CS 222 Computer Organization & Architecture

Lecture 30 [12.04.2019]

DRAM Organization

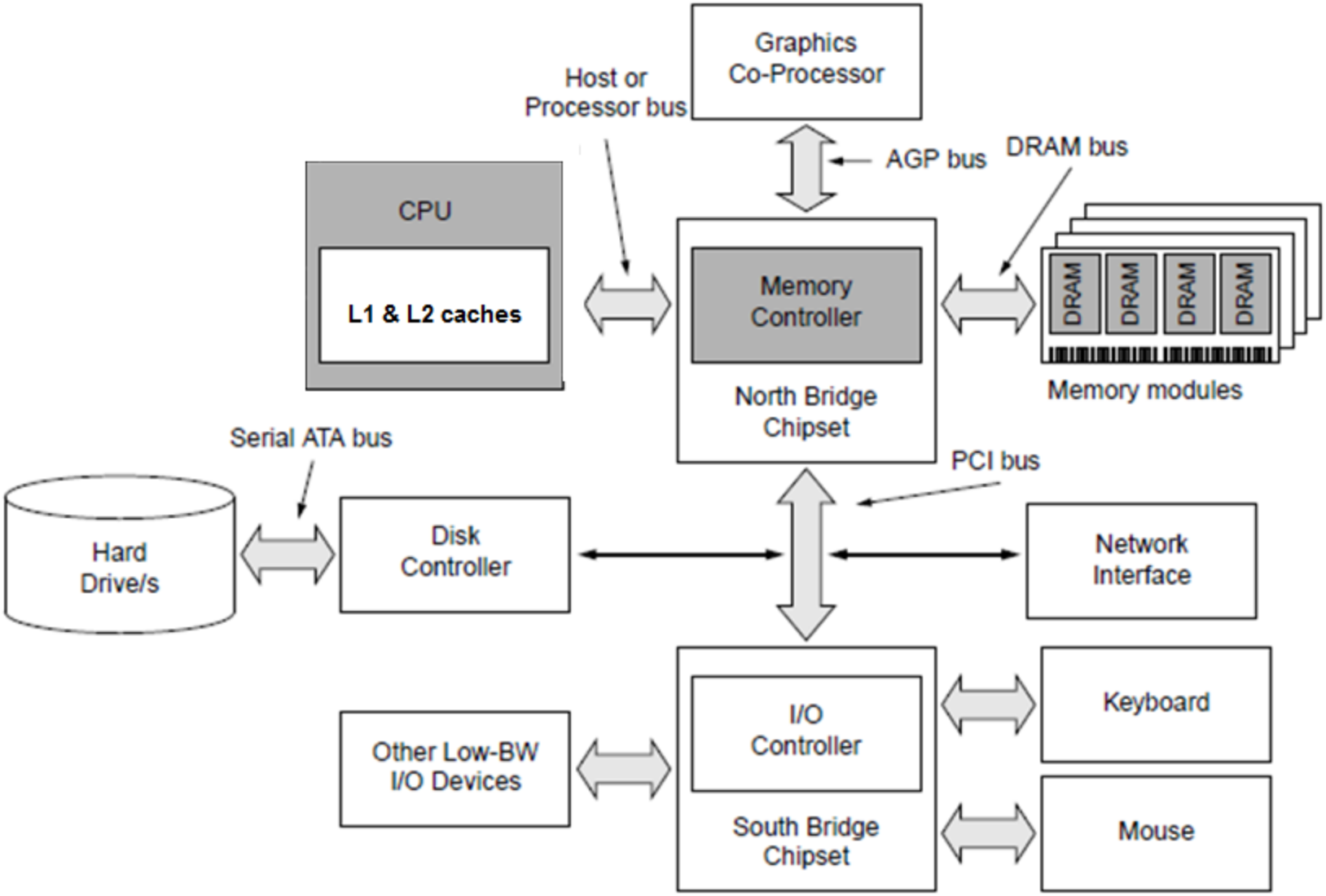
John Jose

Assistant Professor

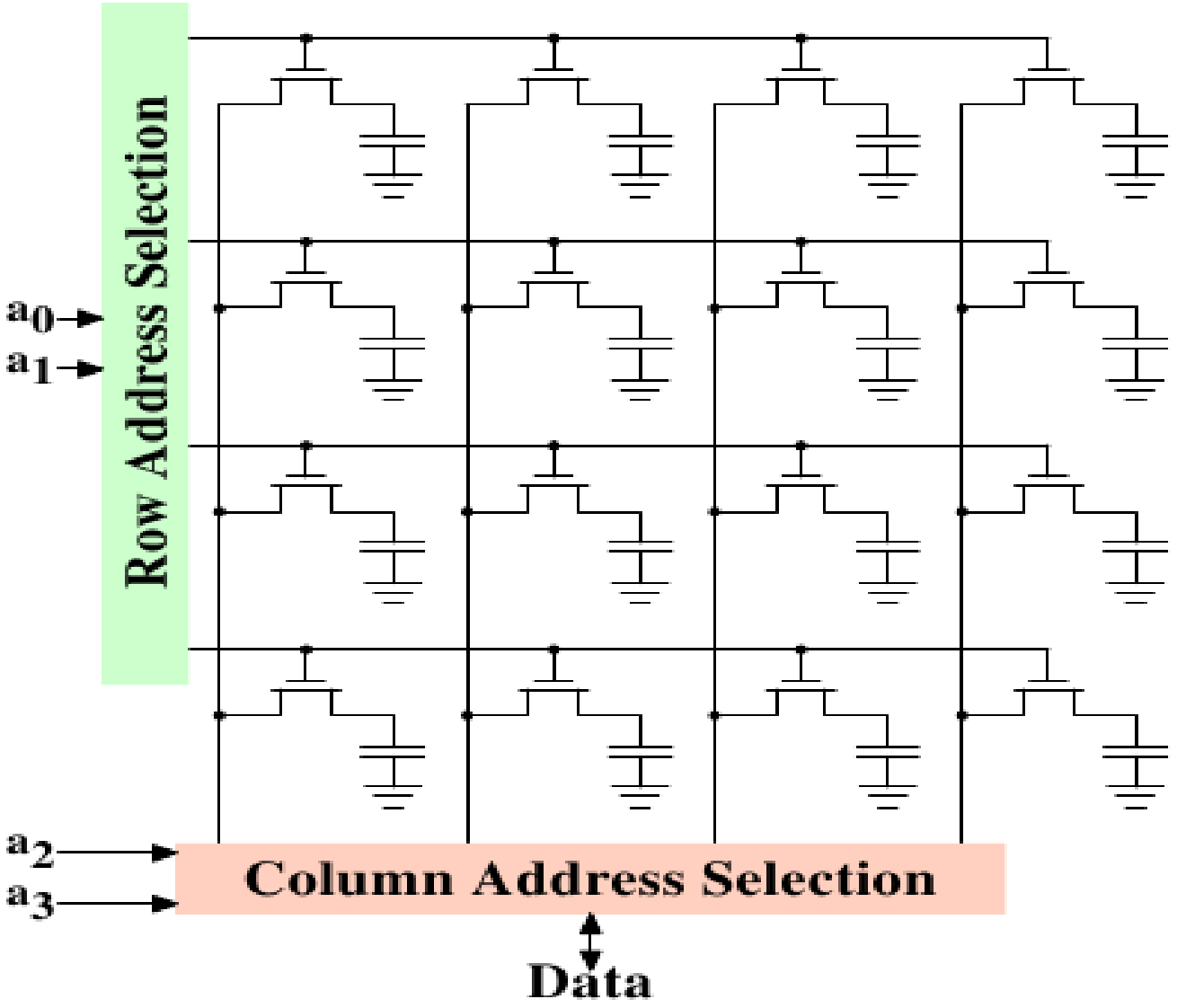
Department of Computer Science & Engineering

Indian Institute of Technology Guwahati, Assam.

Components of a Modern Computer



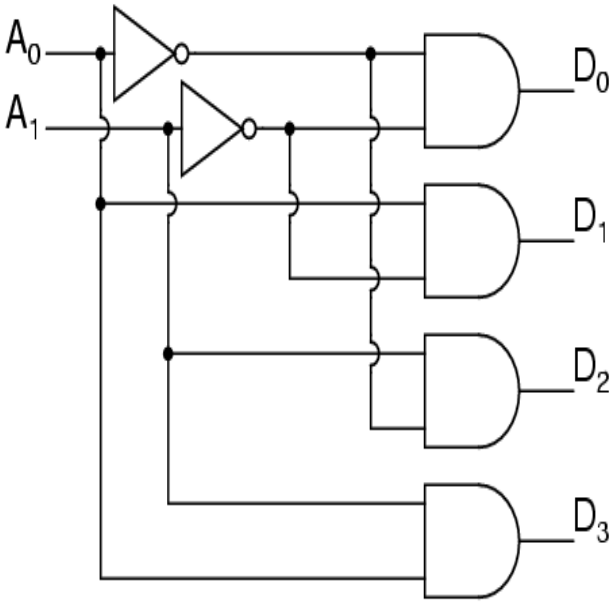
DRAM (Dynamic Random Access Memory)

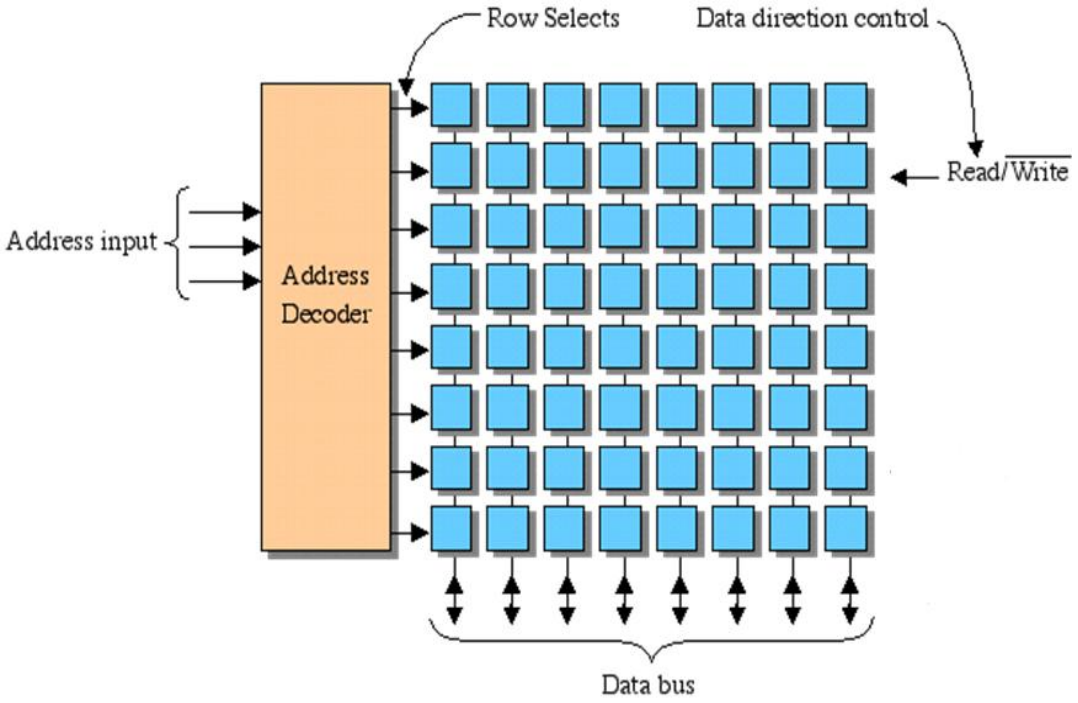


Basic Terminologies

- Physical address space
  - Maximum size of main memory
  - Total number of uniquely identifiable locations
- Physical addressability
  - Minimum size of data in memory can be addressed
  - Byte-addressable, word-addressable, multibyte addressable
- Alignment
  - Does the hardware support unaligned access transparently to software?

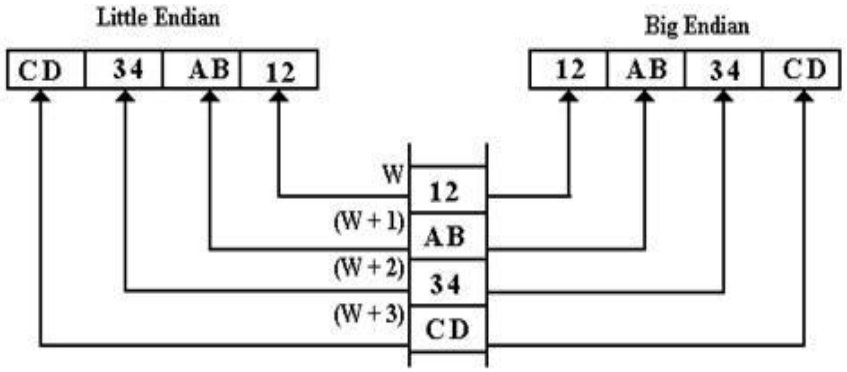
How memory works ?

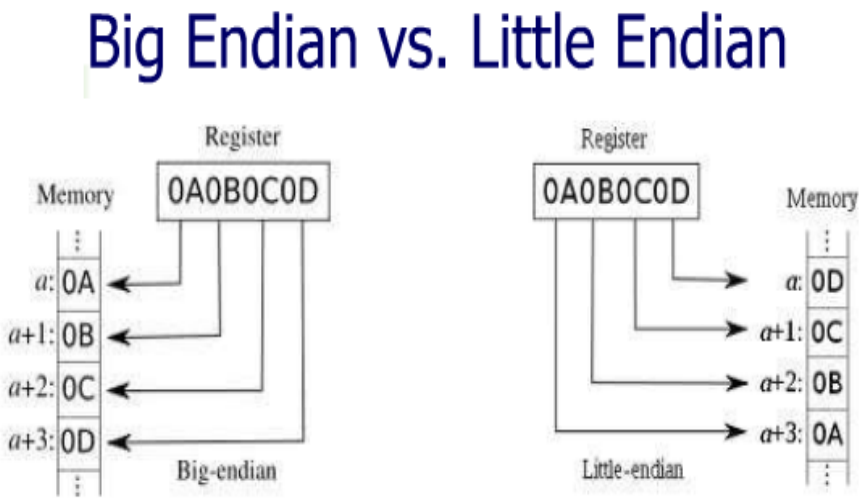




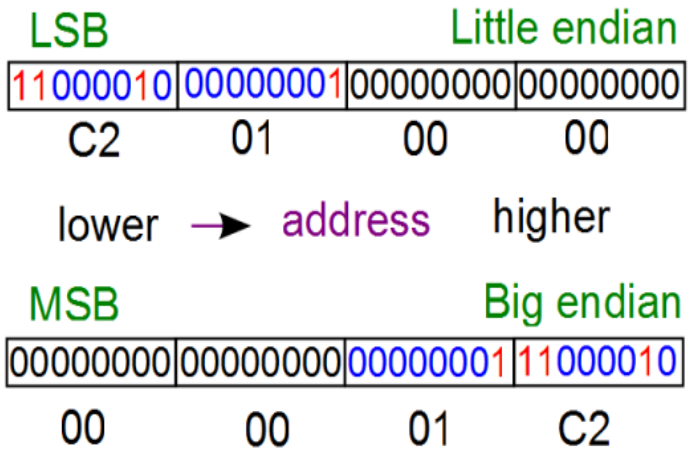
Byte Ordering

- **Big Endian vs Little Endian**

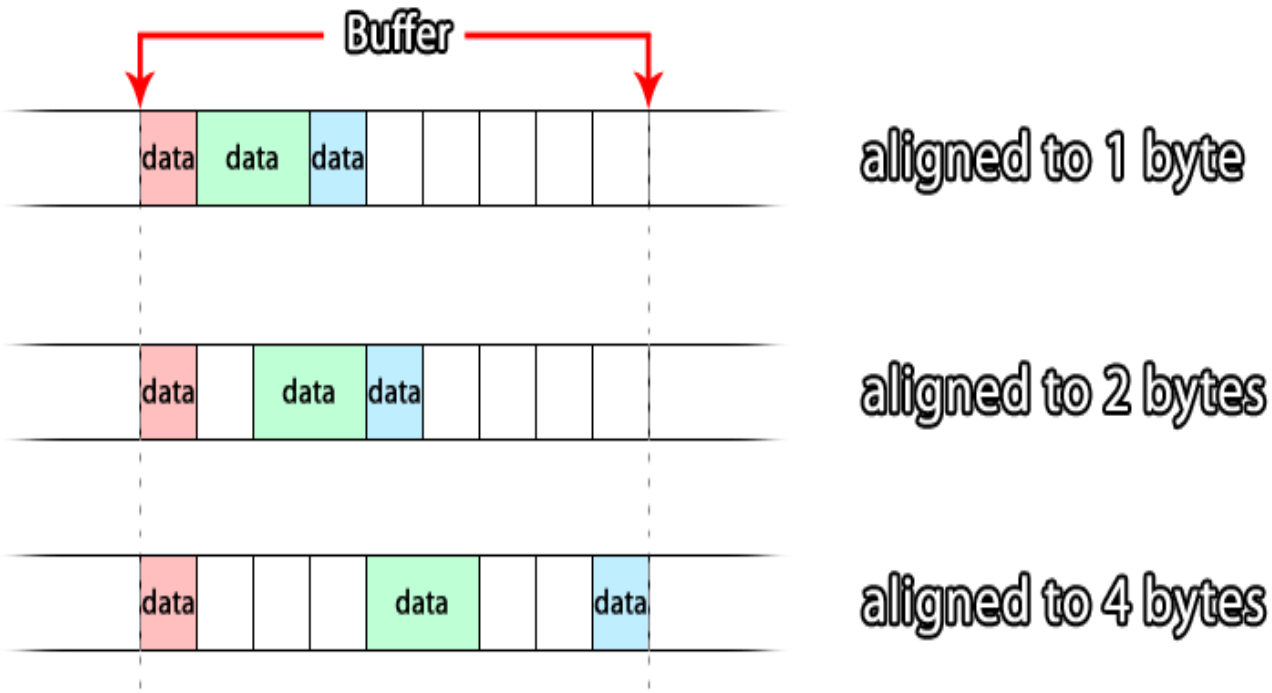




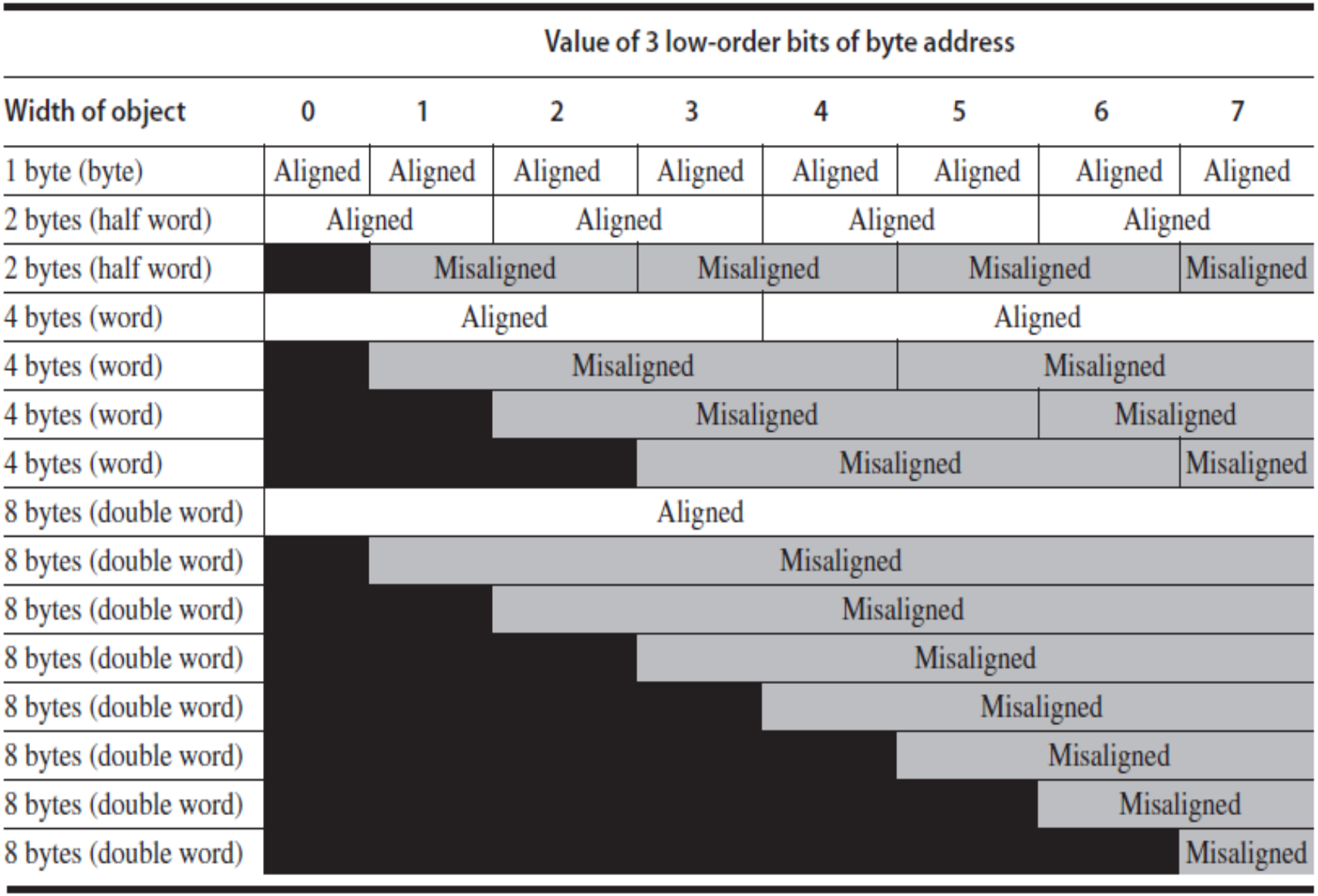
Int i = 450 = $2^8 + 2^7 + 2^6 + 2$ = x000001C2



Byte / Word Alignment



Byte /Word Alignment

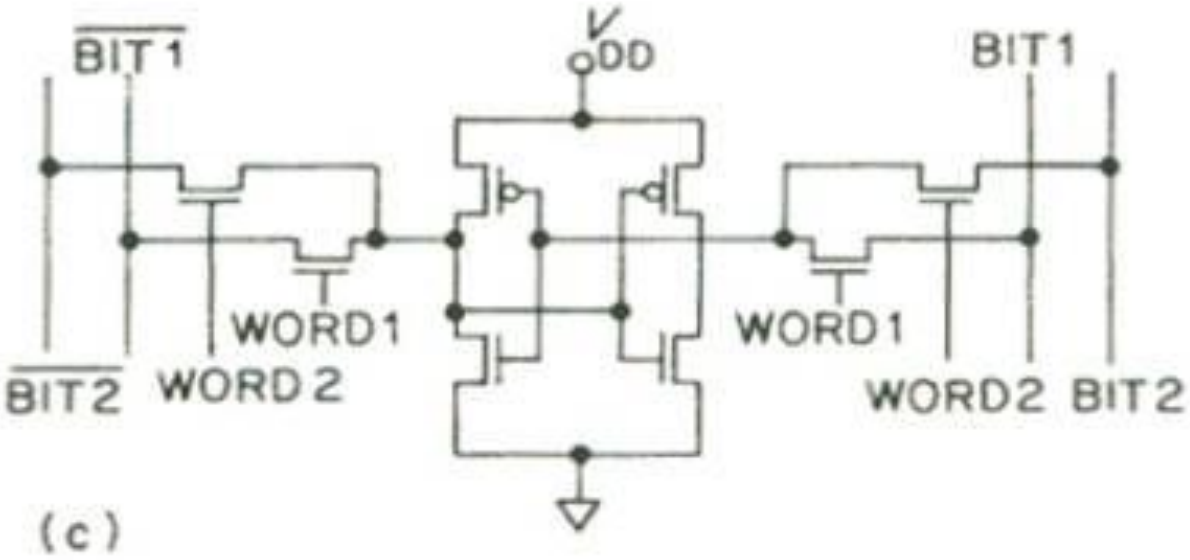


Enabling High Bandwidth Memories

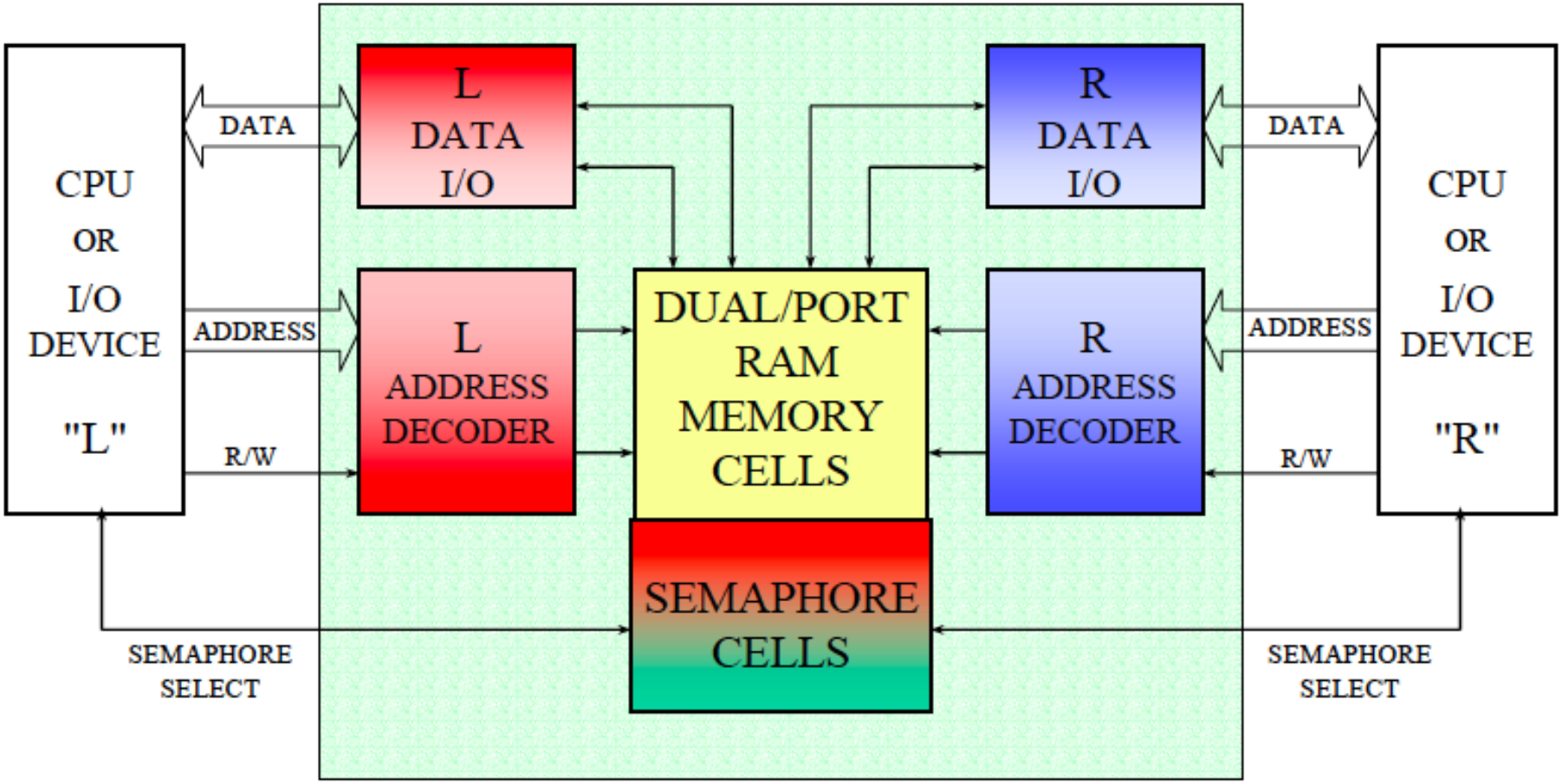
- Multiple Instructions per Cycle can generate multiple cache/memory accesses per cycle
- How do we ensure the cache/memory can handle multiple accesses in the same clock cycle?
- Solutions:
  - Multi-porting
  - Banking (interleaving)

Multiporting

- Each memory cell has multiple read or write ports
- Truly concurrent accesses (no conflicts on reads)
- Expensive in terms of latency, power, area
- How read and write to the same location at same time?
  - Peripheral logic needs to handle this



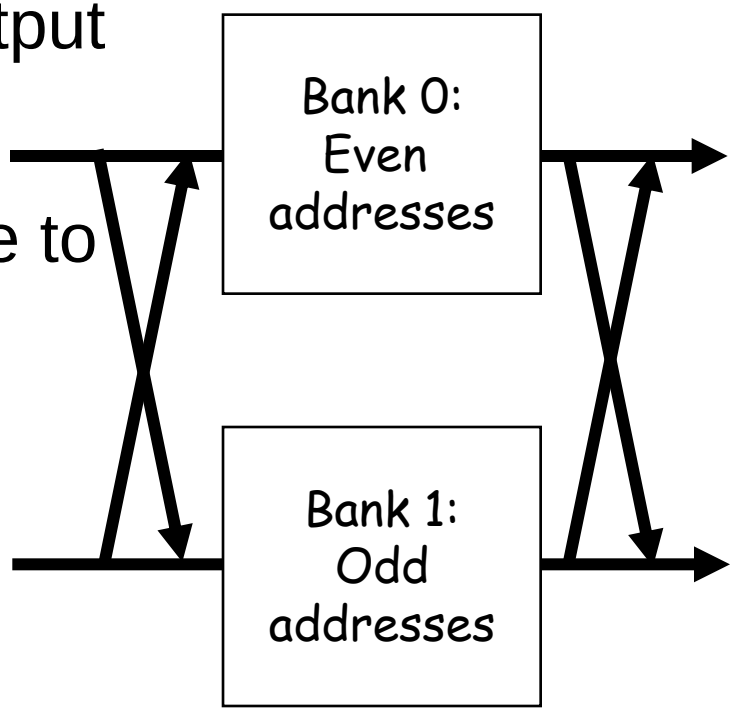
Peripheral Logic for Multiporting



Interleaving

- Address space partitioned into separate banks
- No increase in data store area
- Bits in address determines which bank an address maps
- Cannot satisfy multiple accesses to the same bank
- Crossbar interconnect in input/output

- Bank conflicts - Two accesses are to the same bank difficult to handle

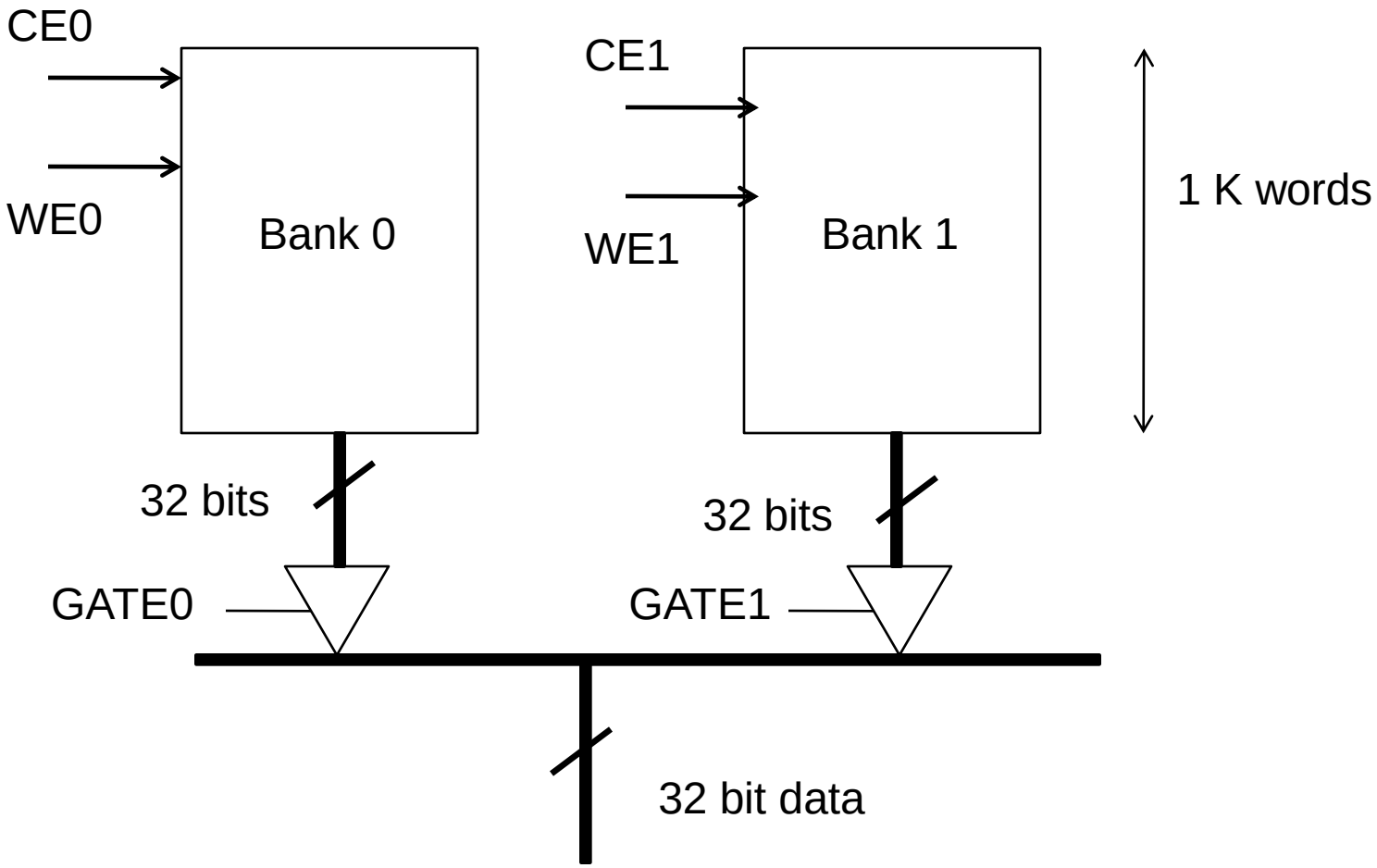


Interleaving

- 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 entire memory into multiple banks that can be accessed independently (in the same cycle or in consecutive cycles)
- Accesses to different banks can be overlapped
- A Key Issue: How do you map data to different banks?

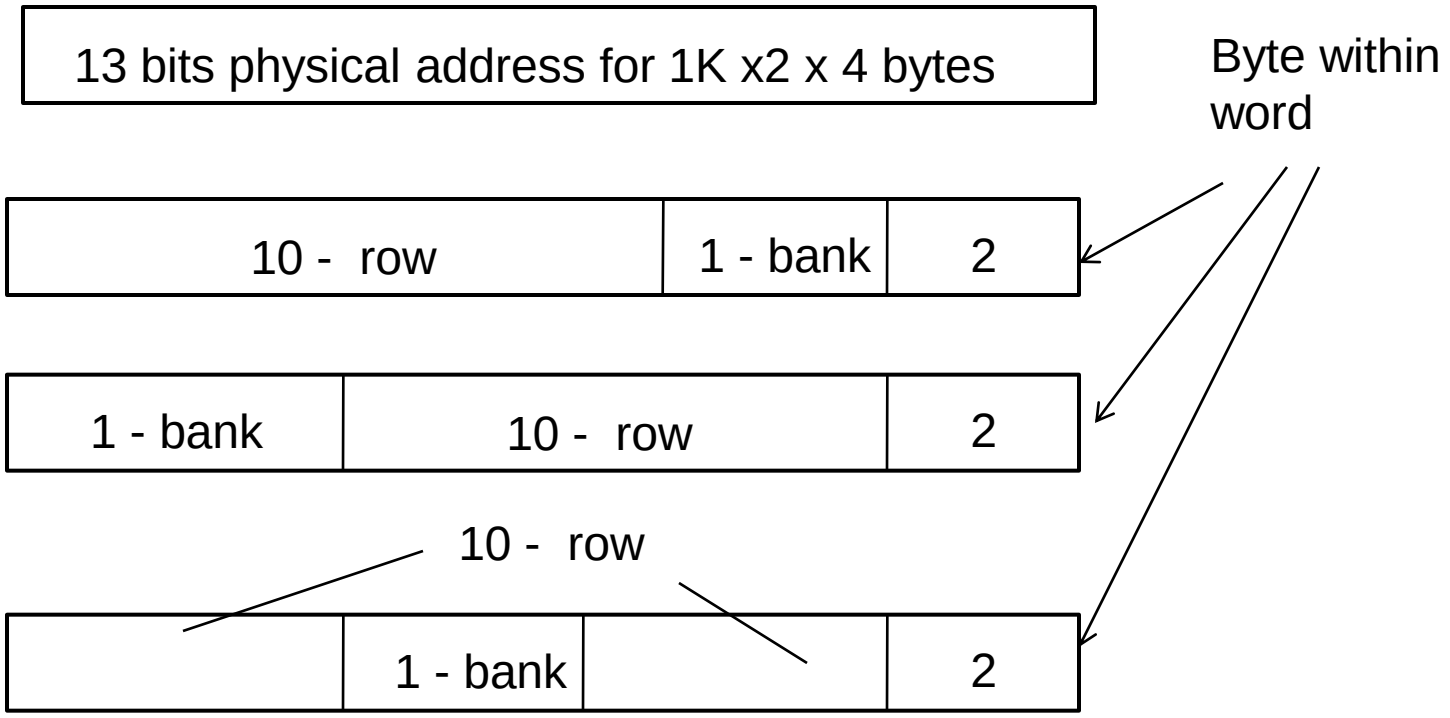
Interleaving

- Assume each bank supplies a word.
- Adjacent words are interleaved across banks



Interleaving

- Physical address split up for interleaving.
- Which bank do consecutive words in memory mapped to?

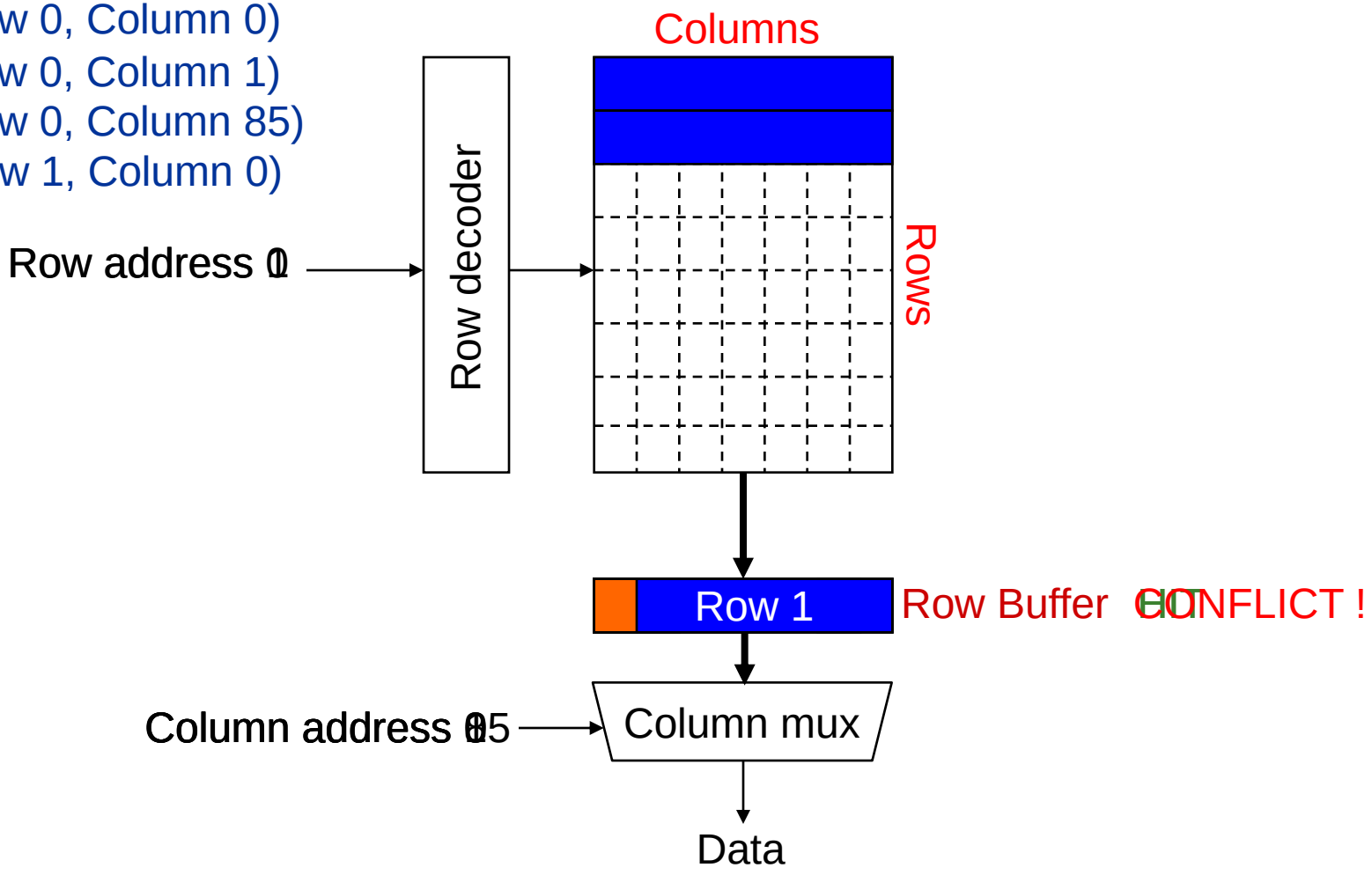


Page Mode DRAM

- A DRAM bank is a 2D array of cells: rows x columns
- A DRAM row is also called a DRAM page
- Sense amplifiers read values to the row buffer
- Each address is a <row,column> pair
- **Access to a closed row**
  - Activate (RAS) - opens row (placed into row buffer)
  - Read/write (CAS)- access column in the row buffer
  - Precharge (PRE) - closes the row and prepares the bank for next access
- **Access to an open row**
  - No RAS only CAS needed

DRAM Bank Operation

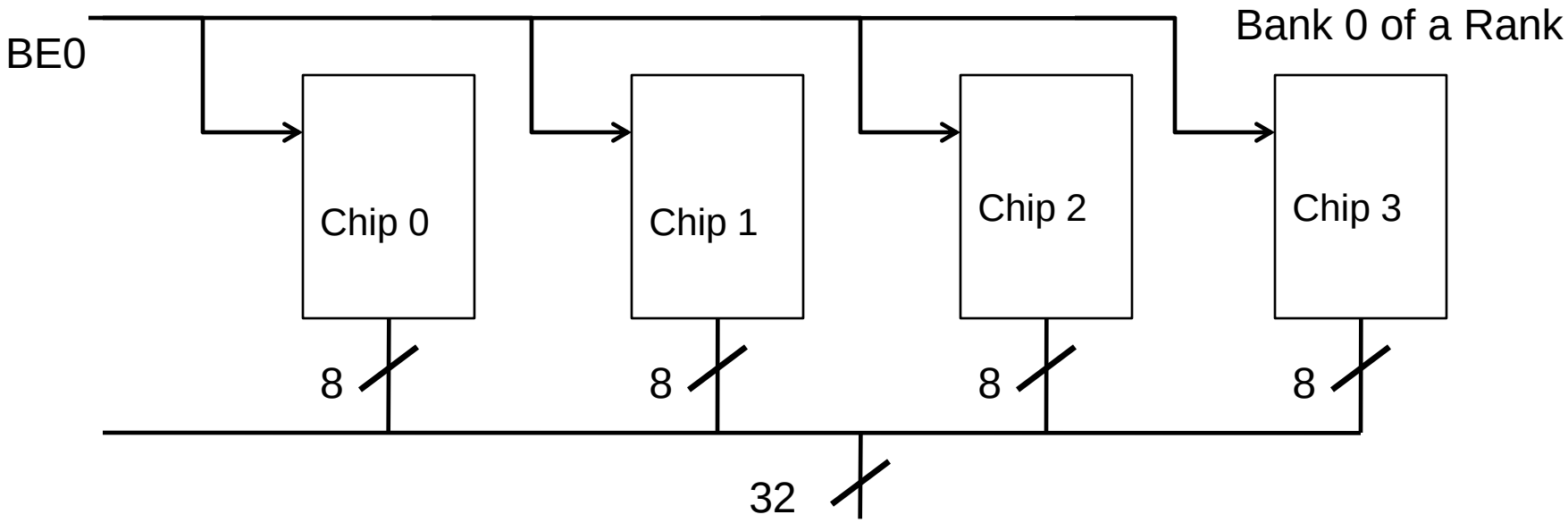
Access Address:
(Row 0, Column 0)
(Row 0, Column 1)
(Row 0, Column 85)
(Row 1, Column 0)



DRAM Subsystem Organization

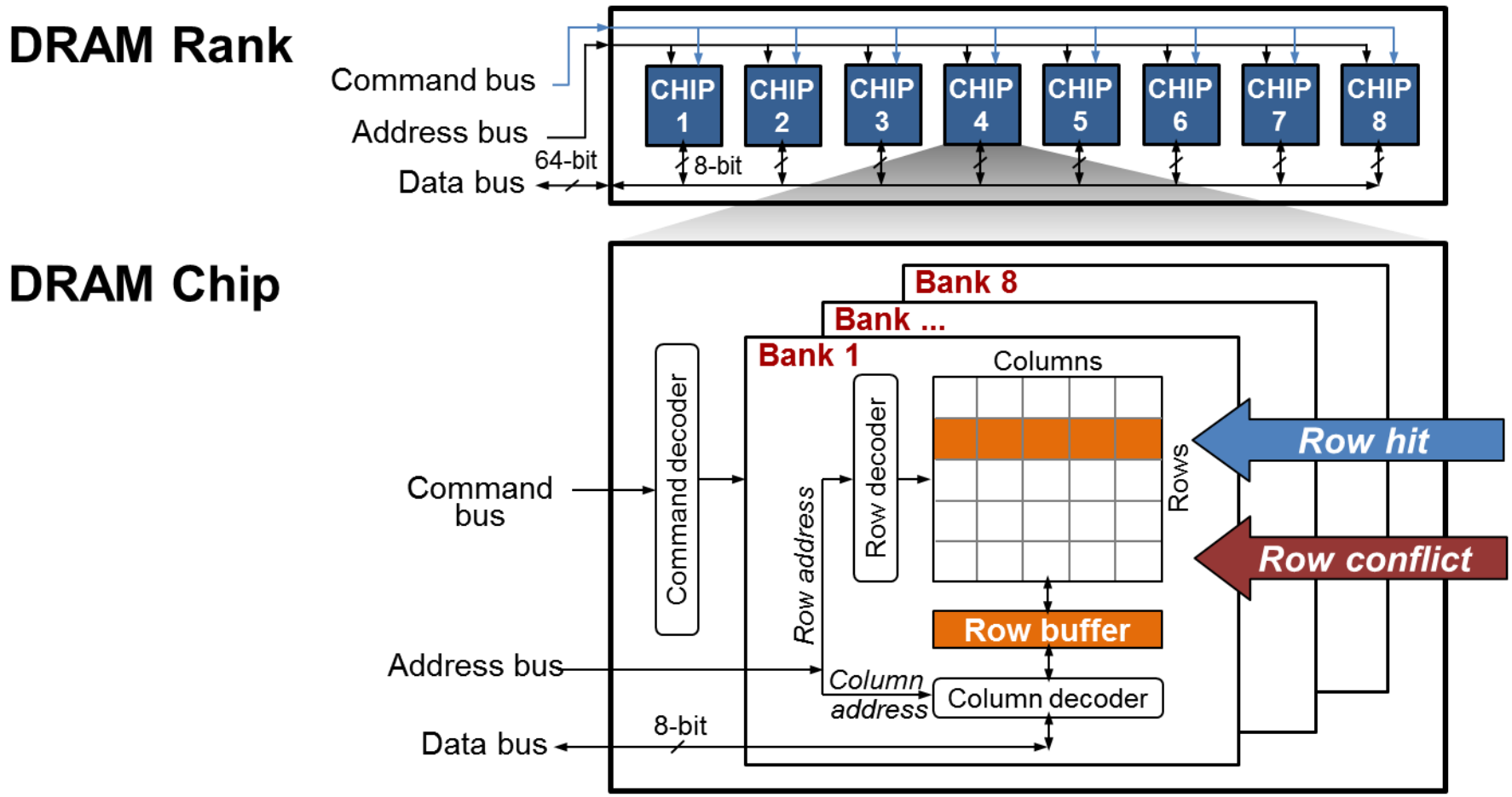
- Channel
- DIMM
- Rank
- Chip
- Bank
- Row
- Column
- B-Cell

DRAM Rank



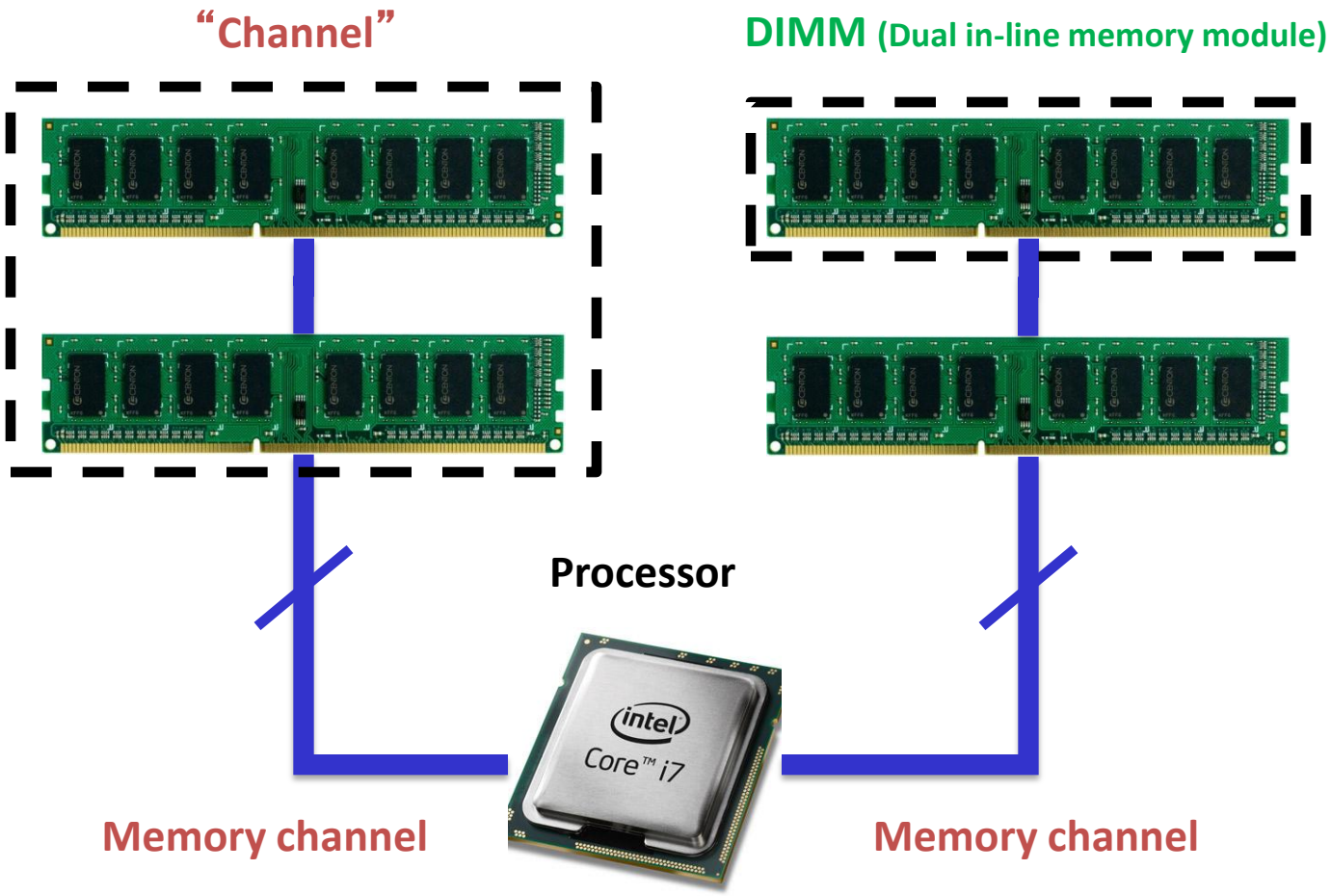
- Rank : A set of chips that respond to same command and same address at the same time but with different pieces of the requested data.
- Easy to produce 8 bit chip than 32 bit chip.
- Produce an 8 bit chip but control and operate them as a rank to get a 32 bit data in a single read.

DRAM Rank



DRAM access latency varies depending on which row is stored in the row buffer

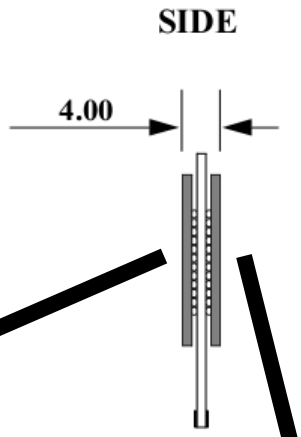
The DRAM subsystem



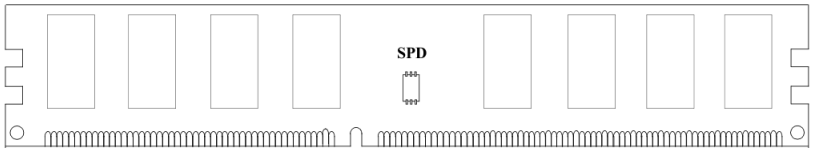
Breaking down a DIMM

DIMM (Dual in-line memory module)

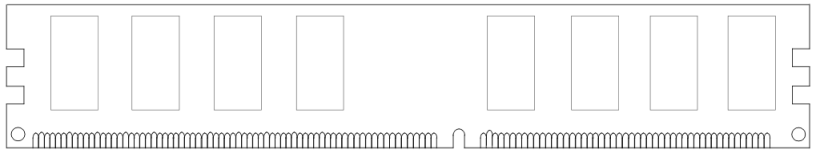
Side view



Front of DIMM



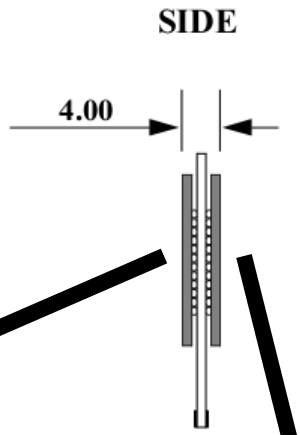
Back of DIMM



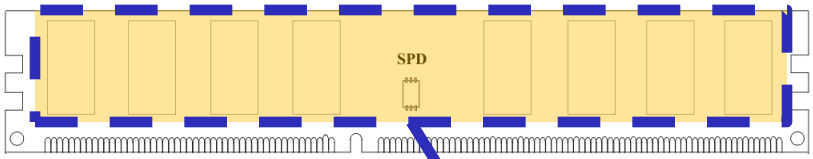
Breaking down a DIMM

DIMM (Dual in-line memory module)

Side view

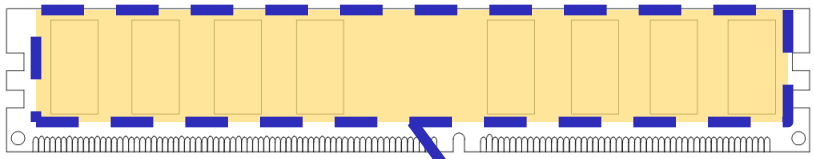


Front of DIMM



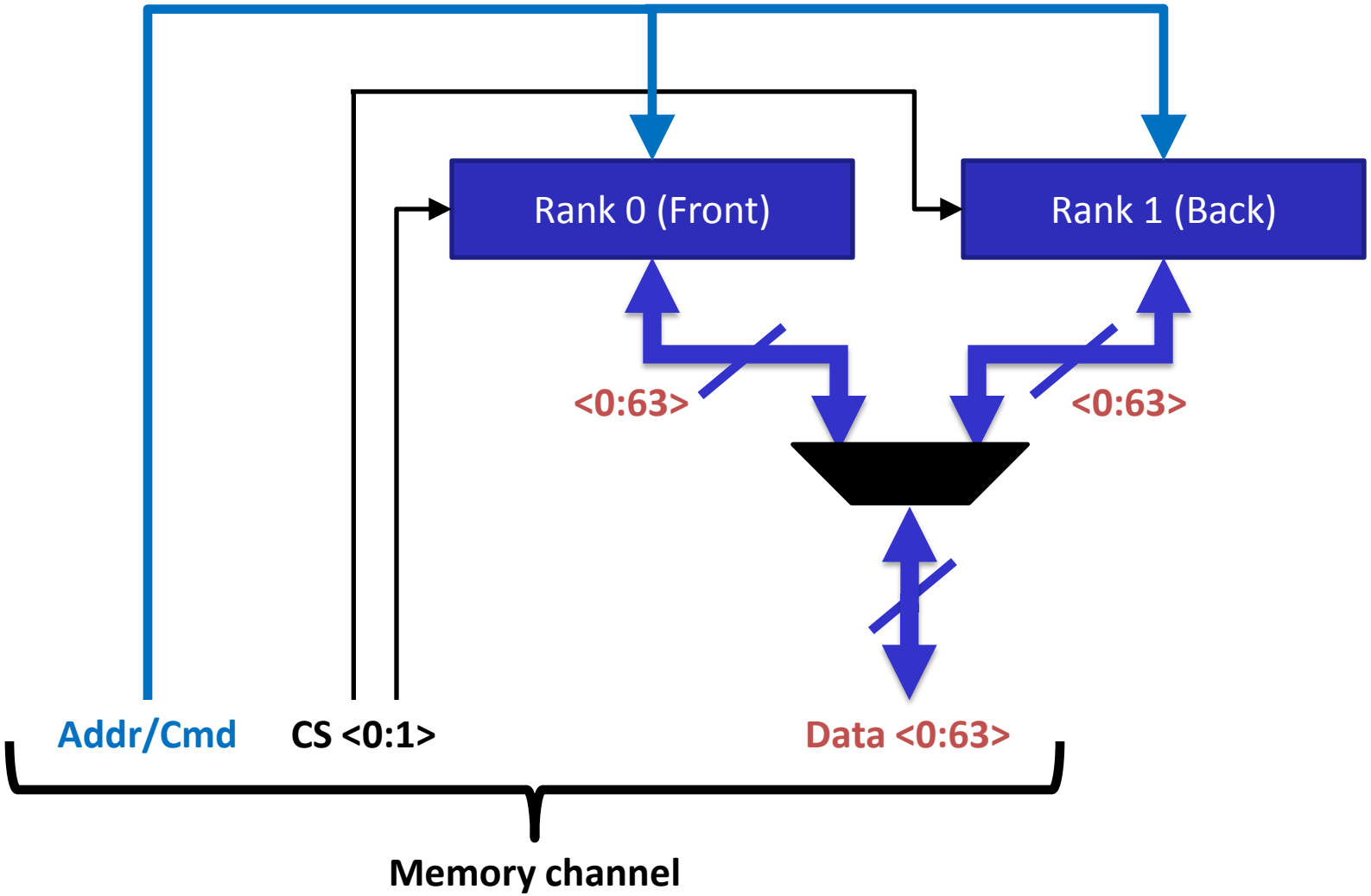
Rank 0: collection of 8 chips

Back of DIMM

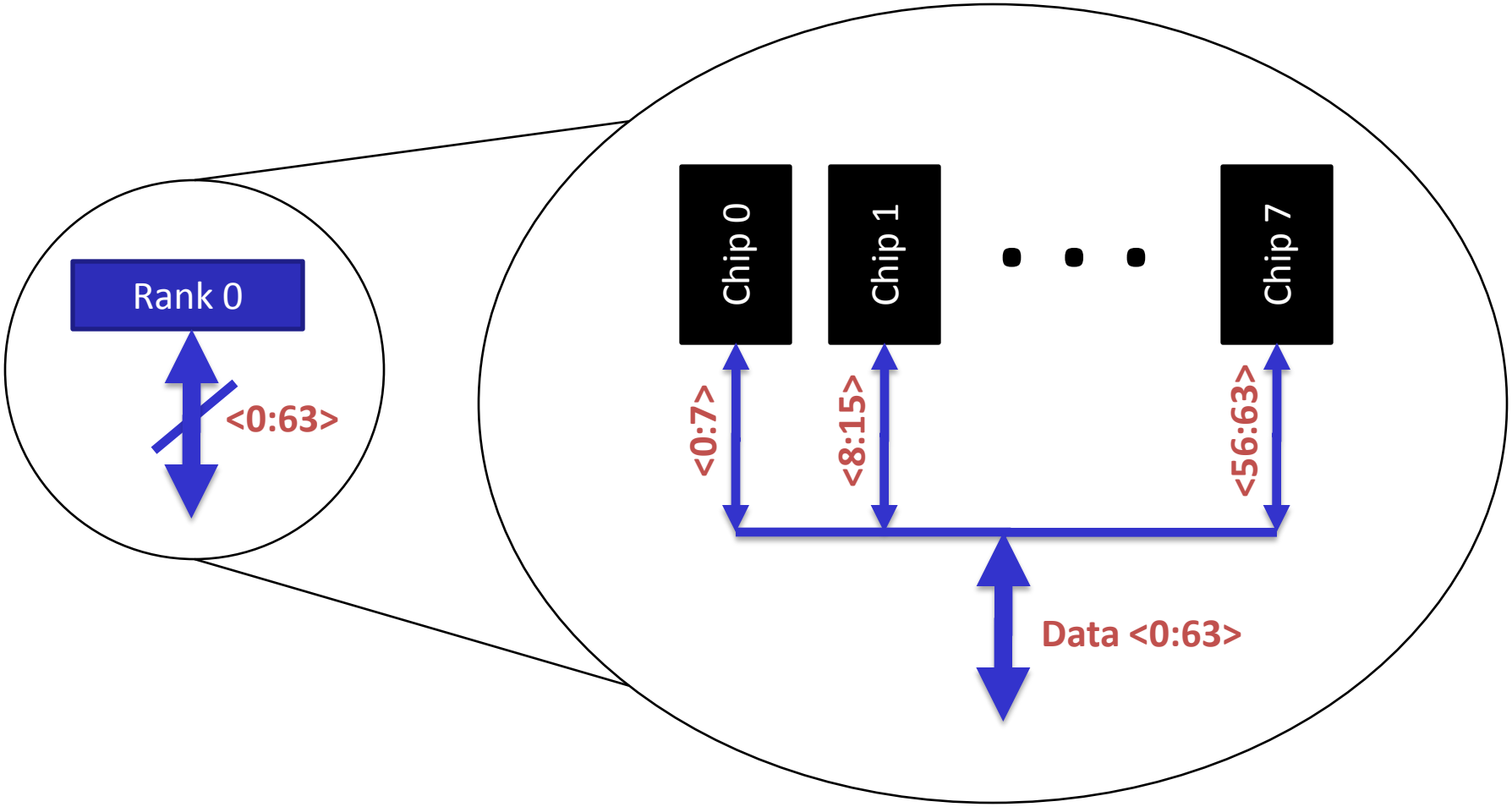


Rank 1

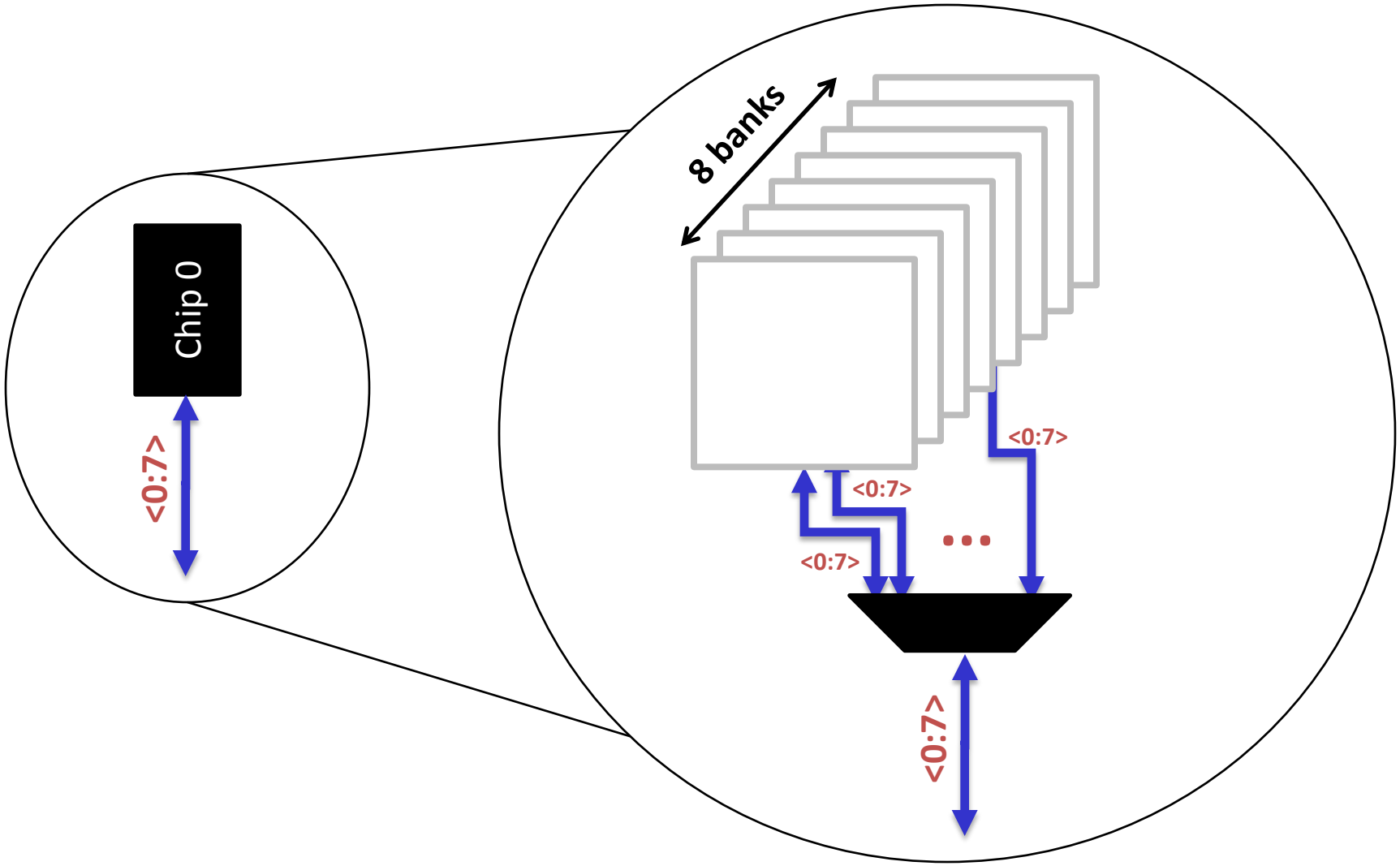
Rank


Rank 0 (Front)
Rank 1 (Back)
<0:63>
<0:63>
Addr/Cmd
CS <0:1>
Data <0:63>
Memory channel

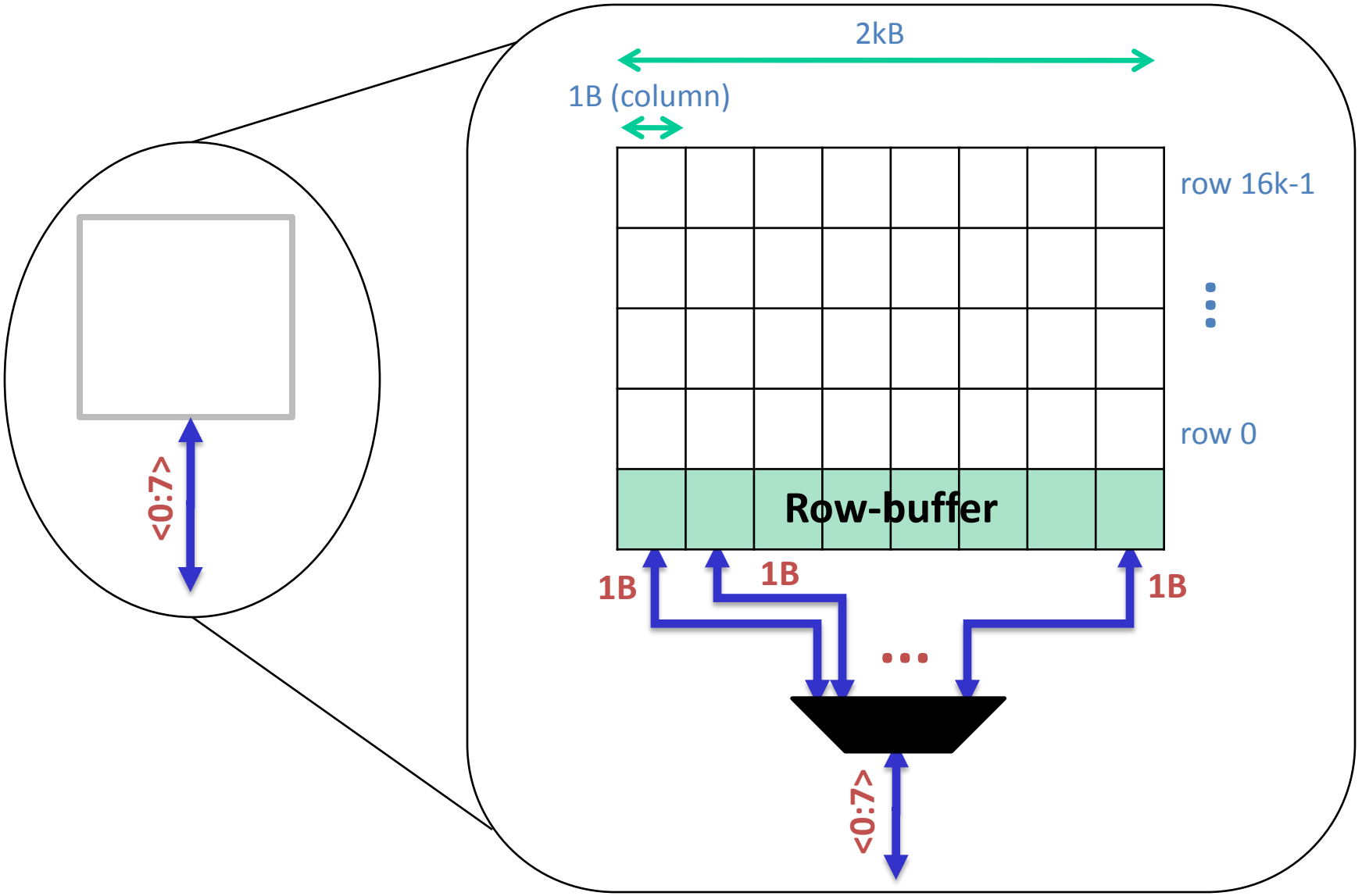
Breaking down a Rank



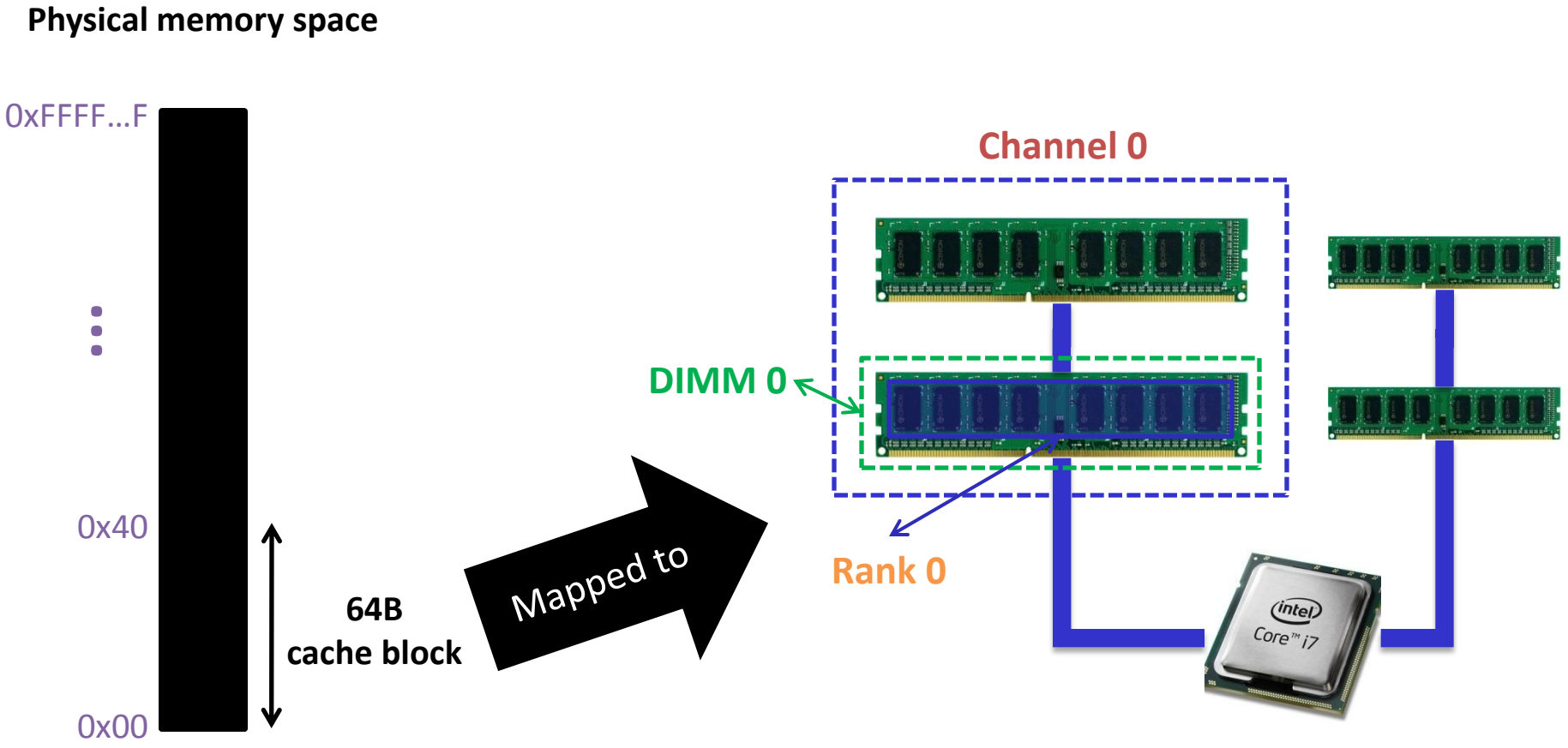
Breaking down a Chip



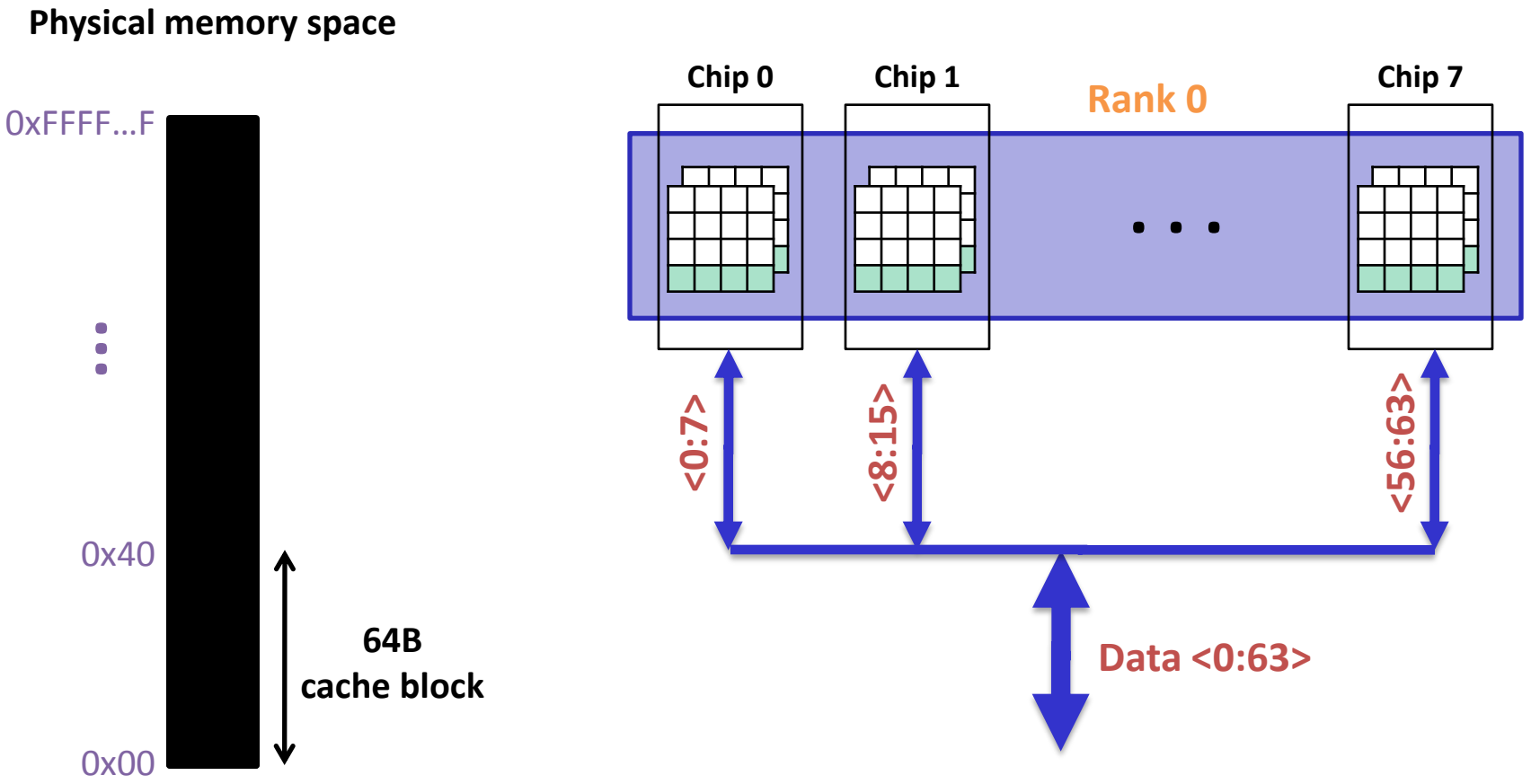
Breaking down a Bank



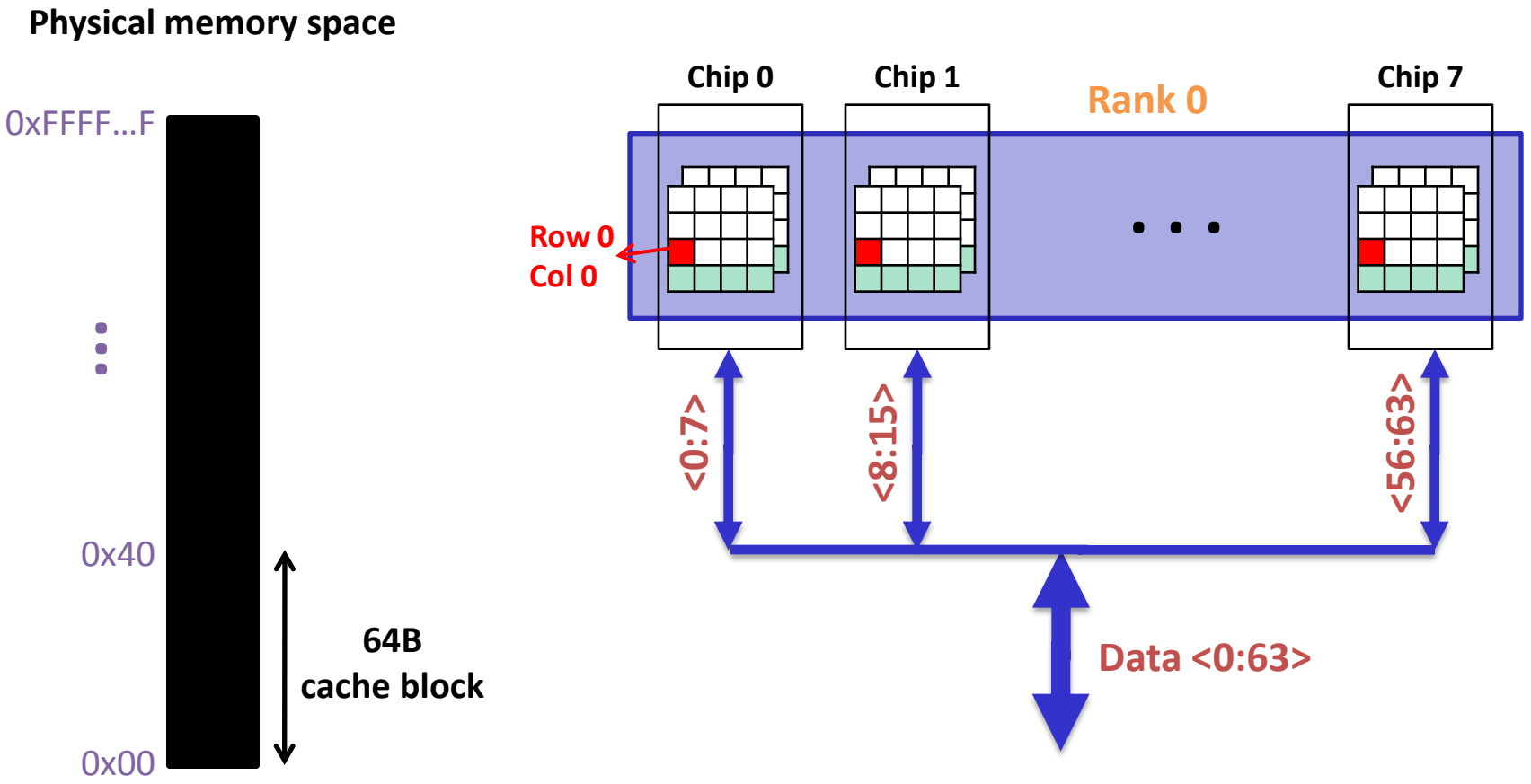
Transferring a cache block



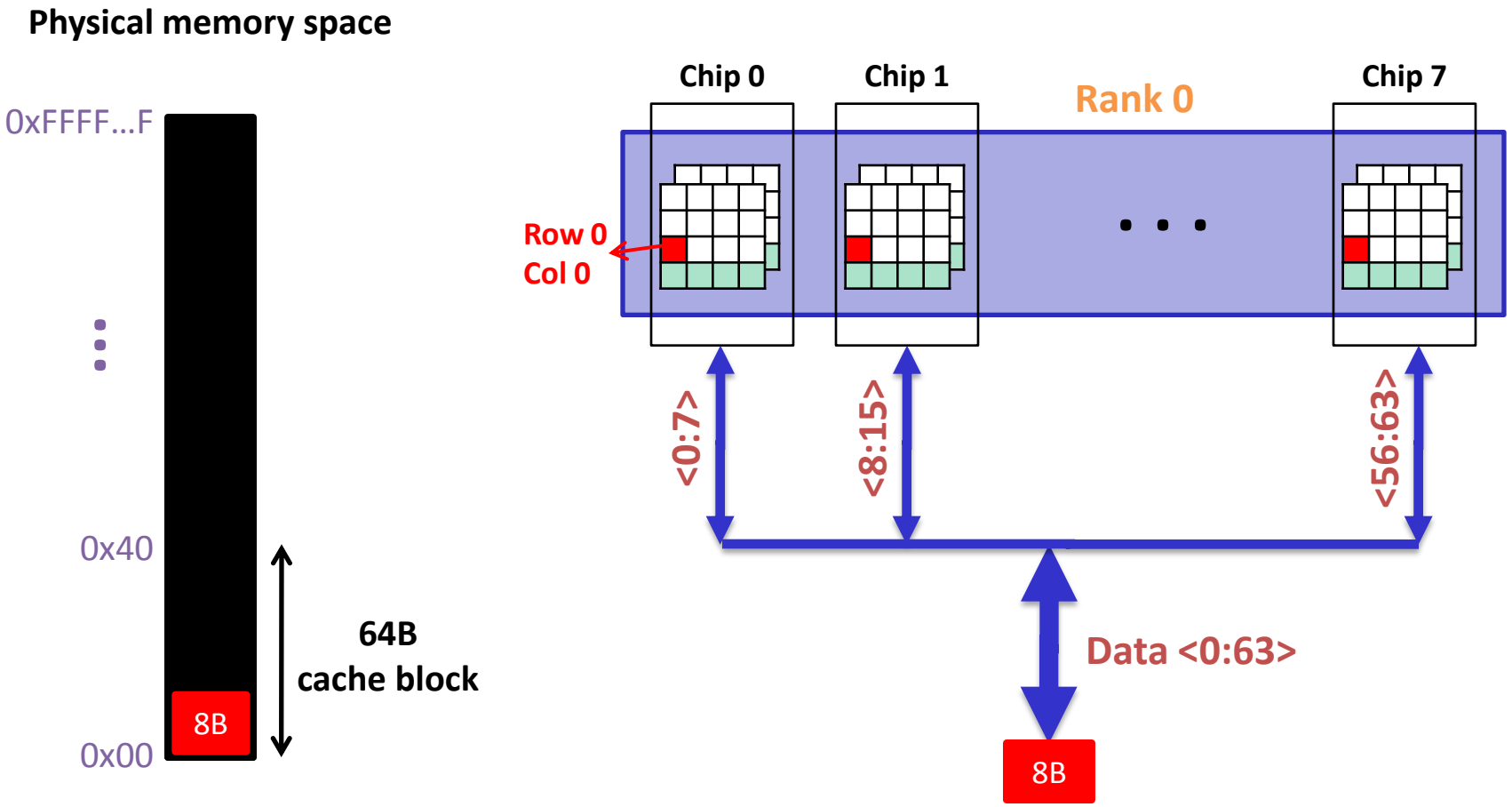
Transferring a cache block



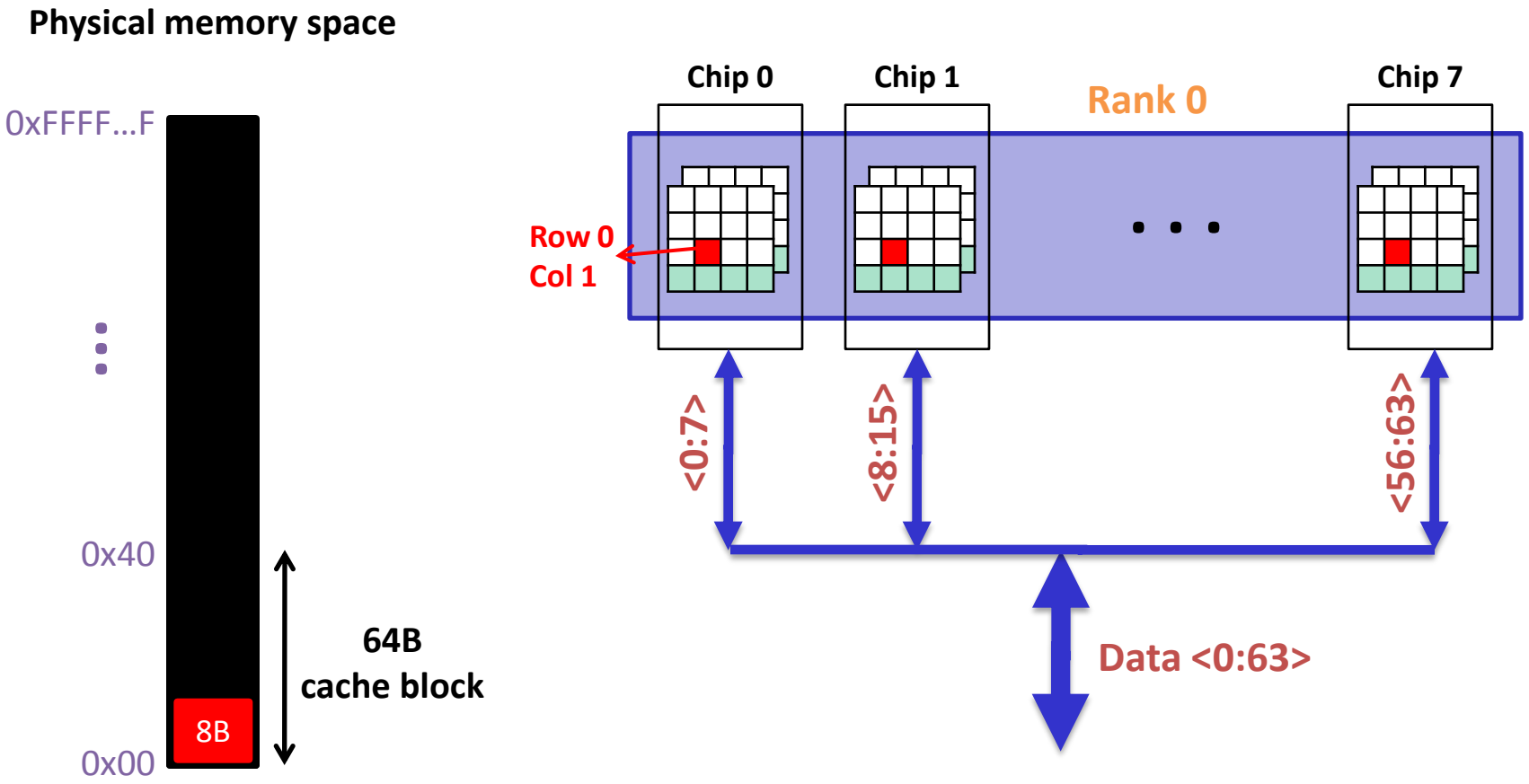
Transferring a cache block



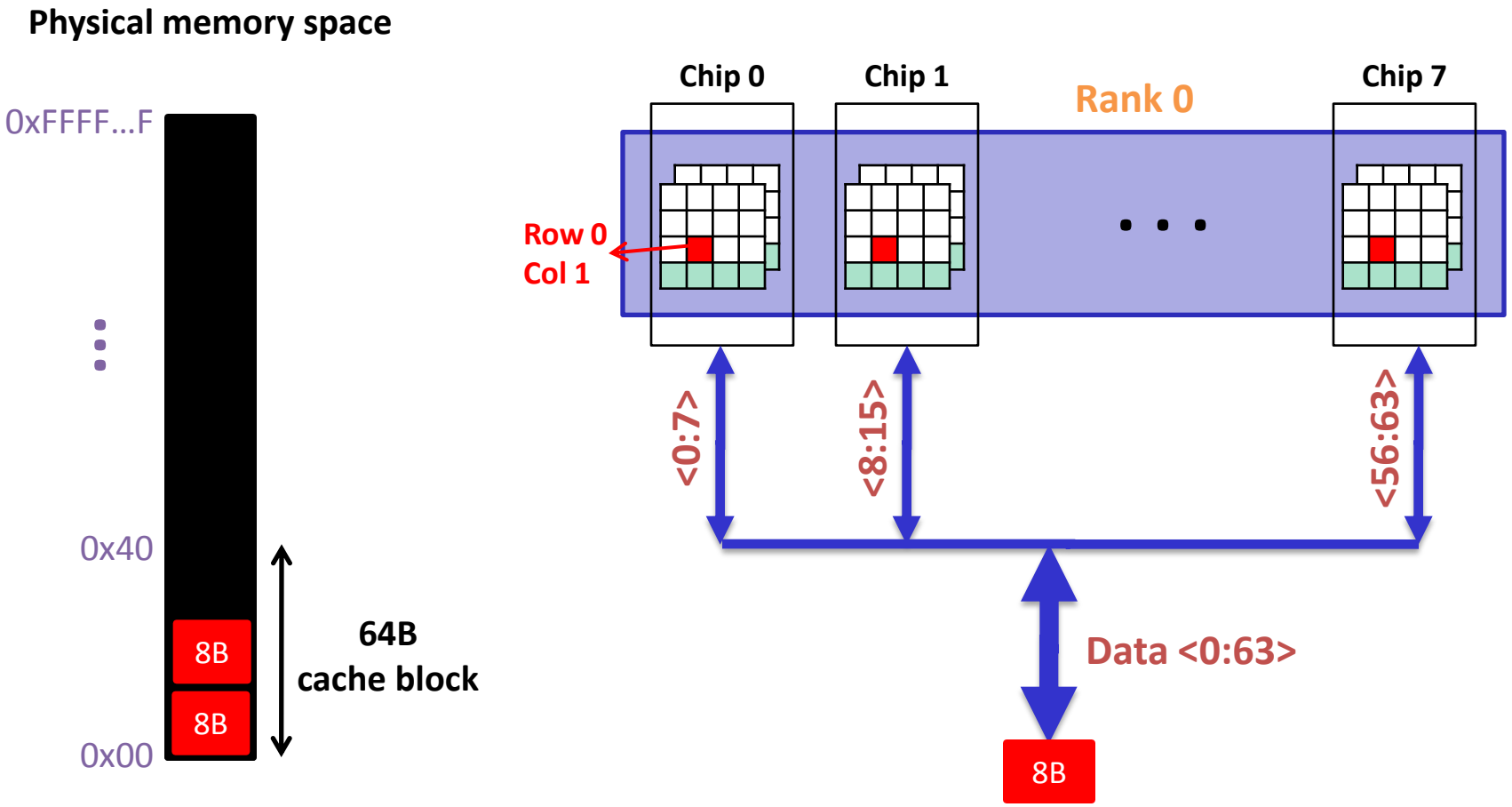
Transferring a cache block



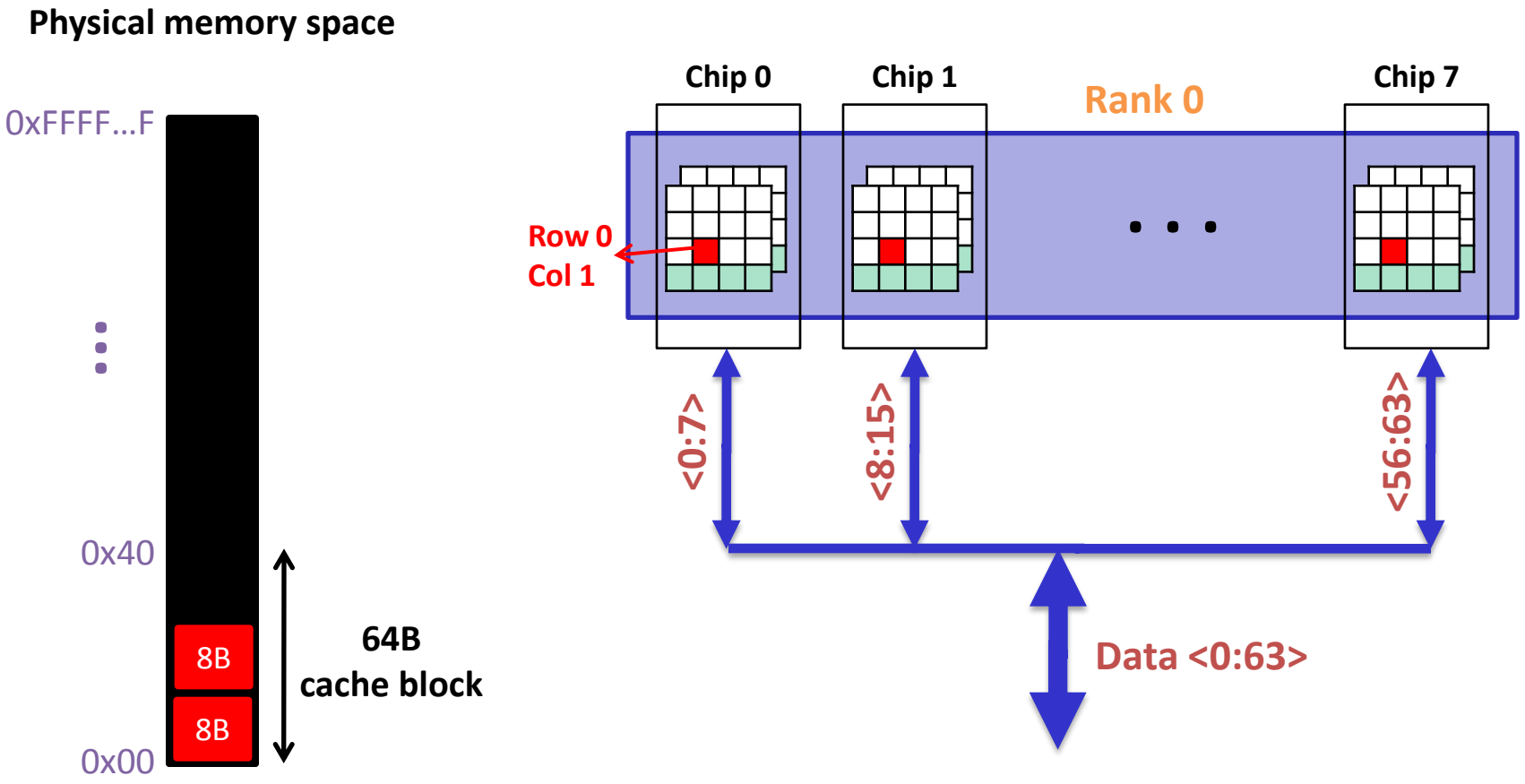
Transferring a cache block



Transferring a cache block



Transferring a cache block



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

johnjose@iitg.ac.in
http://www.iitg.ac.in/johnjose/