ΗΥ 232
Οργάνωση και
Σχεδίαση Υπολογιστών

Διάλεξη 17

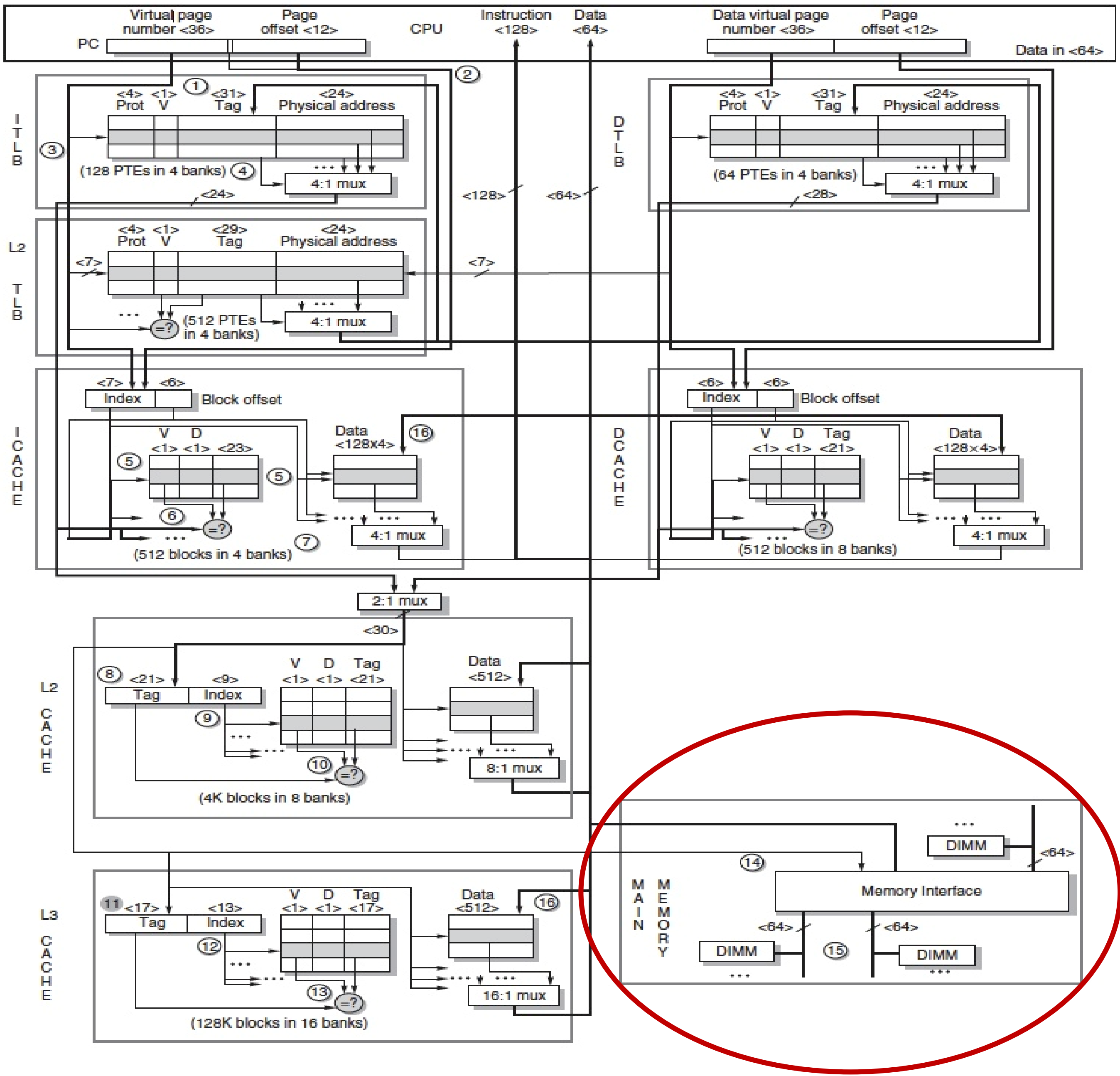
Κύρια Μνήμη (Main Memory)
Ελεγκτής Μνήμης (Memory Controller)

Νίκος Μπέλλας
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ

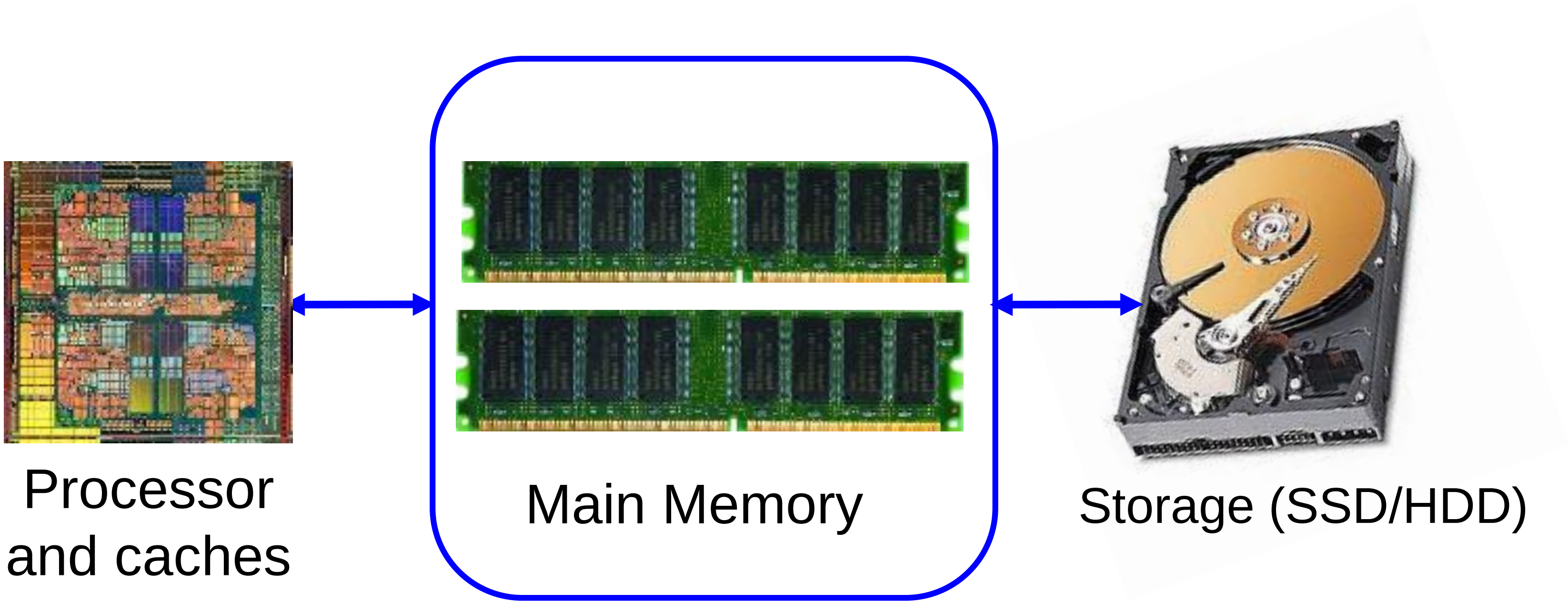
Main Memory Basics

Motivation

- DRAM and the memory subsystem significantly impacts the performance and cost of a system
- Need to understand DRAM technologies
  - to architect an appropriate memory subsystem for an application
  - to utilize chosen DRAM efficiently
  - to design a memory controller



The Main Memory System



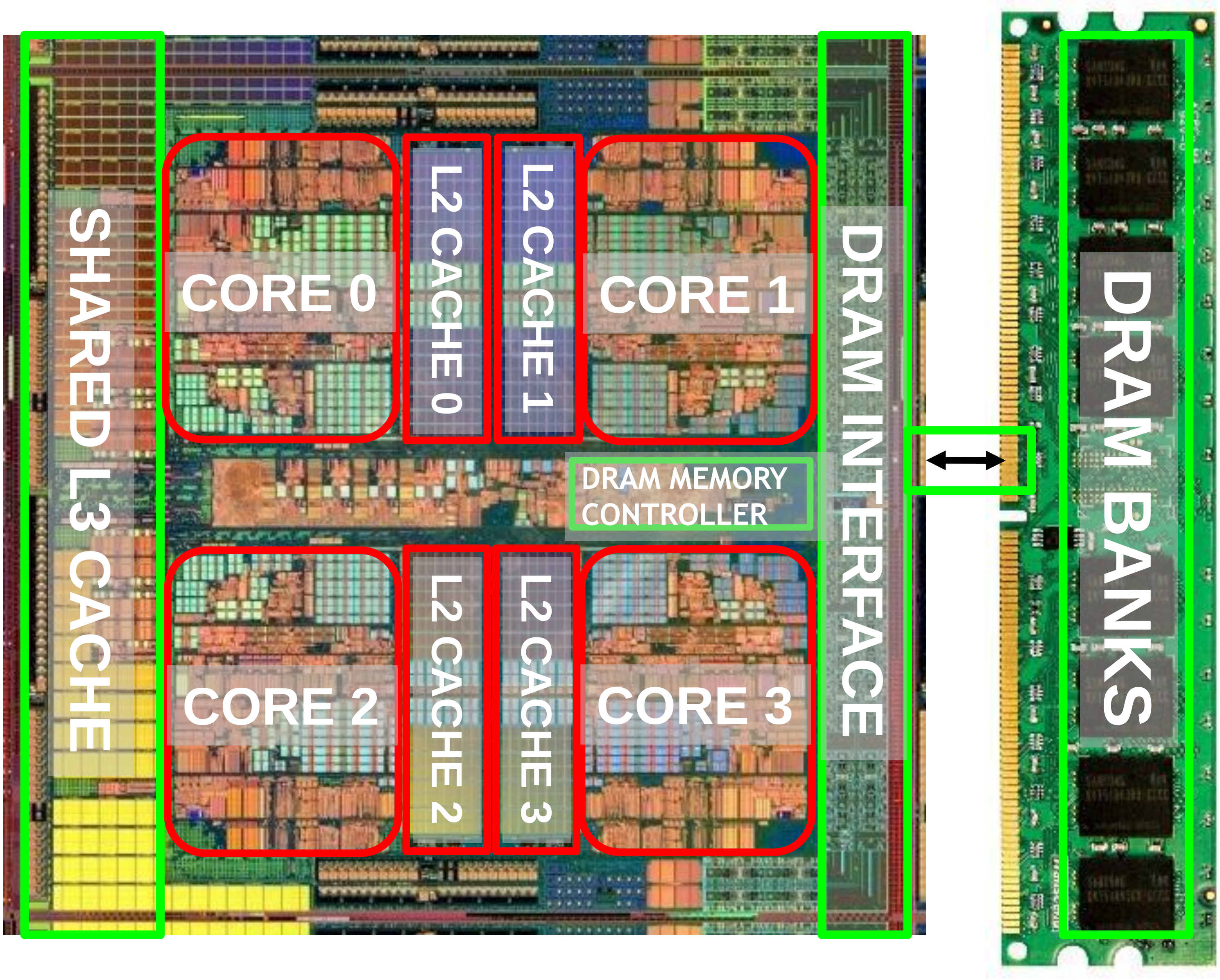
Processor and caches

Main Memory

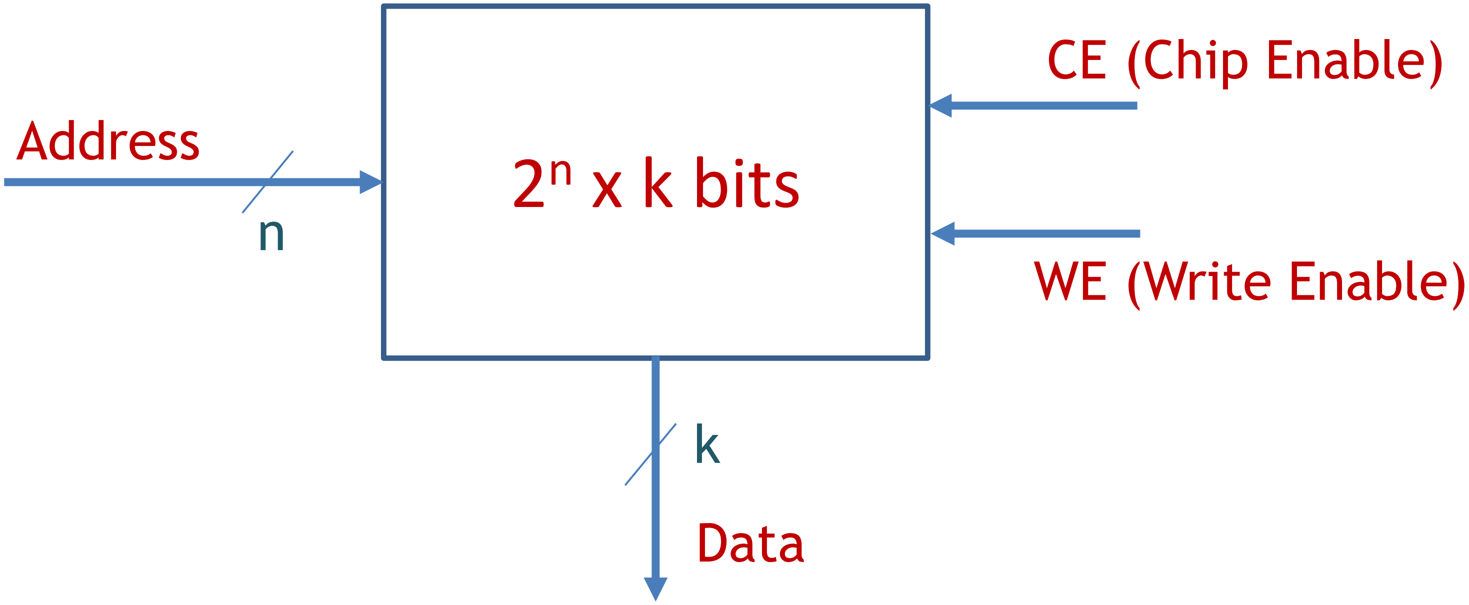
Storage (SSD/HDD)

- 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

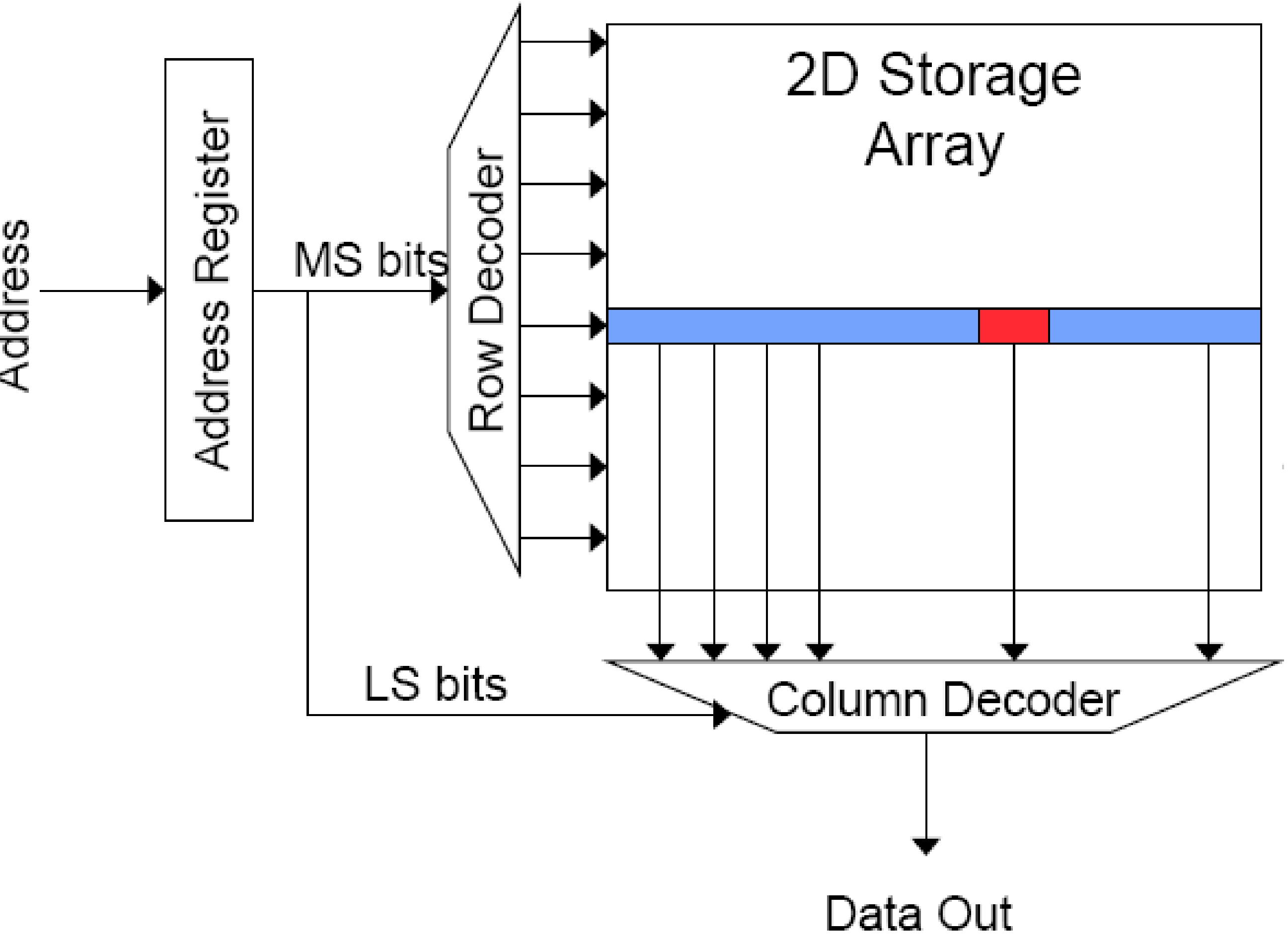
Main Memory in the System



The Main Memory Chip/System Abstraction



Basic Functionality and Organization



- **Read access sequence:**

1. Decode row address & drive word-lines

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

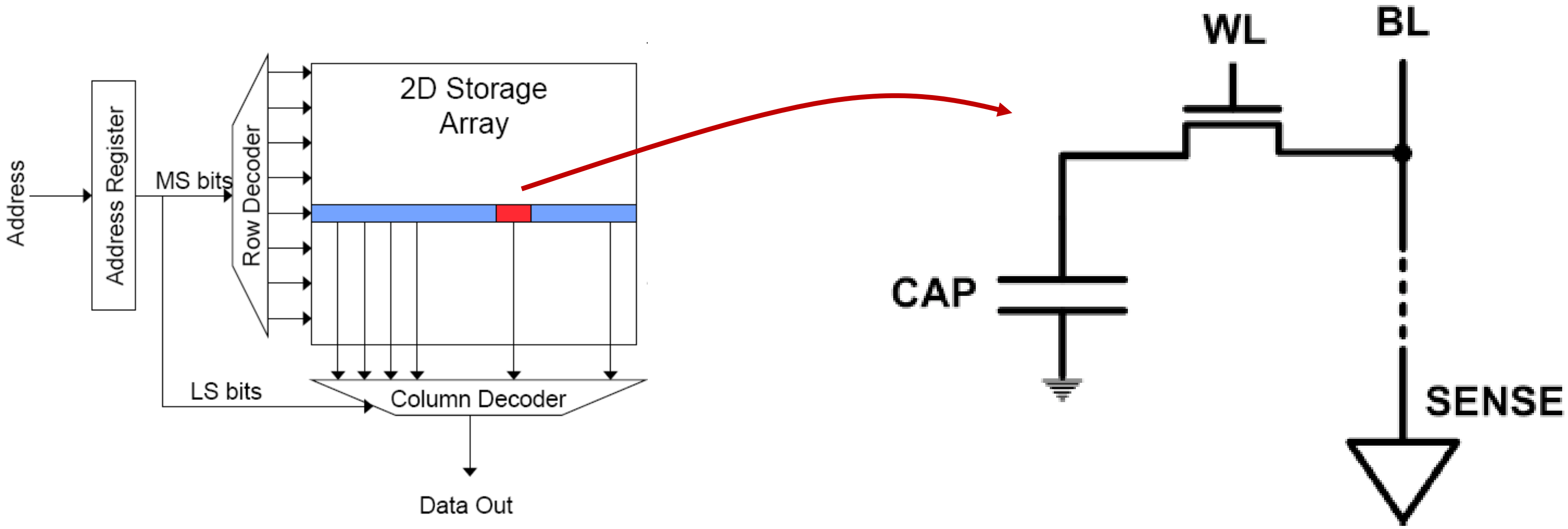
3. Amplify row data

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

5. Precharge bit-lines
   - For next access

The DRAM Storage Cell

- 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
  - There are $2^n x k$ of those capacitors
  - Precharging puts Bit Lines (BL) to high voltage



Main Memory Background

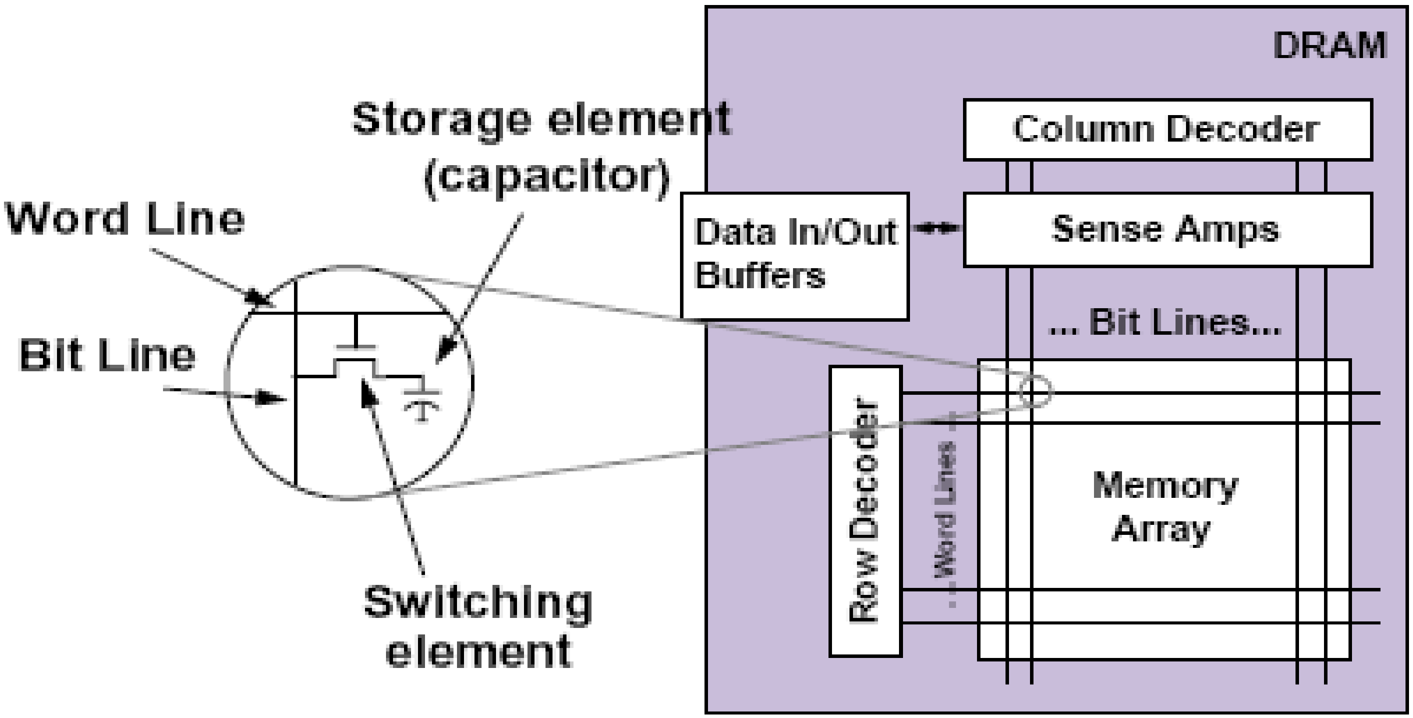
- Performance of Main Memory:
  - Latency: Cache Miss Penalty
    - *Access Time*: time between request and word arrives
    - *Cycle Time*: time between requests (CT > AT)
  - Bandwidth:
  - Main Memory is *DRAM*: Dynamic Random Access Memory
  - Dynamic since needs to be refreshed periodically (64 ms, 1% time)
  - Addresses divided into 2 halves (Memory as a 2D matrix):
    - *RAS* or *Row Access Strobe*
    - *CAS* or *Column Access Strobe*
- Cache uses *SRAM*: Static Random Access Memory
  - No refresh (6 transistors/bit vs. 1 transistor
    Size: DRAM/SRAM *4-8*,
    Cost/Cycle time: SRAM/DRAM *8-16*

DRAMA Internal Organization

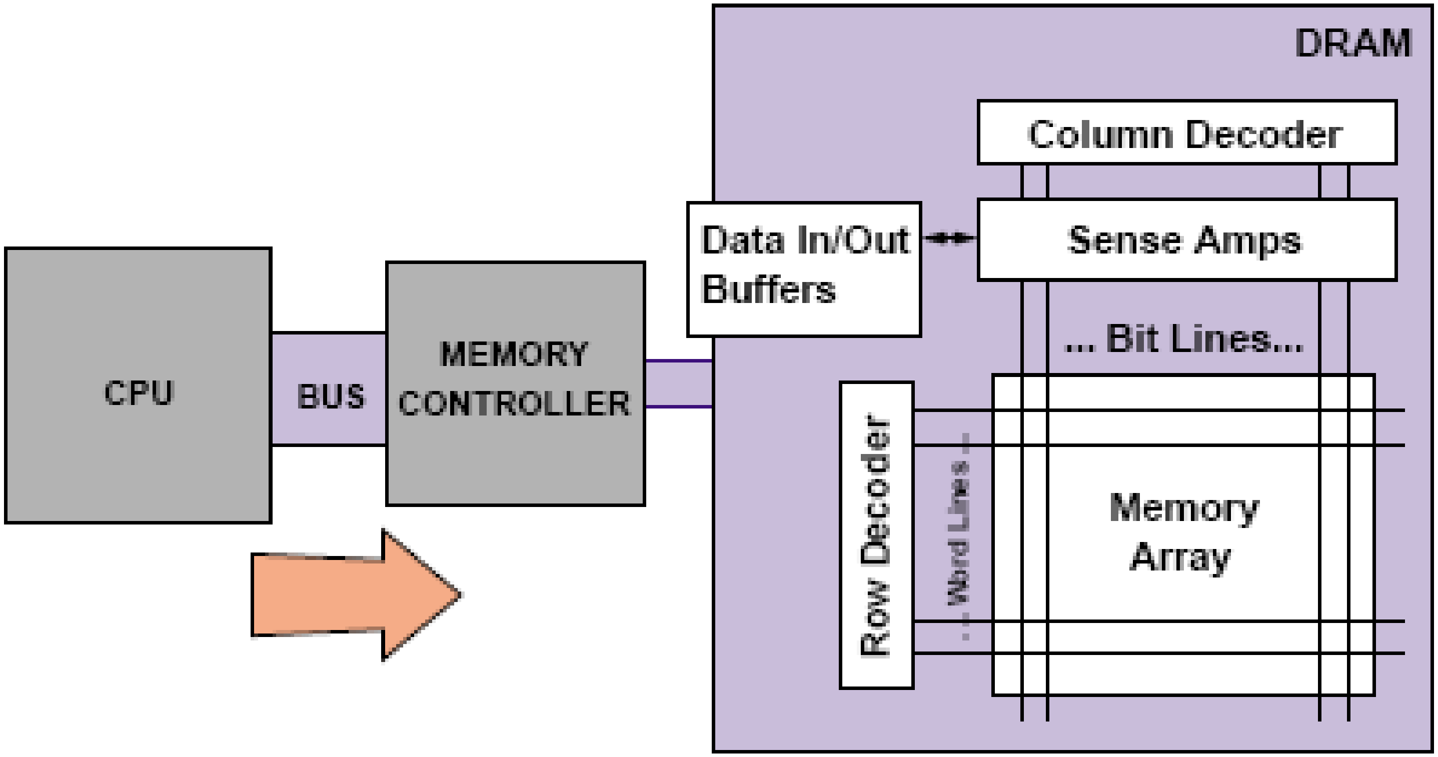
DRAM Types

DRAM internal organization

DRAM ORGANIZATION



DRAM access

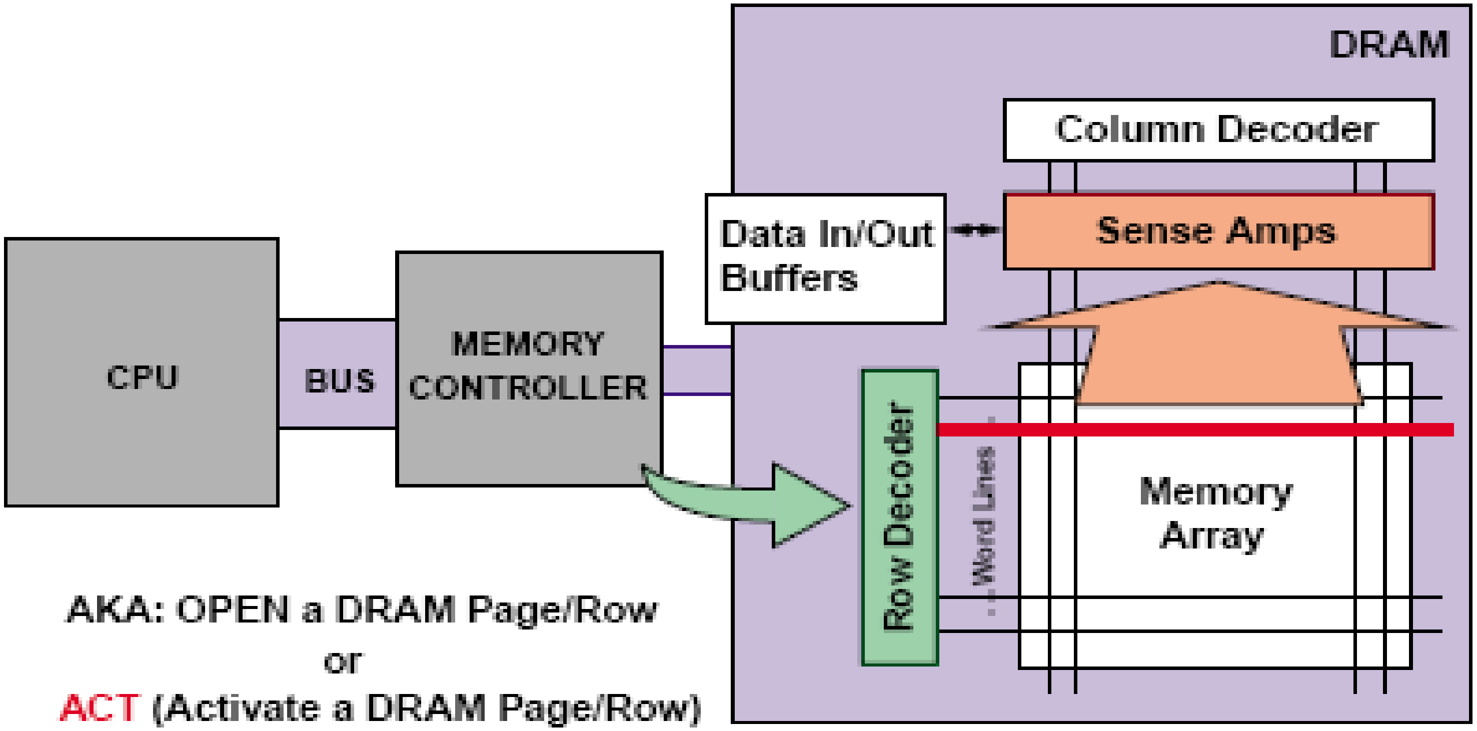


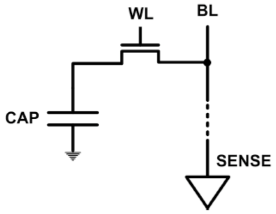
A cache miss triggers a cache line refill from the main memory.
The Memory Controller (MC) receives the request (along with potentially more requests from the same or other masters)
The memory access request consists of:

1. the physical address
2. the data in case of a memory write

DRAM basics

Precharge (PRE) and Row Access (ACT)





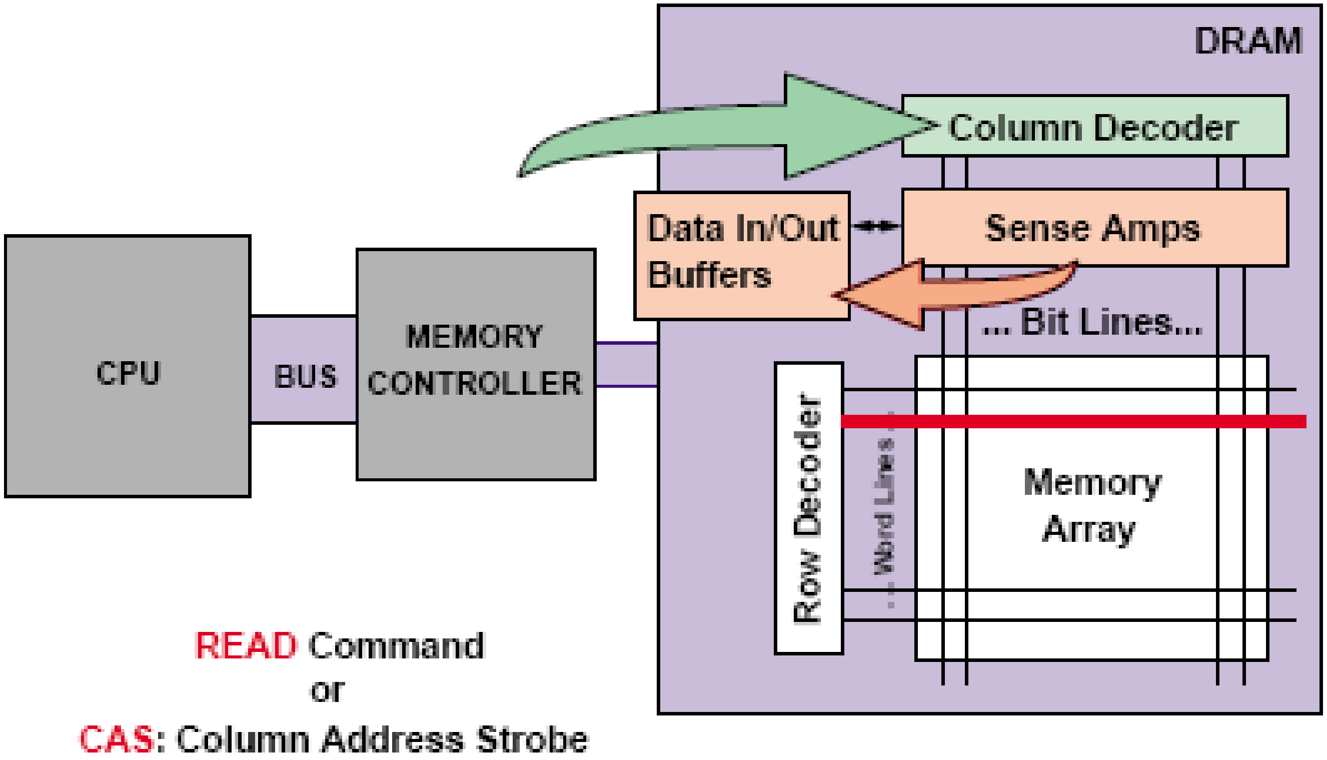
AKA: OPEN a DRAM Page/Row
or
ACT (Activate a DRAM Page/Row)
or
RAS (Row Address Strobe)

The MC breaks the access into two parts:

Row Access:

1. The MC precharges the DRAM array (opens a page). Any previously selected row is flushed from the sense amps
2. It creates the RAS signal to latch the Row Address to an internal latch.
3. The row decoder selects one row of bits that charges the sense amps (opens a row)

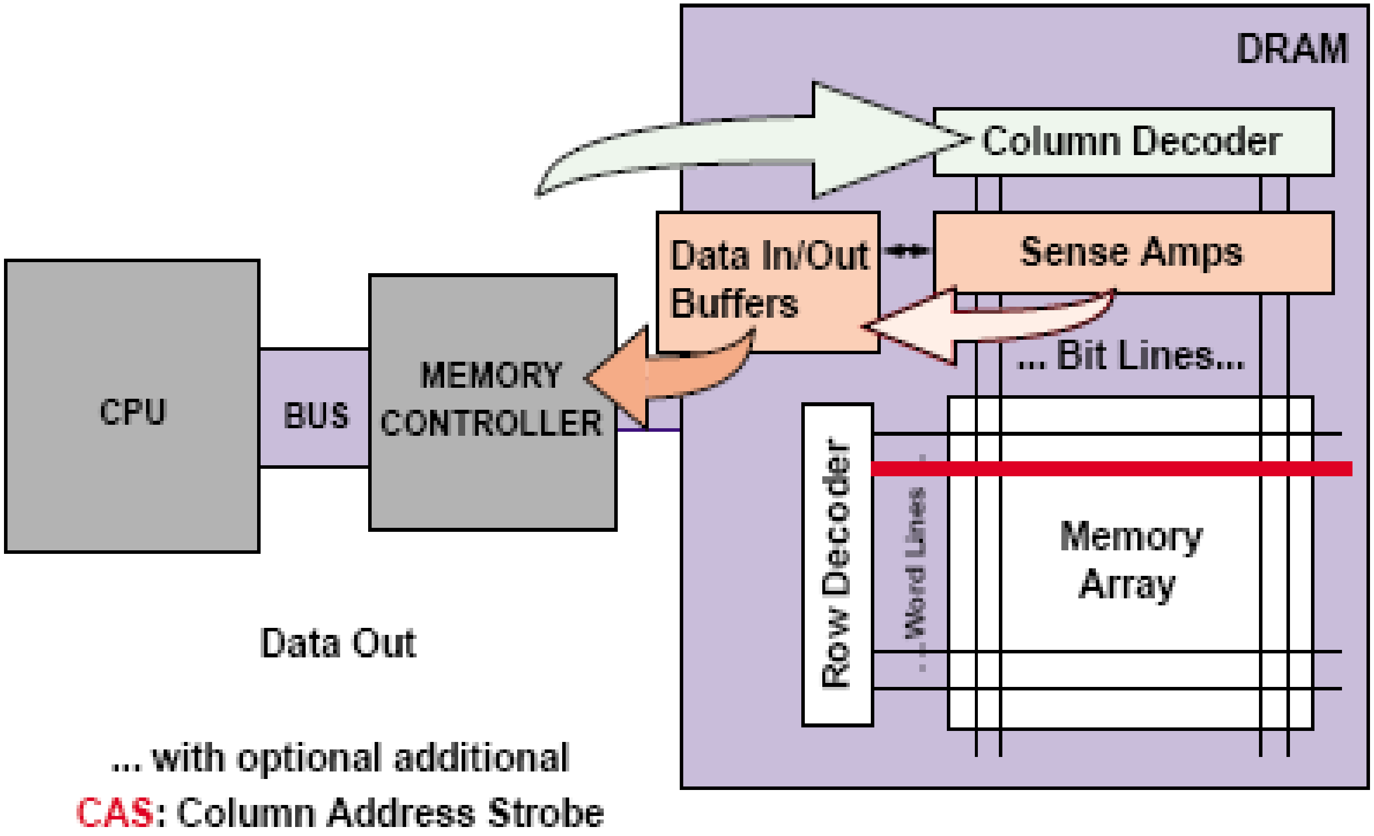
Sense Amps and Column Decoding



Column Access:

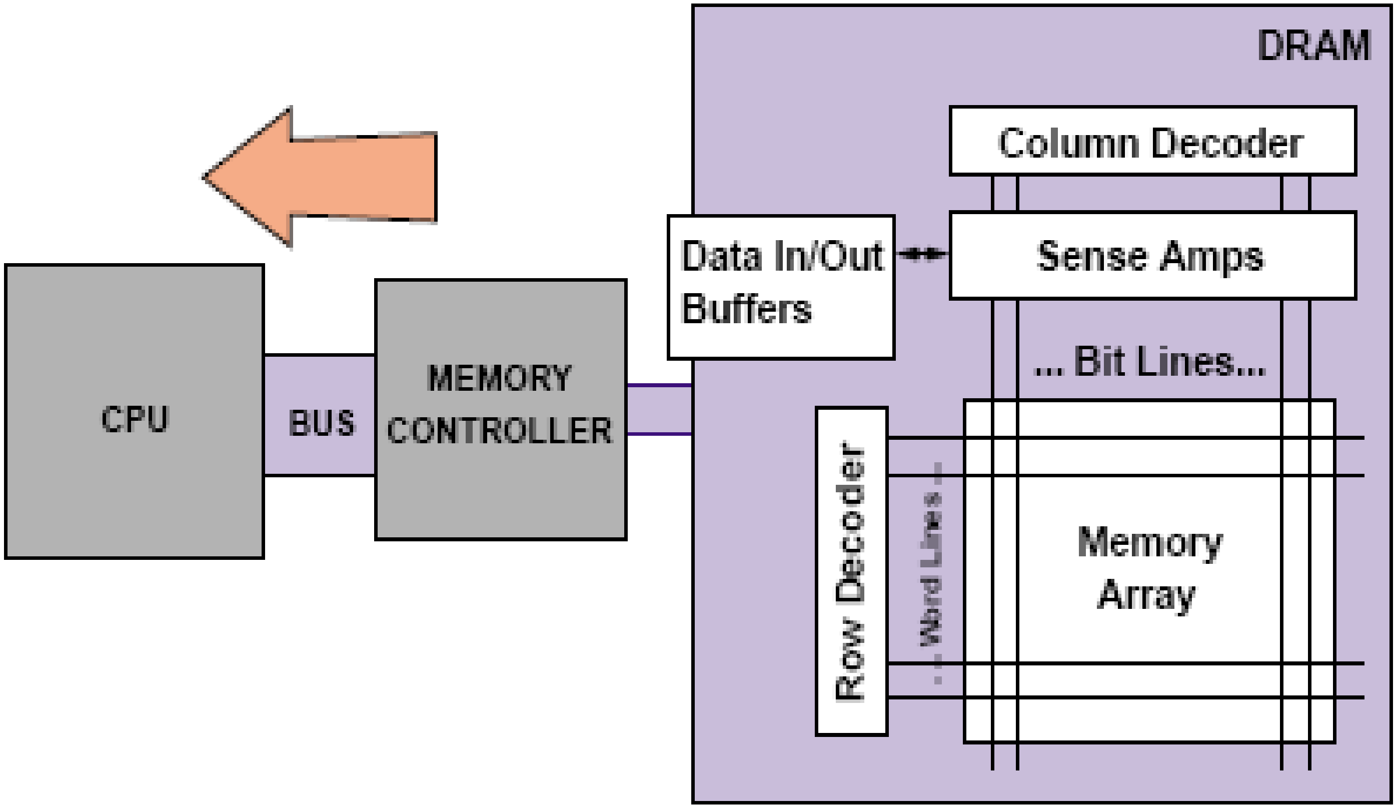
4. It creates the CAS signal to latch the Column Address to an internal latch. The CAS signal can also be used to latch the data in case of writes
5. The column decoder selects the bit to be read out. The CAS signal acts as Output Enable (OE) to drive data out to the output buffers

Read Out (READ)



A new column access in the SAME row reduces access time and increases bandwidth

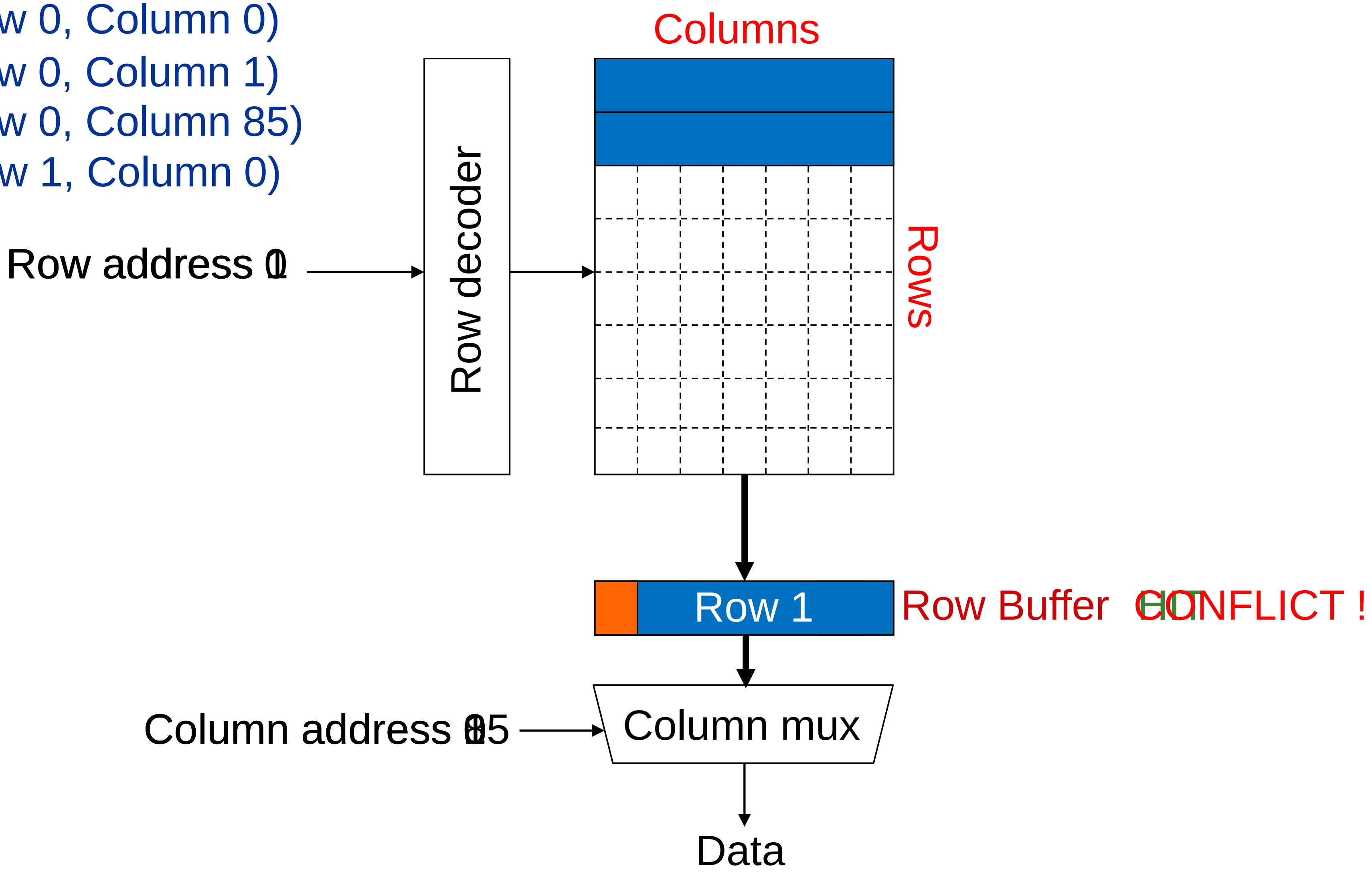
Send back to CPU



The MC redirects the data to the bus to fill the cache

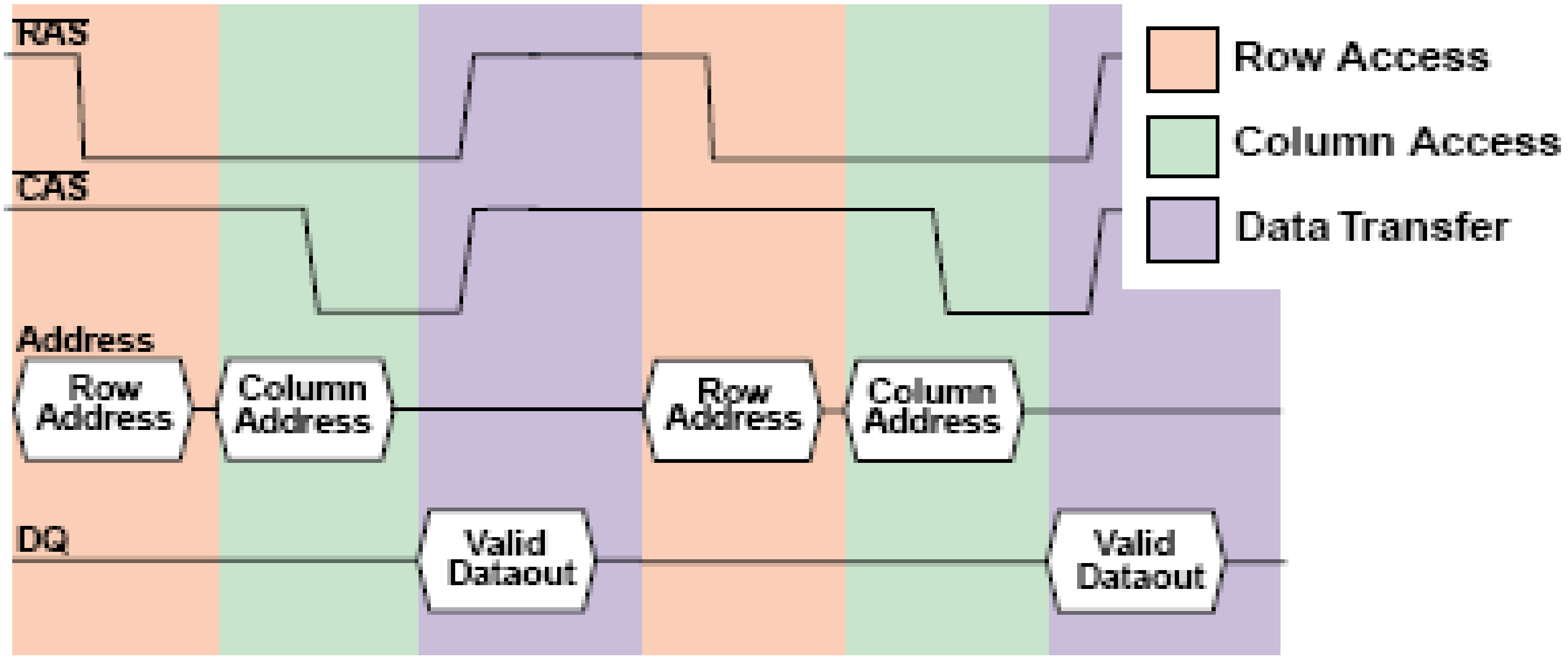
DRAM Bank Operation

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

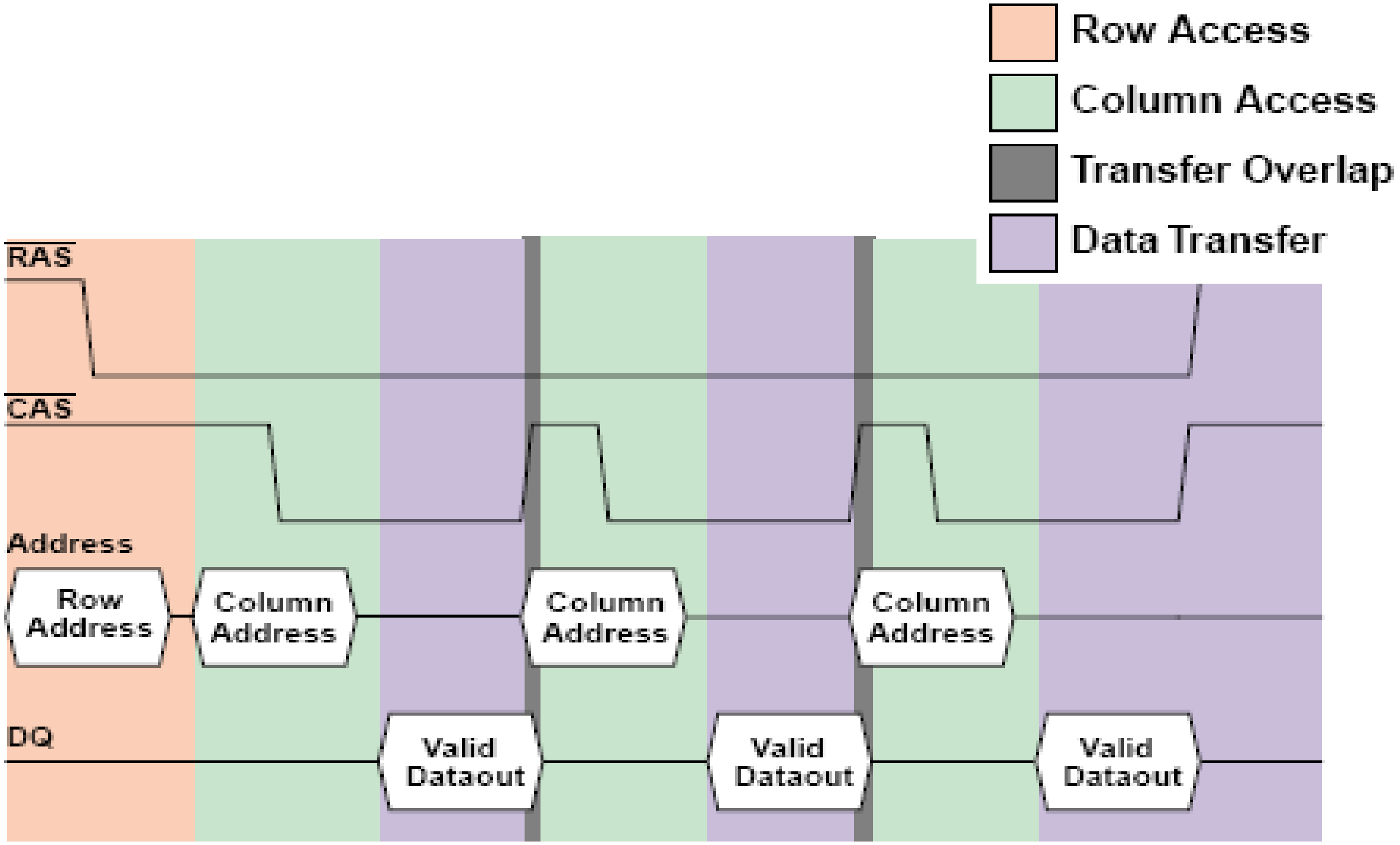


Asynchronous DRAM : Basic timing

Read Timing for Conventional DRAM

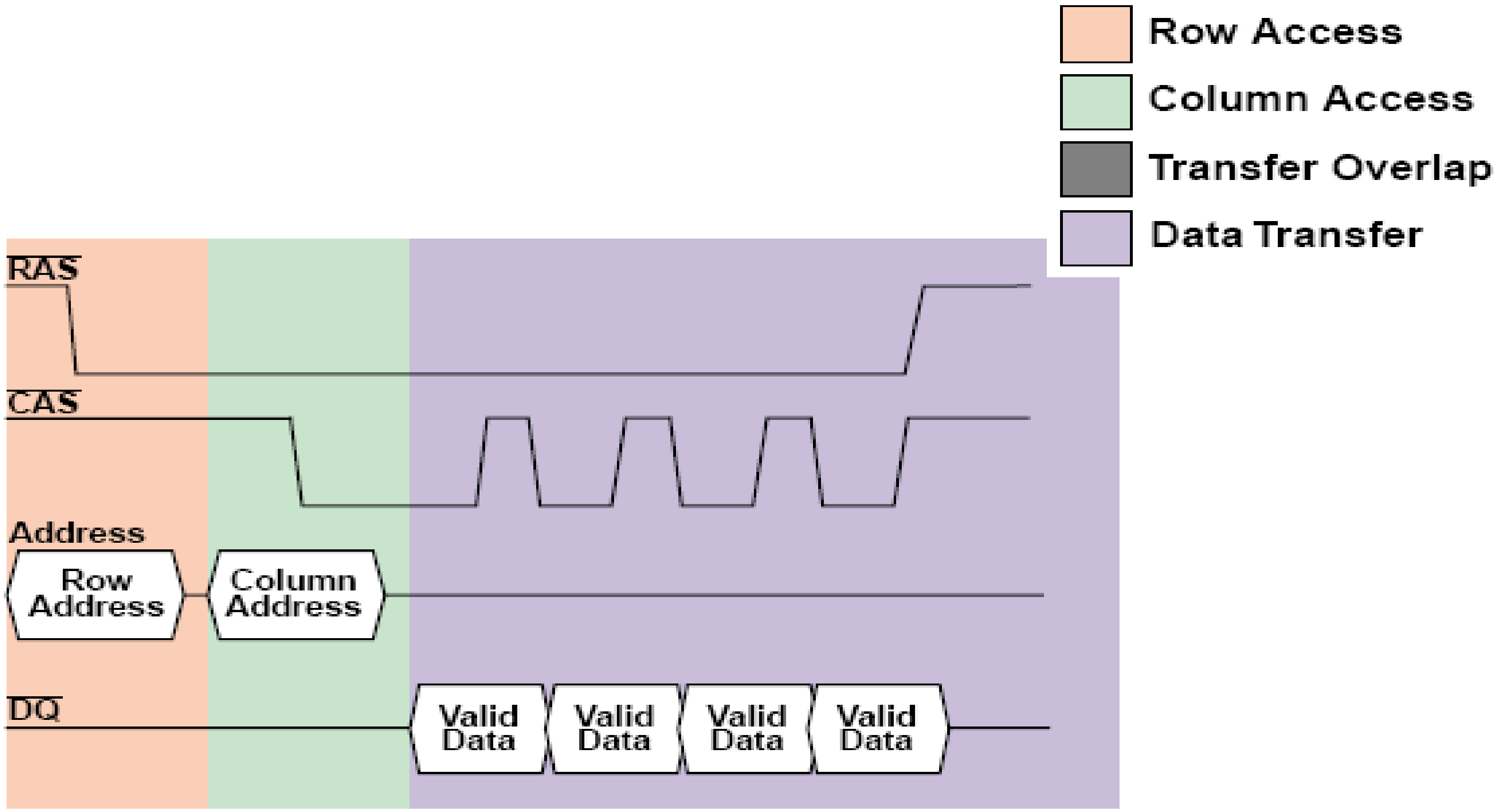


Asynchronous DRAM evolution : Fast Page Mode (FPM)



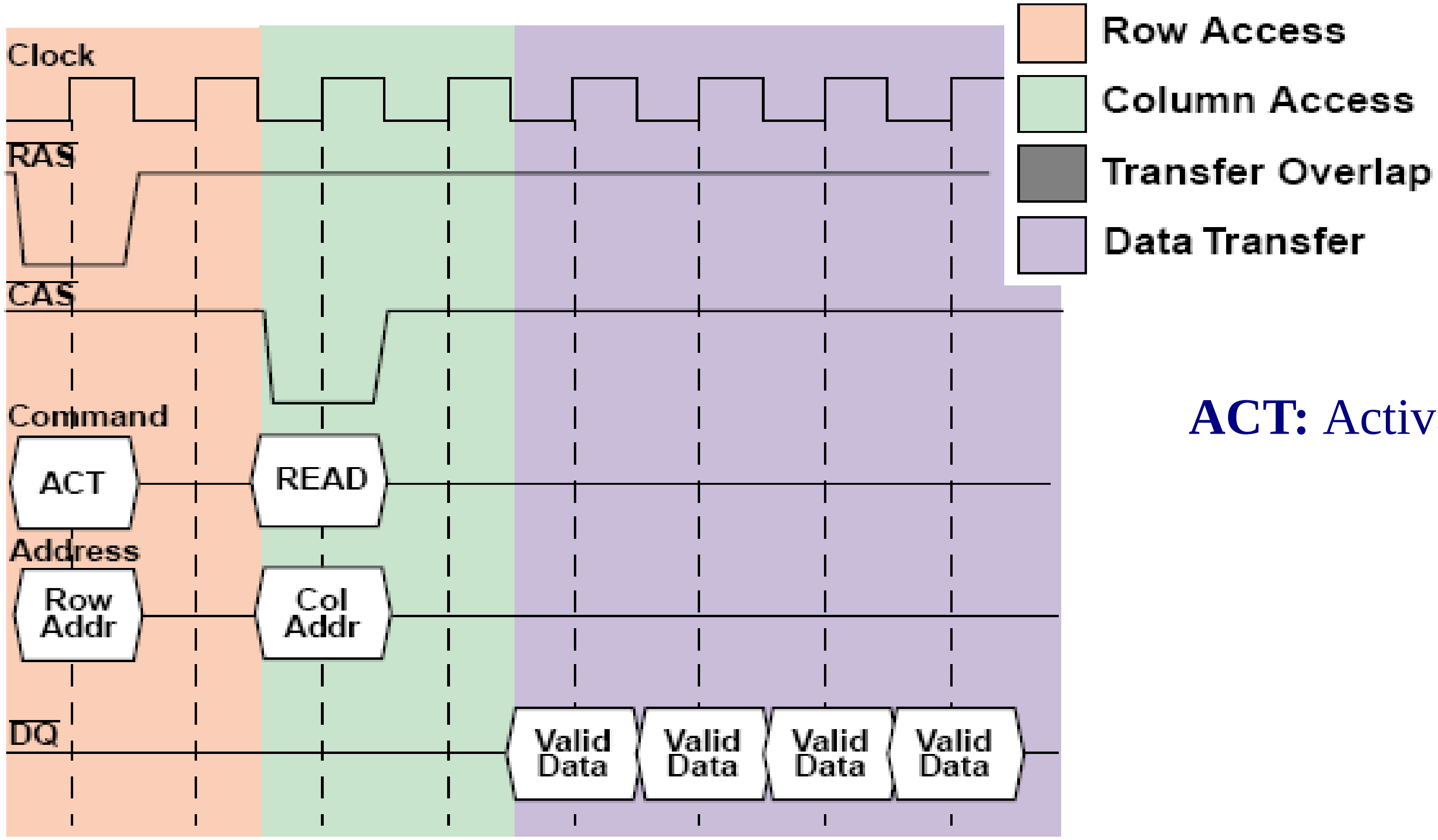
Read row (~1KB) once in the column latch, and reuse data
Data in same row are accessed more quickly.
Exploits spatial locality of memory accesses

Asynchronous DRAM evolution: Burst Mode



Access multiple successive words after the requested word
After initial latency penalty, get 1 word/cycle (e.g. 5-1-1-1)

Synchronous DRAM (SDRAM)

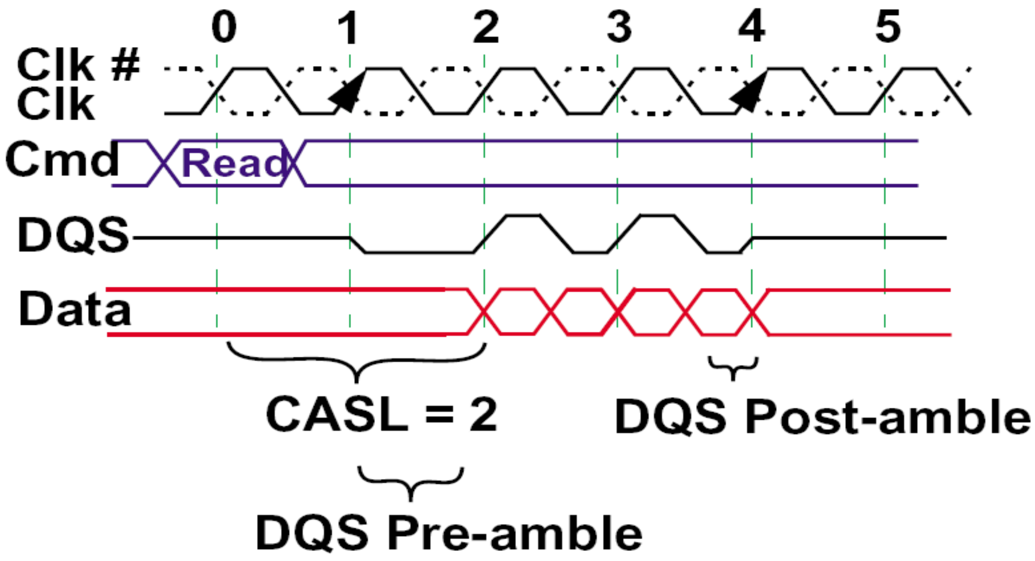


ACT: Activate Row

Add a CLK to avoid synchronization overhead between asynchronous memory array and the bus.

Double Data Rate SDRAM (DDR)

- Transfer data on both positive and negative clock edges
  - doubles peak pin data bandwidth
  - 64-bits transferred at each edge (128 bits per cycle)
  - the frequency of the *memory array* and *bus* is not affected
- Commands still sent only with positive clock edge
  - same pin command bandwidth
  - during random accesses, command bandwidth may limit usable data bandwidth



DDR2 – DDR3 - … - DDRn

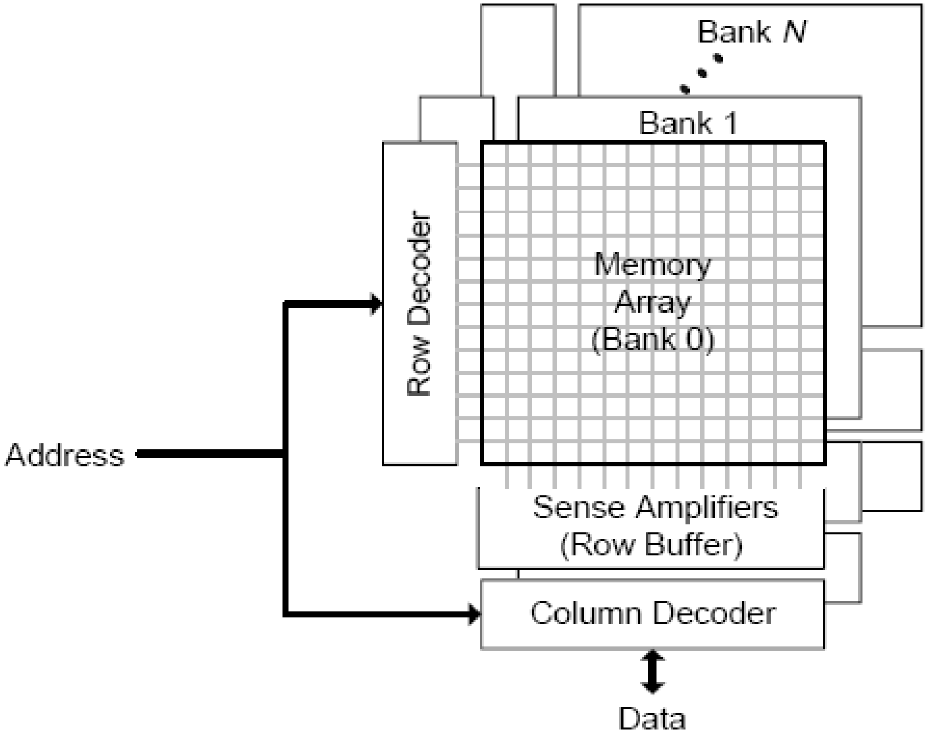
- DDR2 is similar to DDR, with key difference that the bus is clocked twice as fast as in DDR
  - doubles PEAK pin data bandwidth
- Extra buffer stages to sustain high clock frequency
  - Negatively impacts access latency

- Mainly circuit optimizations and improved bus signaling
- Similar for DDR3 (bus clocked four times as fast as in DDR)

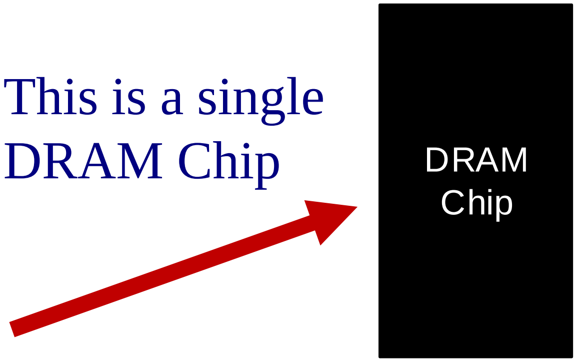
- DDR: Memory Clock = Bus Clock = 133 MHz clock, BW = 266 Mtransfers/sec (DDR266)
- DDR: MC=BC=200 MHz , BW=400 Mtransfers/sec (DDR400)
- DDR2: MC=266MHz, BC=533MHz, BW= 1066Mtransfer/sec (DDR2-1066)
- DDR3: MC=200 MHz, BC=800MHz, BW=1600 Mtransfer/sec (max)
- DDR4: MC=400 MHz, BC=1600MHz, BW=3200 Mtransfer/sec (max)

DRAM at the System Level

SDRAM evolution: Multi-bank memories



This is a single DRAM Chip

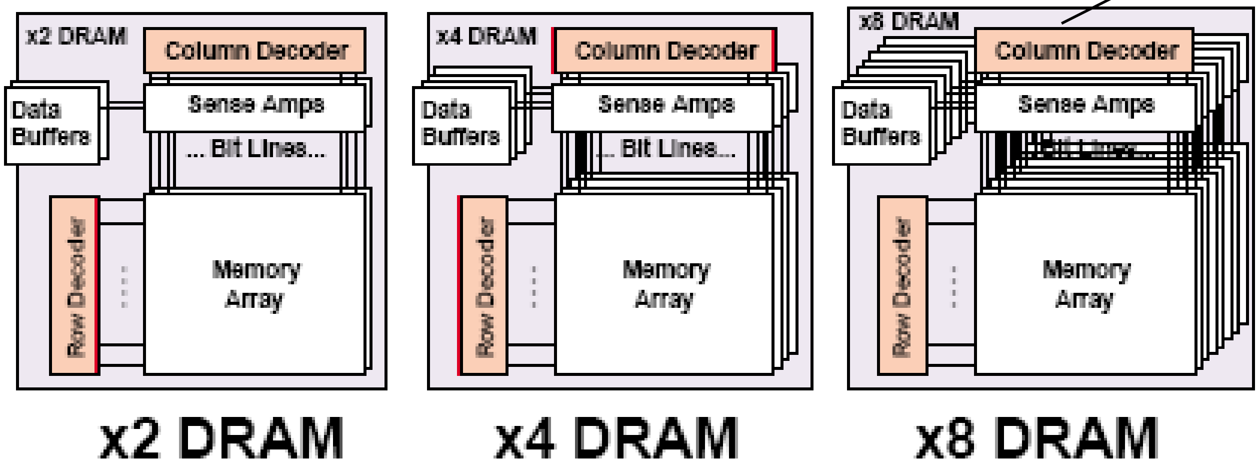


- All modern SDRAMs have multiple, independent banks
- SDRAM command scheme allows overlapped bank operations
  - one bank may be activated and accessed
  - while other banks precharged
  - more efficient use of pin bandwidth

How do we read more than 1 bit?

One bit/array.
Read all arrays simultaneously to get byte

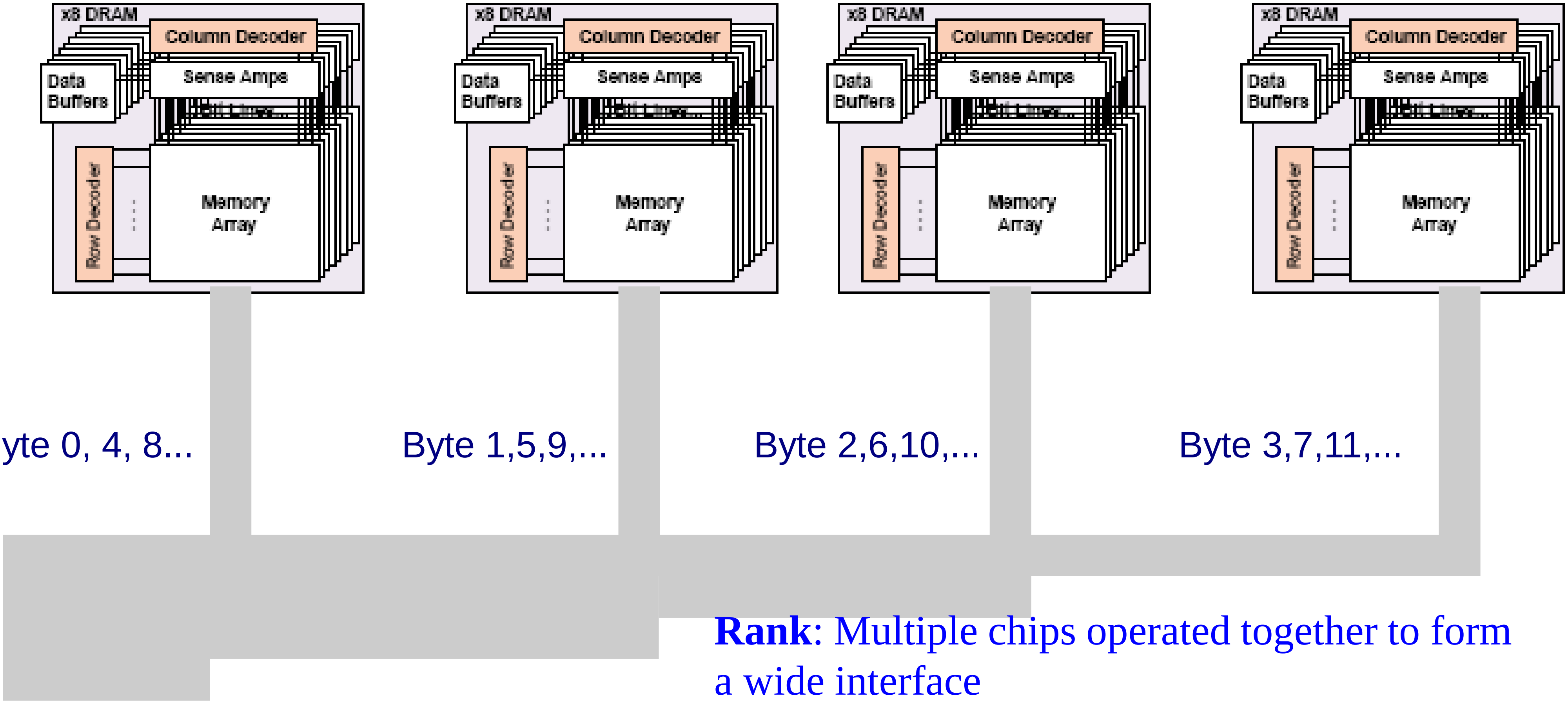
PHYSICAL ORGANIZATION



This is per bank …
Typical DRAMs have 2+ banks

Rank: Wider bus by banks interleaving

Simultaneous access of ALL 4 chips fetches multiple bytes (e.g. for cache fill)



Byte 0, 4, 8...

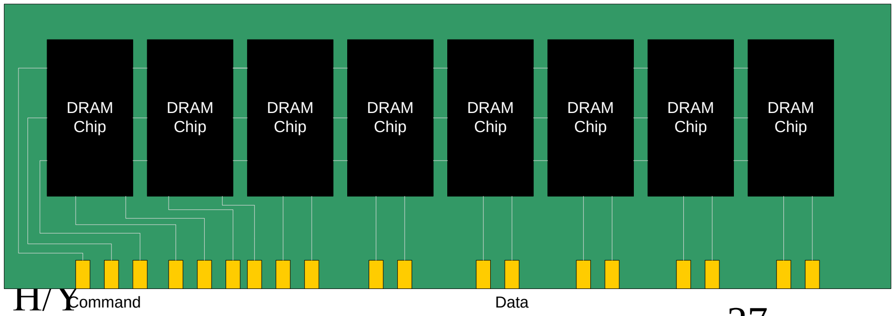
Byte 1,5,9,...

Byte 2,6,10,...

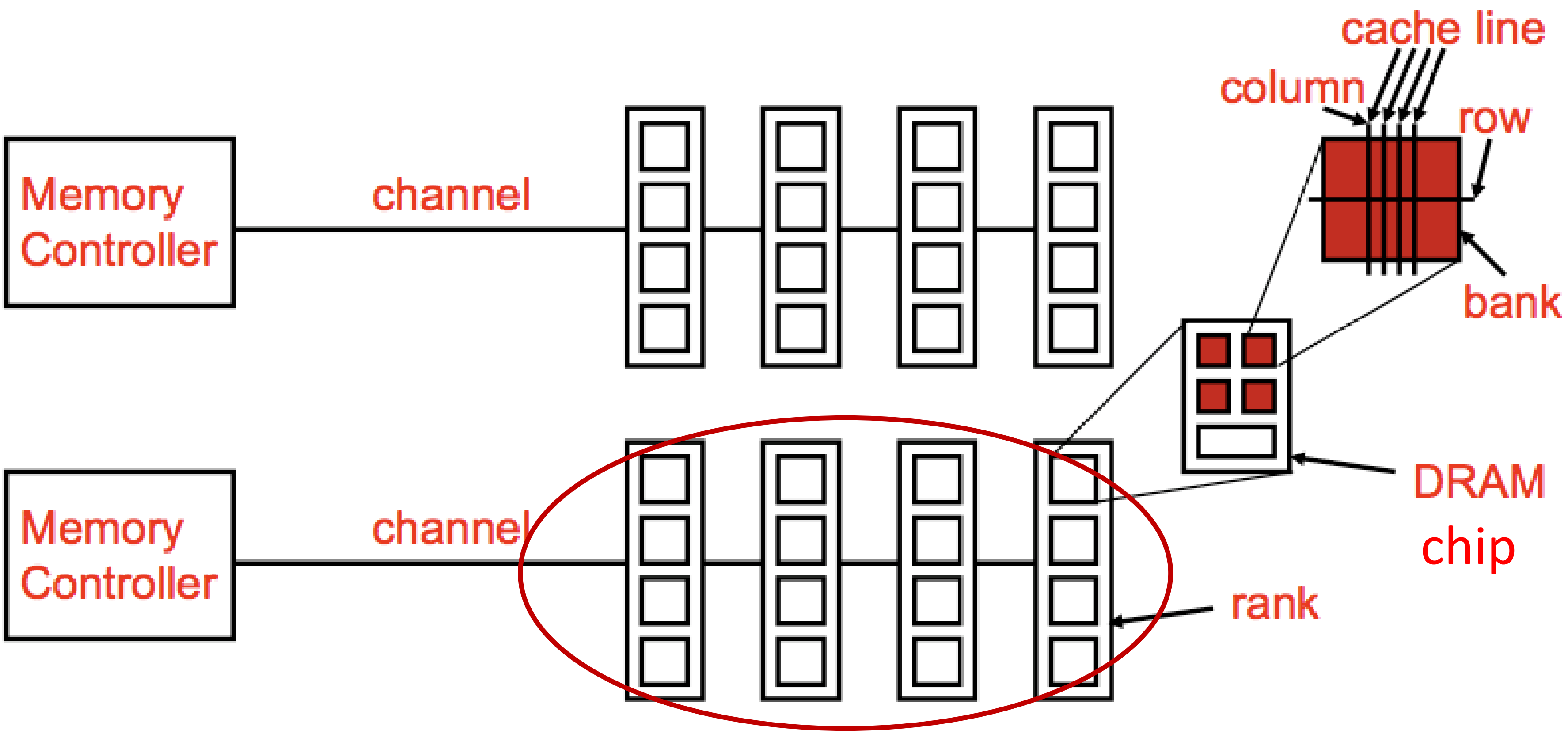
Byte 3,7,11,...

Rank: Multiple chips operated together to form a wide interface

Data Bus 32 bits



Generalized Memory Structure

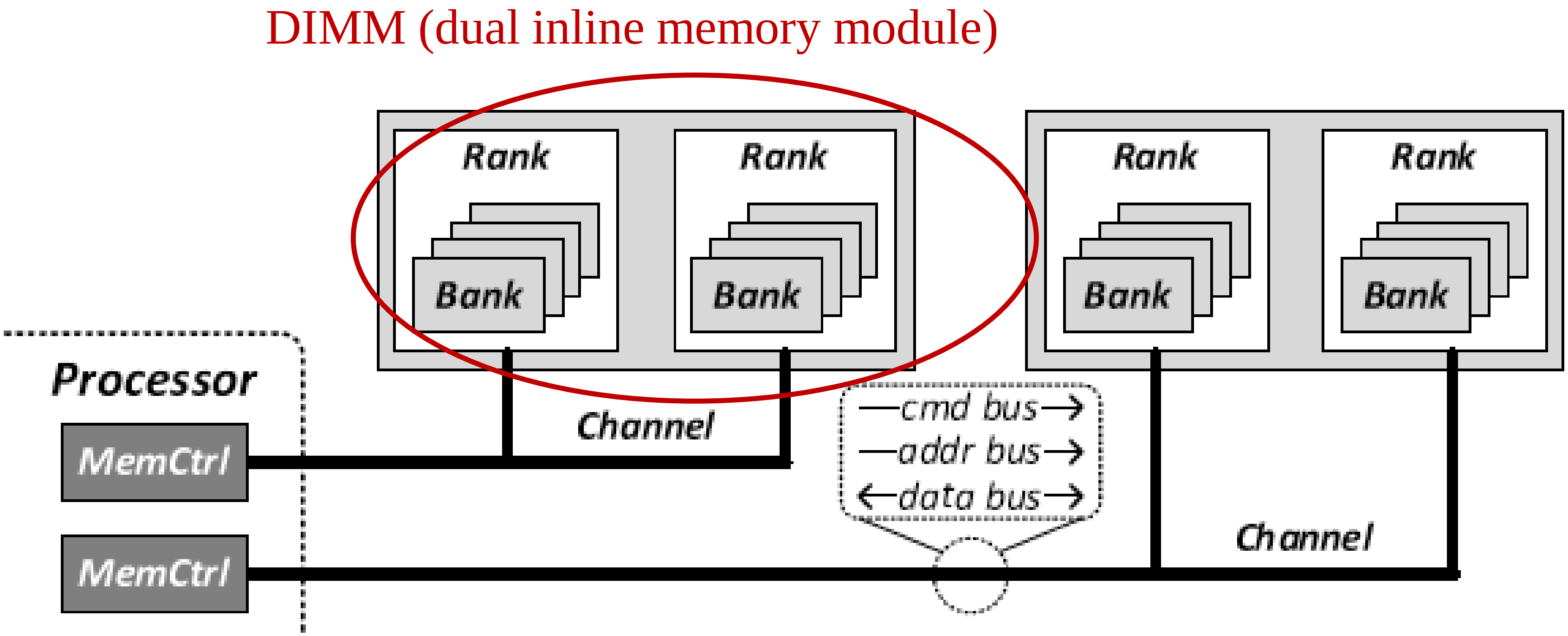


DIMM (dual inline memory module)

A DRAM module consists of one or more ranks

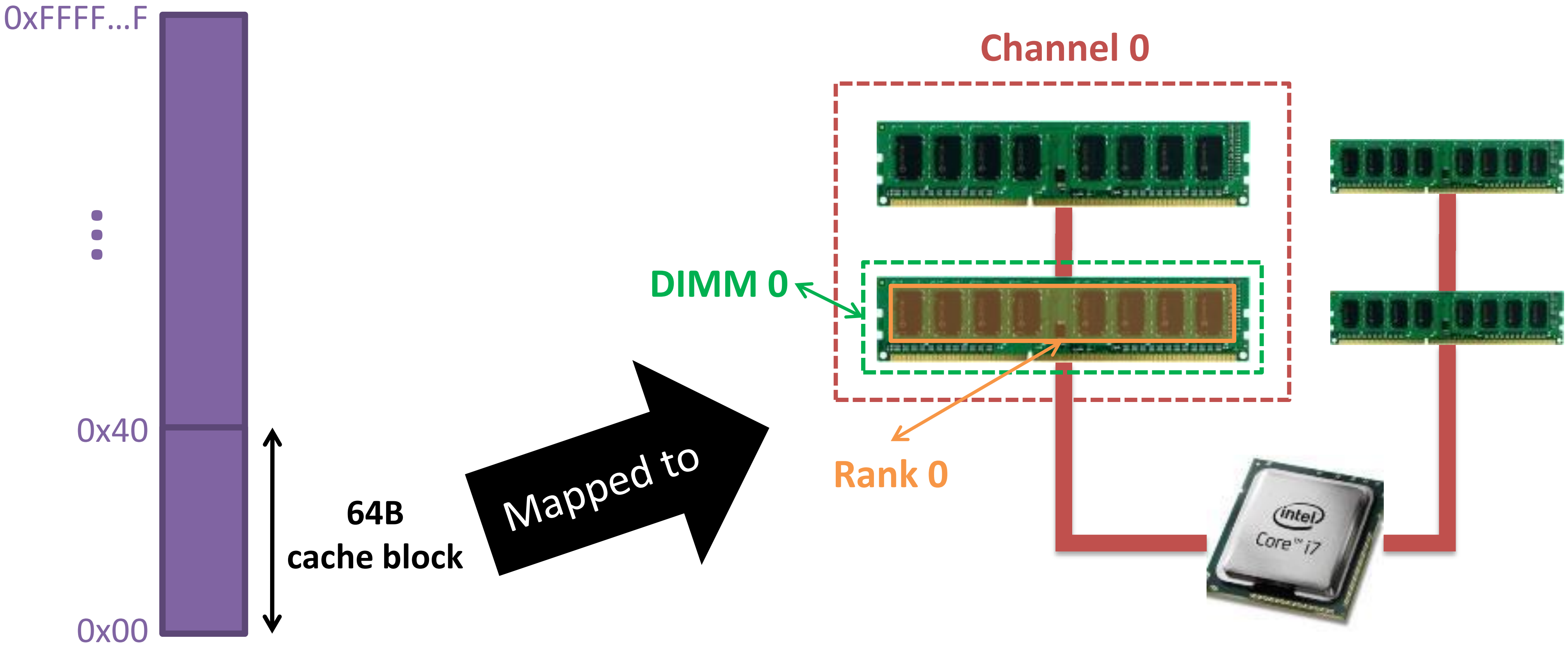
- Also known as DIMMs (dual inline memory modules)
- This is what you plug into your motherboard

Generalized Memory Structure



Example: Transferring a cache block

Physical memory space



0xFFFF…F

0x40

0x00

64B cache block

Mapped to

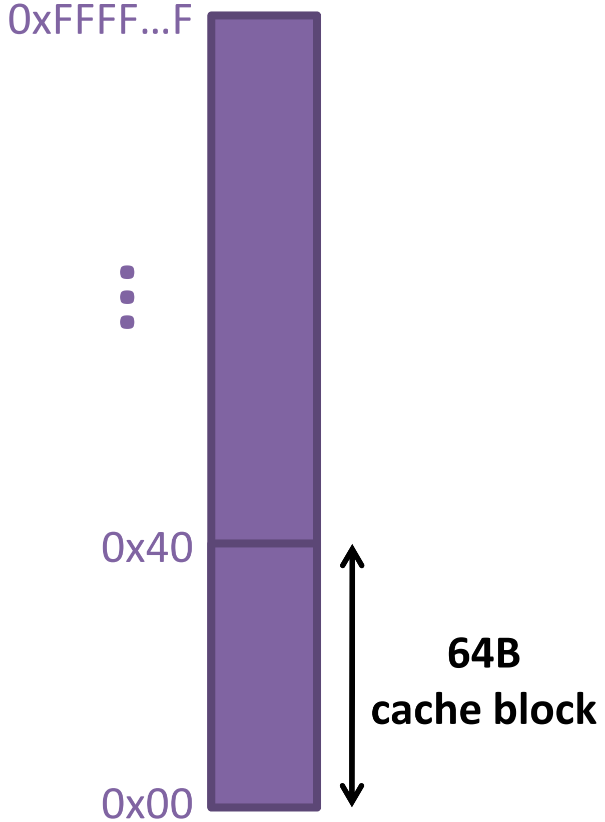
Channel 0

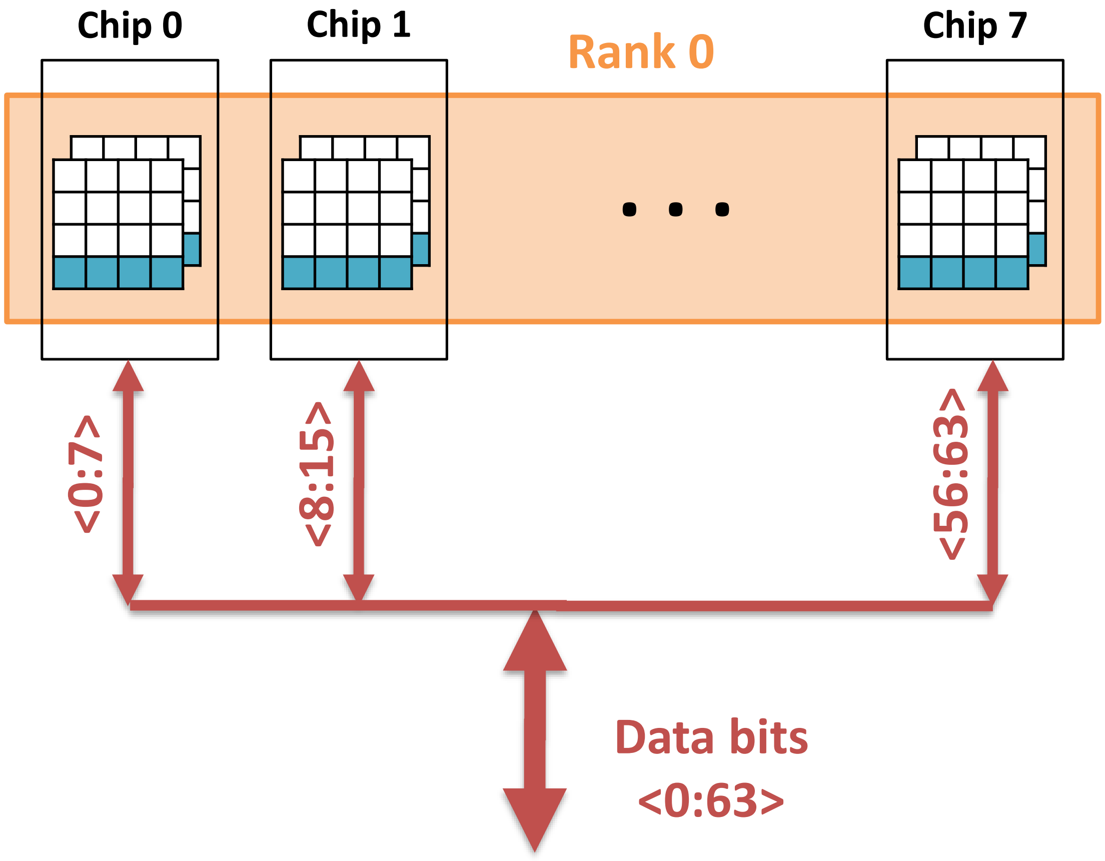
DIMM 0

Rank 0

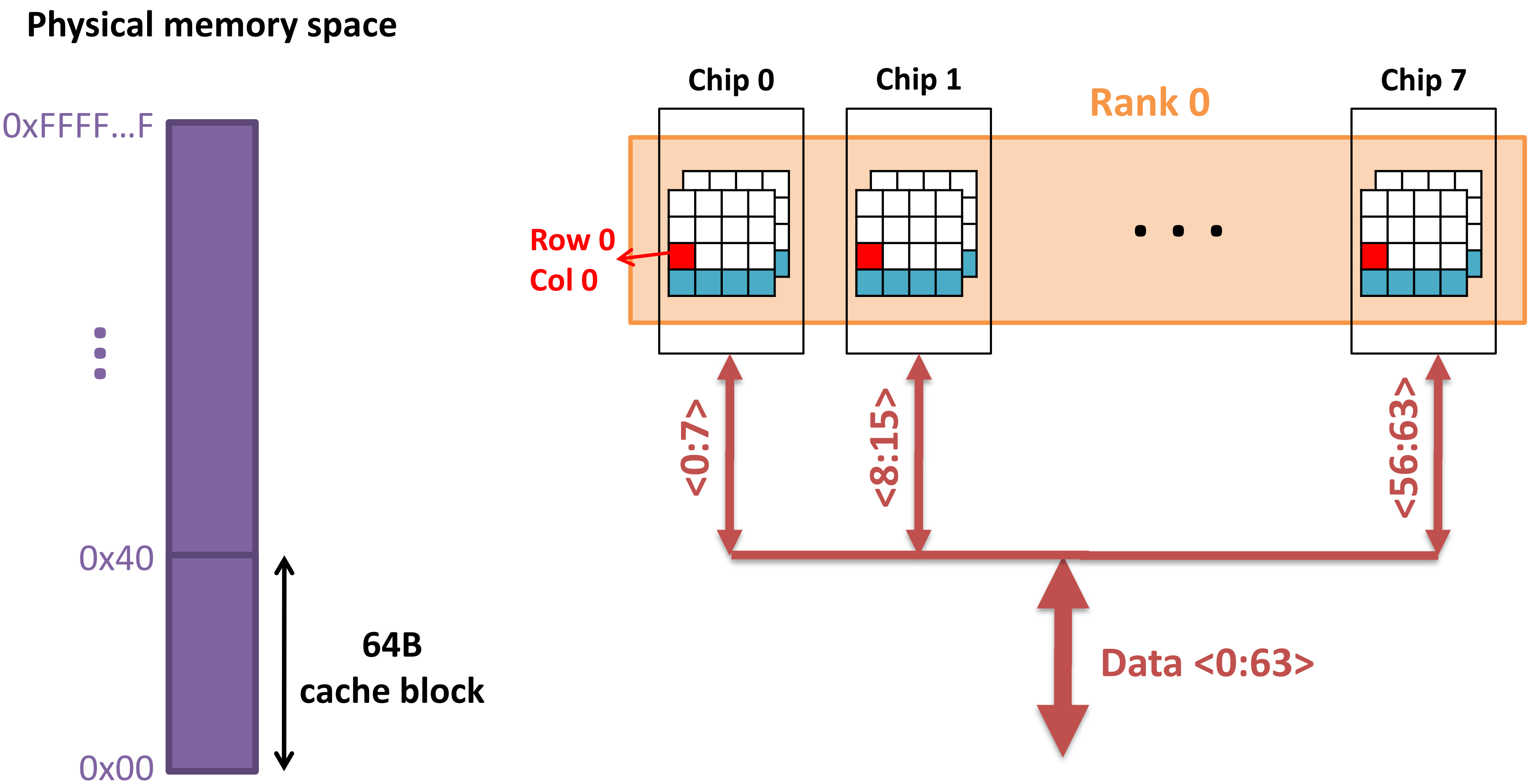
Example: Transferring a cache block

Physical memory space

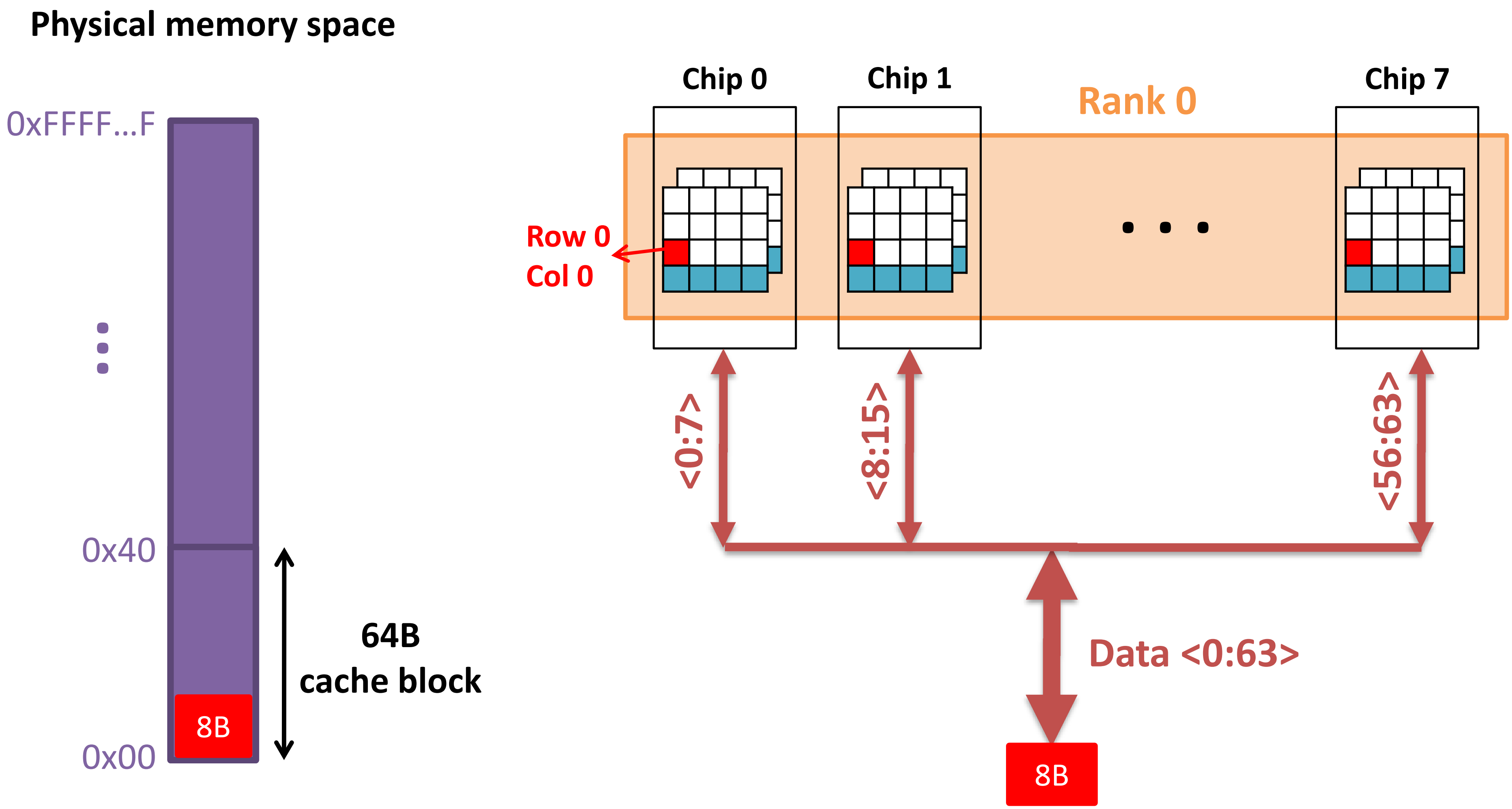




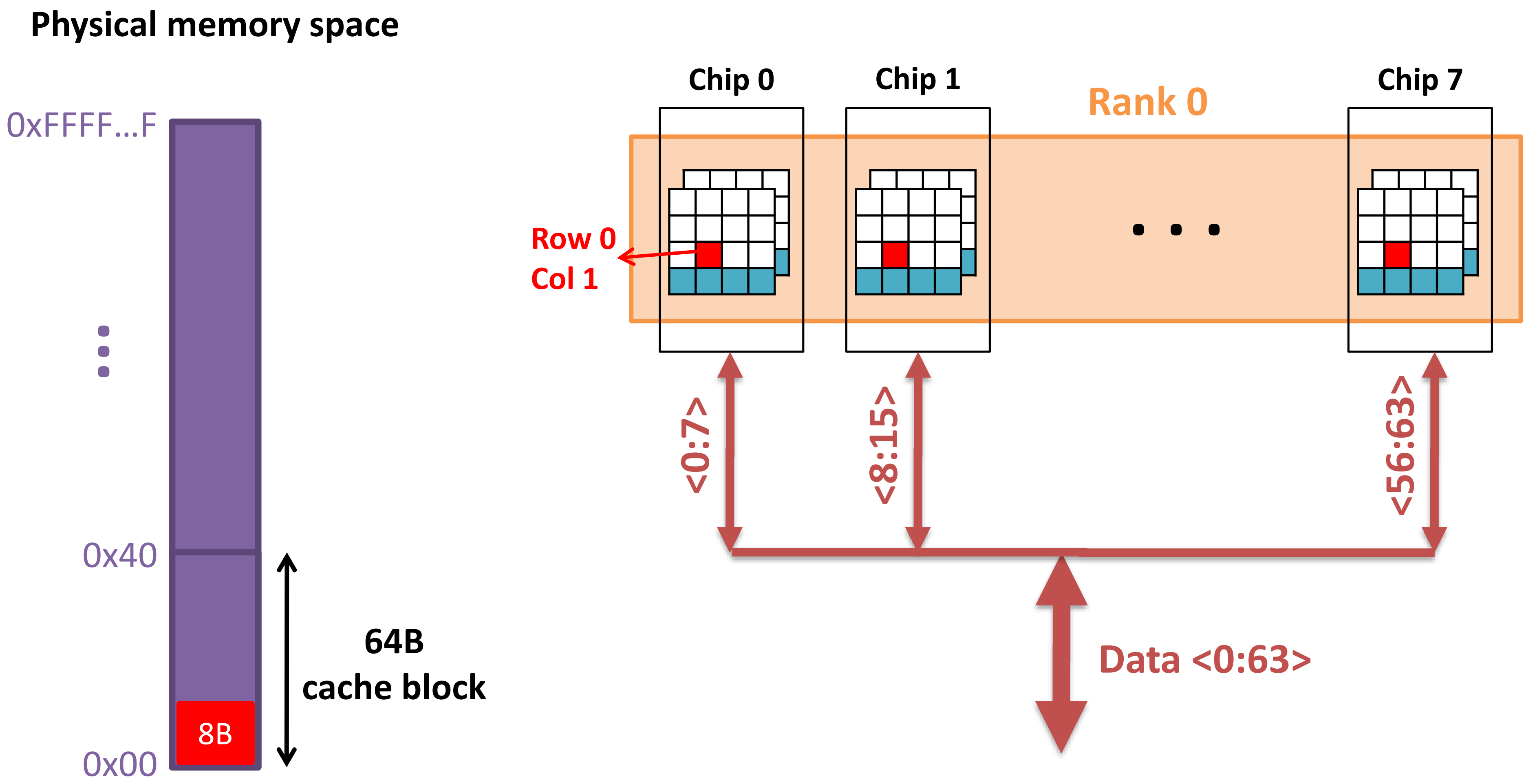
Example: Transferring a cache block



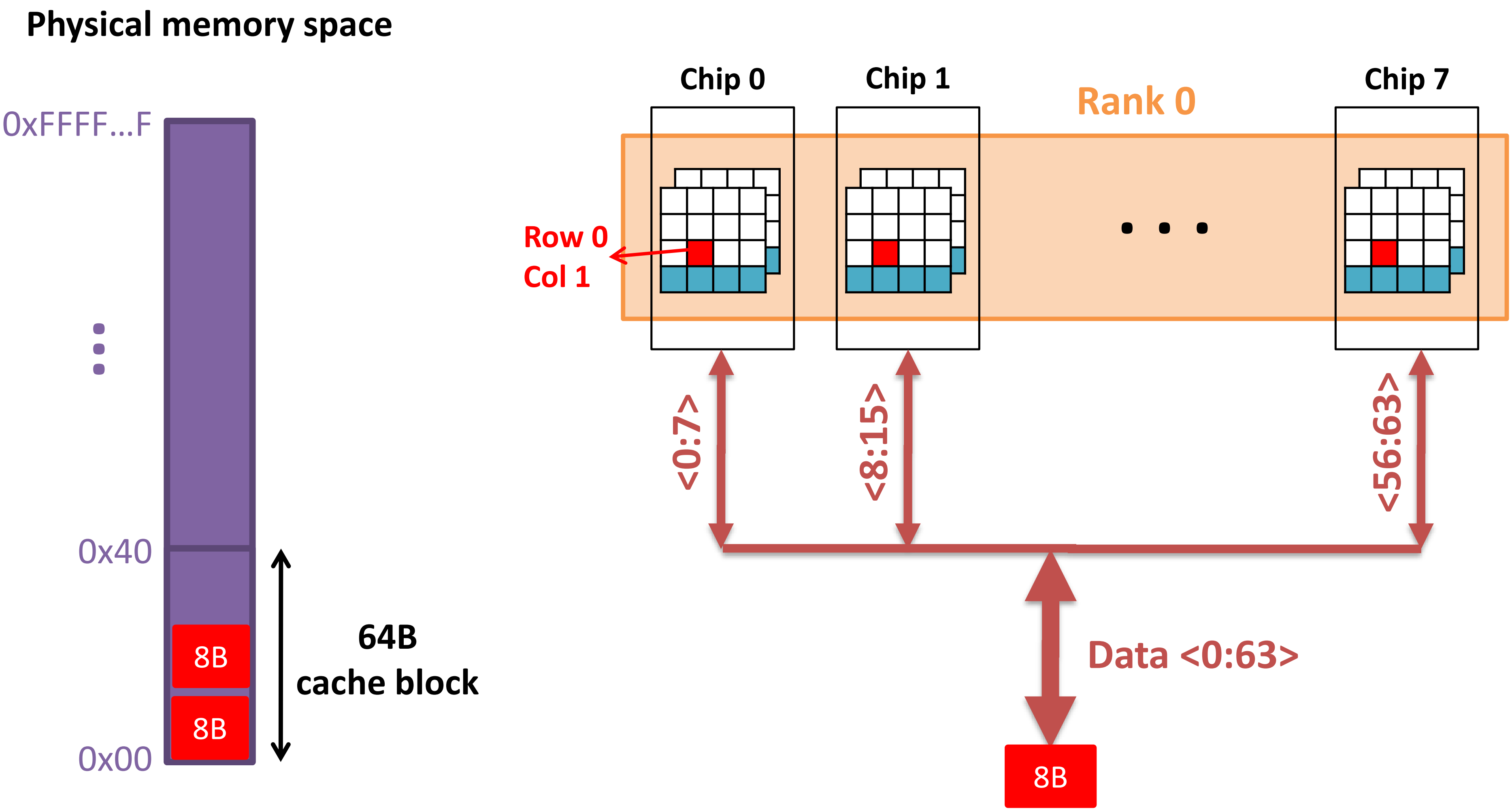
Example: Transferring a cache block



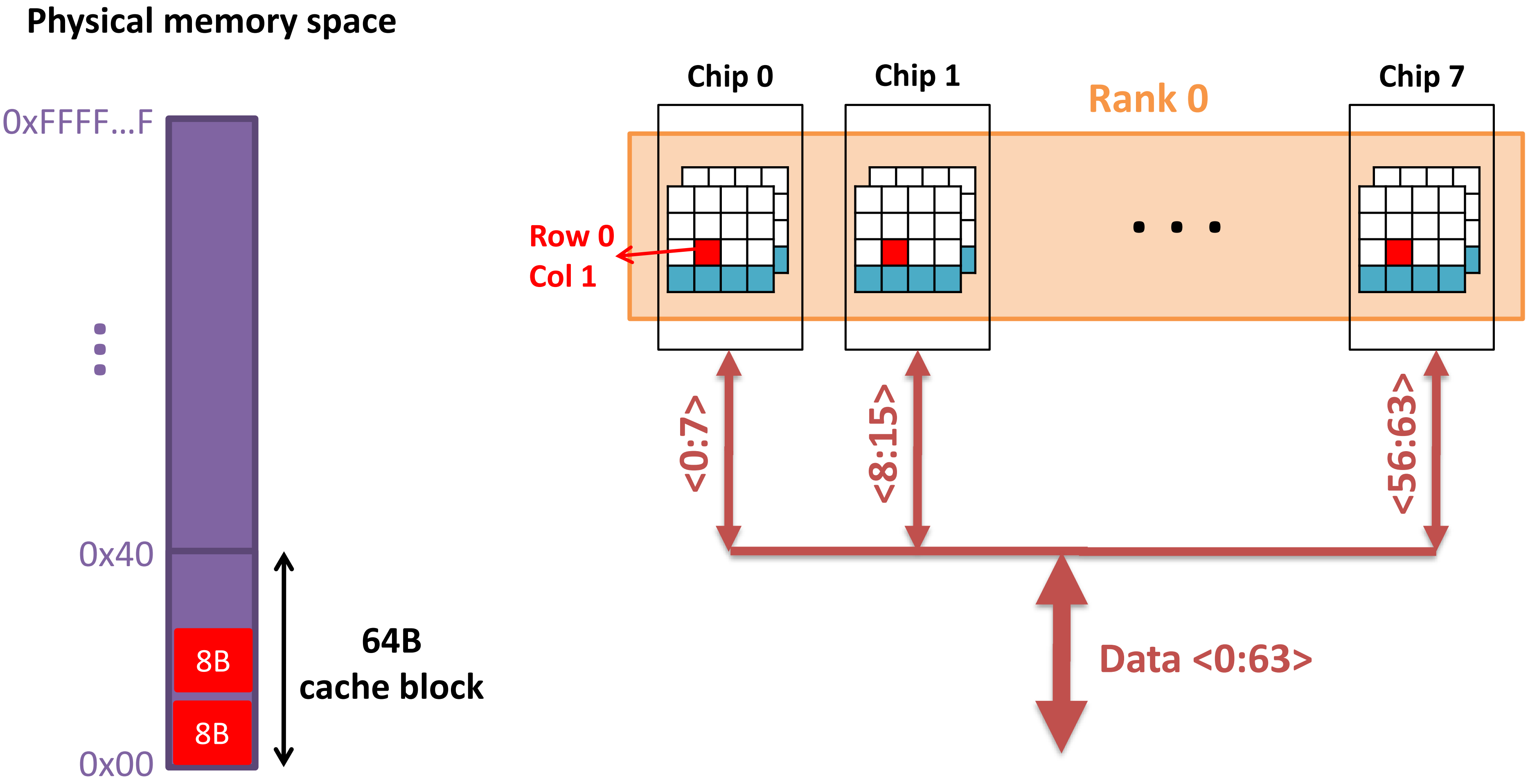
Example: Transferring a cache block



Example: Transferring a cache block



Example: Transferring a cache block

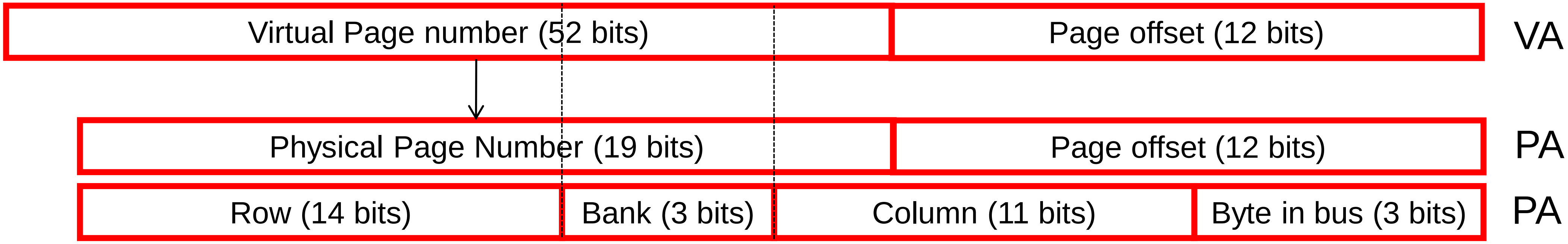


A 64B cache block takes 8 I/O cycles to transfer.

During the process, 8 columns are read sequentially.

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 Page 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.

Basics of Memory Controllers

Memory Access Scheduling : Motivation

- Memory bandwidth a big problem especially for application that do not cache well
  - Multimedia or streaming applications have limited use of the cache due to poor temporal locality
  - Data are read in, processed and thrown away
  - DSP or multimedia processor often limited by poor memory bandwidth
  - Real time requirements (e.g. 30 fps video compression) is an extra bottleneck
- Memory Wall
  - CPU speed improvement (1.2 – 1.52 per year)
  - DRAM latency improvement (1.07 per year)

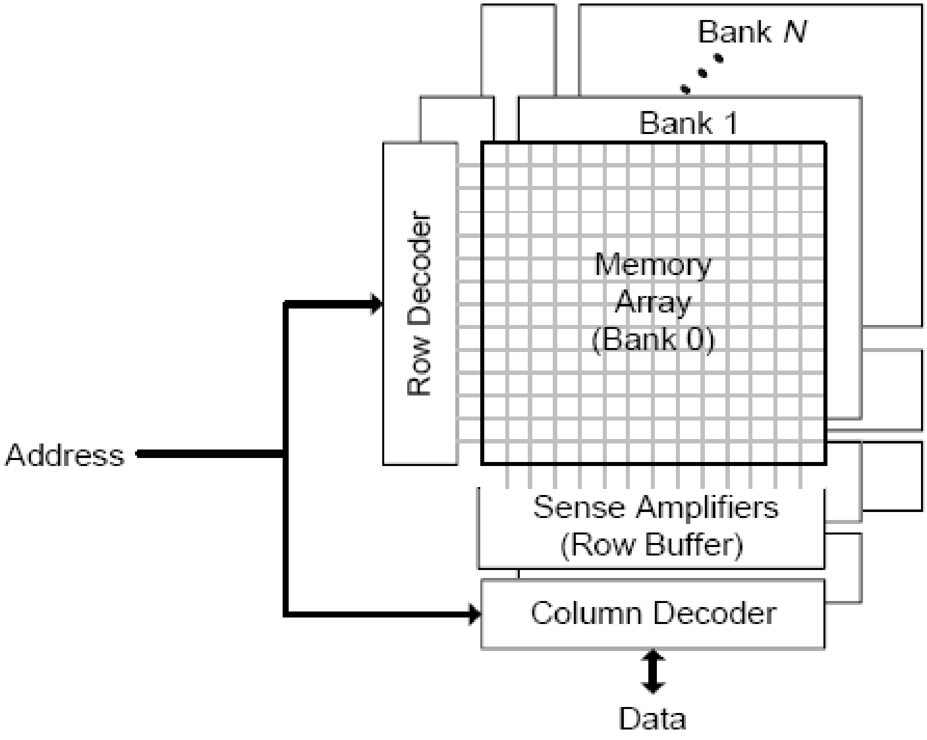
Memory Access Scheduling

- Bandwidth and latency of a memory system STRONGLY dependent on the order of memory accesses

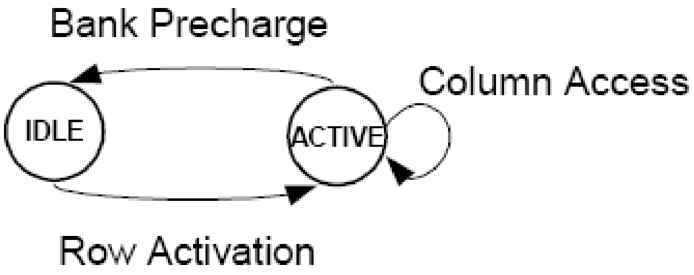
- Modern, multi-bank DRAMs are 3-D structures (Banks, Row, Columns)
  - Access to different columns within a row an order of magnitude faster than accesses to different rows
  - Simultaneous row reads in different banks

- **Memory scheduling** uses the Mem Controller to dynamically reorder access requests to the 3-D memory structure

Memory Access Scheduling

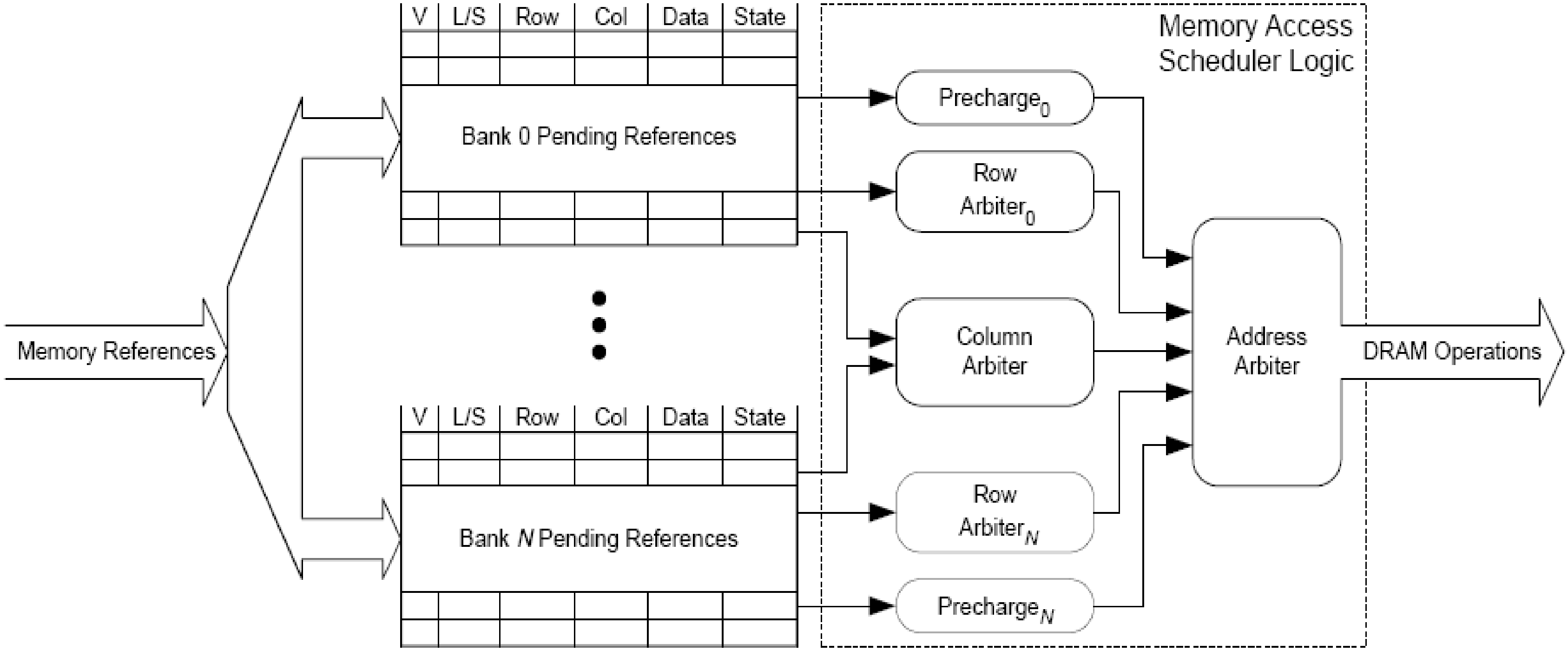


Internal DRAM chip architecture



FSM for bank operation
Each bank has its own FSM
IDLE state: the bank is precharged
ACTIVE state: the bank is being read/written

Memory Access Scheduling

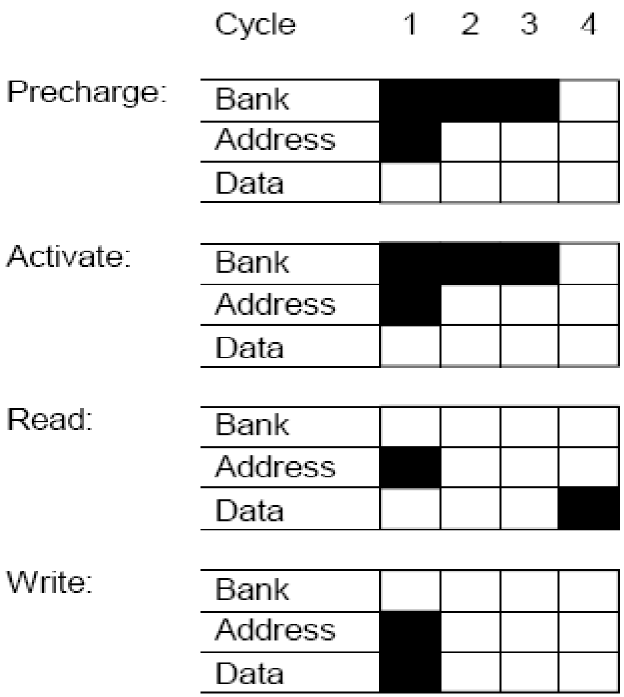


- Given a set of pending memory accesses, a scheduler determines what actions to take next.

- One precharge arbiter per bank, one row arbiter per bank, and a single column arbiter for the common data bus.
- At each cycle, each one of the arbiters takes a decision which request to serve next.
- Arbitration priority determines the exact sequence of accesses
- Split transactions are allowed to break a request and implement out of order request services

Memory Access Scheduling

DRAM operations
P: bank precharge (3 cycle occupancy)
A: row activation (3 cycle occupancy)
C: column access (1 cycle occupancy)



Precharge:	1	2	3	4
Bank	■	■	■	□
Address	■	□	□	□
Data	□	□	□	□

Activate:	1	2	3	4
Bank	■	■	■	□
Address	■	□	□	□
Data	□	□	□	□

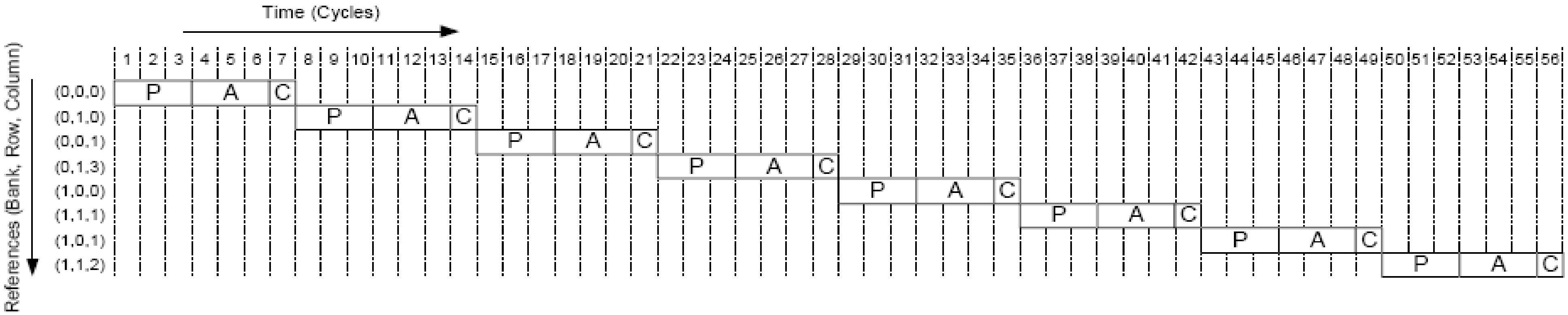
Read:	1	2	3	4
Bank	□	□	□	□
Address	■	□	□	□
Data	□	□	□	■

Write:	1	2	3	4
Bank	□	□	□	□
Address	■	□	□	□
Data	■	□	□	□

Resource utilization

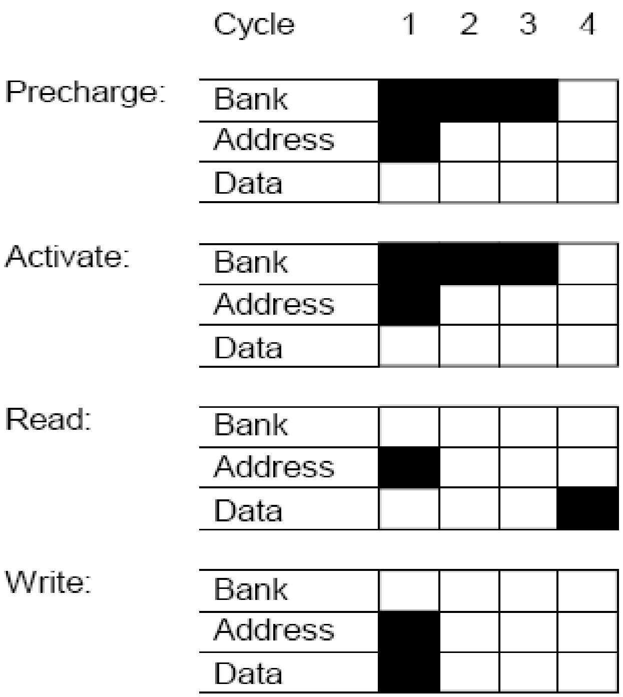
(Bank, Row, Column)

(A) Without access scheduling (56 DRAM Cycles)



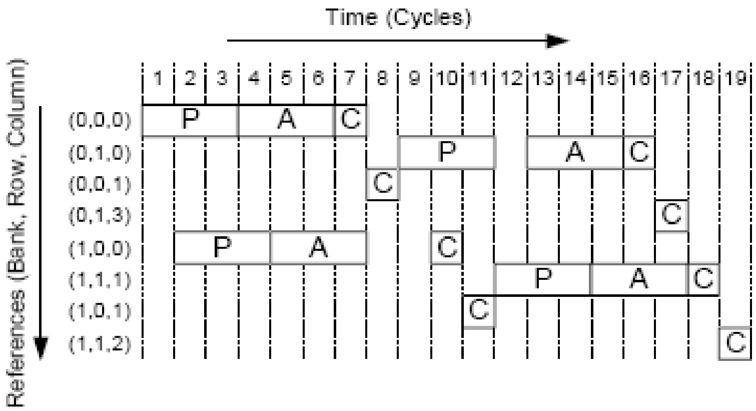
Οργάνωση και Σχεδιάση Η/Υ
(ΗΥ232)

Memory Access Scheduling



Resource utilization

(B) With access scheduling (19 DRAM Cycles)



Example arbitration policy

the row with the fewest pending column accesses is selected next. This minimizes the time that rows with little demand remain active, allowing other rows in the same bank to make progress sooner.