18-640 Foundations of Computer Architecture

Lecture 9:
"Main Memory System Design"

John Paul Shen
September 23, 2014

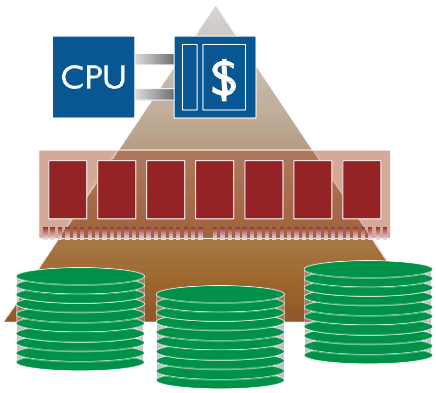
- Required Reading Assignment:
  - Sec. 1 & Sec. 3: Bruce Jacob, "The Memory System: You Can't Avoid It, You Can't Ignore It, You Can't Fake It," Synthesis Lectures on Computer Architecture 2009.
- Recommended Reference:
  - Benjamin C. Lee, Ping Zhou, Jun Yang, Youtao Zhang, Bo Zhao, Engin Ipek, Onur Mutlu, Doug Burger, "Phase-Change Technology and the Future of Main Memory," IEEE Micro, vol. 30, no. 1, pp. 143-143, Jan./Feb. 2010.

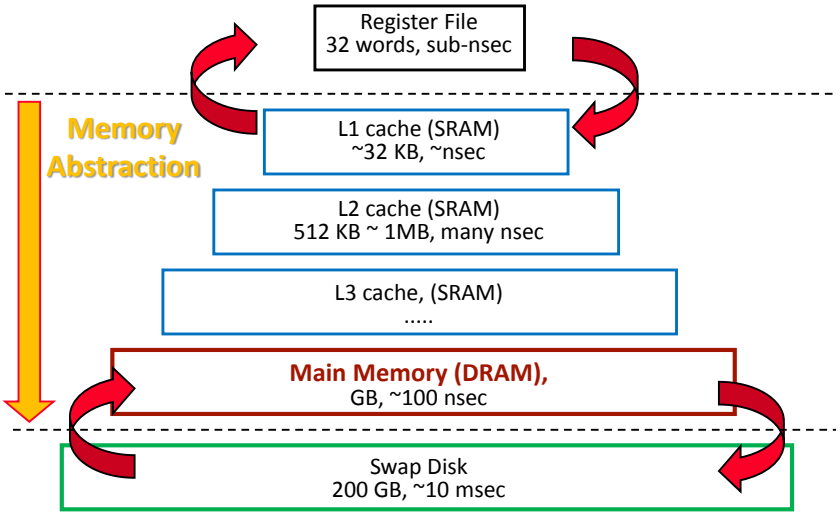
18-640 Foundations of Computer Architecture

Lecture 9:
"Main Memory System Design"

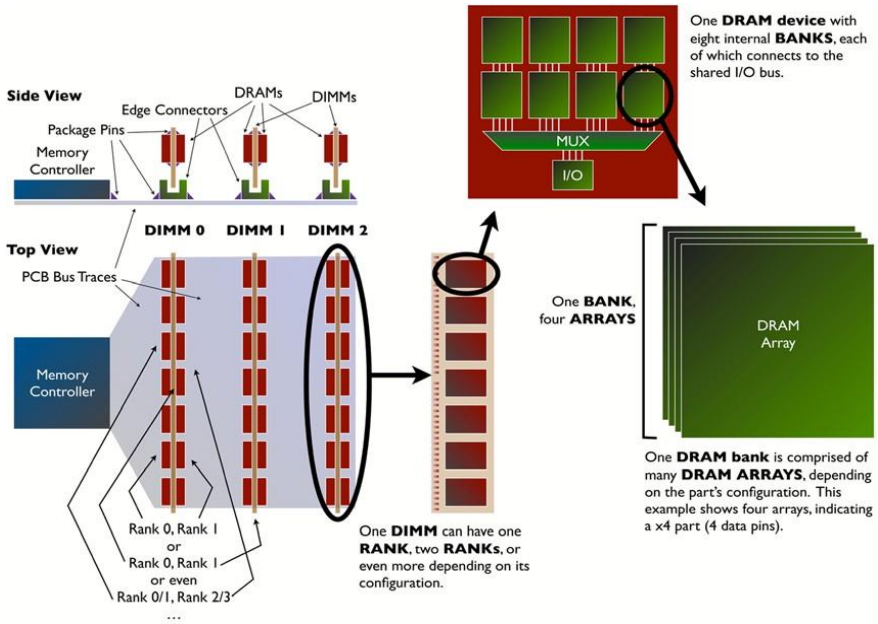
A. Main Memory Implementation
B. DRAM Organization
C. DRAM Operation
D. Memory Controller
E. Emerging Technologies

Memory Hierarchy



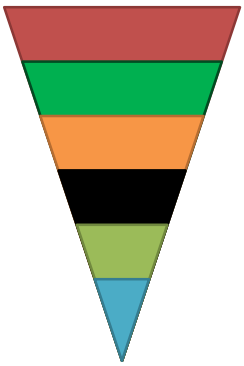


A. Main Memory Implementation

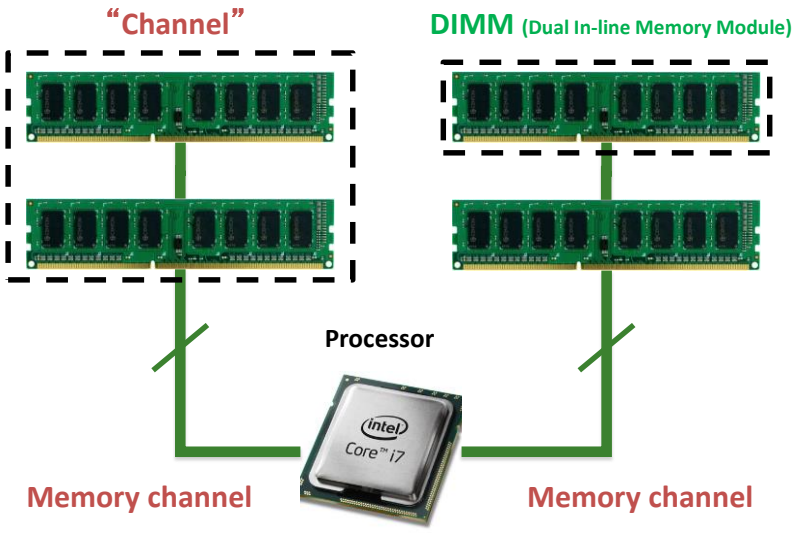


DRAM Subsystem Organization (Top Down View)

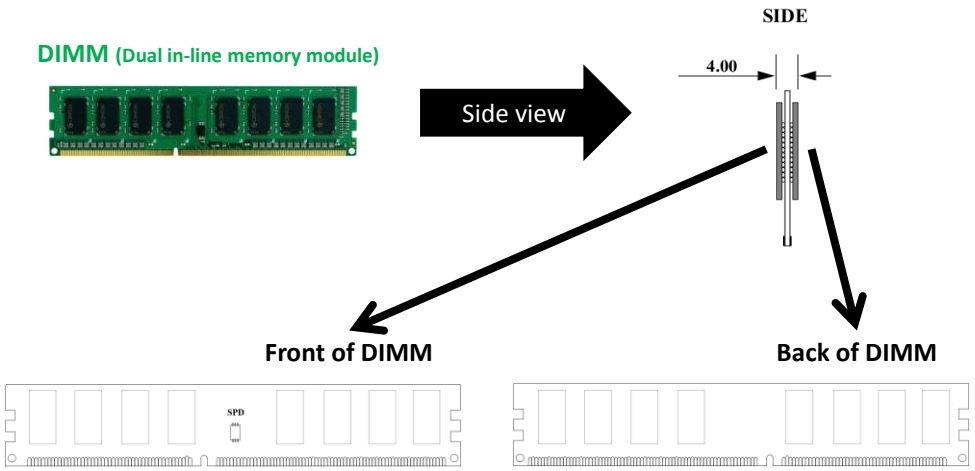
- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column



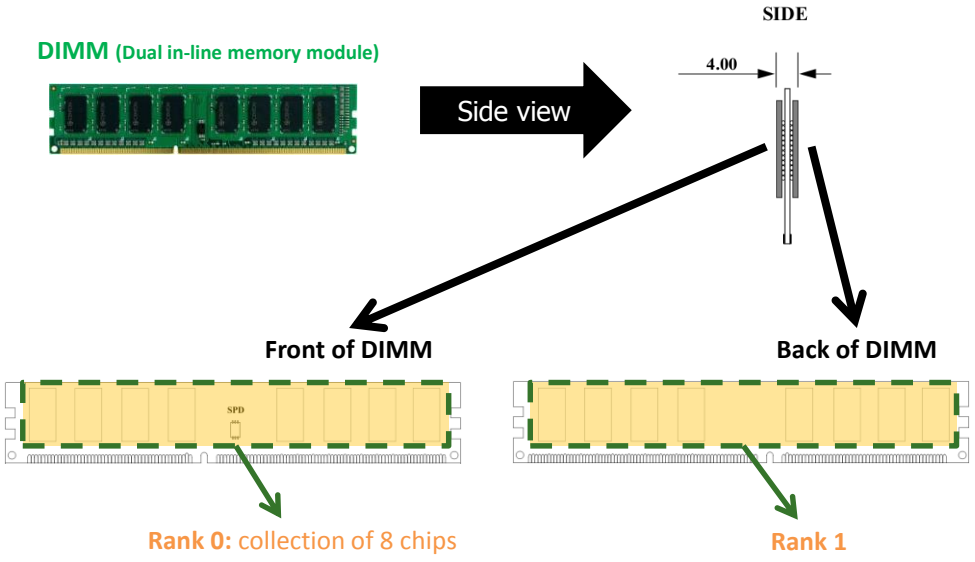
The DRAM Subsystem



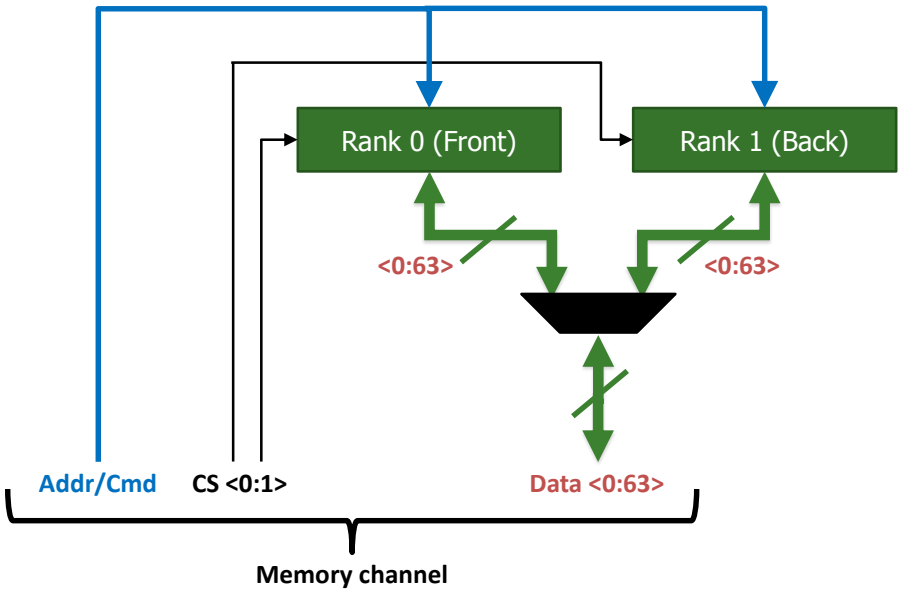
Breaking Down a DIMM



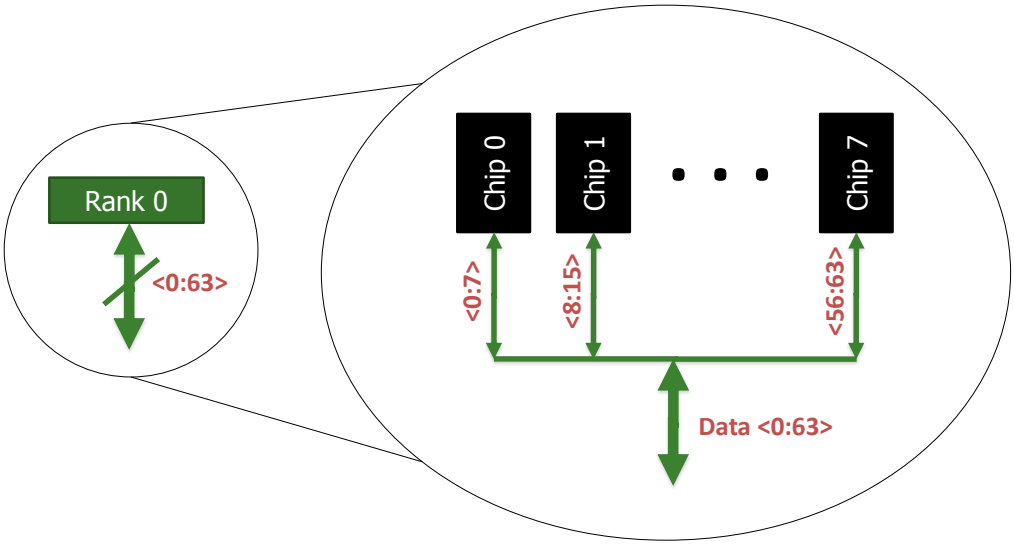
Breaking Down a DIMM



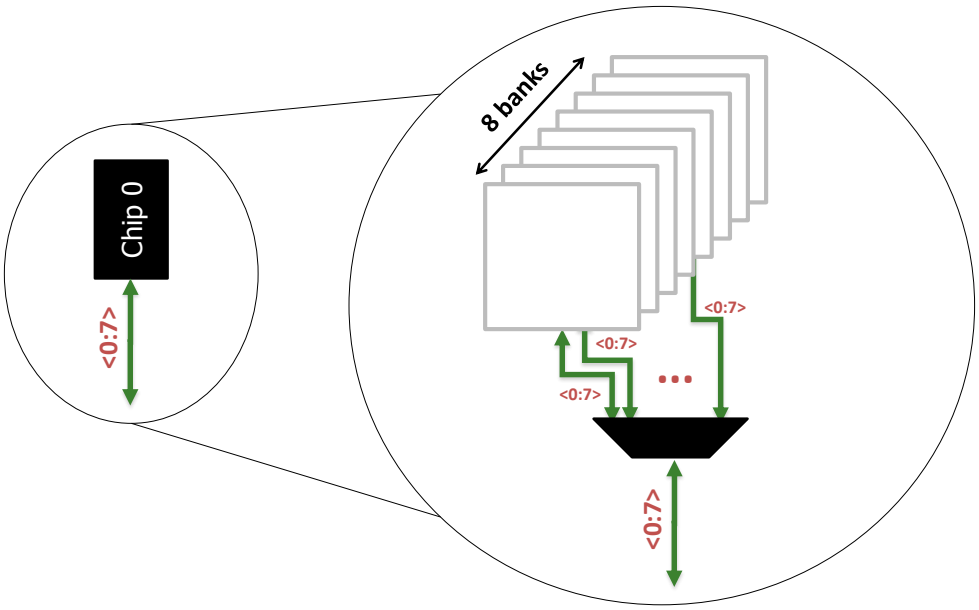
Rank



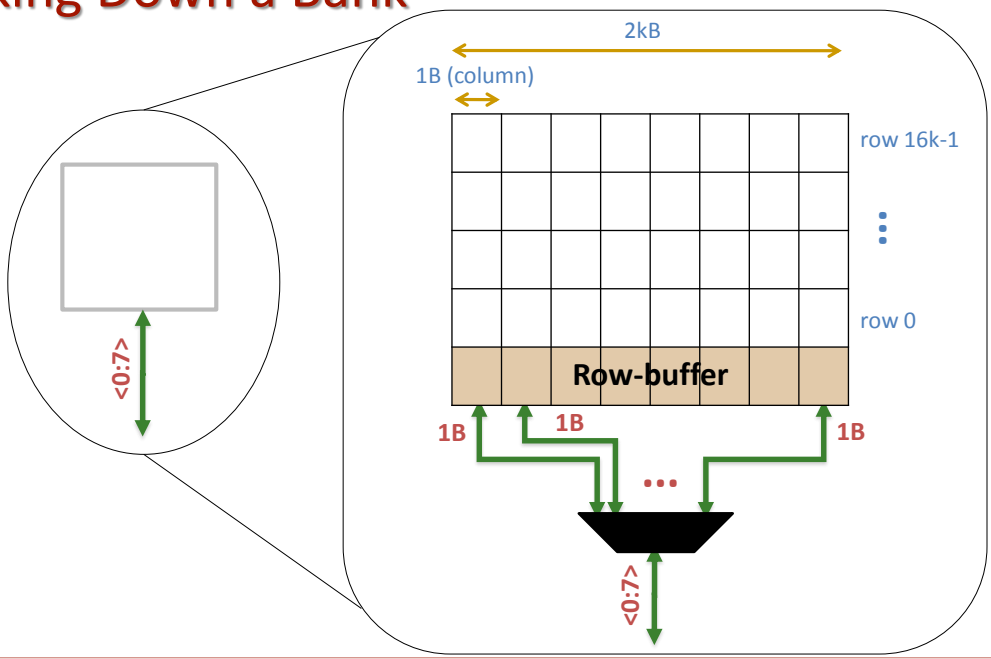
Breaking Down a Rank



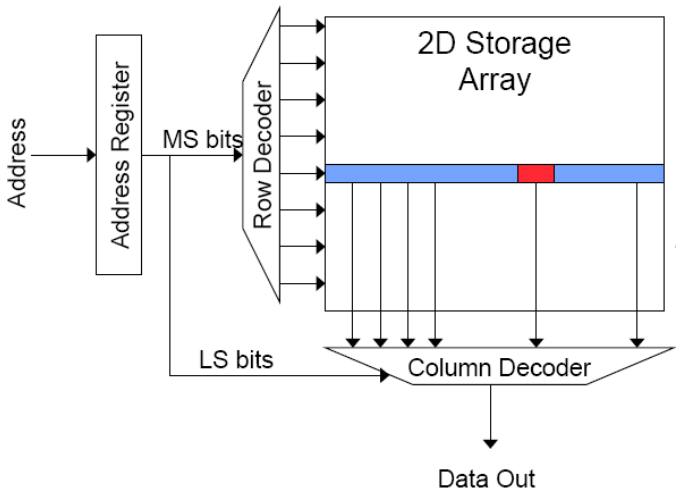
Breaking Down a Chip



Breaking Down a Bank

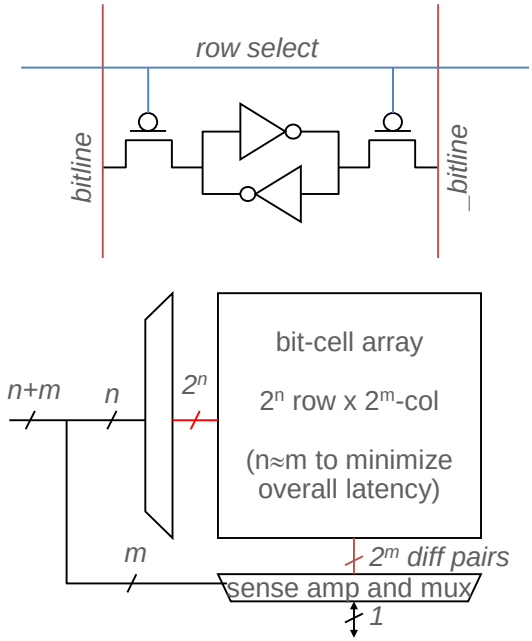


Memory Bank Organization



- Read access sequence:
  1. Decode row address & drive word-line
  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)

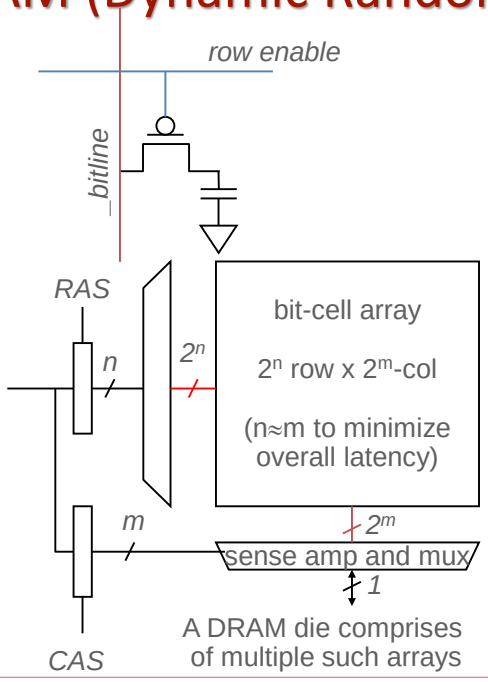


Read Sequence
1. address decode
2. drive row select
3. selected bit-cells drive bitlines (entire row is read together)
4. diff. sensing and col. select (data is ready)
5. precharge all bitlines (for next read or write)

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)



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

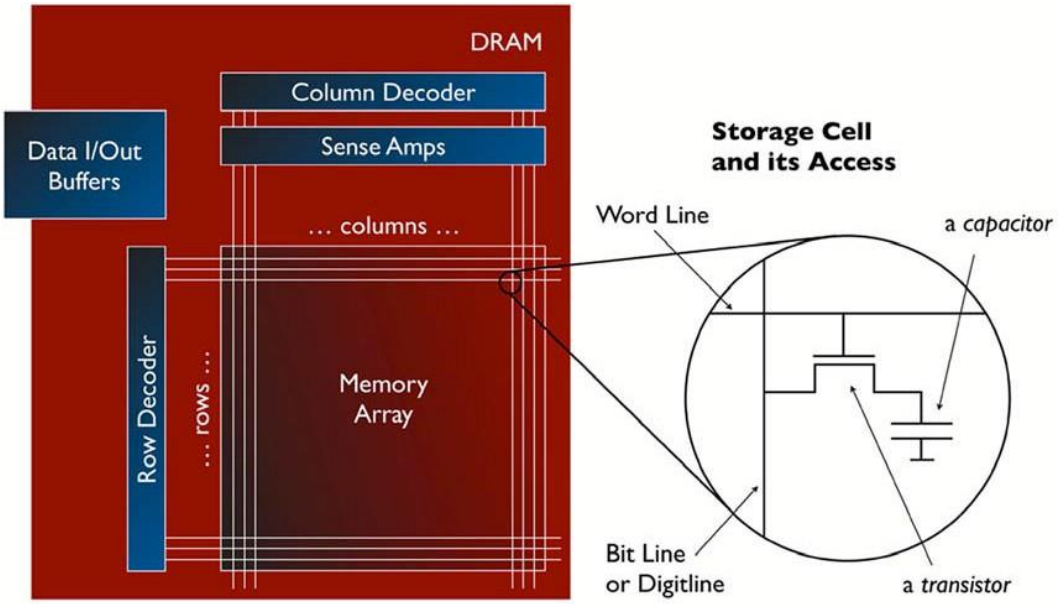
4. a "flip-flopping" sense amp amplifies and regenerates the bitline, data bit is mux'ed out
5. precharge all bitlines

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

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)

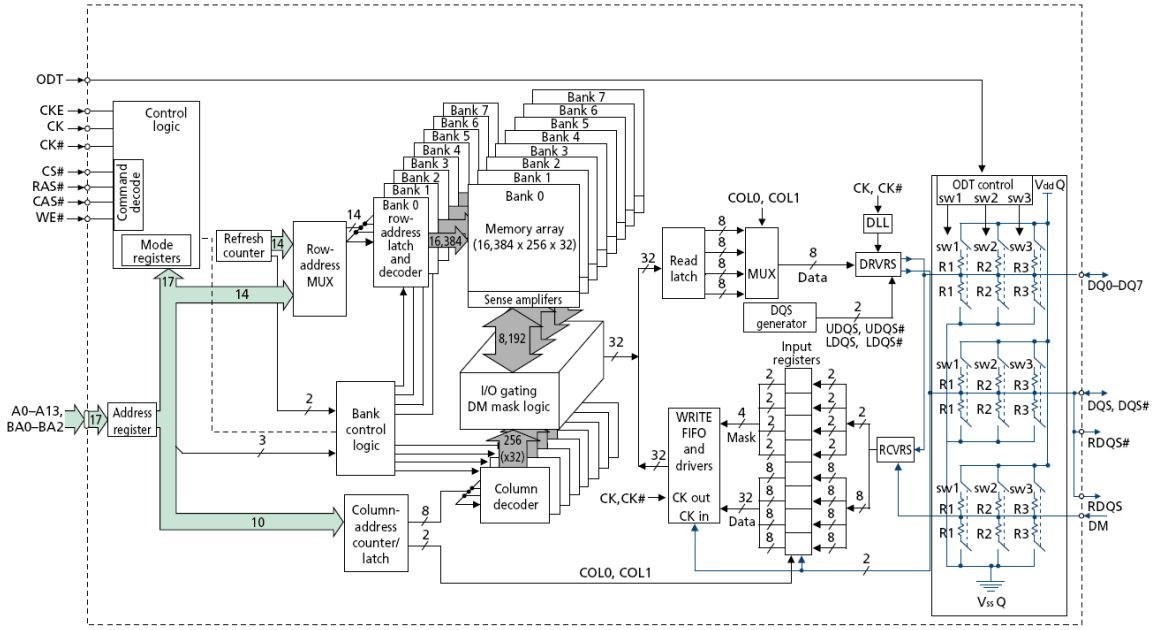
B. DRAM Organization



The DRAM 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)

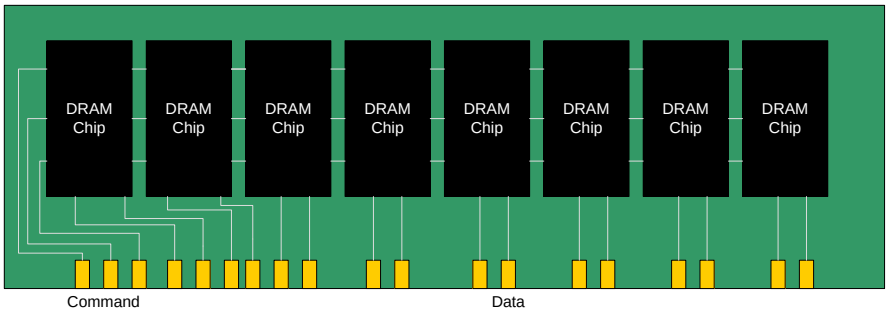
128M x 8-bit DRAM Chip



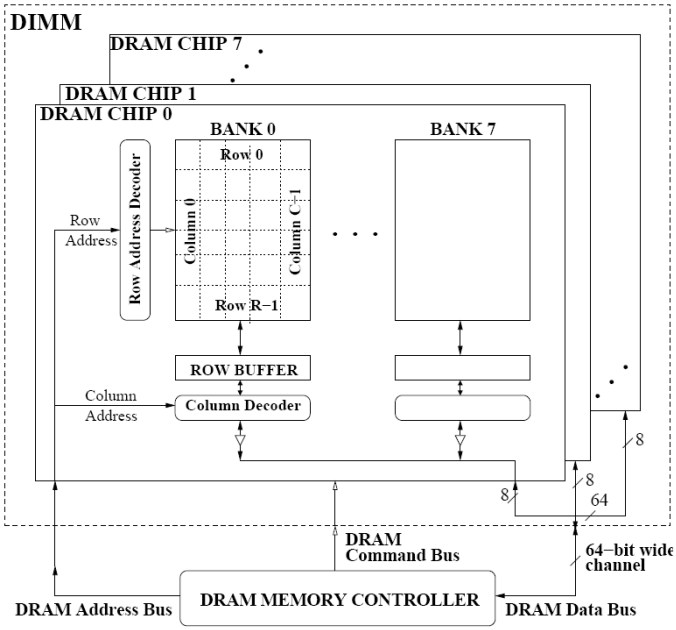
DRAM Rank and Module

- 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)

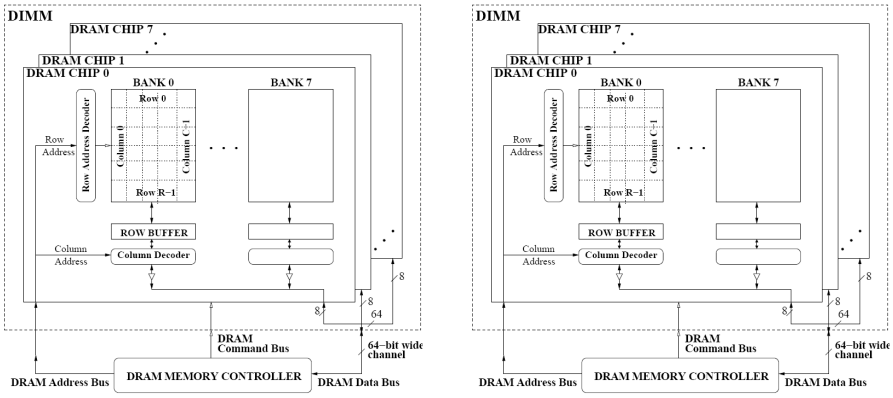


A 64-bit Wide DIMM (One Rank)



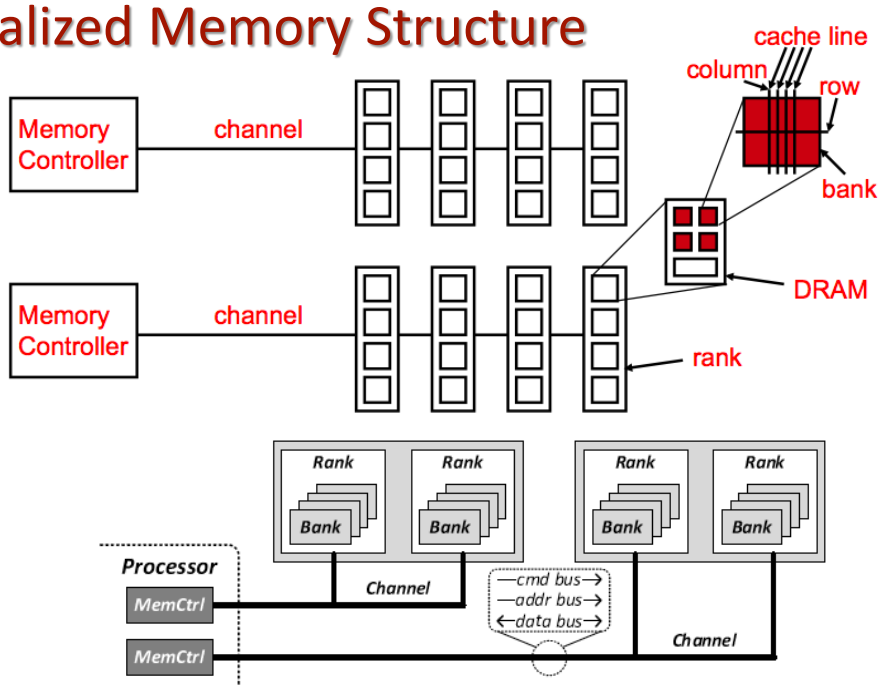
- Advantages:
  - Acts like a high-capacity DRAM chip with a wide interface
  - Flexibility: memory controller does not need to deal with individual chips
- Disadvantages:
  - Granularity: Accesses cannot be smaller than the interface width

DRAM Channels



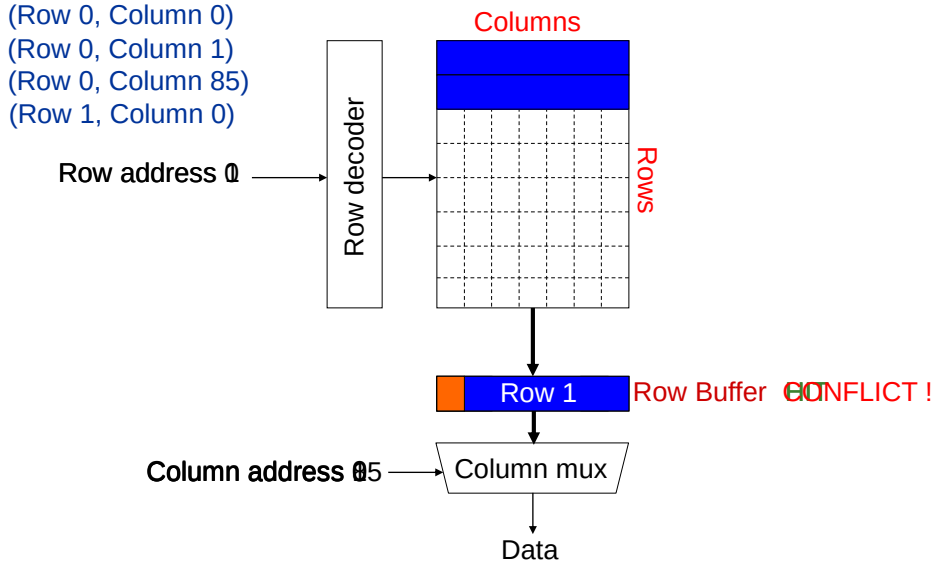
- 2 Independent Channels: 2 Memory Controllers (Above)
- 2 Dependent/Lockstep Channels: 1 Memory Controller with wide interface (Not Shown above)

Generalized Memory Structure



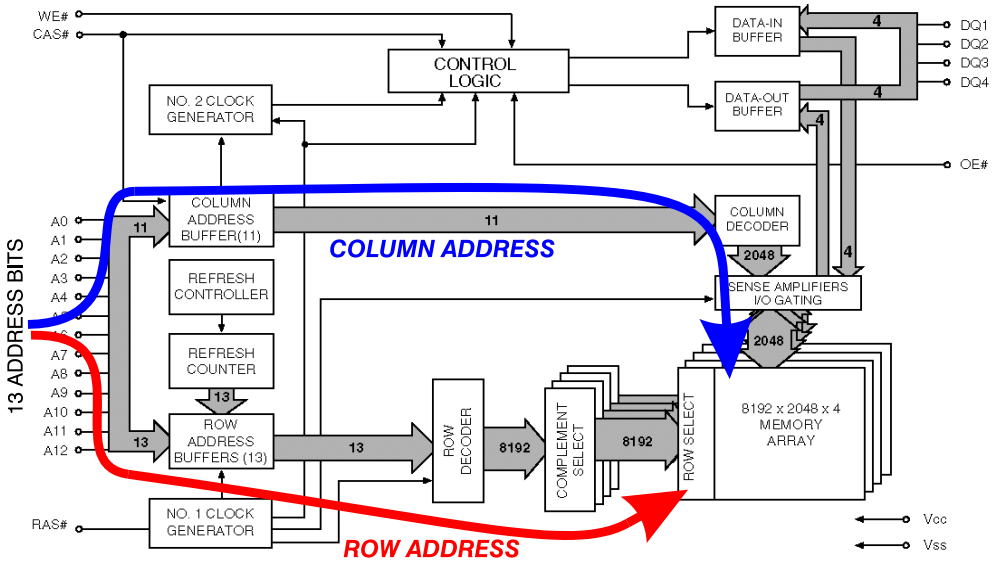
C. DRAM (Bank) Operation

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



Conventional 64MbitDRAM Example from Micron

FUNCTIONAL BLOCK DIAGRAM
MT4LC16M4A7 (13 row addresses)

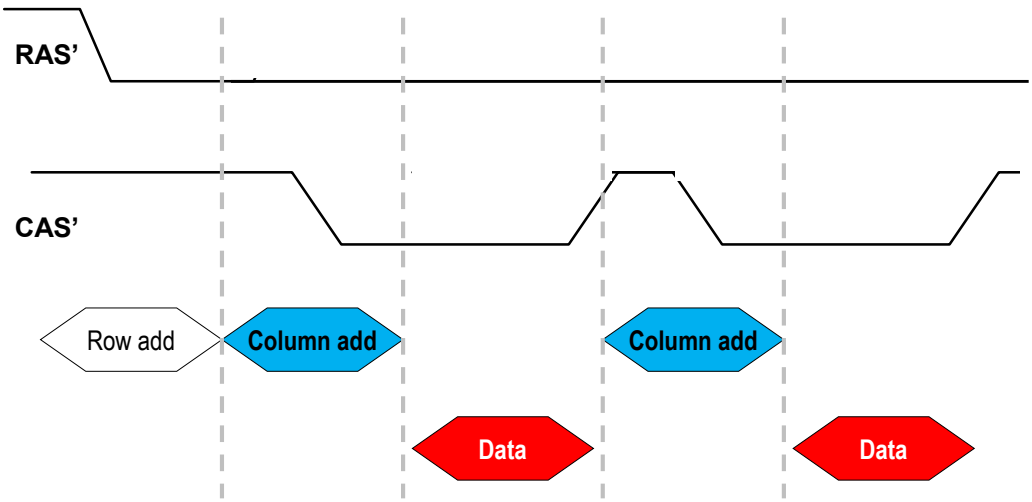


Page Mode DRAM

- 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"

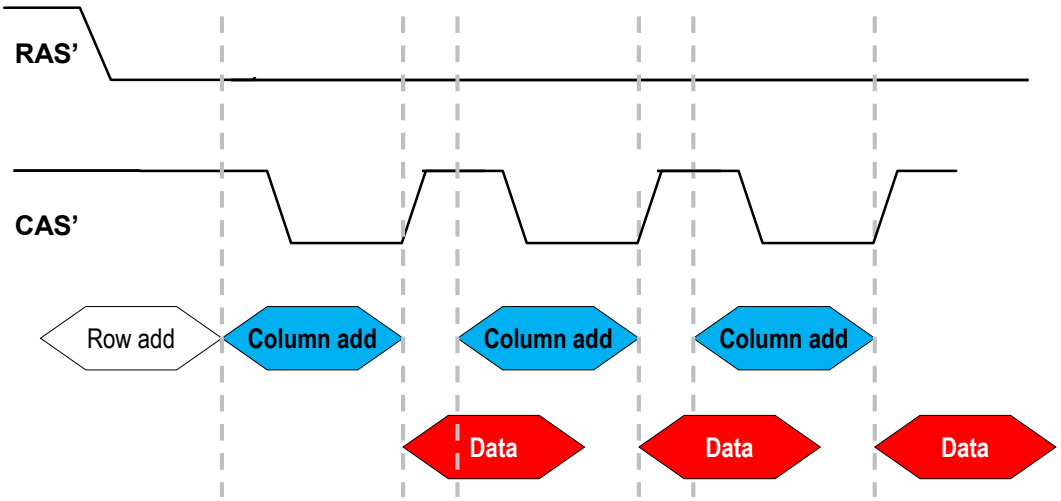
- Each address is a <row,column> pair
- Access to a "closed row"
  - Activate command opens row (placed into row buffer)
  - Read/write command reads/writes column in the row buffer
  - Precharge command closes the row and prepares the bank for next access
- Access to an "open row"
  - No need for activate command

Fast Page Mode (FPM)



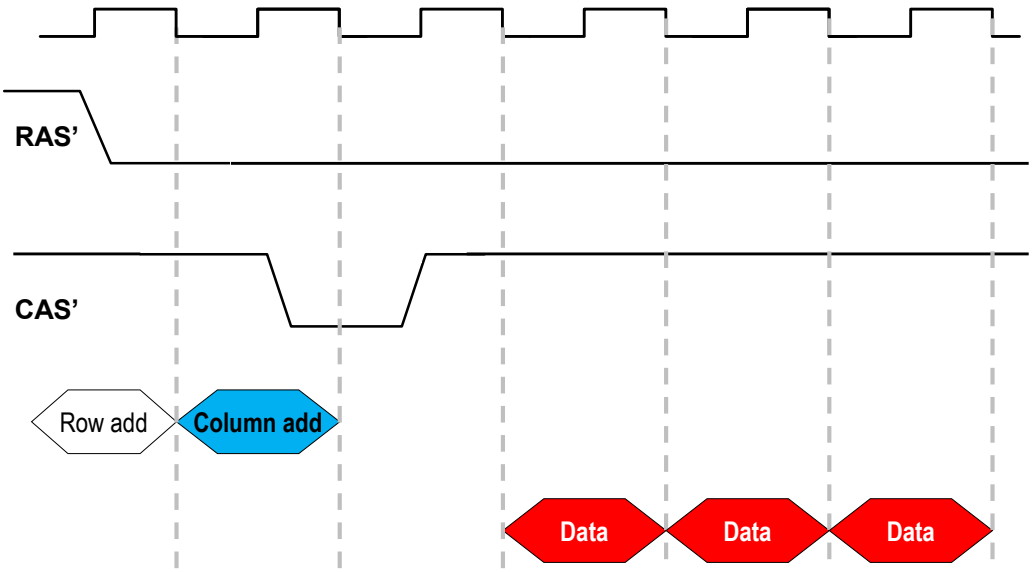
- One row address
- Multiple column addresses

Extended Data Out (EDO)



- As in FPM
- But overlapped Column Address assert with Data Out

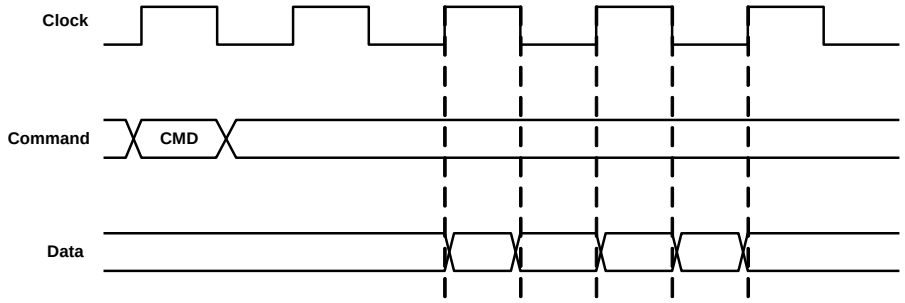
Synchronous DRAM (SDRAM)



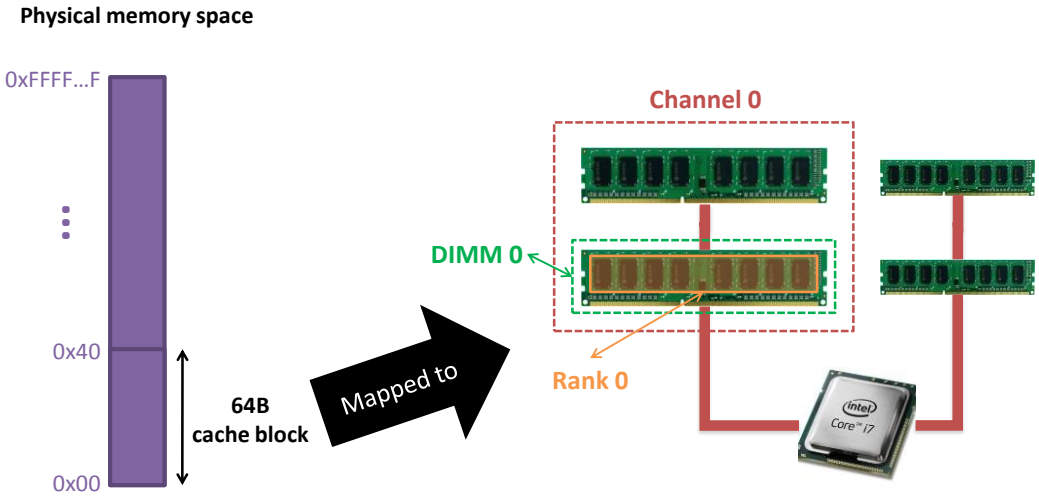
- Single CAS Strobe, multiple transfers

DDR SDRAM Timing

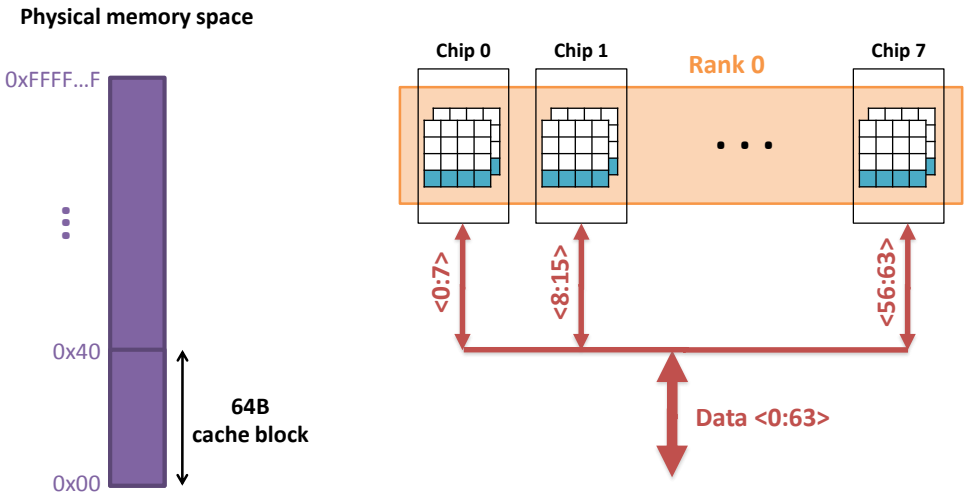
- **Read access**



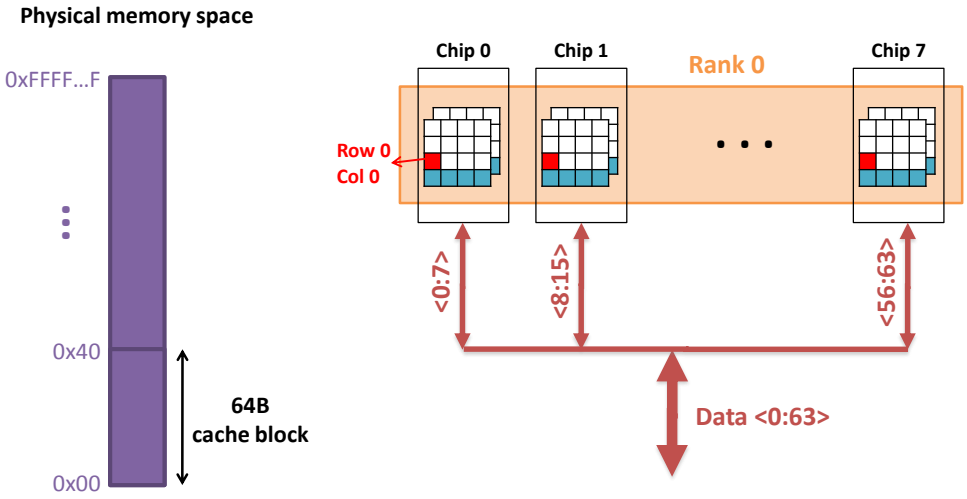
Example: Transferring a Cache Block



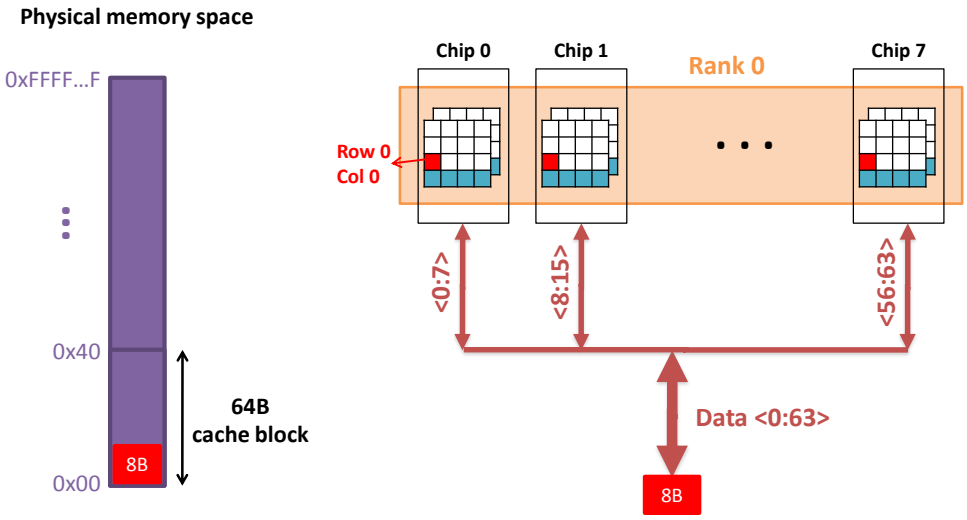
Example: Transferring a Cache Block



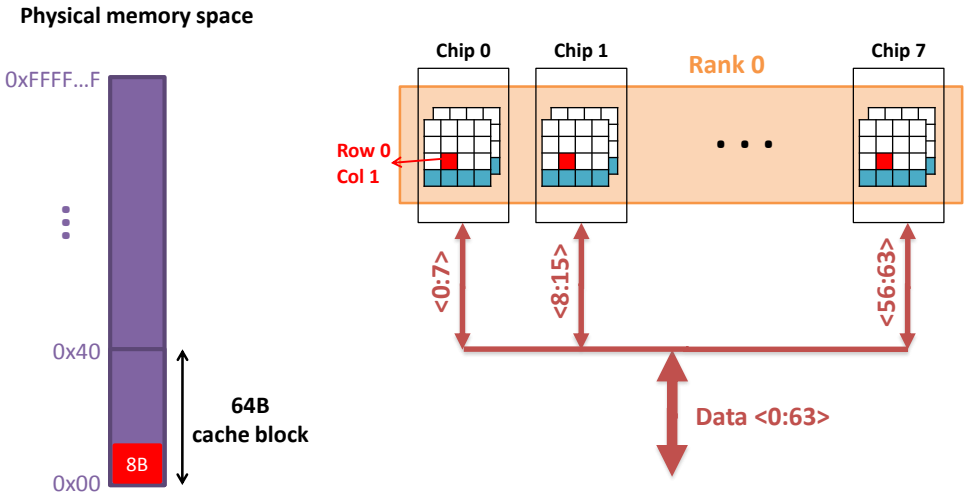
Example: Transferring a Cache Block



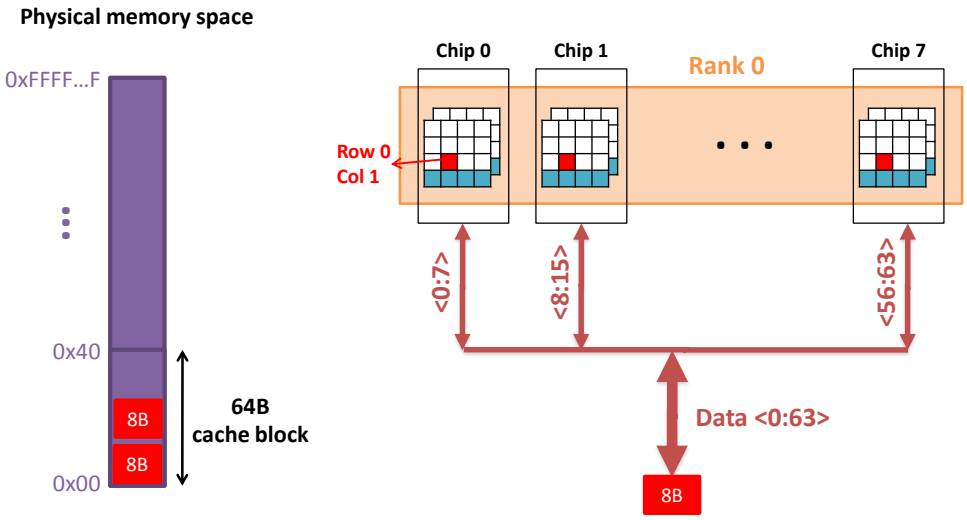
Example: Transferring a Cache Block



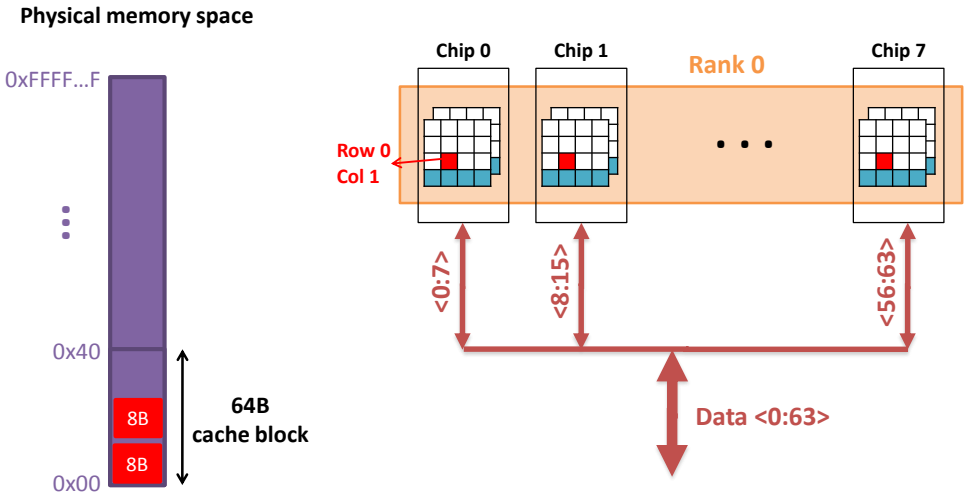
Example: Transferring a Cache Block



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.

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 if row is "open" OR
  - RAS + CAS if array precharged OR
  - PRE + RAS + CAS (worst case)
- DRAM → CPU transfer time (through controller)

Simple Main Memory

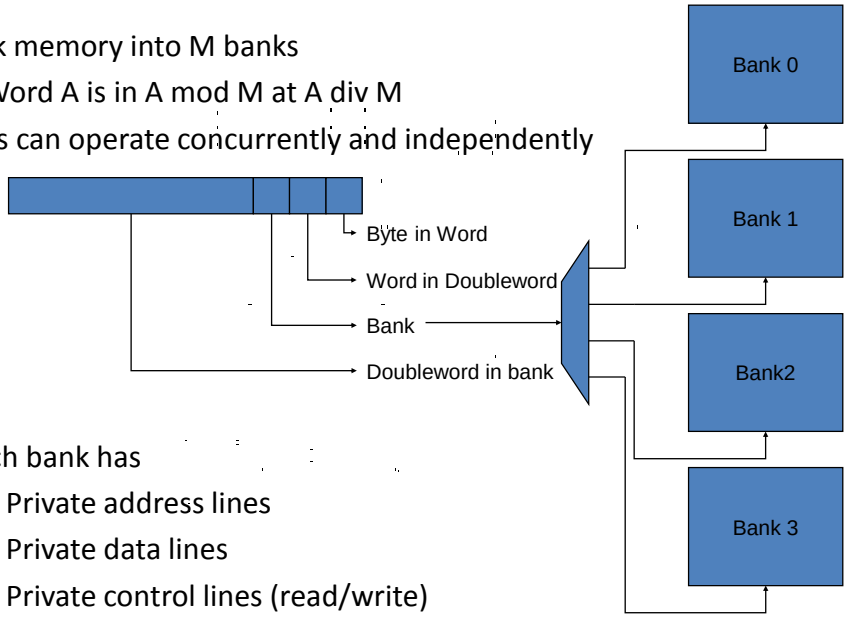
- Consider these parameters:
  - 10 cycles to send address
  - 60 cycles to access each word
  - 10 cycle to send word back
- Miss penalty for a 4-word block
  - (10 + 60 + 10) x 4 = 320
- How can we speed this up?

Wider (Parallel) Main Memory

- Make memory wider
  - Read out all words in parallel
- Memory parameters
  - 10 cycles to send address
  - 60 cycles to access a double word
  - 10 cycles to send it back
- Miss penalty for 4-word block: 2x(10+60+10) = 160
- Costs
  - Wider bus
  - Larger minimum expansion unit (e.g. paired DIMMs)

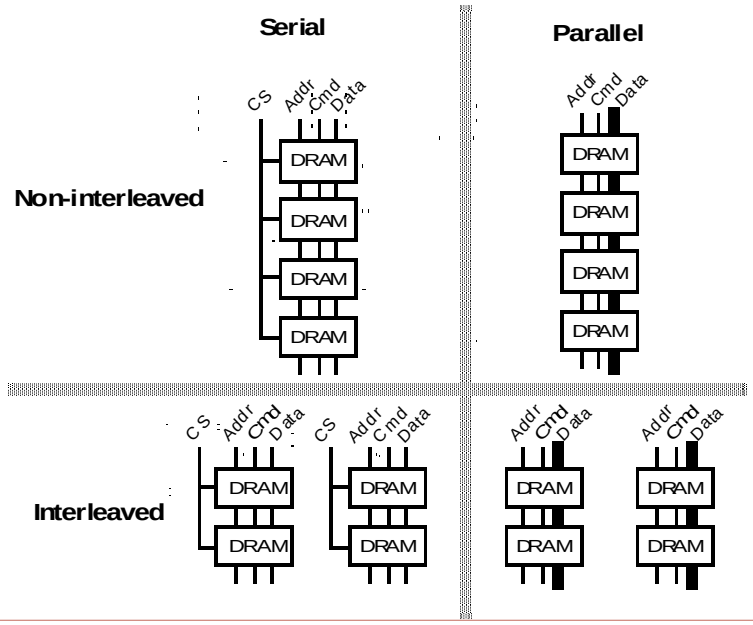
Interleaved Main Memory

- Break memory into M banks
  - Word A is in A mod M at A div M
- Banks can operate concurrently and independently



  - Each bank has
    - Private address lines
    - Private data lines
    - Private control lines (read/write)

Interleaved and Parallel Organization

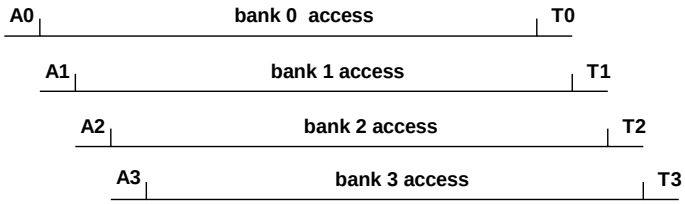


Interleaved Memory Examples

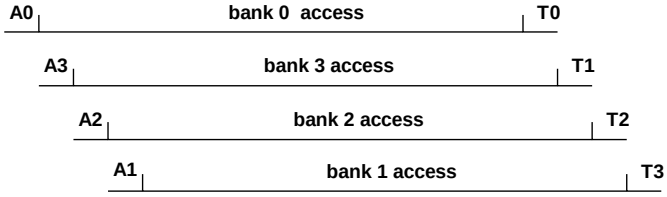
Ai = address to bank i

Ti = data transfer

- Unit Stride:



- Stride 3:



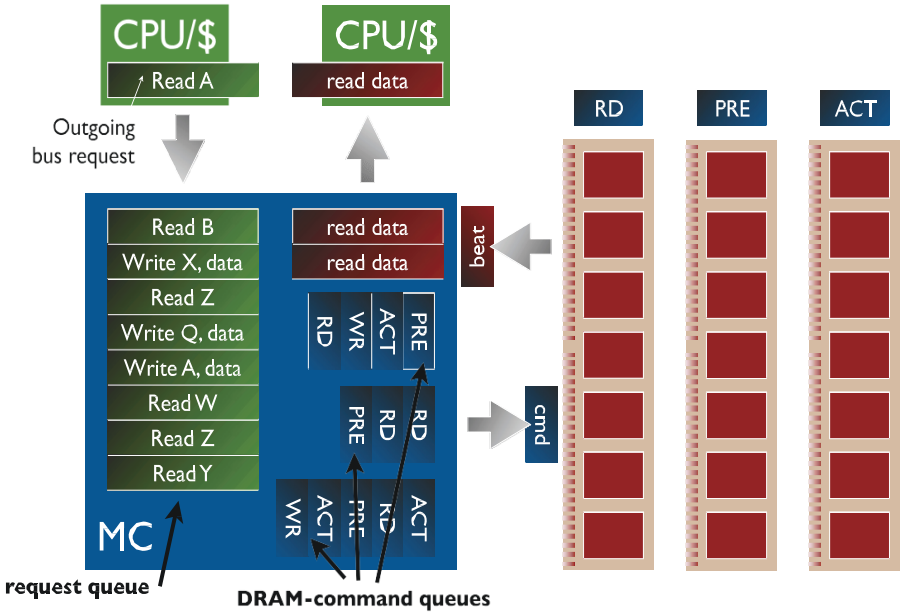
Interleaved Memory Summary

- Parallel memory adequate for sequential accesses
  - Load cache block: multiple sequential words
  - Good for writeback caches
- Banking useful otherwise
  - If many banks, choose a prime number
- Can also do both
  - Within each bank: parallel memory path
  - Across banks
    - Can support multiple concurrent cache accesses (nonblocking)

Constructing a Memory System

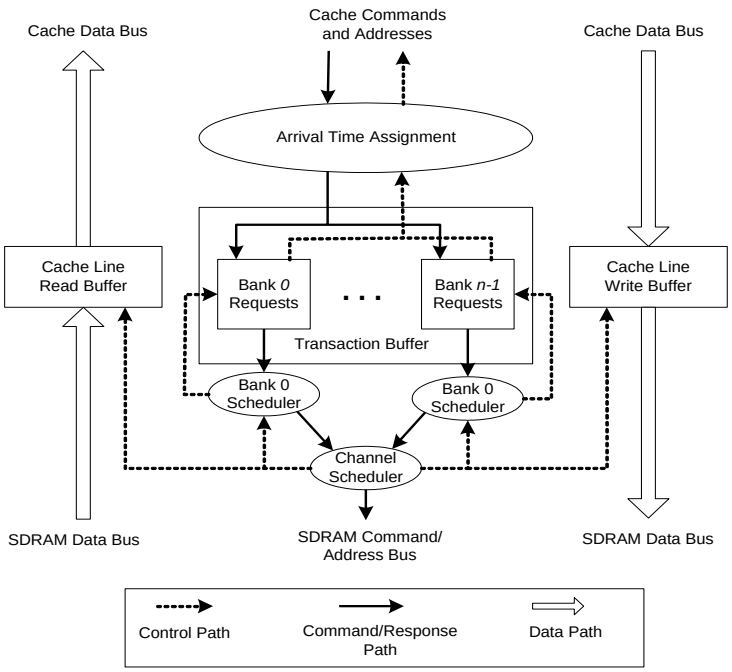
- Combine chips in parallel to increase access width
  - E.g. 8 8-bit wide DRAMs for a 64-bit parallel access
  - DIMM – Dual Inline Memory Module
- Combine DIMMs to form multiple *ranks*
- Attach a number of DIMMs to a memory channel
  - Memory Controller manages a channel (or two lock-step channels)
- Interleave patterns:
  - Rank, Row, Bank, Column, [byte]
  - Row, Rank, Bank, Column, [byte]
    - Better dispersion of addresses
    - Works better with power-of-two ranks

D. Memory Controller



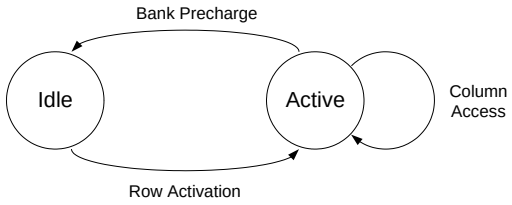
Memory Controllers

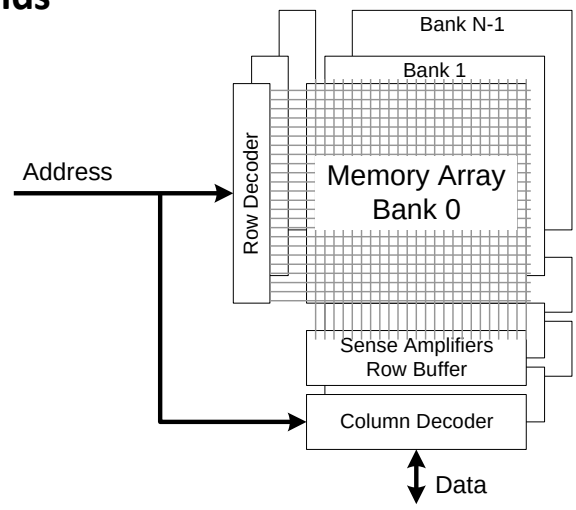
- Contains buffering
  - **In both directions**
- Schedulers manage resources
  - **Channel and banks**



DDR SDRAM Control

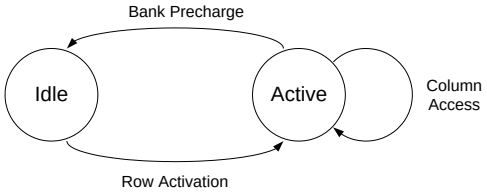
- **Raise level of abstraction: commands**
  - Activate row
    - Read row into row buffer
  - Column access
    - Read data from addressed row
  - Bank Precharge
    - Get ready for new row access

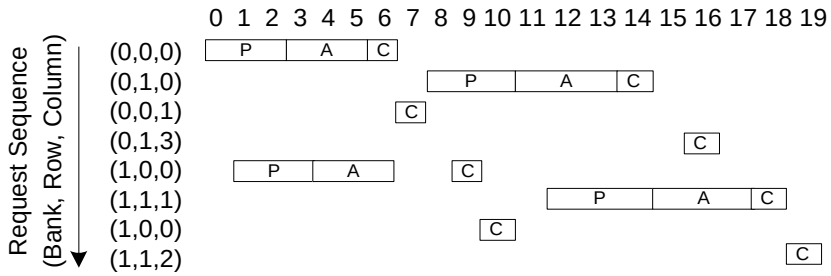




Resource Scheduling

- An interesting optimization problem
- Example:
  - **Precharge: 3 cycles**
  - **Row activate: 3 cycles**
  - **Column access: 1 cycle**
  - **FR-FCFS: 20 cycles**
  - **StrictFIFO: 56 cycles**





P: bank *P*recharge
A: row *A*ctivation
C: *C*olumn Access

DDR SDRAM Policies

- Goal: try to maximize requests to an open row (page)
- Close row policy
  - Always close row, hides precharge penalty
  - Lost opportunity if next access to same row
- Open row policy
  - Leave row open
  - If an access to a different row, then penalty for precharge
- Also performance issues related to rank interleaving
  - Better dispersion of addresses

Memory Scheduling Contest

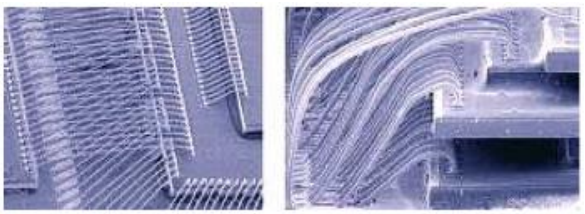
- http://www.cs.utah.edu/~rajeev/jwac12/
- Clean, simple, infrastructure
- Traces provided
- Very easy to make fair comparisons
- Comes with 6 schedulers
- Also targets power-down modes (not just page open/close scheduling)
- Three tracks:
  1. Delay (or Performance),
  2. Energy-Delay Product (EDP)
  3. Performance-Fairness Product (PFP)

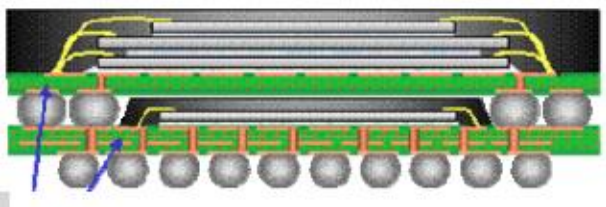
E. Emerging Technologies

- 3D integration
  - Can help increase bandwidth and reduce latency & cost
- Non-volatile memories
  - Can help address SRAM/DRAM shortcomings
  - New features based on non-volatility

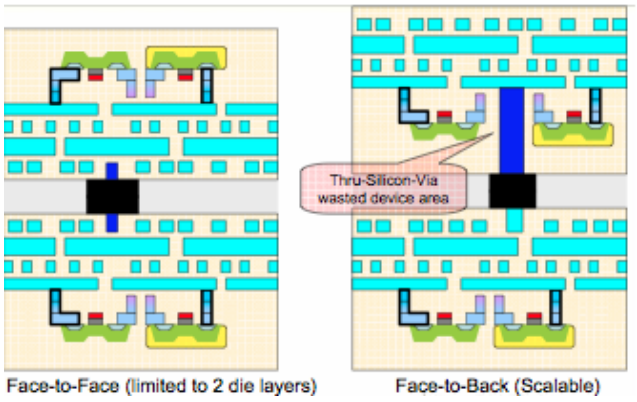
3D Integration: Existing Approaches

- SIP: system-in-package
  - Wire-bonding 3D stacking
- PoP: package-on-package
- Both are popular now
  - E.g., for cellphones





3D Integration with Through-Silicon Vias (TSVs)



- TSVs allow ICs with multiple layers of transistors
  - Via sizes are scaling from ~50um toward <1um pitch
- Options
  - Face-to-face vs face-to-back stacking
  - Wafer vs chip-stacking

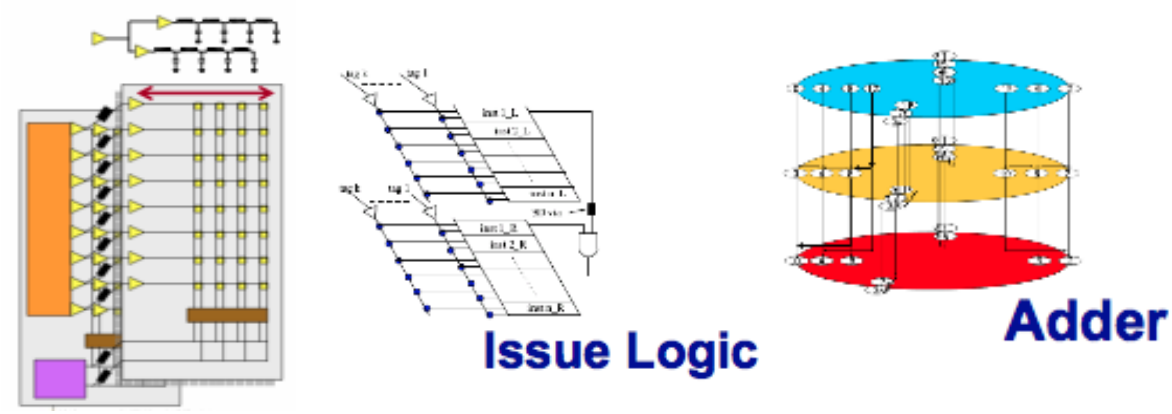
Advantages of 3D

- Area density
  - More functionality for given footprint
- Higher bandwidth
  - More connections between chips
- Lower latency
  - Shorter connections between chips
- Lower interconnect power
  - Due to shorter wires
- Yield
  - One large chip has lower yield than two smaller
- Heterogeneous integration
  - Mix logic, DRAM, and analog chips from separate processes

Disadvantages of 3D

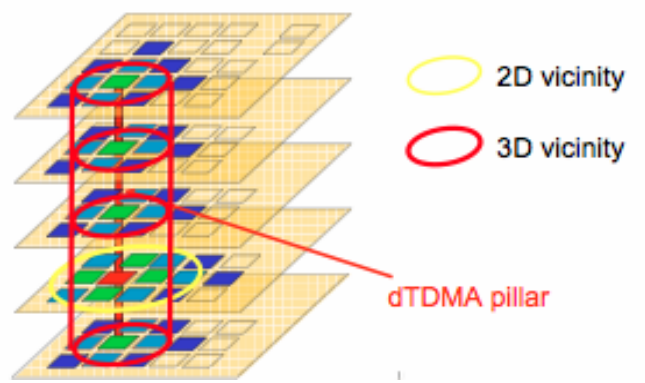
- Cost of 3D integration
  - Extra manufacturing steps
- Yield
  - Each extra step introduces risks
  - Dependencies within stack
- Power density
  - Removing heat from the middle of the stack
  - Heat interference between devices
- Design and testing issues

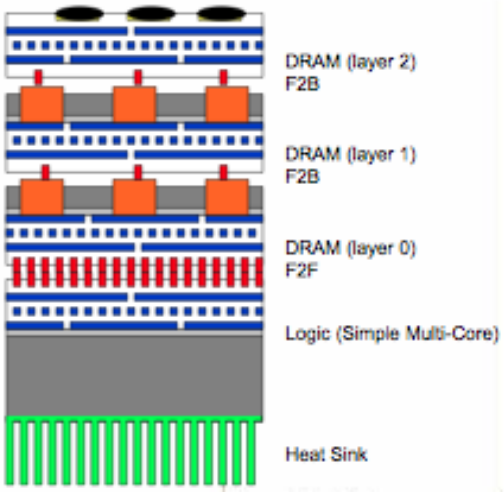
Architectural Uses of 3D



- Stacking array based components
  - E.g., 3D SRAMs and DRAMs (word line spliting, 3D ports)
  - E.g., 3D functional units (adders, shifters, …)
  - Faster/lower power OR larger/more complex

Architectural Uses of 3D



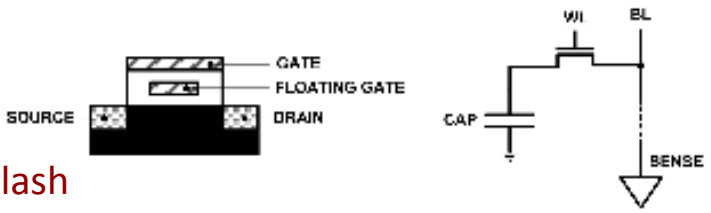


- Stacking distinct components
  - E.g., processor with cache banks on top
  - E.g., processors with DRAM banks on top
    - Lower latency and higher bandwidth to memory
  - E.g., processor with analog I/O circuitry

3D Implications

- How would you build a memory hierarchy given 3D integration?
- How would you build a main memory system given 3D integration?
- How would you build a checkpointing system given 3D integration?

Charge-Based Memories

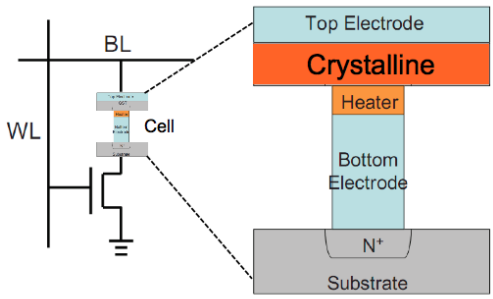


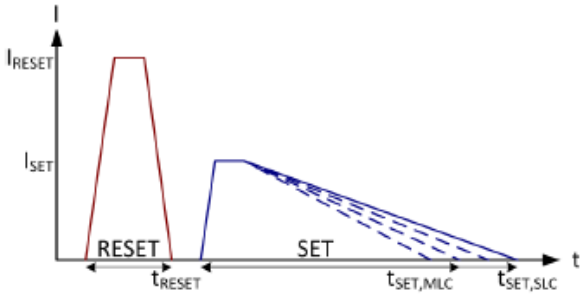
- Charge memories: DRAM, Flash
  - Write data by storing charge (Q)
  - Read data by detecting voltage (V)
- DRAM challenges with technology scaling
  - Smaller capacitor charge, higher transistor leakage
- Flash challenges with technology scaling
  - Tolerance to variability (few electrons per cell), endurance and retention due to thinner tunnel oxide

Resistive Memories

- Resistive memories: PCM, SST-MRAM
  - Write data by pulsing current (dQ/dt)
  - Read data by detecting resistance (R)
- Potential advantages
  - **Scalable** (not relying on few electrons/cell)
  - **Non-volatile**
- Potential disadvantages
  - Write cost (latency, energy)
  - Endurance

Phase Change Memory (PCM)



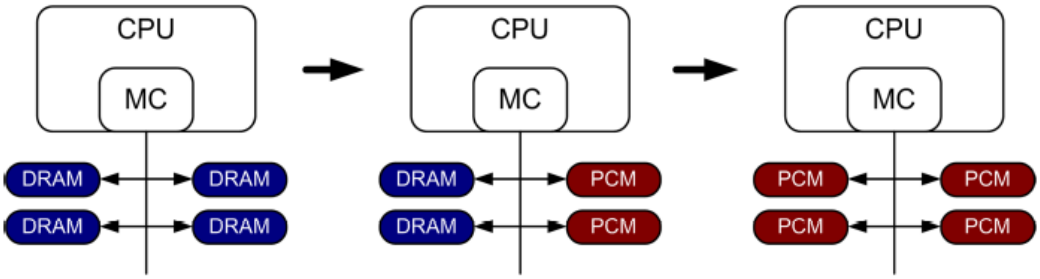


- Set phase via current pulse
  - Non-volatile: amorphous (high R) & crystalline (low R) state
  - Crystallization requires longer time & higher energy
- Detect phase via resistance
  - Retention >10 years

PCM vs DRAM

- Based on PCM prototypes (PCM not in mass production)
- Size: 1.5x of DRAM
  - 9 – 12 F^2
- Endurance: 1e-08x of DRAM
  - 1e+08 writes (phase changes)
- Latency: 4x to 12x of DRAM
  - 50ns read and 150ns write
- Energy: 2x to 40x of DRAM
  - 40uA read, 150uA write
- PCM is non volatile and has negligible idle current
- Both are bit/byte alterable (unlike Flash)

PCM Deployment



- Options
  - DRAM replacement
    - See any issues?
  - Used in conjunction with DRAM
    - What would you allocate in PCM and what in DRAM?

Improving PCM

- Narrow array rows
  - Reduce write energy (proportional to buffer width)
  - Reduce peripheral circuitry, associated area
- On-chip buffers
  - Use DRAM-like buffer and interface
  - Evict modified rows into array
  - Do you see any problem?
- Multiple buffer entries
  - Reduce eviction frequency
  - Improve locality, write coalescing

Improving PCM Wear-out

- Narrow writes
  - Only modified cache lines or words (or even bytes)
  - Requires buffers that track fine-grain state
    - Area/wear-out tradeoff
    - Typically 512B per buffer entry
  - Word-level tracking reduces writes by 5x to 10x
- Other techniques
  - Avoid silent writes (writes that don't change data)
  - Flash-style wear-out leveling

Uses of PCM

- How would you use PCM in systems/software?
- Remember that PCM is
  - Somewhat slower than DRAM
  - Non volatile
  - Byte addressable
- Think of performance, power savings, reliability

Memory Technology Challenges

- Power consumption
  - Dynamic & static memory
- Bandwidth
  - On-chip & off-chip bandwidth
  - Linked to power
- Latency
  - Often interconnect limited
- Density scaling
- Reliability and resilience

Some Perspective

- A DP FP op costs 50pJ and 2-3 clock cycles
- A L1D cache access costs 33pJ and 2-3 clock cycles
- A L2 cache access costs 150pJ and ~15 clock cycles
- A DRAM access costs 2,000pJ and >100 cycles