

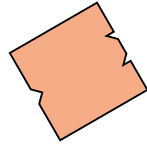
DRAM: Architectures, Interfaces, and Systems

A Tutorial

Bruce Jacob and David Wang

Electrical & Computer Engineering Dept.
University of Maryland at College Park

<http://www.ece.umd.edu/~blj/DRAM/>



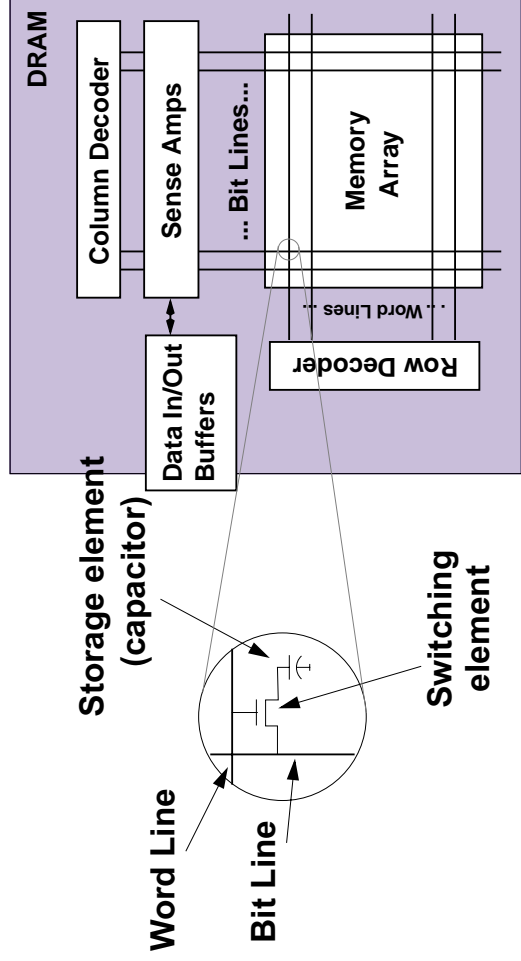
Outline

- Basics
- DRAM Evolution: *Structural Path*
- Advanced Basics
- DRAM Evolution: *Interface Path*
- Future Interface Trends & Research Areas
- Performance Modeling:
Architectures, Systems, Embedded

Break at 10 a.m. — Stop us or starve

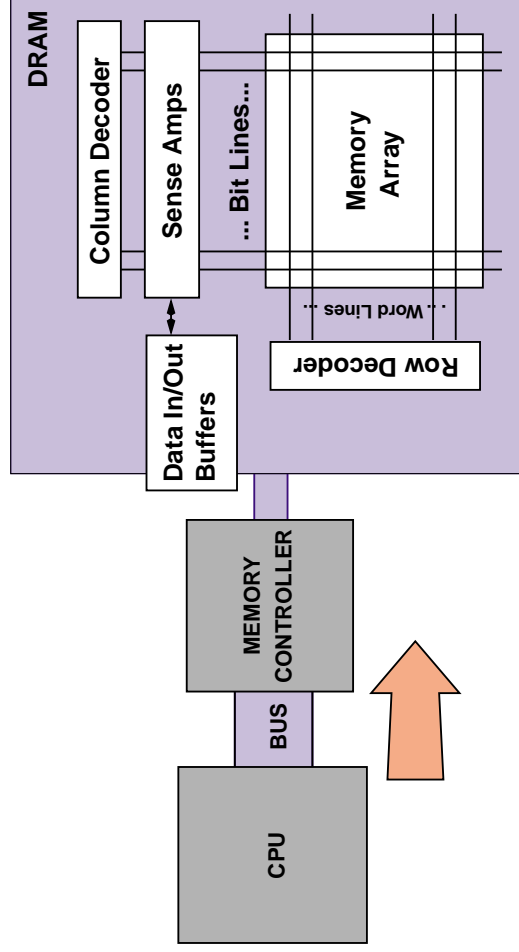
Basics

DRAM ORGANIZATION



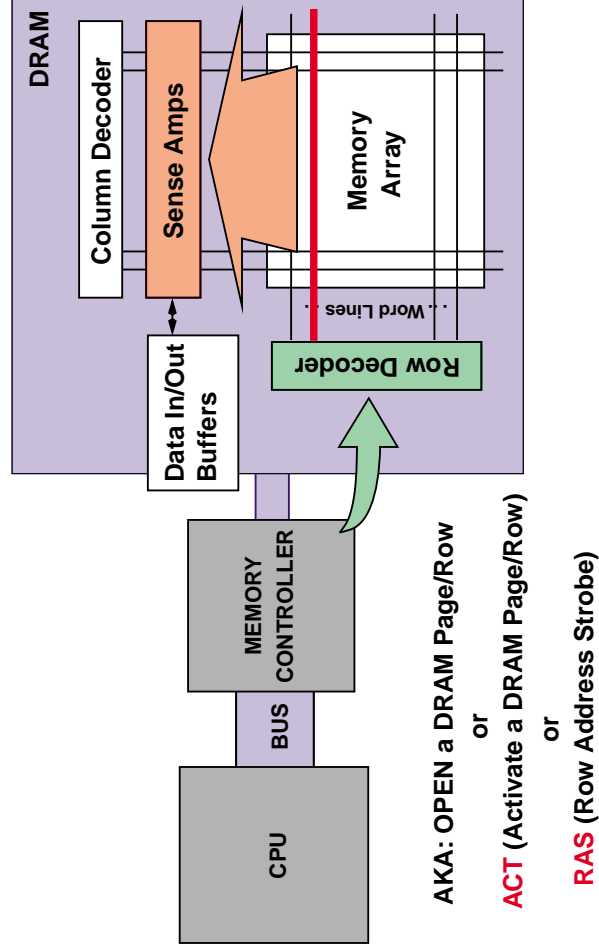
Basics

BUS TRANSMISSION



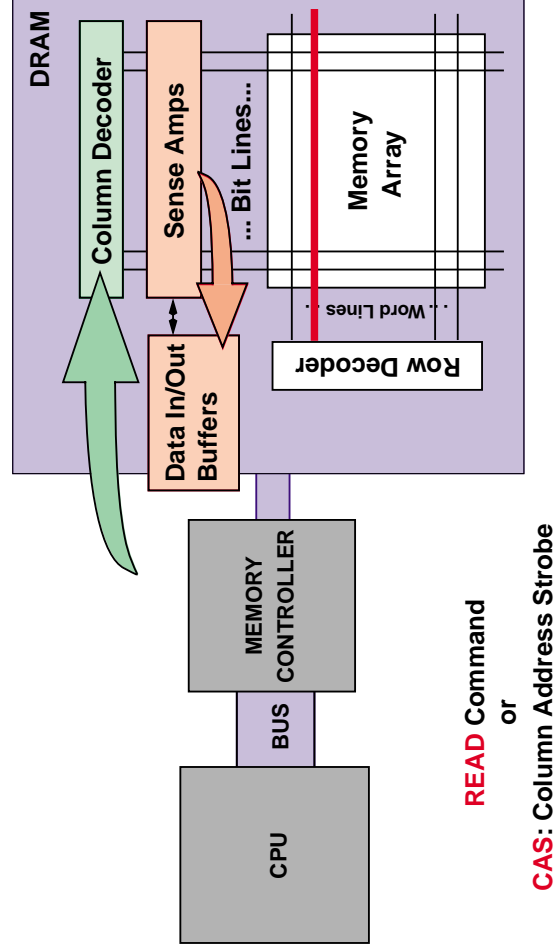
Basics

[PRECHARGE and] ROW ACCESS



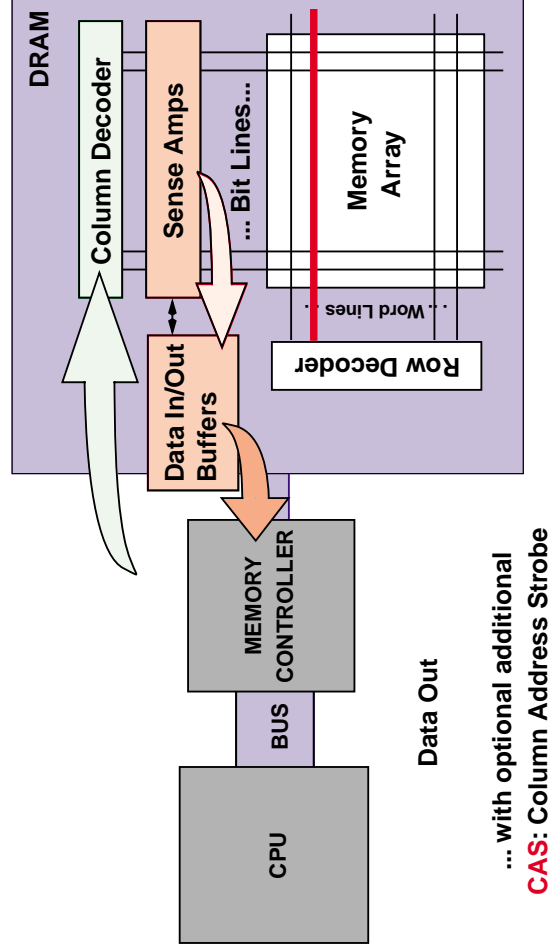
Basics

COLUMN ACCESS



Basics

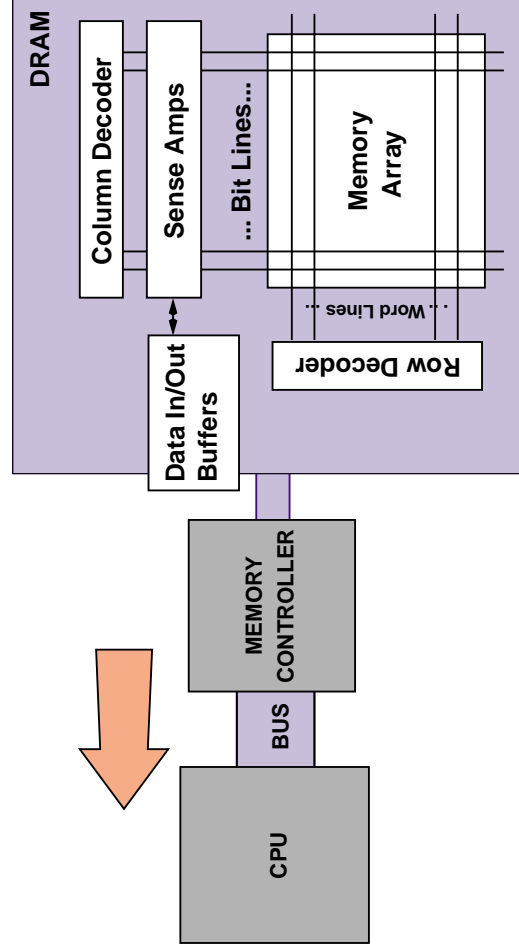
DATA TRANSFER



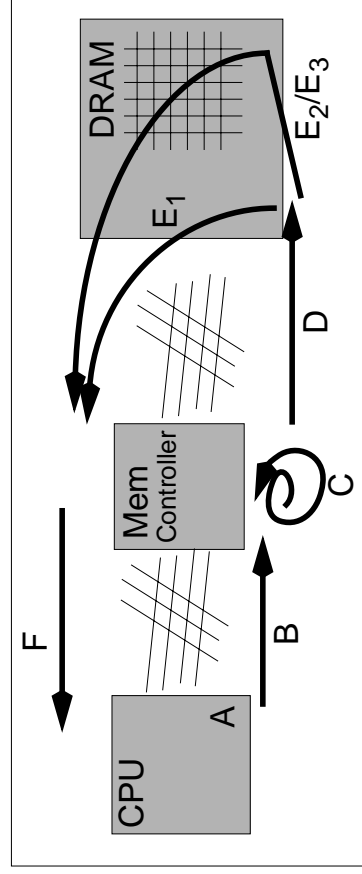
note: page mode enables overlap with CAS

Basics

BUS TRANSMISSION



Basics



A: Transaction request may be delayed in Queue
 B: Transaction request sent to Memory Controller
 C: Transaction converted to Command Sequences
 (may be queued)

D: Command/s Sent to DRAM

E₁: Requires only a **CAS** or

E₂: Requires **RAS + CAS** or

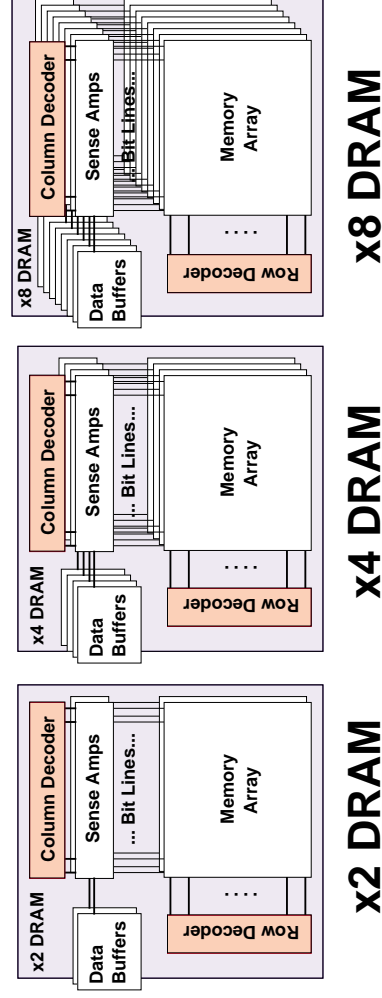
E₃: Requires **PRE + RAS + CAS**

F: Transaction sent back to CPU

“DRAM Latency” = A + B + C + D + E + F

Basics

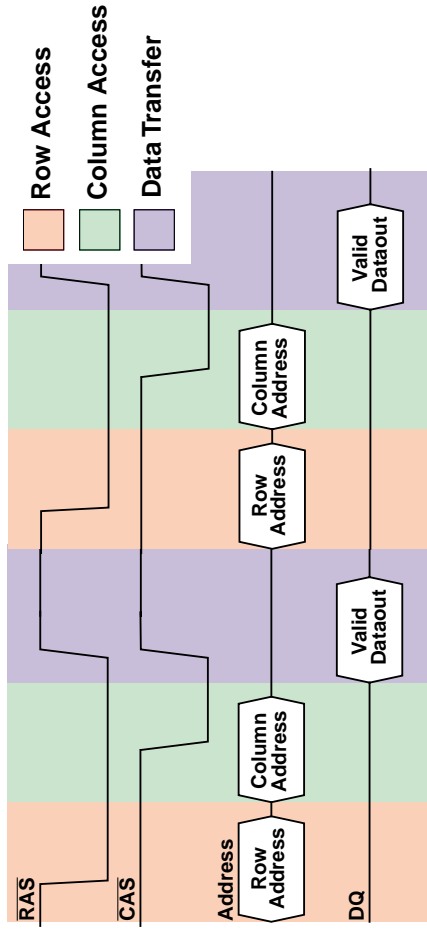
PHYSICAL ORGANIZATION



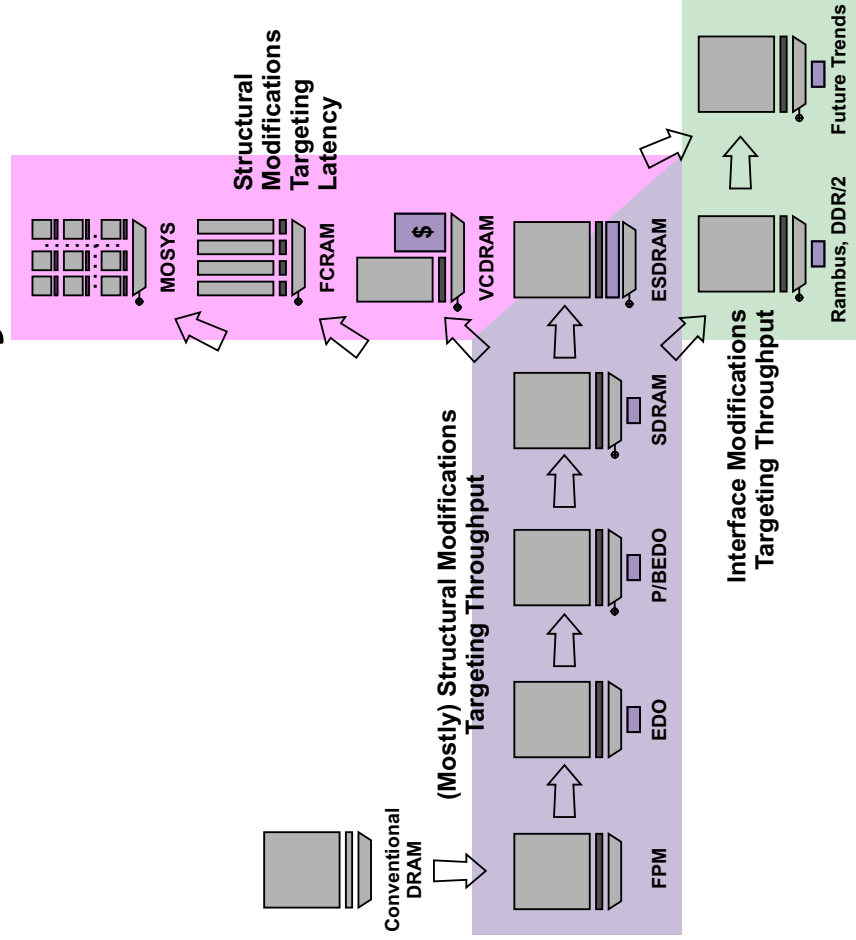
This is per bank ...
 Typical DRAMs have 2+ banks

Basics

Read Timing for Conventional DRAM

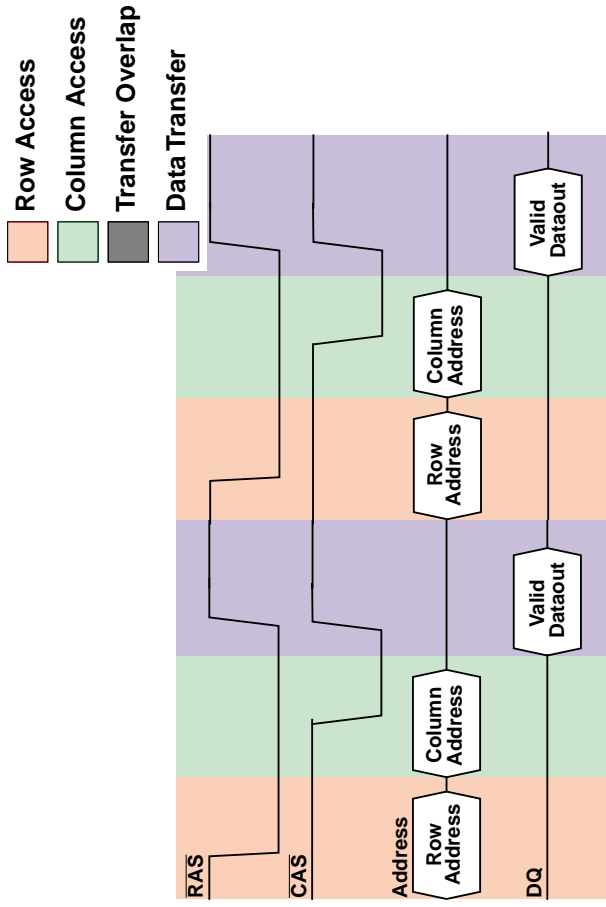


DRAM Evolutionary Tree



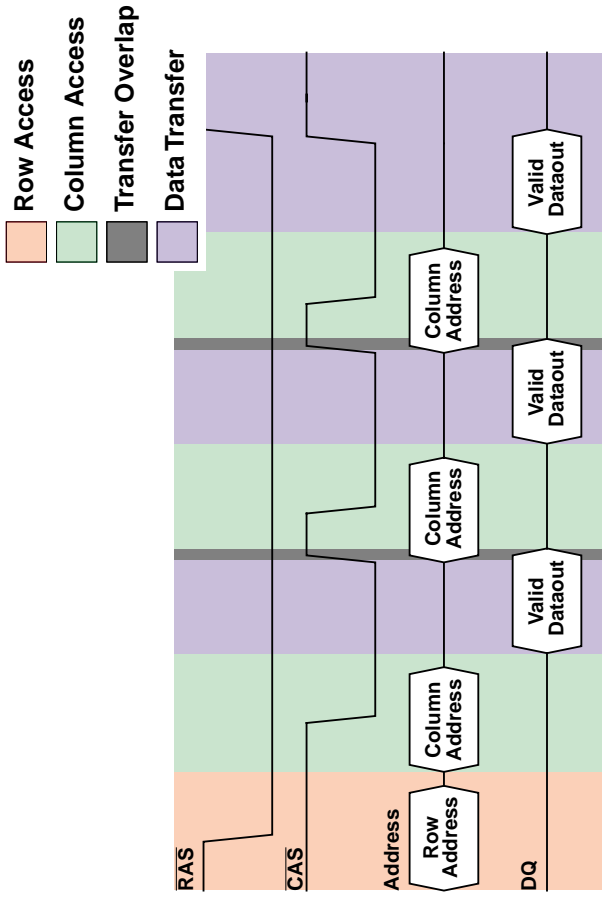
DRAM Evolution

Read Timing for Conventional DRAM



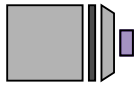
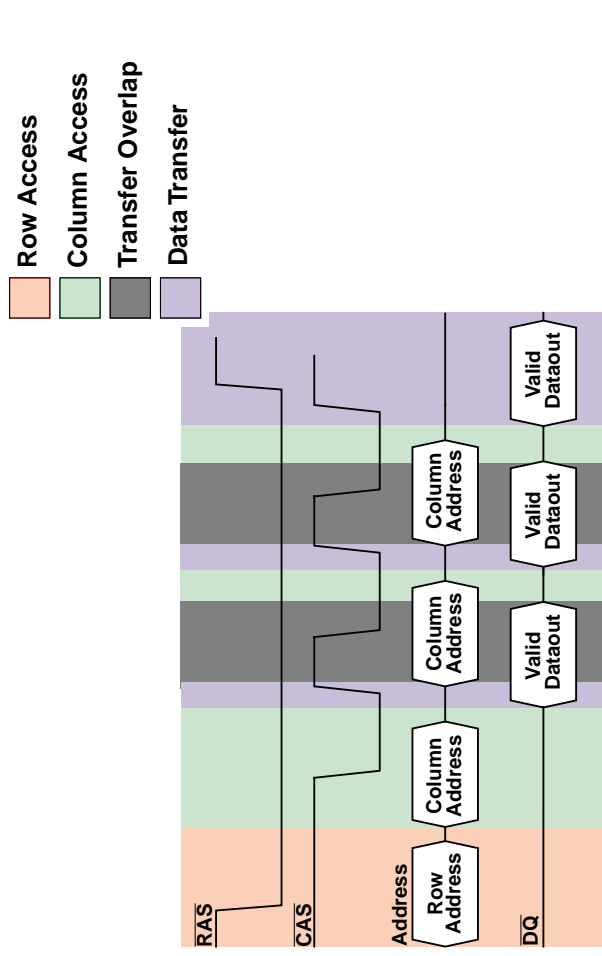
DRAM Evolution

Read Timing for Fast Page Mode



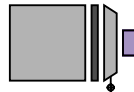
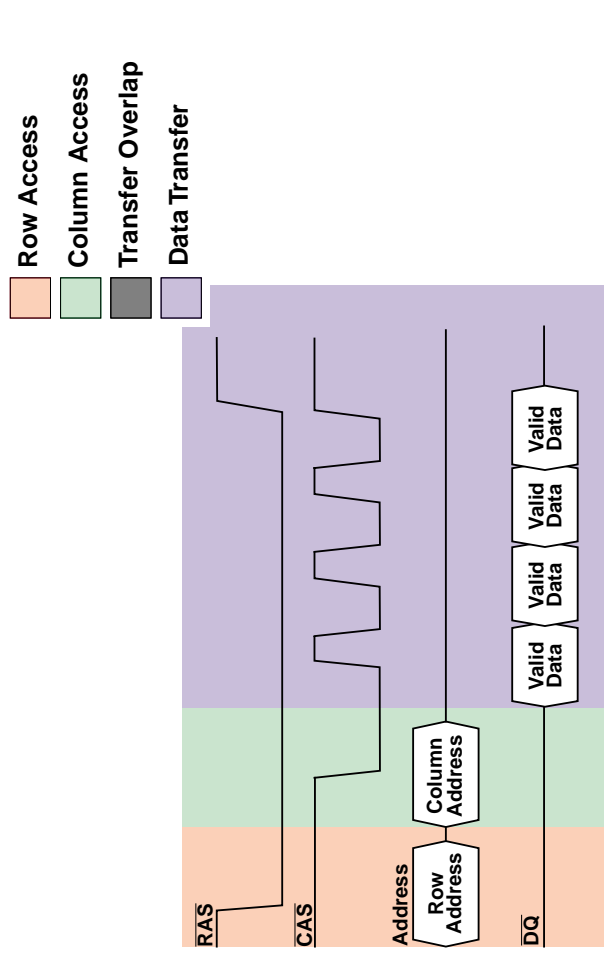
DRAM Evolution

Read Timing for Extended Data Out



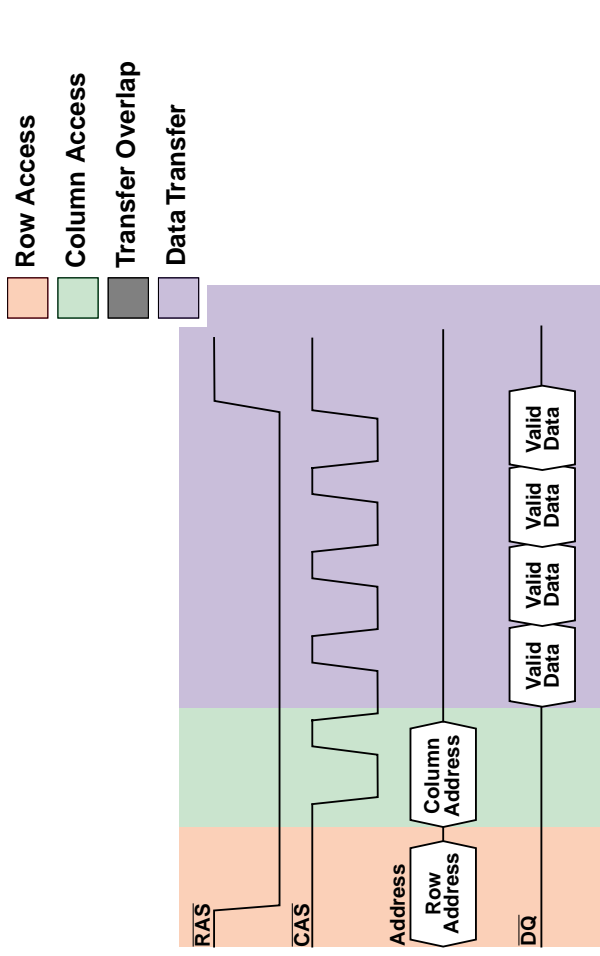
DRAM Evolution

Read Timing for Burst EDO



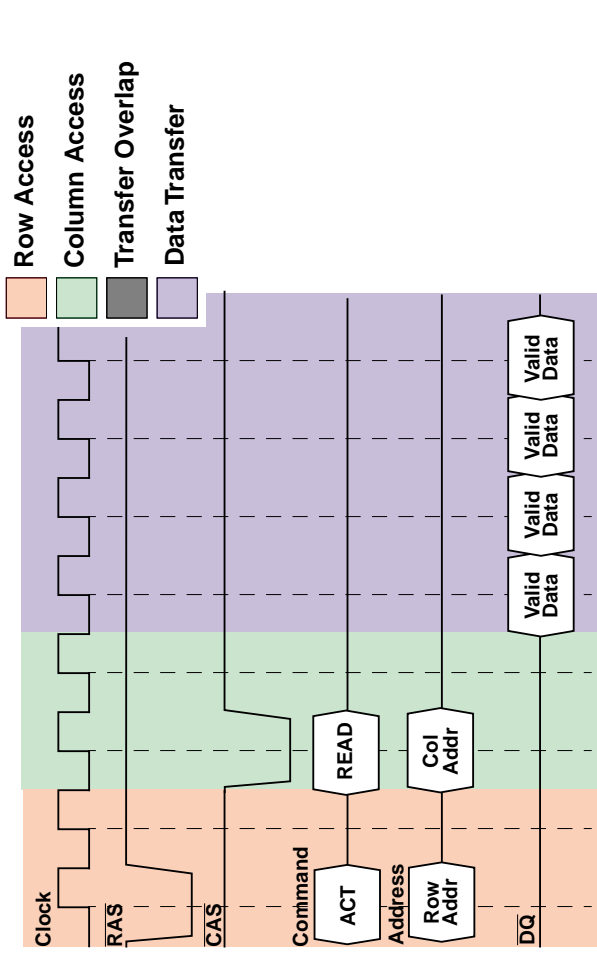
DRAM Evolution

Read Timing for Pipeline Burst EDO



DRAM Evolution

Read Timing for Synchronous DRAM



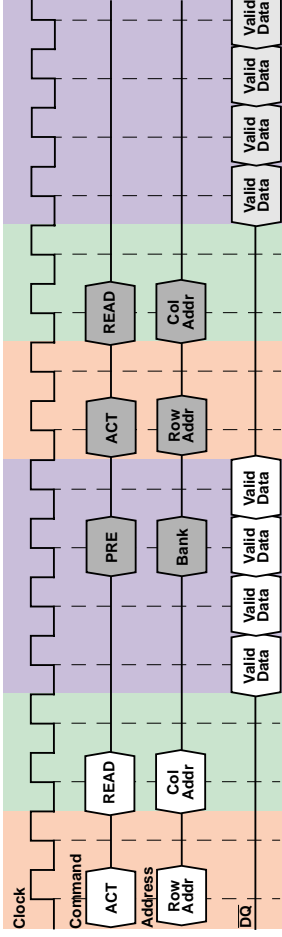
(RAS + CAS + OE ... == Command Bus)

DRAM Evolution

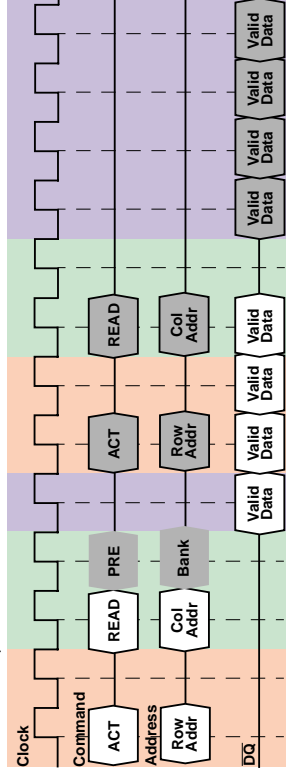


Inter-Row Read Timing for ESDRAM

Regular CAS-2 SDRAM, R/R to same bank



ESDRAM, R/R to same bank

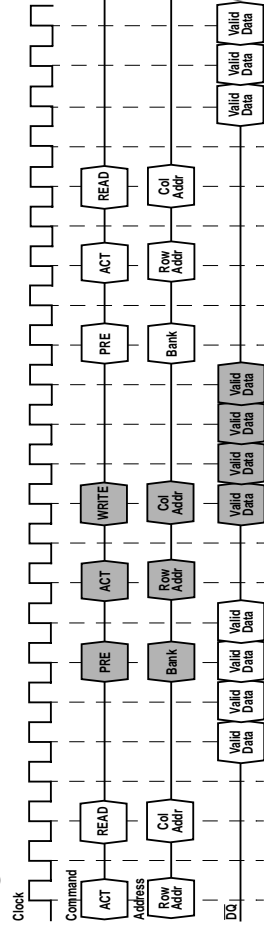


DRAM Evolution

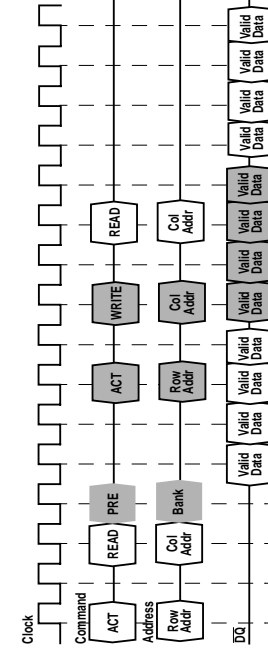


Write-Around in ESDRAM

Regular CAS-2 SDRAM, R/W/R to same bank, rows 0/1/0



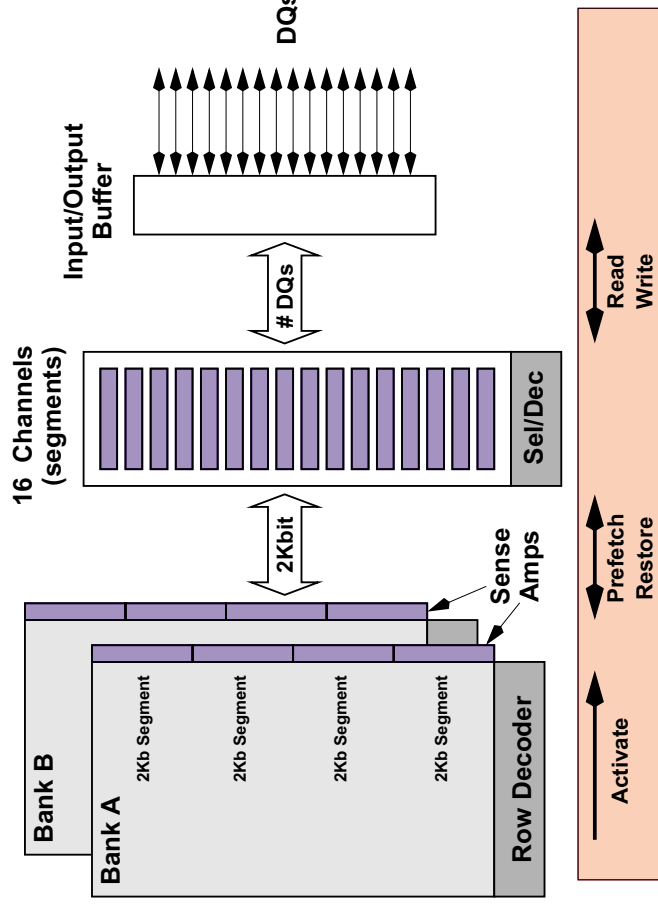
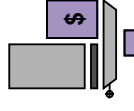
ESDRAM, R/W/R to same bank, rows 0/1/0



(can second READ be this aggressive?)

DRAM Evolution

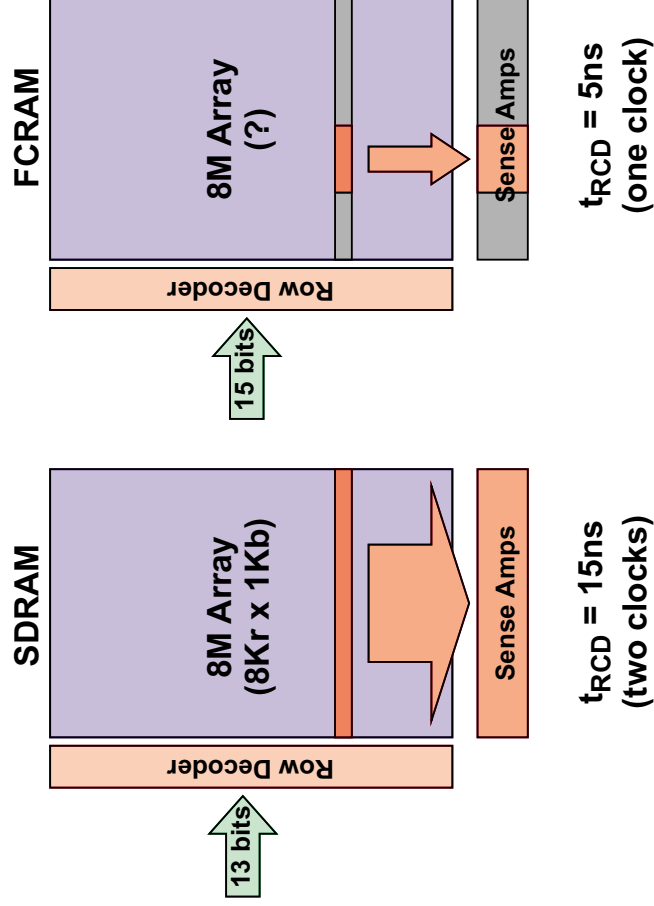
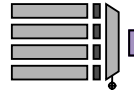
Internal Structure of Virtual Channel



Segment cache is software-managed, reduces energy

DRAM Evolution

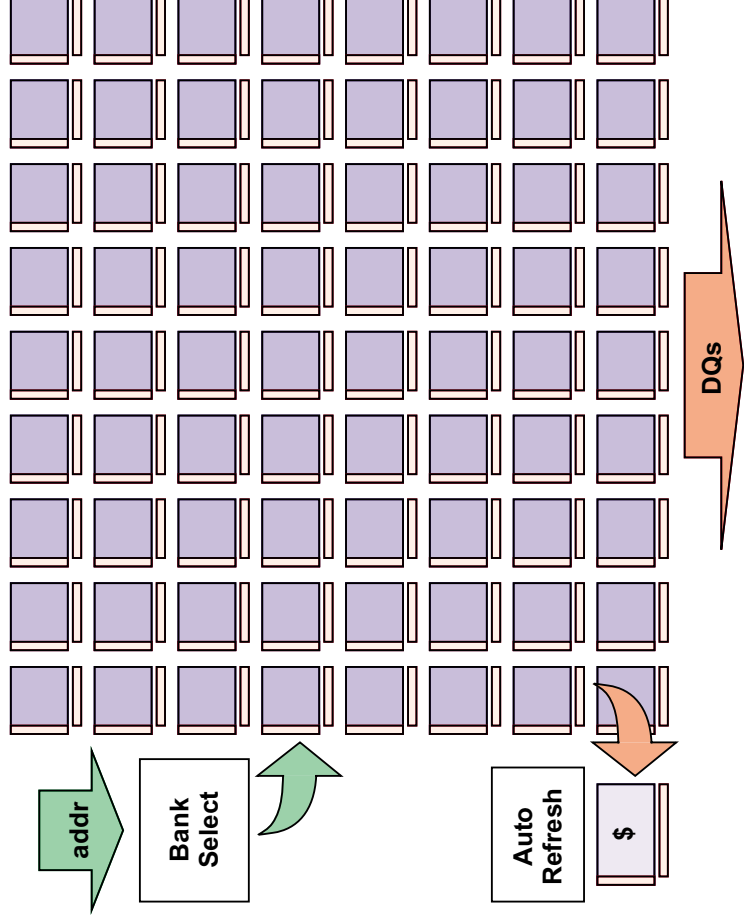
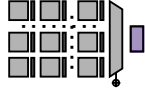
Internal Structure of Fast Cycle RAM



Reduces access time and energy/access

DRAM Evolution

Internal Structure of MoSys 1T-SRAM



DRAM Evolution

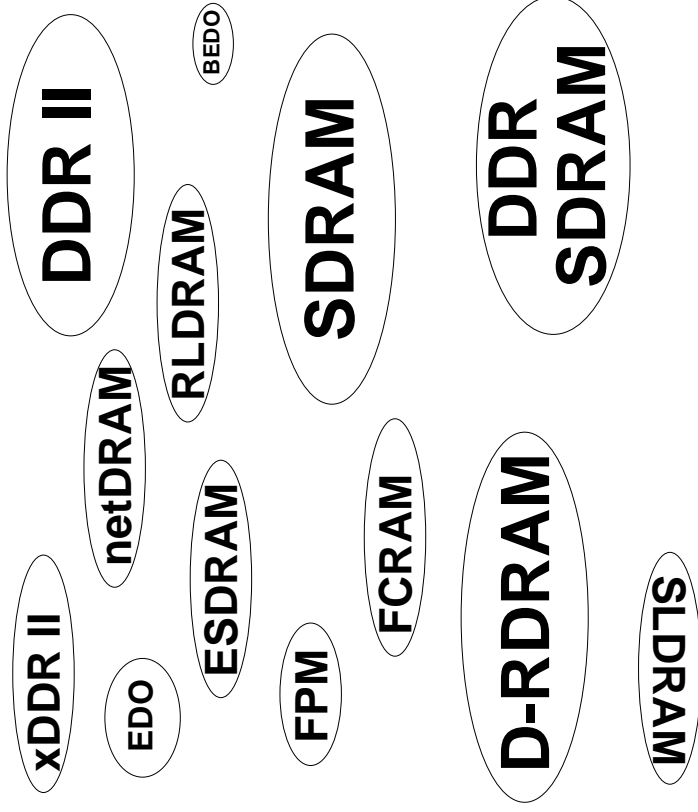
Comparison of Low-Latency DRAM Cores

DRAM Type	Data Bus Speed	Bus Width (per chip)	Peak BW (per Chip)	RAS-CAS (t_{rCD})	RAS-DQ (t_{rAC})
PC133 SDRAM	133	16	266 MB/s	15 ns	30 ns
VCDRAM	133	16	266 MB/s	30 ns	45 ns
FCRAM	200 * 2	16	800 MB/s	5 ns	22 ns
1T-SRAM	200	32	800 MB/s	—	10 ns
DDR 266	133 * 2	16	532 MB/s	20 ns	45 ns
DRDRAM	400 * 2	16	1.6 GB/s	22.5 ns	60 ns
RLDRAM	300 * 2	32	2.4 GB/s	???	25 ns

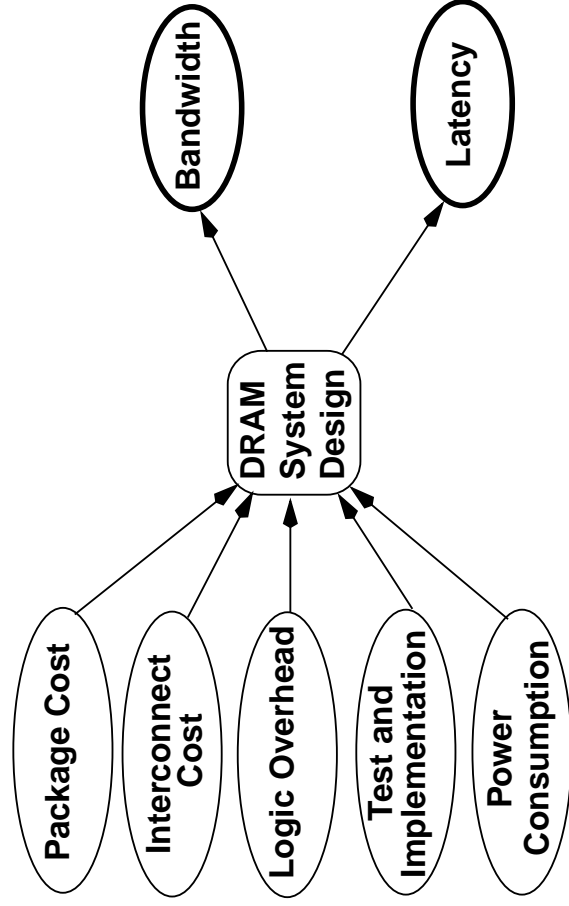
Outline

- **Basics**
- **DRAM Evolution: Structural Path**
- **Advanced Basics**
- **Memory System Details (Lots)**
- **DRAM Evolution: Interface Path**
- **Future Interface Trends & Research Areas**
- **Performance Modeling: Architectures, Systems, Embedded**

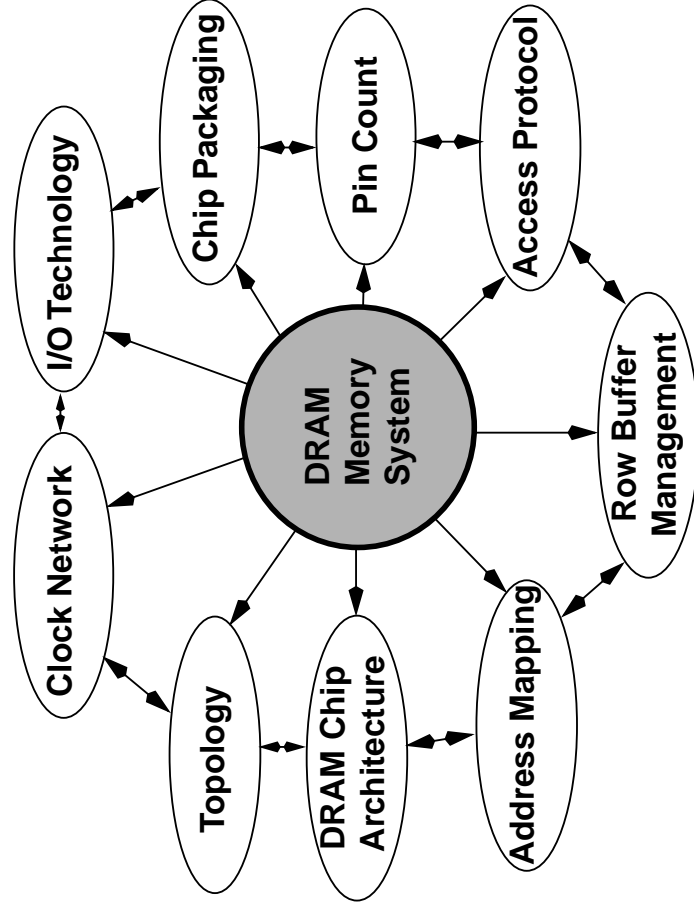
What Does This All Mean?



Cost - Benefit Criterion

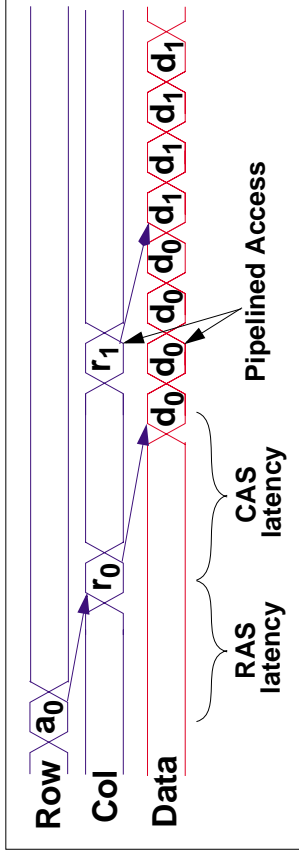


Memory System Design



DRAM Interfaces

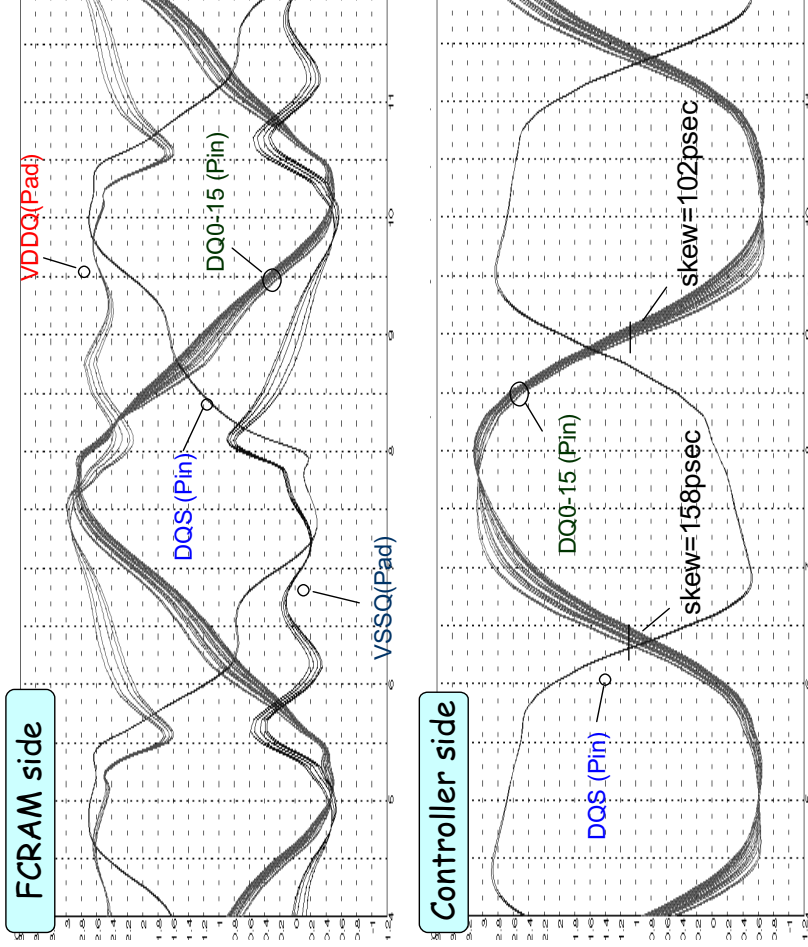
The Digital Fantasy



Pretend that the world looks like this

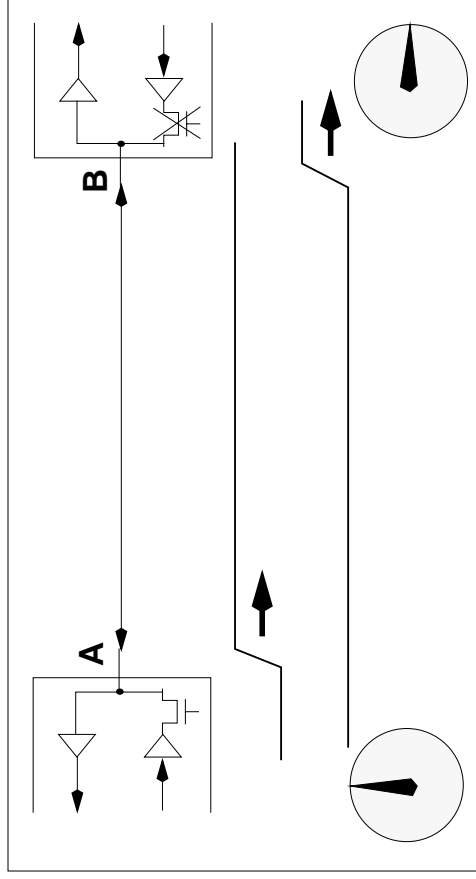
But...

The Real World



*Toshiba Presentation, Denali MemCon 2002

Signal Propagation



Ideal Transmission Line

$\sim 0.66c = 20 \text{ cm/ns}$

PC Board + Module Connectors +
Varying Electrical Loads

= Rather non-Ideal Transmission Line

Clocking Issues

Figure 1:
Sliding Time

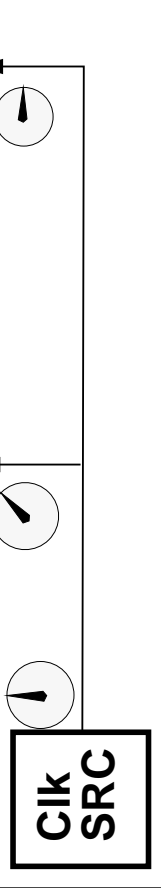
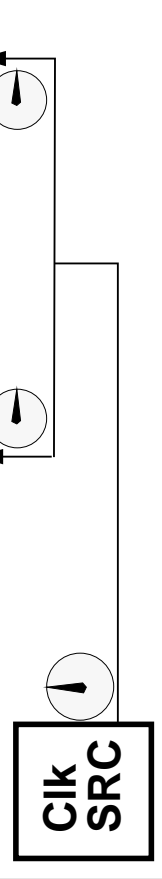
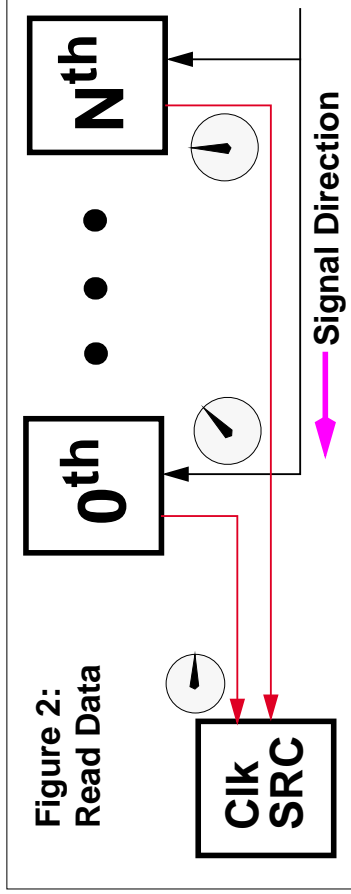
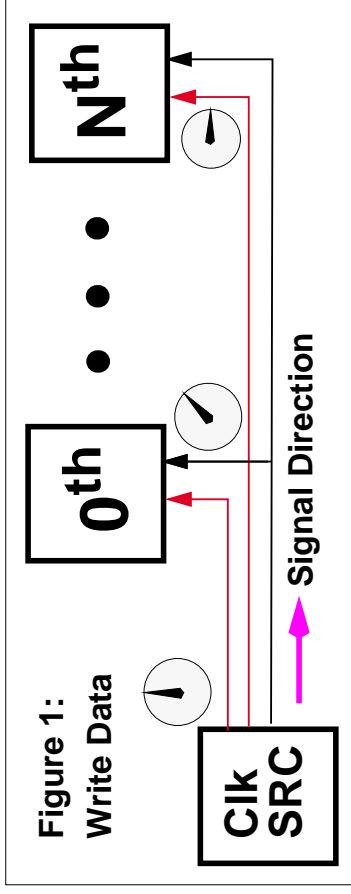


Figure 2:
H Tree?



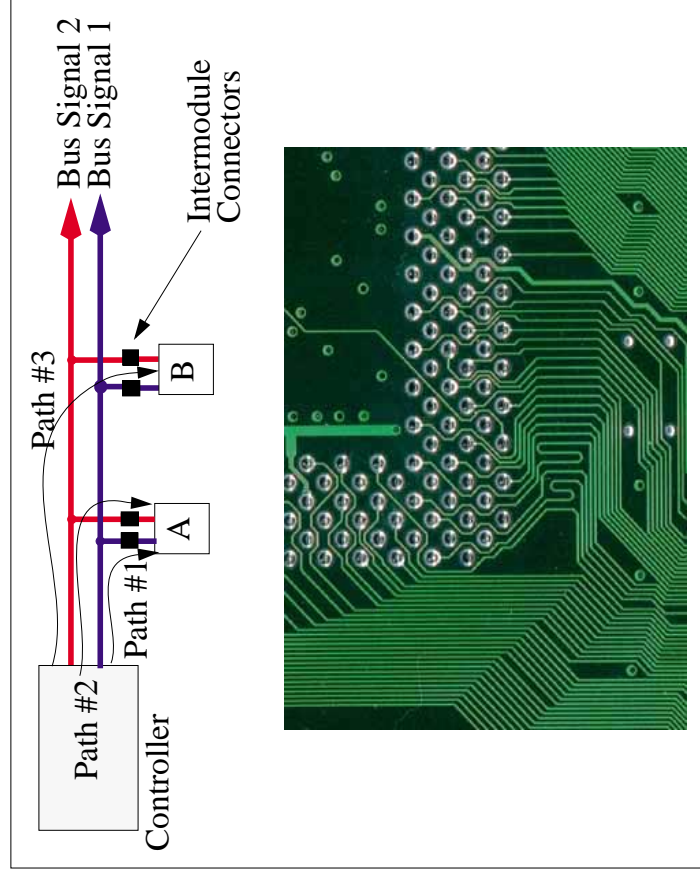
What Kind of Clocking System?

Clocking Issues



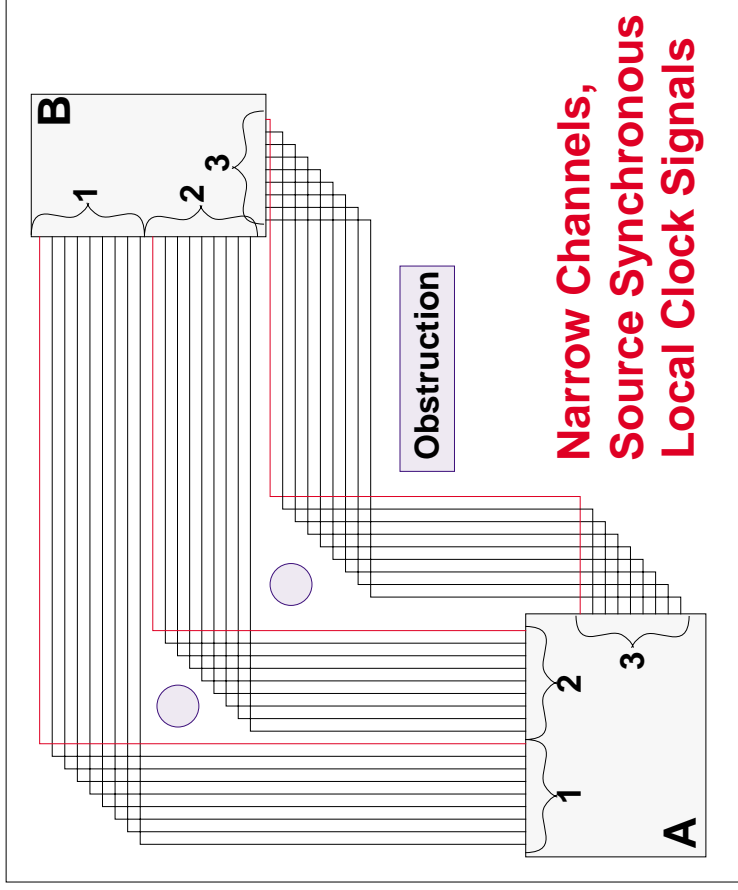
We need different “clocks” for R/W

Path Length Differential

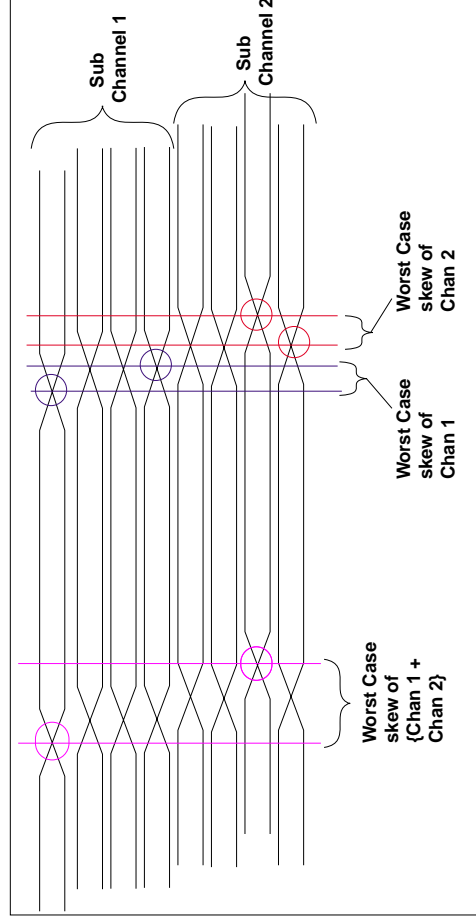


High Frequency AND Wide Parallel Busses are Difficult to Implement

Subdividing Wide Busses

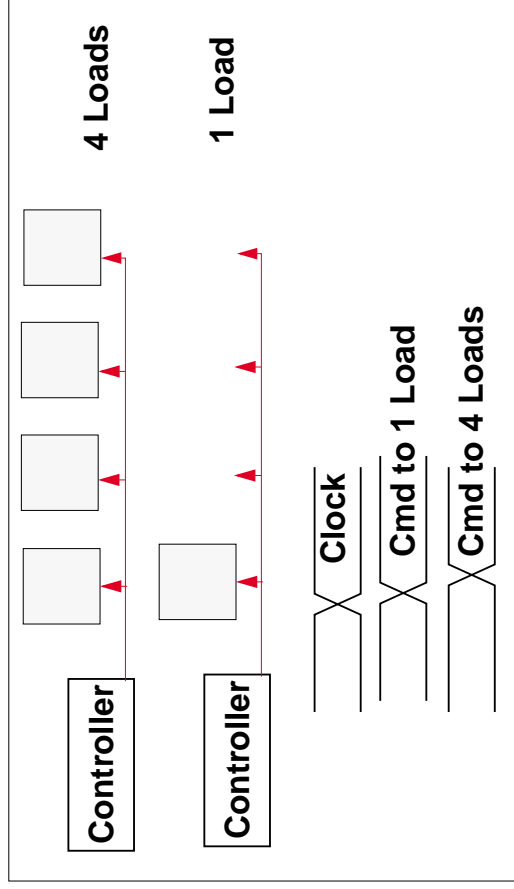


Why Subdivision Helps



Worst Case Skew must be Considered in System Timing

Timing Variations



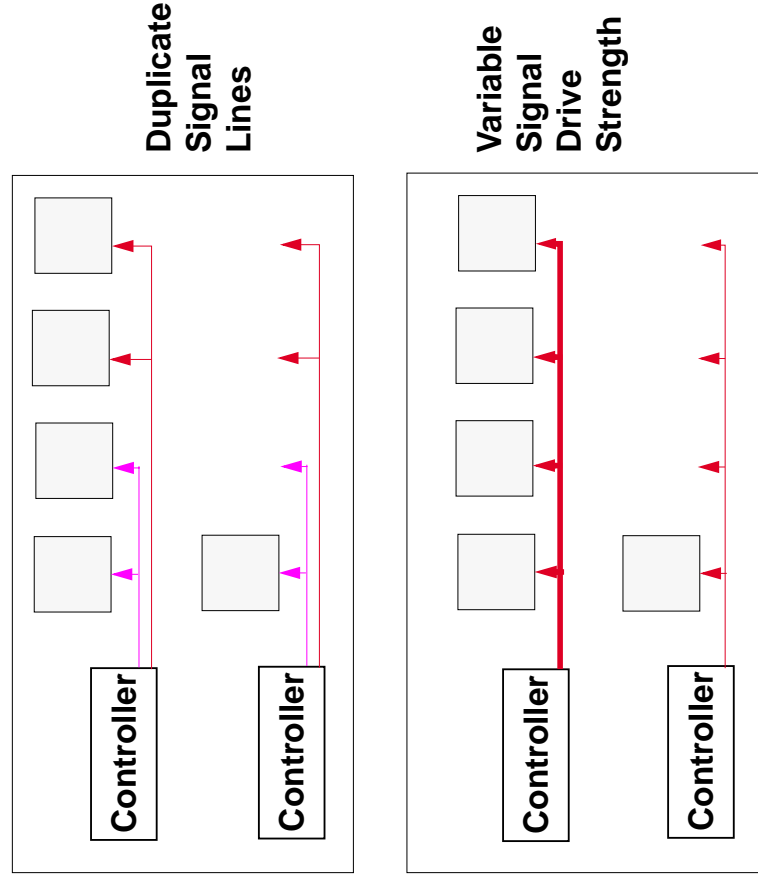
How many DIMMs in System?

How many devices on each DIMM?

Who built the memory module?

Infinite variations on timing!

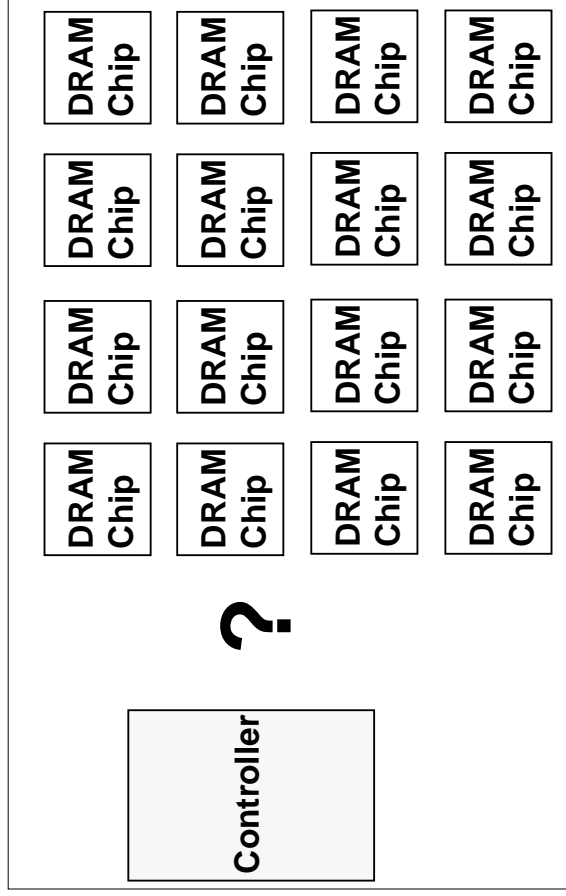
Loading Balance



Duplicate
Signal
Lines

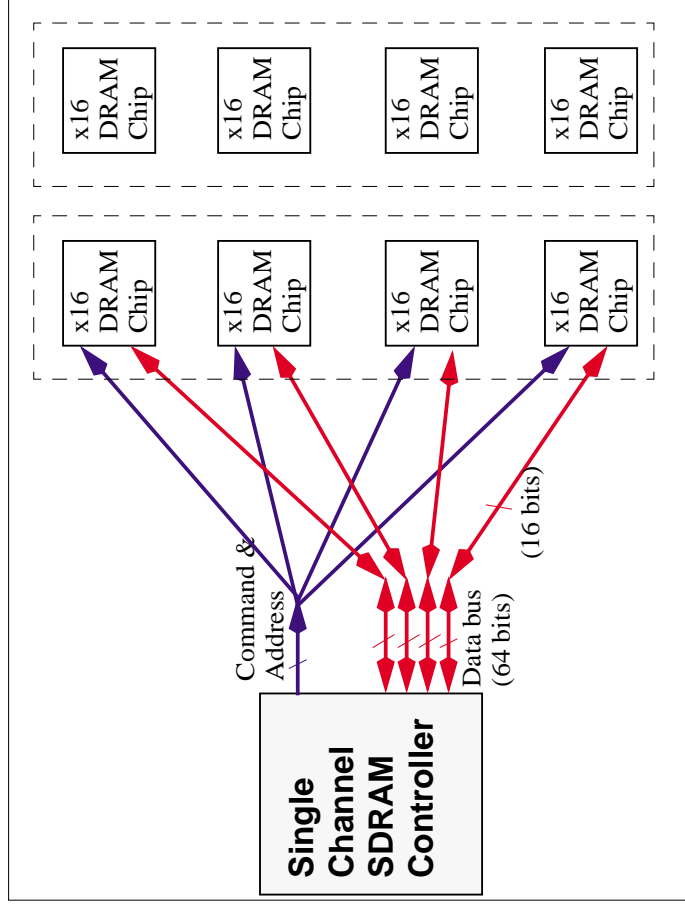
Variable
Signal
Drive
Strength

Topology



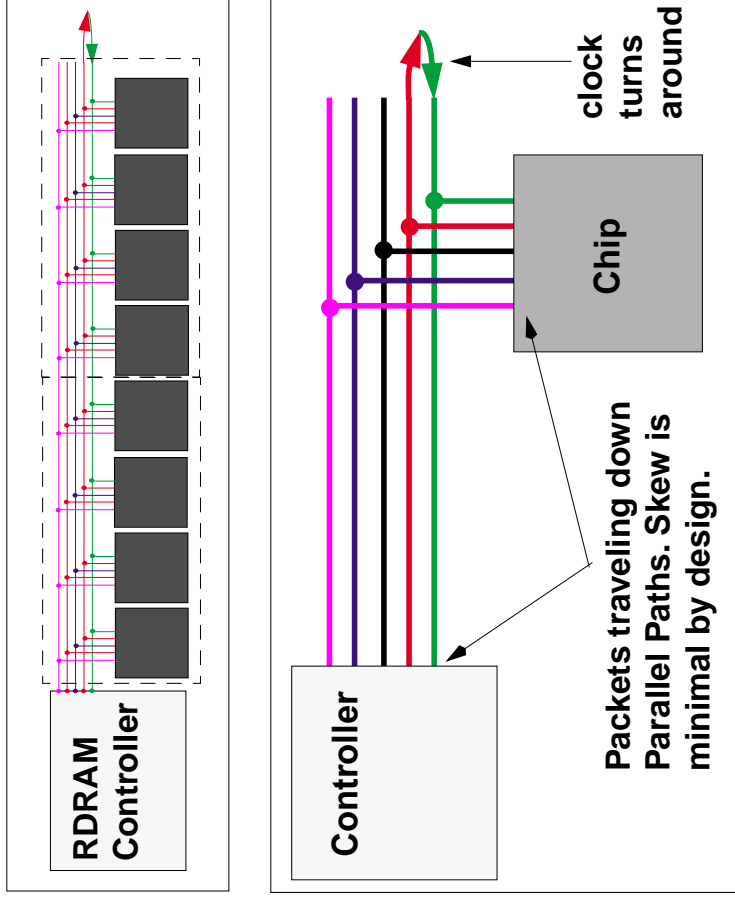
**DRAM System Topology Determines
Electrical Loading Conditions
and Signal Propagation Lengths**

SDRAM Topology Example

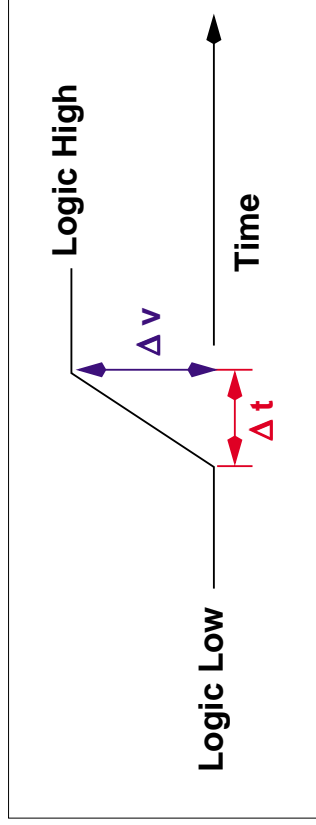


Loading Imbalance

RDRAM Topology Example



I/O Technology



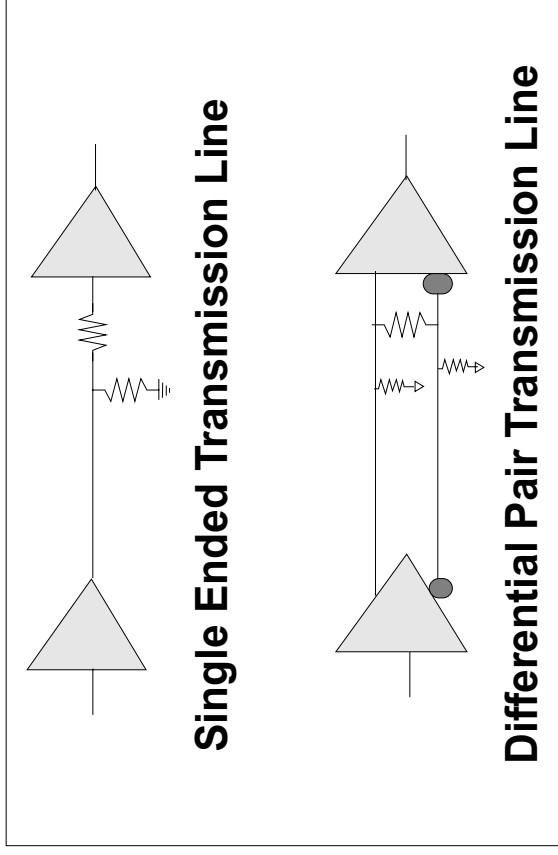
$$\text{Slew Rate} = \frac{\Delta v}{\Delta t}$$

Smaller $\Delta v =$

Smaller Δt at same slew rate

Increase Rate of bits/s/pin

I/O - Differential Pair



Single Ended Transmission Line

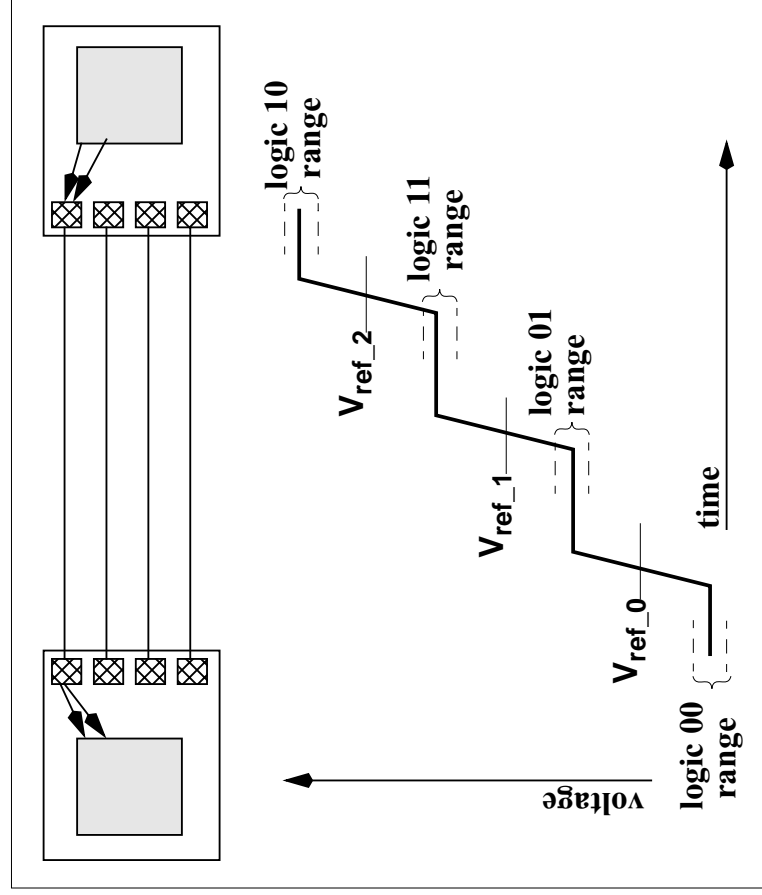
Differential Pair Transmission Line

Increase Rate of bits/s/pin ?

Cost Per Pin?

Pin Count?

I/O - Multi Level Logic



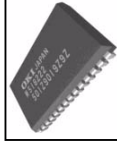
Increase Rate of bits/s/pin

Packaging



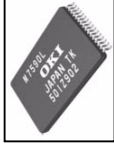
DIP

“good old days”



SOJ

Small Outline J-lead



TSOP

Thin Small Outline
Package



LQFP

Low Profile Quad
Flat Package



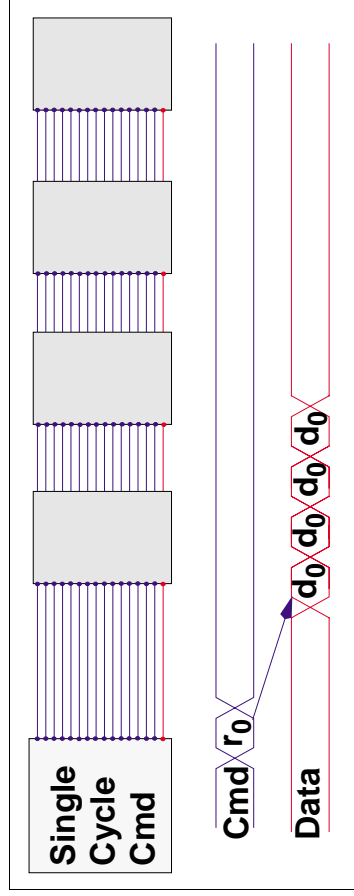
FBGA

Fine Ball Grid Array

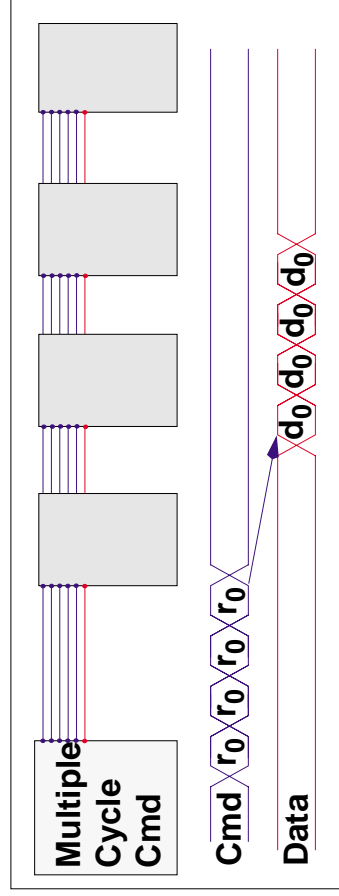
Features	Target Specification	
Package	FBGA	LQFP
Speed	800MBp	550Mbps
V _{dd} /V _{ddq}	2.5V/2.5V (1.8V)	
Interface	SSTL_2	
Row Cycle Time t _{RC}	35ns	

Memory Roadmap for Hynix NetDDR II

Access Protocol

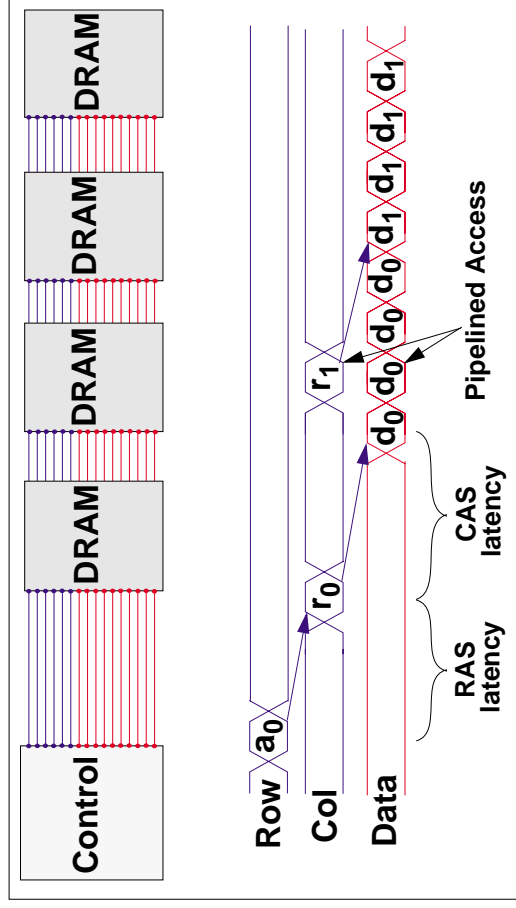


Single Cycle Command



Multiple Cycle Command

Access Protocol (r/r)



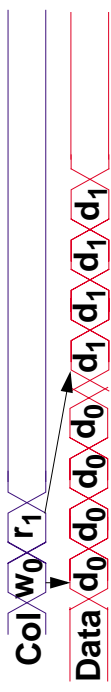
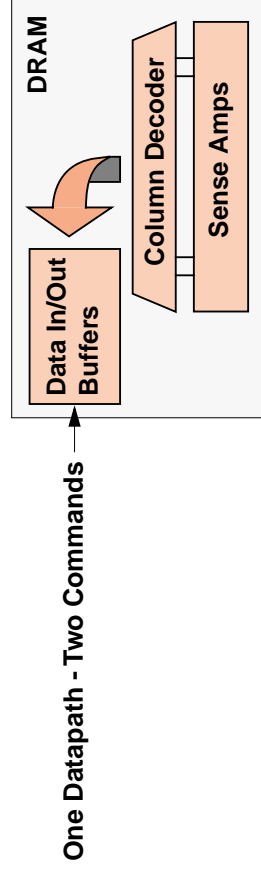
Consecutive Cache Line Read Requests to Same DRAM Row

Command

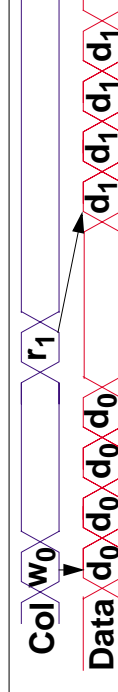
Data

a = Active (open page)
r = Read (Column Read)
d = Data (Data chunk)

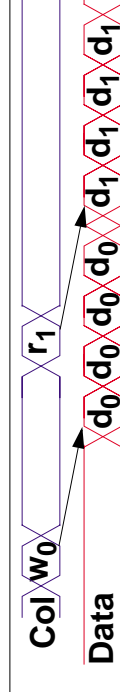
Access Protocol (r/w)



Case 1: Read Following a Write Command to Different DRAM Devices

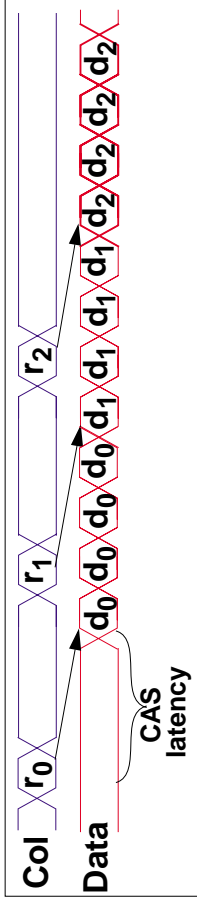


Case 2: Read Following a Write Command to Same DRAM Device

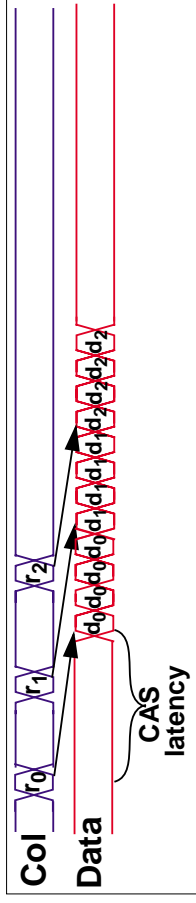


Soln: Delay Data of Write Command to match Read Latency

Access Protocol (pipelines)



Three Back-to-Back Pipelined Read Commands



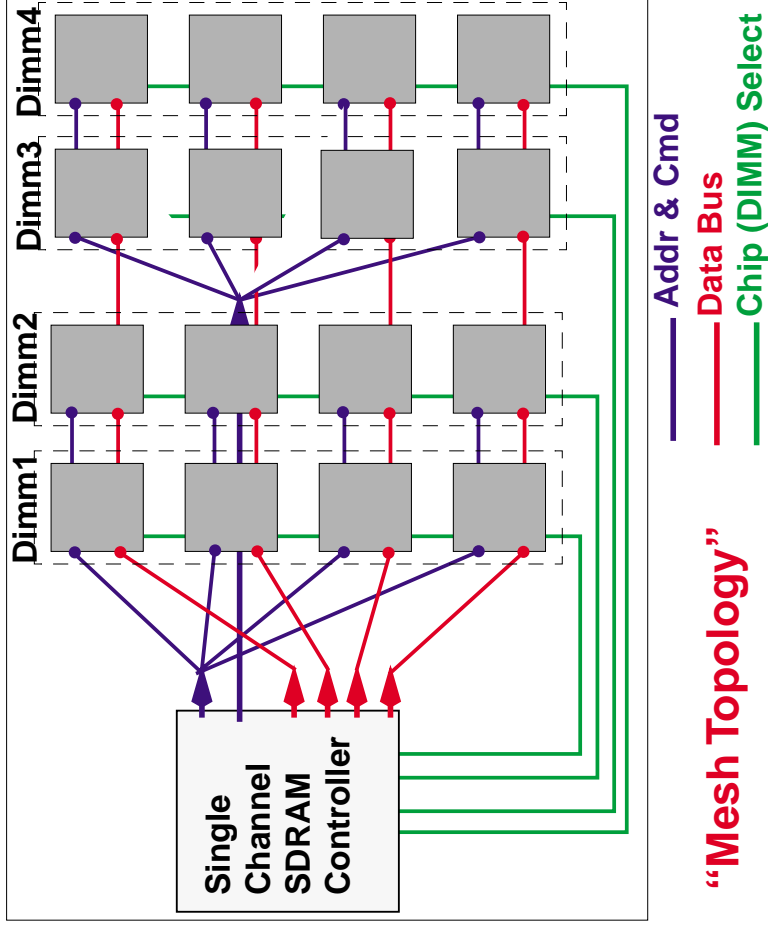
“Same” Latency, 2X pin frequency, Deeper Pipeline

When pin frequency increases, chips must either reduce “real latency”, or support longer bursts, or pipeline more commands.

Outline

- Basics
- DRAM Evolution: *Structural Path*
- Advanced Basics
- DRAM Evolution: *Interface Path*
- SDRAM, DDR SDRAM, RDRAM Memory System Comparisons
- Processor-Memory System Trends
- RDRAM, FCGRAM, DDR II Memory Systems Summary
- Future Interface Trends & Research Areas
- Performance Modeling: *Architectures, Systems, Embedded*

SDRAM System In Detail



SDRAM Chip

133 MHz (7.5ns cycle time)

Multiplexed Command/Address Bus

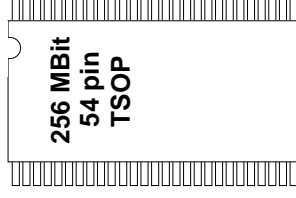
Programmable Burst Length, 1,2,4 or 8

Quad Banks Internally

Supply Voltage of 3.3V

Low Latency, CAS = 2 , 3

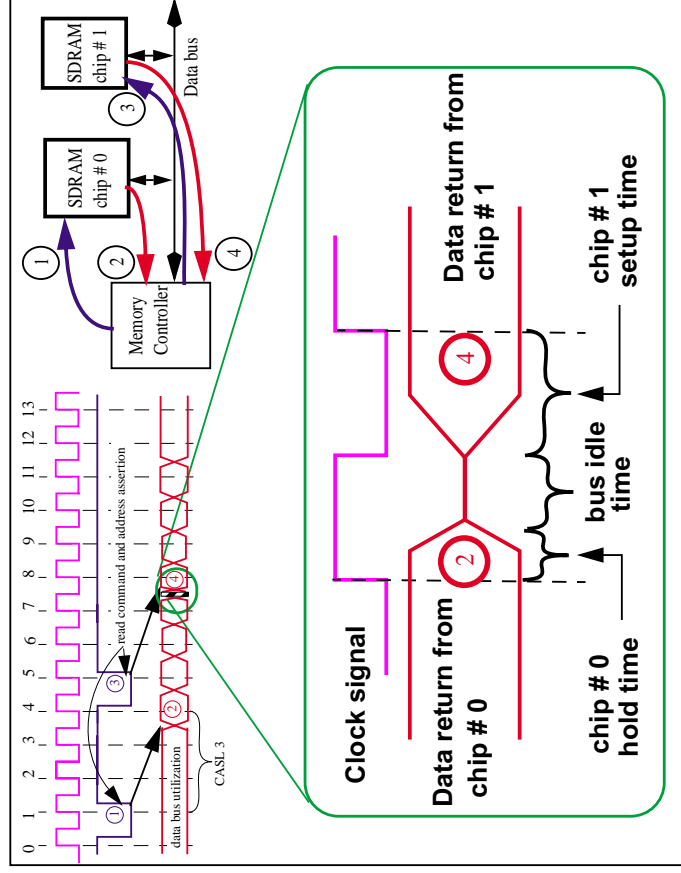
LVTTTL Signaling (0.8V to 2.0V)
(0 to 3.3V rail to rail.)



14 Pwr/Gnd
16 Data
15 Addr
7 Cmd
1 Clk
1 NC

Condition Specification	Cur.	Pwr
Operating (Active) Burst = Continuous	300mA	1W
Operating (Active) Burst = 2	170mA	560mW
Standby (Active) All banks active	60mA	200mW
Standby (powerdown) All banks inactive	2mA	6.6mW

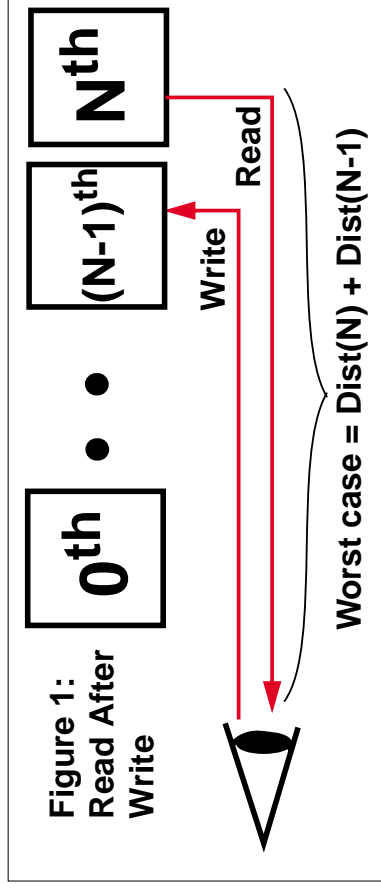
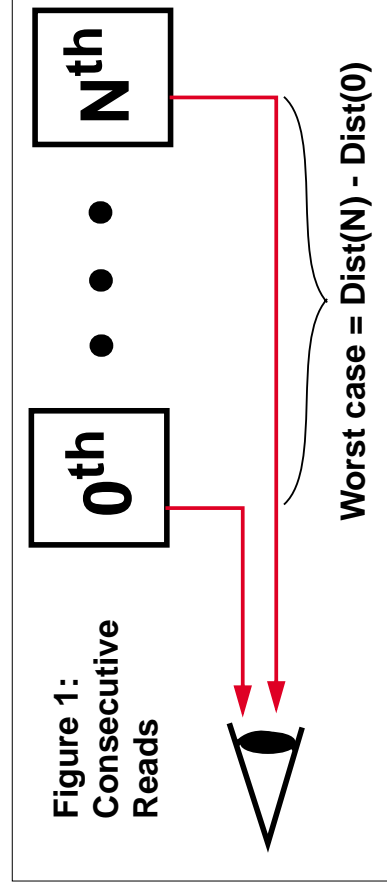
SDRAM Access Protocol (r/r)



Back-to-back Memory Read Accesses to Different Chips in SDRAM

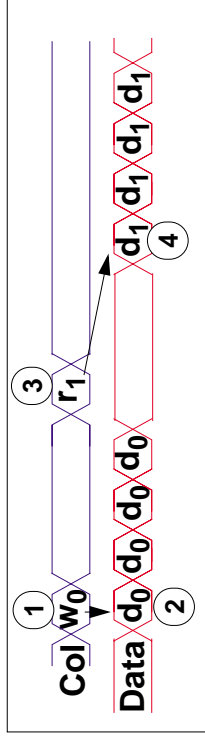
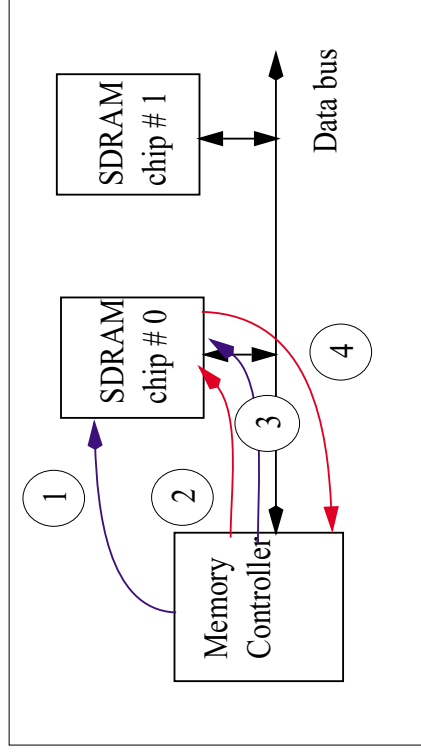
Clock Cycles are still long enough to allow for pipelined back-to-back Reads

SDRAM Access Protocol (w/r)



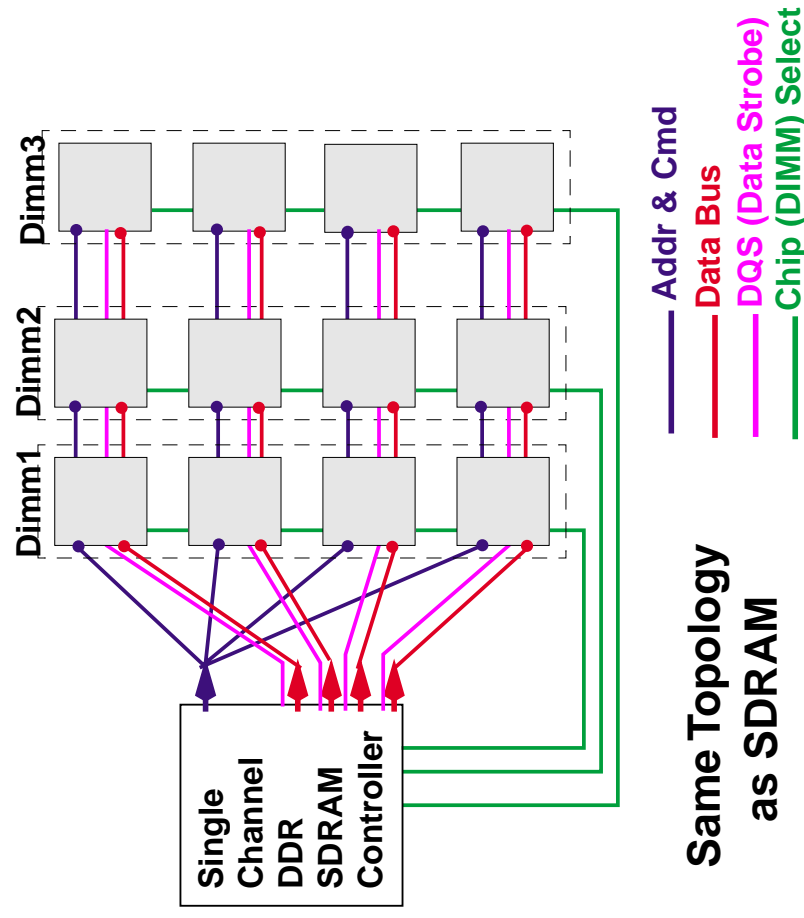
Bus Turn Around

SDRAM Access Protocol (w/r)



Read Following a Write Command to Same SDRAM Device

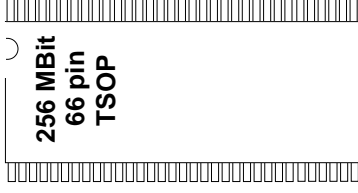
DDR SDRAM System



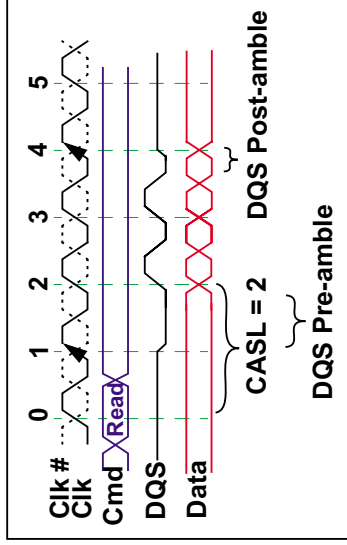
Same Topology
as SDRAM

DDR SDRAM Chip

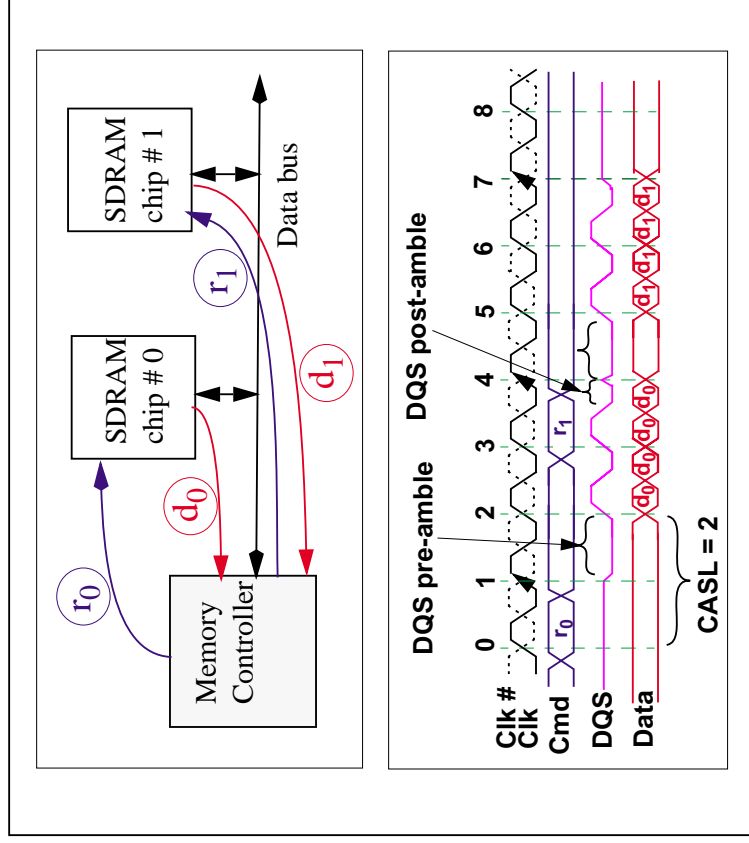
- 133 MHz (7.5ns cycle time)
- Multiplexed Command/Address Bus
- Programmable Burst Lengths, 2, 4 or 8*
- Quad Banks Internally
- Supply Voltage of 2.5V*
- Low Latency, CAS = 2 , 2.5, 3 *
- SSTL-2 Signaling (Vref +/- 0.15V)
(0 to 2.5V rail to rail)



- 16 Pwr/Gnd*
- 16 Data
- 15 Addr
- 7 Cmd
- 2 Clk *
- 7 NC *
- 2 DQS *
- 1 Vref *

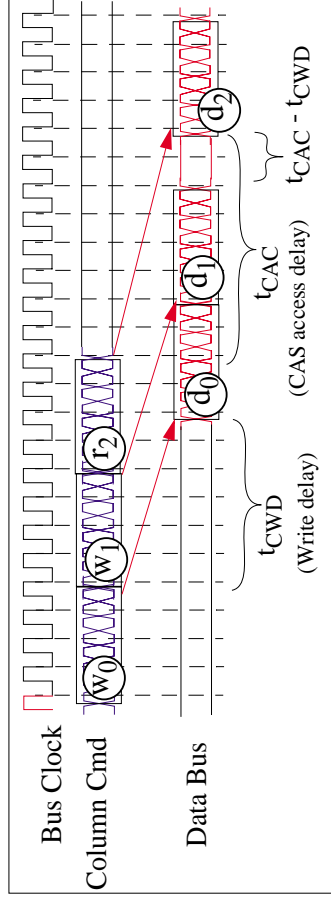
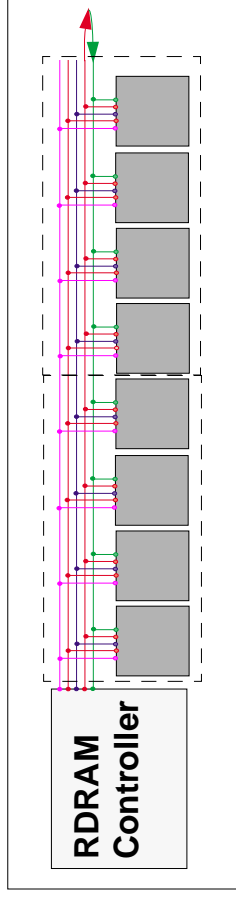


DDR SDRAM Protocol (r/r)



Back-to-back Memory Read Accesses to Different Chips in DDR SDRAM

RDRAM System



Two Write Commands Followed by a Read Command

Packet Protocol : Everything in 8 (half) cycle packets

Direct RDRAM Chip

400 MHz (2.5ns cycle time)

Separate Row-Col Command Busses

Burst Length = 8*

4/16/32 Banks Internally*

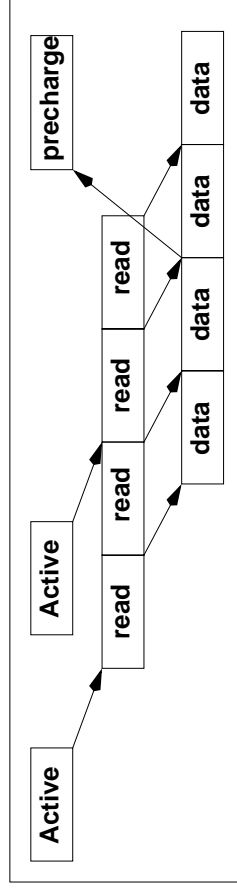
Supply Voltage of 2.5V*

Low Latency, CAS = 4 to 6 full cycles*

**RSL Signaling (Vref +/- 0.2V)
(800 mV rail to rail)**

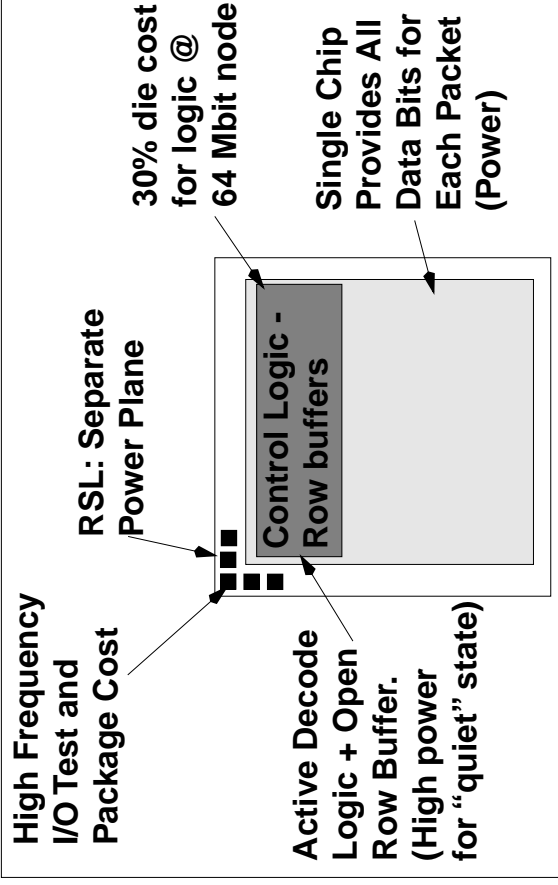
256 MBit
86 pin
FBGA

49 Pwr/Gnd*
16 Data
8 Addr/Cmd
4 Clk*
6 CTL*
2 NC
1 Vref*



**All packets are 8 (half) cycles in length,
the protocol allows near 100% bandwidth
utilization on all channels. (Addr/Cmd/Data)**

RDRAM Drawbacks



Significant Cost Delta for First Generation

System Comparison

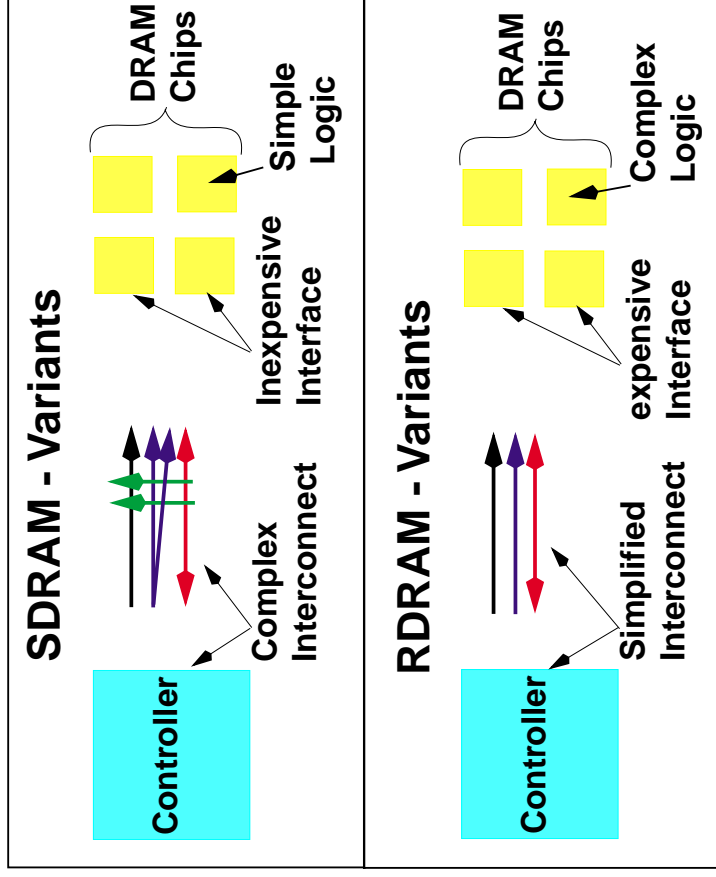
	SDRAM	DDR	RDRAM
Frequency (MHz)	133	133*2	400*2
Pin Count (Data Bus)	64	64	16
Pin Count (Controller)	102	101	33
Theoretical Bandwidth (MB/s)	1064	2128	1600
Theoretical Efficiency (data bits/cycle/pin)	0.63	0.63	0.48
Sustained BW (MB/s)*	655	986	1072
Sustained Efficiency* (data bits/cycle/pin)	0.39	0.29	0.32
RAS + CAS (t_{RAC}) (ns)	45 ~ 50	45 ~ 50	57 ~ 67
CAS Latency (ns)**	22 ~ 30	22 ~ 30	40 ~ 50

133 MHz P6 Chipset + SDRAM CAS Latency ~ 80 ns

*StreamAdd

**Load to use latency

Differences of Philosophy



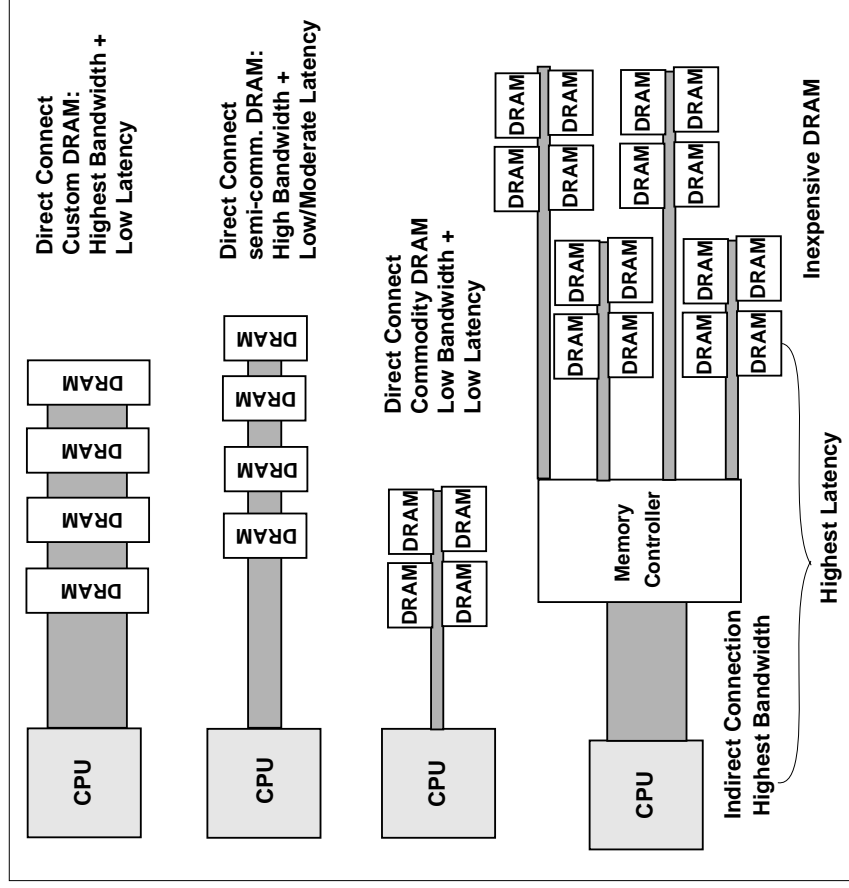
Complexity Moved to DRAM

Technology Roadmap (ITRS)

	2004	2007	2010	2013	2016
Semi Generation (nm)	90	65	45	32	22
CPU MHz	3990	6740	12000	19000	29000
MLogicTransistors/ cm ²	77.2	154.3	309	617	1235
High Perf chip pin count	2263	3012	4009	5335	7100
High Performance chip cost (cents/pin)	1.88	1.61	1.68	1.44	1.22
Memory pin cost (cents/pin)	0.34 - 1.39	0.27 - 0.84	0.22 - 0.34	0.19 - 0.39	0.19 - 0.33
Memory pin count	48-160	48-160	62-208	81-270	105-351

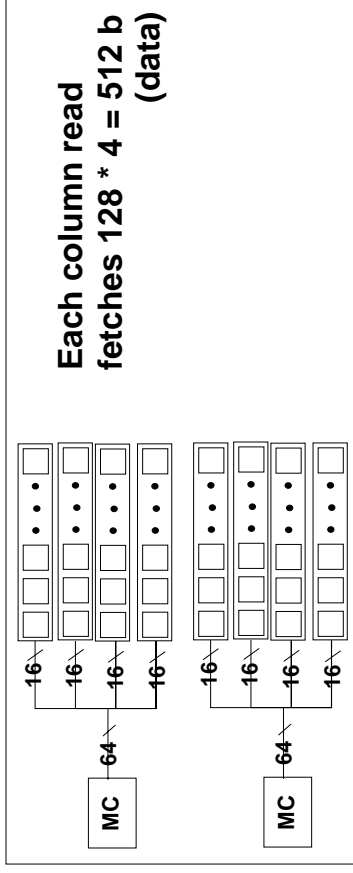
Trend:
Free Transistors &
Costly Interconnects

Choices for Future



EV7 + RDRAM (Compaq/HP)

- RDRAM Memory (2 Controllers)
- Direct Connection to processor
- 75ns Load to use latency
- 12.8 GB/s Peak bandwidth
- 6 GB/s read or write bandwidth
- 2048 open pages ($2 * 32 * 32$)



What if EV7 Used DDR?

- **Peak Bandwidth 12.8 GB/s**
- **6 Channels of 133*2 MHz DDR SDRAM ==**
- **6 Controllers of 6 64 bit wide channels, or**
- **3 Controllers of 3 128 bit wide channels**

System	EV7 + RDRAM	EV7 + 6 controller DDR SDRAM	EV7 + 3 controller DDR SDRAM
Latency	75 ns	~ 50 ns*	~ 50 ns*
Pin count	~265** + Pwr/Gnd	~ 600** + Pwr/Gnd	~ 600** + Pwr/Gnd
Controller Count	2	6***	3***
Open pages	2048	144	72

* page hit CAS + memory controller latency.

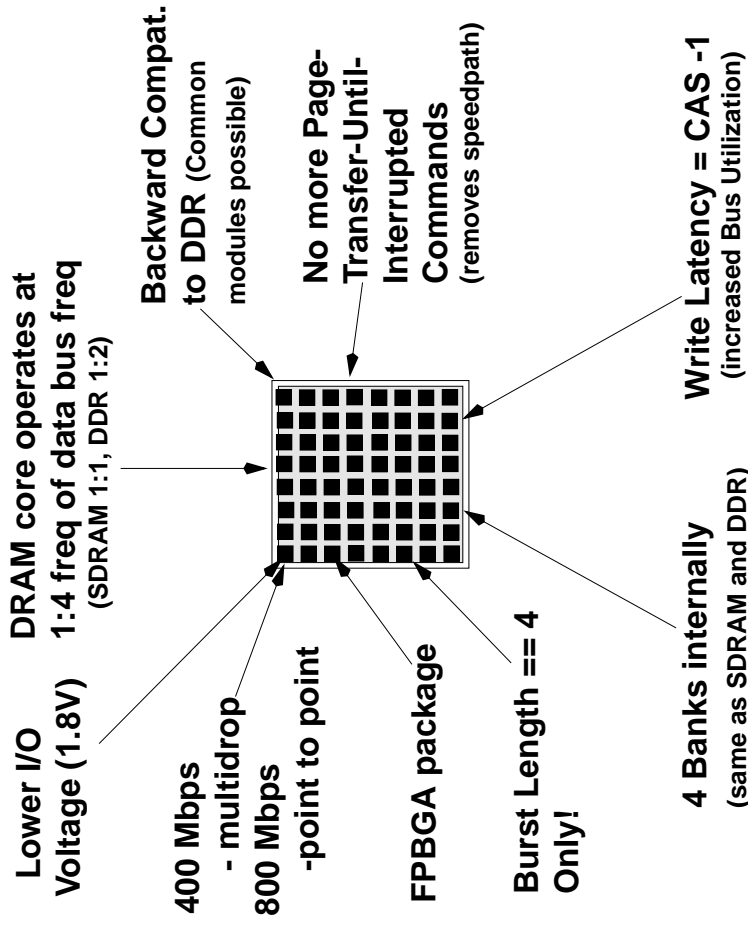
** including all signals, address, command, data, clock, not including ECC or parity

*** 3 controller design is less bandwidth efficient.

What's Next?

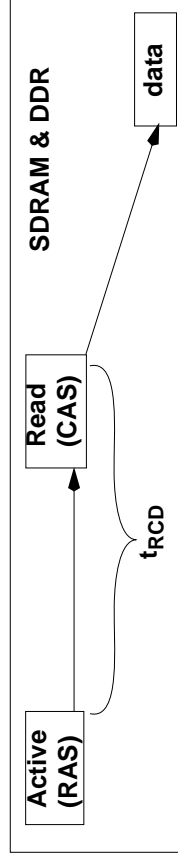
- **DDR II**
- **FCRAM**
- **RLDRAM**
- **RDRAM (Yellowstone etc)**
- **Kentron QBM**

DDR II - DDR Next Gen

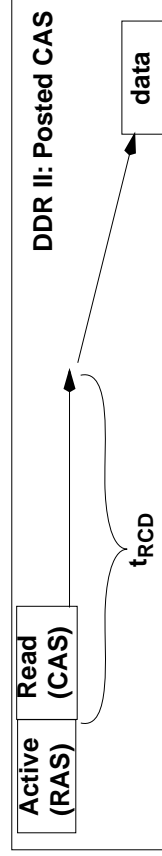


DDR II - Continued

Posted Commands



SDRAM & DDR SDRAM relies on memory controller to know t_{RCD} and issue CAS after t_{RCD} for lowest latency.



Internal counter delays CAS command, DRAM chip issues "real" command after t_{RCD} for lowest latency.

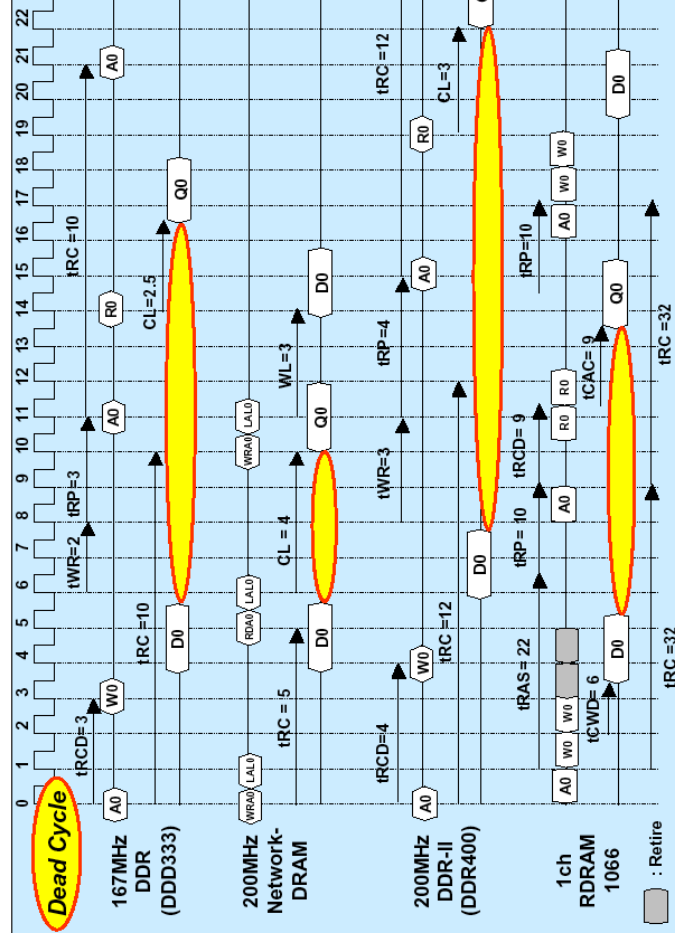
FCRAM

Fast Cycle RAM (aka Network-DRAM)

Features	DDR SDRAM	FCRAM/Network-DRAM
V _{dd} , V _{ddq}	2.5 +/- 0.2V	2.5 +/- 0.15
Electrical Interface	SSTL-2	SSTL-2
Clock Frequency	100~167 MHz	154~200 MHz
t _{RAC}	~40ns	22~26ns
t _{RC}	~60ns	25~30ns
# Banks	4	4
Burst Length	2,4,8	2,4
Write Latency	1 Clock	CASL-1

FCRAM/Network-DRAM looks like DDR+

FCRAM Continued



Faster t_{RC} allows Samsung to claim higher bus efficiency

* Samsung Electronics, Denali MemCon 2002

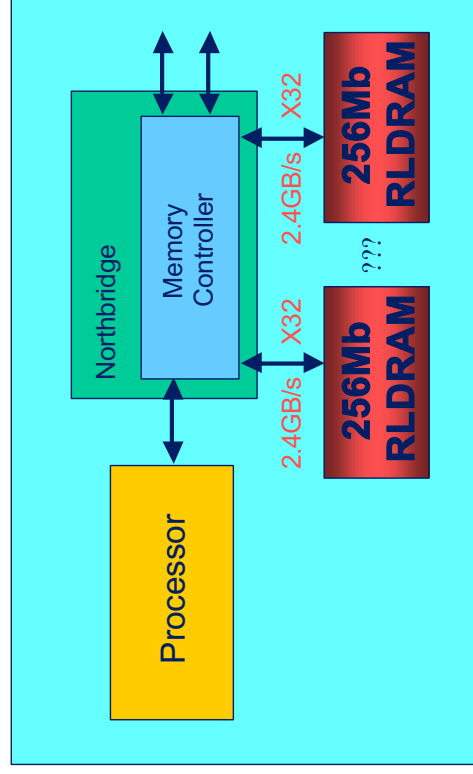
RLDRAM

DRAM Type	Frequency	Bus Width (per chip)	Peak Bandwidth (per Chip)	Random Access Time (t_{RAC})	Row Cycle Time (t_{RC})
PC133 SDRAM	133	16	200 MB/s	45 ns	60 ns
DDR 266	133 * 2	16	532 MB/s	45 ns	60 ns
PC800 RDRAM	400 * 2	16	1.6 GB/s	60 ns	70 ns
FCRAM	200 * 2	16	0.8 GB/s	25 ns	25 ns
RLDRAM	300 * 2	32	2.4 GB/s	25 ns	25 ns

**Comparable to FCRAM in latency
Higher Frequency (No Connectors)
non-Multiplexed Address (SRAM like)**

RLDRAM Continued

High-end PC and Server

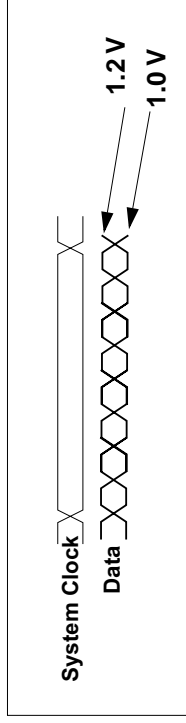


RLDRAM is a great replacement to SRAM in L3 cache applications because of its high density, low power and low cost

* Infineon Presentation, Denali MemCon 2002

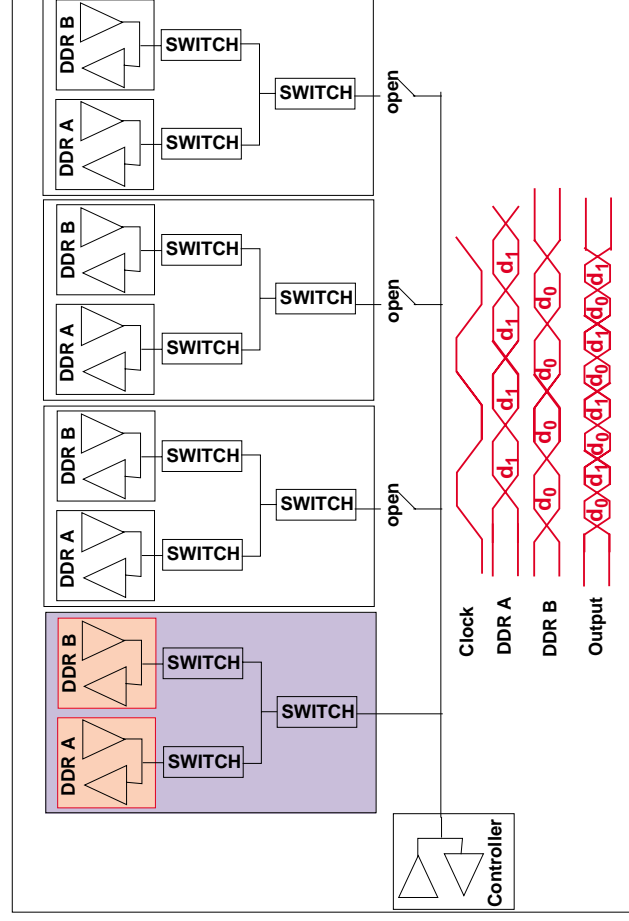
RAMBUS Yellowstone

- Bi-Directional Differential Signals
- Ultra low 200mV p-p signal swings
- 8 data bits transferred per clock
- 400 MHz system clock
- 3.2 GHz effective data frequency
- Cheap 4 layer PCB
- Commodity packaging



Octal Data Rate (ODR) Signaling

Kentron QBM™

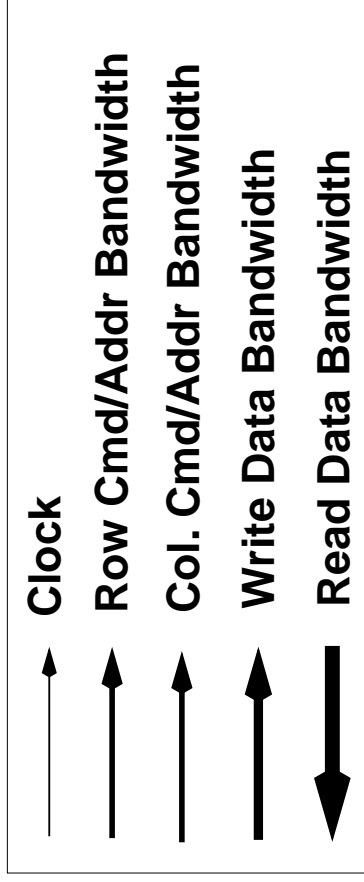


“Wrapper Electronics around DDR memory”
Generates 4 data bits per cycle instead of 2.

Quad Band Memory

A Different Perspective

Everything is bandwidth

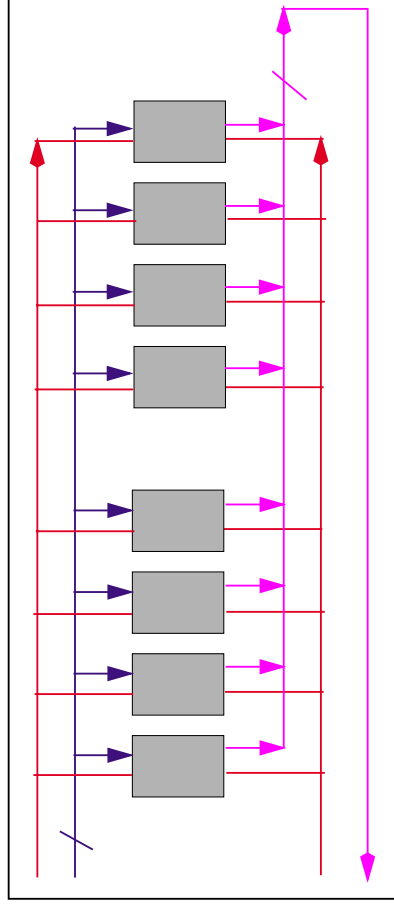


Latency and Bandwidth

Pin-bandwidth and

Pin-transition *Efficiency (bits/cycle/sec)

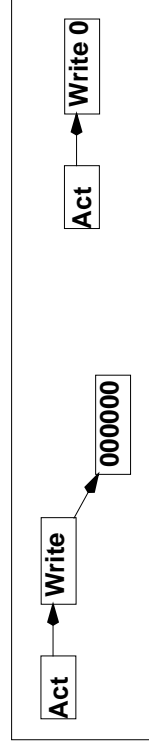
Research Areas: Topology



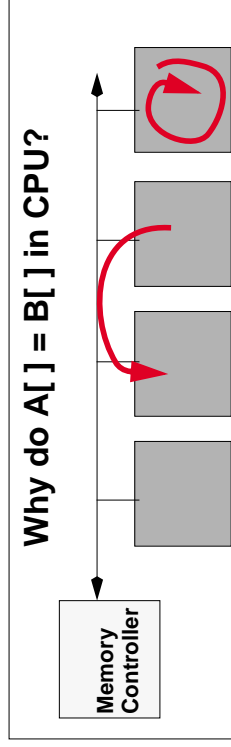
Unidirectional Topology:

- Write Packets sent on Command Bus
- Pins used for Command/Address/Data
- Further Increase of Logic on DRAM chips

Memory Commands?



Instead of $A[] = 0$; Do "write 0"



Why do $A[] = B[]$ in CPU?

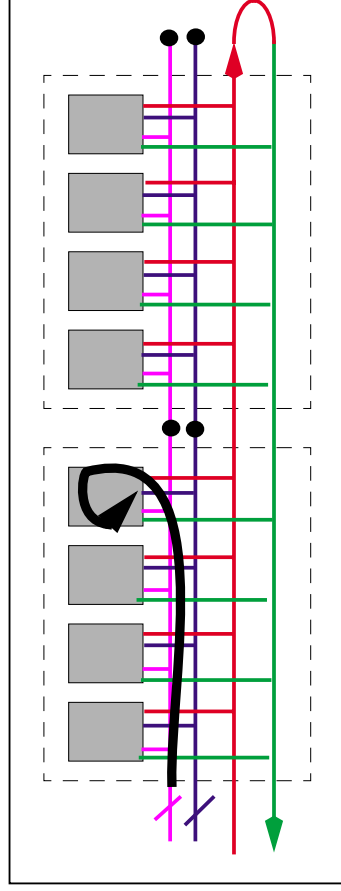
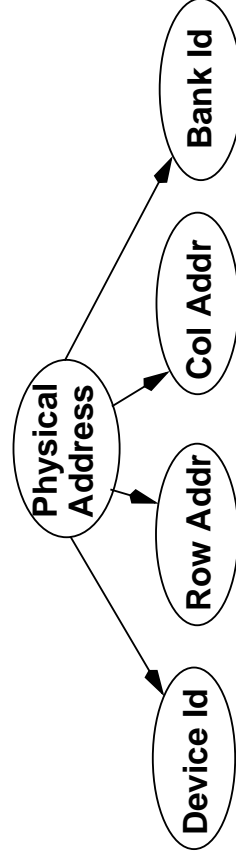
Move Data inside of DRAM or between DRAMs.

Why do STREAMadd in CPU?

$$A[] = B[] + C[]$$

Active Pages *(Chong et. al. ISCA '98)

Address Mapping

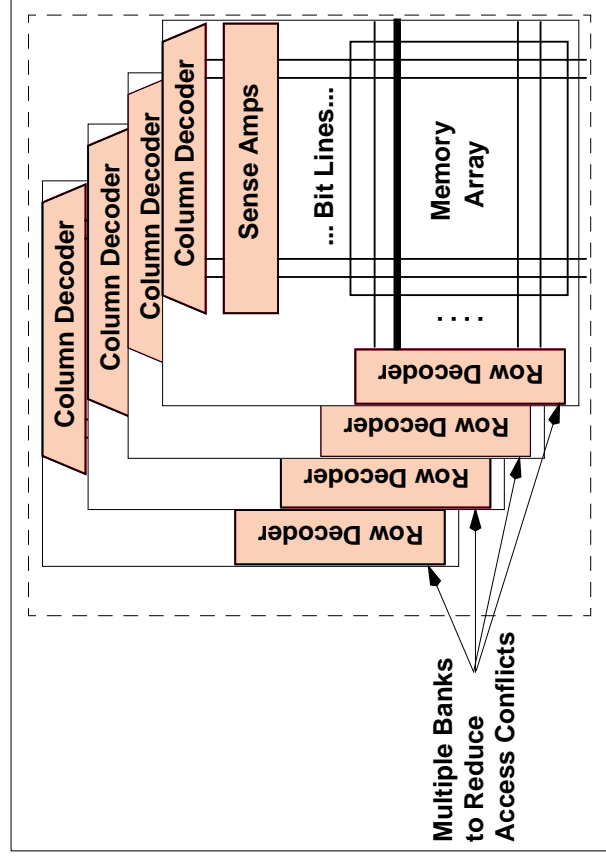


Access Distribution for Temp Control

Avoid Bank Conflicts

Access Reordering for performance

Example: Bank Conflicts

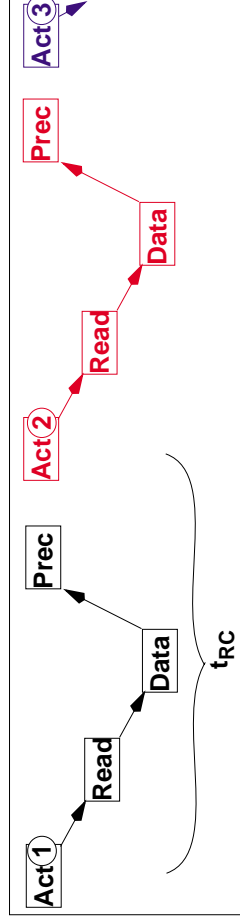


Read 05AE5700 → Device id 3, Row id 266, Bank id 0
 Read 023BB880 → Device id 3, Row id 1BA, Bank id 0
 Read 05AE5780 → Device id 3, Row id 266, Bank id 0
 Read 00CBA2C0 → Device id 3, Row id 052, Bank id 1

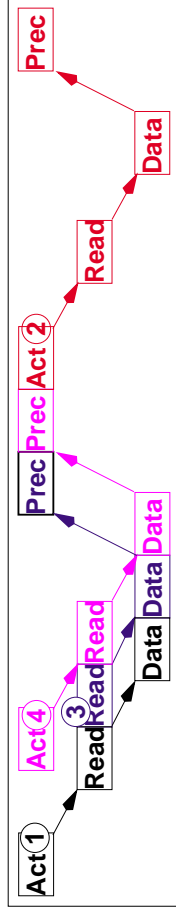
More Banks per Chip == Performance == Logic Overhead

Example: Access Reordering

① Read 05AE5700 → Device id 3, Row id 266, Bank id 0
 ② Read 023BB880 → Device id 3, Row id 1BA, Bank id 0
 ③ Read 05AE5780 → Device id 3, Row id 266, Bank id 0
 ④ Read 00CBA2C0 → Device id 1, Row id 052, Bank id 1



Strict Ordering



Memory Access Re-ordered

Act = Activate Page (Data moved from DRAM cells to row buffer)
 Read = Read Data (Data moved from row buffer to memory controller)
 Prec = Precharge (close page/evict data in row buffer/sense amp)

Outline

- Basics
- DRAM Evolution: *Structural Path*
- Advanced Basics
- DRAM Evolution: *Interface Path*
- Future Interface Trends & Research Areas
- Performance Modeling: *Architectures, Systems, Embedded*

Simulator Overview

CPU: SimpleScalar v3.0a

- 8-way out-of-order
- L1 cache: split 64K/64K, lockup free x32
- L2 cache: unified 1MB, lockup free x1
- L2 blocksize: 128 bytes

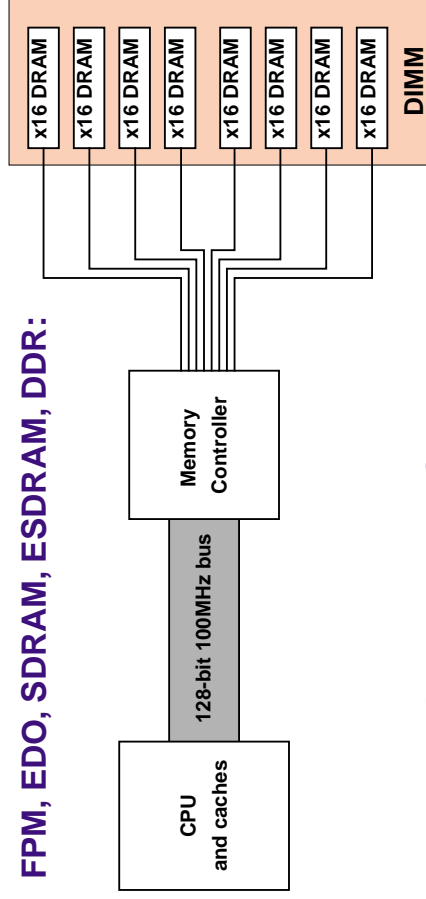
Main Memory: 8 64Mb DRAMs

- 100MHz/128-bit memory bus
- Optimistic *open-page* policy

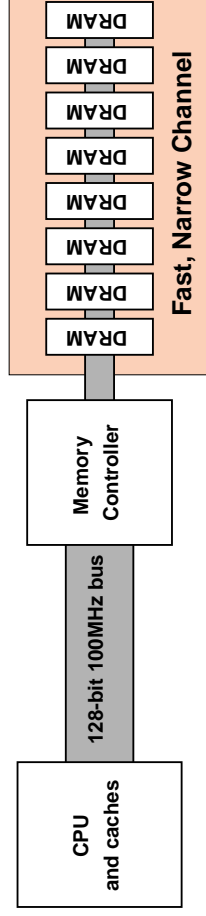
Benchmarks: SPEC '95

DRAM Configurations

FPM, EDO, SDRAM, ESDRAM, DDR:



Rambus, Direct Rambus, SDRAM:

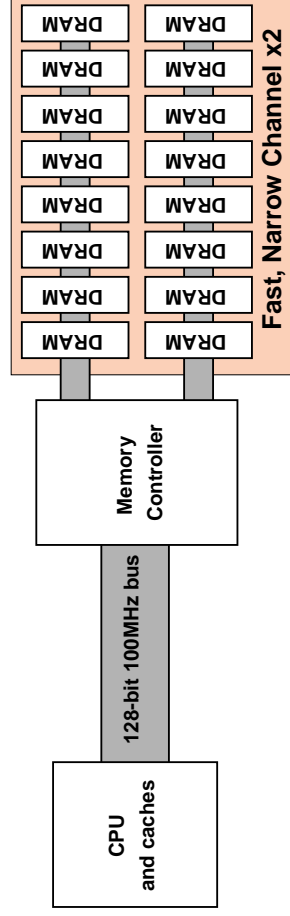


Note: TRANSFER WIDTH of Direct Rambus Channel

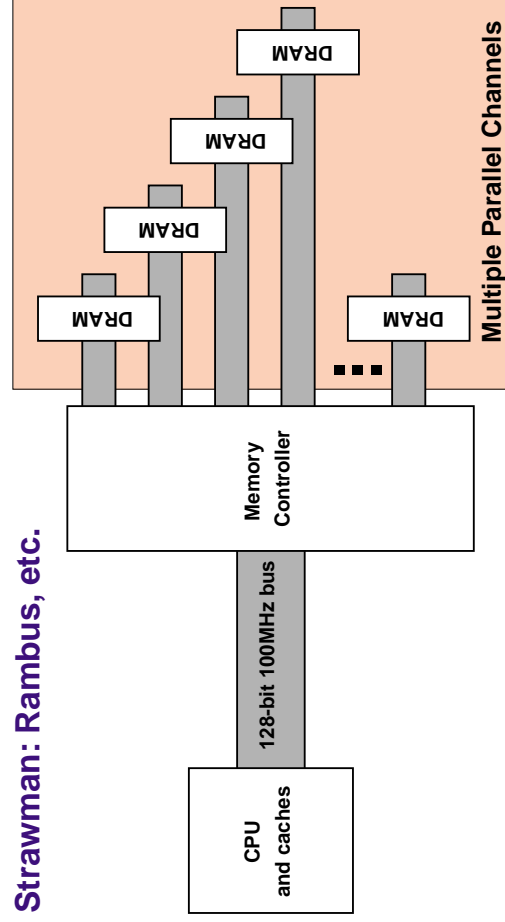
- equals that of ganged FPM, EDO, etc.
- is 2x that of Rambus & SDRAM

DRAM Configurations

Rambus & SDRAM dual-channel:

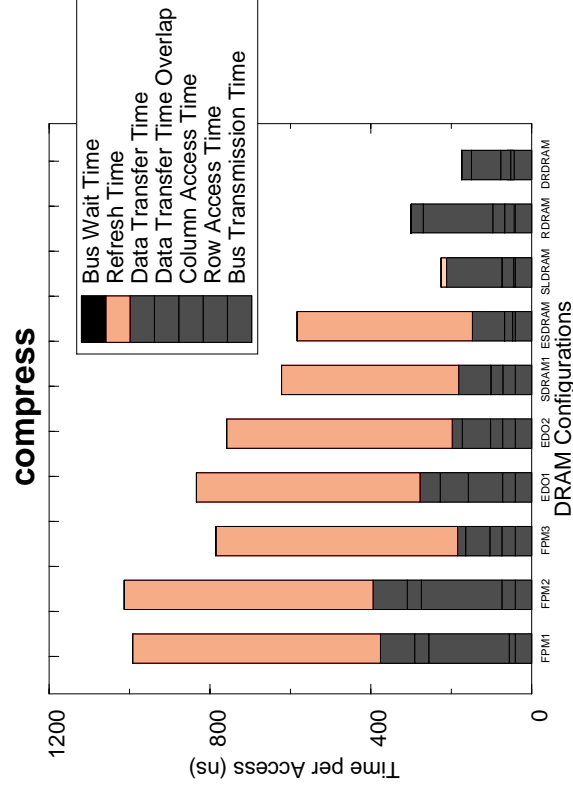


Strawman: Rambus, etc.



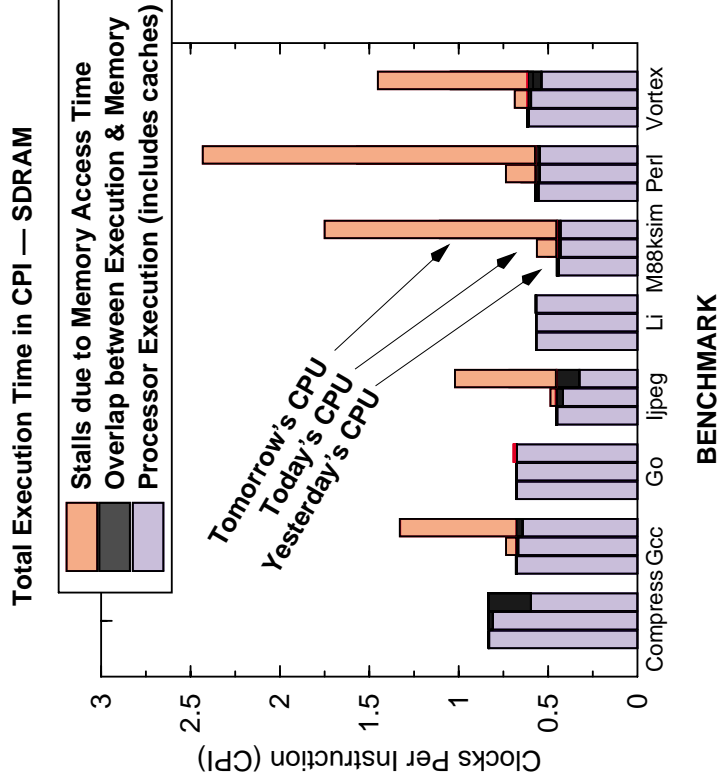
Multiple Parallel Channels

First ... Refresh Matters



Assumes refresh of each bank every 64ms

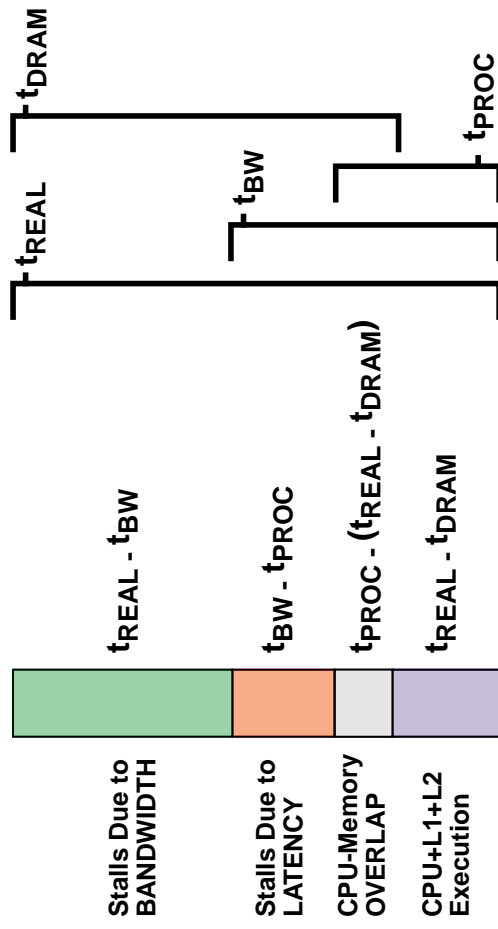
Overhead: Memory vs. CPU



Variable: speed of processor & caches

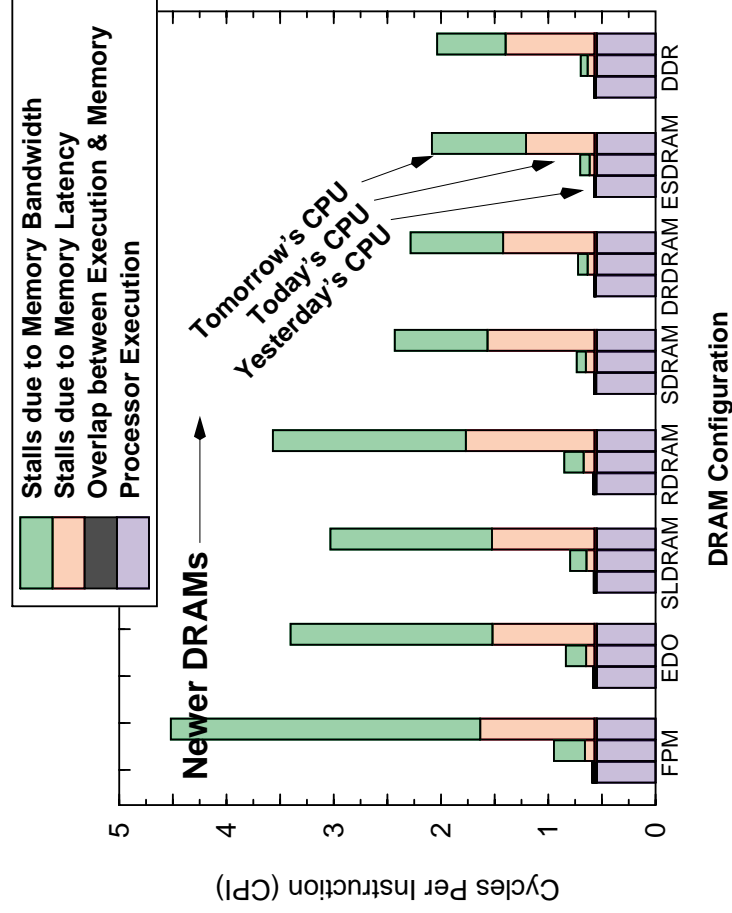
Definitions (var. on Burger, et al)

- t_{PROC} — processor with perfect memory
- t_{REAL} — realistic configuration
- t_{BW} — CPU with wide memory paths
- t_{DRAM} — time seen by DRAM system



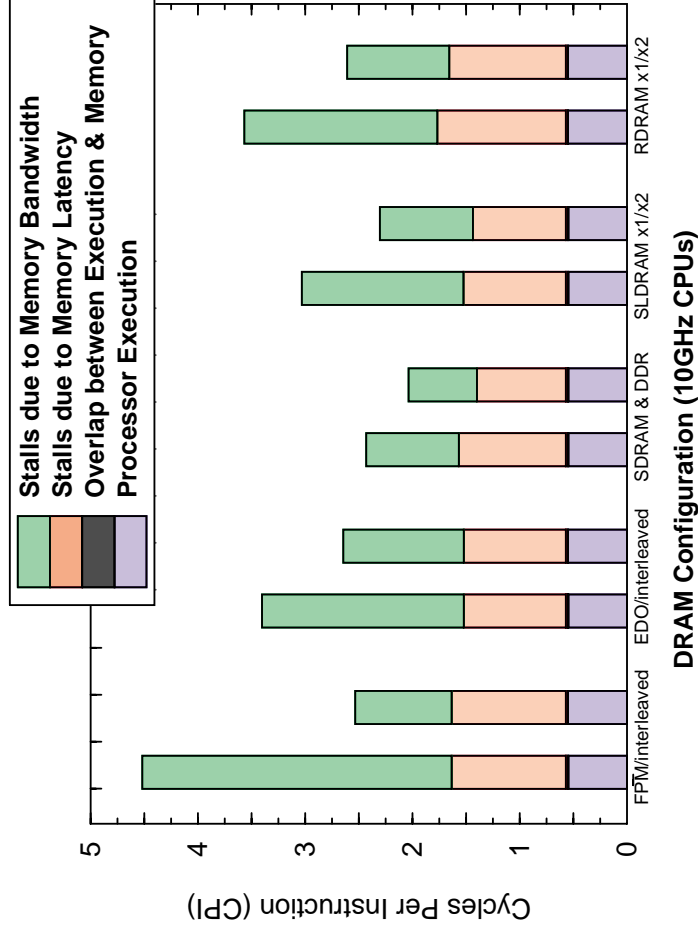
Memory & CPU — PERL

Bandwidth-Enhancing Techniques I:

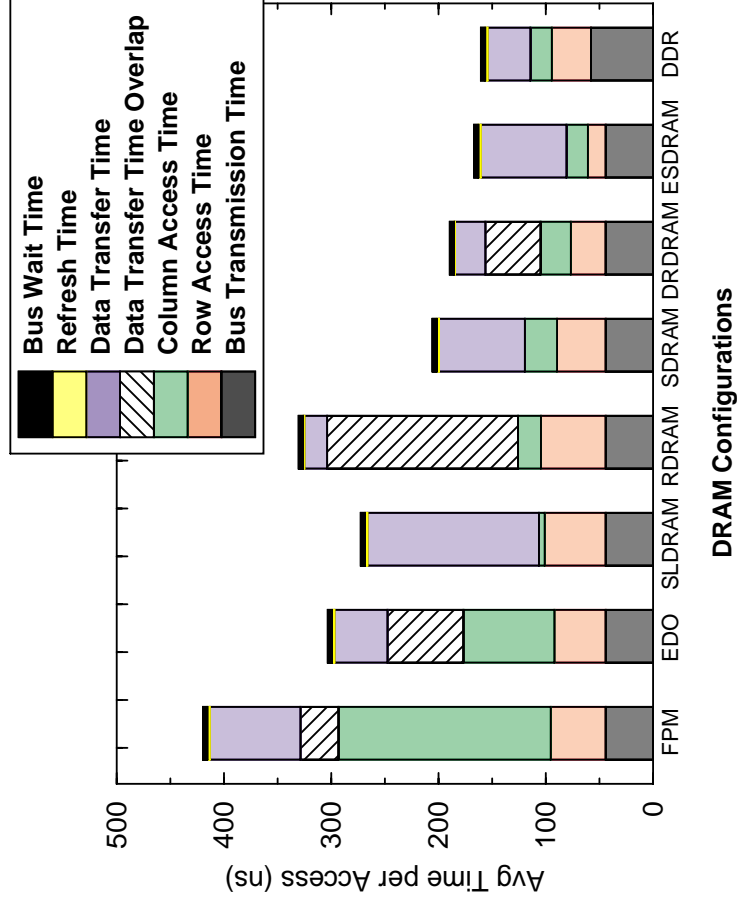


Memory & CPU — PERL

Bandwidth-Enhancing Techniques II:

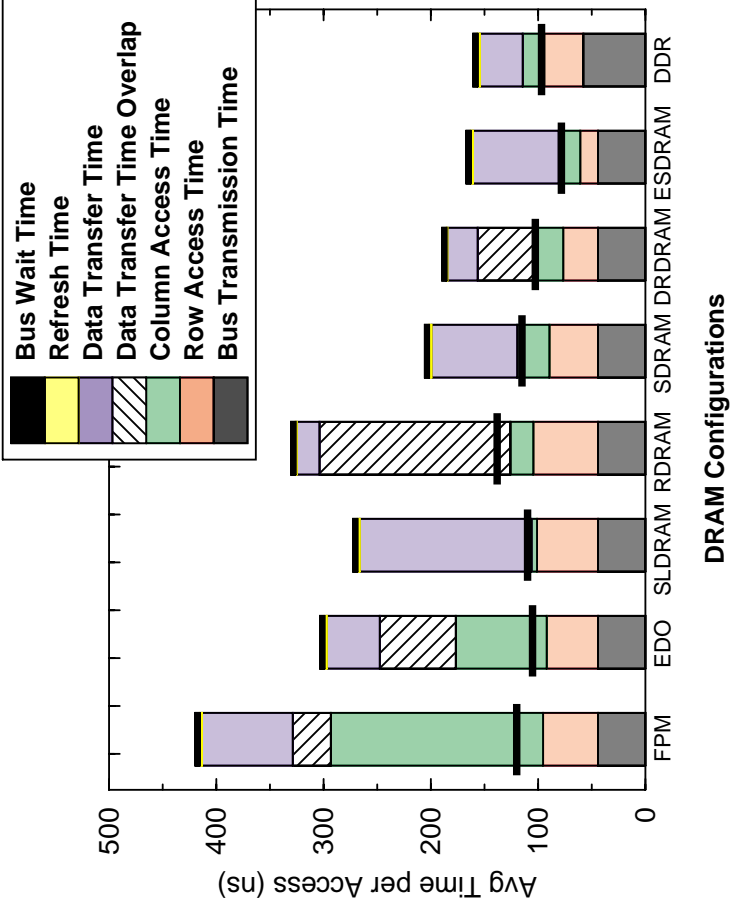


Average Latency of DRAMS



note: SLDRAM & RDRAM 2x data transfers

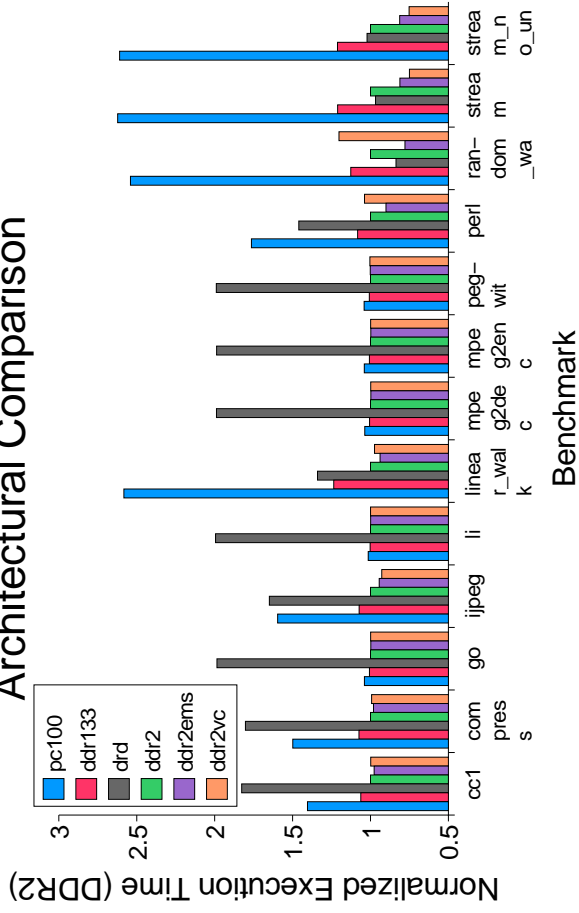
Average Latency of DRAMs



note: SLDRAM & RDRAM 2x data transfers

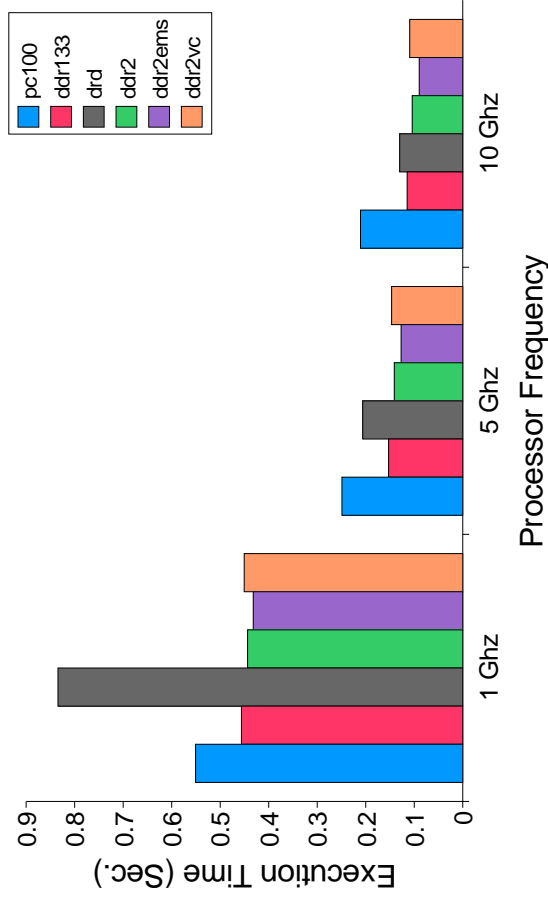
DDR2 Study Results

Architectural Comparison

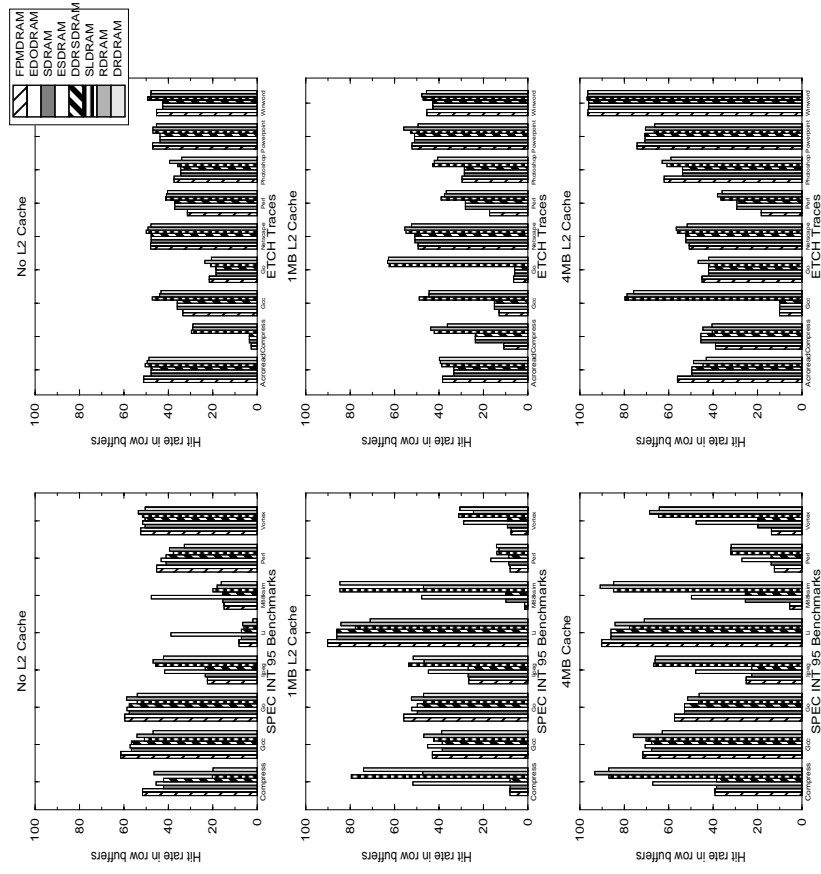


DDR2 Study Results

Perl Runtime

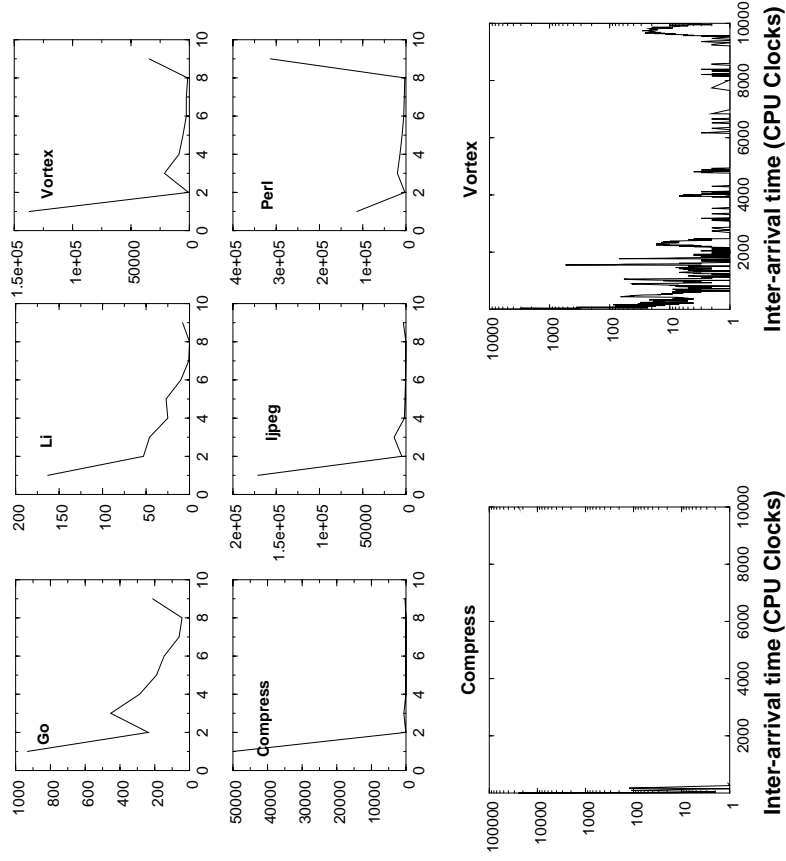


Row-Buffer Hit Rates

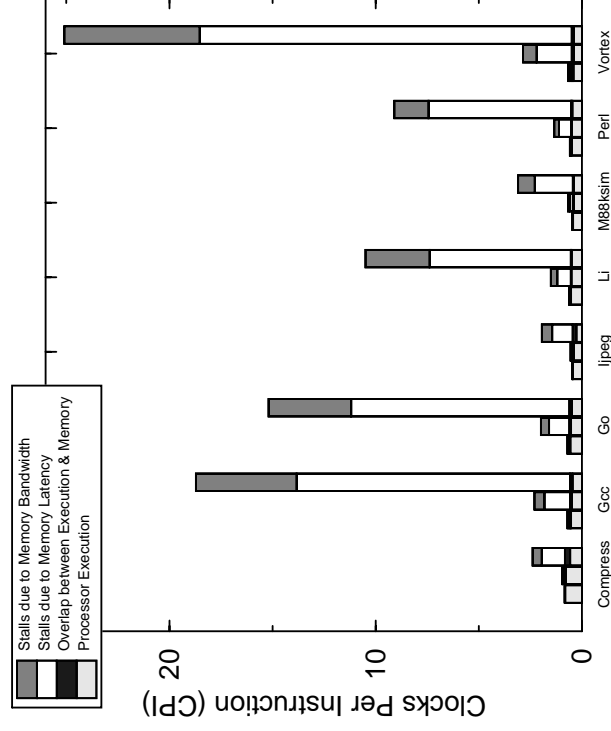


Row-Buffer Hit Rates

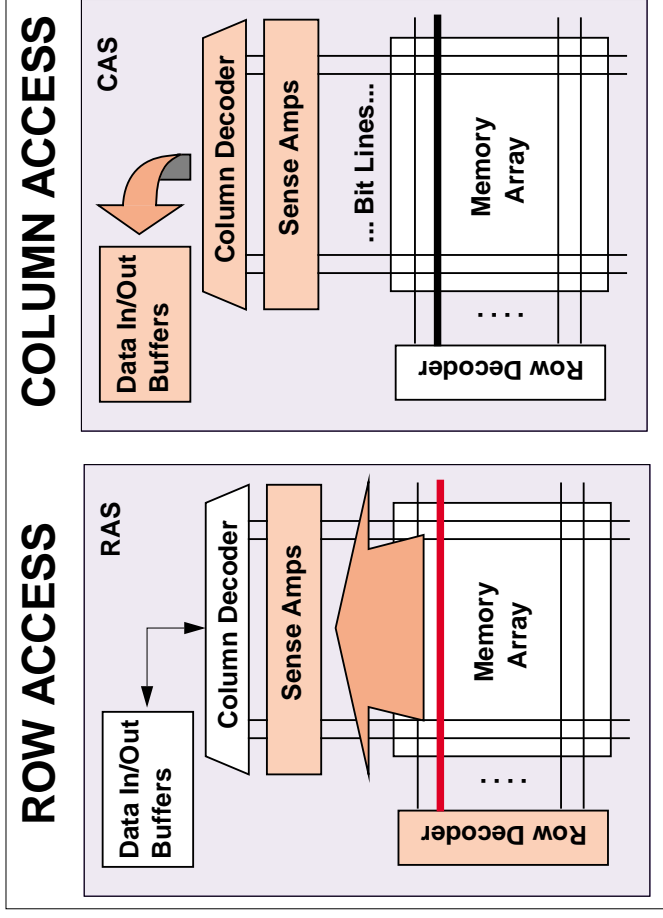
Hits vs. Depth in Victim-Row FIFO Buffer



Row Buffers as L2 Cache



Row Buffer Management



RAS is like Cache Access
Why not Speculate?

Cost-Performance

FPM, EDO, SDRAM, ESDRAM:

- Lower Latency => Wide/Fast Bus
- Increase Capacity => Decrease Latency
- Low System Cost

Rambus, Direct Rambus, SDRAM:

- Lower Latency => Multiple Channels
- Increase Capacity => Increase Capacity
- High System Cost

However, 1 DRDRAM = Multiple SDRAM

Conclusions

100MHz/128-bit Bus is **Current Bottleneck**

- **Solution: Fast Bus/es & MC on CPU**
(e.g. Alpha 21364, Emotion Engine, ...)

Current DRAMs Solving **Bandwidth Problem**
(but **not Latency Problem**)

- **Solution: New cores with on-chip SRAM**
(e.g. ESDRAM, VCDRAM, ...)
- **Solution: New cores with smaller banks**
(e.g. MoSys “SRAM”, FCRAM, ...)

Direct Rambus seems to scale best for future
high-speed CPUs

Outline

- **Basics**
- **DRAM Evolution: Structural Path**
- **Advanced Basics**
- **DRAM Evolution: Interface Path**
- **Future Interface Trends & Research Areas**
- **Performance Modeling: *Systems, Embedded Architectures***

Motivation

Even when we restrict our focus ...

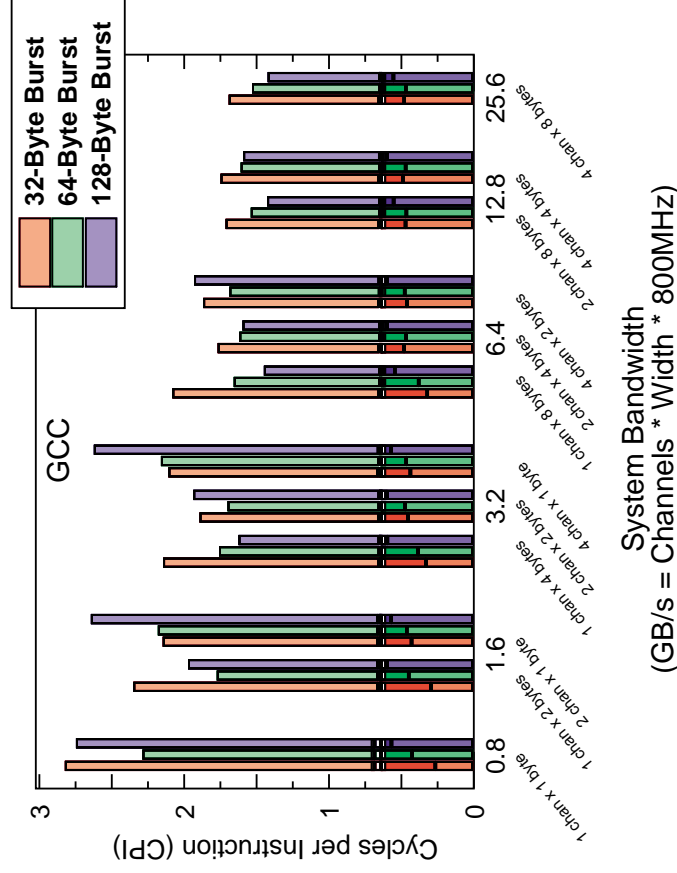
SYSTEM-LEVEL PARAMETERS

- Number of channels Width of channels
- Channel latency Channel bandwidth
- Banks per channel Turnaround time
- Request-queue size Request reordering
- Row-access Column-access
- DRAM precharge CAS-to-CAS latency
- DRAM buffering L2 cache blocksize
- Number of MSHRs Bus protocol

Fully | partially | **not independent (this study)**

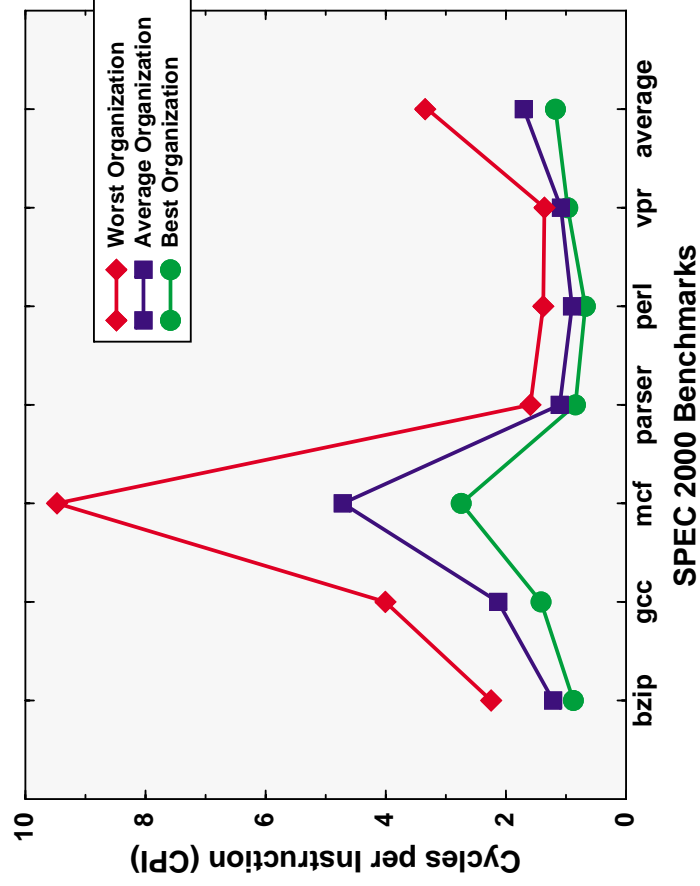
Motivation

... the design space is highly non-linear ...



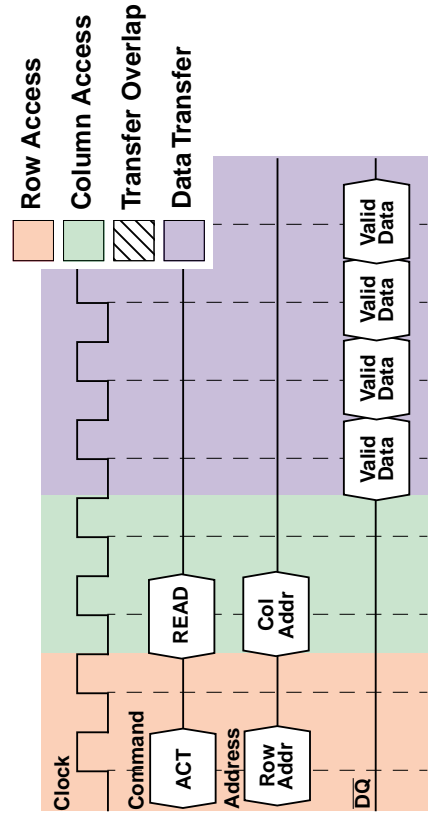
Motivation

... and the cost of poor judgment is high.



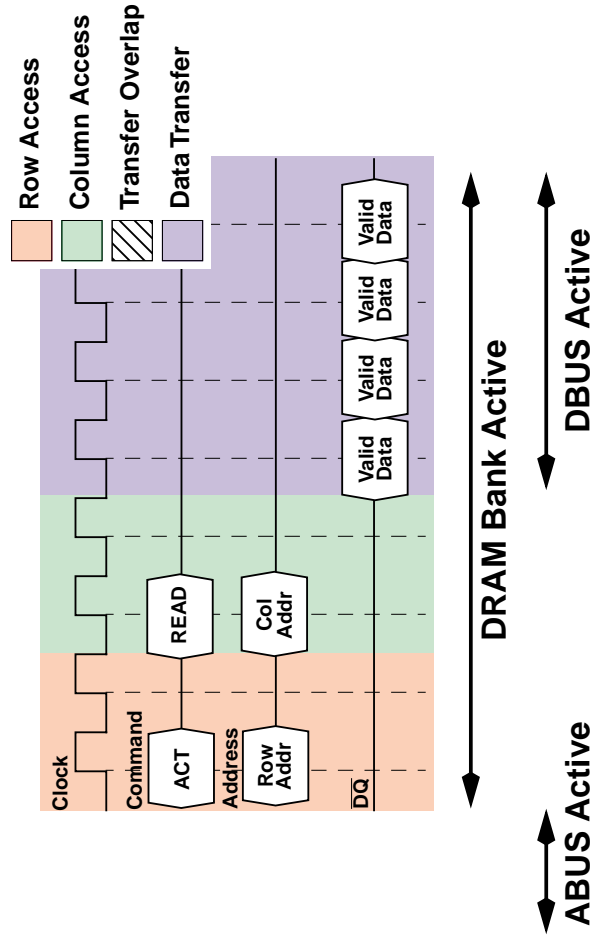
System-Level Model

SDRAM Timing



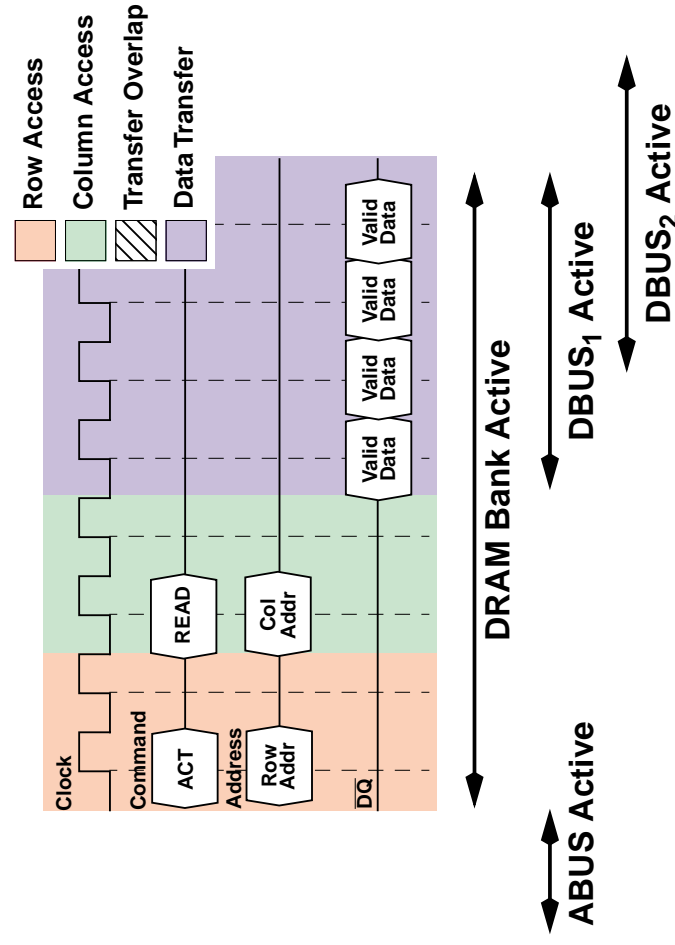
System-Level Model

Timing diagrams are at the DRAM level
... not the system level

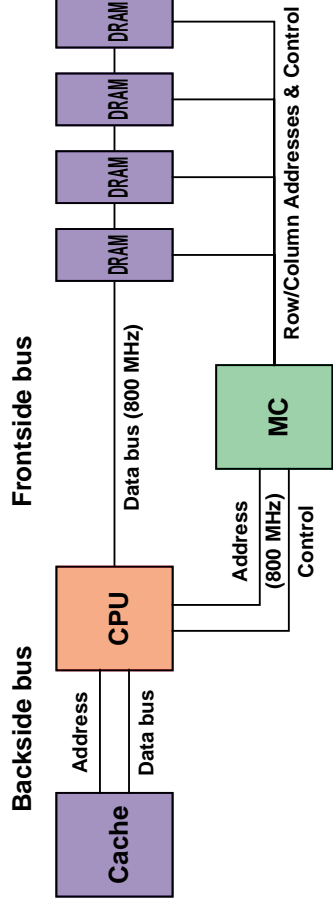


System-Level Model

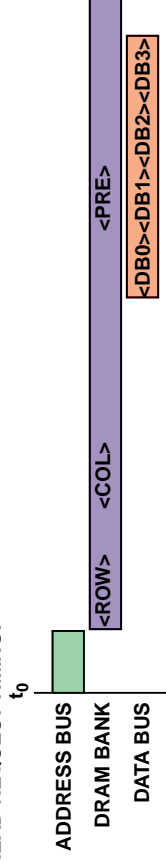
Timing diagrams are at the DRAM level
... not the system level



Request Timing

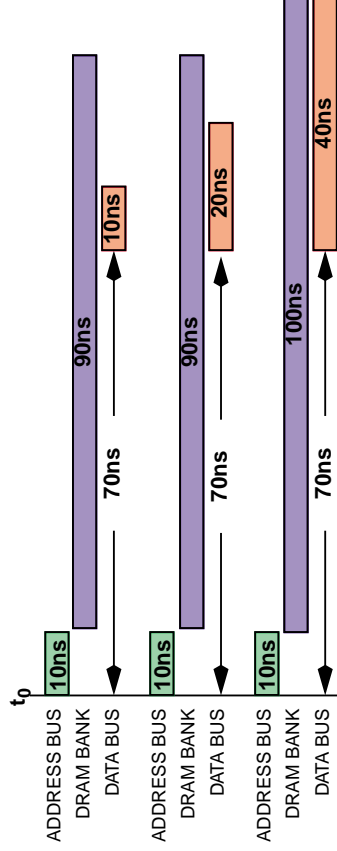


READ REQUEST TIMING:

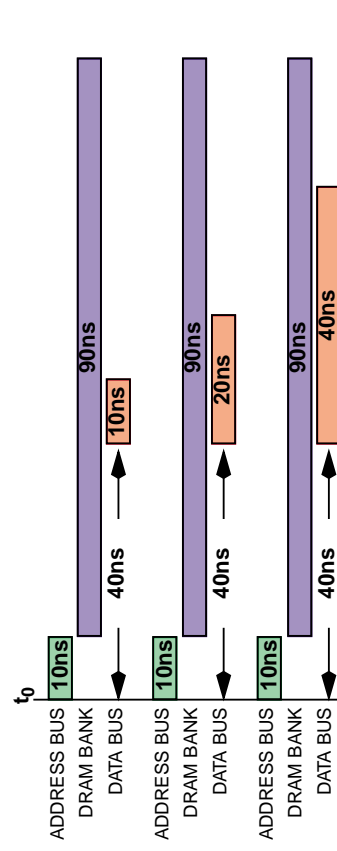


Read/Write Request Shapes

READ REQUESTS:

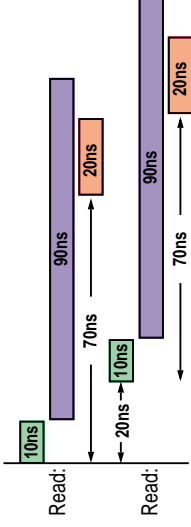


WRITE REQUESTS:



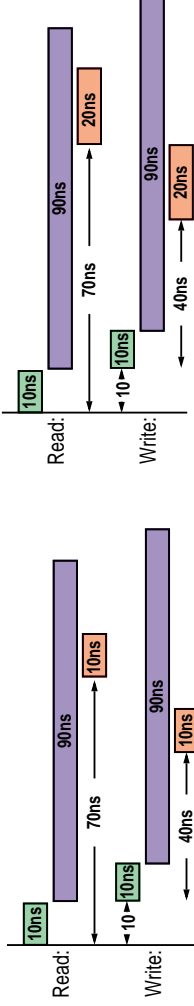
Pipelined/Split Transactions

(a) Legal if R/R to different banks:

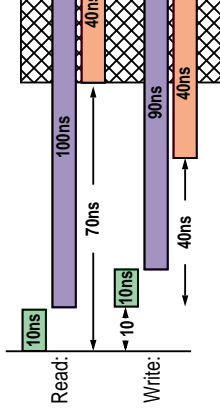


(b) Nesting of writes inside reads is legal if R/W to different banks:

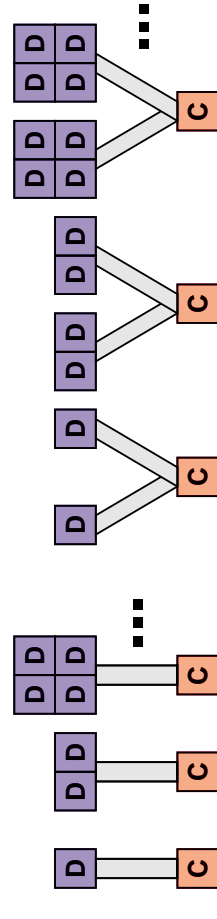
Legal if turnaround $\leq 10\text{ns}$:



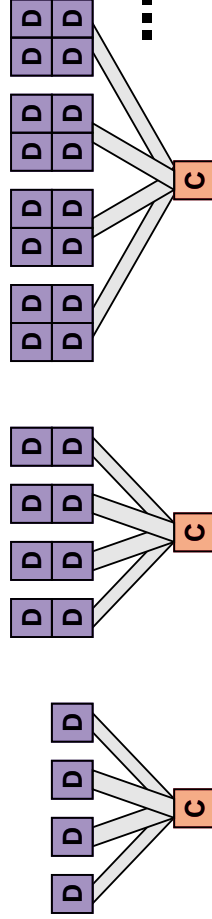
(c) Back-to-back R/W pair that cannot be nested:



Channels & Banks



One independent channel
Banking degrees of 1, 2, 4, ...



Two independent channels
Banking degrees of 1, 2, 4, ...

Four independent channels
Banking degrees of 1, 2, 4, ...

1, 2, 4 800 MHz Channels

8, 16, 32, 64 Data Bits per Channel

1, 2, 4, 8 Banks per Channel (Indep.)

32, 64, 128 Bytes per Burst

Burst Scheduling (Back-to-Back Read Requests)

128-Byte Bursts:



64-Byte Bursts:

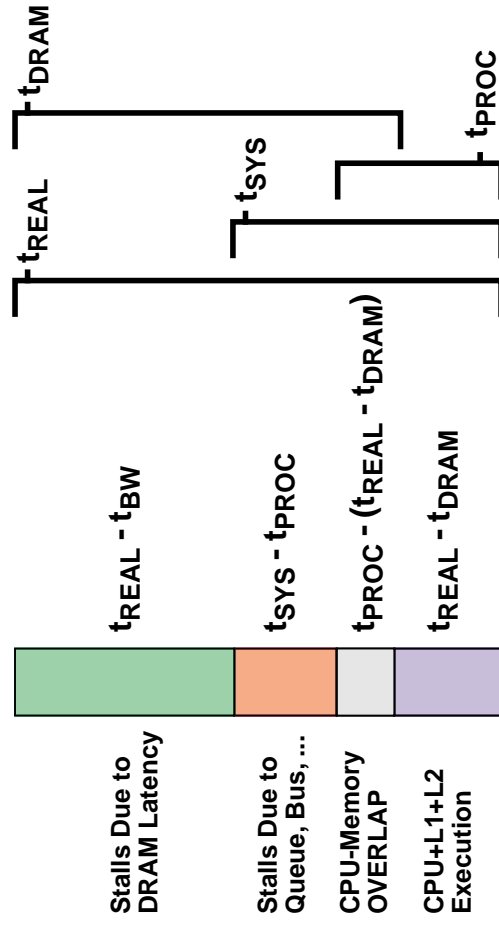


32-Byte Bursts:



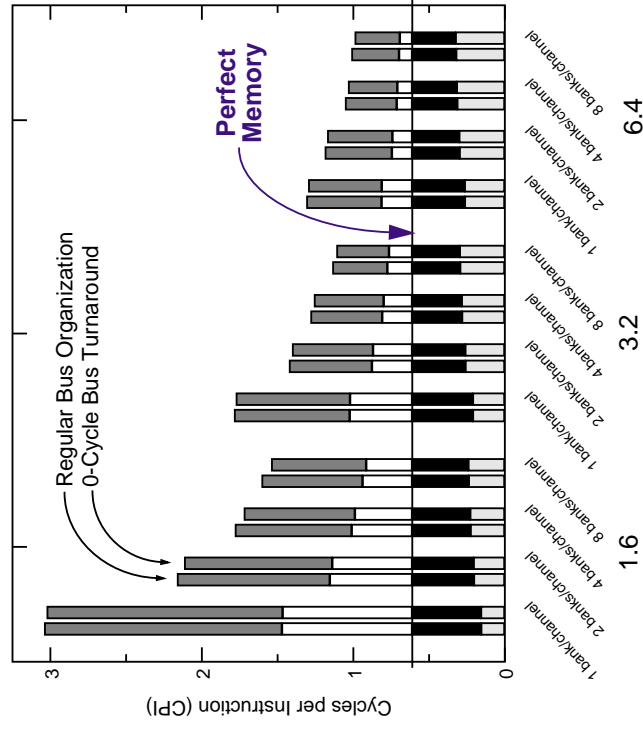
- **Critical-burst-first**
- **Non-critical bursts are promoted**
- **Writes have lowest priority**
(tend back up in request queue ...)
- **Tension between large & small bursts:**
amortization vs. faster time to data

New Bar-Chart Definition



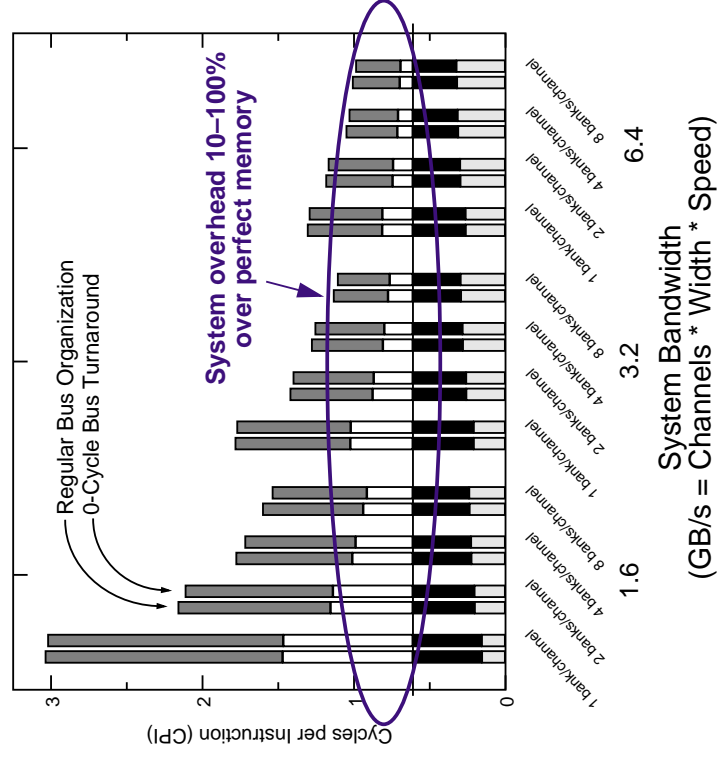
- **t_{PROC} — CPU with 1-cycle L2 miss**
- **t_{REAL} — realistic CPU/DRAM config**
- **t_{SYS} — CPU with 1-cycle DRAM latency**
- **t_{DRAM} — time seen by DRAM system**

System Overhead



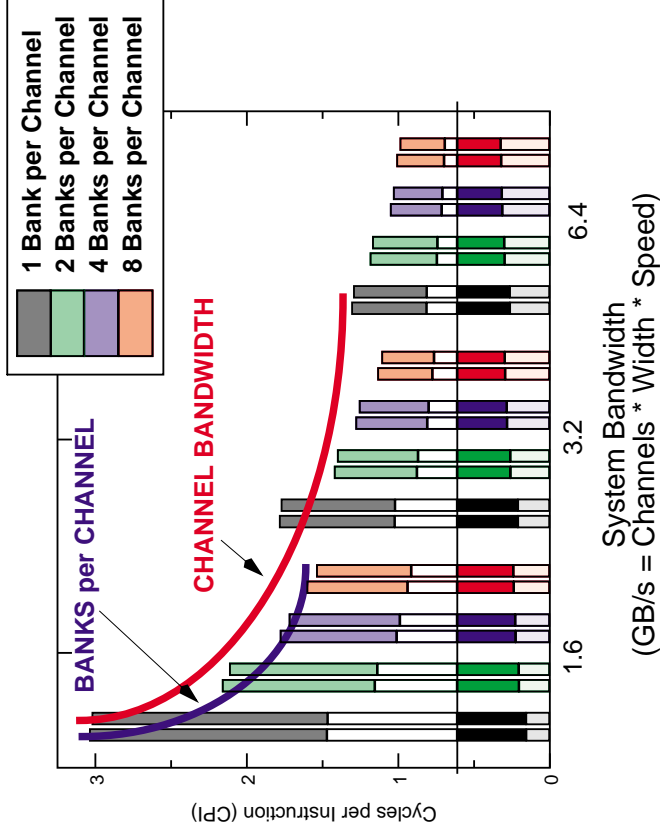
Benchmark = BZIP (SPEC 2000), 32-byte burst, 16-bit bus

System Overhead



Benchmark = BZIP (SPEC 2000), 32-byte burst, 16-bit bus

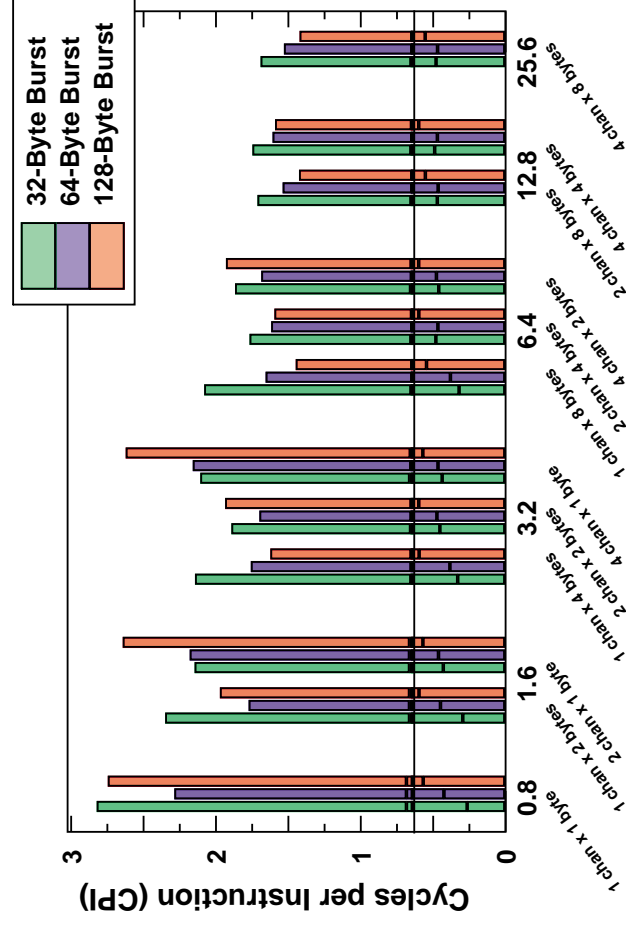
Concurrency Effects



Benchmark = BZIP (SPEC 2000), 32-byte burst, 16-bit bus

Banks/channel as significant as channel BW

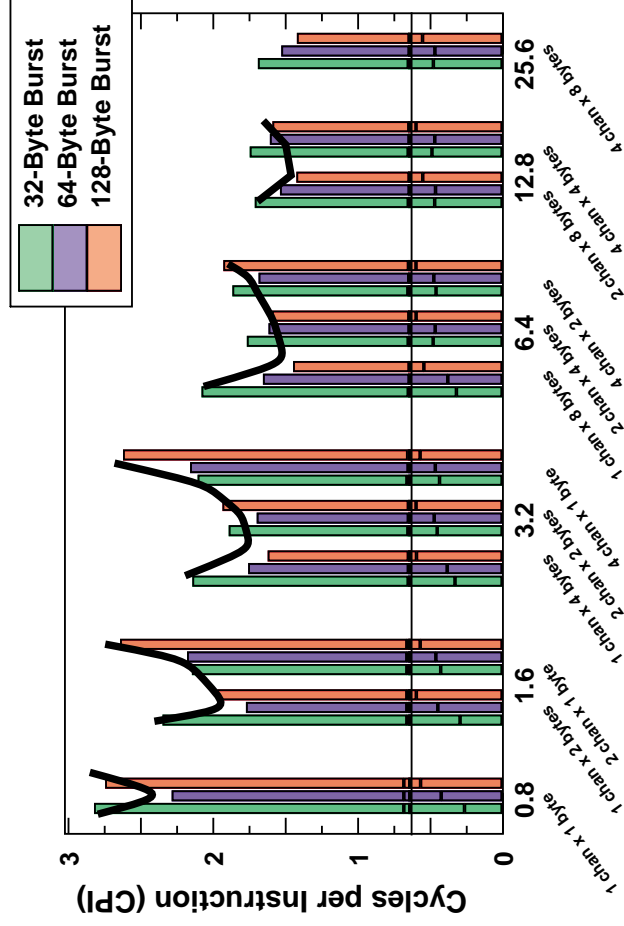
Bandwidth vs. Burst Width



Benchmark = GCC (SPEC 2000), 2 banks/channel

Bandwidth vs. Burst Width

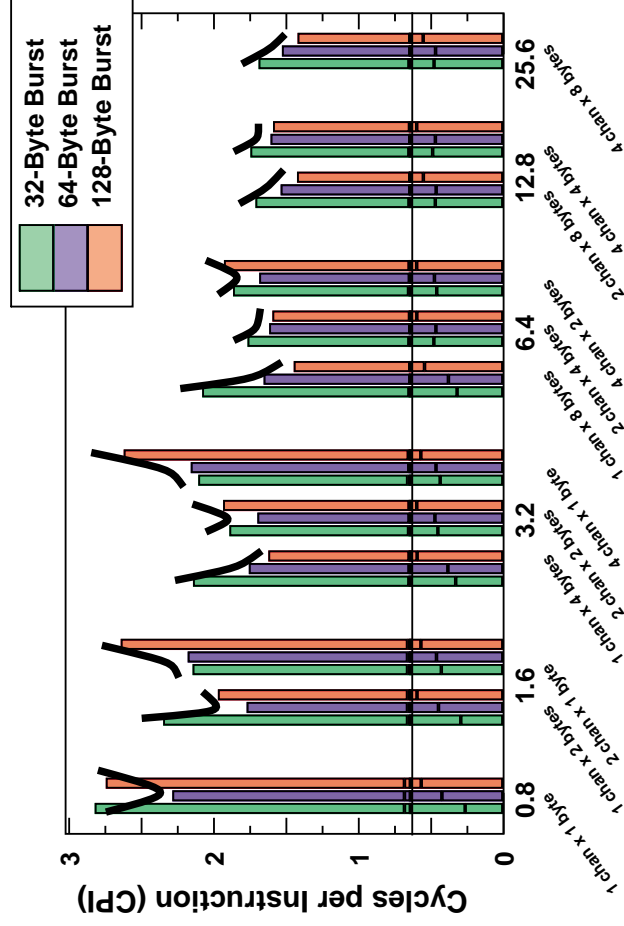
Bandwidth vs. Burst Width



System Bandwidth
(GB/s = Channels * Width * 800MHz)

Benchmark = GCC (SPEC 2000), 2 banks/channel

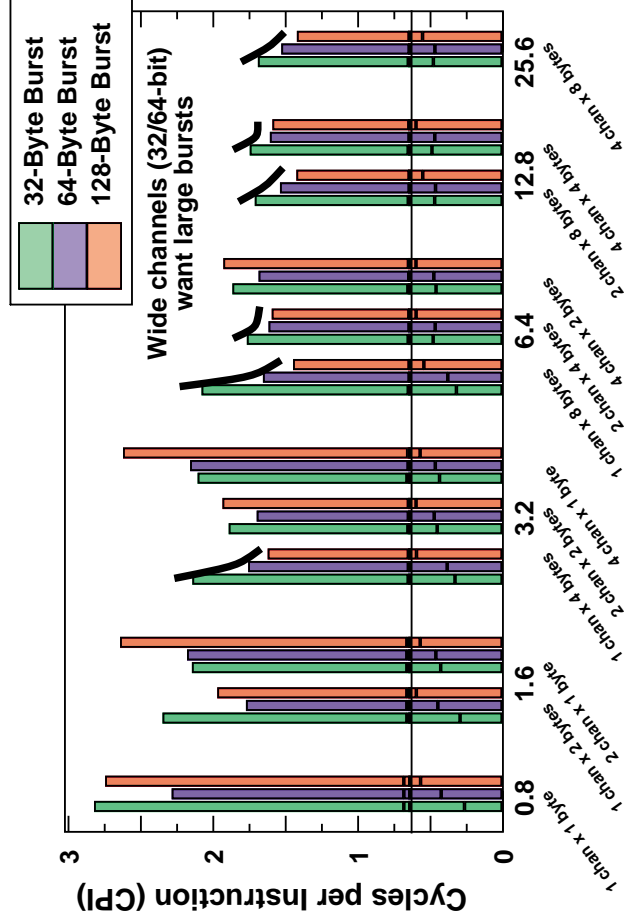
Bandwidth vs. Burst Width



System Bandwidth
(GB/s = Channels * Width * 800MHz)

Benchmark = GCC (SPEC 2000), 2 banks/channel

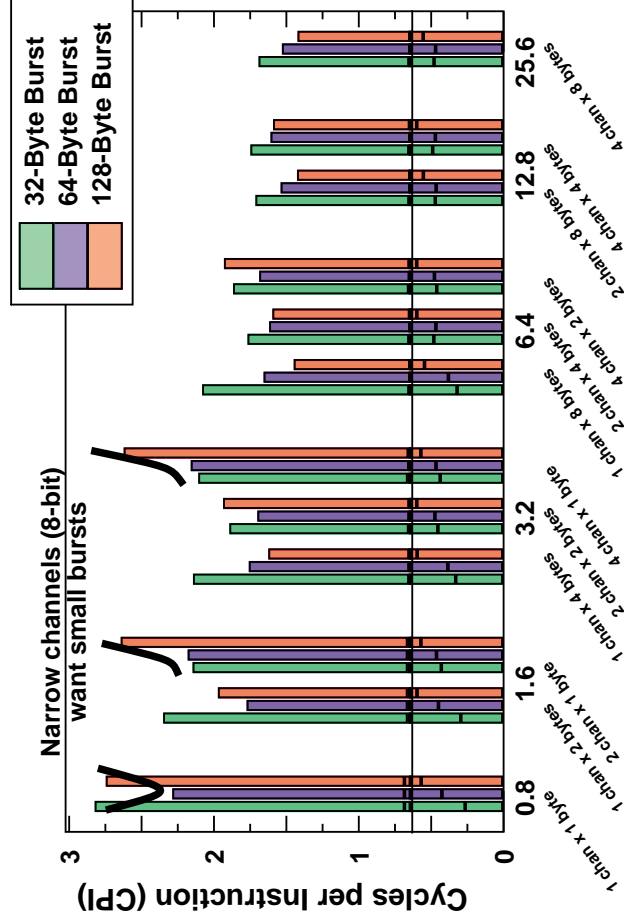
Bandwidth vs. Burst Width



System Bandwidth
(GB/s = Channels * Width * 800MHz)

Benchmark = GCC (SPEC 2000), 2 banks/channel

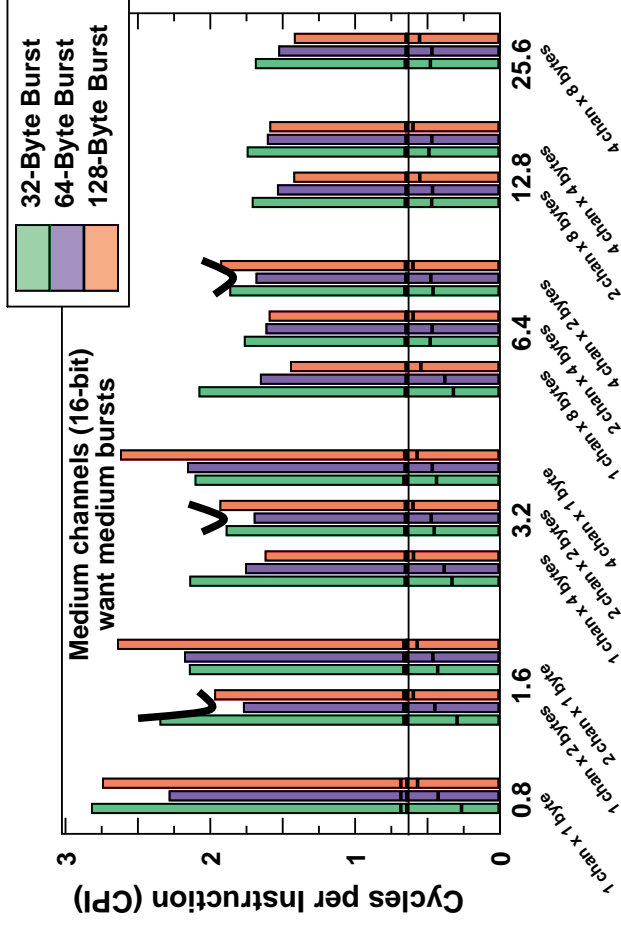
Bandwidth vs. Burst Width



System Bandwidth
(GB/s = Channels * Width * 800MHz)

Benchmark = GCC (SPEC 2000), 2 banks/channel

Bandwidth vs. Burst Width

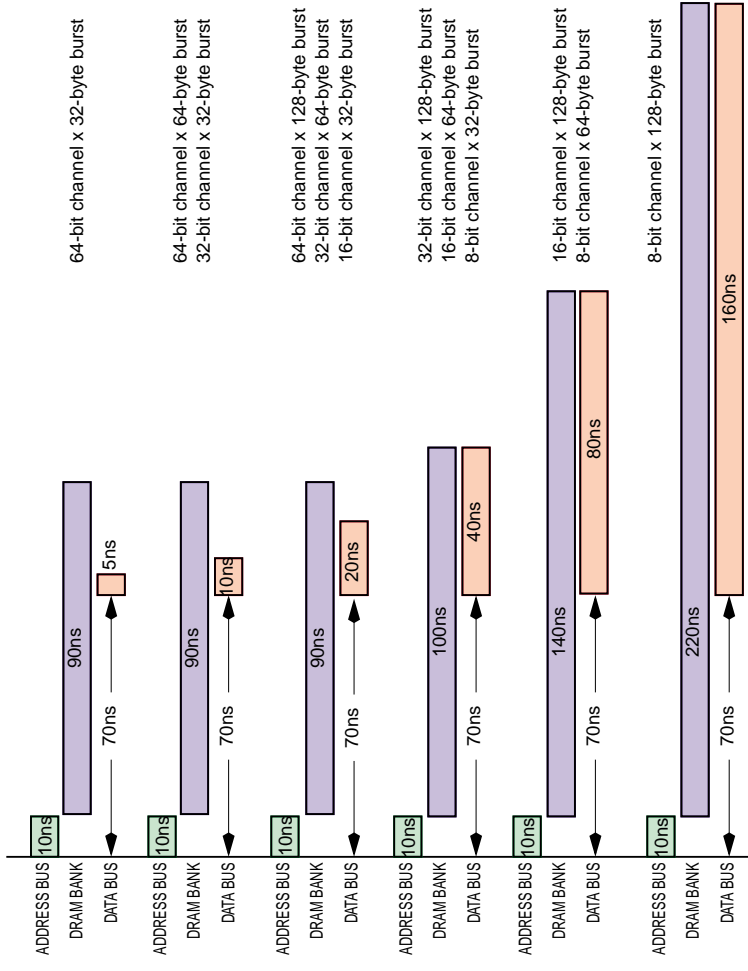


System Bandwidth
(GB/s = Channels * Width * 800MHz)

Benchmark = GCC (SPEC 2000), 2 banks/channel

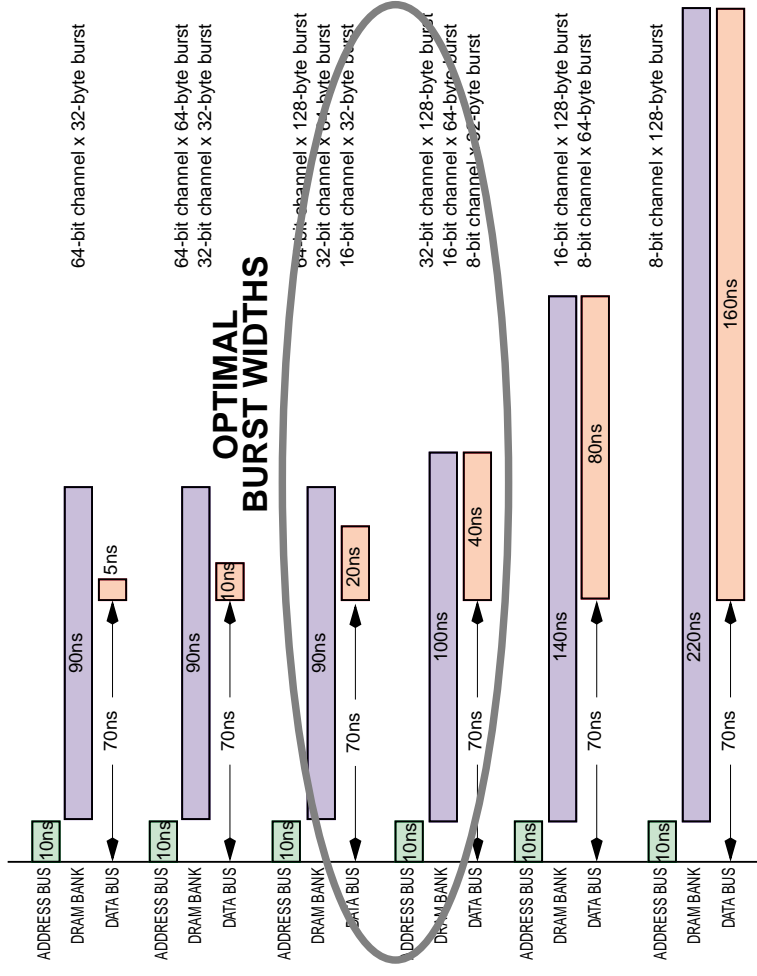
Burst Width Scales with Bus

Range of Burst-Widths Modeled

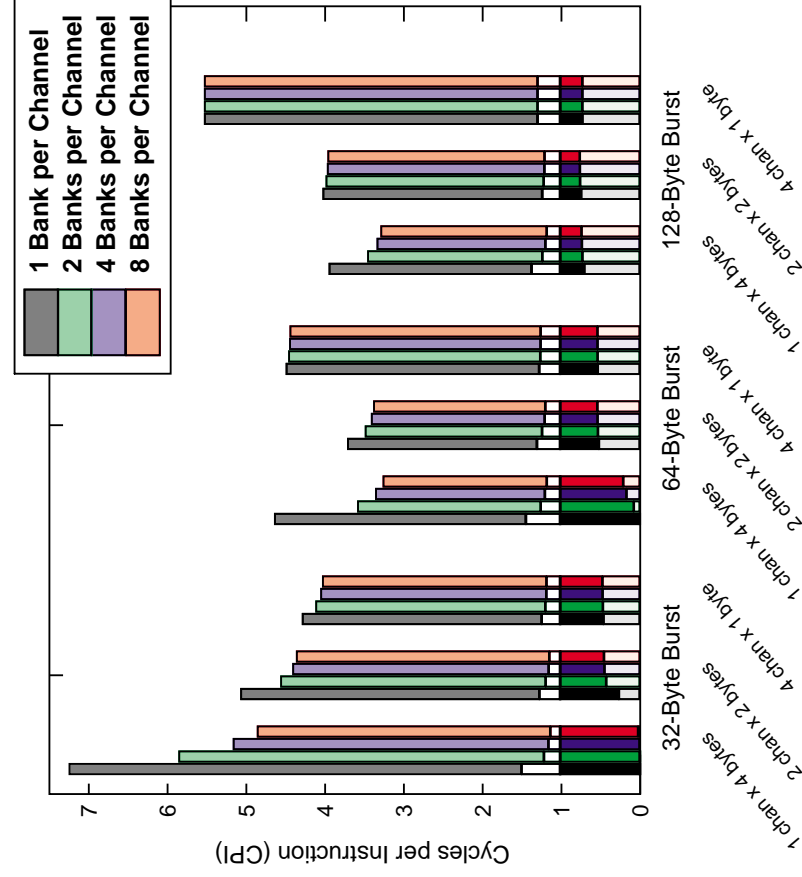


Burst Width Scales with Bus

Range of Burst-Widths Modeled

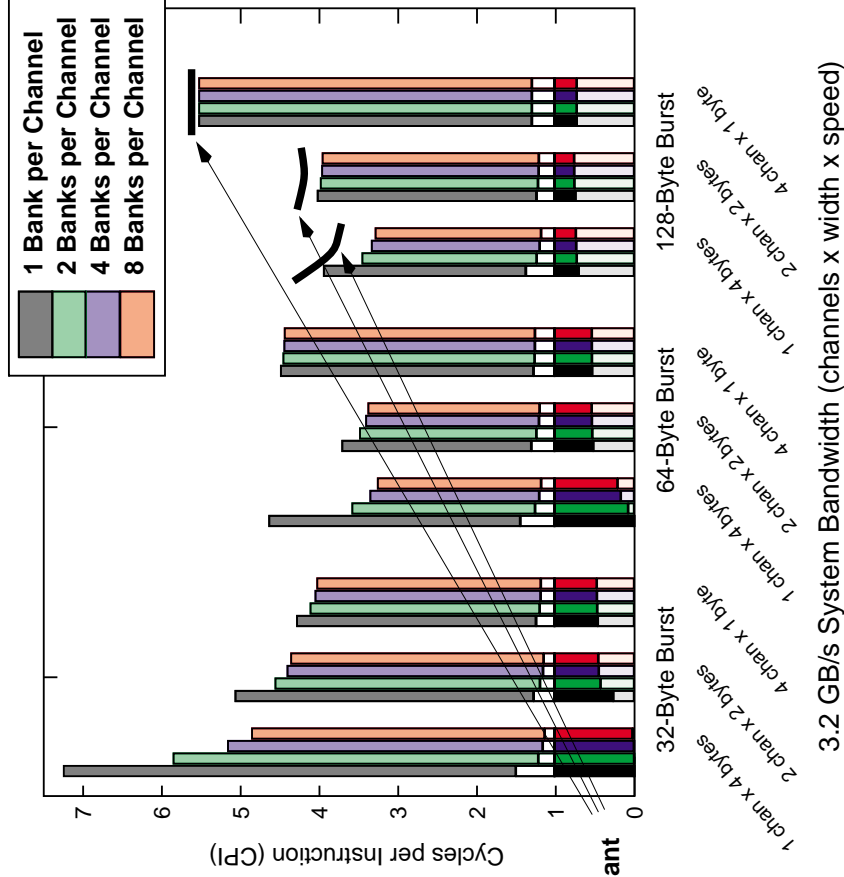


Focus on 3.2 GB/s — MCF



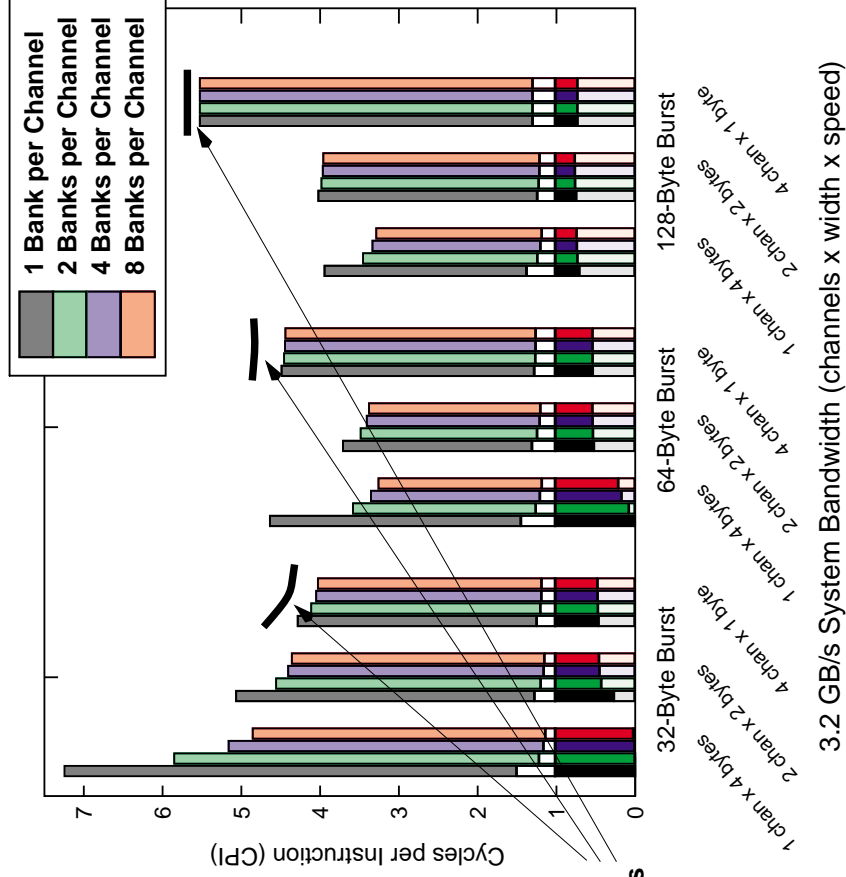
3.2 GB/s System Bandwidth (channels x width x speed)

Focus on 3.2 GB/s — MCF



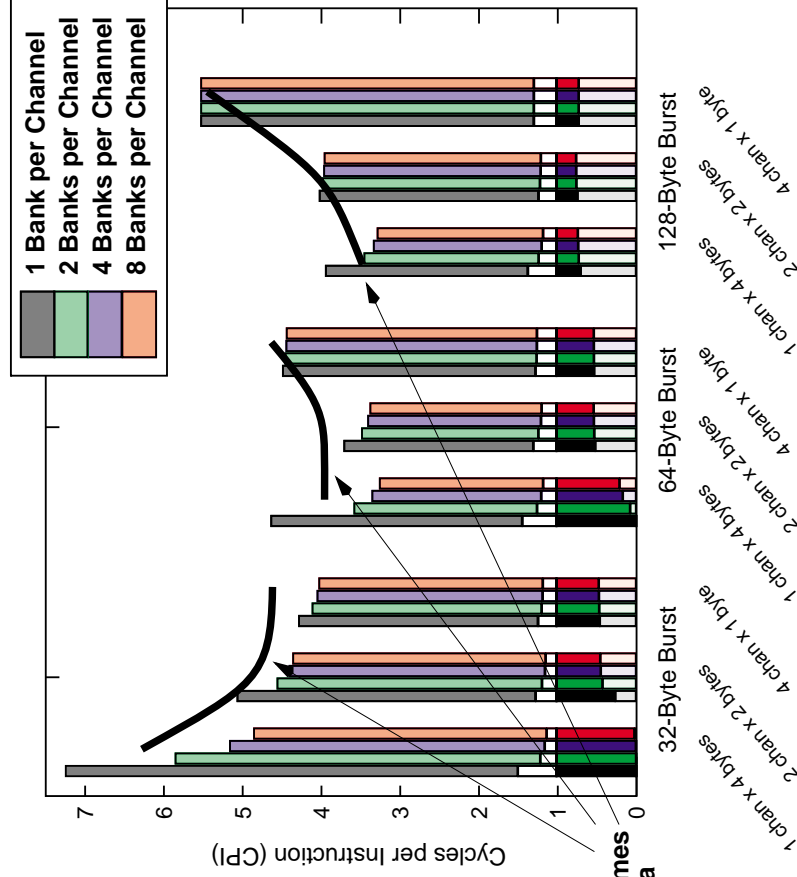
#Banks not particularly important given large burst sizes ...

Focus on 3.2 GB/s — MCF



... even less so with multi-channel systems

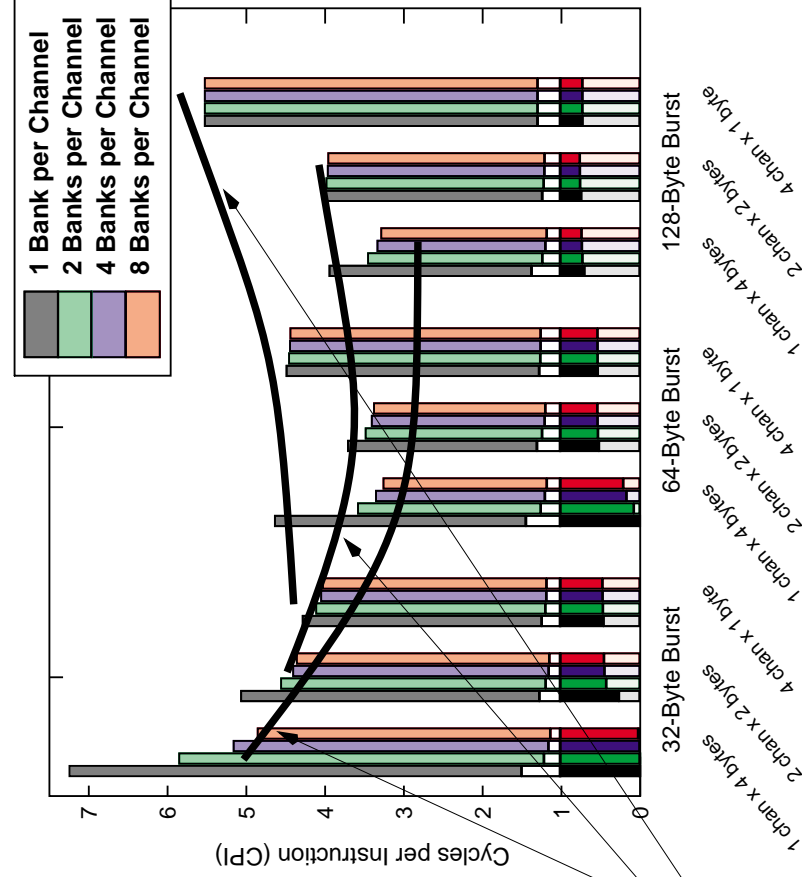
Focus on 3.2 GB/s — MCF



Multi-channel systems sometimes (but not always) a good idea

3.2 GB/s System Bandwidth (channels x width x speed)

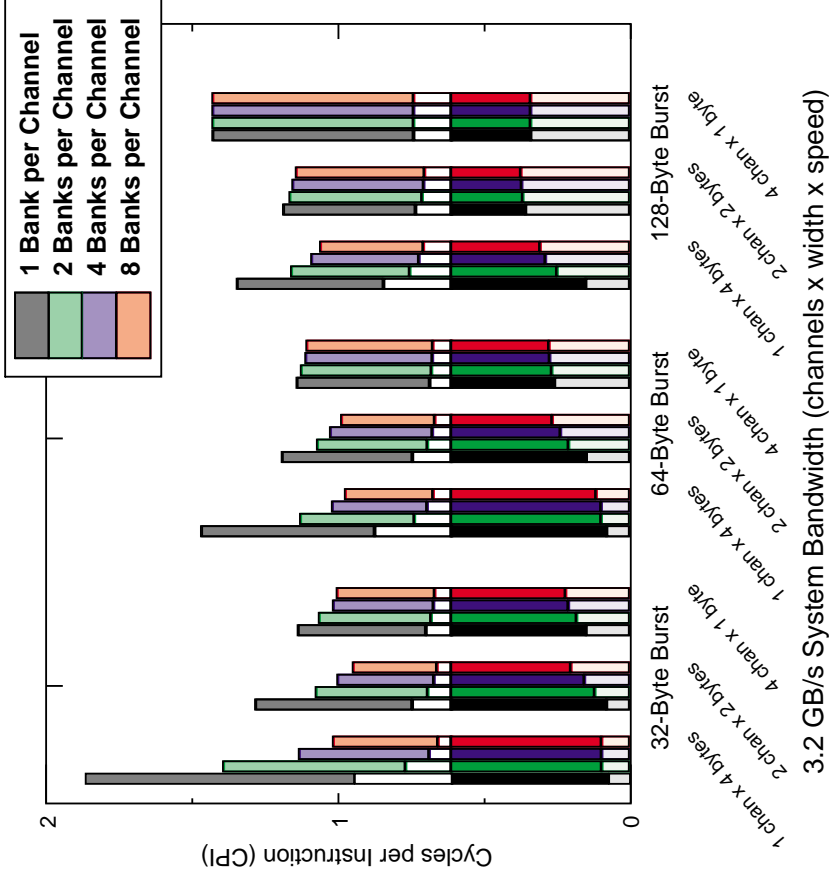
Focus on 3.2 GB/s — MCF



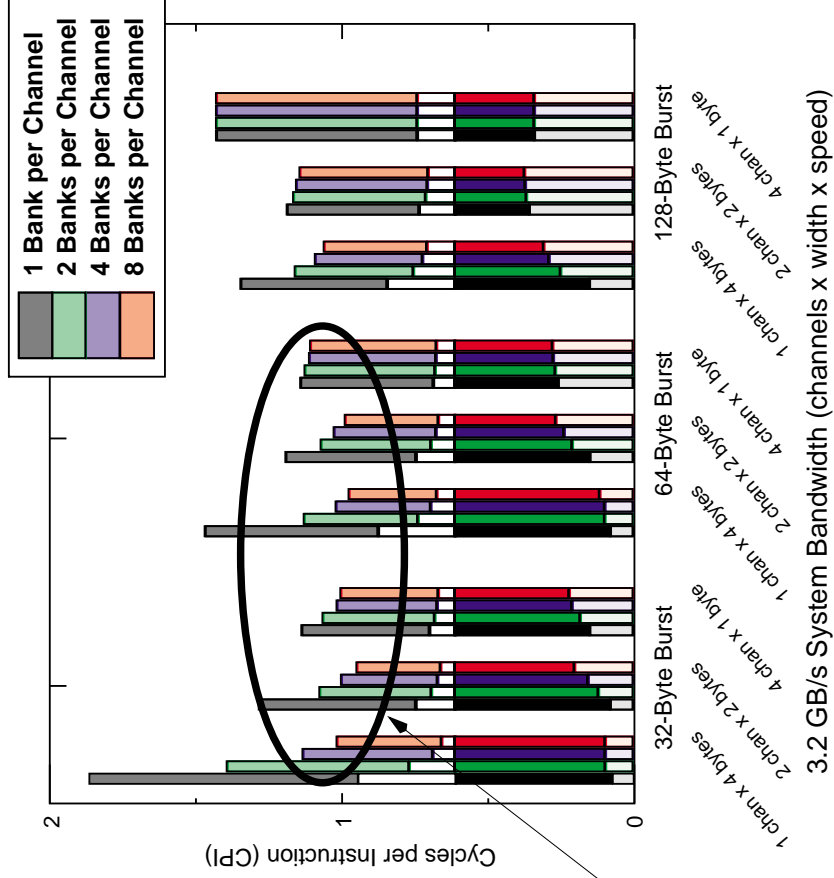
4x 1-byte channels
2x 2-byte channels
1x 4-byte channels

3.2 GB/s System Bandwidth (channels x width x speed)

Focus on 3.2 GB/s — BZIP



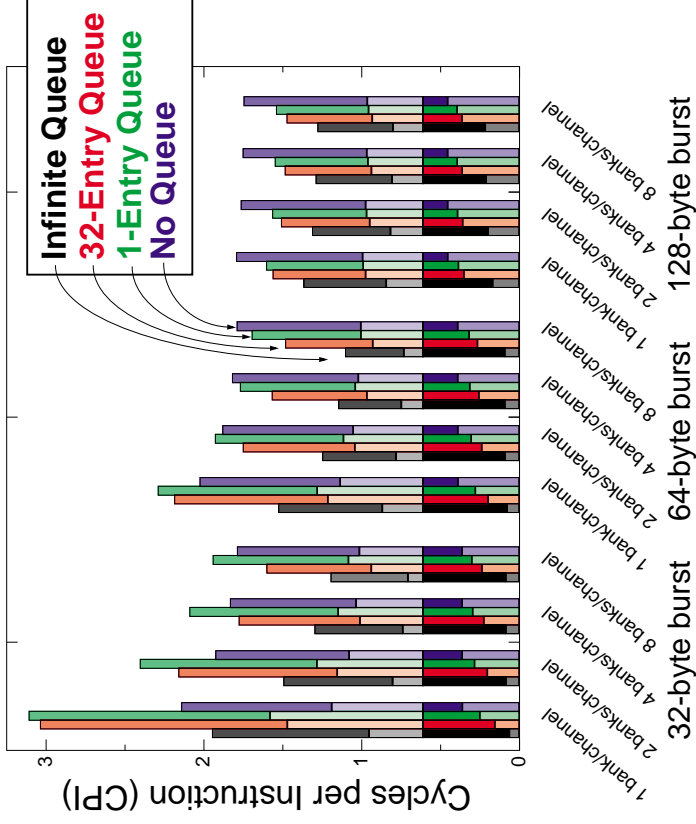
Focus on 3.2 GB/s — BZIP



**BEST CONFIGS are at
SMALLER BURST SIZES**

Queue Size & Reordering

BZIP: 1.6 GB/s (1 channel)



Conclusions

**DESIGN SPACE is NON-LINEAR,
COST of MISJUDGING is HIGH**

CAREFUL TUNING YIELDS 30–40% GAIN

**MORE CONCURRENCY == BETTER
(but not at expense of LATENCY)**

- Via **Channels** → **NOT w/ LARGE BURSTS**
- Via **Banks** → **ALWAYS SAFE**
- Via **Bursts** → **DOESN'T PAY OFF**
- Via **MSHRs** → **NECESSARY**

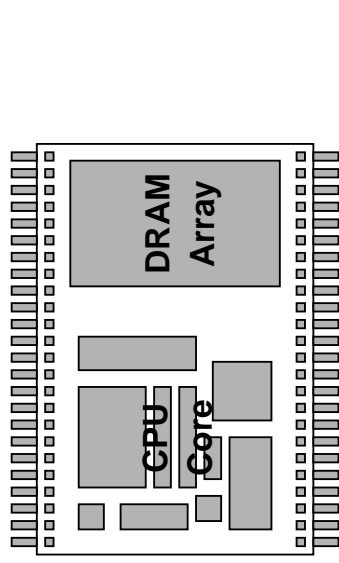
BURSTS AMORTIZE COST OF PRECHARGE

- Typical Systems: 32 bytes (even DDR2)
→ **THIS IS NOT ENOUGH**

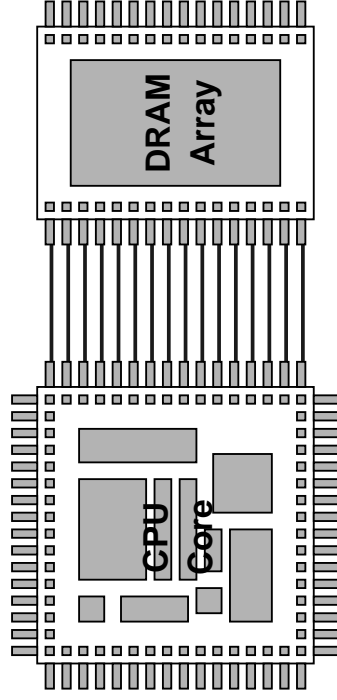
Outline

- Basics
- DRAM Evolution: *Structural Path*
- Advanced Basics
- DRAM Evolution: *Interface Path*
- Future Interface Trends & Research Areas
- Performance Modeling: *Architectures, Systems, Embedded*

Embedded DRAM Primer



Embedded



Not Embedded

Whither Embedded DRAM?

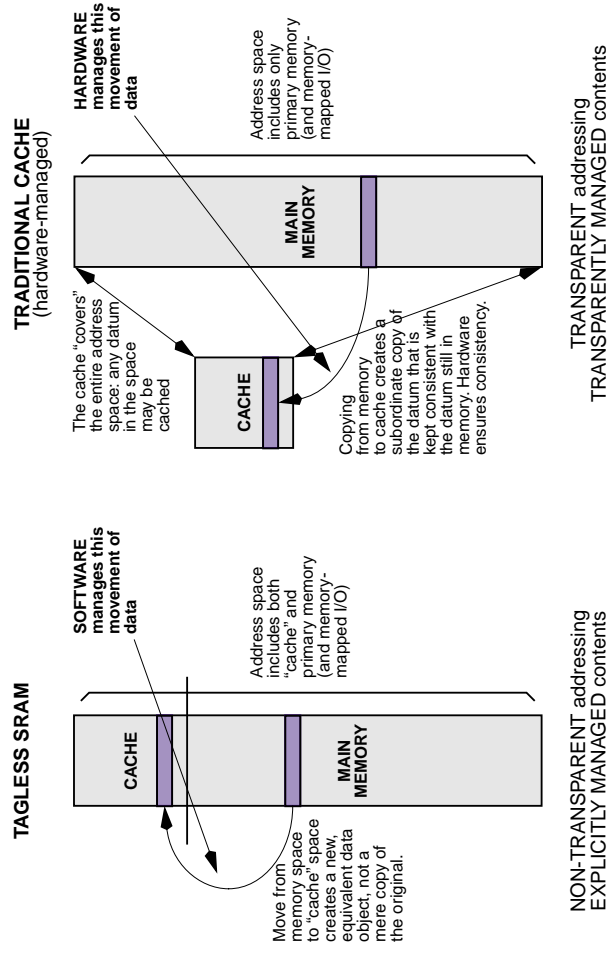
Microprocessor Report, August 1996: “[Five] Architects Look to Processors of Future”

- Two predict imminent merger of CPU and DRAM
- Another states we cannot keep cramming more data over the pins at faster rates (implication: embedded DRAM)
- A fourth wants gigantic on-chip L3 cache (perhaps DRAM L3 implementation?)

SO WHAT HAPPENED?

Embedded DRAM for DSPs

MOTIVATION

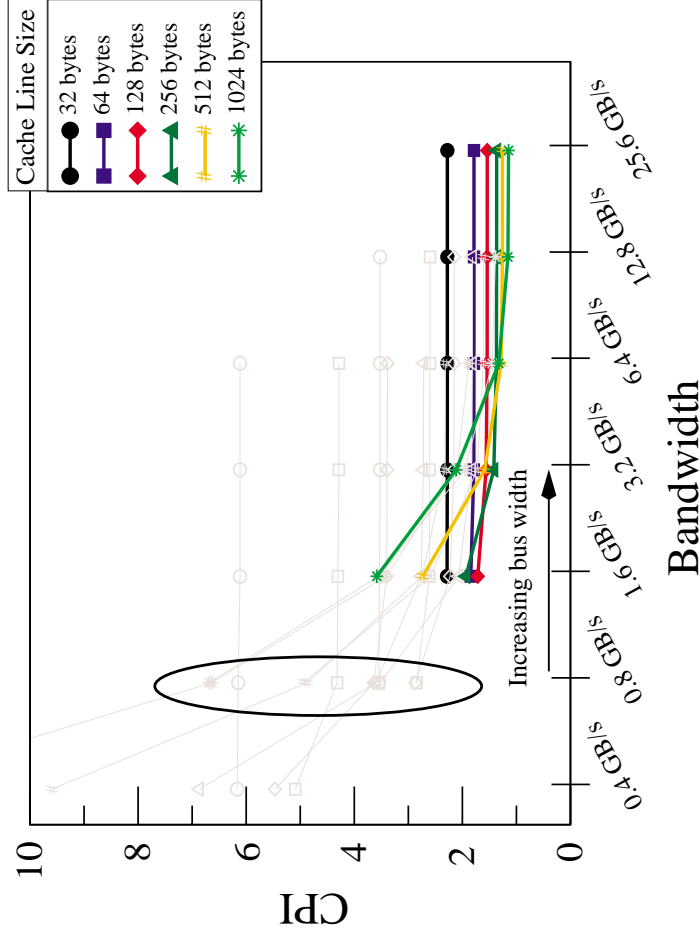


DSP Compilers => Transparent Cache Model

E-DRAM Performance

Embedded Networking Benchmark - Patricia

200MHz C6000 DSP: 200MHz Memory



Performance-Data Sources

“A Performance Study of Contemporary DRAM Architectures,”
Proc. ISCA '99. V. Cuppu, B. Jacob, B. Davis, and T. Mudge.

“Organizational Design Trade-Offs at the DRAM, Memory Bus, and Memory Controller Level: Initial Results,” University of Maryland Technical Report UMD-SCA-TR-1999-2. V. Cuppu and B. Jacob.

“DDR2 and Low Latency Variants,” *Memory Wall Workshop 2000*, in conjunction w/ ISCA '00. B. Davis, T. Mudge, V. Cuppu, and B. Jacob.

“Concurrency, Latency, or System Overhead: Which Has the Largest Impact on DRAM-System Performance?”
Proc. ISCA '01. V. Cuppu and B. Jacob.

“Transparent Data-Memory Organizations for Digital Signal Processors,”
Proc. CASES '01. S. Srinivasan, V. Cuppu, and B. Jacob.

“High Performance DRAMs in Workstation Environments,”
IEEE Transactions on Computers, November 2001.
V. Cuppu, B. Jacob, B. Davis, and T. Mudge.

Recent experiments by Sadagopan Srinivasan, Ph.D. student at University of Maryland.

Acknowledgments

The preceding work was supported in part by the following sources:

- NSF CAREER Award CCR-9983618
- NSF grant EIA-9806645
- NSF grant EIA-0000439
- DOD award AFOSR-F496200110374
- ... and by Compaq and IBM.

CONTACT INFO

Bruce Jacob

**Electrical & Computer Engineering
University of Maryland, College Park**

<http://www.ece.umd.edu/~blj/>
blj@eng.umd.edu

Dave Wang

**Electrical & Computer Engineering
University of Maryland, College Park**

<http://www.wam.umd.edu/~davewang/>
davewang@wam.umd.edu

