15-740/18-740
Computer Architecture
Lecture 19: Main Memory

Prof. Onur Mutlu

Carnegie Mellon University

Last Time

- Multi-core issues in caching
  - OS-based cache partitioning (using page coloring)
  - Handling shared data in caches
  - Non-uniform cache access
  - Caches as bandwidth filters
  - Revisiting cache insertion policies
    - Dynamic insertion policy

Today

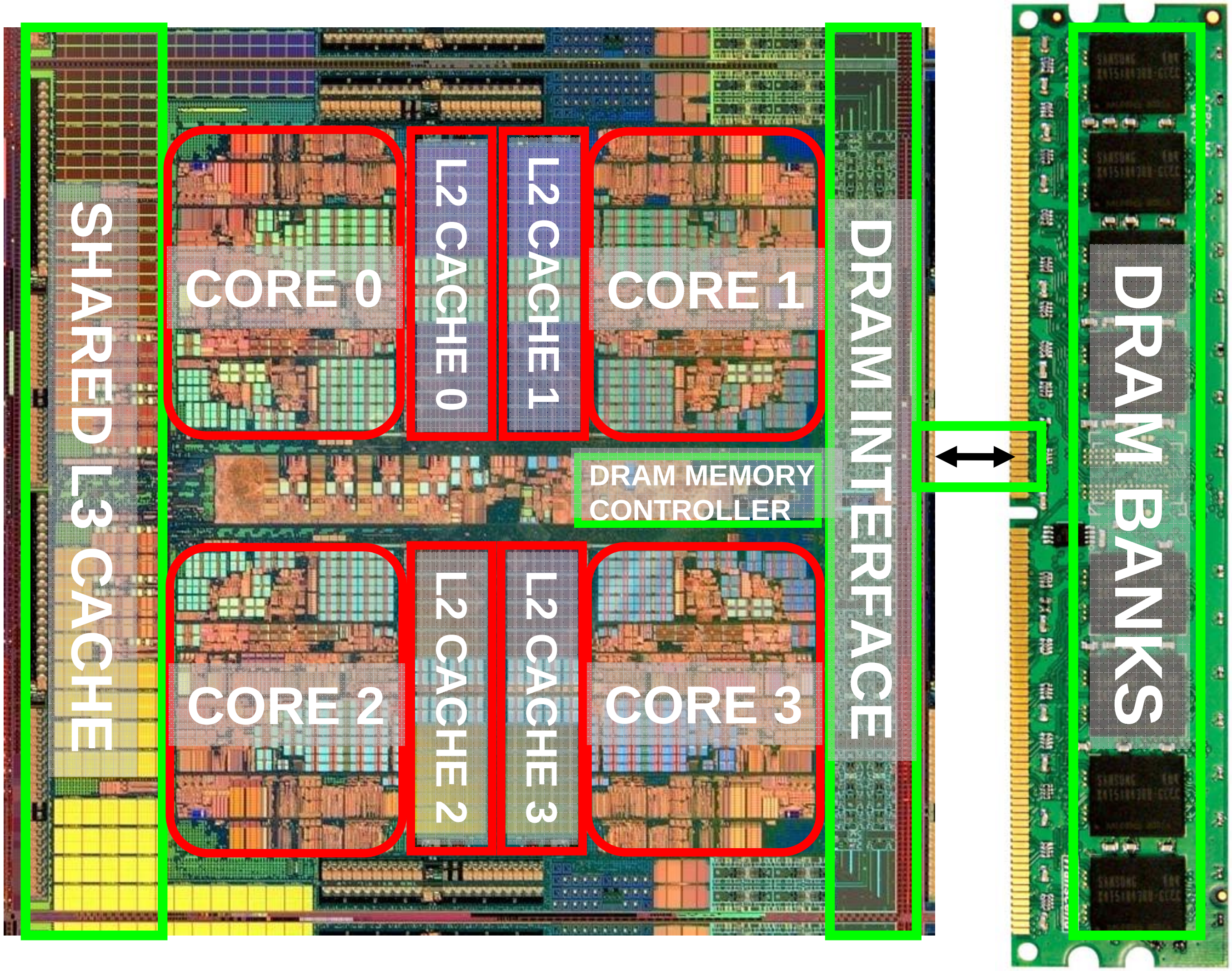
- SRAM vs. DRAM
- Interleaving/Banking
- DRAM Microarchitecture
  - Memory controller
  - Memory buses
  - Banks, ranks, channels, DIMMs
  - Address mapping: software vs. hardware
  - DRAM refresh
- Memory scheduling policies
- Memory power/energy management
- Multi-core issues
  - Fairness, interference
  - Large DRAM capacity

Readings

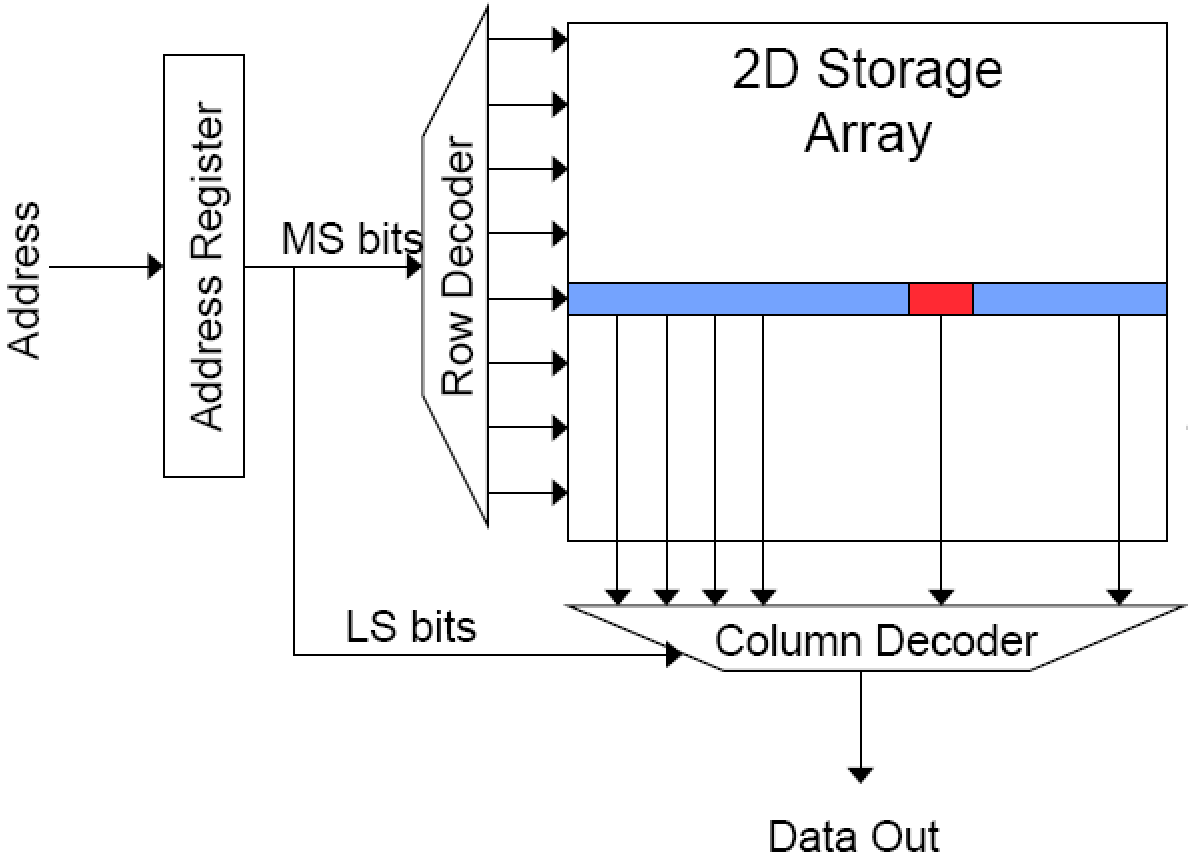
- Required:
  - Mutlu and Moscibroda, "Parallelism-Aware Batch Scheduling: Enabling High-Performance and Fair Memory Controllers," IEEE Micro Top Picks 2009.
  - Mutlu and Moscibroda, "Stall-Time Fair Memory Access Scheduling for Chip Multiprocessors," MICRO 2007.

- Recommended:
  - Zhang et al., "A Permutation-based Page Interleaving Scheme to Reduce Row-buffer Conflicts and Exploit Data Locality," MICRO 2000.
  - Lee et al., "Prefetch-Aware DRAM Controllers," MICRO 2008.
  - Rixner et al., "Memory Access Scheduling," ISCA 2000.

Main Memory in the System



Memory Bank Organization



- Read access sequence:

1. Decode row address & drive word-lines

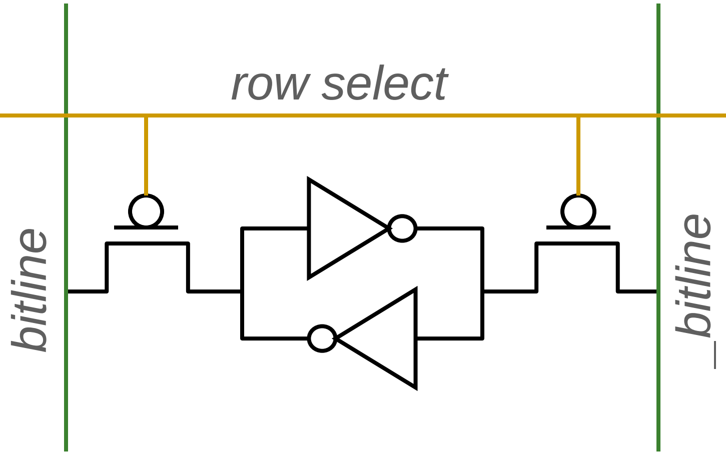
2. Selected bits drive bit-lines
   - Entire row read

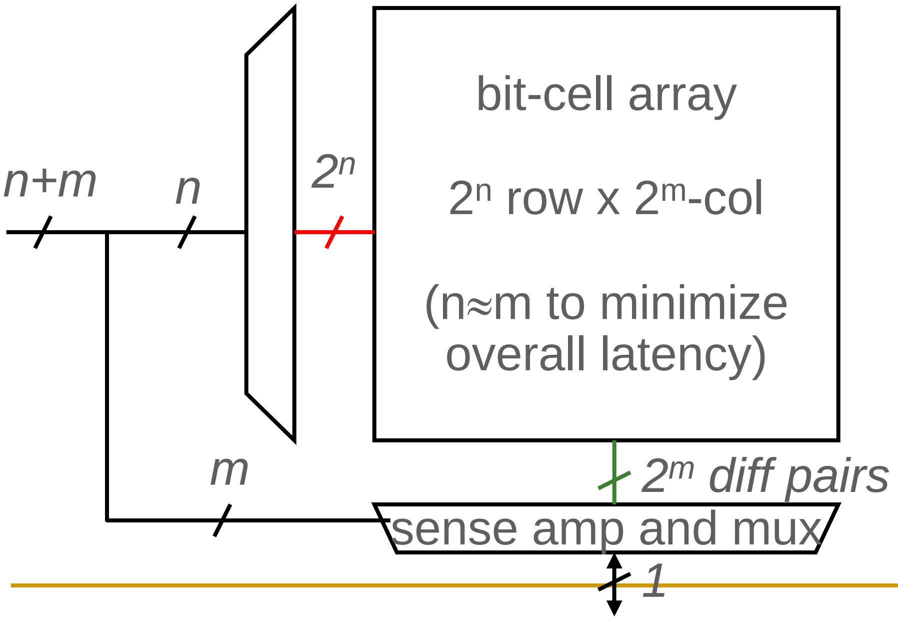
3. Amplify row data

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

5. Precharge bit-lines
   - For next access

SRAM (Static Random Access Memory)





Read Sequence

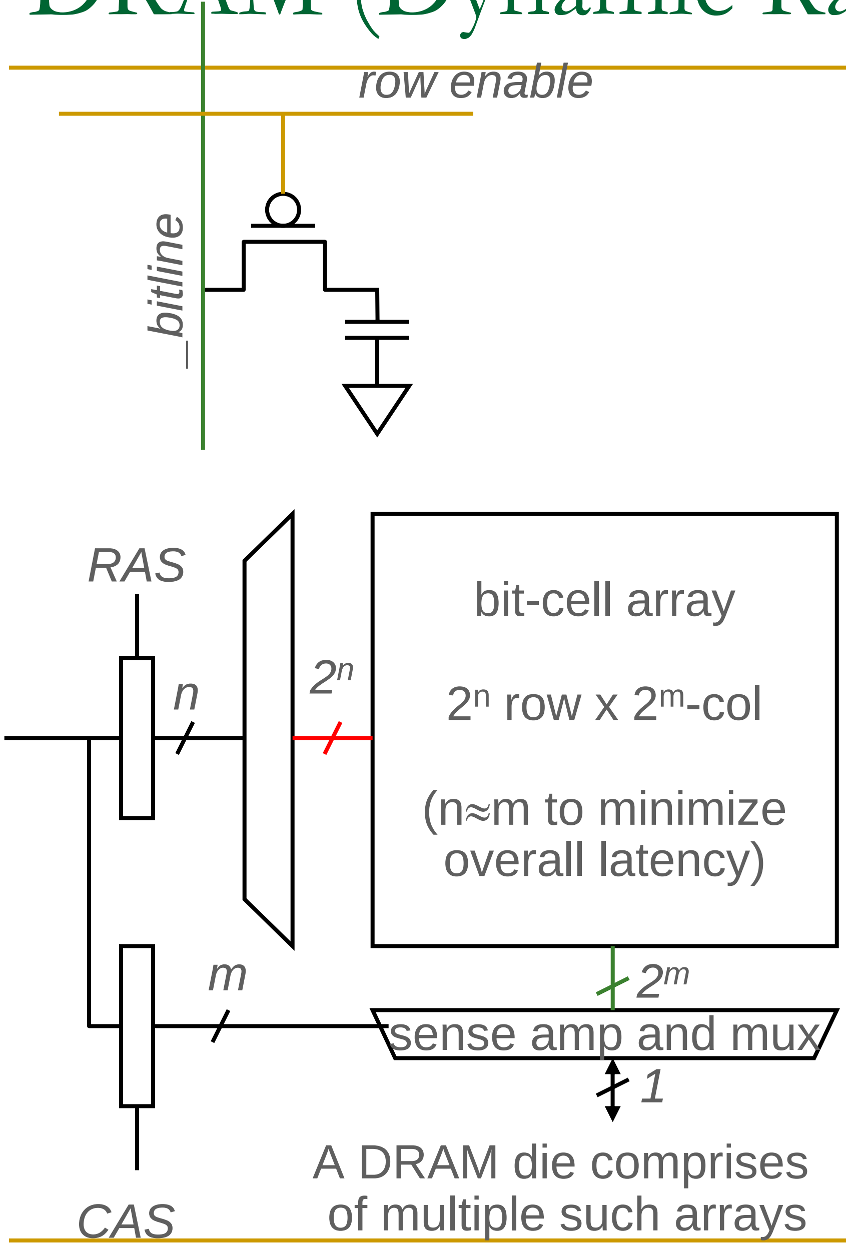
1. address decode
2. drive row select
3. selected bit-cells drive bitlines
   (entire row is read together)
4. diff. sensing and col. select
   (data is ready)
5. precharge all bitlines
   (for next read or write)

Access latency dominated by steps 2 and 3

Cycling time dominated by steps 2, 3 and 5

- step 2 proportional to 2^m
- step 3 and 5 proportional to 2^n

DRAM (Dynamic Random Access Memory)



A DRAM die comprises of multiple such arrays

Bits stored as charges on node capacitance (non-restorative)

- bit cell loses charge when read
- bit cell loses charge over time

Read Sequence

1~3 same as SRAM

4. a "flip-flopping" sense amp amplifies and regenerates the bitline, data bit is mux'ed out
5. precharge all bitlines

Refresh: A DRAM controller must periodically read all rows within the allowed refresh time (10s of ms) such that charge is restored in cells

SRAM vs. DRAM

- SRAM is preferable for register files and L1/L2 caches
  - Fast access
  - No refreshes
  - Simpler manufacturing (compatible with logic process)
  - Lower density (6 transistors per cell)
  - Higher cost

- DRAM is preferable for stand-alone memory chips
  - Much higher capacity
  - Higher density
  - Lower cost

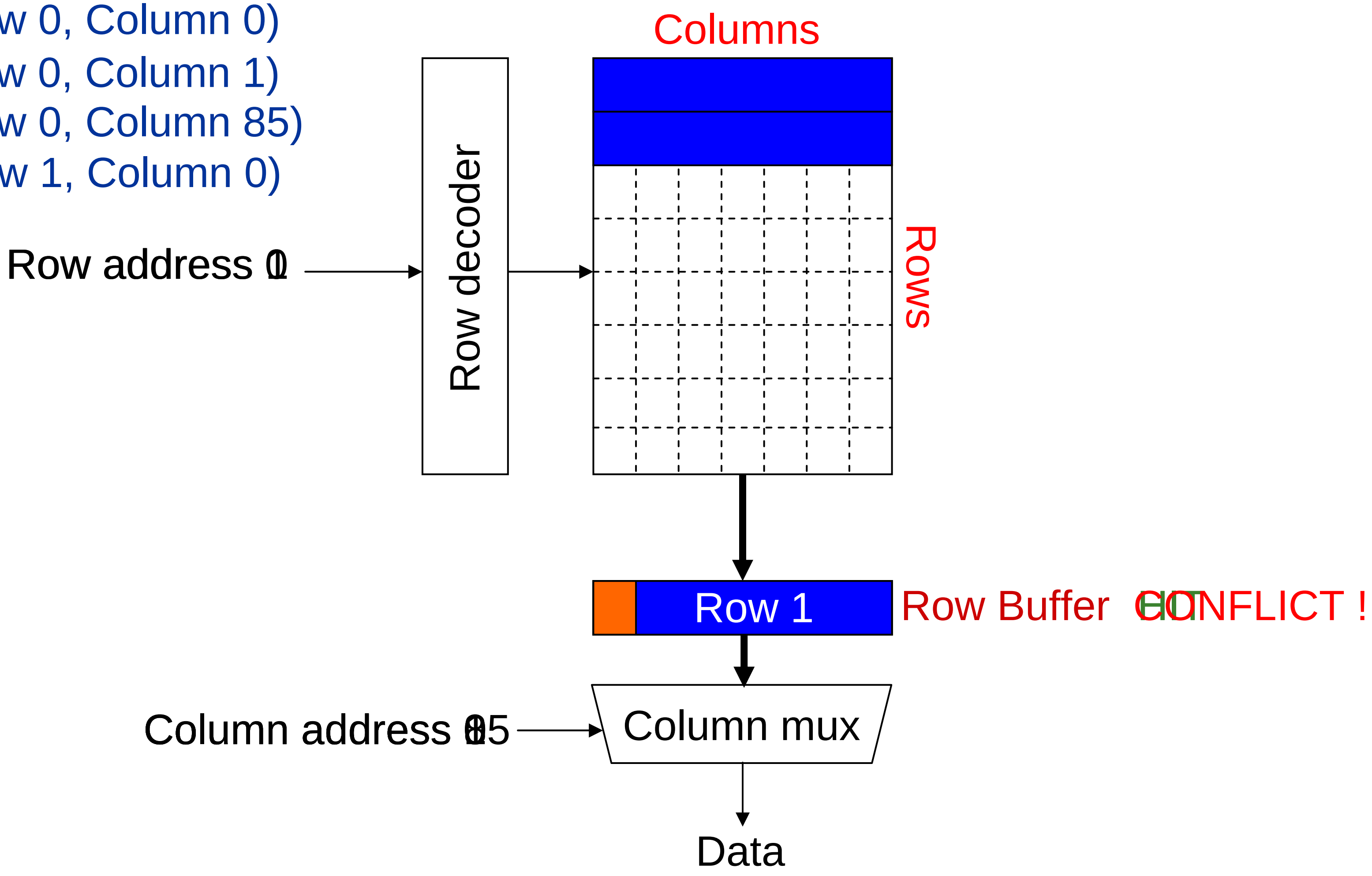
Page Mode DRAM

- 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 activate command

DRAM Bank Operation

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



Latency Components: Basic DRAM Operation

- CPU → controller transfer time
- Controller latency
  - Queuing & scheduling delay at the controller
  - Access converted to basic commands
- Controller → DRAM transfer time
- DRAM bank latency
  - Simple CAS is row is “open” OR
  - RAS + CAS if array precharged OR
  - PRE + RAS + CAS (worst case)
- DRAM → CPU transfer time (through controller)

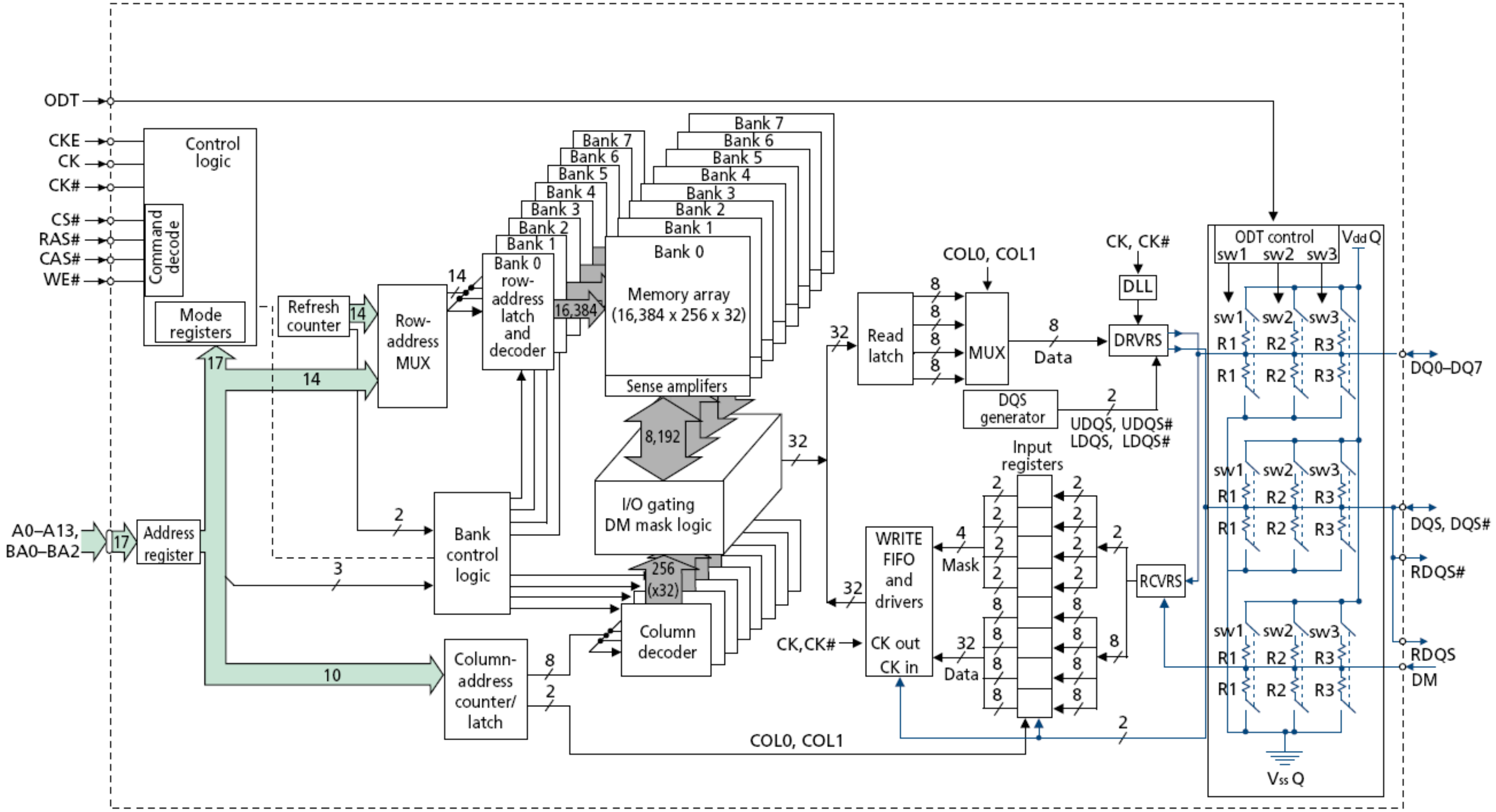
A DRAM Chip and DIMM

- Chip: Consists of multiple banks (2-16 in Synchronous DRAM)
- Banks share command/address/data buses
- The chip itself has a narrow interface (4-16 bits per read)

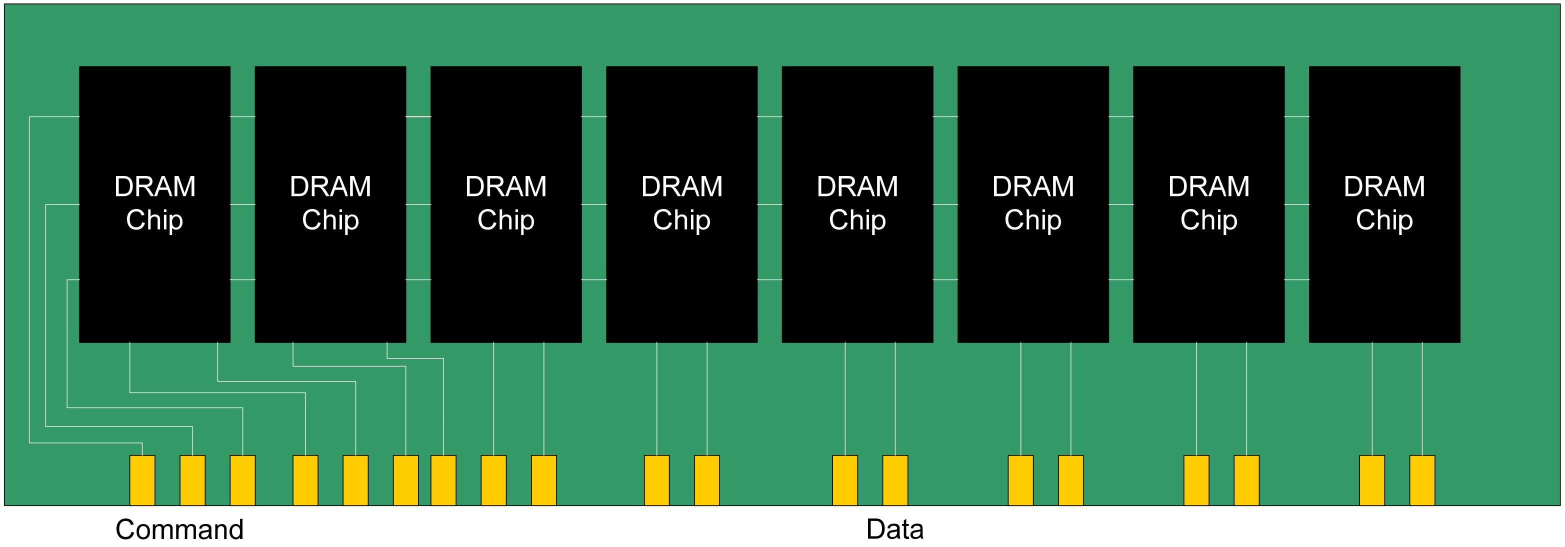
- Multiple chips are put together to form a wide interface
  - Called a module
  - DIMM: Dual Inline Memory Module
  - All chips in one side of a DIMM are operated the same way (rank)
    - Respond to a single command
    - Share address and command buses, but provide different data

- If we have chips with 8-bit interface, to read 8 bytes in a single access, use 8 chips in a DIMM

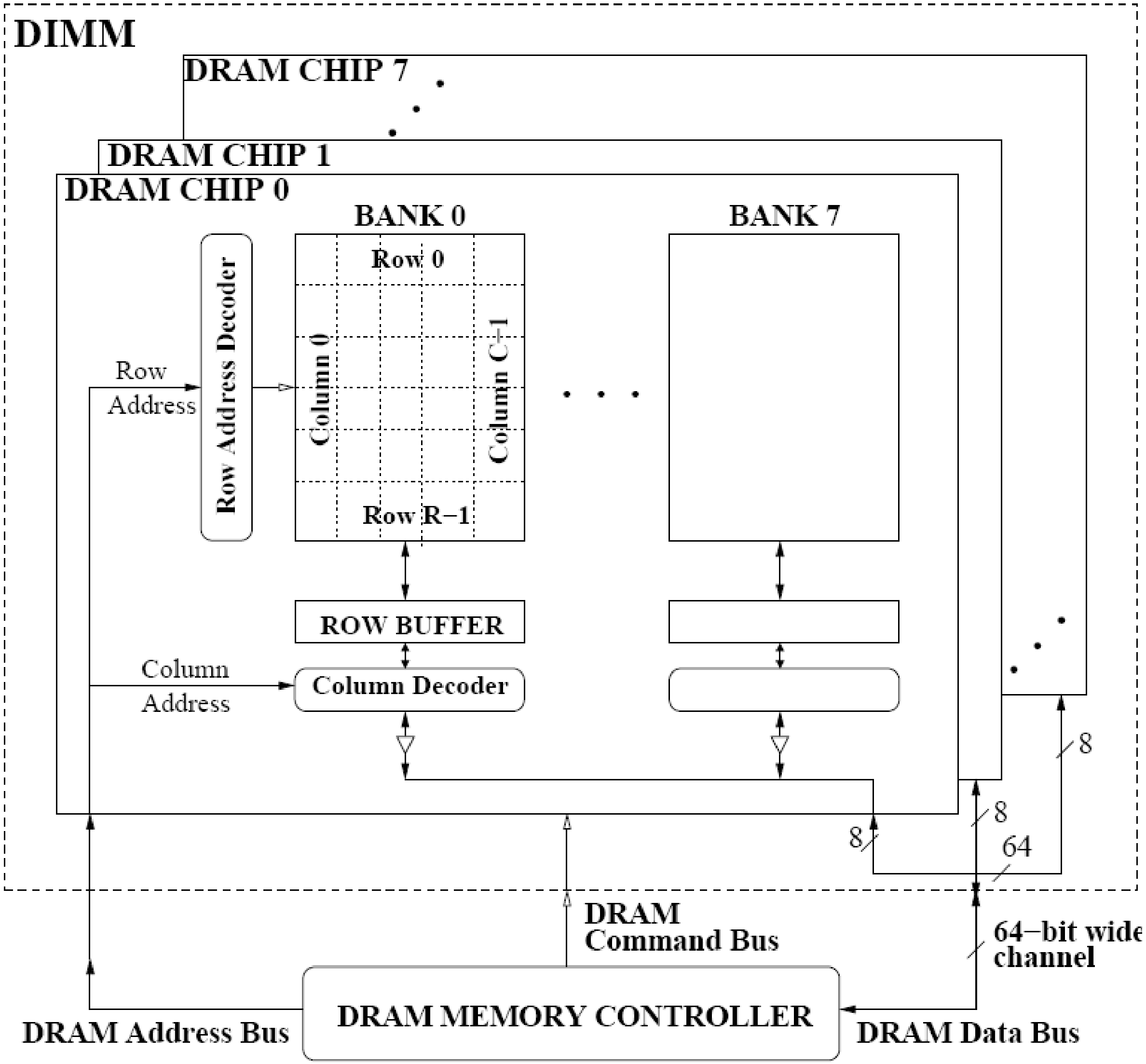
128M x 8-bit DRAM Chip



A 64-bit Wide DIMM



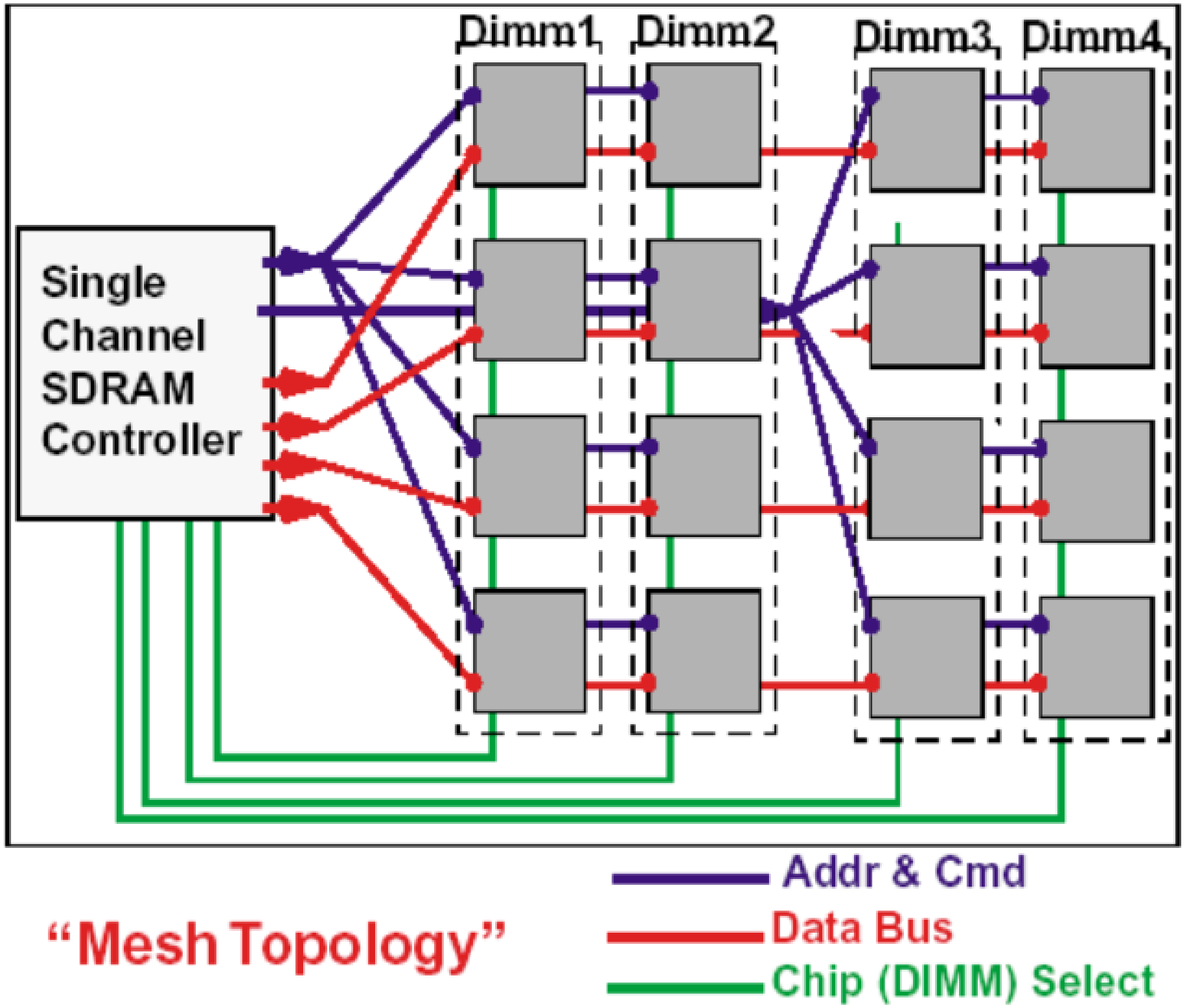
A 64-bit Wide DIMM



- 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

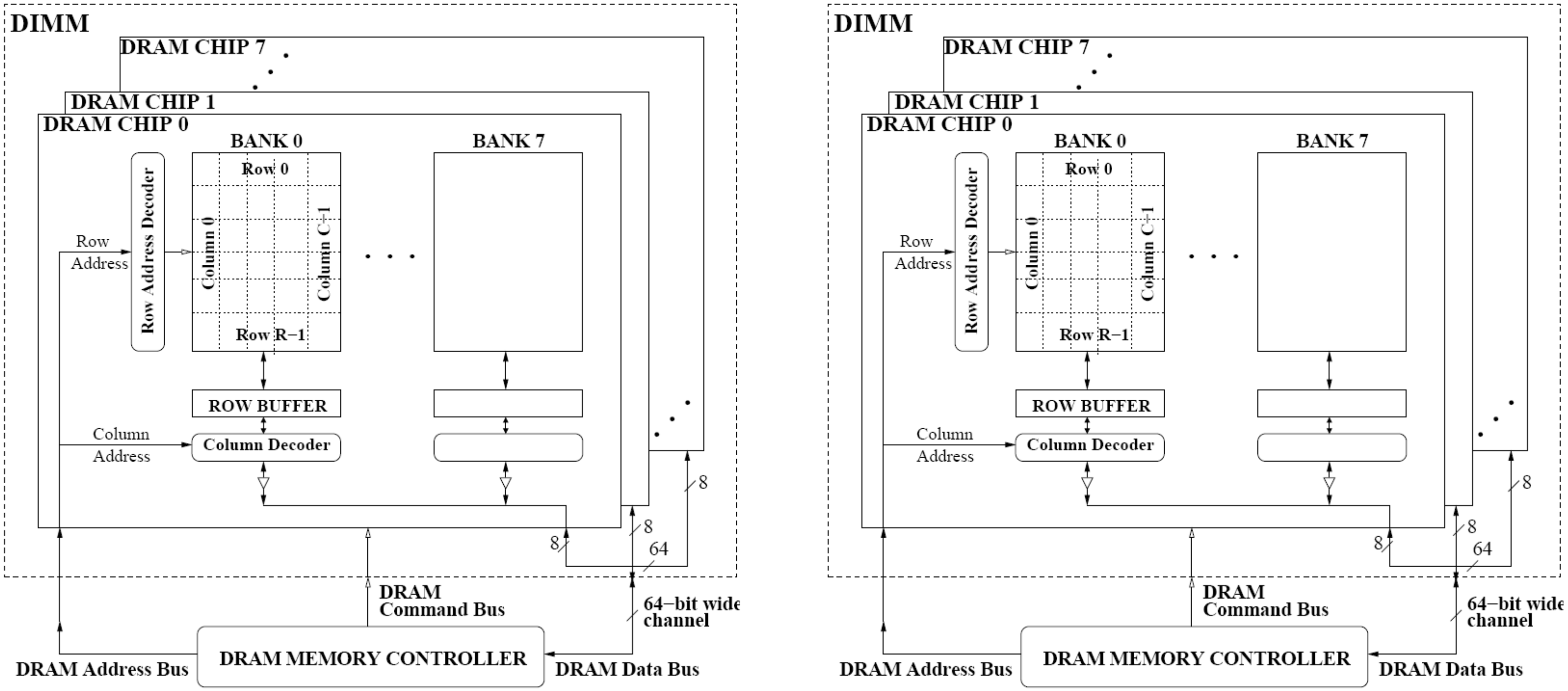
Multiple DIMMs



"Mesh Topology"

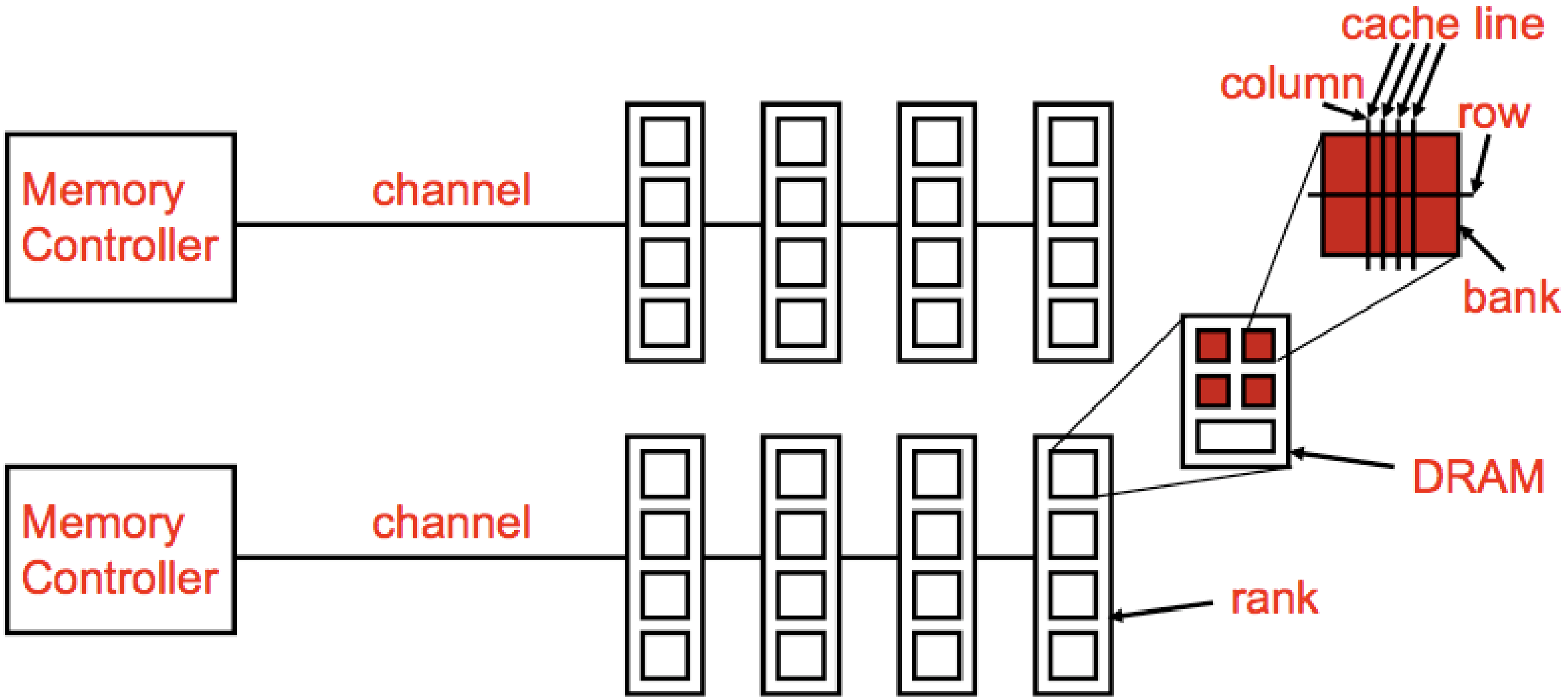
- Advantages:
  - Enables even higher capacity
- Disadvantages:
  - Interconnect complexity and energy consumption can be high

DRAM Channels



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

Generalized Memory Structure

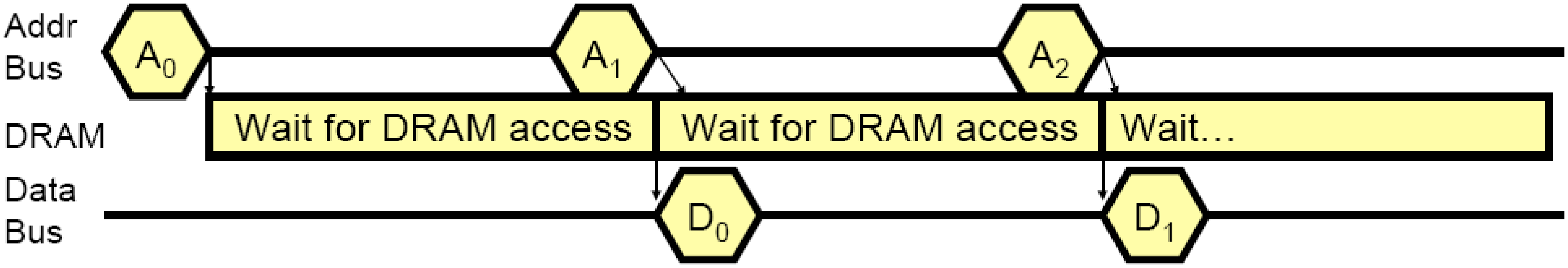


Multiple Banks (Interleaving) and Channels

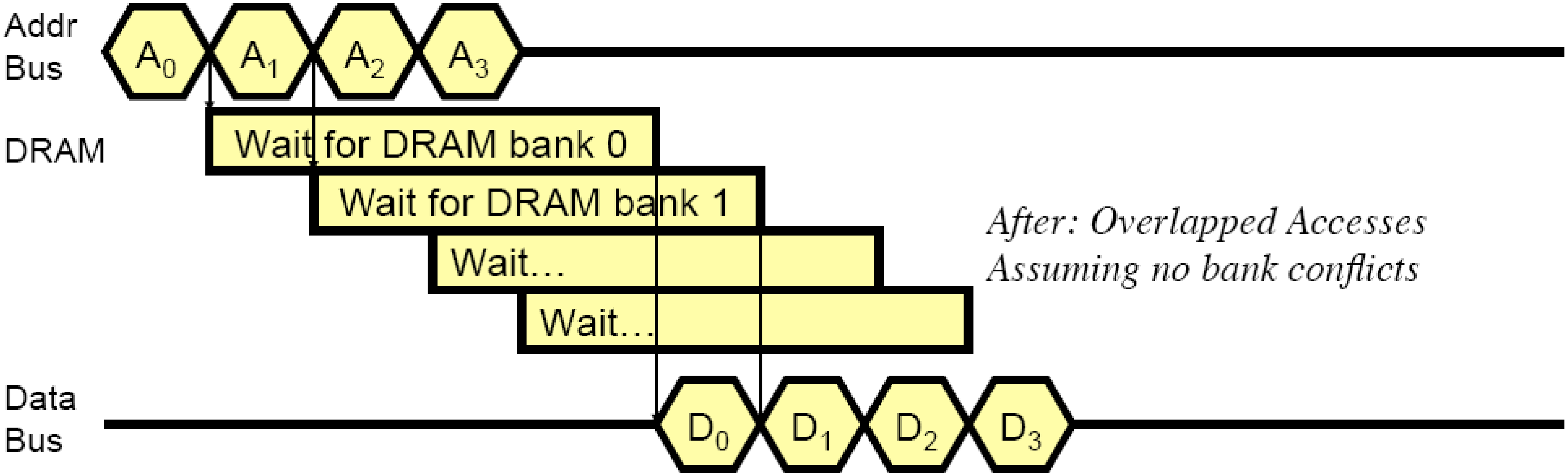
- Multiple banks
  - Enable concurrent DRAM accesses
  - Bits in address determine which bank an address resides in
- Multiple independent channels serve the same purpose
  - But they are even better because they have separate data buses
  - Increased bus bandwidth

- Enabling more concurrency requires reducing
  - Bank conflicts
  - Channel conflicts
- How to select/randomize bank/channel indices in address?
  - Lower order bits have more entropy
  - Randomizing hash functions (XOR of different address bits)

How Multiple Banks/Channels Help



Before: No Overlapping
Assuming accesses to different DRAM rows



After: Overlapped Accesses
Assuming no bank conflicts

Multiple Channels

- Advantages
  - Increased bandwidth
  - Multiple concurrent accesses (if independent channels)

- Disadvantages
  - Higher cost than a single channel
    - More board wires
    - More pins (if on-chip memory controller)

Address Mapping (Single Channel)

- Single-channel system with 8-byte memory bus
  - 2GB memory, 8 banks, 16K rows & 2K columns per bank
- Row interleaving
  - Consecutive rows of memory in consecutive banks

Row (14 bits)	Bank (3 bits)	Column (11 bits)	Byte in bus (3 bits)

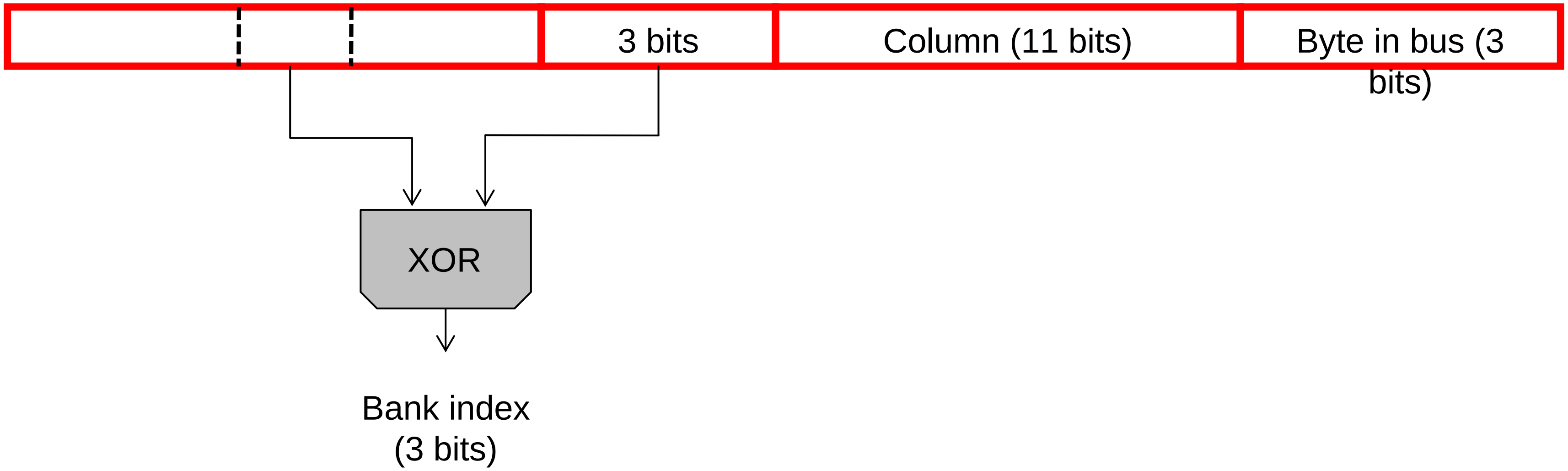
- Cache block interleaving
  - Consecutive cache block addresses in consecutive banks
  - 64 byte cache blocks

Row (14 bits)	High Column	Bank (3 bits)	Low Col.	Byte in bus (3 bits)
	8 bits		3 bits	

  - Accesses to consecutive cache blocks can be serviced in parallel
  - How about random accesses? Strided accesses?

Bank Mapping Randomization

- DRAM controller can randomize the address mapping to banks so that bank conflicts are less likely



Address Mapping (Multiple Channels)

C	Row (14 bits)	Bank (3 bits)	Column (11 bits)	Byte in bus (3 bits)

Row (14 bits)	C	Bank (3 bits)	Column (11 bits)	Byte in bus (3 bits)

Row (14 bits)	Bank (3 bits)	C	Column (11 bits)	Byte in bus (3 bits)

Row (14 bits)	Bank (3 bits)	Column (11 bits)	C	Byte in bus (3 bits)

- Where are consecutive cache blocks?

C	Row (14 bits)	High Column	Bank (3 bits)	Low Col.	Byte in bus (3 bits)
		8 bits		3 bits	

Row (14 bits)	C	High Column	Bank (3 bits)	Low Col.	Byte in bus (3 bits)
		8 bits		3 bits	

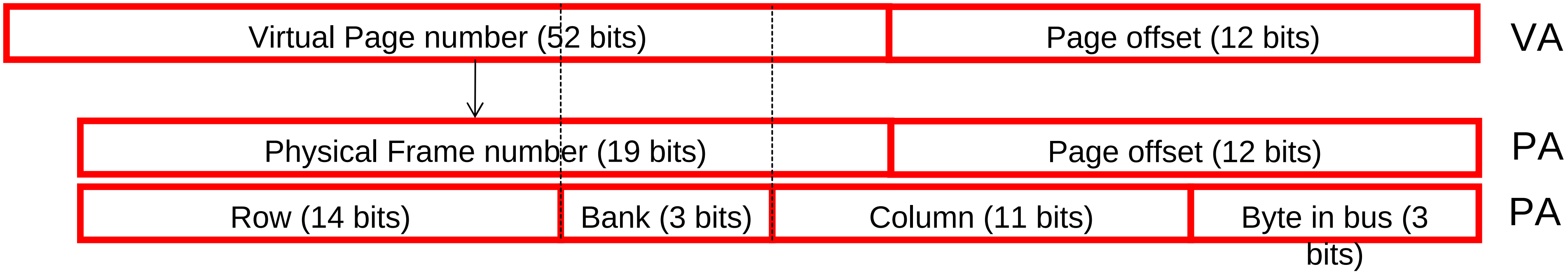
Row (14 bits)	High Column	C	Bank (3 bits)	Low Col.	Byte in bus (3 bits)
	8 bits			3 bits	

Row (14 bits)	High Column	Bank (3 bits)	C	Low Col.	Byte in bus (3 bits)
	8 bits			3 bits	

Row (14 bits)	High Column	Bank (3 bits)	Low Col.	C	Byte in bus (3 bits)
	8 bits		3 bits		

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 Frame 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 control which bank a virtual page is mapped to. It can randomize Page→<Bank,Channel> mappings

- Application cannot know/determine which bank it is accessing

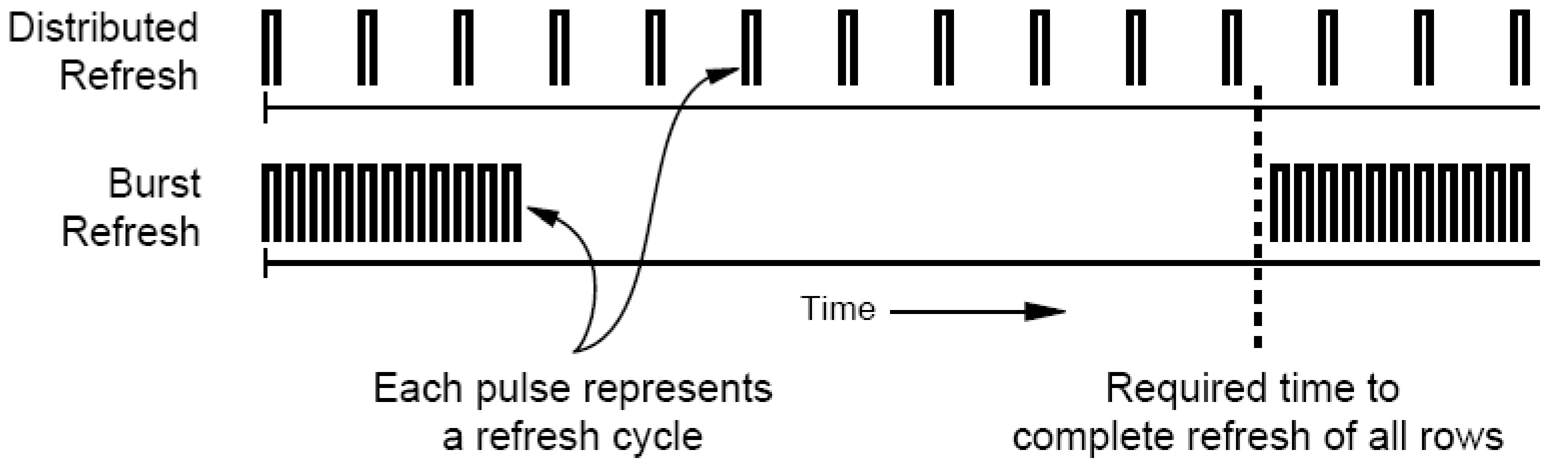
DRAM Refresh (I)

- DRAM capacitor charge leaks over time
- The memory controller needs to read each row periodically to restore the charge
  - Activate + precharge each row every N ms
  - Typical N = 64 ms
- Implications on performance?

  -- DRAM bank unavailable while refreshed

  -- Long pause times: If we refresh all rows in burst, every 64ms the DRAM will be unavailable until refresh ends
- Burst refresh: All rows refreshed immediately after one another
- Distributed refresh: Each row refreshed at a different time, at regular intervals

DRAM Refresh (II)


Distributed Refresh
Burst Refresh
Time
Each pulse represents a refresh cycle
Required time to complete refresh of all rows

- Distributed refresh eliminates long pause times
- How else we can reduce the effect of refresh on performance?
  - Can we reduce the number of refreshes?

DRAM Controller

- Purpose and functions
  - Ensure correct operation of DRAM (refresh)
  - Service DRAM requests while obeying timing constraints of DRAM chips
    - Constraints: resource conflicts (bank, bus, channel), minimum write-to-read delays
    - Translate requests to DRAM command sequences
  - Buffer and schedule requests to improve performance
    - Reordering and row-buffer management
  - Manage power consumption and thermals in DRAM
    - Turn on/off DRAM chips, manage power modes

DRAM Controller Issues

- Where to place?
  - In chipset

    + More flexibility to plug different DRAM types into the system

    + Less power density in the CPU chip
  - On CPU chip

    + Reduced latency for main memory access

    + Higher bandwidth between cores and controller
    - More information can be communicated (e.g. request’s importance in the processing core)