CMPE110 Lecture 16
DRAM

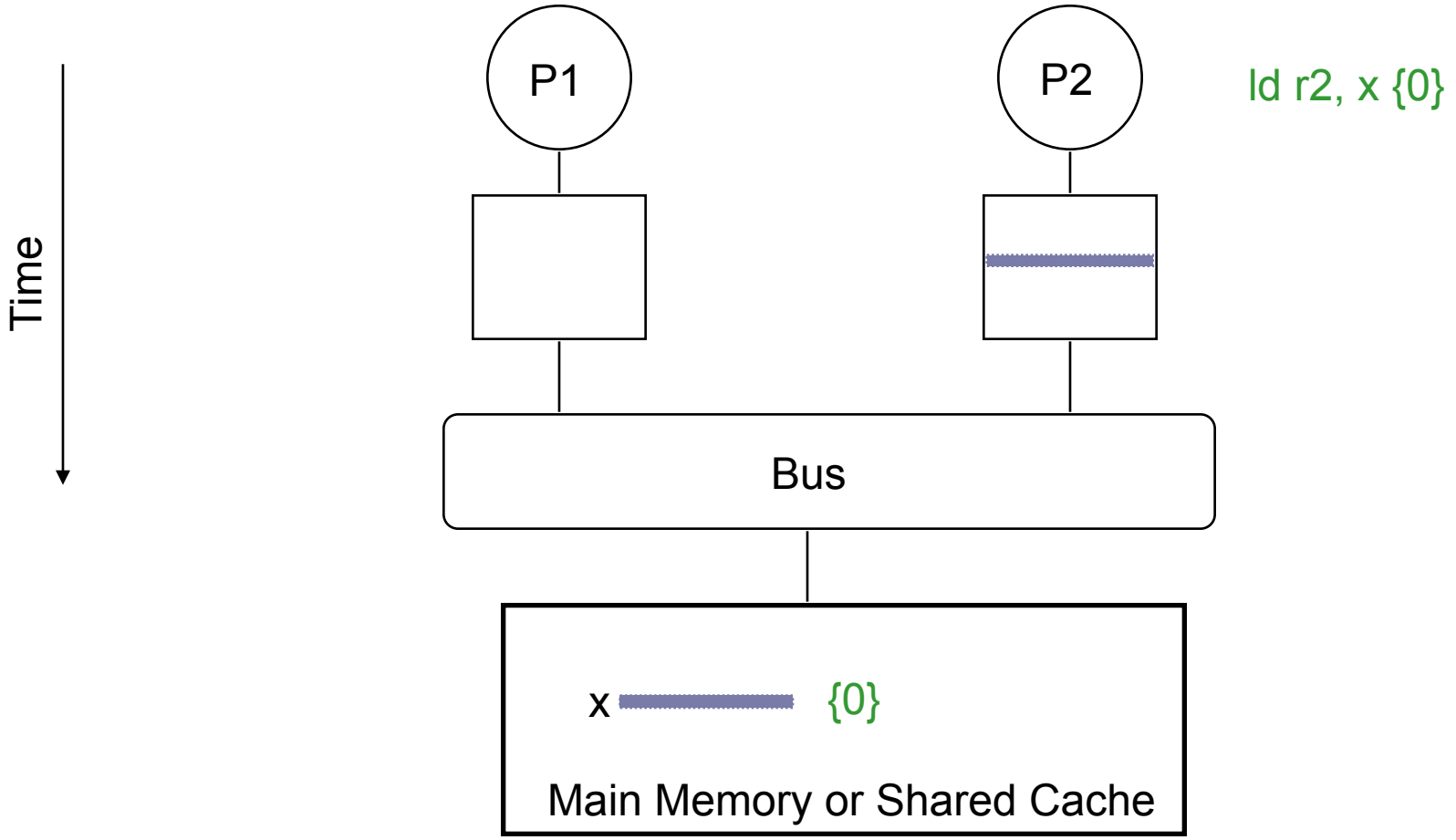
Heiner Litz

https://canvas.ucsc.edu/courses/12652

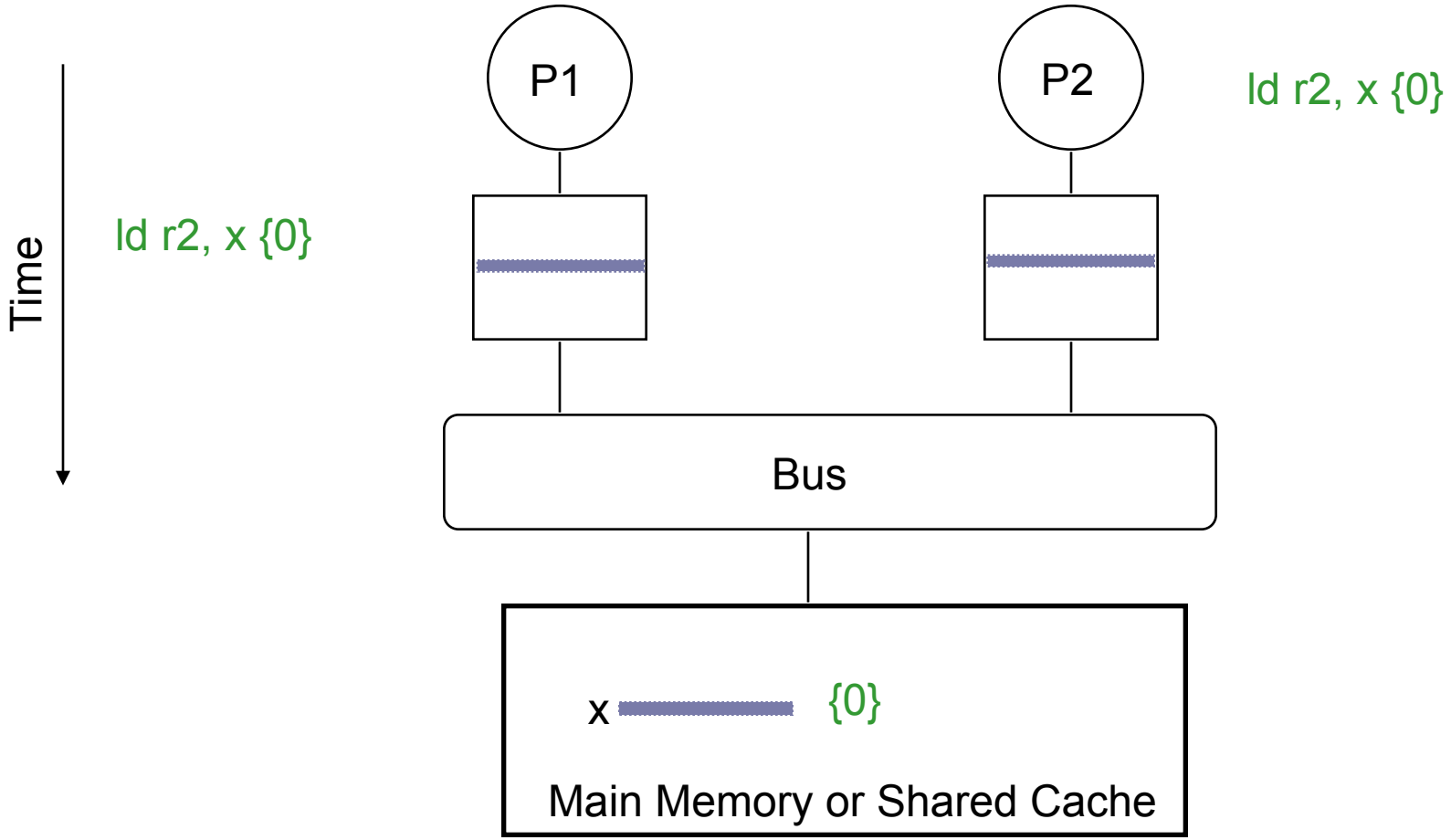
Announcements

Review

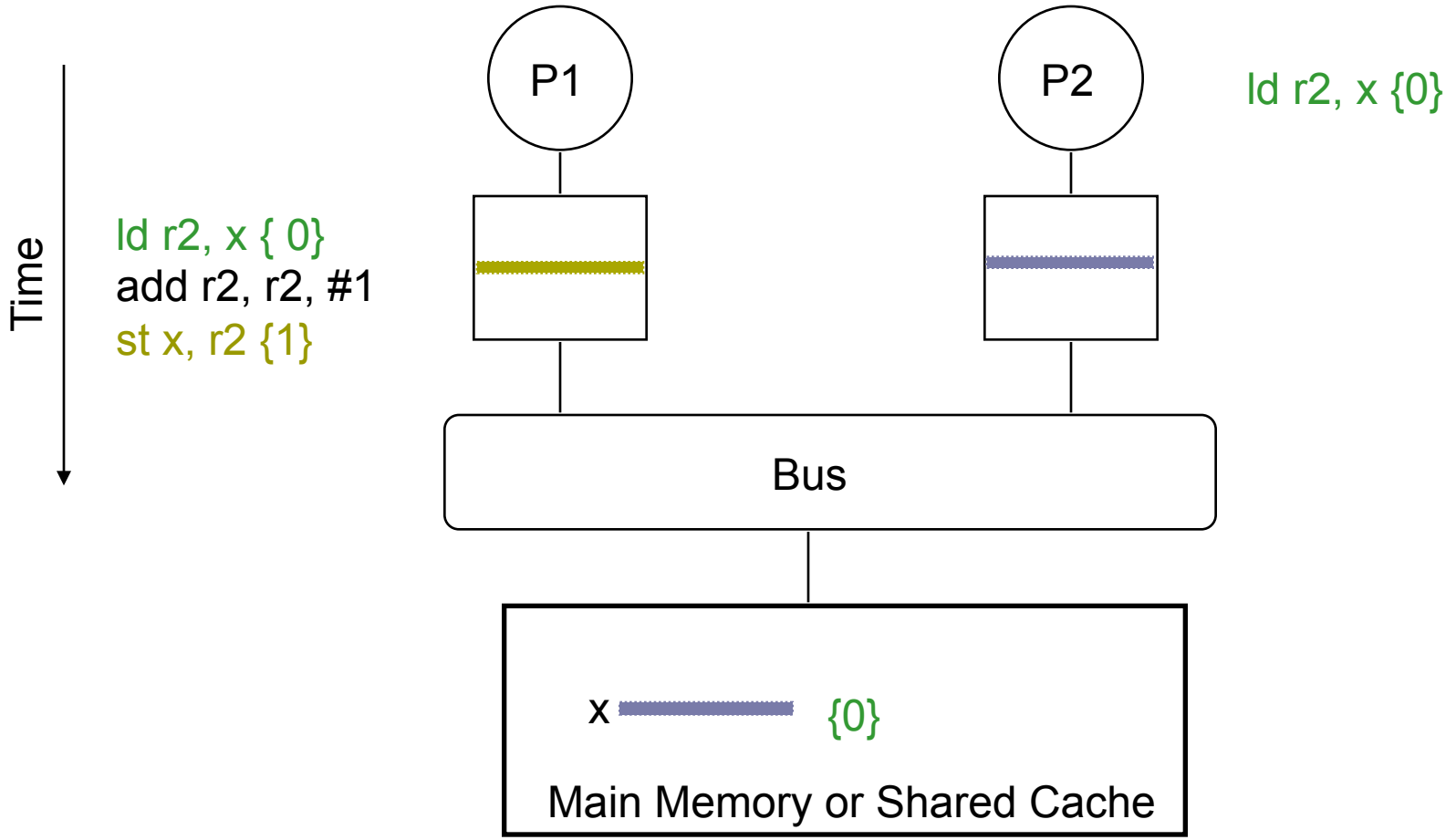
Cache Coherence Problem (Step 1)



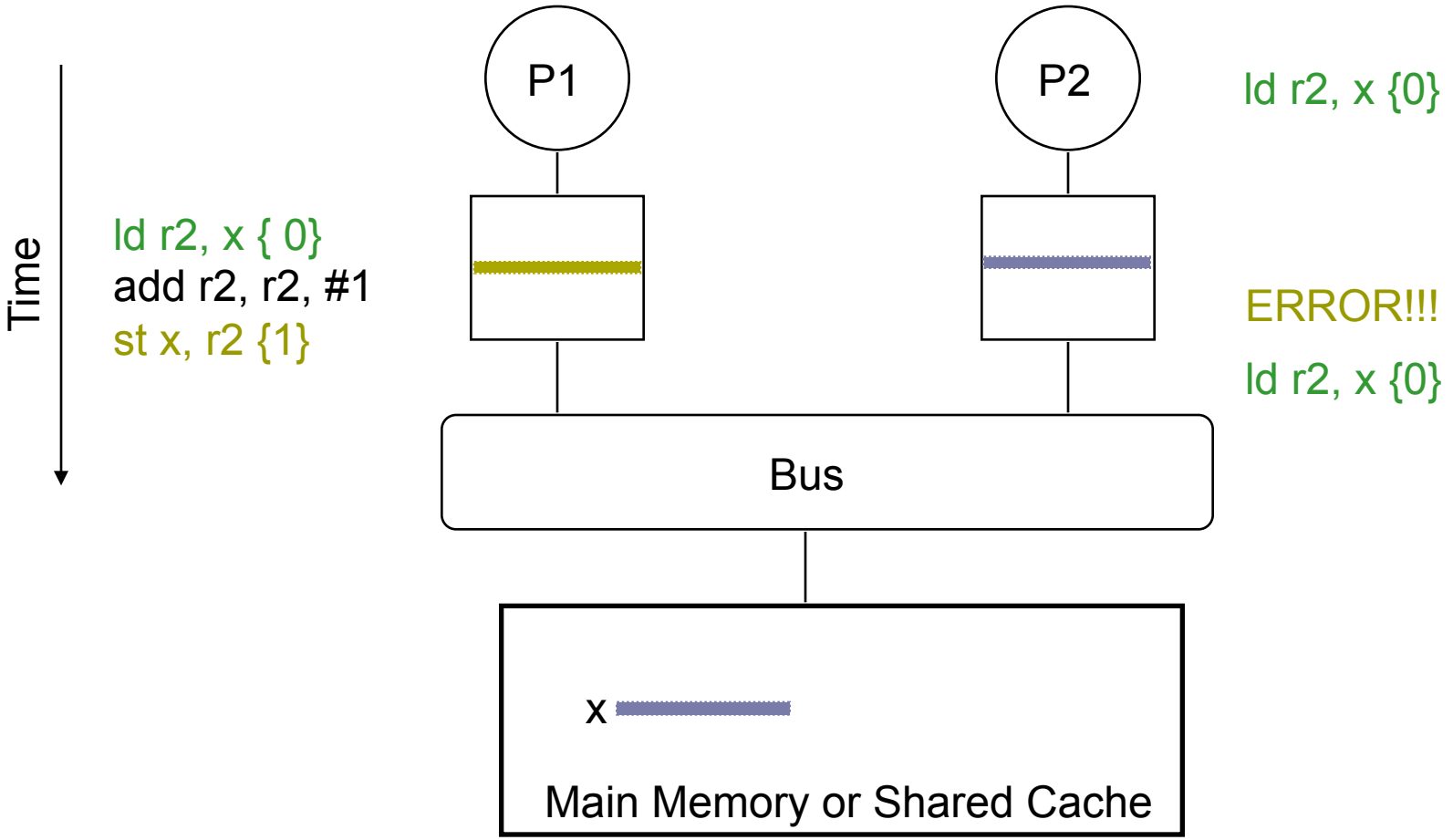
Cache Coherence Problem (Step 2)



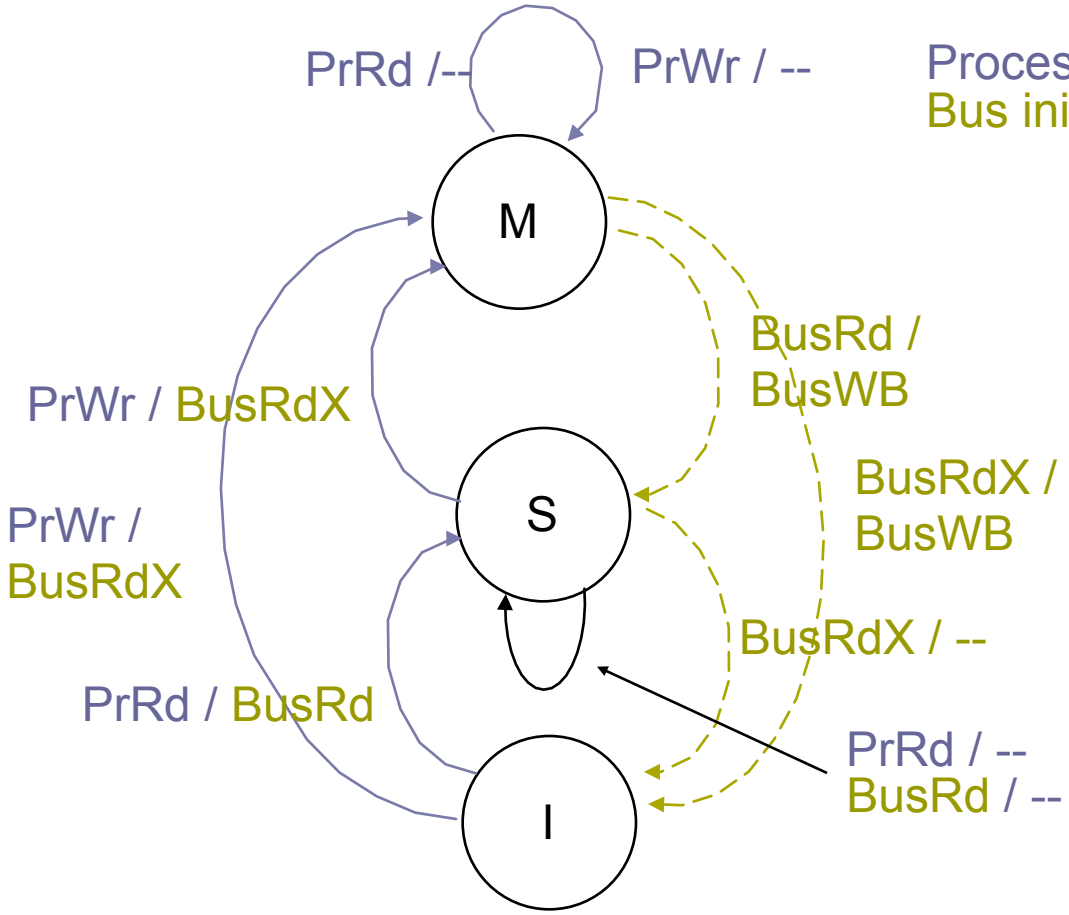
Cache Coherence Problem (Step 3)



Cache Coherence Problem (Step 4)



MSI Coherence Protocol



Abbreviation	Action
PrRd	Processor Read
PrWr	Processor Write
BusRd	Bus Read
BusRdX	Bus Read Exclusive
BusWB	Bus Writeback

Fallacy: Caches are Transparent to Software

- Example
  - Cold cache, 4-byte words, 4-word cache blocks
  - C stores arrays in row-major order
  - Assume N >> 4, and 4MN > cache size

```
int sumarrayrows(int a[M][N])
{
    int i, j, sum = 0;

    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            sum += a[i][j];
    return sum;
}
```

Miss rate =

```
int sumarraycols(int a[M][N])
{
    int i, j, sum = 0;

    for (j = 0; j < N; j++)
        for (i = 0; i < M; i++)
            sum += a[i][j];
    return sum;
}
```

Miss rate =

Matrix Multiplication Example

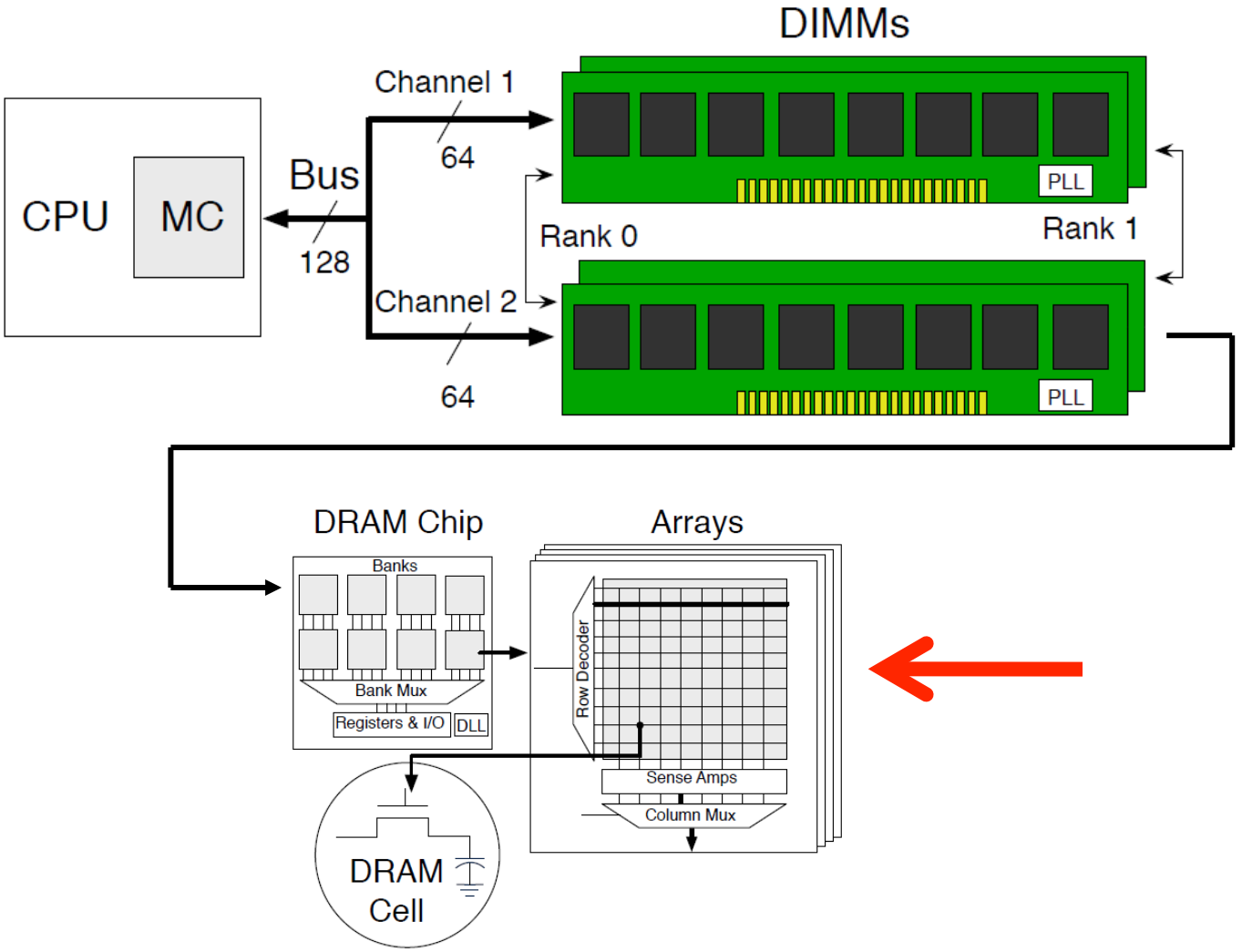
- Description:
  - Multiply N x N matrices
  - $O(N^3)$ total operations
  - Accesses
    - N reads per source element
    - N values summed per destination

```
/* ijk */
for (i=0; i<n; i++)  {
  for (j=0; j<n; j++) {
    sum = 0.0;
    for (k=0; k<n; k++)
      sum += a[i][k] * b[k][j];
    c[i][j] = sum;
  }
}
```

- Writing cache friendly
  - Repeated references to variables are good (temporal locality)
  - Stride-1 reference patterns are good (spatial locality)

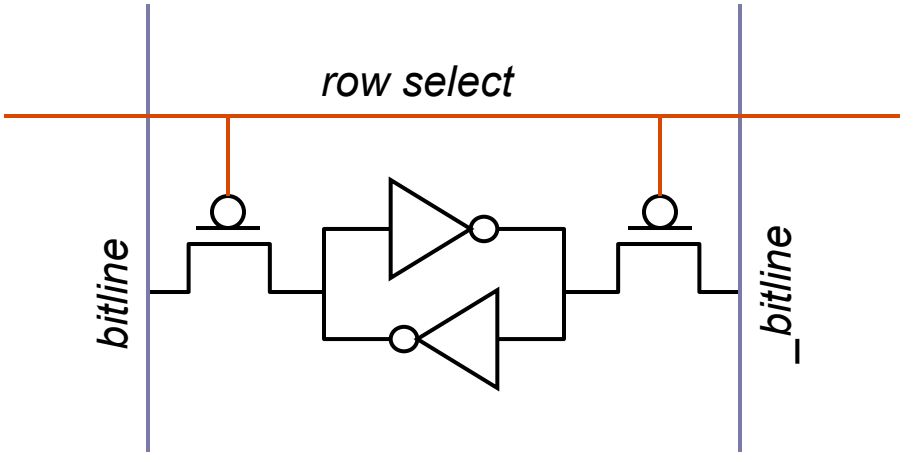
DRAM

Main Memory Overview



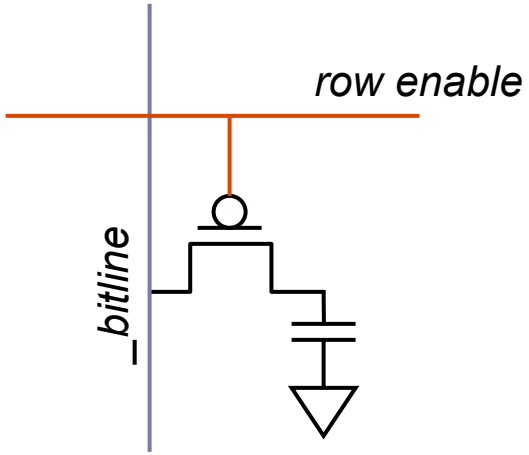
SRAM vs. DRAM

Static Random Access Mem.



- 6T vs. 1T1C
  - Large (~6-10x)
- Bitlines driven by transistors
  - Fast (~10x)

Dynamic Random Access Mem.

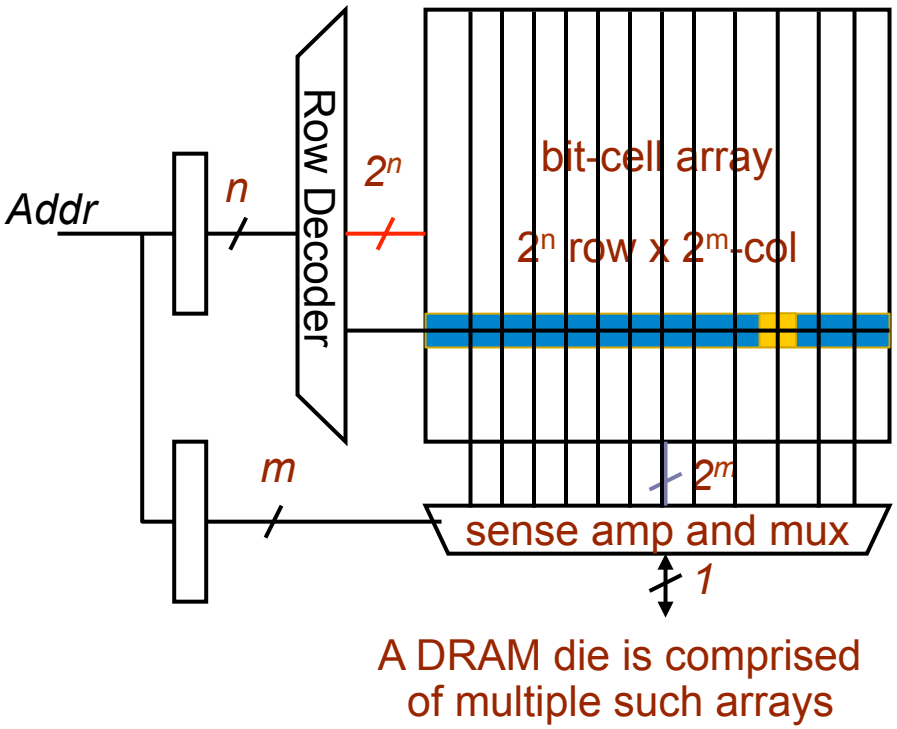


- Bits stored as charges on node capacitance (non-restorative)
  - Bit cell loses charge when read (destructive read)
  - Bit cell loses charge over time
- Must periodically refresh
  - Once every 10s of ms

SRAM vs. DRAM

- SRAM is preferable for register files and L1/L2 caches
  - Fast access
  - No refreshes to worry about
  - 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

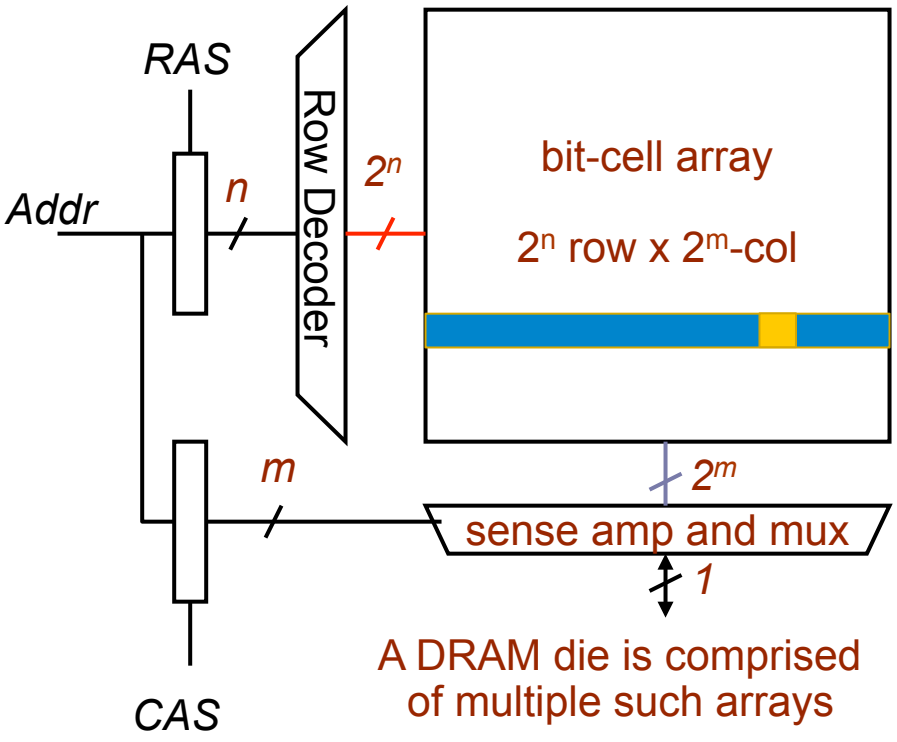
Background: Memory Bank Organization



A DRAM die is comprised of multiple such arrays

- Read access sequence
  - Decode row address & drive word-lines
  - Selected bits drive bit-lines
    - Entire row read
  - Amplify row data
  - Decode column address & select subset of row
  - Send to output
  - Precharge bit-lines for next access

DRAM: Memory-Access Protocol



- 5 basic commands
  - ACTIVATE (open a row)
  - READ (read a column)
  - WRITE
  - PRECHARGE (close row)
  - REFRESH
- To reduce pin count, row and column share same address pins
  - RAS = Row Address Strobe
  - CAS = Column Address Strobe

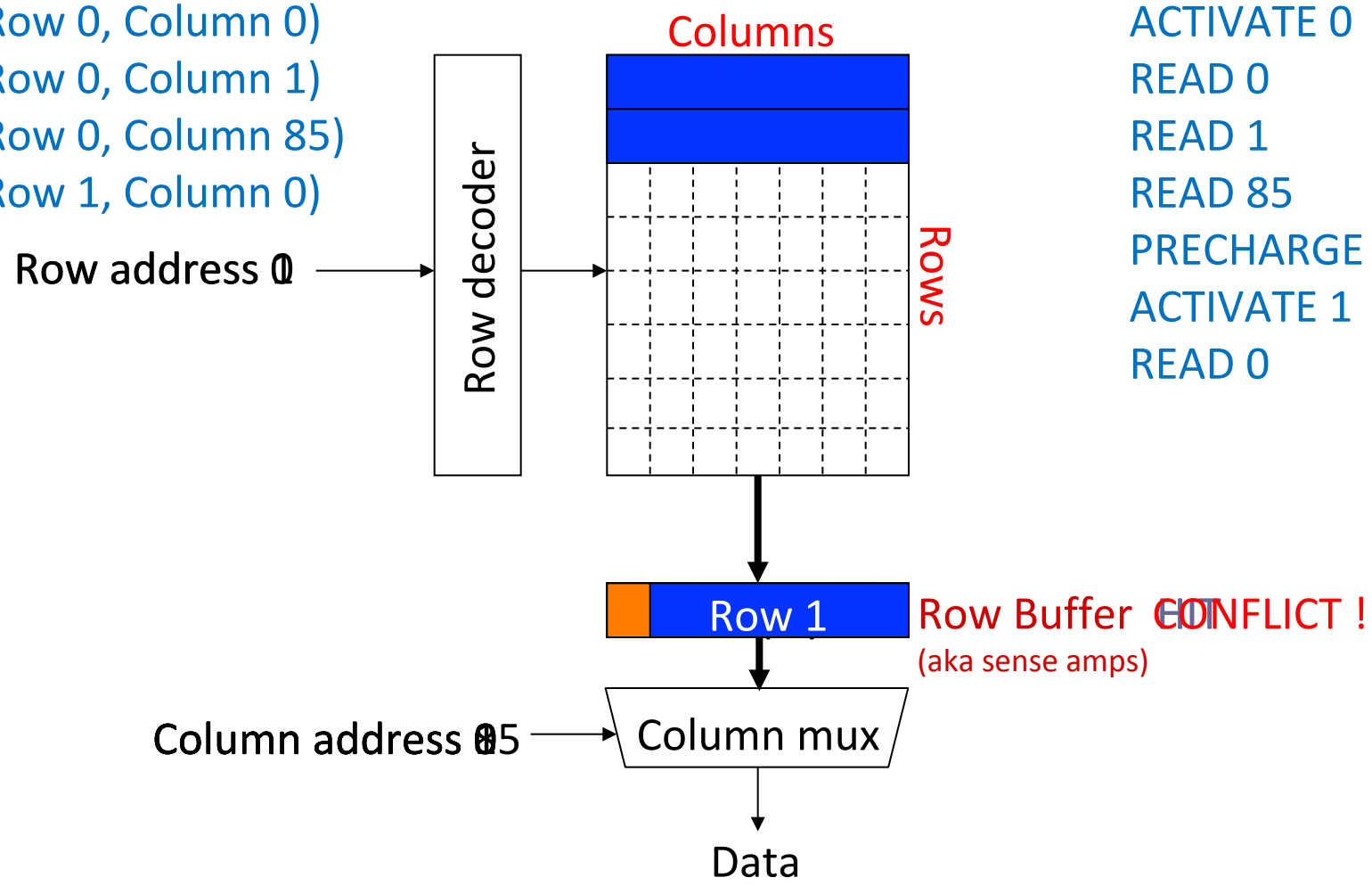
DRAM: Basic Operation

Addresses

(Row 0, Column 0)
(Row 0, Column 1)
(Row 0, Column 85)
(Row 1, Column 0)

Commands

ACTIVATE 0
READ 0
READ 1
READ 85
PRECHARGE
ACTIVATE 1
READ 0



DRAM: Basic Operation

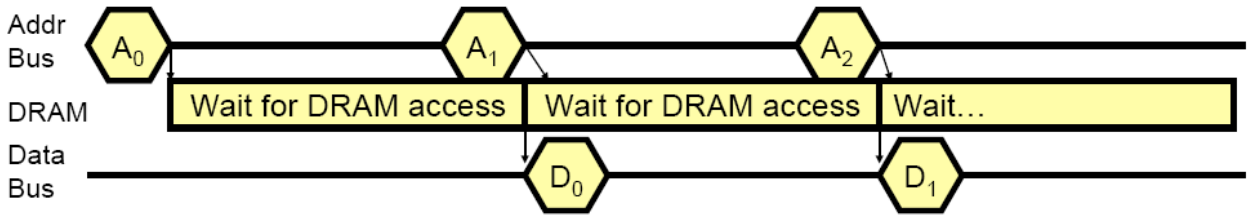
- Access to an “open row”
  - No need for ACTIVATE command
  - READ/WRITE to access row buffer
- Access to a “closed row”
  - If another row already active, must first issue PRECHARGE
  - ACTIVATE to open new row
  - READ/WRITE to access row buffer
  - Optional: PRECHARGE after READ/WRITEs finished

DRAM: Banks

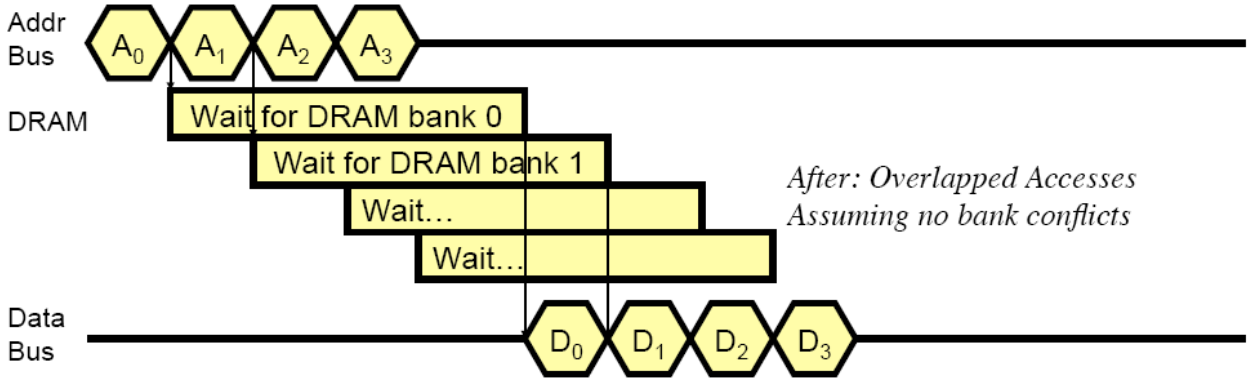
- Modern DRAM chips consist of multiple **banks**
  - Address = (Bank x, Row y, Column z)
- Banks operate independently, but share command/address/data pins
  - Each can have a different row active
  - Can overlap ACTIVATE and PRECHARGE latencies!
    (i.e. READ to bank 0 while ACTIVATING bank 1)

DRAM: Banks

- Enable concurrent DRAM accesses (overlapping)

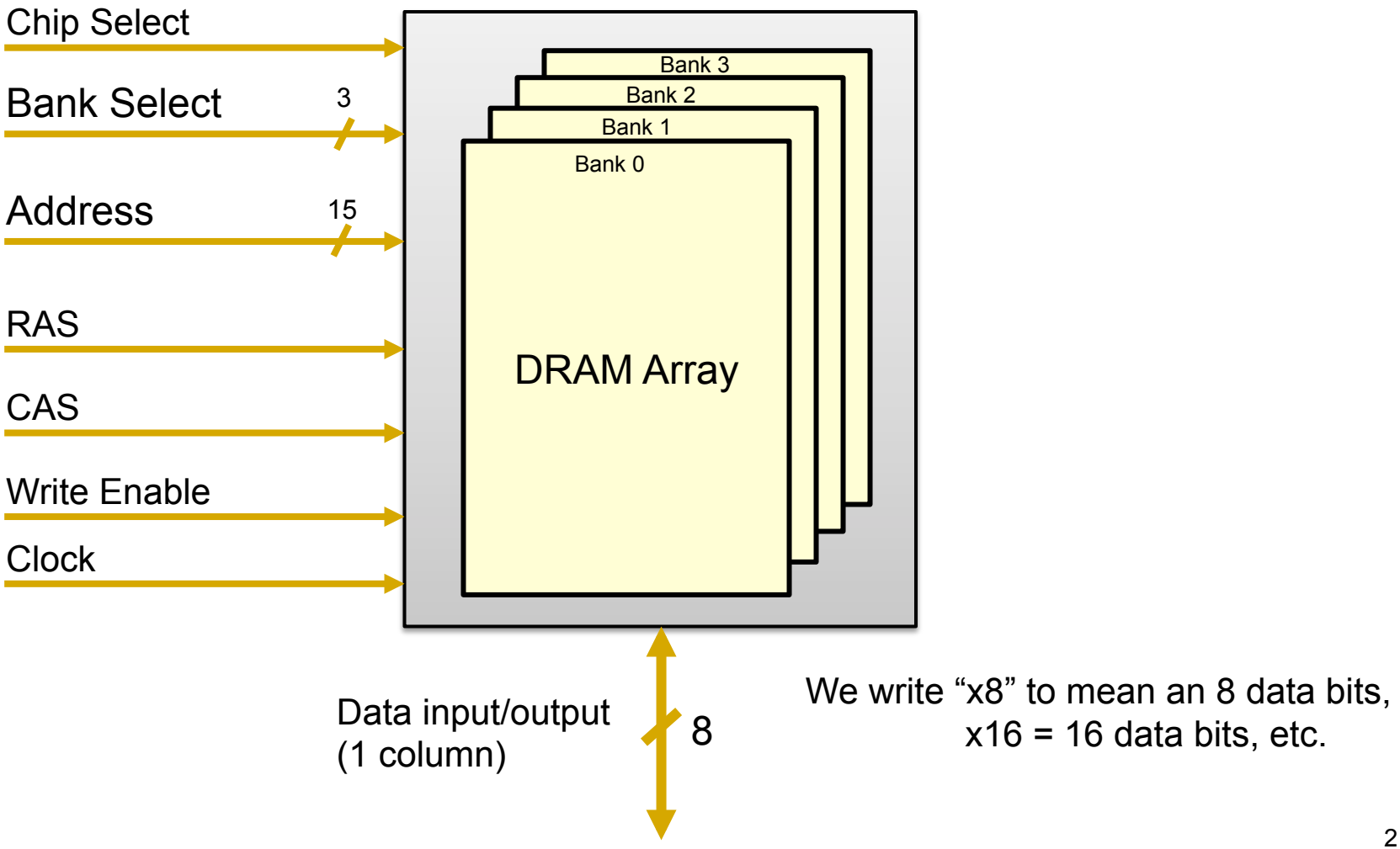


Before: No Overlapping
Assuming accesses to different DRAM rows

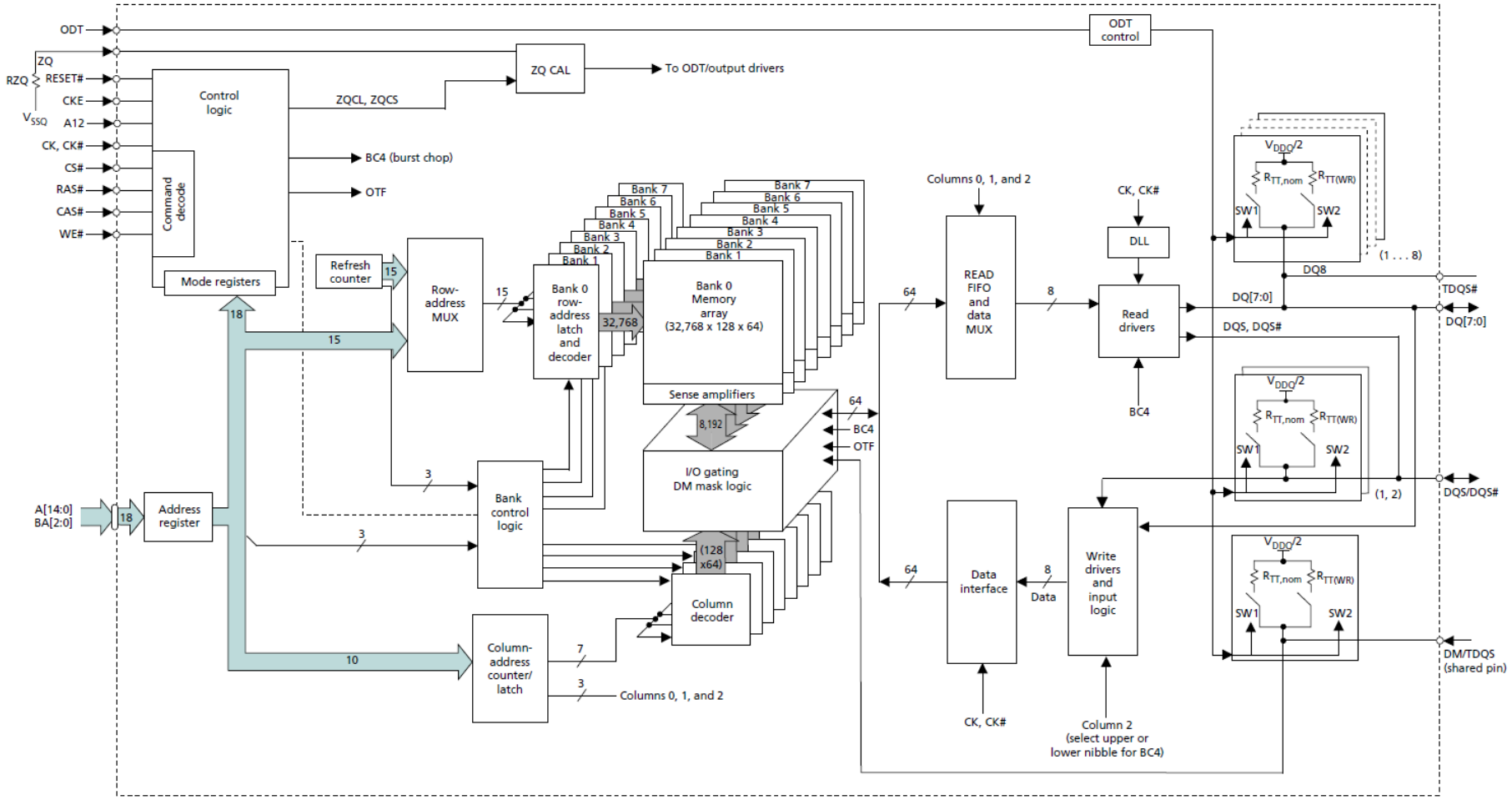


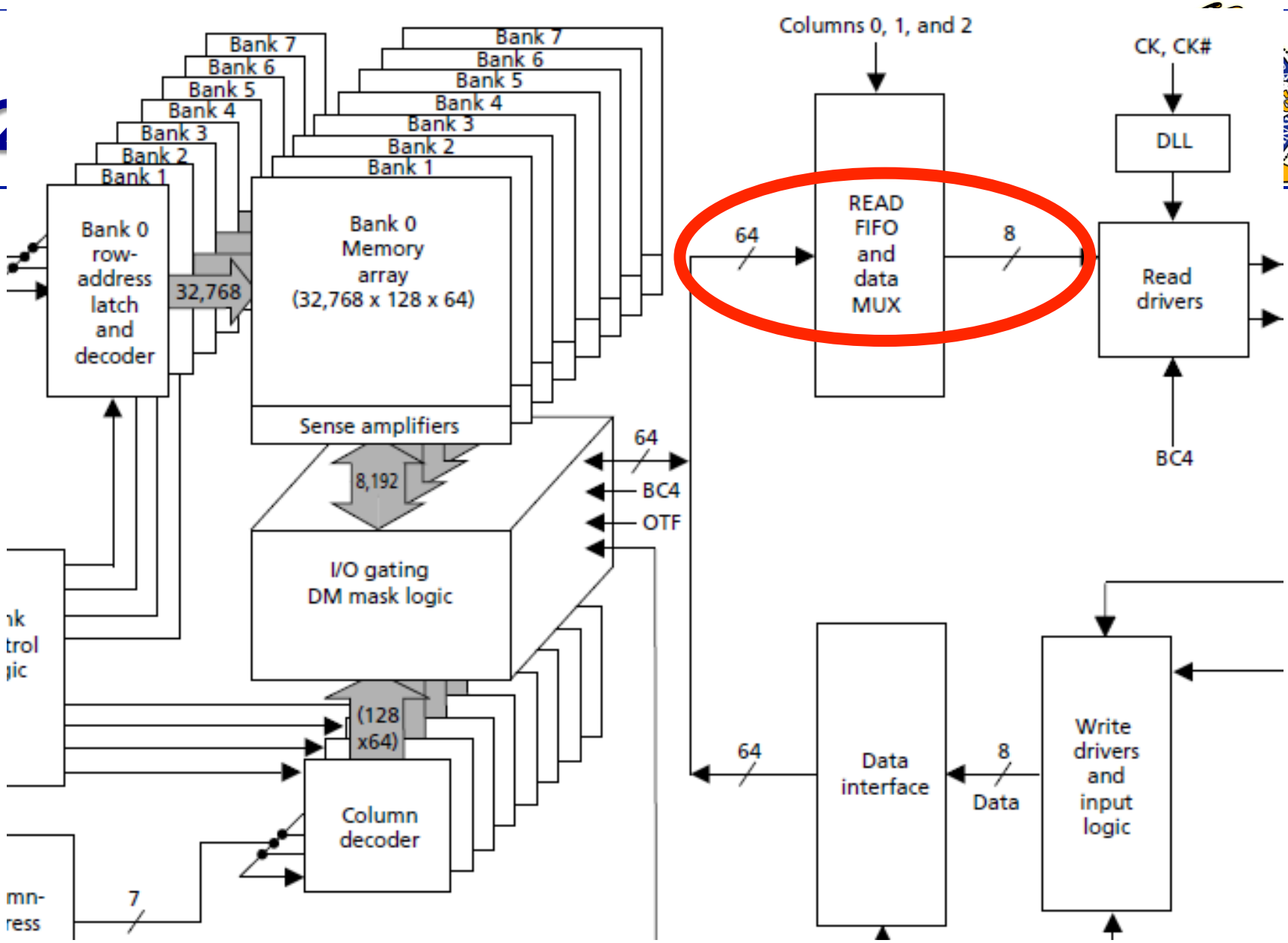
After: Overlapped Accesses
Assuming no bank conflicts

DRAM Chip: High-level View



2Gb x8 DDR3 Chip [Micron]

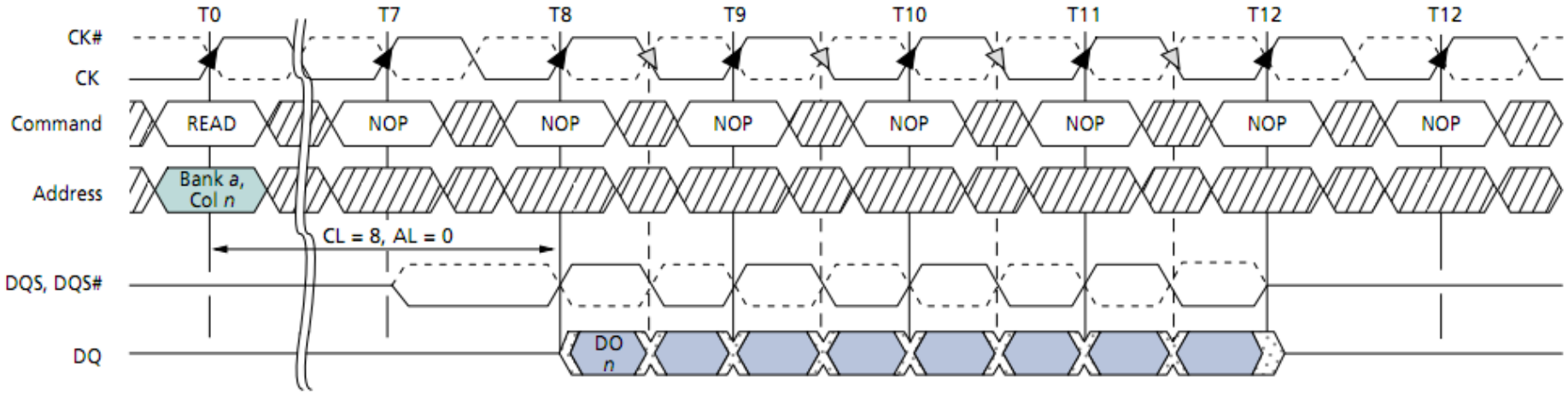




Observe: row width, 64 → 8 bit datapath

DRAM: Burst

- Each READ/WRITE command transfers multiple columns (8 in DDR3)
- DRAM channel clocked faster than DRAM core



- Critical word first?

DDR4 SDRAM: Current industry standard

- Introduced in 2014
- SDRAM = Synchronous DRAM = **Clocked**
- DDR = Double Data Rate
  - Data transferred on both clock edges
  - min 800 Mhz = 1600 MT/s
  - max 2133 MHz = 4266 MT/s
- x4, x8, x16 datapath widths
- Minimum burst length of 8
- 8 banks
- 8Gb, 16Gb, 32Gb, 64Gb capacity common
- Relative to SDR/DDR/DDR2/DDR3: + bandwidth, ~ latency

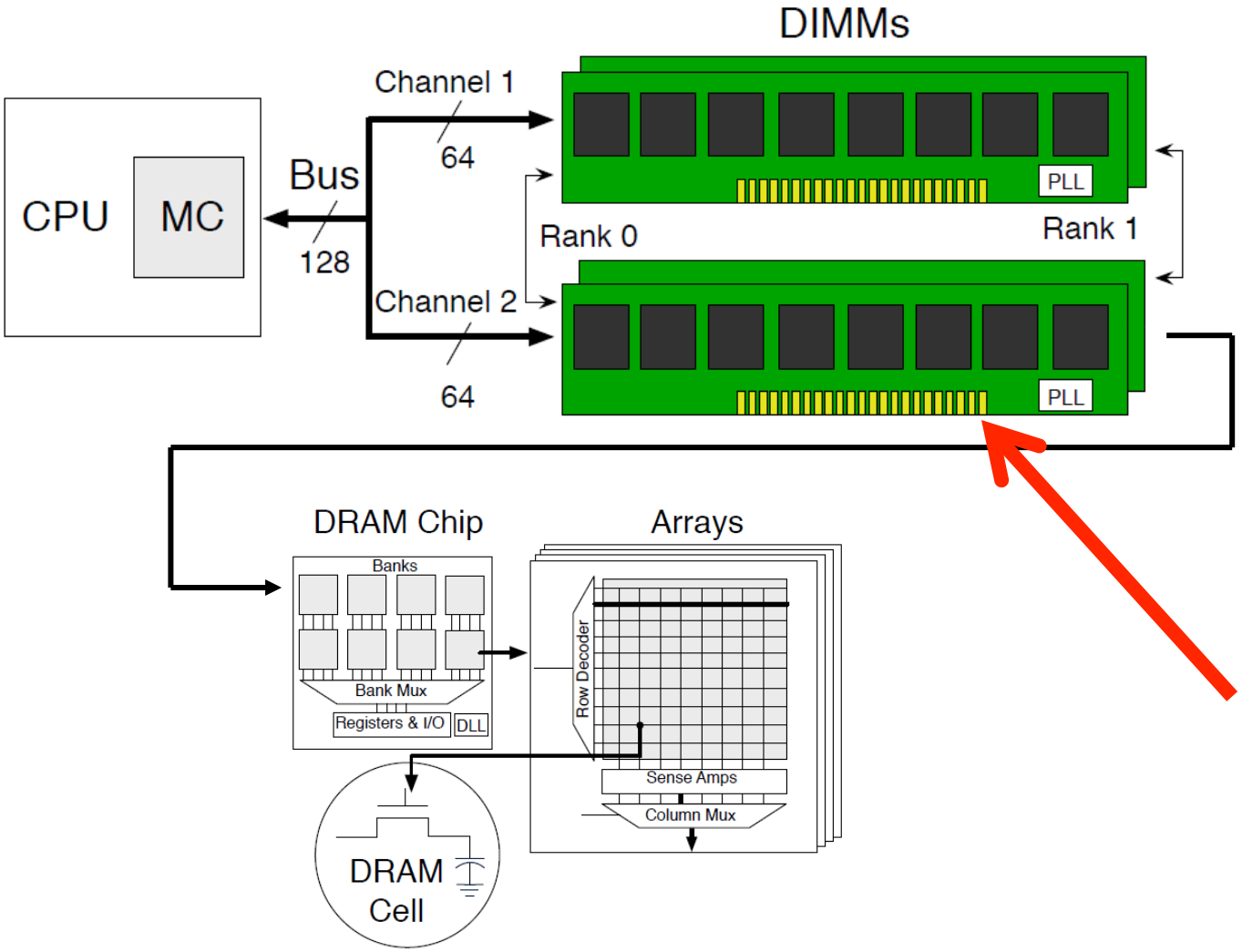
DRAM: Timing Constraints

- Memory controller must respect physical device characteristics
  - **tRCD** = Row to Column command delay
    - How long it takes row to get to sense amps
  - **tCAS** = Time between column command and data out
  - **tCCD** = Time between column commands
    - Rate that you can pipeline column commands
  - **tRP** = Time to precharge DRAM array
  - **tRAS** = Time between RAS and data restoration in DRAM array (minimum time a row must be open)
  - **tRC** = tRAS + tRP = Row "cycle" time
    - Minimum time between accesses to different rows

Latency Components: Basic DRAM Operation

- CPU → memory controller transfer time
- Controller latency
  - Queuing & scheduling delay at the controller
  - Access converted to basic commands
- DRAM bank latency
  - tCAS is row is "open" OR
  - tRCD + tCAS if array precharged OR
  - tRP + tRCD + tCAS (worst case: tRC + tRCD + tCAS)
- DRAM data transfer time
  - BurstLen / (MT/s)
- Memory controller → CPU transfer time

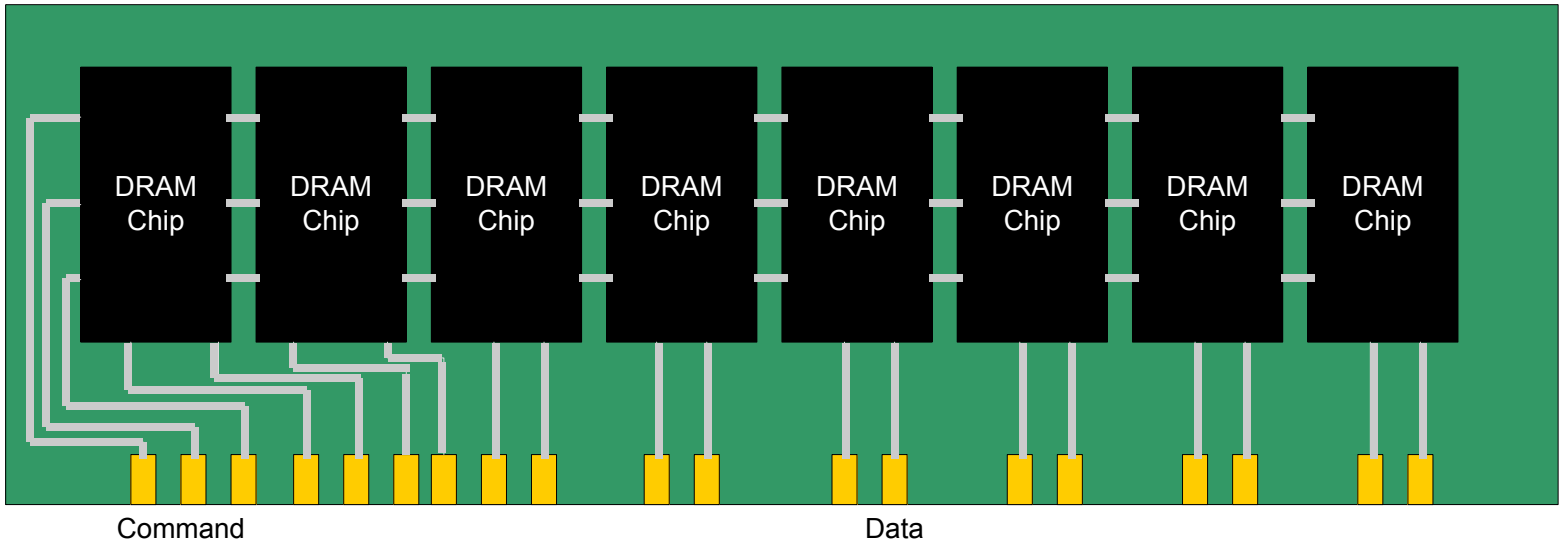
Main Memory Overview



DRAM Modules

- DRAM chips have narrow interface (typically x4, x8, x16)
- Multiple chips are put together to form a wide interface
  - DIMM: Dual Inline Memory Module
  - To get a 64-bit DIMM with x8 DRAM chips, we need to access 8 chips
  - Share command/address lines, but not data
- Advantages
  - Acts like a high-capacity DRAM chip with a wide interface
    - 8x capacity, 8x bandwidth, same latency
- Disadvantages
  - Granularity: Accesses cannot be smaller than the interface width
  - 8x power

A 64-bit Wide DIMM
(physical view)

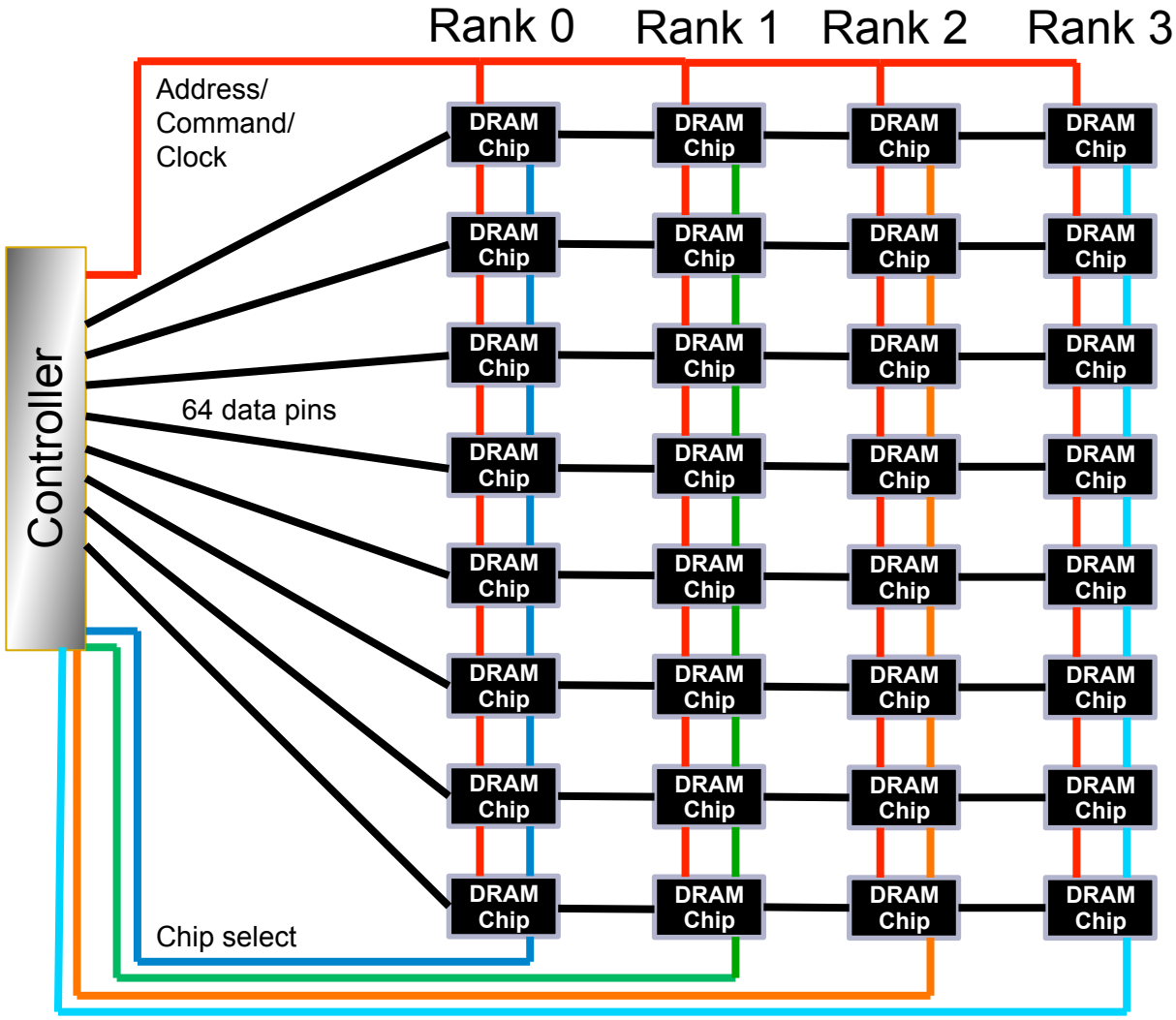


DIMM Capacity

- 64-bit interface
- 2Gb x8 chips
- 64/8 = 8 chips
- 2Gb x 8 = 2GB

- What if we want more capacity on a channel?
  - Option 1: Use narrower chips (i.e. x4)
  - Option 2: Ranks

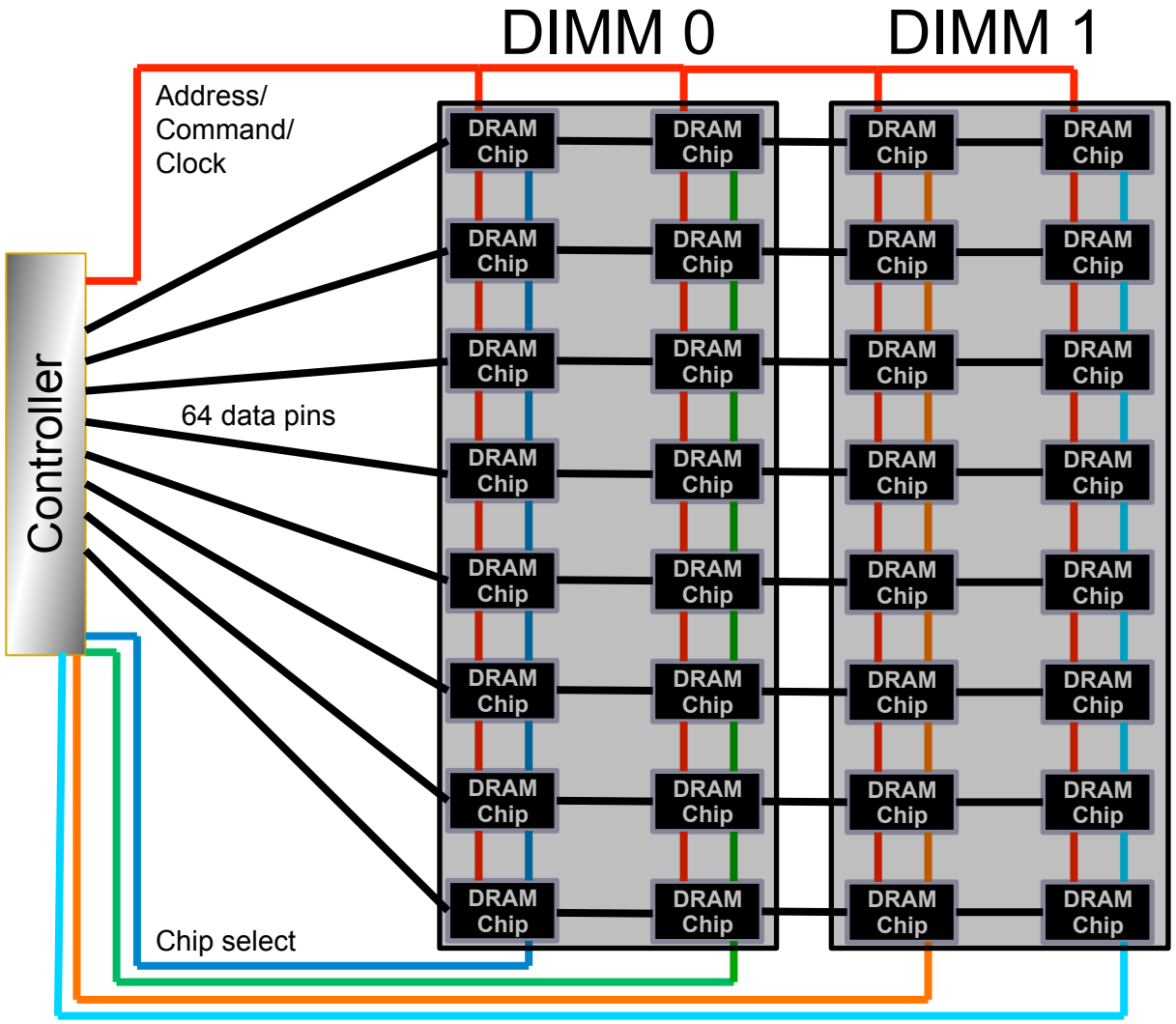
Ranks


Rank 0
Rank 1
Rank 2
Rank 3
Address/
Command/
Clock
Controller
64 data pins
Chip select
DRAM Chip

Ranks

- A DIMM may include multiple Ranks
  - A 64-bit DIMM with 16 chips with x8 interfaces has 2 ranks
- Each 64-bit group of chips is called a rank
  - All chips in a rank respond to a single command
  - Different ranks share command/address/data lines
    - Select between ranks with “Chip Select” signal
  - Ranks provide more “banks” across multiple chips (but don’t confuse rank and bank!)

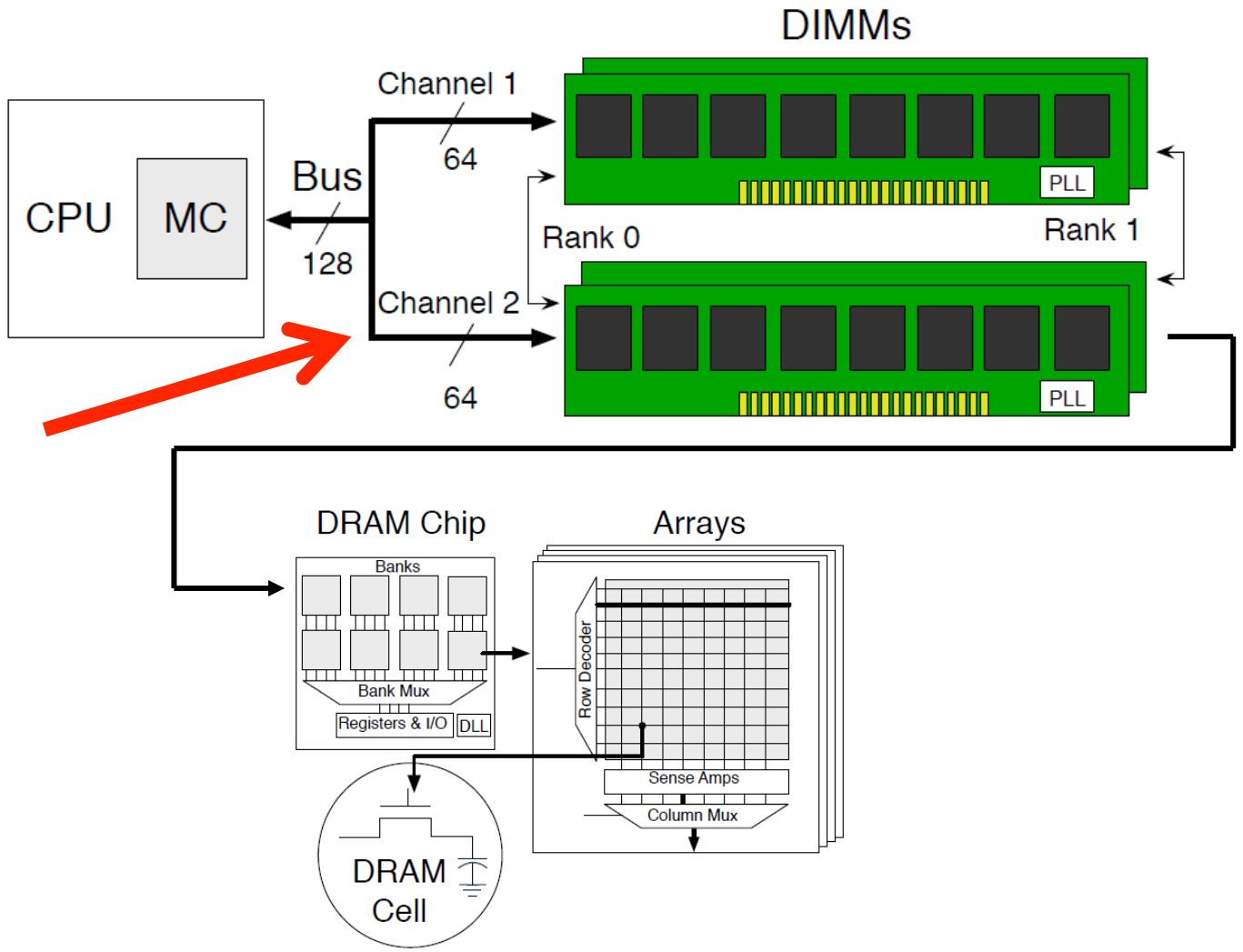
Multiple DIMMs on a Channel



DRAM Capacity

- What if we want even more capacity?
  - Can only put limited number of ranks on channel (signal and driver limitations)
- x2 chips? x1 chips?
  - What happens to power consumption?
- More channels?

Main Memory Overview



DRAM Channels

- Channel: a set of DIMMs in series
  - All DIMMs get the same command, one of the ranks replies
- Multiple-channel options
  - Multiple lock-step channels
    - Single “channel” with wider interface (faster cache line refill!)
    - Sometimes called “Gang Mode”
    - Only works if DIMMs are identical (organization, timing)
  - Multiple independent channels
    - Requires multiple controllers
- Tradeoffs in having multiple channels
  - Cost: pins, wires, controllers
  - Benefit: higher bandwidth, capacity

DRAM Channel Options