CS698Y: Modern Memory Systems
Lecture-15 (DRAM Organization)

Biswabandan Panda

biswap@cse.iitk.ac.in

https://www.cse.iitk.ac.in/users/biswap/CS698Y.html

DRAM Organization

Channel

DIMM

Rank

Chip

Bank

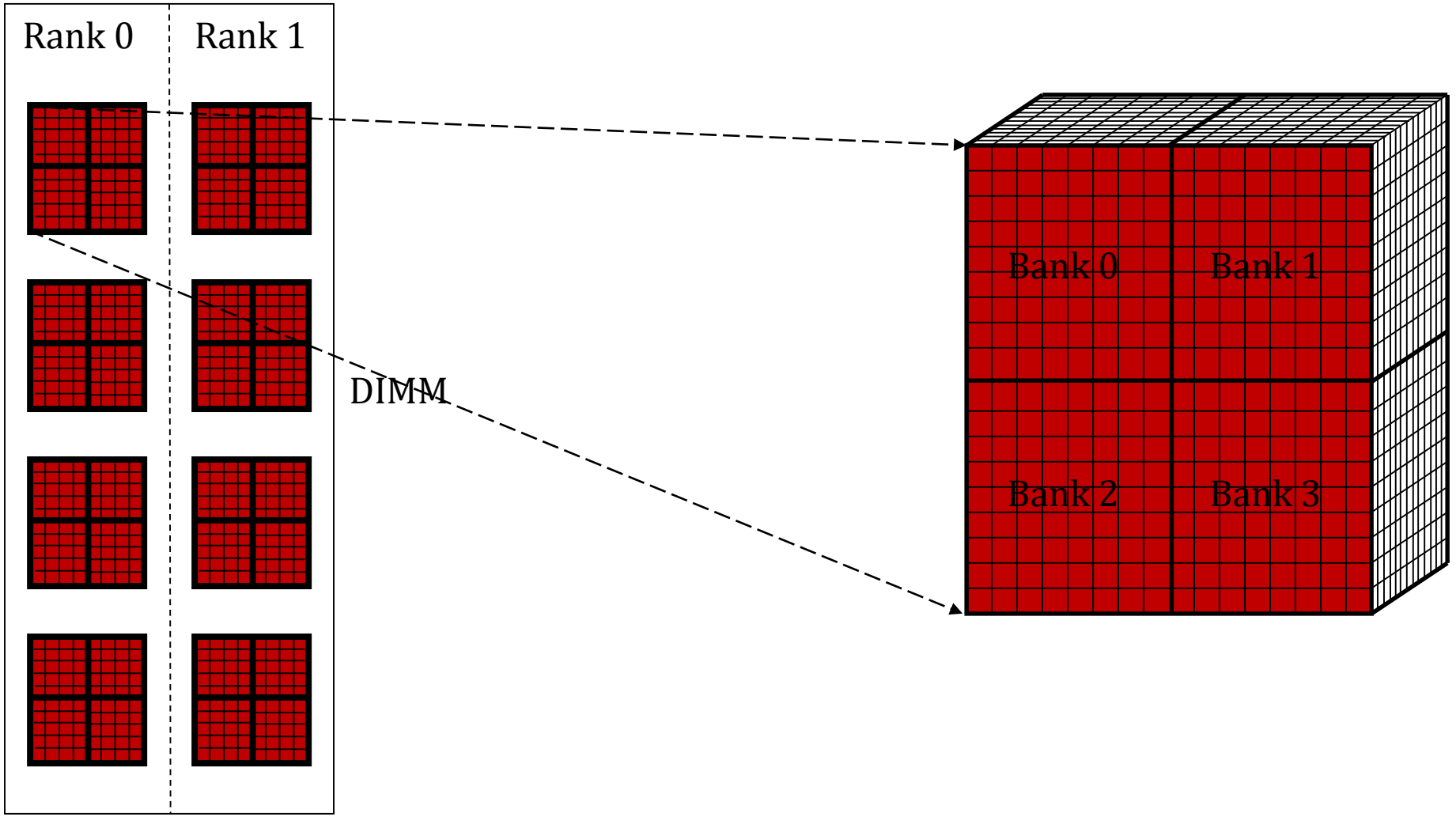
Row

Column

Rank 1 with 8 chips

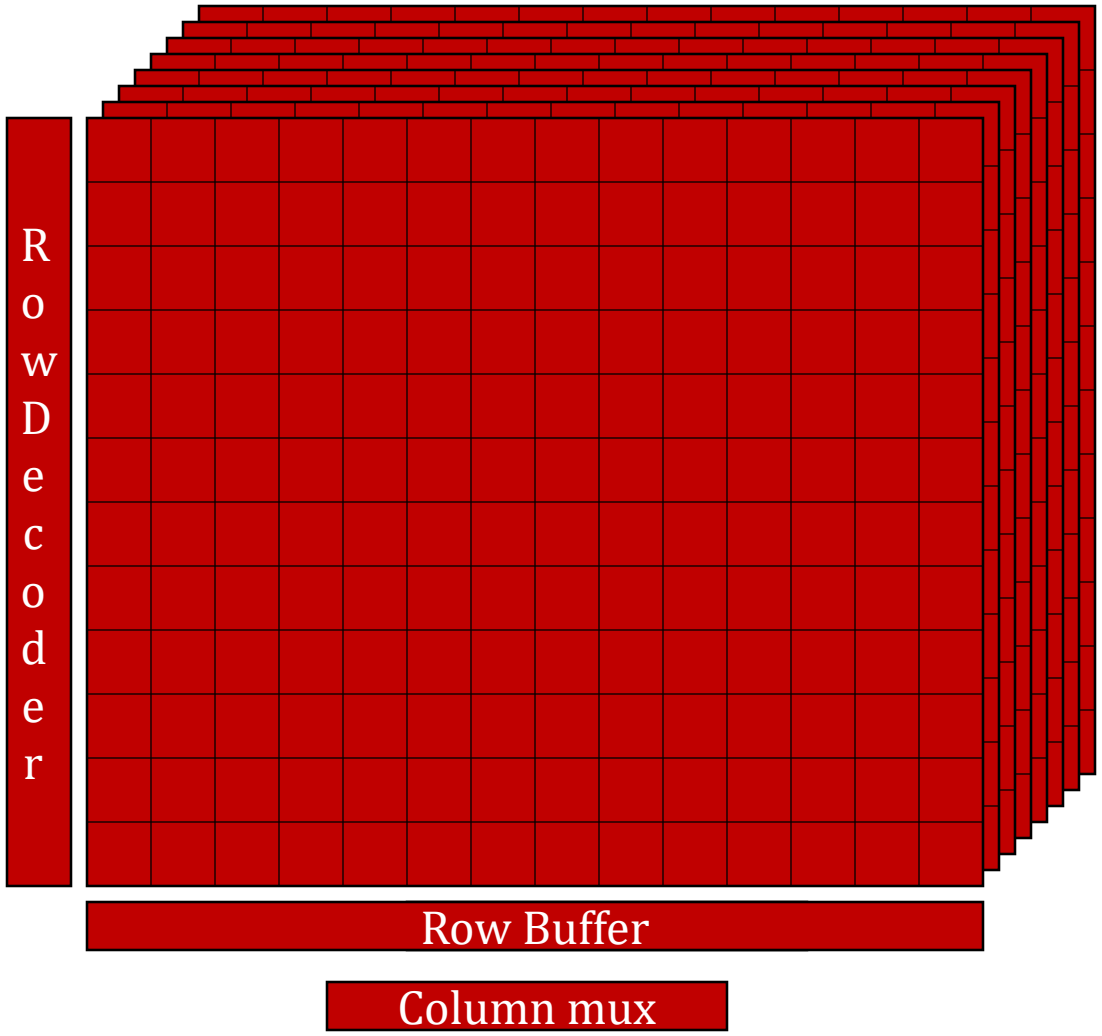
Rank 0 with 8 chips

Ranks, Banks, Rows, and Columns

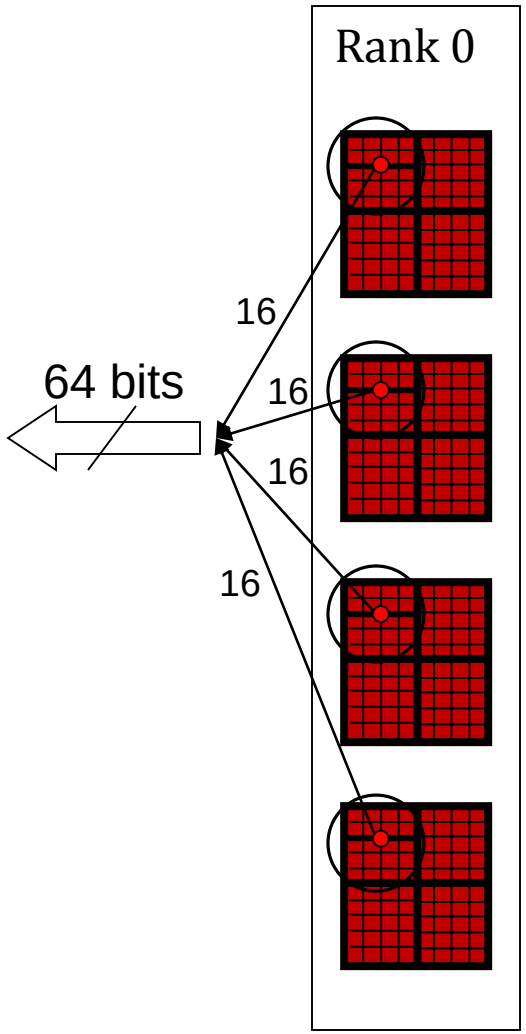


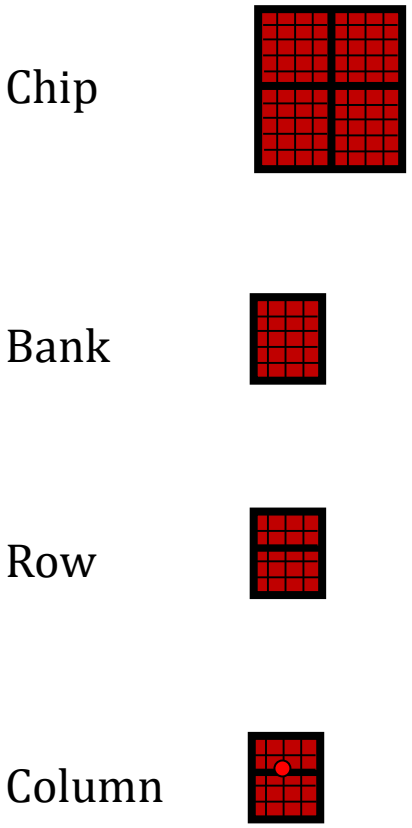
Bank: Collection of DRAM Arrays

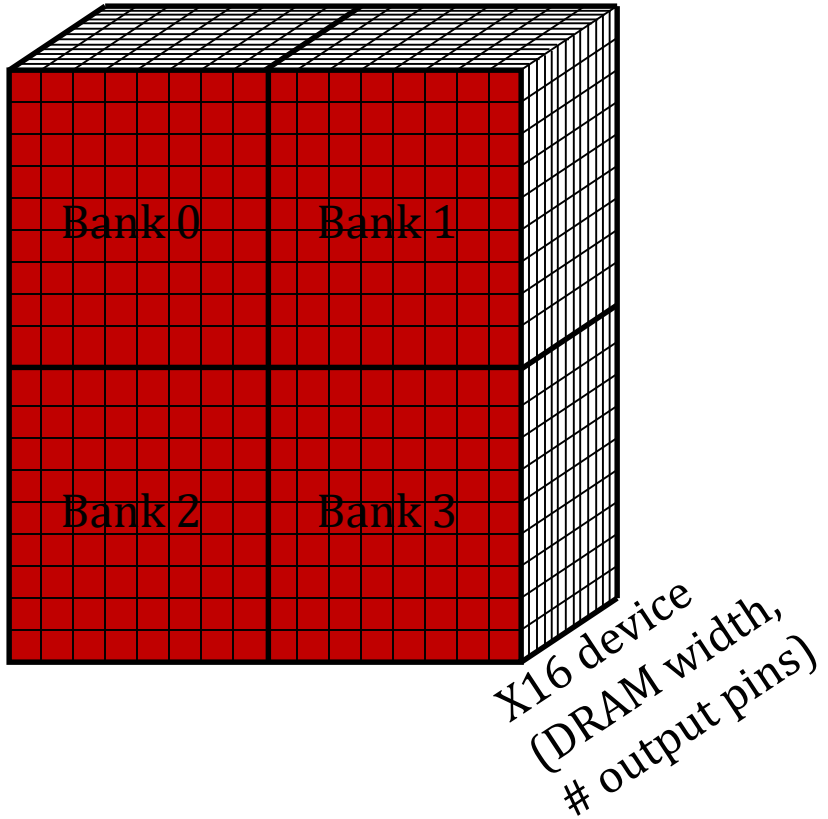
- DRAM Width
  - x4 device
  - x8 device
- Other possible widths
  - x16
  - x32
  - x48
  - x72



Ranks, Banks, Rows, and Columns







16-bit interface: 16 bits from each chip in one go

Let’s Dig Deep

Each rank has 64-bit wide data bus

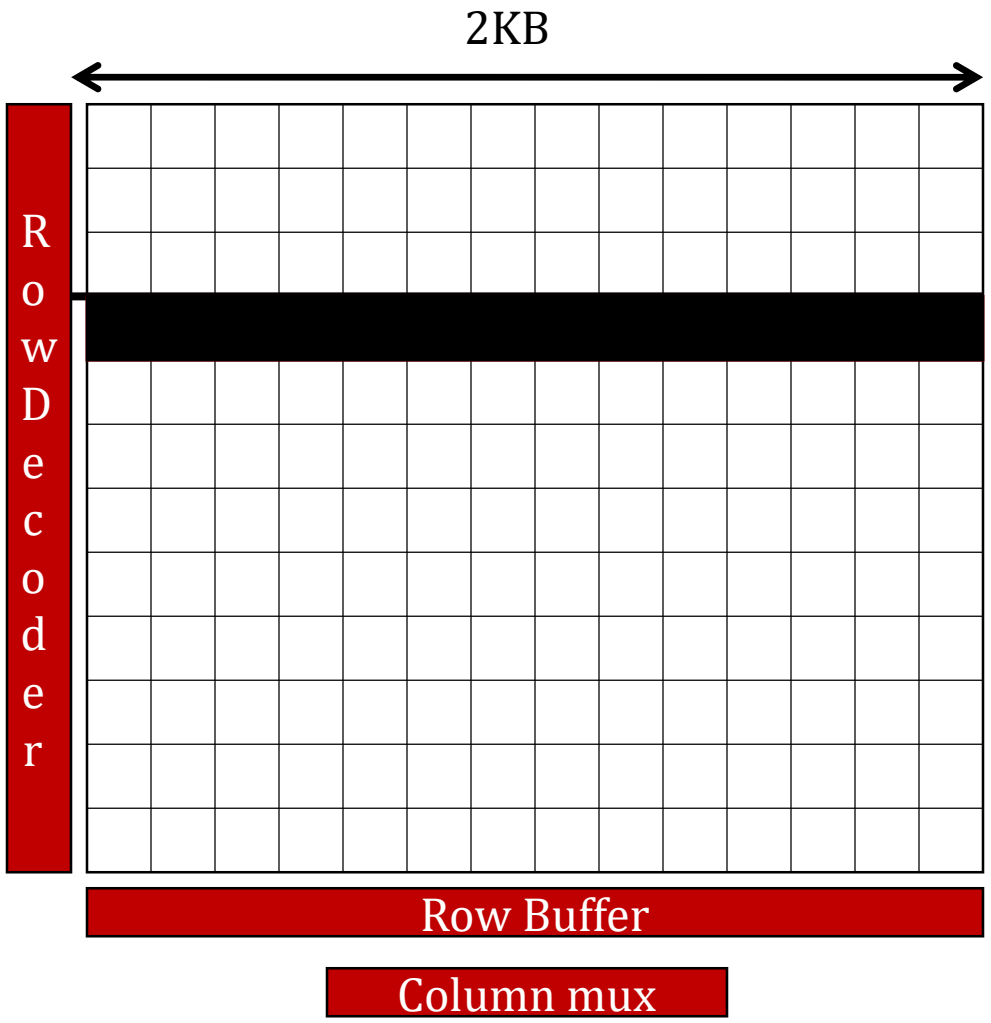
If a rank is of width x8 then # DRAM chips ??

What about x4, # DRAM chips ??

If a rank is of width x8 then # DRAM chips ?? 8

What about x4, # DRAM chips ?? 16

Row (page) and Row buffer (Sense Amplifier)



Logically

Each bank has a row buffer

Stores the last used row

An Example – 4GB DIMM

2Gb * 8 DRAM Chips (one side of the rank)

Total 16 chips + 2 chips for ECC (for both the ranks)

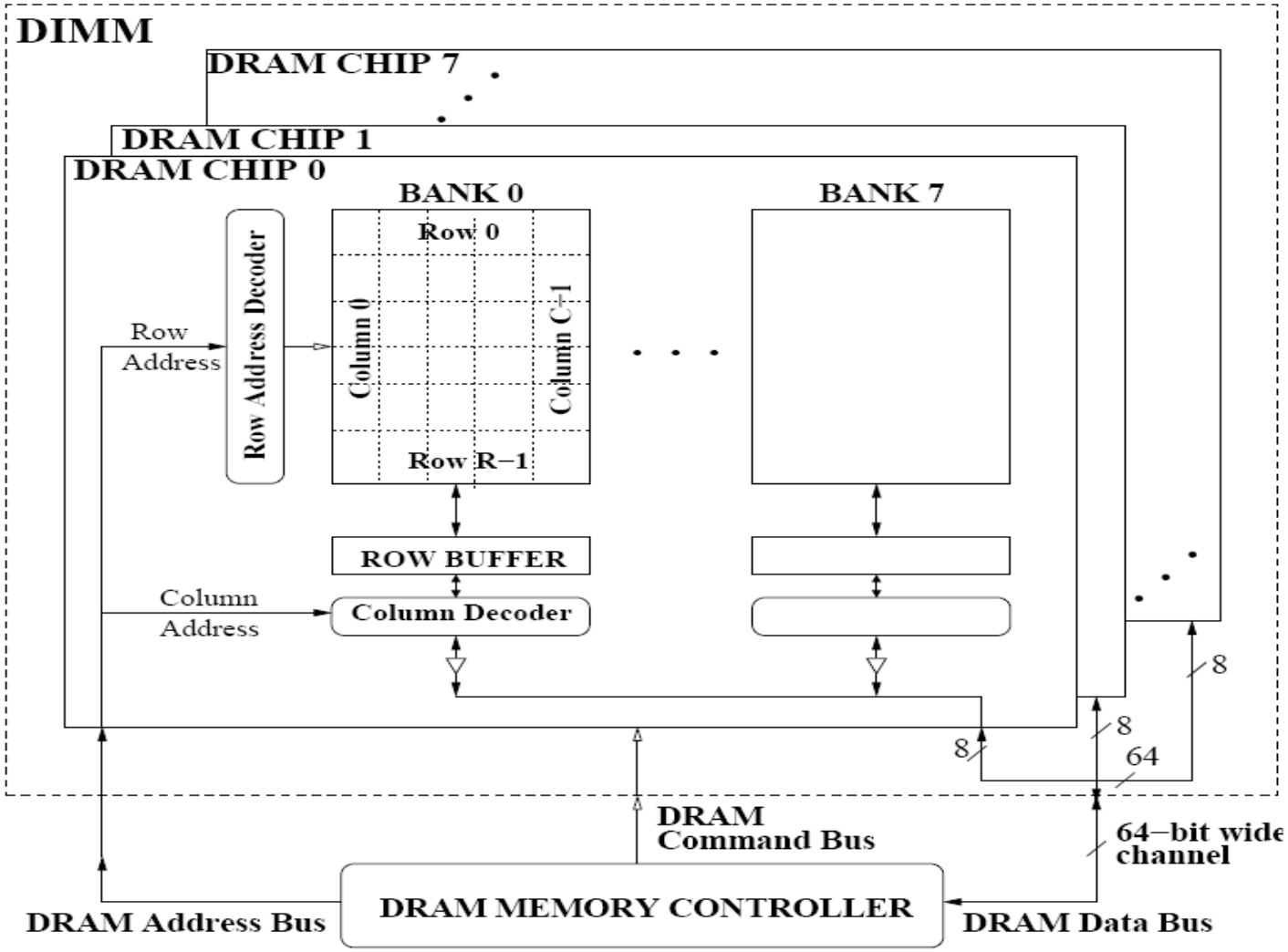
64 bit + 8 bit ECC interface (72 bit wide DIMM)

Transferring a 64B cache line will take 8 transfers of 8B each

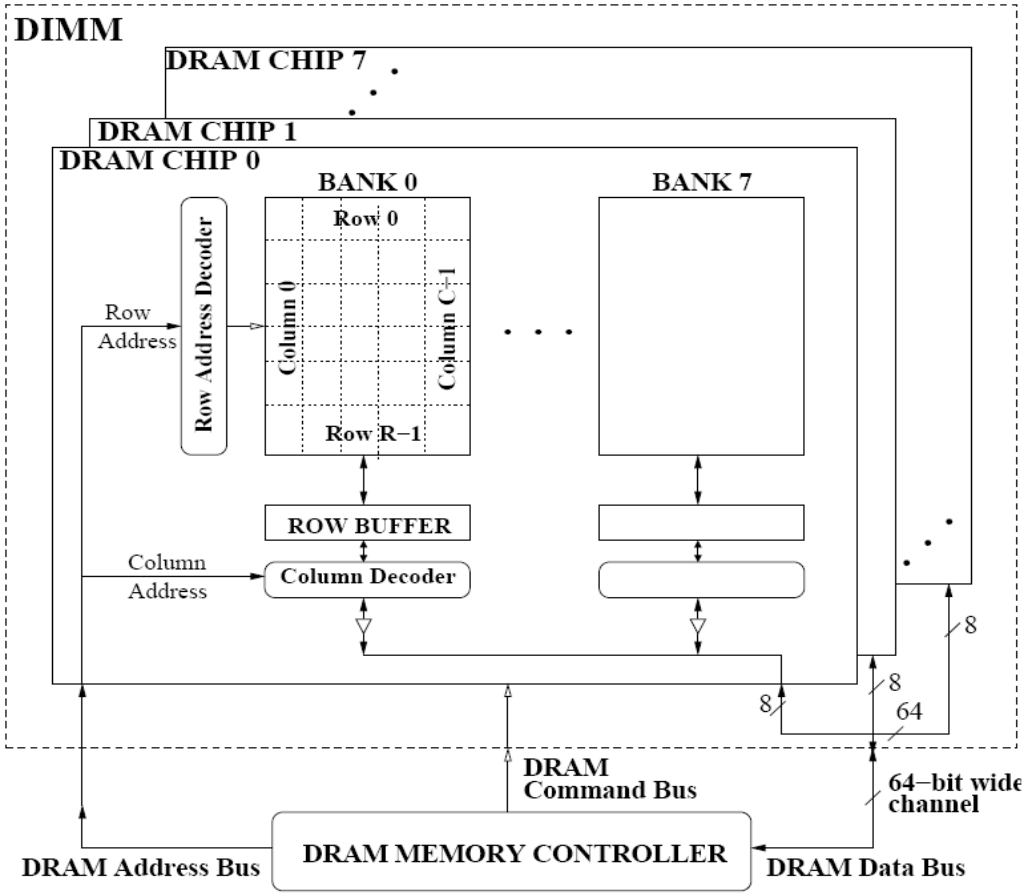
8B will come from 8 chips (8 bits from one chip)

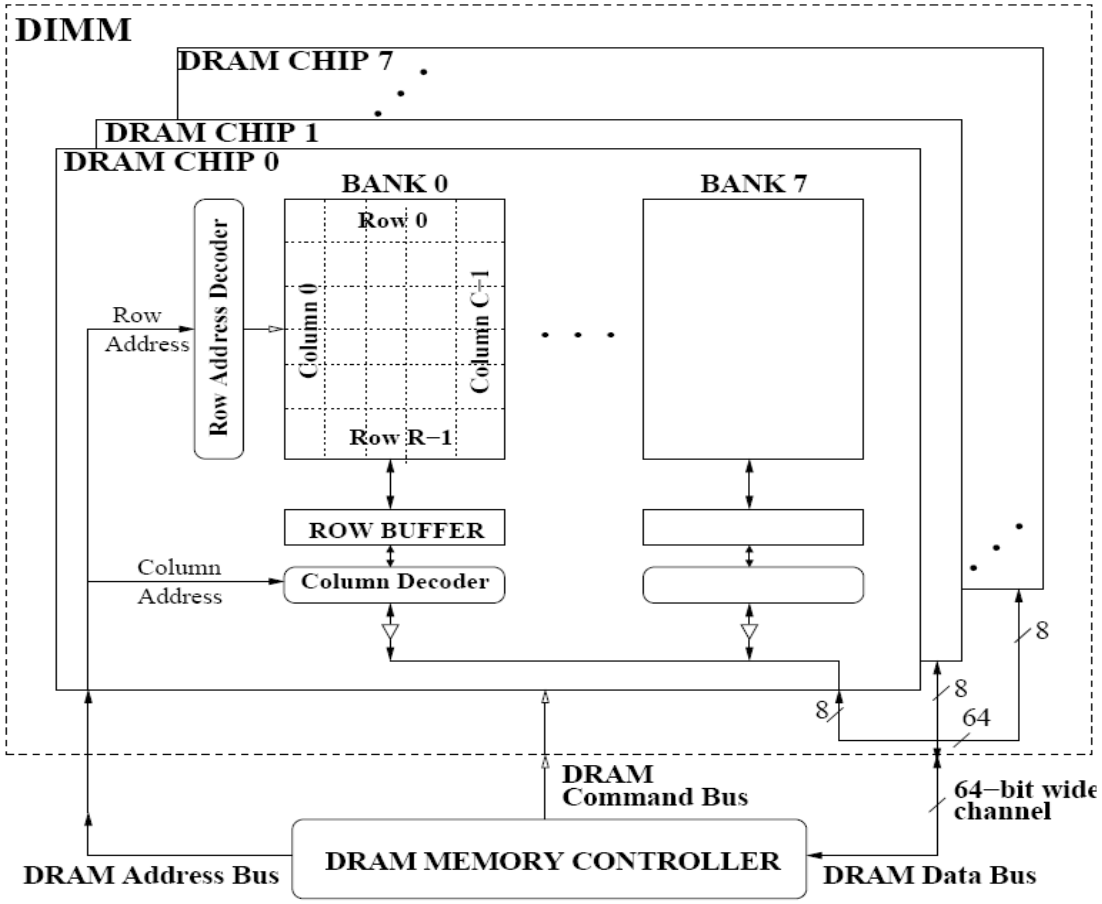
1 bit from each DRAM array assuming 8 DRAM arrays per bank

Another View

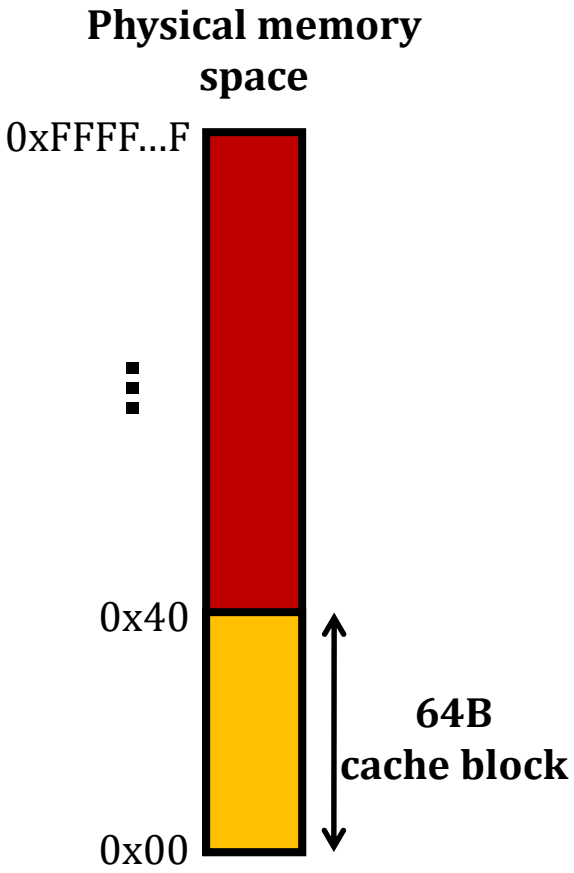


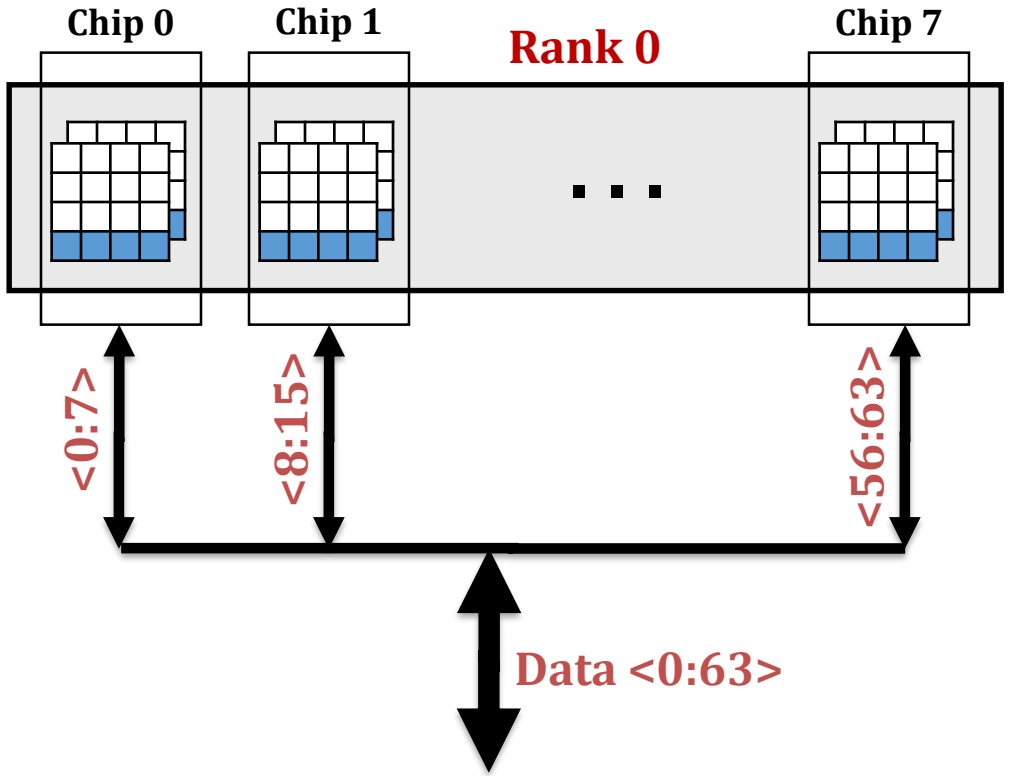
DRAM Channels



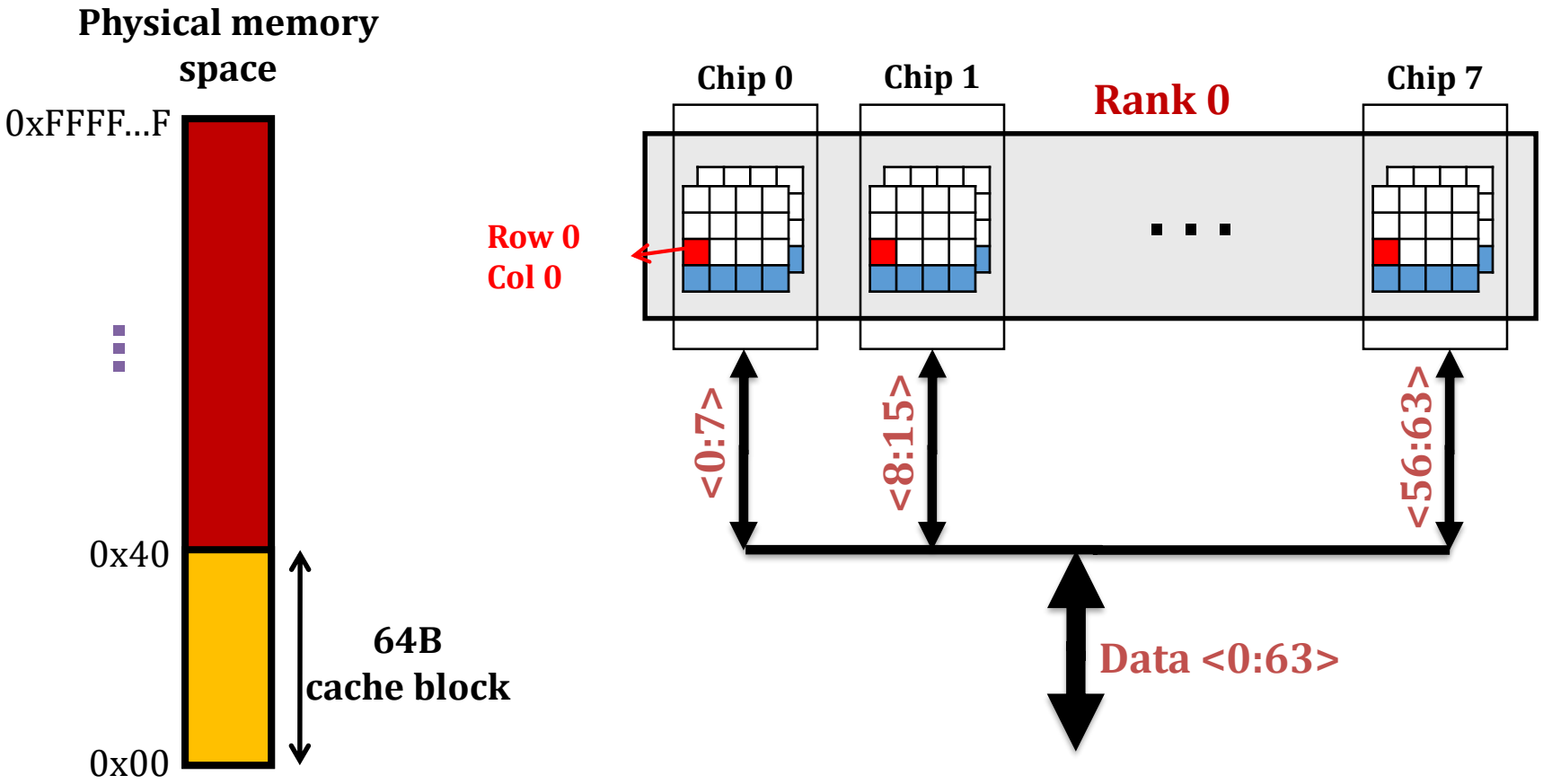


DRAM to LLC

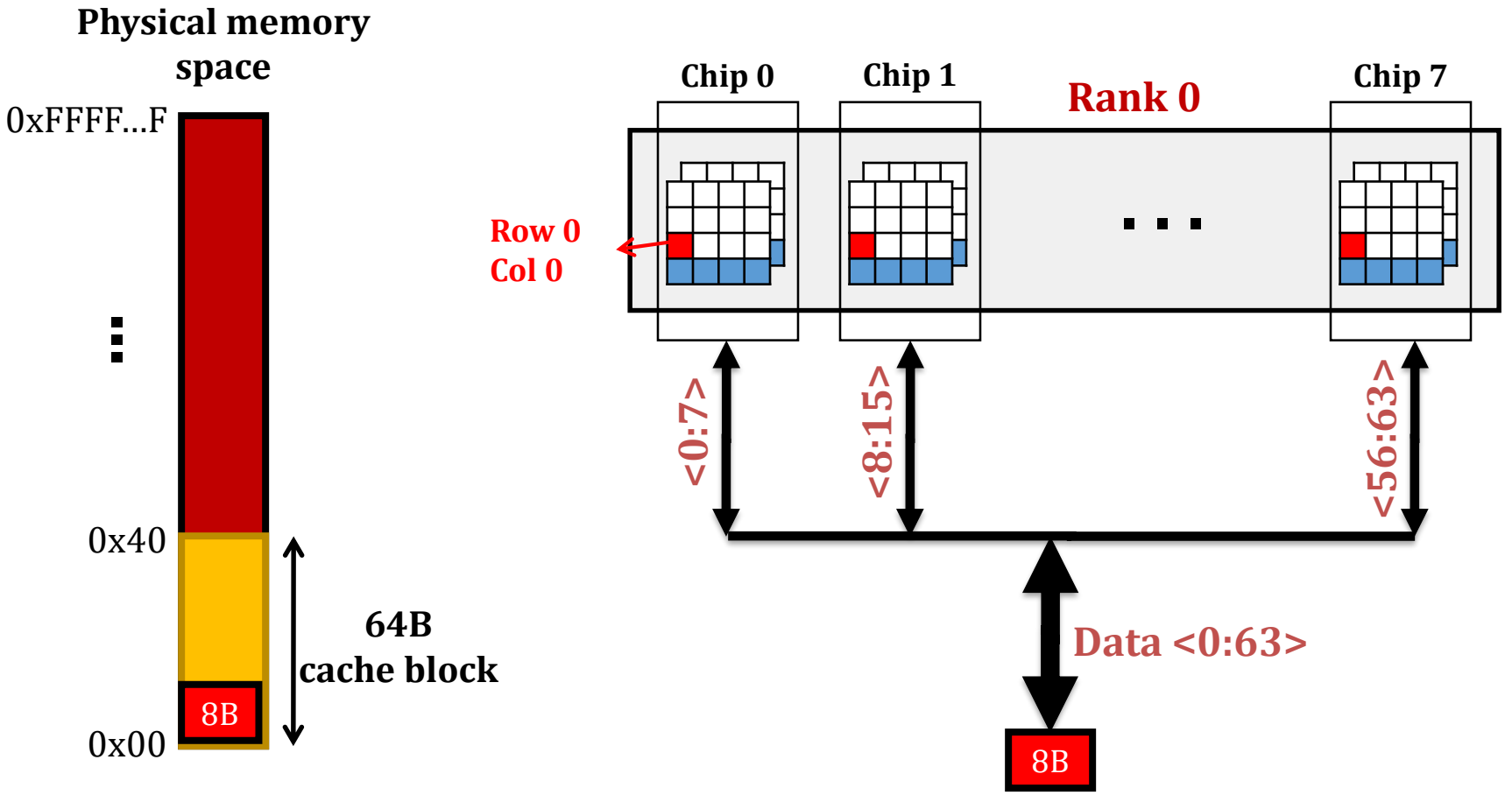

Physical memory space
0xFFFF...F
0x40
0x00
64B cache block


Chip 0
Chip 1
Rank 0
Chip 7
<0:7>
<8:15>
<56:63>
Data <0:63>

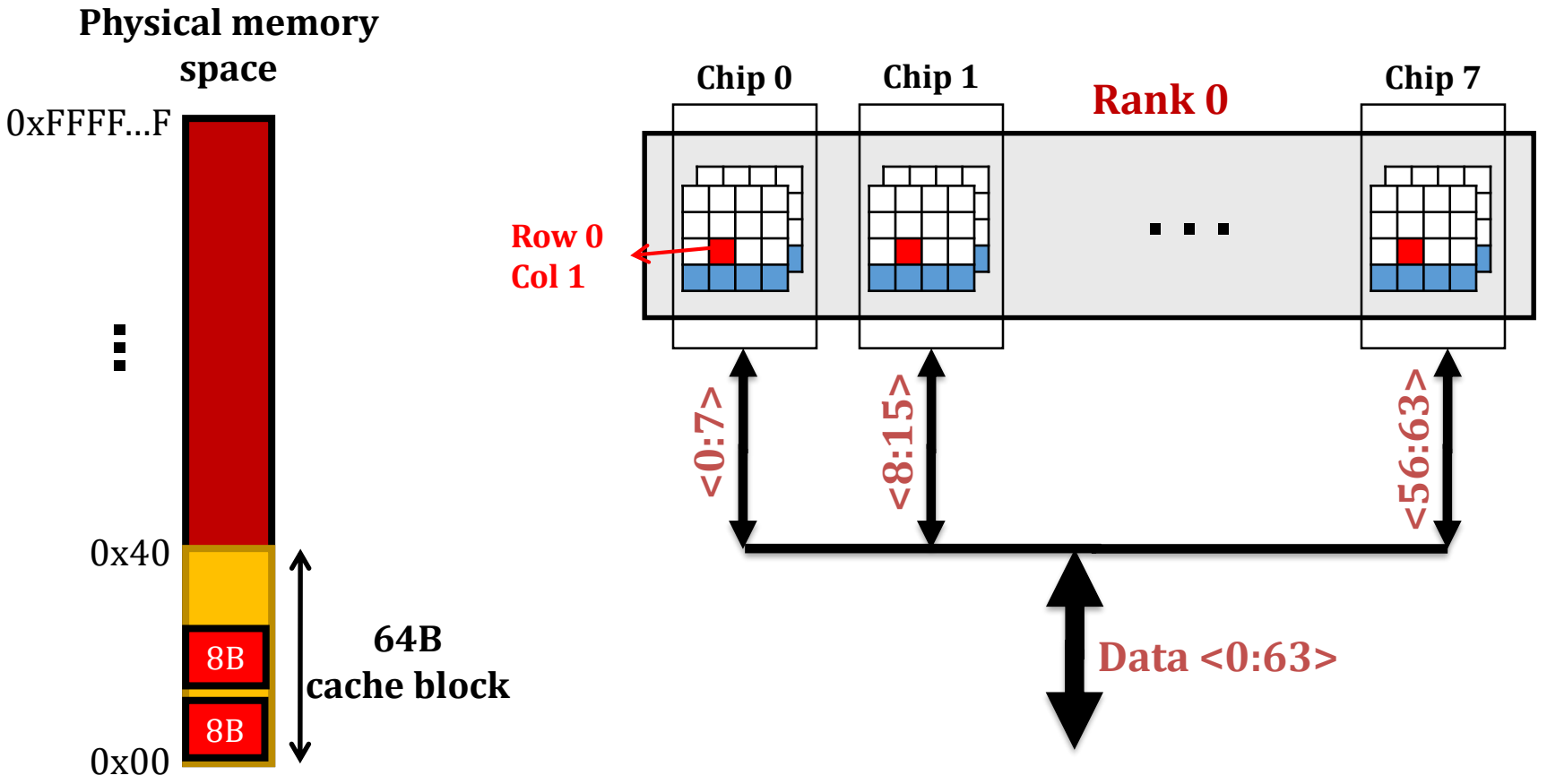
DRAM to LLC


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>

DRAM to LLC


Physical memory space
0xFFFF…F
0x40
0x00
8B
64B cache block
Chip 0
Chip 1
Rank 0
Chip 7
Row 0
Col 0
<0:7>
<8:15>
<56:63>
Data <0:63>
8B

DRAM to LLC

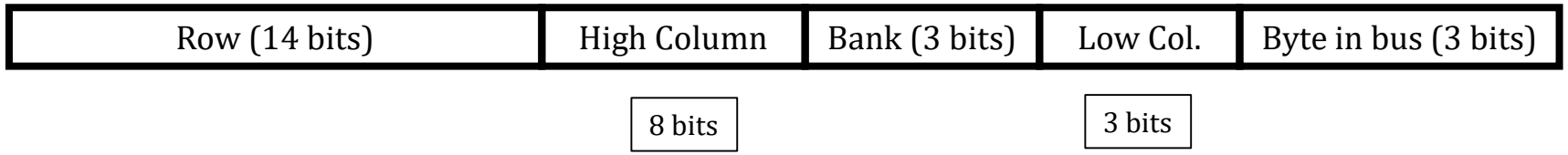


8 cycles (DRAM IO): 1 cycle transfers 8 bytes from a column

DRAM Address Mapping (1 Channel)

2GB DRAM, 8 Banks, 16K rows, 2K Columns per bank

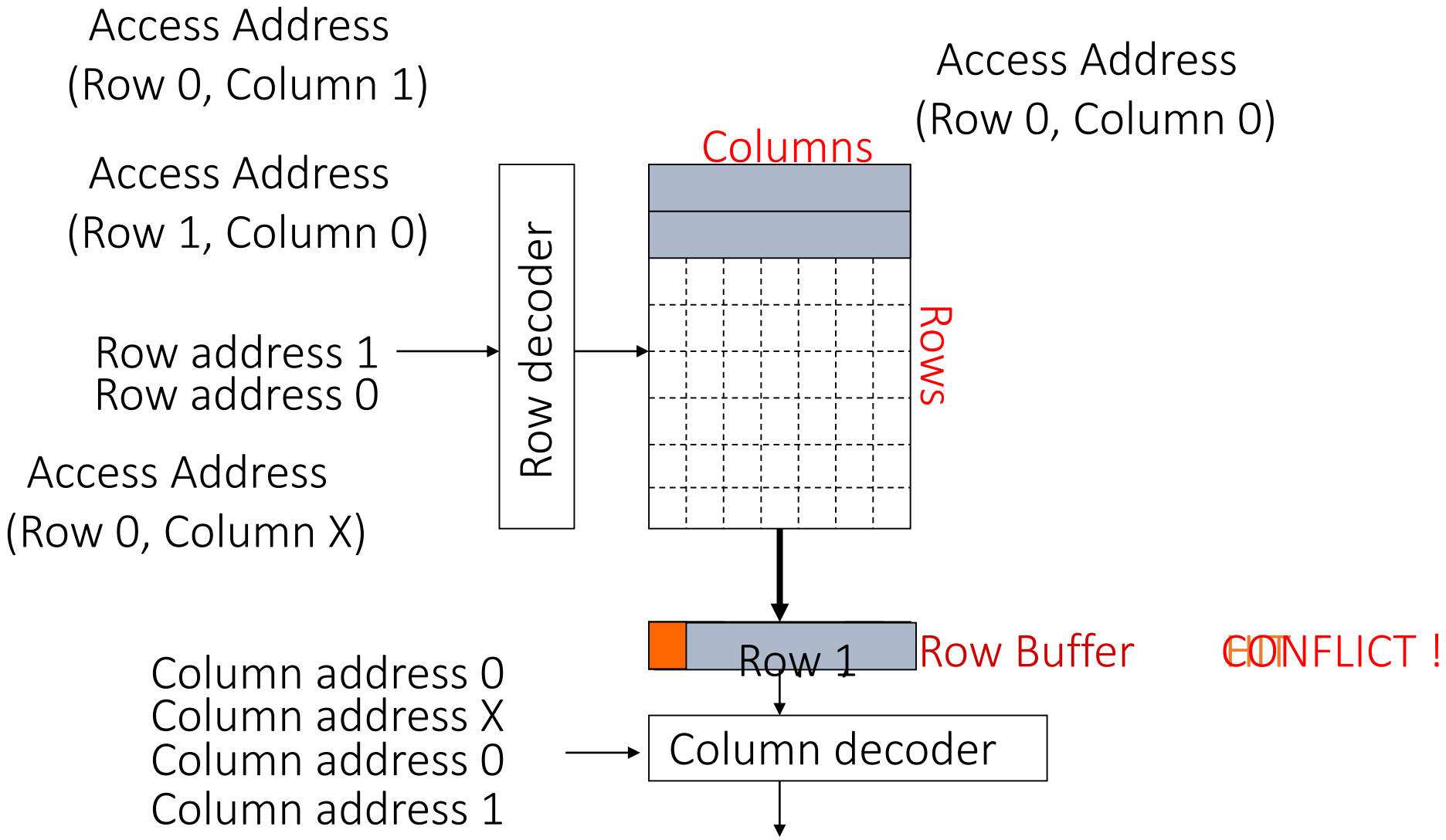
Cache Interleaving: Consecutive cache blocks in consecutive banks



Row Interleaving: Consecutive rows in consecutive banks

What about Multiple Channels?

Accessing a Row



Row Buffer Operations

Access to a closed row (Sequence of Commands):

ACT: Activate the row and place it into row buffer
READ/WRITE: Reads or writes a column
PRECHARGE: Close the row

Access to an open row (Sequence of Commands):

READ/WRITE: Reads or writes a column
PRECHARGE: Close the row

OPEN/CLOSED Row Policies

OPEN PAGE

After an access:
Keep the page in the row-buffer
Consecutive accesses to the same page : Row-buffer Hit
On an access to different page: Close the row and open the new one

CLOSED PAGE

After an access:
Close the page
Consecutive accesses to different page : Low latency
On an access to different page: No need to close the row

OPEN/CLOSED Row Policies

OPEN :
Exploits spatial and temporal locality
Access patterns ?
Latency is limited tCAS

CLOSED :
Exploits random access pattern
Access patterns ?

Let’s Revisit the Latency

PAGE EMPTY

ACT + CAS

PAGE HIT

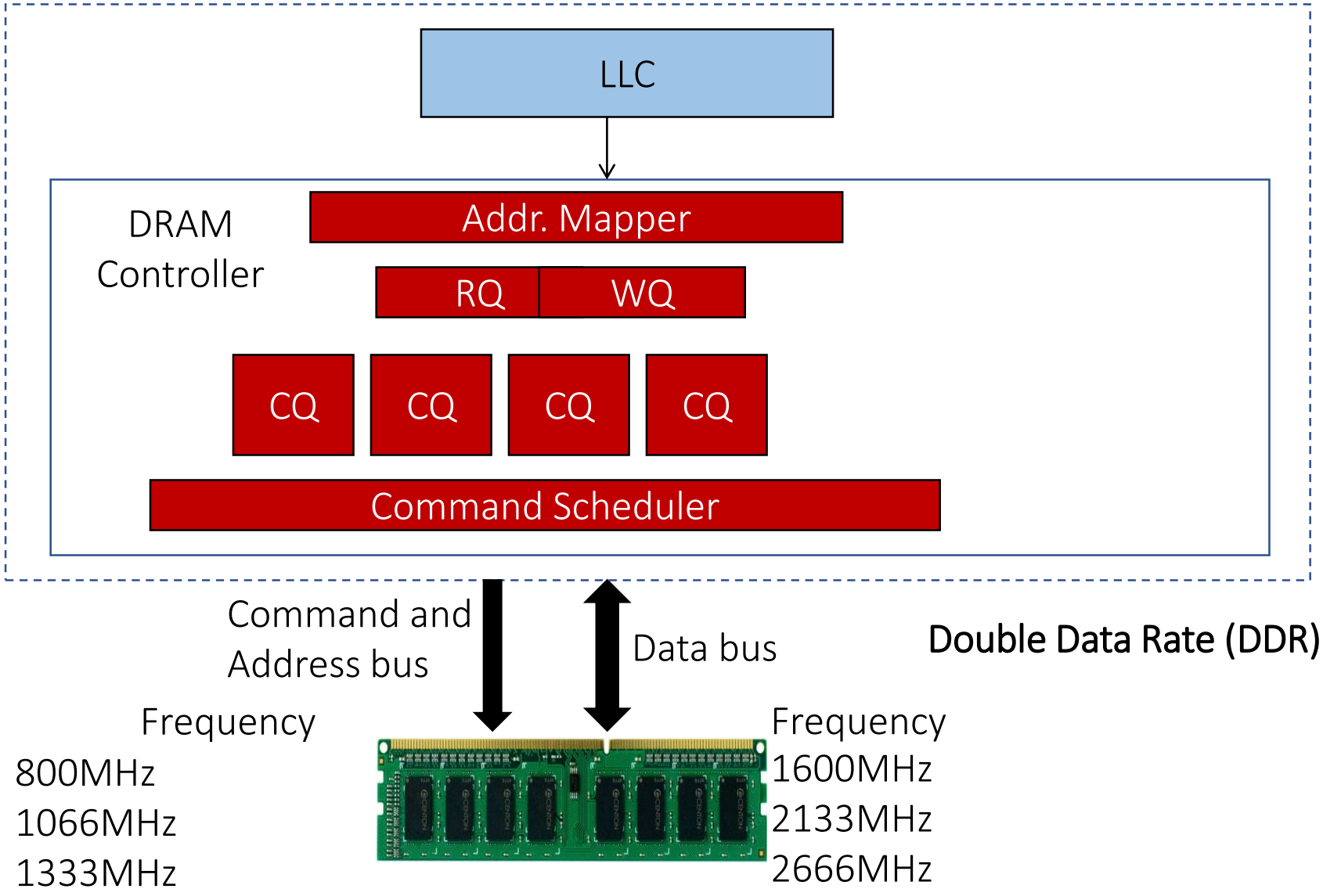
CAS

PAGE MISS

PRE + ACT + CAS

Given a DRAM request,
what is the best/worst-case delay ?

DRAM Controller: An Overview



Reads vs Writes

Reads are critical to performance

Write Queue stores writes and the writes are serviced after # writes reach a threshold

Why?

The direction of the data bus changes from reads to writes. So ??

DRAM controller creates DRAM commands from based on the requests at read Q and write Q

DRAM Scheduling

Based on
Row-buffer locality,
Source of the request,
Loads/Stores
Load criticality

Satisfy all the timing constraints. Around 60

FCFS?

FR-FCFS
[ISCA 00]

Prefers requests with Row hits (column-first) FR: First Ready