COMP2310
Systems, Networks, & Concurrency

Convener: Shoaib Akram
shoaib.akram@anu.edu.au

Australian
National
University

DRAM Organization

- Today, we will focus on the hardware side of modern memory systems
- DRAM **internal organization** remains a mystery
- MOV instruction
  - CPU generates the virtual address
  - MMU performs the translation
  - MMU send physical address to main memory
  - What happens next?
  - How does memory decode the address?
- We make better systems when we understand both hardware and software
  - Today: bring your hardware mindset

The Big Picture

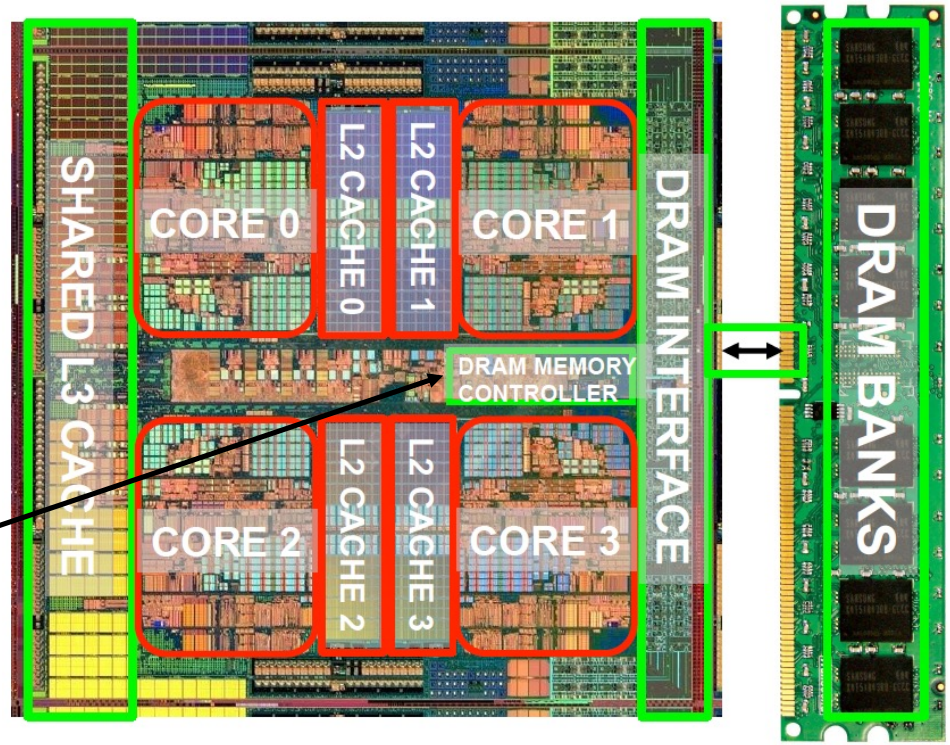
- Out-of-order processor
  - Exploits instruction-level parallelism (ILP) to deliver high performance
  - ILP is the parallel execution of independent instructions via pipelining
  - If the instructions executing in parallel are memory access instructions, the phenomenon is called memory-level parallelism (MLP)
  - For data-intensive applications, MLP is the key to processing large datasets in a reasonable time

- **Exploiting MLP requires support from the cache and memory hierarchy**

- Modern caches are non-blocking
  - Multiple requests are *in flight* at any time
  - Cache controllers maintain the state of each request in a hardware register called miss status handling register (MSHR)

- Internally, DRAM chips are organized to exploit MLP as much as possible
  - **Today:** Gain insight into the internal organization of DRAM chips

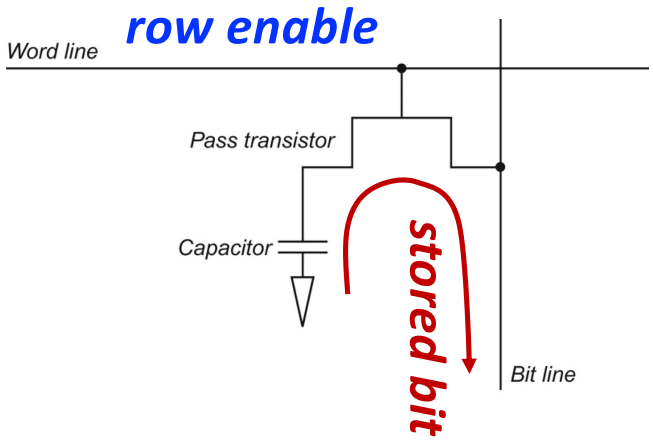
The Big Picture



integrated memory/DRAM controller (iMC)

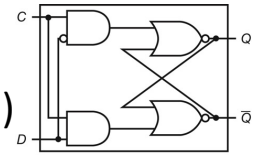
DRAM Storage Cell

- Main memory uses a technology called Dynamic Random Access Memory (DRAM)
- **One** pass or access transistor and **one** capacitor
- The cell is called **1T1C** cell
- Capacitor holds the information to be stored as charge
- Pass transistor controls loading and storing charge to and from the capacitor

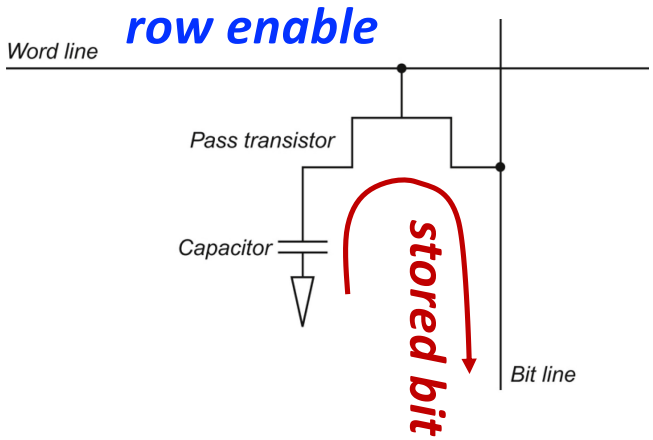


DRAM Cell (Cont'd)

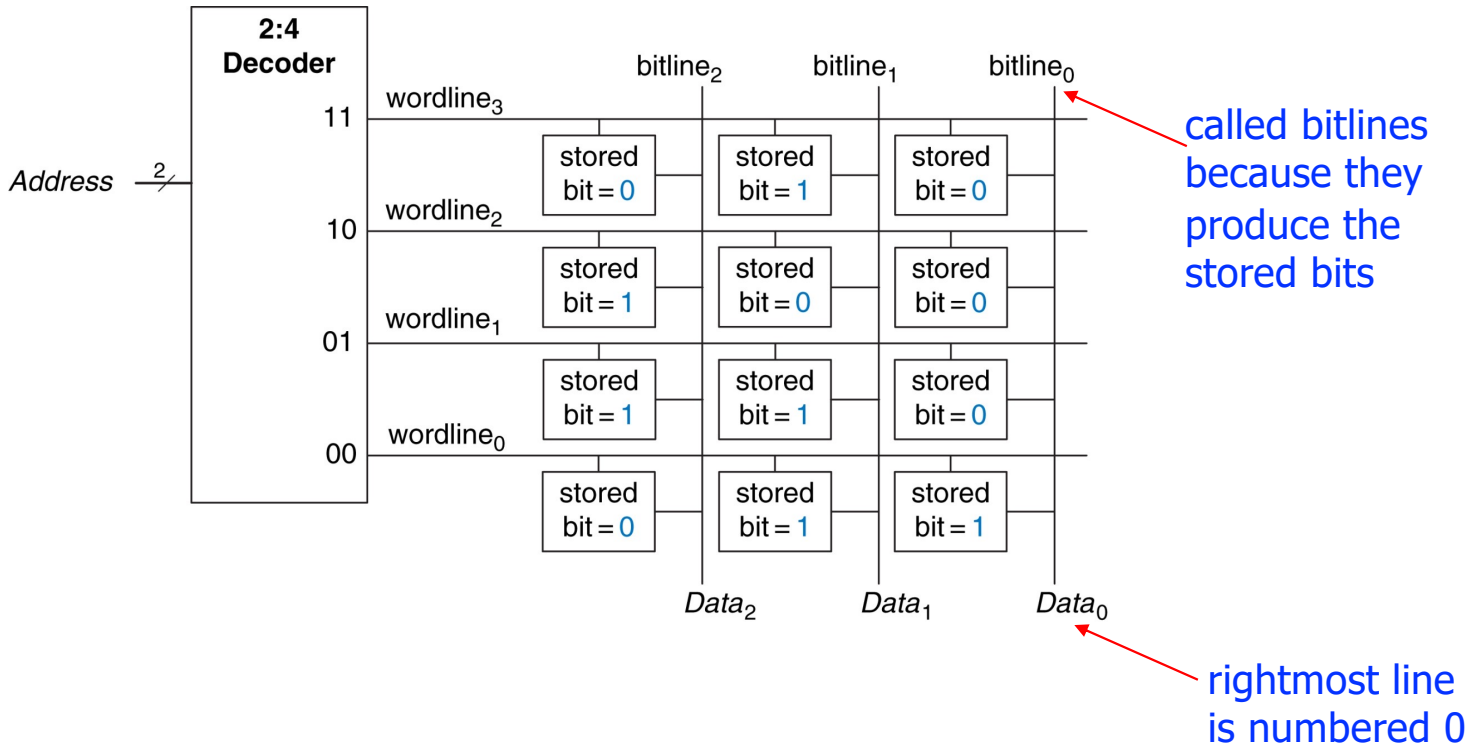
- Capacitor loses charge over time and must be refreshed periodically (the dynamic in DRAM is due to this fact)
- DRAM cells are denser than SRAM cells
  - Contrast 1T1C cells with 4 – 8 transistor SRAM cells)



- DRAM has higher latency than SRAM
  - Charging and discharging capacitors take time
- DRAM is used as off-chip memory due to manufacturing/cost reasons
  - SRAM is used for building registers and caches
  - DRAM is used for building off-chip main memory
- The refresh process in DRAM consumes extra energy
  - In contrast with SRAM cells, DRAM does not constantly draw current and hence is overall more energy-efficient



Recall: Canonical Memory Organization



Memory Wall: Three Aspects

- Applications are increasingly limited by memory capacity and speed

- In old days, for many applications, CPU speed was the key bottleneck
  - Working sets were small and used to fit in cache and memory
  - Fewer memory accesses relative to total instruction count

- An application hit a memory wall when most of the time CPU waits for memory to respond

- Recall that an out-of-order processor needs a very large instruction window to be able to hide last-level cache miss

Memory Wall: Three Aspects

- Capacity
- Latency
- Bandwidth

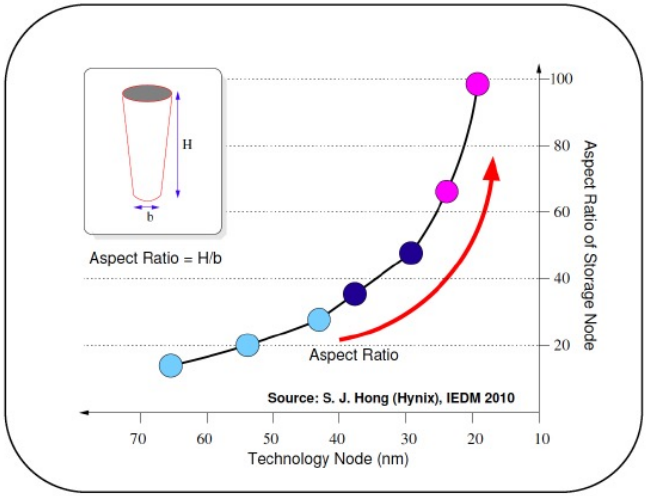
Memory Capacity Wall

- Shrinking DRAM cells below the 10 nanometer (nm) technology node is very complicated

Scaling wall
DRAM does not scale well to small feature sizes (sub 1x nm)

- Main issue is capacitor geometry
  - Shrinking length and width decreases the charge stored in a capacitor
  - Compensate by increasing the height

- The ratio of height to length is called the aspect ratio
  - Capacitor aspect ratios have increased exponentially
  - This has led to:
    - placement and routing issues
    - increased error rates during operation

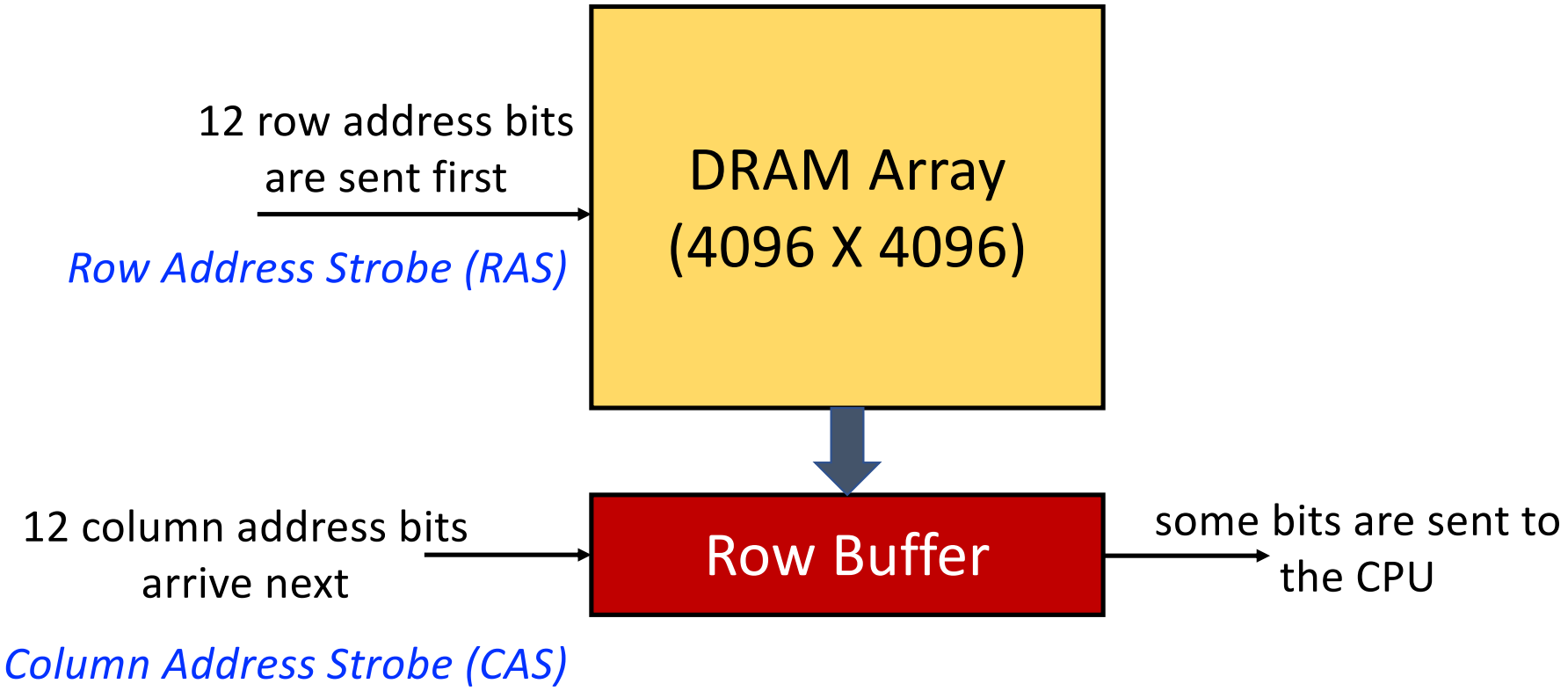


Memory Latency-Bandwidth Wall

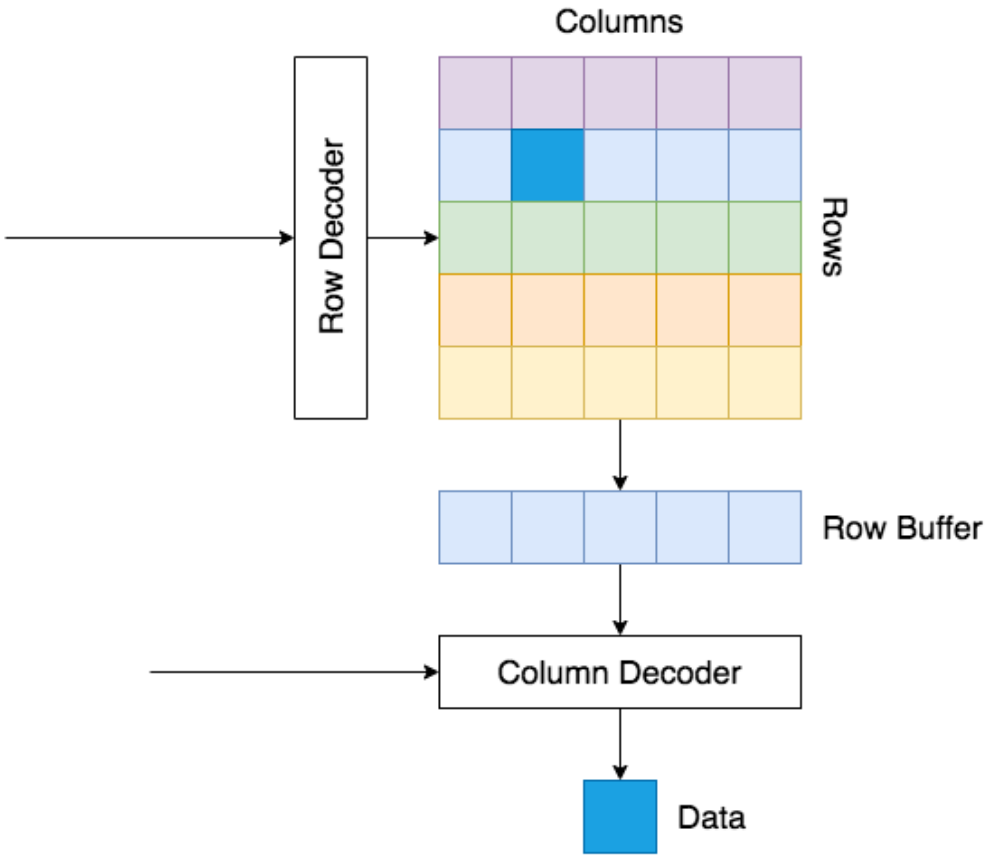
- If DRAM cell (1T1C) cannot be shrunk further, then memory latency cannot reduce anymore
- On the other hand, CPU advances and multicores stress the memory system more each year
- The gap between CPU and memory performance keeps increasing
- Bandwidth has not scaled as much as CPU performance either
  - After the DRAM lookup, data travels over wires (interconnect) to the CPU and this transfer is the key deciding factor in memory bandwidth
  - Unlike transistors, wires are hard to shrink, and their latency has not scaled as much as transistors
  - Limited bandwidth prohibits the CPU from making progress and it eventually stalls

Two Level Decoding

- Large memory arrays are organized into a 2-dimensional array of 1T1C cells
- The resulting matrix of rows and columns uses two-level decoding
- This organization results in practical decoder complexity and ease of mgmt.



Example: Two Level Decoding

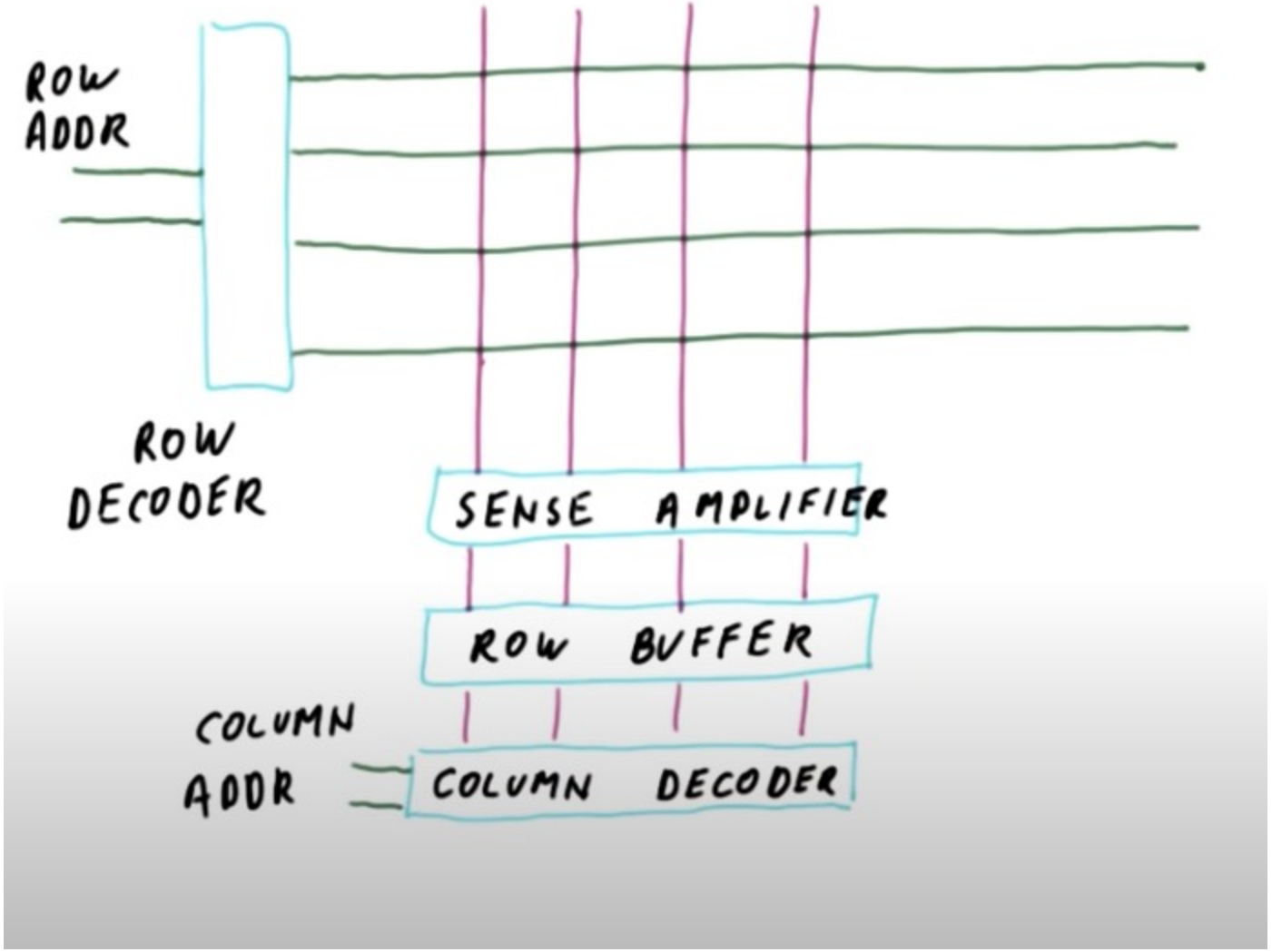


Row Buffer

- A DRAM row is also called a DRAM page
  - No relation to the virtual or physical memory page
  - It is a case of terminology overload!

- After the word line is activated, a DRAM row is read into a row buffer as follows
  - A device called sense amplifier (cross-coupled inverters) senses the small change in voltage on the bit lines and amplifies it (1T1C cells are very small)
    - There is one sense amp. at the end of bit lines
  - The amplified row values are stored into a row buffer (a row of flip-flops)
    - From the row buffer, correct bits are selected using a column address and outputs a single or more bits
  - The capacitor is now drained. DRAM reads are destructive. The sense amp drives the correct values back into the cells.
    - It is almost a read-write process
    - One of the reasons DRAM is slower than SRAM
- Note that one way to refresh the leaky DRAM cells is to read them periodically

DRAM Organization



Row Buffer Mgmt. Policy

- A row can be in one of the two states:
  - Open
    - Leave the contents of the row “just” read in the row buffer
    - If there are subsequent requests to the same row, there will be a row buffer hit
  - Closed
    - Prepare the bit lines to open a different row
      - Phenomenon is called precharging the bit lines
    - Clear the row buffer

What is precharging bit lines?

- Wires have internal capacitance which is much larger than the capacitance of the 1T1C cell
- This internal capacitance depends on many factors and not determined precisely at manufacturing time
- A small capacitor used in DRAM cells will drive the bitlines to a random voltage (not precise)
- To make the process of reads more reliable, bit lines are precisely precharged to a reference voltage (typically VDD/2)

Access to a Closed Row

- **ACTIVATE** Command
  - Decode the physical address
  - Drive the row select line
    - Selected bit cells drive the bit lines
    - Read the entire row into the row buffer
    - Row buffer (sense amplifier) amplifies and regenerates the bit lines
    - Restores the capacitor

- **READ**/**WRITE** Command
  - Drive the column select line
  - Read/write the columns in the row buffer (mux out/in data bits)

- **PRECHARGE** Command
  - Prepares the DRAM array for next access
  - Required for access to a different row
  - Called closing the row

Access to an Open Row

- No need for ACTIVATE command

Typical Command Sequence

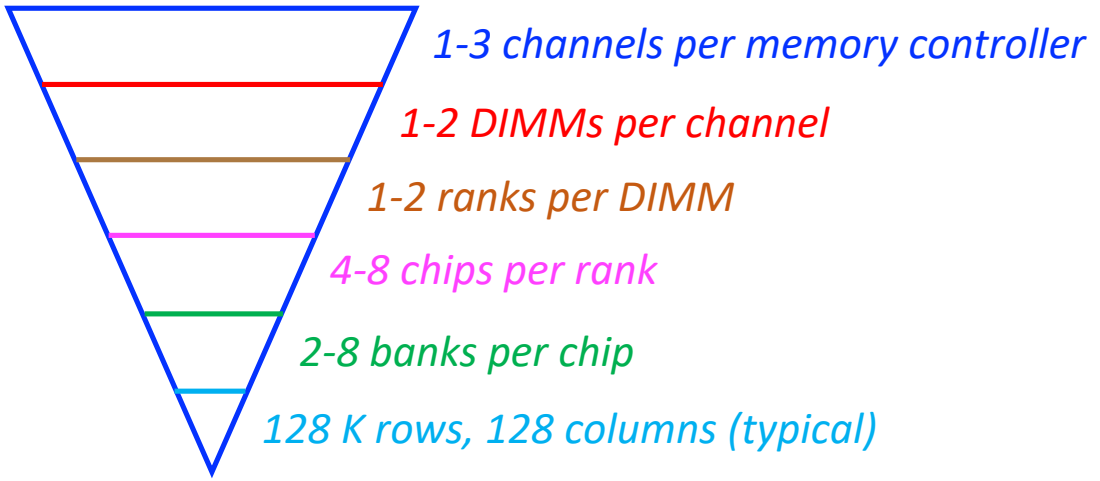
- 1 read, **2 row buffer hits**, 1 read to a different row

- Open row policy
  - ACT, RD, RD, RD, PRE, ACT, RD

- Closed row policy
  - ACT, RD, PRE, ACT, RD, PRE, ACT, RD, PRE, ACT, RD, PRE

Memory Organization

- DRAM is organized into a hierarchy
  - Channel
    - DIMM
      - Rank
        - Chip
          - Bank
            - Row/Column



What do we need from DRAM organization?

- Memory-level parallelism (MLP)
  - Instruction-level parallelism (ILP): execute independent instructions in parallel
  - Similarly, independent memory requests can be resolved in parallel

- Capacity scaling
  - We want to be able to add capacity to the system as needed

- Low latency (10s of nanoseconds)
  - Limited ultimately by physics (technology), but a good organization minimizes latency (e.g., via parallelism)

- High bandwidth
  - **Bandwidth:** Number of bytes returned to the CPU in a given time
  - Measured in bytes per second (30 – 50 GB/s typical for servers)

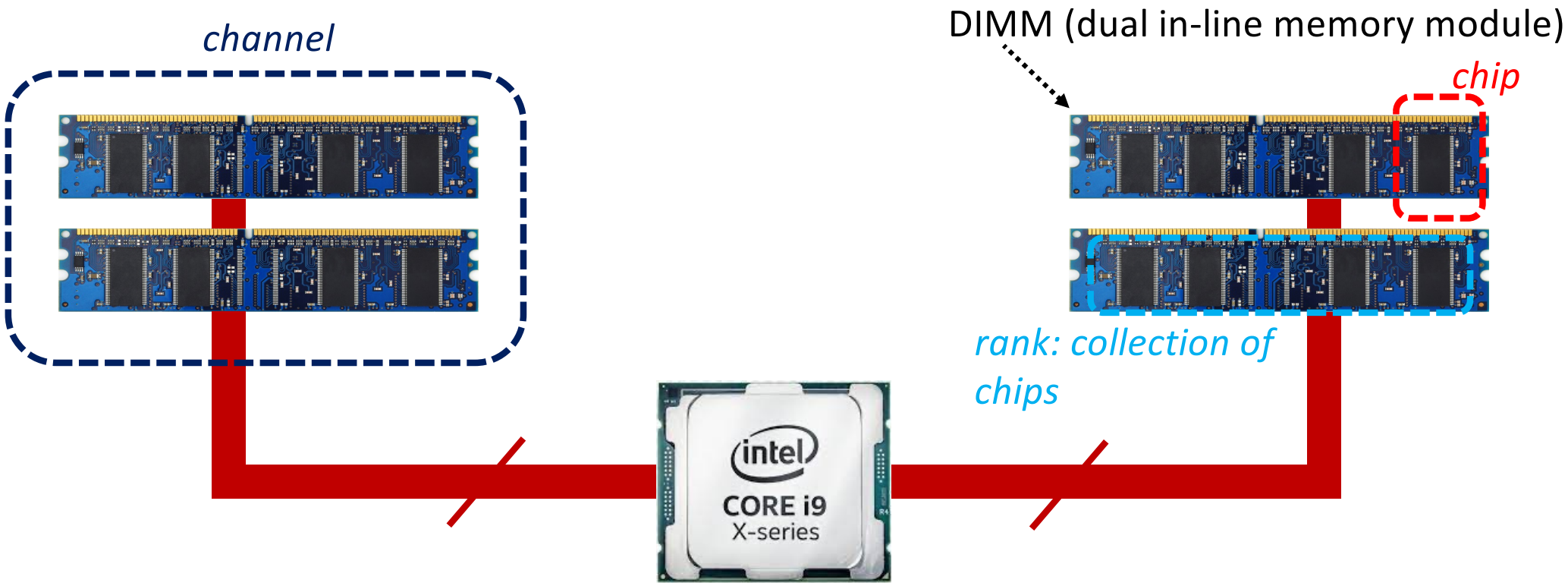
MLP and DRAM Organization

- DRAM organization and hierarchy *delivers* MLP
  - We have channel, DIMM, rank, chip, and bank-level parallelism
  - Once a bank is locked up, subsequent requests needs to wait
  - If there are multiple banks, each one of them can serve a different request simultaneously

Capacity Scaling and DRAM Organization

- DRAM organization as a hierarchy *delivers* capacity scaling
  - Scaling up (user): buy extra DIMMs
  - Scaling up (processor manufacturer): Add more ctlrs/channels to increase capacity (and bandwidth)
  - Scaling up capacity (DRAM manufacturer): Add more banks, increase per-bank capacity, add more rows/columns per bank, store more bits per column

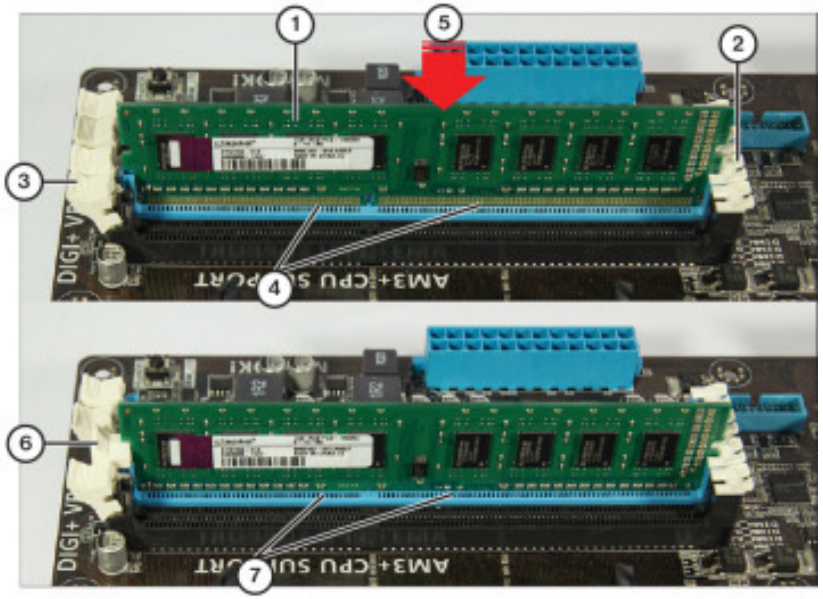
High-Level Organization



- Example organization
  - Two controllers with one channel per controller
  - Two DIMMs connected to each channel

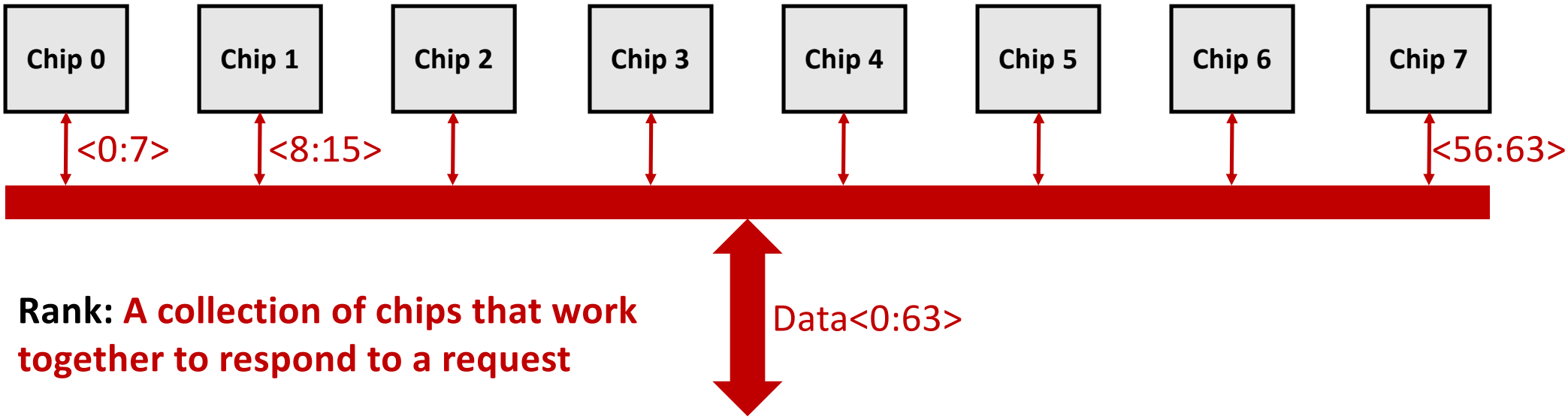
DIMM Overview

- One front-Side rank
- One back-Side rank



Rank (with x8 chips)

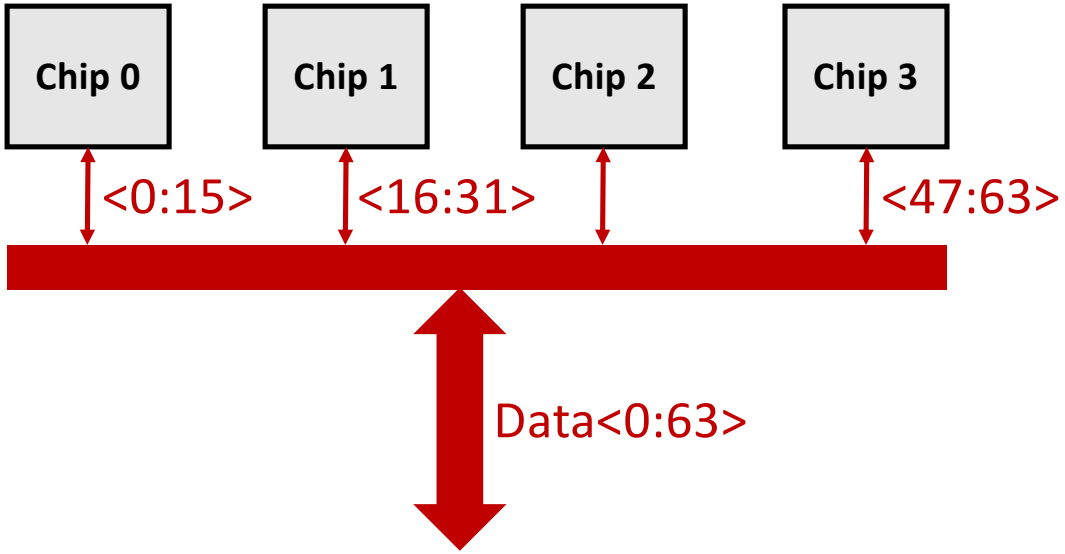
- In x8 DIMMs, there are 8 DRAM chips in a single rank
  - Here, x8 refers to the width of the data bus per chip
  - A 64-bit word is interleaved across 8 chips
  - Each chip delivers 8 bits out of 64 bits



Rank: A collection of chips that work together to respond to a request

Rank (with x16 chips)

- In x16 DIMMs, there are 4 DRAM chips in a single rank
  - Here, x16 refers to the width of the data bus per chip
  - A 64-bit word is interleaved across 4 chips
  - Each chip delivers 16 bits out of 64 bits



Rank (with x4 chips)

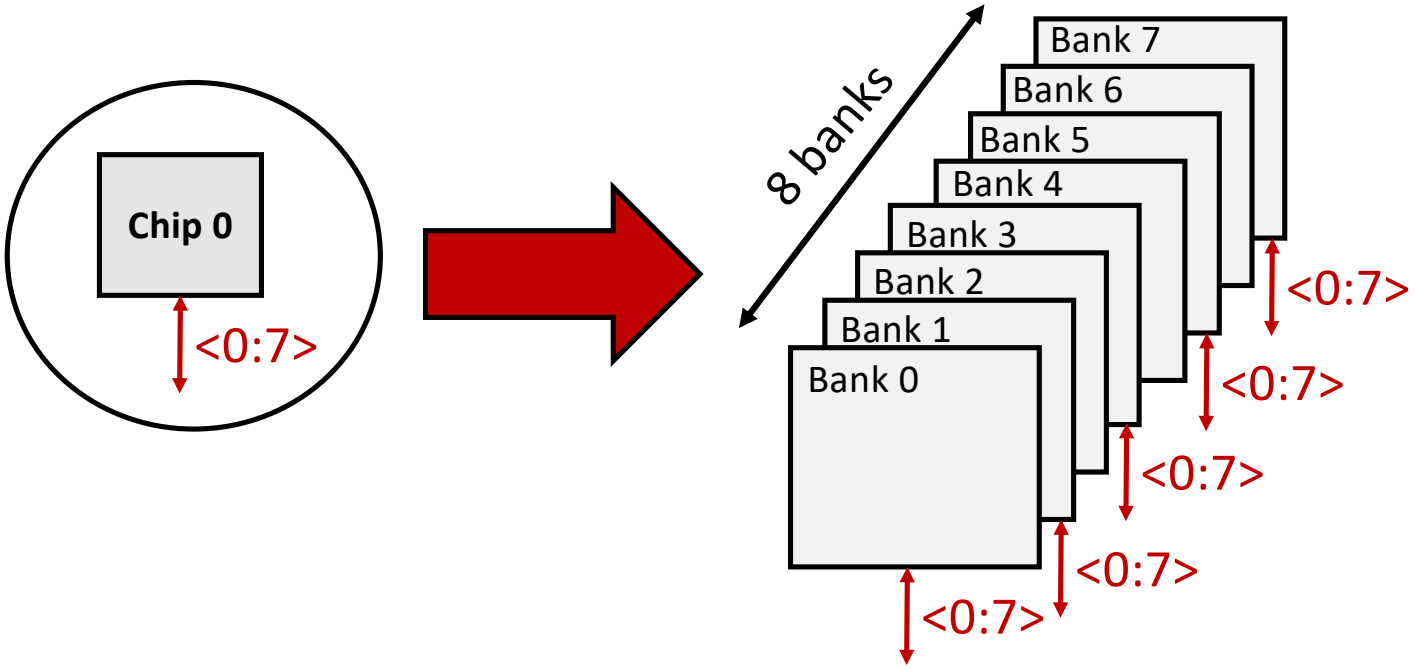
- In x4 DIMMs, there are 16 DRAM chips in a single rank
  - Here, x4 refers to the width of the data bus per chip
  - A 64-bit word is interleaved across 16 chips
  - Each chip delivers 4 bits out of 64 bits

Typical Use Cases

- x16
  - Used in space-limited scenarios
  - Handhelds or where PCB space is limited
  - Scaling up capacity is a problem
- x8
  - High-end servers in data centers
- x4
  - Used in scenarios where very high capacity is a requirement
  - Very high-end servers
  - *Signal integrity is a problem*

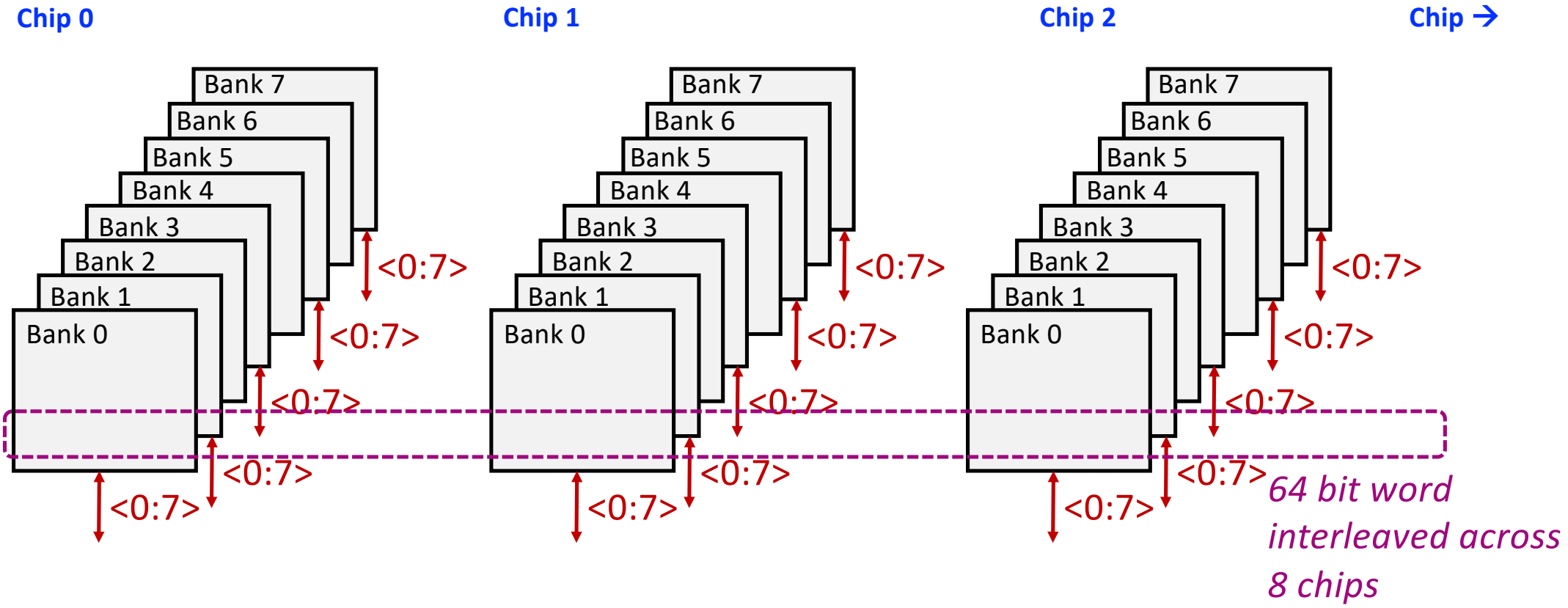
Breaking Down a Chip

- **Bank-level parallelism:** *Different banks serve different requests concurrently*
- The DRAM chip (e.g., x8) only has eight data output pins, so there is some delay due to bank-level sharing of pins
- *There is one row buffer per bank in the chip*

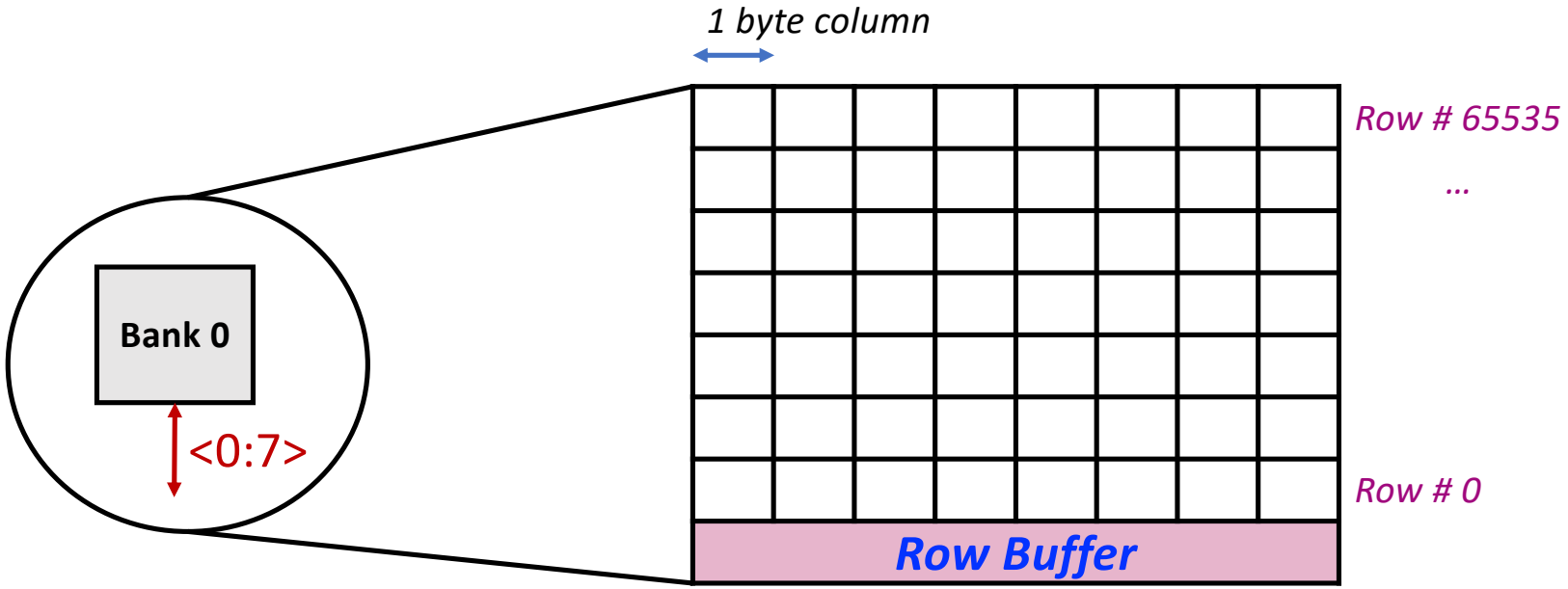


Breaking Down a Rank of Chips

Each bank (e.g., Bank 0) in chips<0:7> receive the same address request and respond with eight bits of the 64-bit word (chip 0 with bits<7:0> and chip 1 with bits<15:8> and so on

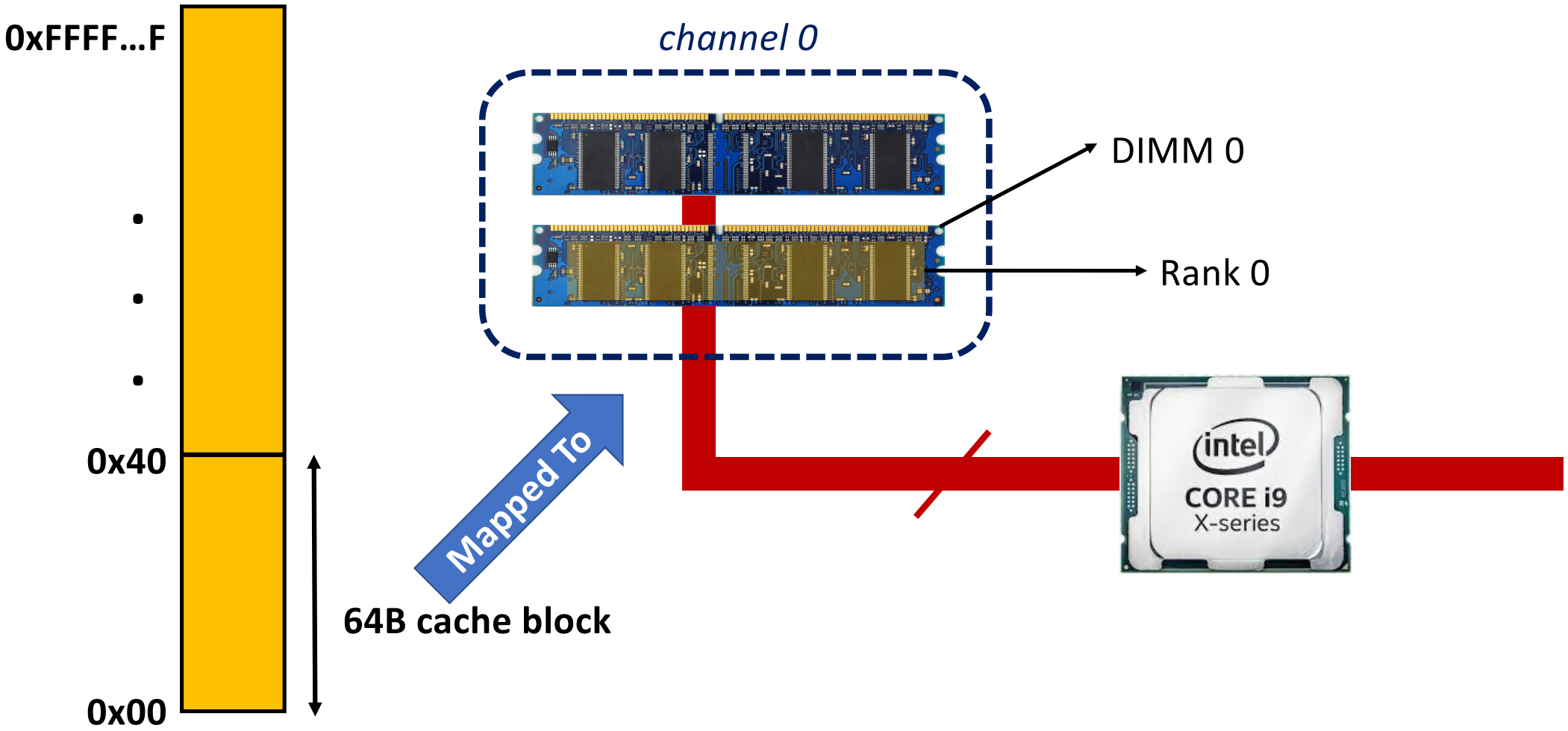


Breaking Down a Bank

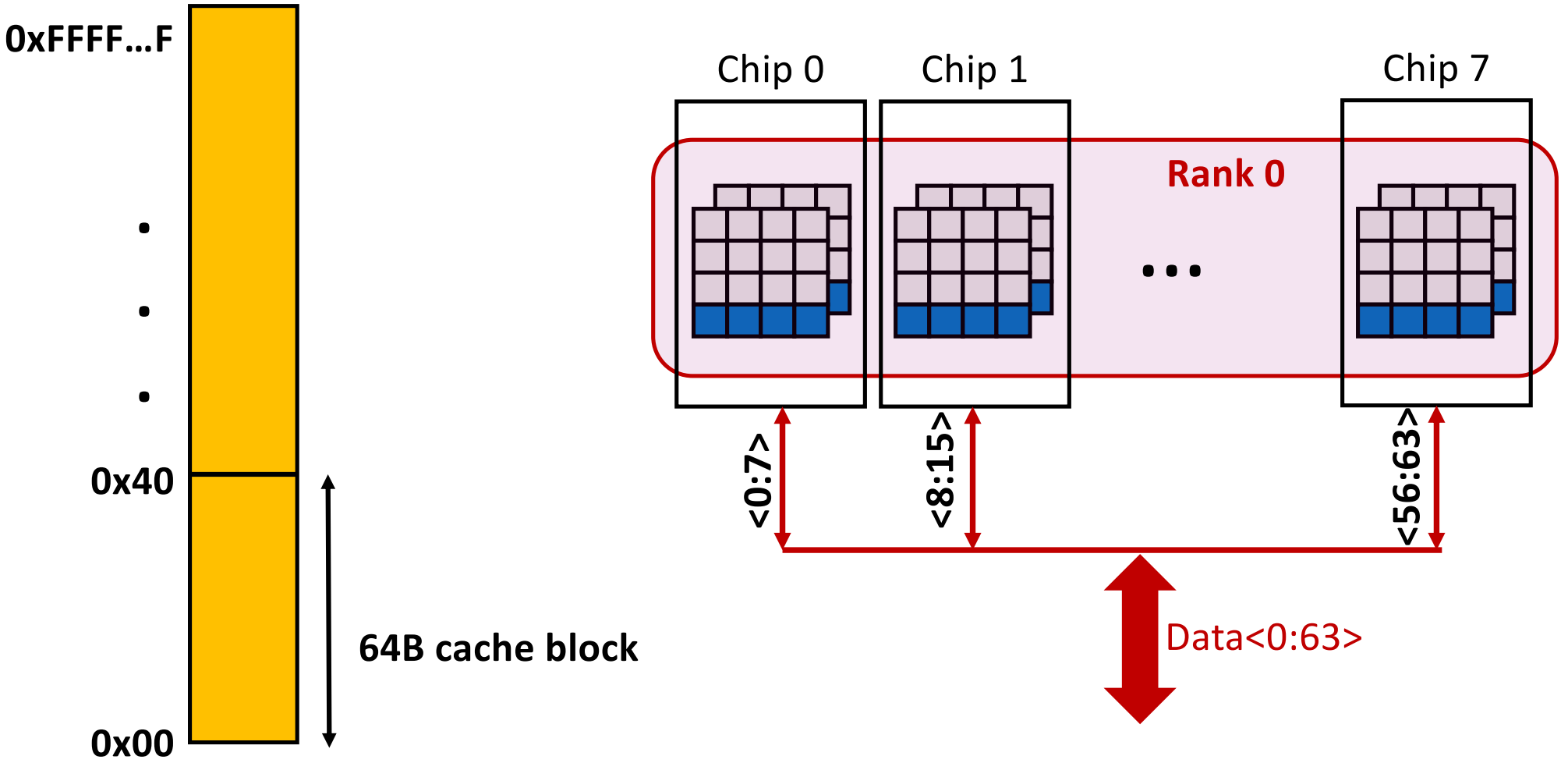


- Terminology: Row is also called an array or a DRAM array
- Bank is made up of multiple *wide* arrays (rows)
- Multiple arrays are also referred to as mats (*matrix of rows and columns*)

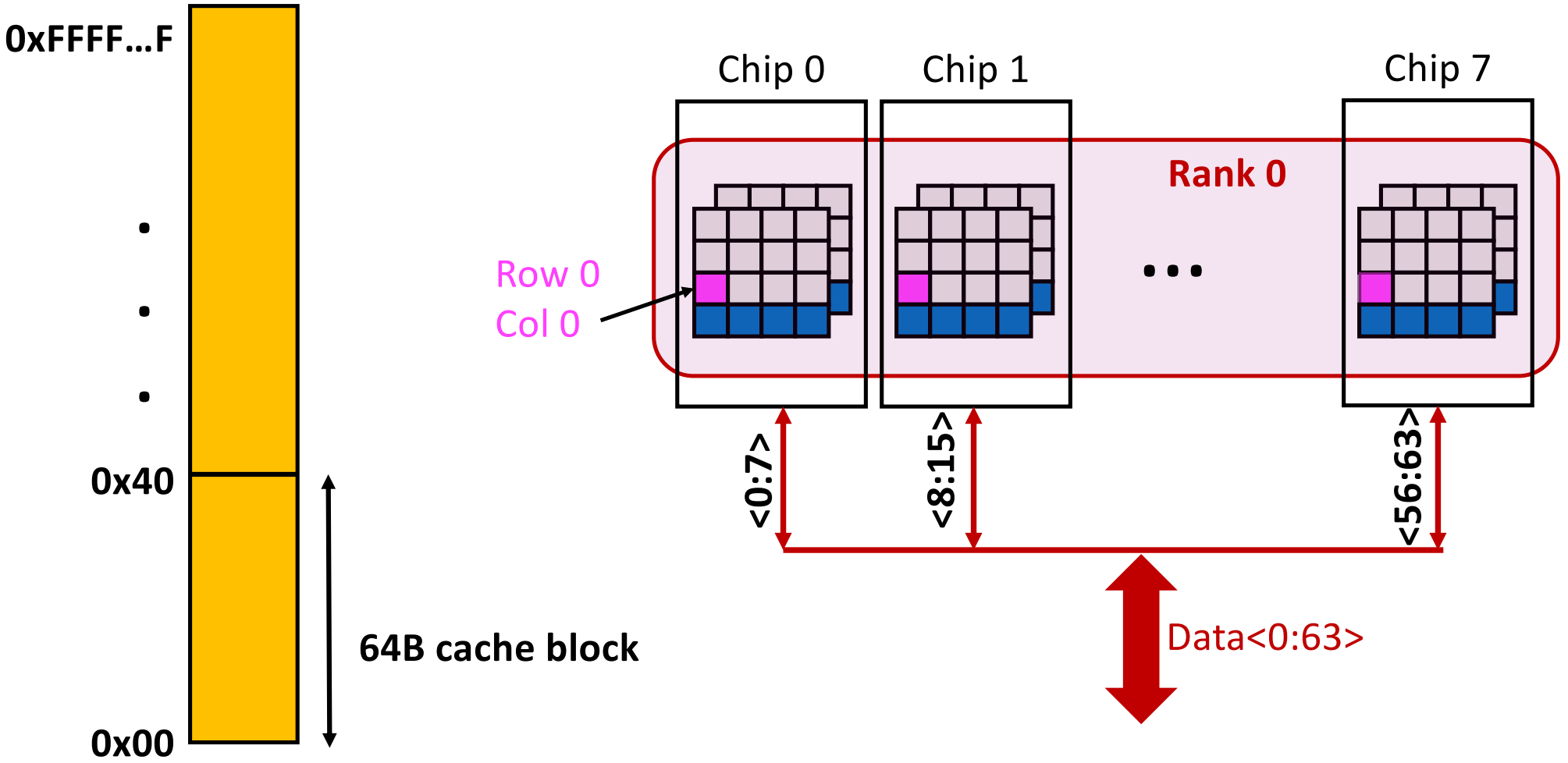
Example: Transferring a Cache Block



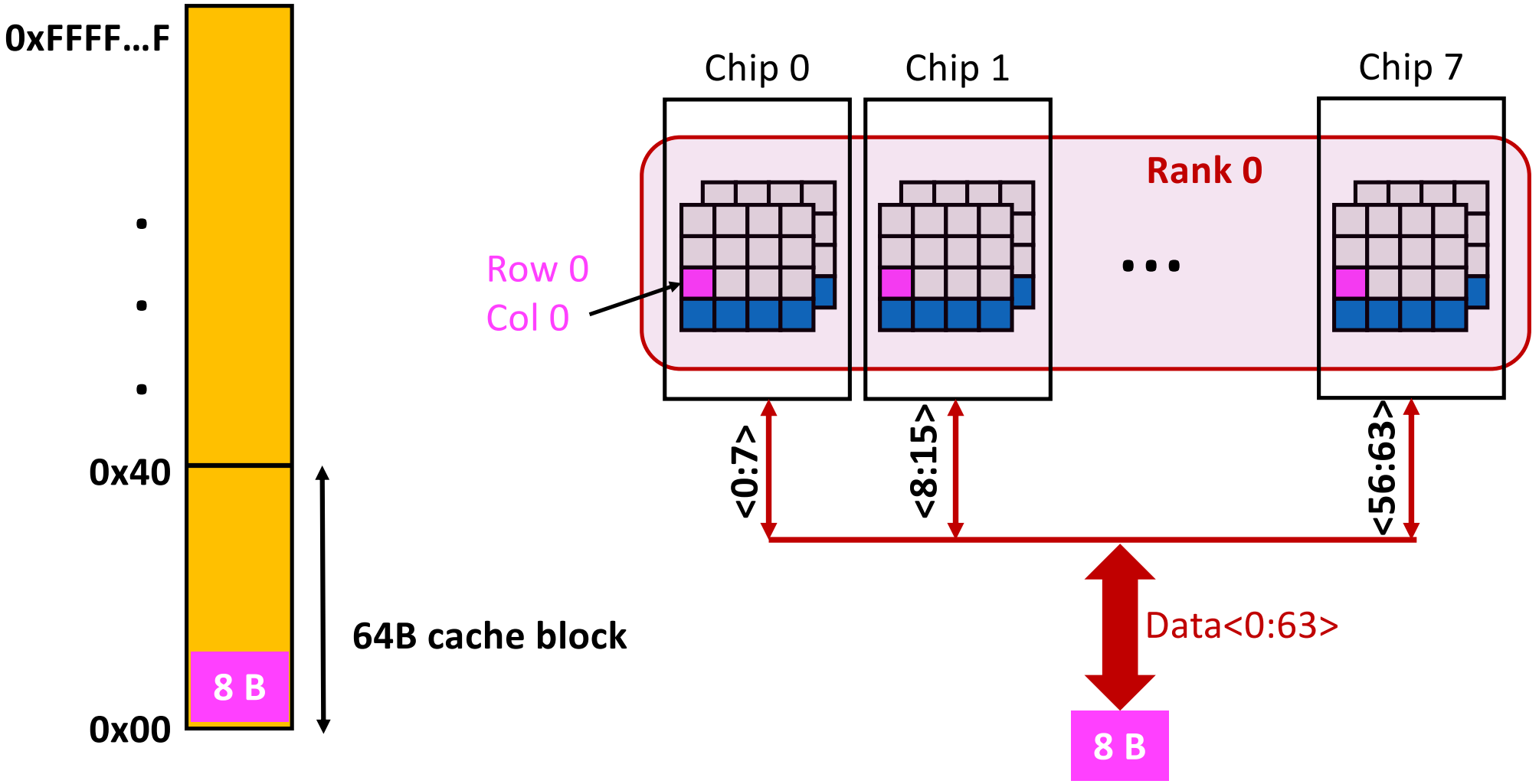
Example: Transferring a Cache Block



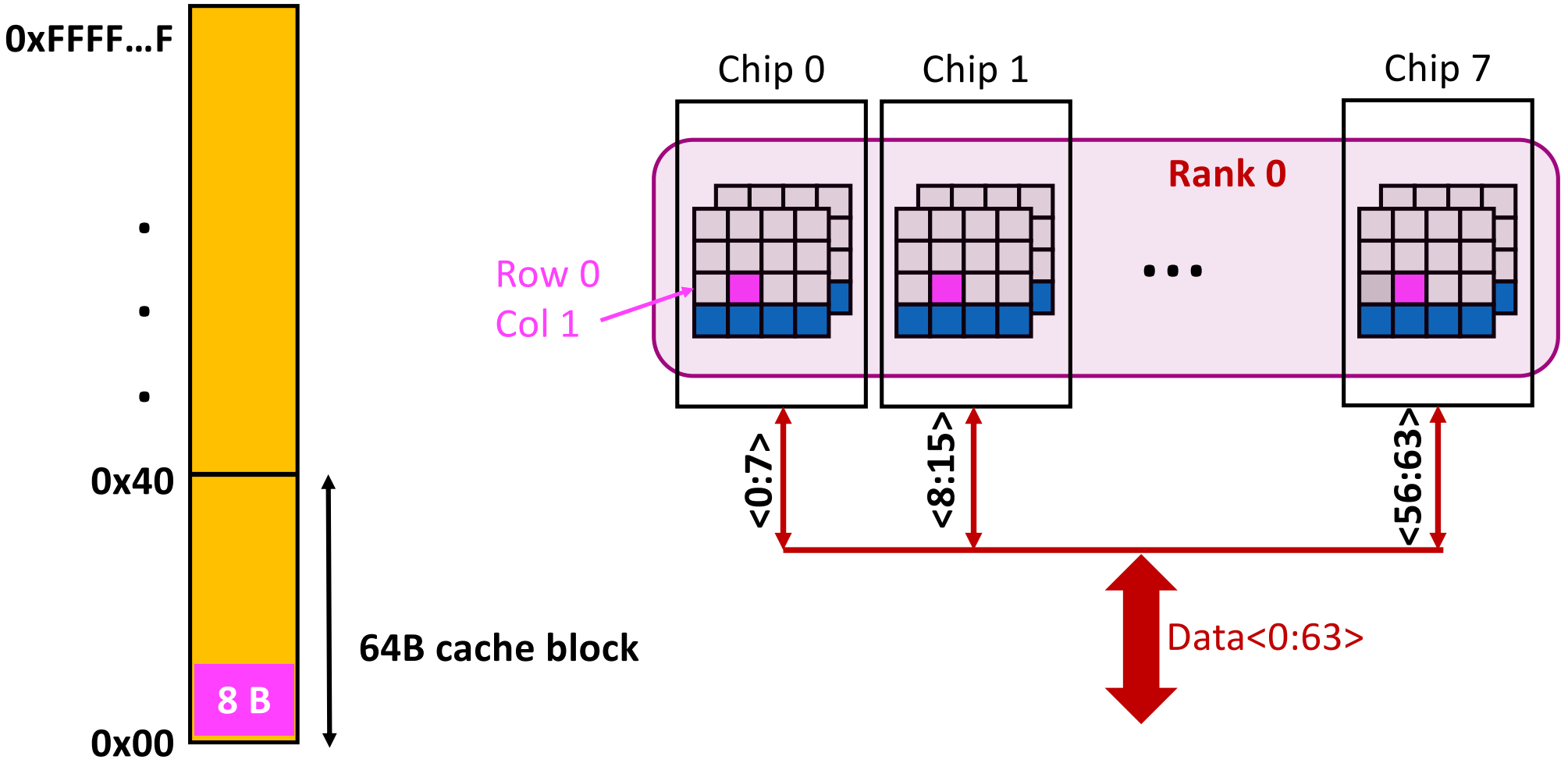
Example: Transferring a Cache Block



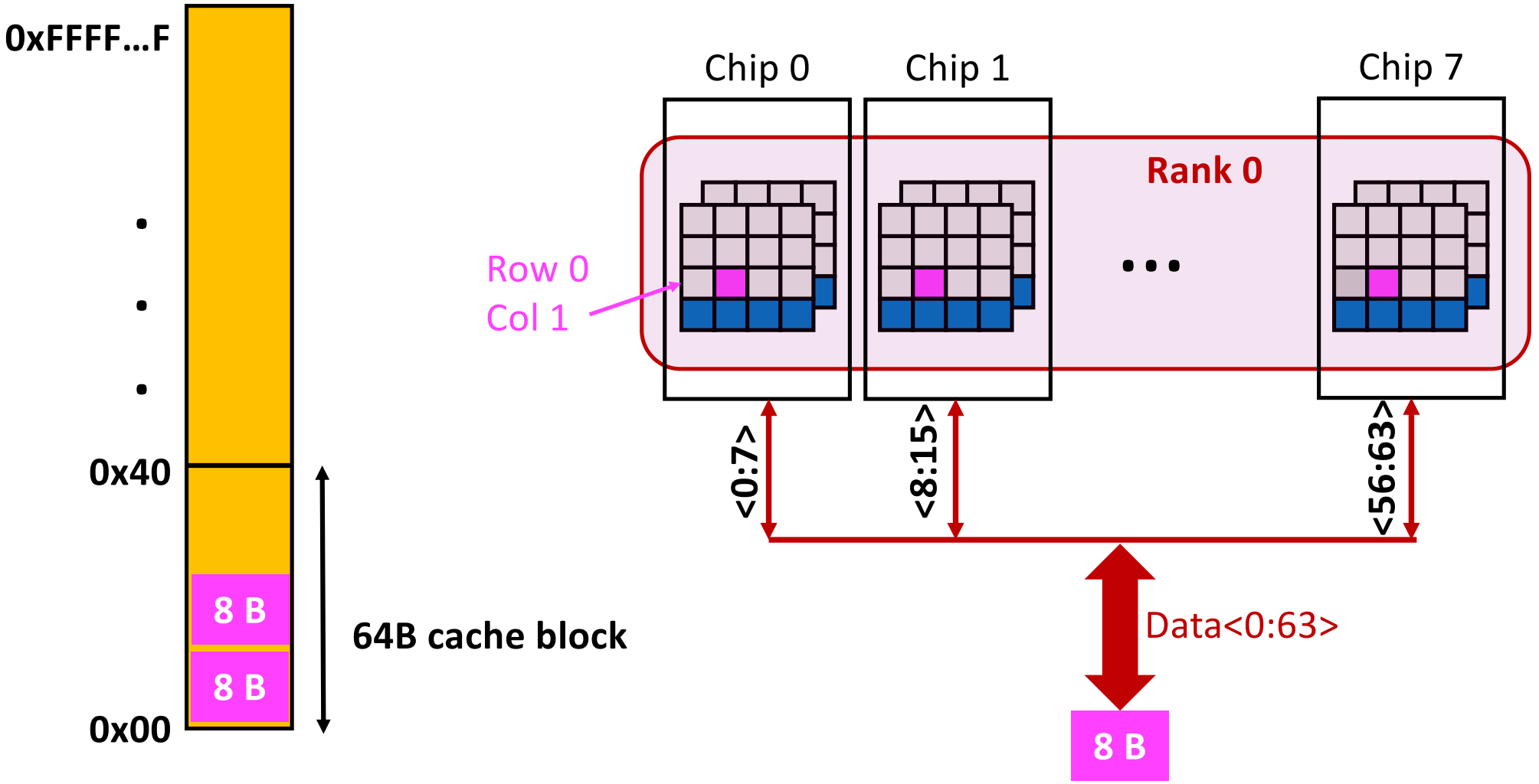
Example: Transferring a Cache Block



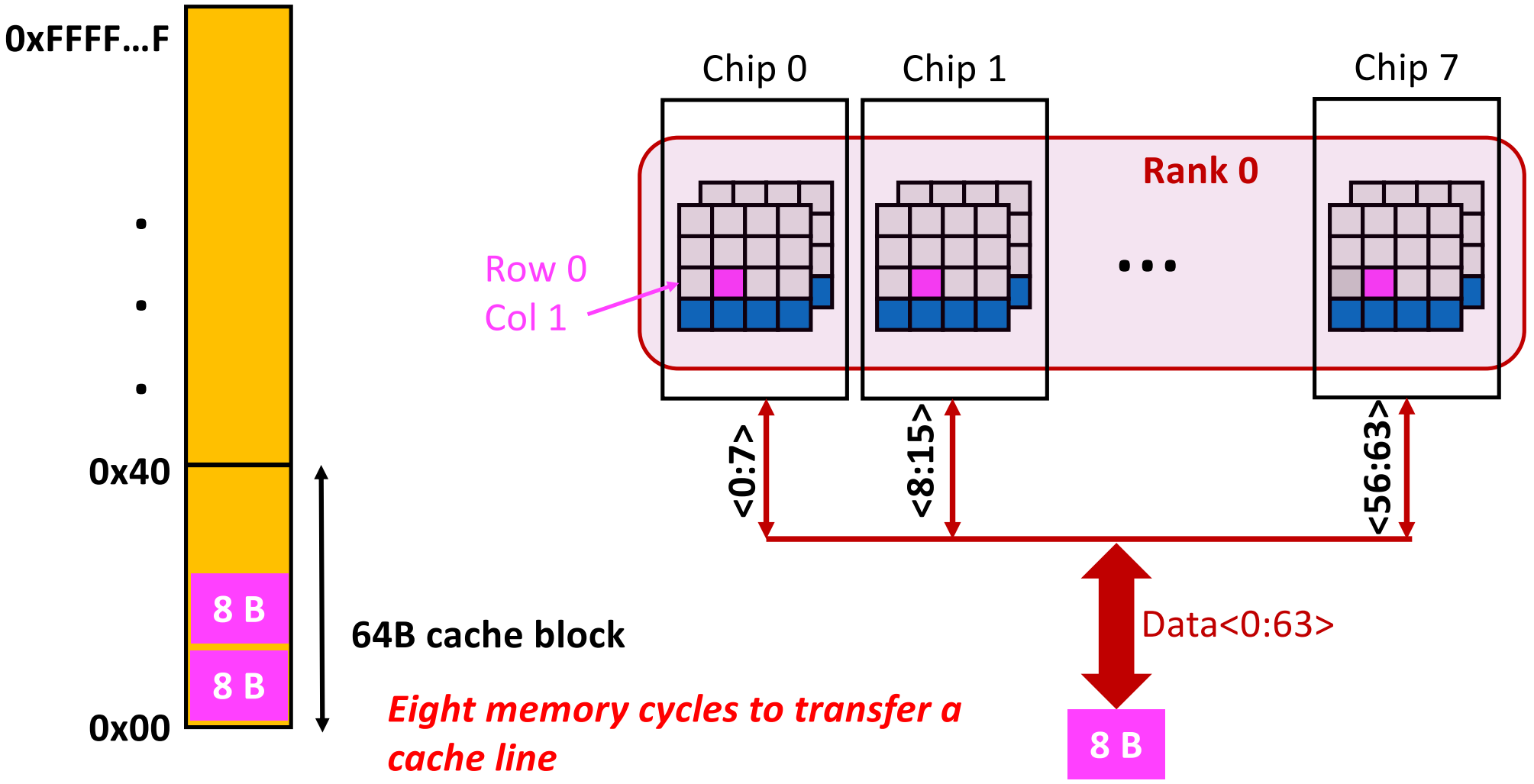
Example: Transferring a Cache Block



Example: Transferring a Cache Block



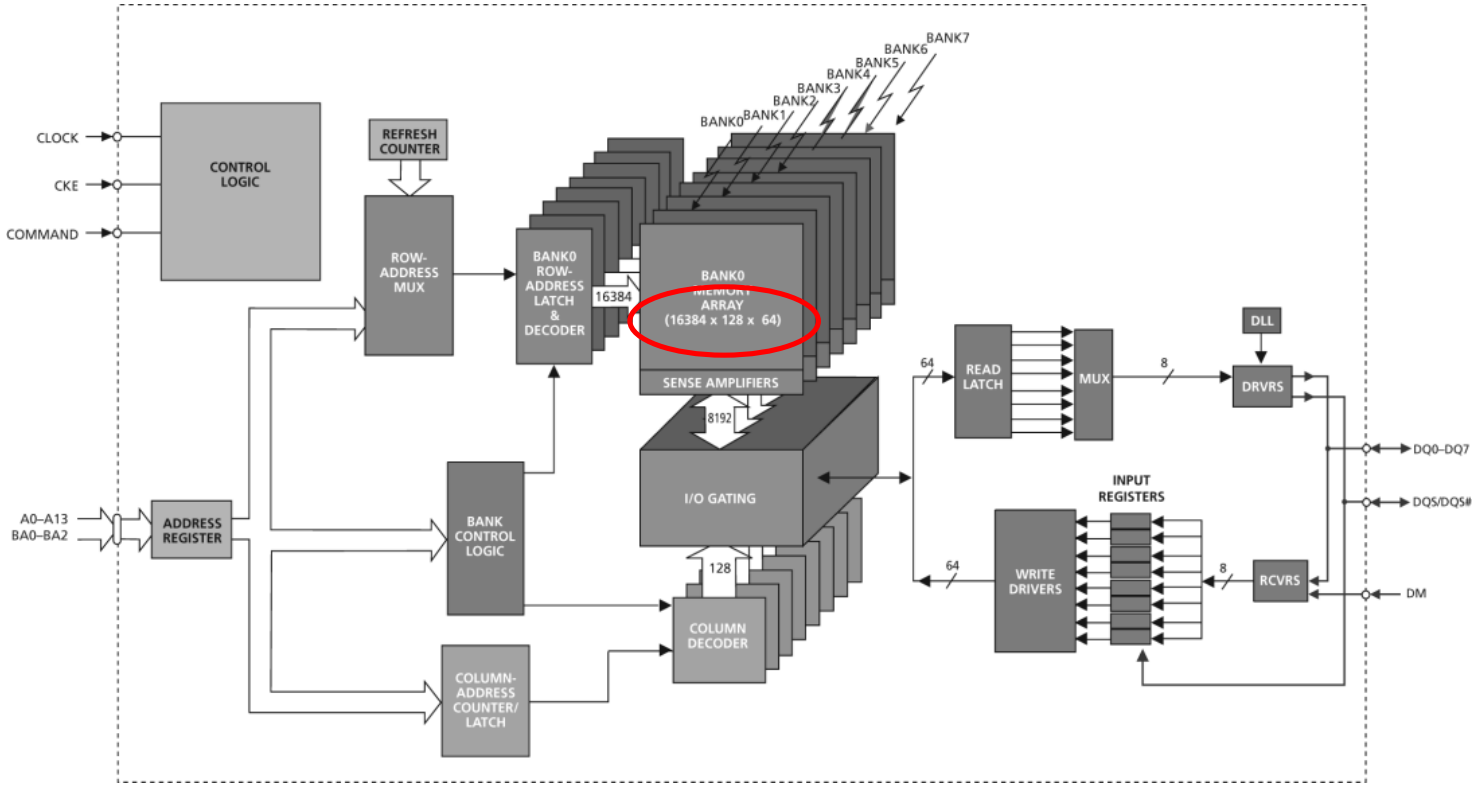
Example: Transferring a Cache Block



Organization Tradeoffs

- DIMM, rank, bank, array forms a hierarchy
- Electrical constraints govern DIMM count attached to a channel or bus
  - Typically a few DIMMs (1 – 2) per channel
- One DIMM can have 1 – 4 ranks
- When do we use wide-output DRAM chips?
  - When energy efficiency is the key constraint
    - Compare activating only 4 x16 chips instead of 16 x4 chips
- When do we use narrow-output DRAM chips?
  - When capacity scaling is the key requirement
    - 16 x4 chips deliver larger capacity than 4 x16 chips

Micron 128 M x8 DRAM Chip



- Here, there are 64 bits per column in a bank (think 3D) instead of just 8
  - Send 8 bits out at a time. Read 64 bits from array (8n prefetch)
- 8 x8 chips → 64 B cache line

Some Resources

Reading data sheets and understanding engineering details and tradeoffs

https://en.bmstu.wiki/index.php?title=DDR3_SDRAM&mobileaction=toggle_view_mobile
https://www.micron.com/-/media/client/global/documents/products/technical-note/dram/tn41_01ddr3_power.pdf
https://www.micron.com/-/media/client/global/documents/products/data-sheet/dram/ddr4/16gb_ddr4_sdram.pdf
https://www.youtube.com/watch?v=w2bFzQTQ9aI&ab_channel=ActuallyHardcoreOverclocking

Row Buffer Mgmt. Policies

- Two policies for managing the row buffer
  - **Open page:** Keep the current row/page open
  - **Closed page:** Precharge the bit lines right away

Open Page Policy

- **Access to a closed row**
  - PRECHARGE command closes the row and prepares the bank for the access
    - This precharge is on the critical path because it precedes read/write access
  - ACTIVATE command opens the row (bring data into the row buffer)
  - READ/WRITE command accesses the column in the row buffer

- Access to an open row
  - **READ/WRITE** command accesses the column in the row buffer

- If an access stream has a high locality, open page policy is helpful
  - Row buffer hits are cheap (it's made out of SRAM cells)
  - Row buffer miss is expensive because precharge is now on the critical path

Closed Page Policy

- Access to a closed row
  - **ACTIVATE** command opens row (→ row buffer)
  - **READ/WRITE** command accesses the column in the row buffer
  - **PRECHARGE** command closes the row and prepares the bank for the next access
    - This precharge can happen in the background and is not on the critical path

- If an access stream has low locality, precharging the bit lines immediately after access is helful for performance

Modern memory controllers use (proprietary) policies somewhere between the two extremes

DRAM Latency: Five Components

1. CPU to DRAM controller (request) transfer time after a load request misses everywhere in the cache hierarchy

2. DRAM Controller latency
   - Queuing and scheduling delay
   - Load/store request translation to DRAM COMMANDS

3. Request transfer time from the DRAM Controller to the selected channel/DIMM

4. Bank latency (*if there is no conflict*)
   - Row buffer hit: 20 ns (move data from row buffer to output pins)
   - Row buffer miss: 60 ns (PRECHARGE + READ + move data to pins)
   - Empty row buffer: 40 ns (READ + move data to pins)

open page policy

closed page policy, precharge immediately

5. Other delays
   - Bank conflict
   - Respecting timing constraints
   - Interconnect/wire

Address Mapping

Problem: We have a 32-bit address. We need to decide the bits we use for selecting the channel, bank, rows, column.

Hypothetical Example:

32-bit address 

Example solution: Use bits <3:5> for selecting one of the eight banks
Use bits <X:Y> for selecting one of the R rows

Bottomline: Depending on the address mapping, contiguous chunks of program data in virtual memory can end up in different banks or the same bank. Recall accesses across banks benefit from bank-level parallelism.

Two Concrete Policies

Assumptions: One channel, 2 GB, 8 banks, 16 K rows, 2K columns per bank
The channel bit is assumed 0 in the scenarios below (only 31 bits shown)

Row interleaving

- Large contiguous chunks (rows) of program data in consecutive banks
- Start filling the row and move to a different bank after 2^{14} bytes (16 KB)

Cache block interleaving

- Consecutive cache blocks (64 B) in consecutive banks
- Place eight 64-bit words (see first 6 bits below) in a one row/bank and then move to a different bank (see next three bits)

Interaction with Virtual Memory

- Operating system (OS) influences where a virtual page ends up in DRAM

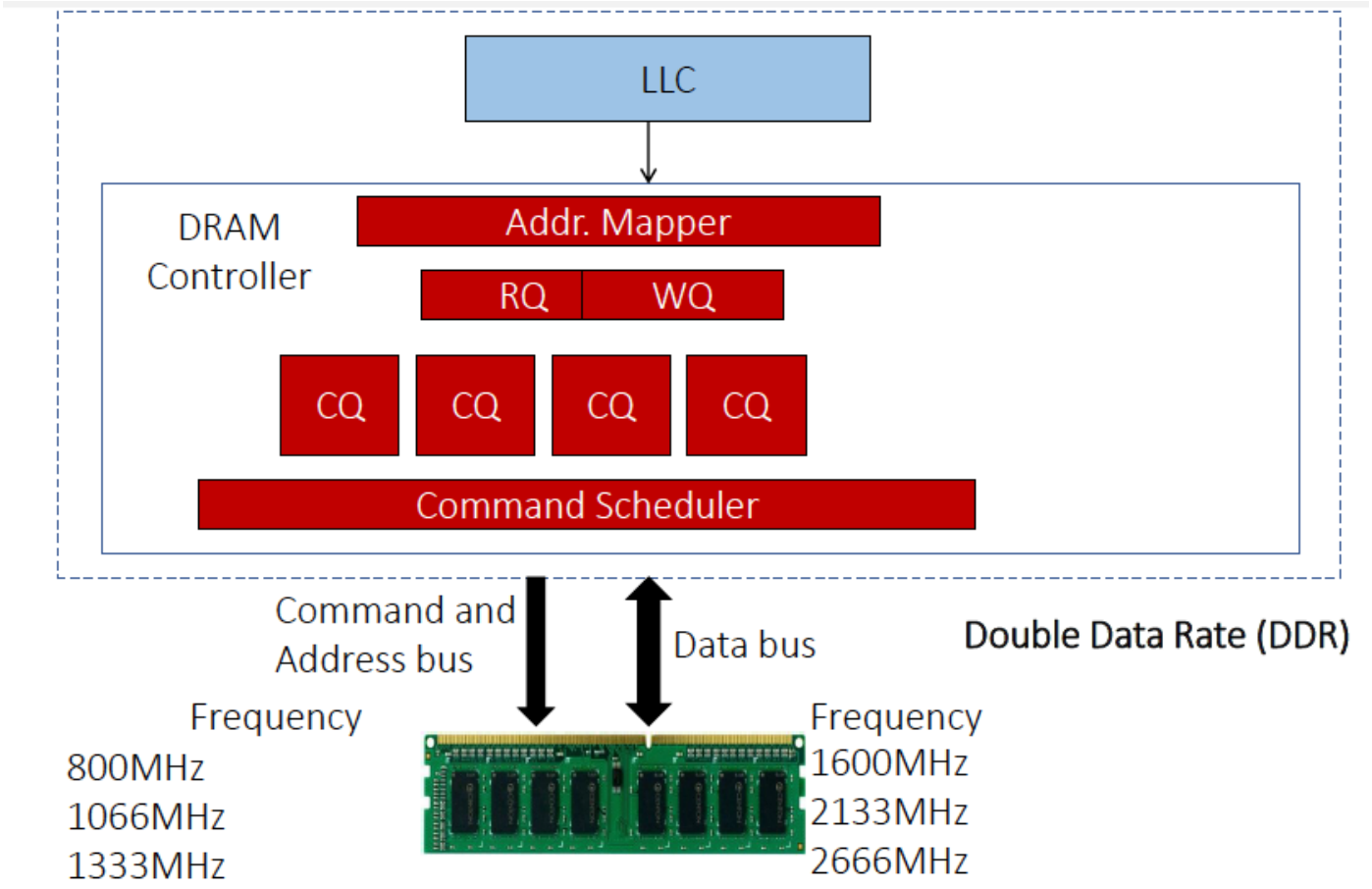
- OS can place consecutive virtual pages in the same row or different rows in different banks

DRAM Refresh

- Capacitors are leaky and must be refreshed
  - Typical refresh interval is 64 ms
  - ACTIVATE + PRECHARGE each row in a bank every 64 ms
  - DRAM refresh is a responsibility of the memory controller

- Performance implications of performance
  - DRAM bank is unavailable during a refresh operation
  - Program can experience long “pause time” due to memory unavailability
  - A serious concern for user-facing (latency-critical) applications such as web search or real-time services

Memory/DRAM Controller



Memory/DRAM Controller

- Placement of controller
  - **Today:** on the processor die (integrated Memory Controller or iMC)
    - Low latency compared to off-chip controller (consumes extra die area)
    - DRAM standards and processor must evolve together
  - **Old days:** outside the processor die (part of the chipset)
    - Decouples DRAM standards/types from the processor
    - Higher access latency

Popular Scheduling Policies

- Tasks of a controller
  - Respect DRAM timing constraints & ensure correct operation (e.g., refresh)
  - Schedule requests to optimize for latency and throughput
- Two popular scheduling policies
  - FCFS (first come, first serve)
    - Issue the first read/write in the queue that is ready for issue
  - FR-FCFS (first ready-first come, first serve)
    - First, prioritize row buffer hits, then prioritize oldest requests

Exercise

For the following access stream, estimate the time it takes for the memory request to finish for three scheduling policies: (1) Open-page, (2) Closed-page, and (3) Oracular. The Oracular policy dynamically switches between open and closed policies based on prior knowledge of the access stream. X, X+1, X+2, X+3 map to the same row, and Y, Y+1 map to a different row in the same bank. Access to an open row (row buffer hit) takes 20 ns, access to a closed row if another row is already open (row buffer conflict) takes 60 ns, and access to an empty row buffer (bit lines are precharged already) takes 40 ns.

Request	Arrival Time	Open	Closed	Oracular
X	0 ns			
Y	30 ns			
X+1	100 ns			
X+3	210 ns			
Y+1	250 ns			
X+2	330 ns			