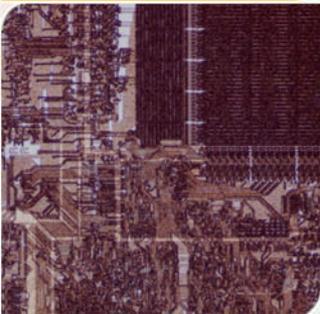


# CS4290/CS6290

Fall 2011

Prof. Hyesoon Kim



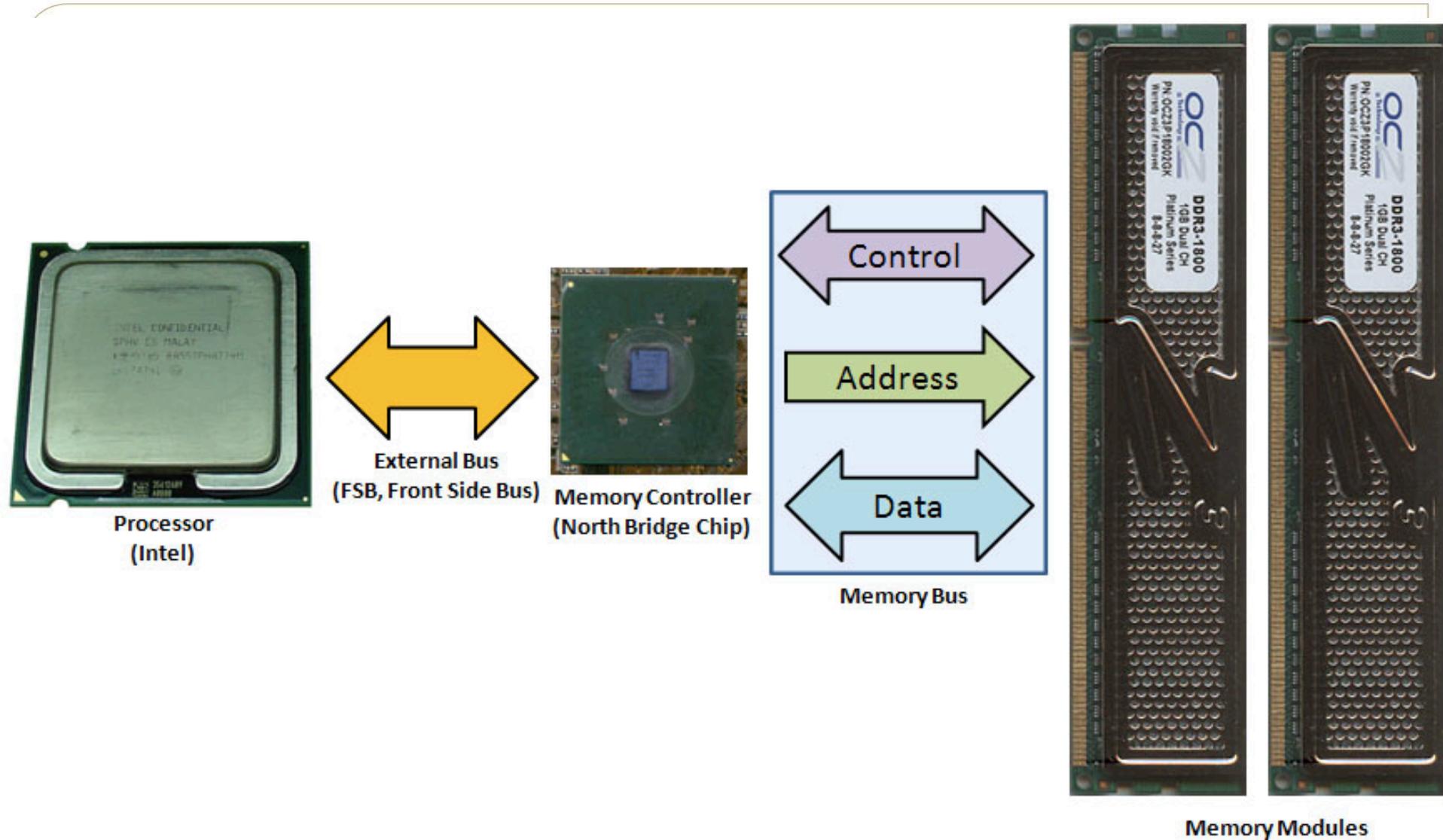
**Georgia  
Tech**



College of  
Computing

Thanks to Prof. Loh & Prof. Prvulovic

# CPU-DRAM



Processor  
(Intel)

External Bus  
(FSB, Front Side Bus)

Memory Controller  
(North Bridge Chip)

Memory Bus

Memory Modules

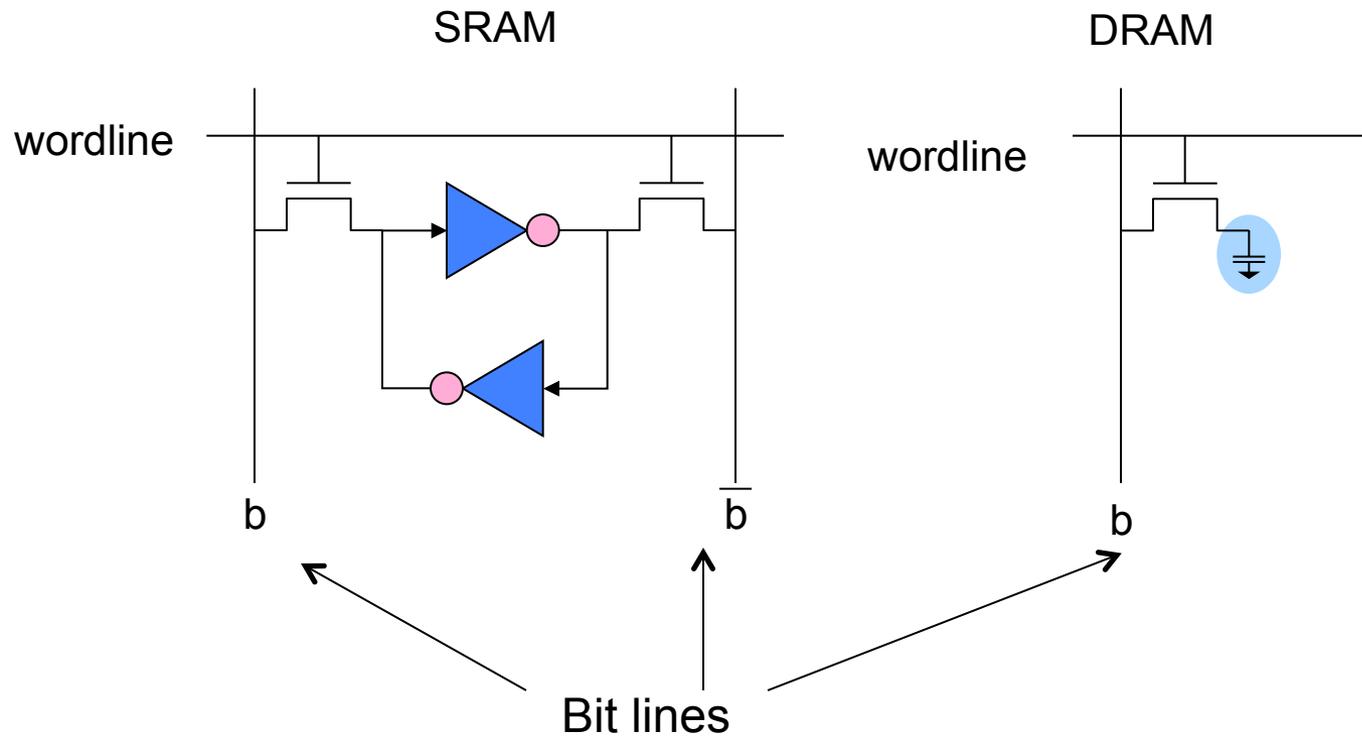


# SRAM vs. DRAM

- DRAM = Dynamic RAM
- SRAM: 6T per bit
  - built with normal high-speed CMOS technology
- DRAM: 1T per bit
  - built with special DRAM process optimized for density



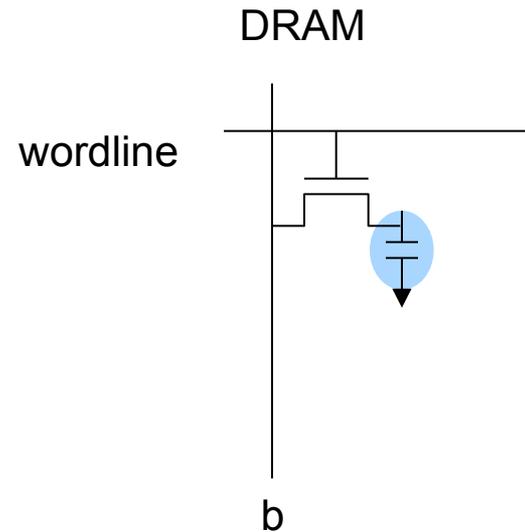
# Hardware Structures





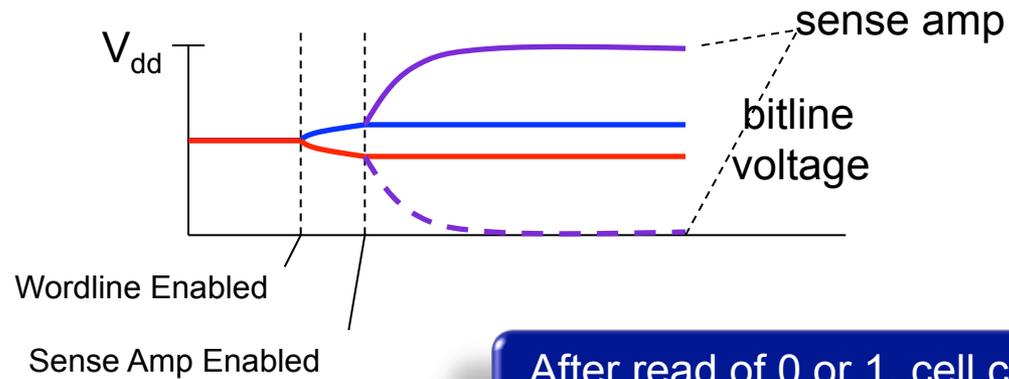
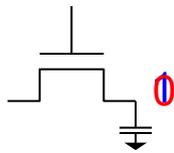
# DRAM Read/Write

- Write
  - Charge bitline HIGH or LOW and set wordline HIGH
- Read
  - Bitline is precharged
  - Wordline is set
  - Depending on the charge bitline becomes slightly higher or lower

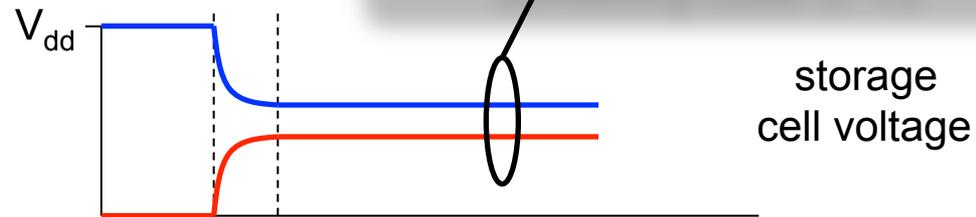




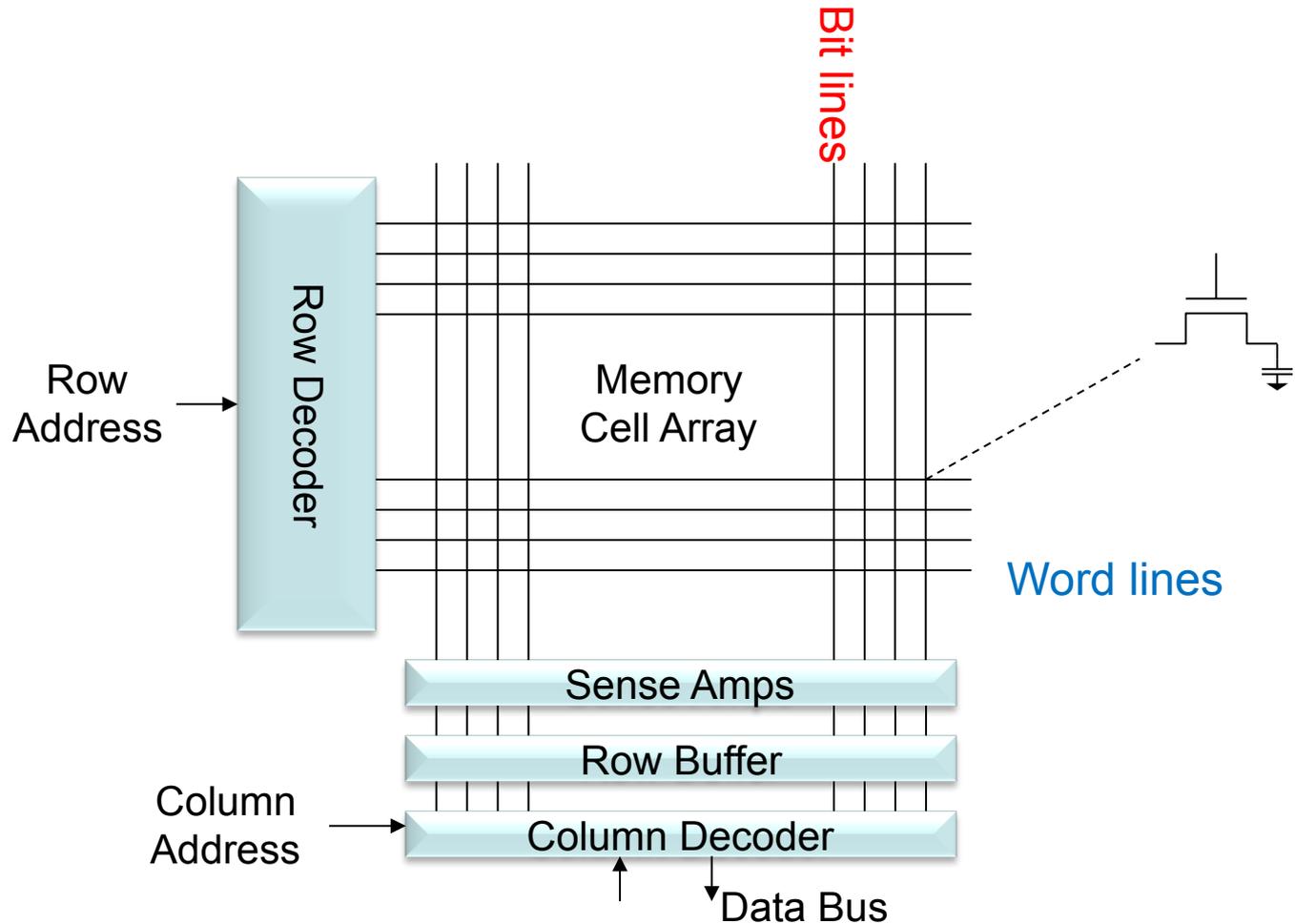
# Destructive Read



After read of 0 or 1, cell contains something close to 1/2



# DRAM Chip Organization

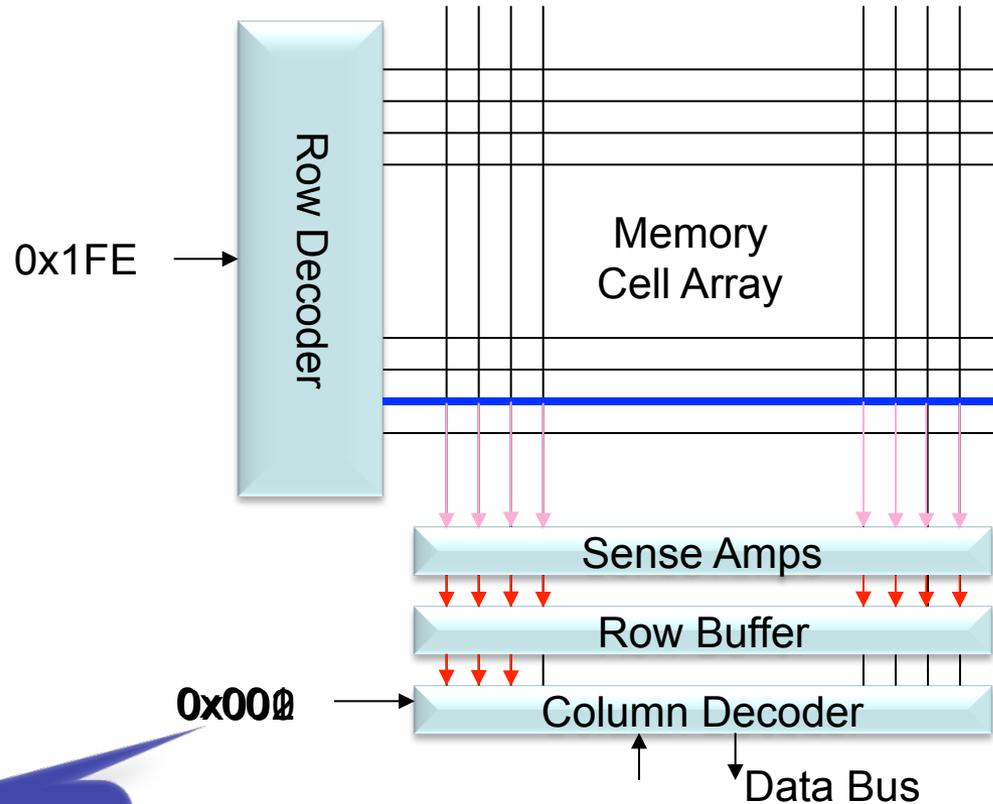


# DRAM Chip Organization (2)



- Differences with SRAM
  - reads are *destructive*: contents are erased after reading
  - Row buffer/DRAM Page
    - Read lots of bits all at once, and then parcel them out based on different column addresses
    - Read from the same row buffer from different locations order
  - “Fast-Page Mode” FPM DRAM organizes the DRAM row to contain bits for a complete page
    - row address held constant, and then fast read from the consecutive locations from the same page

# DRAM Read Operation

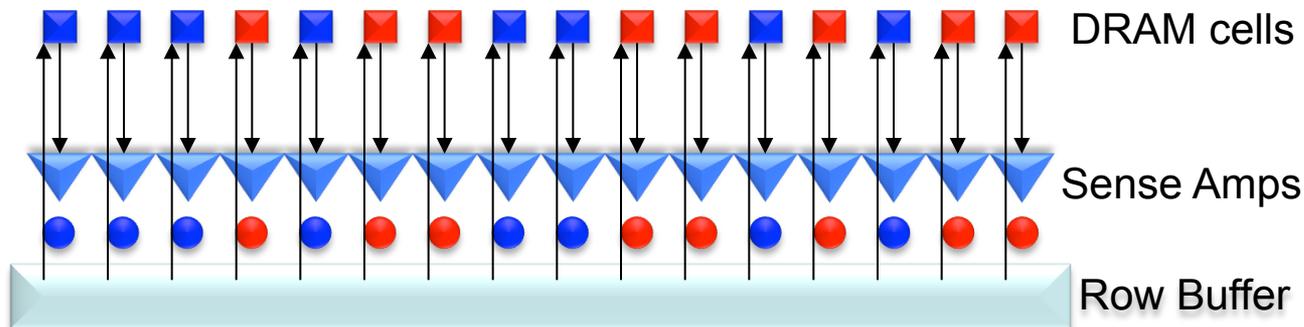


Accesses need not be sequential



# Refresh

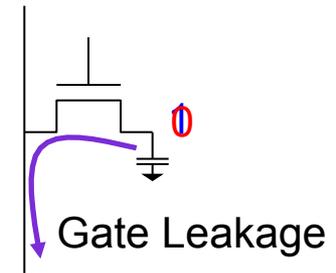
- So after a read, the contents of the DRAM cell are gone
- The values are stored in the row buffer
- Write them back into the cells for the next read in the future





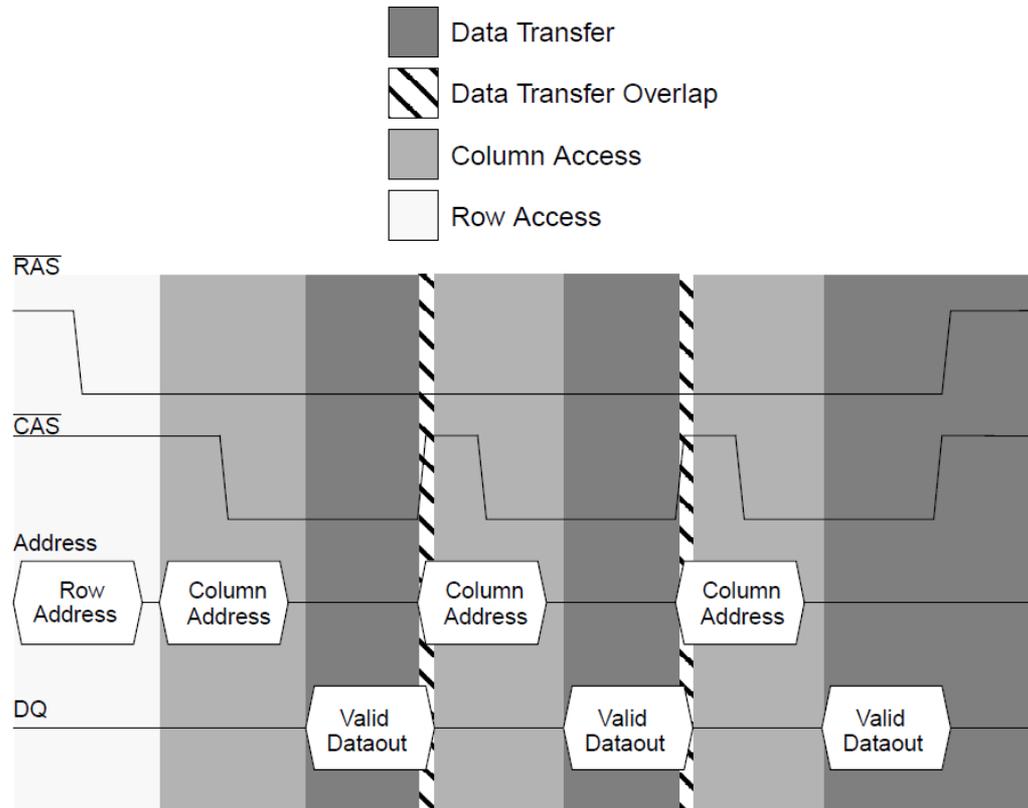
# Refresh (2)

- Fairly gradually, the DRAM cell will lose its contents even if it's not accessed
  - This is why it's called “dynamic”
  - Contrast to SRAM which is “static” in that once written, it maintains its value forever (so long as power remains on)
- All DRAM rows need to be regularly read and re-written



If it keeps its value even if power is removed, then it's “non-volatile” (e.g., flash, HDD, DVDs)

# DRAM Read Timing

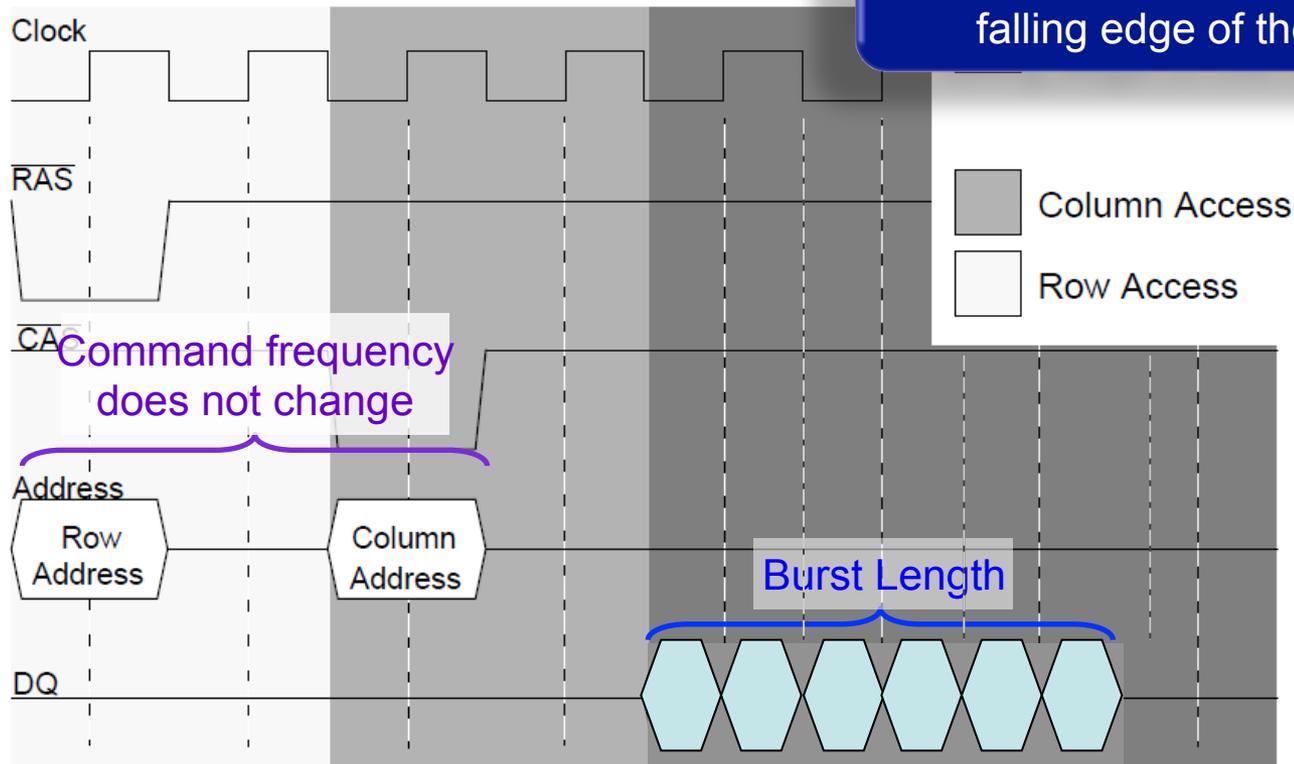


Accesses are asynchronous: triggered by RAS and CAS signals, which can in theory occur at arbitrary times (subject to DRAM timing constraints)

# SDRAM Read Timing



Double-Data Rate (DDR) DRAM transfers data on both rising and falling edge of the clock



Timing figures taken from "A Performance Comparison of Contemporary DRAM Architectures" by Cuppu, Jacob, Davis and Mudge



# Burst Access

- One command access, multiple bytes are read/written.
- Hardware provides multiple burst length option and software can set.



# Example Memory Latency Computation

- FSB freq = 200 MHz, SDRAM
- RAS delay = 2, CAS delay = 2
- Scheduling in memory controller

A0, A1, B0, C0, D3, A2, D0, C1, A3, C3, C2, D1, B1, D2

- Think about hardware complexity...



# REVIEW: VIRTUAL ADDR & CACHE

# Virtual Memory/Physical Memory

- Programmer's view: virtual memory space
- Actual hardware's view: Physical memory space
- In hardware: translation from virtual address to physical address

Virtual Address

Virtual Page Number

Page Offset

Translation

Protection check!

Read/write, kernel/user?

Physical Address

Physical Frame Num

Page Offset

# Need for Translation

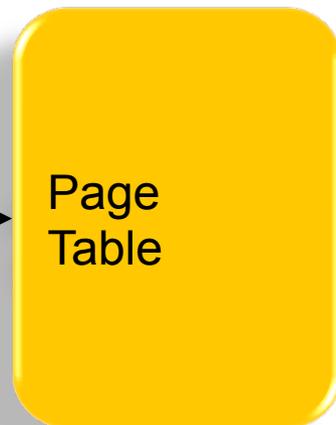


0xFC51908B

Virtual Address

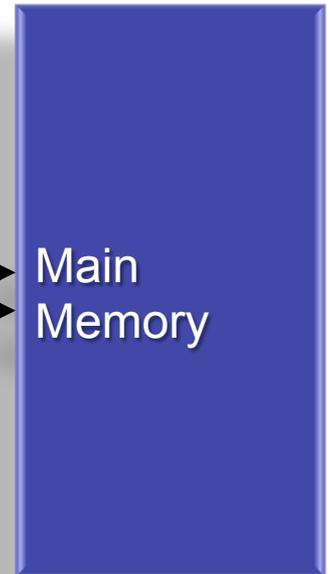


0xFC519



0x00152

Physical Address



0x0015208B



# CPU Memory Access

- Program deals with virtual addresses
  - “Load R1 = 0[R2]”

- On memory instruction

1. Compute virtual address (0[R2])

2. Compute virtual page number

3. Compute physical address of VPN's page table entry

Could be more depending  
On page table organization

4. Load\* mapping

5. Compute physical address

6. Do the actual Load\* from memory



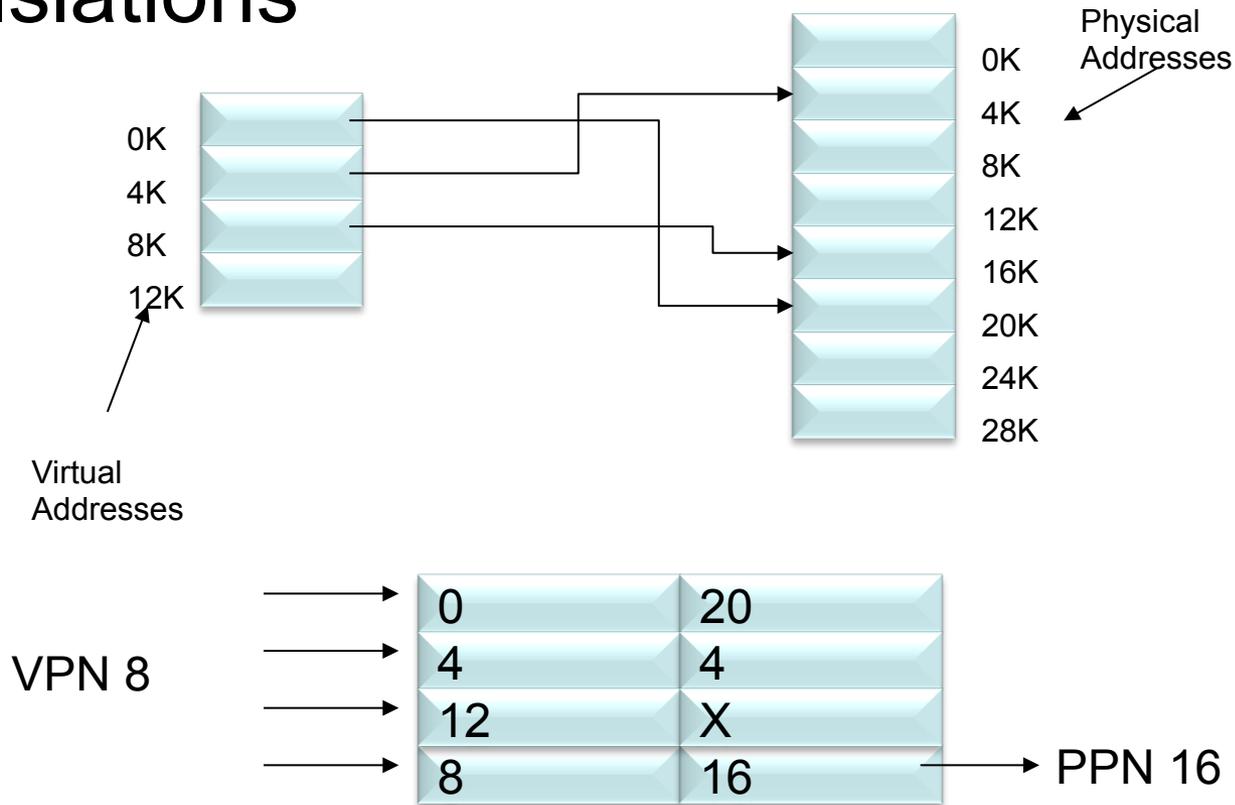
# Impact on Performance?

- Every time you load/store, the CPU must perform two (or more) accesses!
- Even worse, every *fetch* requires translation of the PC!
- Observation:
  - Once a virtual page is mapped into a physical page, it'll likely stay put for quite some time



# Idea: Caching!

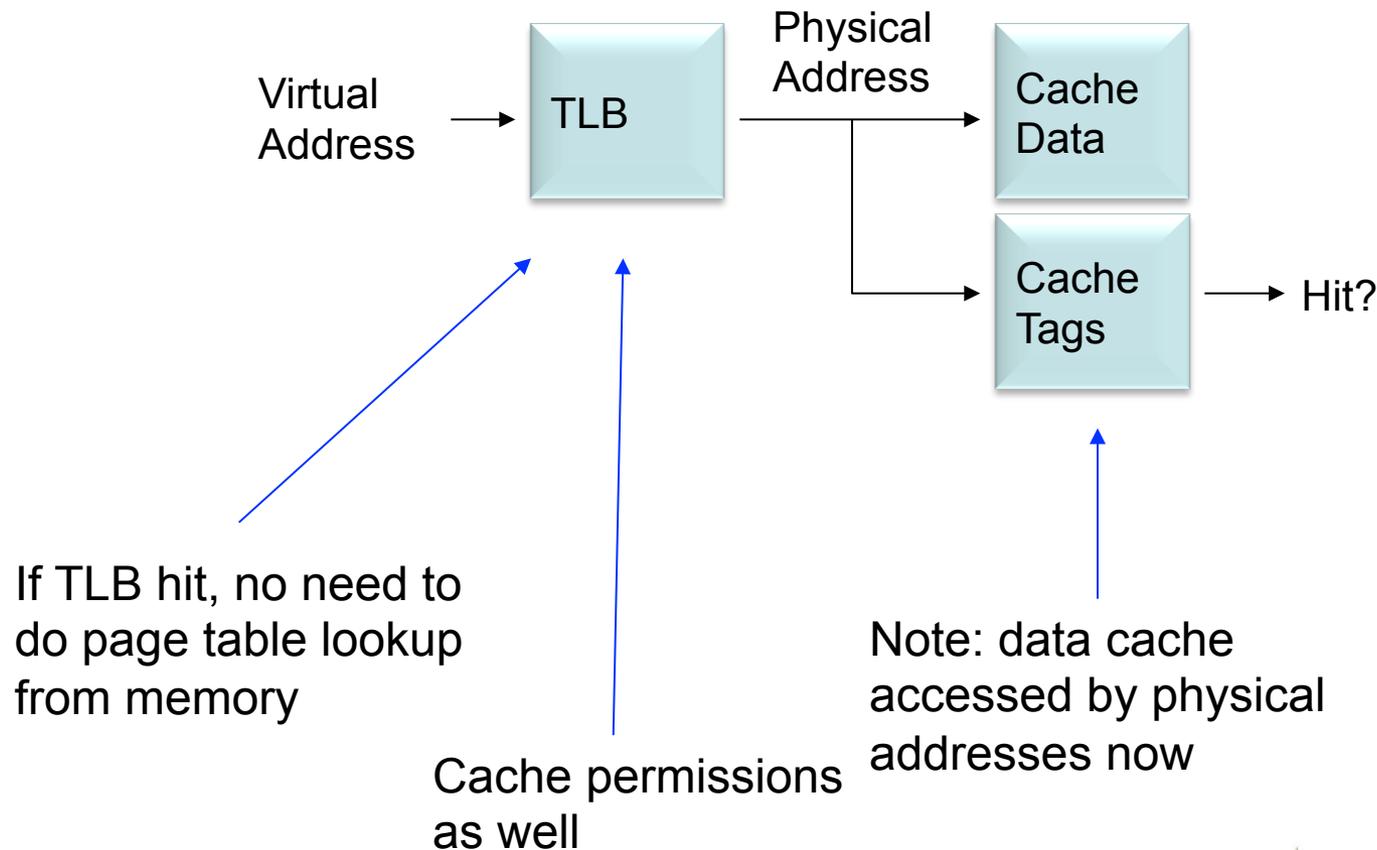
- Not caching of data, but caching of translations



TLB also has protection bits, R/W, kernel/user information

# Translation Cache: TLB

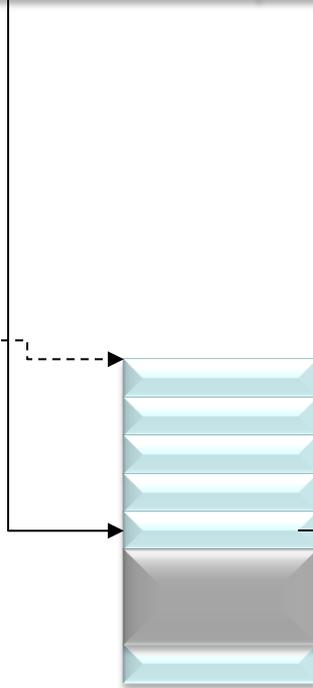
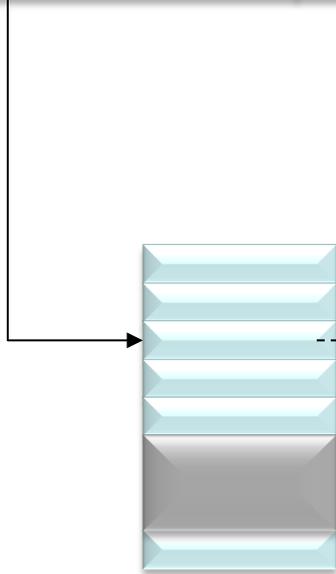
- TLB = Translation Look-aside Buffer





# Multi-Level Page Tables

Virtual Page Number



Physical Page Number



# TLB Miss?

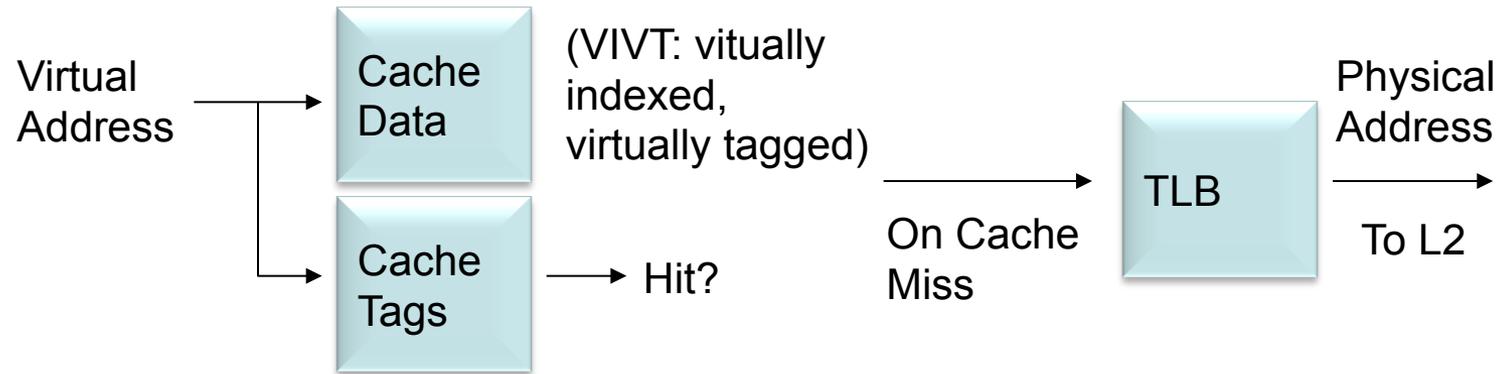
- Software solution
  - Generate an exception
  - O/S
- Hardware solution
  - Hardware page walker
  - TLB miss handler
  - Needs to know TLB miss in advance



# PAPT Cache

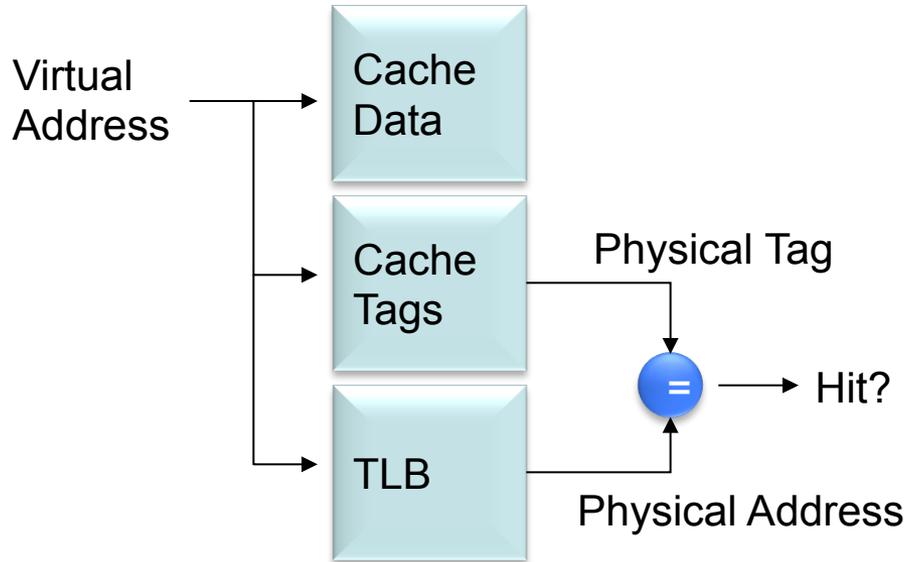
- So far we haven't differentiate physical and virtual addresses so much
- Previous slide showed Physically-Addressed Physically-Tagged cache
  - Sometimes called PIPT (I=Indexed)
- Con: TLB lookup and cache access serialized
  - Caches already take  $> 1$  cycle
- Pro: cache contents valid so long as page table not modified

# Virtually Addressed Cache



- Pro: latency – no need to check TLB
- Con: Cache must be flushed on process change

# Virtually Indexed Physically Tagged



Big page size  
can help here

- Pro: latency – TLB parallelized
- Pro: don't need to flush \$ on process swap
- Con: Limit on cache indexing (can only use bits *not* from the VPN/PPN)



# Virtual Index Physical Tag

Virtual Address



Physical Address



Good



BAD



# Programming

- Programming: Virtual or Physical ?
- Data sharing in parallel programming
  - Virtual or Physical ?
  - Different VAs need to mapped to the same PA
  - Virtual-index-physical-tag Cache
  - $VA1 = PA1 = \{\text{tag1}, \text{index1}, \text{offset1}\}$
  - $VA2 = PA1 = \{\text{tag1}, \text{index2}, \text{offset1}\}$



# Review question

- A computer has an 8KB write-through cache. Each cache block is 64 bits, the cache is 4-way set associative and uses the true LRU replacement policy. Assume a 24-bit address space and byte-addressable memory. How big (in bits) is the tag store



# # of LRU bits

- Assume true-LRU
  - 4-way            2 bits
  - 8-way            3 bits
  - 2-way            0.5 bit or 1 bit
- Pseudo LRU
  - Have fewer bits than true LRU
  - Less accurate but less complex (storage, logic)



# Review question

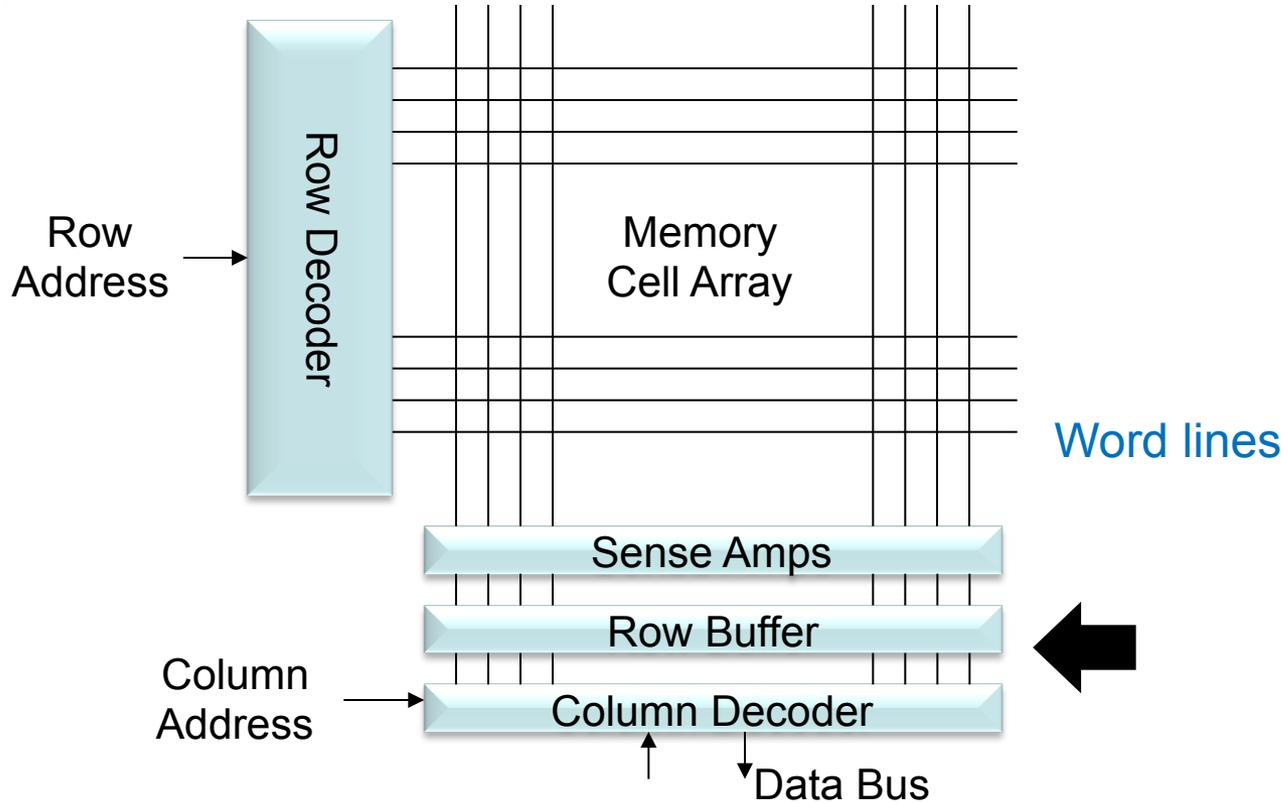
- A computer has an 8KB write-through cache. Each cache block is 64 bits, the cache is 4-way set associative and uses the true LRU replacement policy. Assume a 24-bit address space and byte-addressable memory. How big (in bits) is the tag store



# Review of DRAM

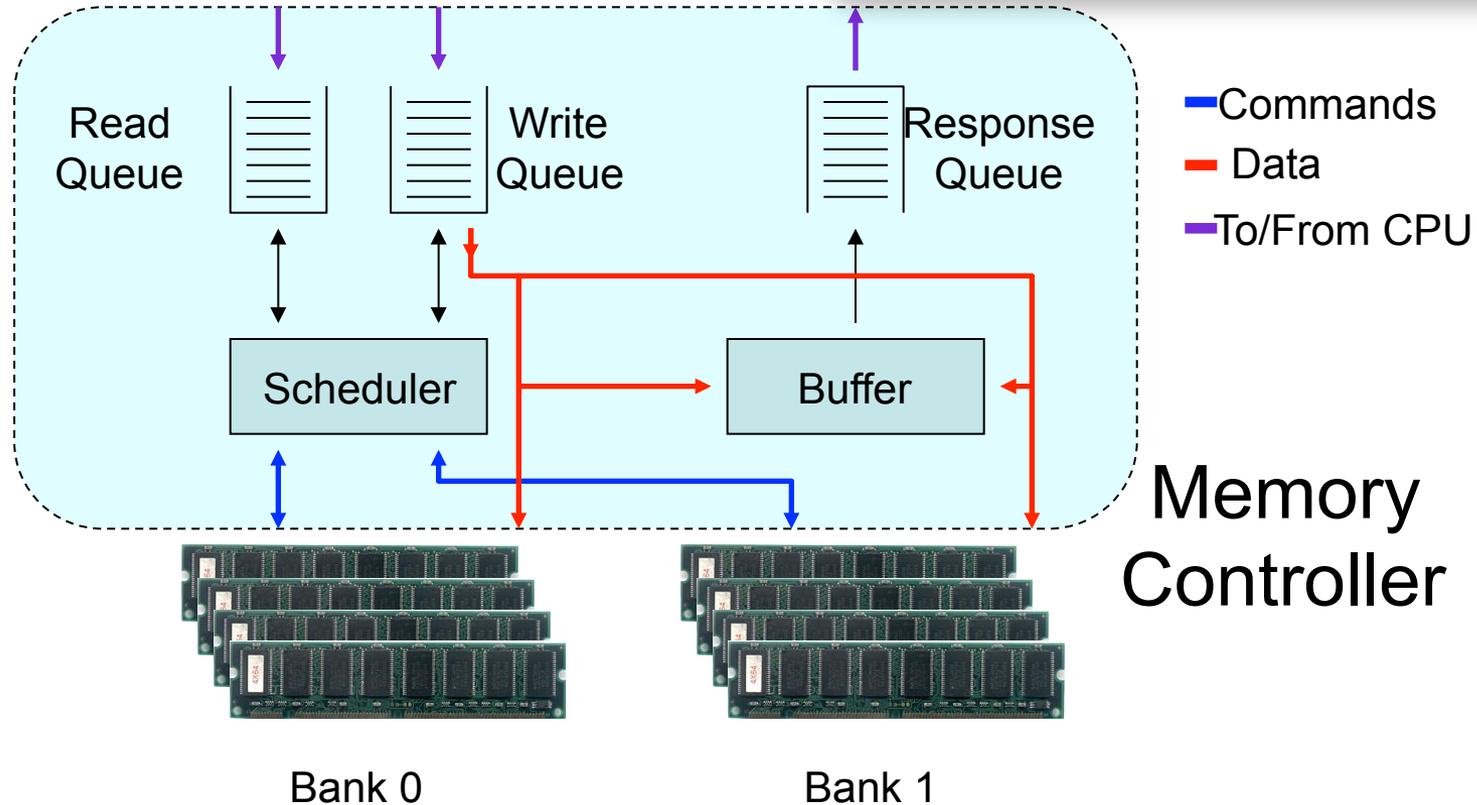
- Main characteristics
  - 1T vs. 6T
  - Destructive read
  - DRAM page
  - Sense amplifier
  - Burst mode

# DRAM Page/Buffer



# Memory Controller

Like Write-Combining Buffer, Scheduler may coalesce multiple accesses together, or re-order to reduce number of row accesses





# Task of Memory Controller

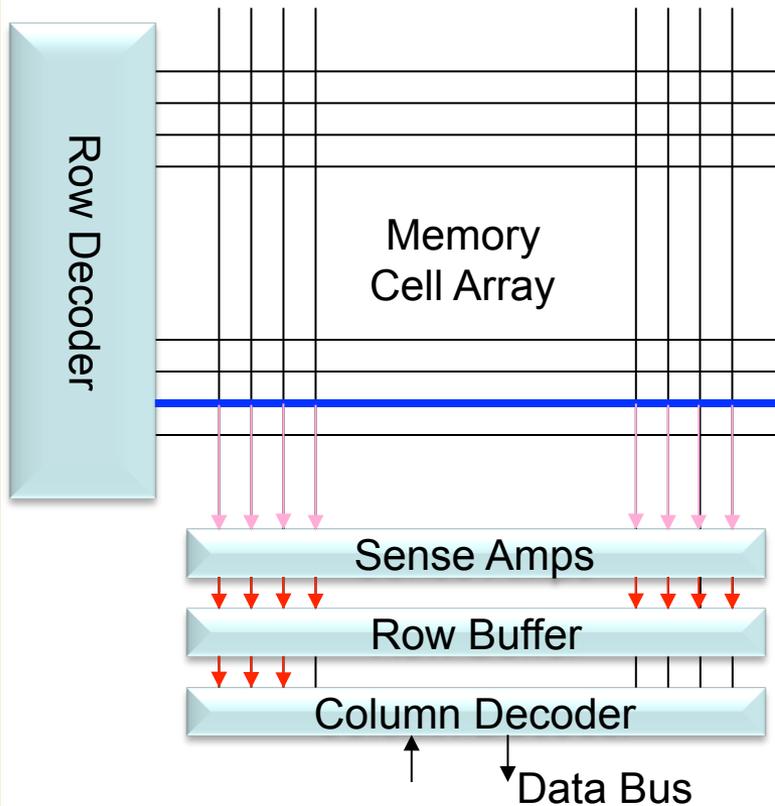
- Manage all data movement between the processor and the memory modules
- Read/Write
- Refresh/Precharge
- Memory request scheduling



# DRAM scheduling

- Scheduling memory requests in the dram system to increase the DRAM utilization
- Suggested Reading
  - Rixner et al., “[Memory Access Scheduling](#),” ISCA 2000.

# DRAM Read Operation



- 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 Read Latency

- 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)



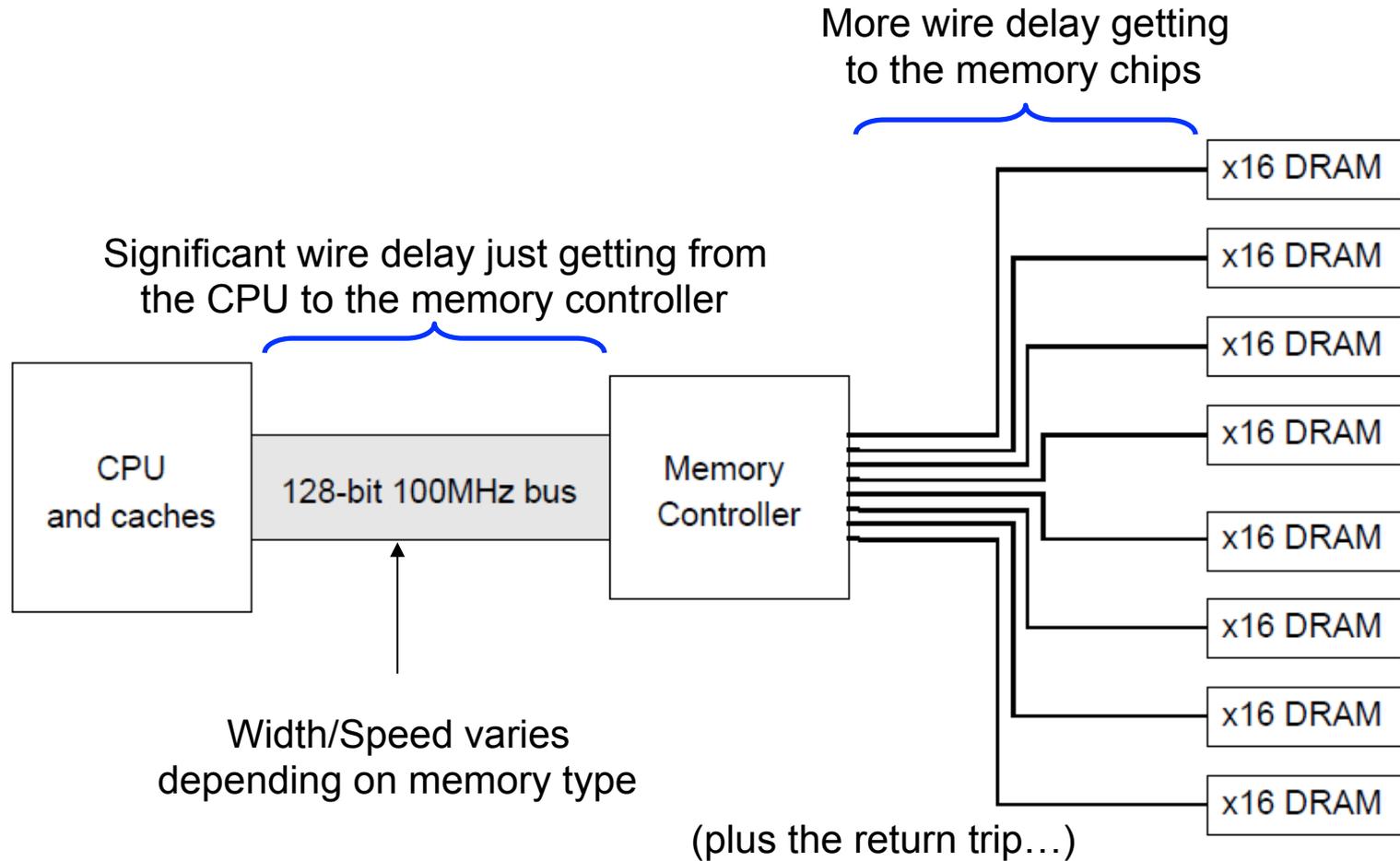
- **Open Page: Keep page open after read**
  - Pros:
    - Temporal, spatial locality
    - Latency is limited by tcas only
  - Cons:
    - Energy consumption, pay the cost of closing a page.
    - Page close+page open + ras + cas +bus transfer time
- **Closed Page: page close after read**
  - Good for random access patterns
  - Page open+ras+cas+bus transfer time



# Review & Outline

- DRAM scheduler: FCFS/ FRFCFS
- DRAM memory system organization

# Layout Latency





# DRAM Scheduling Policies-I

- **FCFS** (first come first served)
  - Oldest request first
- **FR-FCFS** (first ready, first come first served)
  1. Row-hit first
  2. Oldest first

Goal: Maximize row buffer hit rate → **maximize DRAM throughput**

  - Actually, scheduling is done at the **command level**
    - Column commands (read/write) prioritized over row commands (activate/precharge)
    - Within each group, older commands prioritized over younger ones



# DRAM Scheduling Policies-II

- A scheduling policy is essentially 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 core are dependent on it?



# Why are DRAM Controllers Difficult to Design?

- Need to obey **DRAM timing constraints** for correctness
  - There are many (50+) timing constraints in DRAM
  - tWTR: Minimum number of cycles to wait before issuing a read command after a write command is issued
  - tRC: Minimum number of cycles between the issuing of two consecutive activate commands to the same bank
  - ...
- Need to **keep track of many resources** to prevent conflicts
  - Channels, banks, ranks, data bus, address bus, row buffers
- Need to handle **DRAM refresh**
- Need to optimize for performance (in the presence of constraints)
  - Reordering is not simple
  - Predicting the future?



# Example Memory Latency Computation

- FSB freq = 200 MHz, SDRAM
- RAS delay = 2, CAS delay = 2, Precharge = 2
- Scheduling in memory controller
- Scheduler queue size = 6  
A0, A1, B0, C0, D3, A2, D0, C1, A3, C3, C2, D1, B1, D2
- FCFS time?
  
- FRFCFS time?  
A0, A1, A2, B0, C0, C1, C3, C2, D3, D0, D1, D2, A3, B1

# Rank/bank/row/column/channel

- Bank, row, column → DRAM chip configuration
  - Banks: different banks can be operated independently
- Rank → a set of DRAM devices that operate in **lockstep** fashion to command in a memory (i.e. chips inside the same rank are accessed simultaneously)
- Channel → CPU and memory communication channel

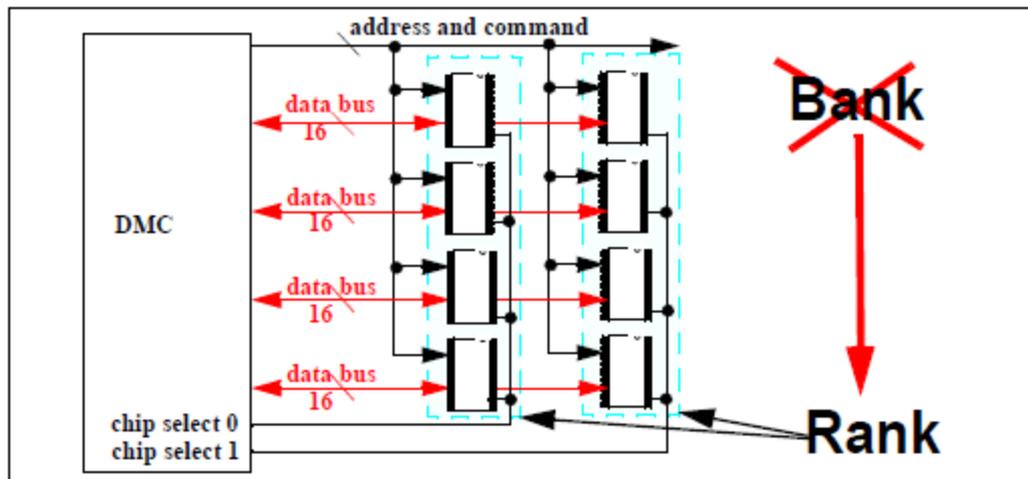
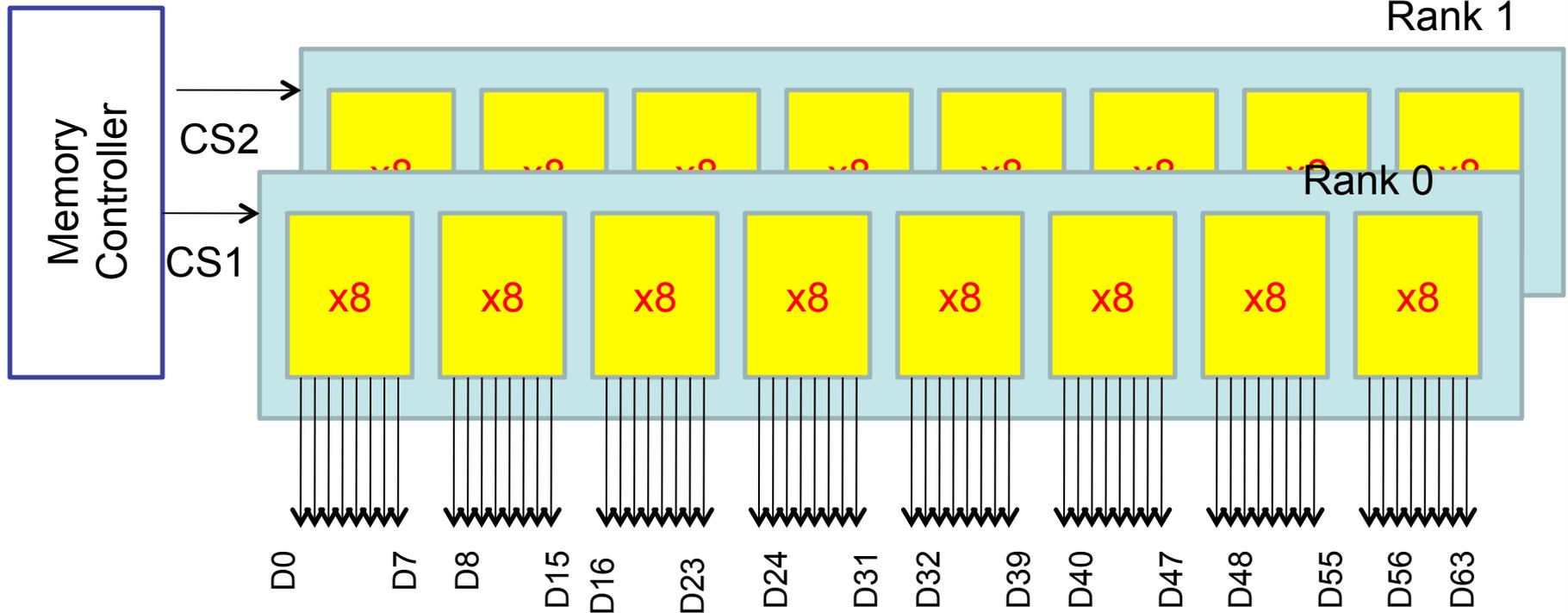
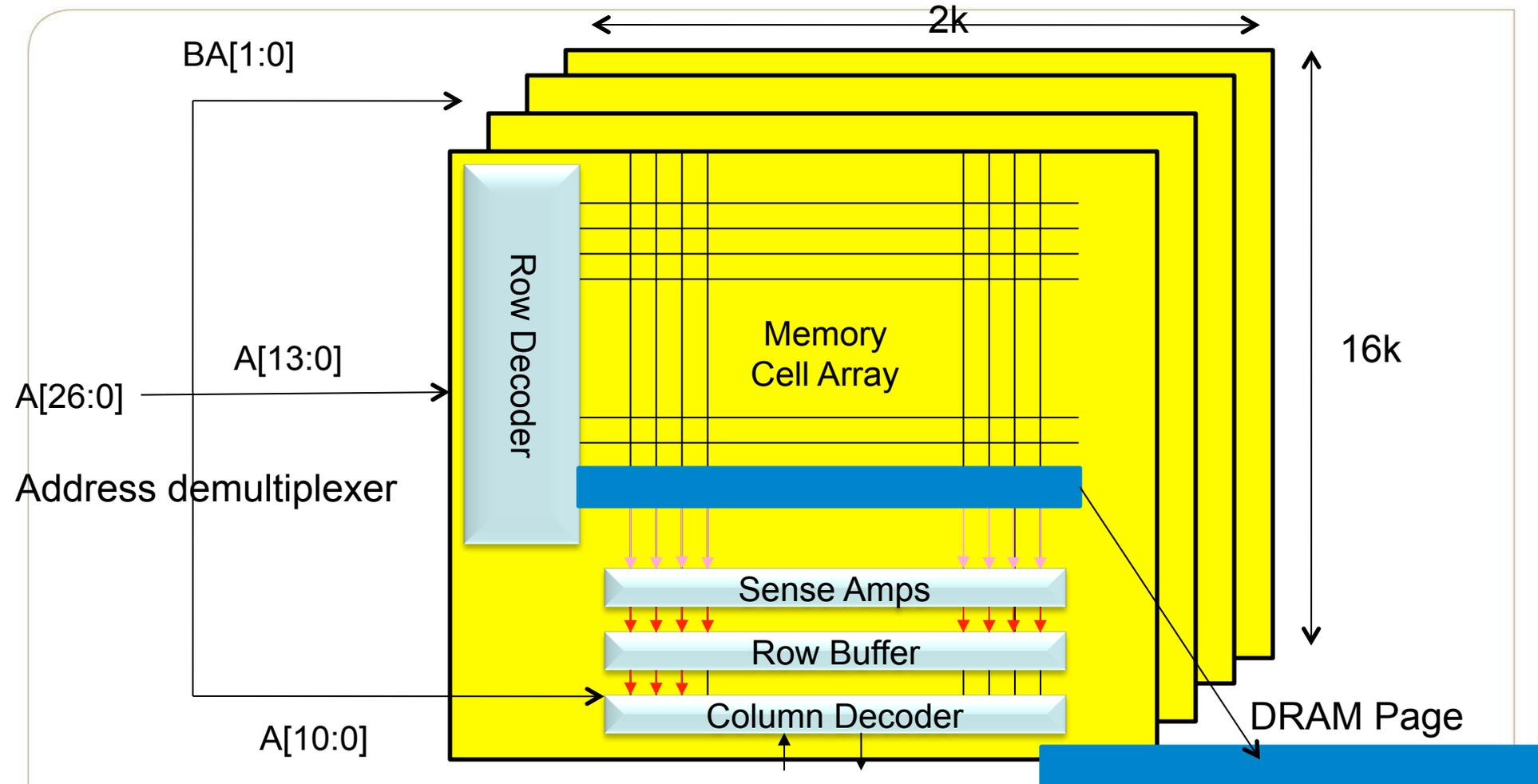


Figure 3.5: Memory System with 2 ranks of DRAM devices.

# DRAM Ranks



# DRAM BANK: 512MB 4-bank DRAM



Data Outs D[3:0] A DRAM Page =  $2k \times 4B = 8KB$

# BANKS

Figure 3.6 shows an SDRAM device with 4 banks. Modern DRAM devices contain

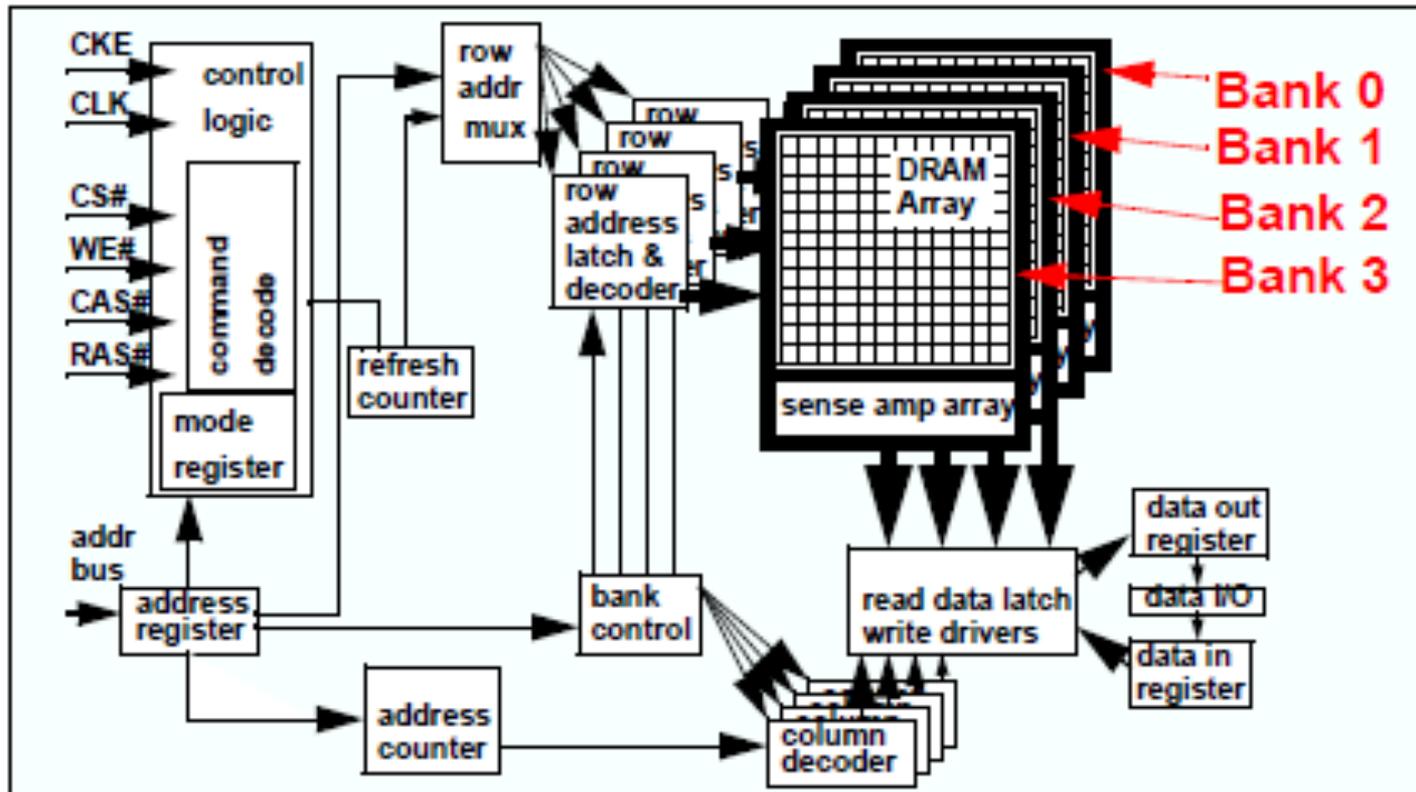


Figure 3.6: SDRAM device with 4 banks of DRAM arrays internally.



# Bank & Interleaving

	Bank id
--	---------

0
1
2
3

4
5
6
7

8
9
10
11

12
13
14
15

	Bank id
--	---------

0
4
8
12

1
5
9
13

2
6
10
14

3
7
11
15

- Interleaving: why?

# Column Size

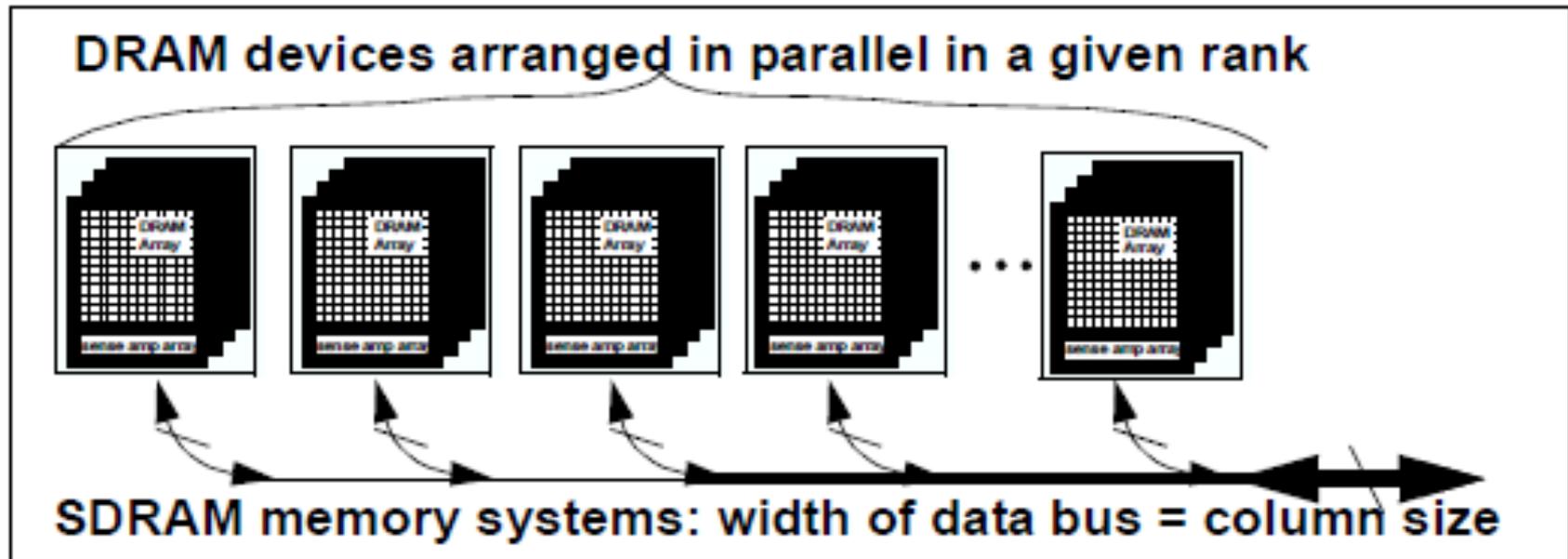
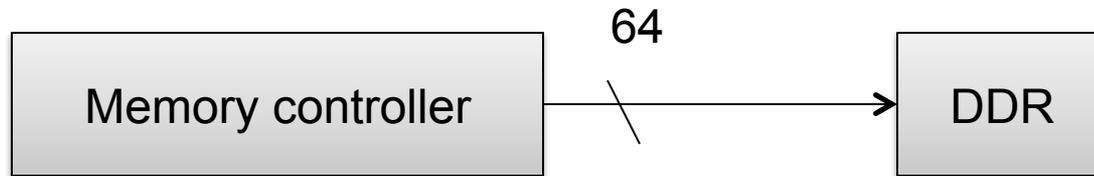
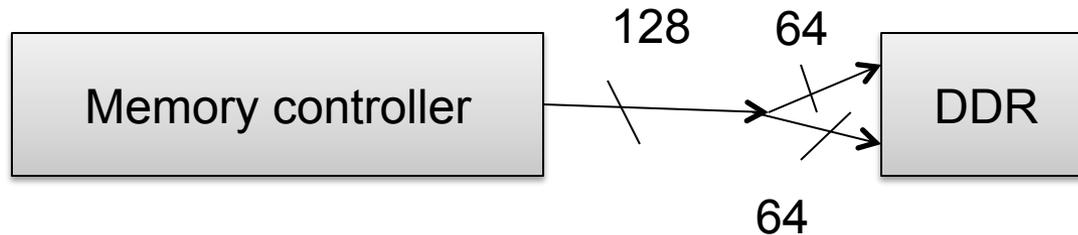


Figure 3.8: Classical DRAM system topology, width of data bus equals column size.

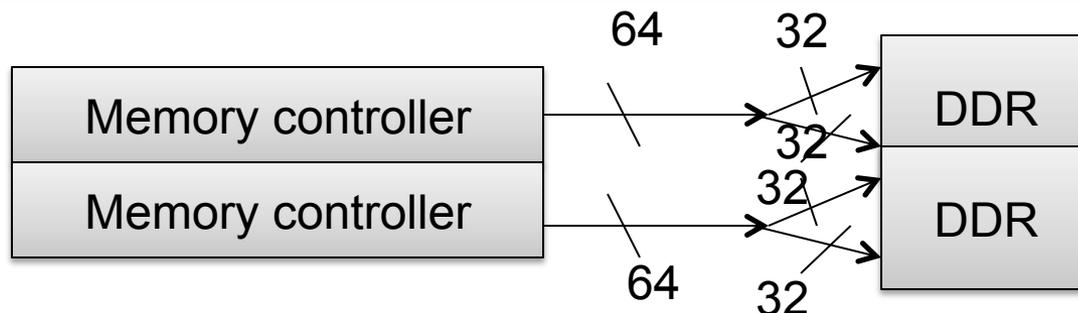
# Channel



- One physical channel of 64 bit width



- Two physical channel of 64bit wide busses
- One logical channel



- Two channels: 64bit wide per channel

# Mesh Topology

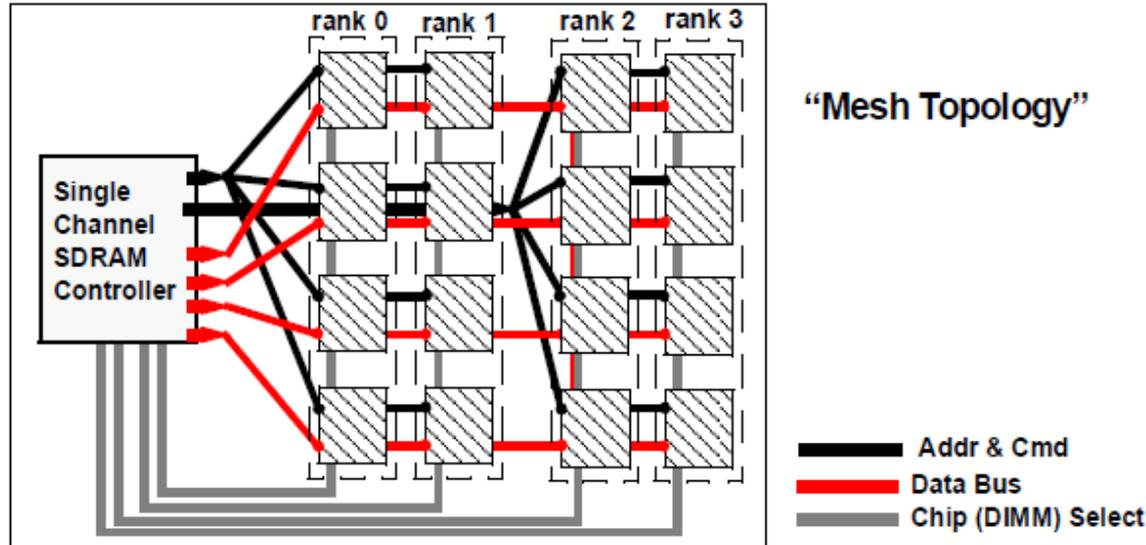


Figure 3.16: Topology of a generic DRAM memory system.

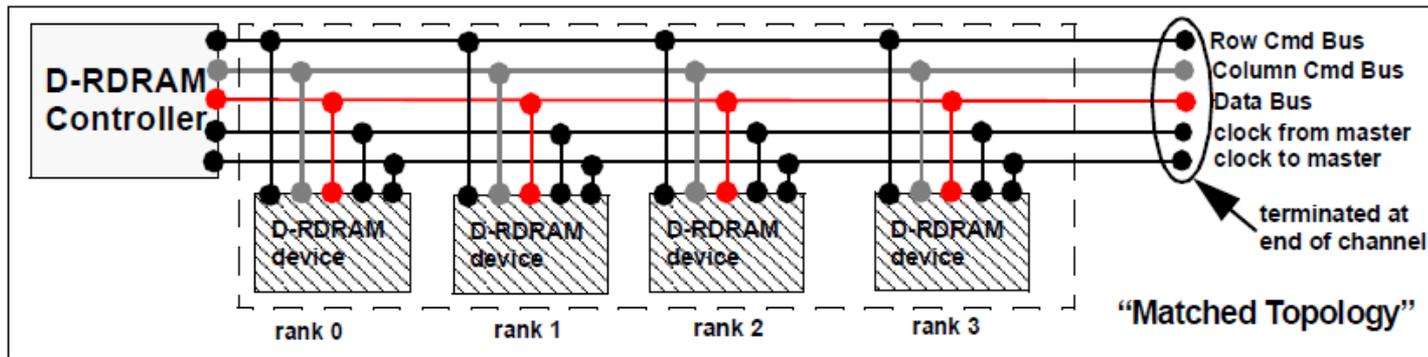
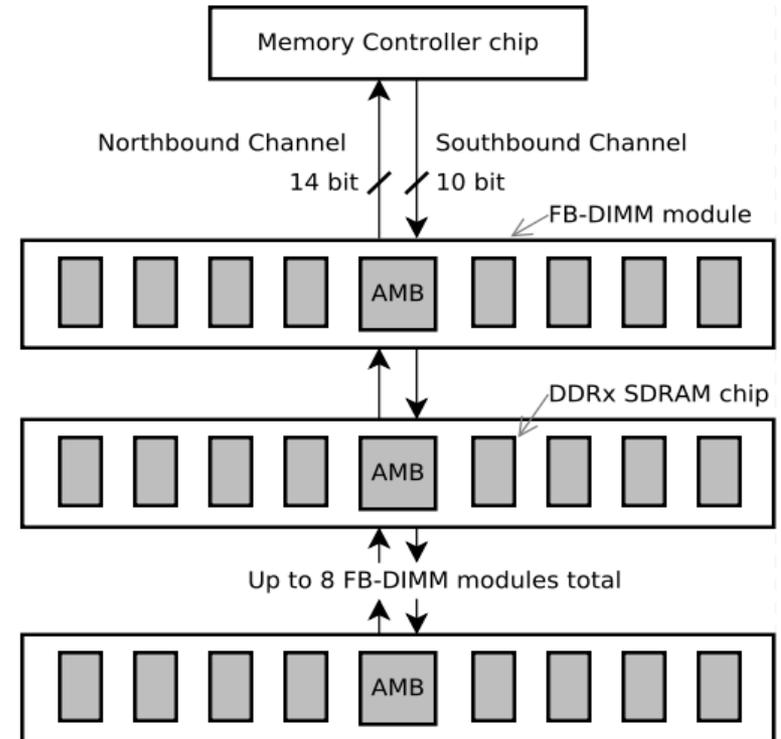


Figure 3.17: Topology of a generic Direct RDRAM memory system.



# FB-DIMM

- AMB (advanced memory buffer)
- Each DIMM has their own DIMM memory controller
- Increase bandwidth
- ~ DDR2



# Memory




**PATRIOT**  
MEMORY

●●●●● (44)

Compare

**Intel P67 platform/ XMP ready**  
Patriot Gamer 2 Series 8GB (2 x 4GB) 240-Pin DDR3 SDRAM DDR3 1333 (PC3 10666) Desktop Memory

- DDR3 1333 (PC3 10666)
- Timing 9-9-9-24
- Voltage 1.65V



**CORSAIR**

●●●●● (8)

Compare

**CORSAIR Vengeance 8GB (2 x 4GB) 240-Pin DDR3 SDRAM DDR3 1866 (PC3 15000) Desktop Memory Model**

[View Details](#)

- DDR3 1866 (PC3 15000)
- Timing 9-10-9-27
- Cas Latency 9



**crucial**

●●●●● (9)

Compare

**Crucial Ballistix sport 4GB (2 x 2GB) 240-Pin DDR3 SDRAM DDR3 1600 Desktop Memory Model**

- DDR3 1600
- Timing 10-10-10-28
- Cas Latency 10



**G.SKILL**

●●●●● (639)

Compare

**Free 4GB SDHC flash card w/ purchase, limited offer**  
G.SKILL Ripjaws Series 8GB (2 x 4GB) 240-Pin DDR3 SDRAM DDR3 1600 (PC3 12800) Desktop Memory

- DDR3 1600 (PC3 12800)
- Timing 9-9-9-24-2N
- Cas Latency 9



# Naming

CL (CAS):ck cycles between sending a column address to the memory and the beginning of the data in response

tRCD: Clock cycles between RAS to CAS delay

tRP: Clock cycles between row precharge and activate (PRE)

tRC: from RAS to read&write

Standard name	Memory clock (MHZ)	Cycle time (ns)	I/O bus clock (MHz)	Data rate (MT/s)	Peak transfer rate (MB/s)	Timing (CL-tRCD-tRP)	CAS latency (ns)
DDR3-1333	166.66	6	666.66	1333.33	10666.66	7-7-7-7 8-8-8-8 ....	10.5 12 ....

DDR3-M transfer second

I/O frequency =  $\frac{1}{2}$  M transfer frequency

DIMM name = M transfer second \*2 (dual) \* 8B

e.g.) DDR3-1600 = PC12800 = 1600\*2\*8

# 240 Pin



**Table 5: Pin Assignments**

240-Pin DDR3 UDIMM Front								240-Pin DDR3 UDIMM Back							
Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol
1	VREFDQ	31	DQ25	61	A2	91	DQ41	121	Vss	151	Vss	181	A1	211	Vss
2	Vss	32	Vss	62	VDD	92	Vss	122	DQ4	152	DM3	182	VDD	212	DM5
3	DQ0	33	DQS3#	63	CK1	93	DQS5#	123	DQ5	153	NC	183	VDD	213	NC
4	DQ1	34	DQS3	64	CK1#	94	DQS5	124	Vss	154	Vss	184	CK0	214	Vss
5	Vss	35	Vss	65	VDD	95	Vss	125	DM0	155	DQ30	185	CK0#	215	DQ46
6	DQS0#	36	DQ26	66	VDD	96	DQ42	126	NC	156	DQ31	186	VDD	216	DQ47
7	DQS0	37	DQ27	67	VREFCA	97	DQ43	127	Vss	157	Vss	187	NC	217	Vss
8	Vss	38	Vss	68	NC	98	Vss	128	DQ6	158	NC	188	A0	218	DQ52
9	DQ2	39	NC	69	VDD	99	DQ48	129	DQ7	159	NC	189	VDD	219	DQ53
10	DQ3	40	NC	70	A10	100	DQ49	130	Vss	160	Vss	190	BA1	220	Vss
11	Vss	41	Vss	71	BA0	101	Vss	131	DQ12	161	NC	191	VDD	221	DM6
12	DQ8	42	NC	72	VDD	102	DQS6#	132	DQ13	162	NC	192	RAS#	222	NC
13	DQ9	43	NC	73	WE#	103	DQS6	133	Vss	163	Vss	193	S0#	223	Vss
14	Vss	44	Vss	74	CAS#	104	Vss	134	DM1	164	NC	194	VDD	224	DQ54
15	DQS1#	45	NC	75	VDD	105	DQ50	135	NC	165	NC	195	ODT0	225	DQ55
16	DQS1	46	NC	76	NC	106	DQ51	136	Vss	166	Vss	196	A13	226	Vss
17	Vss	47	Vss	77	NC	107	Vss	137	DQ14	167	NC	197	VDD	227	DQ60
18	DQ10	48	NC	78	VDD	108	DQ56	138	DQ15	168	RESET#	198	NC	228	DQ61
19	DQ11	49	NC	79	NC	109	DQ57	139	Vss	169	NC	199	Vss	229	Vss
20	Vss	50	CKE0	80	Vss	110	Vss	140	DQ20	170	VDD	200	DQ36	230	DM7
21	DQ16	51	VDD	81	DQ32	111	DQS7#	141	DQ21	171	NC	201	DQ37	231	NC
22	DQ17	52	BA2	82	DQ33	112	DQS7	142	Vss	172	NC/A14 <sup>1</sup>	202	Vss	232	Vss
23	Vss	53	NC	83	Vss	113	Vss	143	DM2	173	VDD	203	DM4	233	DQ62
24	DQS2#	54	VDD	84	DQS4#	114	DQ58	144	NC	174	A12	204	NC	234	DQ63
25	DQS2	55	A11	85	DQS4	115	DQ59	145	Vss	175	A9	205	Vss	235	Vss
26	Vss	56	A7	86	Vss	116	Vss	146	DQ22	176	VDD	206	DQ38	236	VDDSPD
27	DQ18	57	VDD	87	DQ34	117	SA0	147	DQ23	177	A8	207	DQ39	237	SA1
28	DQ19	58	A5	88	DQ35	118	SCL	148	Vss	178	A6	208	Vss	238	SDA
29	Vss	59	A4	89	Vss	119	SA2	149	DQ28	179	VDD	209	DQ44	239	Vss
30	DQ24	60	VDD	90	DQ40	120	VTT	150	DQ29	180	A3	210	DQ45	240	VTT

Notes: 1. Pin 172 is NC for 1GB and A14 for 2GB.



# Ann.

- L3 will be posted by tonight.
- Cache & DRAM (DRAM page) & MSHR
- Due (10/20)
  
- Exam & Lab 2 grade: will be posted by tonight.
- You can pick up your exam paper
  - Friday 4-5 pm (or send email)