COEN-4730/EECE-5730 Computer Architecture

# Lecture 5
# Main Memory
## *(Ch.2)*

**Cristinel Ababei**
*Dept. of Electrical and Computer Engineering*
MARQUETTE UNIVERSITY
BE THE DIFFERENCE.

*Credits: Slides adapted from presentations of Sudeep Pasricha and others: Kubiatowicz, Patterson, Mutlu, Elsevier*





# Outline

- Main memory organization
- DRAM basics
- Quest for DRAM Performance
- Memory controller
- Future

# Main Memory Background

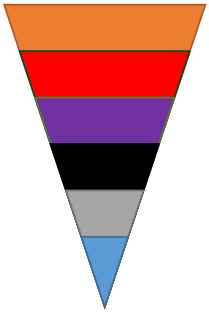
- Performance of Main Memory:
  - Latency: affects cache Miss Penalty
  - Bandwidth: I/O & Large Block →affect Miss Penalty
- Main Memory is DRAM: Dynamic Random Access Memory
  - Dynamic since needs to be refreshed periodically (8 ms, 1% time)
  - Addresses divided into 2 halves (Memory as a 2D matrix):
    - *RAS* or *Row Address Strobe*
    - *CAS* or *Column Address Strobe*
- Cache uses SRAM: Static Random Access Memory
  - No refresh (6 transistors per bit vs. 1 transistor + 1 capacitor per bit)
- While a lot is done in terms of cache organization (to reduce processor-DRAM performance gap), innovations in main memory is needed as well



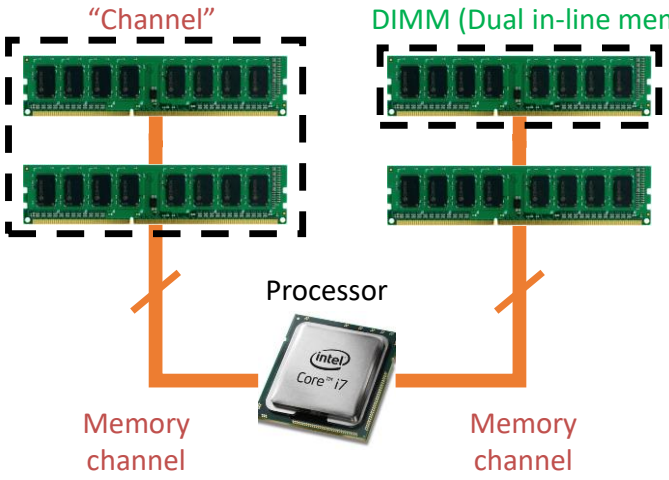


# Memory Subsystem Organization

- Memory subsystem organization
  - Channel
  - DIMM
  - Rank
  - Chip
  - Bank
  - Row/Column



# Memory Subsystem



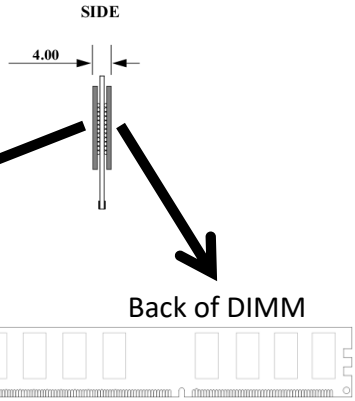

5

5

# Breaking down a DIMM

DIMM (Dual in-line memory module)

Side view




Serial Presence Detect (SPD)

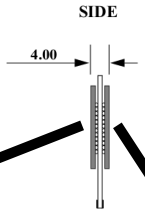
- Stored in EEPROM on module
- Has info to configure memory controllers

6

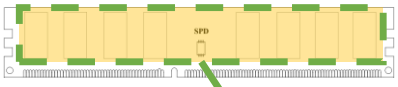
# Breaking down a DIMM

DIMM (Dual in-line memory module)

Side view




Front of DIMM




Back of DIMM

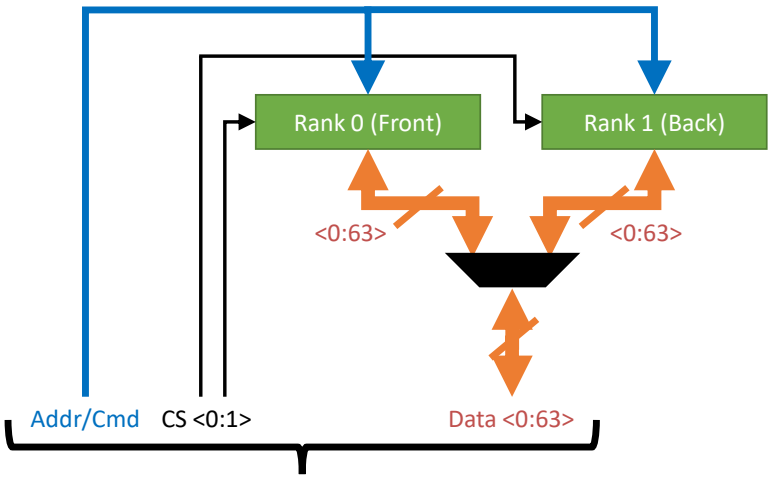
Rank 0: collection of 8 chips

Rank 1

7

7

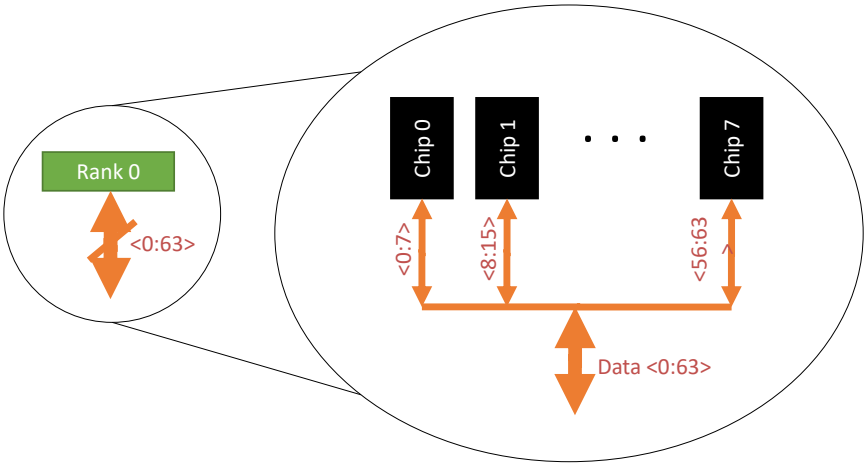
# Rank




8

8

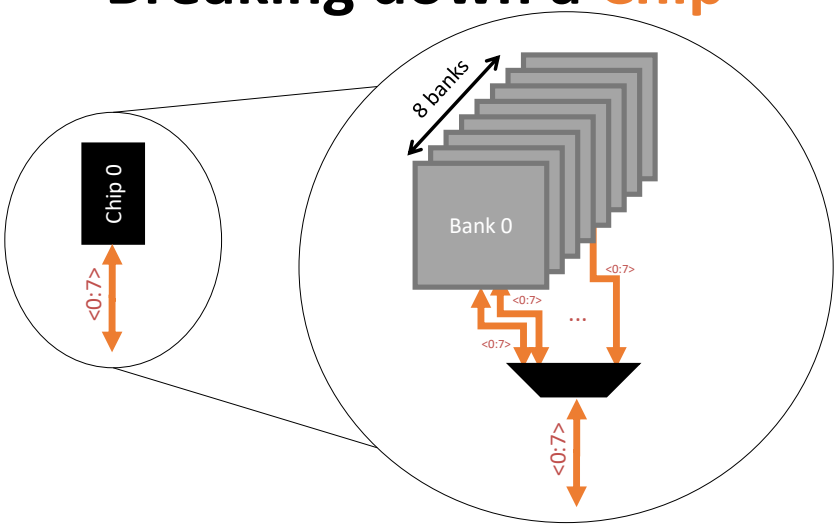
# Breaking down a Rank




9

9

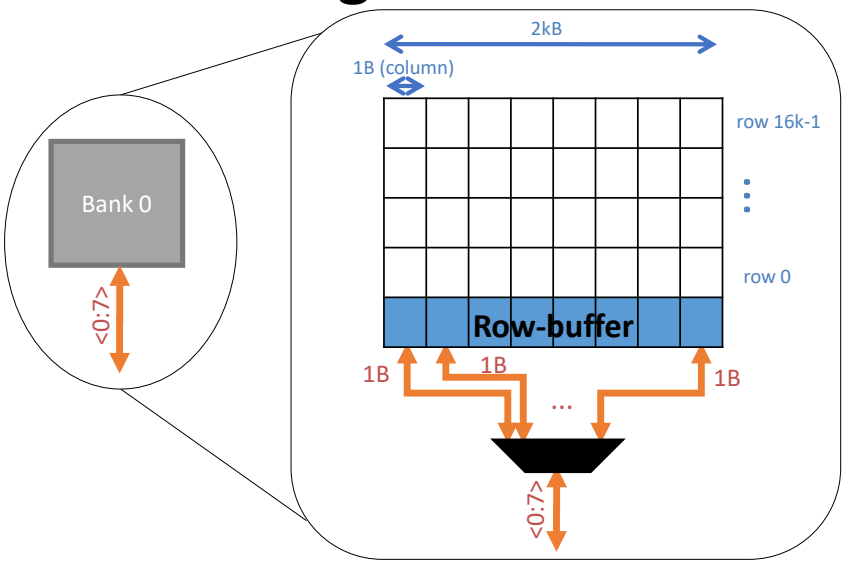
# Breaking down a Chip




10

10

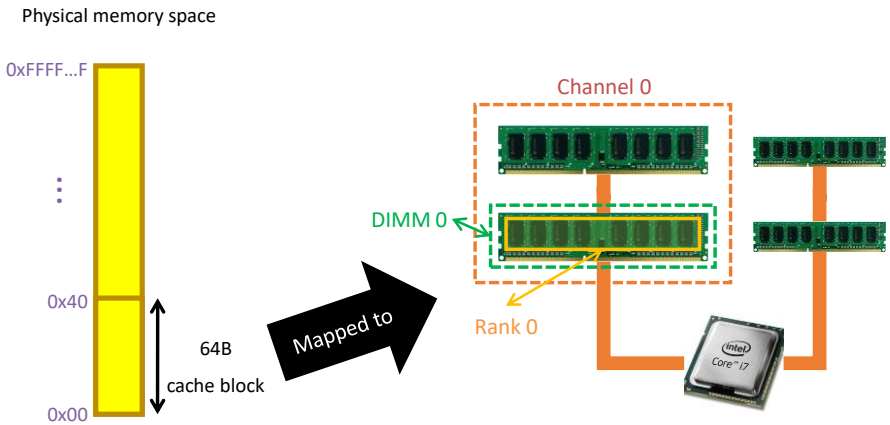
# Breaking down a Bank




11

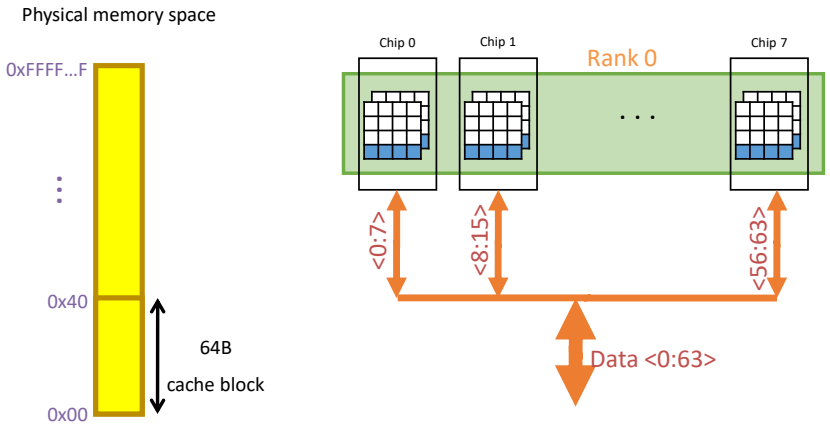
11

# Example: Transferring a Cache Block




12

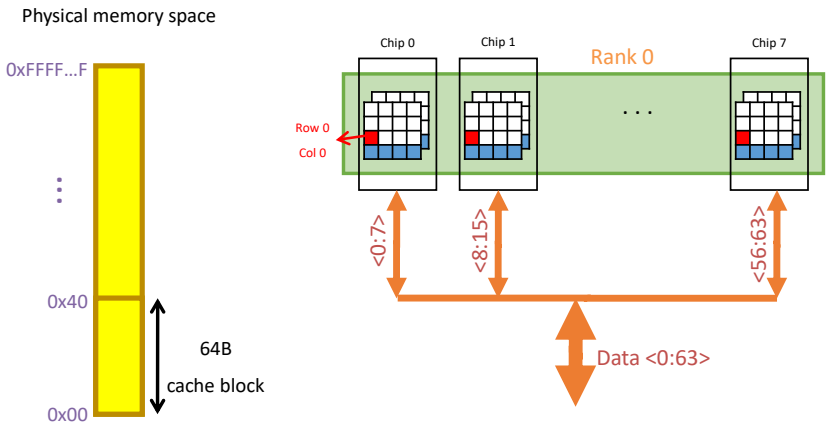
# Example: Transferring a Cache Block



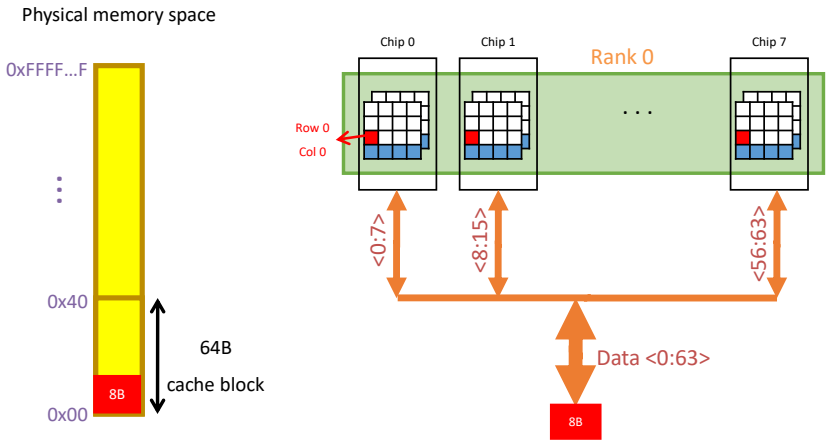





# Example: Transferring a Cache Block



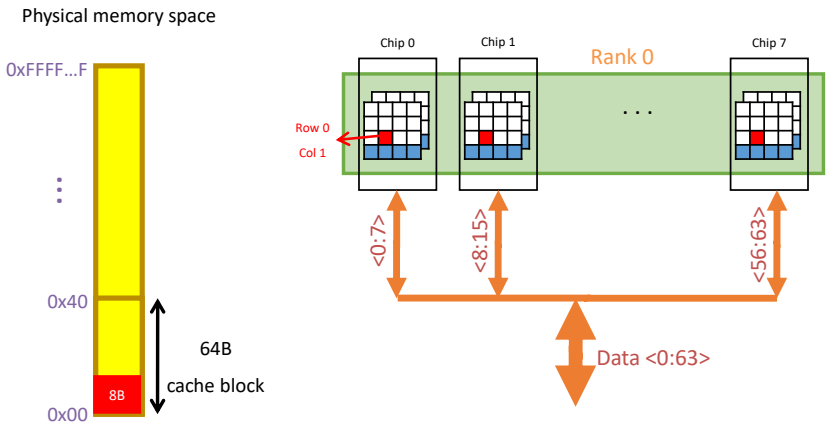
# Example: Transferring a cache block




15

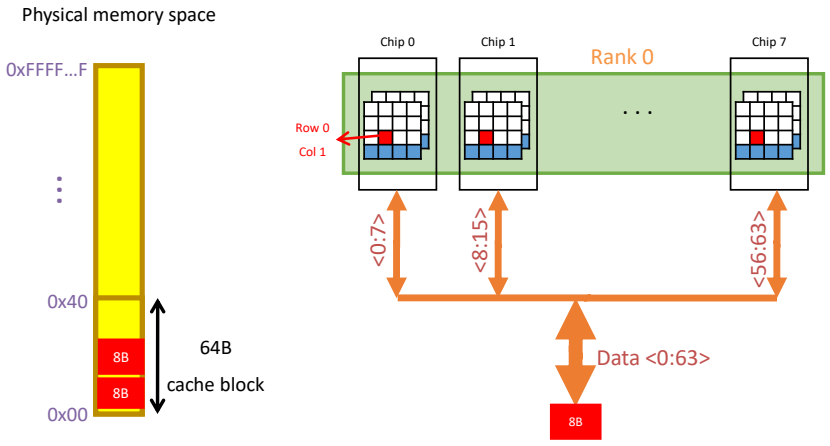
15

# Example: Transferring a cache block




16

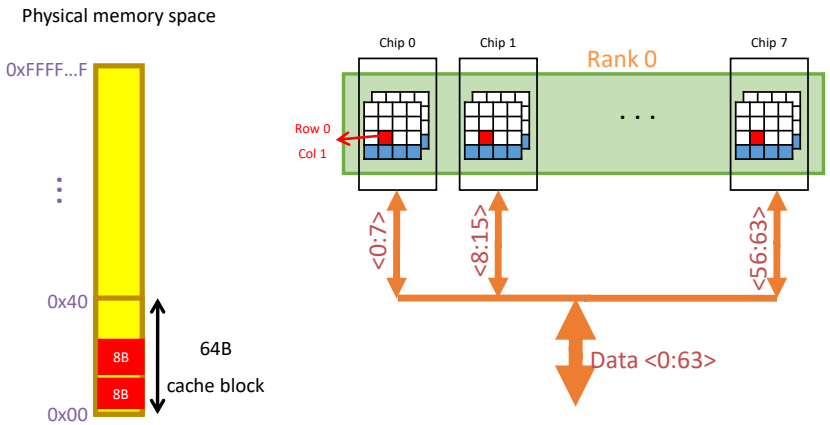
# Example: Transferring a Cache Block




17

17

# Example: Transferring a Cache Block




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

**During the process, 8 columns are read sequentially.**

18

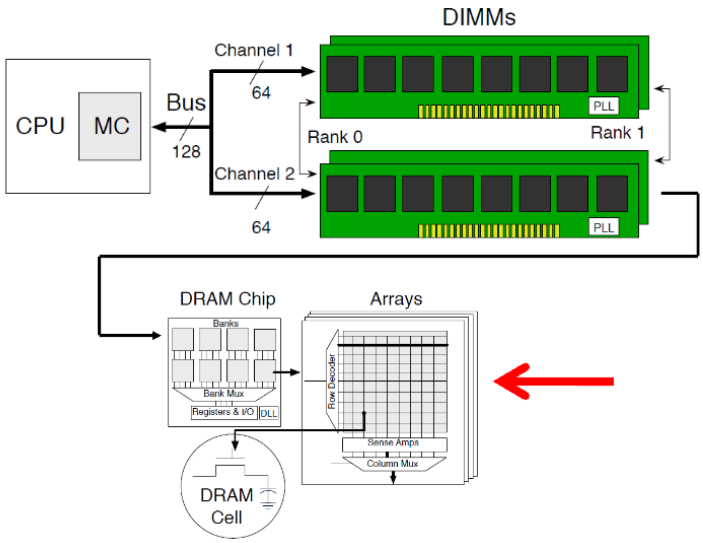
# Outline

- Main memory organization
- DRAM basics
- Quest for DRAM Performance
- Memory controller
- Future

19

19

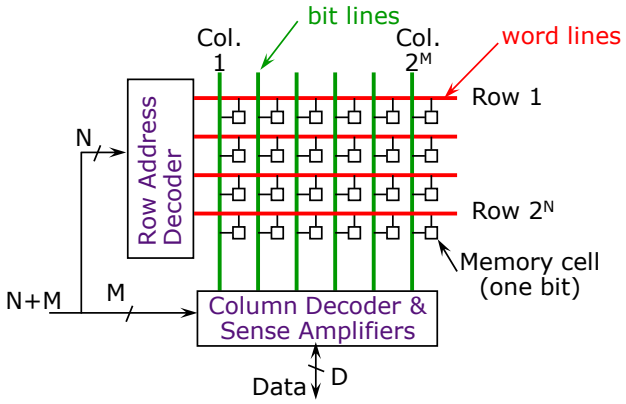
# DRAM Overview




20

20

# DRAM Architecture

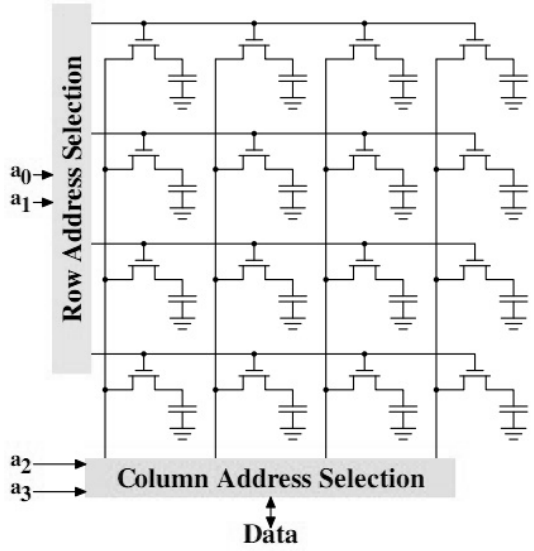



- **Bits stored in 2-dimensional arrays on chip**
- **Modern chips have around 4 logical banks on each chip**
  - **Each logical bank physically implemented as many smaller arrays**

21

21

# DRAM Array

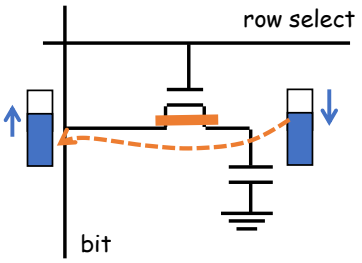



22

22

# 1-T Memory Cell (DRAM)

- **Write:**
  1. Drive bit line
  2. Select row
- **Read:**
  1. Precharge bit line to VDD/2
  2. Select row
  3. Cell and bit line share charges
     - » Minute voltage changes on the bit line
  4. Sense (fancy sense amp)
     - » Can detect changes of ~1 million electrons
  5. Write: restore the value
- **Refresh**
  1. Just do a dummy read to every cell.



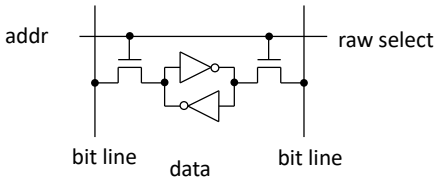

*Read is really a read followed by a restoring Write*

23

# SRAM vs. DRAM

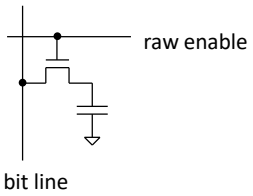
**The primary difference between different memory types is the bit cell**

**SRAM Cell**




- Larger cell ⇒ lower density, higher cost/bit
- No dissipation
- Read non-destructive
- No refresh required
- Simple read ⇒ faster access
- Standard IC process ⇒ natural for integration with logic

**DRAM Cell**




- Smaller cell ⇒ higher density, lower cost/bit
- Needs periodic refresh, and refresh after read
- Complex read ⇒ longer access time
- Special IC process ⇒ difficult to integrate with logic circuits

24

24

# DRAM Operation: Three Steps

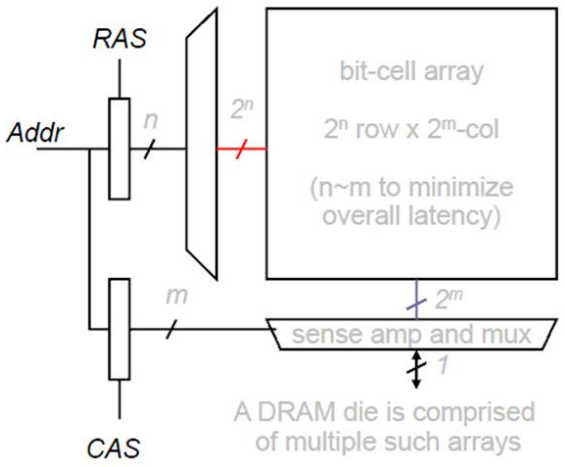
- Precharge
  - Charges bit lines to known value, required before next row access
- Row access
  - Decode row address, enable addressed row (often multiple Kb in row)
  - Contents of storage cell share charge with bitlines
  - Small change in voltage detected by sense amplifiers which latch whole row of bits
  - Sense amplifiers drive bitlines full rail to recharge storage cells
- Column access
  - Decode column address to select small number of sense amplifier latches (4, 8, 16, or 32 bits depending on DRAM package)
  - On read, send latched bits out to chip pins
  - On write, charge sense amplifier latches; which then charge storage cells to required value
  - Can perform multiple column accesses on same row without another row access (burst mode)

25

25

# DRAM: Memory-Access Protocol

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




26

26

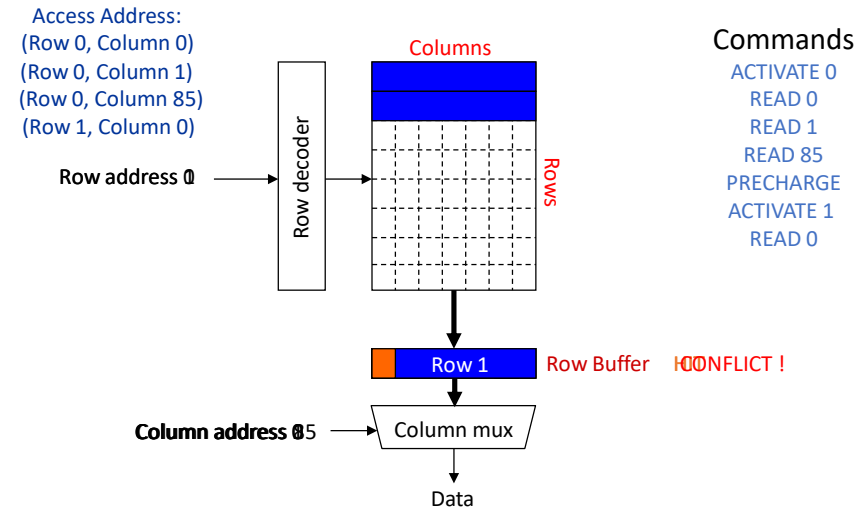
# DRAM: Basic Operation

- 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

27

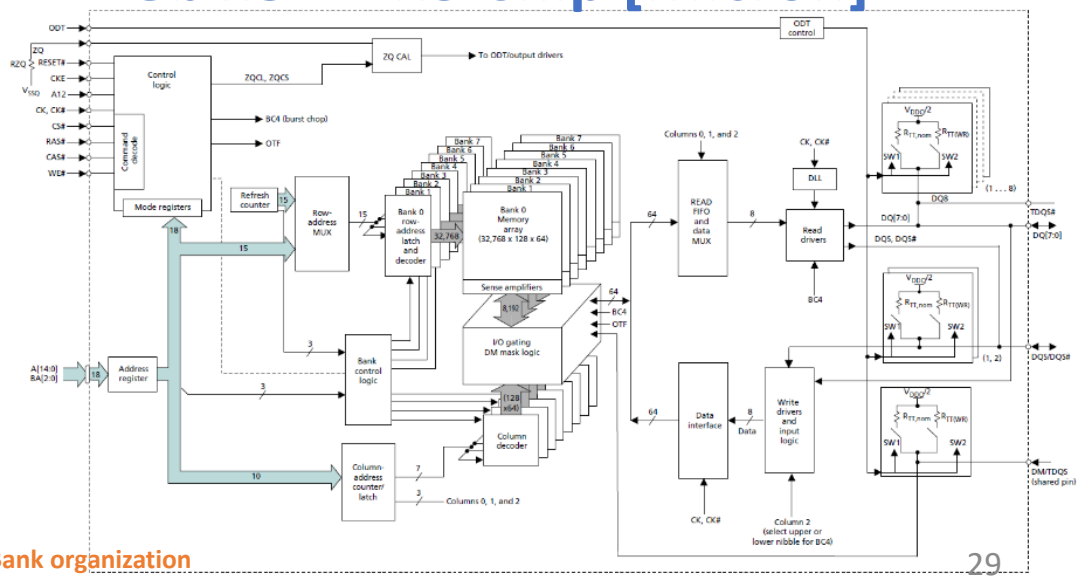


# DRAM Bank Operation




28

# 2Gb x8 DDR3 Chip [Micron]




Observe Bank organization

29

29

# Outline

- Main memory organization
- DRAM basics
- Quest for DRAM Performance
- Memory controller
- Future

30

30

# Quest for DRAM Performance

1. Fast Page mode
   - Add timing signals that allow repeated accesses to row buffer without another row access time
   - Such a buffer comes naturally, as each array will buffer 1024 to 2048 bits for each access
2. Synchronous DRAM (SDRAM)
   - Add a clock signal to DRAM interface, so that repeated transfers would not bear overhead to synchronize with DRAM controller
3. Double Data Rate (DDR SDRAM)
   - Transfer data on both the rising edge and falling edge of the DRAM clock signal ⇒ doubling the peak data rate
   - DDR2 lowers power by dropping the voltage from 2.5 to 1.8 volts + offers higher clock rates: up to 400 MHz
   - DDR3 drops to 1.5 volts + higher clock rates: up to 800 MHz
   - DDR4 drops to 1-1.2 volts + higher clock rates: up to 1600 MHz





# Memory Optimizations

| | | | Best case access time (no precharge) | | | Precharge needed |
|---|---|---|---|---|---|---|
| Production year | Chip size | DRAM type | RAS time (ns) | CAS time (ns) | Total (ns) | Total (ns) |
| 2000 | 256M bit | DDR1 | 21 | 21 | 42 | 63 |
| 2002 | 512M bit | DDR1 | 15 | 15 | 30 | 45 |
| 2004 | 1G bit | DDR2 | 15 | 15 | 30 | 45 |
| 2006 | 2G bit | DDR2 | 10 | 10 | 20 | 30 |
| 2010 | 4G bit | DDR3 | 13 | 13 | 26 | 39 |
| 2016 | 8G bit | DDR4 | 13 | 13 | 26 | 39 |

# Memory Optimizations

| Standard | I/O clock rate | M transfers/s | DRAM name | MiB/s/DIMM | DIMM name |
|---|---|---|---|---|---|
| DDR1 | 133 | 266 | DDR266 | 2128 | PC2100 |
| DDR1 | 150 | 300 | DDR300 | 2400 | PC2400 |
| DDR1 | 200 | 400 | DDR400 | 3200 | PC3200 |
| DDR2 | 266 | 533 | DDR2-533 | 4264 | PC4300 |
| DDR2 | 333 | 667 | DDR2-667 | 5336 | PC5300 |
| DDR2 | 400 | 800 | DDR2-800 | 6400 | PC6400 |
| DDR3 | 533 | 1066 | DDR3-1066 | 8528 | PC8500 |
| DDR3 | 666 | 1333 | DDR3-1333 | 10,664 | PC10700 |
| DDR3 | 800 | 1600 | DDR3-1600 | 12,800 | PC12800 |
| DDR4 | 1333 | 2666 | DDR4-2666 | 21,300 | PC21300 |



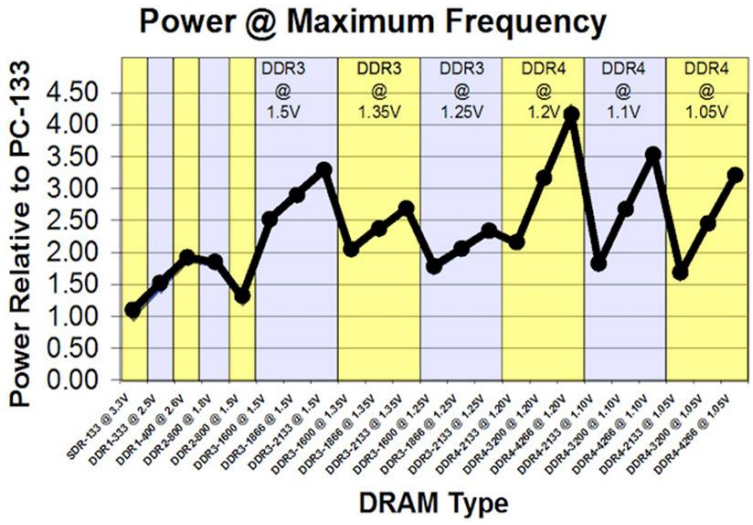


# Graphics Memory

| Product | Density | Banks | Part Num. | PKG & Speed | Org. | Interf. | Ref. | Voltage(V) | PKG. | PKG Type | Status |
|---|---|---|---|---|---|---|---|---|---|---|---|
| gDDR3 SDRAM | 1Gb G-die | 8Banks | K4W1G1646G | BC08/1A 11/12/15 | 64Mx16 | SSTL_15 | 8K/64ms | 1.5V ± 0.075V | 96ball FBGA | Halogen-Free, Lead-Free & Flip-Chip | Mass Production |
| | 2Gb C-die | | K4W2G1646C | HC1A/11 12/15 | 128Mx16 | | | | | Halogen-Free & Lead-Free | Mass Production |
| | DDP 4Gb D-die | | K4W4G1646D | BC12 | 256Mx16 | | | | | | CS Jan.'11 |
| GDDR3 SGRAM | 512Mb I-die | 8Banks | K4J52324KI | HC7A/08 1A/12/14 | 16Mx32 | POD_18 | 8K/32ms | 1.8V ± 0.1V | 136ball FBGA | Halogen-Free & Lead-Free | Mass Production |
| | 1Gb G-die | | K4J10324KG | HC1A/14 | 32Mx32 | | | | | | CS Aug.'11 |
| GDDR5 SGRAM | 1Gb G-die | 16Banks | K4G10325FG | HC03/04/05 | 32Mx32 | POD_15 | 8K/32ms | 1.5V ± 0.045V | 170ball FBGA | Halogen-Free & Lead-Free | Mass Production |
| | 2Gb C-die | | K4G20325FC | HC03/04/05 | 64Mx32 | POD_15 | 16K/32ms | 1.5V ± 0.045V | 170ball FBGA | Halogen-Free & Lead-Free | Mass Production |

- **Achieve 2-5 X bandwidth per DRAM vs. DDR3**
  - **Wider interfaces (32 vs. 16 bit)**
  - **Higher clock rate**
    - **Possible because they are attached via soldering instead of socketted DIMM modules**
  - **E.g. Samsung GDDR5**
    - **2.5GHz, 20 GBps bandwidth on 32-bit bus (160GBps on 256-bit bus)**

# DRAM Power: Not always up, but…

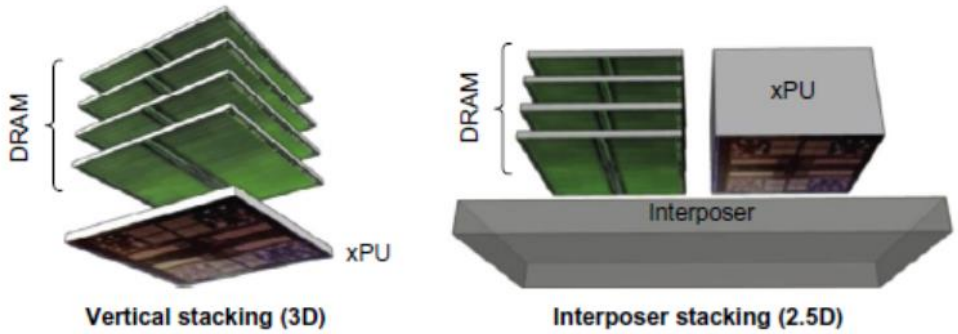




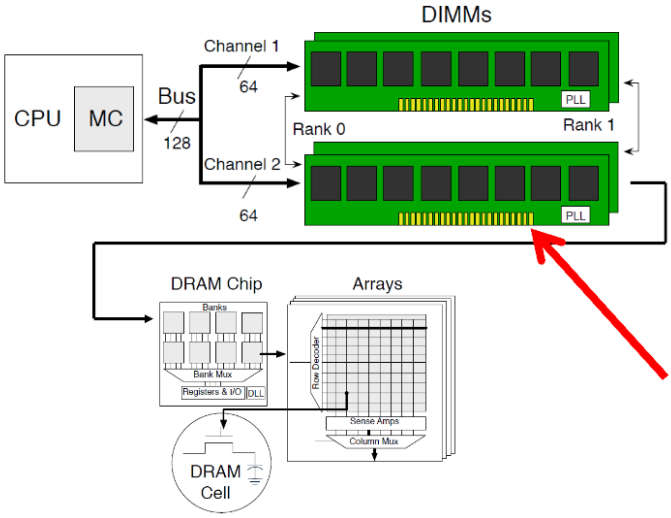



# Stacked/Embedded DRAMs

- Stacked DRAMs in same package as processor
  - High Bandwidth Memory (HBM)



# DRAM Modules




37

# DRAM Modules

- DRAM chips have narrow interface (typically x4, x8,x16)
- Multiple chips are put together to form wide interface
  - DIMM: Dual Inline Memory Module
  - E.g., 64-bit DIMM needs to access 8 chips with 8-bit interface
  - Share command/address lines, but not data

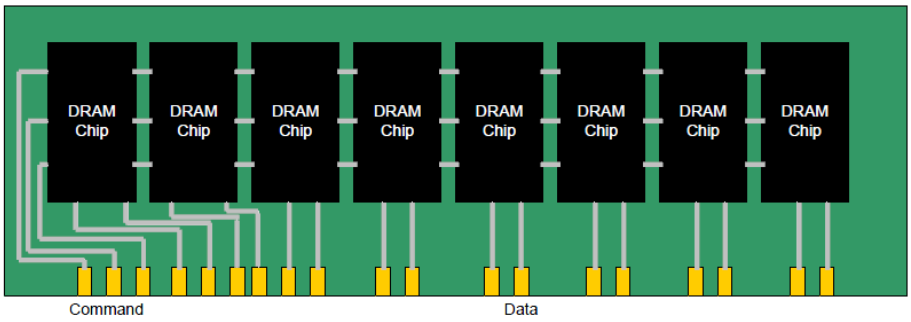
- Advantages
  - Acts like a high-capacity DRAM chips with wide interface
  - 8x capacity, 8x bandwidth, same latency

- Disadvantages
  - Granularity: Accesses cannot be smaller than the interface width
  - 8x power

38

38

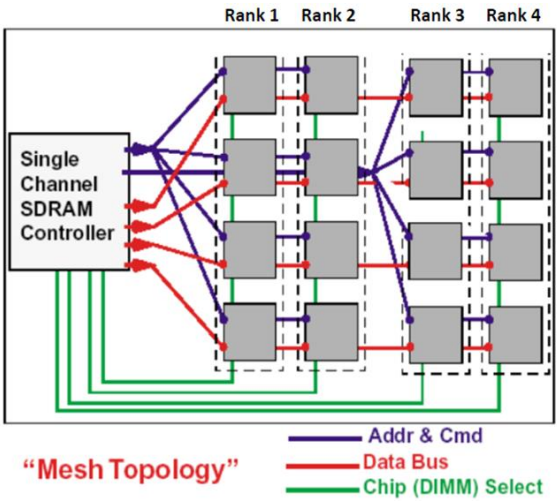
# A 64-bit Wide DIMM (physical view)




39

# Increasing Capacity: Multiple DIMMs on a Channel

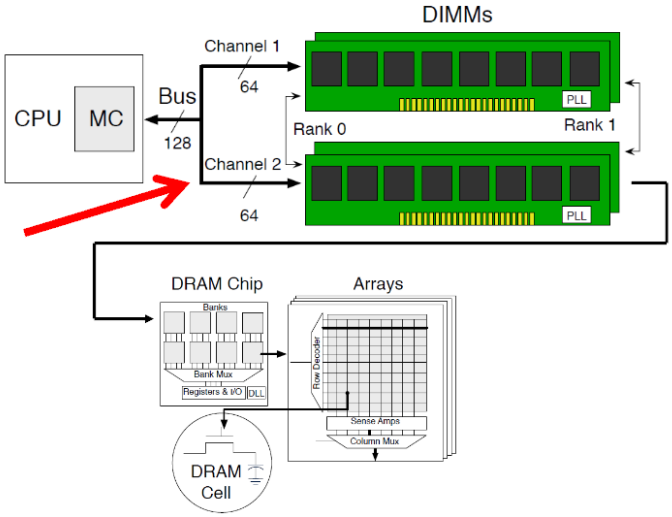
- Advantages
  - Enables even capacity
- Disadvantages
  - Interconnect latency
  - Complexity
  - Higher energy usage
  - Address/Command signal integrity is a challenge




40

40

# DRAM Channels




41

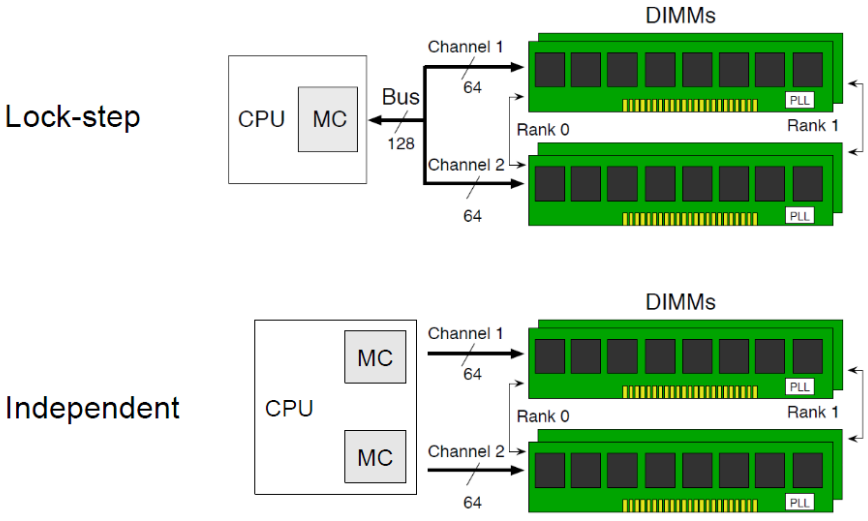
# DRAM Channels

- Channel: a set of DIMMS in series
  - All DIMMs get the same command, one of the ranks replies
- System options
  - Single channel system
  - Multiple dependent (lock-step) channels
    - Single controller with wider interfaces
    - Only works if DIMMS are identical
  - Multiple independent channels
    - Requires multiple controllers
- Tradeoffs
  - Cost: pins, wires, controller
  - Benefit: higher bandwidth, capacity, flexibility

42

42

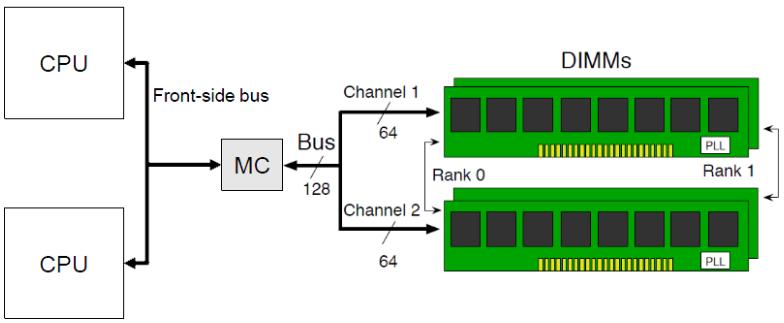
# DRAM Channel Options




43

43

# Multi-CPU (old school)

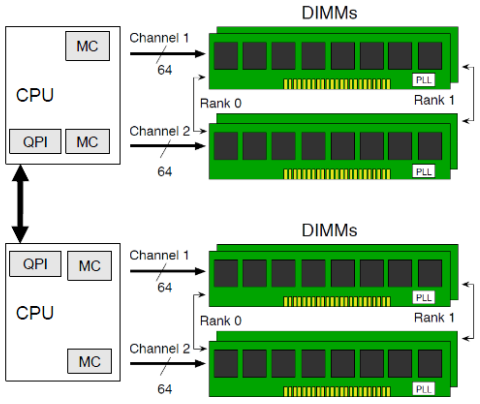



- External MC adds latency
- Capacity doesn't grow w/ # of CPUs

44

44

# NUMA Topology (modern)




- Capacity grows w/ # of CPUs
- NUMA: "Non-uniform Memory Access"

45

45

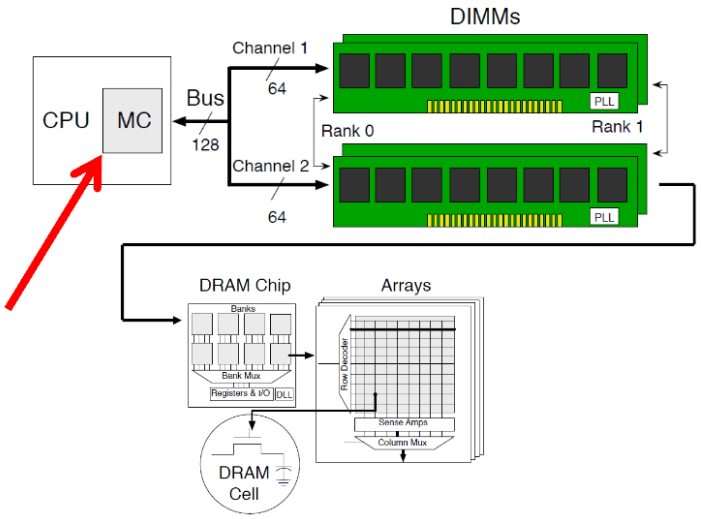
# Outline

- Main memory organization
- DRAM basics
- Quest for DRAM Performance
- Memory controller
- Future

46

46

# Memory Controller




47

47

# DRAM Controller Functionality

- Obey timing constraints of DRAM
- Map physical addresses to DRAM addresses
- Row buffer management policies
- DRAM request scheduling
- DRAM refresh strategies
- DRAM power management
- DRAM reliability

→ DRAM controllers are challenging to design!

48

48

# Latency Components: Basic DRAM Operation

- CPU → 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)
- Controller → CPU transfer time

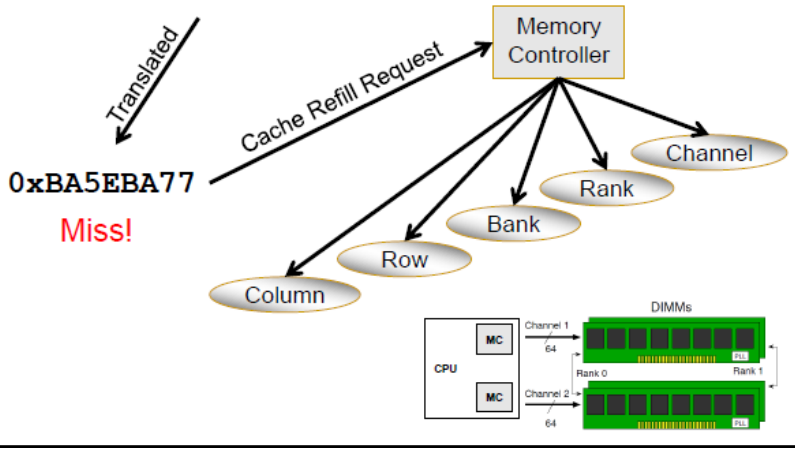
49



# DRAM Addressing

## Memory Controller has a significant impact on access latency

```
LD R1, Mem[foo]
```




50

# 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

51

51

# DRAM: Timing Constraints

- There are dozens of these…
  - tWTR = Write to read delay
  - tWR = Time from end of last write to PRECHARGE
  - tFAW = Four ACTIVATE window
  - …
- Makes performance analysis, memory controller design difficult
- DRAM datasheets freely available, abundant with such constraints

52

## DRAM Controller Scheduling Policies (I)

- FCFS (first come first served)
  - Oldest request first
- FR-FCFS (first ready, first come first served)
  - Row-hit first
  - Oldest first
  - Goal: maximize row buffer hit rate → maximize DRAM throughput

53

53

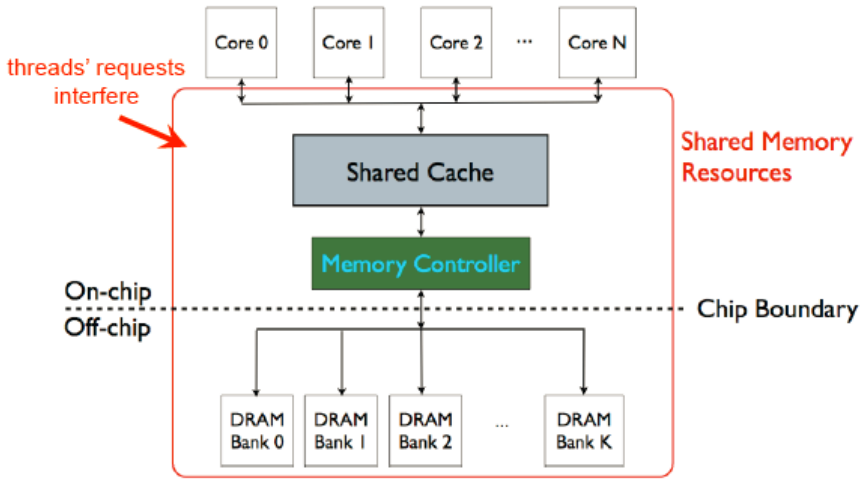
## DRAM Controller Scheduling Policies (II)

- A scheduling policy is a prioritization order
- Prioritization can be based on
  - Request age
  - Row buffer hit/miss status
  - Request type (prefetch, read, write)
  - Requestor type (load miss or store miss)
  - Request criticality
    - Oldest miss in the core?
    - How many instructions in the core depend on it?

54

54

# Problem: Memory Request Interference




55

55

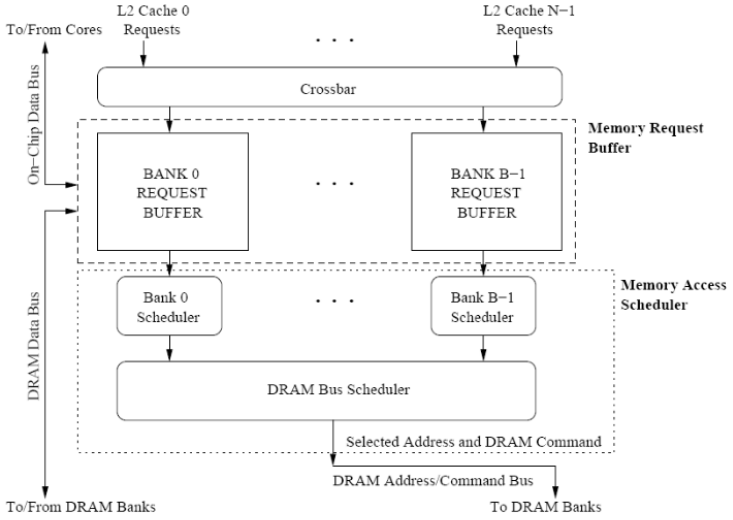
# Problem: Memory Request Interference

- Problem: Threads share the memory system, but memory system does not distinguish threads' requests
  - Memory system algorithms thread-unaware and thread-unfair
- Existing memory systems
  - Free-for-all, demand-based sharing of the memory system
  - Aggressive threads can deny service to others
  - Do not try to reduce or control inter-thread interference
- Solution #1: Smart resources: Design each shared resource to have a configurable fairness/QoS mechanism
  - Fair/QoS-aware memory schedulers, interconnects, caches, arbiters
- Solution #2: Dumb resources: Keep each resource free-for-all, but control access to memory system at the cores/sources
  - Fairness via Source Throttling; Estimate thread slowdowns in the entire system and throttle cores that slow down others; Coordinated Prefetcher Throttling

56

56

# A Modern DRAM Controller



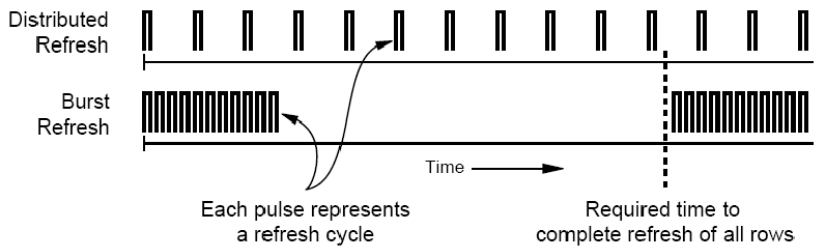





# DRAM Refresh

- 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

# DRAM Refresh




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

59

59

# DRAM Power Management

- DRAM chips have power modes
- Idea: When not accessing a chip power it down

- Power states
  - Active (highest power)
  - All banks idle (i.e. precharged)
  - Power-down
  - Self-refresh (lowest power)

- State transitions incur latency during which the chip cannot be accessed

60

# Mobile DRAM characteristics

| Technology Parameter | DDR3 | LPDDR2 |
|---|---|---|
| Timing (tCAS, tRAS, tRC) | 15, 38, 50ns | 15, 42, 57ns |
| Active current (Read, Write) | 180, 185mA | 210, 175mA |
| Idle current (Powerdown, Standby) | 35, 45mA | 1.6, 23mA |
| Powerdown exit latency | 24ns | 7.5ns |
| Operating voltage | 1.5V | 1.2V |
| Typical operating frequency | 800MHz | 400MHz |
| Device width | 8 | 16 |

- Same core as DDR3 devices
  - Same capacity per device , same access latency, same active currents
- IO interface optimized for very low static power
  - Including faster power down modes, no termination
- Same chip bandwidth
  - Wider interface operating at slower clock rate

61

61

# DRAM Reliability

- DRAMs are susceptible to soft and hard errors
- Dynamic errors can be
  - Detected by parity bits
    - Usually 1 parity bit per 8 bits of data
  - Detected and fixed by the use of Error Correcting Codes (ECCs)
    - E.g. SECDED Hamming code can detect two errors and correct a single error with a cost of 8 bits of overhead per 64 data bits
- In very large systems, the possibility of multiple errors as well as complete failure of a single memory chip becomes significant
  - Chipkill (advanced form of ECC) was introduced by IBM to solve this problem
  - Chipkill distributes data and ECC information, so that the complete failure of a single memory chip can be handled by supporting the reconstruction of the missing data from the remaining memory chips
  - IBM and SUN servers and Google Clusters use it
  - Intel calls their version SDDC

62

62

# Outline

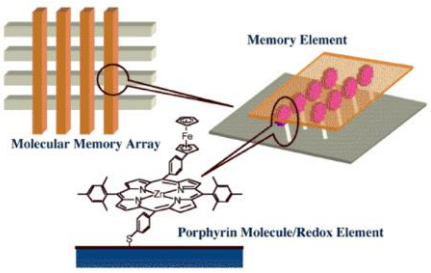
- Main memory organization
- DRAM basics
- Quest for DRAM Performance
- Memory controller
- Future

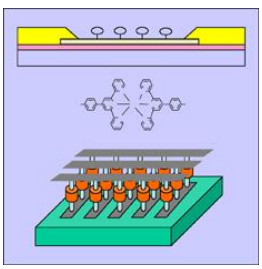
63

63

# Molecular RAM

- Molecular RAM technology
  - Use special compounds such as porphyrin-based polymers to store electric charge
  - Once a certain voltage threshold is achieved the material oxidizes, releasing an electric charge. The process is reversible, in effect creating an electric capacitor.
  - Some universities, Hewlett-Packard have announced work on molecular memories.
  - NASA is also supporting research on non-volatile molecular memories.



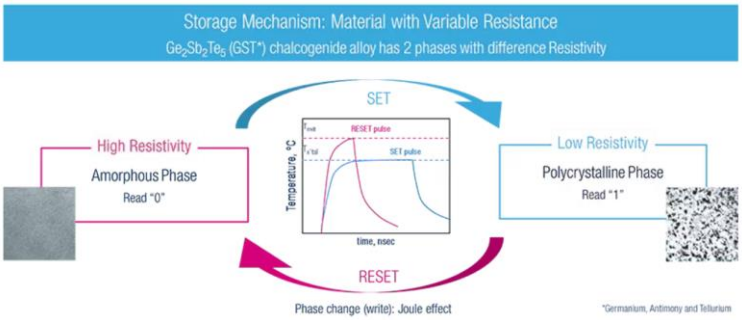



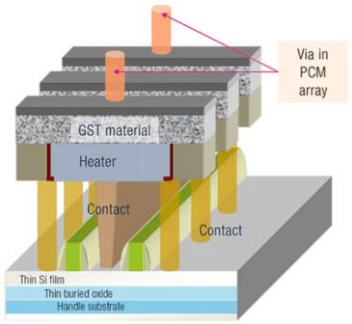
64

64

# Phase Change Memory (PCM)

- Phase Change Memory (PCM) technology
  - Uses a *glass* that can be changed between amorphous and crystalline states. Nonvolatile
  - https://www.st.com/content/st_com/en/about/innovation---technology/PCM.html



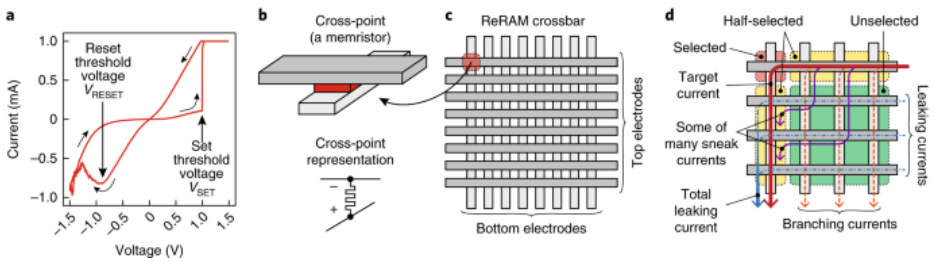




65

65

# Resistive random-access memory (ReRAM)

- Resistive random-access memory (ReRAM or RRAM)
  - A type of non-volatile (NV) random-access (RAM) computer memory that works by changing the resistance across a dielectric solid-state material, often referred to as a memristor.
  - https://www.crossbar-inc.com/technology/reram-advantages/
  - https://www.lumenci.com/post/reram
  - …

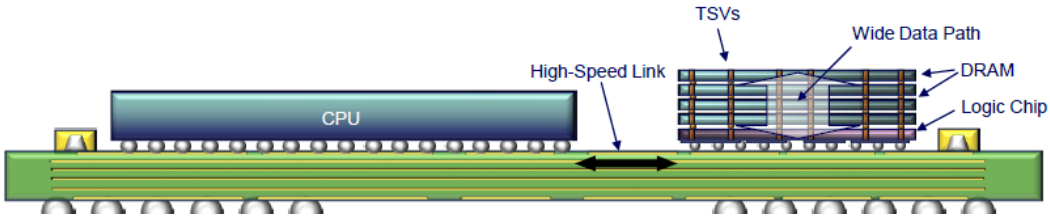



66

66

# 3D Integration: Micron Hybrid Memory Cube (HMC)

- 3D-stacked device with memory + logic
- Links between CPU and logic layer HMC
- High capacity, low power, high bandwidth
  - https://www.micron.com/-/media/client/global/documents/products/data-sheet/hmc/gen2/hmc_gen2.pdf




67



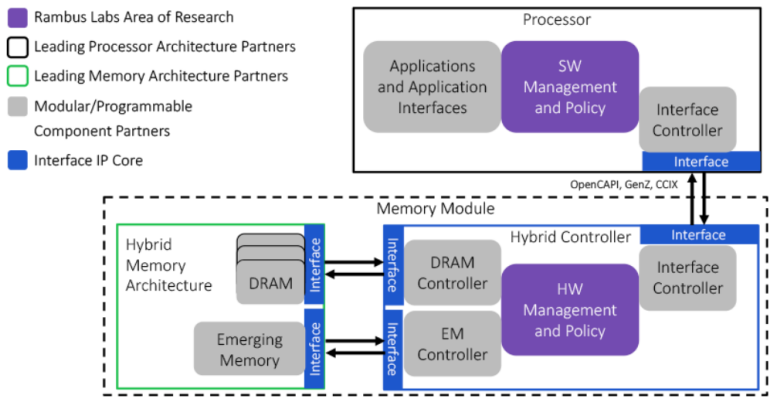
# 3D Integration: AMD Xilinx

- AMD 3D ICs utilize stacked silicon interconnect (SSI) technology
  - https://www.xilinx.com/products/silicon-devices/3dic.html


68

# Hybrid Memory Research

- Emerging Solutions - Hybrid Memory Research
  - https://www.rambus.com/emerging-solutions/hybrid-memory/




69

69

# Resources

- http://www.monolithic3d.com/
- http://isscc.org/index.html
- ISSCC – Memory Trends
  - https://www.isscc.org/trends
  - https://static1.squarespace.com/static/6130ef779c7a2574bd4b8888/t/626325a757094e52110f7db0/1650664892028/ISSCC2022PressKit.pdf
- An article
  - https://www.eejournal.com/article/can-any-emerging-memory-technology-topple-dram-and-nand-flash/

70

70