

# How Not to Configure Your DRAM System

**Bruce Jacob**

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

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

## **OUTLINE:**

- **Motivation & DRAM Primer**
- **Yesterday's Experiments & Results**
- **Today's Experiments & Results**
- **Conclusions**



# Sources

**“A Performance Study of Contemporary DRAM Architectures,” *Proc. ISCA '99.***

**V. Cuppu, B. Jacob, B. Davis, and T. Mudge**

**“DDR2 and Low Latency Variants,” *Memory Wall Workshop*, in conjunction w/ ISCA '00.**

**B. Davis, T. Mudge, V. Cuppu, and B. Jacob.**

**Recent experiments by Vinodh Cuppu,  
Ph.D. student at University of Maryland**

# Dilemma: THIS ...

## STATUS QUO in MEMORY-SYSTEM RESEARCH:

...

```
if ( INSTR.loadstore ) {  
    if ( L1_cache_miss( INSTR.daddr ) ) {  
        if ( L2_cache_miss( INSTR.daddr ) ) {  
  
            cycles += DRAM_LATENCY;  
  
        }  
    }  
}
```

...

## ... or THIS ...

### STATUS QUO in MEMORY-SYSTEM RESEARCH:

...

```
if ( INSTR.loadstore ) {  
    if ( L1_cache_miss( INSTR.daddr ) ) {  
        if ( L2_cache_miss( INSTR.daddr ) ) {  
  
            INSTR.ready = now() + DRAM_LATENCY;  
  
        }  
    }  
}
```

...

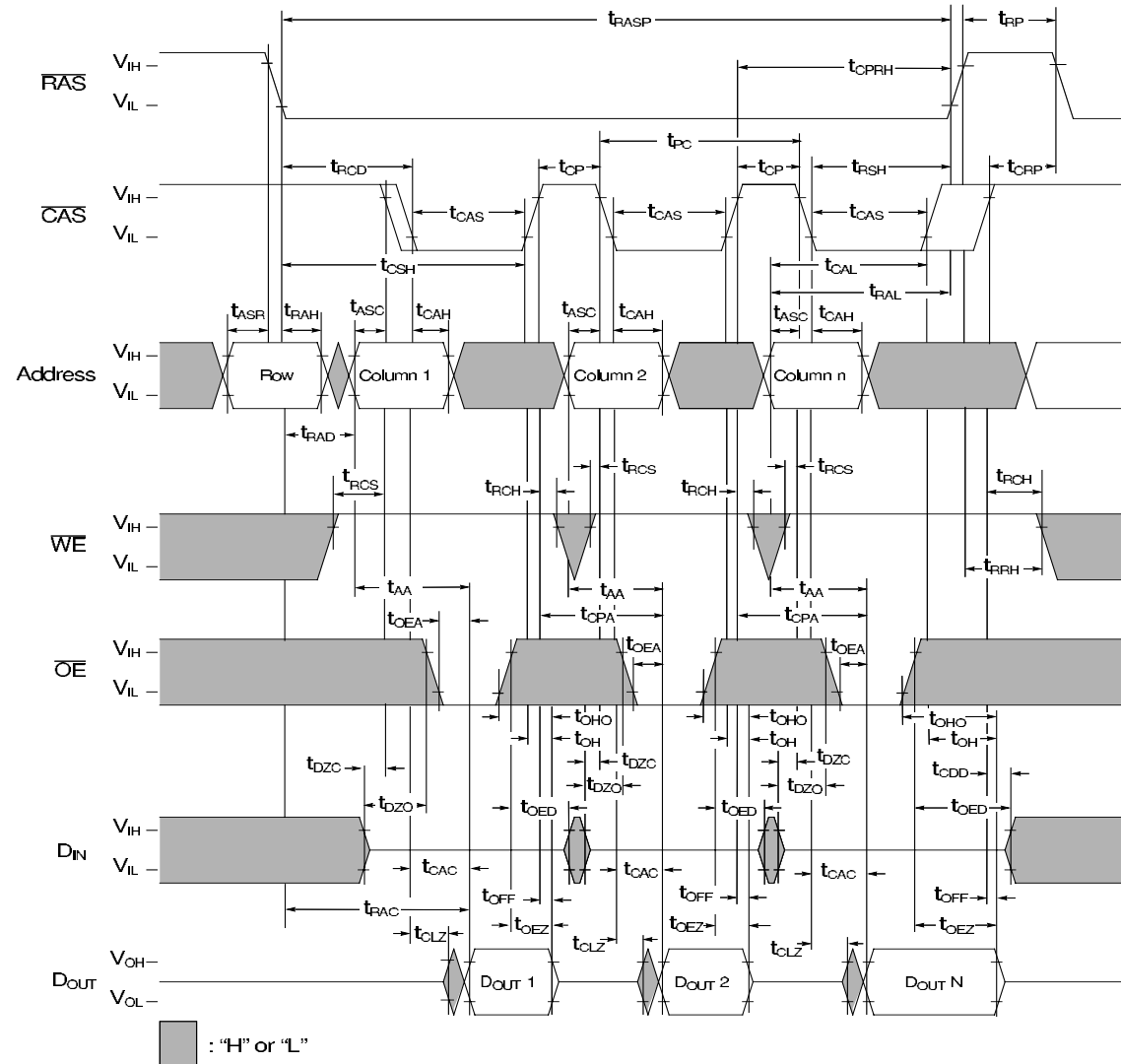
HOW NOT TO  
CONFIGURE  
YOUR DRAM SYSTEM

Bruce Jacob

University of  
Maryland

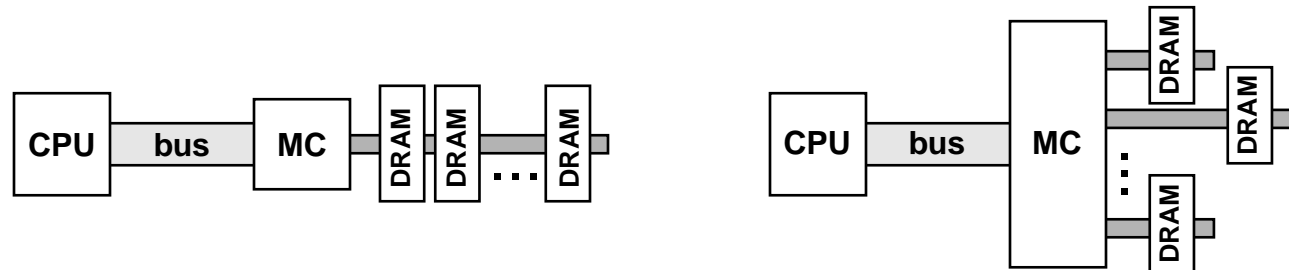
# ... or THIS

Fast Page Mode Read Cycle

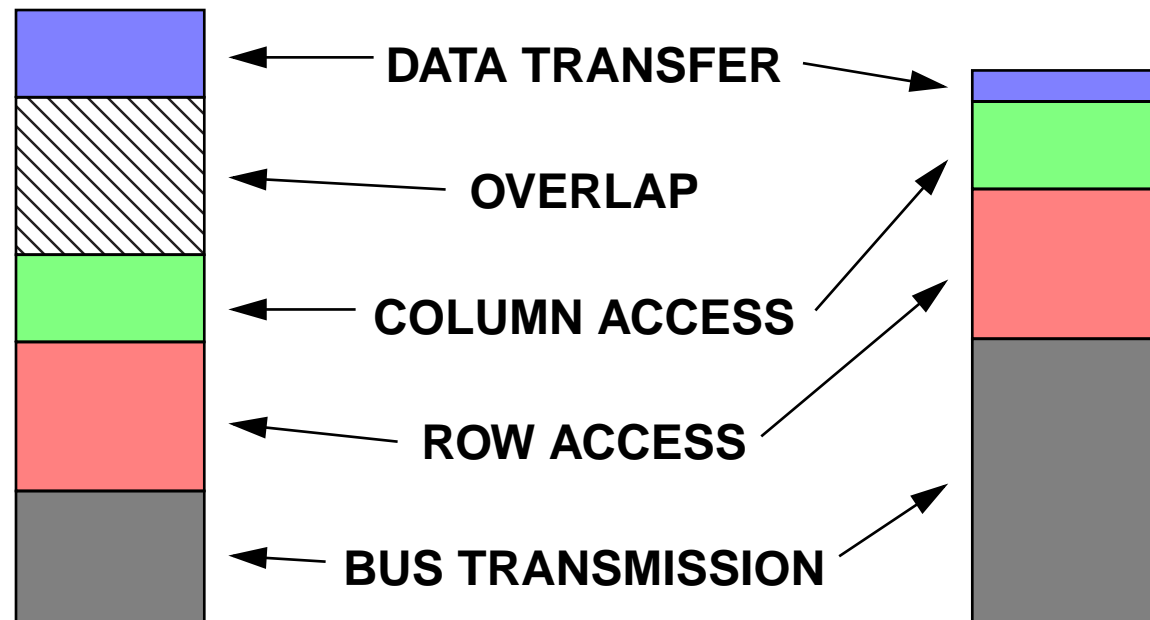


# Motivation

## HERE'S WHAT YOU MISS:

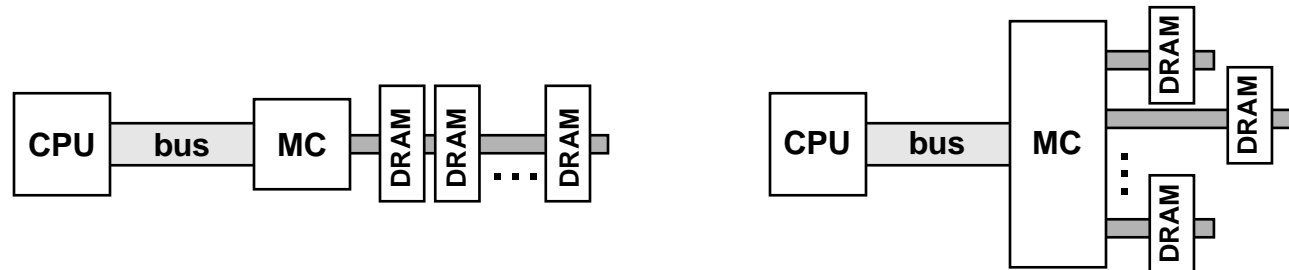


## DRAM LATENCY:



# Motivation

## HERE'S WHAT YOU MISS:



## DRAM LATENCY:

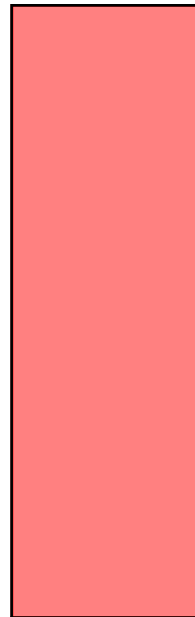
DATA TRANSFER?

OVERLAP?

COLUMN ACCESS?

ROW ACCESS?

BUS TRANSMISSION?



# Goal

## PRELIMINARY DRAM STUDY:

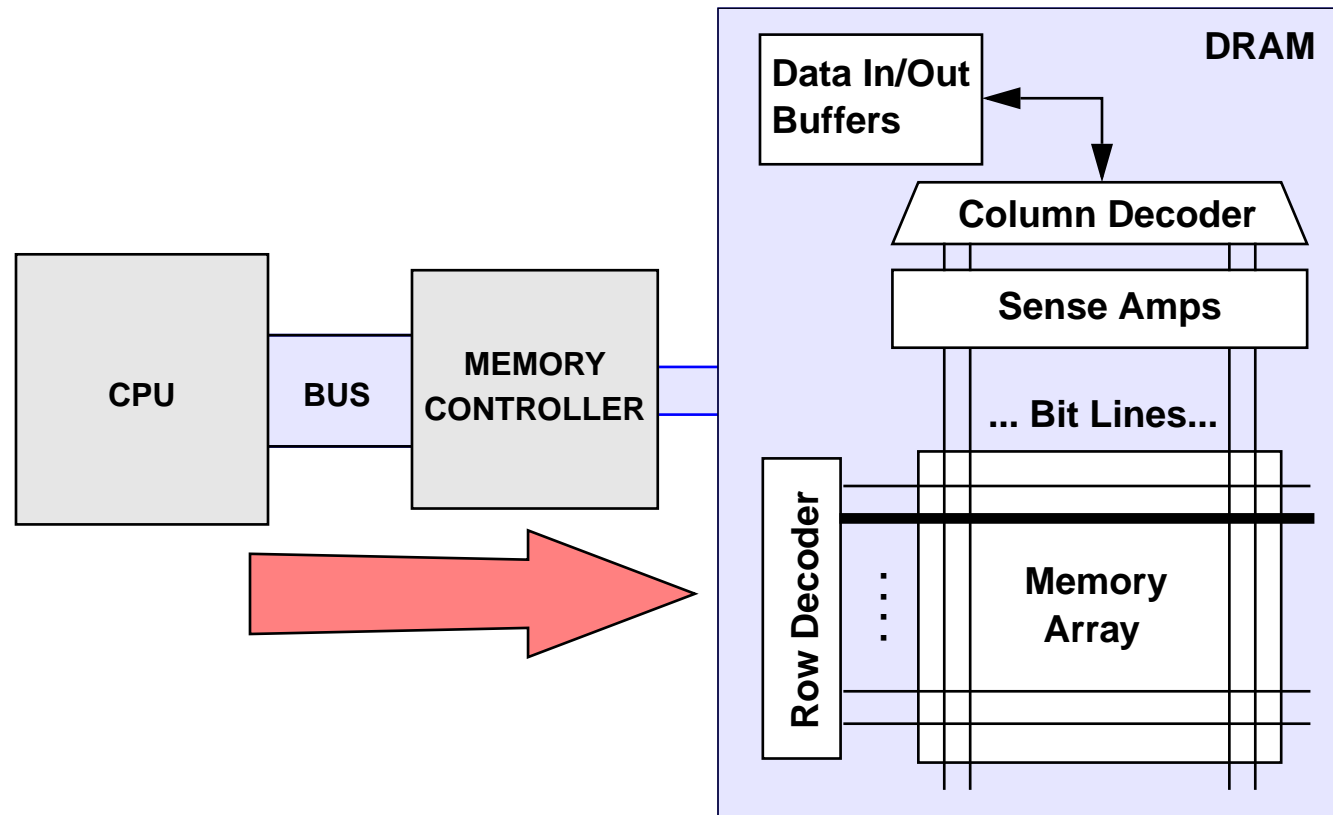
- **Bus Transmission**
- **Row Access**
- **Column Access**
- **Data Transfer**
- **Bus Wait/Synch Time**
- **Stalls Due to Refresh**
- **The OVERLAP of These Components  
(with each other)  
(with CPU execution)**

**MODEL EXISTING TECHNOLOGY**



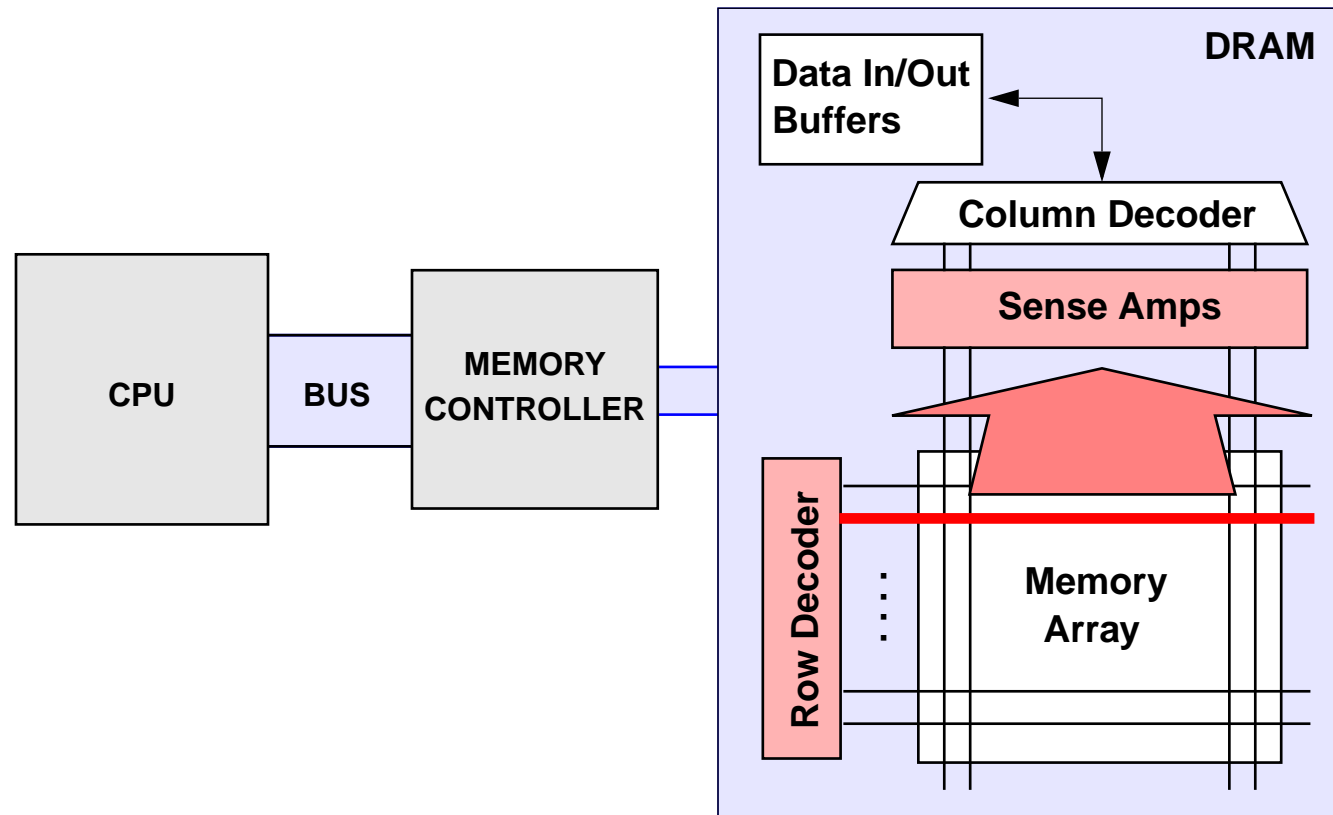
# DRAM Primer

## BUS TRANSMISSION



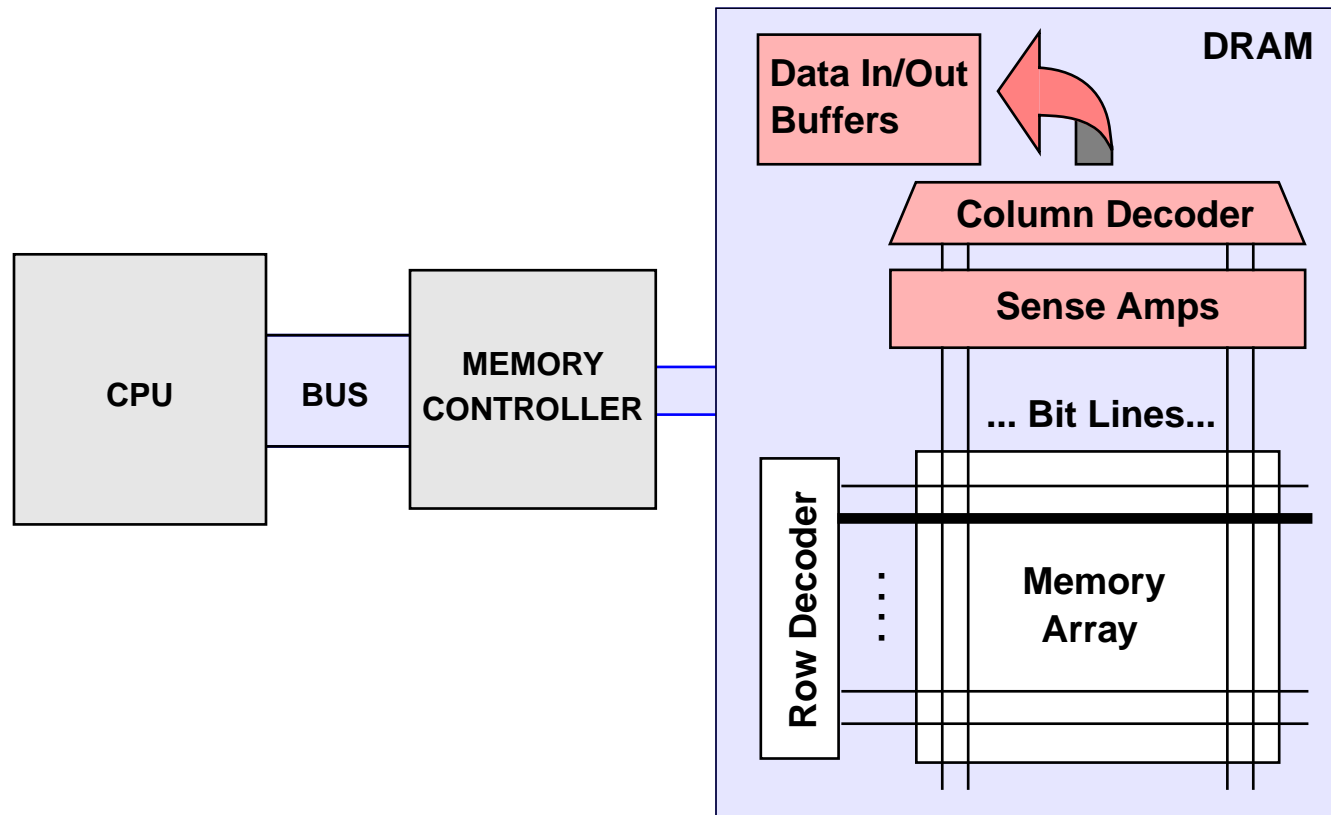
# DRAM Primer

## ROW ACCESS



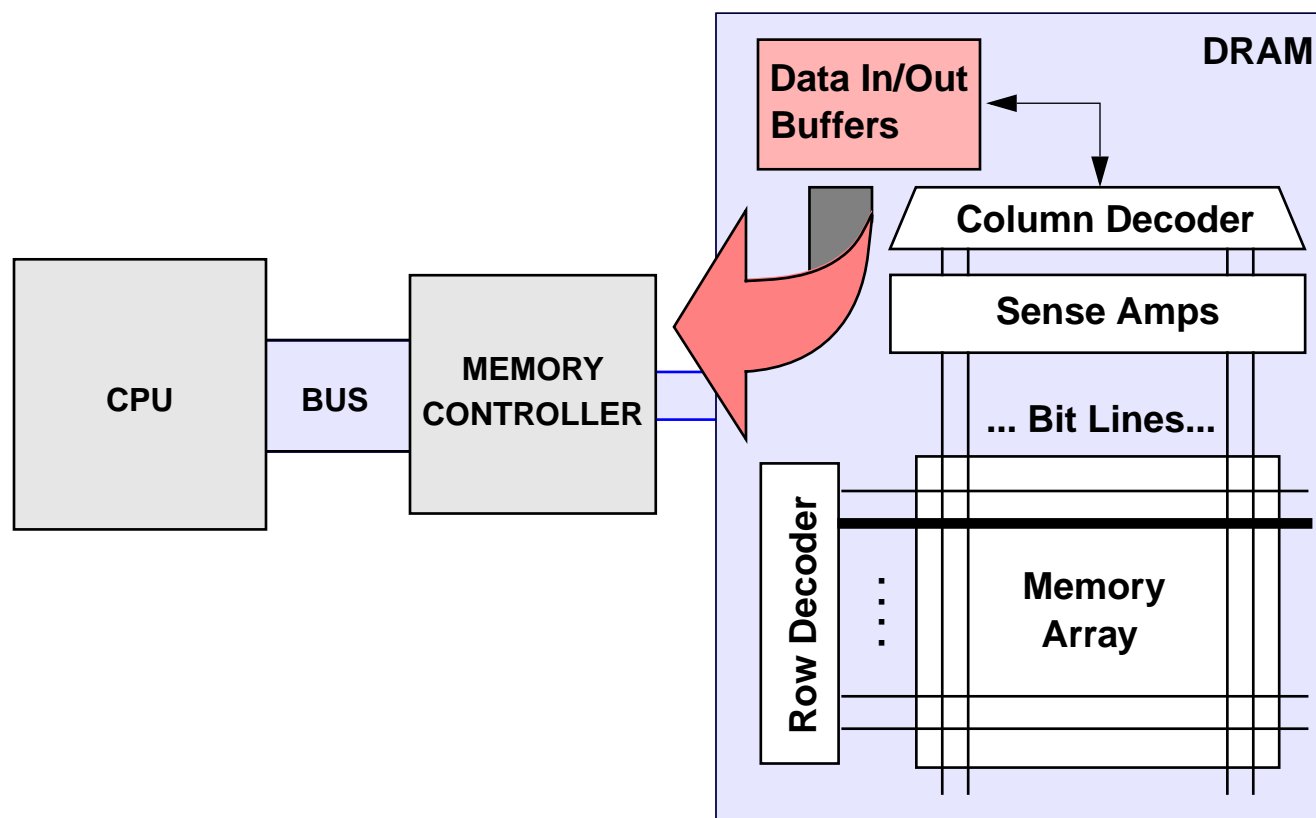
# DRAM Primer

## COLUMN ACCESS



# DRAM Primer

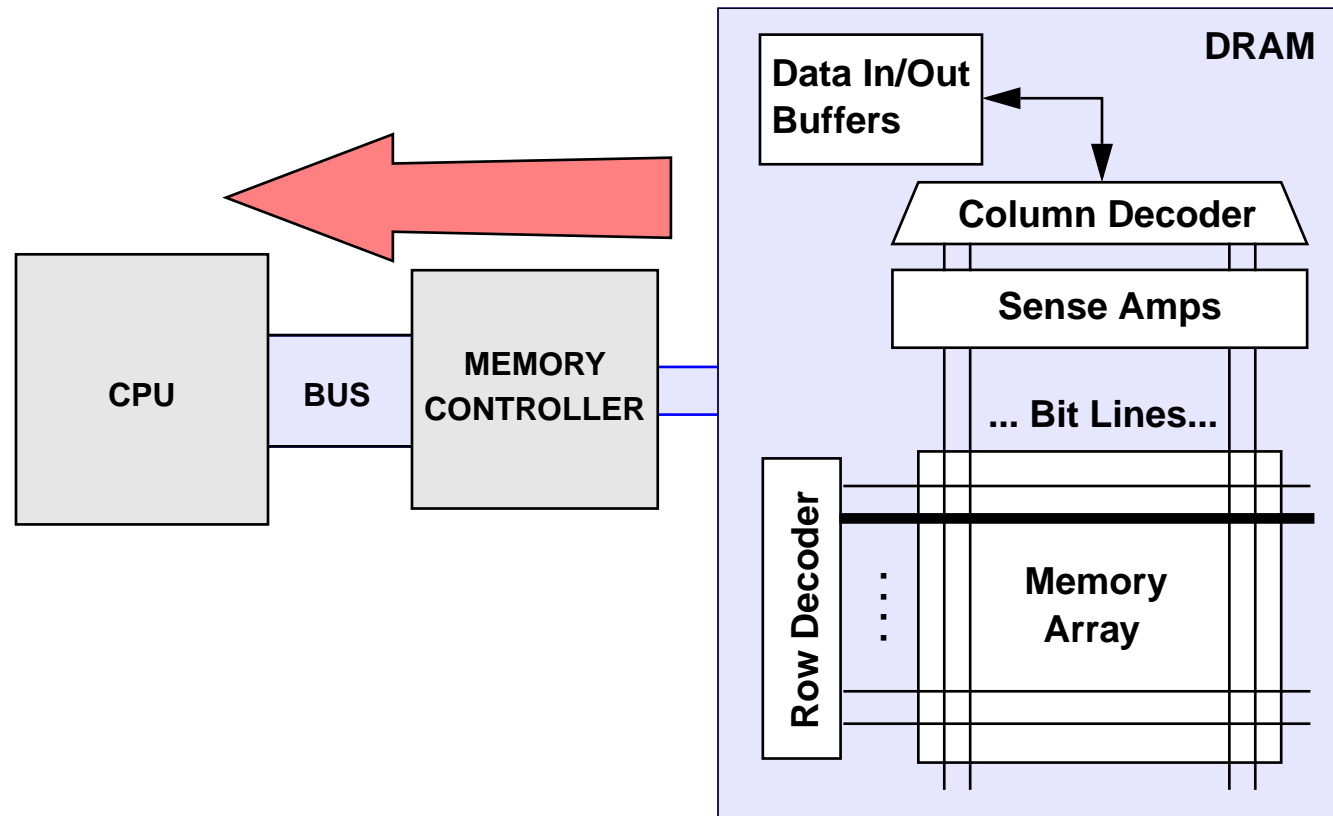
## DATA TRANSFER



**note: page mode enables overlap with COL**

# DRAM Primer

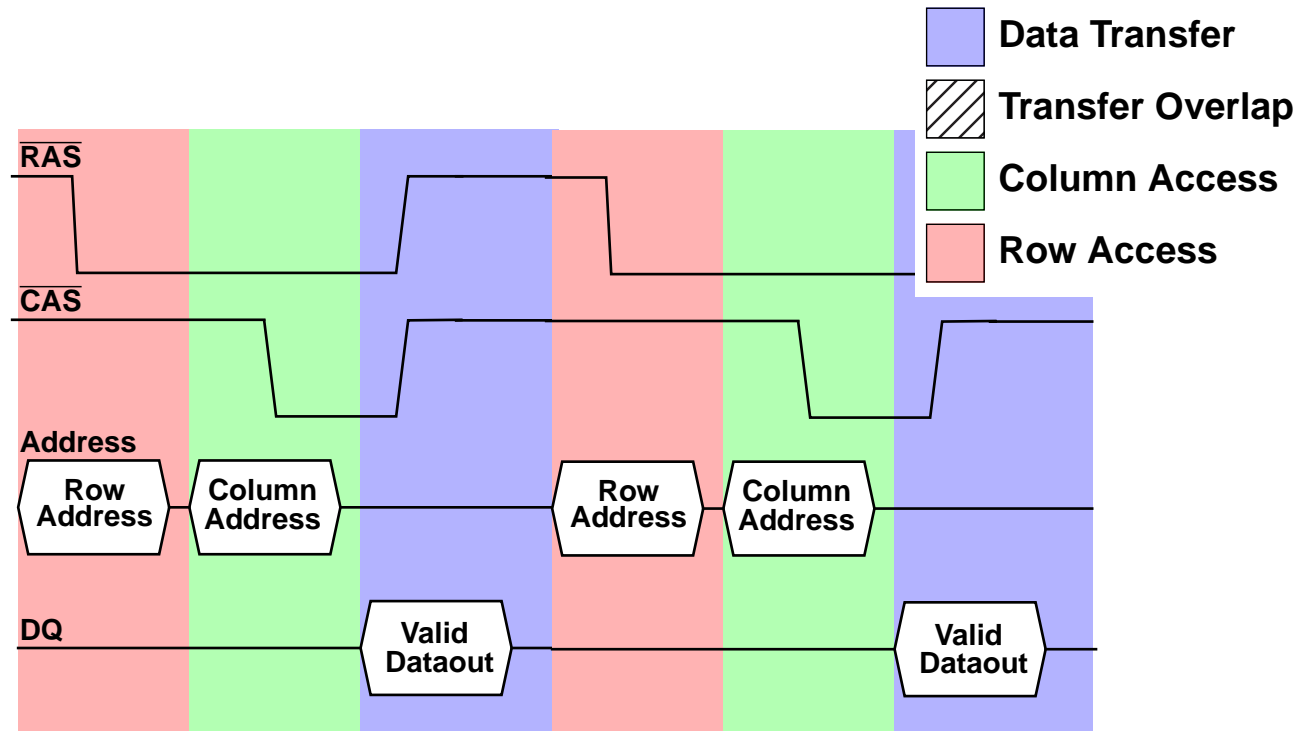
## BUS TRANSMISSION



**note: overlapped component not shown**

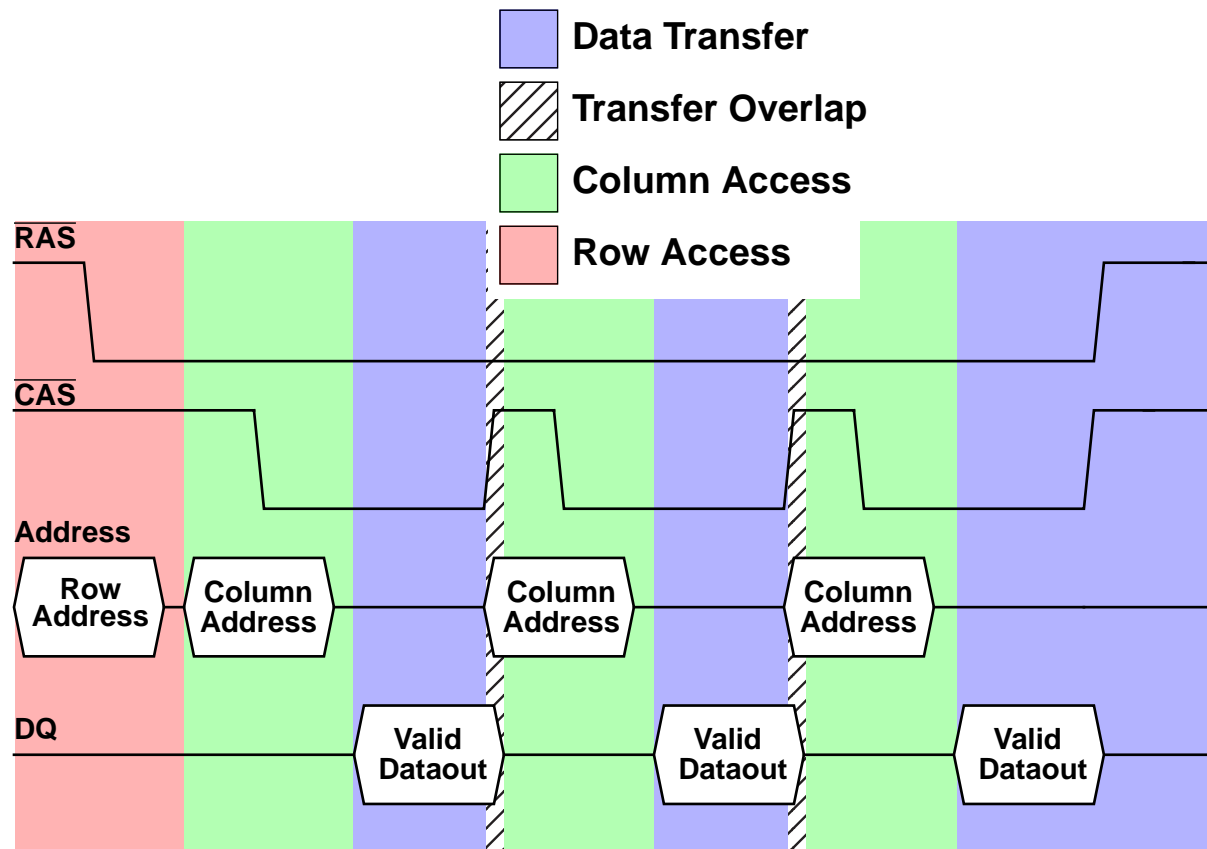
# DRAM Primer

## Read Timing for Conventional DRAM



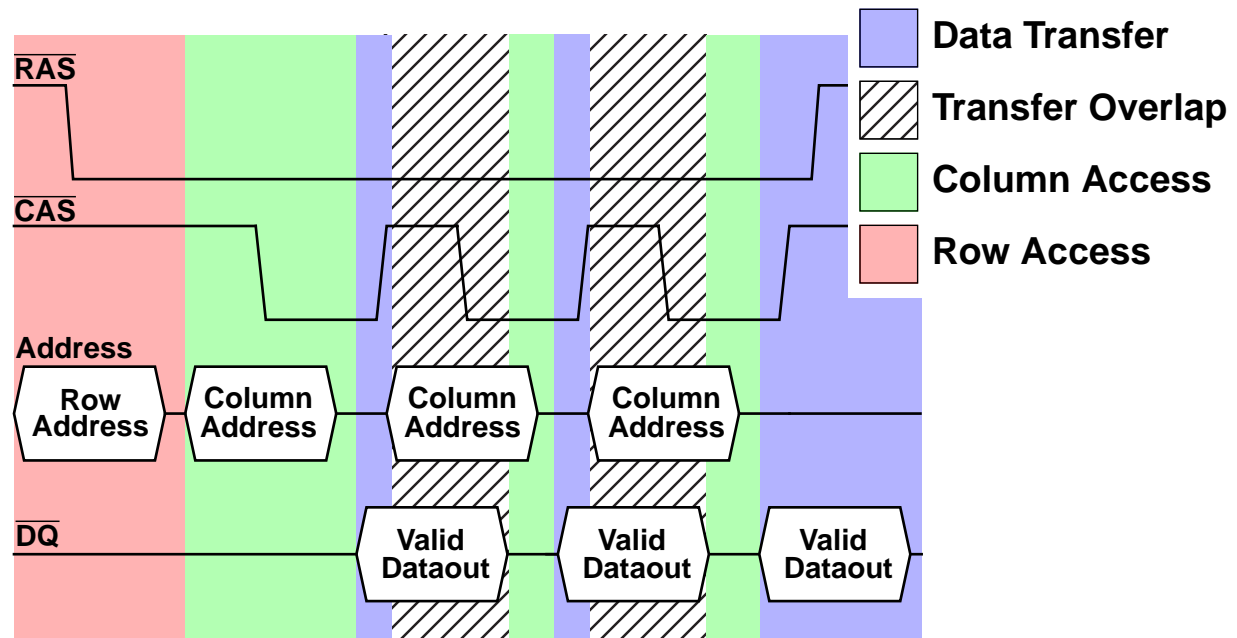
# DRAM Primer

## Read Timing for Fast Page Mode DRAM



# DRAM Primer

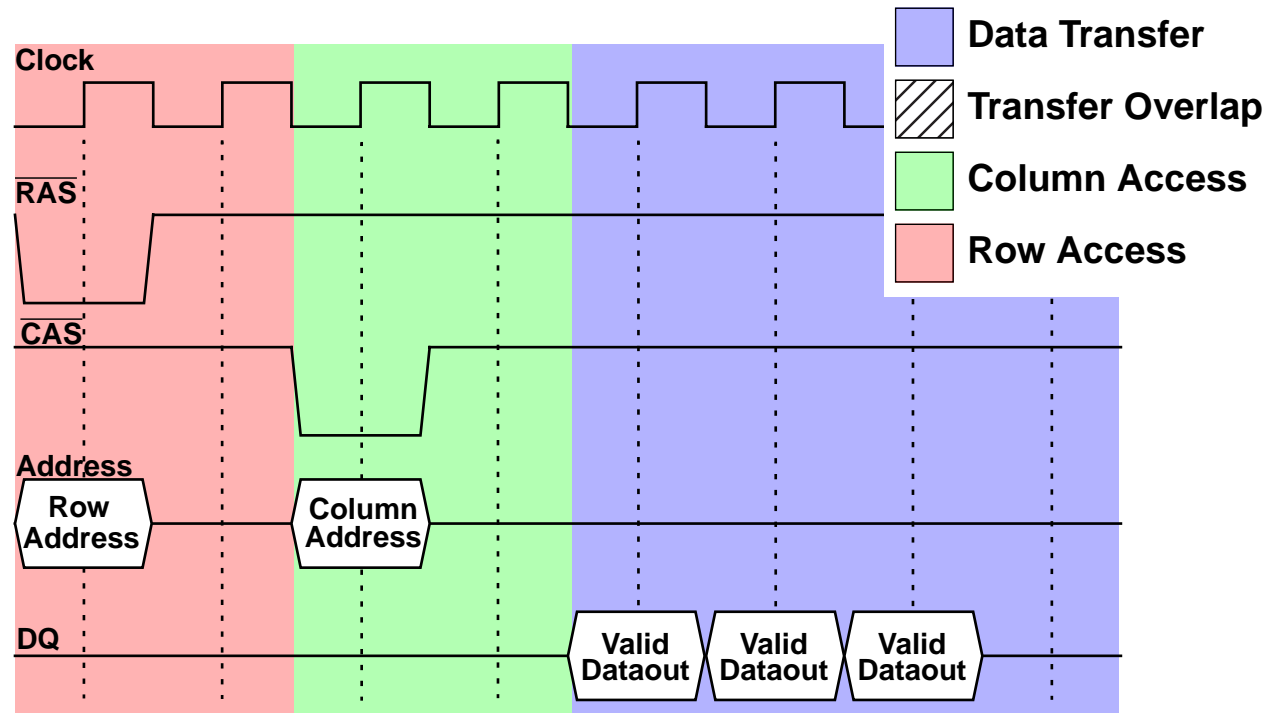
## Read Timing for Extended Data Out DRAM





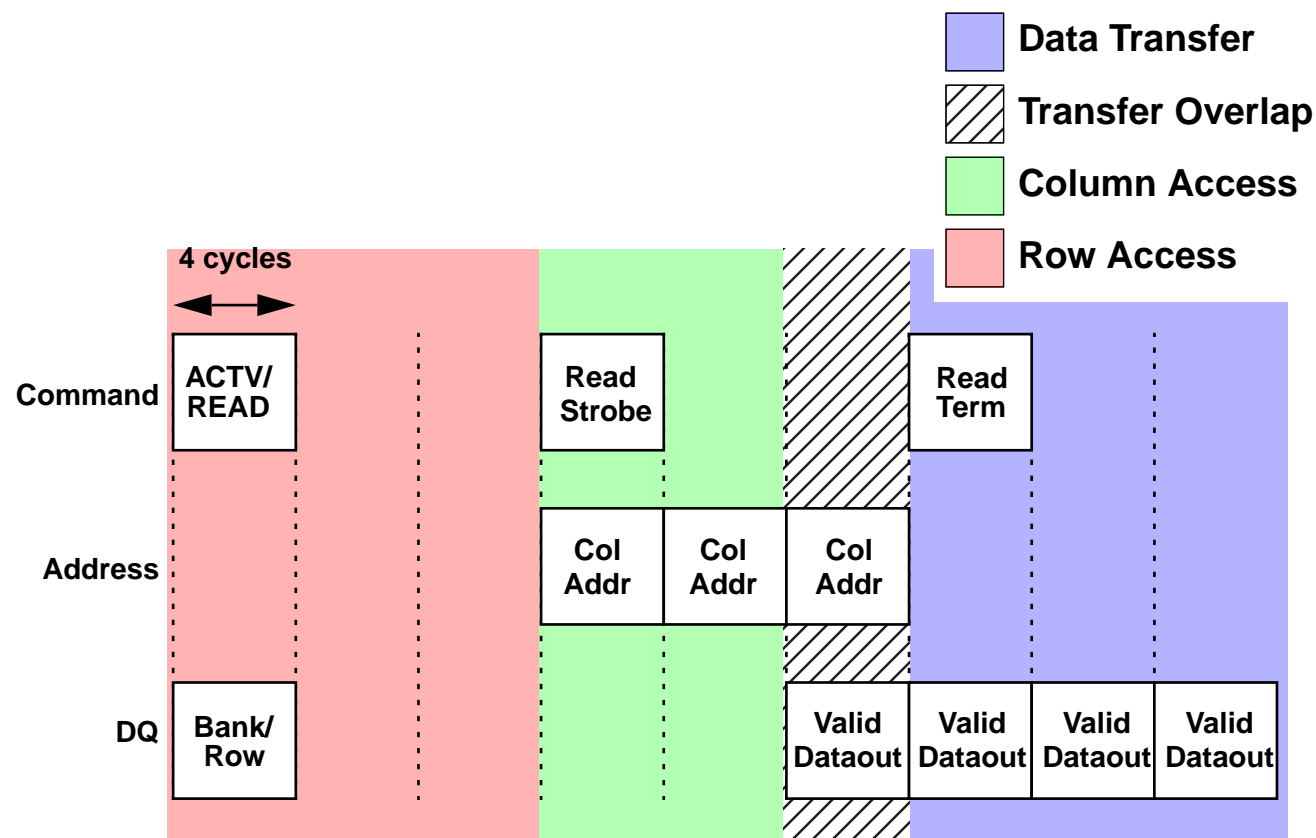
# DRAM Primer

## Read Timing for Synchronous DRAM



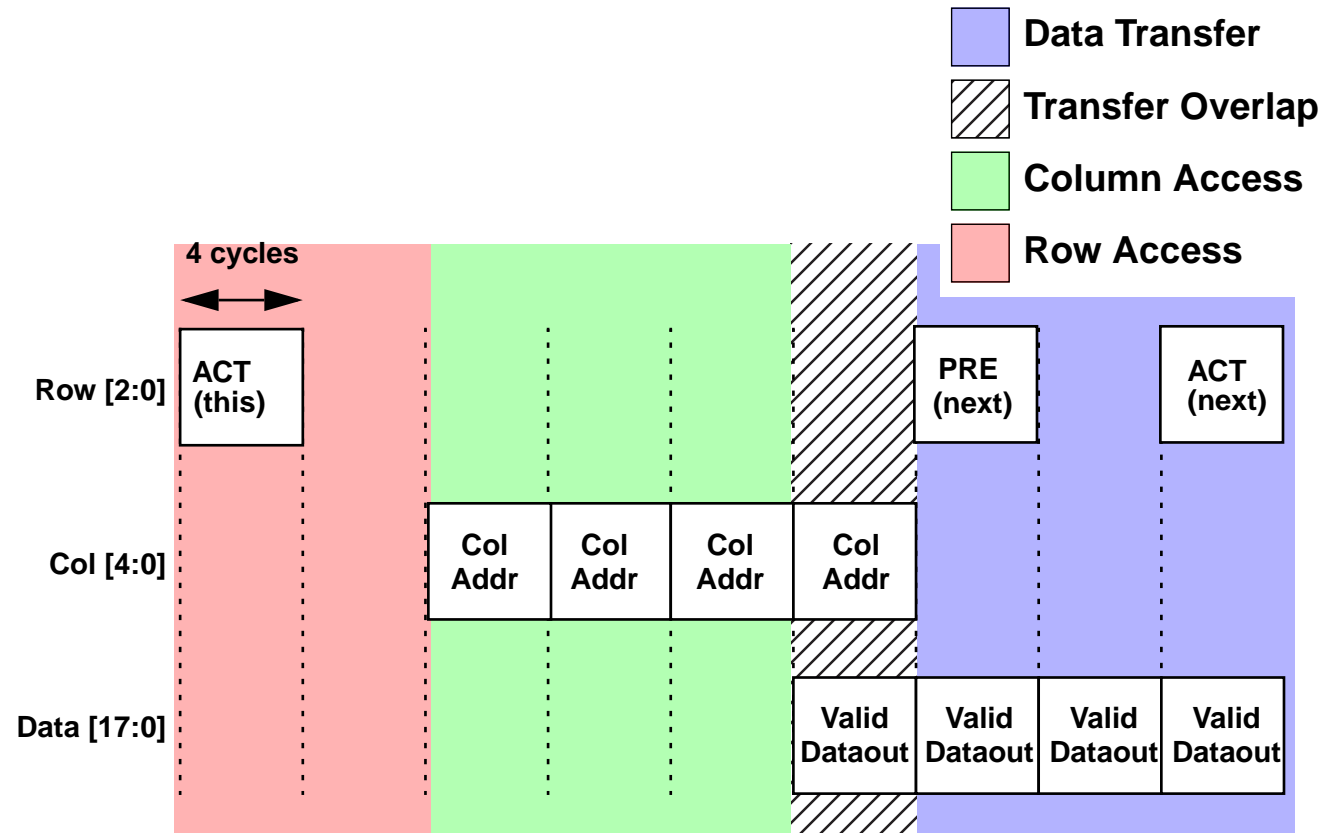
# DRAM Primer

## Read Timing for Rambus DRAM



# DRAM Primer

## Read Timing for Direct Rambus DRAM



# 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**

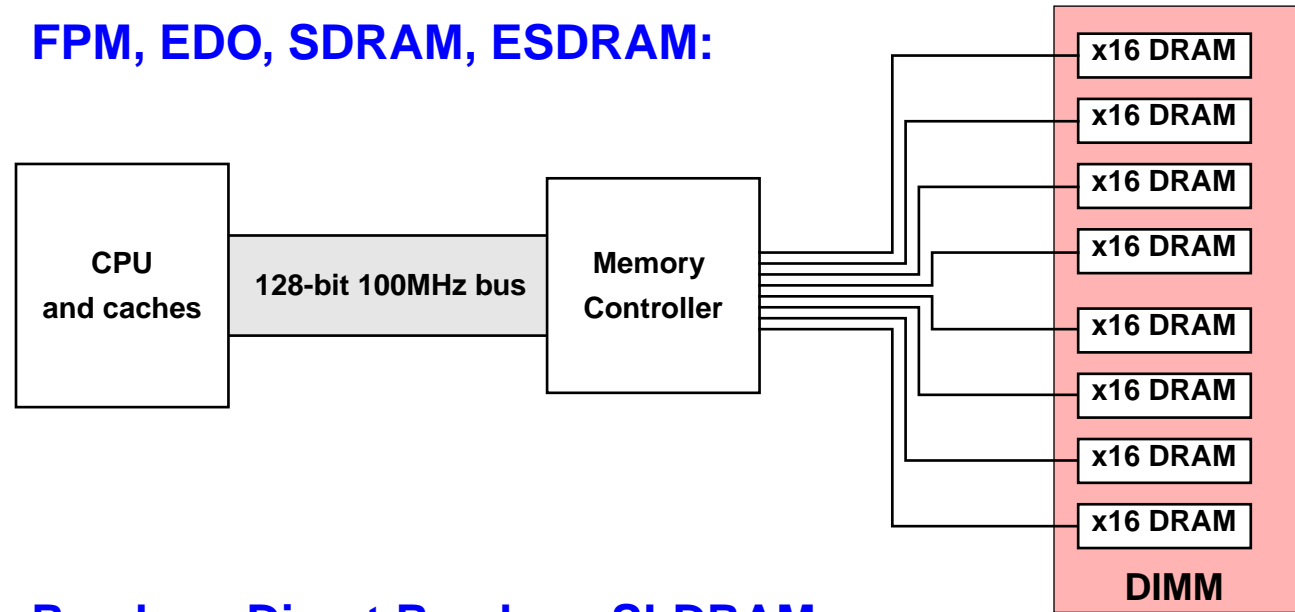
## Main Memory: 8 64Mb DRAMs

- **100MHz/128-bit memory bus**
- **Optimistic *open-page* policy  
(*close-immediately* can be calculated)**

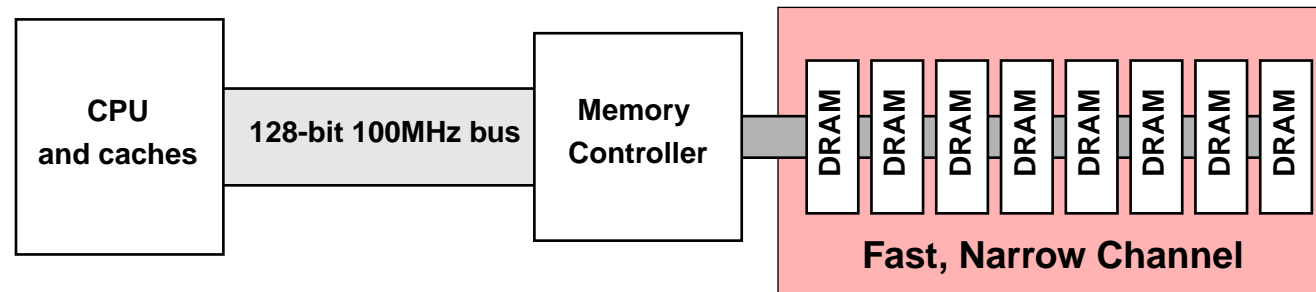
**Represents a “typical” workstation**

# DRAM Configurations

## FPM, EDO, SDRAM, ESDRAM:



## Rambus, Direct Rambus, SLDRAM:

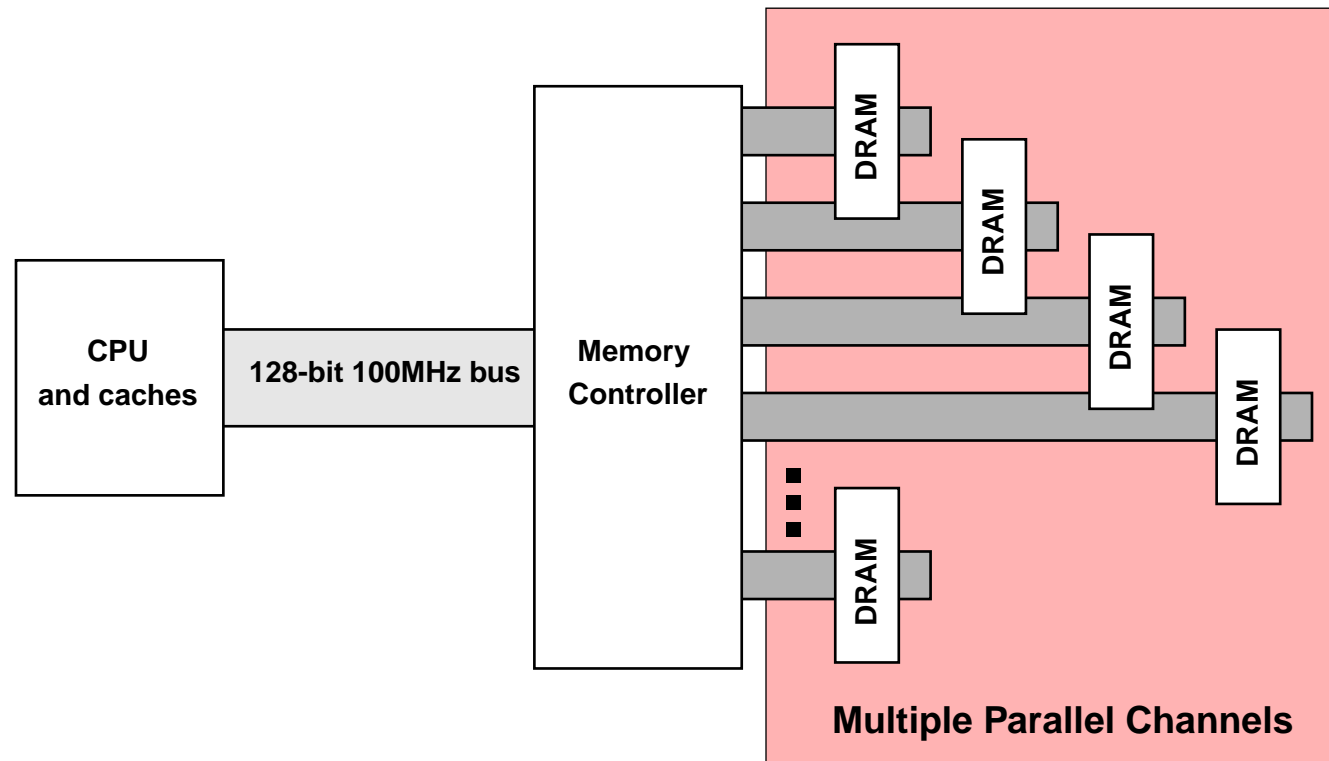


**Note: TRANSFER WIDTH of Direct Rambus Channel**

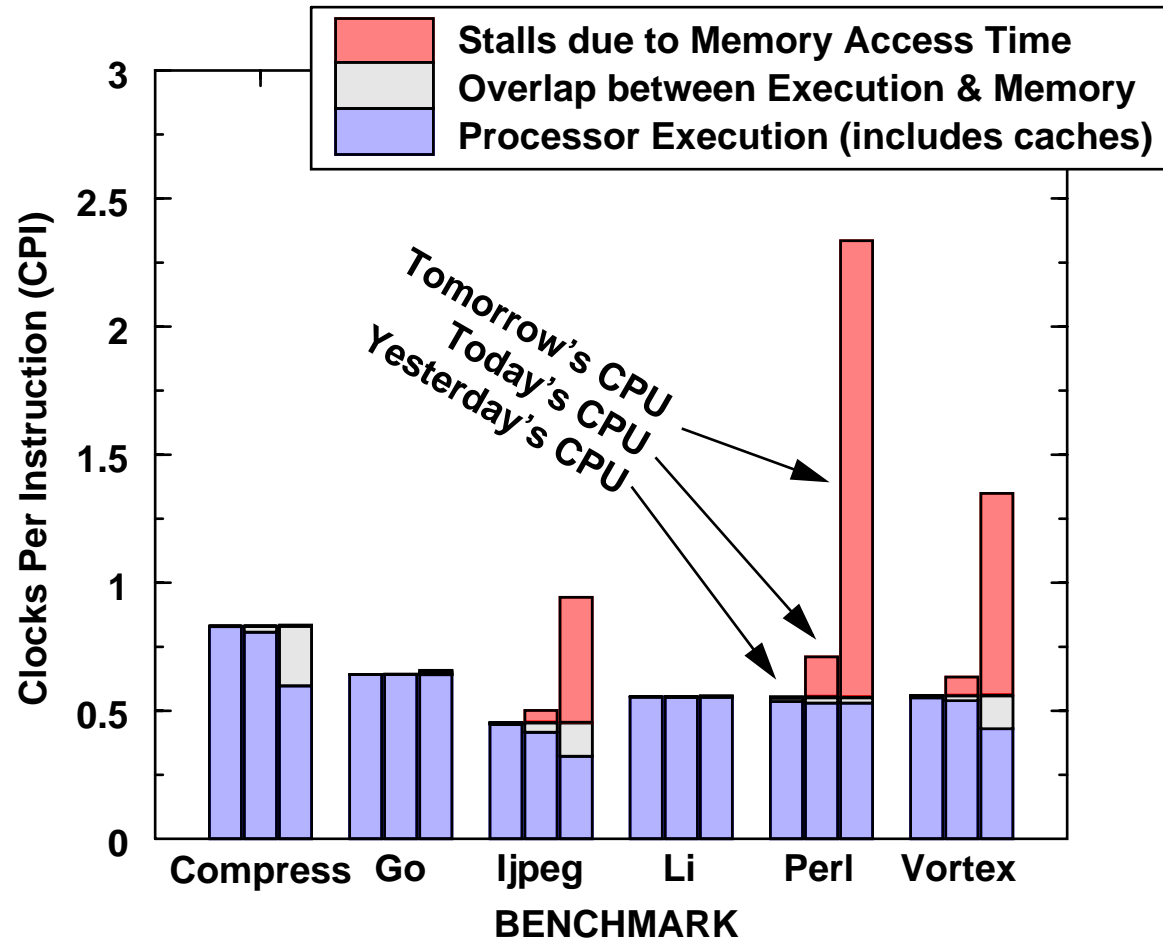
- equals that of ganged FPM, EDO, etc.
- is 2x that of Rambus & SLDRAM

# DRAM Configurations

## Strawman: Rambus, etc.



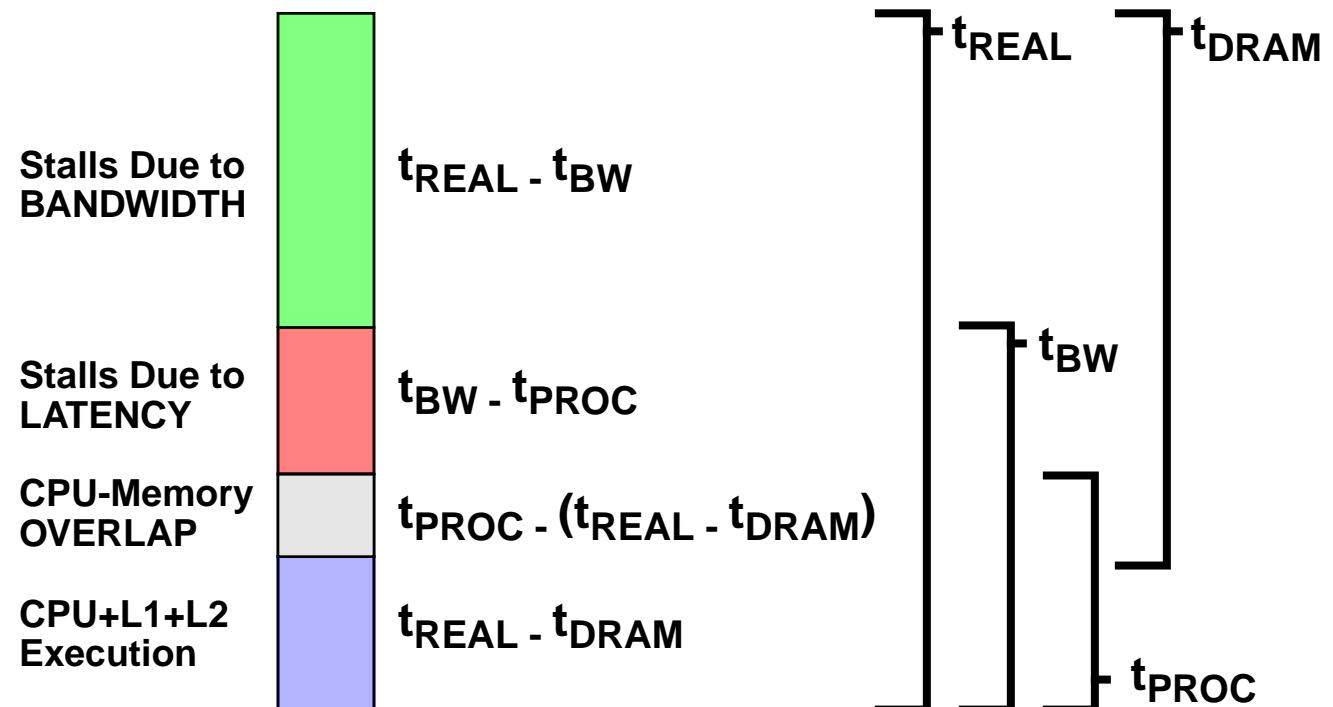
# Overhead: Memory vs. CPU



**Variable: speed of processor & caches**

# Definitions (var. on Burger, et al)

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



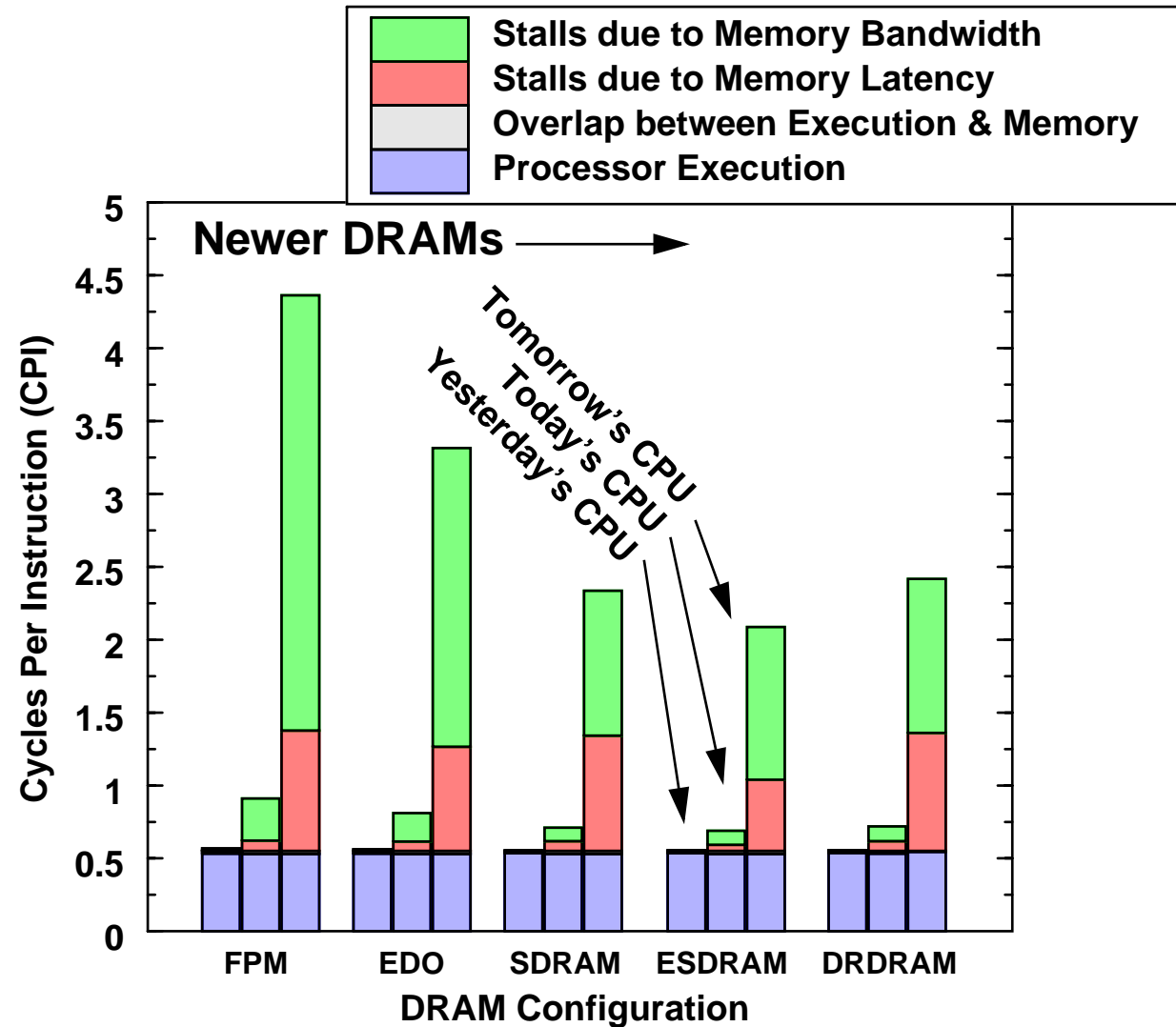


HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Memory & CPU — PERL

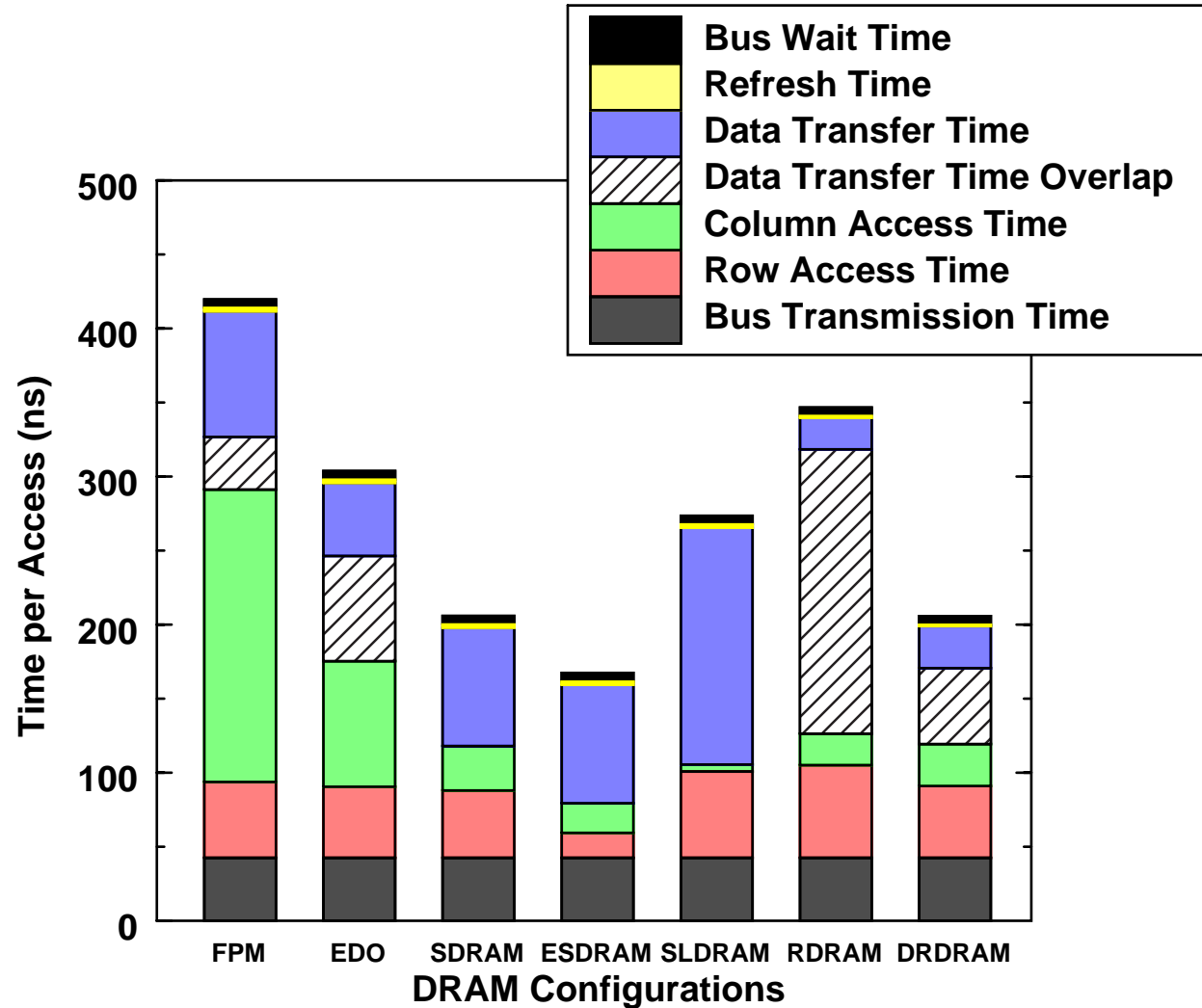


HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Average Latency of DRAMs



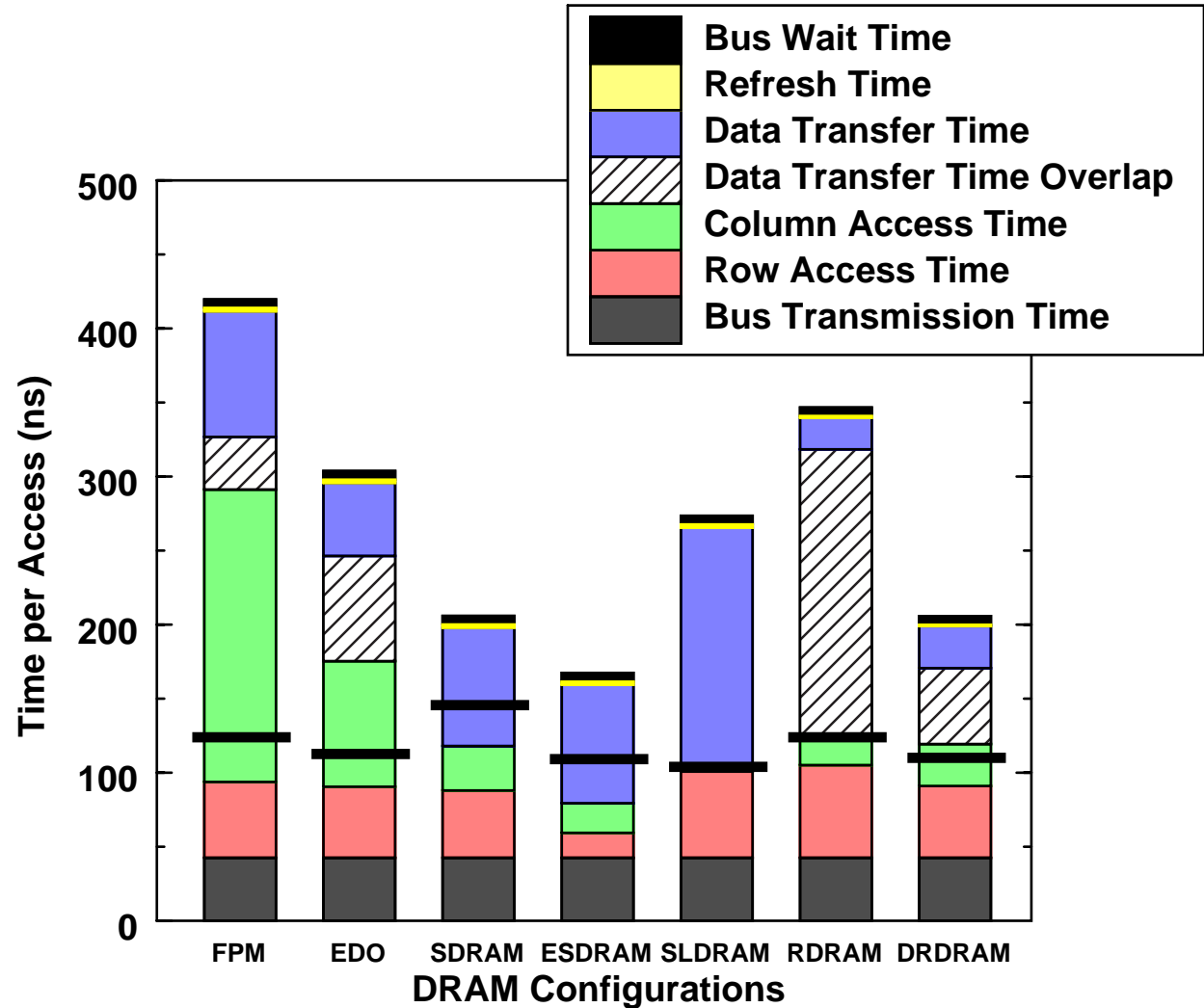
**note: SLDRAM & RDRAM 2x data transfers**

HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Average Latency of DRAMs



**note: SLDRAM & RDRAM 2x data transfers**

# Cost-Performance

## FPM, EDO, SDRAM, ESDRAM:

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

## Rambus, Direct Rambus, SLDRAM:

- **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, Sony Emotion, ...)

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

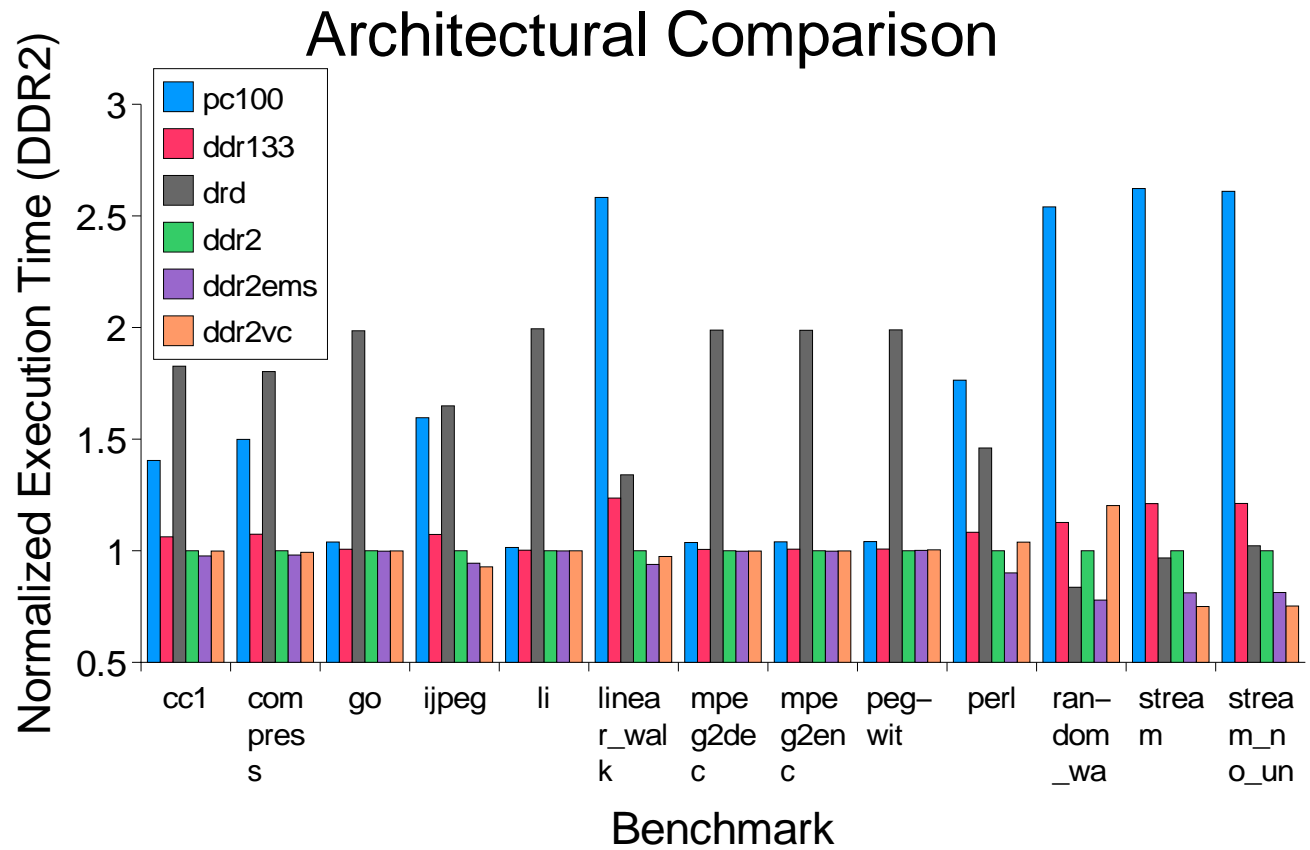
- **Solution: New cores with on-chip SRAM**  
(*e.g.* ESDRAM, VCDRAM, ...)
- **Solution: New cores with smaller banks**  
(*e.g.* MoSys “SRAM”, FCRAM, ...)

# Recent Work

**Detailed Study of DDR2 Proposals  
in Concurrent Environment, Including  
Comparison with DRDRAM**

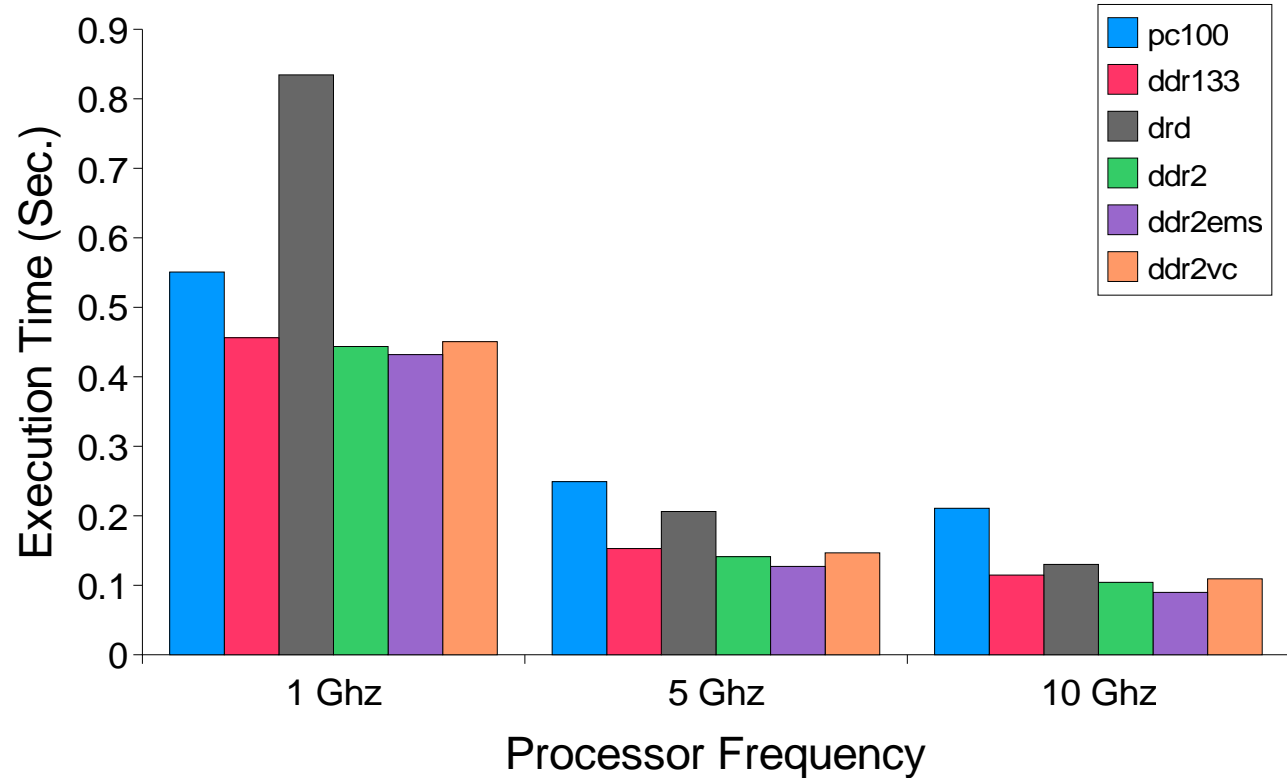
**Highly Concurrent System Organizations  
(Multiple Channels, Queueing Mechanisms,  
Priority Schemes, Optimal Burst Sizes)**

# DDR2 Study Results



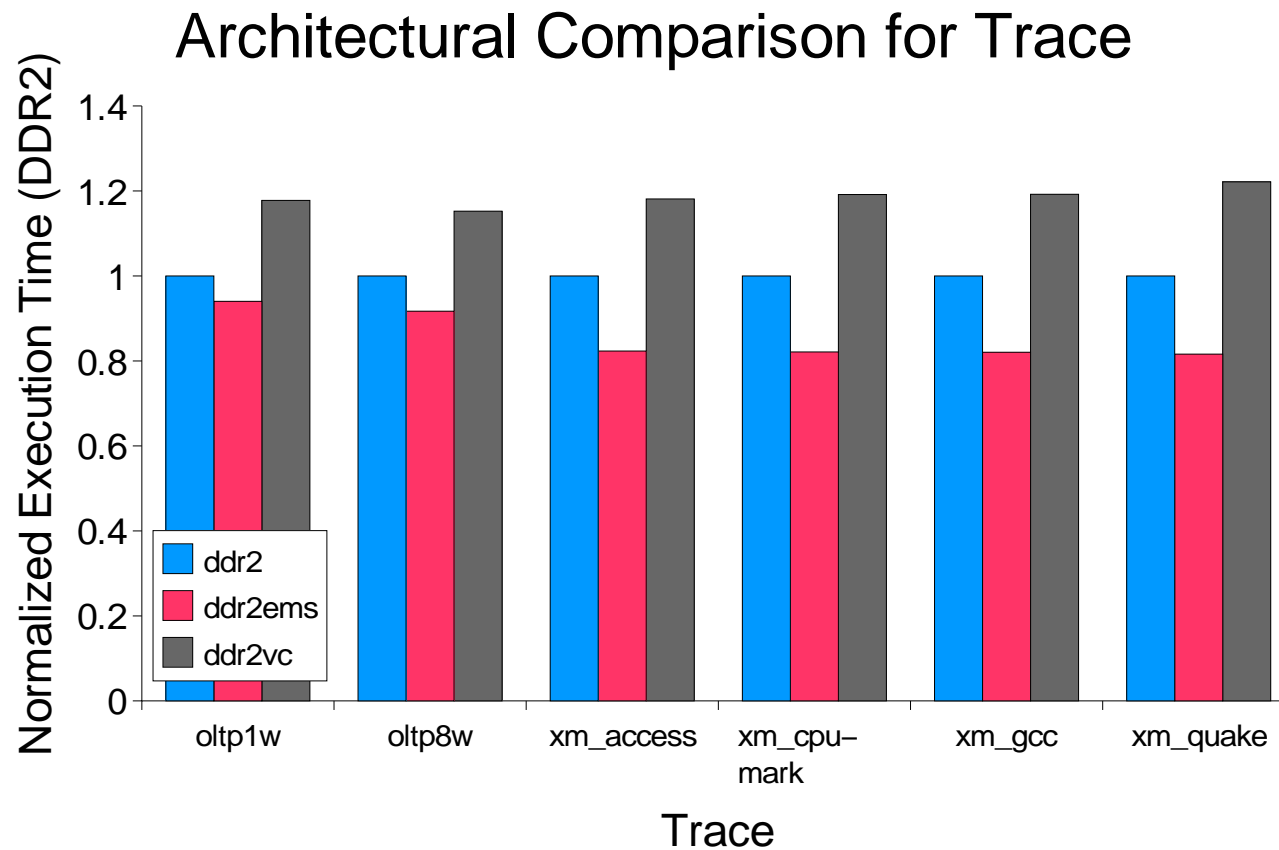
# DDR2 Study Results

## Perl Runtime

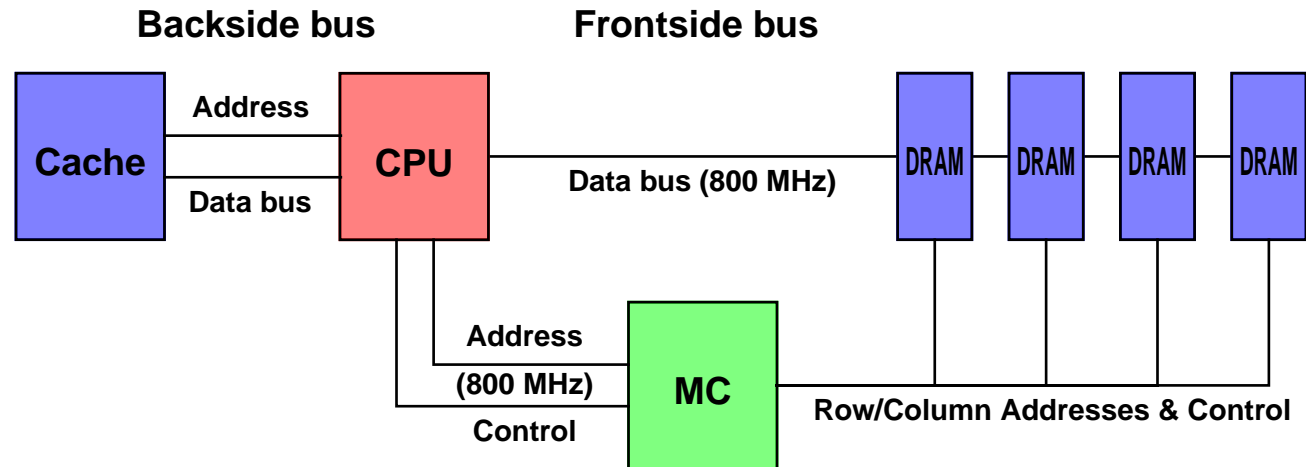




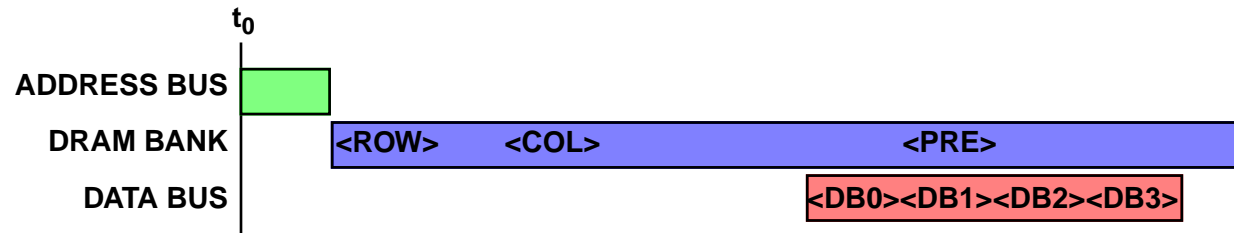
# DDR2 Study Results



# Concurrency Study: Timing

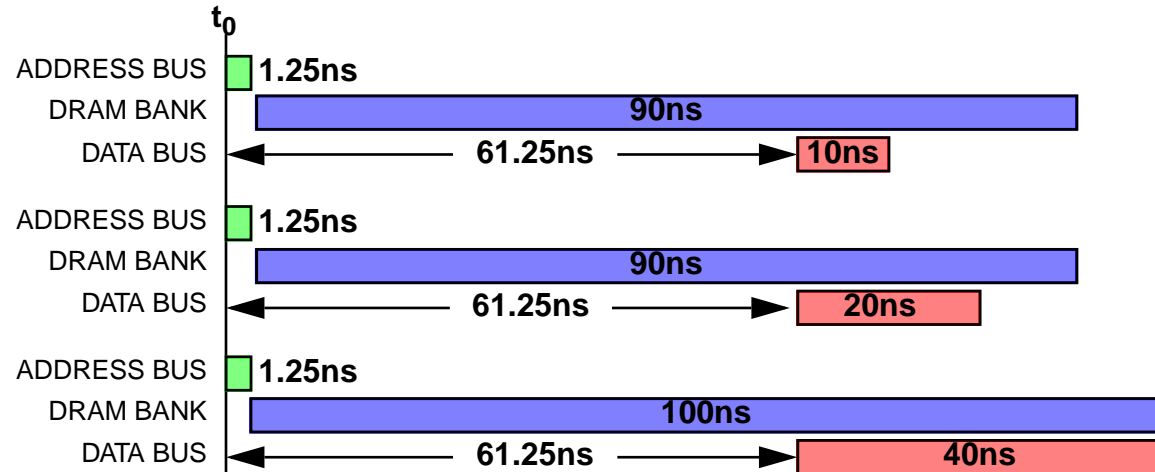


## READ REQUEST TIMING:

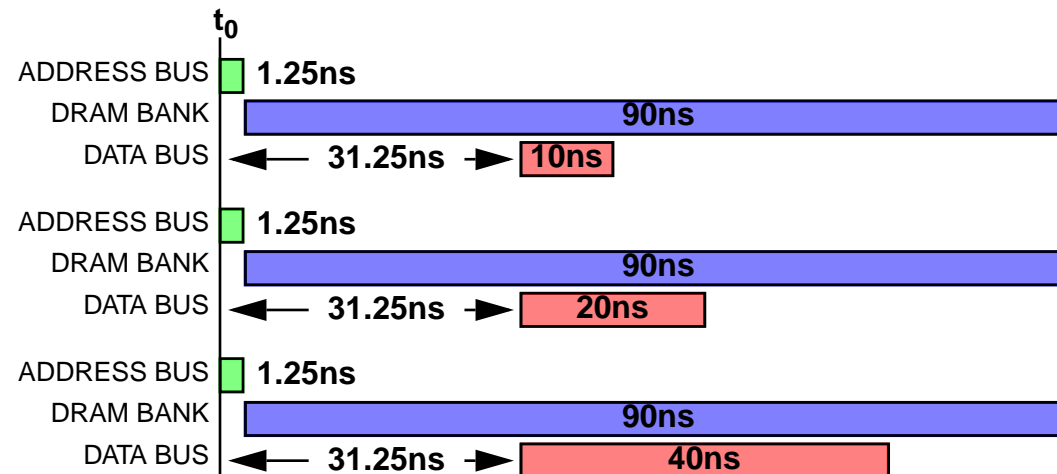


# Read/Write Request Shapes

## READ REQUESTS:

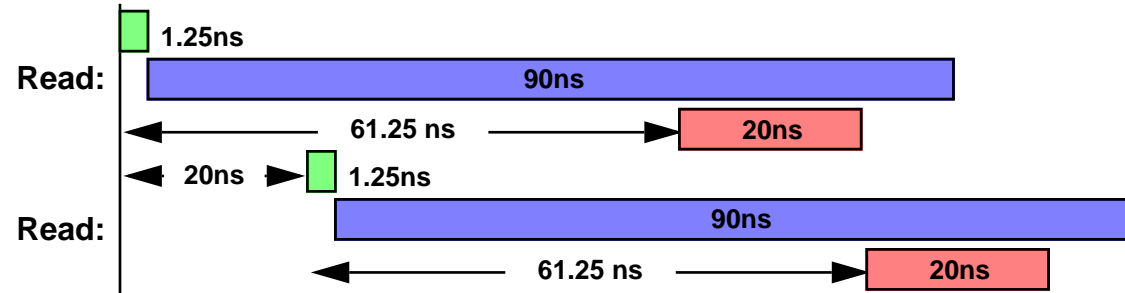


## WRITE REQUESTS:

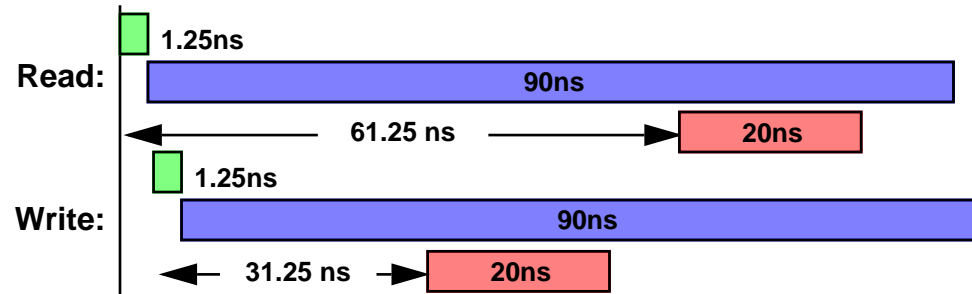


# Pipelined/Split Transactions

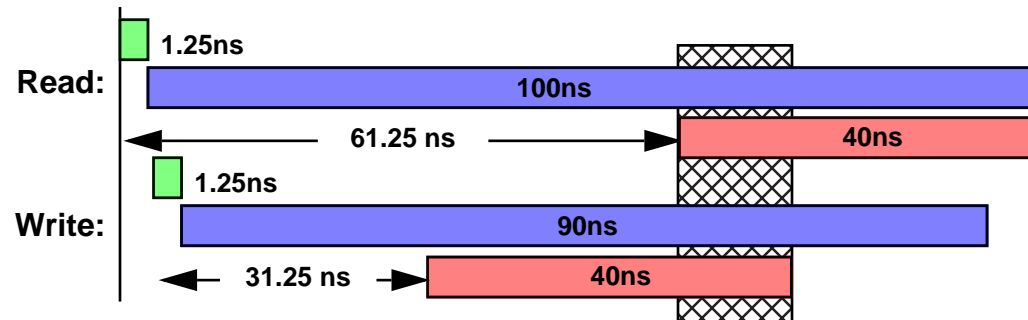
(a) Legal if R/R to different banks:



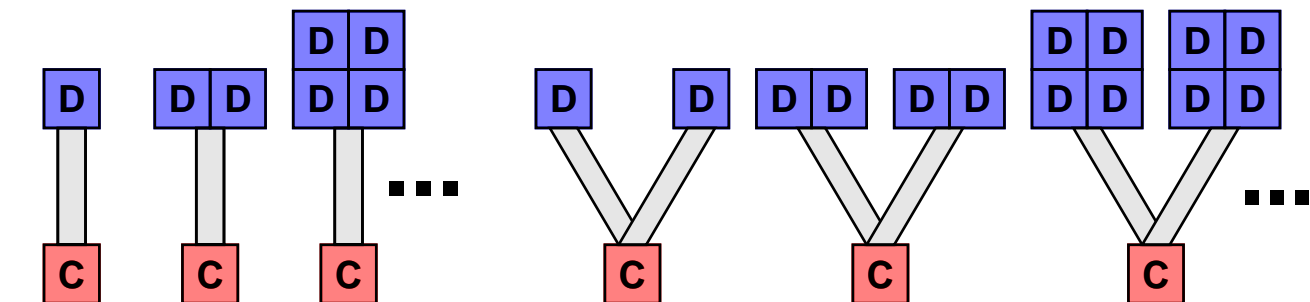
(b) Legal if turnaround  $\leq 8.75\text{ns}$  and R/W to different banks:  
(note: write can start up to 7.5ns later if turnaround = 1.25ns)



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

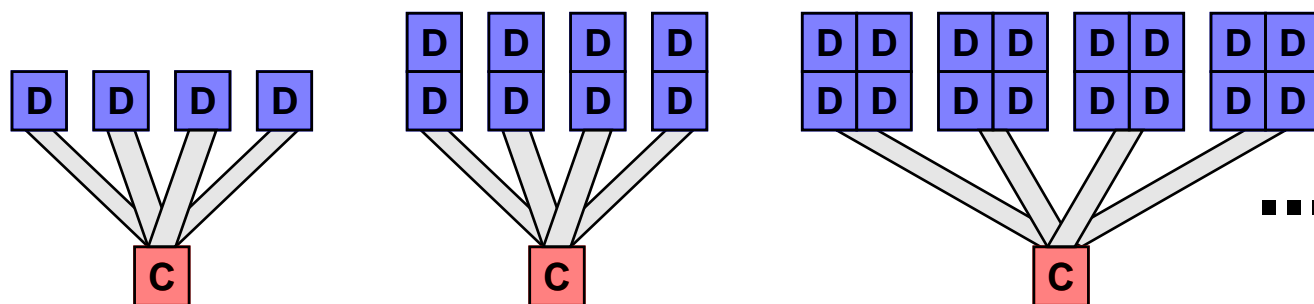


# Channels & Banks



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:



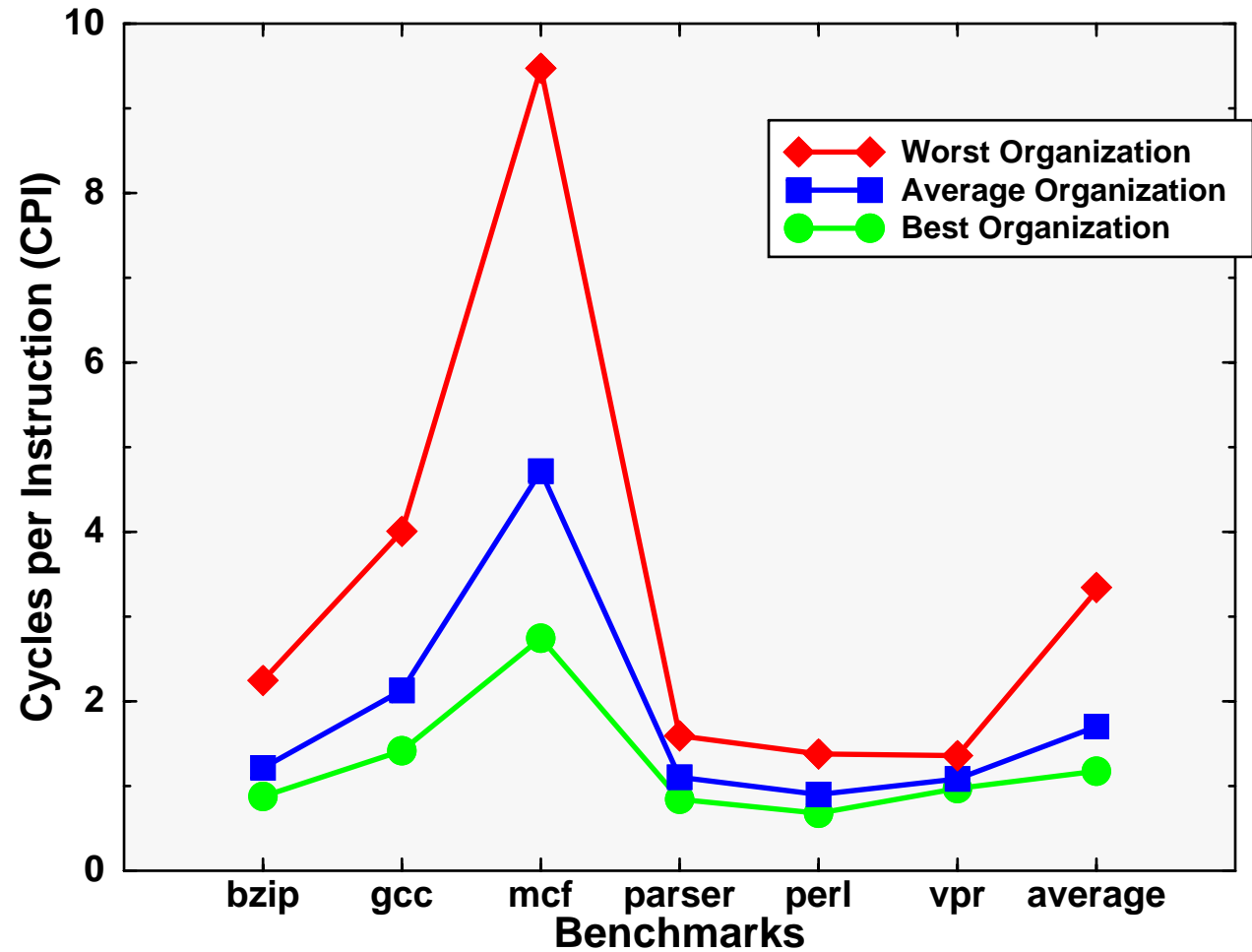
- **Critical-burst-first**
- **Non-critical bursts are promoted**
- **Writes have lowest priority  
(tend back up in request queue ...)**
- **Tension between large & small bursts:  
amortization vs. faster time to data**

HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

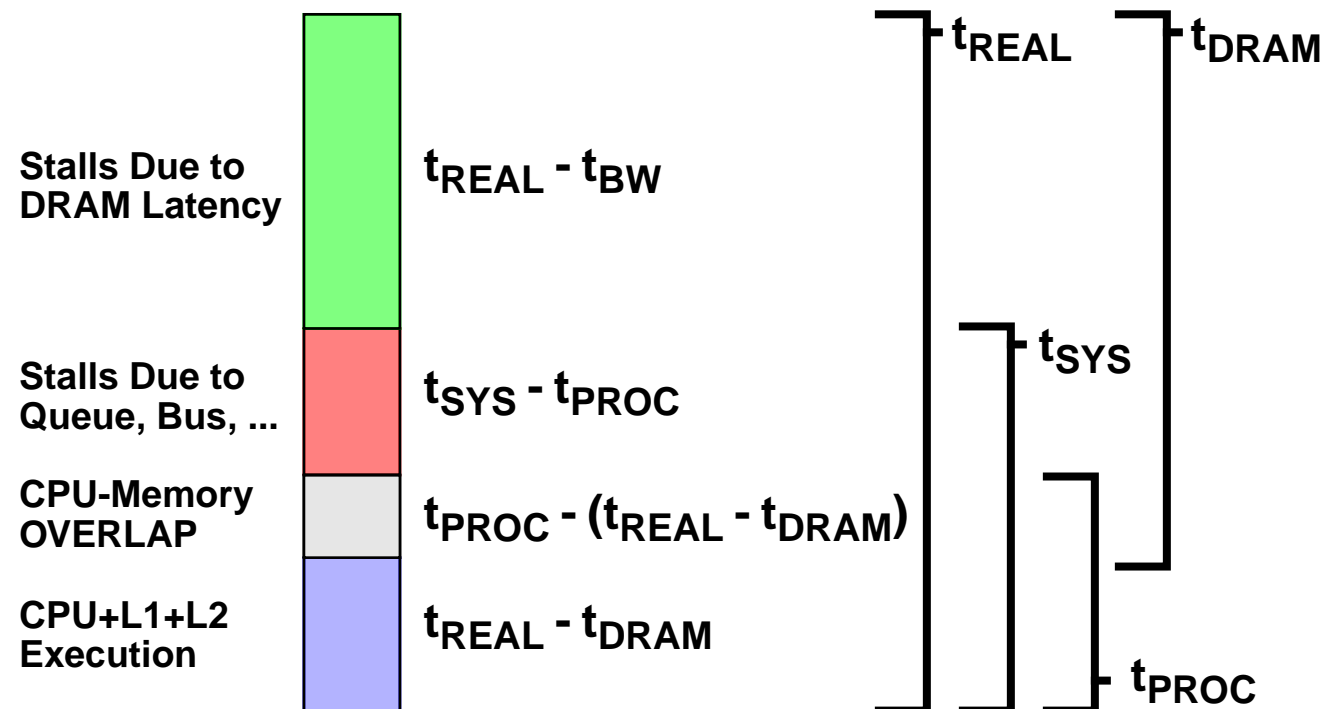
University of  
Maryland

# The Bottom Line



# New Bar-Chart Definition

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



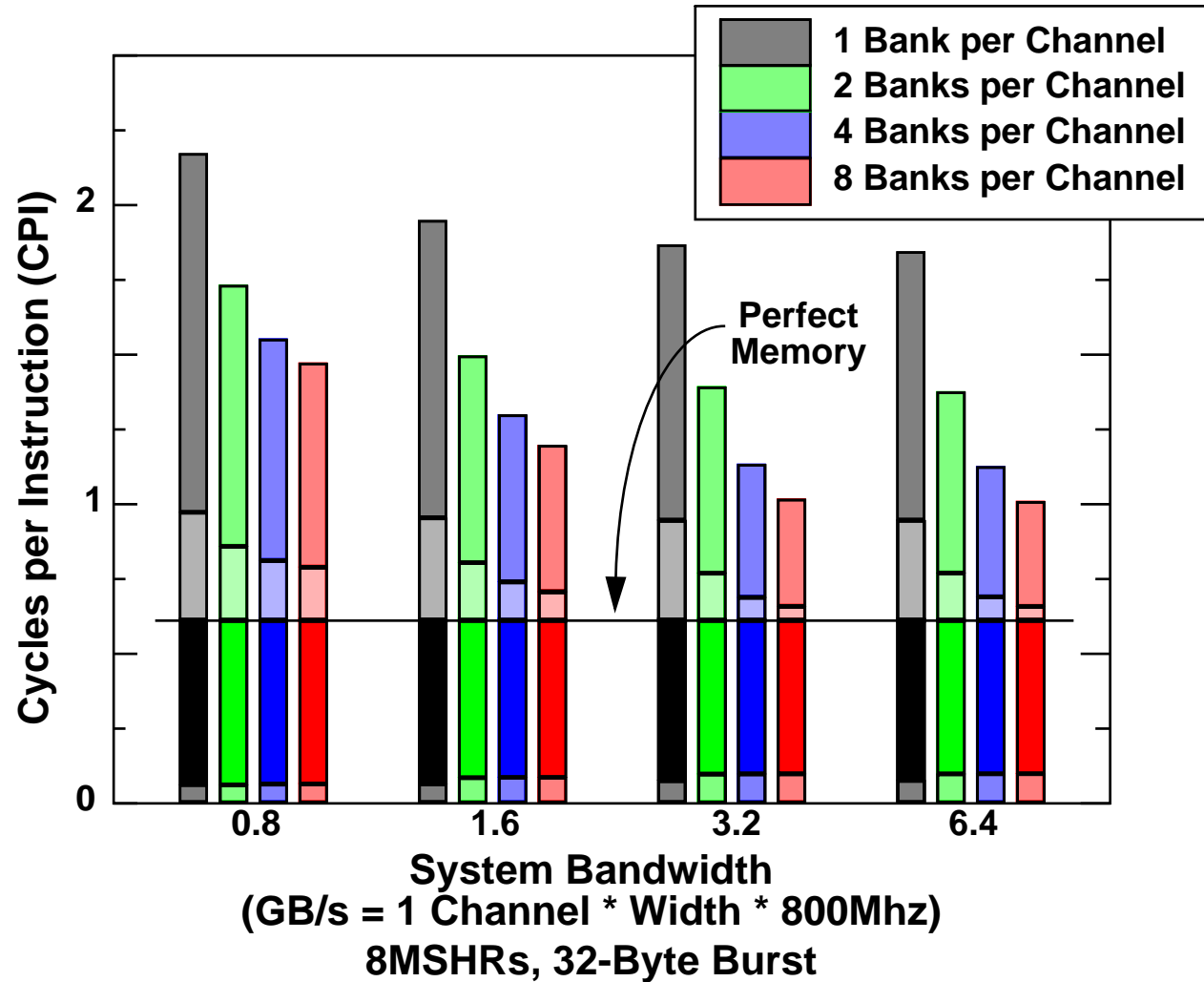


HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# System Overhead



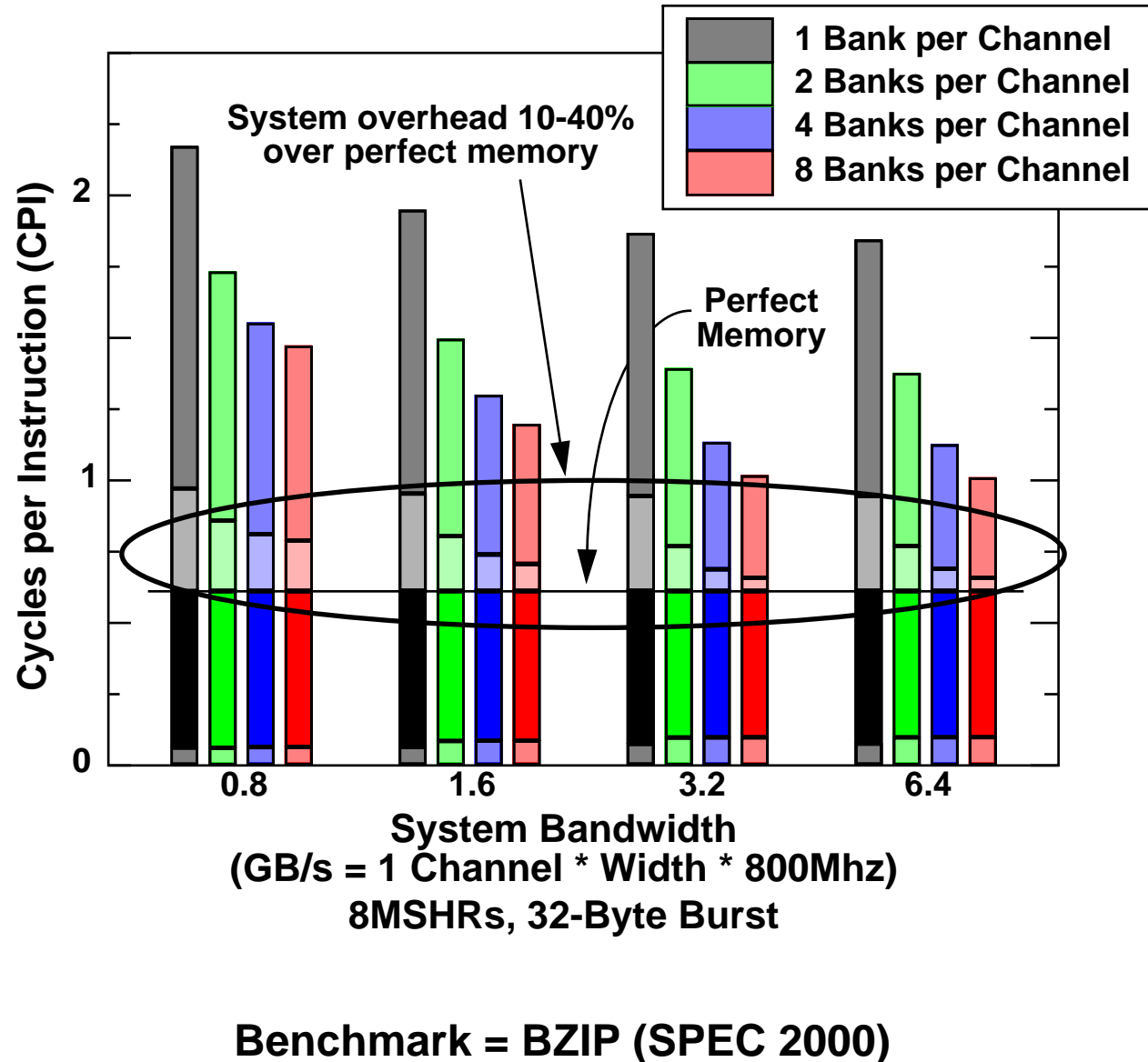
Benchmark = BZIP (SPEC 2000)

HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# System Overhead

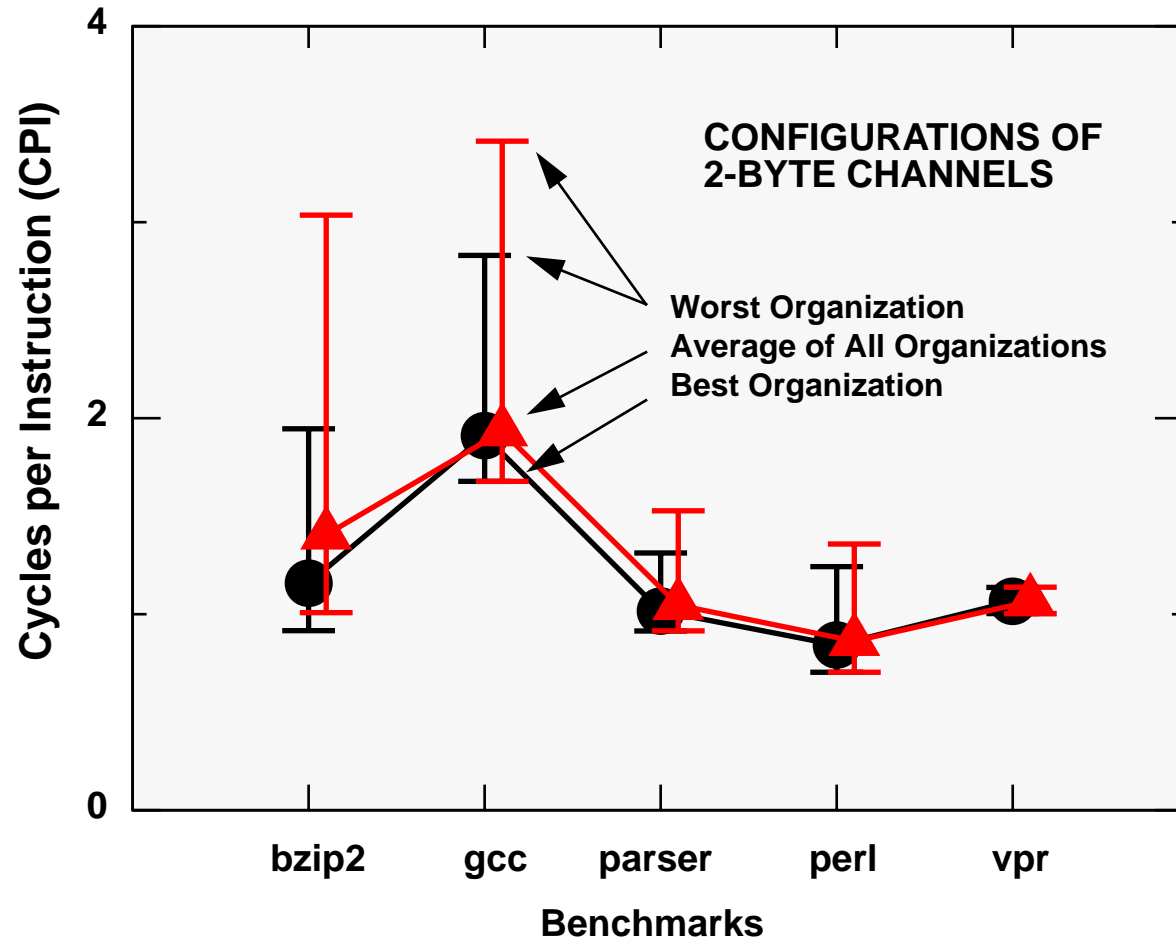


HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# It's Not Queue Size ...



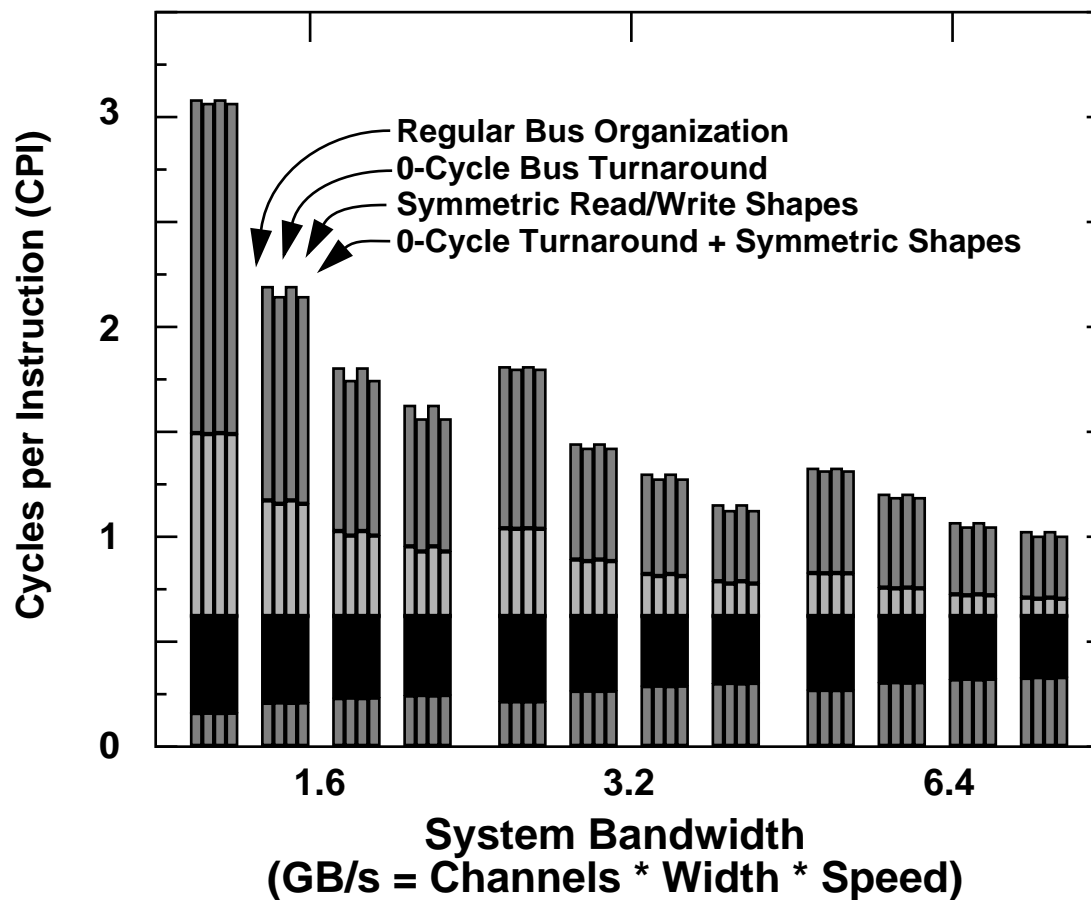
**Black = infinite request queue,  
Red = 32-entry request queue**

HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# ... It's Also Not Turnaround ...



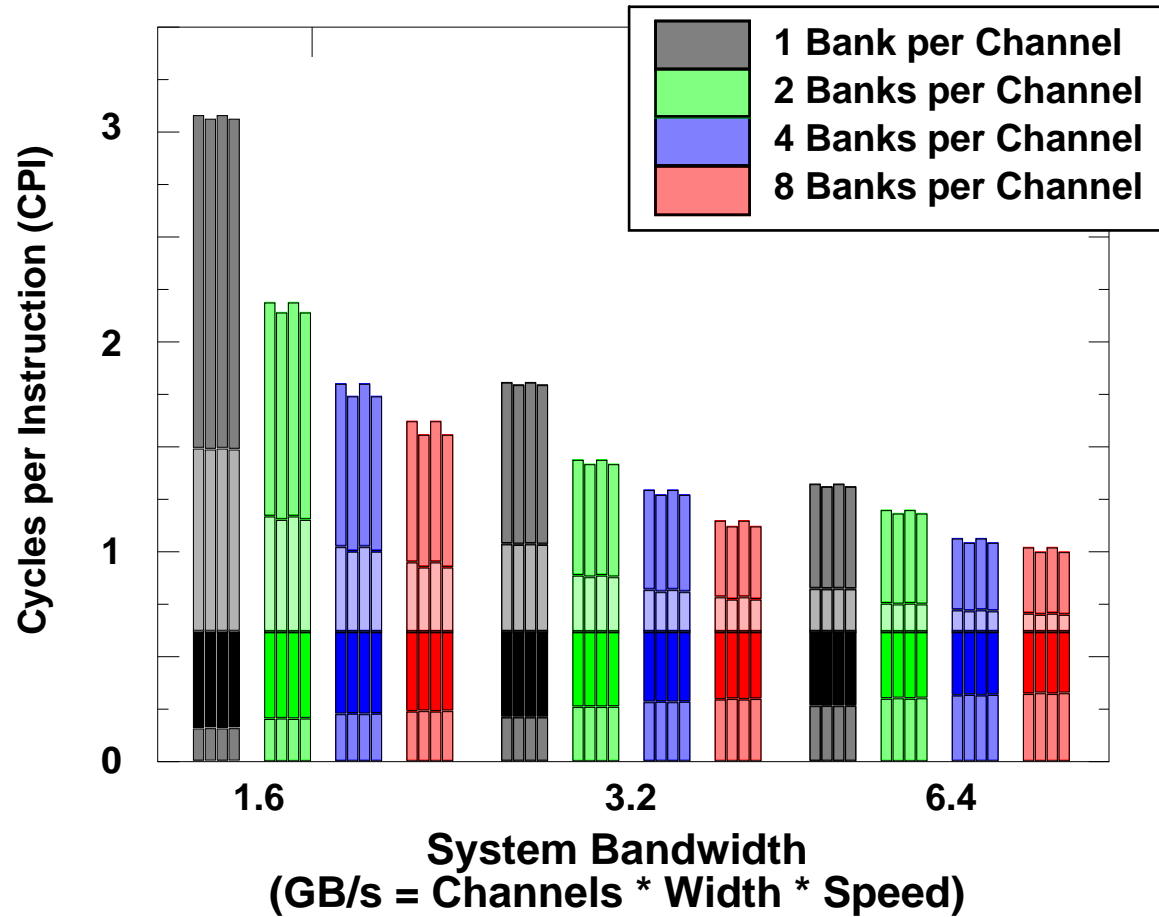
Benchmark = BZIP (SPEC 2000), 32-byte burst, 16-bit bus

HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# ... It's Related to **Concurrency**



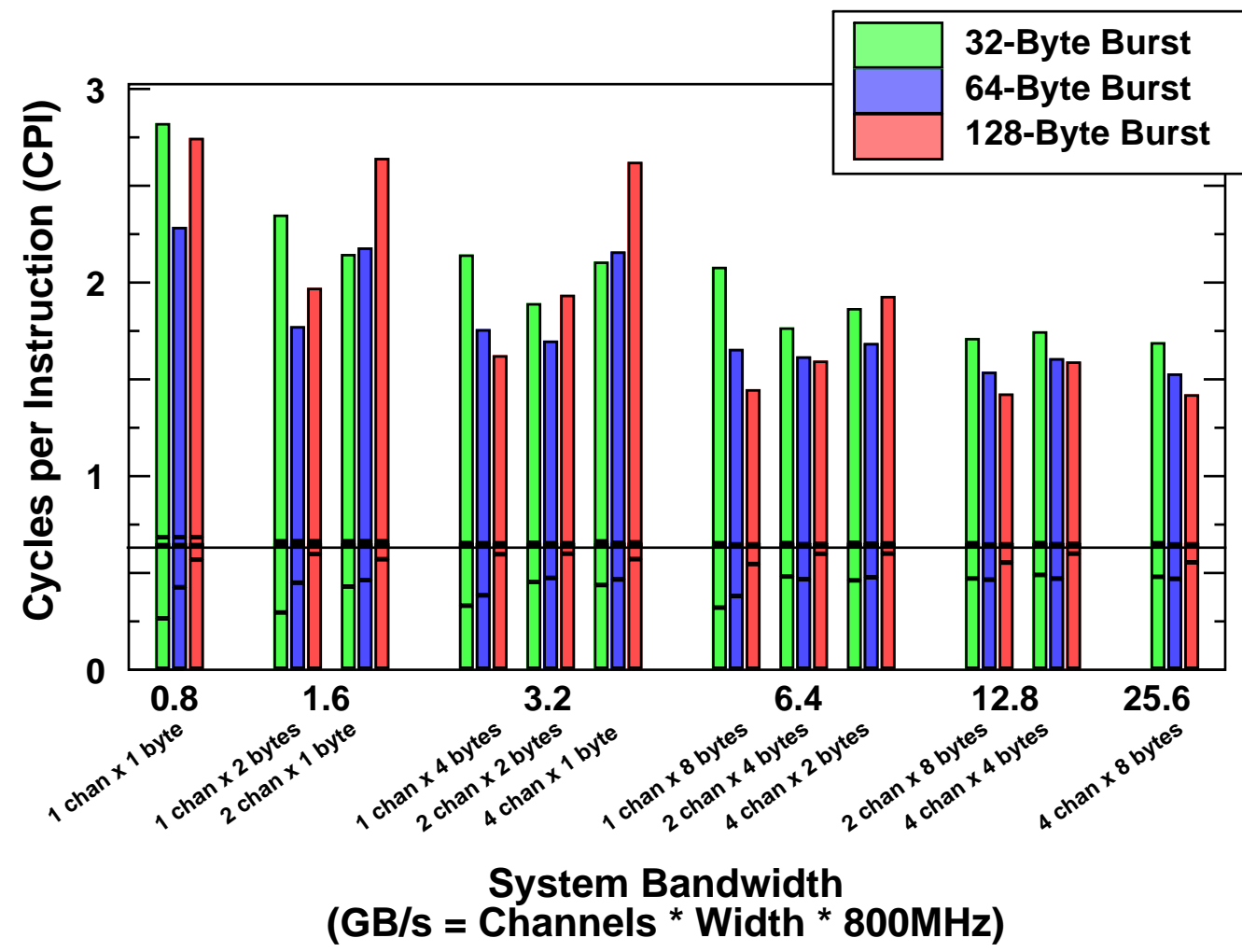
Benchmark = BZIP (SPEC 2000), 32-byte burst, 16-bit bus

HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Bandwidth vs. Burst Width



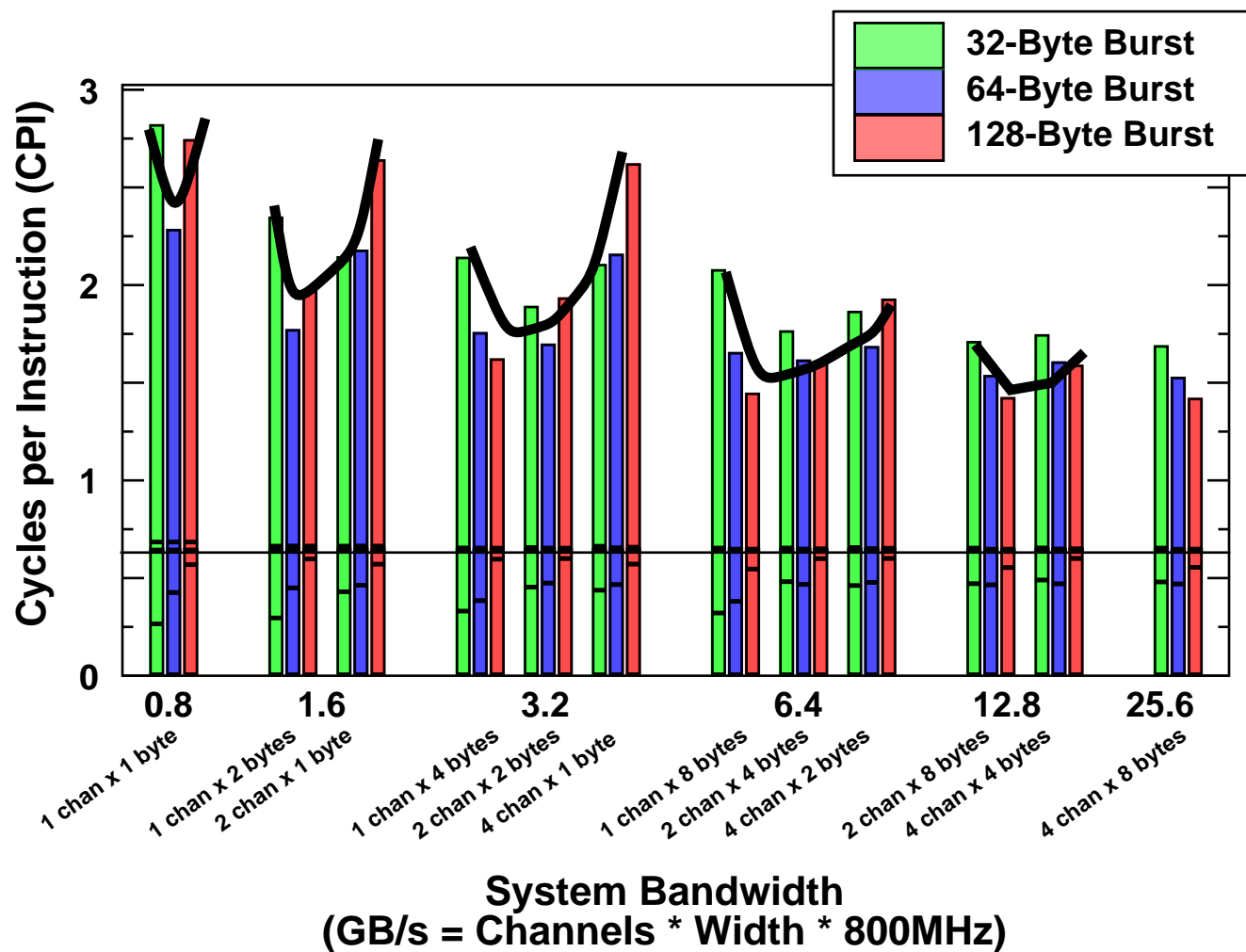
Benchmark = GCC (SPEC 2000), 2 banks/channel

HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Bandwidth vs. Burst Width



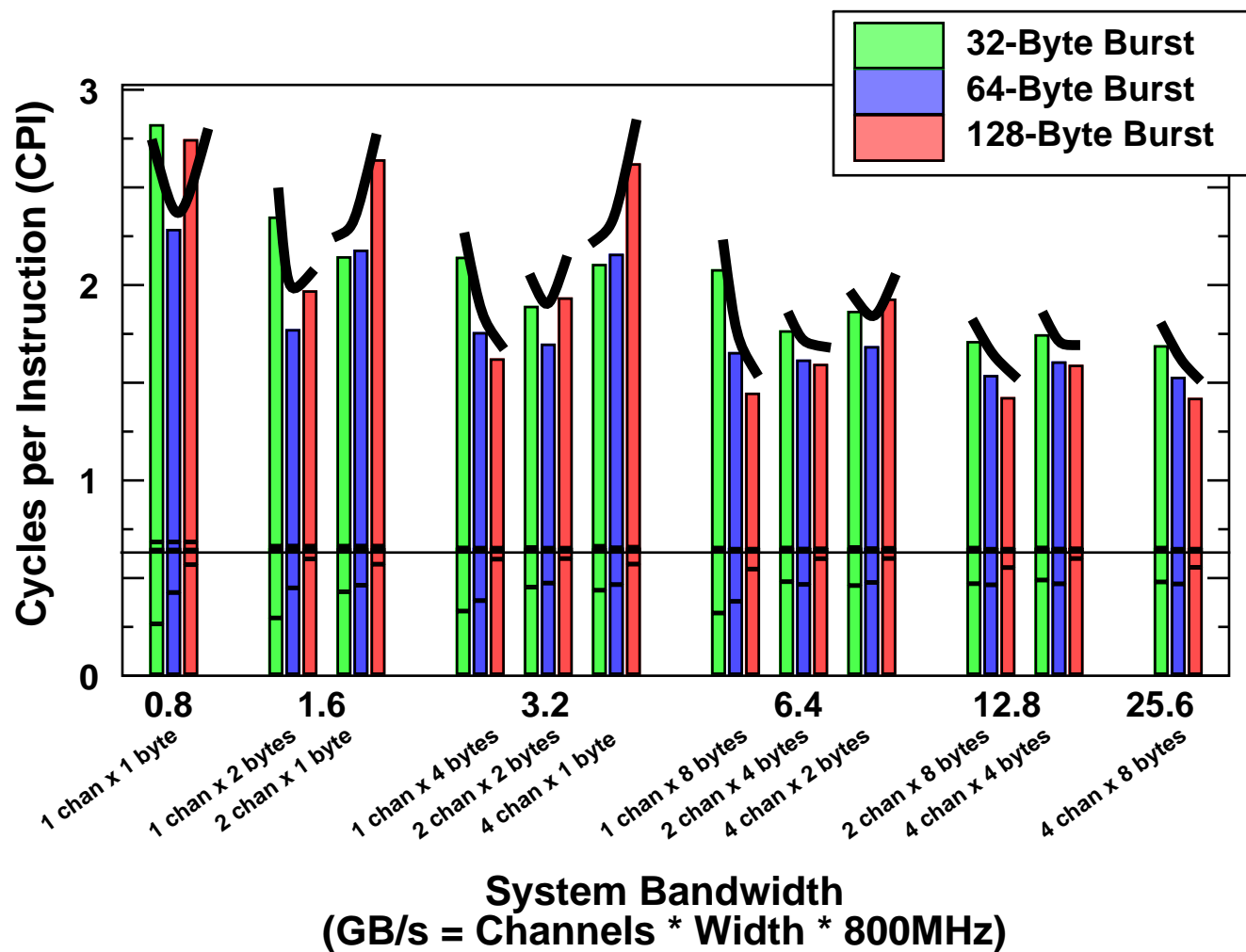
Benchmark = GCC (SPEC 2000), 2 banks/channel

HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Bandwidth vs. Burst Width

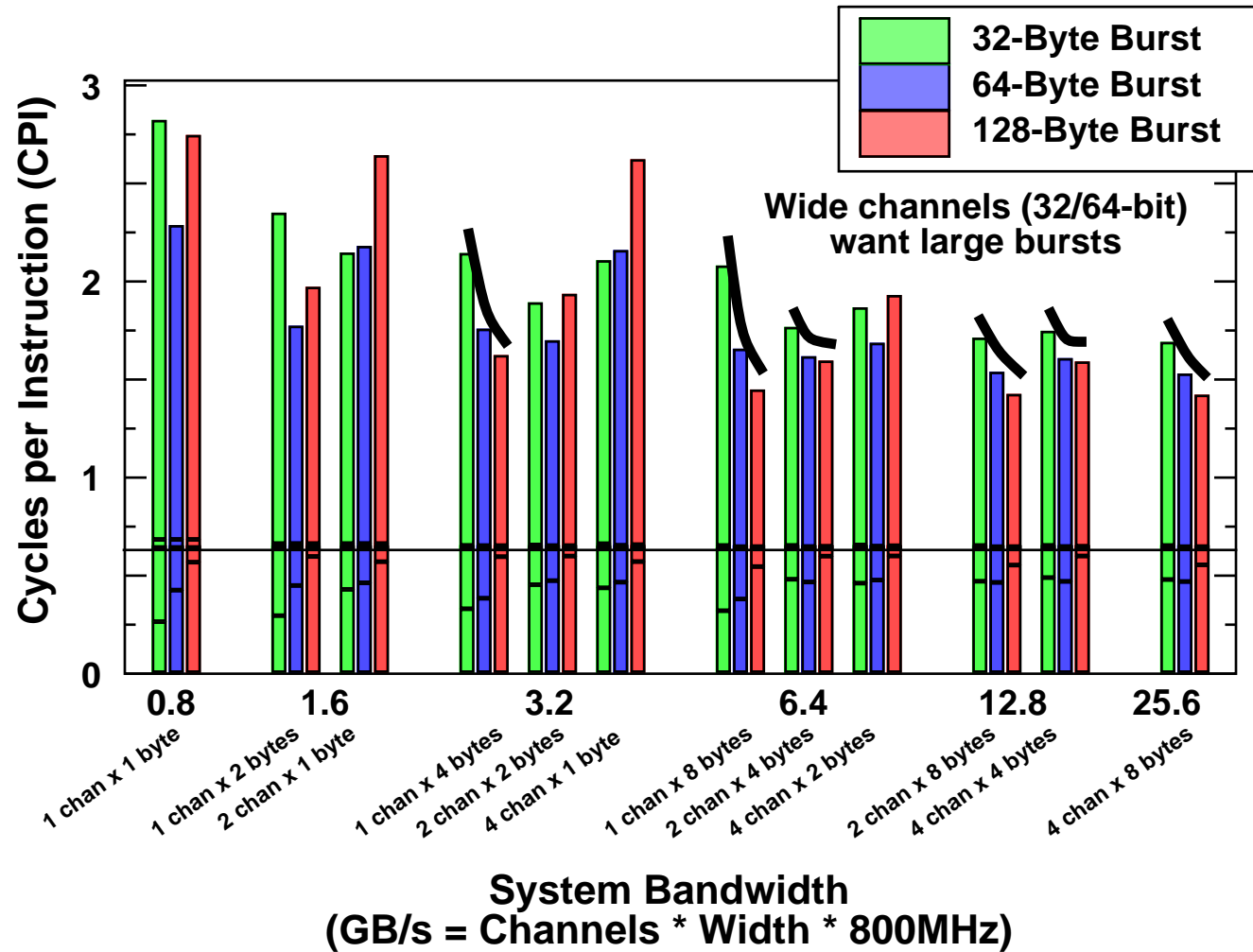


Benchmark = GCC (SPEC 2000), 2 banks/channel



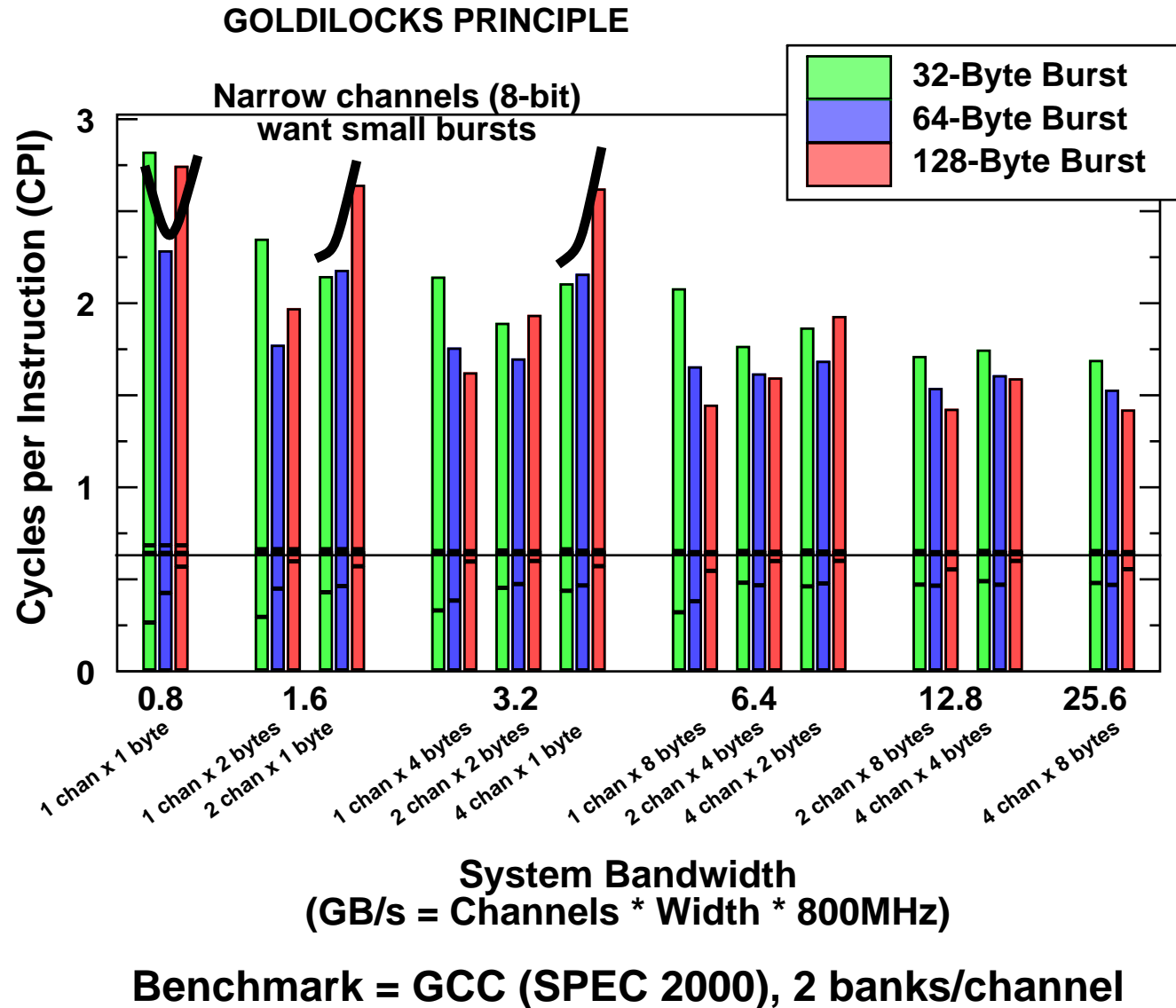
# Bandwidth vs. Burst Width

## GOLDBLOCKS PRINCIPLE



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

# Bandwidth vs. Burst Width



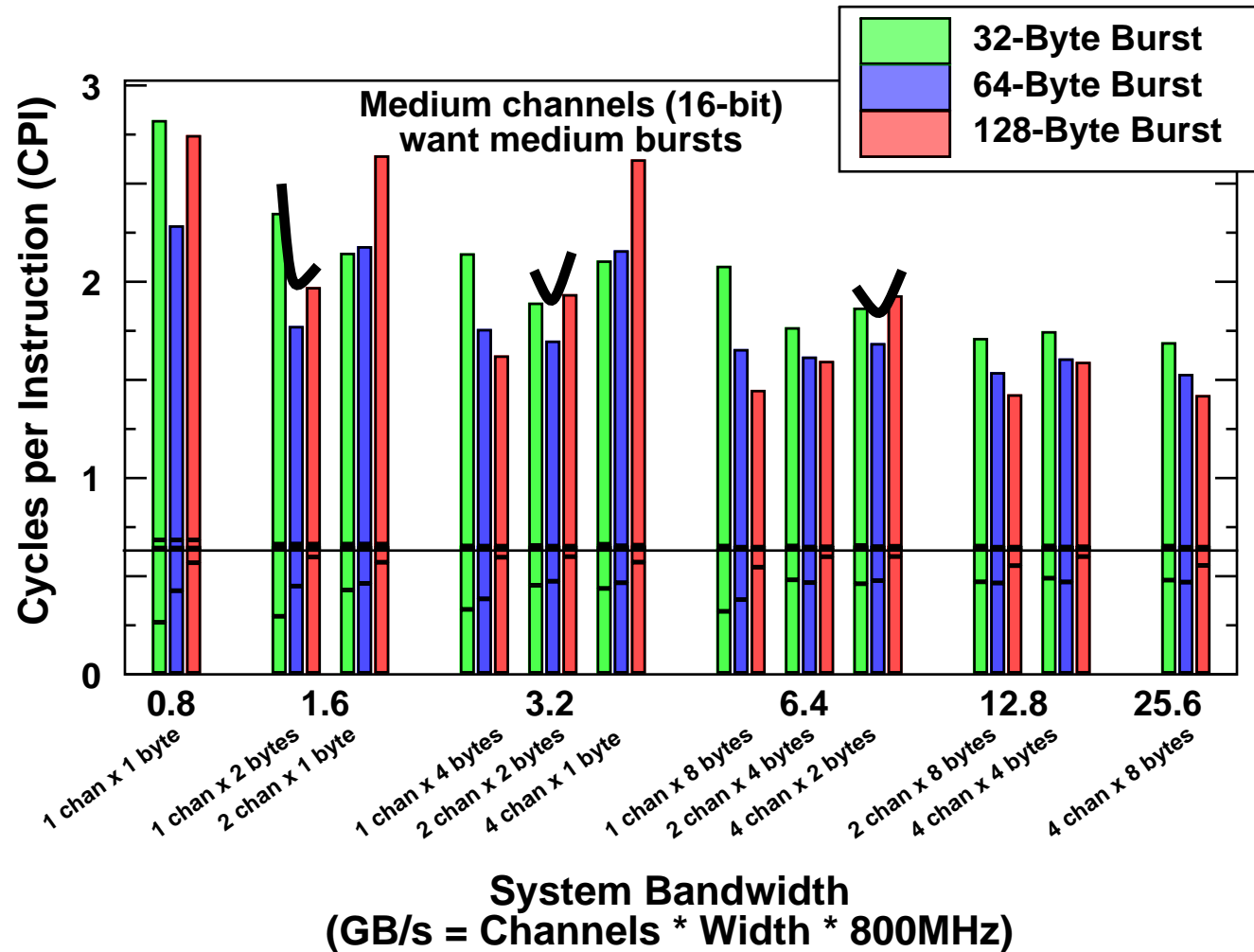
HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Bandwidth vs. Burst Width

## GOLDBLOCKS PRINCIPLE



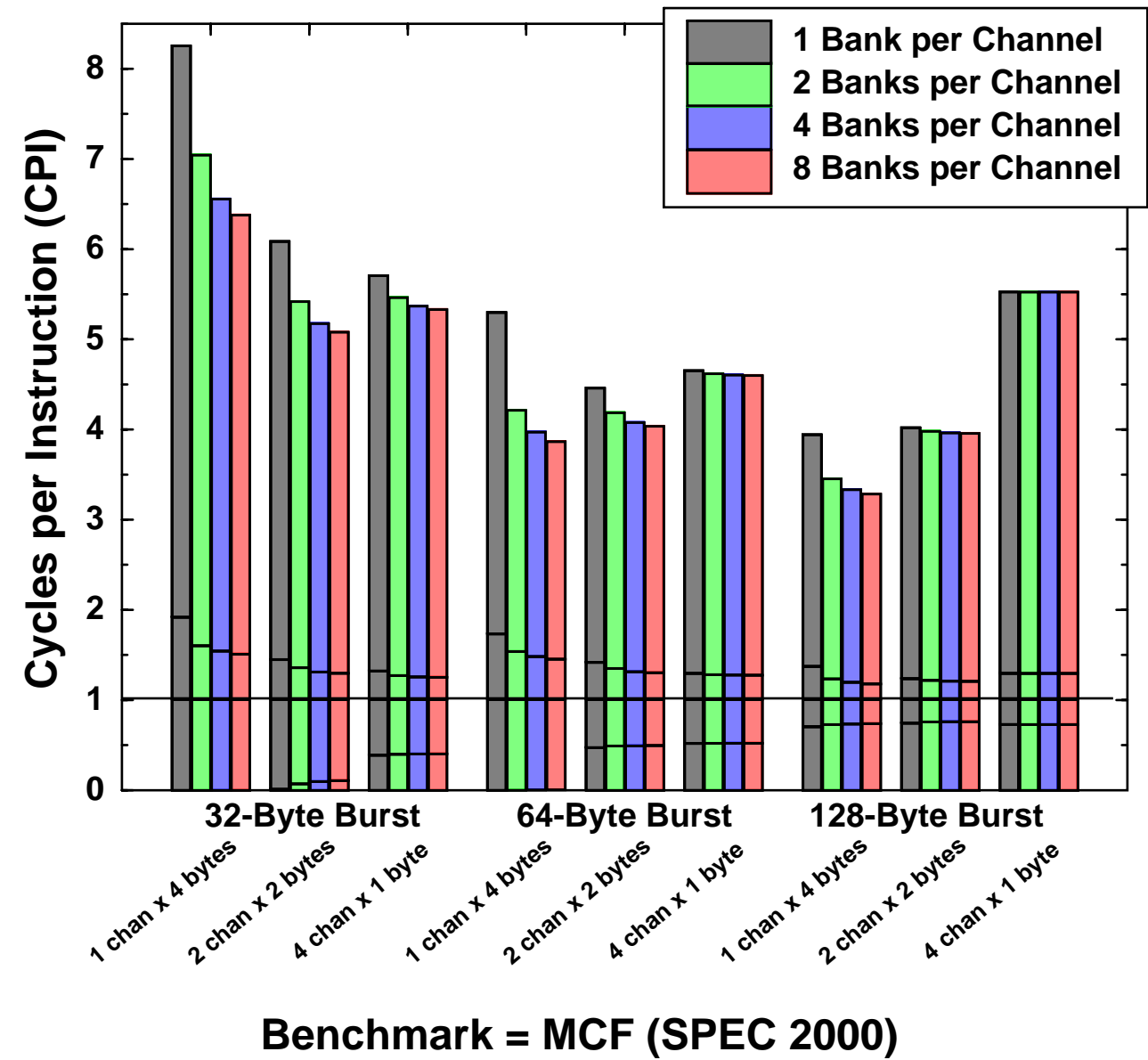
Benchmark = GCC (SPEC 2000), 2 banks/channel

HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

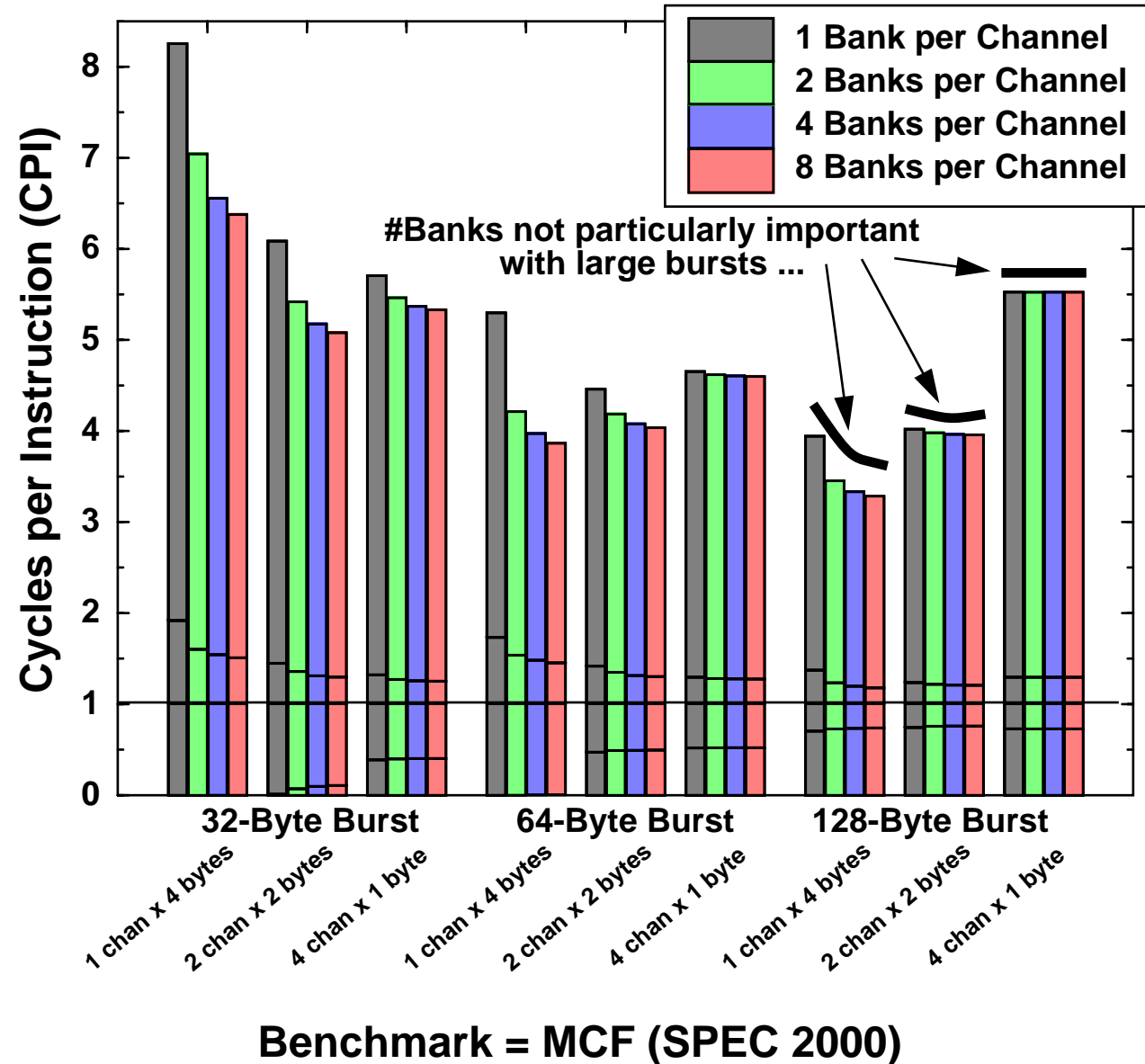
Bruce Jacob

University of  
Maryland

# Focus on 3.2 GB/s — MCF



# Focus on 3.2 GB/s — MCF

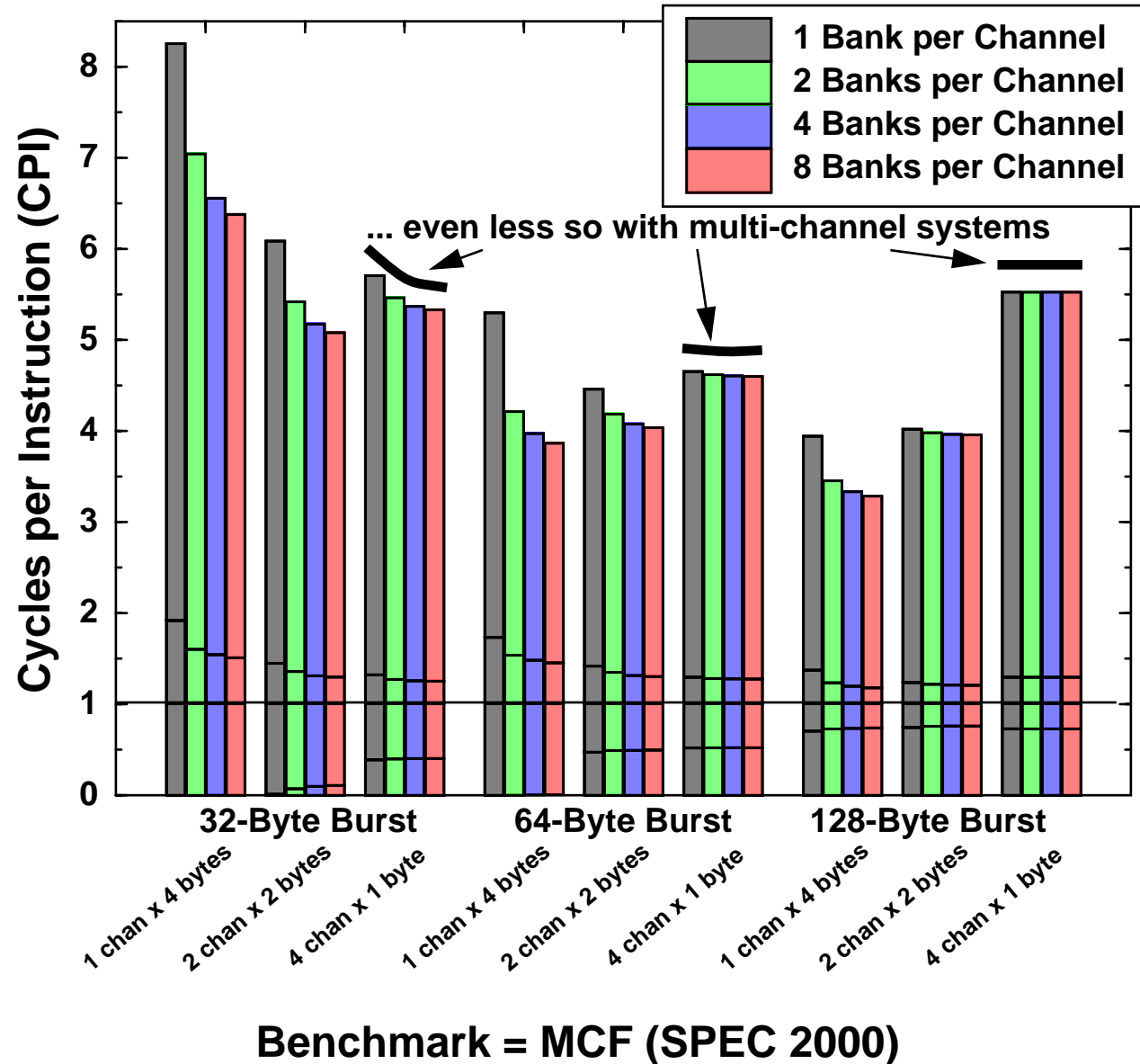


HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Focus on 3.2 GB/s — MCF

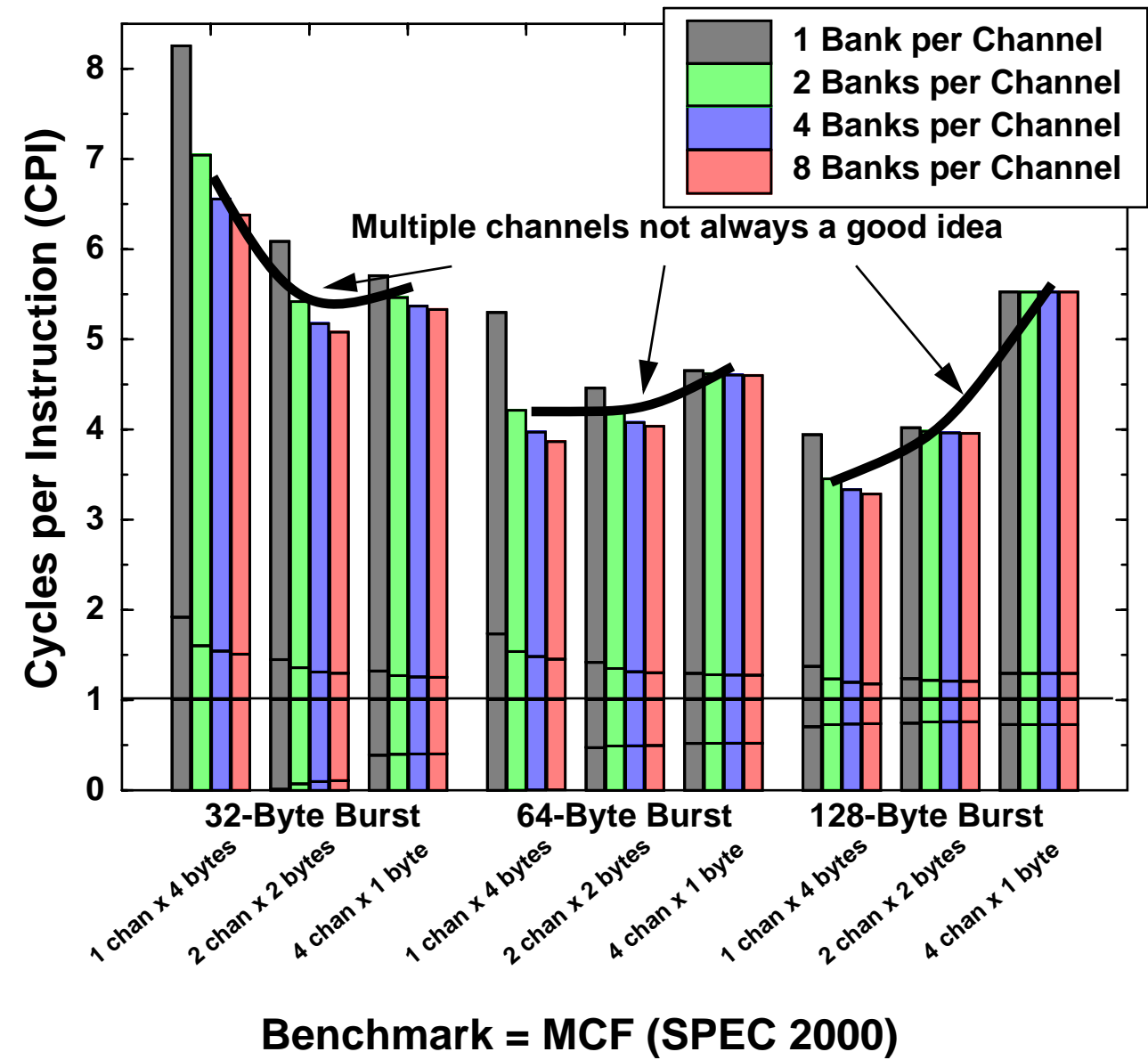


HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Focus on 3.2 GB/s — MCF

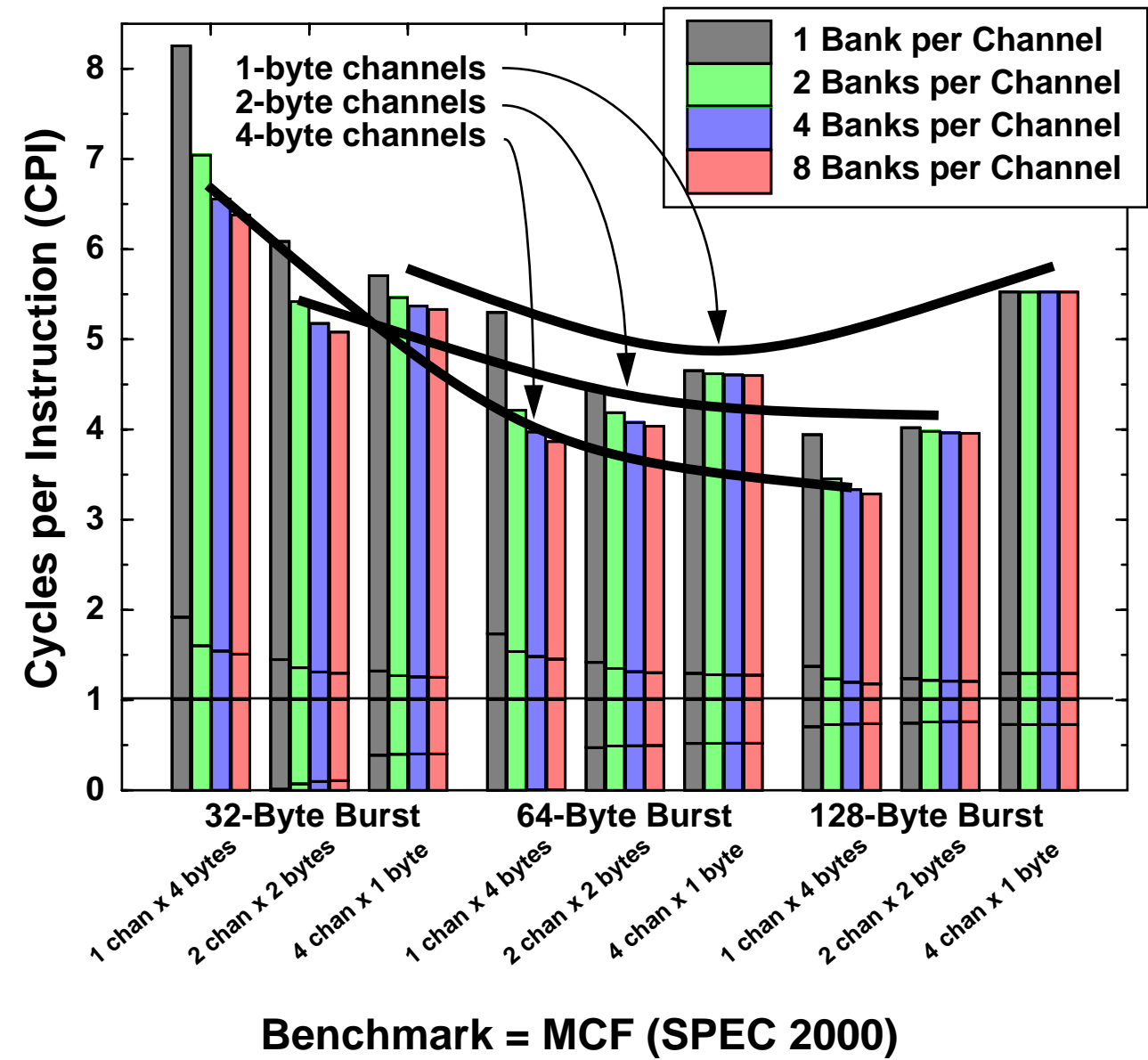


HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Focus on 3.2 GB/s — MCF



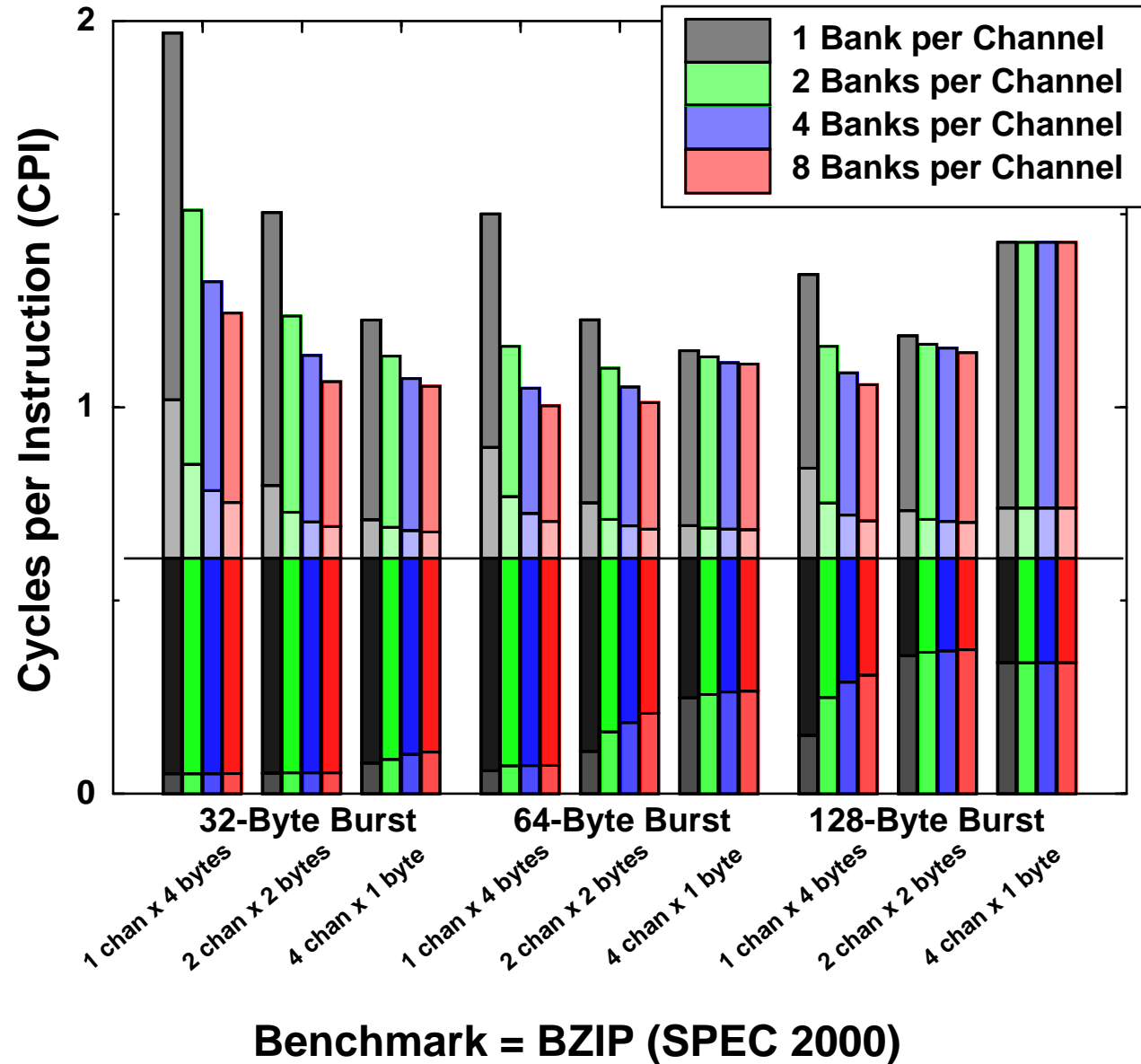


HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Focus on 3.2 GB/s — BZIP

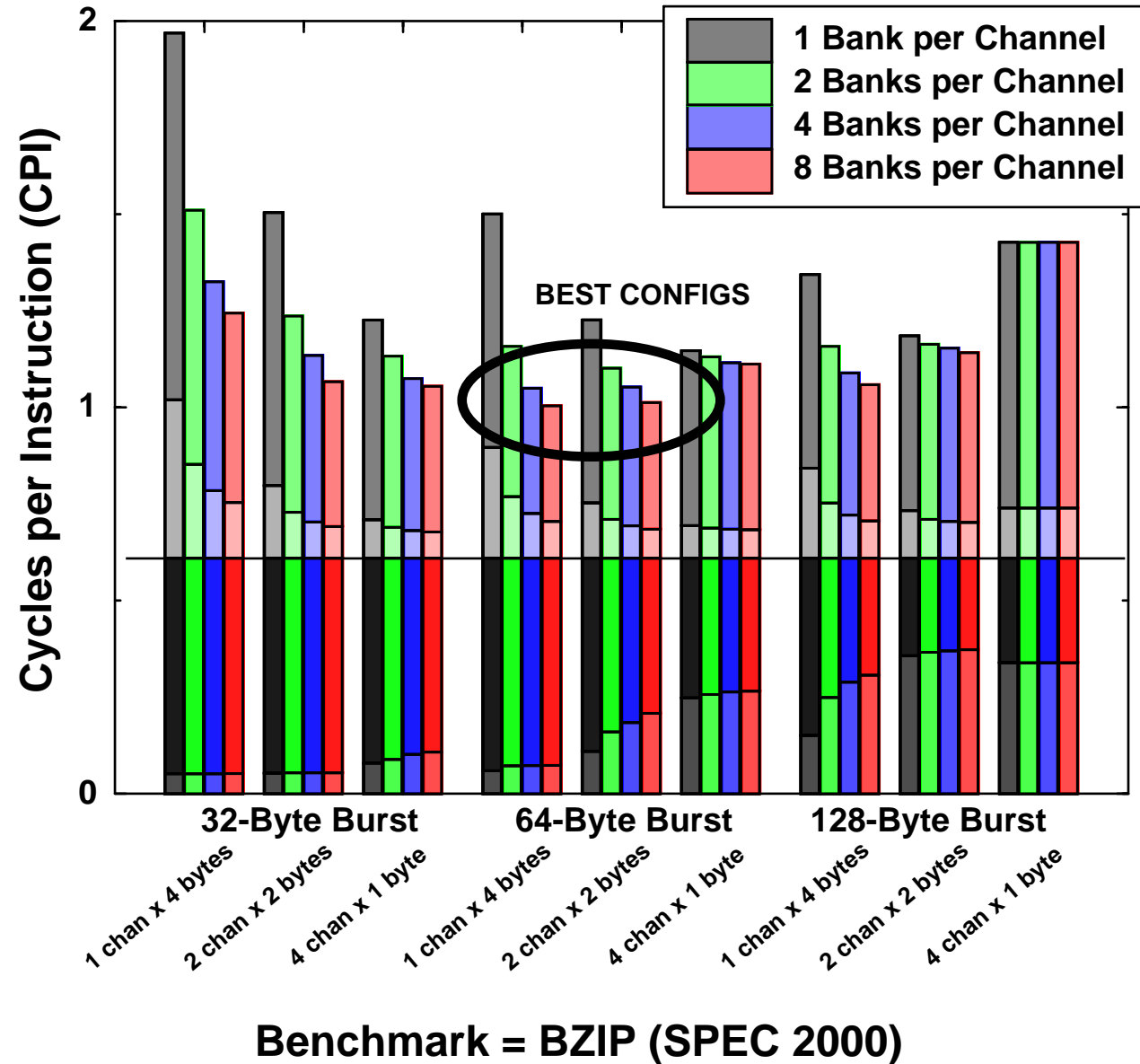


HOW NOT TO  
CONFIGURE  
YOUR DRAM  
SYSTEM

Bruce Jacob

University of  
Maryland

# Focus on 3.2 GB/s — BZIP



# Conclusions

**CAREFUL TUNING YIELDS 30–40% GAIN**

**MORE CONCURRENCY == BETTER**

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

**WIDER == BETTER (Thank you, Pontiac)**

- **Gang Multiple RAMBUS Channels**

**BURSTS AMORTIZE COST OF PRECHARGE**

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

# CONTACT INFO:

**Prof. Bruce Jacob**

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

**`http://www.ece.umd.edu/~blj/`**

**`blj@eng.umd.edu`**

