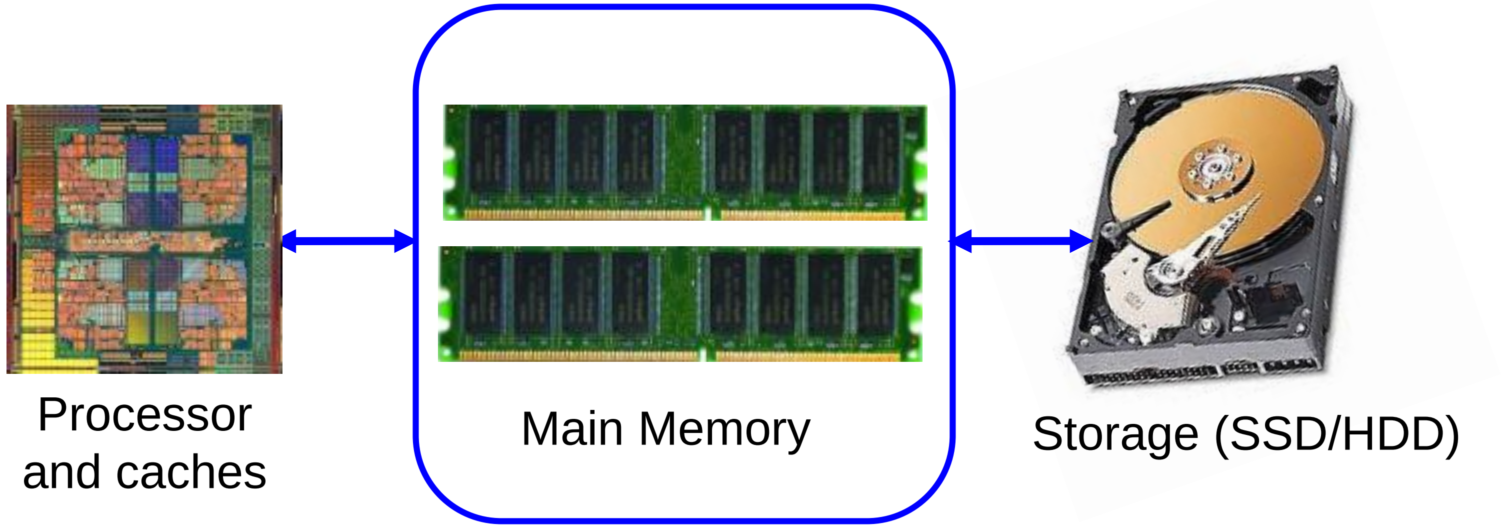
Computer Architecture

12. Main Memory

Jianhua Li
College of Computer and Information
Hefei University of Technology

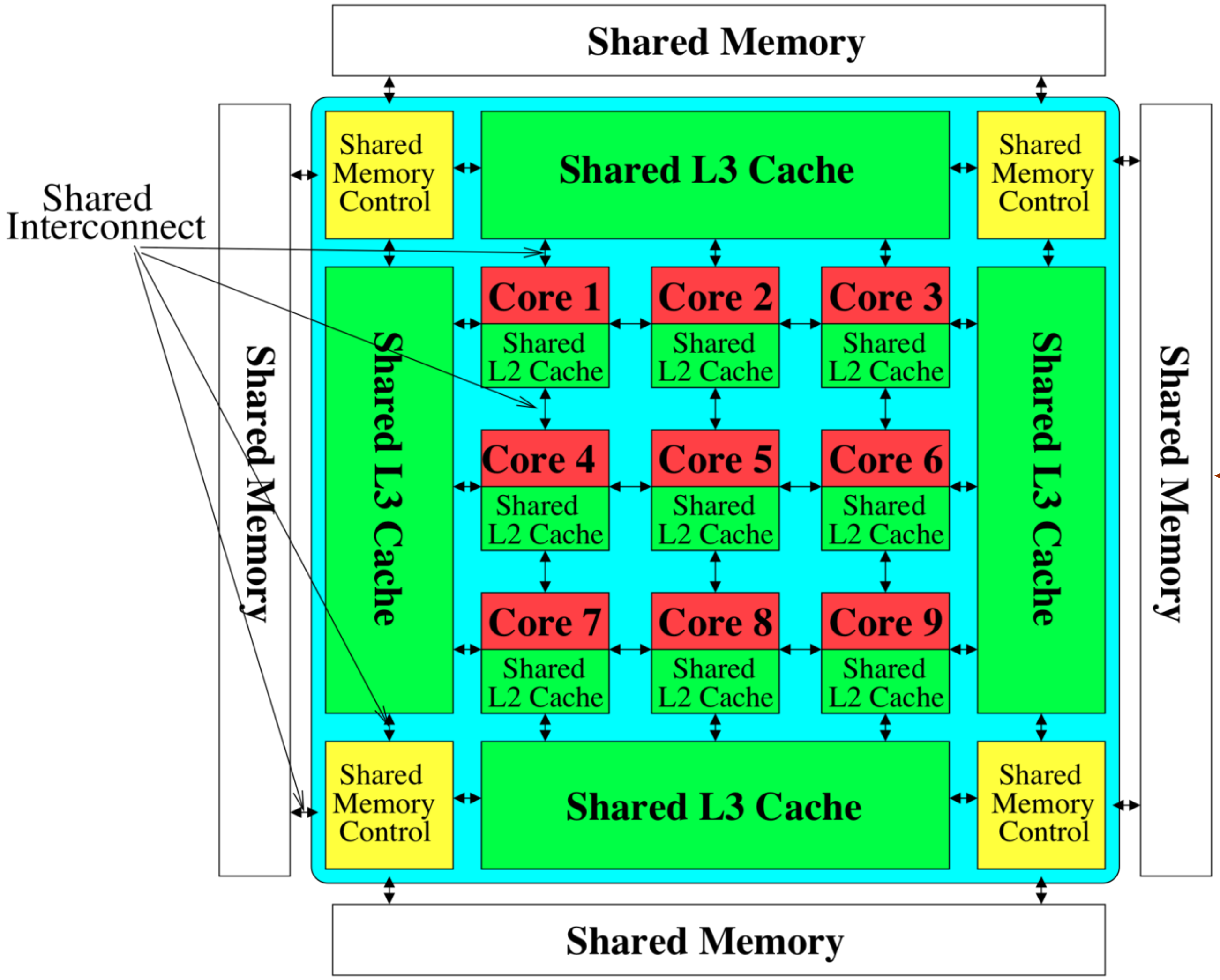
The Main Memory System

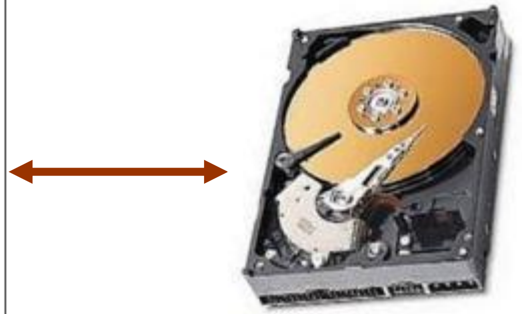


- Main memory is a critical component of all computing systems: server, mobile, embedded, desktop, sensor

- Main memory system must scale (in *size*, *technology*, *efficiency*, *cost*, and *management algorithms*) to match the growing demands of bandwidths

Memory System: A Shared Resource View





Storage

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/reinvent the main memory system
  - to fix DRAM issues and enable emerging technologies
  - to satisfy all requirements

Major Trends Affecting Main Memory

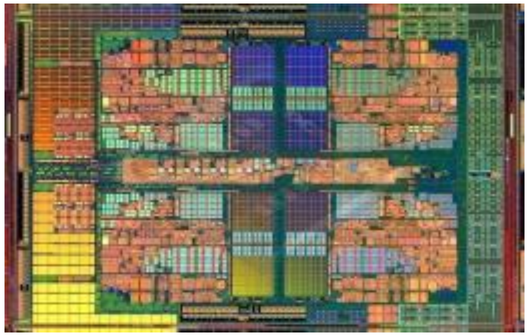
- Need for main memory capacity, bandwidth, QoS increasing

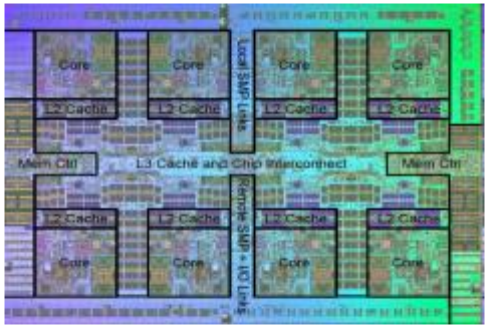
- Main memory energy/power is a key system design concern

- DRAM technology scaling is ending

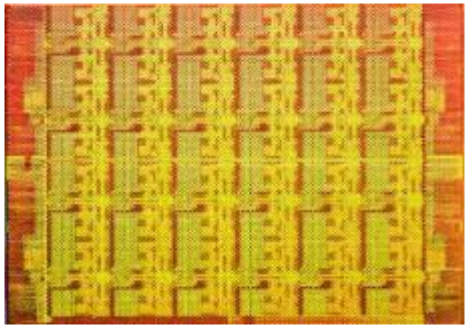
Demand for Memory Capacity

- More cores ➔ More concurrency ➔ Larger working set


AMD Barcelona: 4 cores



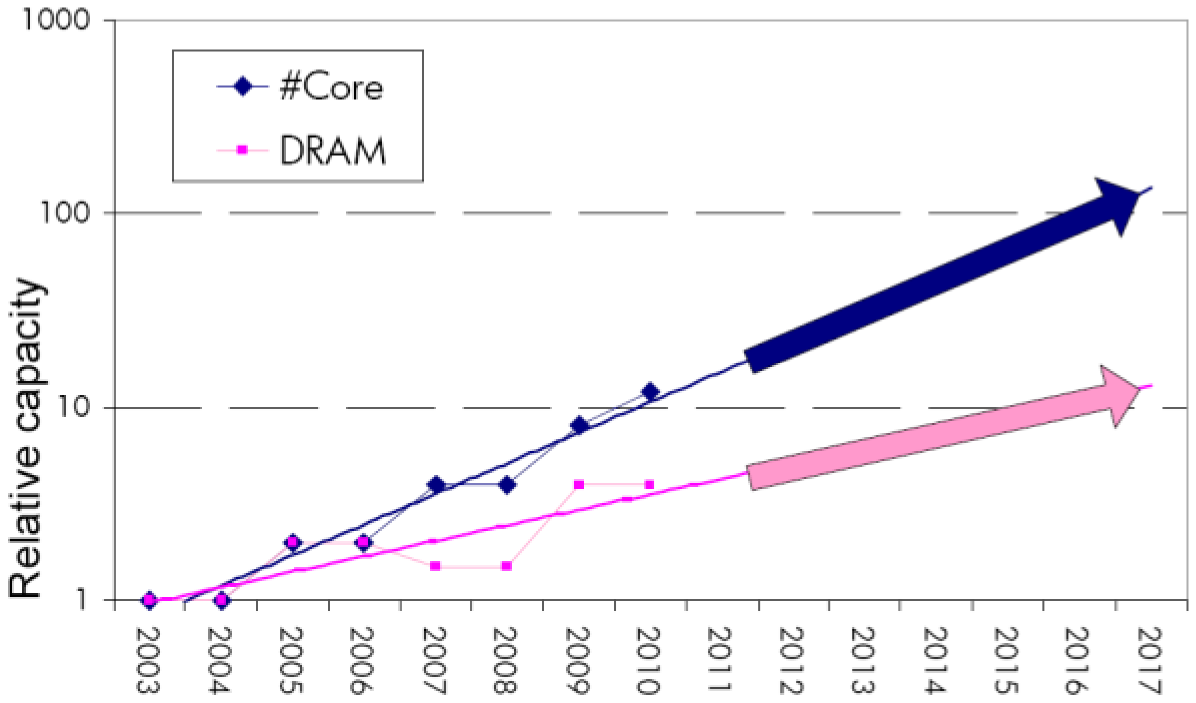
IBM Power7: 8 cores


Intel SCC: 48 cores

- Modern applications are (increasingly) data-intensive
- Many applications/virtual machines (will) share main memory
  - Cloud computing/servers: Consolidation to improve efficiency
  - GP-GPUs: Many threads from multiple parallel applications
  - Mobile: Interactive + non-interactive consolidation

Example: The Memory Capacity Gap

Core count doubling ~ every 2 years
DRAM DIMM capacity doubling ~ every 3 years



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

Major Trends Affecting Main Memory

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

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

- DRAM technology scaling is ending

Major Trends Affecting Main Memory

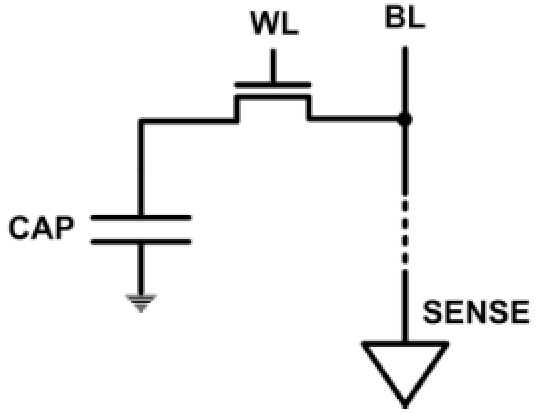
- Need for main memory capacity, bandwidth, QoS increasing
  - Multi-core: increasing number of cores
  - Data-intensive applications: increasing demand for data
  - Consolidation: Cloud computing, GPUs, mobile, heterogeneity
- Main memory energy/power is a key system design concern
  - IBM servers: ~50% energy spent in off-chip memory hierarchy [Lefurgy, IEEE Computer 2003]
  - DRAM consumes power when idle and needs periodic refresh
- DRAM technology scaling is ending

Major Trends Affecting Main Memory

- Need for main memory capacity, bandwidth, QoS increasing
  - Multi-core: increasing number of cores
  - Data-intensive applications: increasing demand for data
  - Consolidation: Cloud computing, GPUs, mobile, heterogeneity
- Main memory energy/power is a key system design concern
  - IBM servers: ~50% energy spent in off-chip memory hierarchy [Lefurgy, IEEE Computer 2003]
  - DRAM consumes power when idle and needs periodic refresh
- DRAM technology scaling is ending
  - ITRS projects DRAM will not scale easily below X nm
  - Scaling has provided many benefits:
    - higher capacity, higher density, lower cost, lower energy

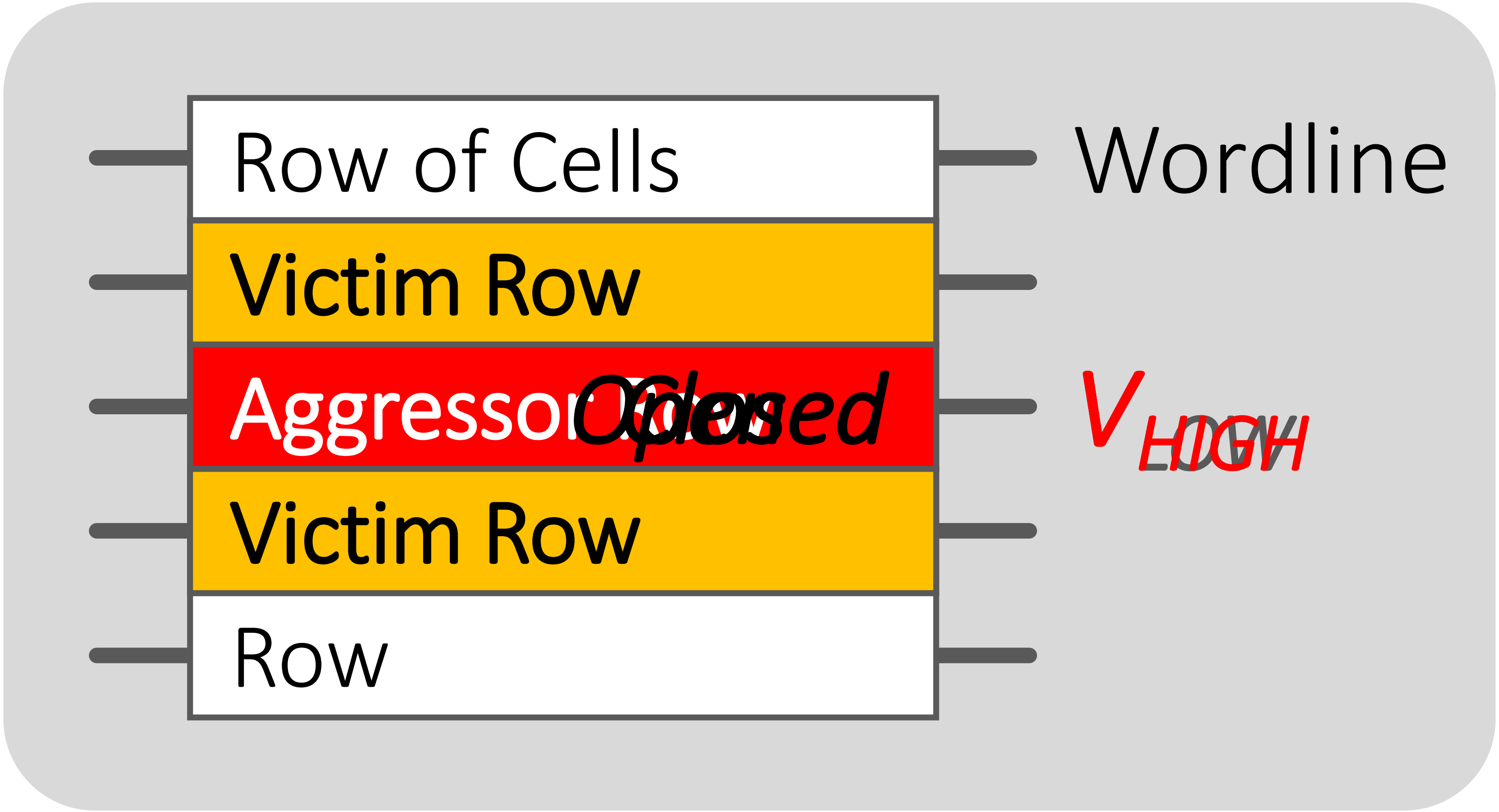
The DRAM Scaling Problem

- 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

Evidence of the DRAM Scaling Problem

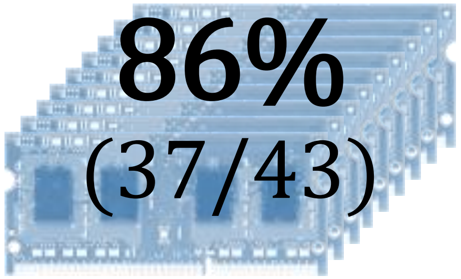
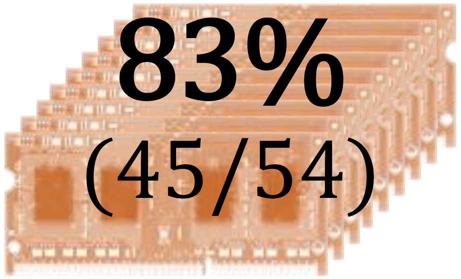
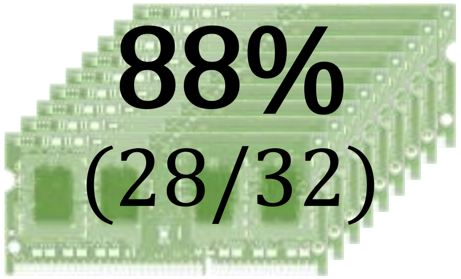


Repeatedly opening and closing a row enough times within a refresh interval induces disturbance errors in adjacent rows in most real DRAM chips you can buy today

Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

Most DRAM Modules Are At Risk

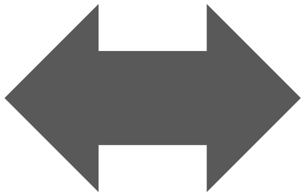
A company	B company	C company
86% (37/43)	**83%** (45/54)	**88%** (28/32)
Up to **1.0×10^7** errors	Up to **2.7×10^6** errors	Up to **3.3×10^5** errors

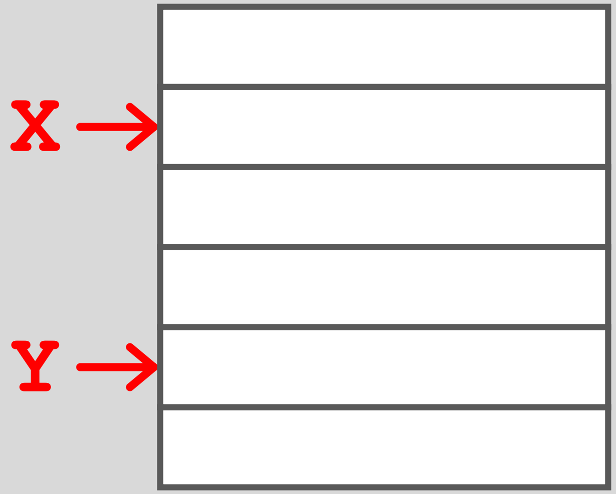
Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

x86 CPU

DRAM Module

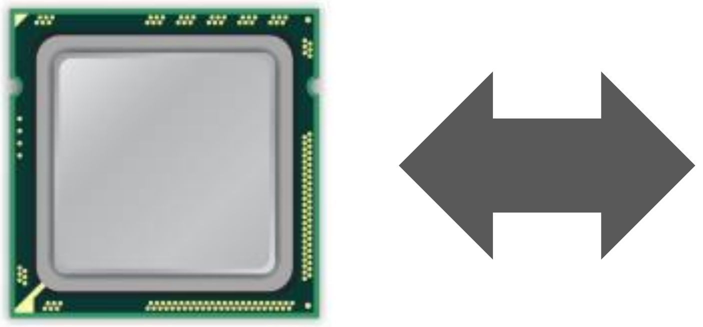


```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```

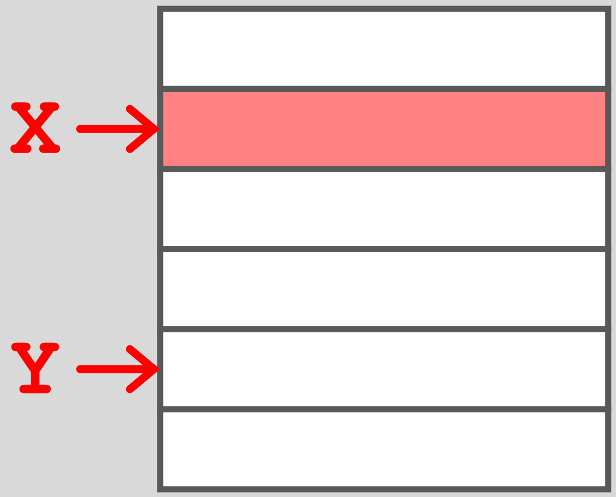


x86 CPU

DRAM Module

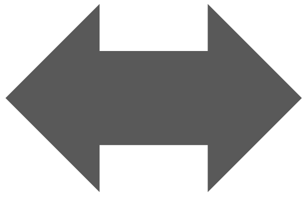


```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```

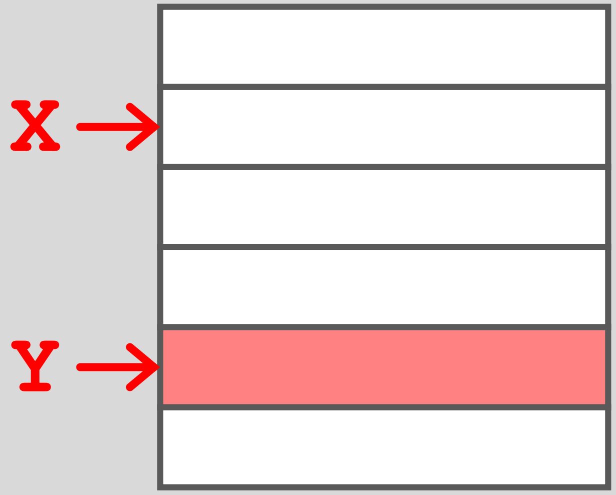


x86 CPU

DRAM Module

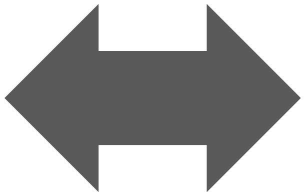


```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```

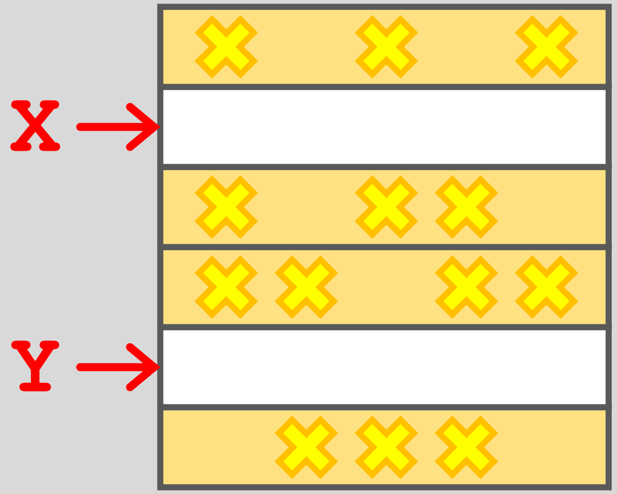


x86 CPU

DRAM Module



```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```



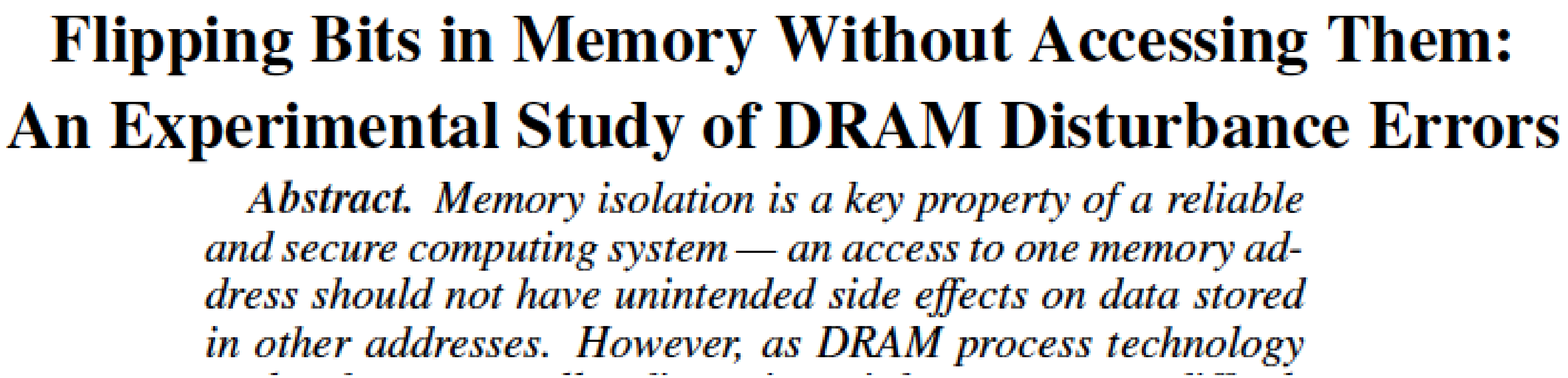
Observed Errors in Real Systems

CPU Architecture	Errors	Access-Rate
Intel Haswell (2013)	22.9K	12.3M/sec
Intel Ivy Bridge (2012)	20.7K	11.7M/sec
Intel Sandy Bridge (2011)	16.1K	11.6M/sec
AMD Piledriver (2012)	59	6.1M/sec

- *A real reliability & security issue*
- *In a more controlled environment, we can induce as many as ten million disturbance errors*

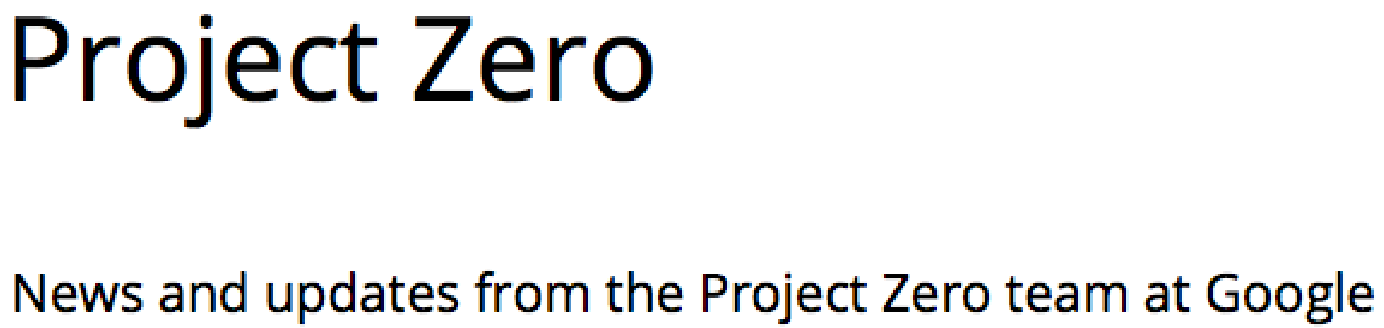
Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

Security Implications



Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

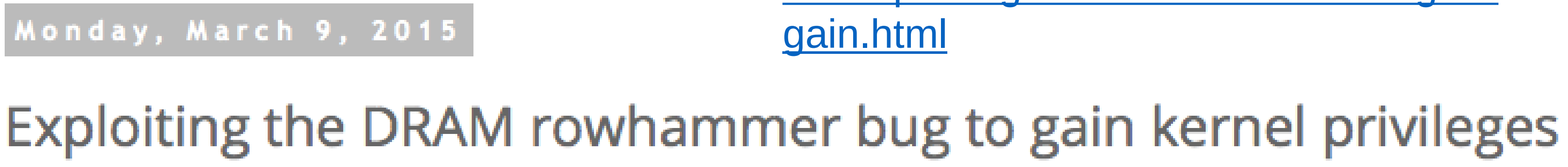
Abstract. Memory isolation is a key property of a reliable and secure computing system — an access to one memory address should not have unintended side effects on data stored in other addresses. However, as DRAM process technology



Project Zero

News and updates from the Project Zero team at Google

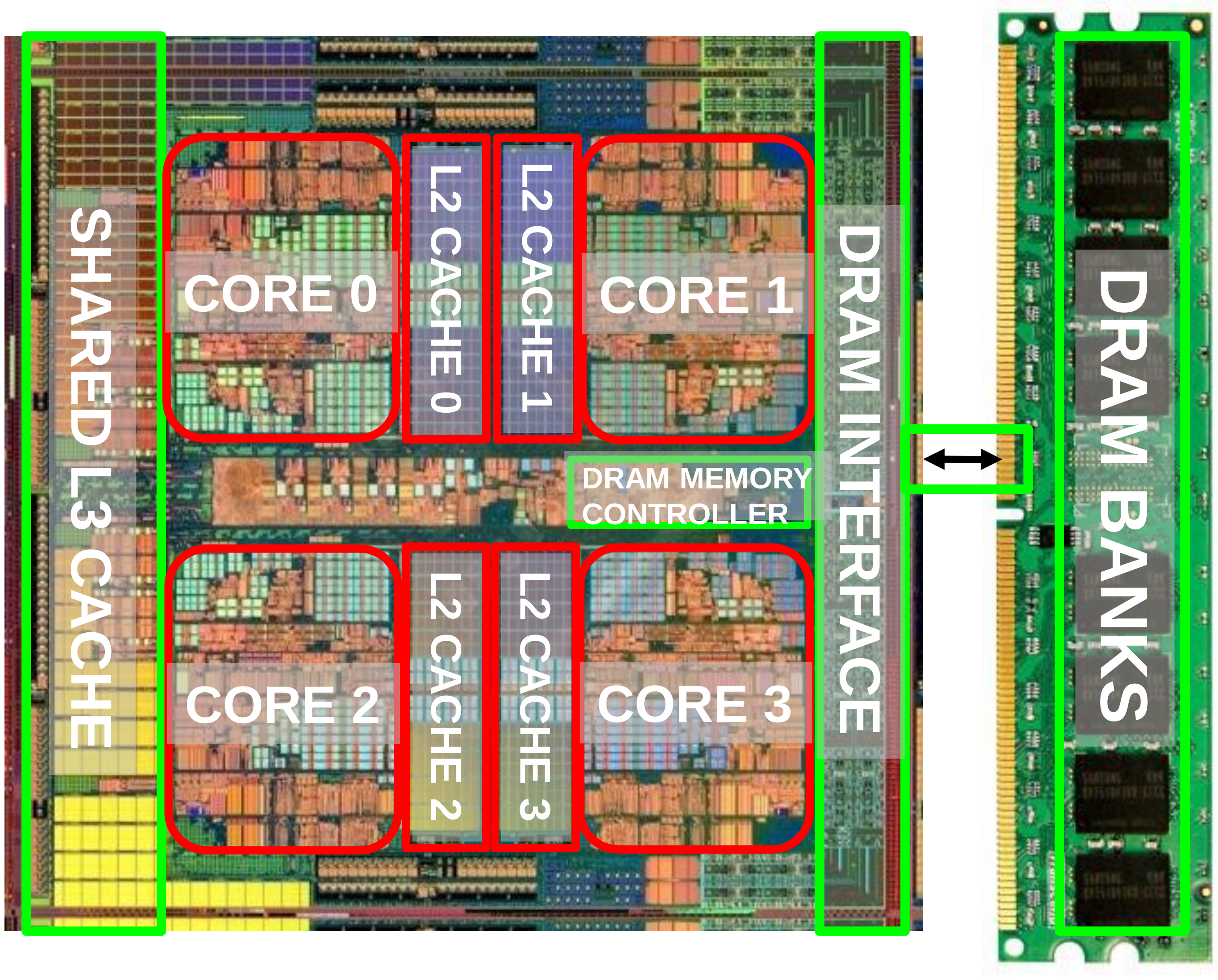
http://googleprojectzero.blogspot.com/2015/03/exploiting-dram-rowhammer-bug-to-gain.html



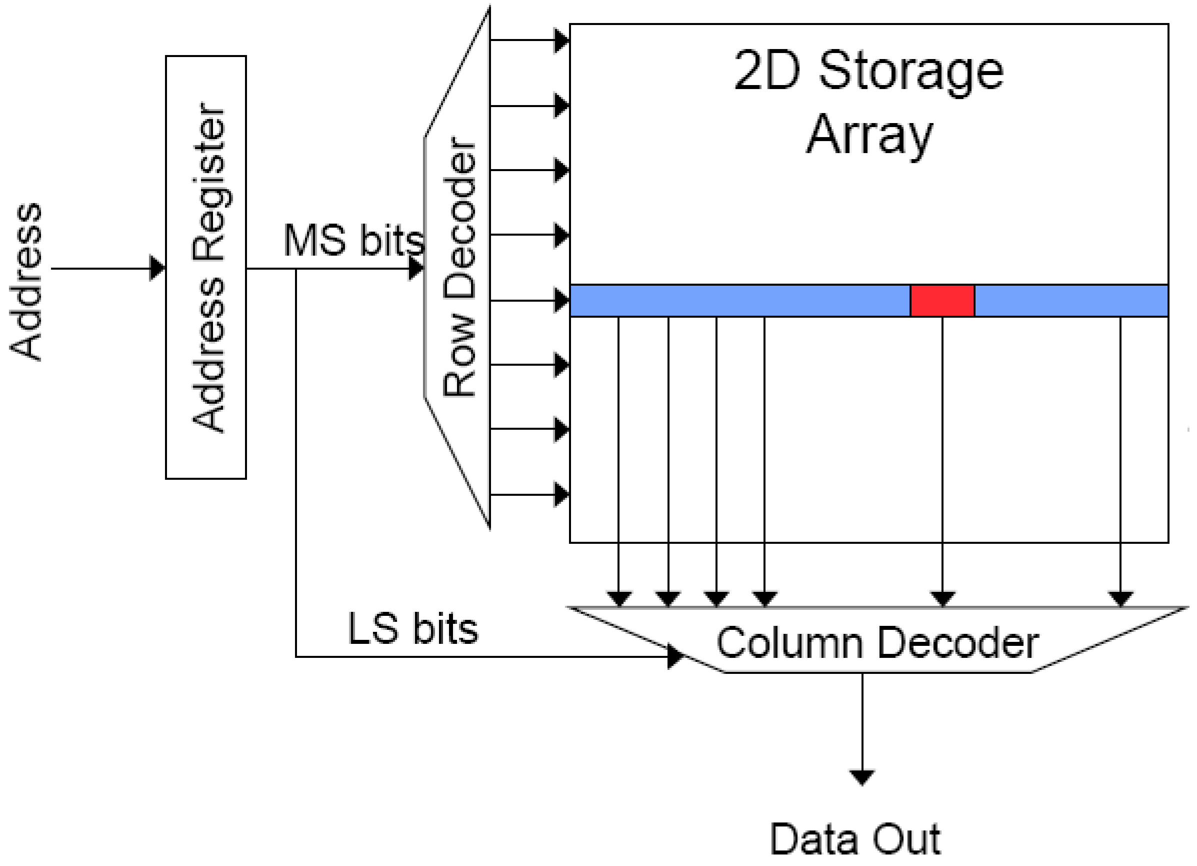
Monday, March 9, 2015

Exploiting the DRAM rowhammer bug to gain kernel privileges

Main Memory in the System



Memory Bank 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

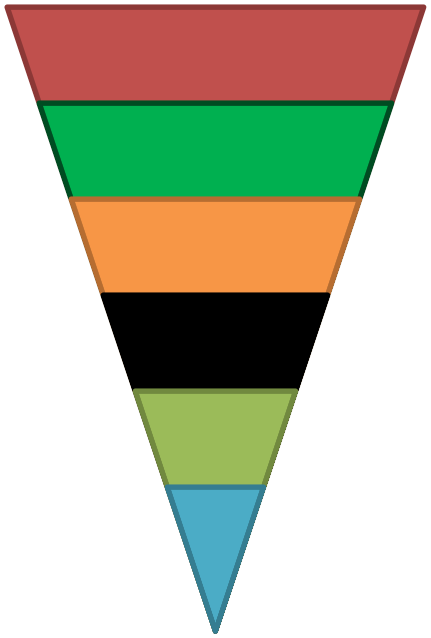
  5. Precharge bit-lines
     - For next access

DRAM vs. SRAM

- DRAM
  - Slower access (capacitor)
  - Higher density (1T 1C cell)
  - Lower cost
  - Requires refresh (power, performance, circuitry)
  - Manufacturing requires putting capacitor and logic together
- SRAM
  - Faster access (no capacitor)
  - Lower density (6T cell)
  - Higher cost
  - No need for refresh
  - Manufacturing compatible with logic process (no capacitor)

DRAM Subsystem Organization

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



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 an activate command

DRAM Bank Operation

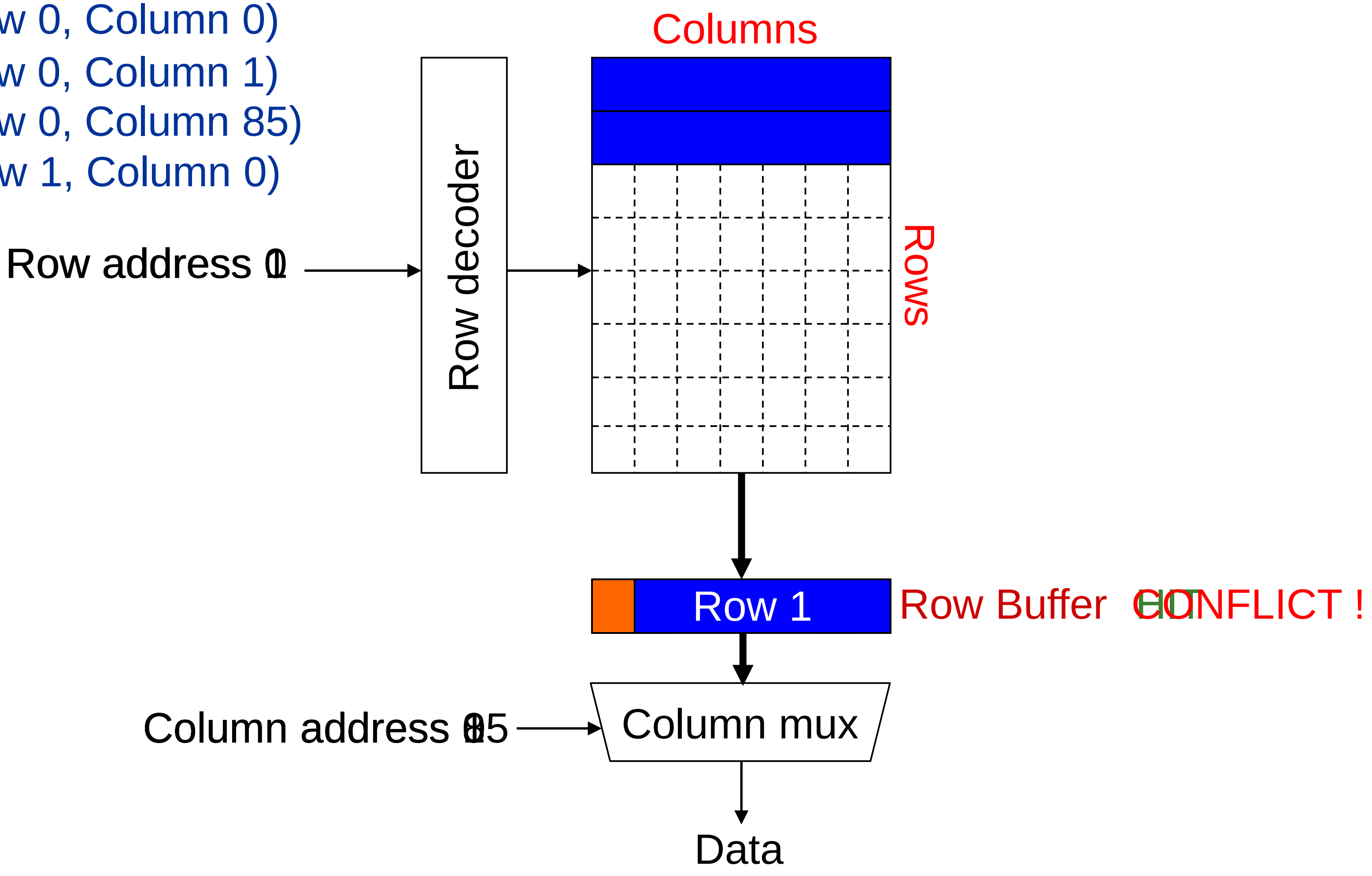
Access Address:

(Row 0, Column 0)

(Row 0, Column 1)

(Row 0, Column 85)

(Row 1, Column 0)

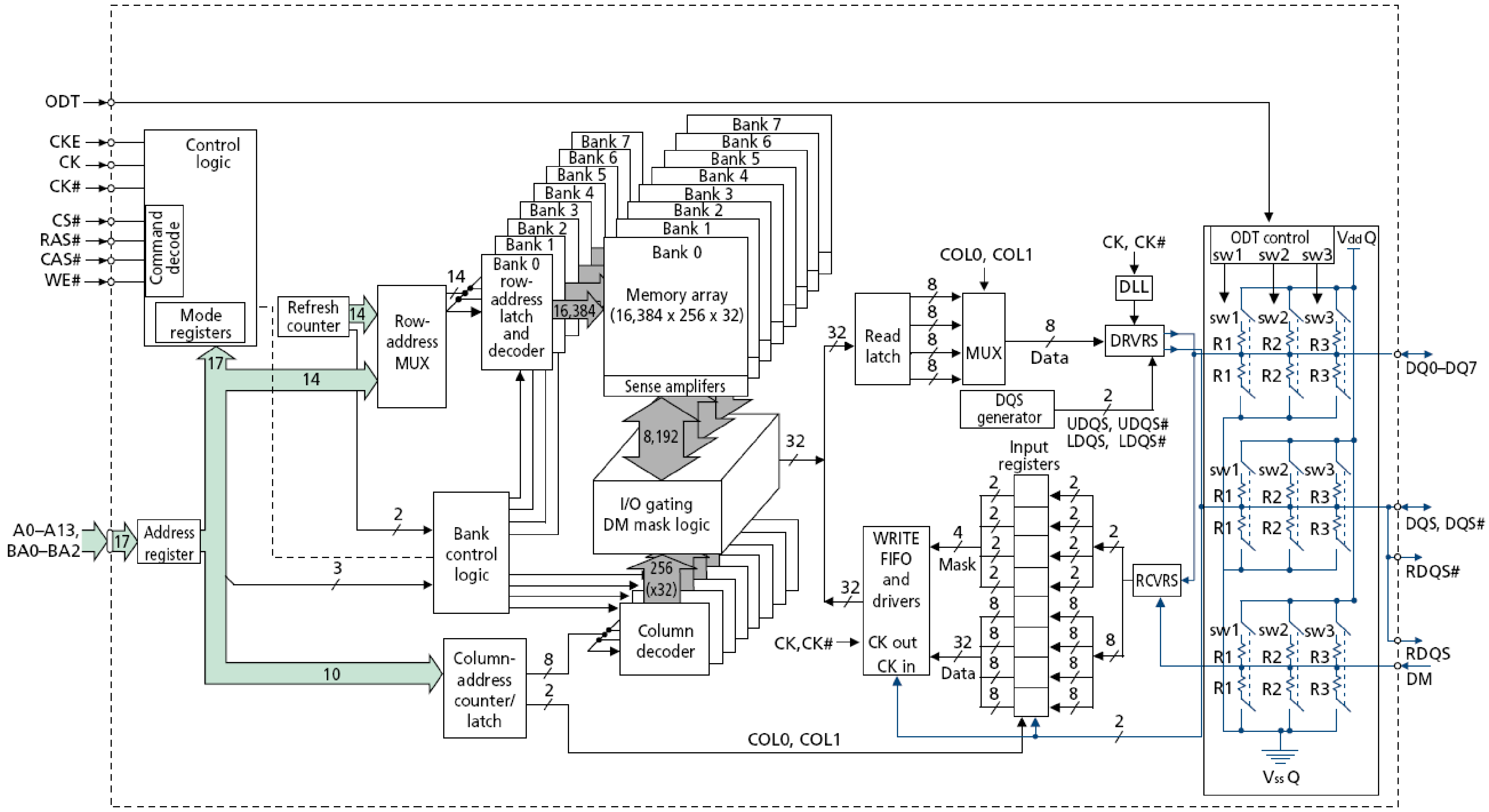


The DRAM Chip

- Consists of multiple banks (8 is a common number today)
- Banks share command/address/data buses
- The chip itself has a narrow interface (4-16 bits per read)

- Changing the number of banks, size of the interface (pins), whether or not command/address/data buses are shared has significant impact on DRAM system cost

128M x 8-bit DRAM Chip



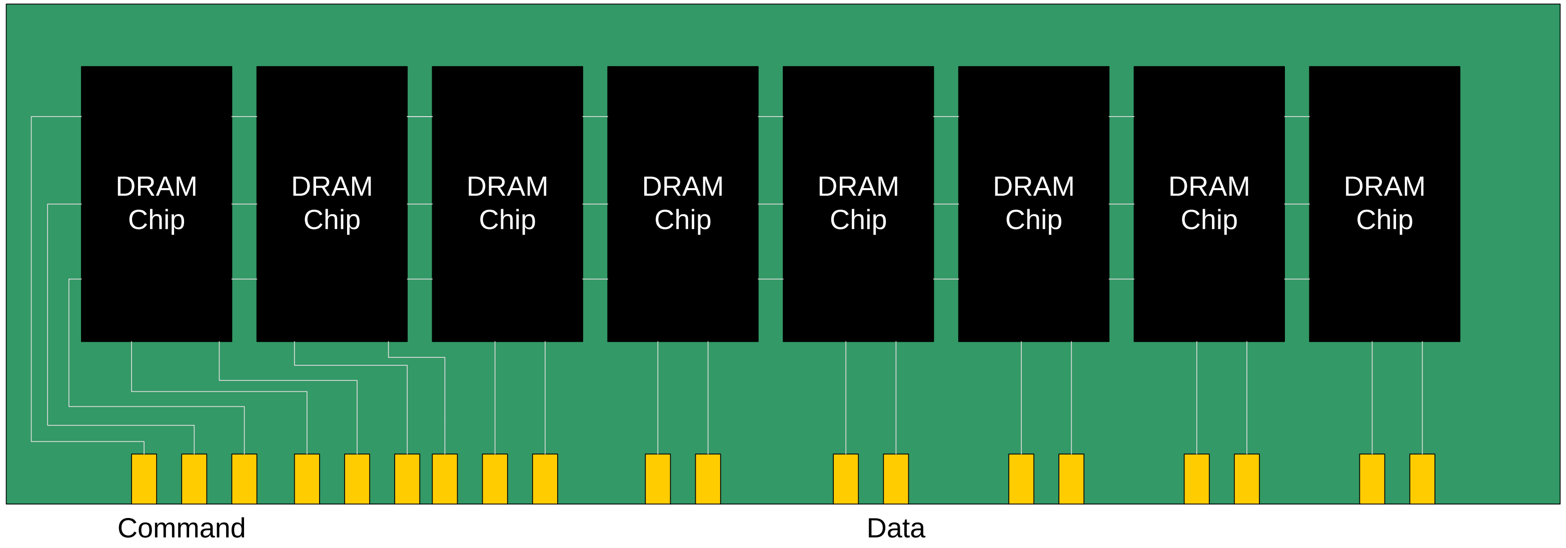
DRAM Rank and Module

- 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**

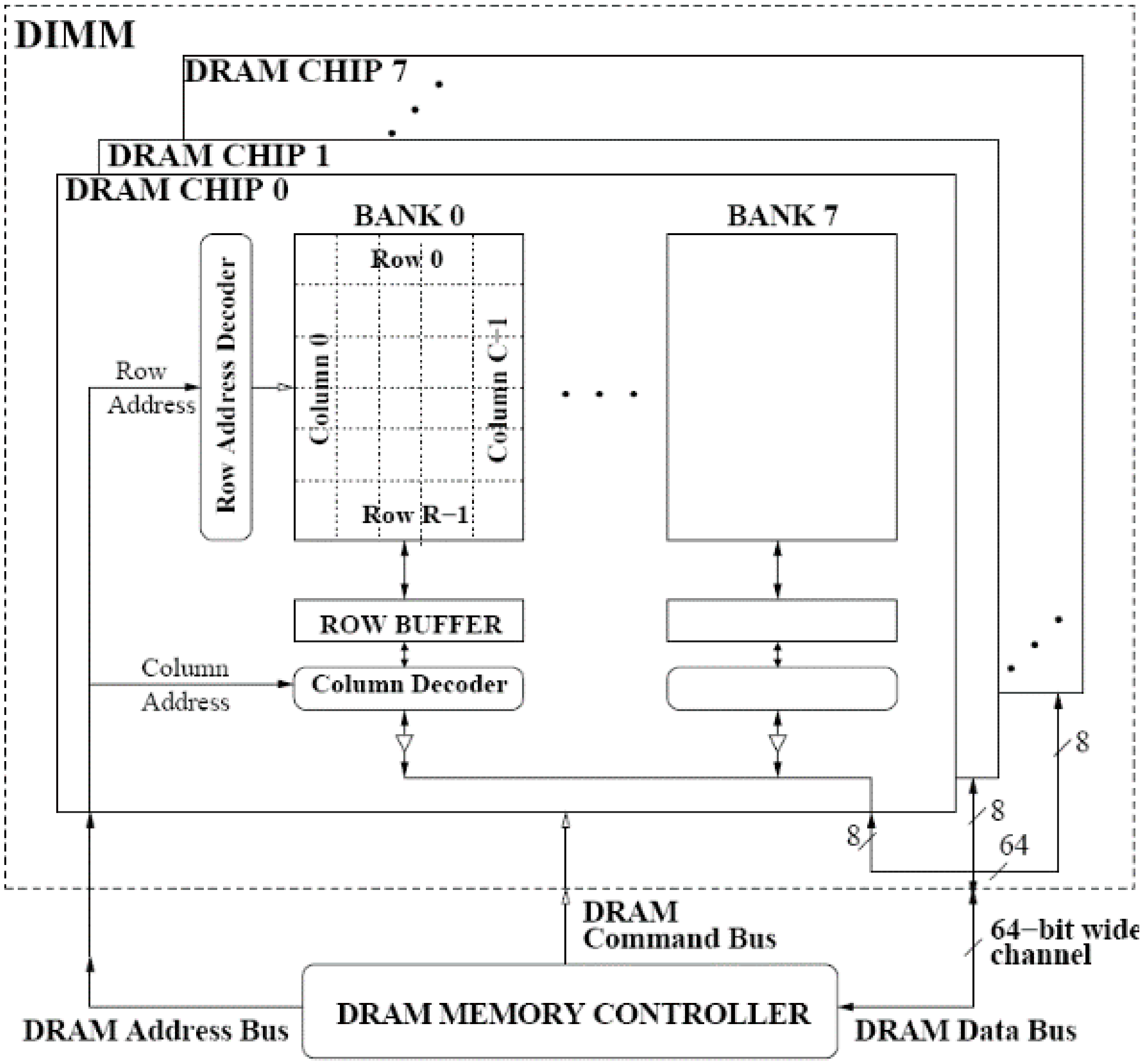
- A DRAM module consists of one or more ranks
  - E.g., DIMM (dual inline memory module)
  - This is what you plug into your motherboard

- 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)

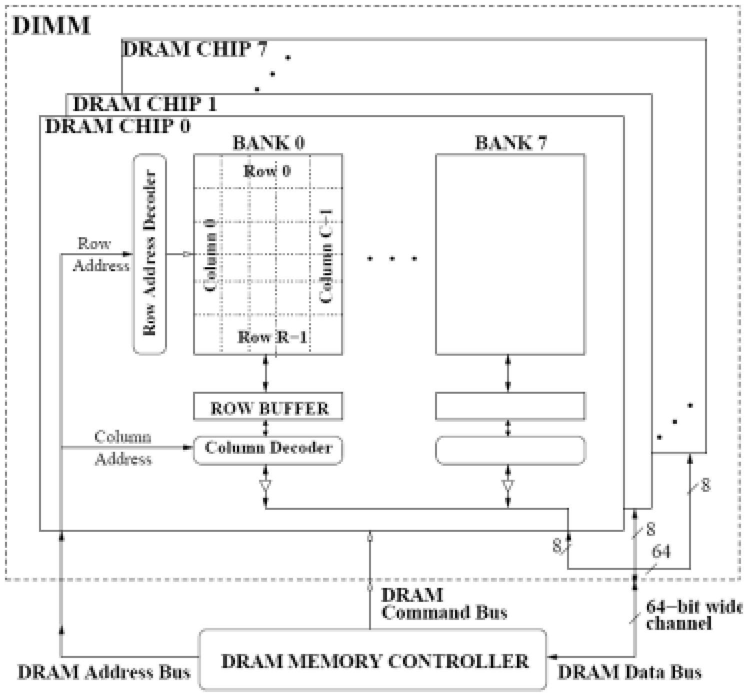


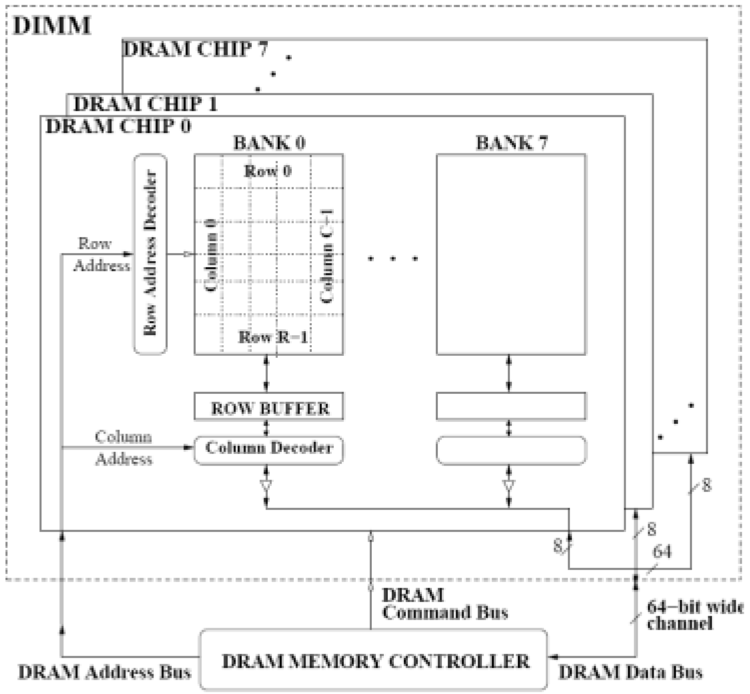
A 64-bit Wide DIMM (One Rank)



- 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

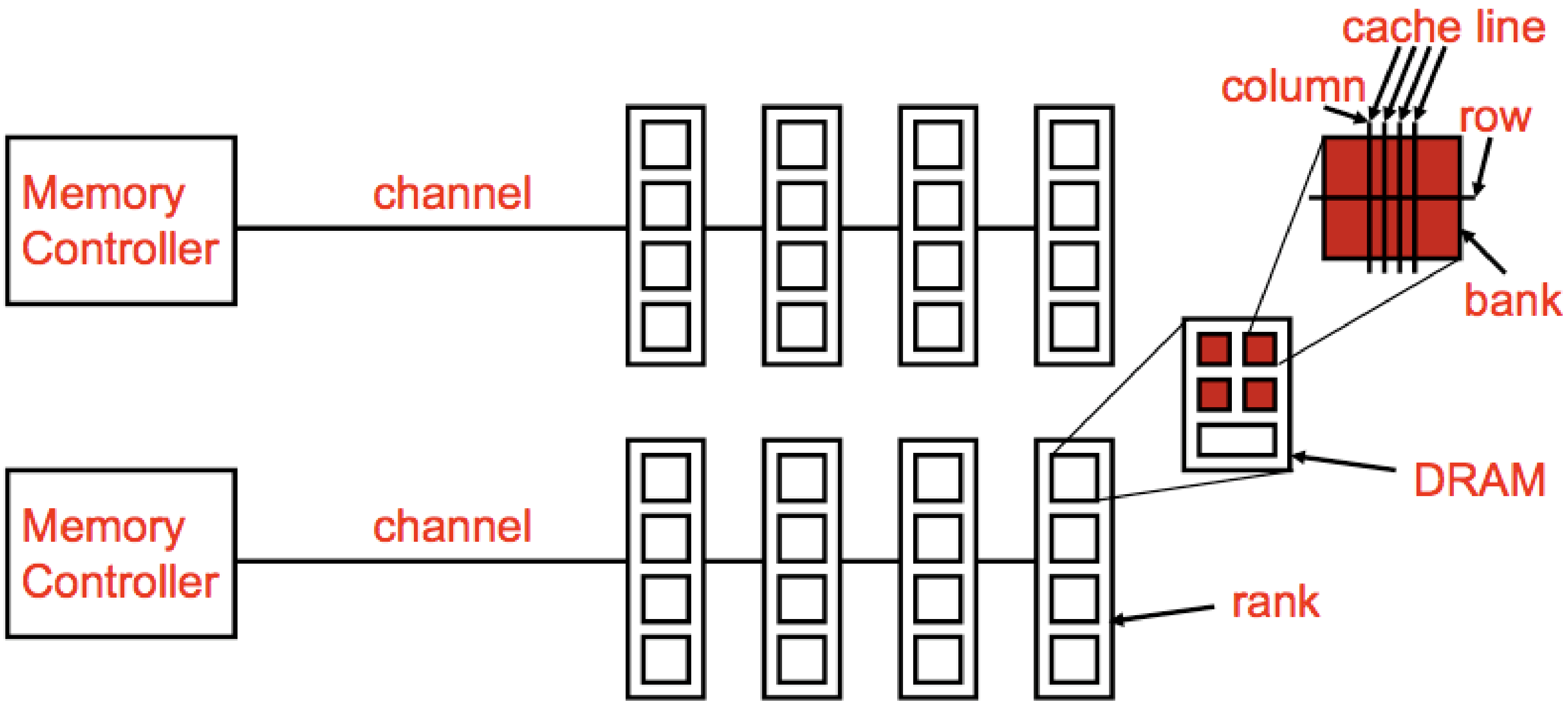
DRAM Channels



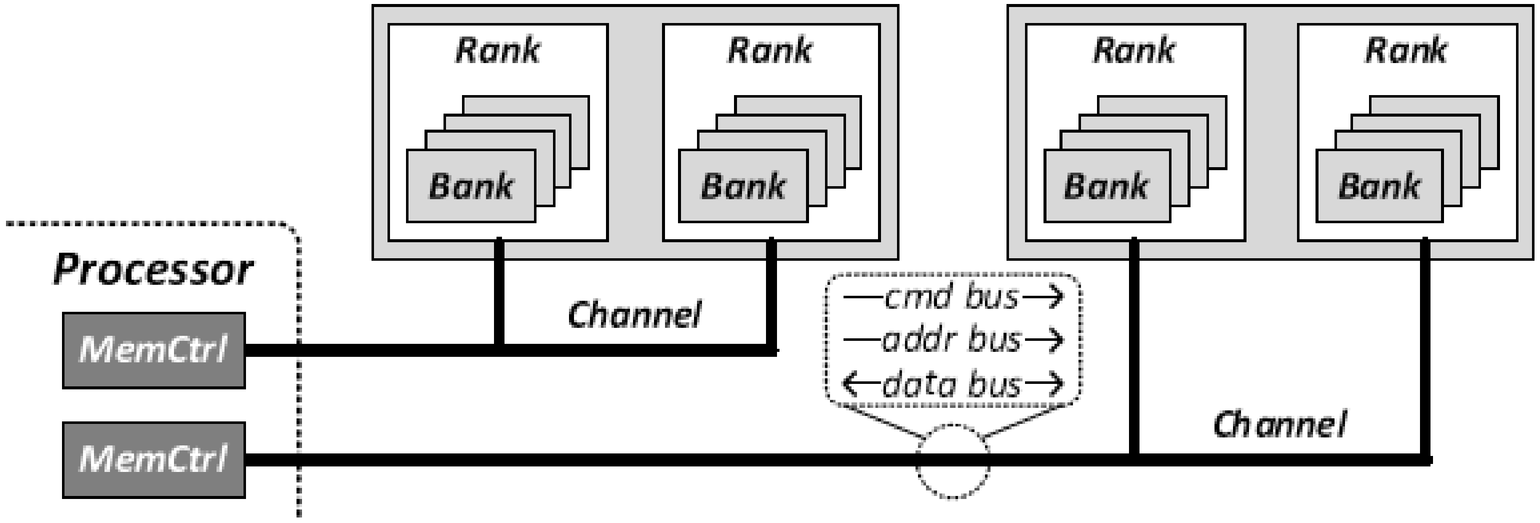


- 2 Independent Channels: 2 Memory Controllers (Above)
- 2 Dependent/Lockstep Channels: 1 Memory Controller with wide interface (Not Shown)

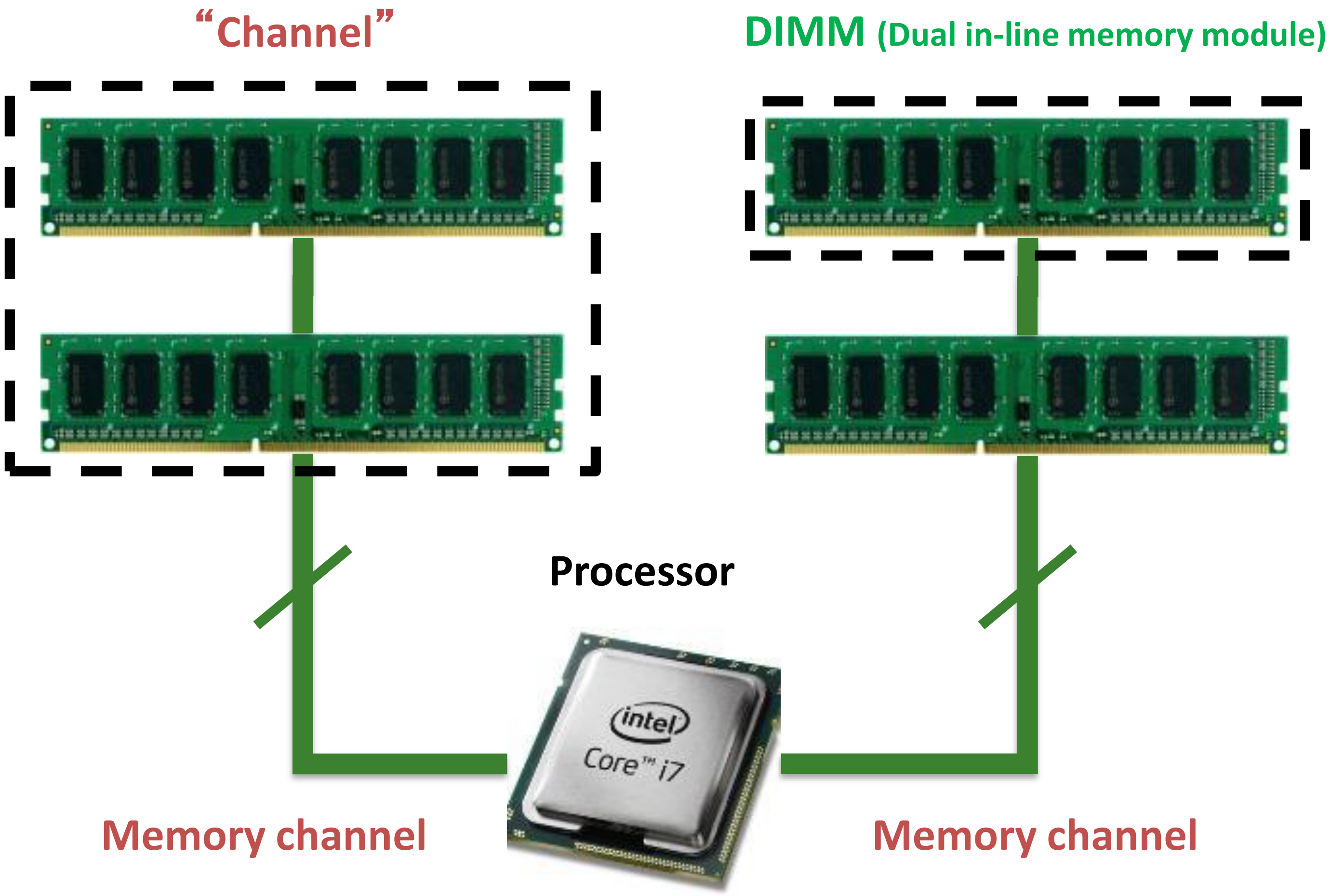
Generalized Memory Structure



Generalized Memory Structure



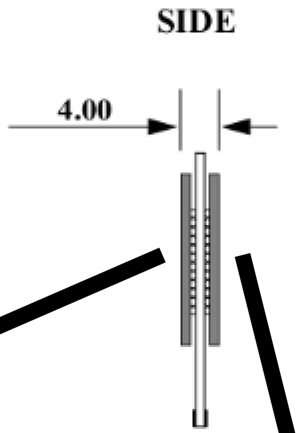
The DRAM subsystem



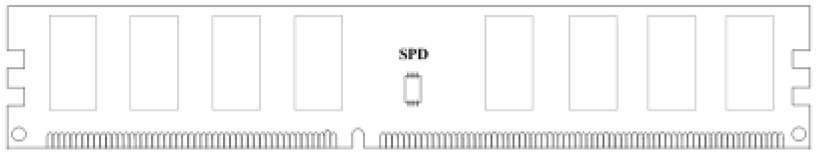
Breaking down a DIMM

DIMM (Dual in-line memory module)

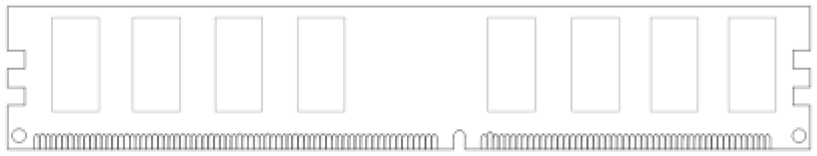
Side view



Front of DIMM



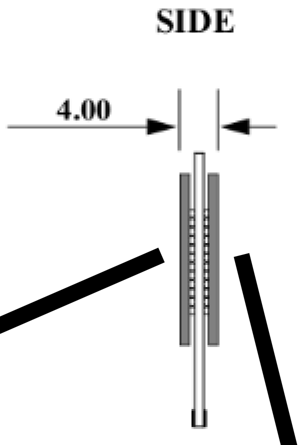
Back of DIMM



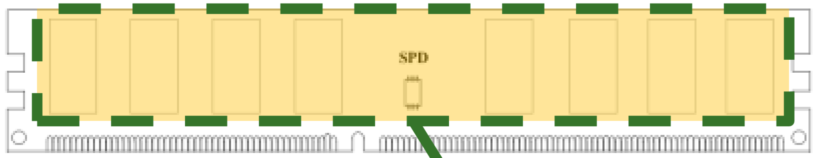
Breaking down a DIMM

DIMM **(Dual in-line memory module)**

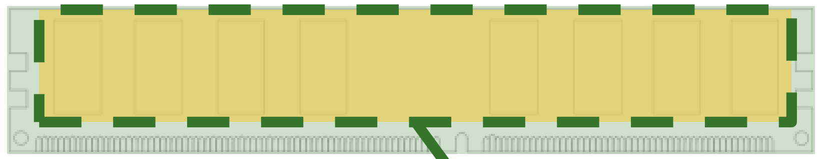
Side view



Front of DIMM



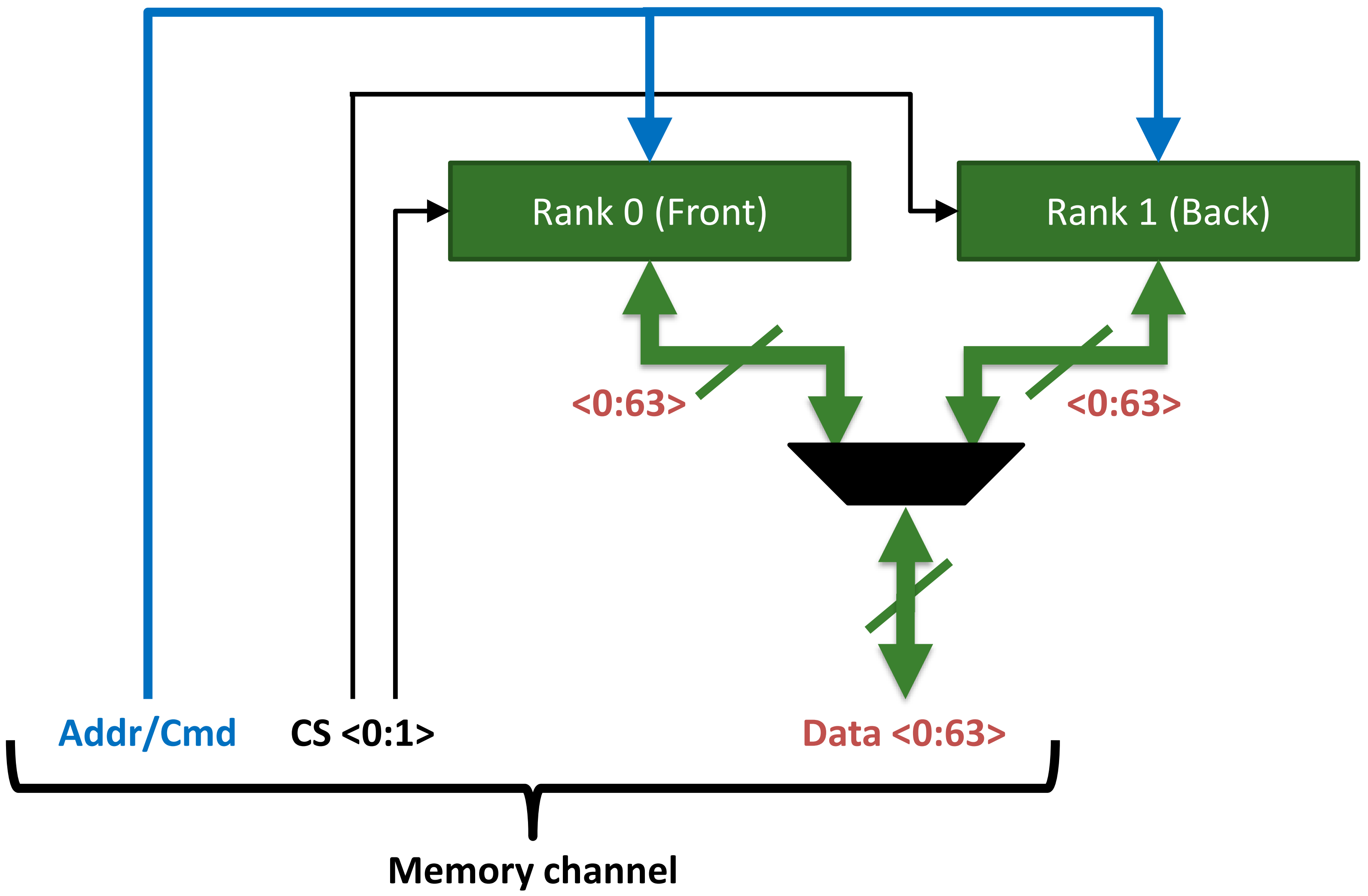
Back of DIMM



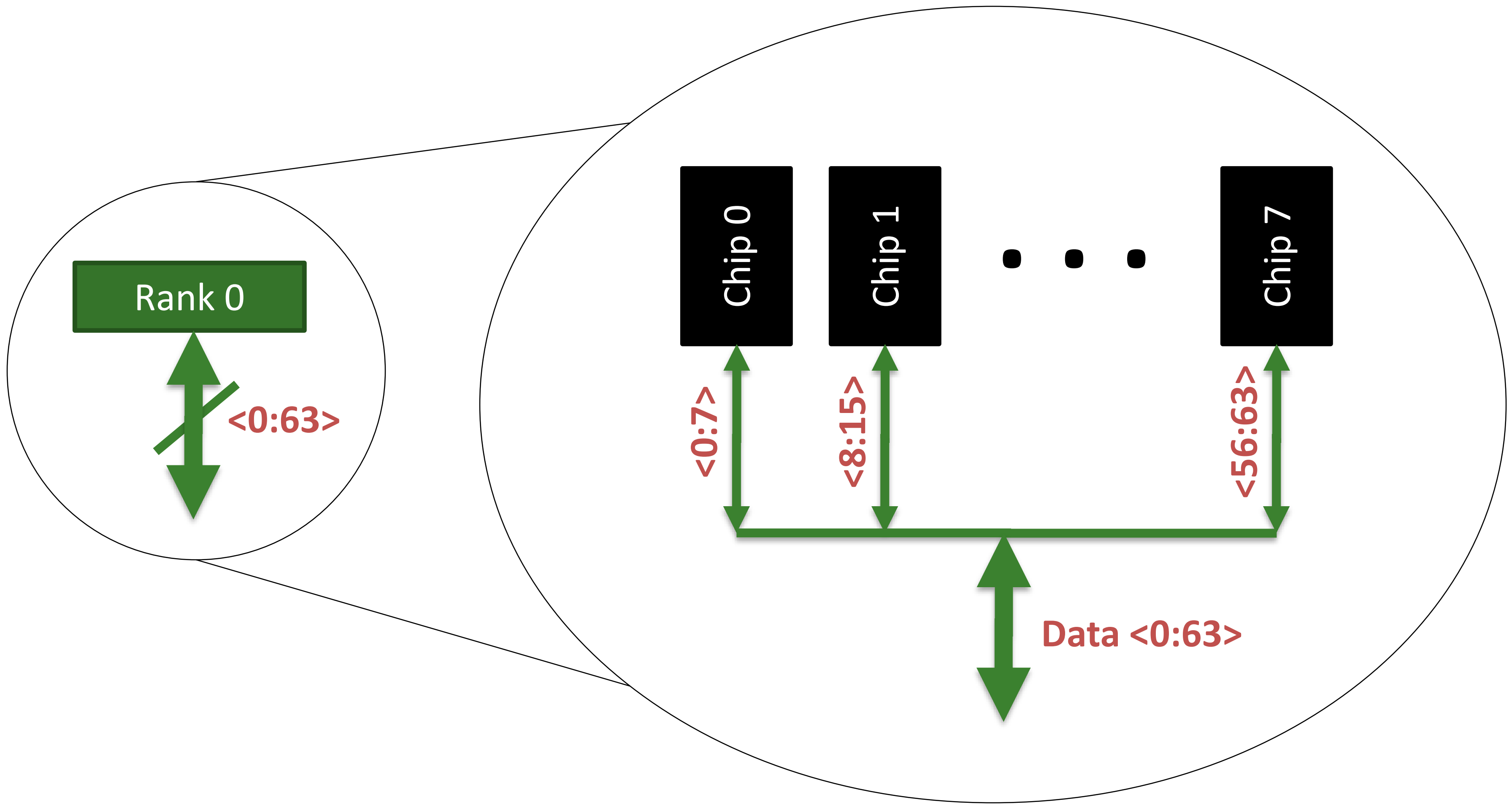
Rank 0: collection of 8 chips

Rank 1

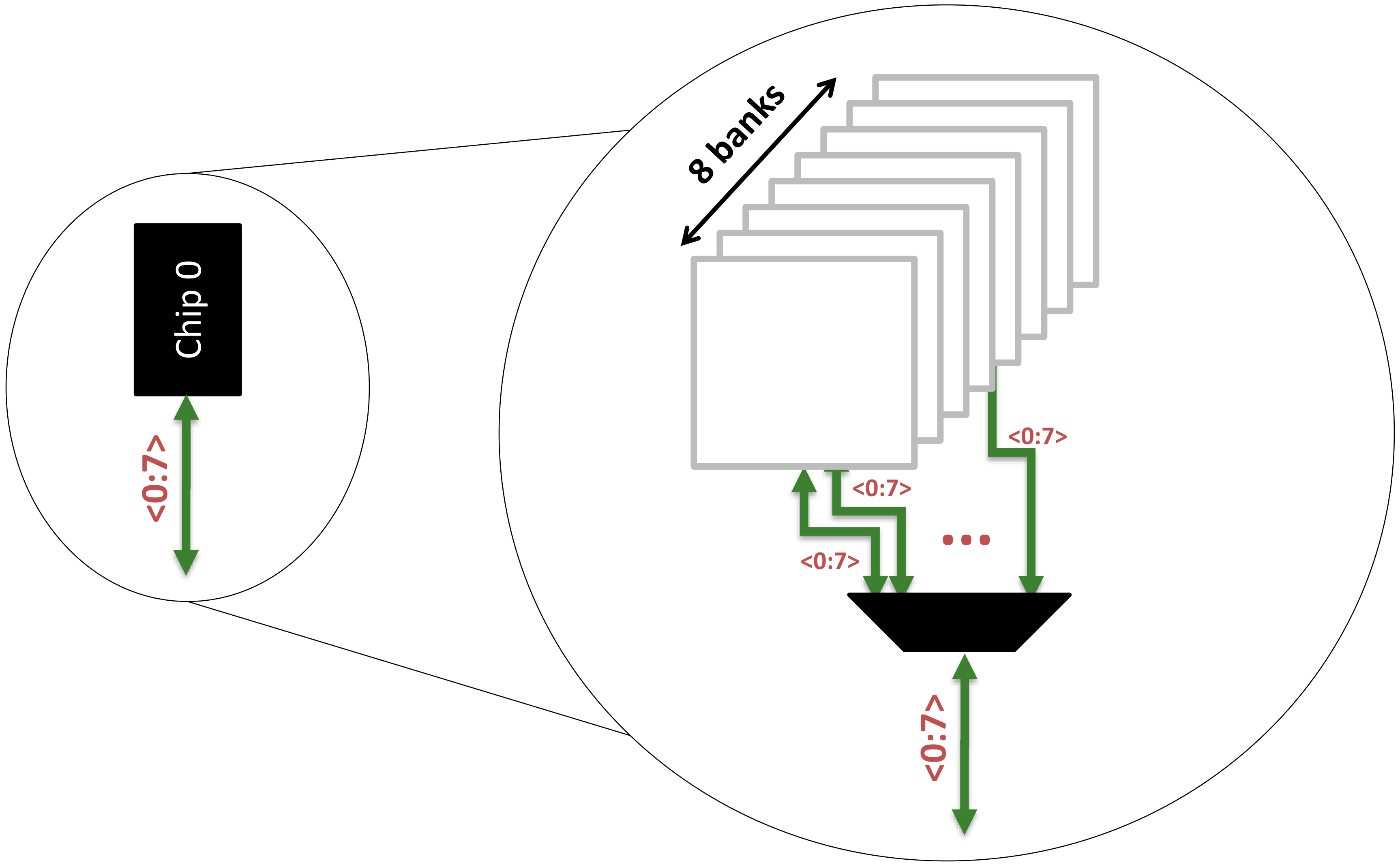
Rank


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

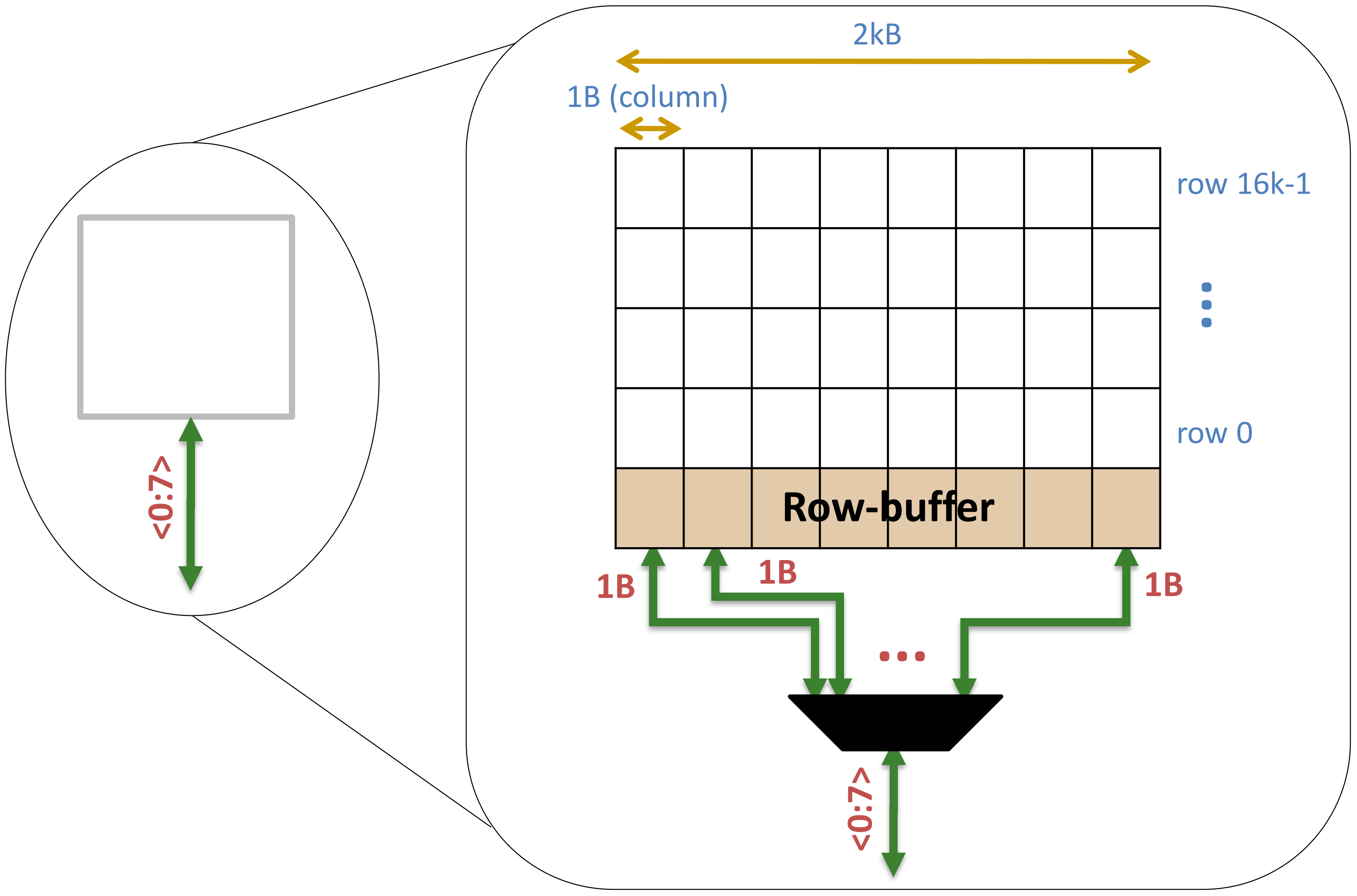
Breaking down a Rank



Breaking down a Chip

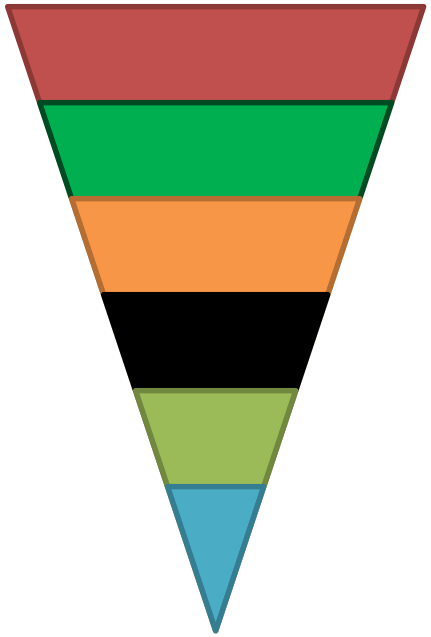


Breaking down a Bank

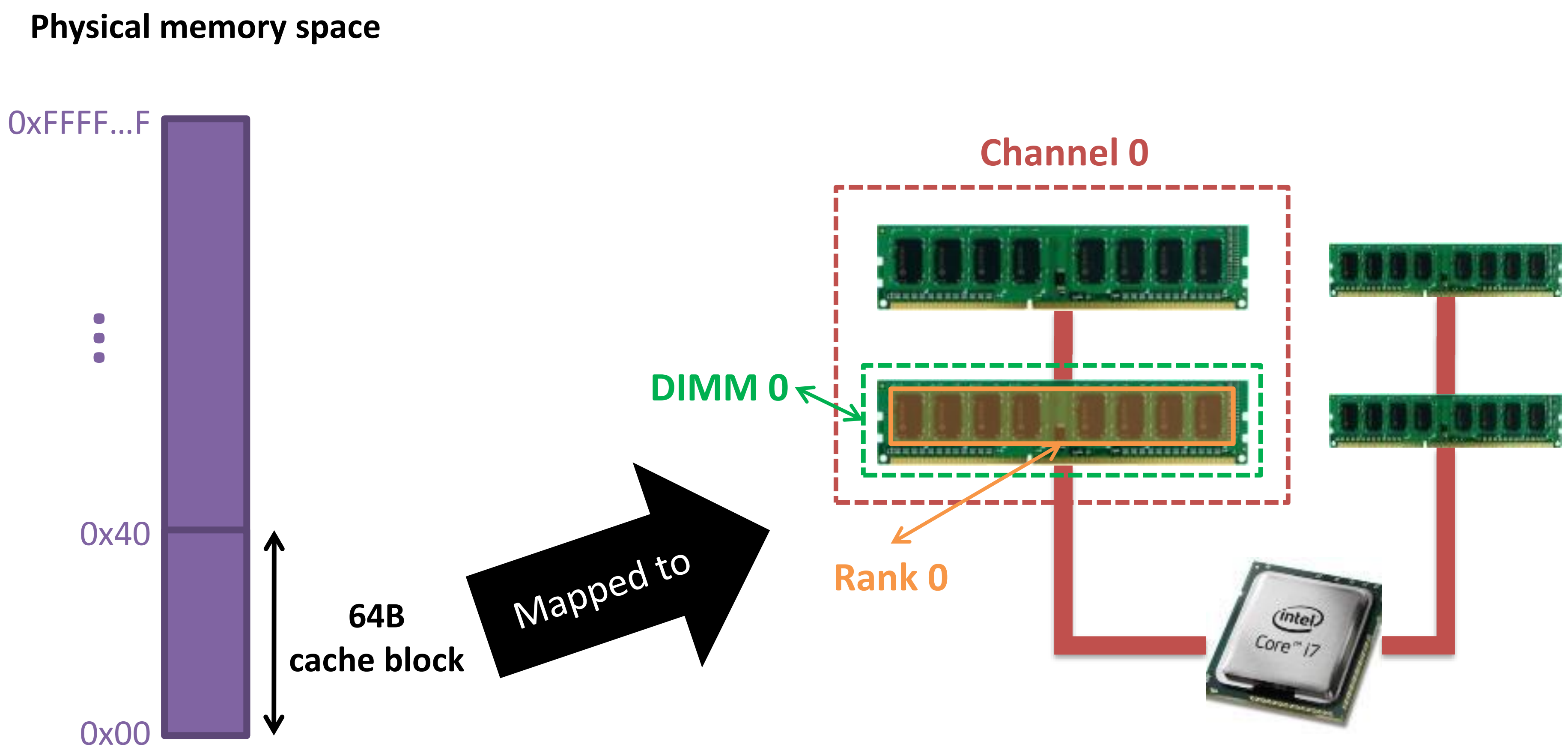


DRAM Subsystem Organization

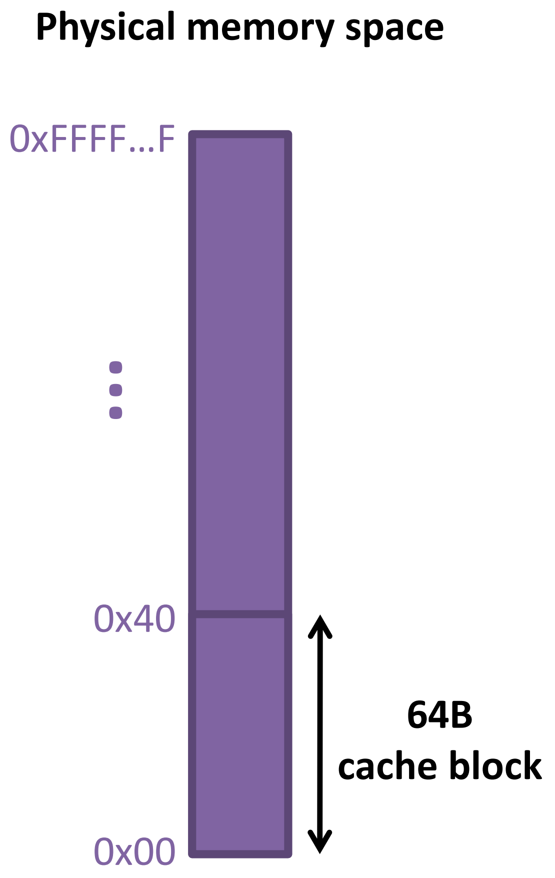
- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column
- Cell

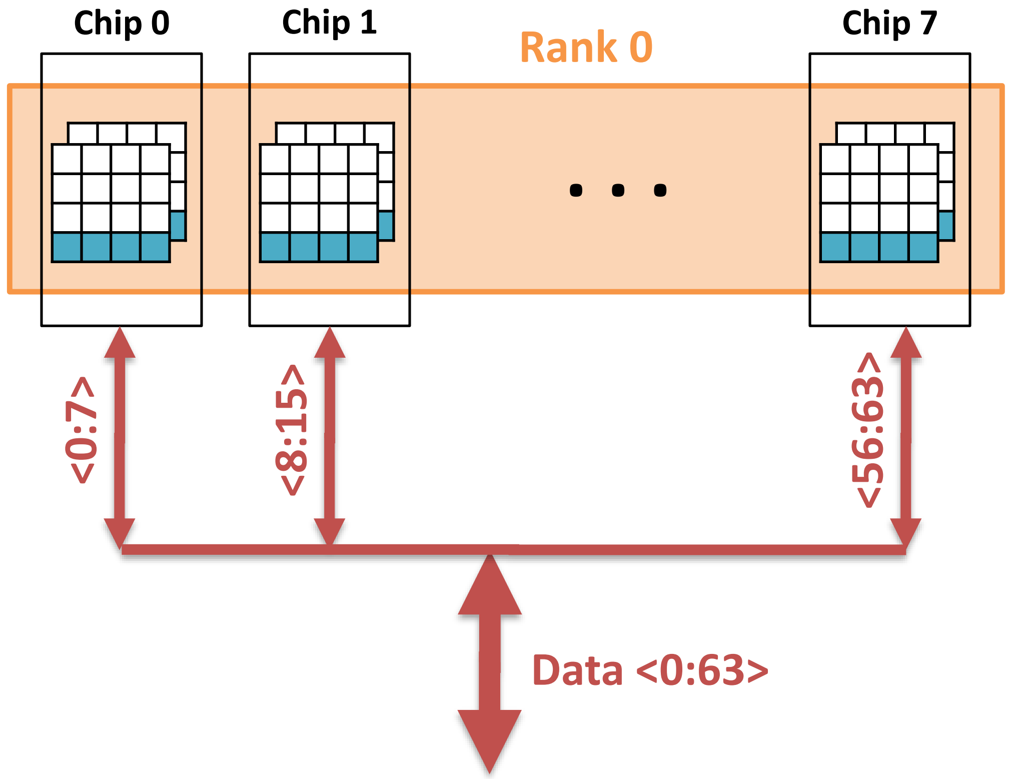


Example: Transferring a cache block

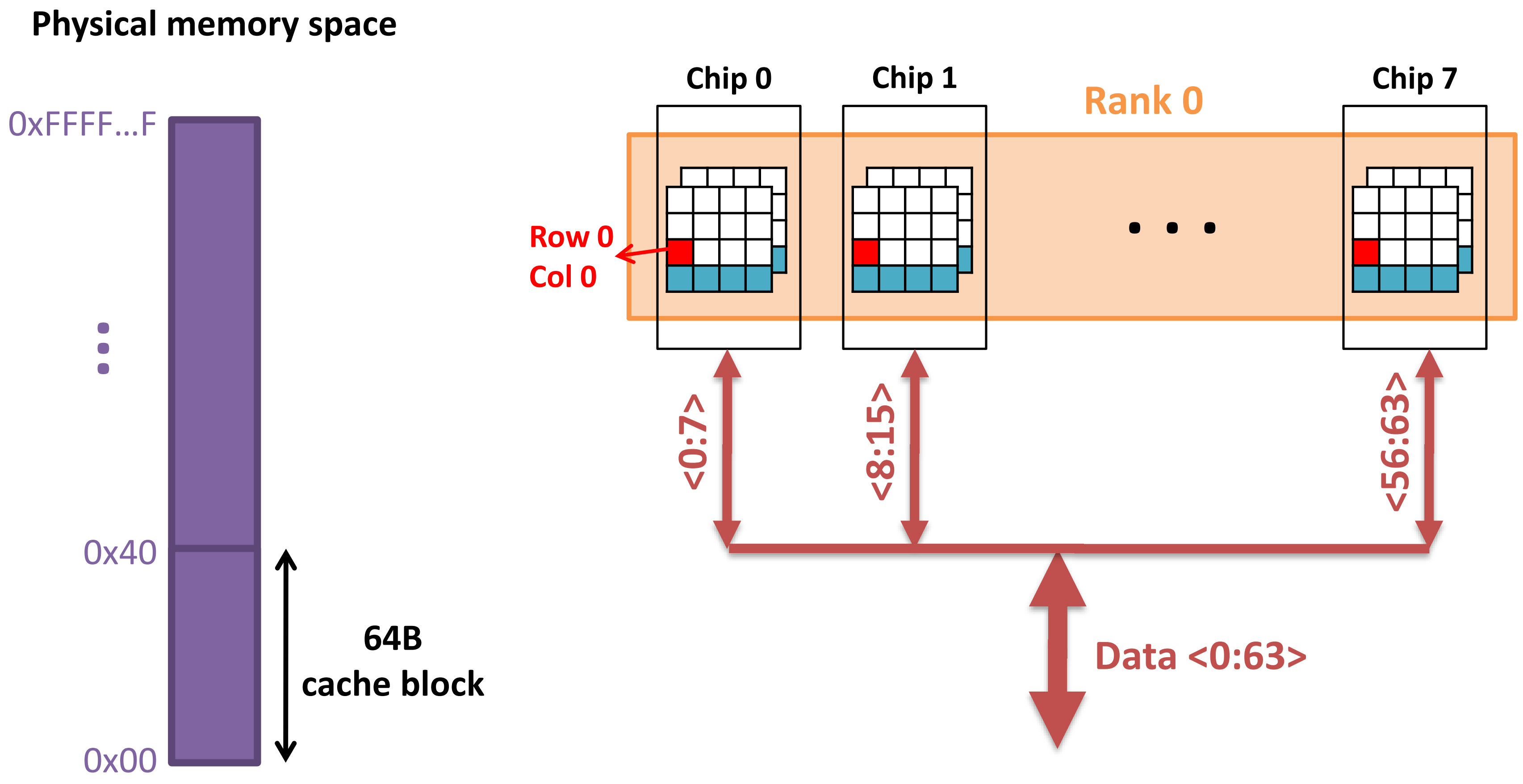


Example: Transferring a cache block

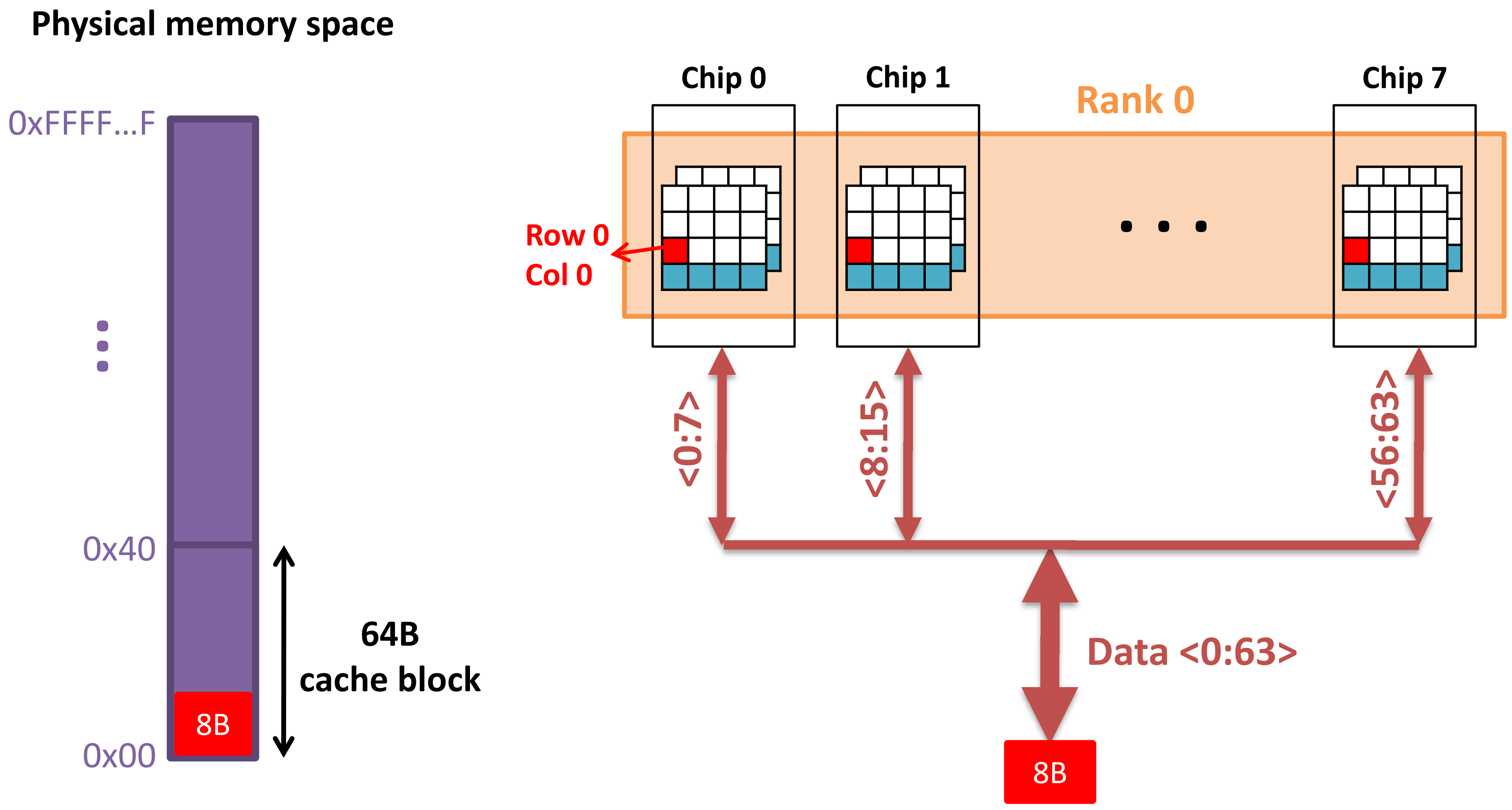




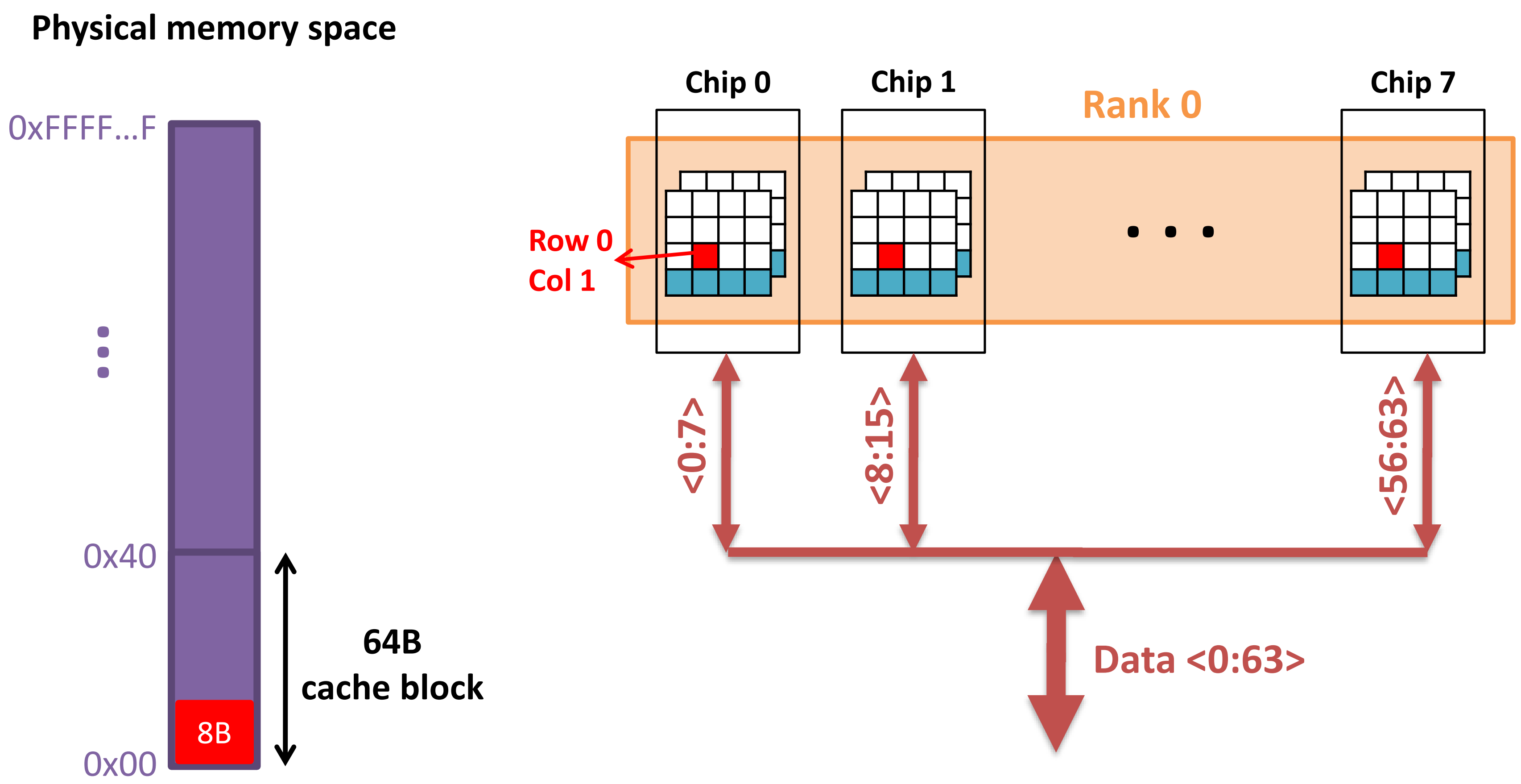
Example: Transferring a cache block



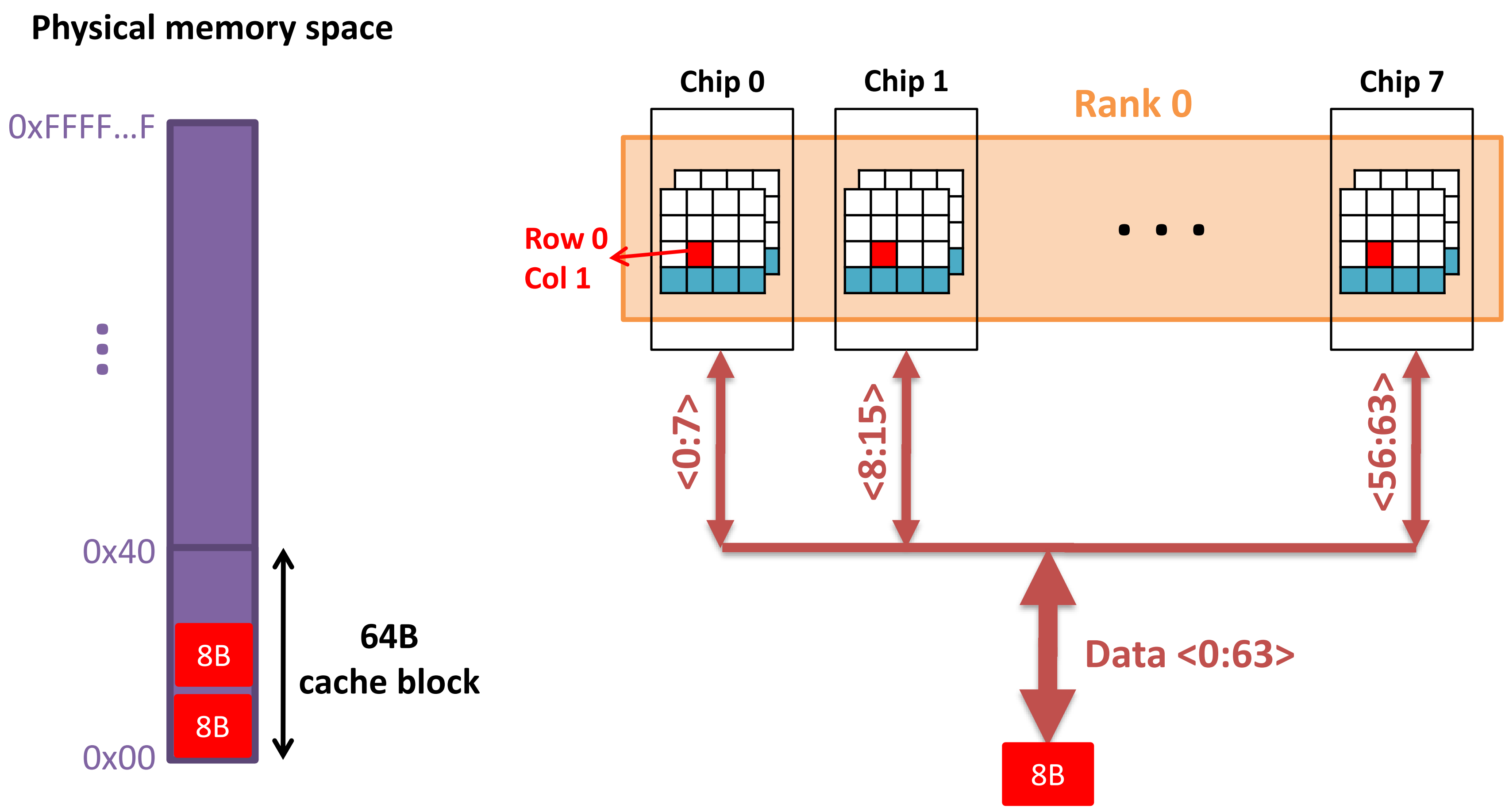
Example: Transferring a cache block



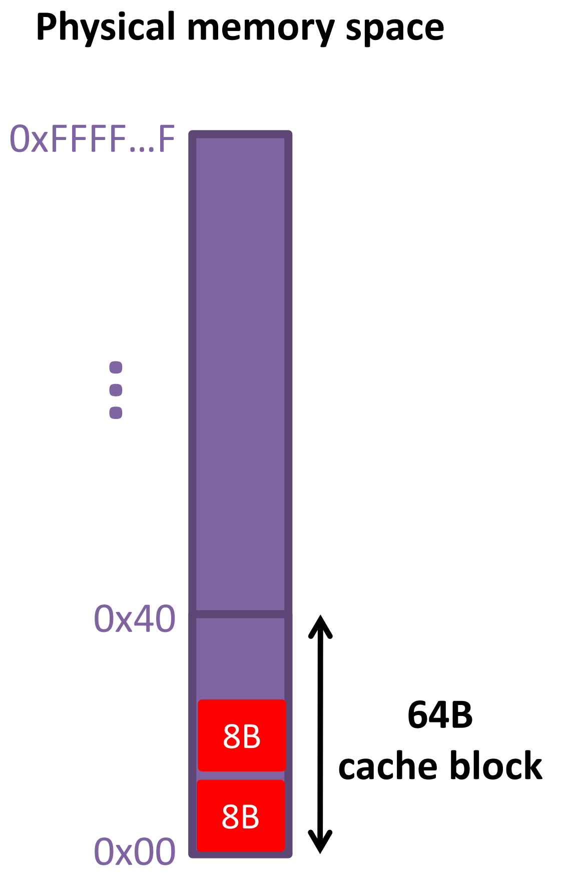
Example: Transferring a cache block

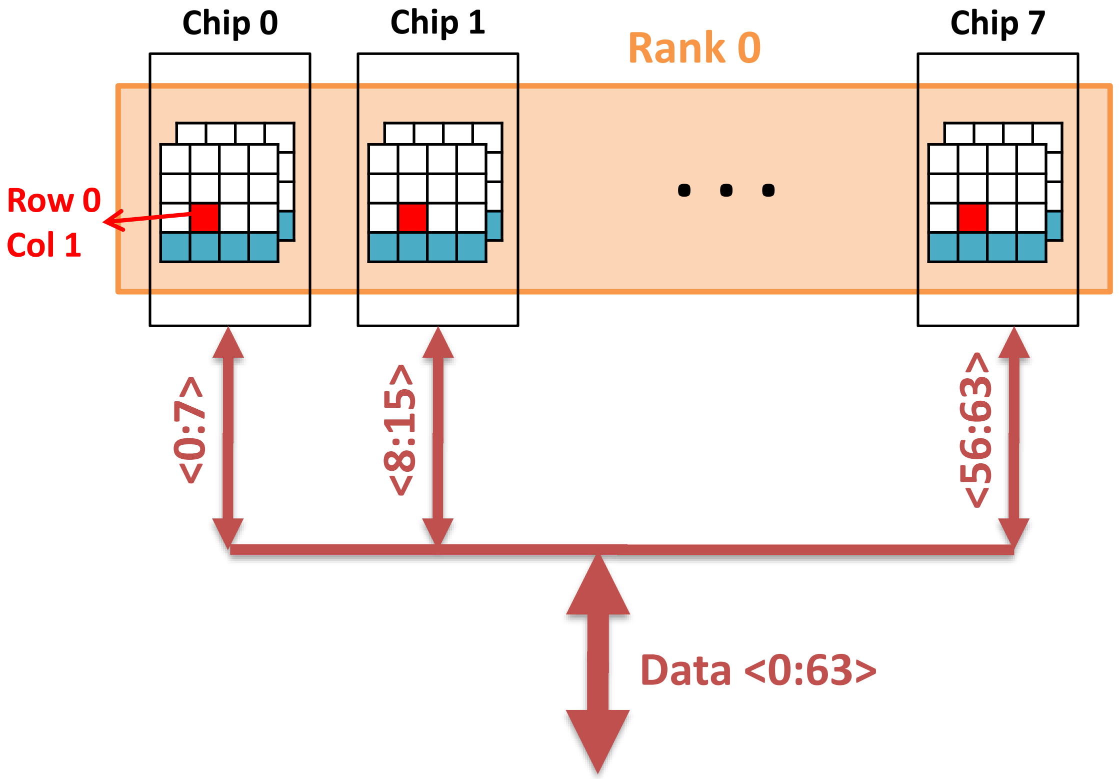


Example: Transferring a cache block



Example: Transferring a cache block





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

During the process, 8 columns are read sequentially.

Next Topic

Virtual Memory