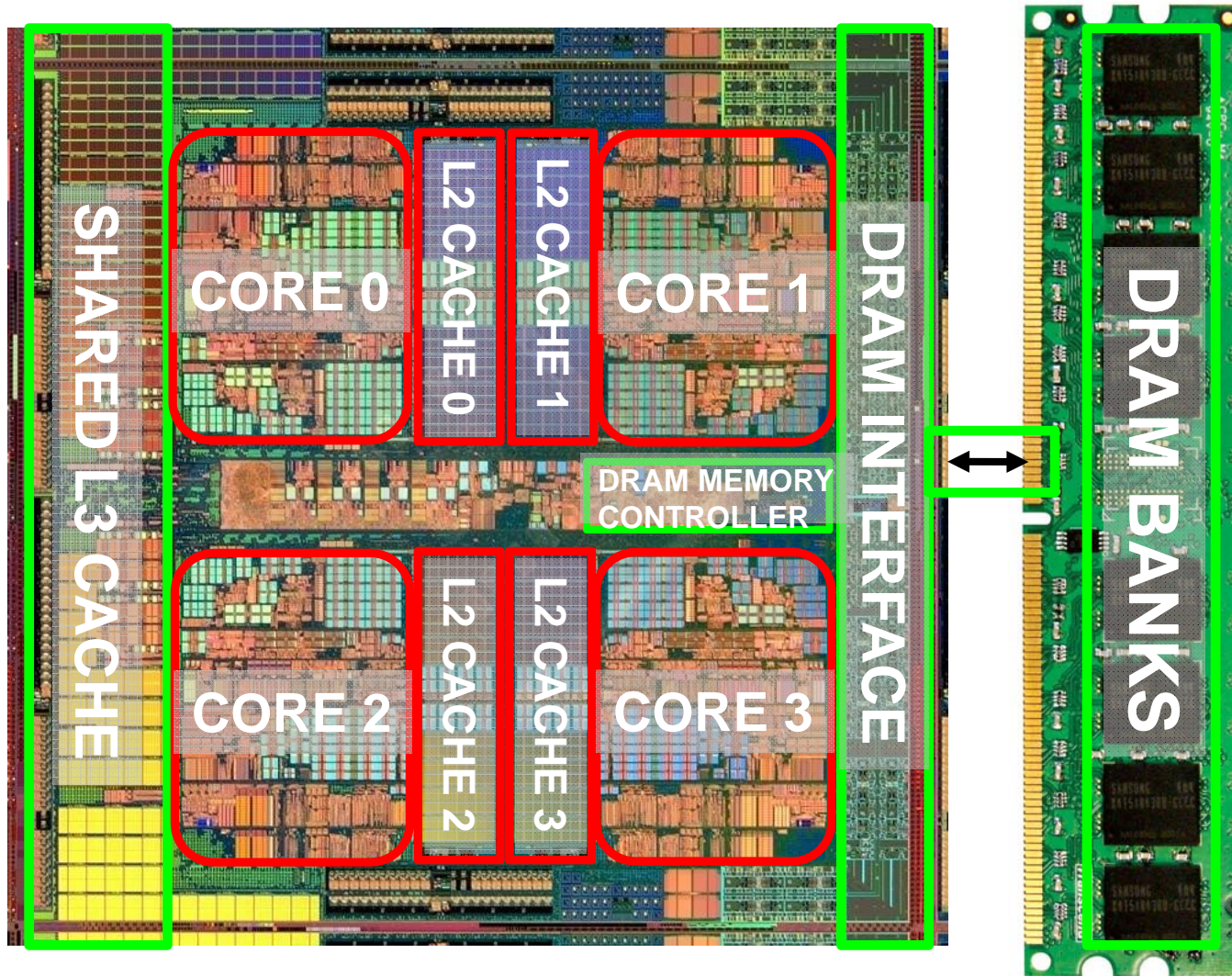
Prof. Onur Mutlu

Carnegie Mellon University

(reorganized by Seth)

Main Memory

Main Memory in the System

Ideal Memory

- Zero access time (latency)
- Infinite capacity
- Zero cost
- Infinite bandwidth (to support multiple accesses in parallel)

The Problem

- **Ideal memory's requirements oppose each other**

- **Bigger is slower**
  - Bigger → Takes longer to determine the location

- **Faster is more expensive**
  - Memory technology: SRAM vs. DRAM

- **Higher bandwidth is more expensive**
  - Need more banks, more ports, higher frequency, or faster technology

Memory Technology: DRAM

- Dynamic random access memory

- Capacitor charge state indicates stored value
  - Whether the capacitor is charged or discharged indicates storage of 1 or 0
  - 1 capacitor
  - 1 access transistor

- Capacitor leaks through the RC path
  - DRAM cell loses charge over time
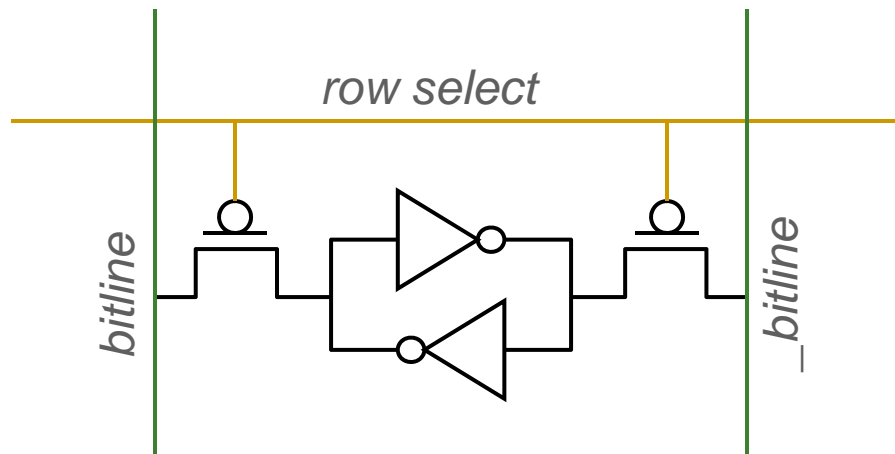 - DRAM cell needs to be refreshed
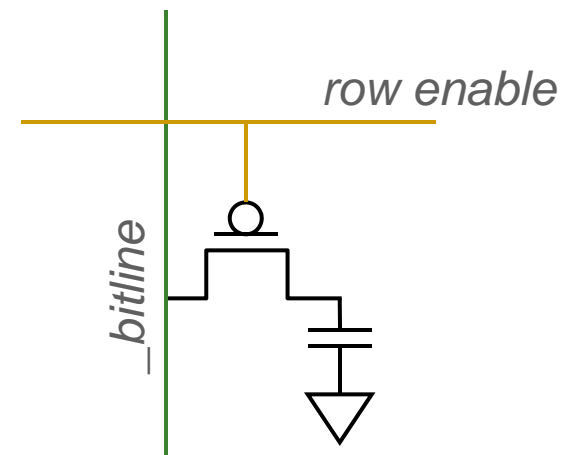
  - Read Liu et al., "RAIDR: Retention-aware Intelligent DRAM Refresh," ISCA 2012.

Memory Technology: SRAM

- Static random access memory
- Two cross coupled inverters store a single bit
  - Feedback path enables the stored value to persist in the "cell"
  - 4 transistors for storage
  - 2 transistors for access

An Aside: Phase Change Memory

- Phase change material (chalcogenide glass) exists in two states:
    - Amorphous: Low optical reflexivity and high electrical resistivity
    - Crystalline: High optical reflexivity and low electrical resistivity
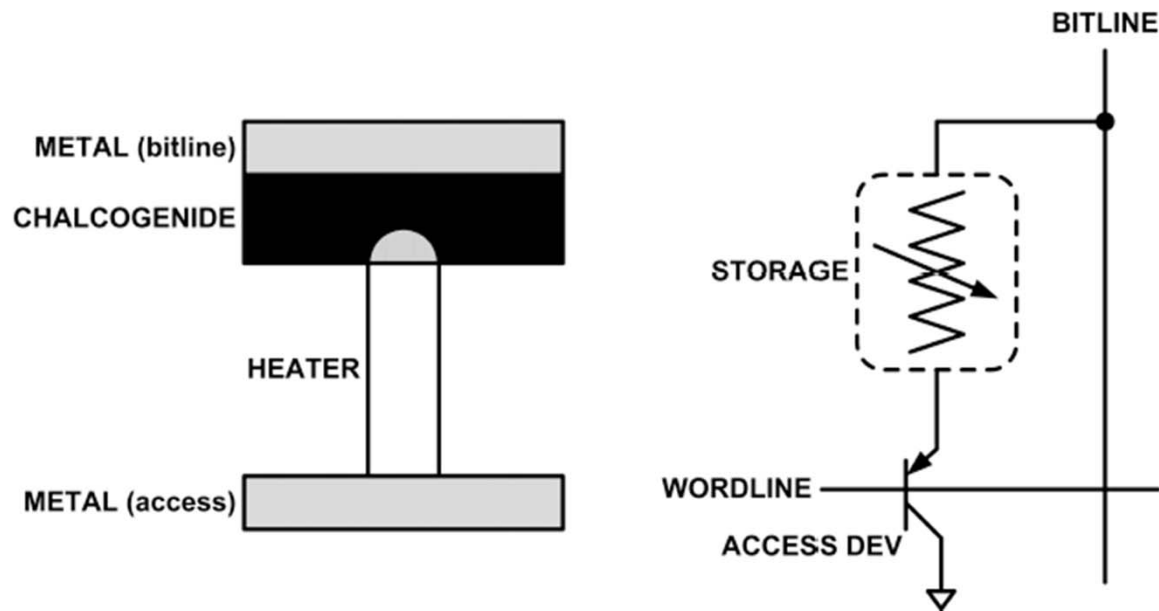
PCM is resistive memory: High resistance (0), Low resistance (1)

Lee, Ipek, Mutlu, Burger, "Architecting Phase Change Memory as a Scalable DRAM Alternative," ISCA 2009.

Memory Bank: A Fundamental Concept

- **Interleaving (banking)**
  - **Problem**: a single monolithic memory array takes long to access and does not enable multiple accesses in parallel

  - **Goal**: Reduce the latency of memory array access and enable multiple accesses in parallel

  - **Idea**: Divide the array into multiple banks that can be accessed independently (in the same cycle or in consecutive cycles)
    - Each bank is smaller than the entire memory storage
    - Accesses to different banks can be overlapped

  - **An issue**: How do you map data to different banks? (i.e., how do you interleave data across banks?)

Memory Bank Organization and Operation
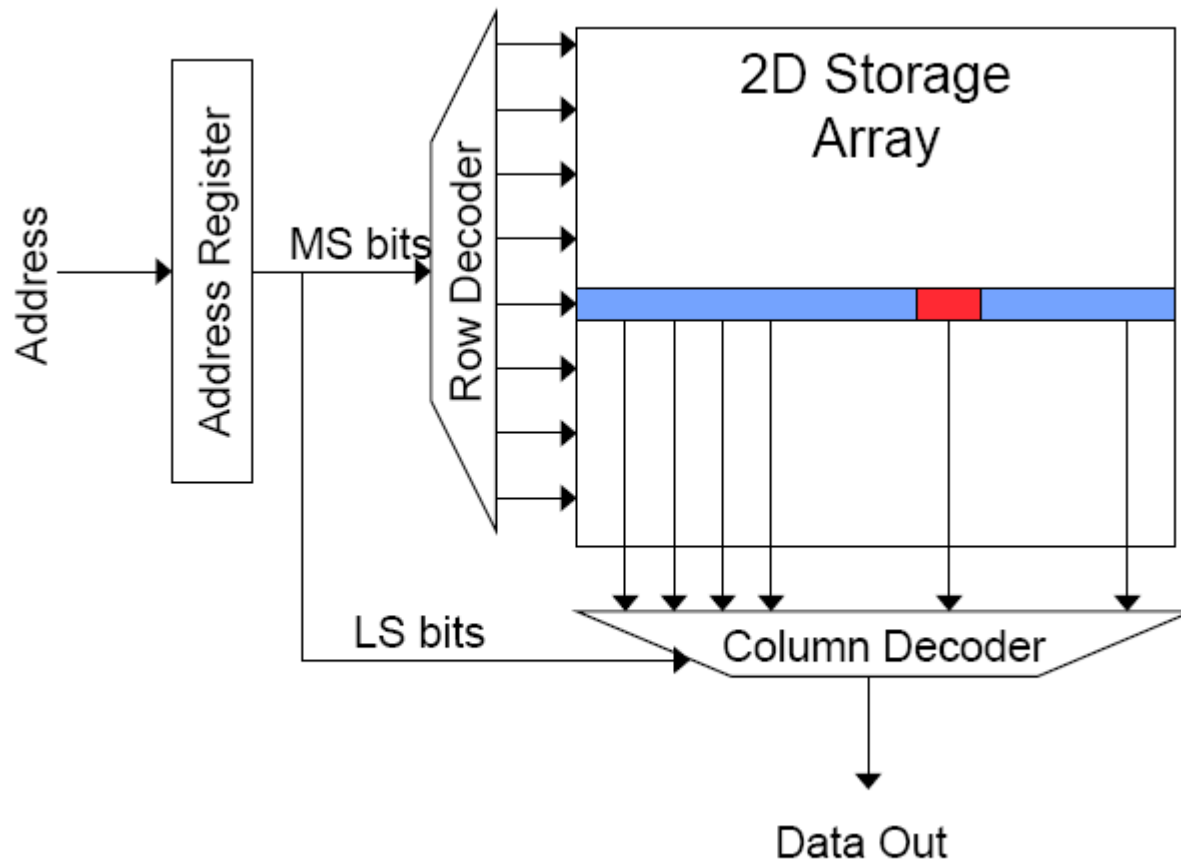
- Read access sequence:

  1. Decode row address & drive word-lines

  2. Selected bits drive bit-lines
     - Entire row read

  3. Amplify row data

  4. Decode column address & select subset of row
     - Send to output

  5. Precharge bit-lines
   - For next access

Why Memory Hierarchy?

- We want both fast and large

- But we cannot achieve both with a single level of memory

- Idea: Have multiple levels of storage (progressively bigger and slower as the levels are farther from the processor) and ensure most of the data the processor needs is kept in the fast(er) level(s)
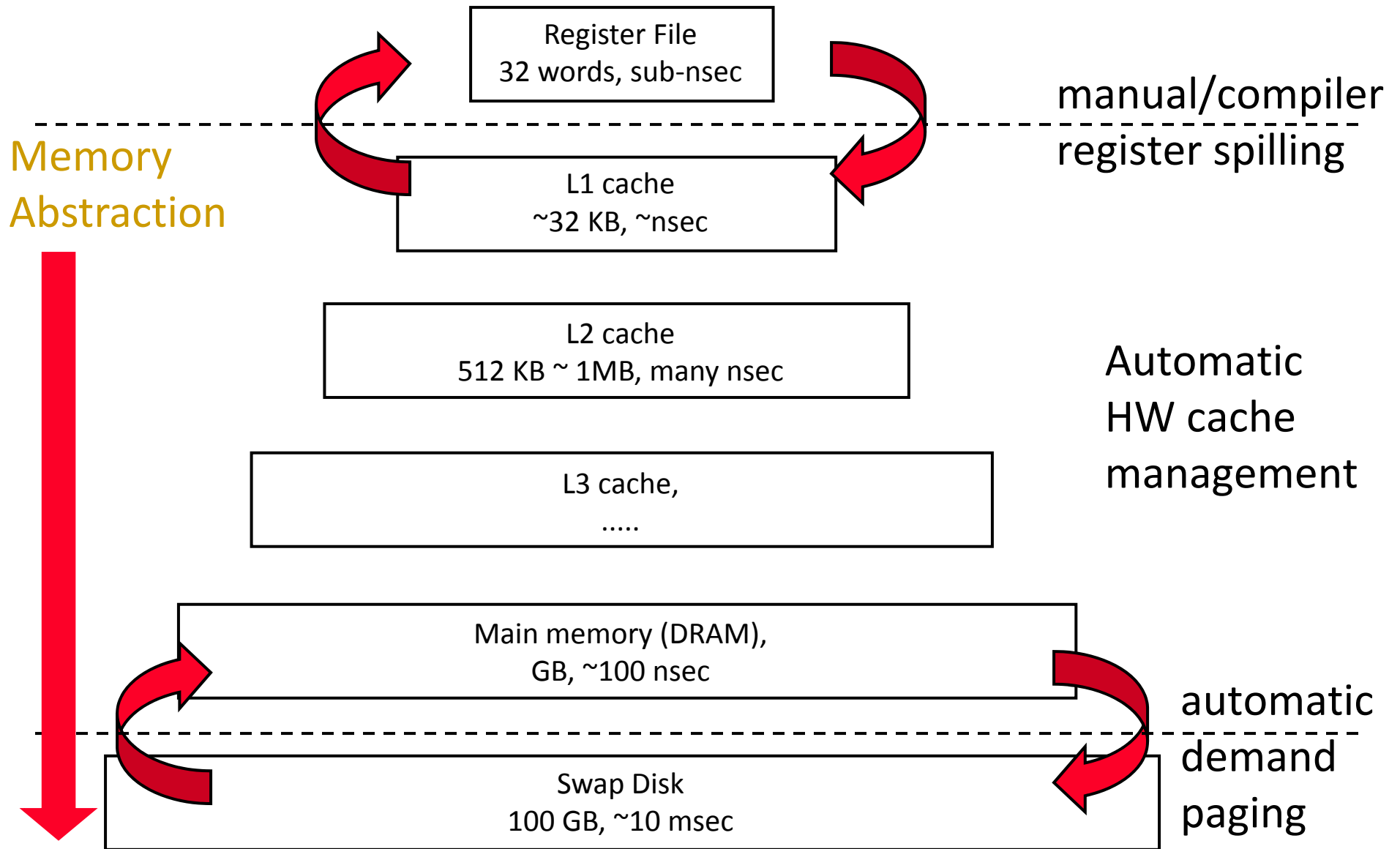
Memory Hierarchy

- Fundamental tradeoff
  - Fast memory: small
  - Large memory: slow
- Idea: Memory hierarchy
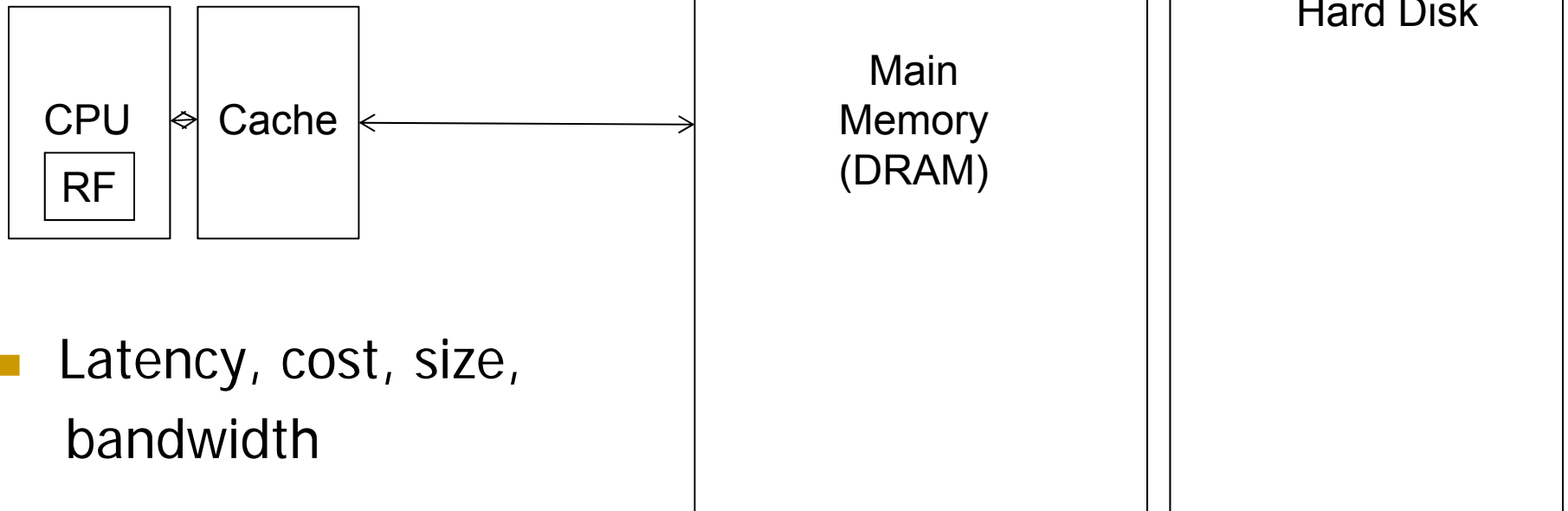
- Latency, cost, size, bandwidth

```
+-------------+    +---------+        +-----------+   +-----------+
|   CPU       |<-->|  Cache  |<------>|   Main    |   |           |
|   +------+  |    |         |        |  Memory   |   | Hard Disk |
|   |  RF  |  |    |         |        |  (DRAM)   |   |           |
|   +------+  |    |         |        |           |   |           |
+-------------+    +---------+        +-----------+   +-----------+
```

A Modern Memory Hierarchy

Memory
Abstraction

Register File
32 words, sub-nsec

manual/compiler
register spilling

L1 cache
~32 KB, ~nsec

L2 cache
512 KB ~ 1MB, many nsec

Automatic
HW cache
management

L3 cache,
.....

Main memory (DRAM),
GB, ~100 nsec
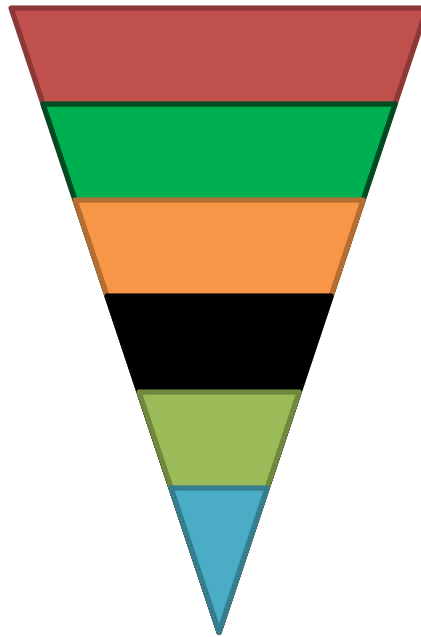
automatic
demand
paging

Swap Disk
100 GB, ~10 msec

13

The DRAM Subsystem

DRAM Subsystem Organization

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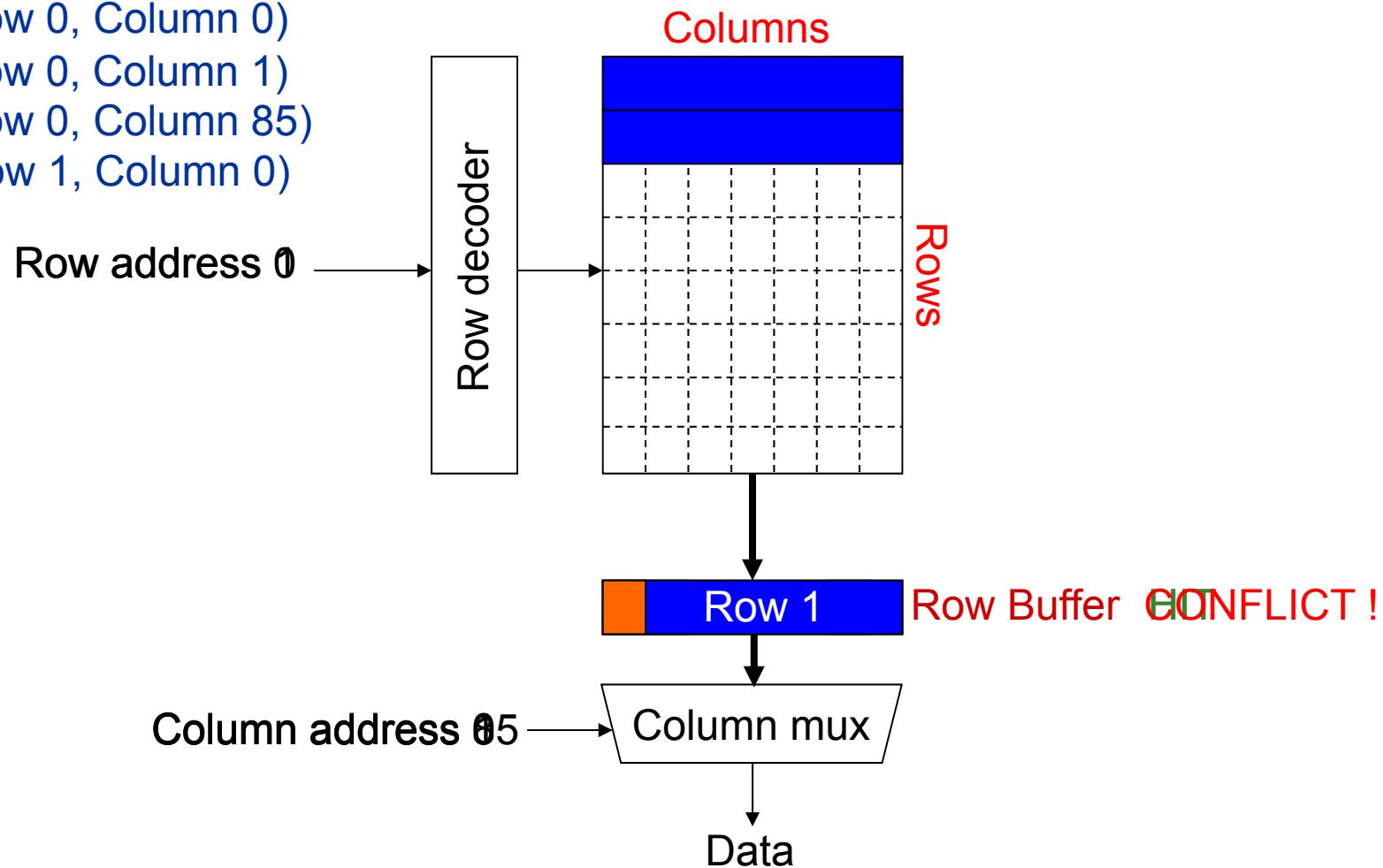- DIMM
- Rank
- Chip
- Bank
- Row/Column

Page Mode DRAM

- A DRAM bank is a 2D array of cells: rows x columns
- A "DRAM row" is also called a "DRAM page"
- "Sense amplifiers" also called "row buffer"

- Each address is a <row,column> pair
- Access to a "closed row"
  - Activate command opens row (placed into row buffer)
  - Read/write command reads/writes column in the row buffer
  - Precharge command closes the row and prepares the bank for next access
- Access to an "open row"
  - No need for activate command

DRAM Bank Operation

Access Address:
(Row 0, Column 0)
(Row 0, Column 1)
(Row 0, Column 85)
(Row 1, Column 0)

Columns

Row address 0 1 → Row decoder →

Rows

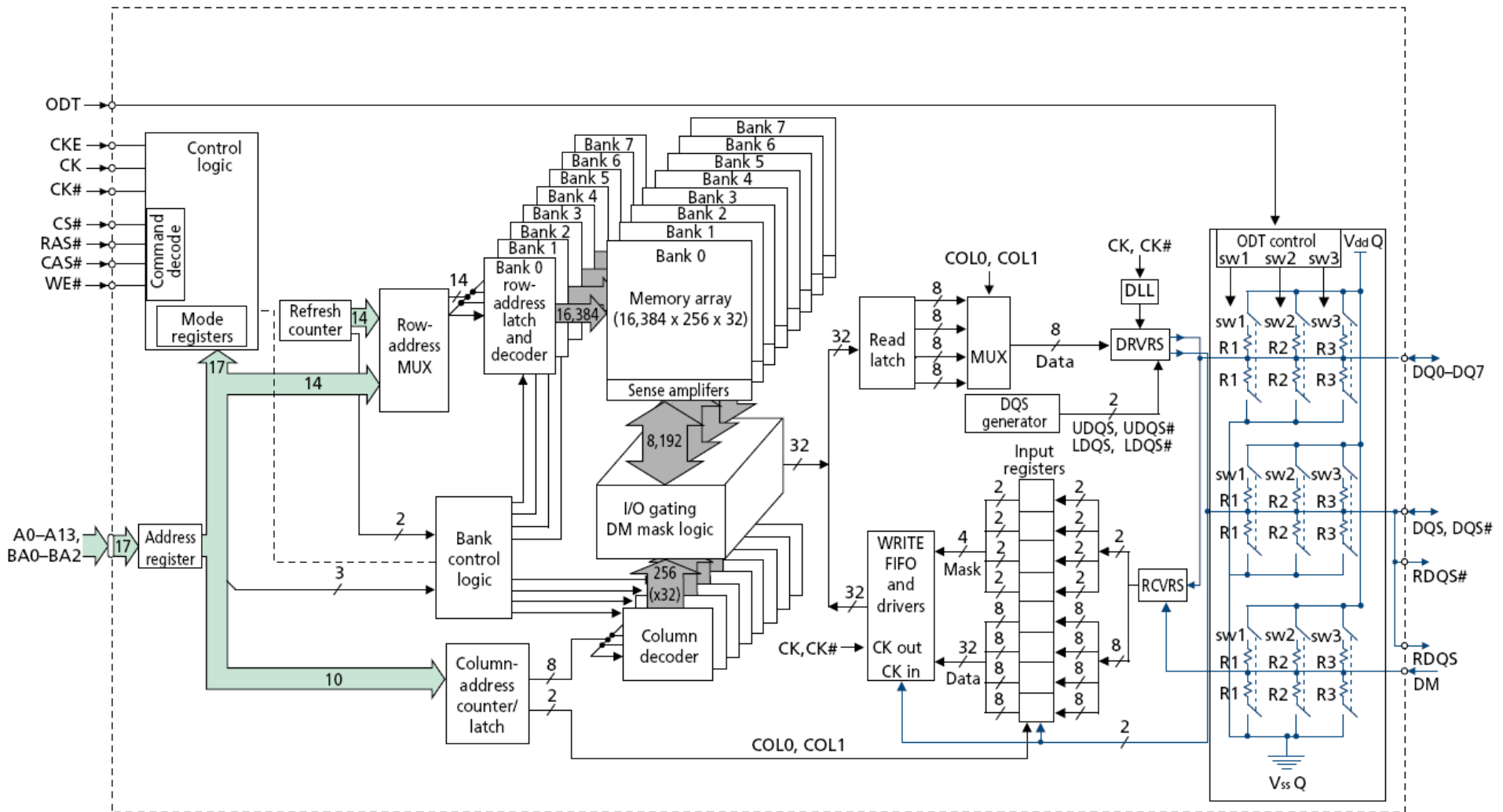Row 1 Row Buffer CONFLICT ! HIT

Column address 0 85 → Column mux

Data

The DRAM Chip

- Consists of multiple banks (2-16 in Synchronous DRAM)
- Banks share command/address/data buses
- The chip itself has a narrow interface (4-16 bits per read)

DRAM Rank and Module

- **Rank: Multiple chips operated together to form a wide interface**

- All chips comprising a rank are controlled at the same time
  - Respond to a single command
  - Share address and command buses, but provide different data

- A DRAM module consists of one or more ranks
  - E.g., DIMM (dual inline memory module)
  - This is what you plug into your motherboard

- If we have chips with 8-bit interface, to read 8 bytes in a single access, use 8 chips in a DIMM

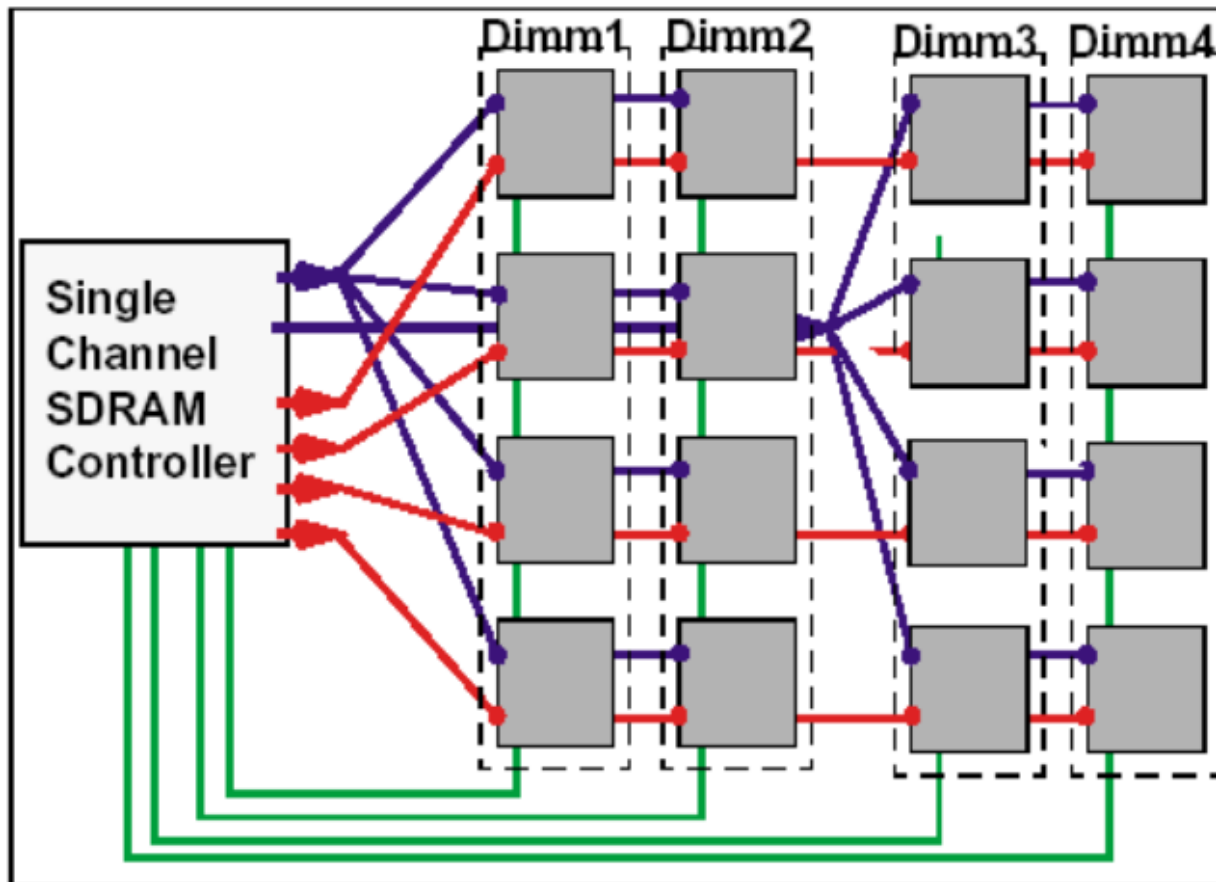# A 64-bit Wide DIMM (One Rank)
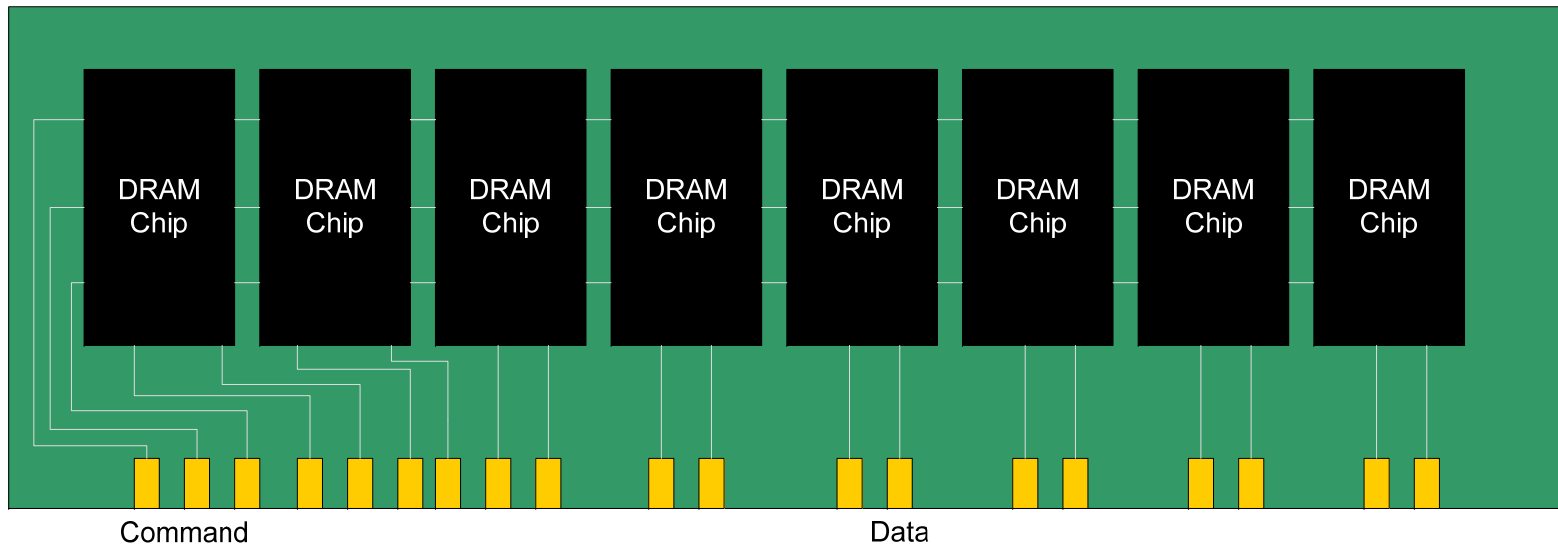
A 64-bit Wide DIMM (One Rank)
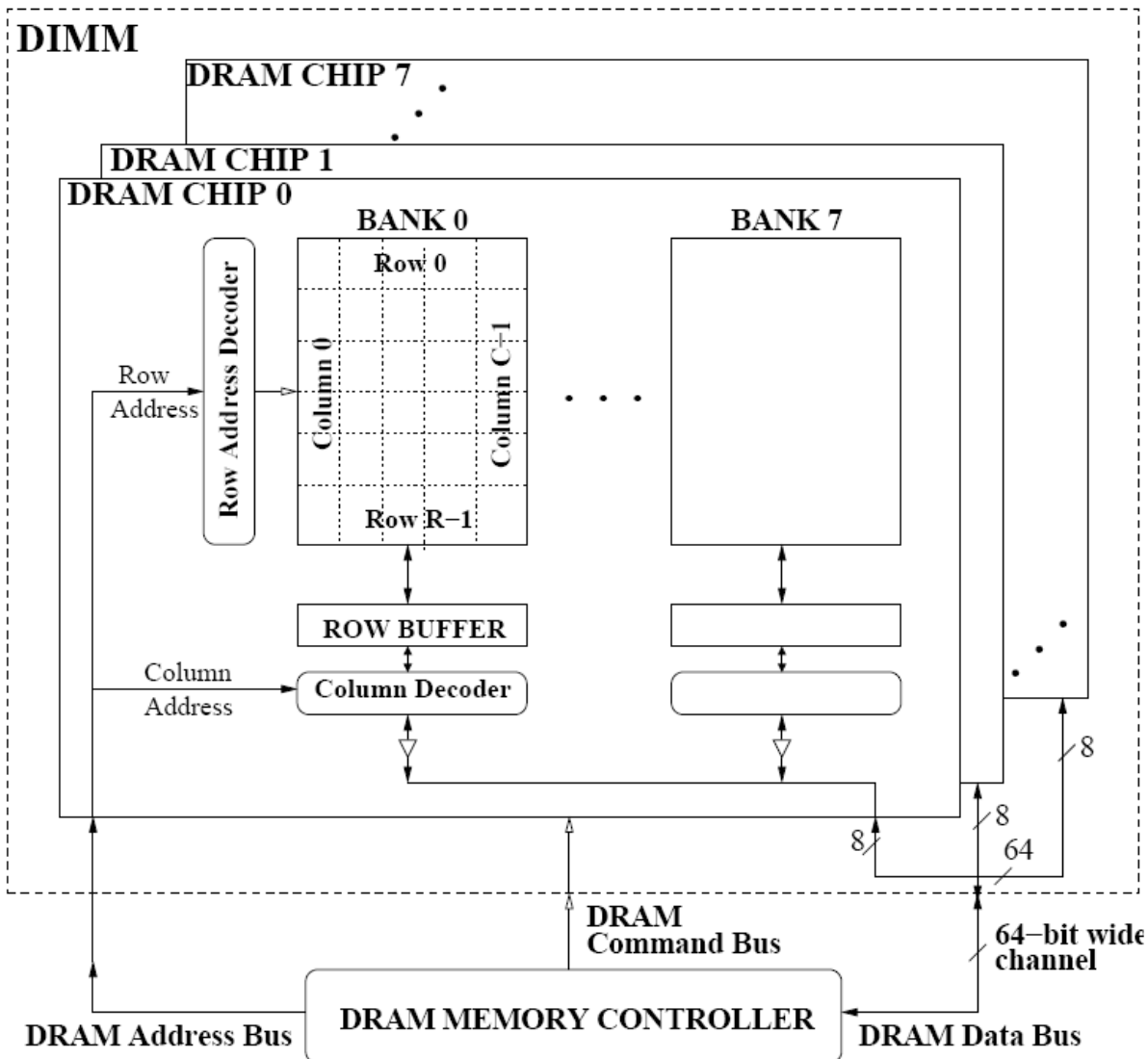
- **Advantages:**
  - Acts like a high-capacity DRAM chip with a wide interface
  - Flexibility: memory controller does not need to deal with individual chips

- **Disadvantages:**
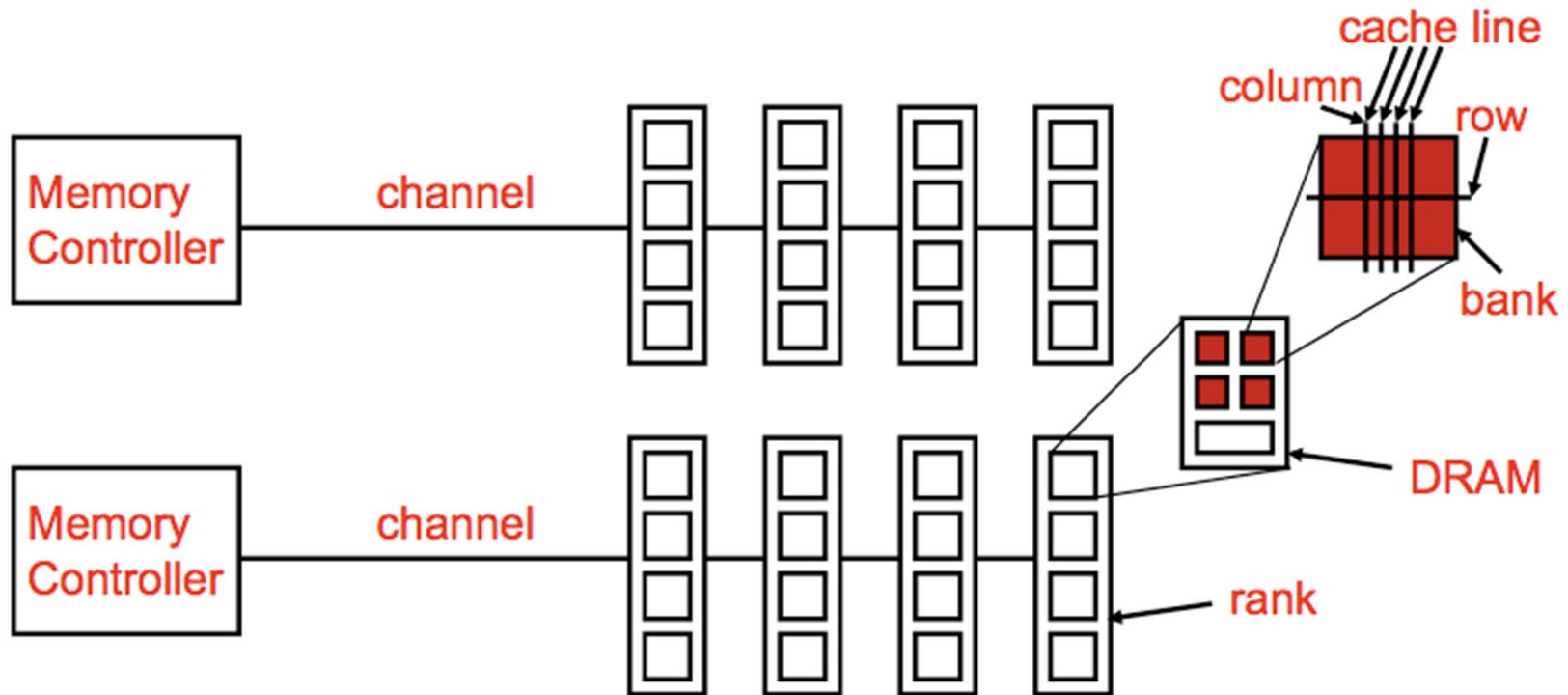 - Granularity: Accesses cannot be smaller than the interface width

Multiple DIMMs

- **Advantages:**
  - Enables even higher capacity

- **Disadvantages:**
  - Interconnect complexity and energy consumption can be high
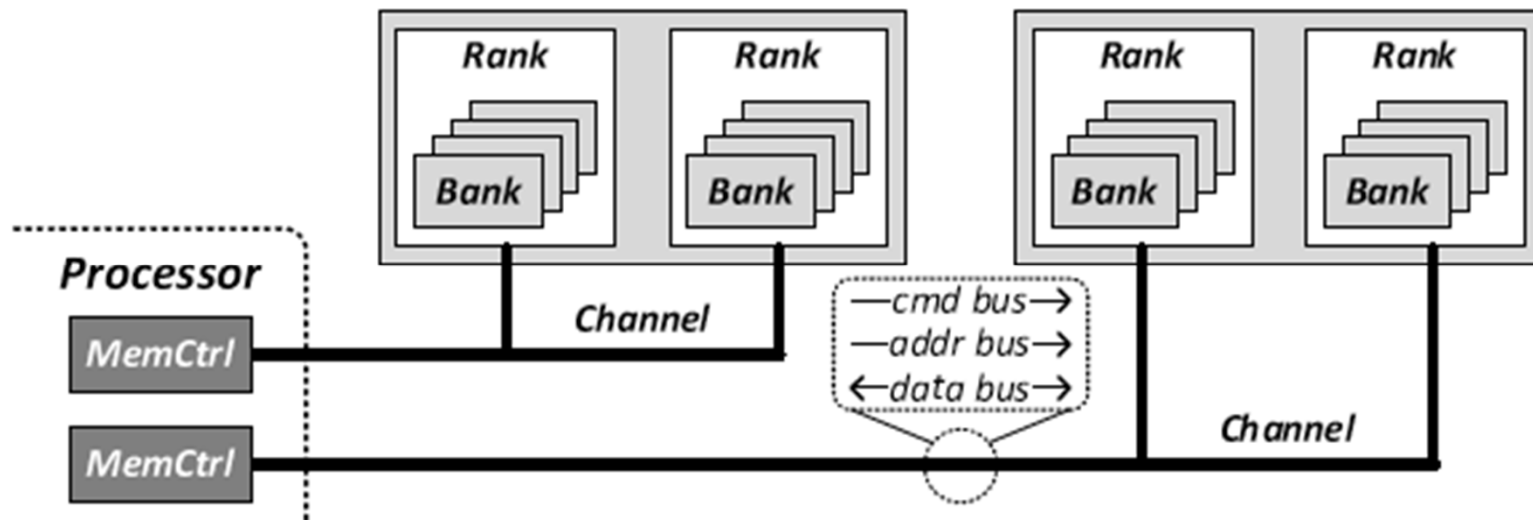
Diagram labels:

Dimm1 Dimm2 Dimm3 Dimm4

Single Channel SDRAM Controller

"Mesh Topology"

Addr & Cmd
Data Bus
Chip (DIMM) Select

DRAM Channels

- 2 Independent Channels: 2 Memory Controllers (Above)
- 2 Dependent/Lockstep Channels: 1 Memory Controller with wide interface (Not shown above)

Generalized Memory Structure

Generalized Memory Structure

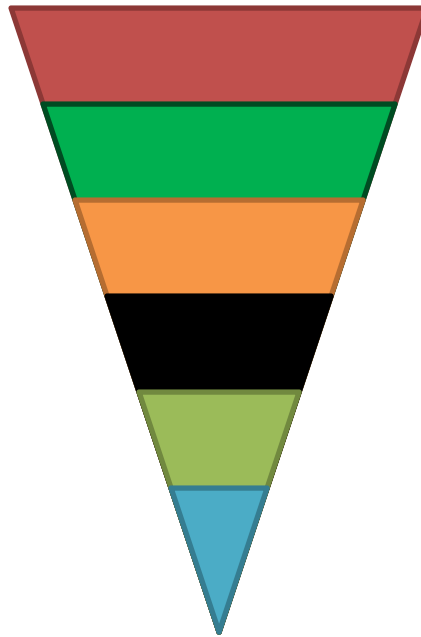Kim+, "A Case for Exploiting Subarray-Level Parallelism in DRAM," ISCA 2012.

The DRAM Subsystem
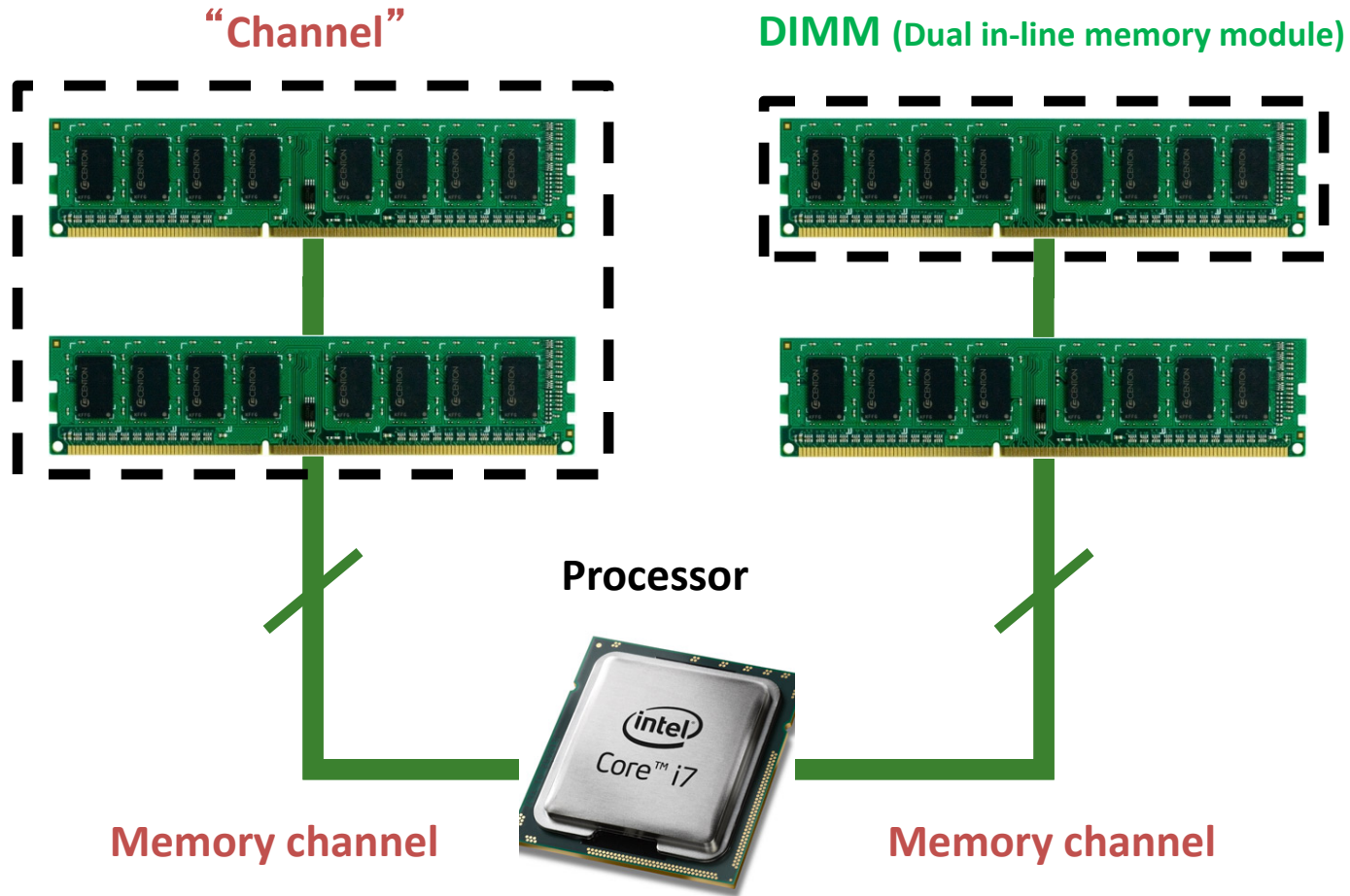# The Top Down View

DRAM Subsystem Organization
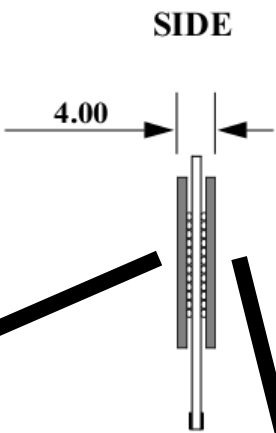
- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column

The DRAM subsystem

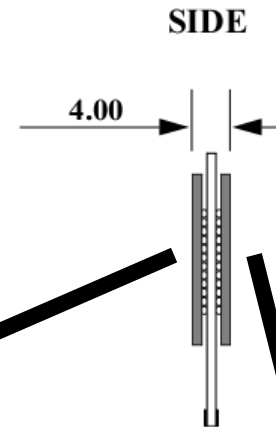# Breaking down a DIMM
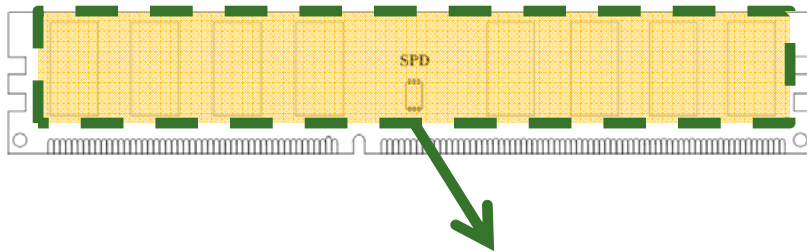
DIMM **(Dual in-line memory module)**

Side view

SIDE

4.00

Front of DIMM

SPD

Back of DIMM

Breaking down a DIMM

DIMM (Dual in-line memory module)

SIDE

4.00

Side view

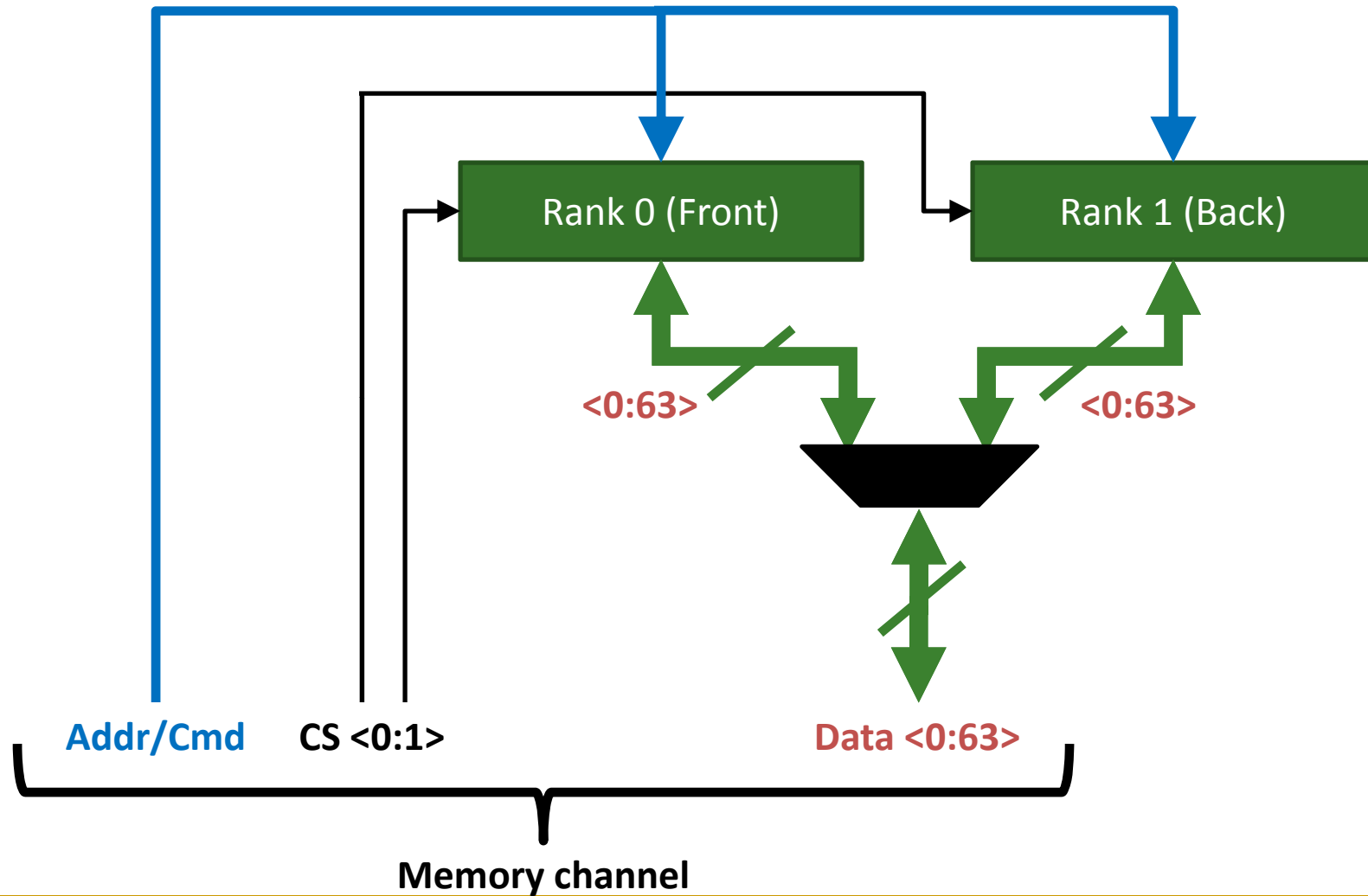**Front of DIMM**

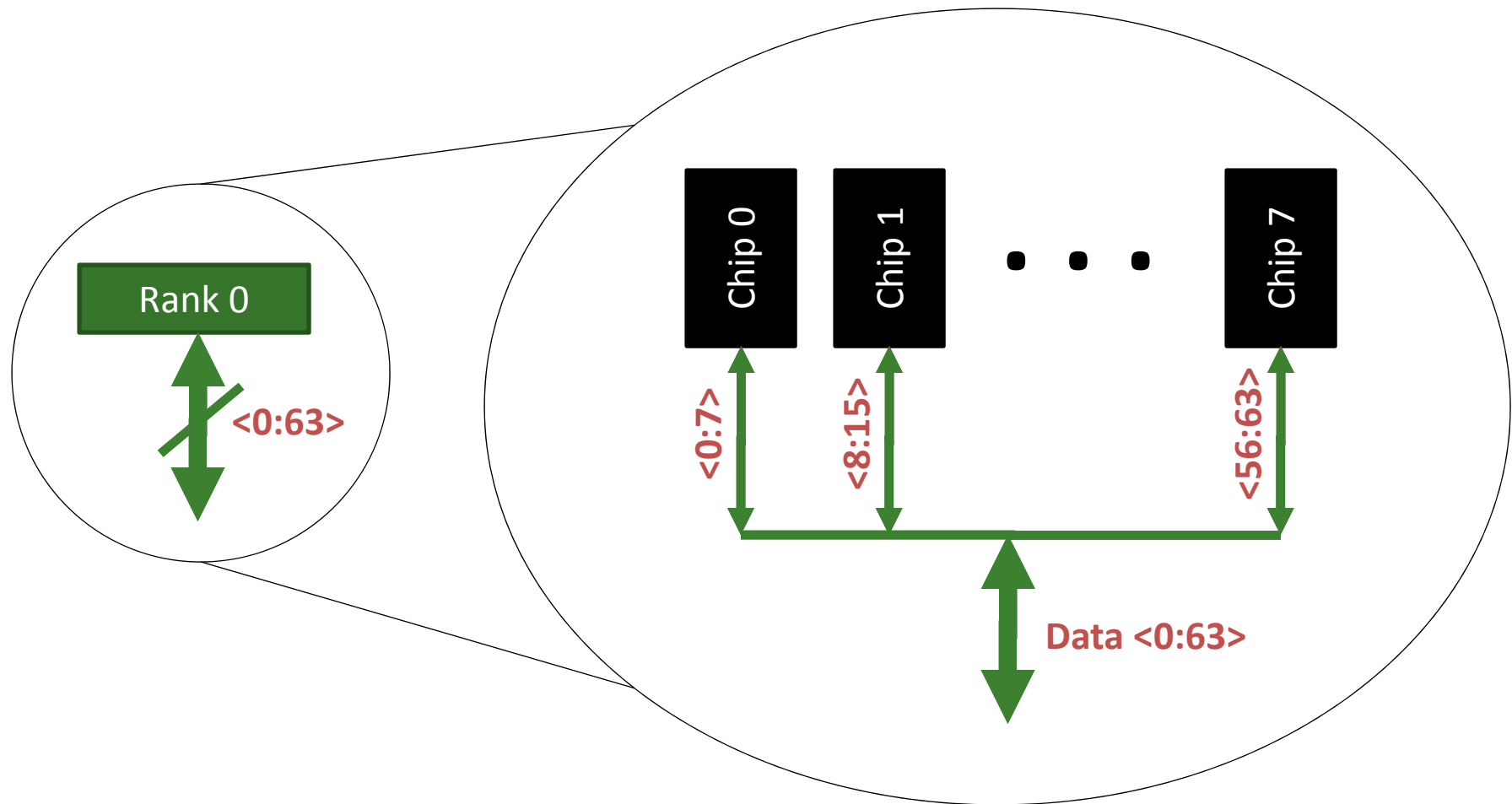**Back of DIMM**
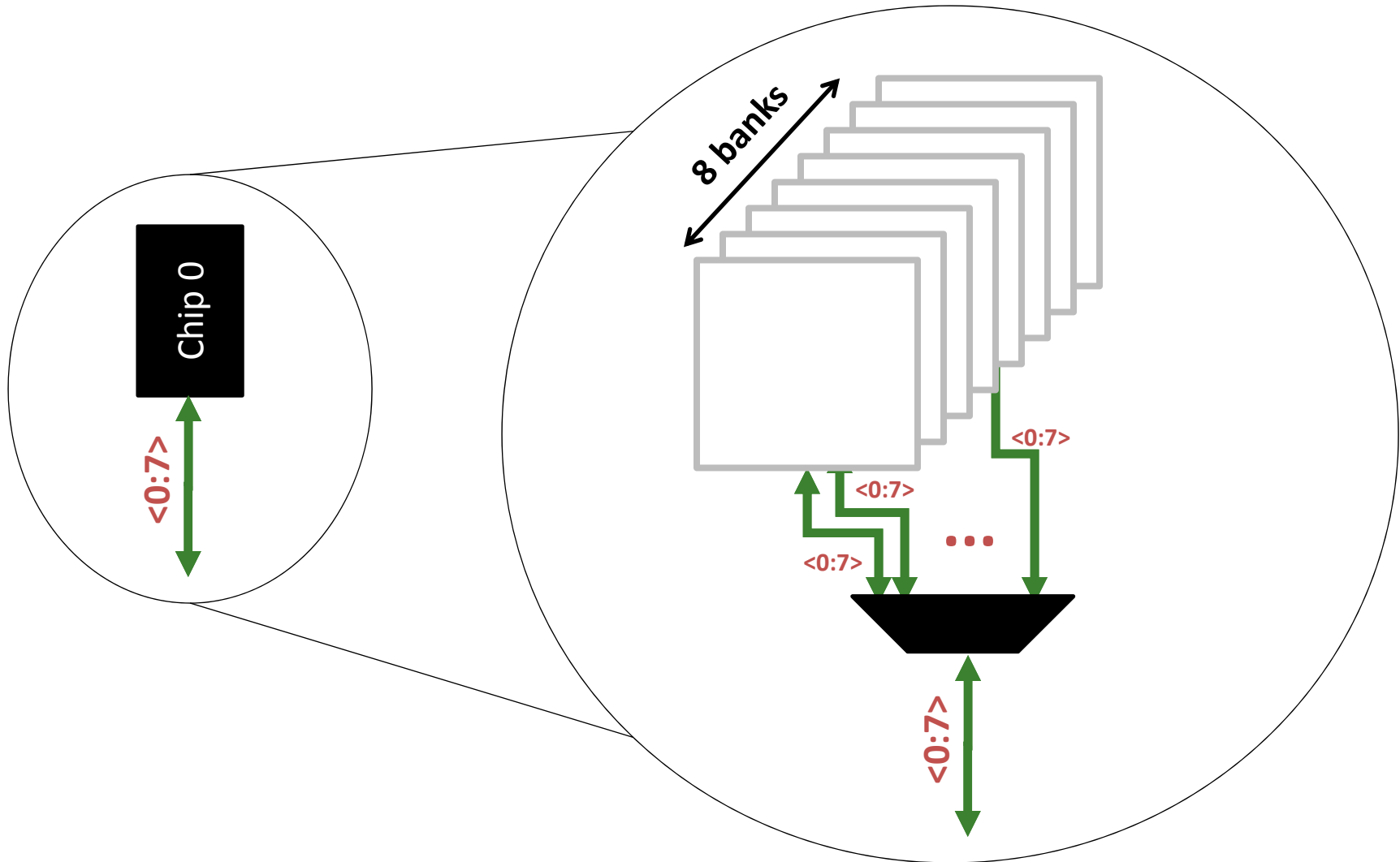
SPD

Rank 0: collection of 8 chips
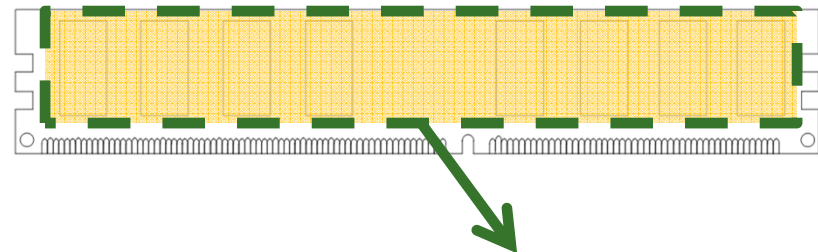
Rank 1

Rank

Breaking down a Rank

Breaking down a Chip

Breaking down a Bank
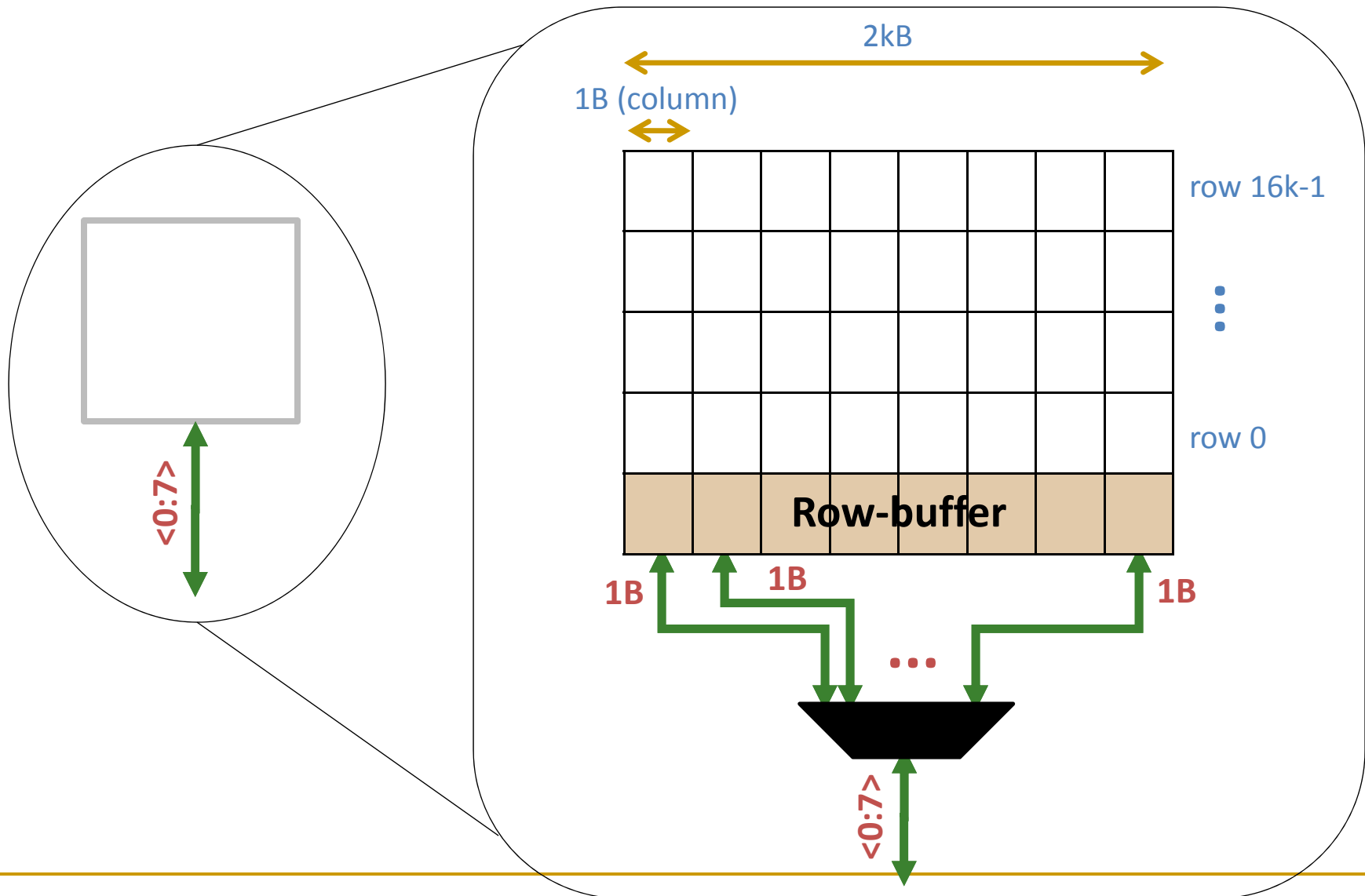
DRAM Subsystem Organization

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

Example: Transferring a cache block
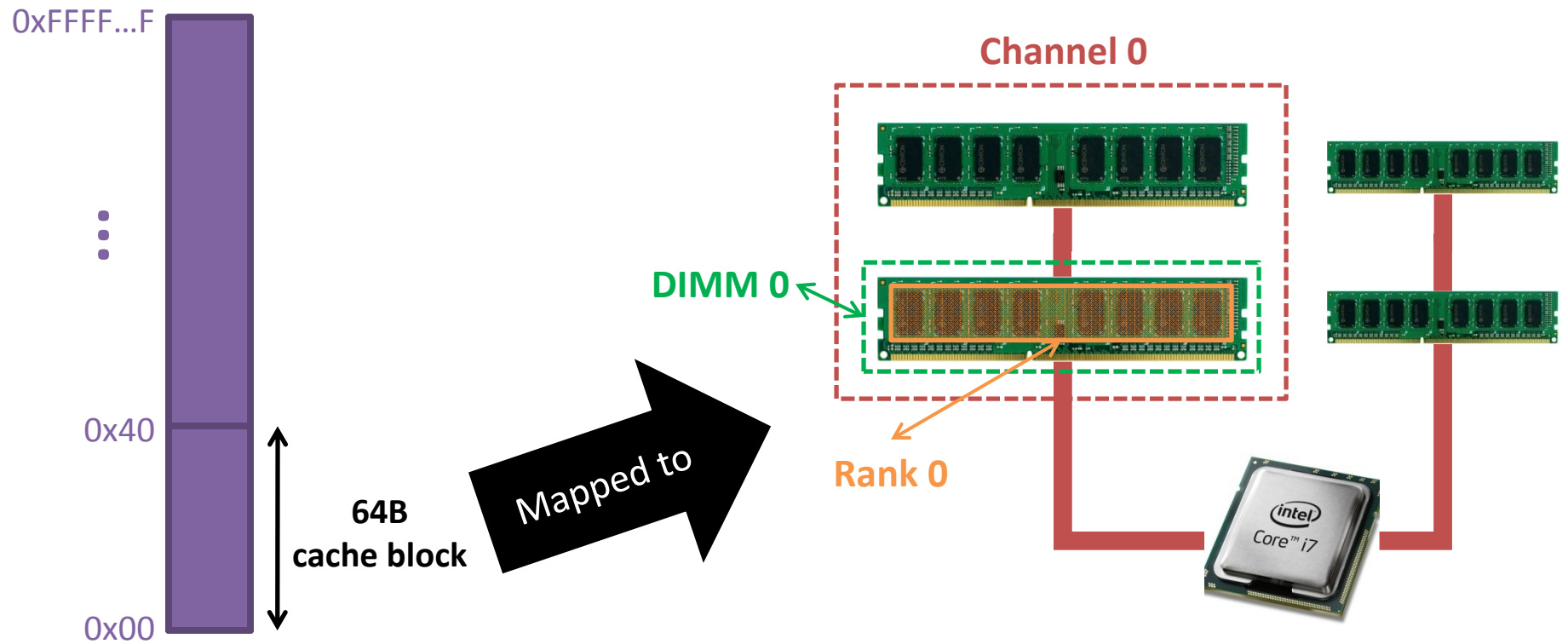
Example: Transferring a cache block

Example: Transferring a cache block

Physical memory space

0xFFFF...F

0x40

0x00

64B
cache block

Chip 0   Chip 1   **Rank 0**   Chip 7

Row 0
Col 0

. . .

<0:7>

<8:15>

<56:63>

Data <0:63>

Example: Transferring a cache block

Example: Transferring a cache block

Example: Transferring a cache block

Physical memory space

0xFFFF...F

0x40

0x00

8B

8B

64B cache block

Chip 0   Chip 1   Rank 0   Chip 7

Row 0
Col 1

. . .

<0:7>   <8:15>   <56:63>

Data <0:63>

8B
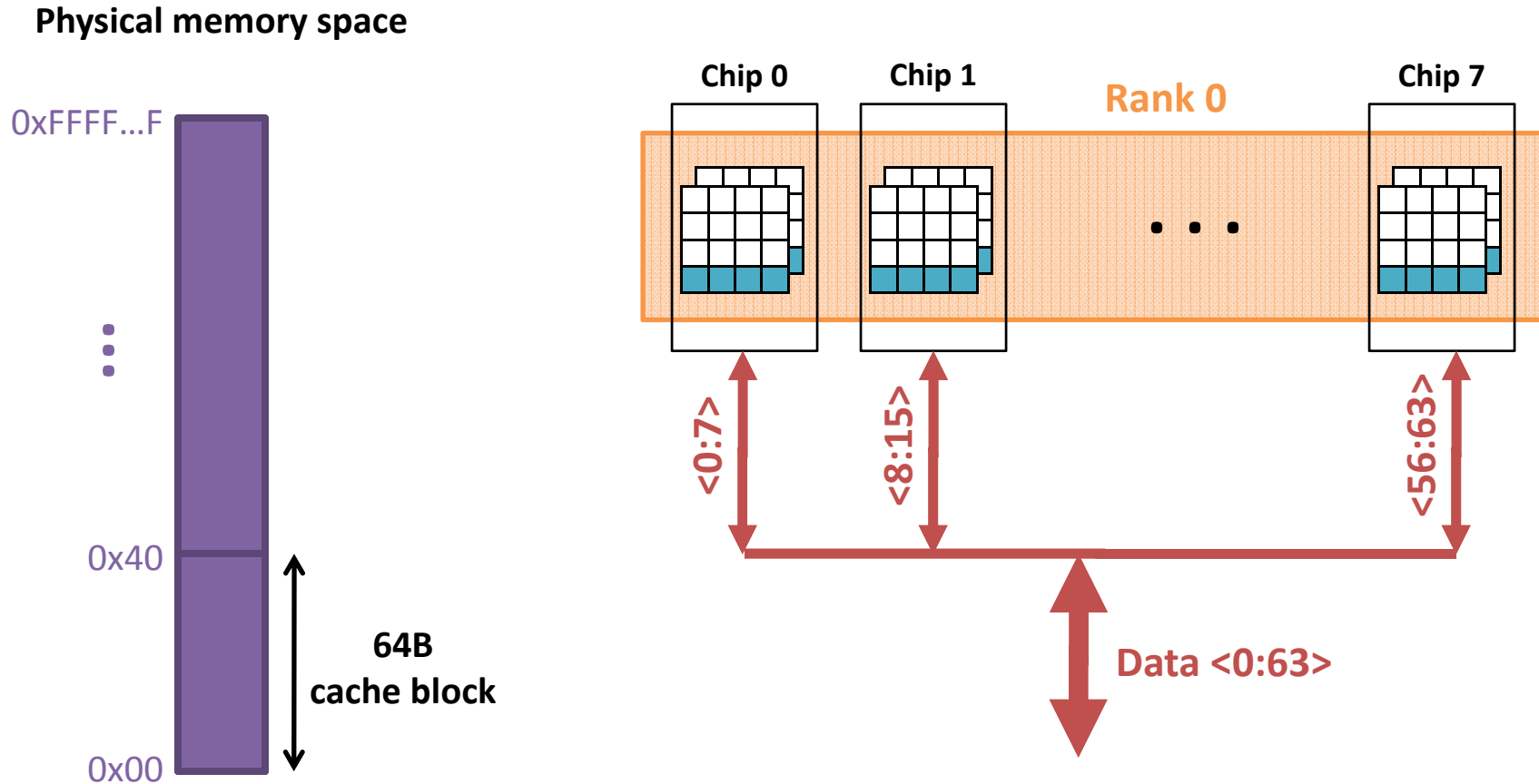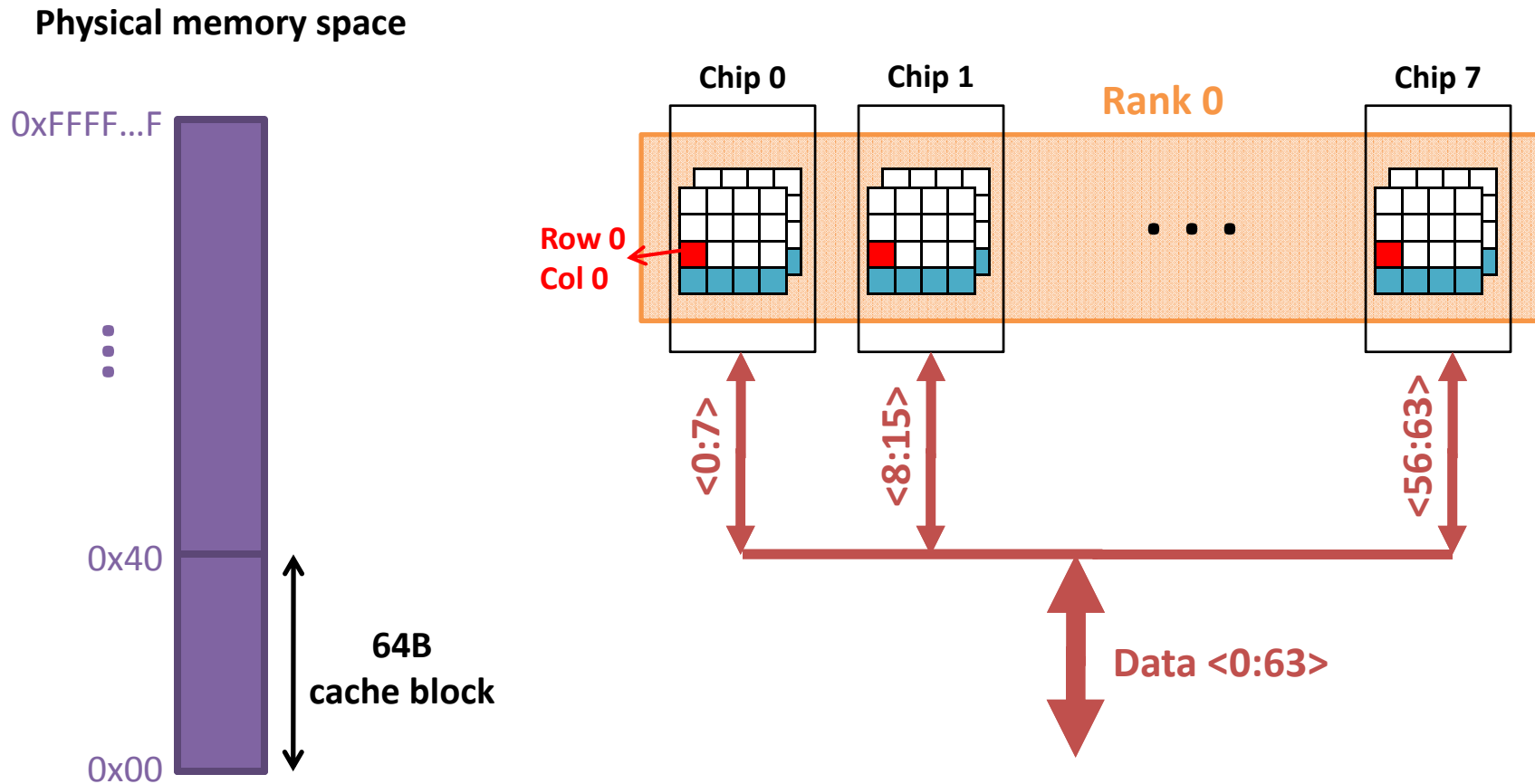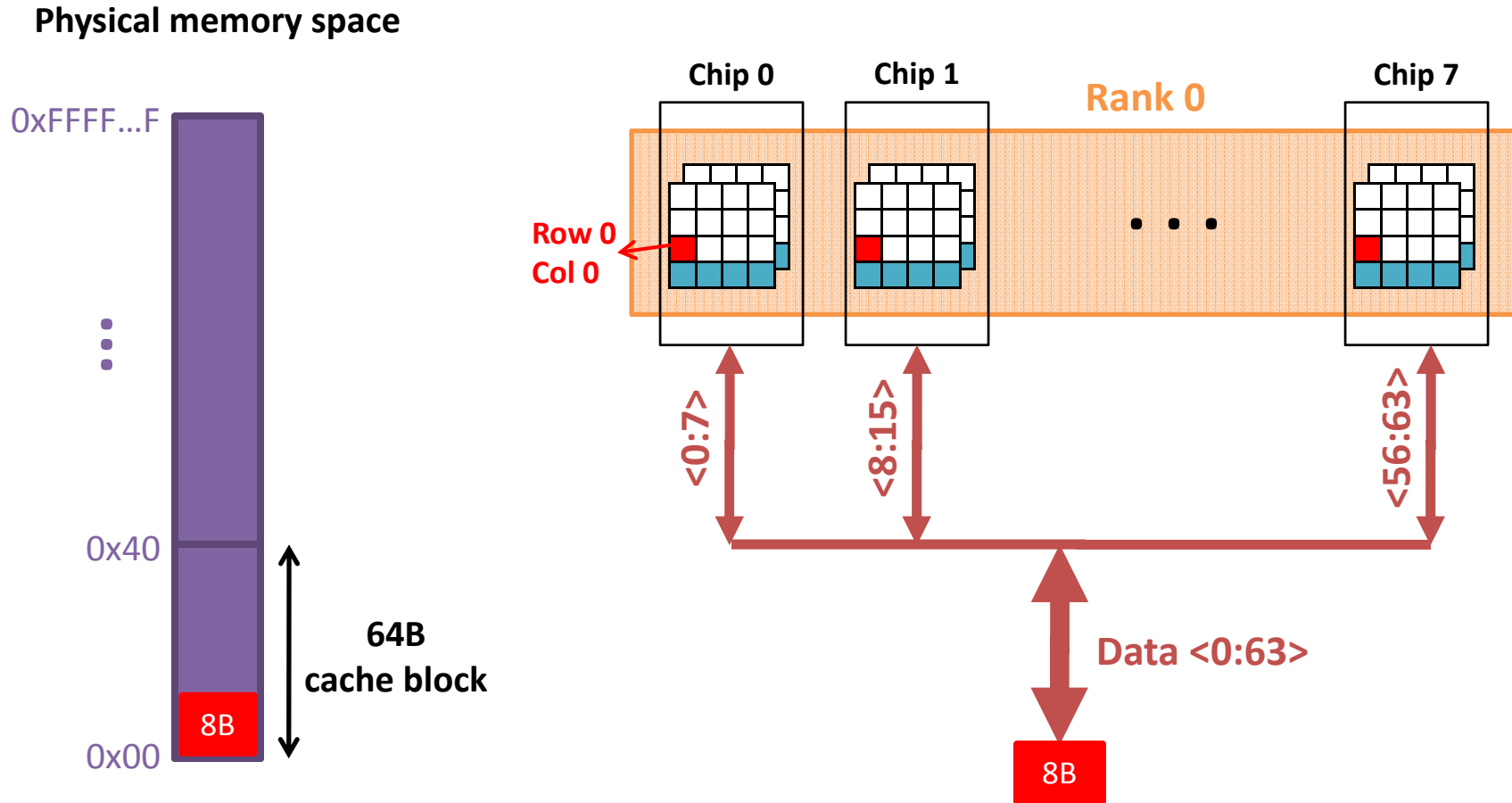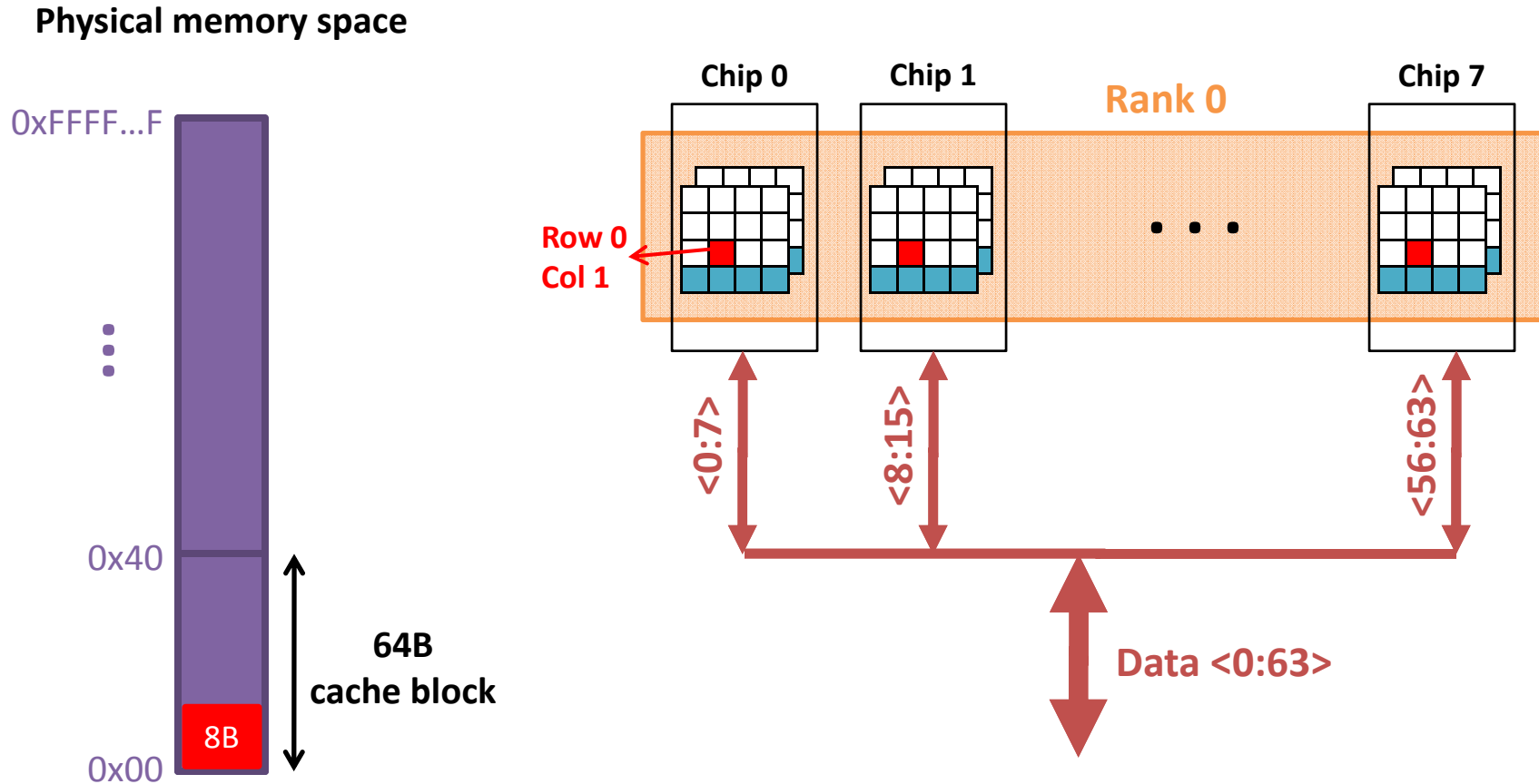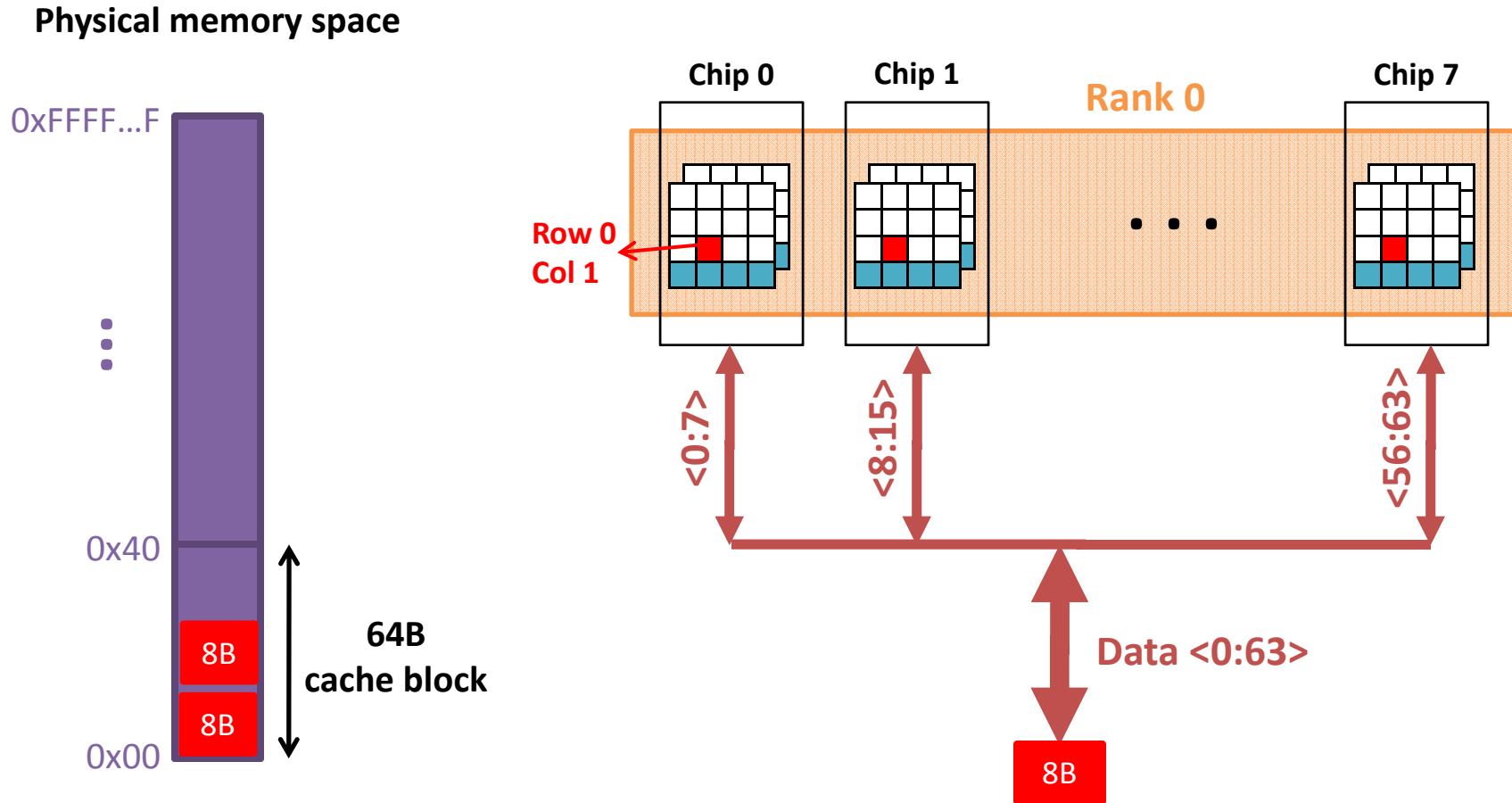
Example: Transferring a cache block
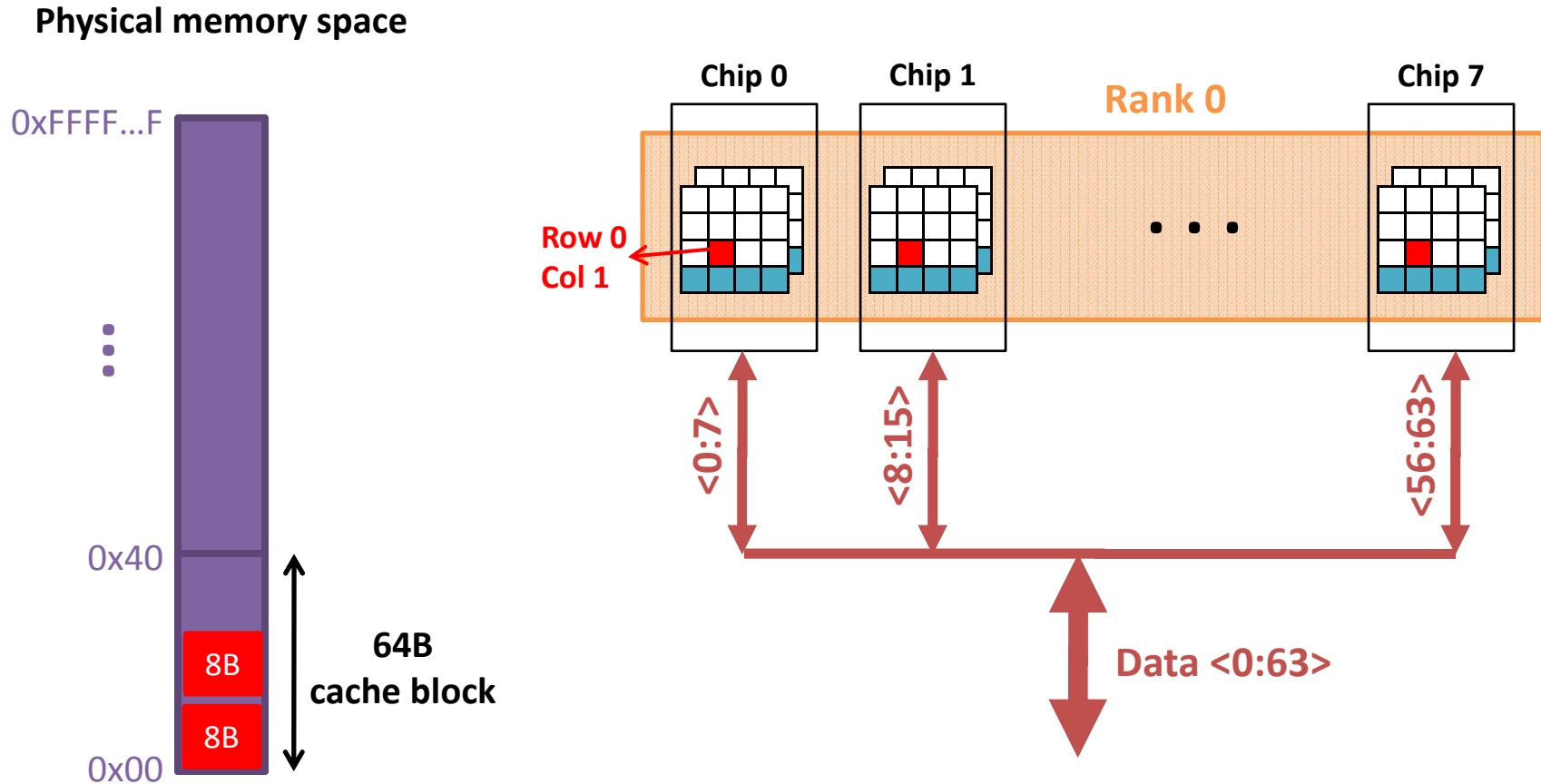
A 64B cache block takes 8 I/O cycles to transfer.

During the process, 8 columns are read sequentially.

Latency Components: Basic DRAM Operation

- CPU → controller transfer time

- Controller latency
  - Queuing & scheduling delay at the controller
  - Access converted to basic commands

- Controller → DRAM transfer time

- DRAM bank latency
  - Simple CAS (column address strobe) if row is "open" OR
  - RAS (row address strobe) + CAS if array precharged OR
  - PRE + RAS + CAS (worst case)

- DRAM → Controller transfer time
  - Bus latency (BL)

- Controller to CPU transfer time

Multiple Banks (Interleaving) and Channels

- **Multiple banks**
  - Enable concurrent DRAM accesses
  - Bits in address determine which bank an address resides in
- **Multiple independent channels serve the same purpose**
  - But they are even better because they have separate data buses
  - Increased bus bandwidth

- **Enabling more concurrency requires reducing**
  - Bank conflicts
  - Channel conflicts
- **How to select/randomize bank/channel indices in address?**
  - Lower order bits have more entropy
  - Randomizing hash functions (XOR of different address bits)

How Multiple Banks Help
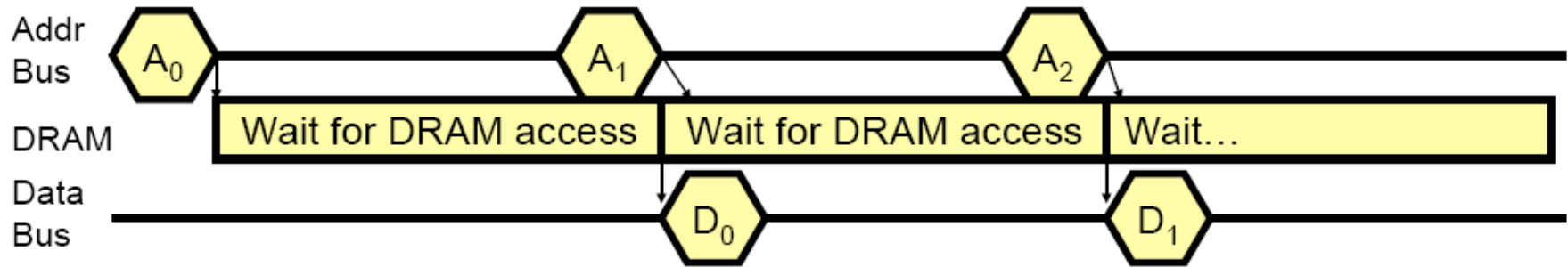
Before: No Overlapping
Assuming accesses to different DRAM rows
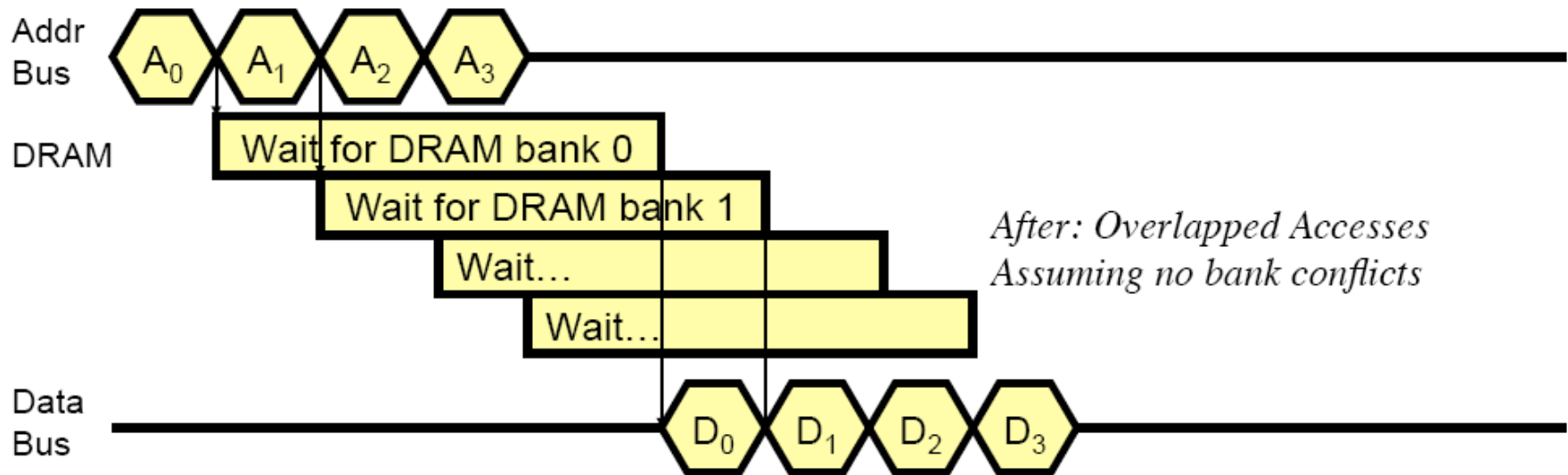
After: Overlapped Accesses
Assuming no bank conflicts

Address Mapping (Single Channel)

- **Single-channel system with 8-byte memory bus**
  - 2GB memory, 8 banks, 16K rows & 2K columns per bank

- **Row interleaving**
  - Consecutive rows of memory in consecutive banks

Row (14 bits)	Bank (3 bits)	Column (11 bits)	Byte in bus (3 bits)
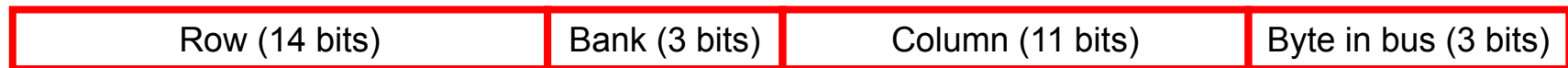
  - Accesses to consecutive cache blocks serviced in a pipelined manner

- **Cache block interleaving**
  - Consecutive cache block addresses in consecutive banks
  - 64 byte cache blocks

Row (14 bits)	High Column	Bank (3 bits)	Low Col.	Byte in bus (3 bits)
	8 bits		3 bits	
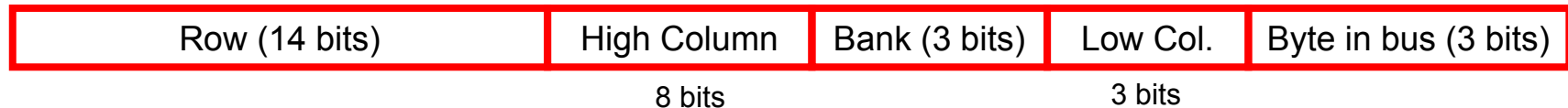
  - Accesses to consecutive cache blocks can be serviced in parallel

Bank Mapping Randomization

- DRAM controller can randomize the address mapping to banks so that bank conflicts are less likely

| | | 3 bits | Column (11 bits) | Byte in bus (3 bits) |

XOR

Bank index
(3 bits)

Address Mapping (Multiple Channels)

| C | Row (14 bits) | Bank (3 bits) | Column (11 bits) | Byte in bus (3 bits) |

| Row (14 bits) | C | Bank (3 bits) | Column (11 bits) | Byte in bus (3 bits) |

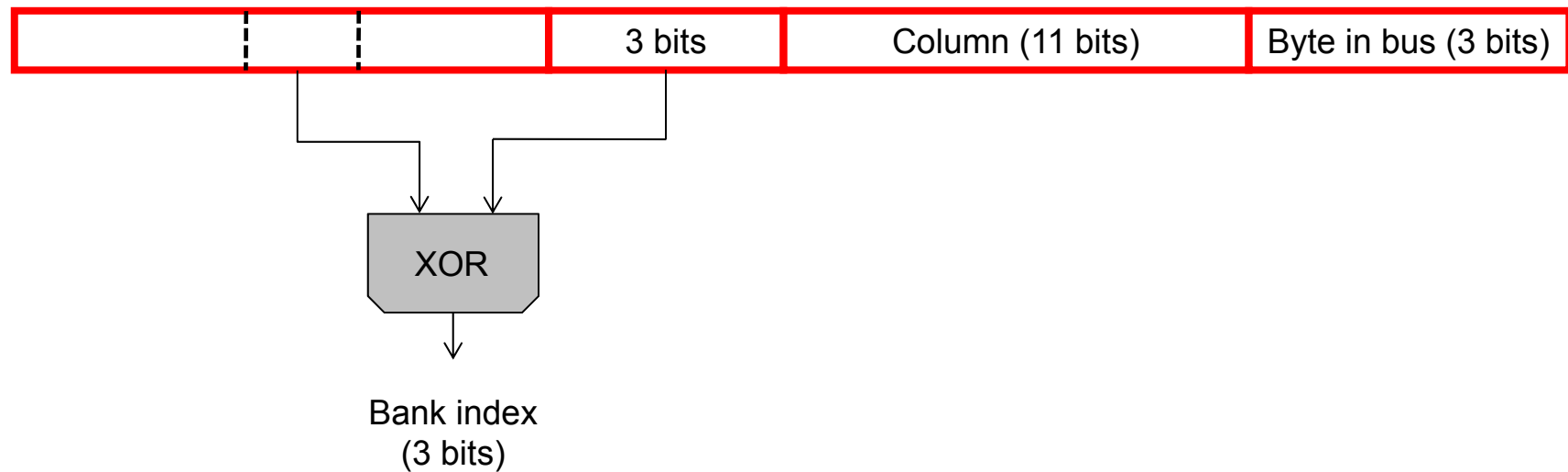| Row (14 bits) | Bank (3 bits) | C | Column (11 bits) | Byte in bus (3 bits) |

| Row (14 bits) | Bank (3 bits) | Column (11 bits) | C | Byte in bus (3 bits) |
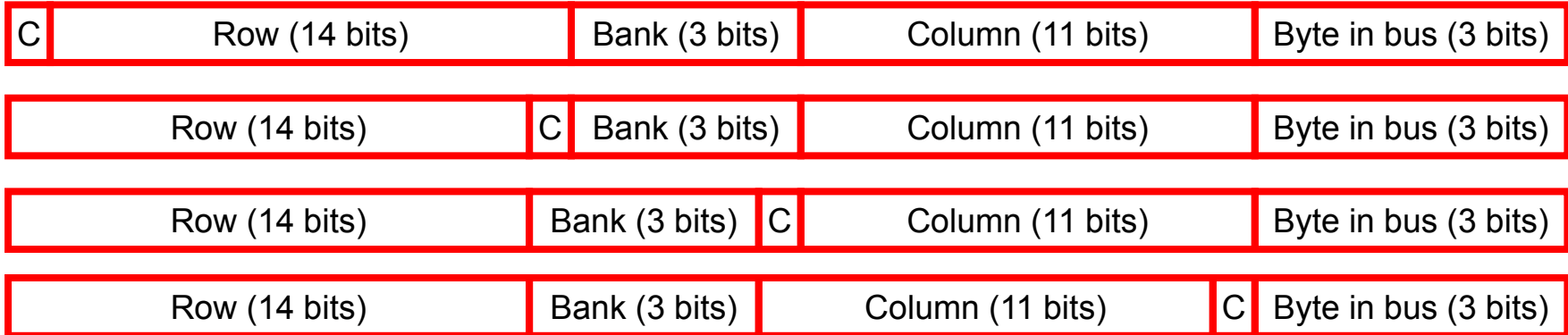
- **Where are consecutive cache blocks?**

| C | Row (14 bits) | High Column | Bank (3 bits) | Low Col. | Byte in bus (3 bits) |

8 bits   3 bits

| Row (14 bits) | C | High Column | Bank (3 bits) | Low Col. | Byte in bus (3 bits) |

8 bits   3 bits

| Row (14 bits) | High Column | C | Bank (3 bits) | Low Col. | Byte in bus (3 bits) |

8 bits   3 bits

| Row (14 bits) | High Column | Bank (3 bits) | C | Low Col. | Byte in bus (3 bits) |

8 bits   3 bits

| Row (14 bits) | High Column | Bank (3 bits) | Low Col. | C | Byte in bus (3 bits) |

8 bits   3 bits
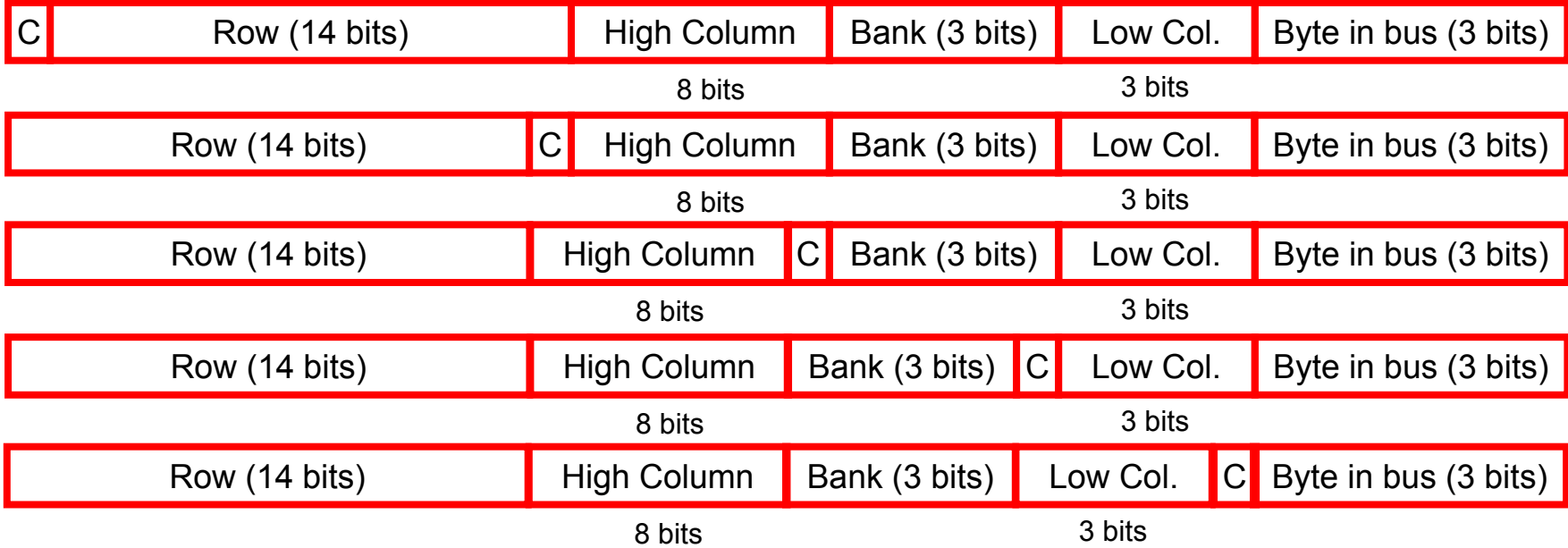
Interaction with Virtual➔Physical Mapping

- Operating System influences where an address maps to in DRAM

Virtual Page number (52 bits)		Page offset (12 bits)	VA

Physical Frame number (19 bits)		Page offset (12 bits)	PA

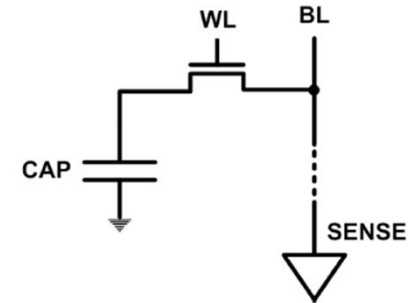| Row (14 bits)	Bank (3 bits)	Column (11 bits)	Byte in bus (3 bits)	PA
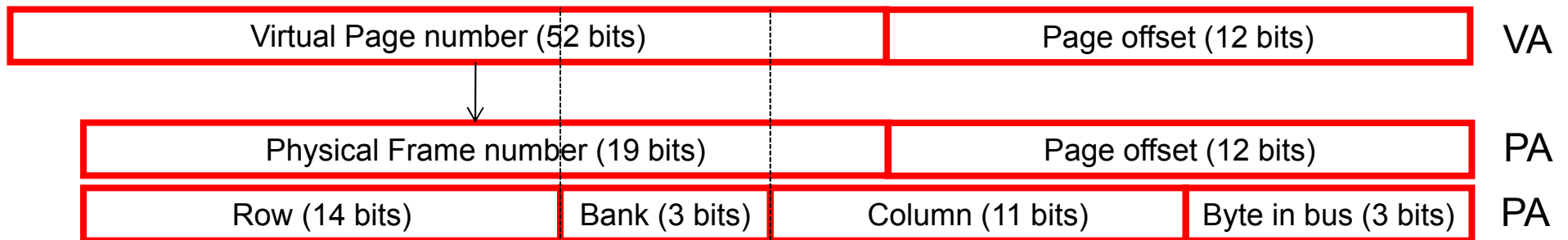
- Operating system can influence which bank/channel/rank a virtual page is mapped to.

- It can perform page coloring to
  - Minimize bank conflicts
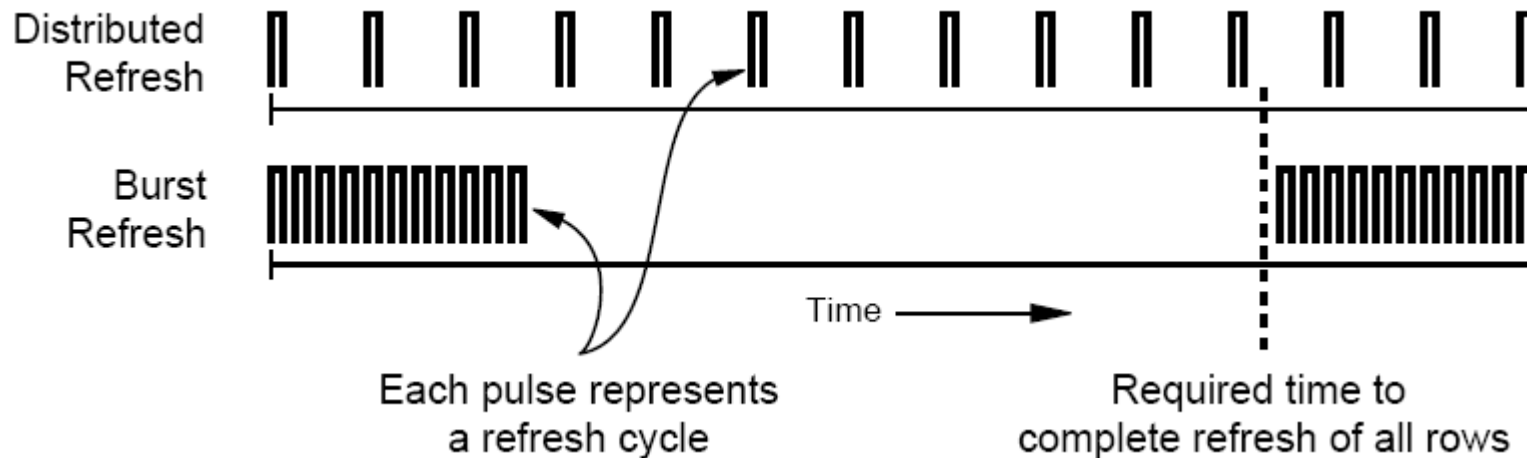 - Minimize inter-application interference **[Muralidhara+ MICRO'11]**

DRAM Refresh (I)

- DRAM capacitor charge leaks over time

- The memory controller needs to read each row periodically to restore the charge

  - Activate + precharge each row every N ms
  - Typical N = 64 ms

- Implications on performance?

  -- DRAM bank unavailable while refreshed

  -- Long pause times: If we refresh all rows in burst, every 64ms the DRAM will be unavailable until refresh ends

- Burst refresh: All rows refreshed immediately after one another

- Distributed refresh: Each row refreshed at a different time, at regular intervals

DRAM Refresh (II)

Distributed Refresh

Burst Refresh

Each pulse represents a refresh cycle

Time →

Required time to complete refresh of all rows

- Distributed refresh eliminates long pause times
- How else we can reduce the effect of refresh on performance?
  - Can we reduce the number of refreshes?

Downsides of DRAM Refresh

-- Energy consumption: Each refresh consumes energy

-- Performance degradation: DRAM rank/bank unavailable while refreshed

-- QoS/predictability impact: (Long) pause times during refresh
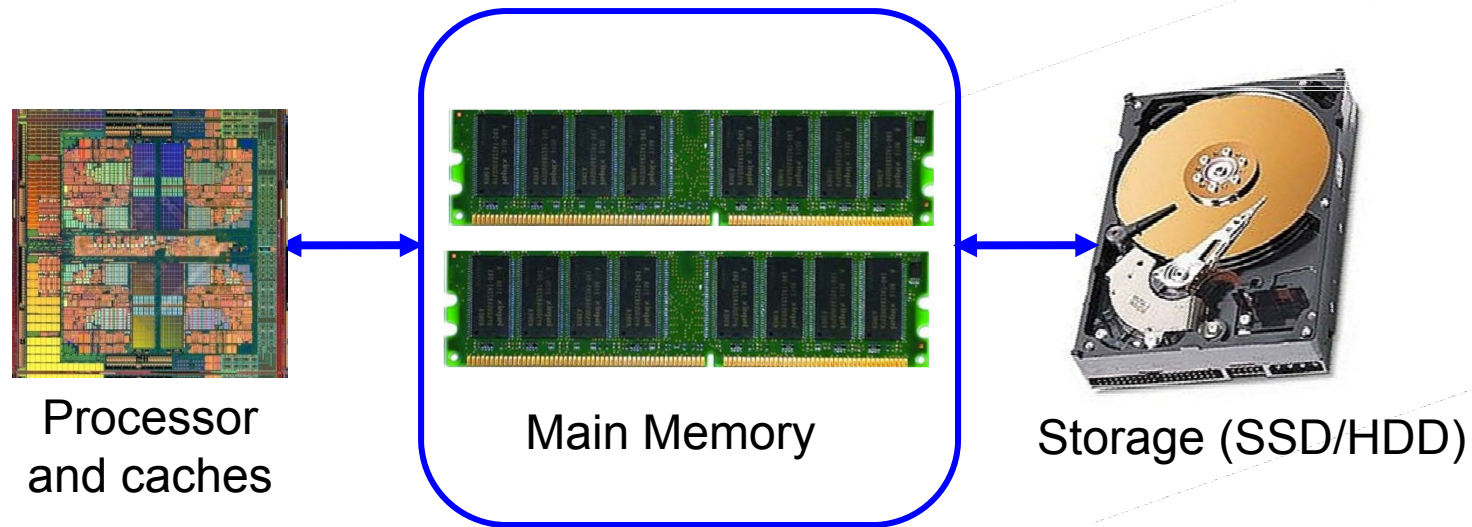
-- Refresh rate limits DRAM density scaling
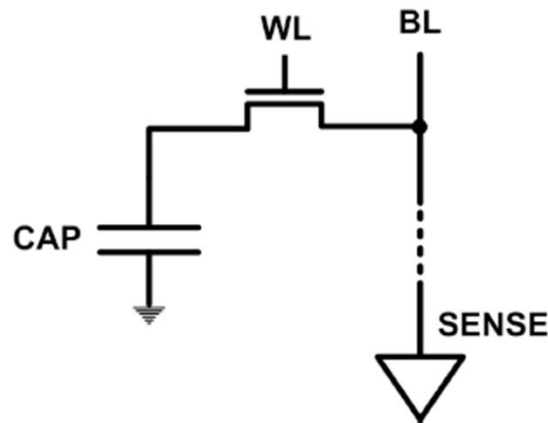
Liu et al., "RAIDR: Retention-aware Intelligent DRAM Refresh," ISCA 2012.

Trends

The Main Memory System

Processor and caches   ←→   Main Memory   ←→   Storage (SSD/HDD)

- **Main memory is a critical component of all computing systems**: server, mobile, embedded, desktop, sensor

- **Main memory system must scale** (in *size*, *technology*, *efficiency*, *cost*, and *management algorithms*) to maintain performance growth and technology scaling benefits

SAFARI

Memory System: A *Shared Resource* View
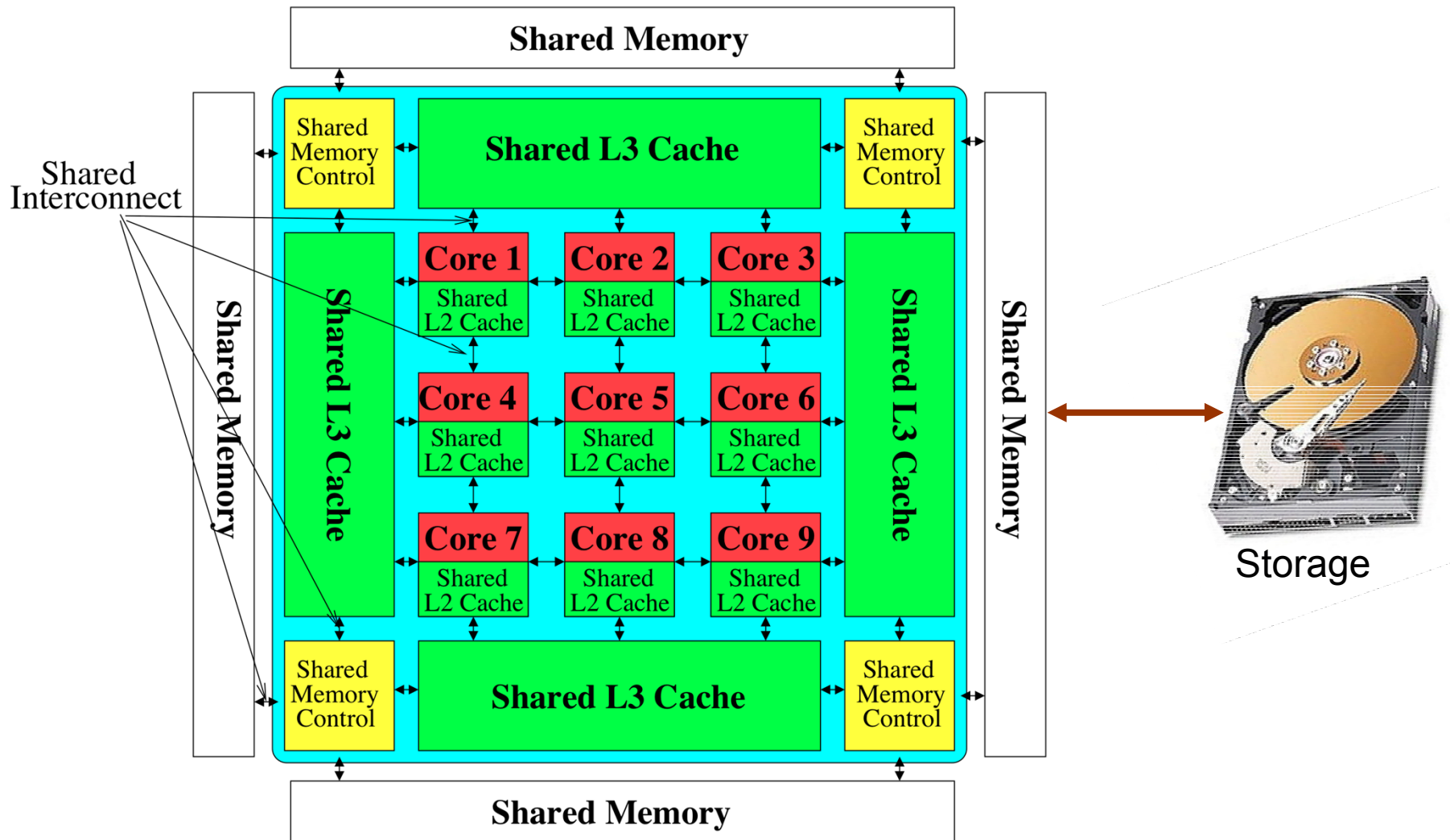
State of the Main Memory System

- Recent technology, architecture, and application trends
  - lead to new requirements
  - exacerbate old requirements

- DRAM and memory controllers, as we know them today, are (will be) unlikely to satisfy all requirements

- Some emerging non-volatile memory technologies (e.g., PCM) enable new opportunities: memory+storage merging

- We need to rethink the main memory system
  - to fix DRAM issues and enable emerging technologies
  - to satisfy all requirements

SAFARI

Major Trends Affecting Main Memory (I)

- Need for main memory capacity, bandwidth, QoS increasing

- Main memory energy/power is a key system design concern

- DRAM technology scaling is ending
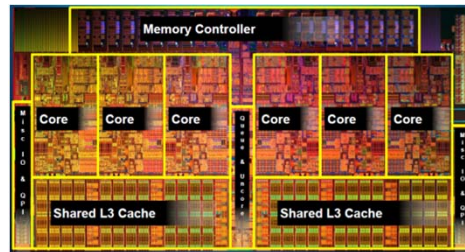
Major Trends Affecting Main Memory (II)

- **Need for main memory capacity, bandwidth, QoS increasing**
  - Multi-core: increasing number of cores
  - Data-intensive applications: increasing demand/hunger for data
  - Consolidation: cloud computing, GPUs, mobile

- Main memory energy/power is a key system design concern

- DRAM technology scaling is ending
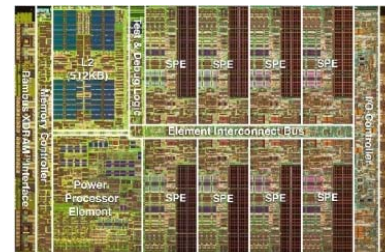
Example Trend: Many Cores on Chip

- Simpler and lower power than a single large core
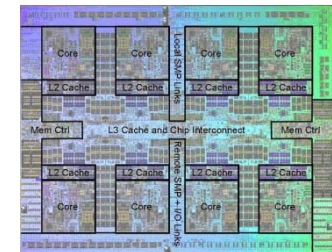- Large scale parallelism on chip
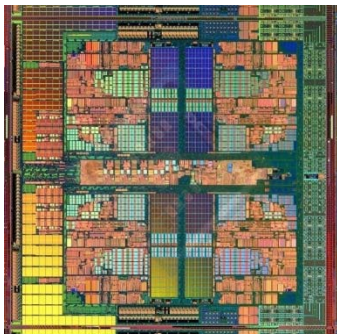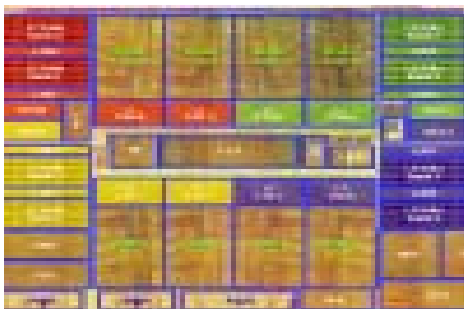

AMD Barcelona
4 cores

Intel Core i7
8 cores

IBM Cell BE
8+1 cores

IBM POWER7
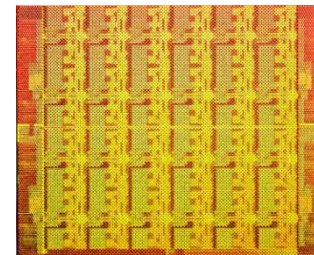8 cores

Sun Niagara II
8 cores
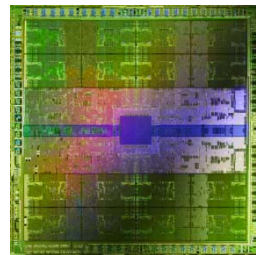
Nvidia Fermi
448 "cores"

Intel SCC
48 cores, networked
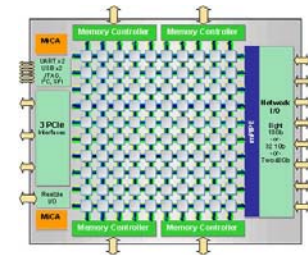
Tilera TILE Gx
100 cores, networked

Consequence: The Memory Capacity Gap

Core count doubling ~ every 2 years
DRAM DIMM capacity doubling ~ every 3 years
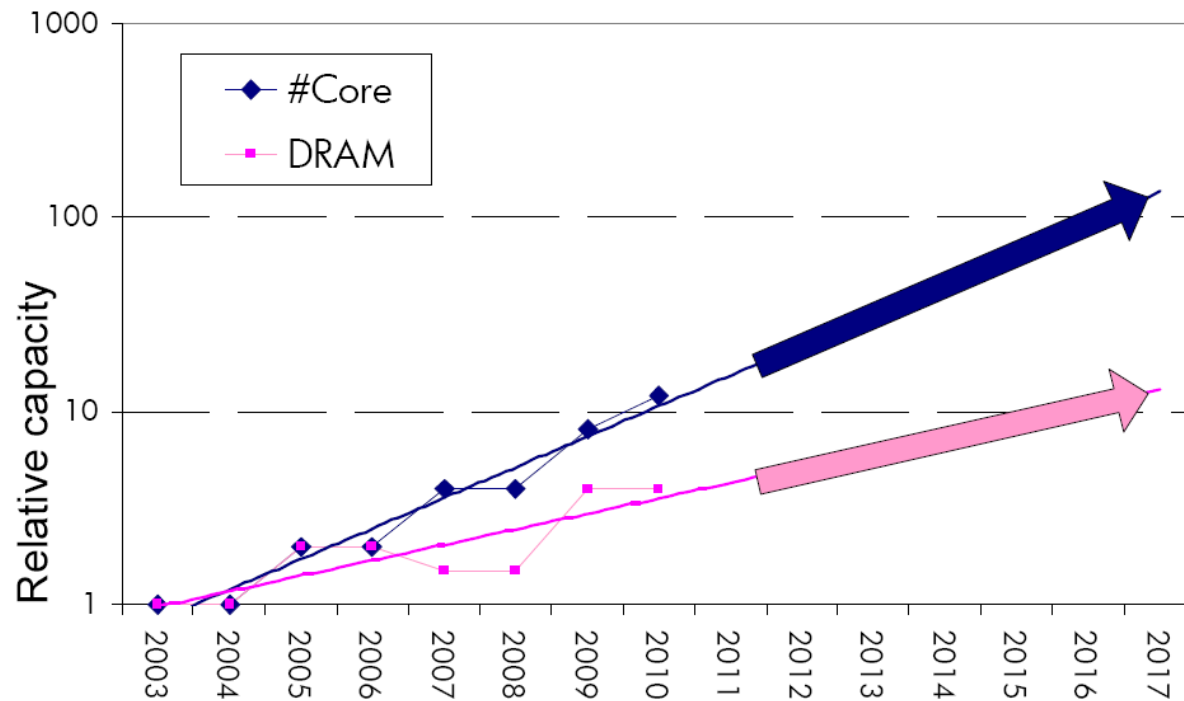
Source: Lim et al., ISCA 2009.

- *Memory capacity per core* expected to drop by 30% every two years
- Trends worse for *memory bandwidth per core*!

Major Trends Affecting Main Memory (III)

- Need for main memory capacity, bandwidth, QoS increasing

- **Main memory energy/power is a key system design concern**
  - ~40-50% energy spent in off-chip memory hierarchy [Lefurgy, IEEE Computer 2003]
  - DRAM consumes power even when not used (periodic refresh)

- DRAM technology scaling is ending

Major Trends Affecting Main Memory (IV)

- Need for main memory capacity, bandwidth, QoS increasing

- Main memory energy/power is a key system design concern

- DRAM technology scaling is ending
  - ITRS projects DRAM will not scale easily below X nm
  - Scaling has provided many benefits:
    - higher capacity (density), lower cost, lower energy

The DRAM Scaling Problem

- **DRAM stores charge in a capacitor (charge-based memory)**
    - Capacitor must be large enough for reliable sensing
    - Access transistor should be large enough for low leakage and high retention time
    - Scaling beyond 40-35nm (2013) is challenging [ITRS, 2009]
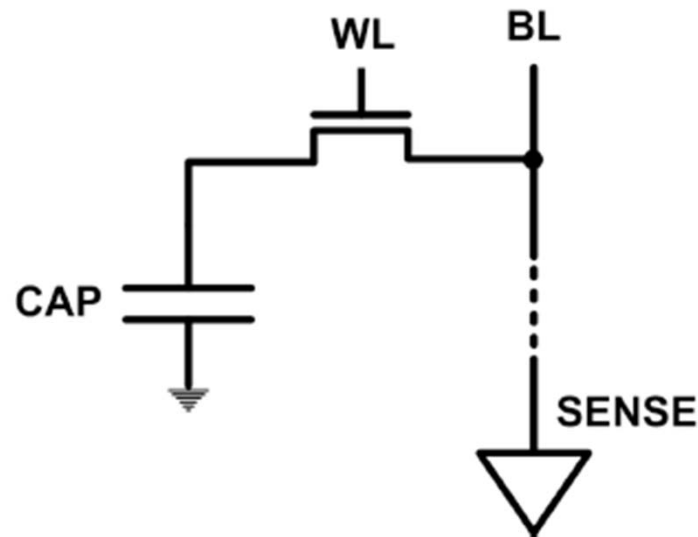
- **DRAM capacity, cost, and energy/power hard to scale**

Solutions to the DRAM Scaling Problem

- Two potential solutions
  - Tolerate DRAM (by taking a fresh look at it)
  - Enable emerging memory technologies to eliminate/minimize DRAM

- Do both
  - Hybrid memory systems

Solution 1: Tolerate DRAM

- Overcome DRAM shortcomings with
  - System-DRAM co-design
  - Novel DRAM architectures, interface, functions
  - Better waste management (efficient utilization)

- Key issues to tackle
  - Reduce refresh energy
  - Improve bandwidth and latency
  - Reduce waste
  - Enable reliability at low cost

- Liu, Jaiyen, Veras, Mutlu, "RAIDR: Retention-Aware Intelligent DRAM Refresh," ISCA 2012.
- Kim, Seshadri, Lee+, "A Case for Exploiting Subarray-Level Parallelism in DRAM," ISCA 2012.
- Lee+, "Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture," HPCA 2013.
- Liu+, "An Experimental Study of Data Retention Behavior in Modern DRAM Devices" ISCA'13.
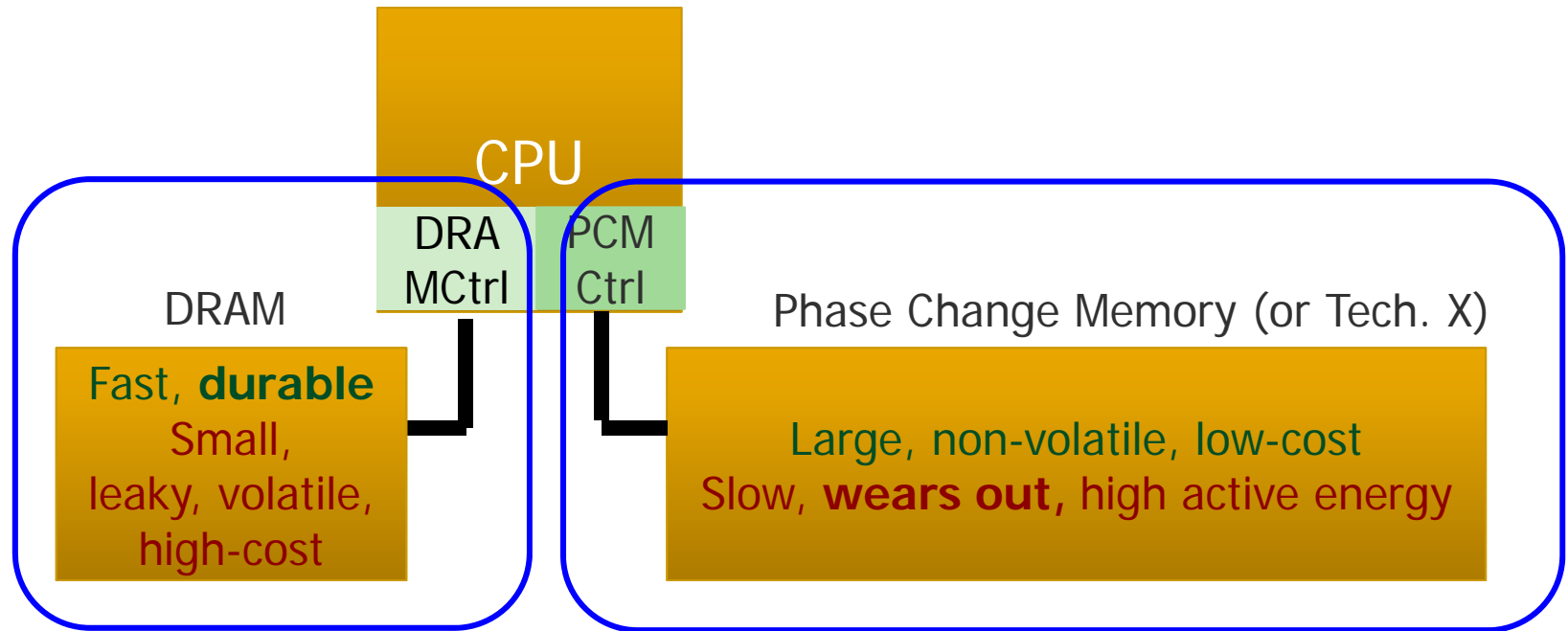- Seshadri+, "RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data," 2013.

SAFARI

Solution 2: Emerging Memory Technologies

- **Some emerging resistive memory technologies seem more scalable than DRAM (and they are non-volatile)**

- Example: Phase Change Memory
  - Expected to scale to 9nm (2022 [ITRS])
  - Expected to be denser than DRAM: can store multiple bits/cell

- But, emerging technologies have shortcomings as well
  - Can they be enabled to replace/augment/surpass DRAM?

- Lee, Ipek, Mutlu, Burger, "Architecting Phase Change Memory as a Scalable DRAM Alternative," ISCA 2009, CACM 2010, Top Picks 2010.
- Meza, Chang, Yoon, Mutlu, Ranganathan, "Enabling Efficient and Scalable Hybrid Memories," IEEE Comp. Arch. Letters 2012.
- Yoon, Meza et al., "Row Buffer Locality Aware Caching Policies for Hybrid Memories," ICCD 2012 Best Paper Award.

SAFARI

Hybrid Memory Systems

CPU

DRA MCtrl   PCM Ctrl

DRAM
Fast, **durable**
Small, leaky, volatile, high-cost

Phase Change Memory (or Tech. X)
Large, non-volatile, low-cost
Slow, **wears out,** high active energy

Hardware/software manage data allocation and movement
to achieve the best of multiple technologies

Meza+, "Enabling Efficient and Scalable Hybrid Memories," IEEE Comp. Arch. Letters, 2012.
Yoon, Meza et al., "Row Buffer Locality Aware Caching Policies for Hybrid Memories," ICCD 2012 Best Paper Award.

SAFARI

An Orthogonal Issue: Memory Interference

- Problem: Memory interference is uncontrolled → uncontrollable, unpredictable, vulnerable system

- Goal: We need to control it → Design a QoS-aware system

- Solution: Hardware/software cooperative memory QoS
  - Hardware designed to provide a configurable fairness substrate
    - Application-aware memory scheduling, partitioning, throttling
  - Software designed to configure the resources to satisfy different QoS goals

  - E.g., fair, programmable memory controllers and on-chip networks provide QoS and predictable performance
    [2007-2012, Top Picks'09,'11a,'11b,'12]