

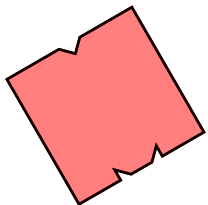
A Holistic Approach to DRAM

Prof. Bruce Jacob

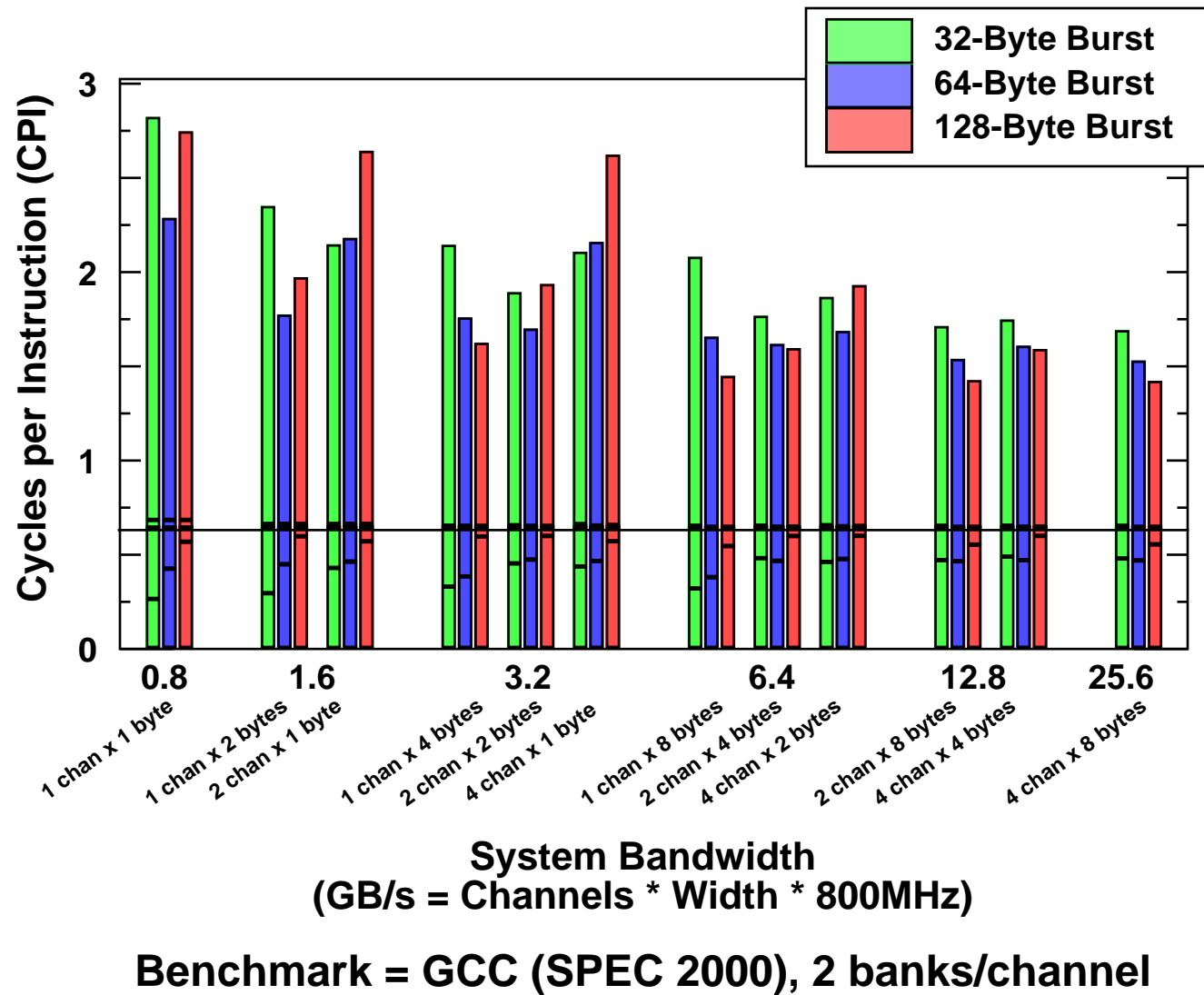
**Electrical & Computer Engineering
University of Maryland, College Park**

OUTLINE

- **Anecdotes, Vision**
- **Our Past & Present Work**
- **Anecdotes Revisited**
- **Conclusions**

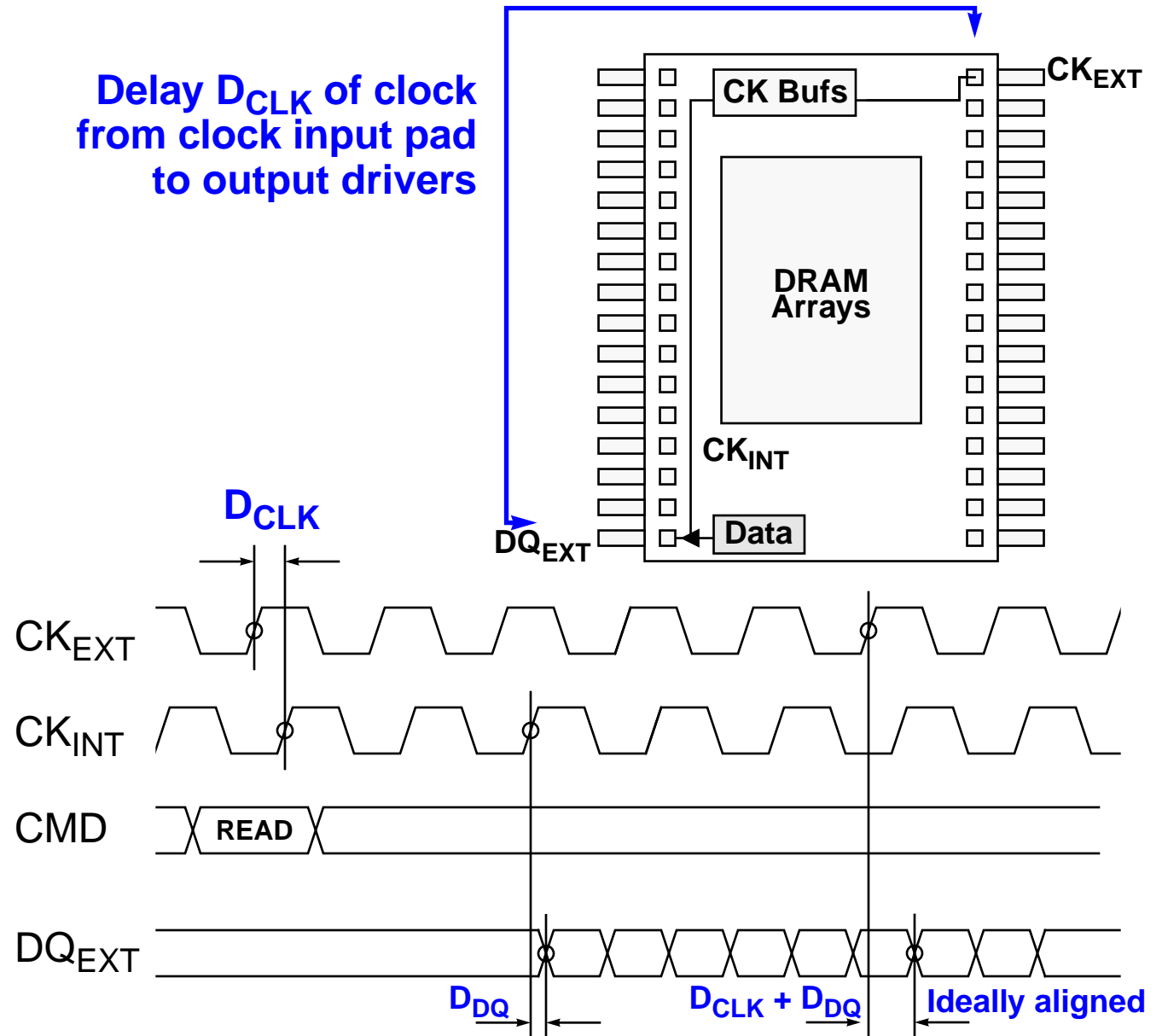


Anecdote I: System Issues



Anecdote II: DDR's DLL

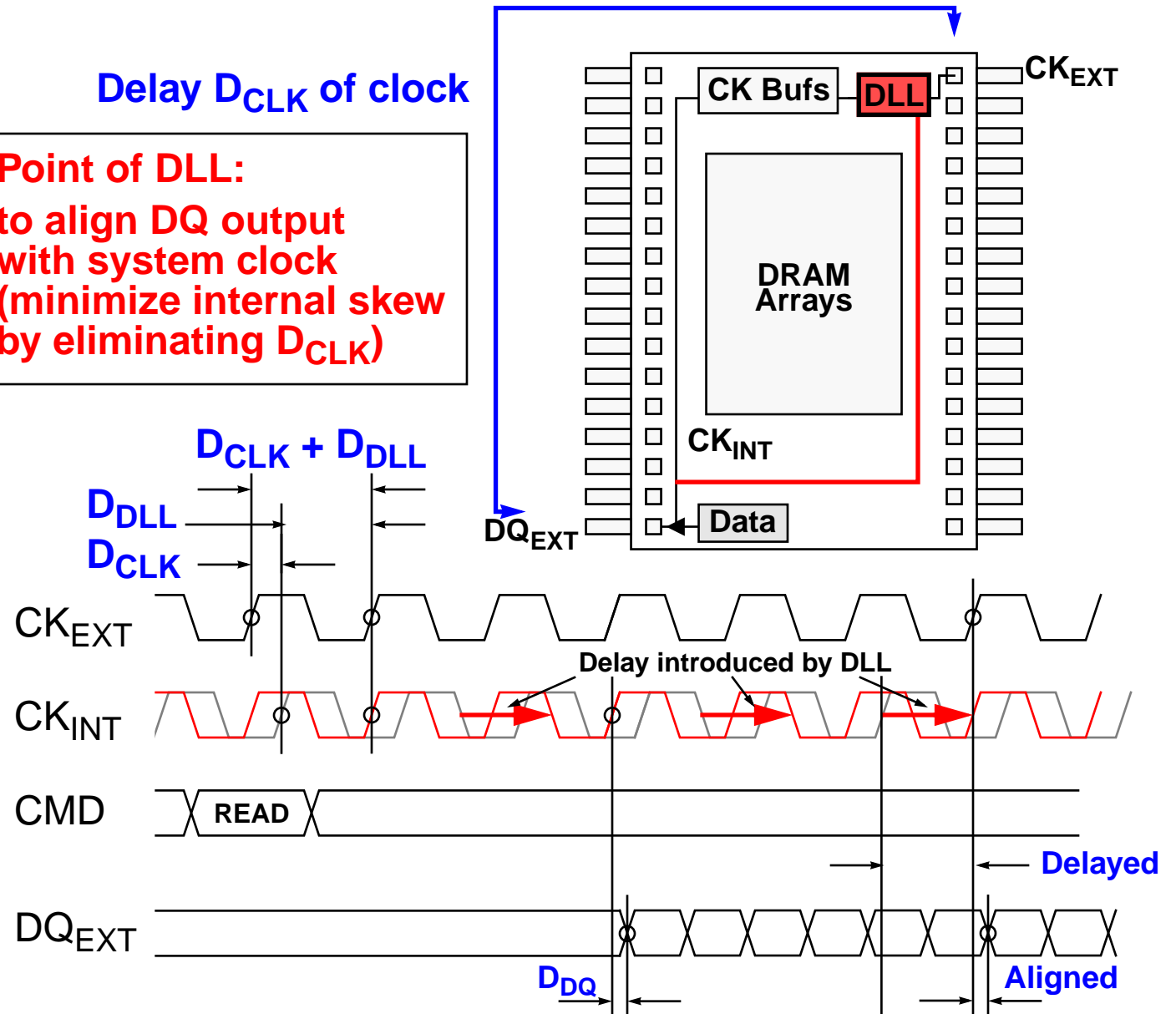
Delay D_{CLK} of clock
from clock input pad
to output drivers



Anecdote II: DDR's DLL

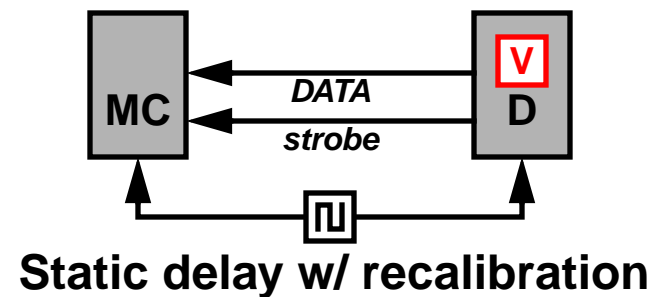
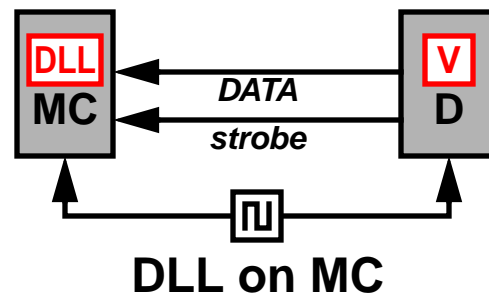
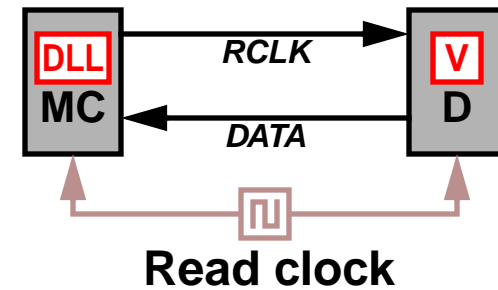
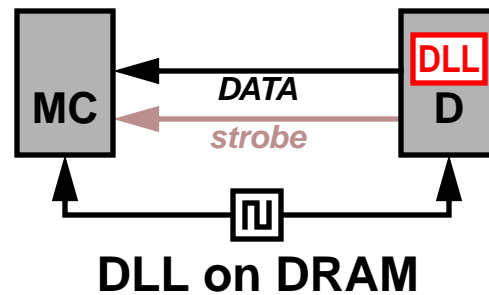
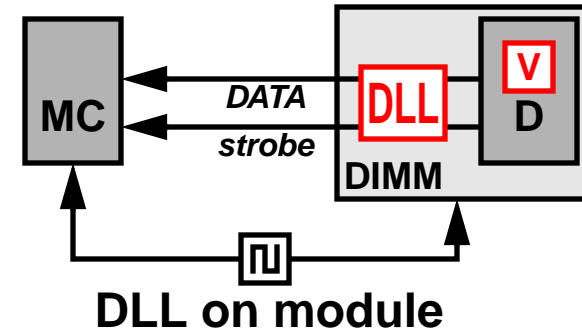
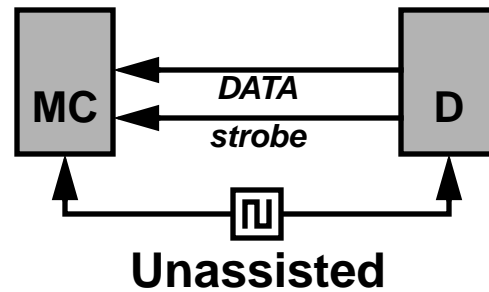
Delay D_{CLK} of clock

Point of DLL:
to align **DQ** output
with system clock
(minimize internal skew
by eliminating D_{CLK})



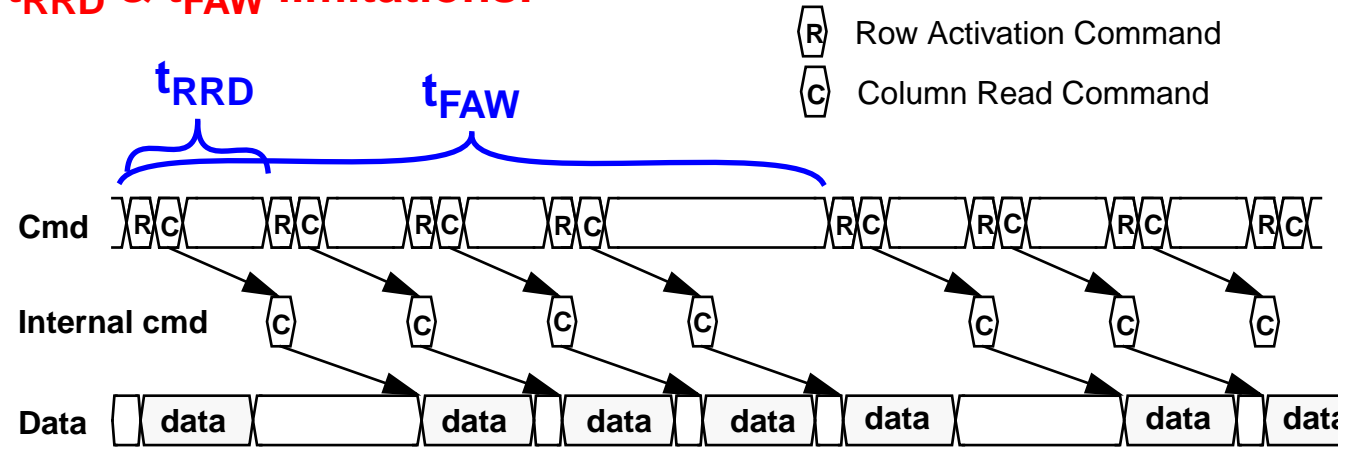
Anecdote II: DDR's DLL

A handful of alternatives:

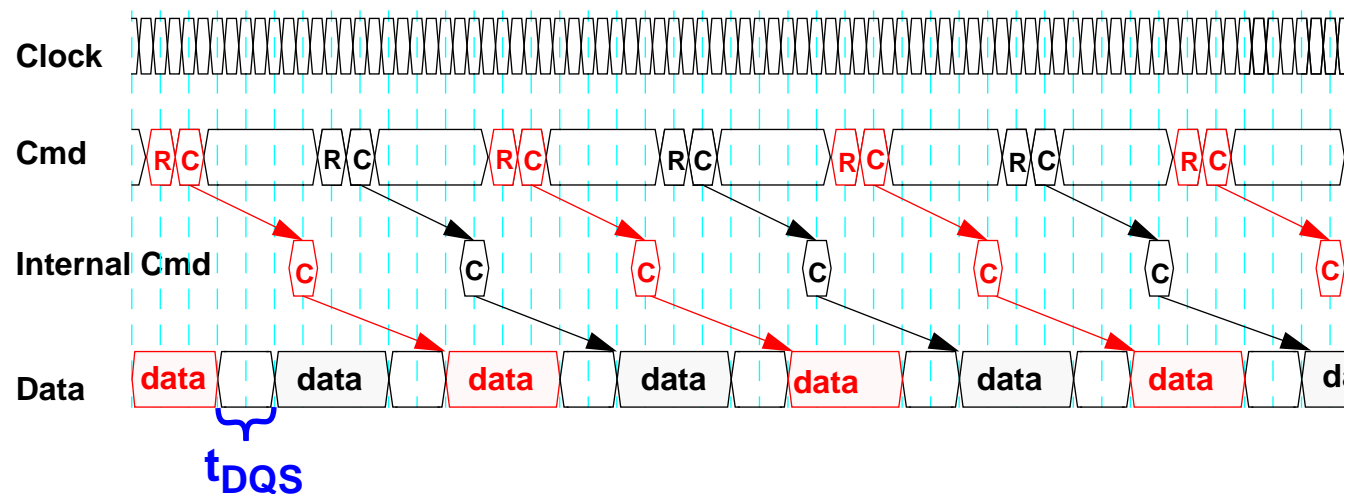


Anecdote III: Circuit v System

t_{RRD} & t_{FAW} limitations:



t_{DQS} limitations:



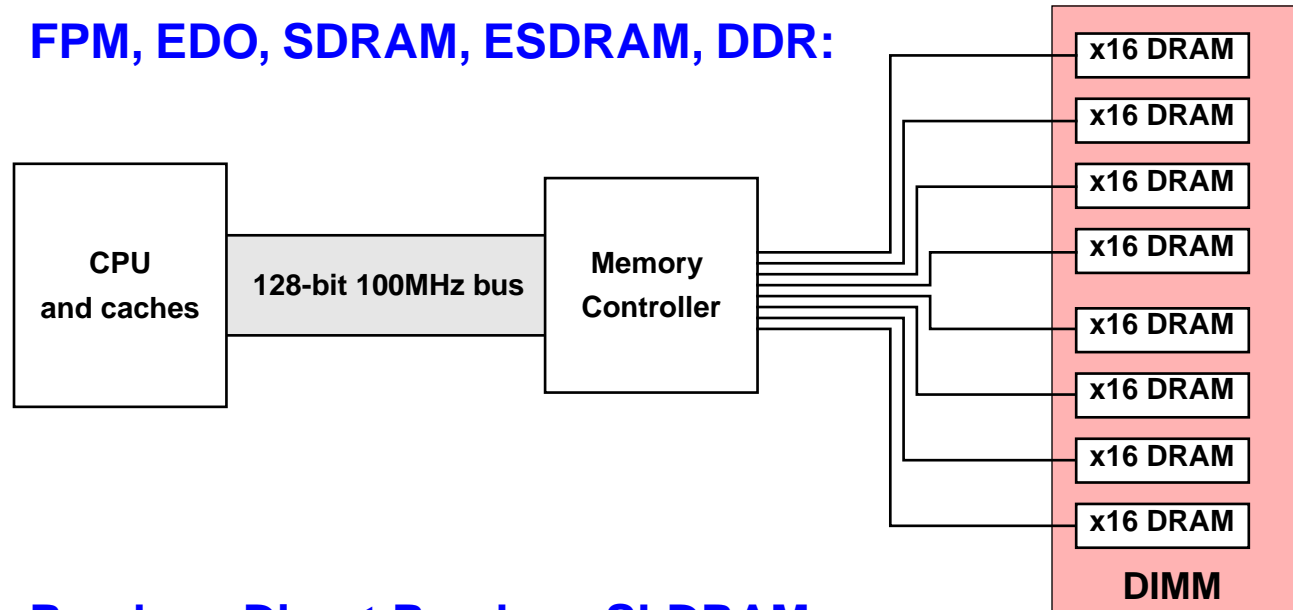
Vision

Must make **circuit-level decisions**
considering **system-level ramifications**

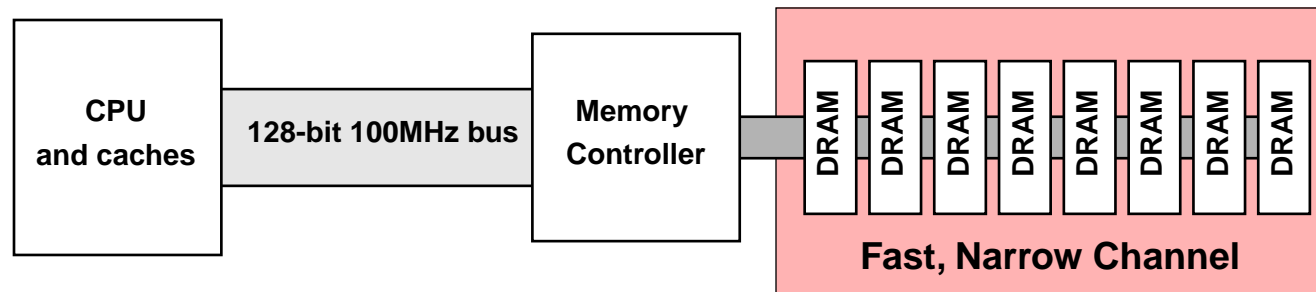
(holistic approach)

Past Work: Device-Level

FPM, EDO, SDRAM, ESDRAM, DDR:

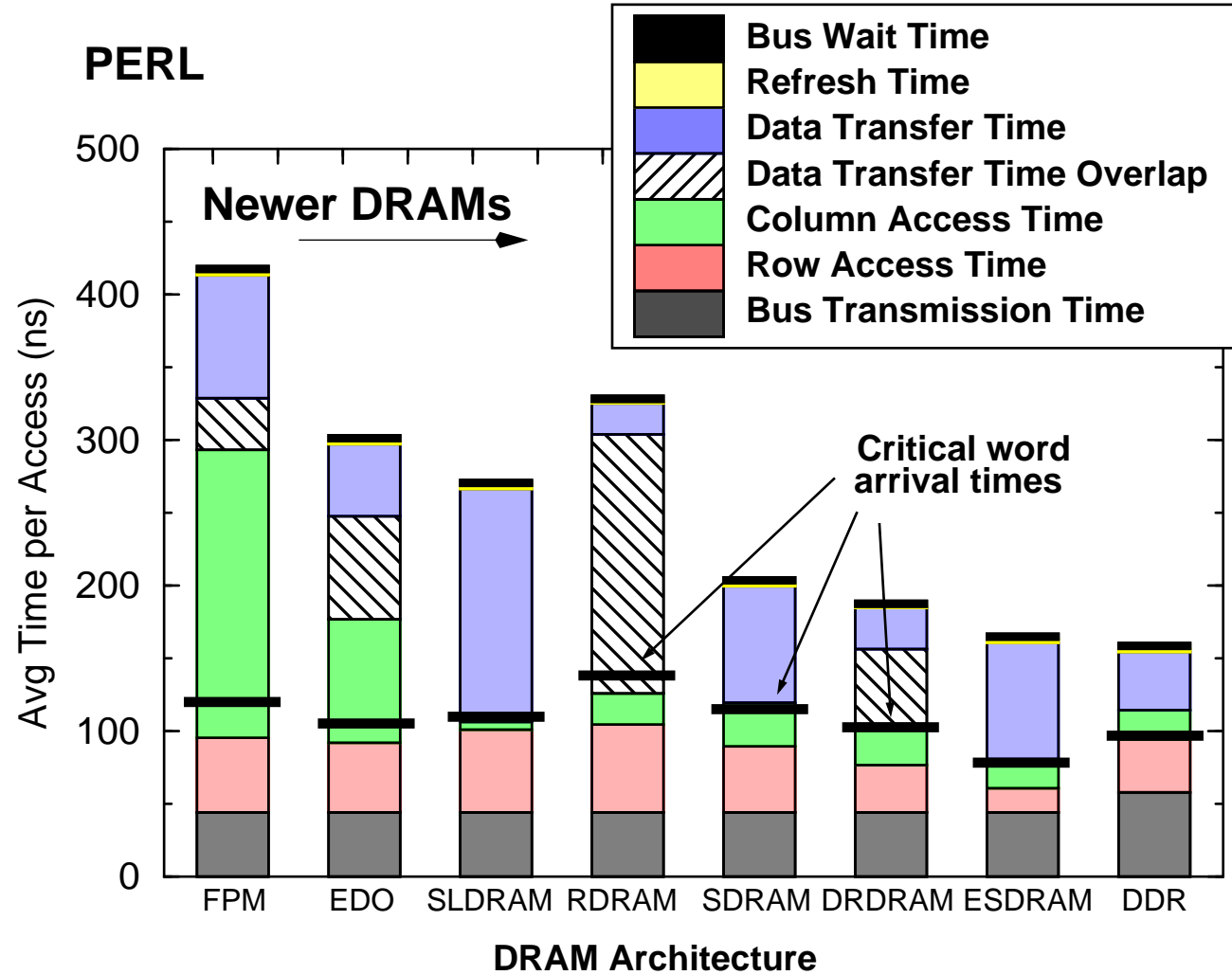


Rambus, Direct Rambus, SLDRAM:



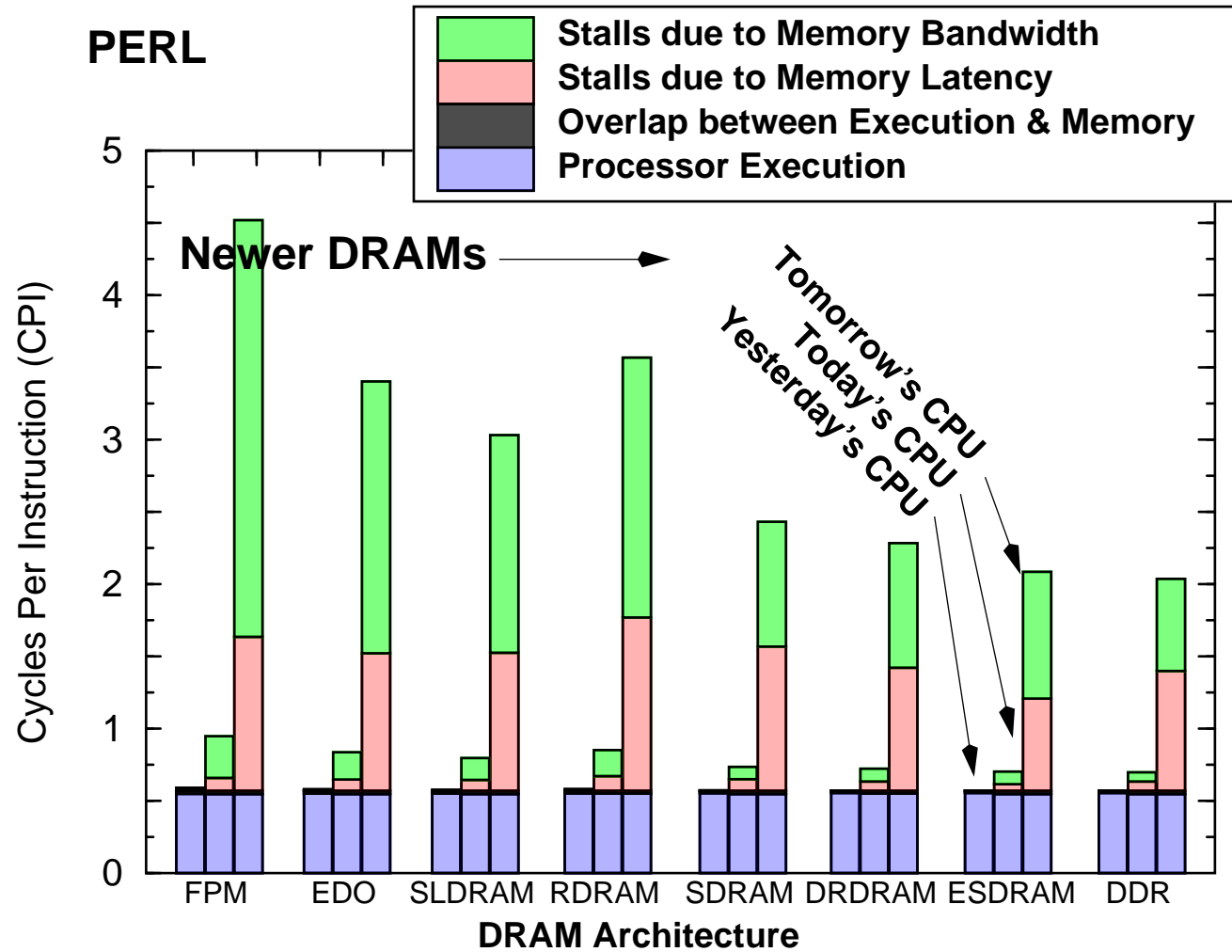
Past Work: Device-Level

Average Latencies



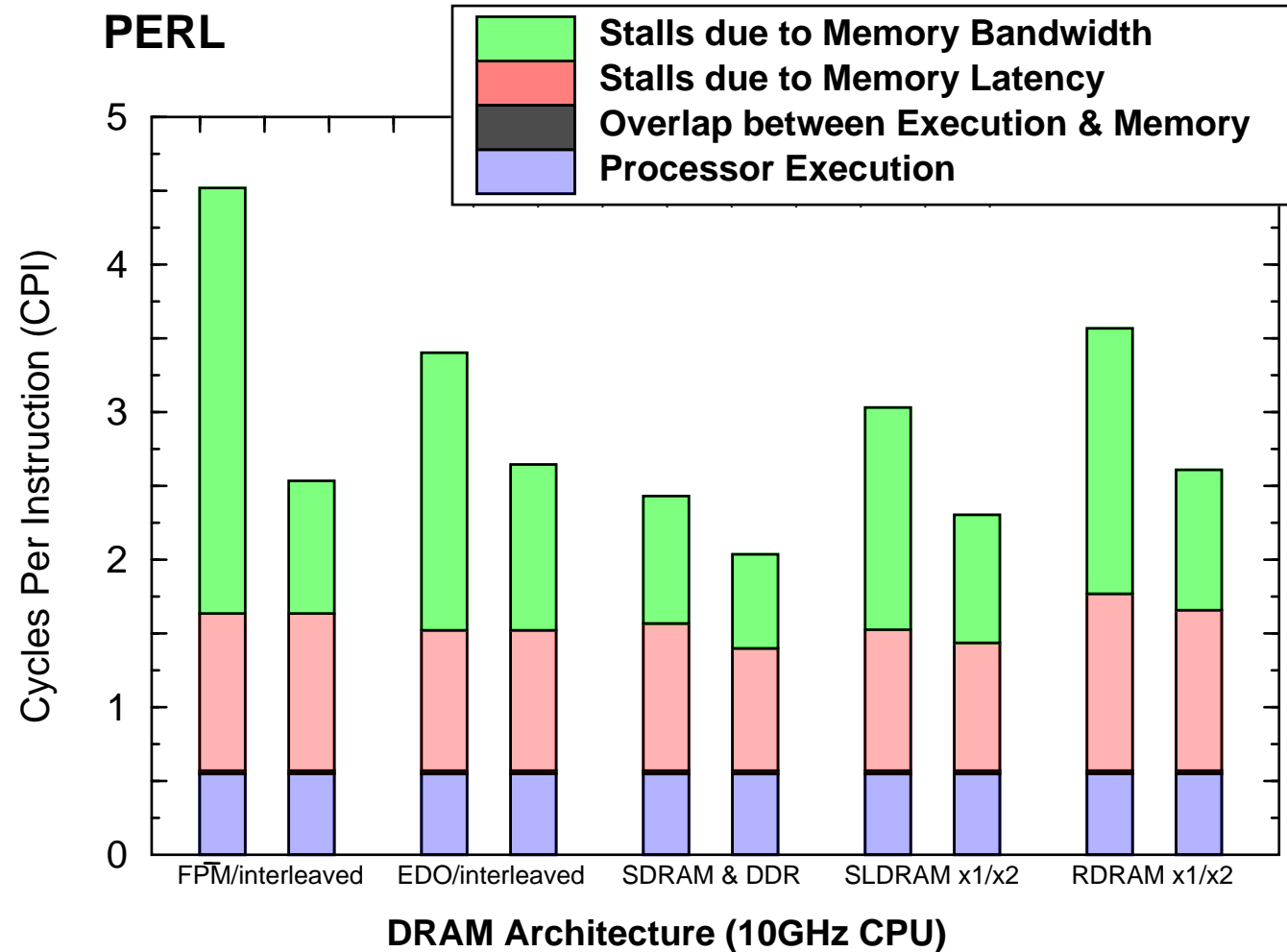
Past Work: Device-Level

Bandwidth-Enhancing Techniques I:



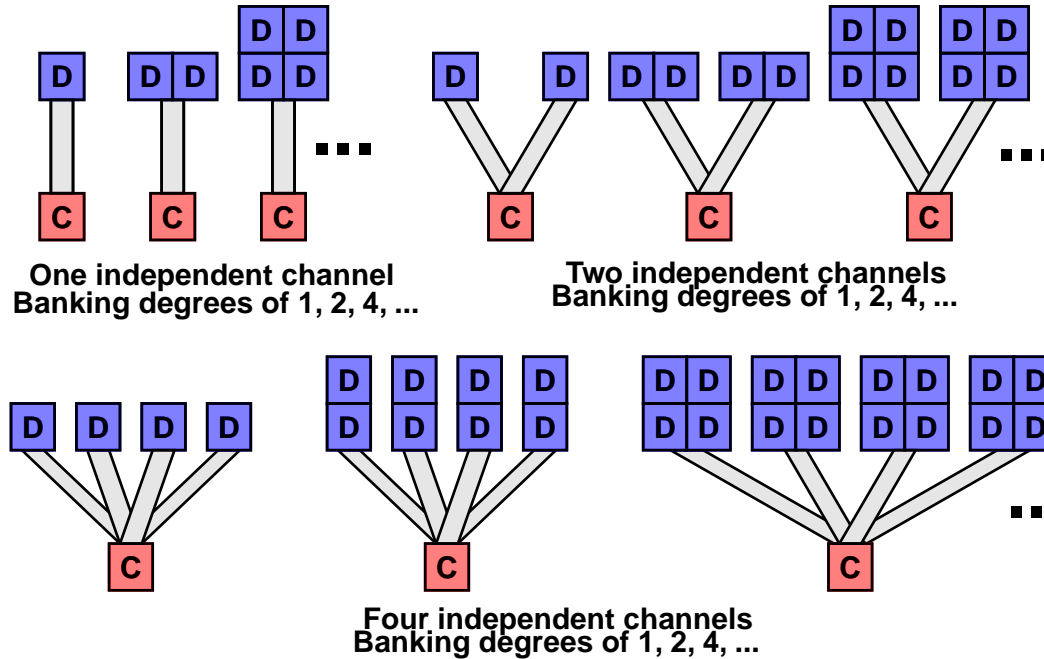
Past Work: Device-Level

Bandwidth-Enhancing Techniques II:



Past Work: System-Level

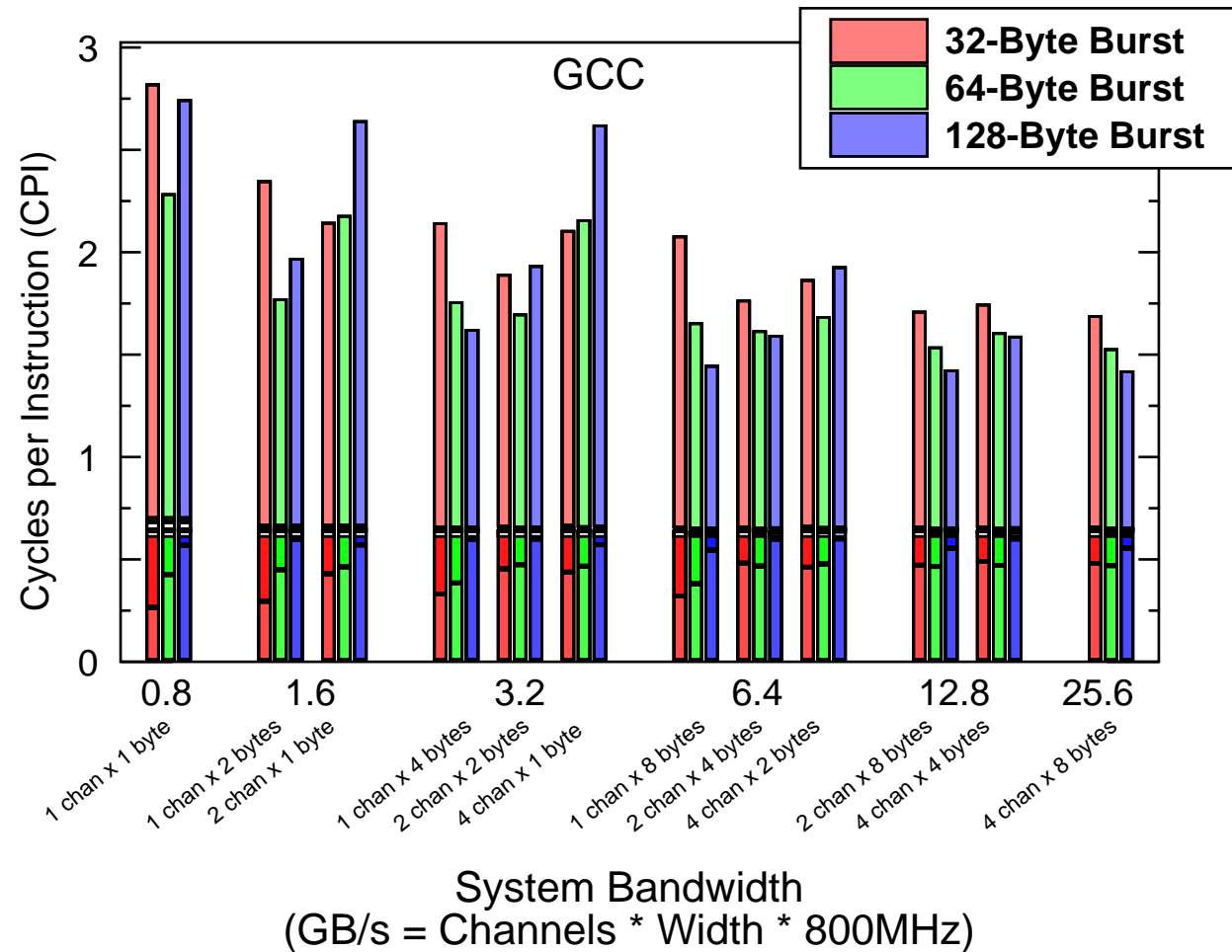
Even when we restrict our focus ...



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

Past Work: System-Level

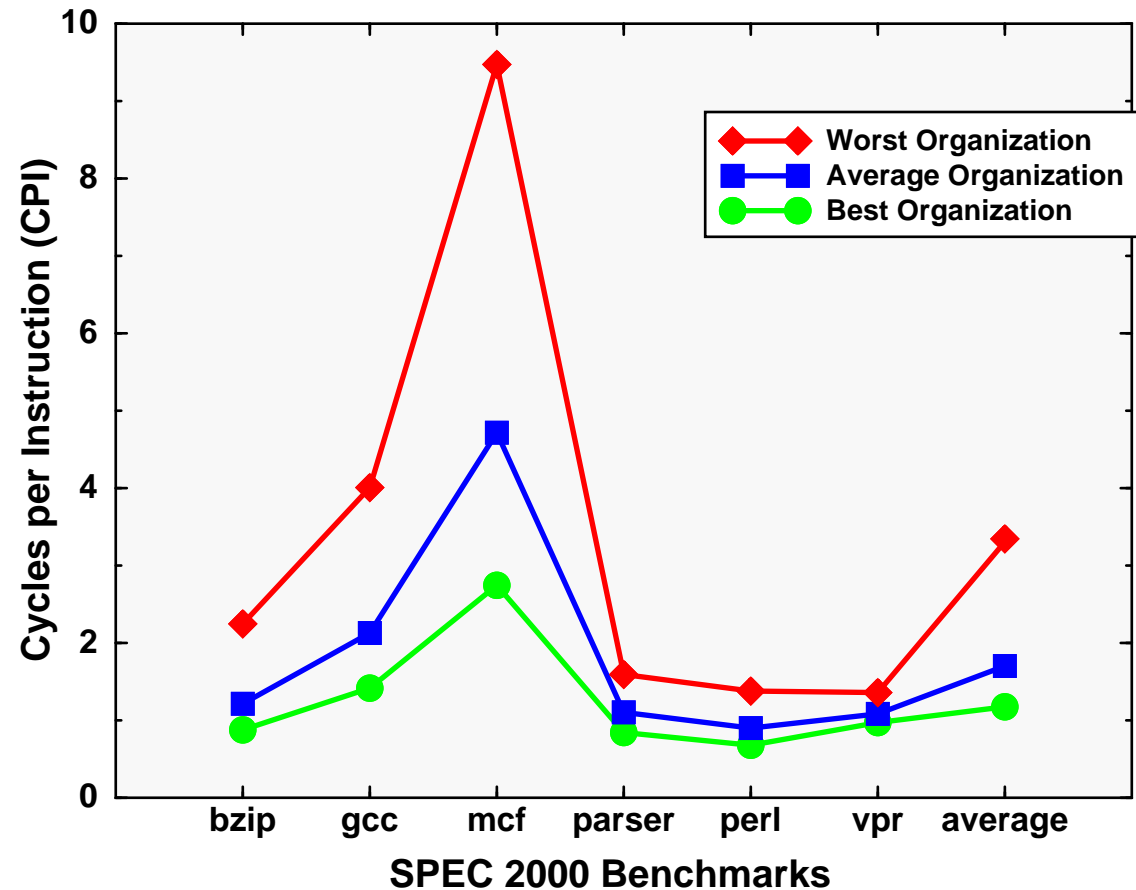
... the design space is FAR from regular ...



[Cuppu & Jacob ISCA 2001]

Past Work: System-Level

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



[Cuppu & Jacob ISCA 2001]

An Aside

Past work used first-order models.

**Present work uses models
accurate to *second & third order* effects ...**

[Definition: Zero'th Order]

...

```
if ( INSTR.is_loadstore ) {  
    if ( L1_cache_miss( INSTR.daddr ) ) {  
        if ( L2_cache_miss( INSTR.daddr ) ) {
```

```
            cycles += DRAM_LATENCY;
```

OR

```
            INSTR.ready = now() + DRAM_LATENCY;
```

```
        }  
    }  
}
```

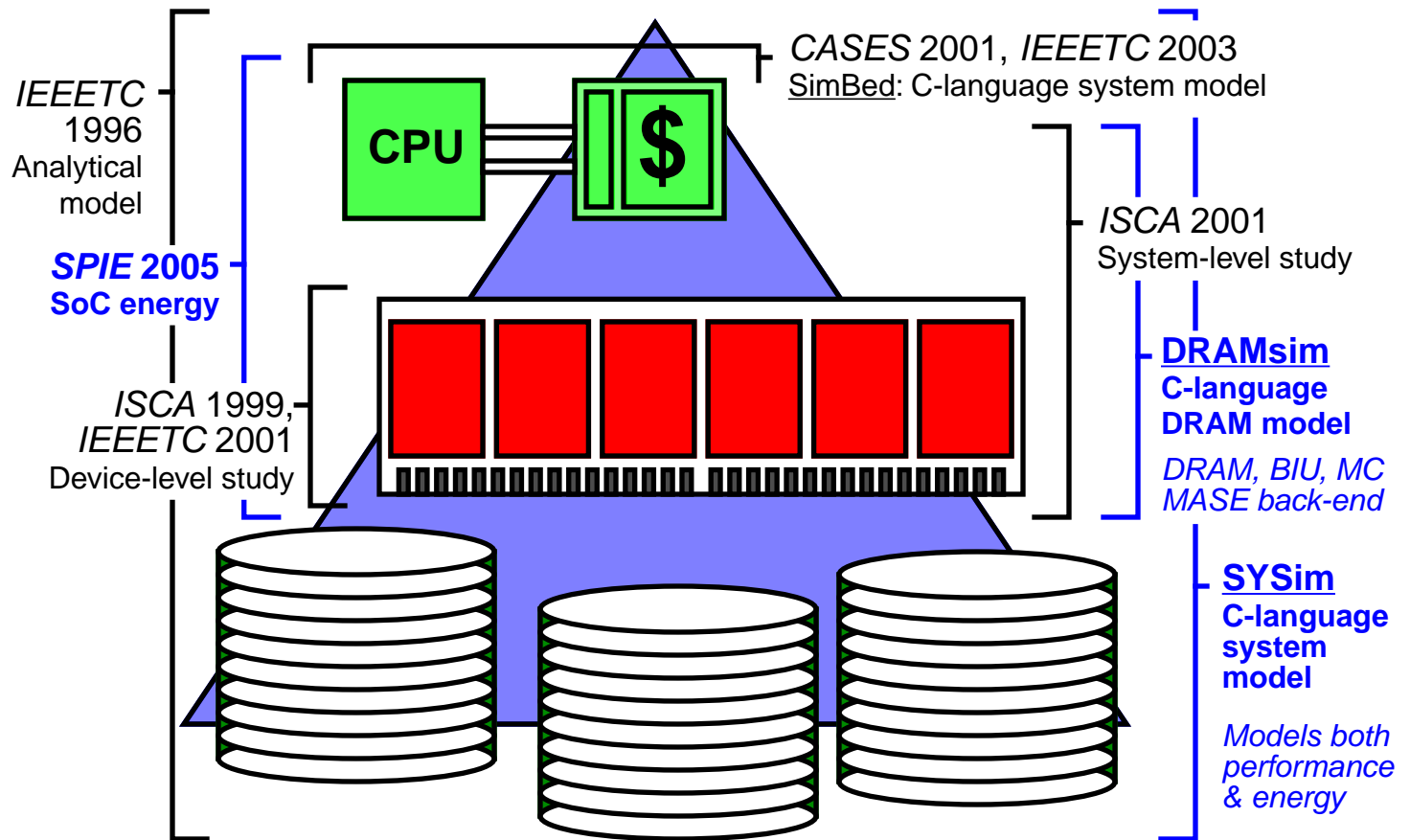
...

An Aside

Past work used first-order models.

**Present work uses models
accurate to *second & third order* effects ...**

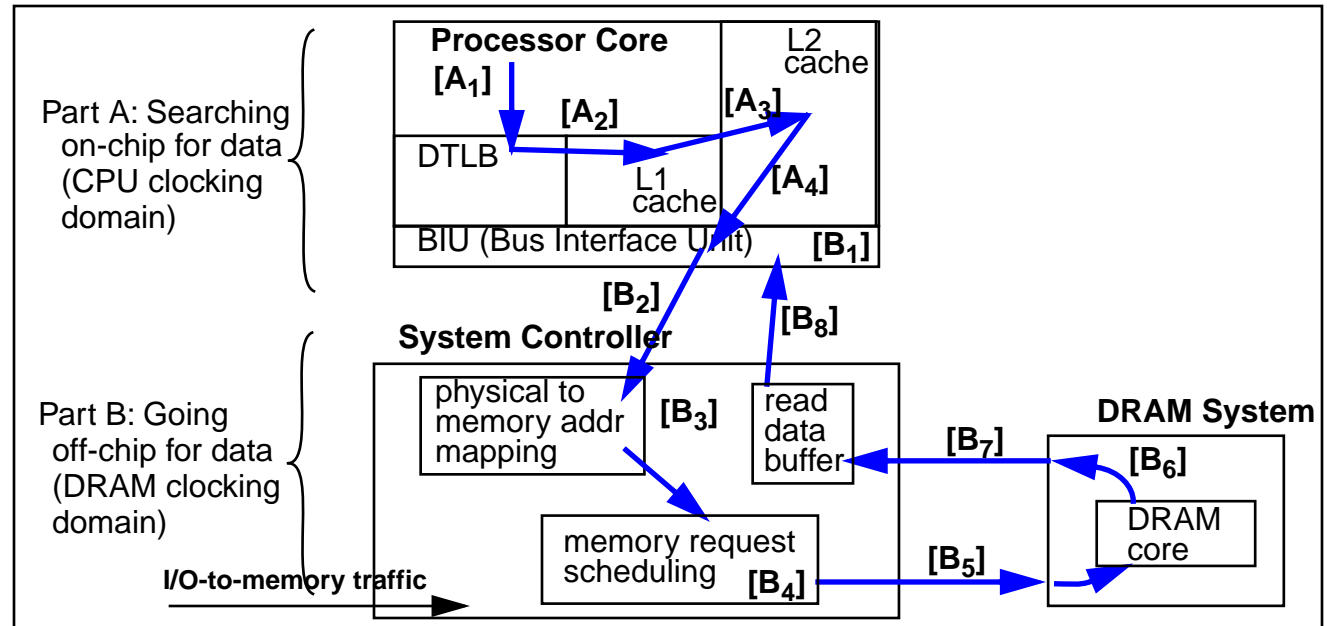
Past & Present Work



- IEEETC 1996:** System-level analytical tool for cost/performance
- ISCA 1999, IEEETC 2001:** DRAM device-level characterization
- CASES 2001, IEEETC 2003:** Performance & energy modeling of CPU and SRAM (model executes unmodified RTOS)
- ISCA 2001:** DRAM system-level characterization
- SPIE 2005:** SystemC modeling of energy in systems-on-chip

Present Work: DRAMsim

Execution of a Load Instruction



Stages of instruction execution:



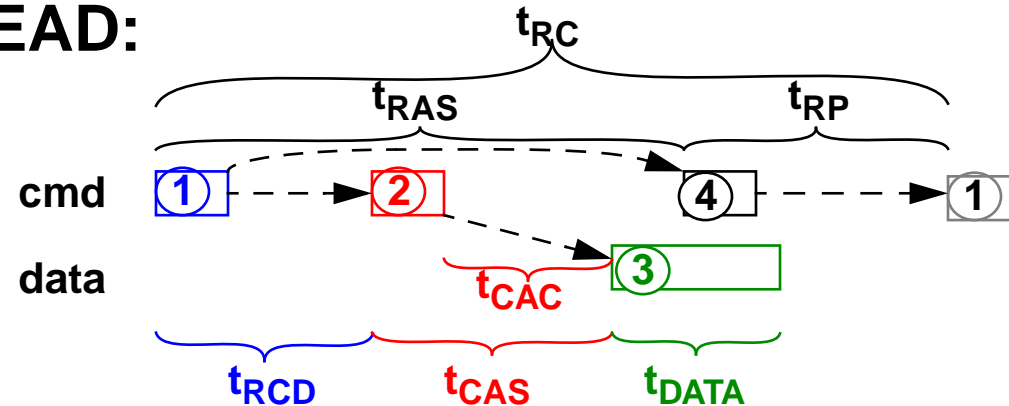
virtual to physical address translation (DTLB access) [A ₁]	[A ₂] L1 D-Cache access. If miss then proceed to	[A ₃] L2 Cache access. If miss then send to BIU	Bus Interface Unit (BIU) obtains data from main memory [A ₄ + B]
---	--	---	---

[B ₁] BIU arbitrates for ownership of address bus **	[B ₂] request sent to system controller	[B ₃] phys. addr. to memory addr. translation.	[B ₄] mem. request scheduling**	[B ₅] mem. addr. Setup (RAS/CAS)	[B ₆ , B ₇] DRAM dev. obtains data and returns to controller	[B ₈] system controller returns data to CPU
--	---	--	---	--	---	---

** Steps not required for some processor/system controllers. protocol dependant.

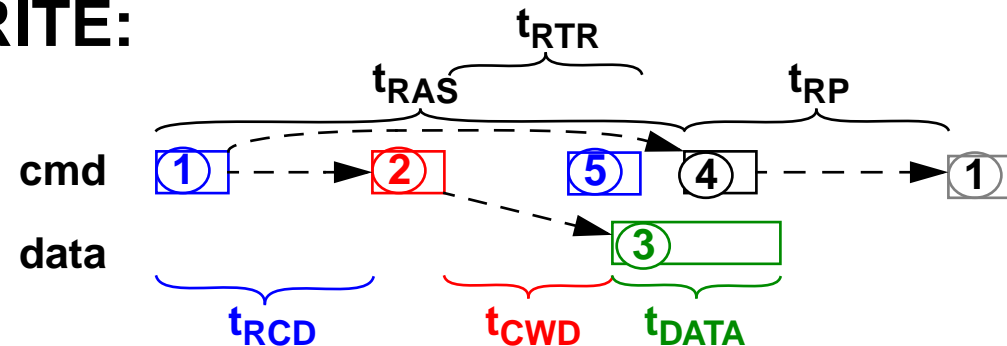
Present Work: DRAMsim

READ:



- ① Active: Open Row, t_{RCD} time later, a CAS command may be issued to the DRAM chip
- ② CAS: Column Read command, t_{CAS} time later, data begins to be placed onto the Data bus. We use t_{CAC} to factor out command transmission time.
- ③ Data: The number of cycles that the data transmits over the Data bus
- ④ Precharge: Close the Row, this command may be issued t_{RAS} time after the Active command. After t_{RP} time, another active command may be issued.

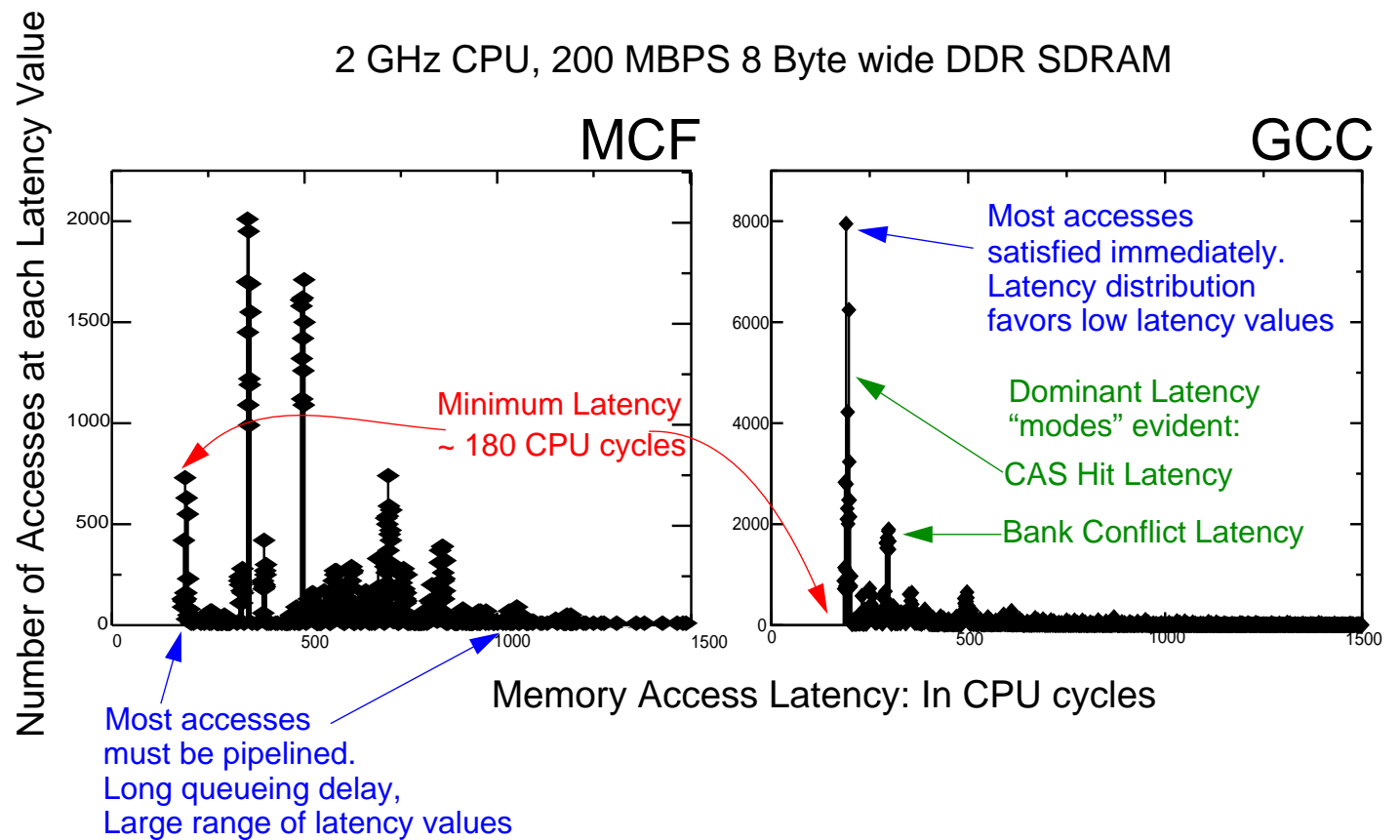
WRITE:



- ② CWD: Column Write Delay, the number of cycles that the controllers must wait before placing the data onto the data bus.
- ⑤ RTR: Retirement delay, this is for systems with write delay buffers.(RDRAM)

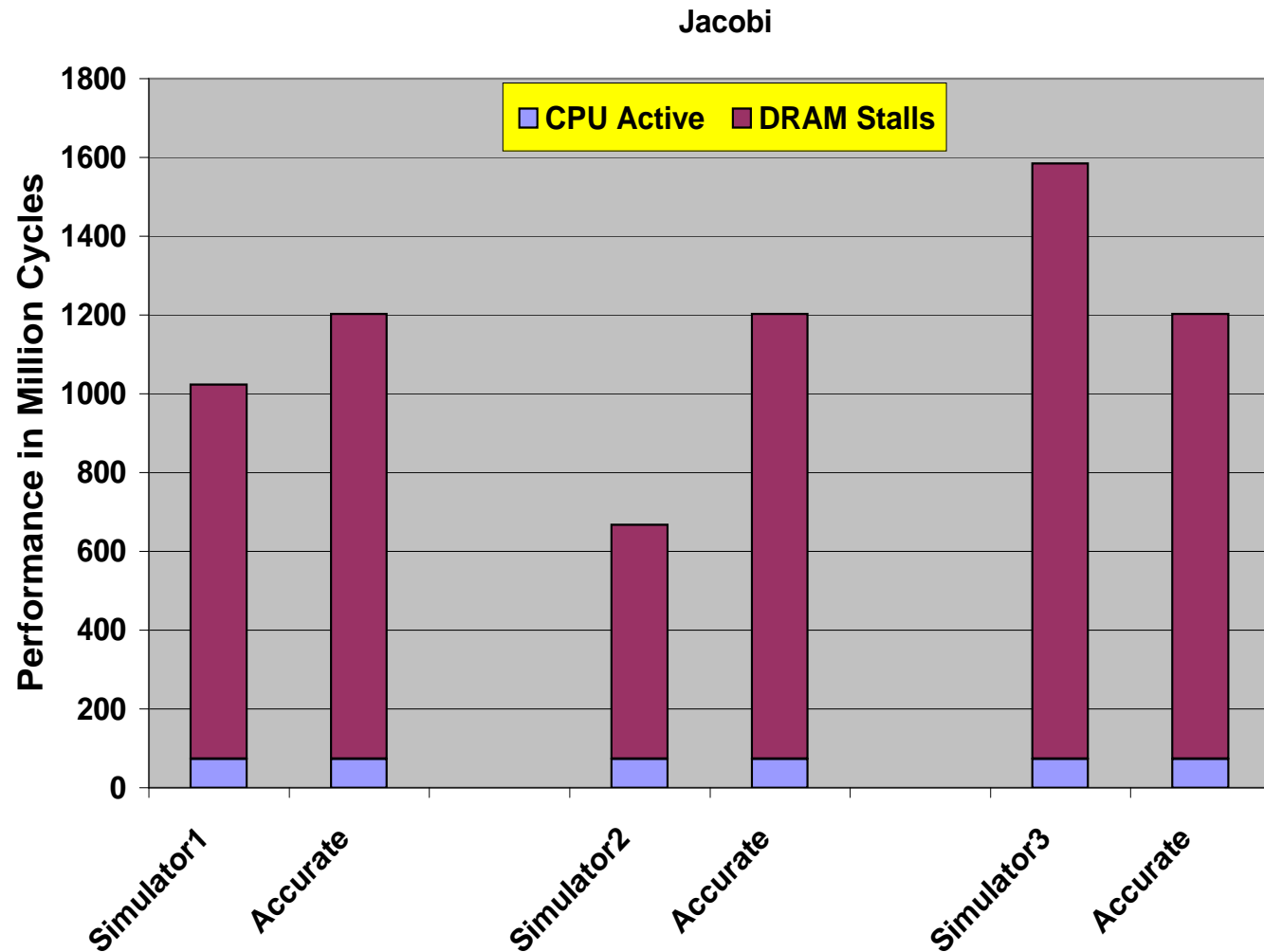
Present Work: DRAMsim

Memory Access Latency Distribution



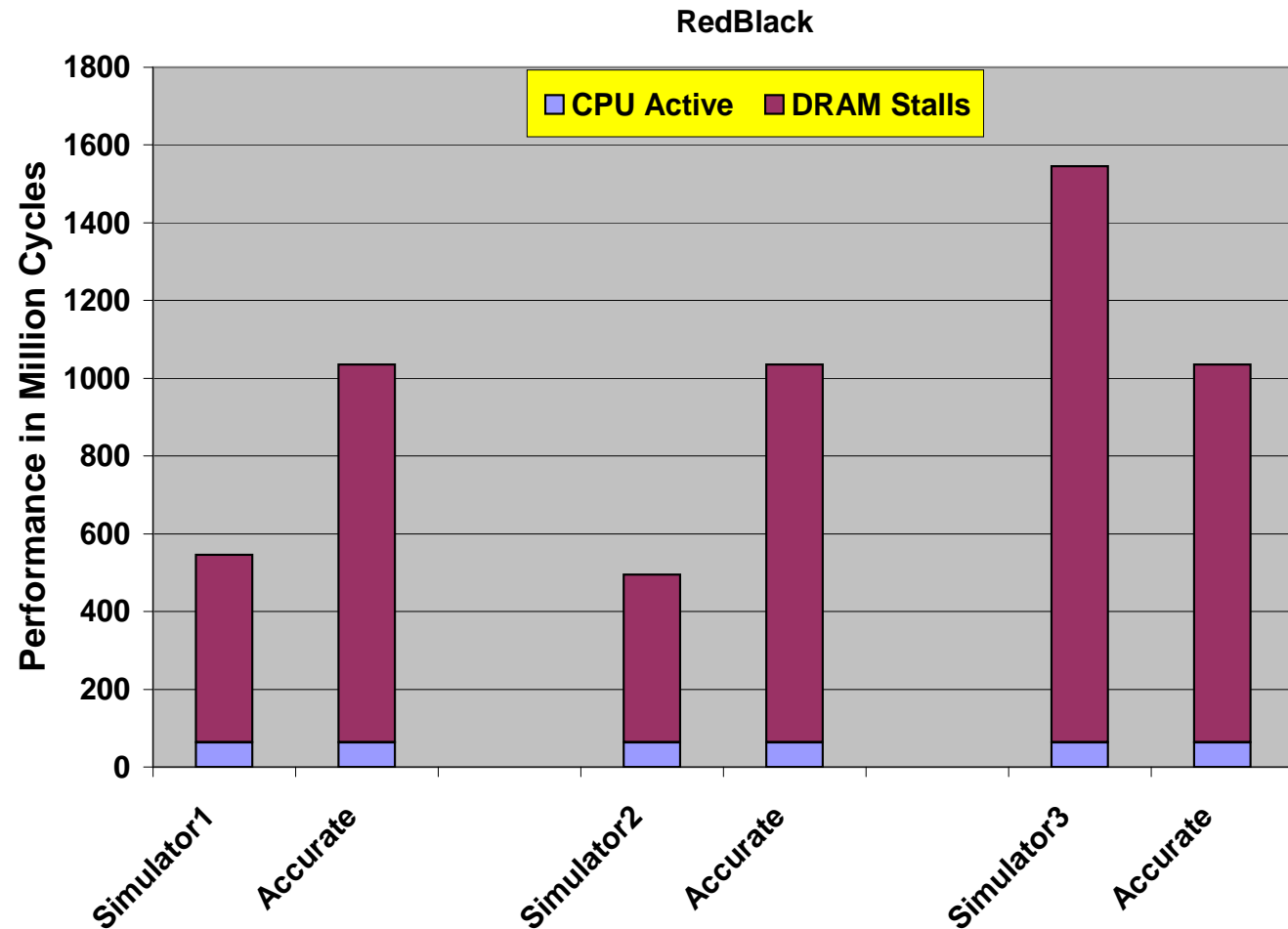
Present Work: *Why?*

Benefit: *Performance Numbers (DRAMsim)*



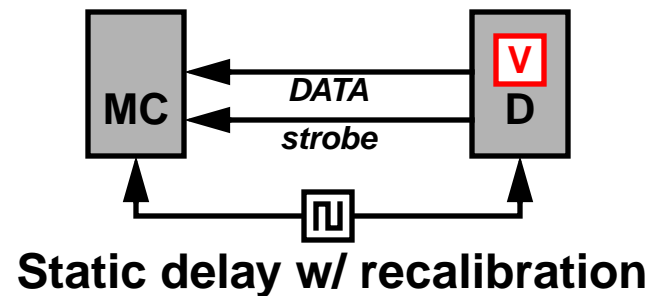
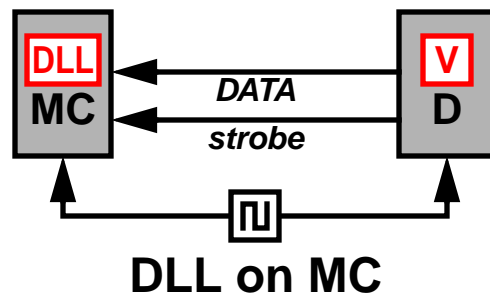
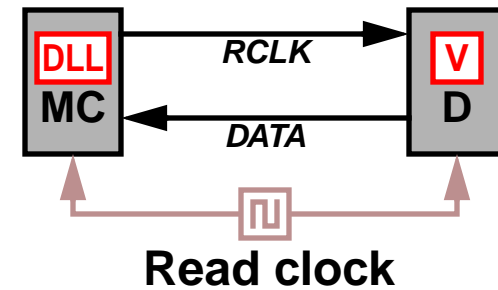
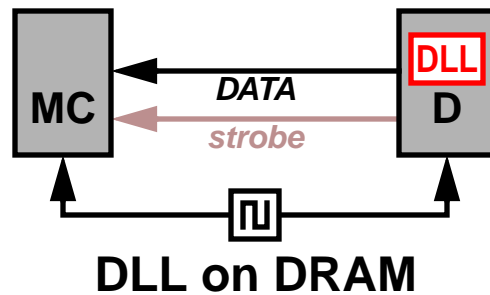
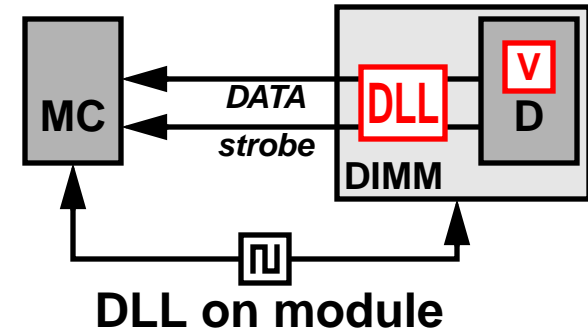
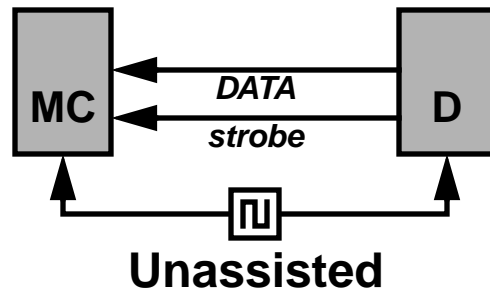
Present Work: *Why?*

Benefit: *Performance Numbers (DRAMsim)*



Present Work: *Why?*

Benefit: *Insights (Anecdote II, revisited)*



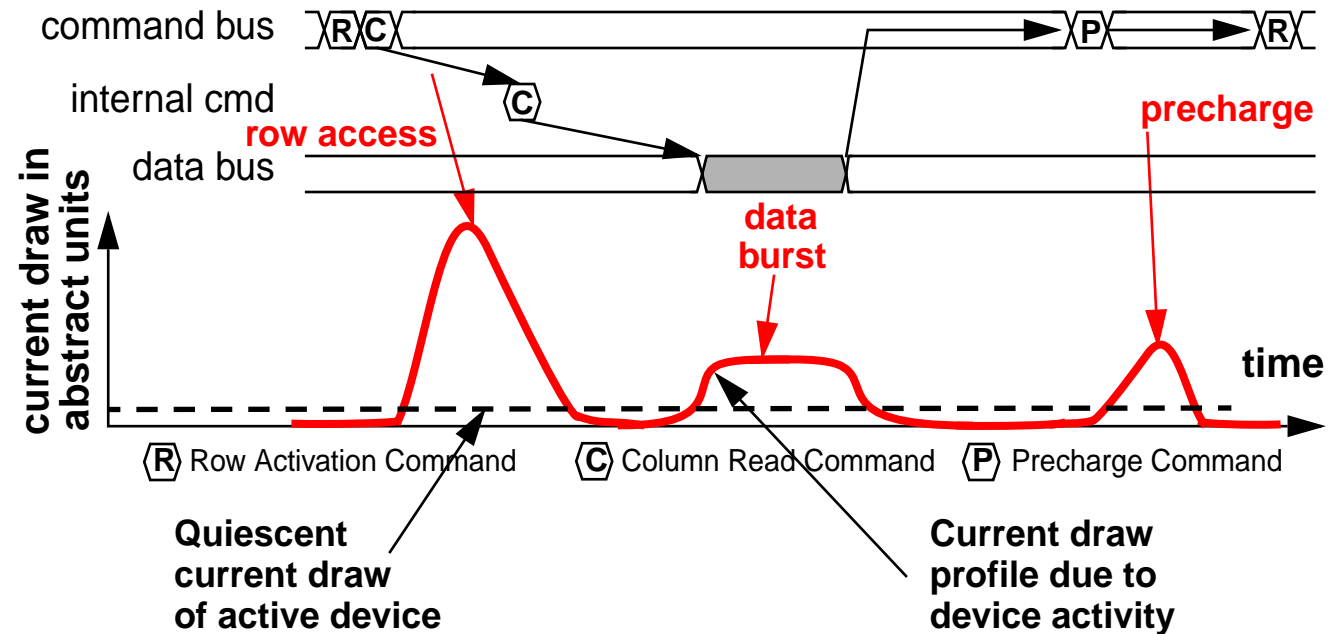
Present Work: *Why?*

Benefit: *Insights (Anecdote II, revisited)*

SCHEME	COST	EFFECTIVENESS (Uncertainty in read)
No DLL	0	$D_{\text{CLK}} + X_{\text{mit}} + \text{wire} + \text{Recv} + \text{Clk skew}$
on DRAM	16xDLL	$X_{\text{mit}} + \text{wire} + \text{Recv} + \text{Clk skew}$
on MC	2xDLL 16xVern	$\text{wire} + \text{Recv}$
on DIMM	2xDLL 16xVern	$\text{wire} + \text{Recv} + \text{Clk skew}$
Read CLK	2xDLL 16xVern	$\text{wire} + \text{Recv}$
Static	16xVern	$X_{\text{mit}} + \text{wire} + \text{Recv}$

- **Cost** = for 2-DIMM system, 8 DRAM parts per DIMM
note: “cost” applies to both die area *and* power
- **Uncertainty** = *very rough, intuitive idea*

Anecdote III, revisited



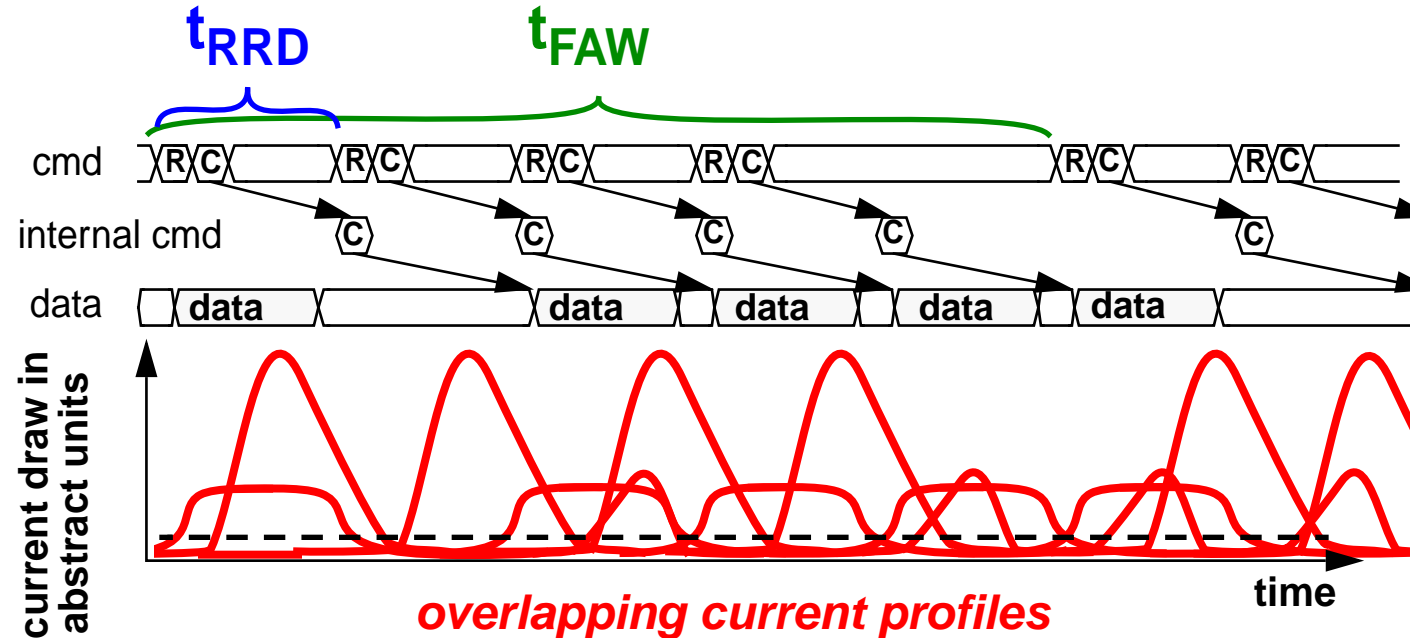
Power consumption in DRAM devices:

- Row activation, data read-out, bank precharge: all are relatively expensive operations
- Current draw of operation additive to quiescent value

... So what's the big deal?

Anecdote III, revisited

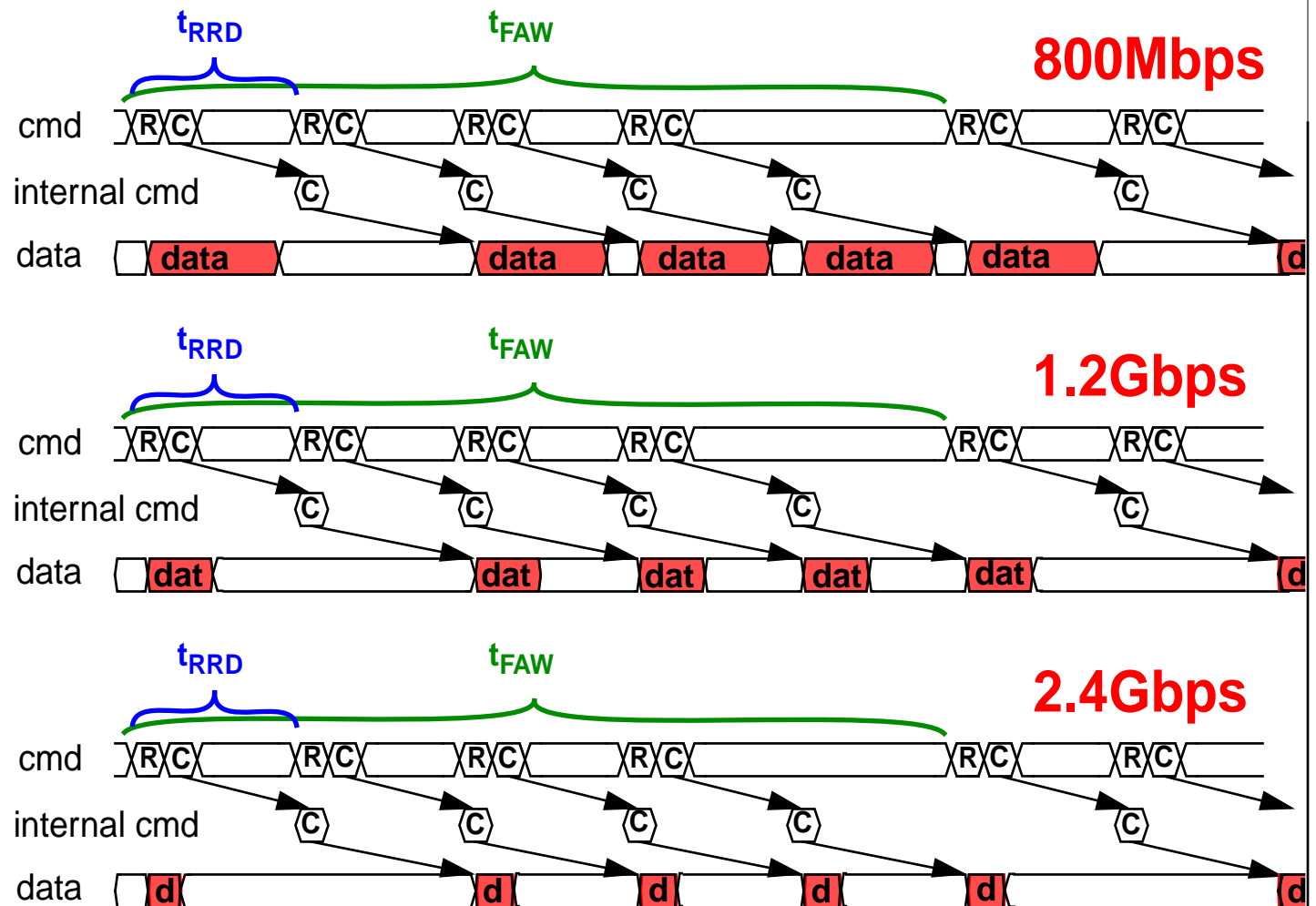
t_{RRD} & t_{FAW} *protocol-level limitations*
placed upon device to limit
maximum current draw



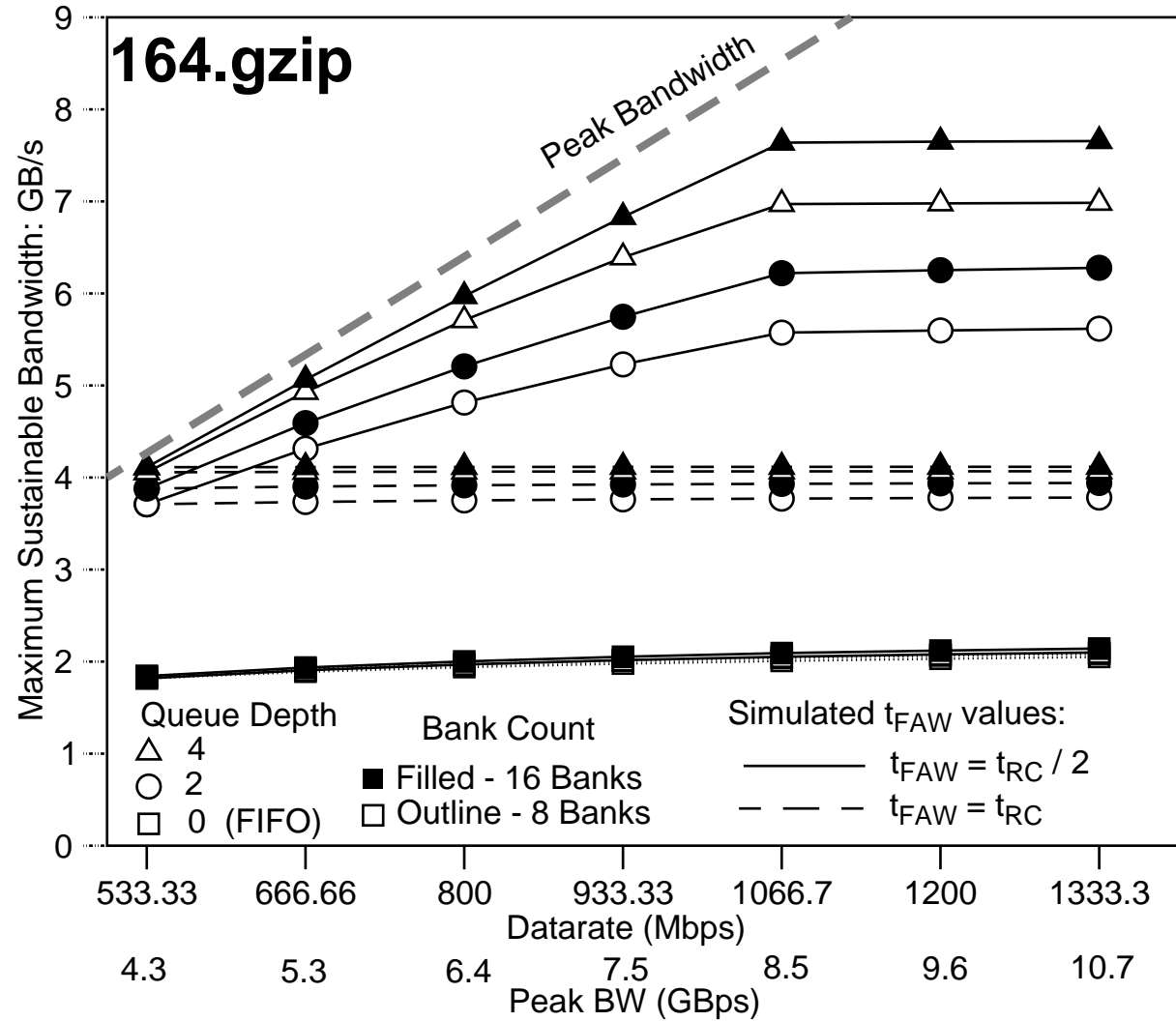
- Severely limits bus efficiency from single rank
- Problem worsens in future: parameters defined in *nanoseconds*, not *cycles*

Anecdote III, revisited

- t_{RRD} & t_{FAW} — Problem worsens in future:
parameters defined in *nanoseconds*, not *cycles*



Max. Sustainable Bandwidth



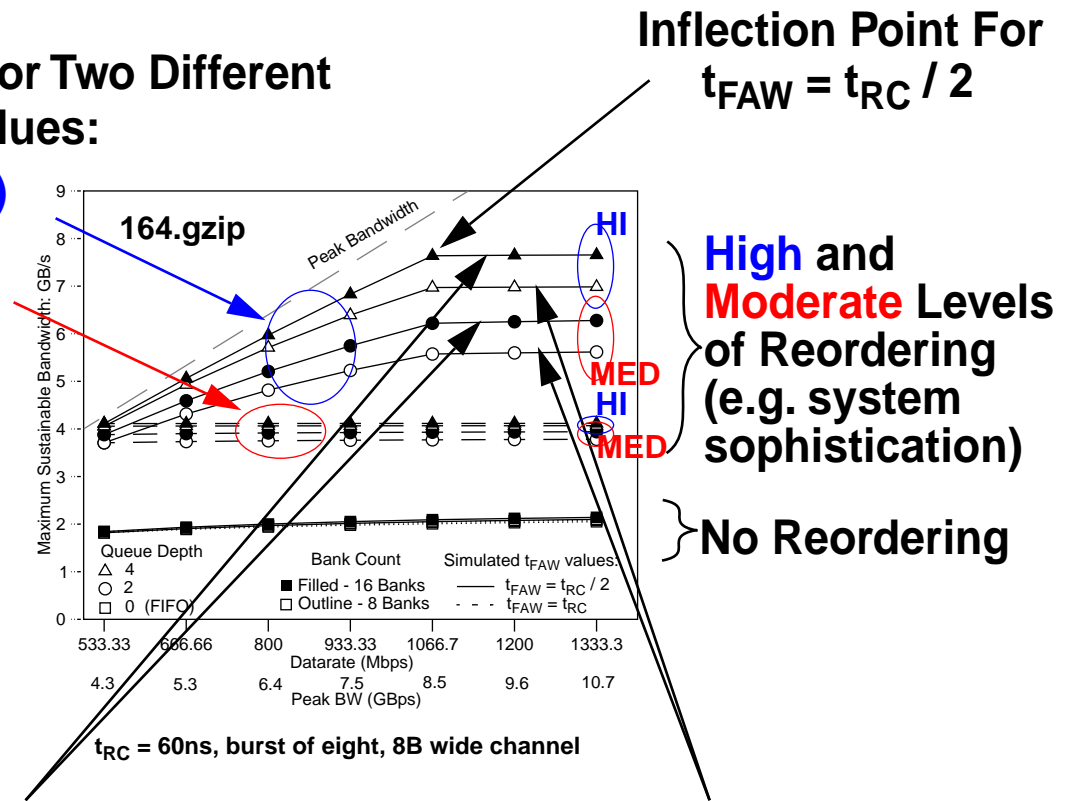
$t_{RC} = 60\text{ns}$, burst of eight, 8B wide channel

Max. Sustainable Bandwidth

t_{FAW} Impact for Two Different Simulated Values:

($t_{FAW} = t_{RC}/2$)

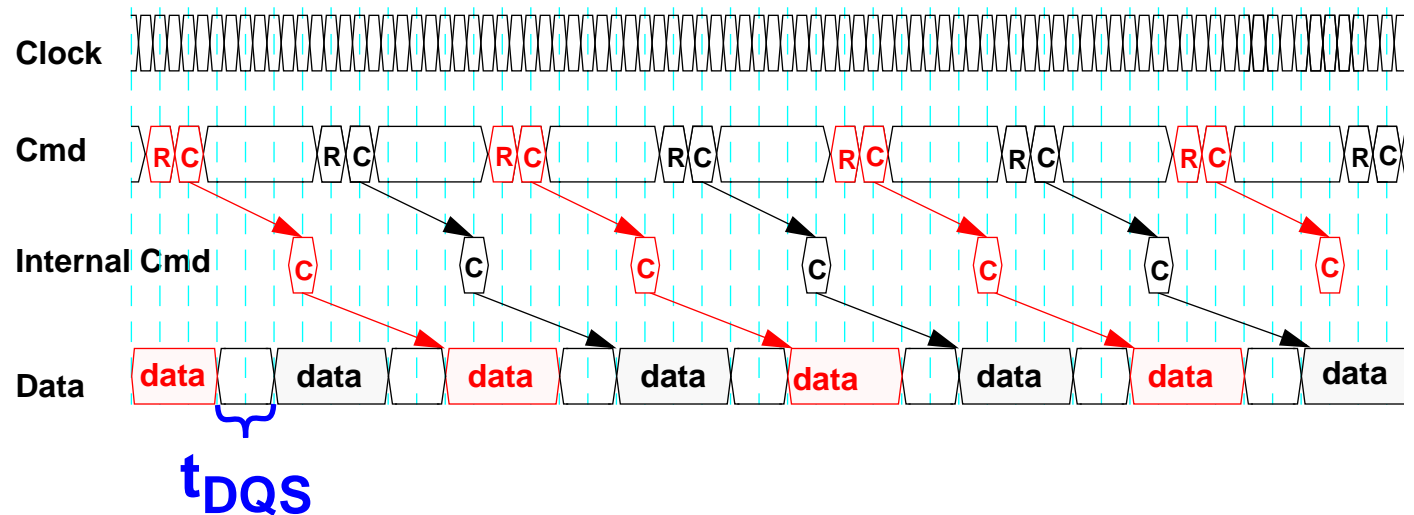
($t_{FAW} = t_{RC}$)



16 BANKS improves bandwidth over **8 BANKS** by ~10%
(how does this compare with incremental cost?)

But Wait, There's More ...

t_{DQS} *protocol-level limitation placed upon ranks to prevent data-bus collisions on rank hand-off*



- Severely limits bus efficiency from multiple ranks
- Luckily, it is defined in *cycles* and not *nanoseconds*

Solution I: Scheduling

Problems created by $t_{FAW} + t_{RRD} + t_{DQS}$

- $t_{FAW} + t_{RRD}$ Must spread out ACT commands
- t_{DQS} Must switch ranks infrequently

Salient point: t_{FAW} does not place limit on *total* number of open banks

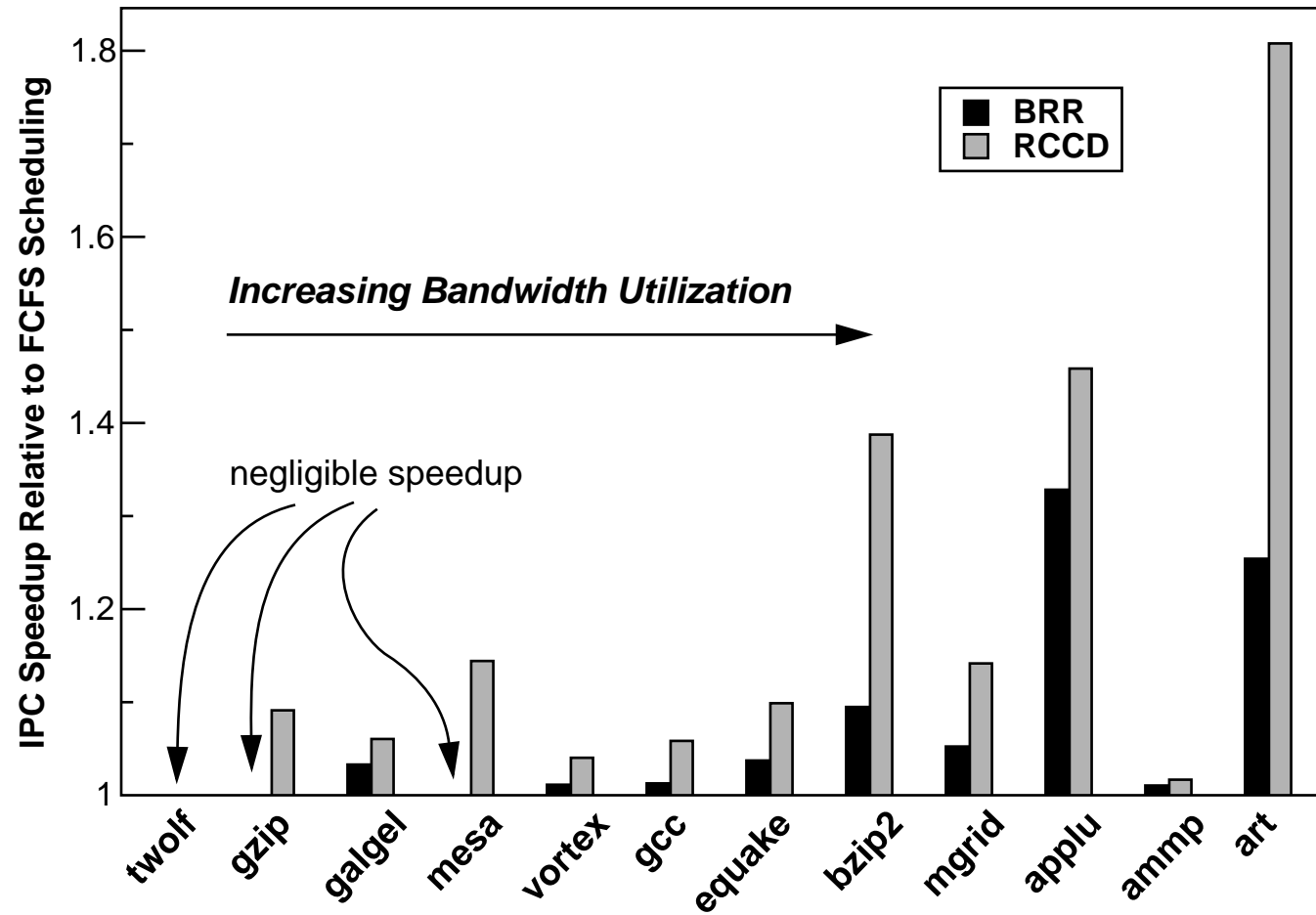
**Problem can be solved with scheduling:
*row-column command decoupling (RCCD)***

- Schedule ACT commands far before their corresponding READ commands
- Schedule large number of bank-reads before switching ranks

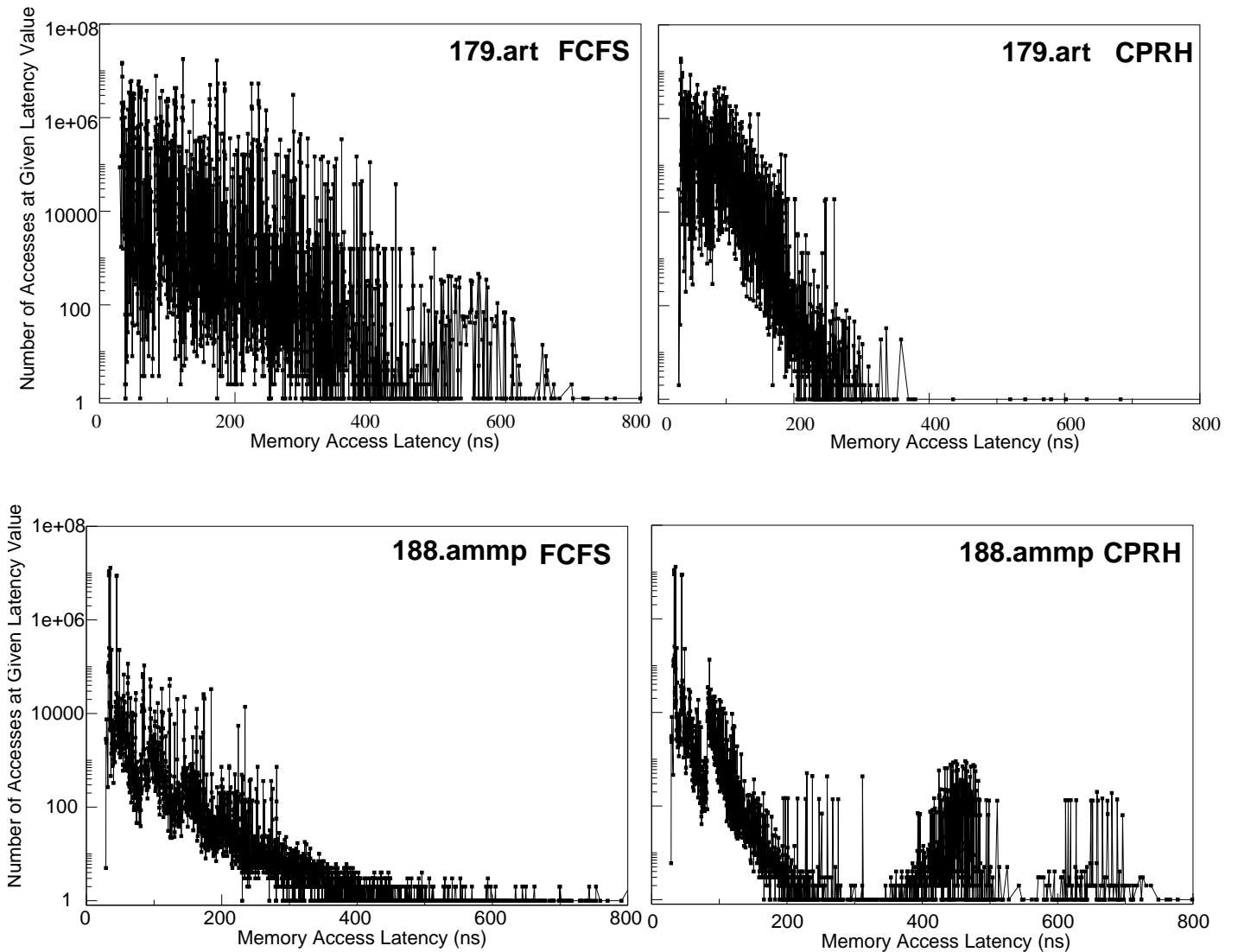
[patent pending]

Solution I: Scheduling

IPC Speedup Relative to FCFS Scheduling



Solution I: Scheduling

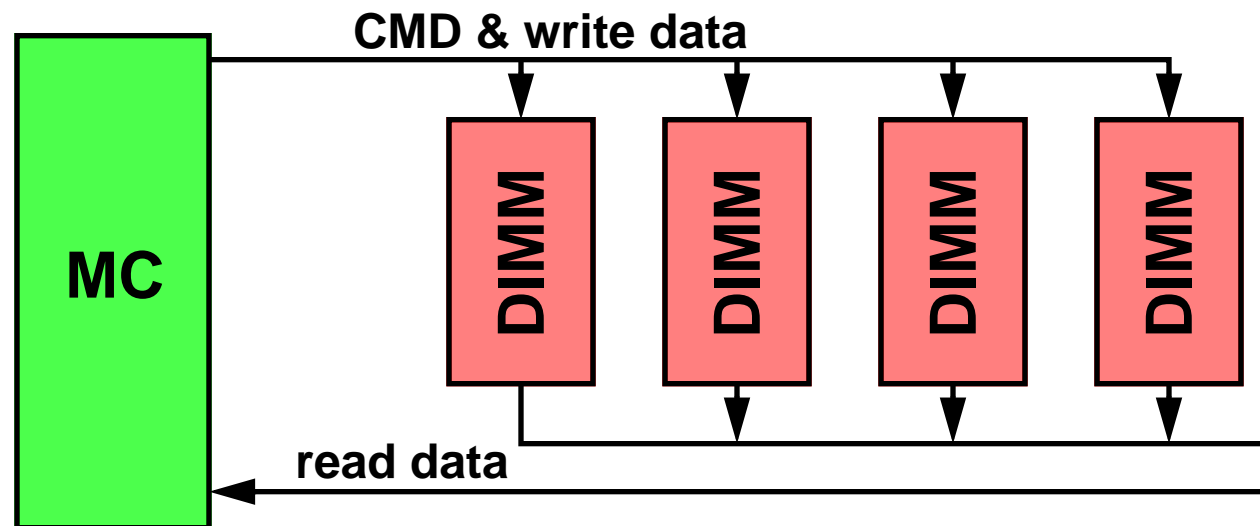


Solution II: Topology, etc.

Problems solved by $t_{FAW} + t_{RRD} + t_{DQS}$

- $t_{FAW} + t_{RRD}$ Instantaneous current draw in device
- t_{DQS} Bus collisions on rank handoffs

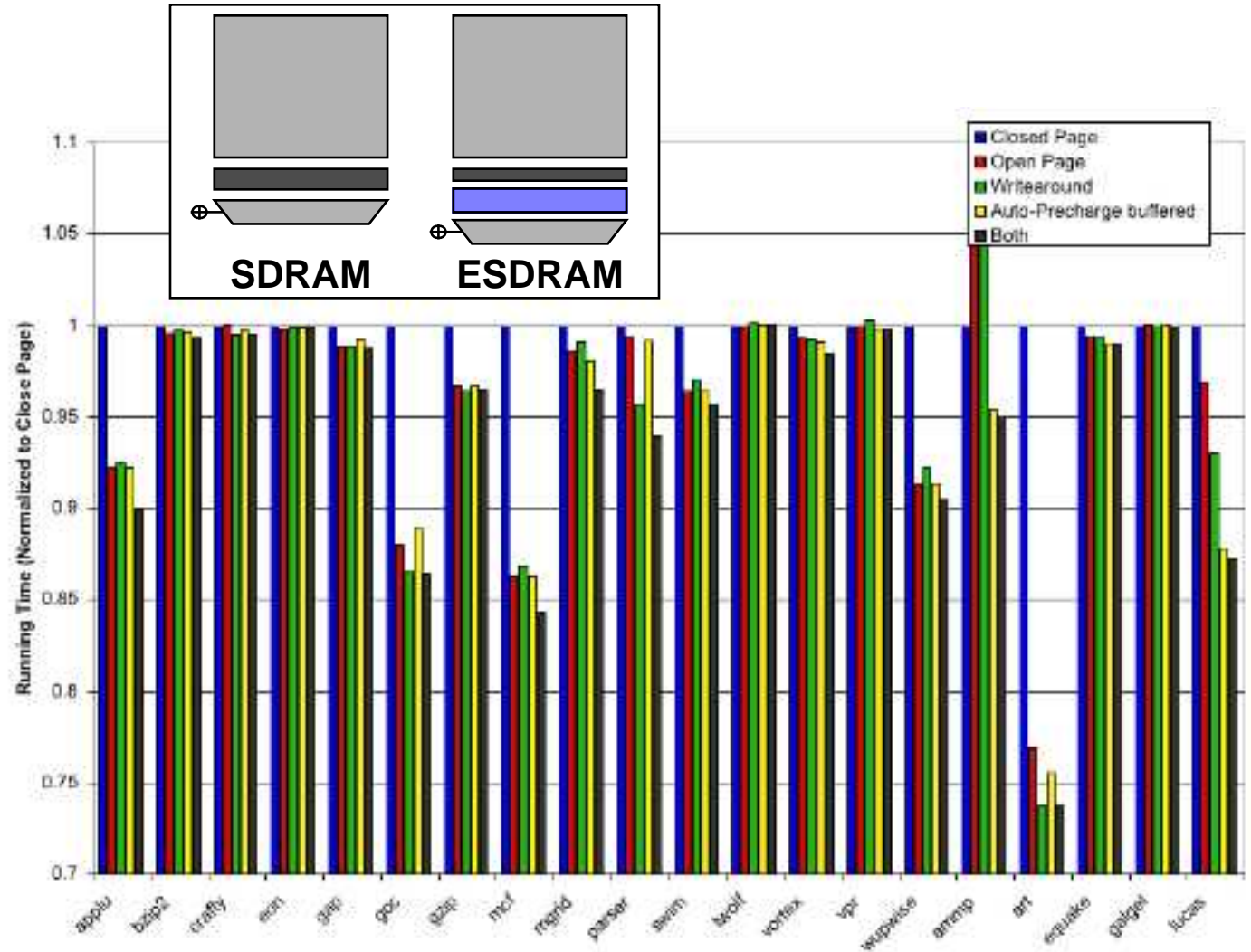
Any alternative solution will do ...



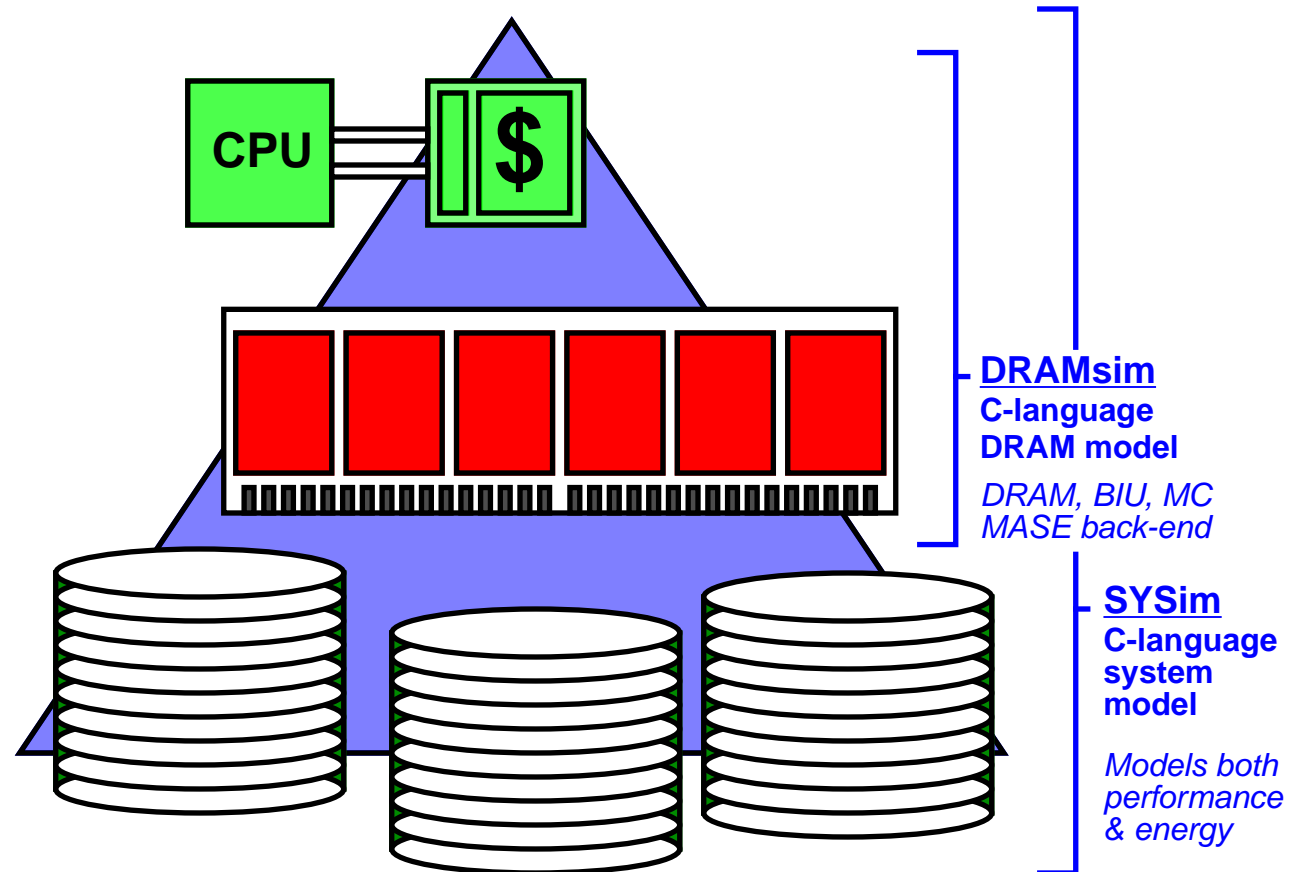
- Topology eliminates collisions (can account for static DIMM-DIMM skew with Vernier-type solution)
- Note: solution requires source-synchronous clocking

Other Items, e.g. ESDRAM

How effective is the write-around feature?



Perspective, Future Work



- Investigating these types of issues (local-vs-system) at all levels of the hierarchy (cache, disk, etc).
- MKP book to come (“holistic approach” ??)
- Goal: SYSim will be open-source release w/ book

Summary

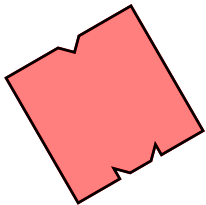
No longer appropriate to optimize subsystems in isolation: local optima do not yield globally optimal system

Systemic behaviors: unanticipated interactions yielding inefficiencies

Specific instances:

- $t_{FAW} + t_{RRD} + t_{DQS}$ severely limits BW
- Choice of DLL on DDR SDRAMs to de-skew parts

Many problems can be addressed by system-level solutions; can be better than circuit-level solutions



Et Cetera

MEMORY-SYSTEMS RESEARCH SUPPORT:

- NSF (CAREER Award)
- Intel
- Cray

(CURRENT) MEMSYS GRAD STUDENTS:

- **Dave Wang:** *DRAMsim*, $t_{FAW} + t_{RRD} + t_{DQS}$ studies, etc.
- **Ankush Varma:** SystemC system-on-chip energy model
- **Nuengwong Tuaycharoen:** *SYSim* development
- **Sean Leventhal:** Other studies (e.g. ESDRAM, MC, etc.)

CONTACT INFO:

- **Prof. Bruce Jacob**
ECE Dept., University of Maryland, College Park, MD
- <http://www.ece.umd.edu/~blj/>
blj@umd.edu

