

# ECE 485/585

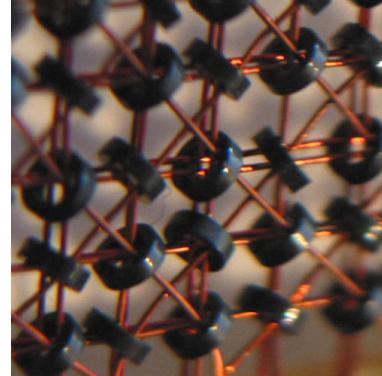
# Microprocessor System Design

Prof. Mark G. Faust

Maseeh College of Engineering  
and Computer Science

**PORTLAND STATE  
UNIVERSITY**

# Memory



**"640K ought to be enough for anybody."**

-- Bill Gates, 1981

# Memory

- Taxonomy of Memories
- Memory Hierarchy
- SRAM
  - Basic Cell, Devices, Timing
- Memory Organization
  - Multiple banks, interleaving
- DRAM
  - Basic Cell, Timing
  - DRAM Evolution
- DRAM modules
- Error Correction
- Memory Controllers

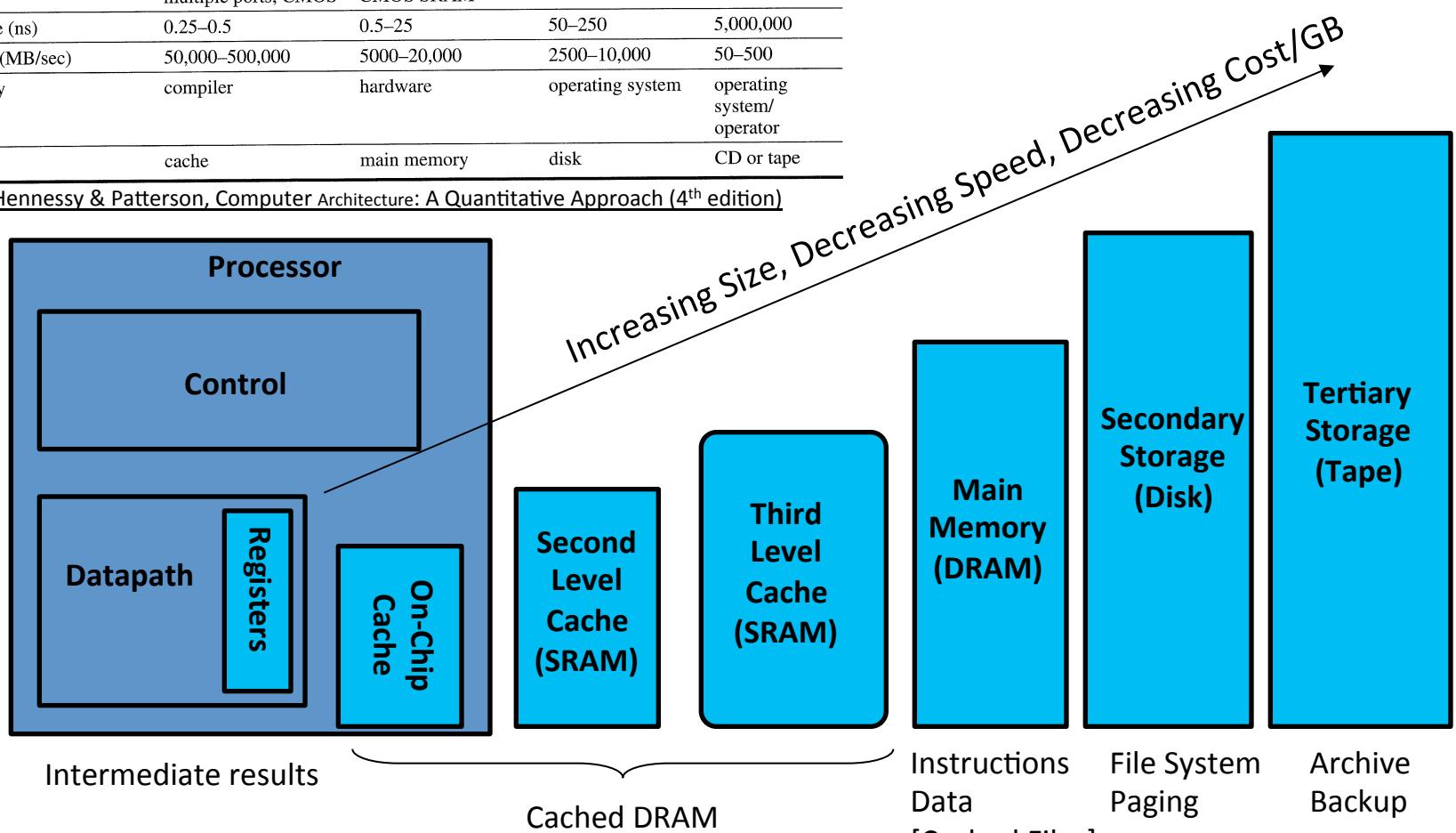
# Memory Taxonomy

Read/Write Memory		Read Only
Volatile	Non-Volatile	
Non-Random Access	Random Access	
Shift Register FIFO CAM	SRAM DRAM	EPROM E <sup>2</sup> PROM Flash NAND NOR NVRAM
		Mask ROM PROM

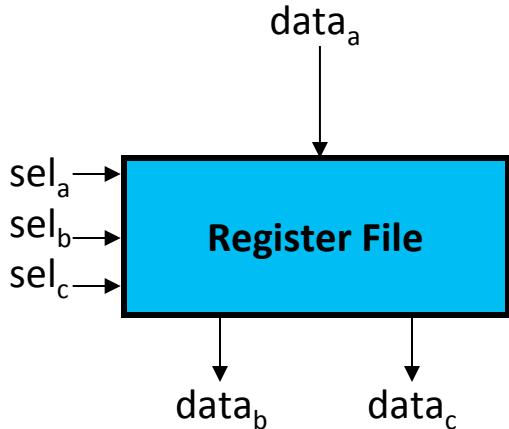
# Computer Memory Hierarchy

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	< 16 MB	< 512 GB	> 1 TB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25–0.5	0.5–25	50–250	5,000,000
Bandwidth (MB/sec)	50,000–500,000	5000–20,000	2500–10,000	50–500
Managed by	compiler	hardware	operating system	operating system/operator
Backed by	cache	main memory	disk	CD or tape

From Hennessy & Patterson, Computer Architecture: A Quantitative Approach (4<sup>th</sup> edition)

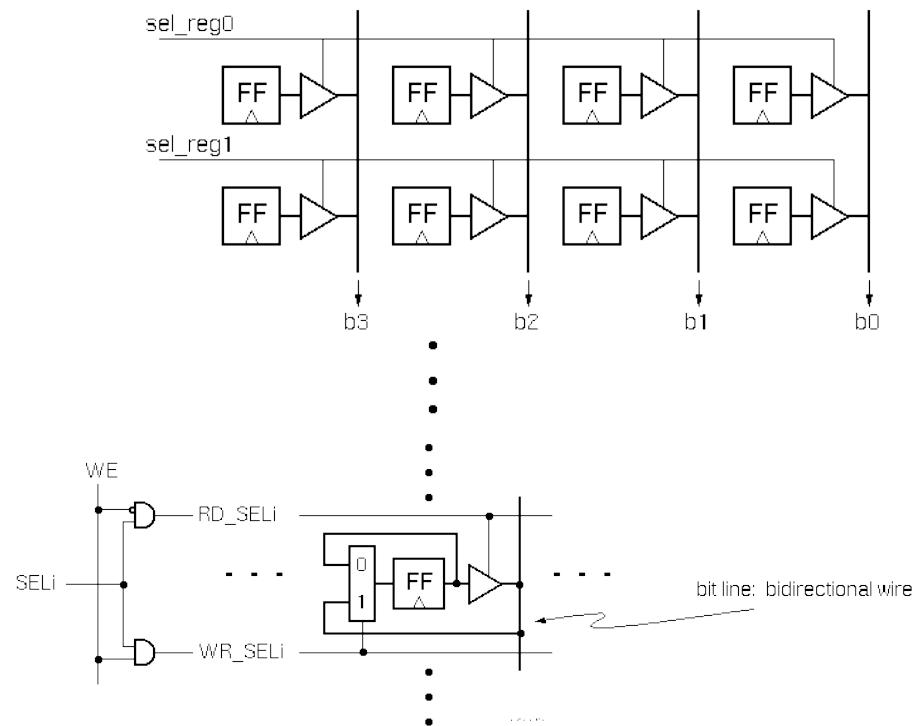


# Register Files

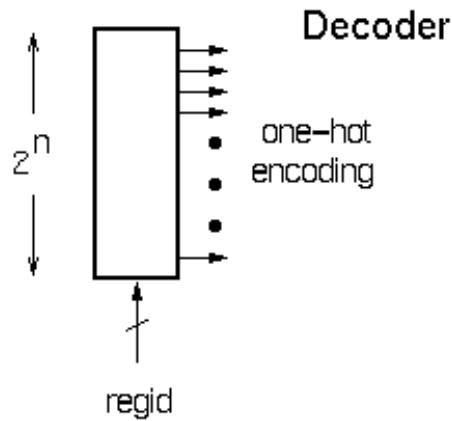


- For read operations, the register file is equivalent to a 2-D array of flip-flops with tri-state outputs
- For write operations, we add some additional circuitry to the basic cell

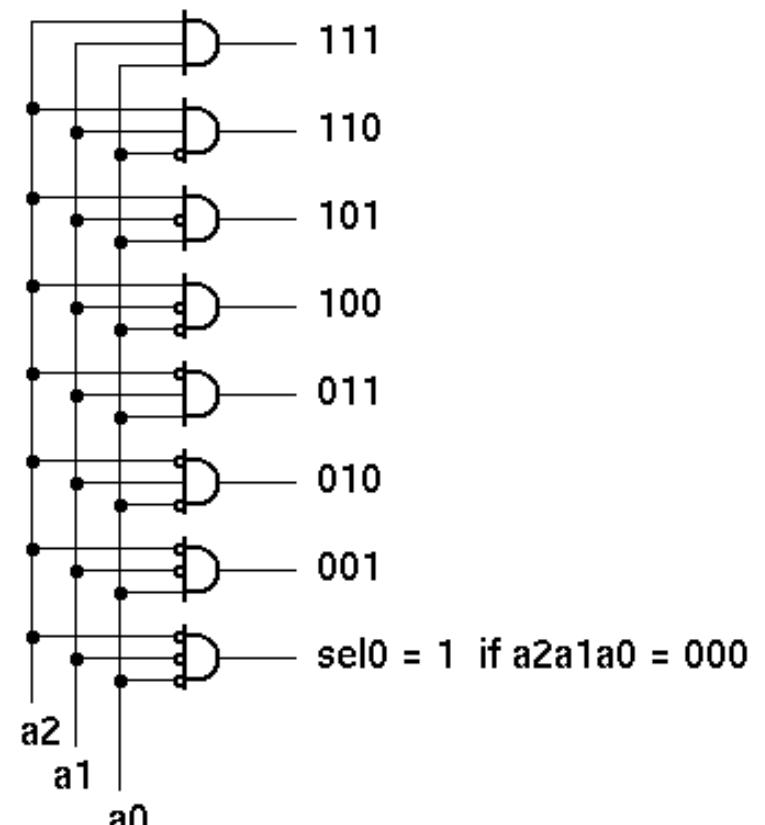
- General Purpose Registers
- Usually have multiple ports
  - Support CPU architecture's datapaths
  - Ability to read two operands, write one
- Operate at CPU speed



# Address Decoding



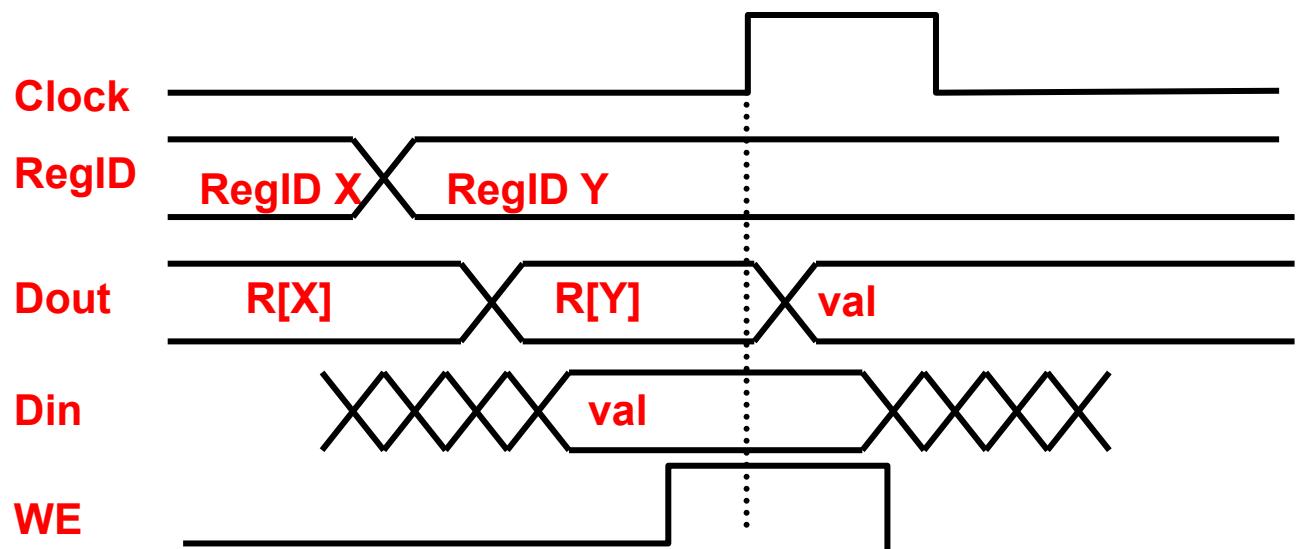
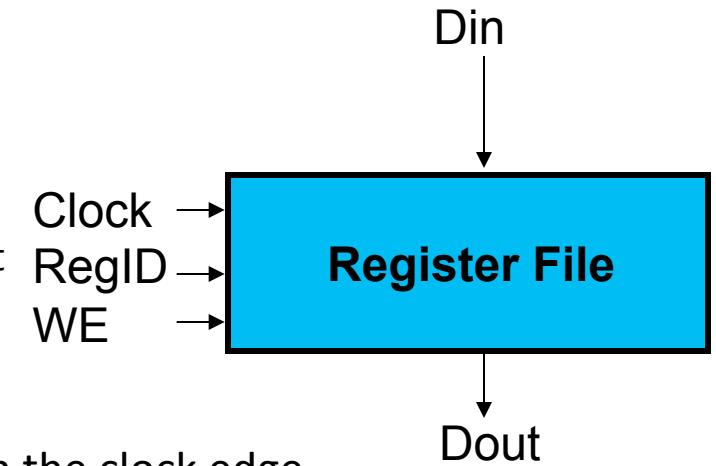
regid		sel_reg1	sel_reg0
000	00000001		
001	00000010		
010	00000100		
011	00001000		
100	00010000		
101	00100000		
110	01000000		
111	10000000		



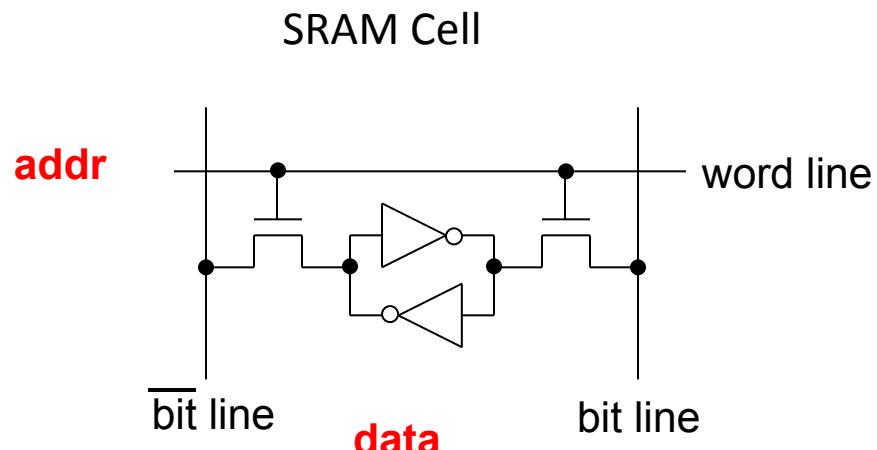
- Address decoder generates a one-hot code (1-of-n code) from the address
  - binary to unary
- The output is used for row selection

# Accessing Register Files

- Read
  - “Address following”
    - Change address
      - Data from new address appears on output
    - Asynchronous
- Write is synchronous
  - If WE, input data is written to selected word on the clock edge



# SRAM Technology



Which will be longer: bit lines or word lines?

Bit lines!

For density and low power, want tiny transistors

Unable to drive long bit lines

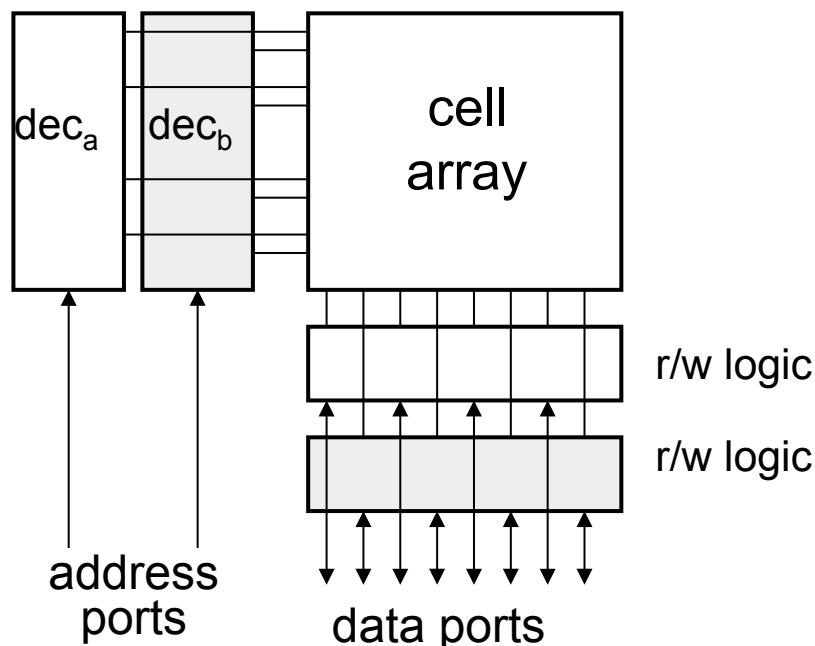
Pre-charge bit lines ( $V_{dd}/2$ ) before read

Use differential between bit and  $\bar{bit}$

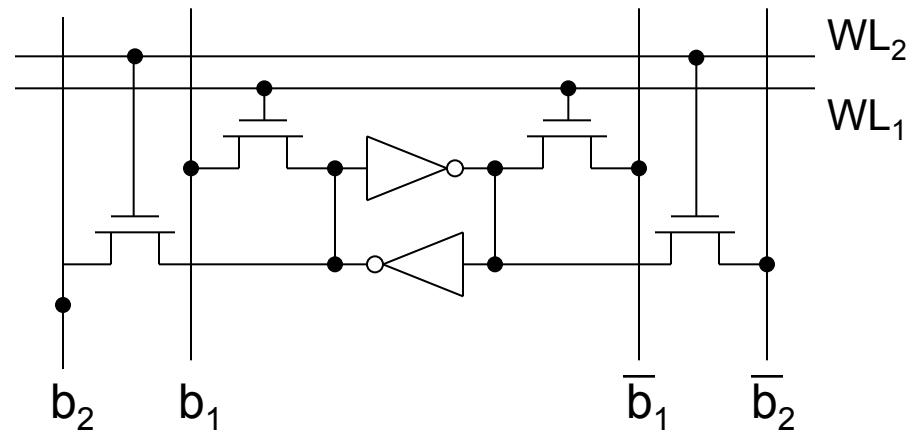
- Write
  - Write bit and  $\bar{bit}$  onto bit lines
  - Select desired word (“row”)
  - Turns on pass transistors
  - Writes new value to cell
  - [One inverter input will be low, turning its output high]
- Read
  - Select desired word (“row”)
  - One bit line will be pulled low
  - Other will remain high
  - Takes long time for bit line to be pulled low with tiny transistor
  - Don’t need to wait – can just sense difference between two bit lines!

# Dual-ported Memory Internals

- Add decoder, another set of read/write logic, bits lines, word lines



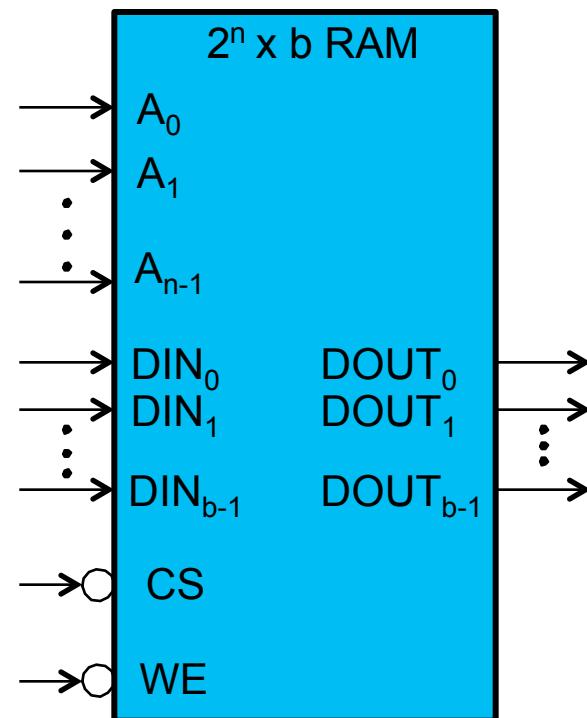
- Example cell: SRAM



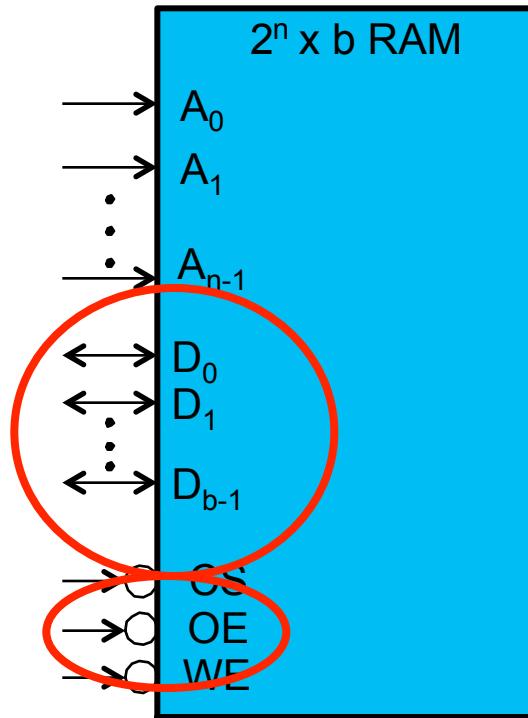
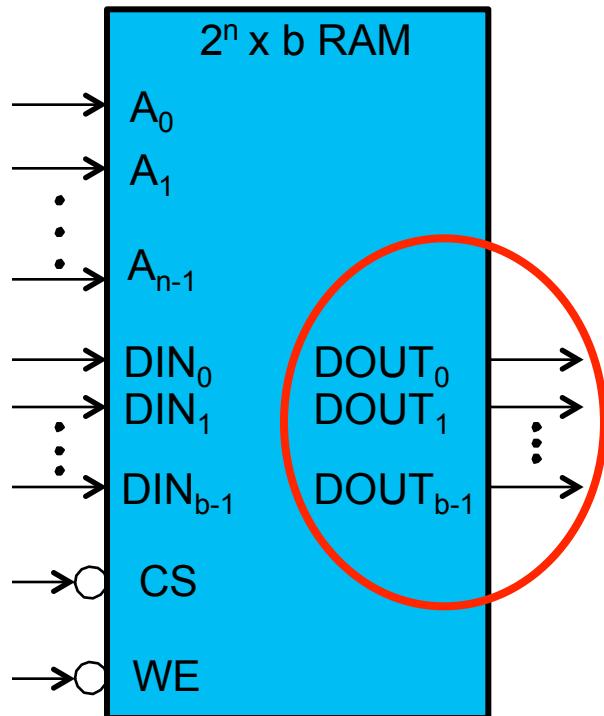
- Repeat everything but cross-coupled inverters.
- This scheme extends up to a couple more ports, then need to add additional transistors.

# Basic SRAM

- Size in bits (organization)
  - 1Mb ( $256K \times 4$ )  $\rightarrow$  256K words of 4 bits
  - 1Mb ( $128K \times 8$ )  $\rightarrow$  128K words of 8 bits
- Most Control Signals are Active Low
- Chip Select (/CS) effectively an enable
- Write Enable (/WE) controls read/write
- Write
  - /WE is asserted (Low)
  - /CS is asserted (Low)
- Read
  - /WE is de-asserted (High)
  - /CS is asserted (Low)

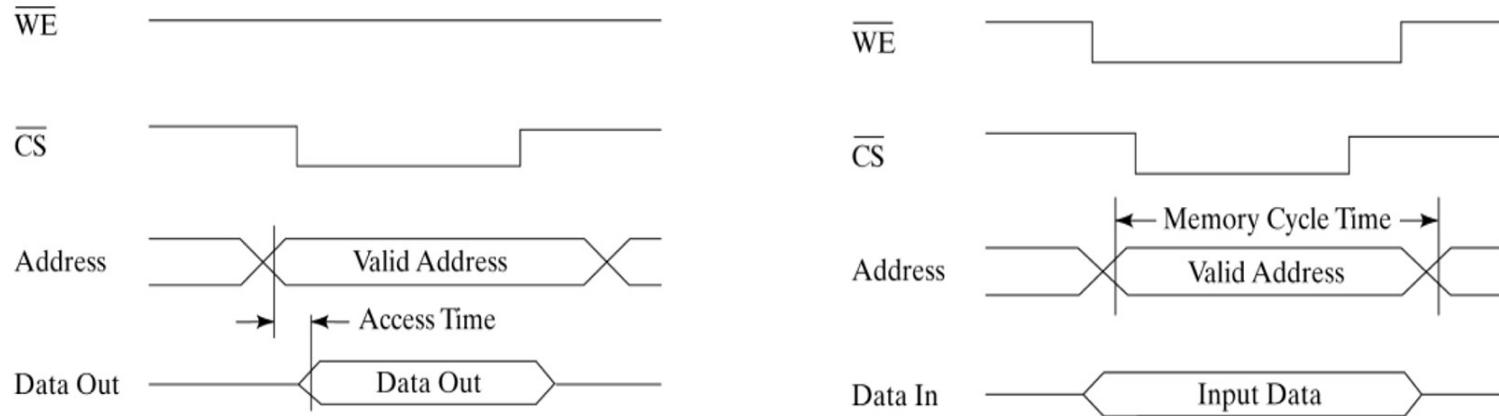


# SRAM Variations



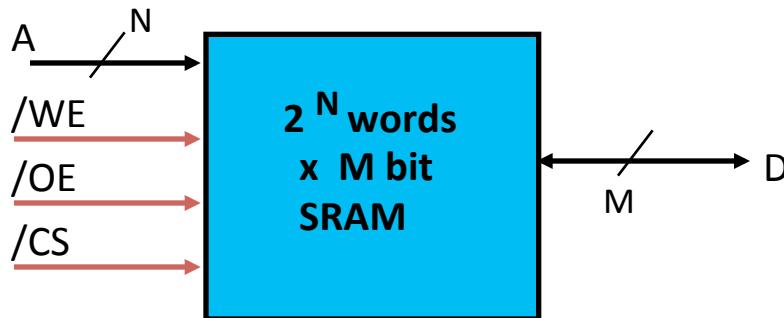
- Dedicated Din & Dout
  - Trade pin count (\$) for higher performance
  - No bidirectional “turnaround” time required
- Din & Dout often combined to save pins (\$)
- A new control signal, Output Enable (/OE)

# Simplified SRAM timing diagram



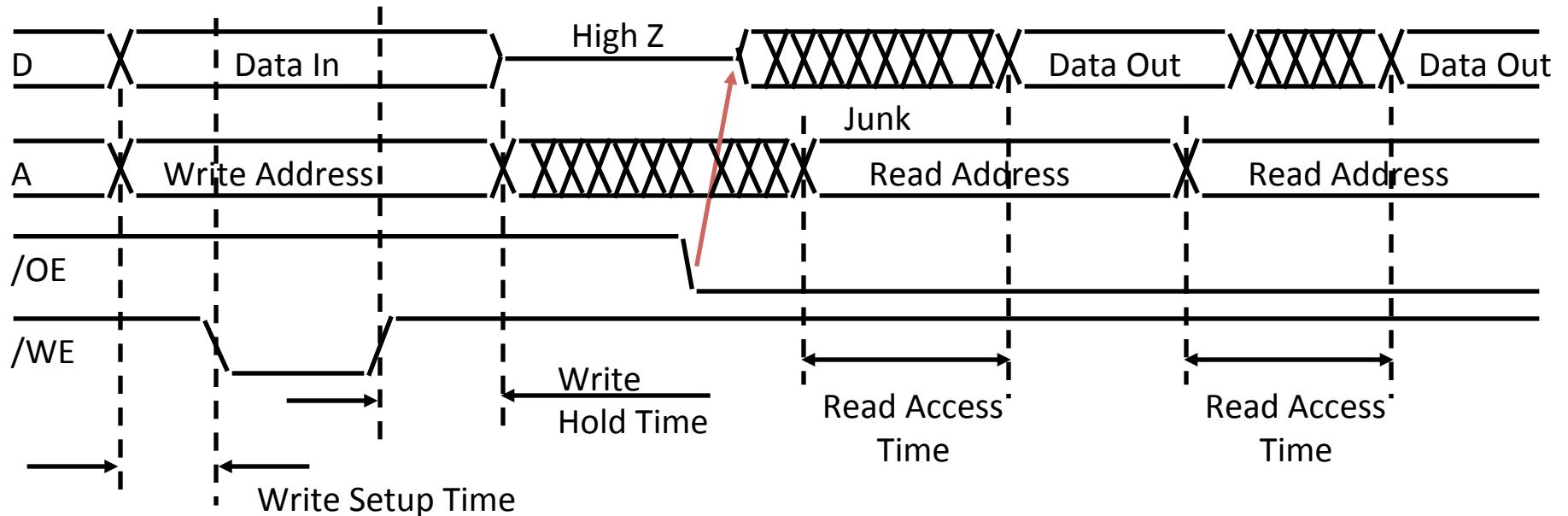
- Read: Valid address, then /CS (Chip Select) asserted
- Access Time: Address good to data valid
- Cycle Time: Minimum time between subsequent memory operations
- Write: Valid address and data with /WE asserted, then /CS asserted
  - Address must be stable a setup time before /WE and /CS go low
  - And hold time after one goes high

# Typical SRAM Timing

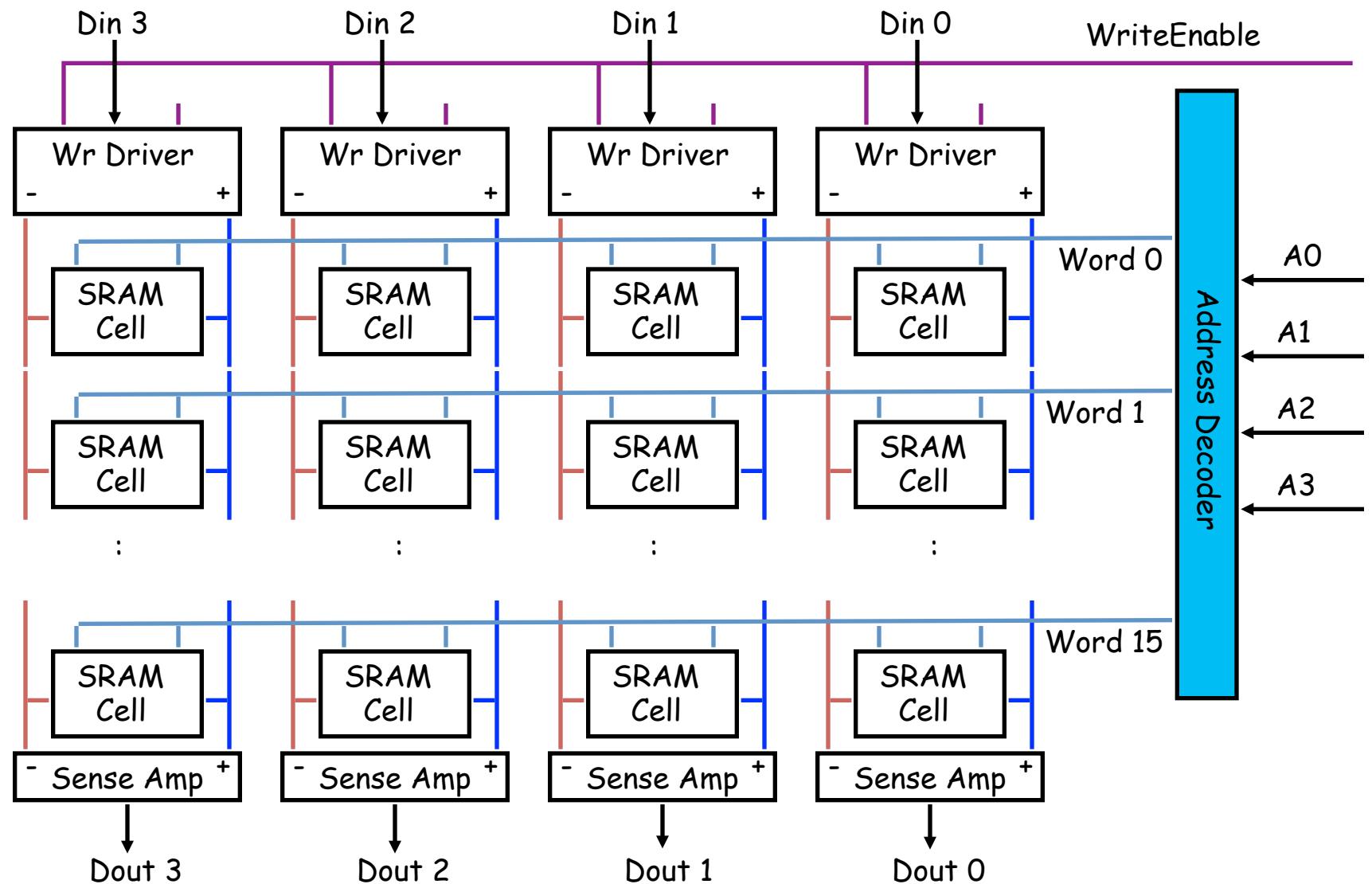


/OE determines direction  
Hi = Write, Lo = Read

Write Timing:

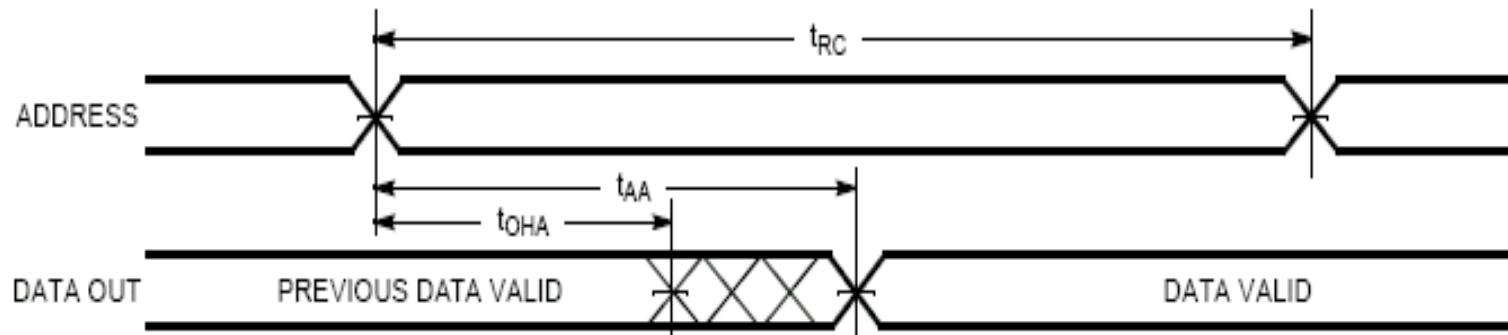


# Internal SRAM Organization (16x4)



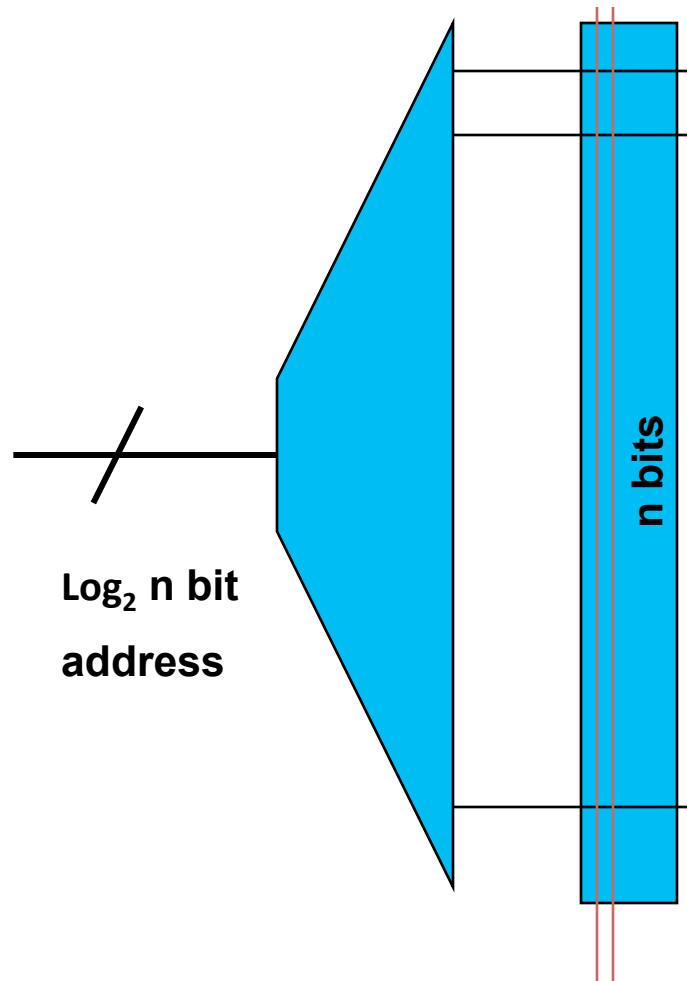
# Example: Cypress SRAM

Read Cycle No. 1<sup>[11, 12]</sup>



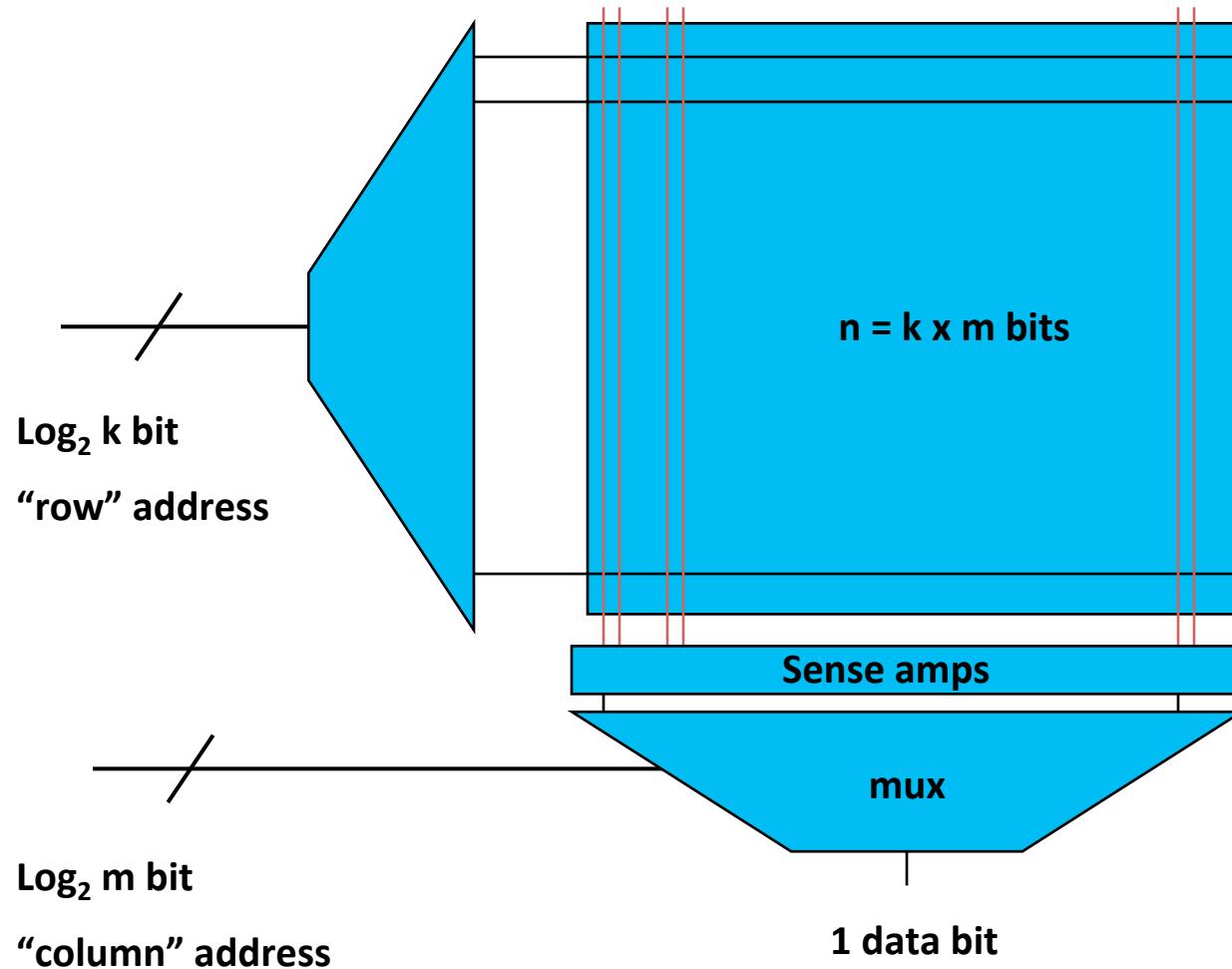
- Note “address following” mode
- Key SRAM timing parameters
  - $t_{AA}$  – Address access time: time between a valid address being applied and valid data available on data outputs
  - $t_{RC}$  – Read cycle time: Minimum time that one address must be held on the address lines before a second address can be presented
- $t_{AA}$  represents latency
- $t_{RC}$  represents bandwidth (throughput)

# What happens as number of bits increases?

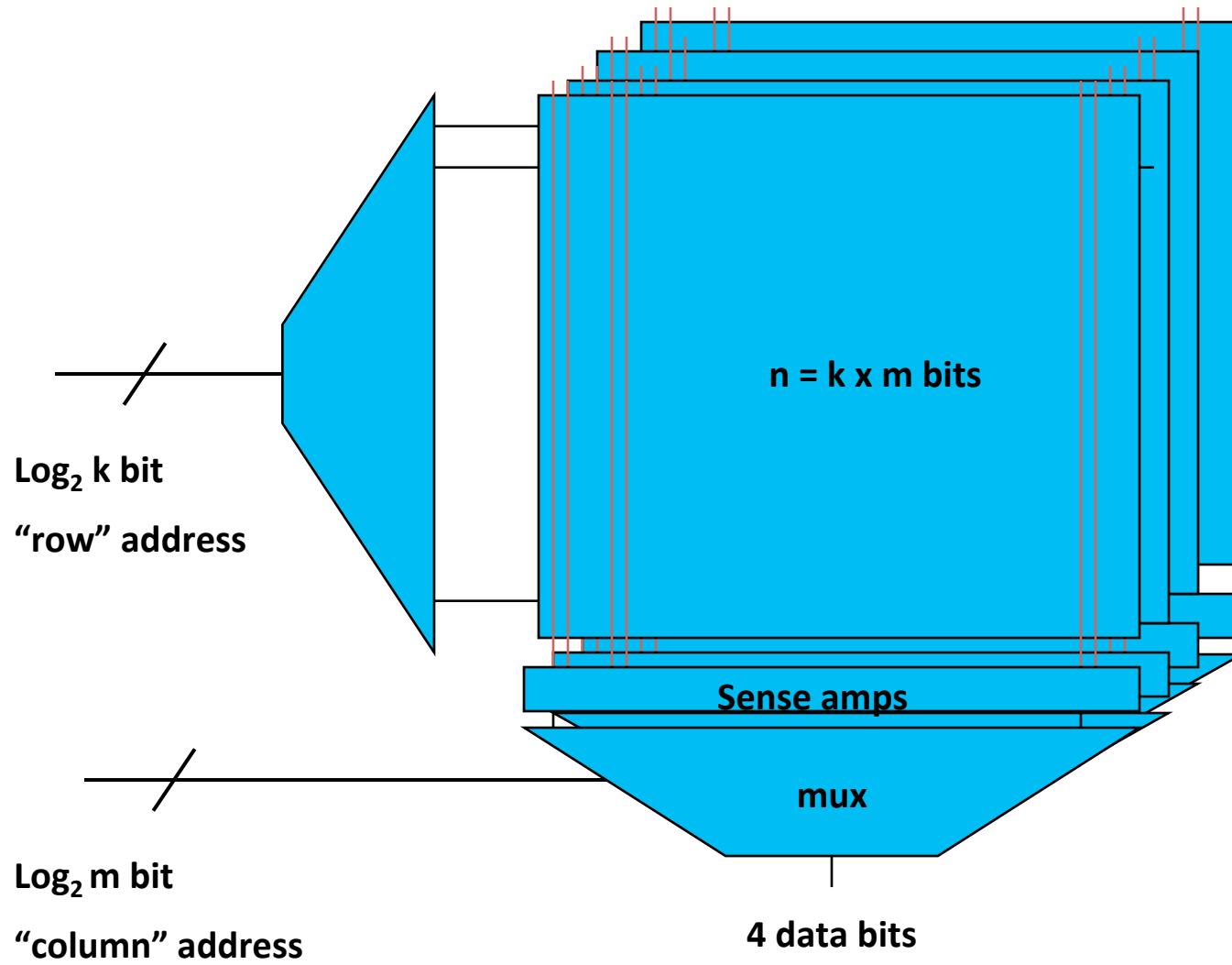


- Decoder larger and slower
- Bit lines increase in length
  - Large distributed RC load
  - Larger, slower transistors
- Remember
  - Treat output as differential signal
  - Pre-charge both bit lines high
  - Memory cell pulls only one low
  - Sense bit value by comparing sense lines
- → Make it shorter and wider!

# Inside a Tall Thin RAM is...

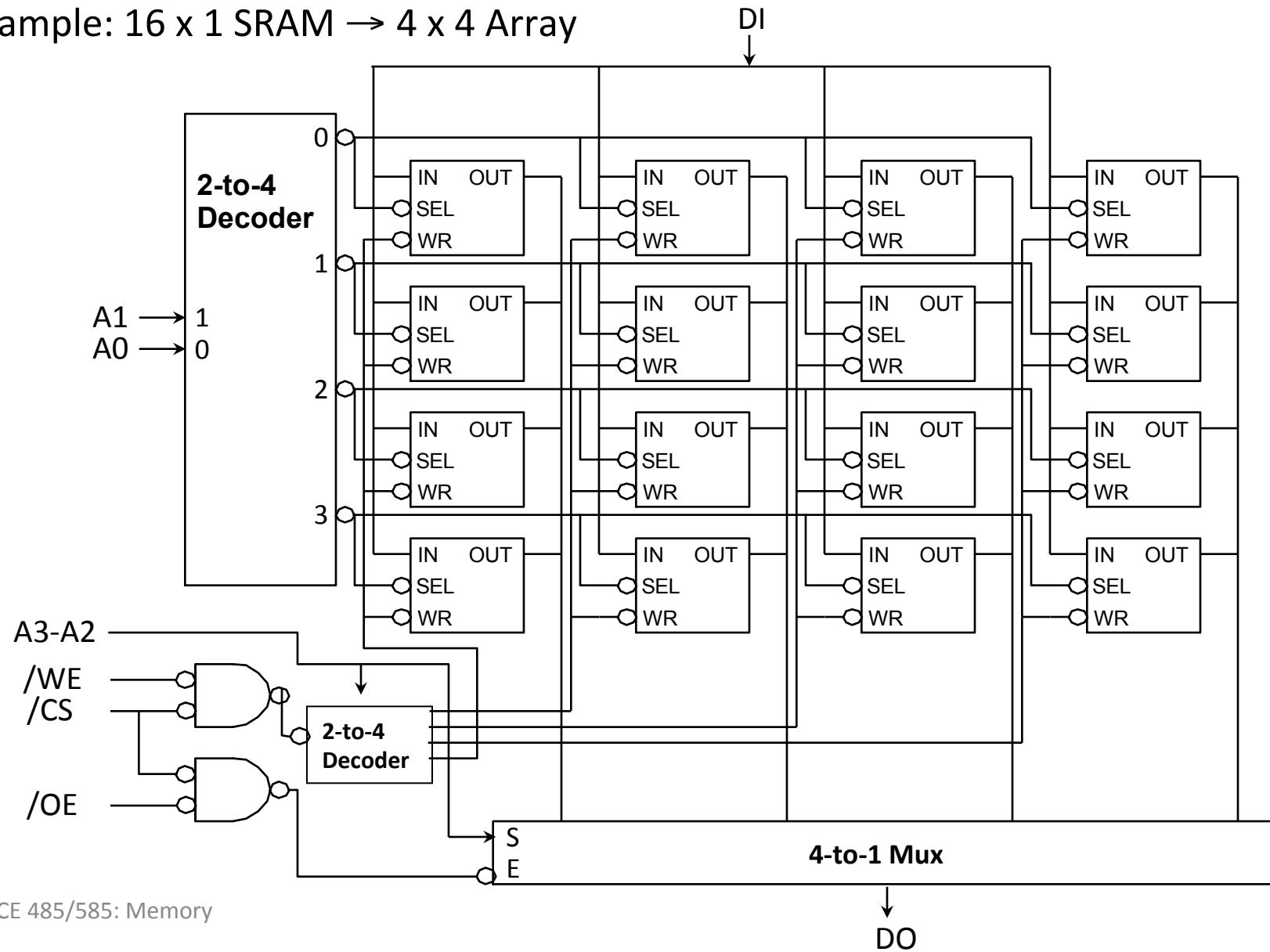


# Replicate for Desired Width



# Physical SRAM Array Should Be Square

Example: 16 x 1 SRAM  $\rightarrow$  4 x 4 Array

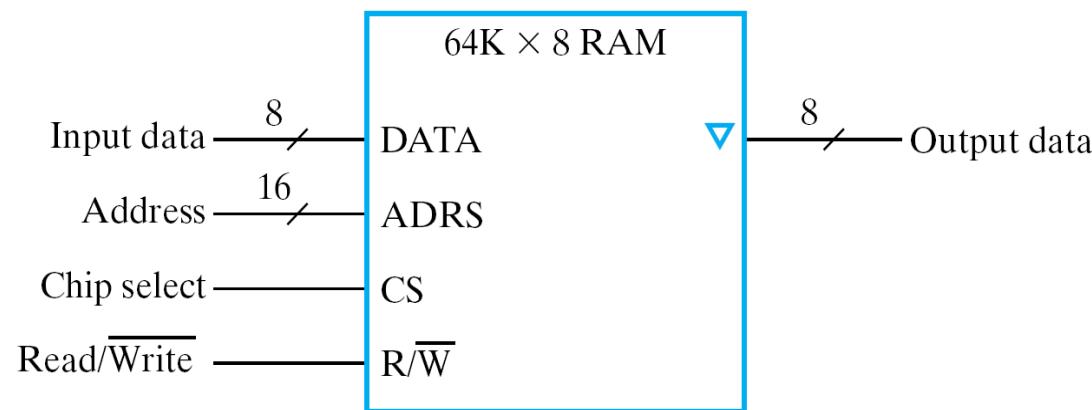


# Evolutionary Modifications

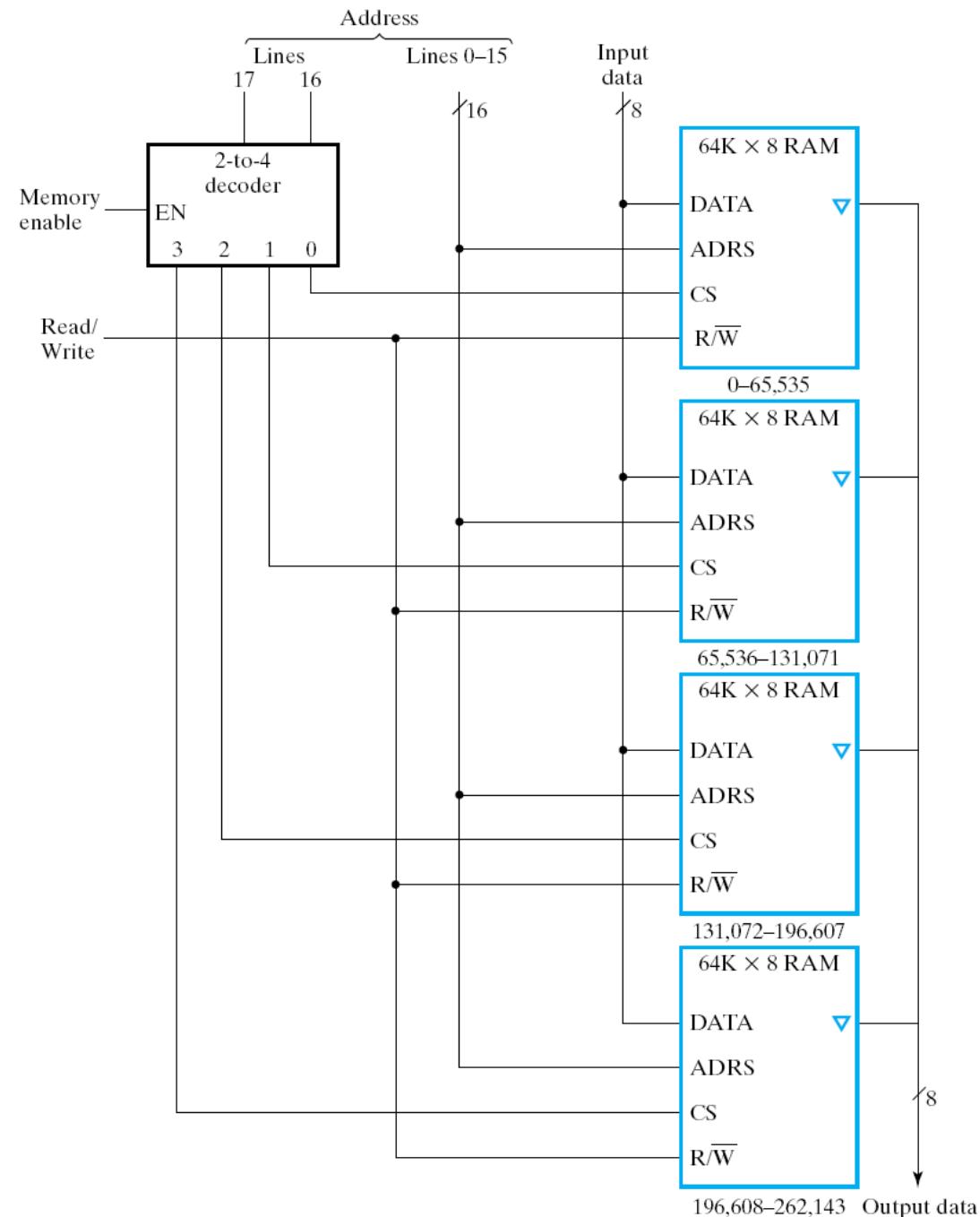
- Add Output Enable
  - Can turn off drivers and immediately start driving data for write without bus contention
- Add latches to input data
  - Chip Select functions this way
  - Only need to hold data until latched
- Add latches to output data
  - Outputs available to be read while next access begun
- Provide synchronous interface

# Memory Organization

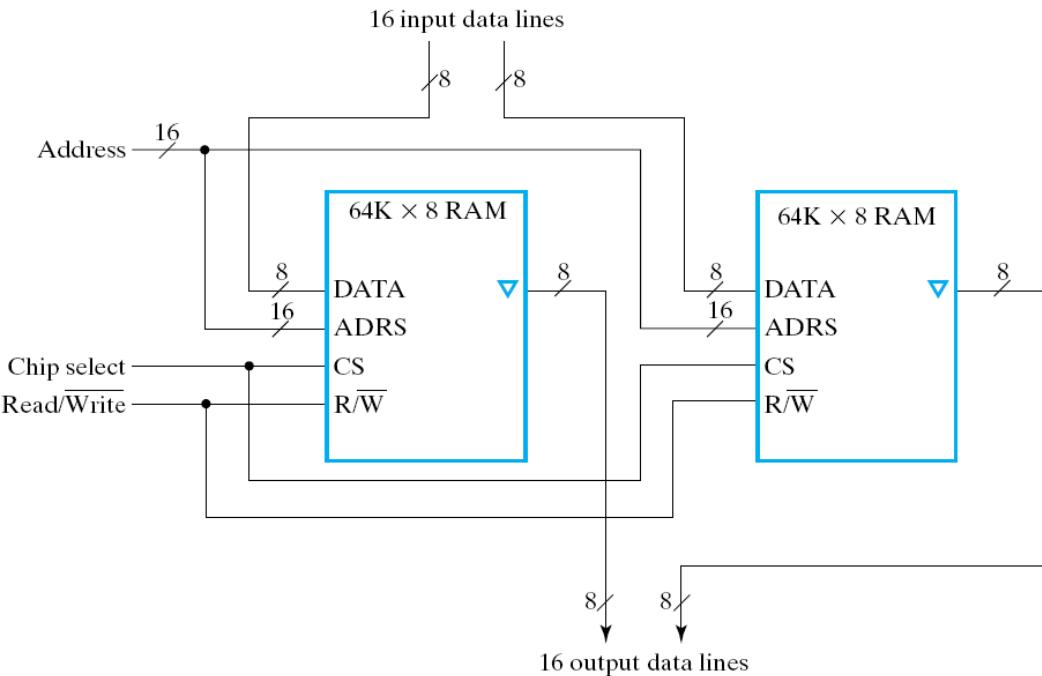
How do we build memory subsystems out of memory devices?



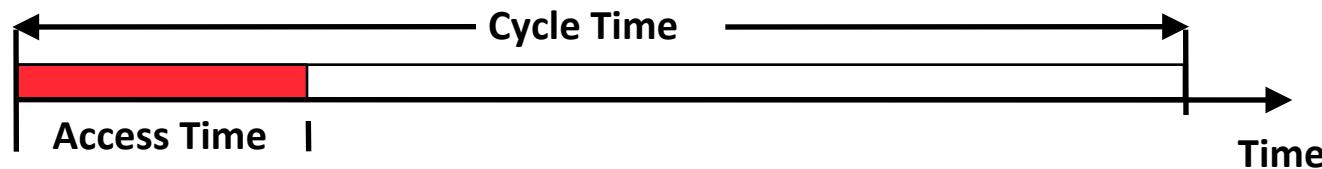
256K x 8 Memory System  
 4x 64K x 8 RAM chips  
 $256K \rightarrow 18$  address lines



## 64K x 16 Memory System 2x 64K x 8 RAM chips



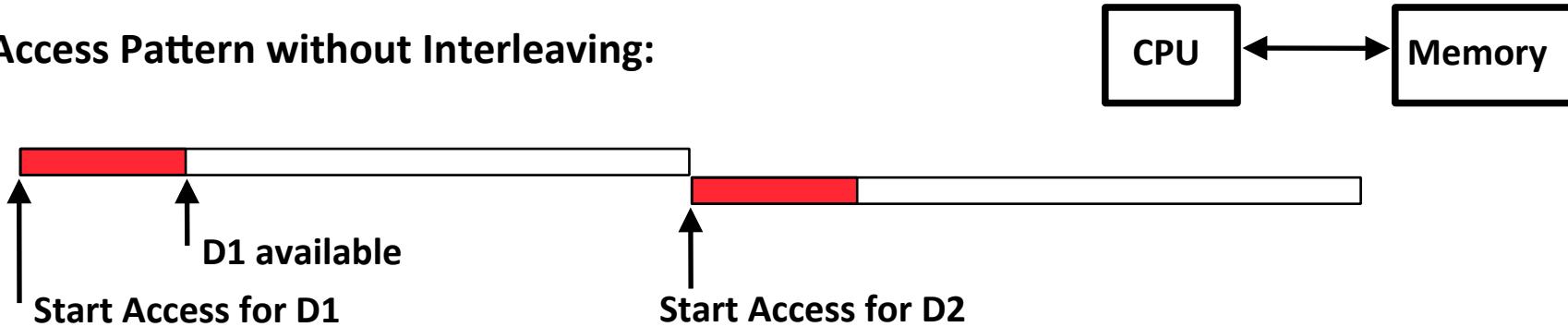
# Improving Memory System Performance



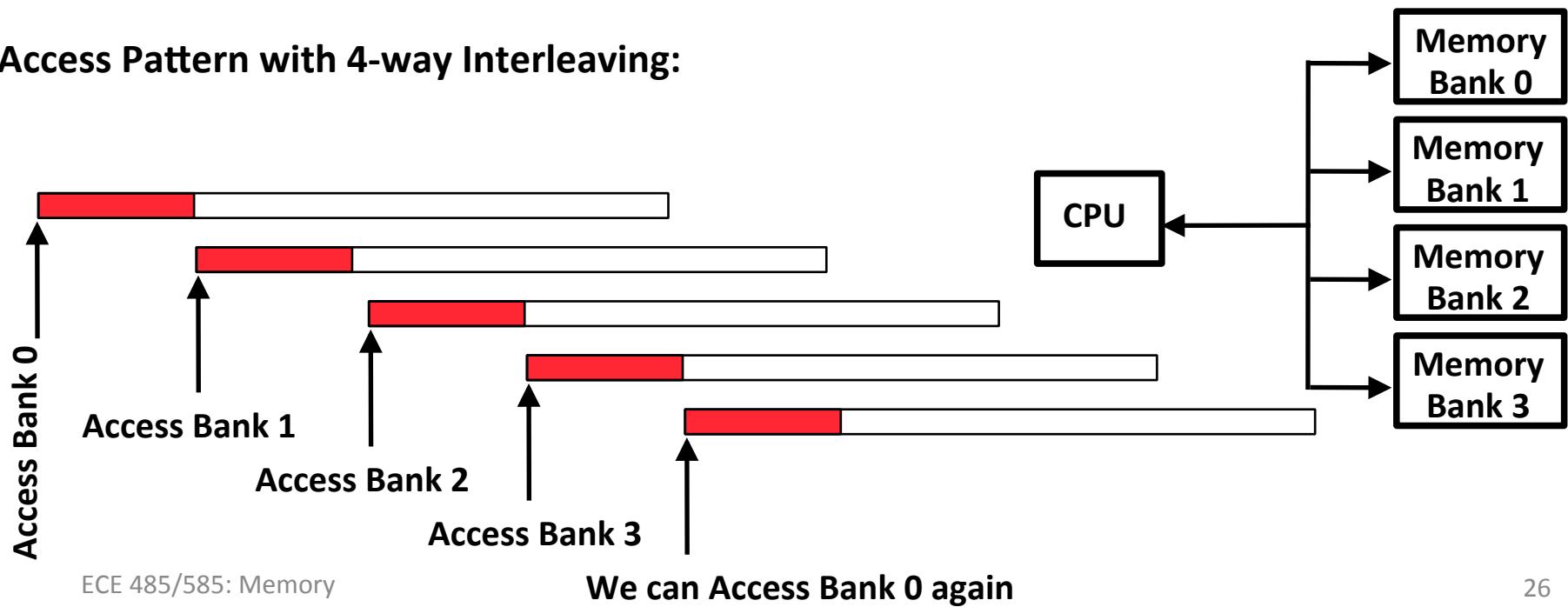
- DRAM (Read/Write) Cycle Time >> DRAM (Read/Write) Access Time
  - 2:1 Why?
- DRAM (Read/Write) Cycle Time
  - How frequently can you initiate an access?
  - Analogy: A little kid can only ask his father for money on Saturday
- DRAM (Read/Write) Access Time
  - How quickly will you get what you want once you initiate an access?
  - Analogy: As soon as he asks, his father will give him the money
- DRAM Bandwidth Limitation analogy
  - What happens if he runs out of money on Wednesday?
  - Ask Mom!

# Increasing Bandwidth: Interleaving

Access Pattern without Interleaving:



Access Pattern with 4-way Interleaving:



# Memory Interleaving

read 00000

read 00001

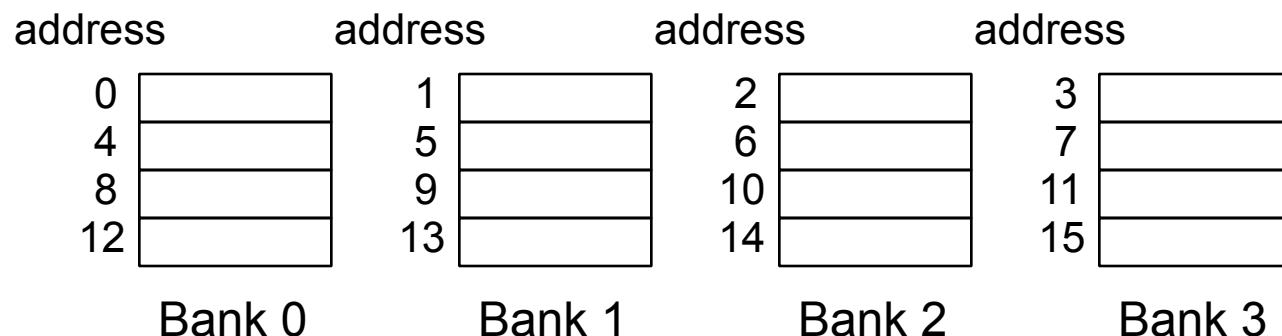
read 00002

read 00003

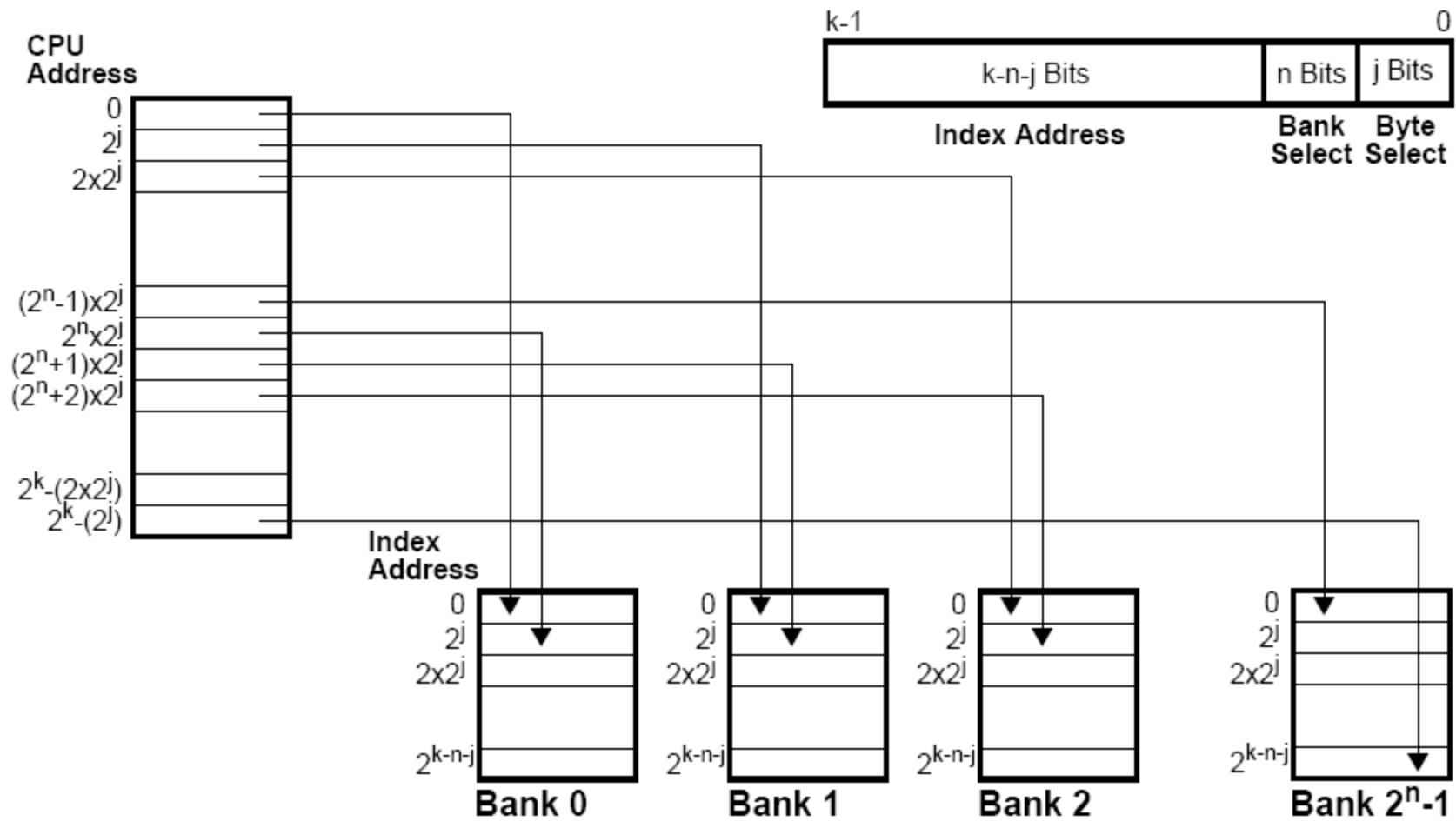
read 00004

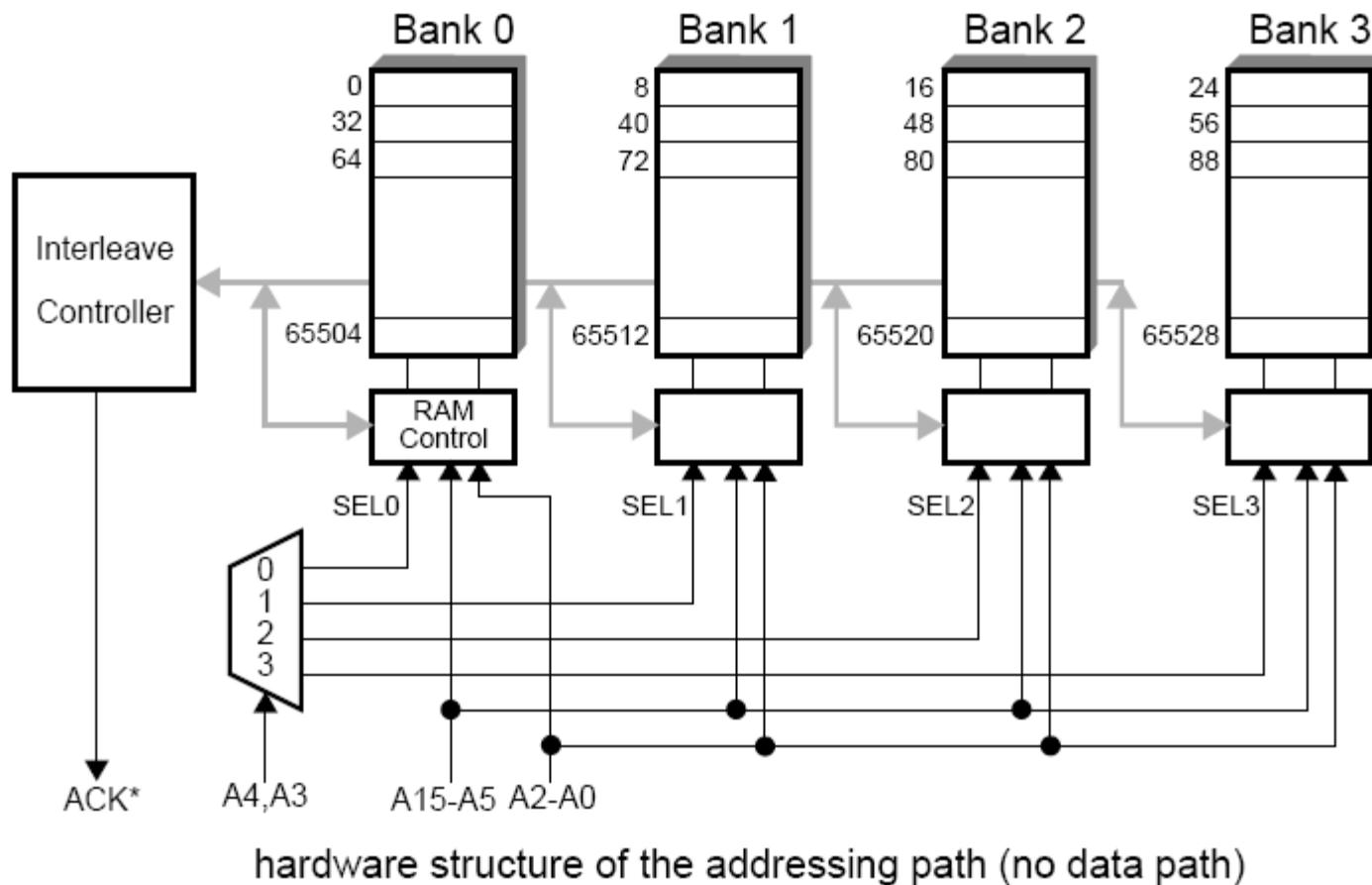
```
for (i = 0; i < 16; i++)  
    A[i] = A[i] * c + d;
```

(assume A[0] at address 0)

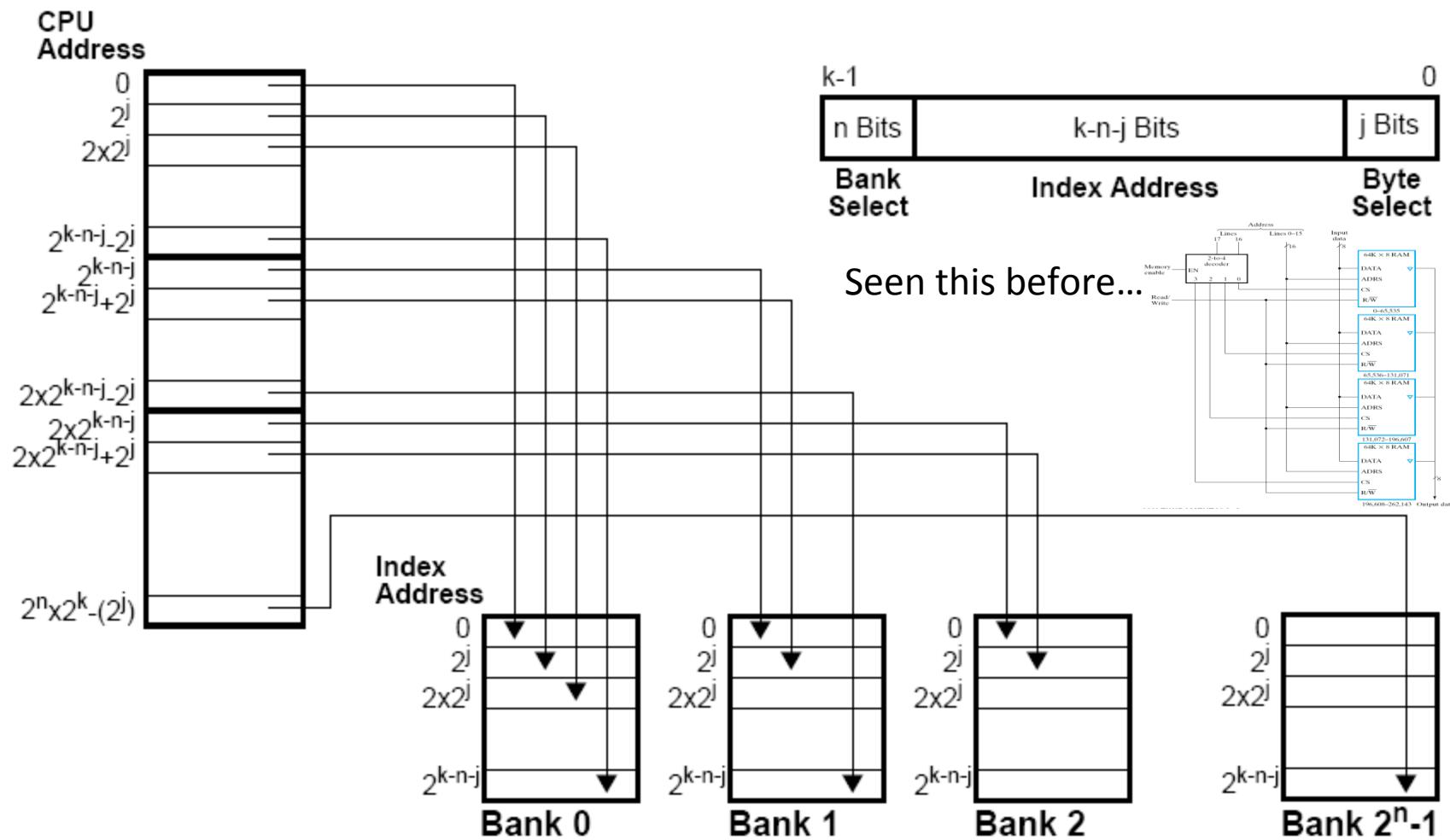


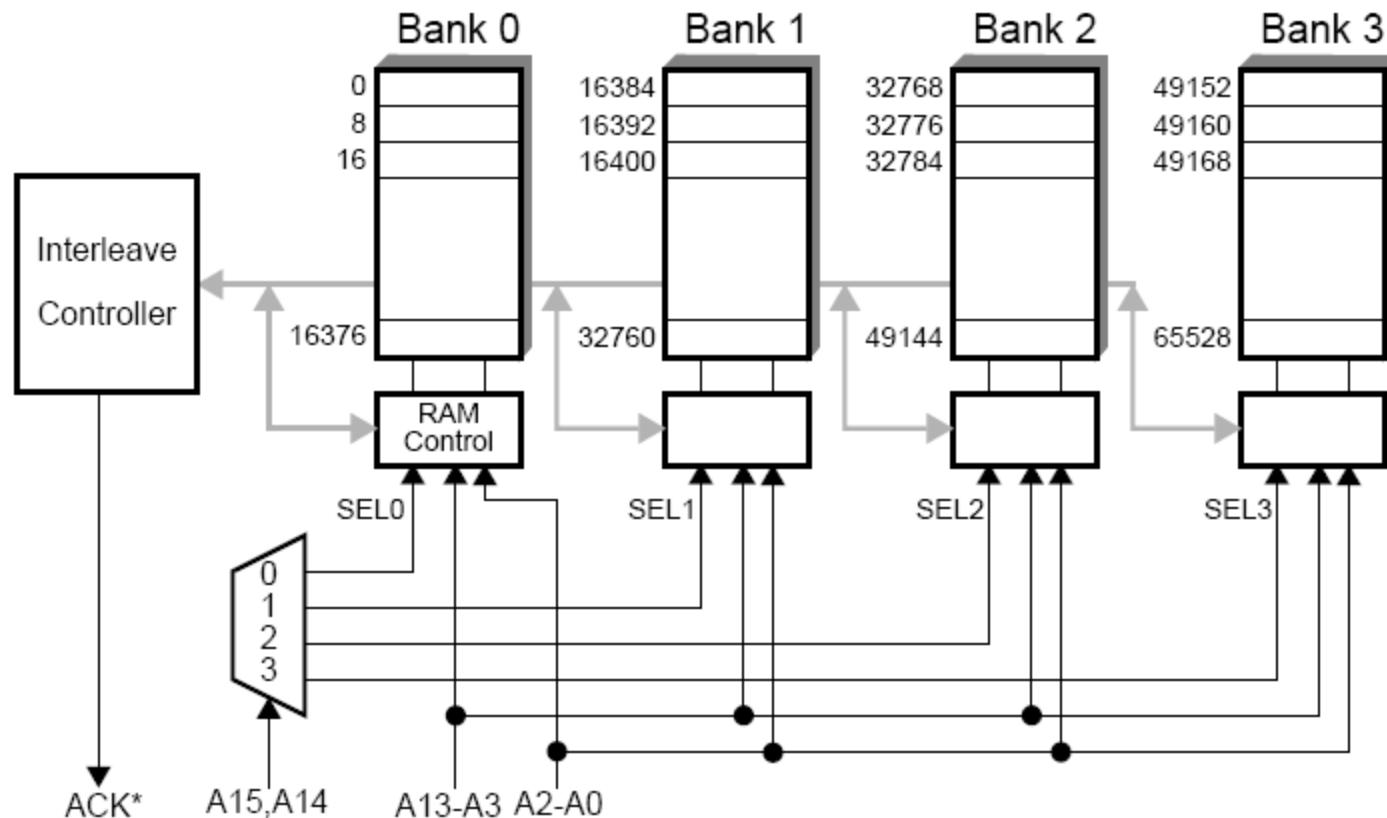
# Low Order Memory Interleaving





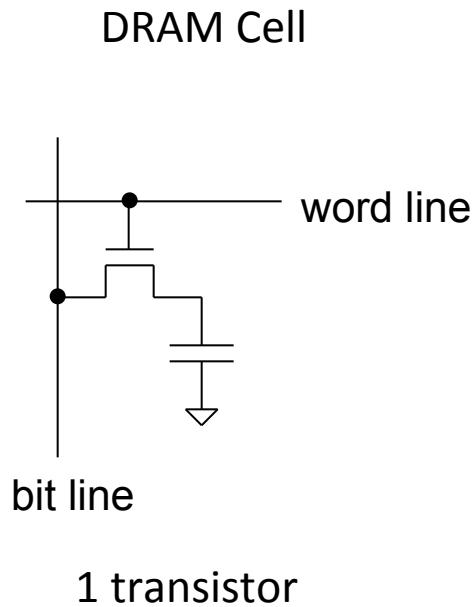
# High Order Memory Interleaving





hardware structure of the addressing path (no data path)

# DRAM Technology



Charge stored on tiny capacitor

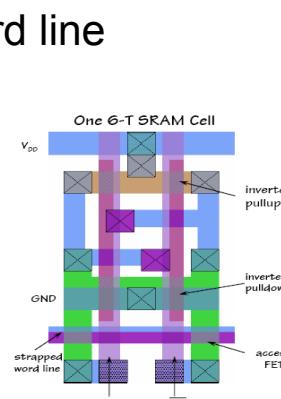
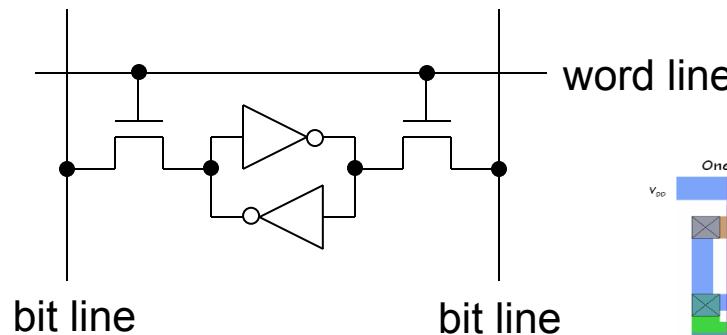
Read is destructive → read must restore value

Charge leaks over time → refresh

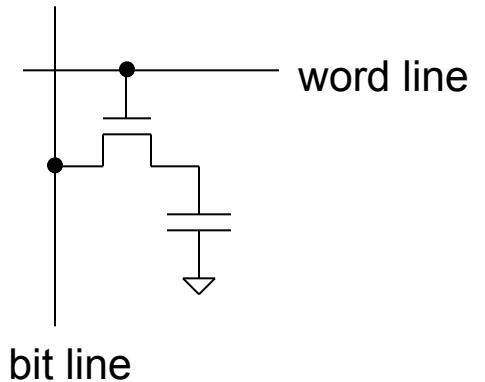
- Write
  - Drive bit line
  - Select desired word (“row”)
- Read
  - Pre-charge bit line
  - Select desired word (“row”)
  - Sense charge
  - Write value back (restore)
- Refresh!
  - Periodically read each cell
    - (forcing write-back)

# Volatile Memory Comparison

SRAM Cell



DRAM Cell

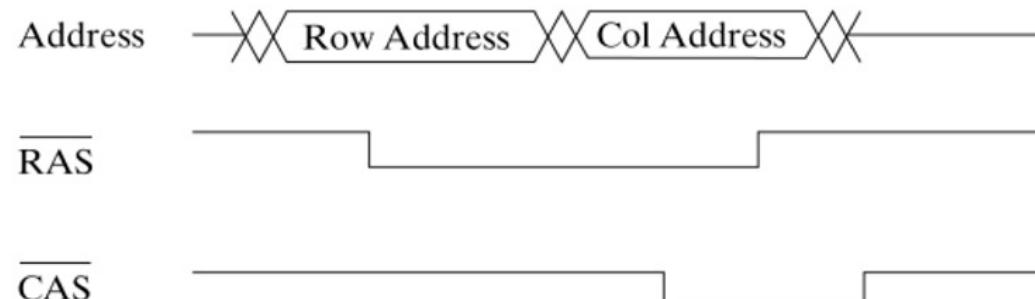
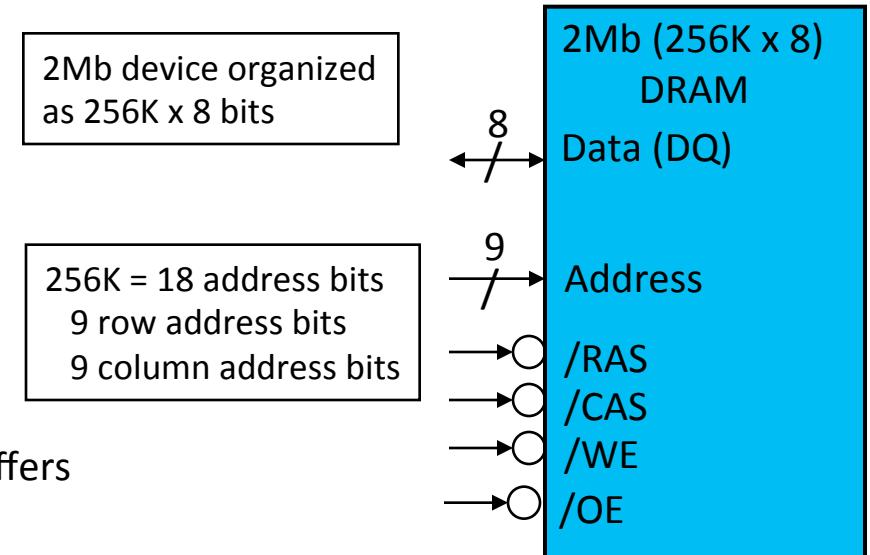


- Larger cell
  - lower density, higher cost/bit
- No dissipation
- Read non-destructive
- No refresh required
- Simple read  $\Rightarrow$  faster access
- Standard IC process  $\Rightarrow$  natural for integration with logic

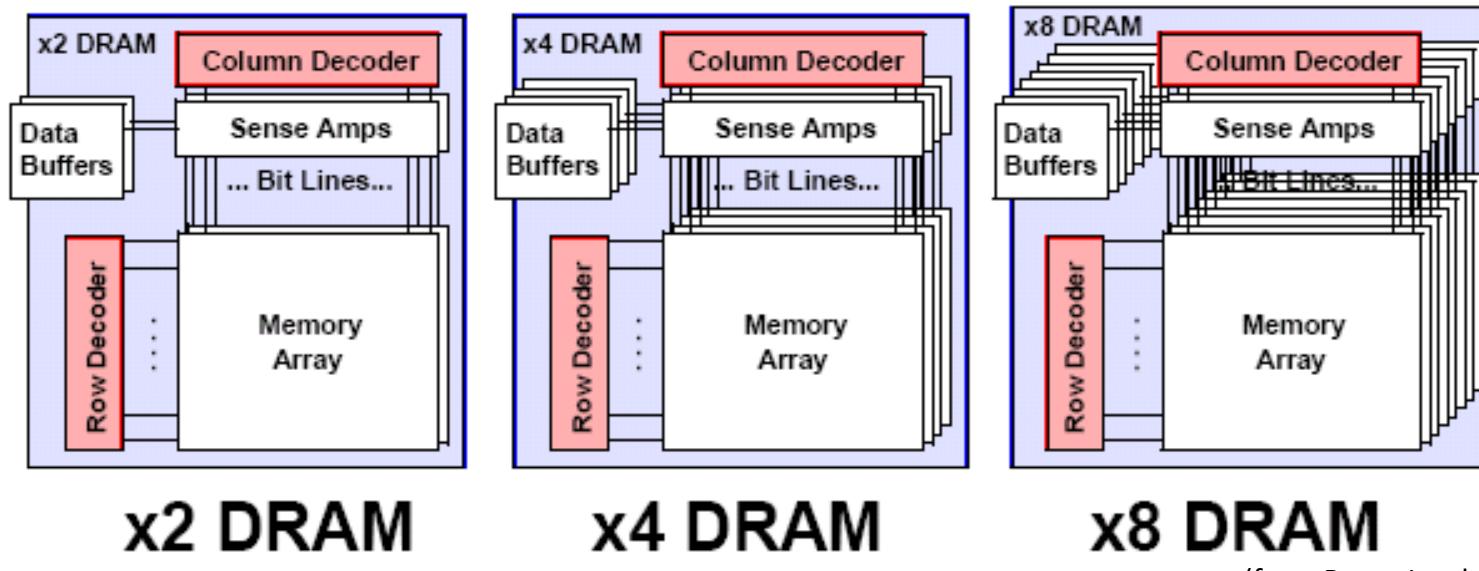
- Smaller cell
  - higher density, lower cost/bit
- Needs periodic refresh
- Refresh after read
- Complex read  $\Rightarrow$  longer access time
- Special IC process  $\Rightarrow$  difficult to integrate with logic circuits
- Density impacts addressing
  - (multiplex address lines)

# DRAM Device Pin Outs

- Cost Rules!
  - Fewer pins, smaller package, less \$
- So Multiplex
  - Data (In/Out)
    - /WE asserted (low) for write
    - /OE asserted (low) to enable output buffers
  - Address (Row/Column)
    - /RAS (Row Address Strobe) asserted after row placed on address pins
    - /CAS (Column Address Strobe) asserted after column placed on address pins



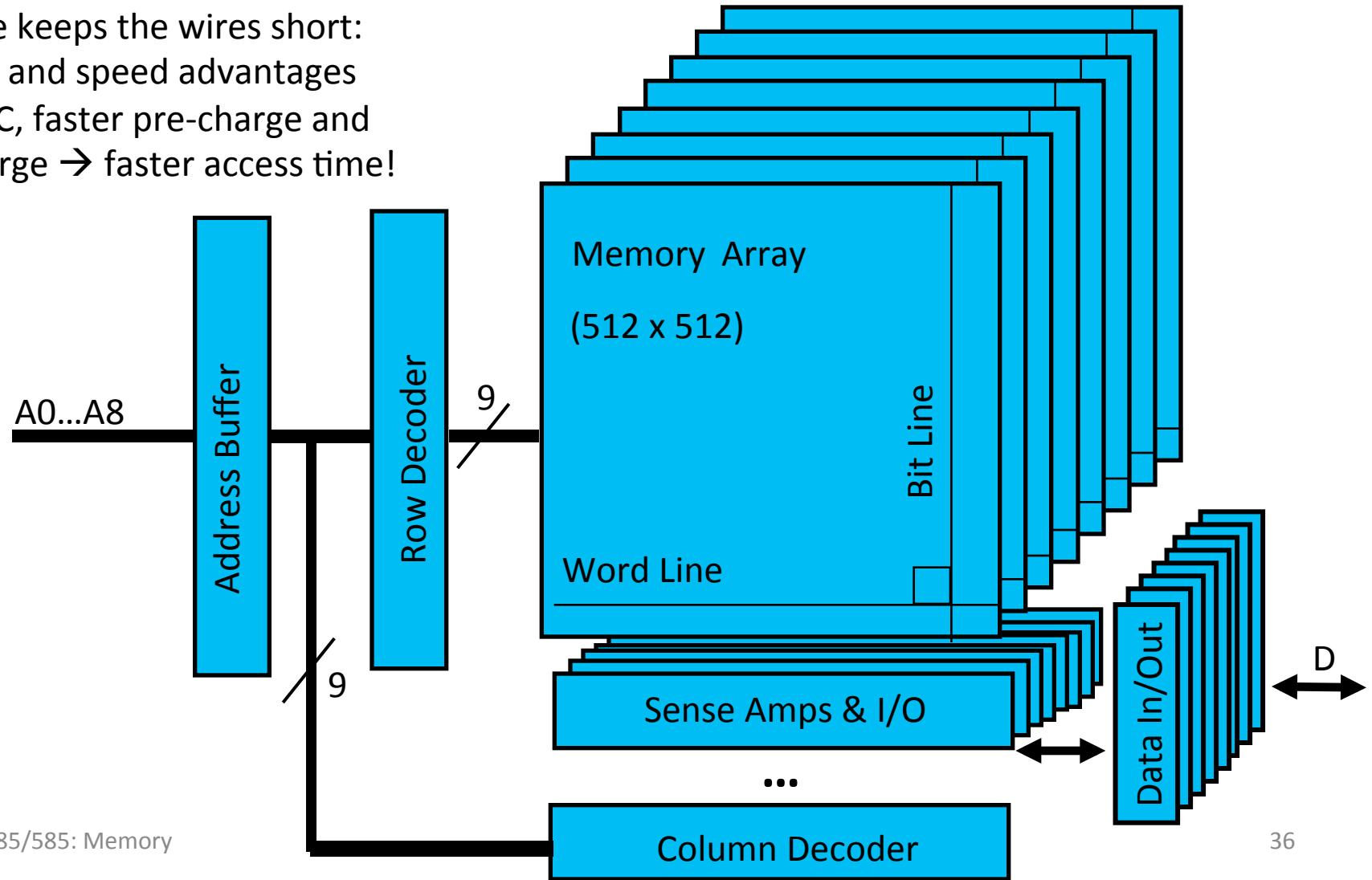
How do we keep square row/column organization but get devices of more than x1 bit?



# Internal DRAM Organization

## 2Mb as 256K x 8

Square keeps the wires short:  
Power and speed advantages  
Less RC, faster pre-charge and  
discharge → faster access time!

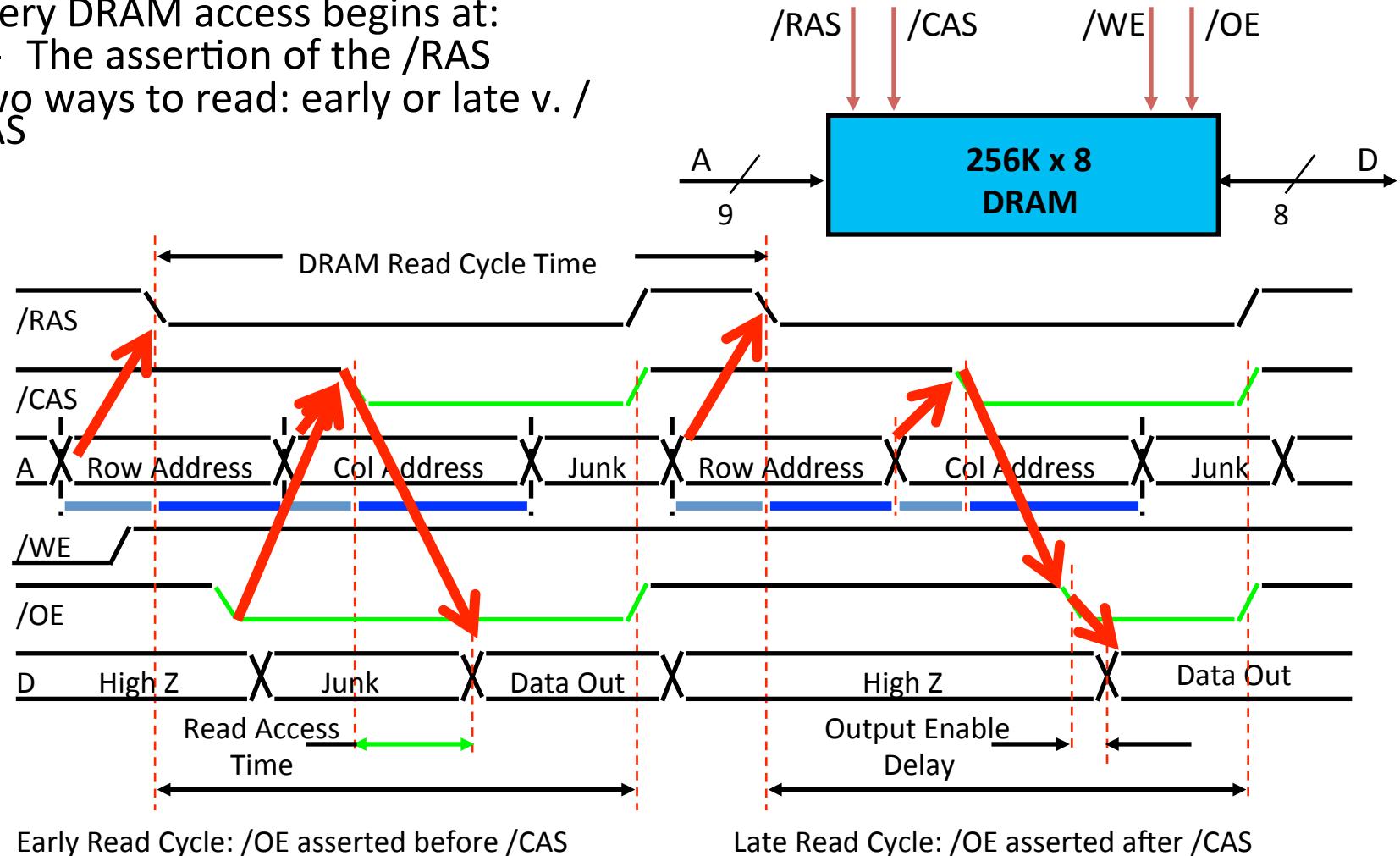


# DRAM Timing

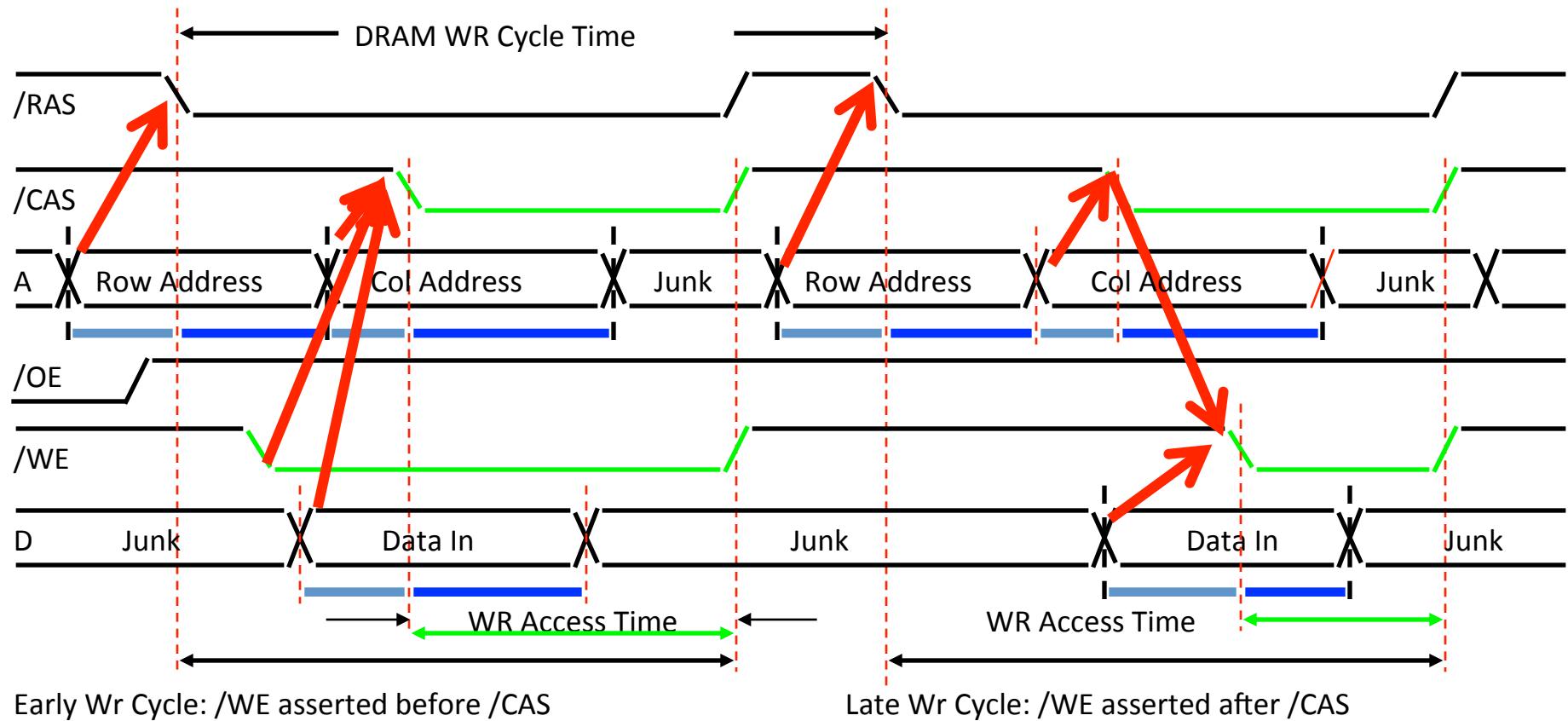
- Read cycle - RAS + CAS
  - (RAS asserted) Entire row is latched in register
  - (CAS asserted) Data in register is multiplexed to output
  - (RAS de-asserted) Data in register is rewritten
  - (CAS de-asserted) Output is released
- Write cycle - RAS + WE + CAS
  - (RAS asserted) Entire row is latched in register
  - (WE asserted) Data is stable
  - (CAS asserted) Write Data to register
  - (WE de-asserted) Data is no longer stable
  - (RAS de-asserted) Data in register is rewritten
  - (CAS de-asserted) Operation complete
- Refresh cycle - RAS Only
  - (RAS asserted) Entire row is latched in register
  - (RAS de-asserted) The data in the register is rewritten

# DRAM Read Timing

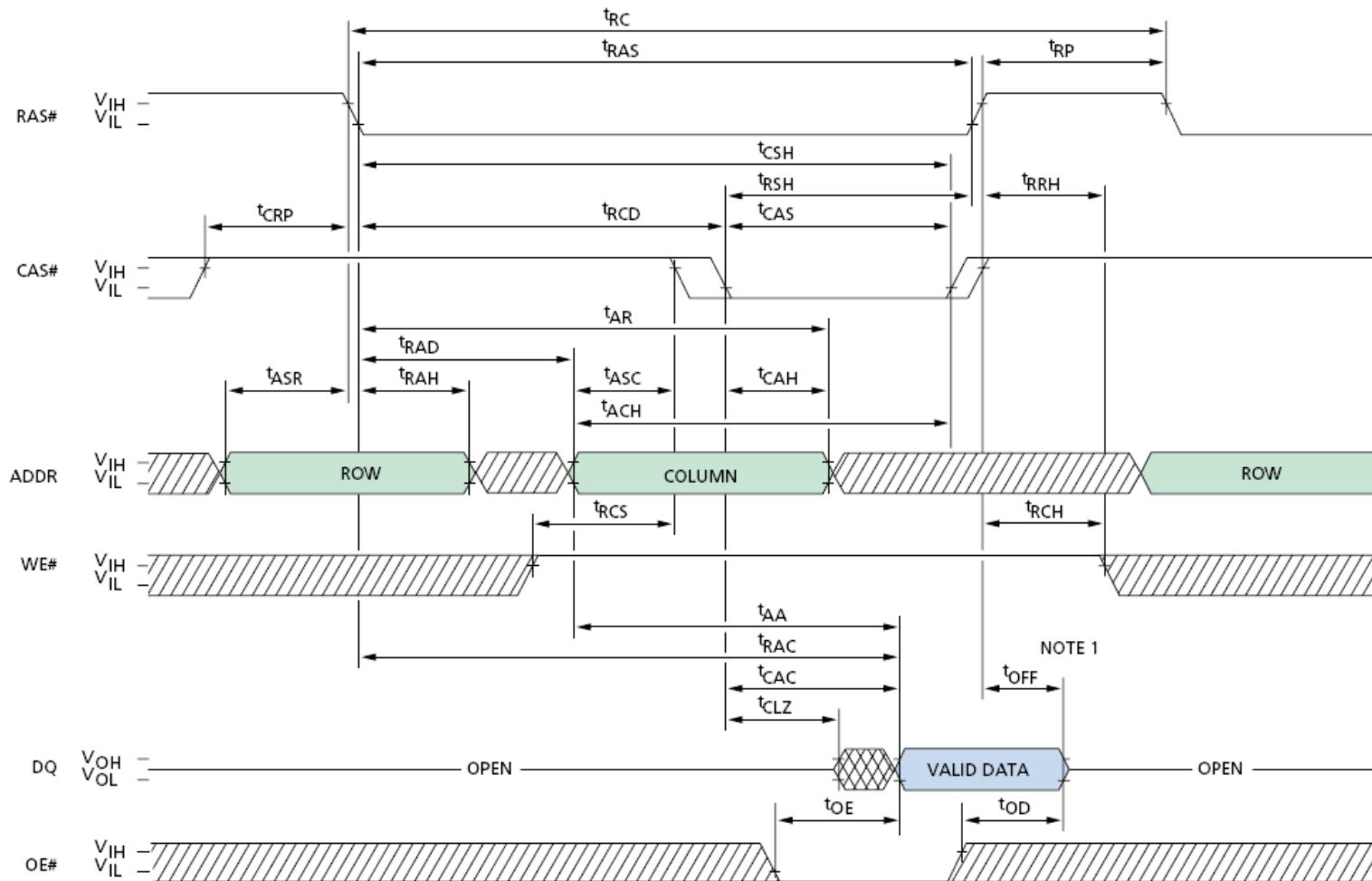
- Every DRAM access begins at:
  - The assertion of the /RAS
- Two ways to read: early or late v. / CAS



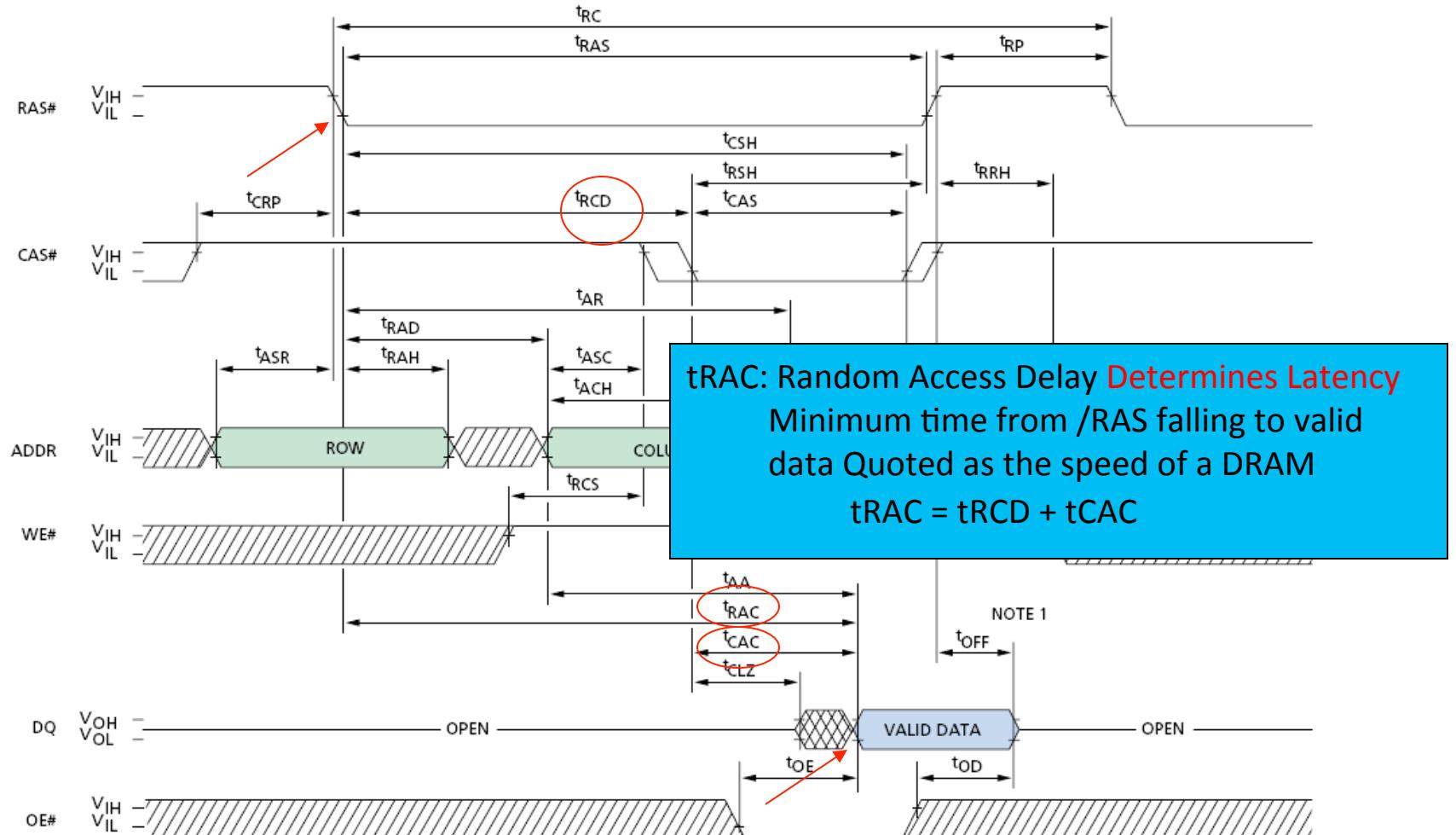
# DRAM Write Timing



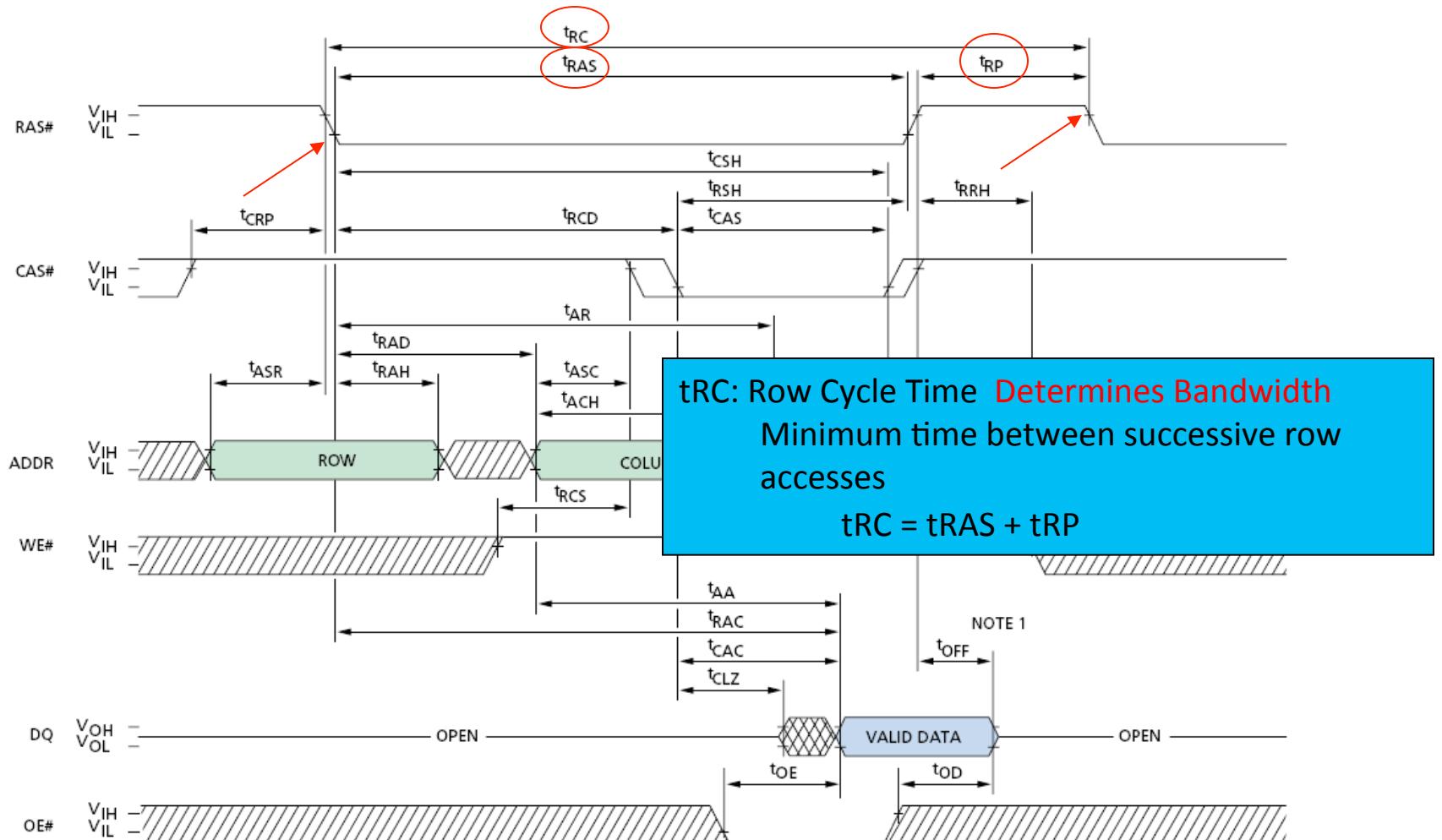
# Actual DRAM Read Cycle



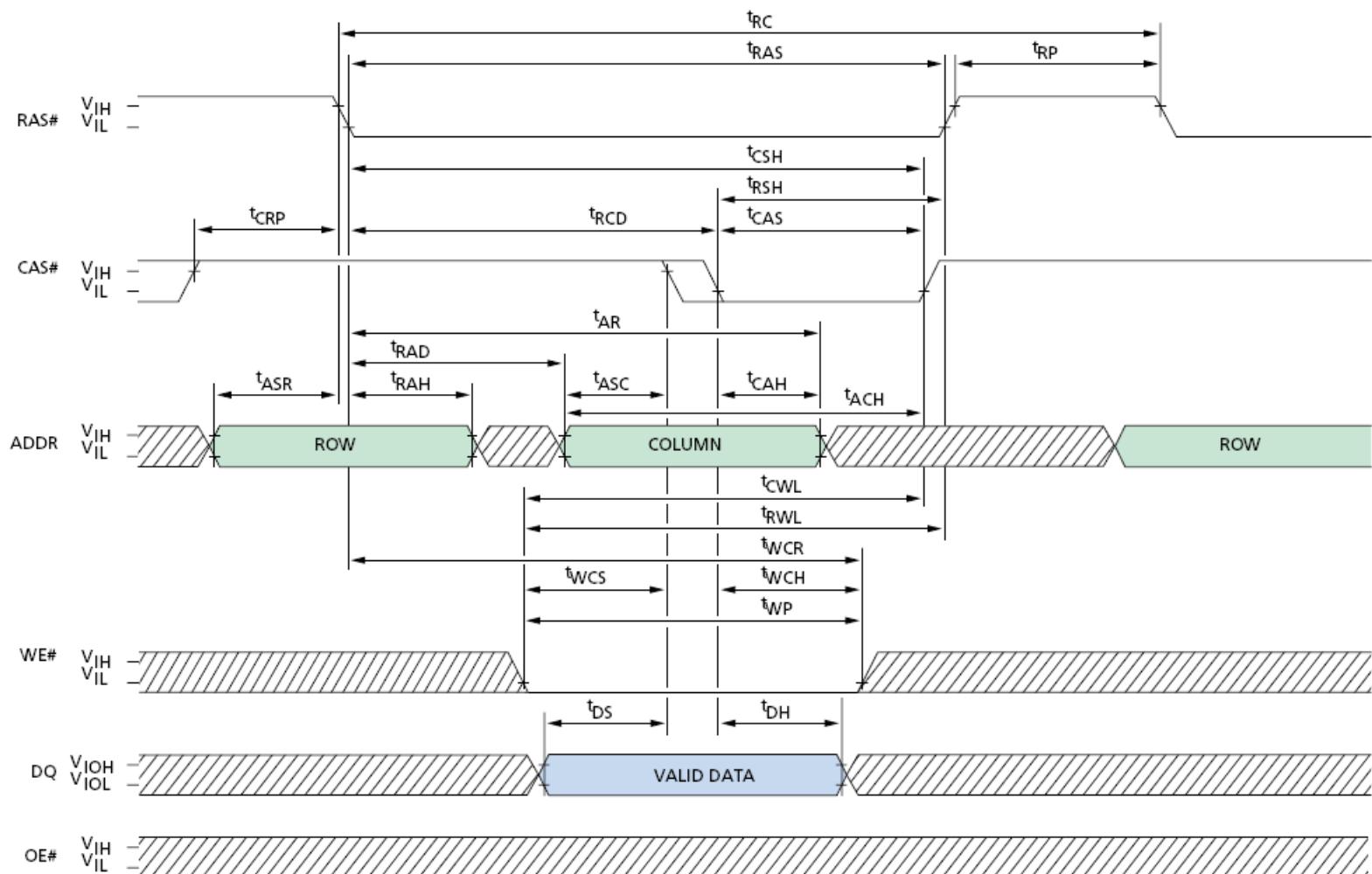
# Actual DRAM Read Cycle



# Actual DRAM Read Cycle



# Actual DRAM Write Cycle



# Key DRAM Timing Parameters

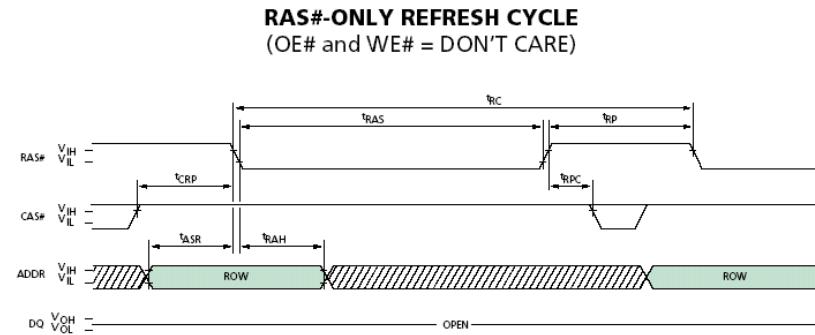
- $t_{RAC}$ : Random Access Delay **Determines Latency**
  - Minimum time from /RAS falling to valid data output
  - Quoted as the speed of a DRAM
    - $t_{RAC} = t_{RCD} + t_{CAC}$
- $t_{RCD}$ : Row Command Delay (RAS/CAS Delay)
  - Minimum time between a row command and a column command
- $t_{CAC}$ : Column Access Time
  - Delay from falling /CAS to valid data out
- $t_{RC}$ : Row Cycle Time **Determines Bandwidth**
  - Minimum time between successive row accesses
    - $t_{RC} = t_{RAS} + t_{RP}$
- $t_{RAS}$ : Row Address Strobe
  - Minimum time /RAS must be maintained
- $t_{RP}$ : Row Pre-Charge Delay
  - Minimum time to pre-charge so another /RAS can begin

# Refresh



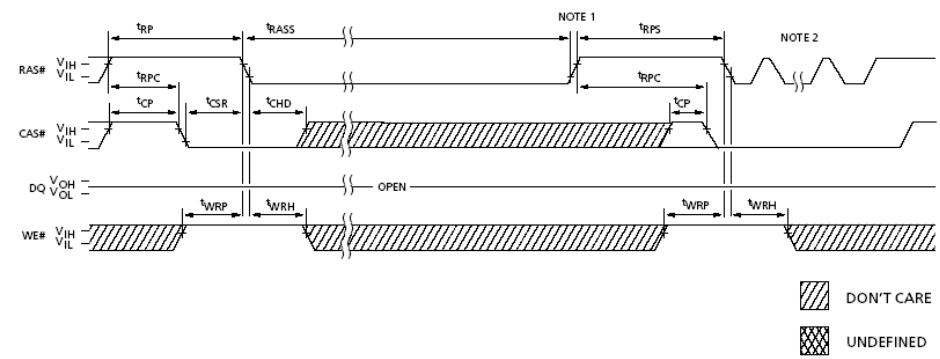
16 MEG x 4  
FPM DRAM

- Depends upon device
  - Refresh period – time by which all rows must be refreshed
    - Typical – 64ms
  - Refresh interval – time between refresh of each row
    - Typical 15.6us.
  - Interleave with normal operation
    - “distributed” vs “burst” refresh
  - Refresh entire row at a time
  - Use /RAS and /CAS to signal
- RAS-only
  - /RAS cycling, no /CAS cycling
  - External memory controller maintains address of last row refreshed
- CAS-before-RAS (CBR)
  - DRAM maintains refresh address
- Hidden Refresh
  - Following Read or Write
  - /CAS left low, /RAS cycled
  - Output data remains valid
  - Requires time, delaying subsequent read/write

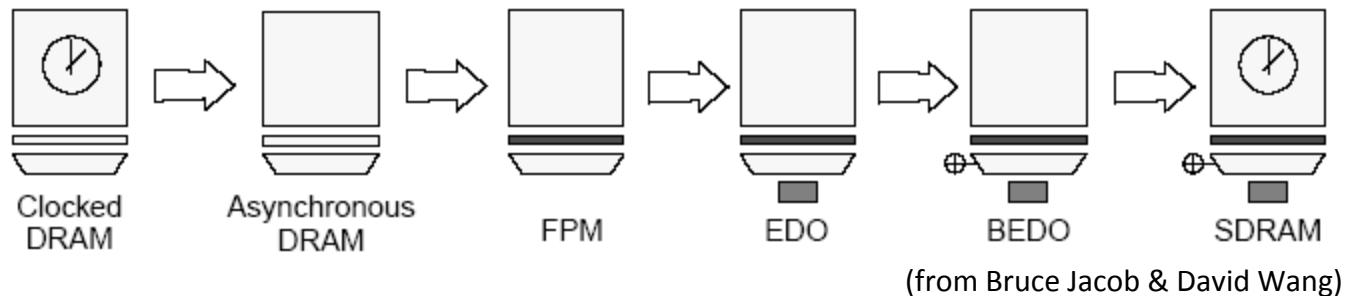


16 MEG x 4  
FPM DRAM

**SELF REFRESH CYCLE**  
(Addresses and OE# = DON'T CARE)



# DRAM Evolution

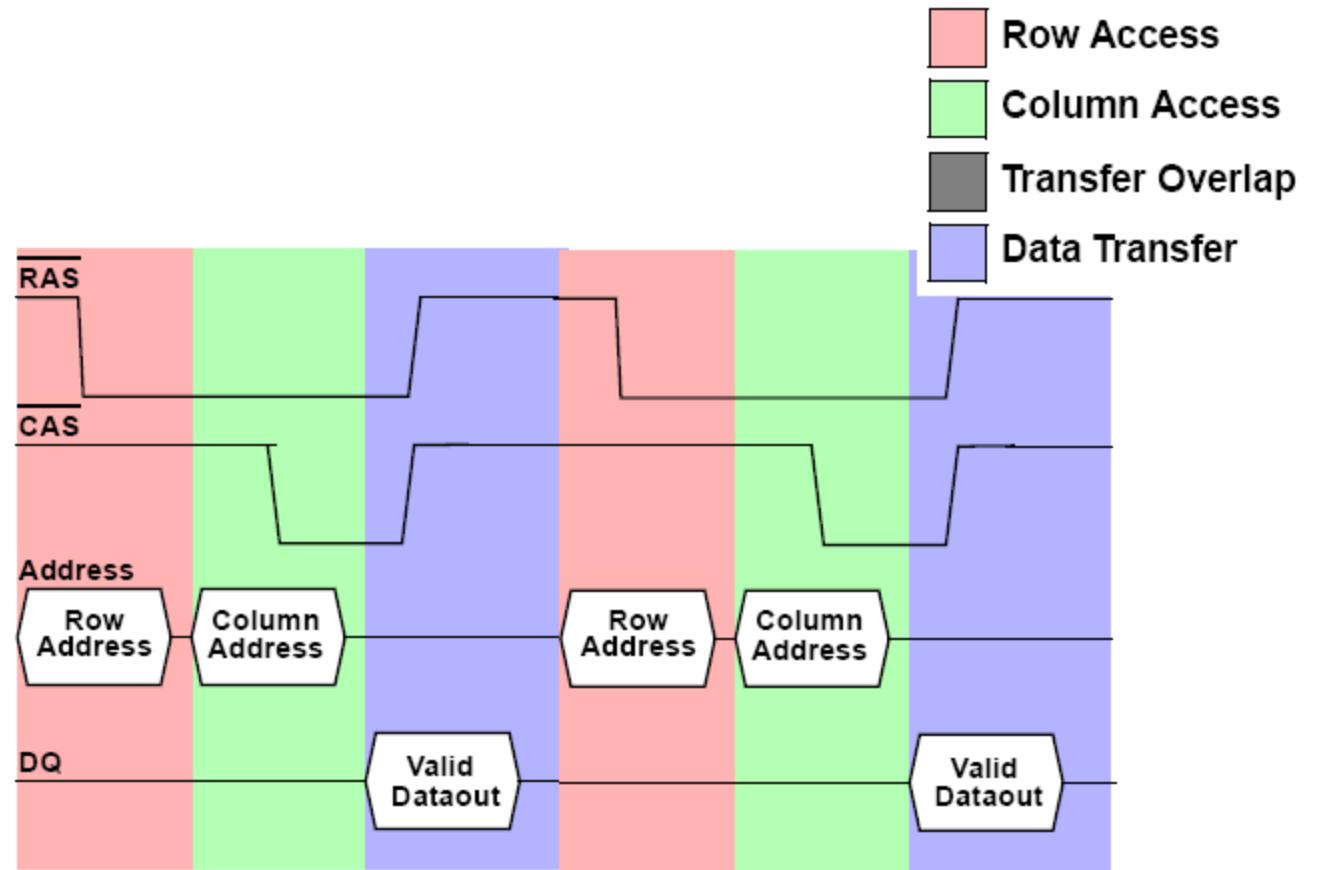
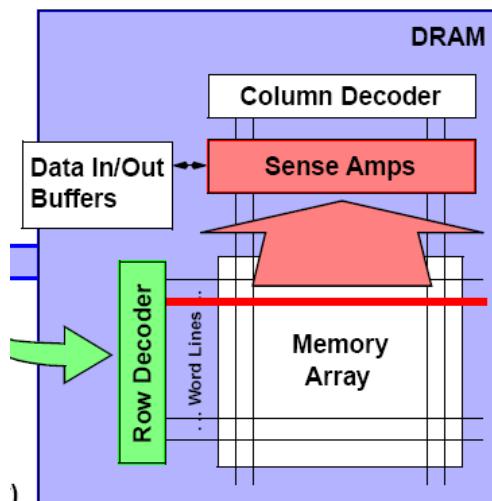


- FPM – Fast Page Mode (< 1995)
- EDO – Extended Data Out (1996 – 1999)
- BEDO – Burst Extended Data Out
- SDRAM – Synchronous DRAM (>1995)
  - SDRAM
  - DDR SDRAM (double data rate)
  - DDR2
  - DDR3
- Other DRAM technologies
  - Rambus DRAM (RDRAM)
  - Concurrent Rambus DRAM
  - Direct Rambus DRAM (DRDRAM)
- Numerous Specialty DRAMs not shown
  - Virtual Channel Memory (VCDRAM)
  - Enhanced SDRAM (ESDRAM)
  - MoSys 1T-SRAM
  - Reduced Latency DRAM (RLDRAM)
  - Fast Cycle DRAM (FCRAM)

By 2002 most PCs using SDRAM and DDR SDRAM

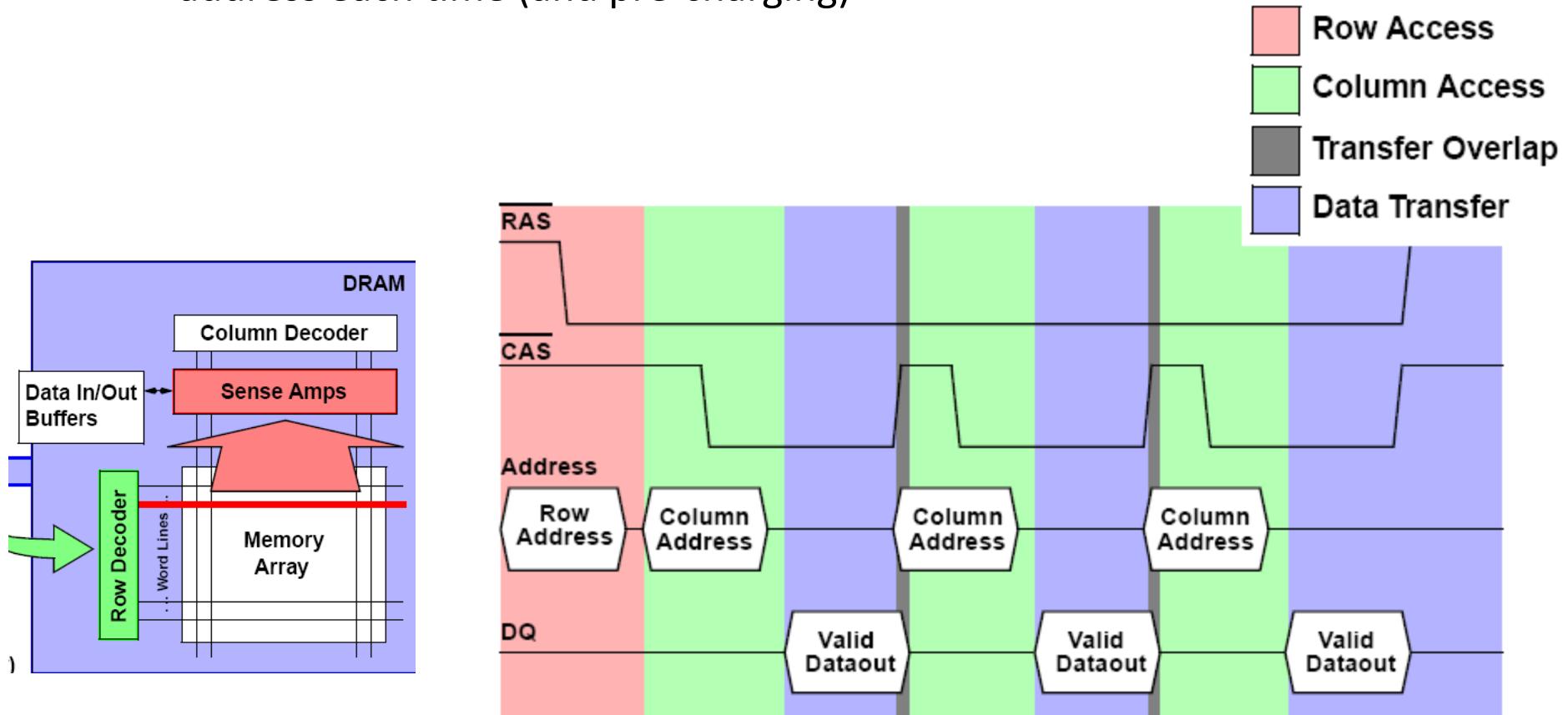
By 2010 PCs shipping with DDR3 in volume

# Conventional DRAM Read Timing



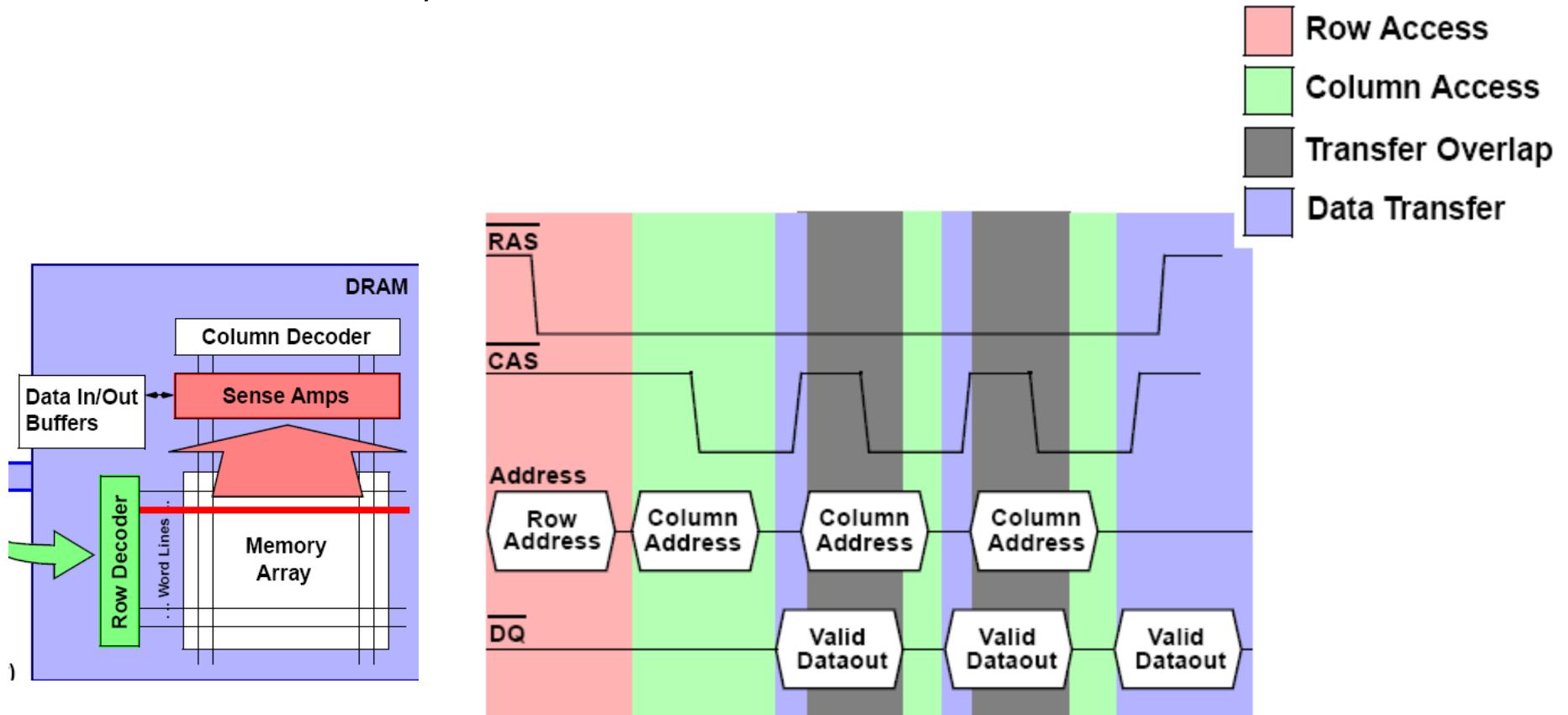
# Fast Page Mode DRAM Read Timing

- Innovation – Hold entire row (page) in sense amplifiers
- Benefit – CPU can access each column in row without providing row address each time (and pre-charging)



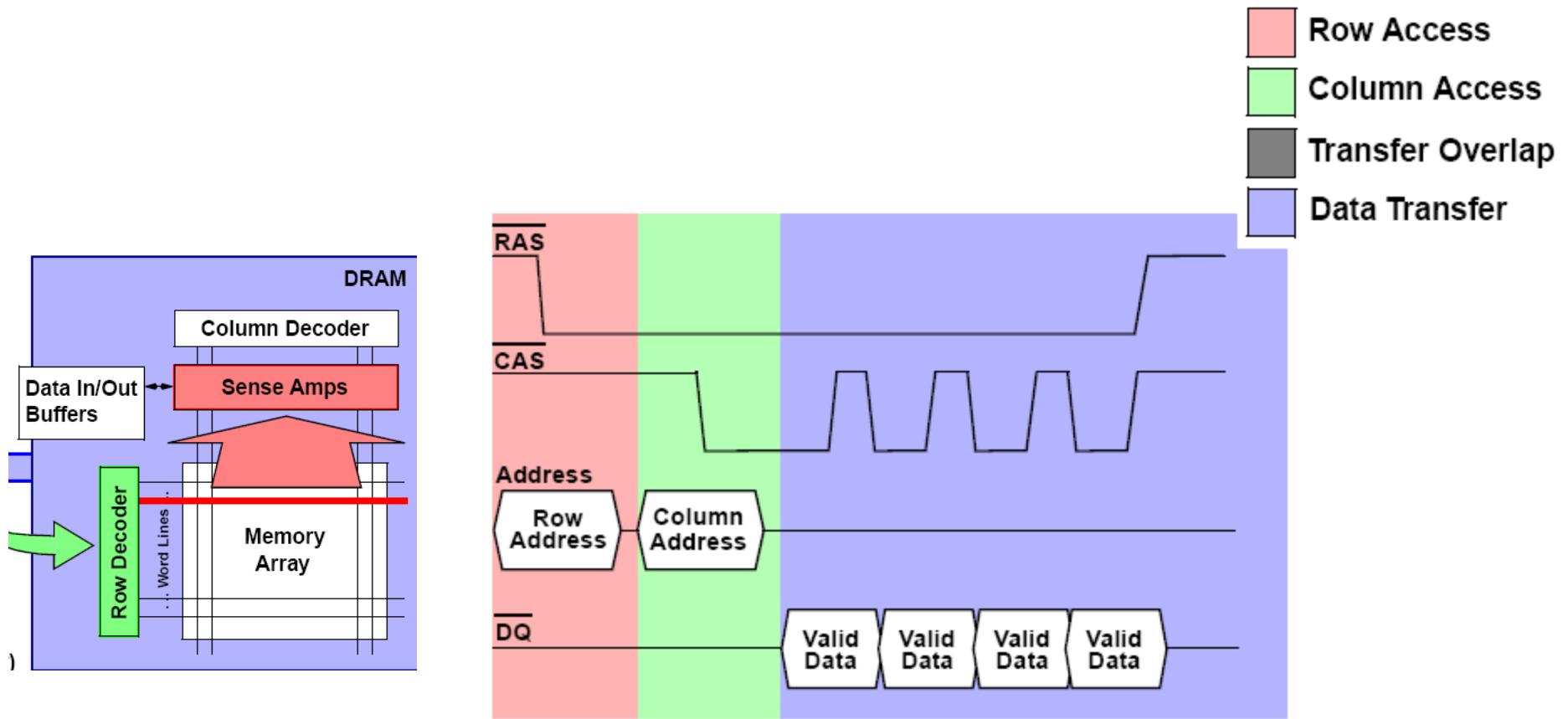
# Extended Data Out DRAM Read Timing

- Innovation – Add latch between sense amplifiers and output pins
- Benefit – Can begin pre-charging sooner (data from prior access remains valid)



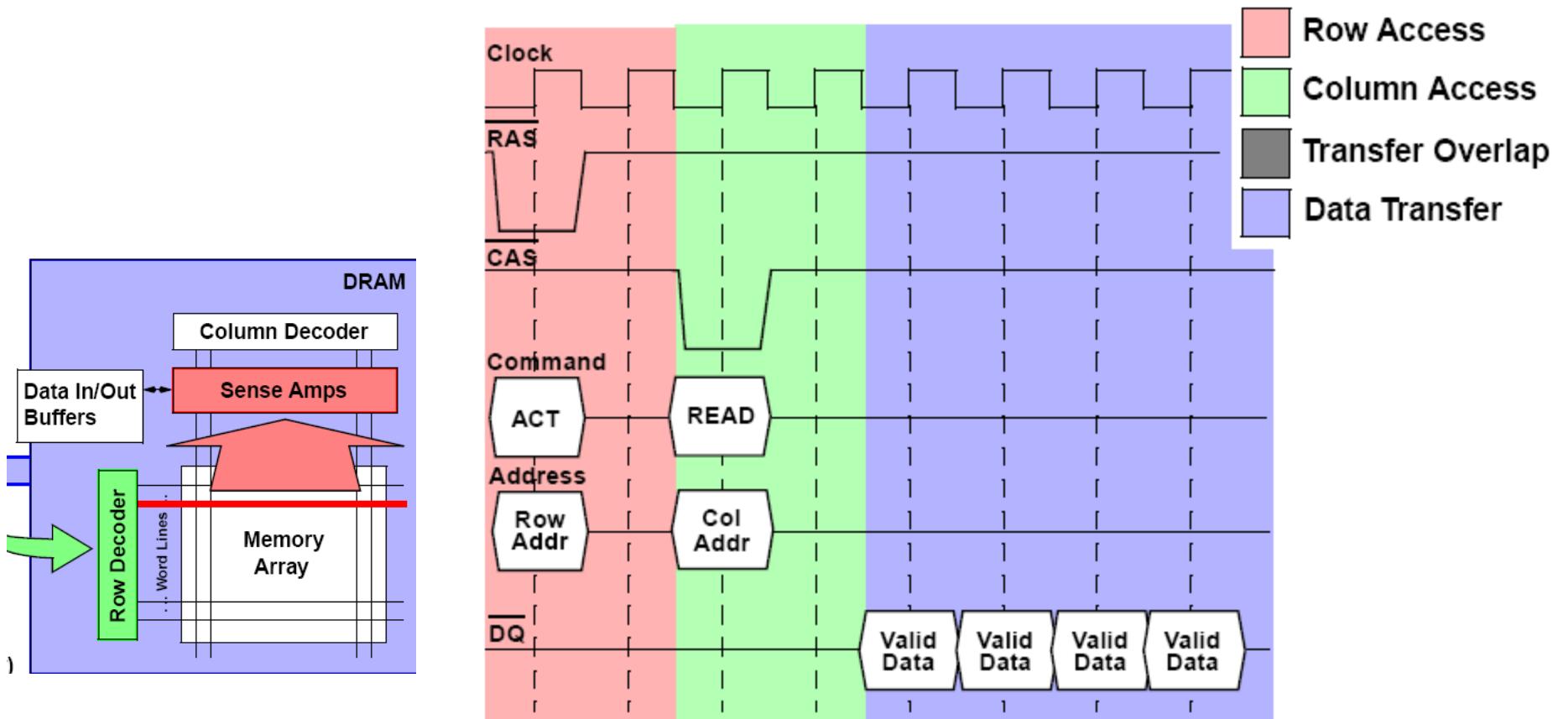
# Burst EDO DRAM Read Timing

- Innovation – DRAM provides column data sequentially
- Benefit – No need to transfer column address



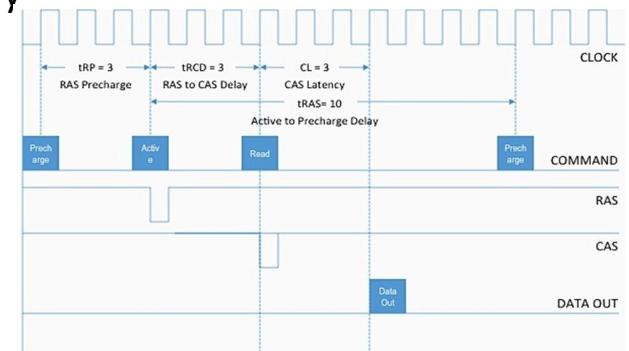
# Synchronous DRAM Read Timing

- Innovation – Pipeline access, command interface (ACTIVATE, READ)
- Benefit – No need to transfer column address

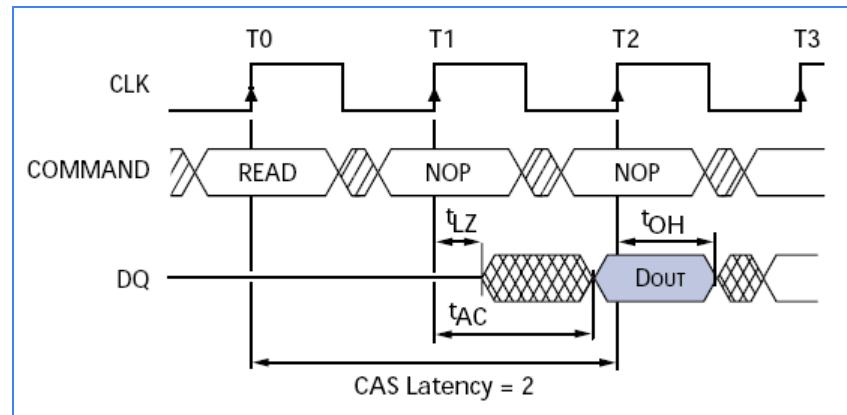
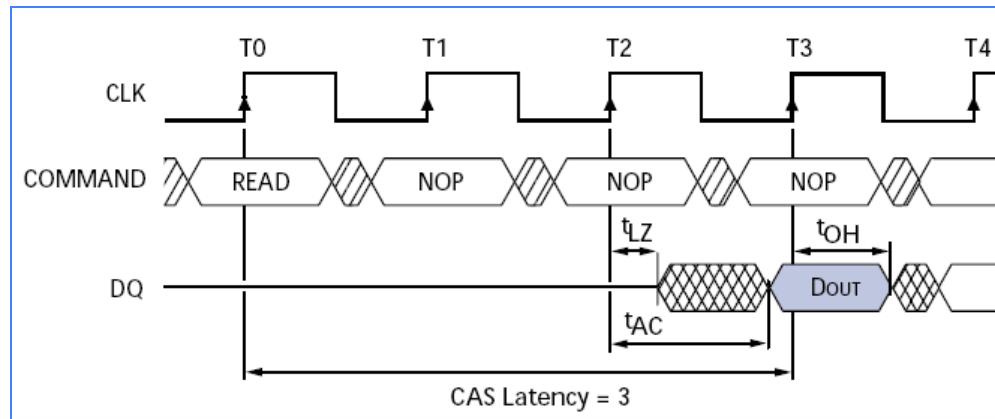


# Key SDRAM Timing Parameters

- $t_{RCD}$ : **Determines Latency**
  - Minimum time between an Activate command and Read command
  - Analogous to DRAM parameter  $t_{RCD}$  : Row Command Delay (RAS/CAS Delay)
- CL: CAS Latency **Determines Latency**
  - Time between Read command and first data valid
  - Analogous to DRAM parameter  $t_{CAC}$ : Column Access Time
- $t_{RAS}$ 
  - Time between Activate command and end of restoration of data in DRAM array
  - Analogous to DRAM parameter  $t_{RAS}$ : Row Address Strobe
- $t_{RP}$ 
  - Time to pre-charge DRAM array in preparation for another row access
  - Analogous to DRAM parameter  $t_{RP}$ : Row Precharge Delay
- $t_{RC}$  **Determines Bandwidth**
  - Time between successive row access to different rows
  - $t_{RC} = t_{RAS} + t_{RP}$
  - Analogous to DRAM parameter  $t_{RC}$ : Row Cycle Time

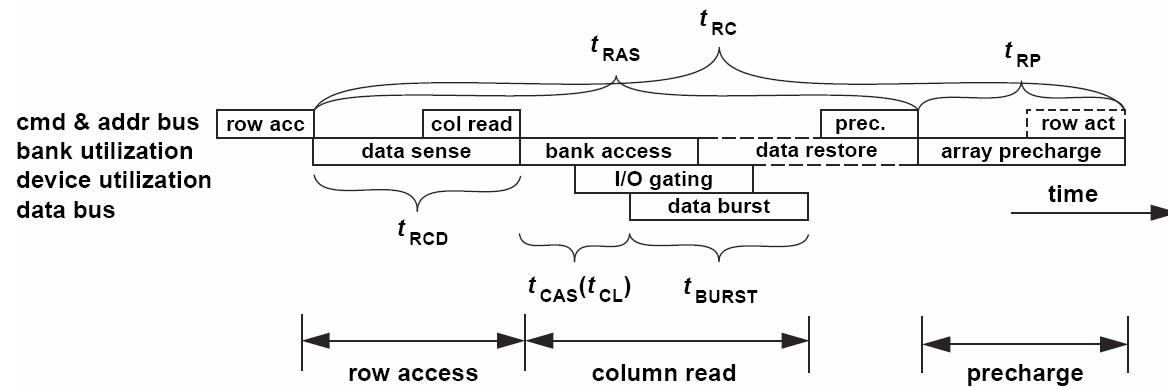


# CL – CAS Latency



SDR DRAM examples  
(DDR can have CAS latency of 2.5)

# Key SDRAM Timing Parameters



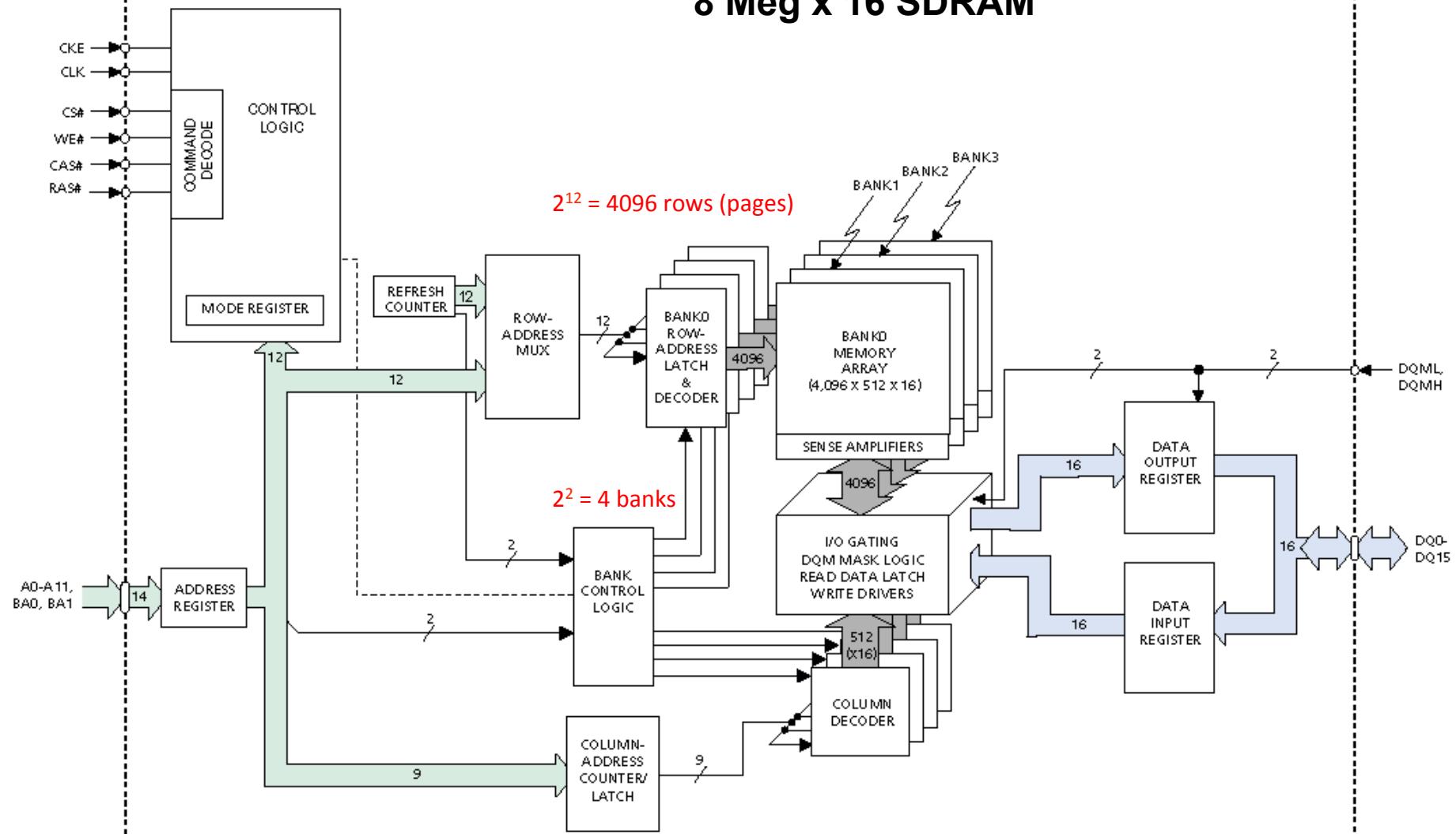
# SDRAM (Synchronous DRAM)

- Adopted for Pentium use
- Synchronous (clocked) interface (simplified timing)
- RAS, CAS signals combined to make “command”
- Ideal for cache line fill when bus width < cache line size
- Burst read/write
- Initial latency, then data every clock cycle
- Internal interleaved banks allow multiple rows (pages) to be “open” for read/write
- Self-contained refresh

# SDRAM Details

- Multiple “banks” of cell arrays are used to reduce access time:
  - Each bank is 4K rows by 512 “columns” by 16 bits (for our part)
- Read and Write operations are split into RAS (row access) followed by CAS (column access)
- These operations are controlled by sending commands
  - Commands are sent using the RAS, CAS, CS, & WE pins.
- Address pins are “time multiplexed”
  - During RAS operation, address lines select the bank and row
  - During CAS operation, address lines select the column.
- “ACTIVE” command “opens” a bank/row for operation
  - Transfers contents of the entire row to a buffer
- Subsequent “READ” or “WRITE” commands access the contents of the row buffer
- For burst reads and writes during “READ” or “WRITE” the starting address of the block is supplied.
  - Burst length is programmable as 1, 2, 4, 8 or a “full page” (entire row) with a burst terminate option
- Special commands are used for initialization (burst options etc.)
- A burst operation takes  $\approx 4 + n$  cycles (for n words)

# Functional Block Diagram 8 Meg x 16 SDRAM



# Banks Incorporated Into SDRAM

Memory address

Row	Bank	Column
-----	------	--------

- Why row/bank/column, not bank/row/column?
  - Consider spatial locality
  - Imagine accessing a series of sequential memory addresses
  - After exhausting a column, references to another bank
  - Consider if row/bank reversed
    - Bank would rarely be used, lose benefit of interleaving

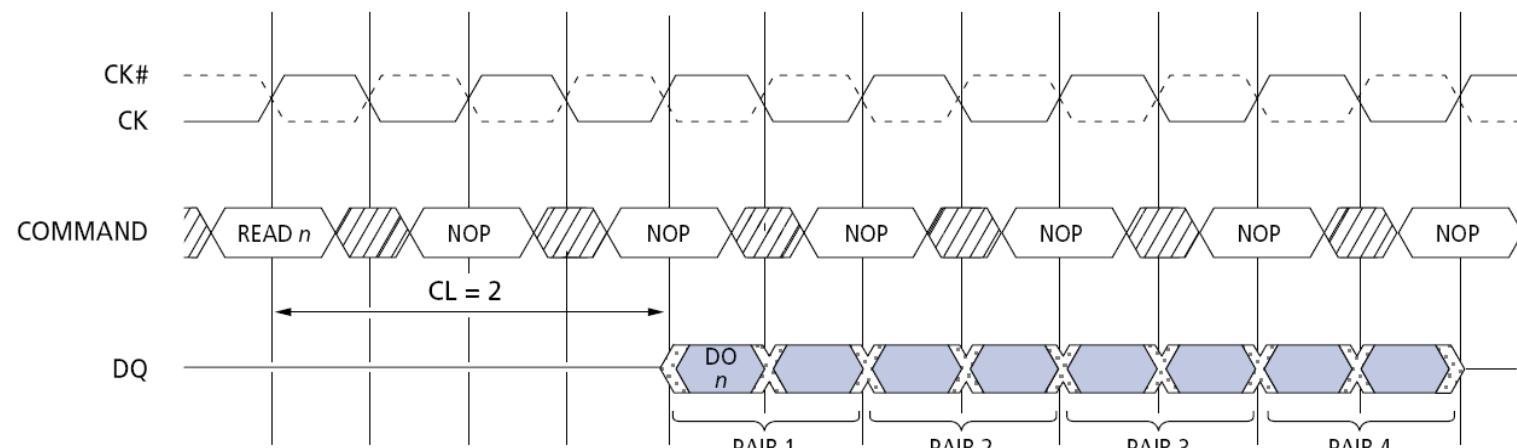
# Page Size

- ACTIVATE command reads data for all columns in the page
  - Reads data into sense amplifiers
  - Writes data back to data cells
- Consequently, page size is a factor in power consumption

$$\text{Page Size (bytes)} = \frac{\text{Number of Columns} \times \text{DQs}}{8}$$

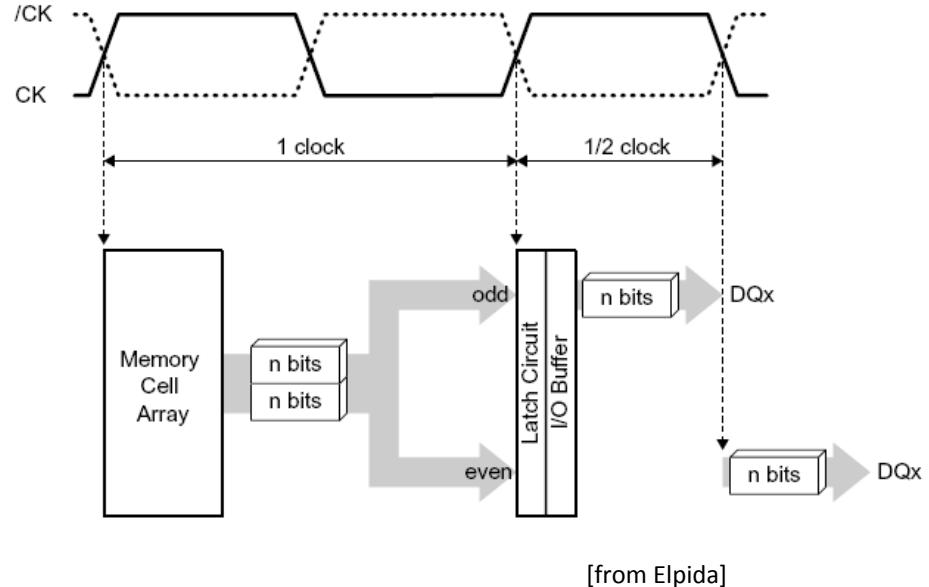
# DDR (Double Data Rate) SDRAM

- Innovation – Transfer data on rising and falling edges of clock
  - Same internal SDRAM core but 2n-prefetch
- Benefit – Twice the bandwidth, same control and signals as SDRAM
- Significant Differences –
  - Differential clock
  - Source synchronous (DQS)
  - Burst length of 2,4,8 only
  - CL = 2, 2.5, 3
  - SSTL-2 Stub Series Terminated Logic (2.5V) vs LVTTL (3.3V)

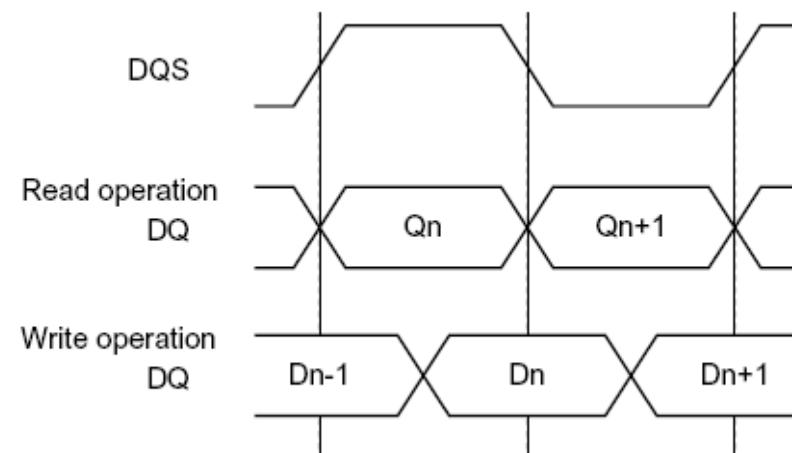


# DDR DRAM

- **$2n$  prefetch**
  - Use same DRAM core (cell array)
  - Fetch twice as many bits
  - Same latency for first data transfer

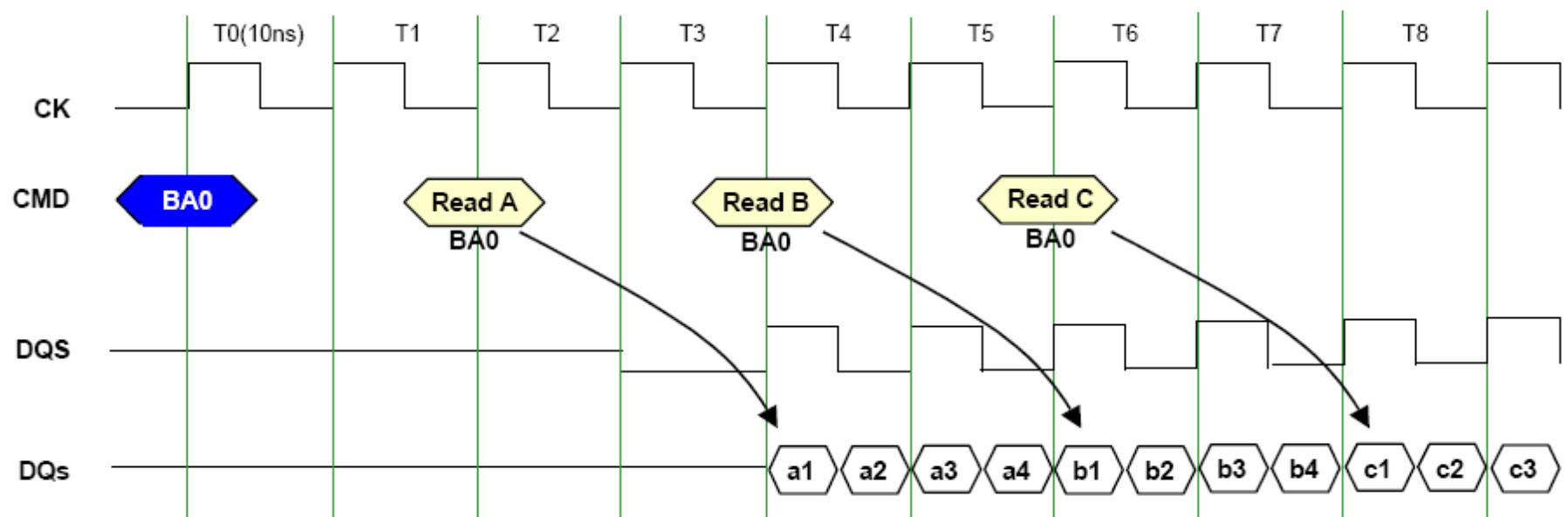


- **Source synchronous**
  - Data transfer is twice clock rate
  - Data strobe sent alongside data
    - Read: supplied by DRAM
      - Data aligned with strobe edge
    - Write: supplied by controller
      - Data centered on strobe edge



# DDR SDRAM Access Examples

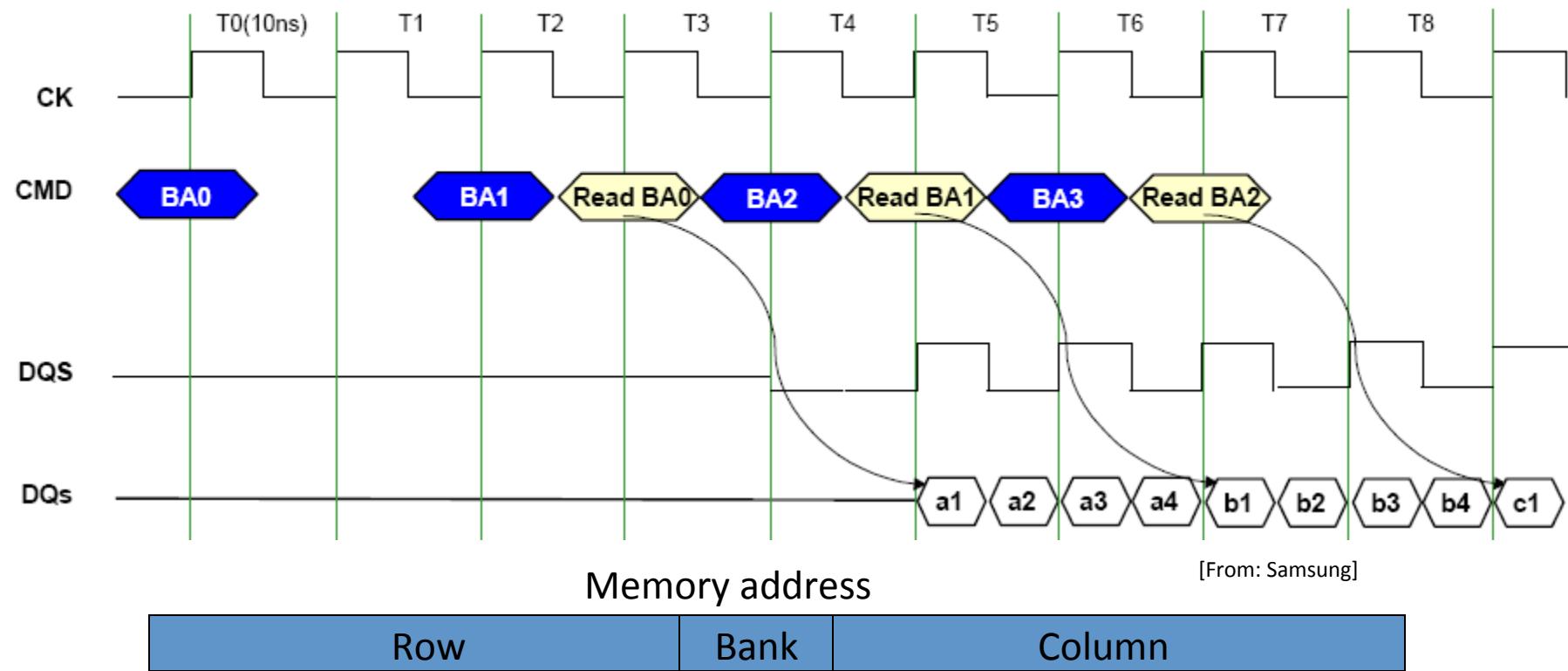
Reads from same open page/bank



[From: Samsung]

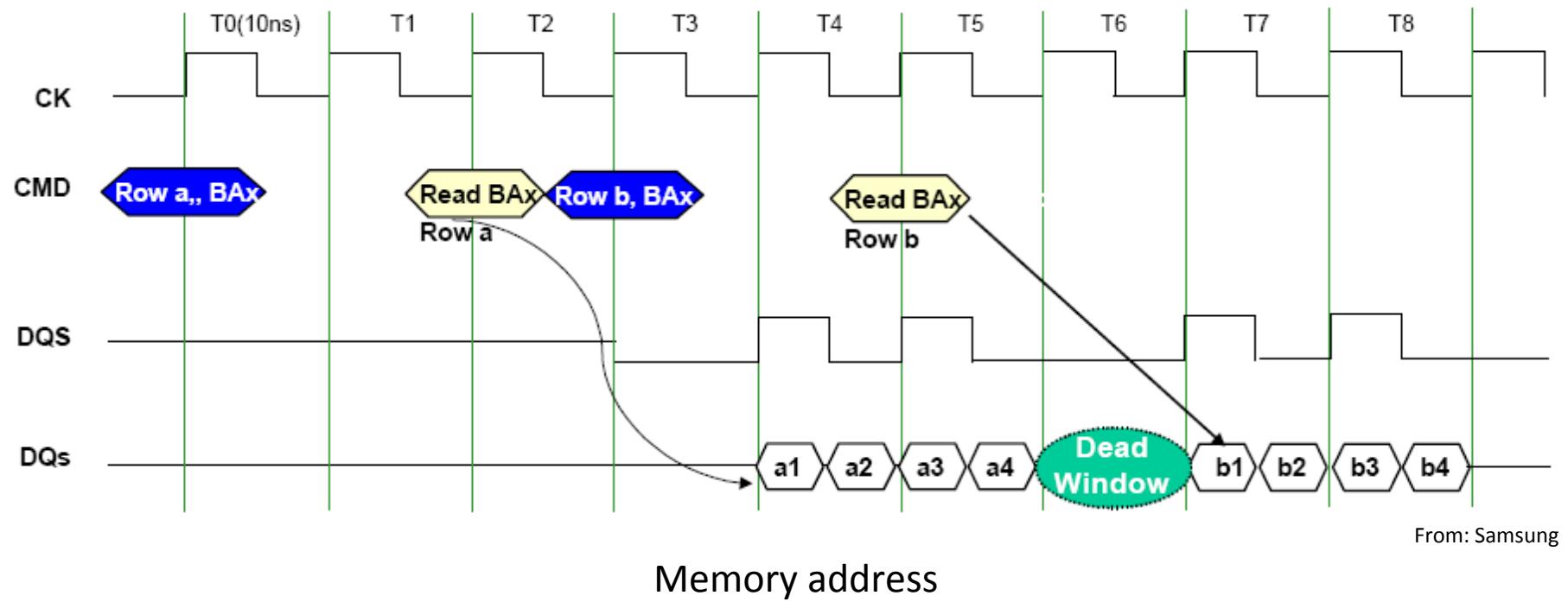
# DDR SDRAM Access Examples

Reads from different banks, open row

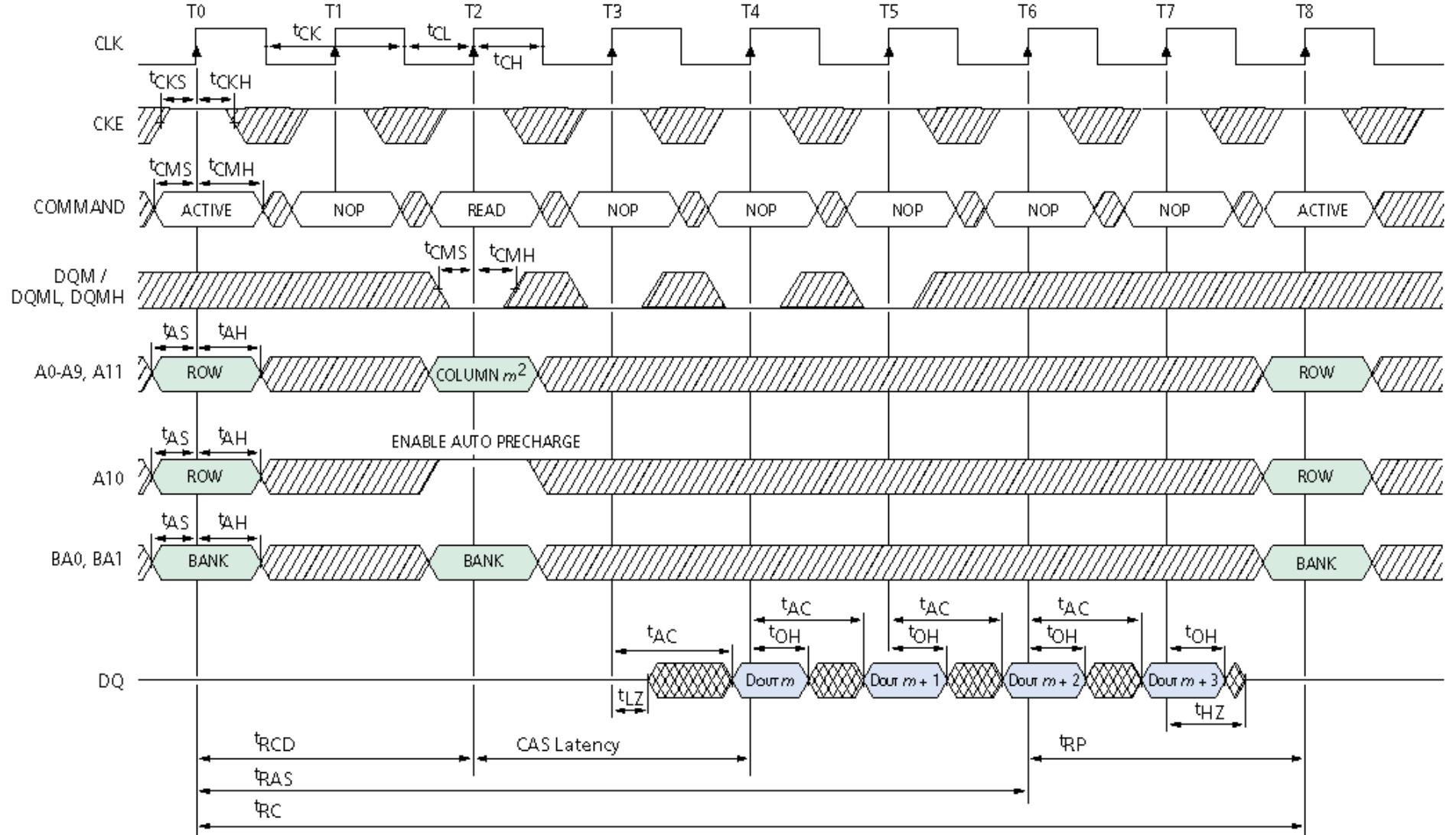


# DDR SDRAM Access Examples

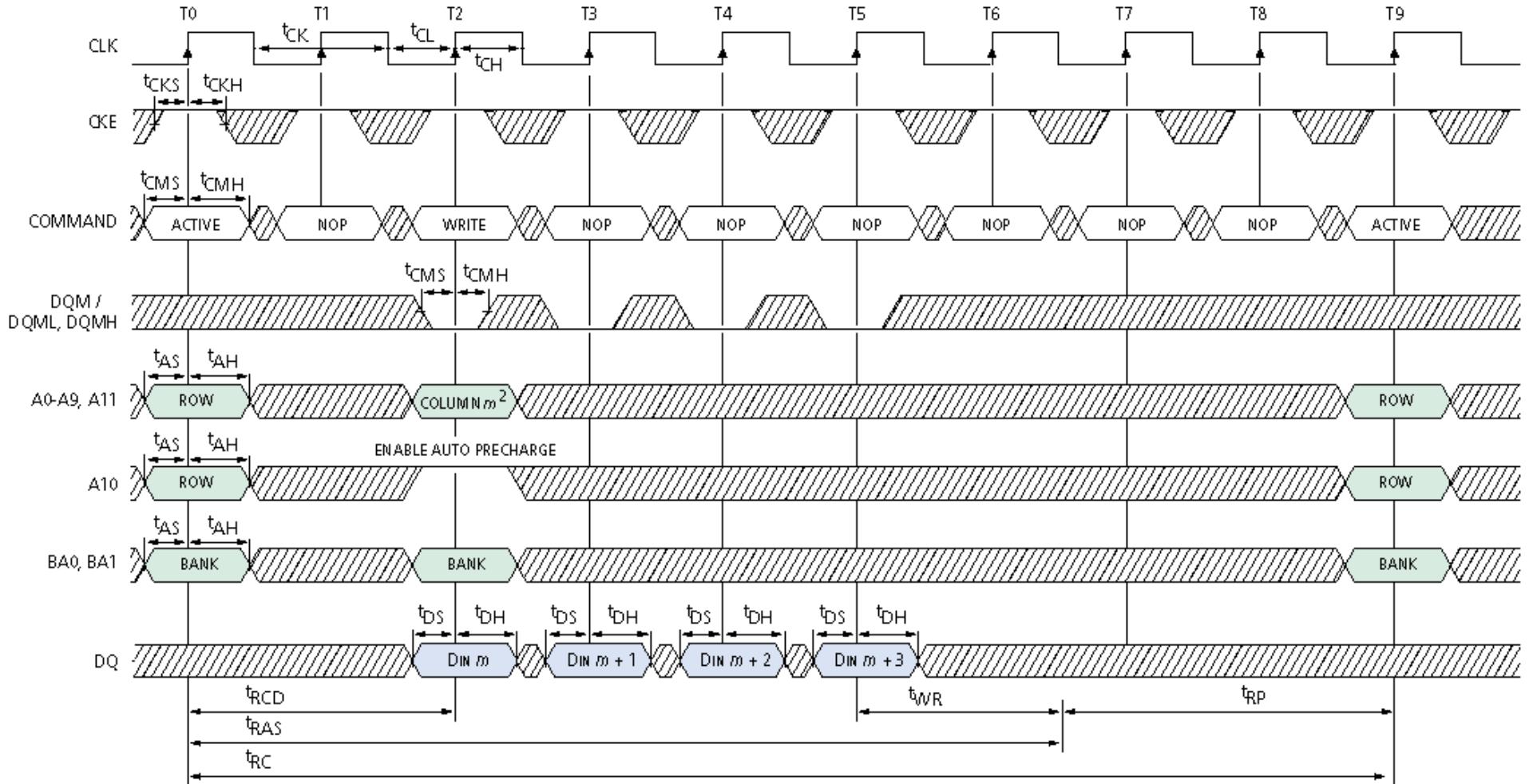
Reads from different row



# READ burst (with auto precharge)



# WRITE burst (with auto precharge)

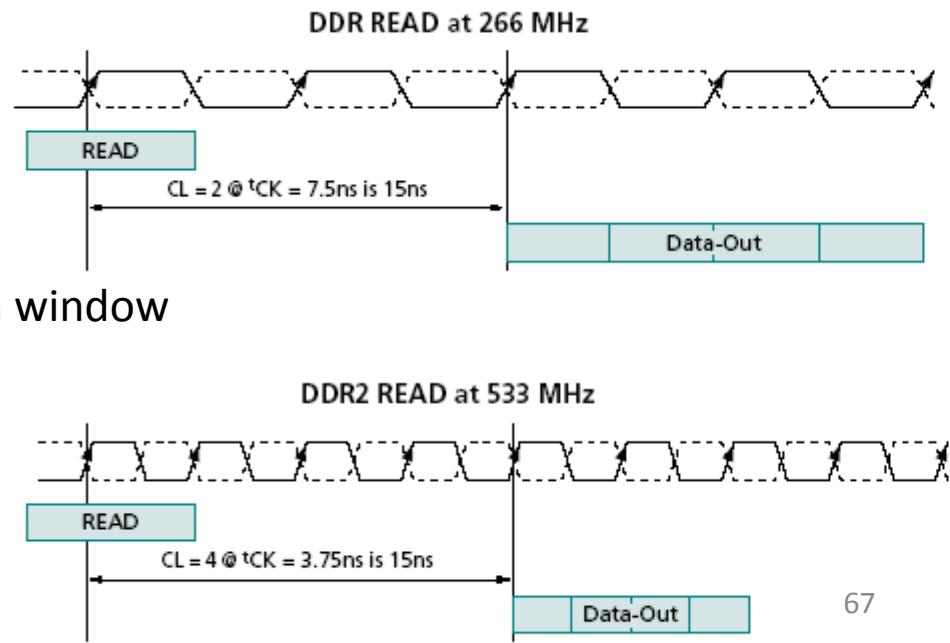


See datasheet for more details.

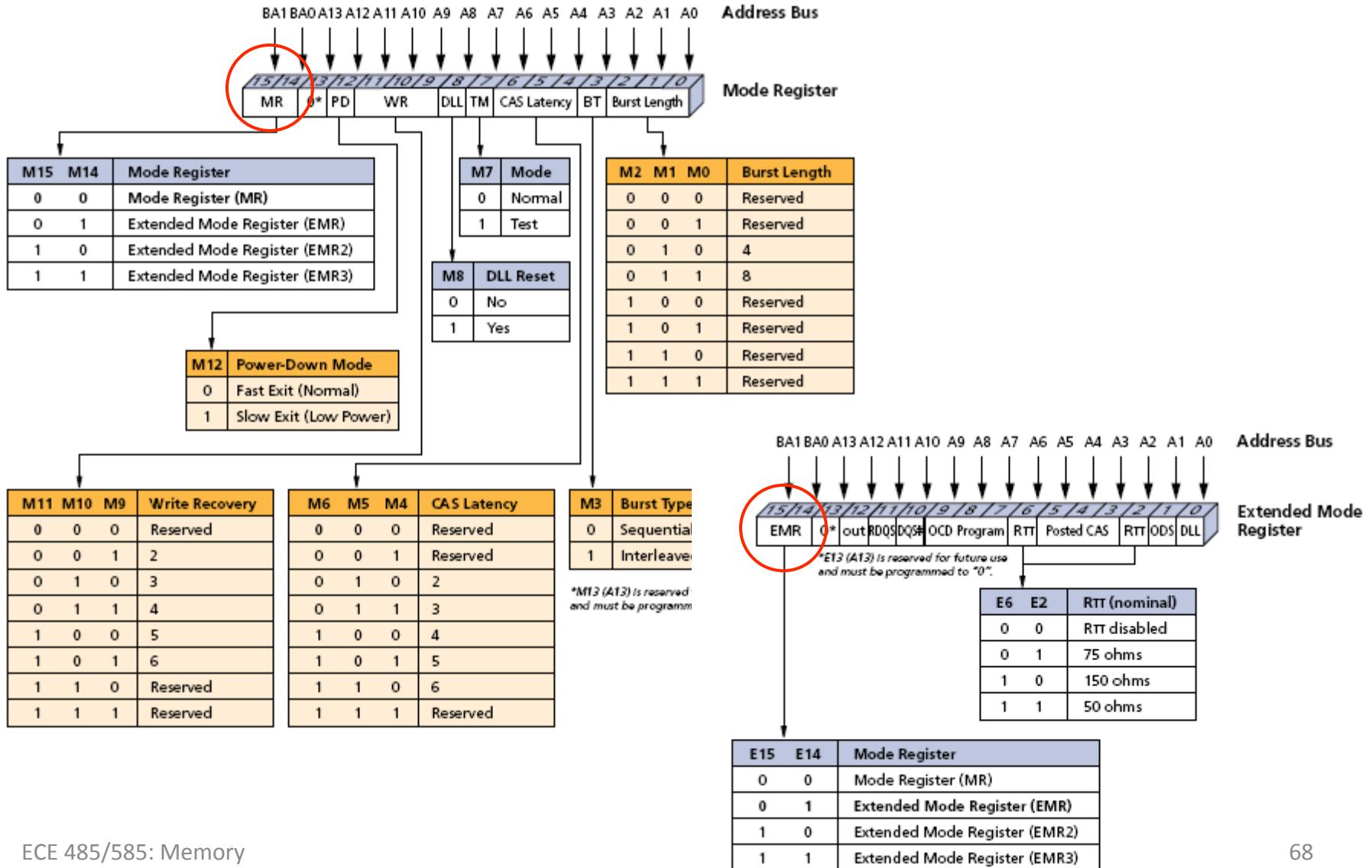
Verilog simulation models available.

# DDR-2 SDRAM

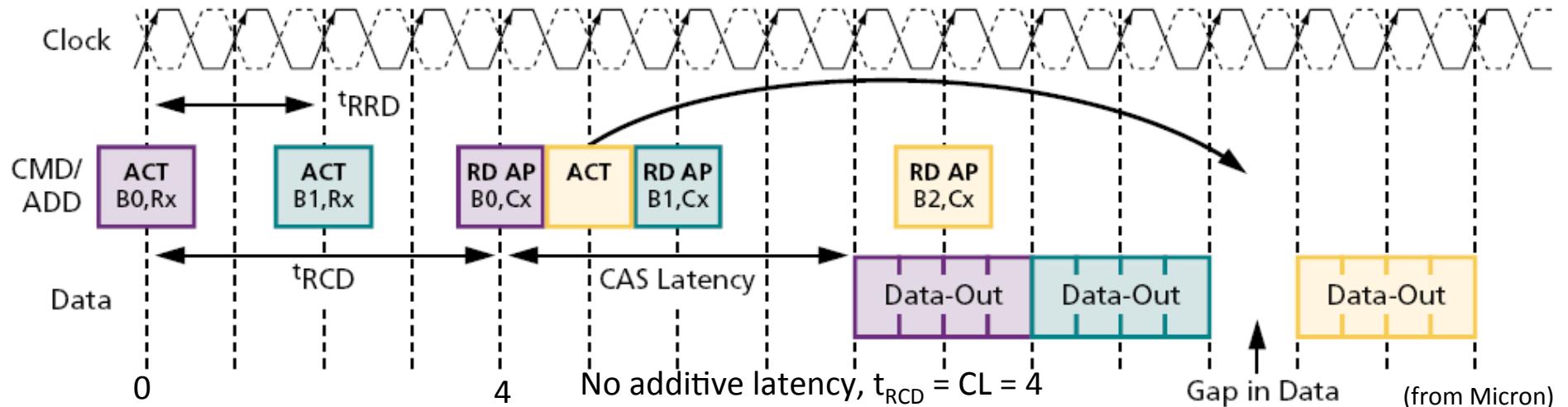
- During 2009-2010 the dominant high volume (PC) memory technology
- Innovation – 4n-prefetch, faster clocks
- Benefit – Increased bandwidth, same control and signals as DDR SDRAM
- Significant Differences
  - SSTL-18 (1.8V) vs. SSTL-2 (2.5V)
  - Low power (from lower supply voltage and new low power modes)
  - (Optional) differential strobe (DQS, DQS#)
  - ODT (On Die Termination)
  - 400 MHz clocks vs. 200 MHz
  - CL = 3,4,5
  - 2Gb devices
  - 4/8 banks vs. 4 banks
    - $t_{FAW}$  = four-bank activation window
  - Burst lengths of 4, 8
    - (no 2 because of 4n-prefetch)
  - “additive latency”



# Mode Register Changes



# Additive Latency



Can't place ACT command in cycle 4

slot occupied by RD AP: B0,Cx

So ACT is delayed by full cycle

so RD AP: B2, Cx is delayed

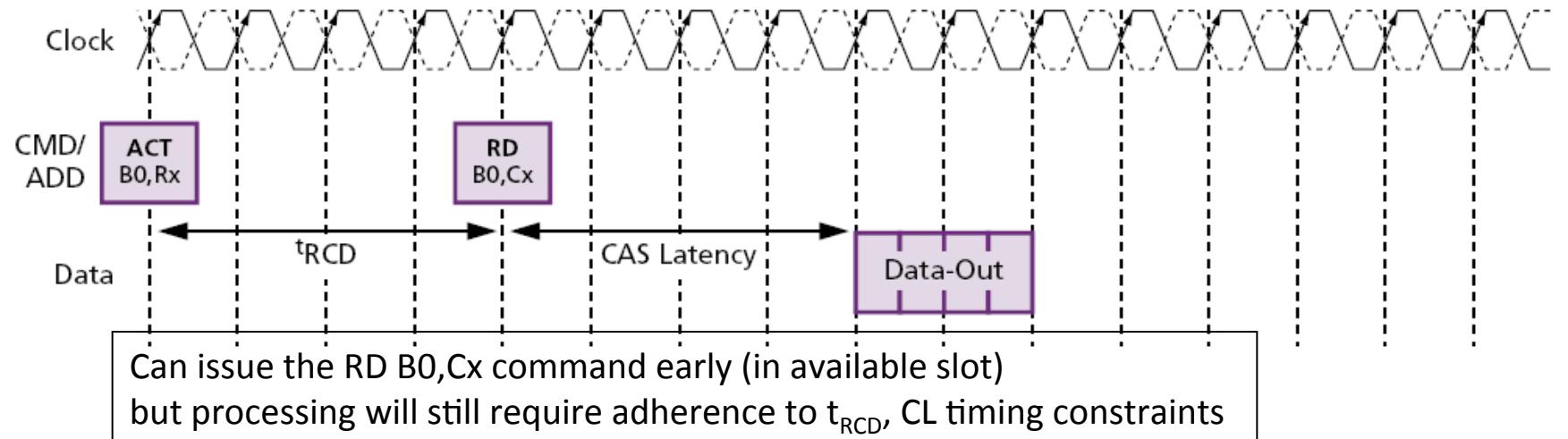
and resulting data out is delayed

ACT B<n>,R<x> = activate row <x> in bank <n>

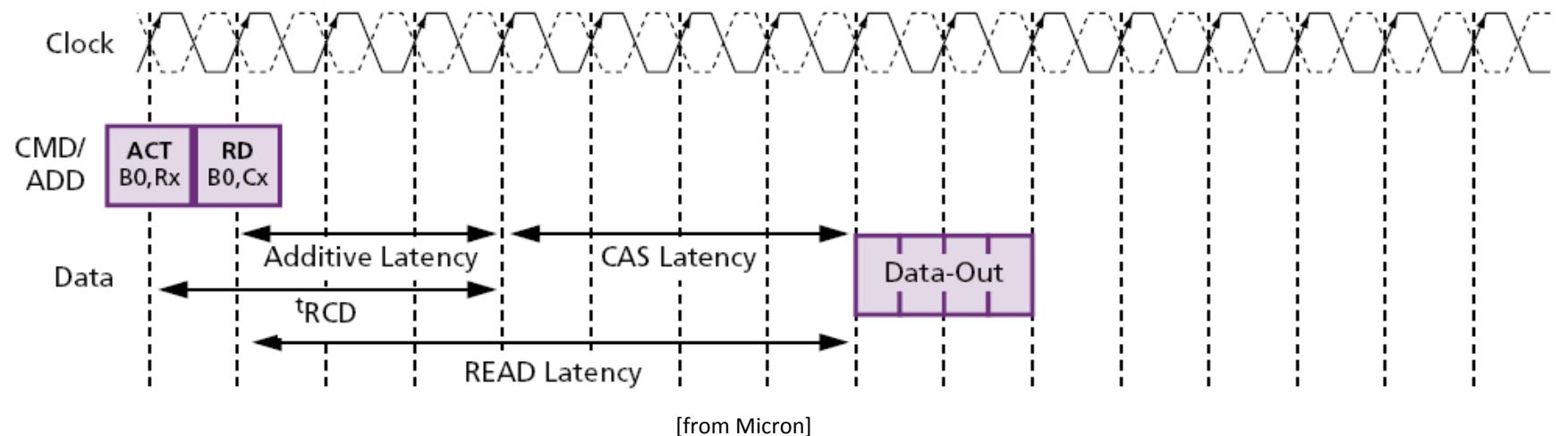
RD AP B<n>,C<x> = read column <x> from activated bank <n> (auto pre-charge)

# Additive Latency

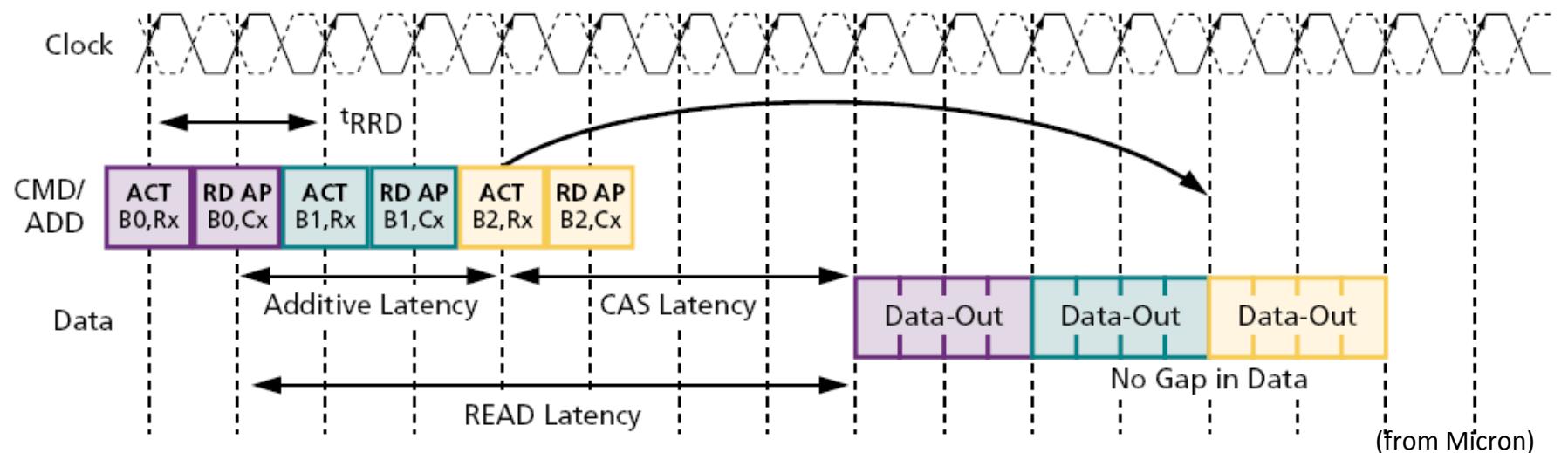
DDR2 with Additive Latency = 0



DDR2 with Additive Latency = ( $t_{RCD} - 1$ )



# Additive Latency



Permits continuous data read from DRAM

# DDR-3

- Current generation SDRAM
- Key Differences
  - 8n prefetch
  - SSTL-15 (1.5V) vs. SSTL-18 (1.8V)
    - Reduced power consumption (~30%)
  - 667-800 MHz clocks
    - 2x bandwidth of DDR-2
  - 8 banks vs. 4 banks
    - More open banks – less latency
- Adoption rate
  - Introduction in 2007 (insignificant quantities)
  - Samsung 8 Gb DDR3 described at ISSCC February 2009
  - Currently dominant PC memory (DDR4 specification underway)

# Micron DDR3 Datasheet



2Gb: x4, x8, x16 DDR3 SDRAM  
Features

## DDR3 SDRAM

MT41J512M4 – 64 Meg x 4 x 8 Banks

MT41J256M8 – 32 Meg x 8 x 8 Banks

MT41J128M16 – 16 Meg x 16 x 8 Banks

Speed Grade	Data Rate (MT/s)	Target tRCD-tRP-CL	tRCD (ns)	tRP (ns)	CL (ns)
-107 <sup>1, 2</sup>	1866	13-13-13	13.91	13.91	13.91
-125 <sup>1, 2</sup>	1600	11-11-11	13.75	13.75	13.75
-15 <sup>3</sup>	1333	10-10-10	15	15	15
-15E <sup>1</sup>	1333	9-9-9	13.5	13.5	13.5
-187	1066	8-8-8	15	15	15
-187E	1066	7-7-7	13.1	13.1	13.1

Parameter	512 Meg x 4	256 Meg x 8	128 Meg x 16
Configuration	64 Meg x 4 x 8 banks	32 Meg x 8 x 8 banks	16 Meg x 16 x 8 banks
Refresh count	8K	8K	8K
Row addressing	32K (A[14:0])	32K (A[14:0])	16K (A[13:0])
Bank addressing	8 (BA[2:0])	8 (BA[2:0])	8 (BA[2:0])
Column addressing	2K (A[11, 9:0])	1K (A[9:0])	1K (A[9:0])

# DDR3 Commands

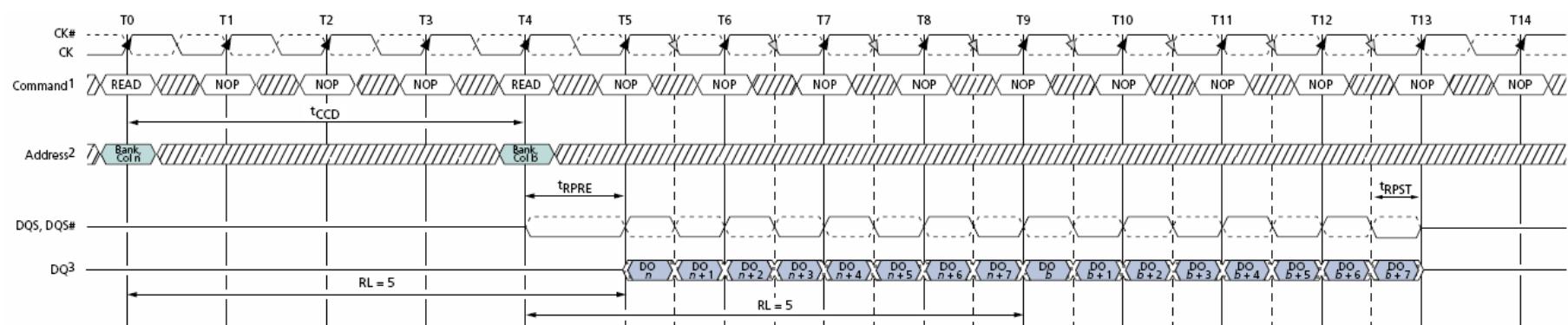
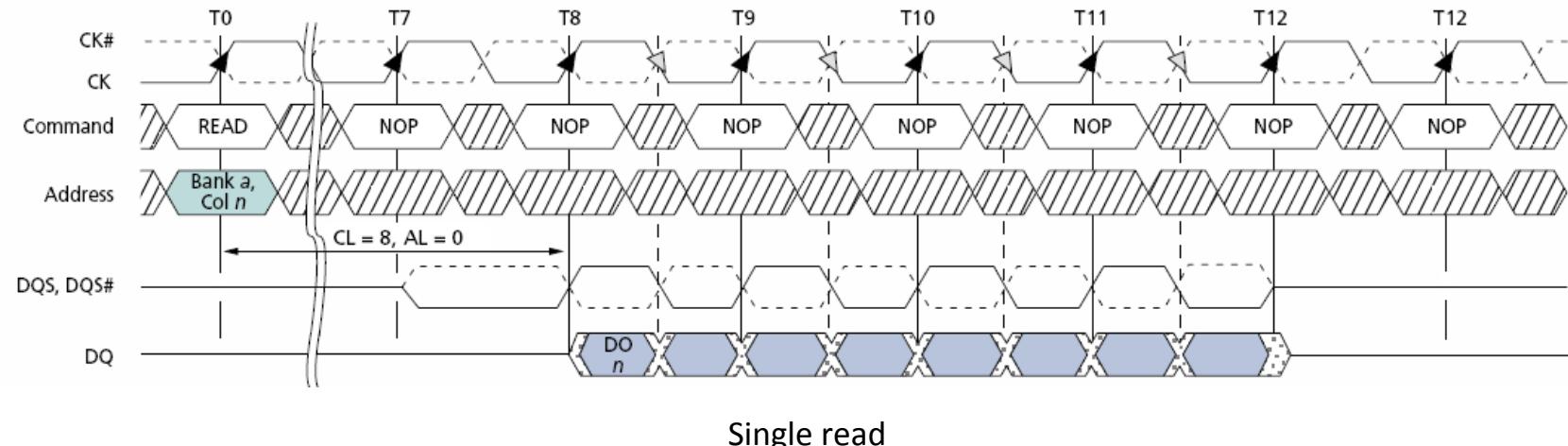
Function	Symbol	CKE		CS#	RAS#	CAS#	WE#	BA [2:0]	An	A12	A10	A[11, 9:0]	Notes	
		Prev. Cycle	Next Cycle											
MODE REGISTER SET	MRS	H	H	L	L	L	L	BA	OP code					
REFRESH	REF	H	H	L	L	L	H	V	V	V	V	V		
Self refresh entry	SRE	H	L	L	L	L	H	V	V	V	V	V	6	
Self refresh exit	SRX	L	H	H	V	V	V	V	V	V	V	V	6, 7	
				L	H	H	H							
Single-bank PRECHARGE	PRE	H	H	L	L	H	L	BA	V	V	L	V		
PRECHARGE all banks	PREA	H	H	L	L	H	L	V		V	H	V		
Bank ACTIVATE	ACT	H	H	L	L	H	H	BA	Row address (RA)					
WRITE	BL8MRS, BC4MRS	WR	H	H	L	H	L	L	BA	RFU	V	L	CA	8
	BC4OTF	WRS4	H	H	L	H	L	L	BA	RFU	L	L	CA	8
	BL8OTF	WRS8	H	H	L	H	L	L	BA	RFU	H	L	CA	8
WRITE with auto precharge	BL8MRS, BC4MRS	WRAP	H	H	L	H	L	L	BA	RFU	V	H	CA	8
	BC4OTF	WRAPS4	H	H	L	H	L	L	BA	RFU	L	H	CA	8
	BL8OTF	WRAPS8	H	H	L	H	L	L	BA	RFU	H	H	CA	8
READ	BL8MRS, BC4MRS	RD	H	H	L	H	L	H	BA	RFU	V	L	CA	8
	BC4OTF	RDS4	H	H	L	H	L	H	BA	RFU	L	L	CA	8
	BL8OTF	RDS8	H	H	L	H	L	H	BA	RFU	H	L	CA	8
READ with auto precharge	BL8MRS, BC4MRS	RDAP	H	H	L	H	L	H	BA	RFU	V	H	CA	8
	BC4OTF	RDAPS4	H	H	L	H	L	H	BA	RFU	L	H	CA	8
	BL8OTF	RDAPS8	H	H	L	H	L	H	BA	RFU	H	H	CA	8
NO OPERATION	NOP	H	H			H	H	H	V	V	V	V	V	9
Device DESELECTED	DES	H	H	H	X	X	X	X	X	X	X	X	X	10
Power-down entry	PDE	H	L	L	H	H	H	V	V	V	V	V	V	6
				H	V	V	V							
Power-down exit	PDX	L	H	L	H	H	H	V	V	V	V	V	V	6, 11
				H	V	V	V							
ZQ CALIBRATION LONG	ZQCL	H	H	L	H	H	L	X	X	X	H	X		12
ZQ CALIBRATION SHORT	ZQCS	H	H	L	H	H	L	X	X	X	L	X		

# DDR3 Commands (Notes)

1. Commands are defined by states of CS#, RAS#, CAS#, WE#, and CKE at the rising edge of the clock. The MSB of BA, RA, and CA are device-, density-, and configuration-dependent.
2. RESET# is LOW enabled and used only for asynchronous reset. Thus, RESET# must be held HIGH during any normal operation.
3. The state of ODT does not affect the states described in this table.
4. Operations apply to the bank defined by the bank address. For MRS, BA selects one of four mode registers.
5. "V" means "H" or "L" (a defined logic level), and "X" means "Don't Care."
6. See Table 70 (page 114) for additional information on CKE transition.
7. Self refresh exit is asynchronous.
8. Burst READs or WRITEs cannot be terminated or interrupted. MRS (fixed) and OTF BL/BC are defined in MR0.
9. The purpose of the NOP command is to prevent the DRAM from registering any unwanted commands. A NOP will not terminate an operation that is executing.
10. The DES and NOP commands perform similarly.
11. The power-down mode does not perform any REFRESH operations.
12. ZQ CALIBRATION LONG is used for either ZQinit (first ZQCL command during initialization) or ZQoper (ZQCL command after initialization).

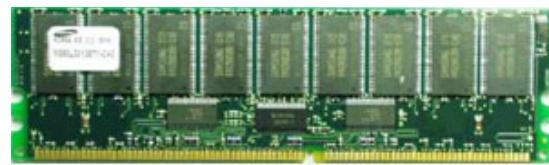
(RFU → Reserved for Future Use)

# DDR3 Read Cycle

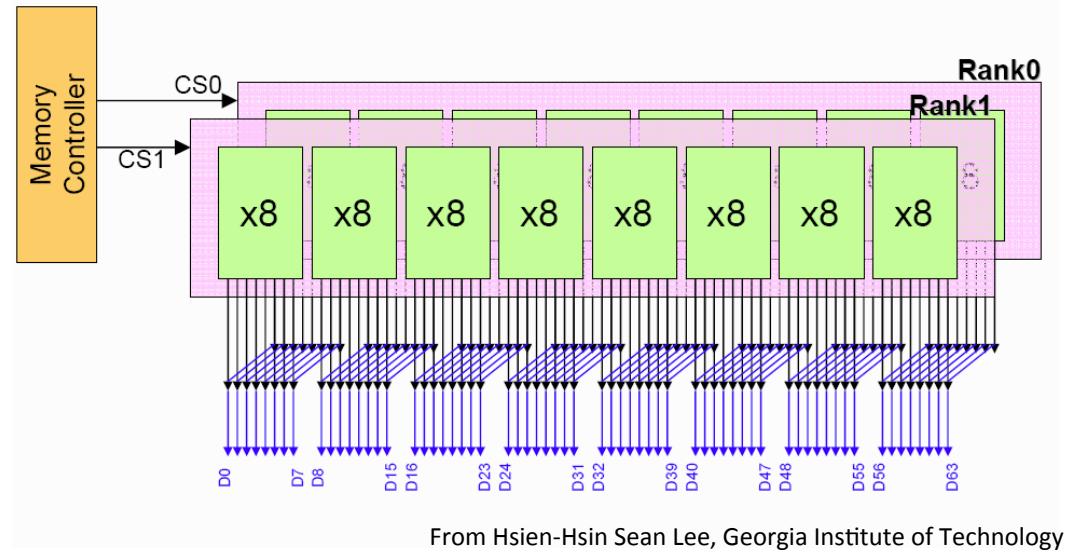


Back-to-back reads to open page

# Memory Modules

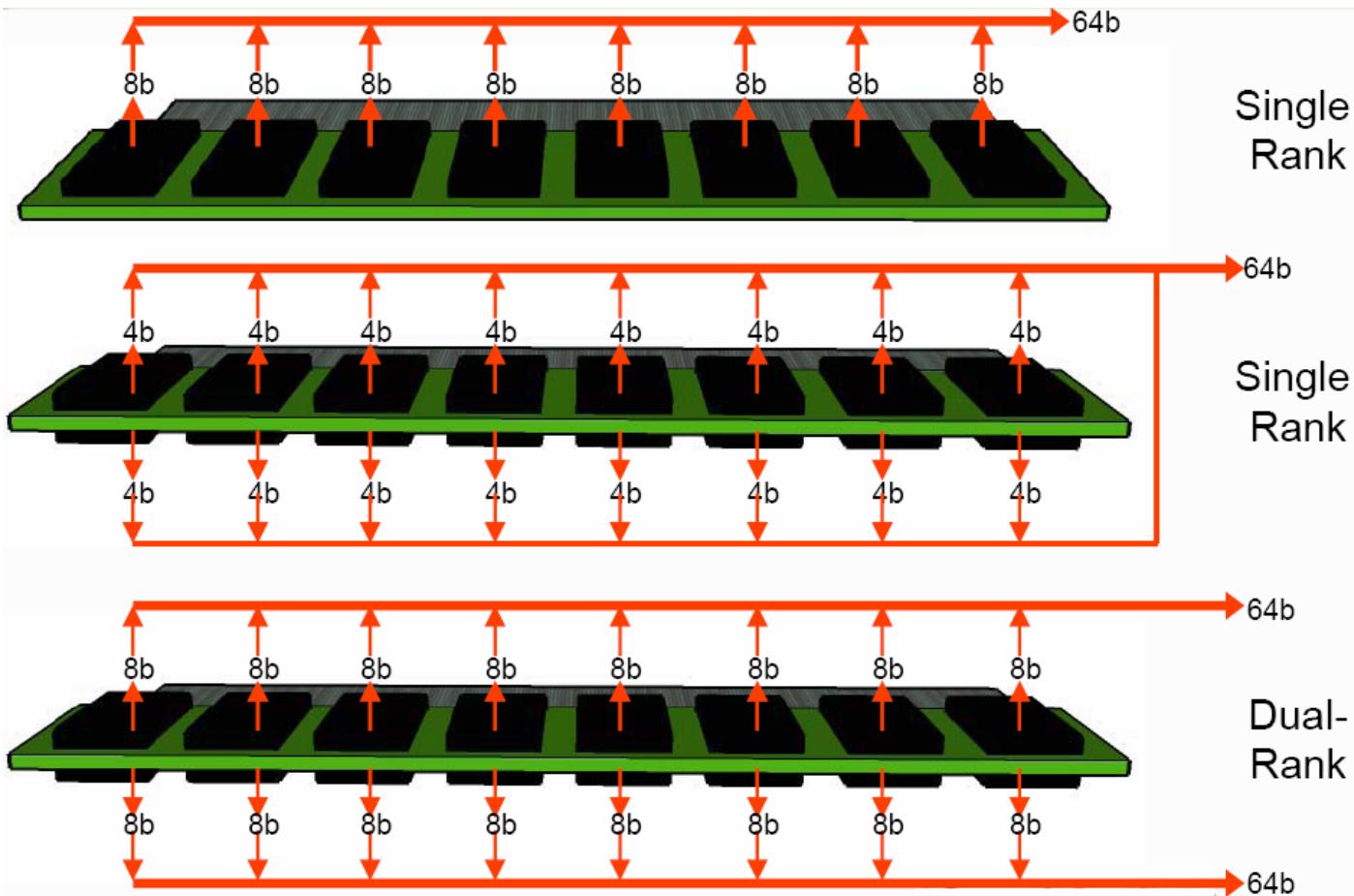


184 pin DDR SDRAM DIMM

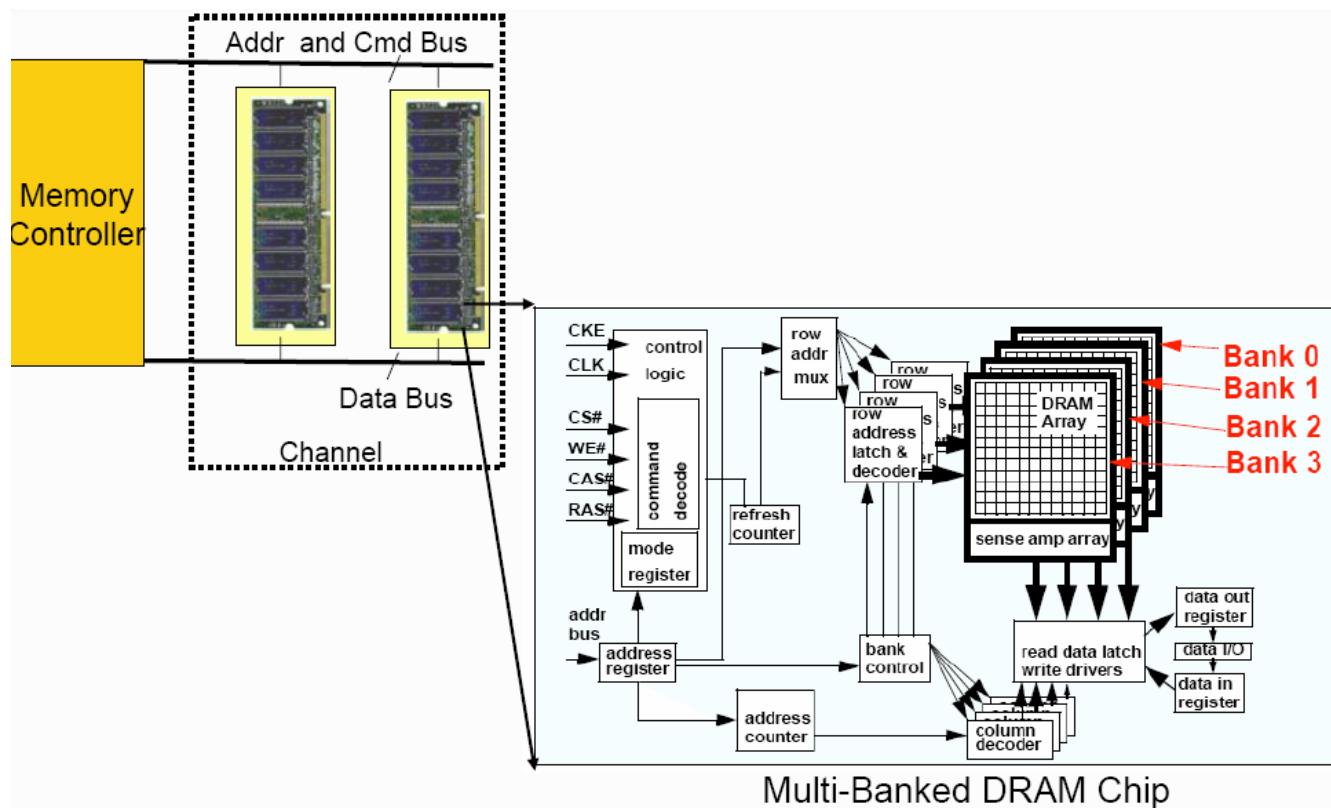


- All chips in a “rank” receive same address and control signals
- Each chip responsible for subset of data bits in its rank
- Module acts as high capacity DRAM with wide data path
  - Example: 8 chips, each 8 bits wide = 64 bits
- Easy to add/replace memory in a system
  - No need to solder or remove individual chips
- Memory granularity issue
  - What's the smallest increment in memory size?

# DRAM Ranks

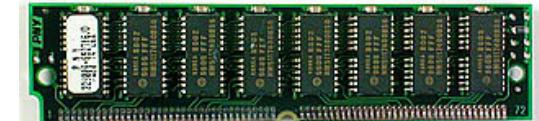


# Organization of DRAM Modules



# Memory Modules

- SIMM (Single Inline Memory Module)
  - 30-pin: some 286, most 386, some 486 systems
    - Page Mode, Fast Page mode devices
  - 72-pin: some 386, most 486, nearly all Pentium (before DIMM)
    - Fast Page Mode, EDO devices
- DIMM (Dual Inline Memory Module)
  - Dominant today
- SODIMM (Small Outline DIMM)
  - Used in notebooks, Apple iMac
- RIMM (Rambus RDRAM Module)
- SPD – Serial Presence Detect
  - 8-pin serial EEPROM on memory module
  - Key parameters for SDRAM controller
    - Number of row/column addresses
    - Number of ranks
    - Module width
    - Refresh rate/type
    - Error checking (none, parity, ECC)
    - Latency
    - Timing parameters



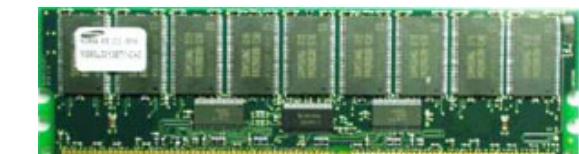
SIMM



168 pin SDRAM DIMM



SODIMM



184 pin DDR SDRAM DIMM



200 pin DDR2, DDR3 SDRAM DIMM



RIMM

# DRAM and DIMM Nomenclature

Device name	Clock	M transfers per sec	MB/sec Per DIMM	DIMM name
DDR200	100 MHz	200	1,600 MB/s	PC-1600
DDR266	133 MHz	266	2,133 MB/s	PC-2100
DDR333	166 MHz	333	2,666 MB/s	PC-2700
DDR400	200 MHz	400	3,200 MB/s	PC-3200
DDR2-400	200 MHz	400	3,200 MB/s	PC2-3200
DDR2-533	266 MHz	533	4,266 MB/s	PC2-4200
DDR2-667	333 MHz	666	5,333 MB/s	PC2-5300
DDR2-800	400 MHz	800	6,400 MB/s	PC2-6400
DDR2-1066	533 MHz	1066	8,533 MB/s	PC2-8500
DDR3-800	400 MHz	800	6,400 MB/s	PC3-6400
DDR3-1066	533 MHz	1066	8,500 MB/s	PC3-8500
DDR3-1333	666 MHz	1333	10,666 MB/s	PC3-10600
DDR3-1600	800 MHz	1600	12,800 MB/s	PC3-12800

M transfers/second = 2 x Clock Rate (DDR)

DRAM name incorporates M transfers per second

MB/sec = 8 x M transfers per second

DIMM name incorporates MB/sec (rounded)

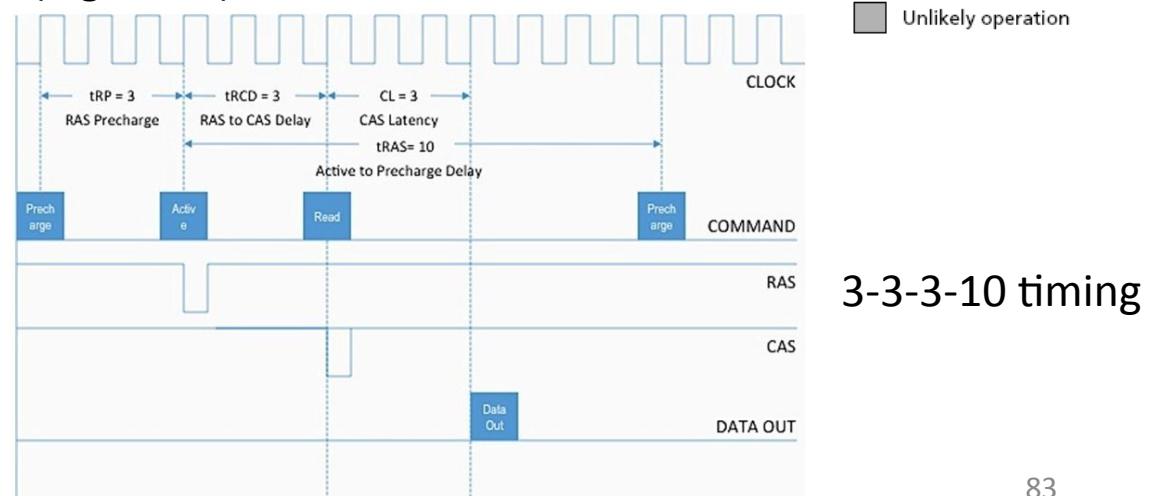
# DRAM History

- DRAMs: capacity +60%/yr, cost –30%/yr
  - 2.5X cells/area, 1.5X die size in 3 years
- DRAM fab costs \$2B
  - DRAM only: density, leakage v. speed
- Rely on increasing number of computers & memory per computer (>60% market)
  - SIMM or DIMM is replaceable unit
- Commodity, second source industry  
=> high volume, low profit, conservative
  - Standardization: JEDEC (Joint Electronic Devices Engineering Council)
    - EIA (Electronics Industries Alliance)
- Order of importance: 1) Cost/bit 2) Capacity
  - First RAMBUS: 10X BW, +30% cost => little impact

# DRAM/SDRAM Latency Specifications

- DRAM
  - Used 4 numbers (e.g. 4-1-1-1)
  - Indicates number of CPU cycles for 1st and successive accesses
- SDRAM
  - CAS Latency (CAS or CL)
  - Delay in clock cycles between request and the time the first data is available
  - PC133 module might be described as CAS-2, CAS=2, CL2, CL-2, or CL=2
- SDR-DRAM
  - CAS Latency of 1, 2, or 3
- DDR-DRAM
  - CAS Latency of 2 or 2.5
- When three numbers appear (e.g. 3-2-2)
  - CAS Latency ( $t_{CAC}$ )
  - RAS-to-CAS delay ( $t_{RCD}$ )
  - RAS pre-charge time ( $t_{RP}$ )
- DDR3 seeing use of four
  - CAS Latency ( $t_{CAS}$   $t_{CL}$ )
  - RAS-to-CAS delay ( $t_{RCD}$ )
  - RAS pre-charge time ( $t_{RP}$ )
  - RAS access time ( $t_{RAS}$ )

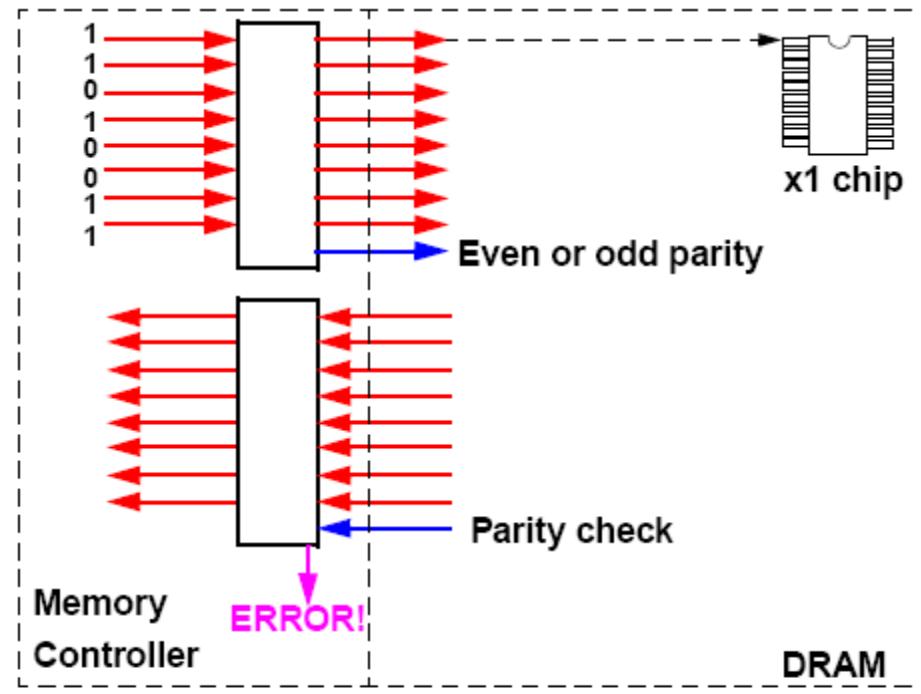
	$t_{CK}$	RL	3	4	5	6
DDR2-400	5ns		15ns	20ns	25ns	30ns
DDR2-533	3.75ns		11.25ns	15ns	18.75ns	22.5ns
DDR2-667	3ns		9ns	12ns	15ns	18ns
DDR2-800	2.5ns		7.5ns	10ns	12.5ns	15ns



# Error Correction

- Motivation
  - Failures/time proportional to number of bits
  - As DRAM cells size & voltages shrink, more vulnerable
- Why not issue on your PC?
  - Failure rate was low
  - Few consumers would know what to do anyway
  - DRAM banks too large now
  - Servers always corrected memory systems
- Sources
  - Alpha particles (impurities in IC manufacturing)
  - Cosmic rays (vary with altitude)
    - Bigger problem in Denver and on space-bound electronics
  - Noise
- Need to handle failures throughout memory subsystem
  - DRAM chips, module, bus
  - DRAM chips don't incorporate ECC
  - Store the ECC bits in DRAM alongside the data bits
  - Chipset (or integrated controller) handles ECC

# Error Detection: Parity

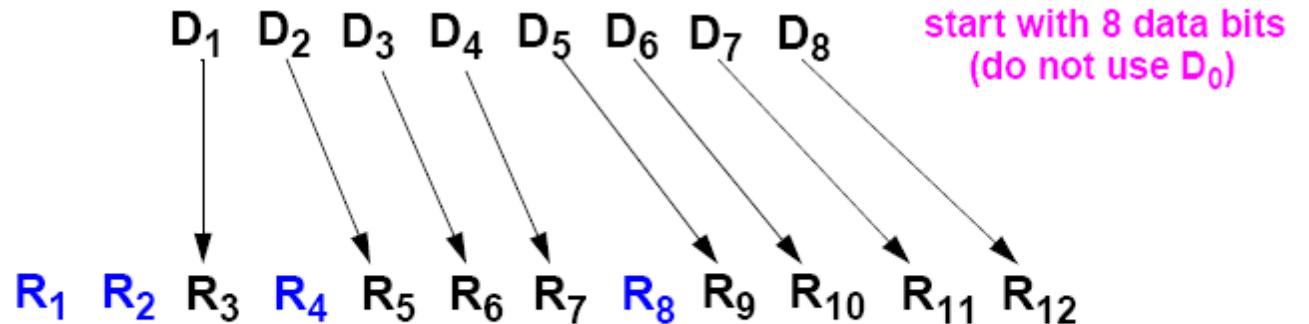


- Odd bit error detection
- No error correction capability
- Overhead: 1 bit per byte

[from Bruce Jacob]

# Error Correction Codes (ECC)

Single bit error correction  
requires  $n+1$  check bits for  $2^n$  data bits



Reserve  $R_m$  bit positions where  $m$  is a power of 2.  
Move data bits into available bit positions. (skip  $R_0$ )  
Display “m” in binary format.

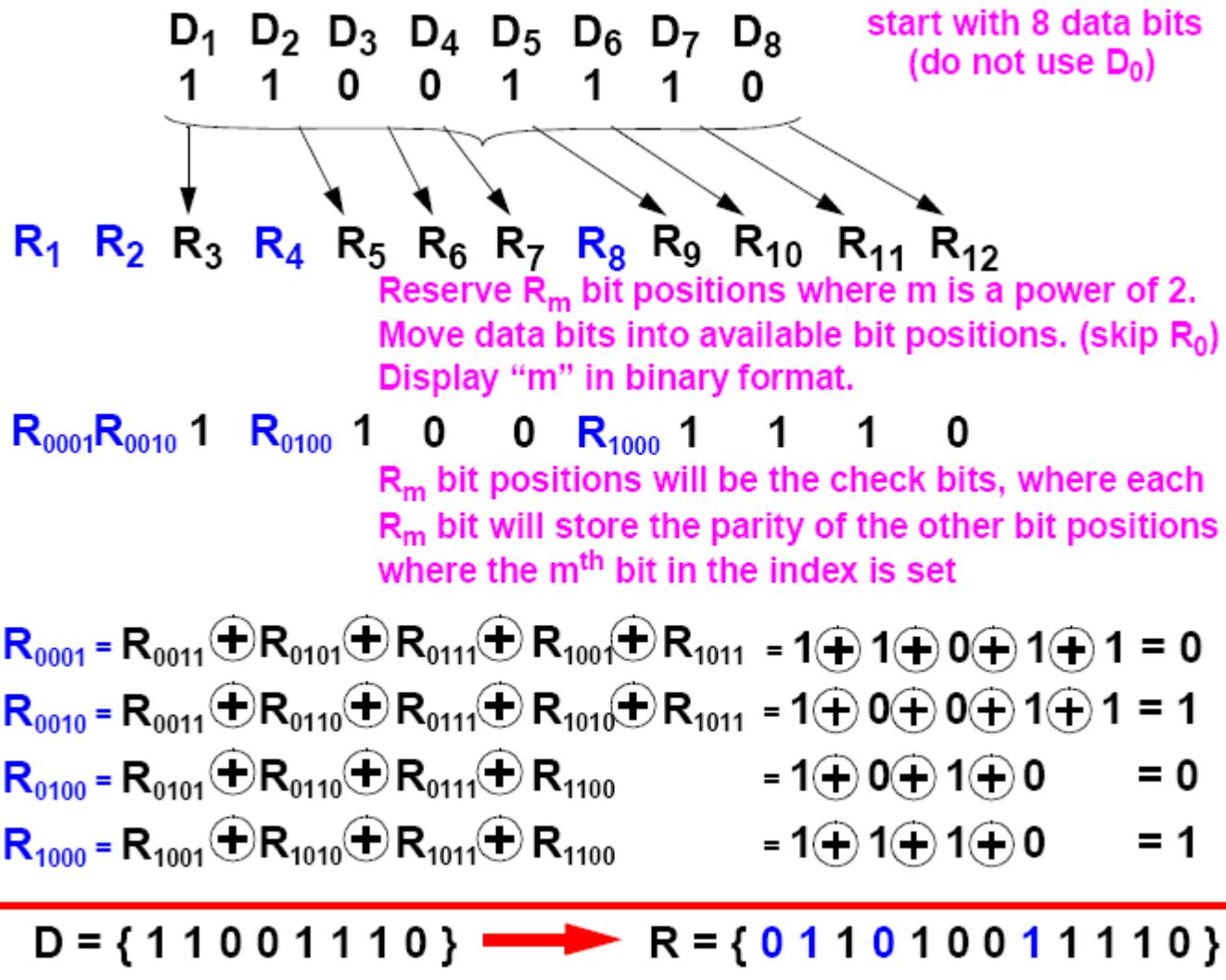
$R_{0001} R_{0010} R_{0011} R_{0100} R_{0101} R_{0110} R_{0111} R_{1000} R_{1001} R_{1010} R_{1011} R_{1100}$

$R_m$  bit positions will be the check bits, where each  $R_m$  bit will store the parity of the other bit positions where the  $m^{\text{th}}$  bit in the index is set

$$R_{0001} = R_{0011} \oplus R_{0101} \oplus R_{0111} \oplus R_{1001} \oplus R_{1011}$$

$$R_{0010} = R_{0011} \oplus R_{0110} \oplus R_{0111} \oplus R_{1010} \oplus R_{1011}$$

# Error Correction Codes (ECC)



# Error Correction Codes (ECC)

An example: decoding and verifying

$$R = \{ 0 1 1 0 1 0 0 1 1 1 1 0 \}$$

$$R = \{ 0 1 1 0 1 0 0 1 1 1 0 0 \}$$

One bit error. Can we detect and correct?

Recompute check bits

$$R_{0001} = R_{0011} \oplus R_{0101} \oplus R_{0111} \oplus R_{1001} \oplus R_{1011} = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$R_{0010} = R_{0011} \oplus R_{0110} \oplus R_{0111} \oplus R_{1010} \oplus R_{1011} = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$R_{0100} = R_{0101} \oplus R_{0110} \oplus R_{0111} \oplus R_{1100} = 1 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$R_{1000} = R_{1001} \oplus R_{1010} \oplus R_{1011} \oplus R_{1100} = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

XOR old check bits against new check bits

	$R_{1000}$	$R_{0100}$	$R_{0010}$	$R_{0001}$	
	1	0	1	0	Old
$\oplus$	0	0	0	1	New
$R_{1011}$	1	0	1	1	Difference !

Bit position 11 is rotten

# Error Correction Codes (ECC)

An example: multiple bit errors

$$R = \{ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \}$$

$$R = \{ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \} \quad \text{Multi bit error. Can we detect and correct?}$$

Recompute check bits

$$R_{0001} = R_{0011} \oplus R_{0101} \oplus R_{0111} \oplus R_{1001} \oplus R_{1011} = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$R_{0010} = R_{0011} \oplus R_{0110} \oplus R_{0111} \oplus R_{1010} \oplus R_{1011} = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$R_{0100} = R_{0101} \oplus R_{0110} \oplus R_{0111} \oplus R_{1100} = 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$R_{1000} = R_{1001} \oplus R_{1010} \oplus R_{1011} \oplus R_{1100} = 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

XOR old check bits against new check bits

	$R_{1000}$	$R_{0100}$	$R_{0010}$	$R_{0001}$	
	1	0	1	0	Old
$R_{0111}$	1	1	0	1	New

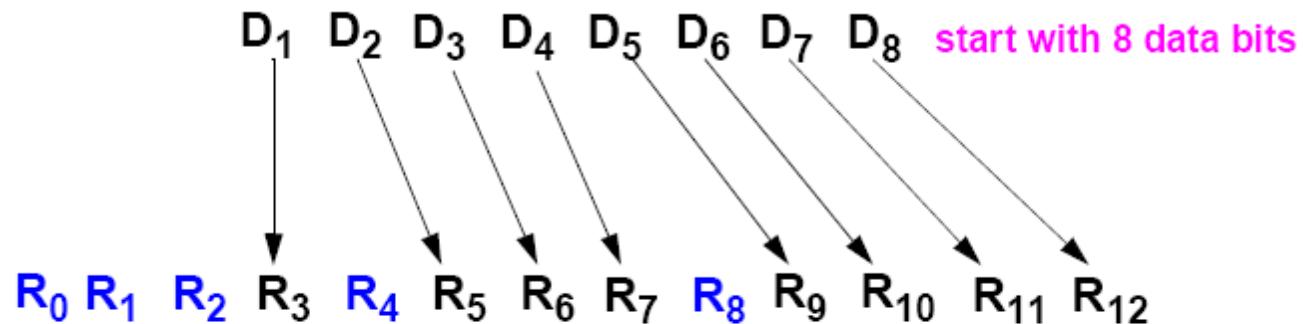
⊕

→      0      1      1      1      Difference !

# Error Correction Codes (ECC)

Add another check bit – SECDED

Single Error Correction Double Error Detection



$$R_0 = R_1 \oplus R_2 \oplus \dots \oplus R_{(2^n + n + 1)}$$

requires  $n+2$  check bits for  $2^n$  data bits

# Error Correction Codes (ECC)

Double Error Detection Example (even parity)

$$R = \{ 1 0 1 1 0 1 0 0 1 1 1 1 0 \}$$

$$R = \{ 1 0 1 1 0 1 0 0 1 1 1 0 1 \} \quad \text{Multi bit error. Can we detect and correct?}$$

Recompute check bits

$$R_{0001} = R_{0011} \oplus R_{0101} \oplus R_{0111} \oplus R_{1001} \oplus R_{1011} = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$R_{0010} = R_{0011} \oplus R_{0110} \oplus R_{0111} \oplus R_{1010} \oplus R_{1011} = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$R_{0100} = R_{0101} \oplus R_{0110} \oplus R_{0111} \oplus R_{1100} = 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$R_{1000} = R_{1001} \oplus R_{1010} \oplus R_{1011} \oplus R_{1100} = 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

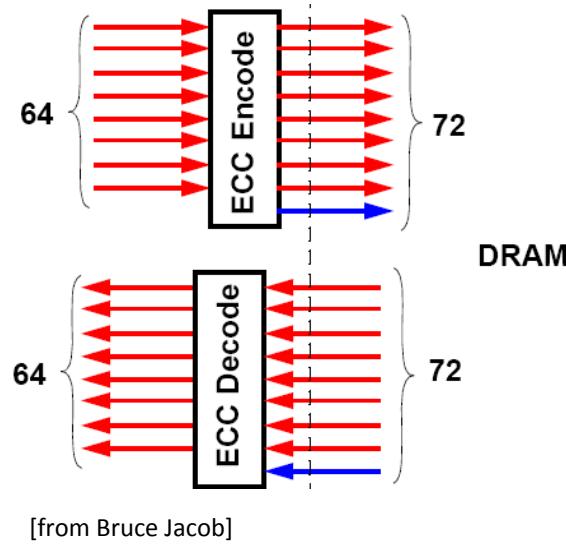
XOR old check bits against new check bits

$R_{1000}$	$R_{0100}$	$R_{0010}$	$R_{0001}$	
Old	New			
1	0	1	0	
⊕	1	1	0	1
	0	1	1	Difference !

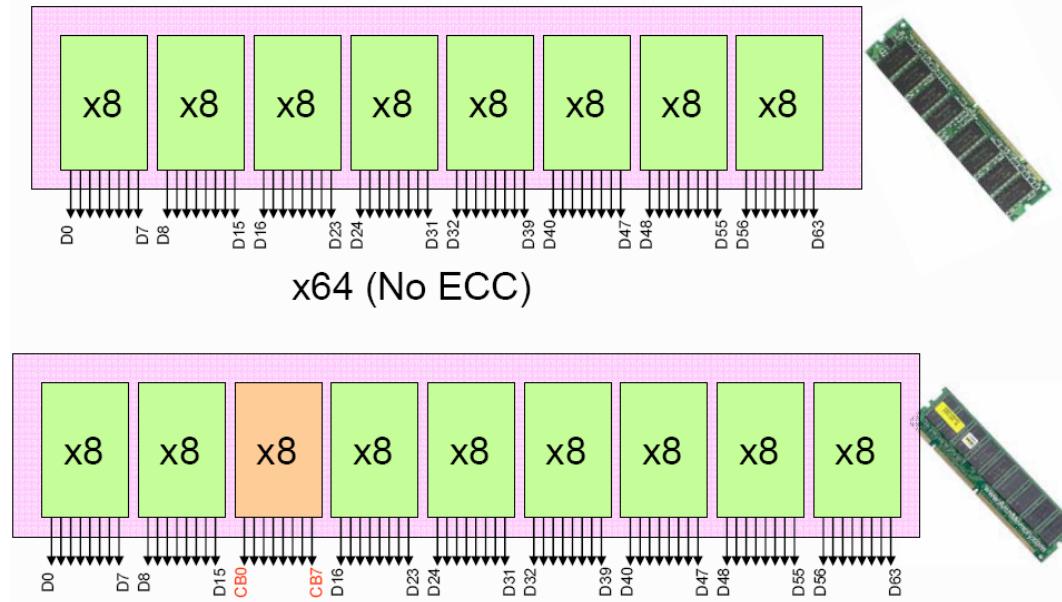
XOR check bits tell us there is error, but  $R_0$  parity says all is well. This is a 2 bit error, cannot be corrected.

→ Actually it says there wasn't an odd number of bit errors

# Error Correction Codes (ECC)

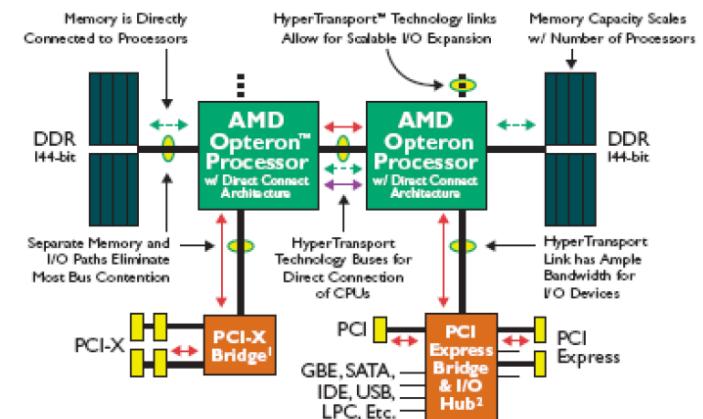
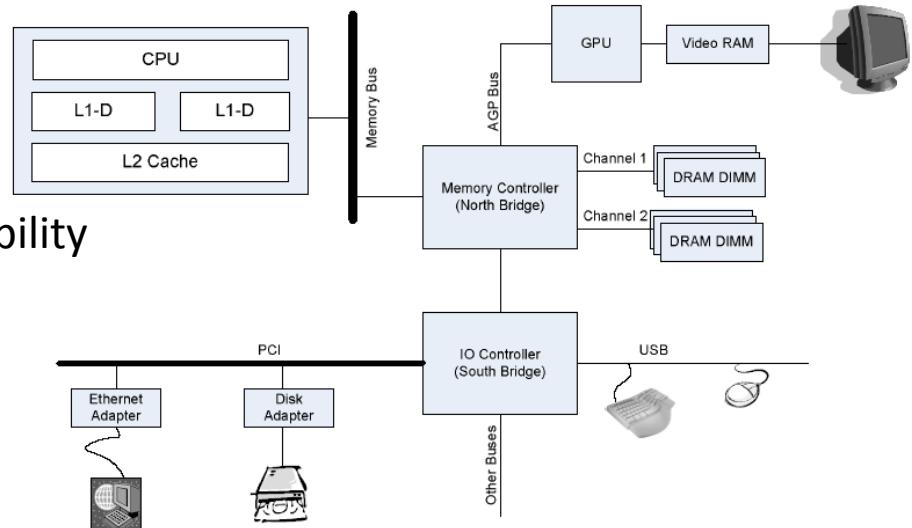


64-bit data path + 8 bits ECC stored to DRAM module

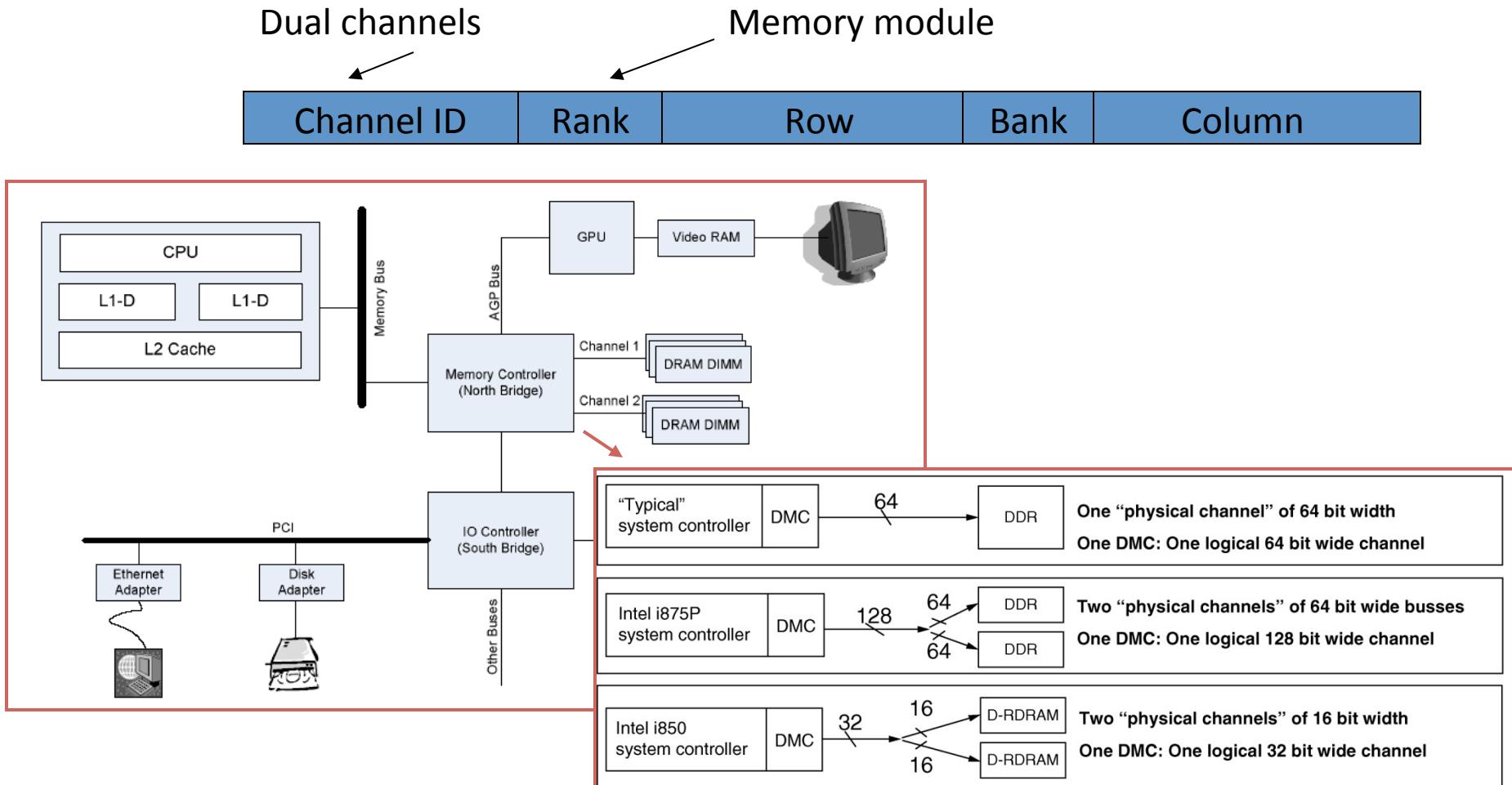


# Memory Controllers

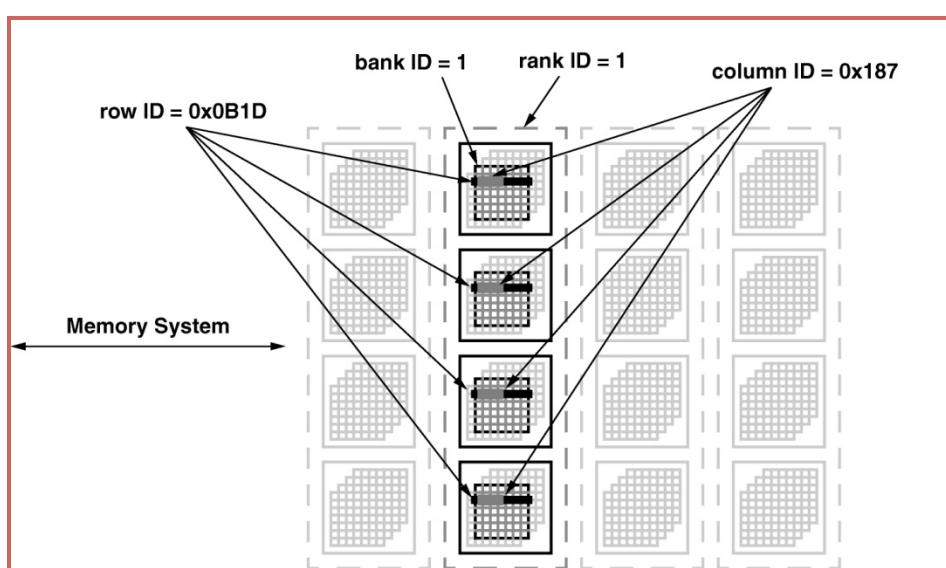
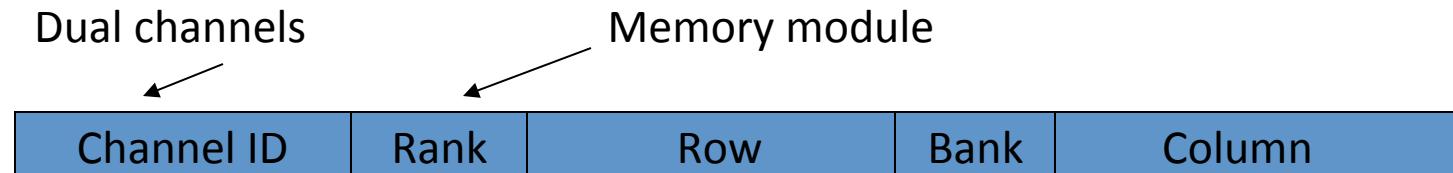
- Handle the actual interface to memory
  - Determine memory configuration/capability
  - Memory Timing/Signal interface
  - Address Mapping
    - Physical Address to Memory Topology
  - Error Correction
  - Scheduling
  - Refresh
- Reside in North Bridge of chipset
  - Intel prior to Nehalem
  - MCH (Memory Controller Hub)
  - Isolates CPU from memory technology/device changes
- Integrated with microprocessor
  - AMD, Intel Nehalem
  - Low latency for high performance
  - Opens possibility for processor-directed hints



# Address Mapping

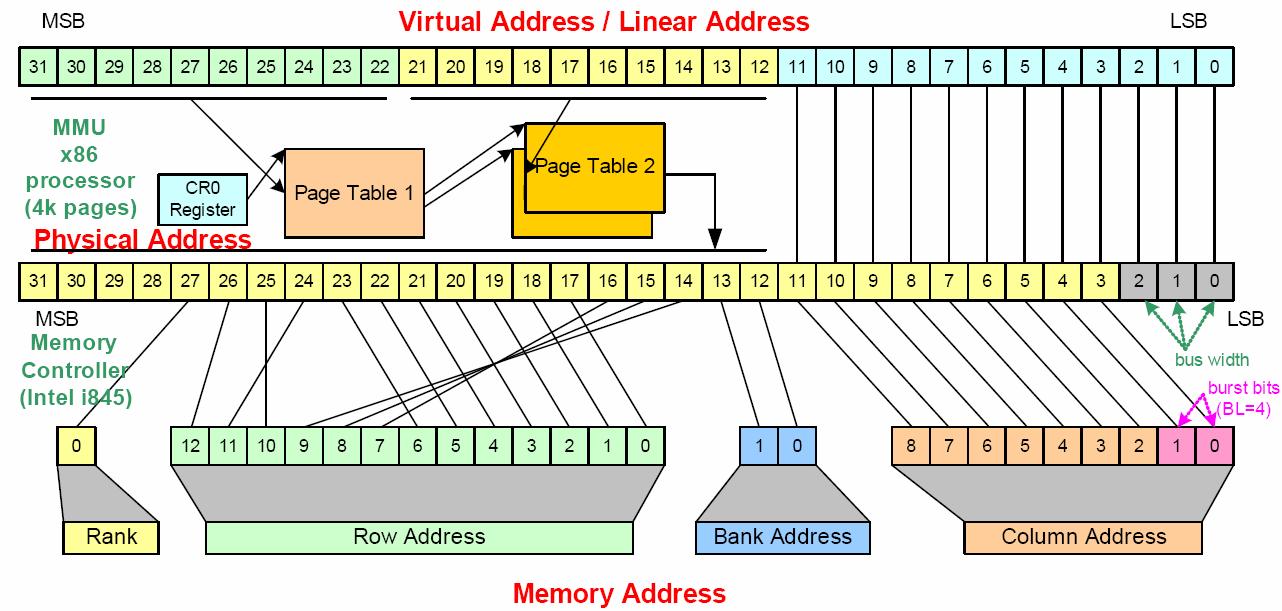


# Address Mapping



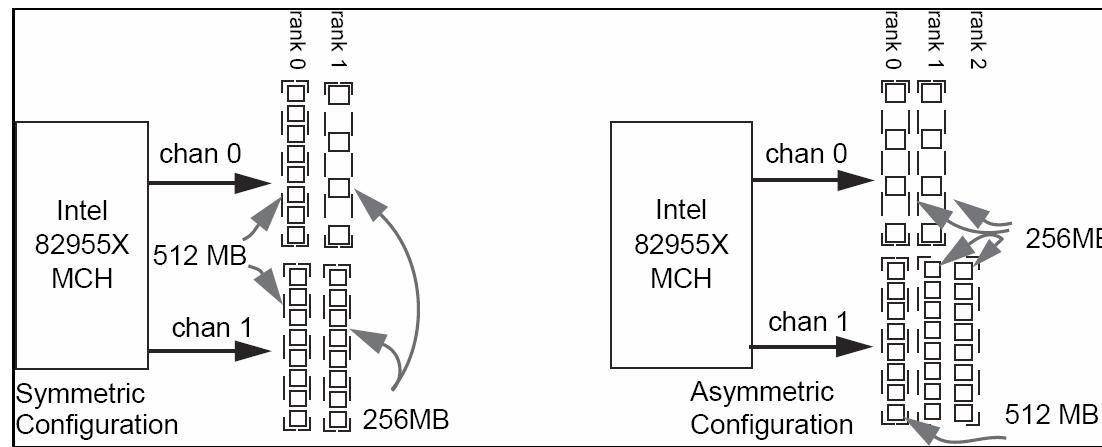
- Channel**  
Physical path between CPU and memory
- Rank**  
Group of DRAM chips operating in lockstep  
Same address, control, CS  
Responsible for subset of same “word”
- Bank**  
Set of independent memory arrays in DRAM chip
- Row/Column**  
Address of bit cell in a bank  
May be several “planes” to achieve n bits “wide”

# Address Mapping



[from Simon Albert, PhD thesis]

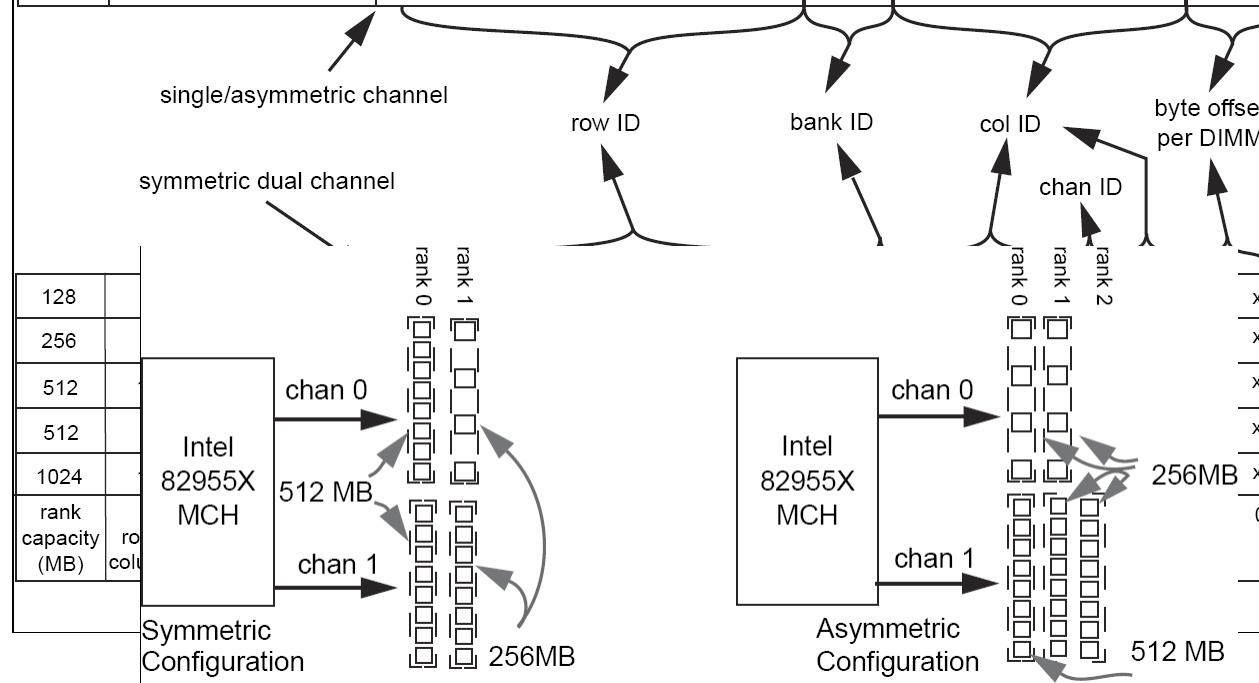
# Symmetric and Asymmetric Channels



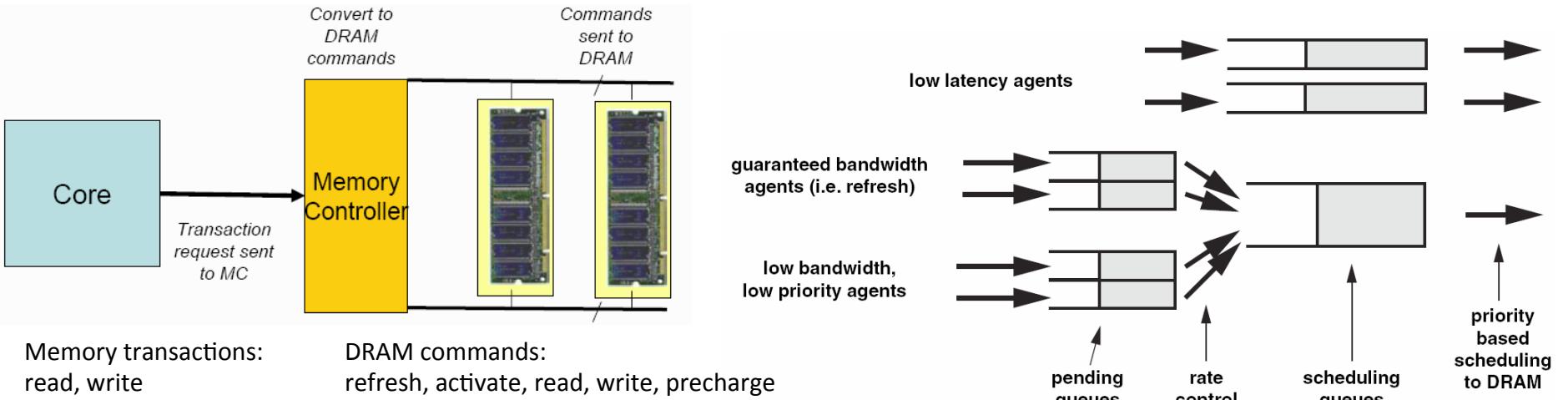
# Address Mapping

## Per channel, per-rank address mapping scheme for single/asymmetric channel mode

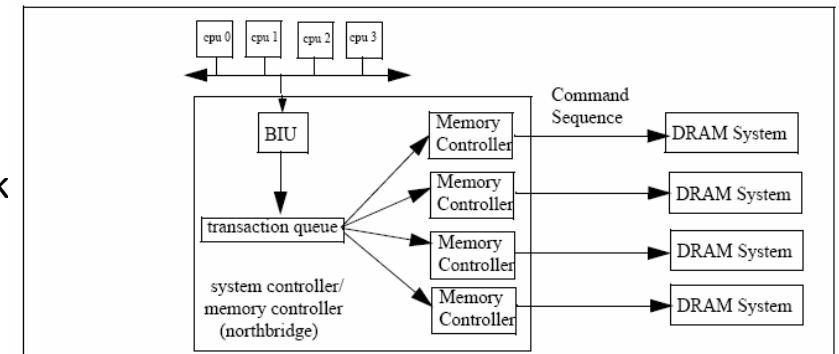
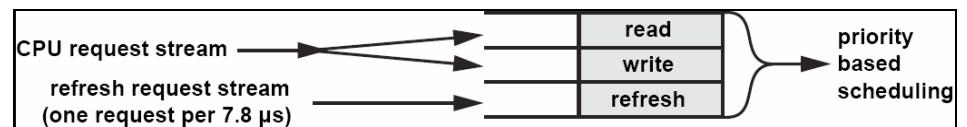
rank capacity (MB)	rank configuration row count x bank count x column count x column size	physical address																				x x x x								
		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0												x x x x								x x x x								
		10	9	8	7	6	5	4	3	2	1	0	11	12	0	1	8	7	6	5	4	3	2	1	0	x	x			
128	8192 x 4 x 512 x 8	10	9	8	7	6	5	4	3	2	1	0	11	12	0	1	8	7	6	5	4	3	2	1	0	x	x			
256	8192 x 4 x 1024 x 8	12	10	9	8	7	6	5	4	3	2	1	0	11	1	0	9	8	7	6	5	4	3	2	1	0	x	x		
512	16384 x 4 x 1024 x 8	13	12	10	9	8	7	6	5	4	3	2	1	0	11	1	0	9	8	7	6	5	4	3	2	1	0	x	x	
512	8192 x 8 x 1024 x 8	12	11	10	9	8	7	6	5	4	3	2	1	0	0	1	2	9	8	7	6	5	4	3	2	1	0	x	x	
1024	16384 x 8 x 1024 x 8	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	1	2	9	8	7	6	5	4	3	2	1	0	x	x



# Memory Scheduling

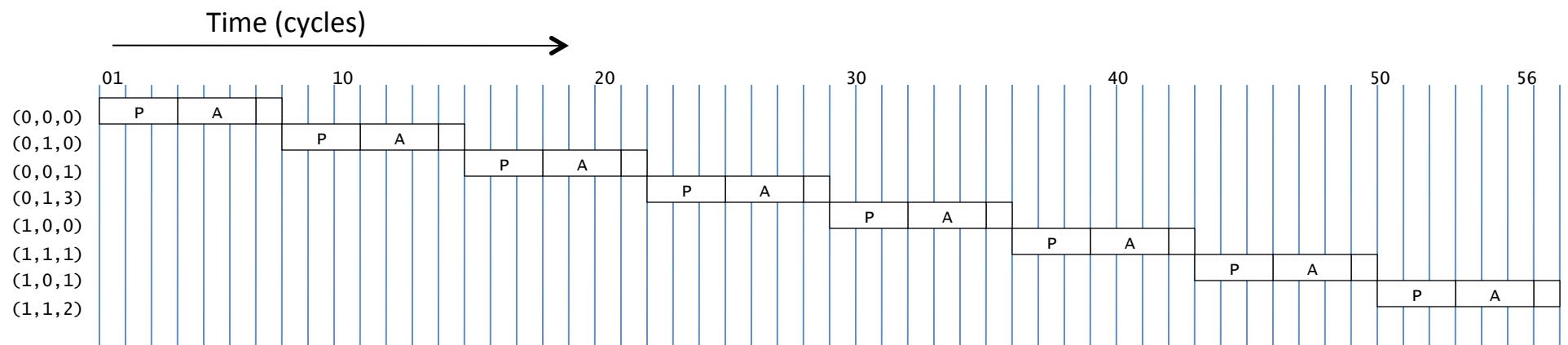


- **Memory scheduling policy**
  - Handle transaction requests
    - Possibly from different cores
  - Refresh
  - Prioritize low/high priority
    - CPU cache line fill request
    - Prefetch
  - Prioritize Read over Write
  - Re-order to take advantage of open page in bank
  - Page policy
    - Open Page
    - Close Page

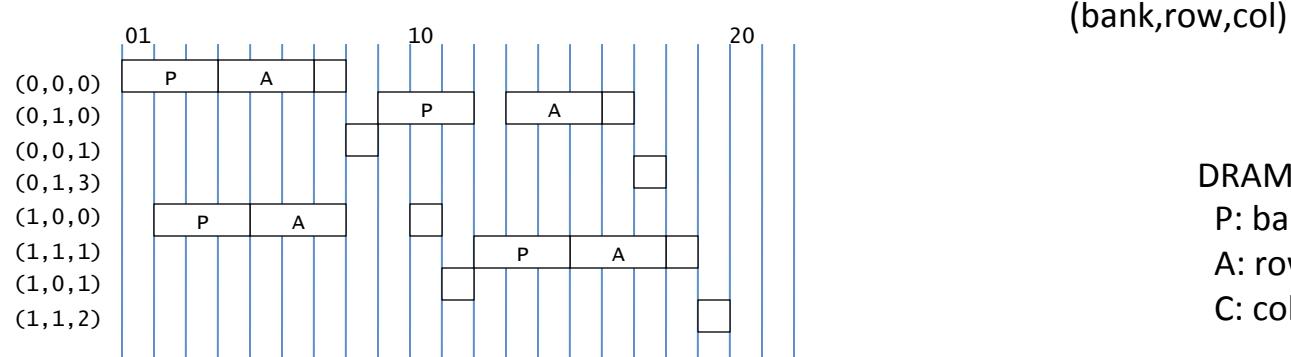


# Memory Scheduling

Without access scheduling (56 DRAM cycles)



With access scheduling (19 DRAM cycles)



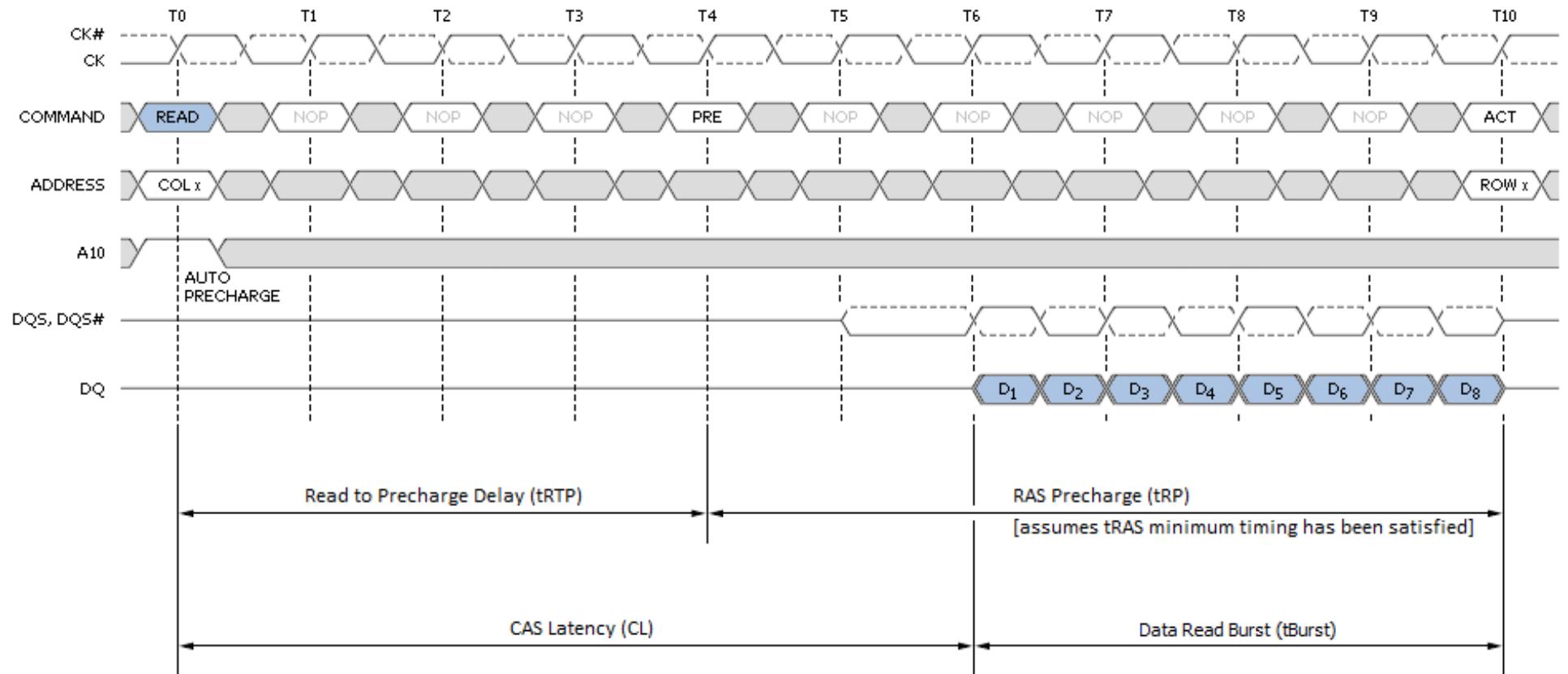
DRAM commands

P: bank precharge (3 cycles)

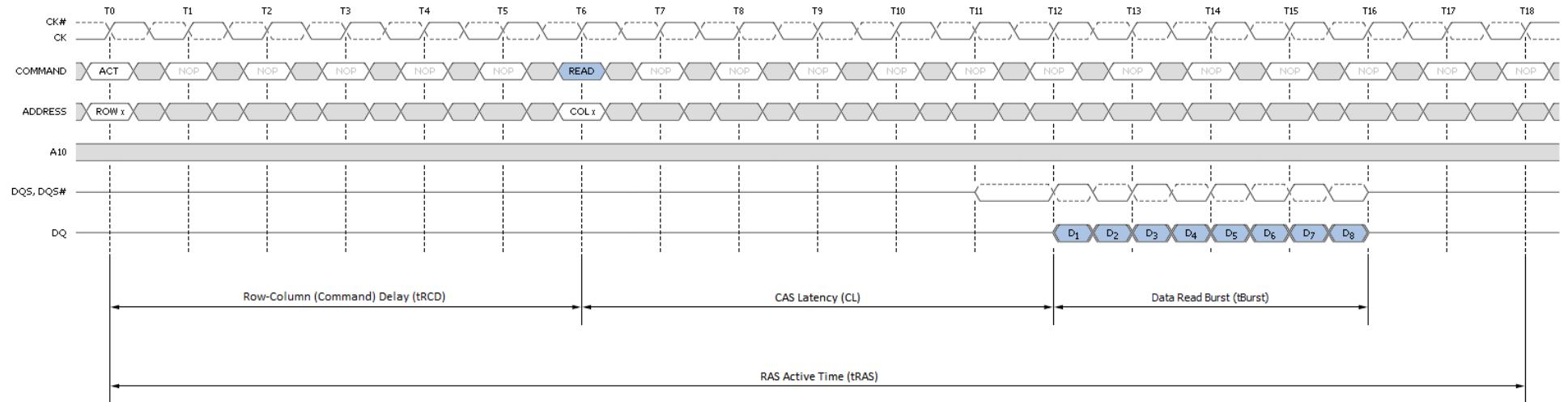
A: row activation (3 cycles)

C: column access (1 cycle)

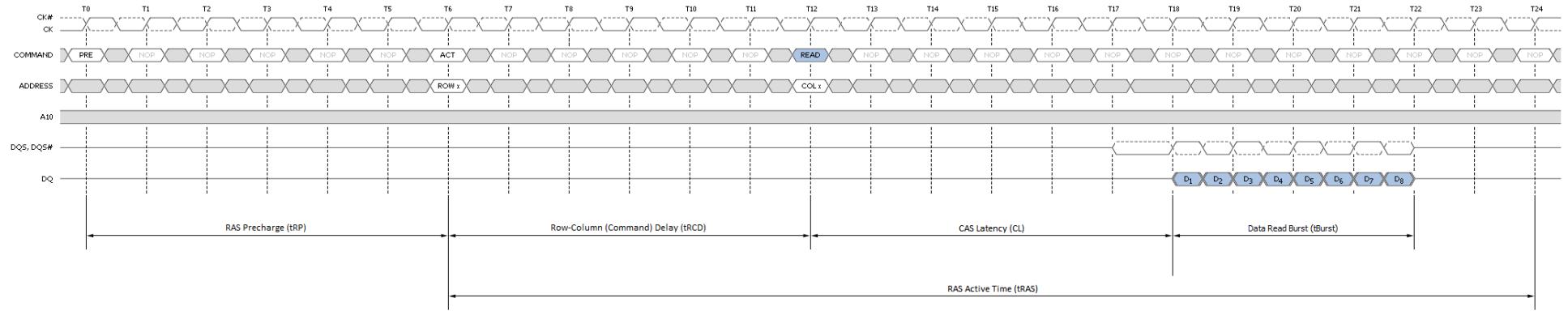
# Page Hit



# Page Empty



# Page Miss



Case 1: For an Idle Bank the read latency is tRCD + tCAS

Case 2: For an Active Bank the read latency can vary

- Minimum: tCAS for access to an active Page
- Maximum: tRP + tRCD + tCAS when the Page must be closed to open another

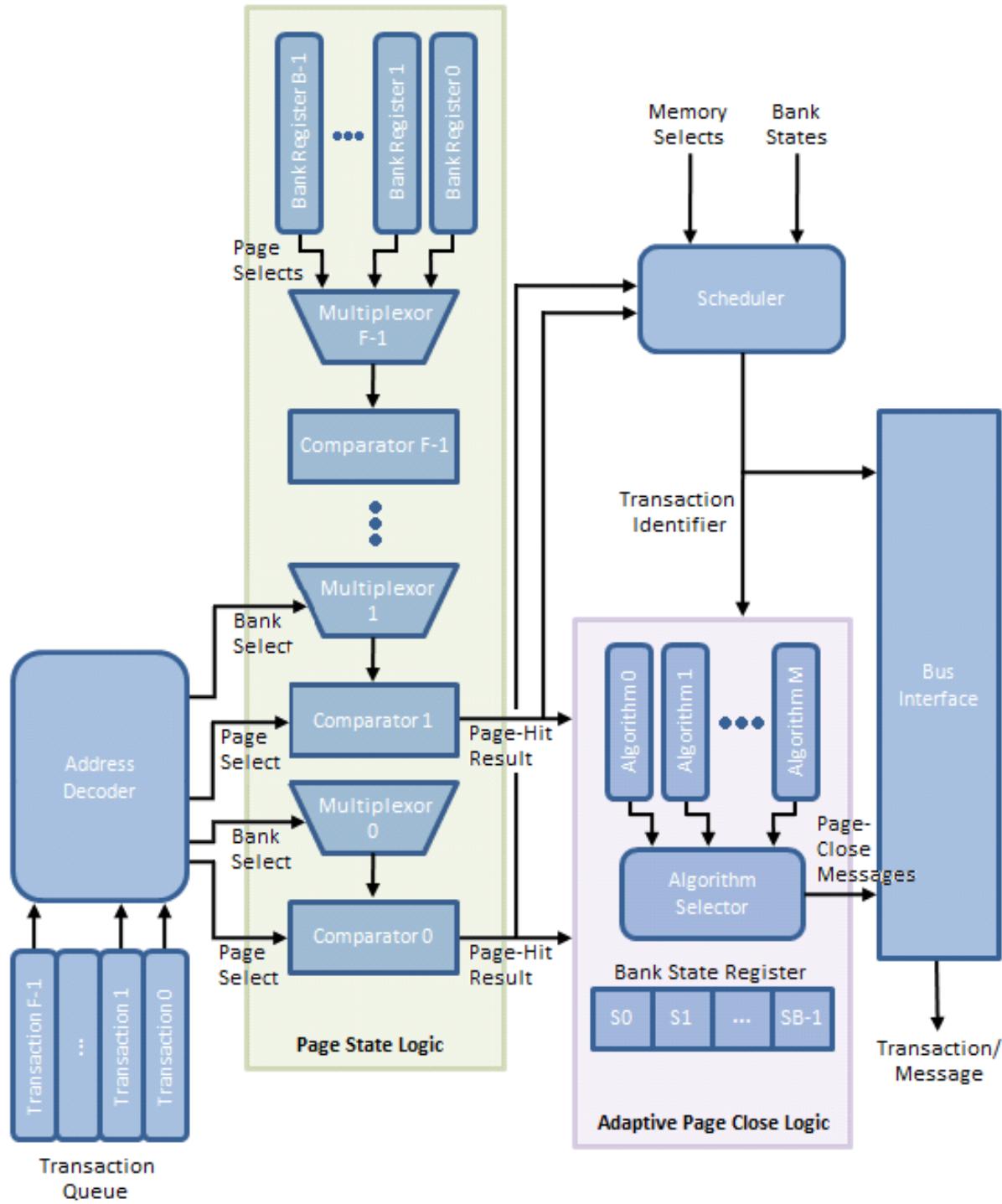
If  $n\%$  of accesses result in a hit to an open Page, then average read latency is:

$$n \times tCAS + (1-n) \times (tRP + tRCD + tCAS)$$

Then the break-even point for leaving the Page open or closing it would be:

$$tRCD + tCAS = n \times tCAS + (1-n) \times (tRP + tRCD + tCAS)$$

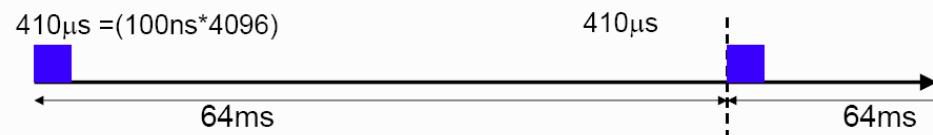
$$n = \frac{tRP}{tRP + tRCD}$$



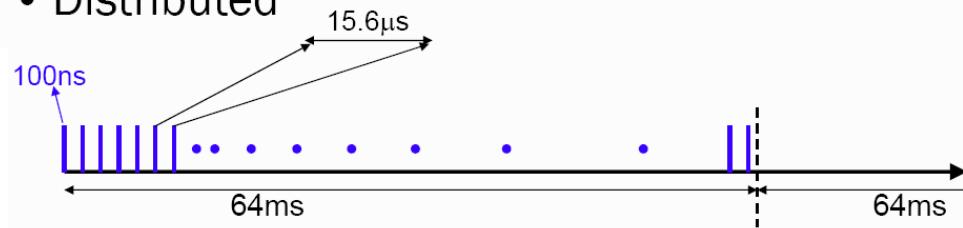
# Refresh Revisited

- Leaky storage
- Periodic Refresh across DRAM rows
- Un-accessible when refreshing
- Read, and write the same data back
- Example:
  - 4k rows in a DRAM
  - 100ns read cycle
  - Decay in 64ms

- Bursty

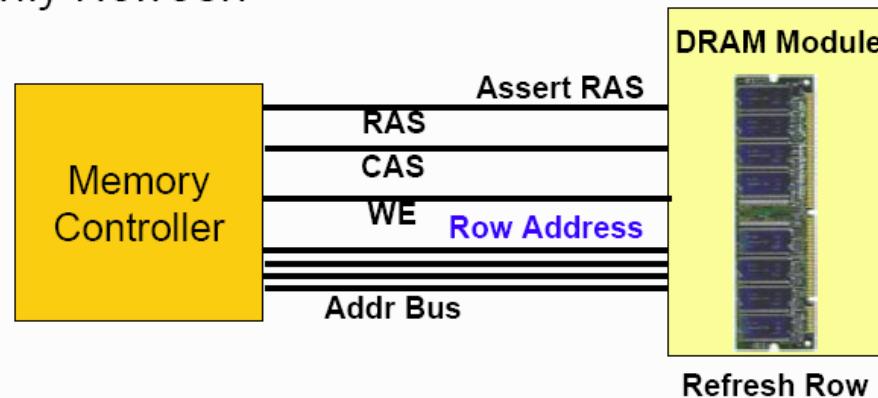


- Distributed

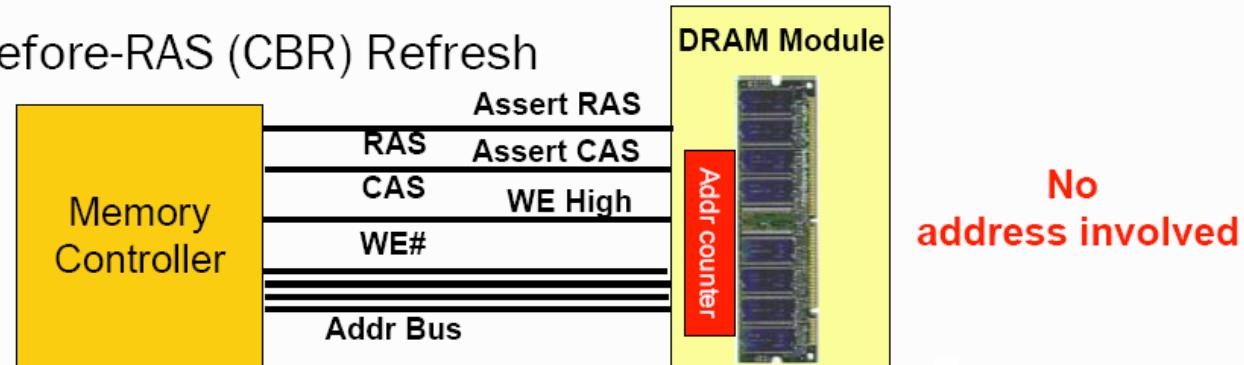


# Refresh

- RAS-Only Refresh



- CAS-Before-RAS (CBR) Refresh



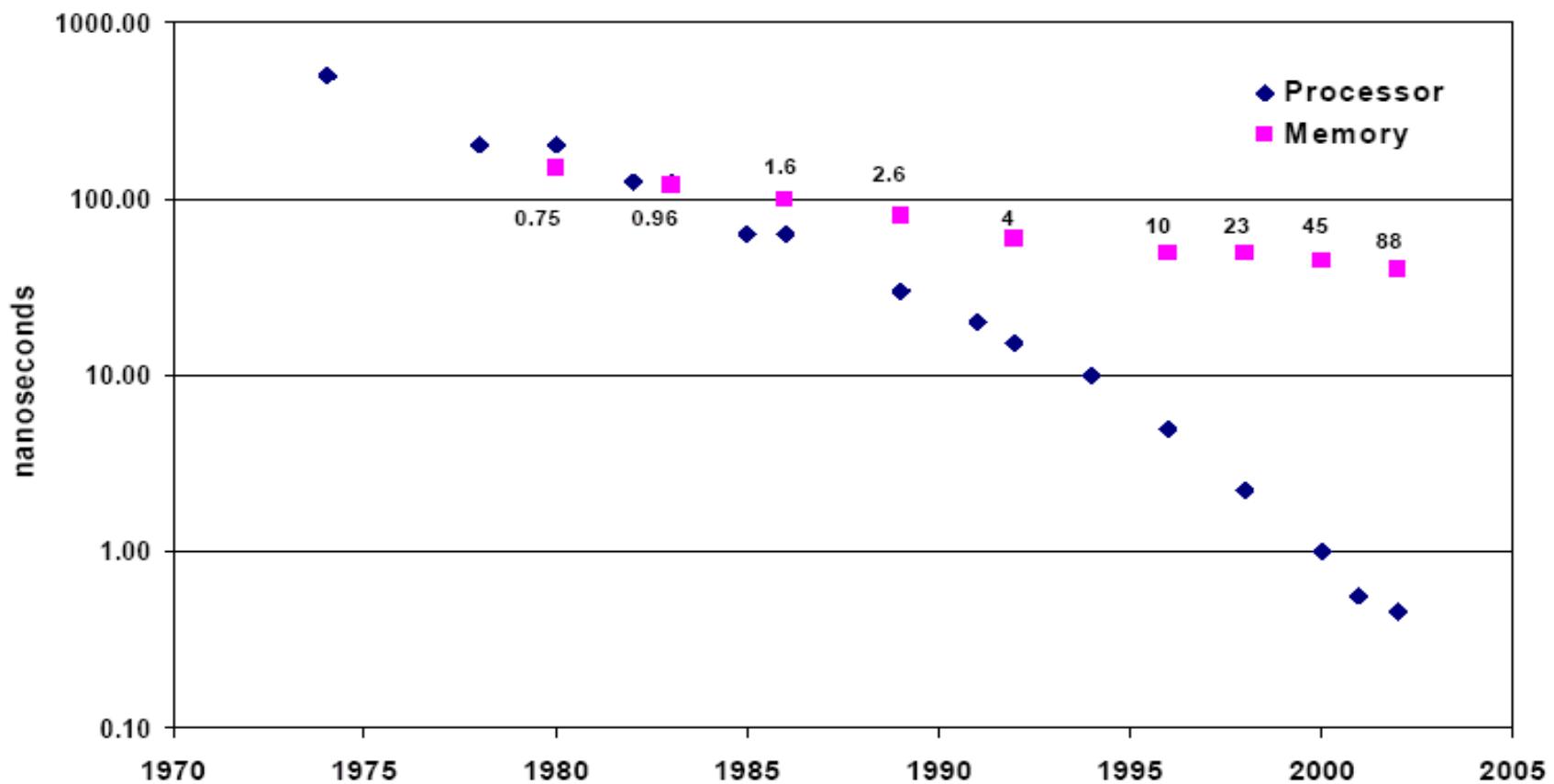
# Memory Technology Trends

Year of introduction	Chip size	Row access strobe (RAS)		Column access strobe (CAS)/data transfer time (ns)	Cycle time (ns)
		Slowest DRAM (ns)	Fastest DRAM (ns)		
1980	64K bit	180	150	75	250
1983	256K bit	150	120	50	220
1986	1M bit	120	100	25	190
1989	4M bit	100	80	20	165
1992	16M bit	80	60	15	120
1996	64M bit	70	50	12	110
1998	128M bit	70	50	10	100
2000	256M bit	65	45	7	90
2002	512M bit	60	40	5	80
2004	1G bit	55	35	5	70
2006	2G bit	50	30	2.5	60

**Figure 5.13 Times of fast and slow DRAMs with each generation.** (Cycle time is defined on page 310.) Performance improvement of row access time is about 5% per year. The improvement by a factor of 2 in column access in 1986 accompanied the switch from NMOS DRAMs to CMOS DRAMs.

From Hennessy & Patterson, Computer Architecture: A Quantitative Approach (4<sup>th</sup> edition)

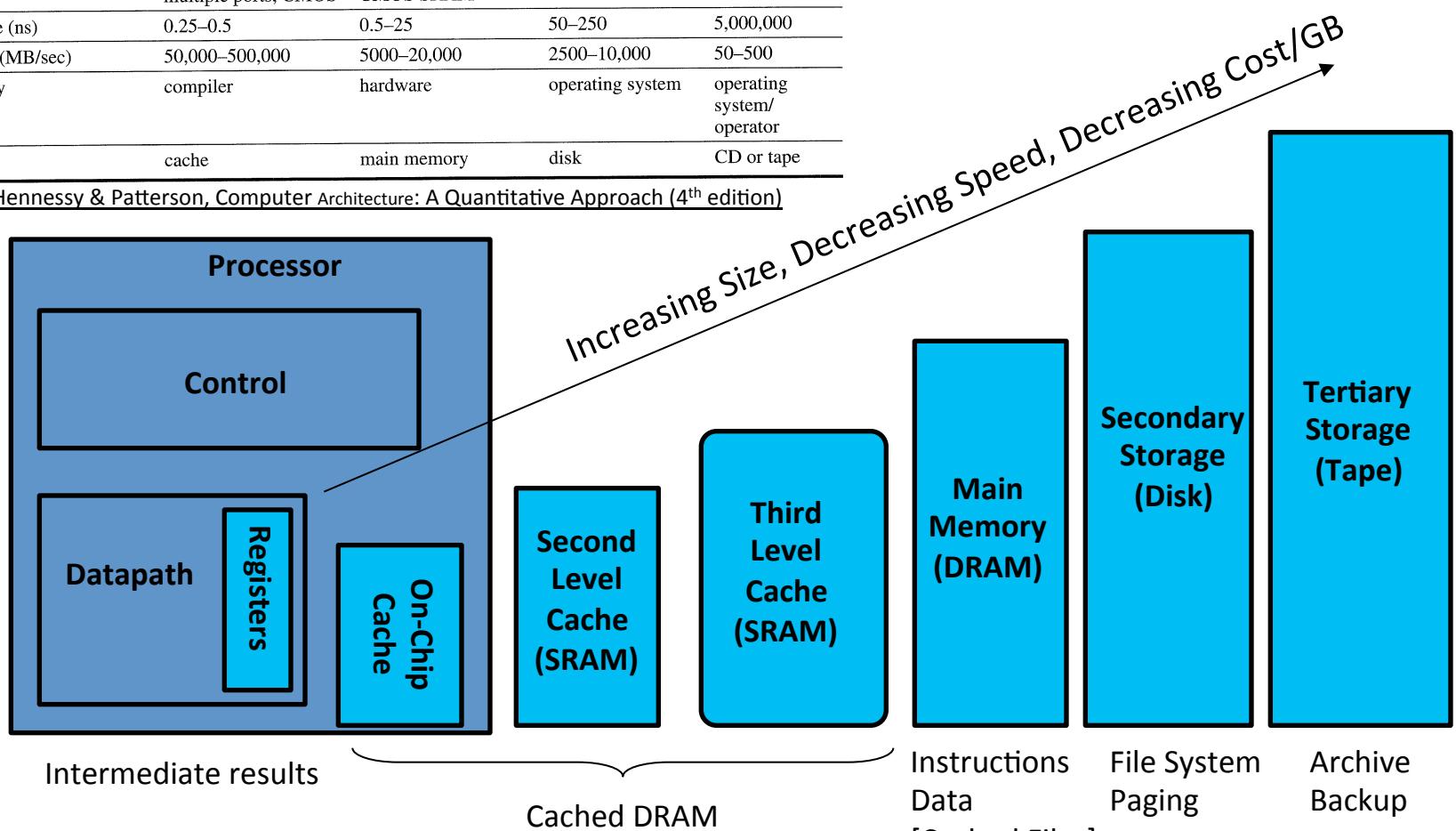
# Processor/Memory Gap



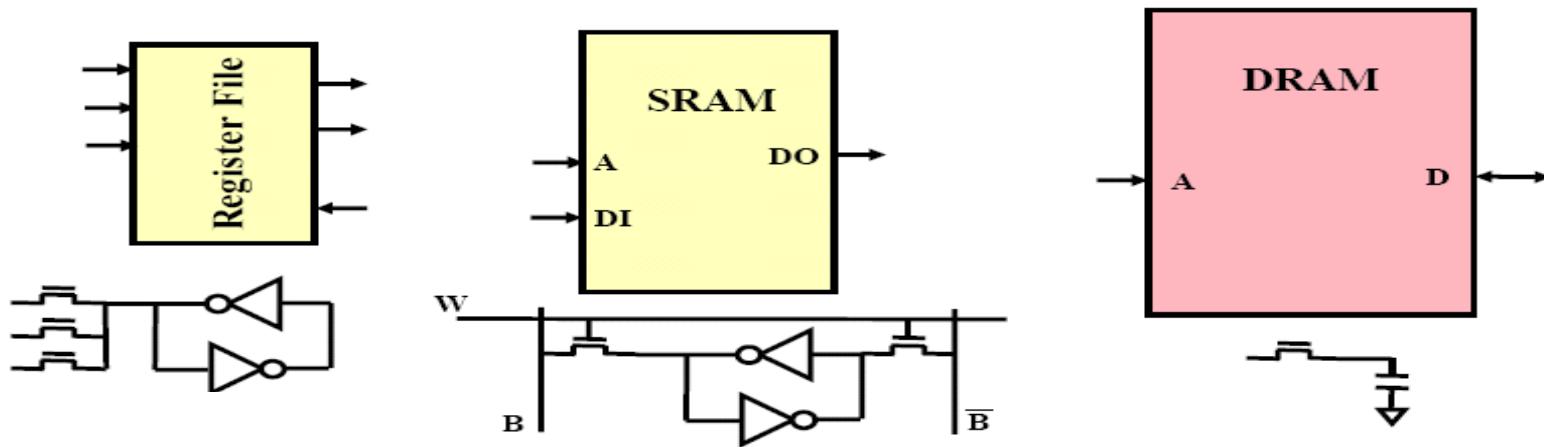
# Computer Memory Hierarchy

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	< 16 MB	< 512 GB	> 1 TB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25–0.5	0.5–25	50–250	5,000,000
Bandwidth (MB/sec)	50,000–500,000	5000–20,000	2500–10,000	50–500
Managed by	compiler	hardware	operating system	operating system/operator
Backed by	cache	main memory	disk	CD or tape

From Hennessy & Patterson, Computer Architecture: A Quantitative Approach (4<sup>th</sup> edition)



# Registers, SRAM, DRAM

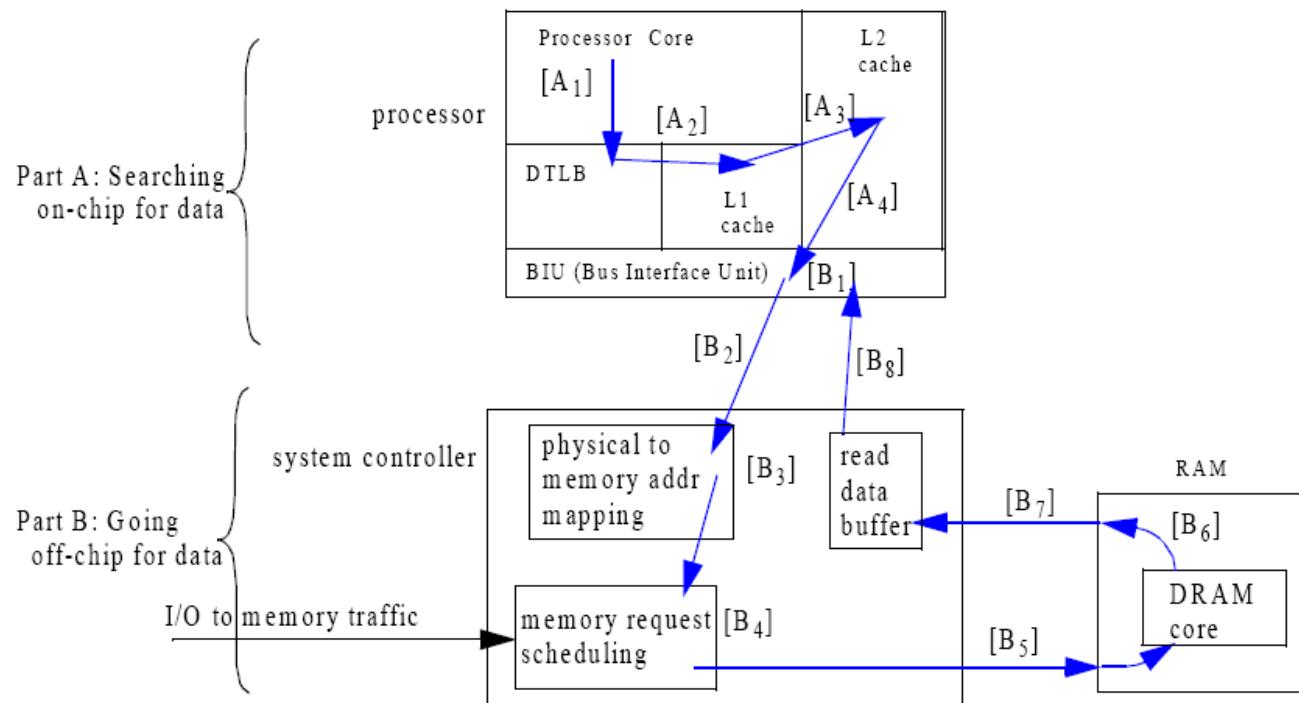


- Simple Interface
- “At speed” access (CPU)
- Multi-ported
- Simple interface
- Moderate density
- Moderate cost/bit
- Single or double ported
- Primary design goal: speed
- Can be integrated with logic
- Complex interface
- Multiple clock cycles to access
- Burst/page modes
- Very high density
- Low cost/bit
- Primary design goals: density, \$
- Usually single ported
- Specialized fab process
  - Rarely integrated with logic

# Intel Pentium 4 3.2 GHz Server

Component	Access Speed (Time for data to be returned)
Registers	1 cycle = 0.3 nanoseconds
L1 Cache	3 cycles = 1 nanoseconds
L2 Cache	20 cycles = 7 nanoseconds
L3 Cache	40 cycles = 13 nanoseconds
Memory	300 cycles = 100 nanoseconds

# Putting it all together...



- Improving Bandwidth
  - Wider memory access/bus
  - Banks & interleaving
  - Page/Burst Modes
- Improving Latency
  - Remove redundant steps
  - Integration
    - Refresh row counter in DRAM chip
    - Memory controller on processor chip
  - Caching

# Why Spend So Much Time on Memory?

- Huge impact on computing performance (and increasingly, power consumption)
- Perhaps no other single technology has impacted the evolution of PC architecture as much
  - Caches
  - Microprocessor architecture (pre-fetch...)
  - Bus Width, Speed
  - Chipsets
- PCs aren't the only high performance application for memory system design (<50% Q2 2012)
  - Embedded Systems
  - Video/Graphics/Game Processors
  - Digital Signal Processing (DSP)
  - Automated Test Equipment (ATE)

# Acronyms and Definitions

RAM – random access memory

ROM – read only memory

PROM – programmable read only memory

EPROM – erasable PROM

EEPROM/E<sup>2</sup>PROM – electrically erasable PROM

CAM – content addressable memory

DRAM – dynamic RAM (requires refresh)

SRAM – static RAM (no refresh)

SDRAM – synchronous DRAM

NVRAM – non-volatile RAM (often RAM with battery backup)

SDR SDRAM – single data rate SDRAM

DDR SDRAM – double data rate SDRAM

RDRAM – RAMBUS DRAM

ECC – Error Correction Codes

DIMM – Dual Inline Memory Module

# Acronyms and Definitions

Rank – Group of memory chips with same control signals (each chip typically a subset of the bits that comprise a memory word)

Bank – Portion of DRAM chip with own sense amplifiers, open page