Contemporary DRAM Architectures and Beyond

Bruce Jacob

Electrical & Computer Engineering
University of Maryland, College Park
`http://www.ece.umd.edu/~blj/`

OUTLINE:

- **Motivation & Background**
- **Experiments**
- **Results**
- **More Recent Results**

Sources

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

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

Recent experiments by Brian Davis,
Ph.D. student at University of Michigan

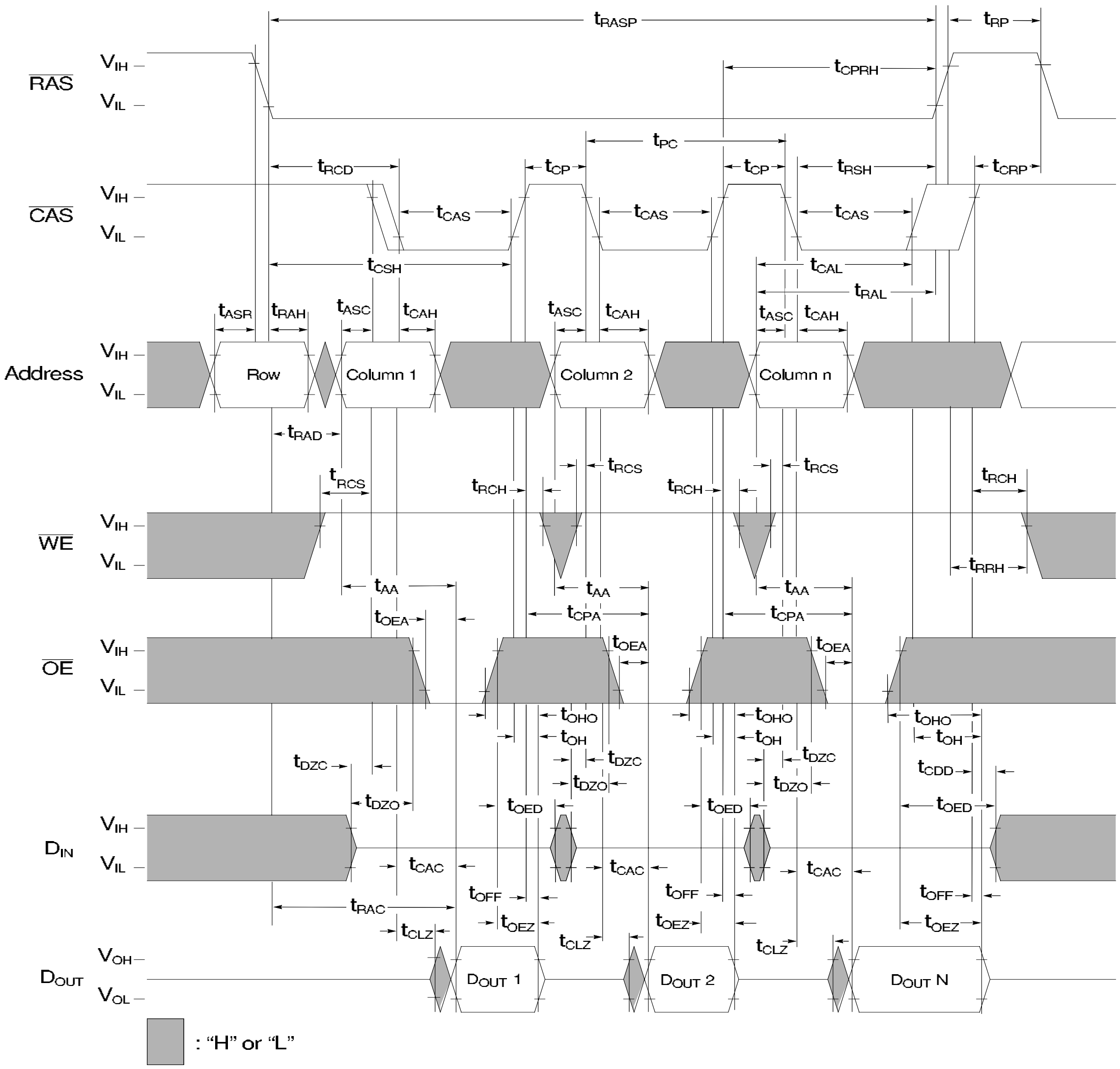
Dilemma: THIS ...

STATUS QUO in
MEMORY-SYSTEM RESEARCH:

```
...

if (memory_instruction(INSTR)) {
    if (L1_cache_miss( data_addr(INSTR) ){
        if (L2_cache_miss( data_addr(INSTR) ){

            cycles += DRAM_LATENCY;

        }
    }
}

...
```

... or THIS

Fast Page Mode Read Cycle



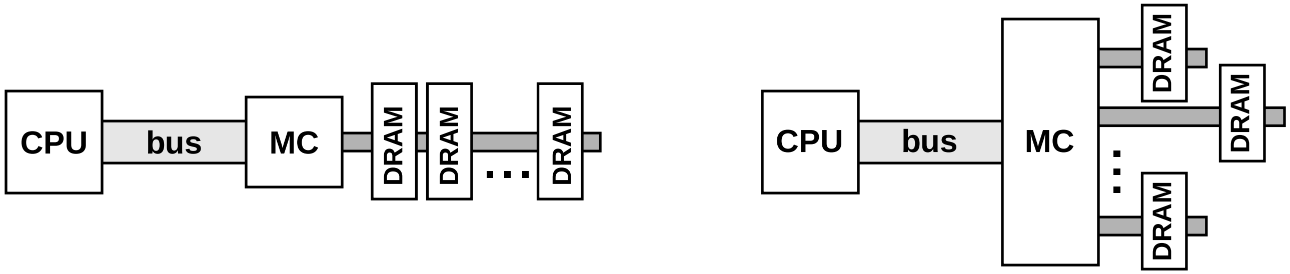
CONTEMPORARY DRAM ARCHITECTURES AND BEYOND

Bruce Jacob

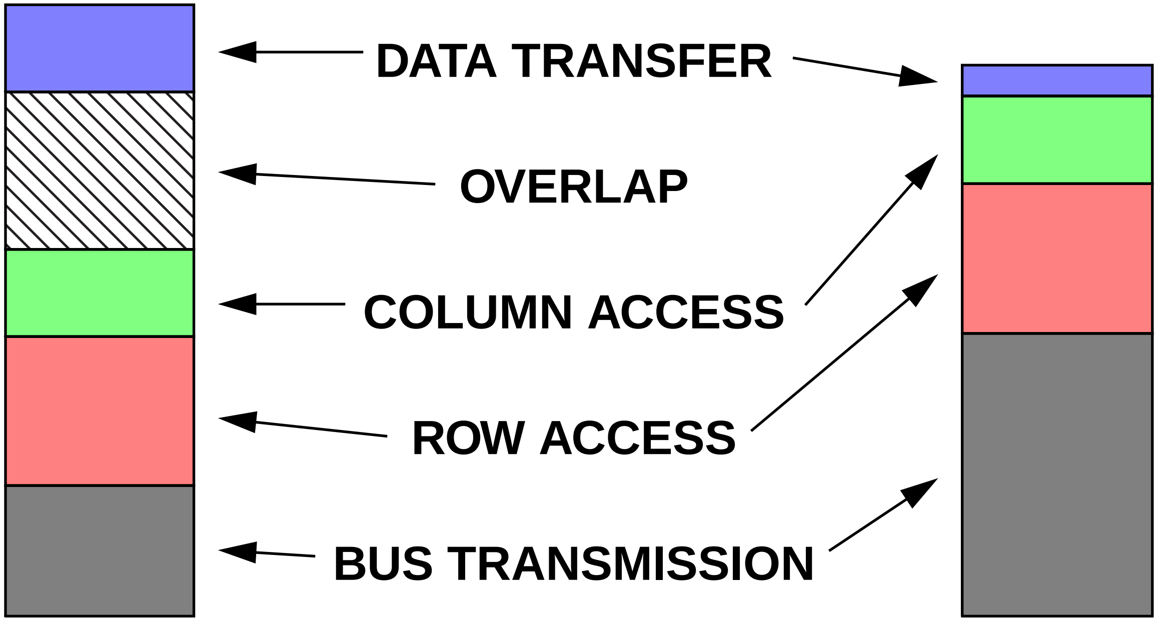
University of Maryland

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)

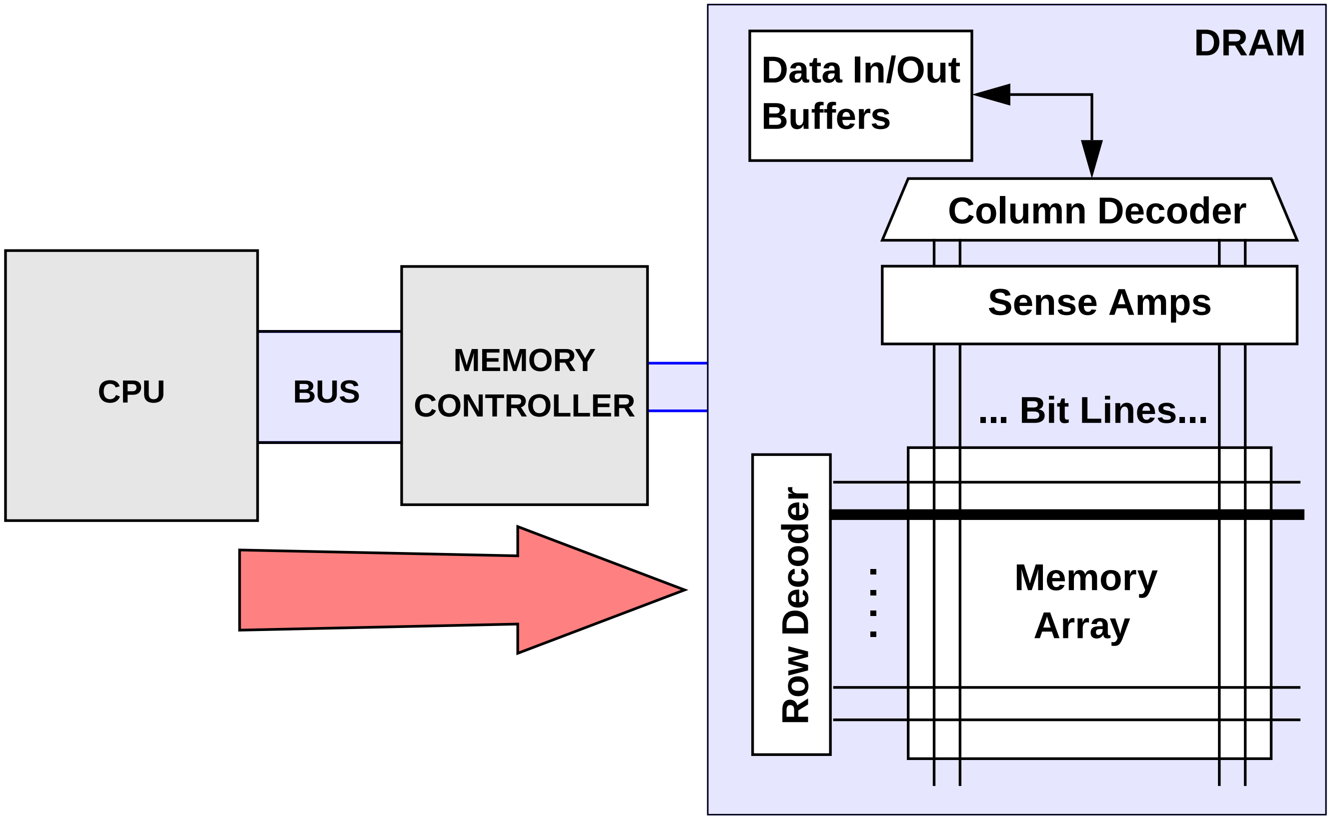
MODEL EXISTING TECHNOLOGY

Bruce Jacob

University of Maryland

DRAM Primer

BUS TRANSMISSION



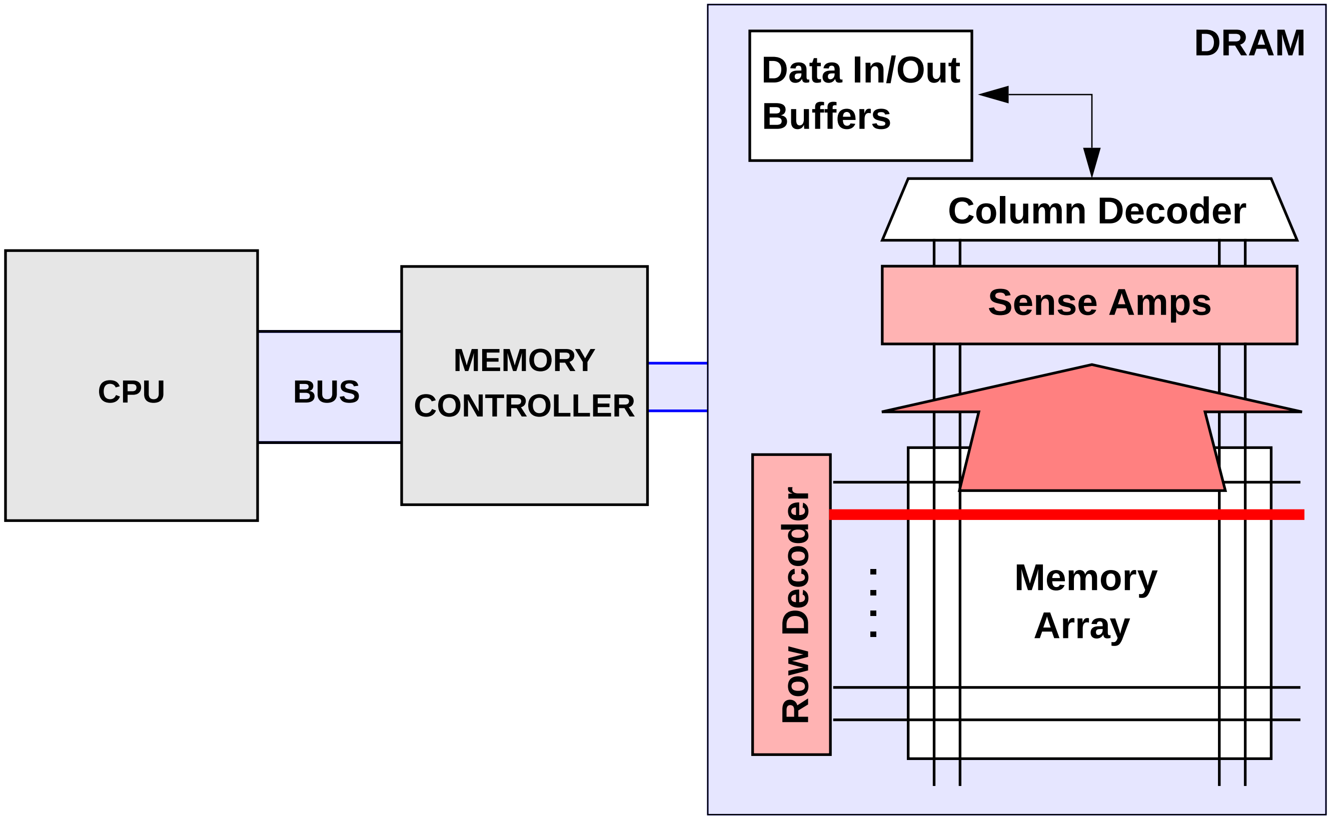
CONTEMPORARY DRAM ARCHITECTURES AND BEYOND

Bruce Jacob

University of Maryland

DRAM Primer

ROW ACCESS

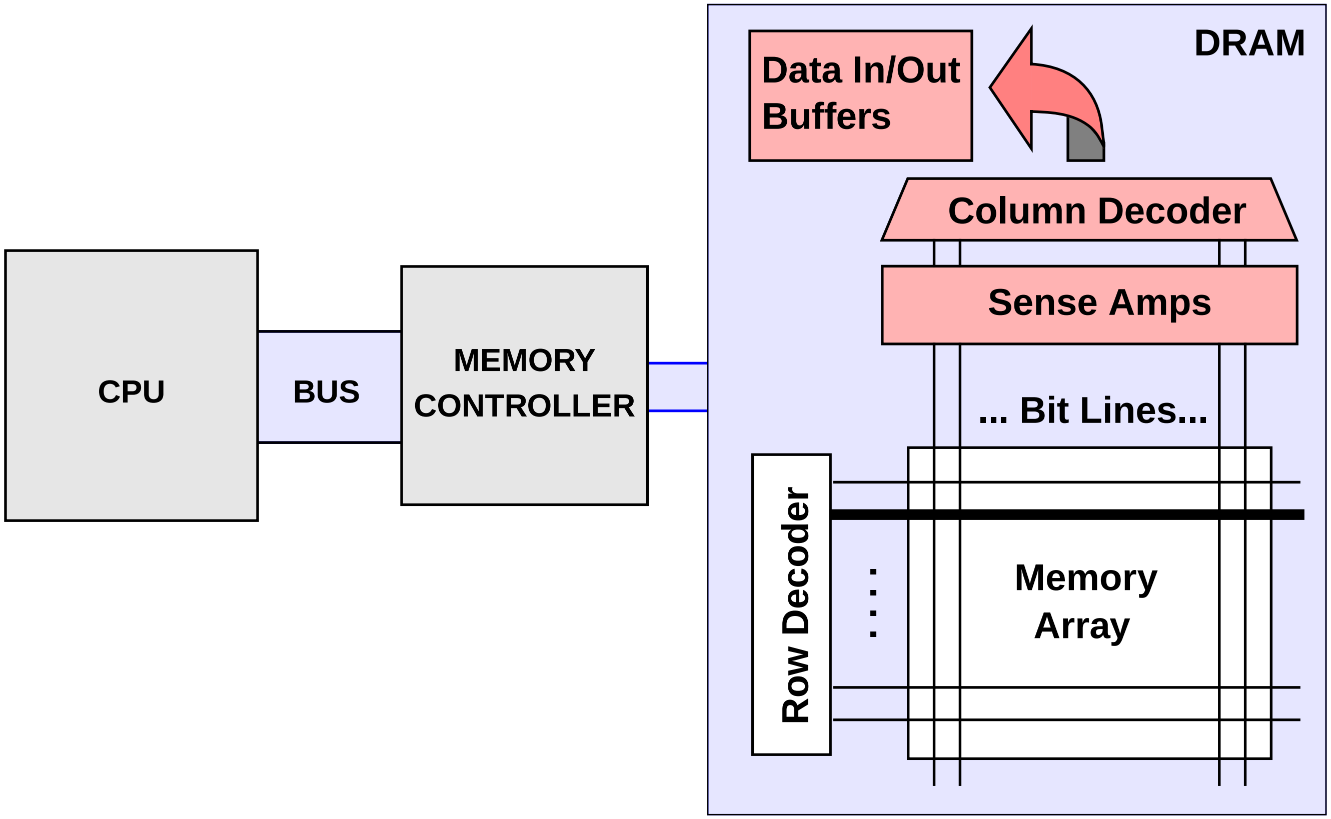


Bruce Jacob

University of Maryland

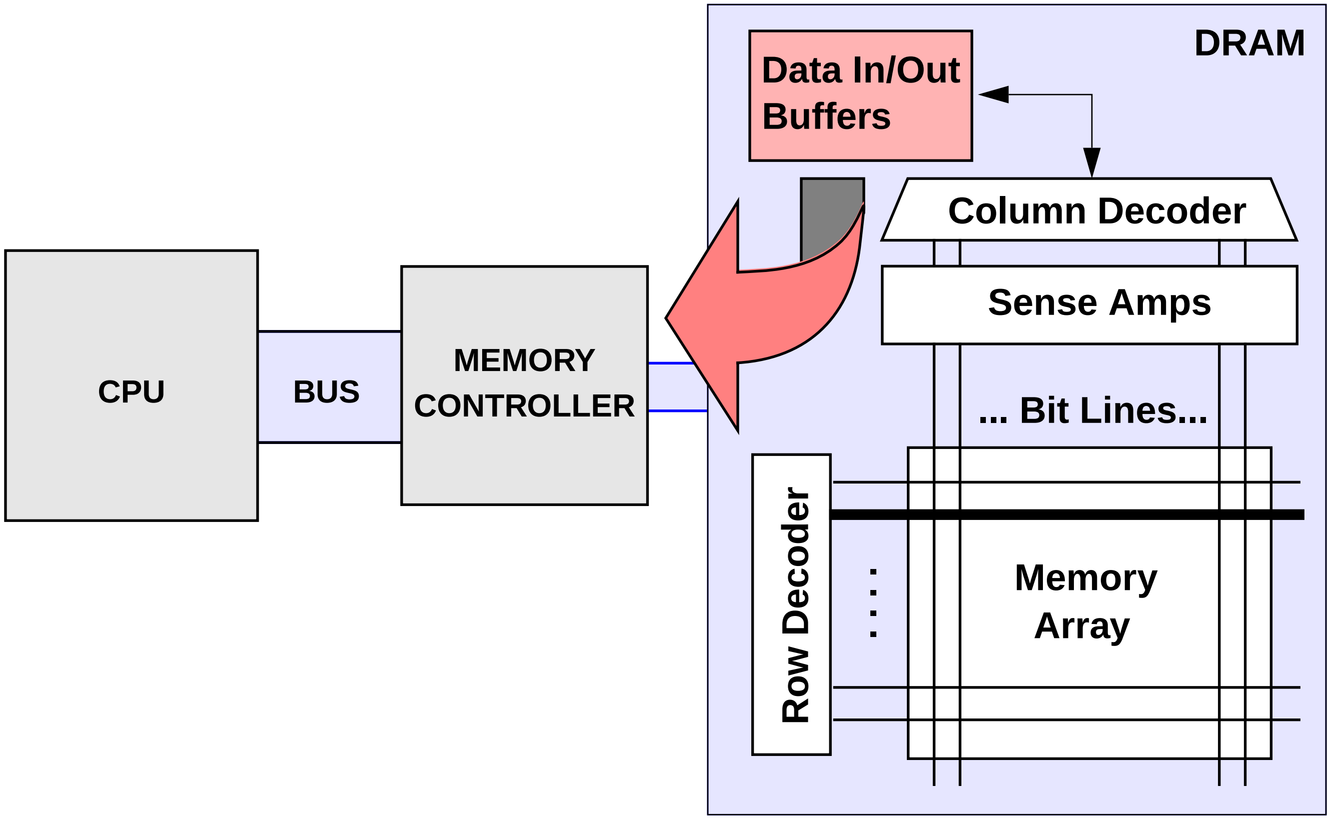
DRAM Primer

COLUMN ACCESS



DRAM Primer

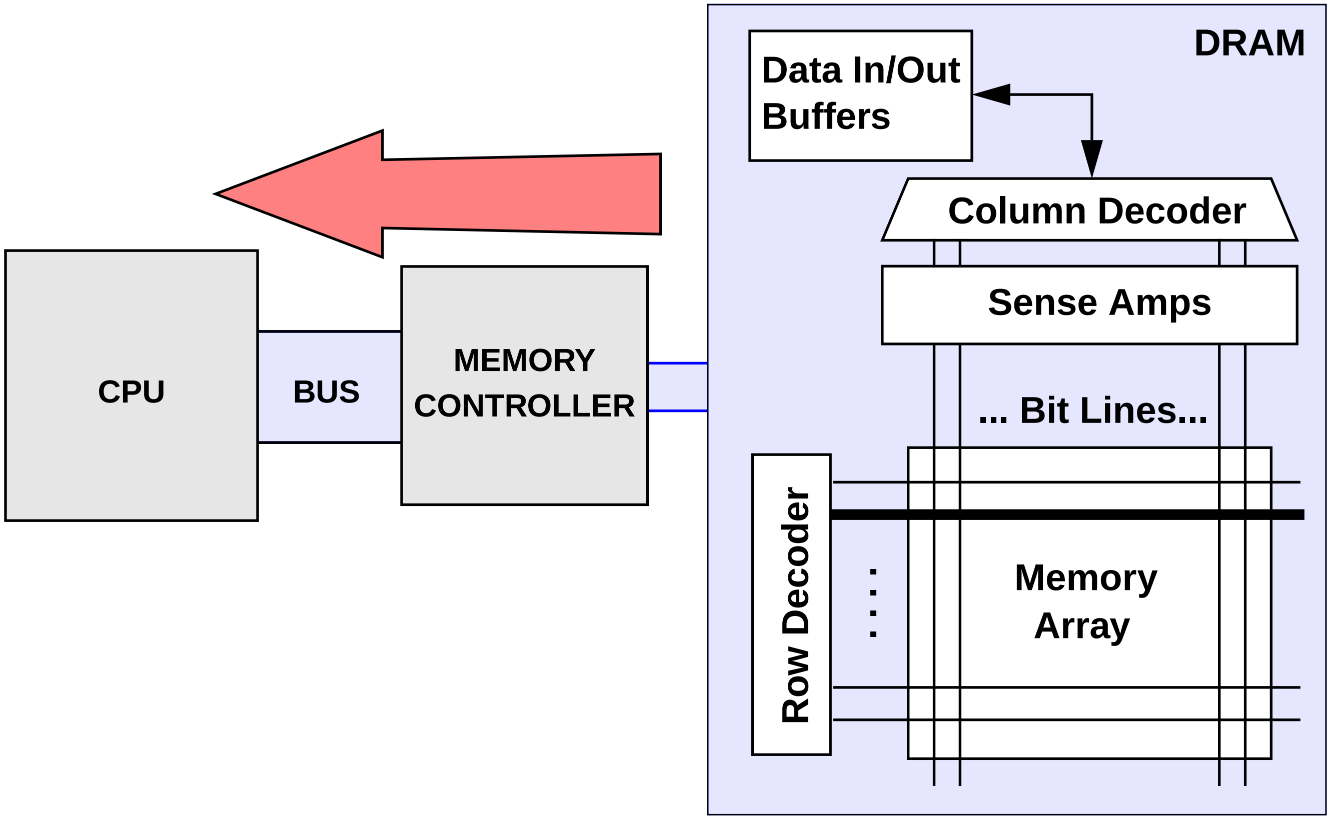
DATA TRANSFER



note: page mode enables overlap with COL

DRAM Primer

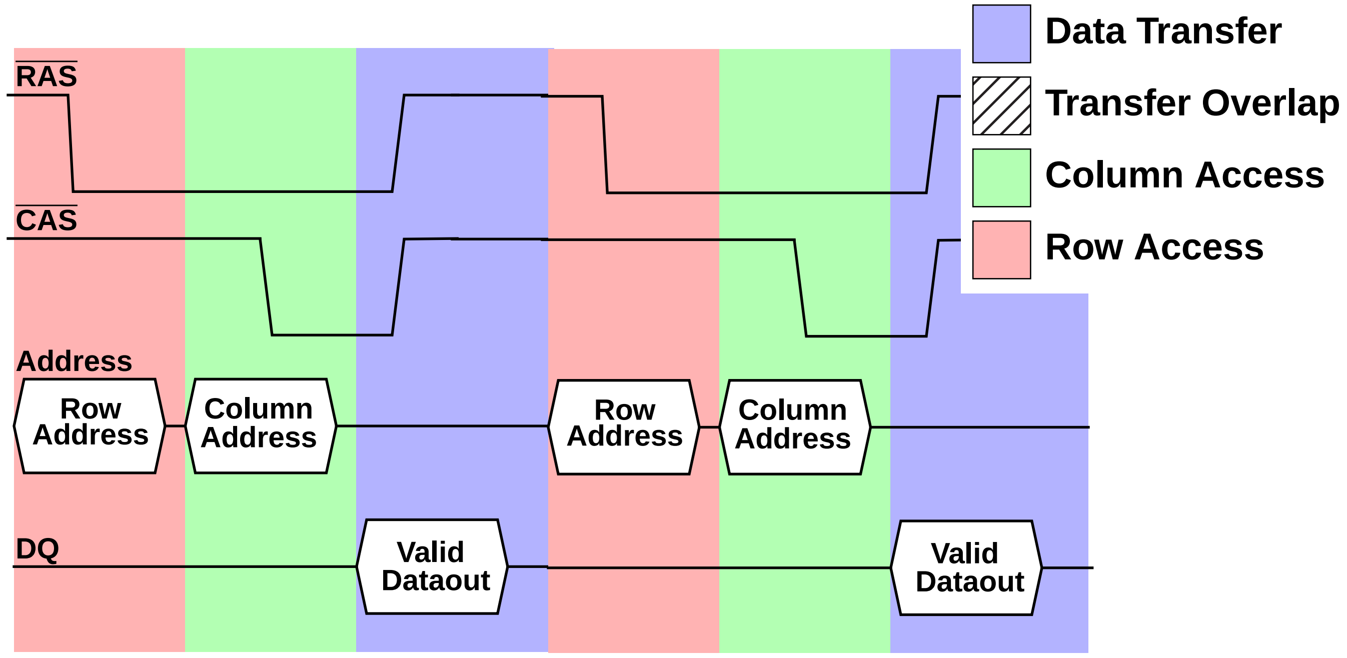
BUS TRANSMISSION



note: overlapped component not shown

DRAM Primer

Read Timing for Conventional DRAM



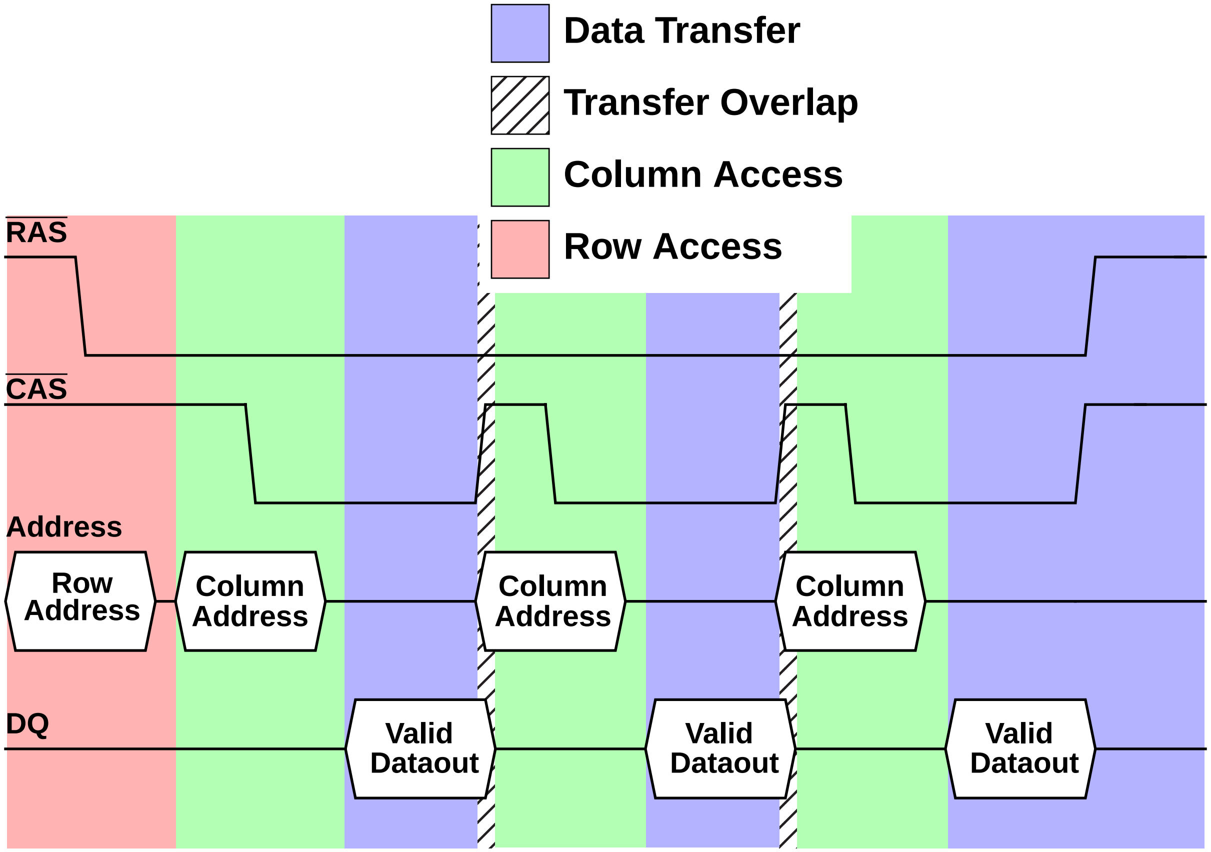
CONTEMPORARY
DRAM
ARCHITECTURES
AND BEYOND

Bruce Jacob

University of
Maryland

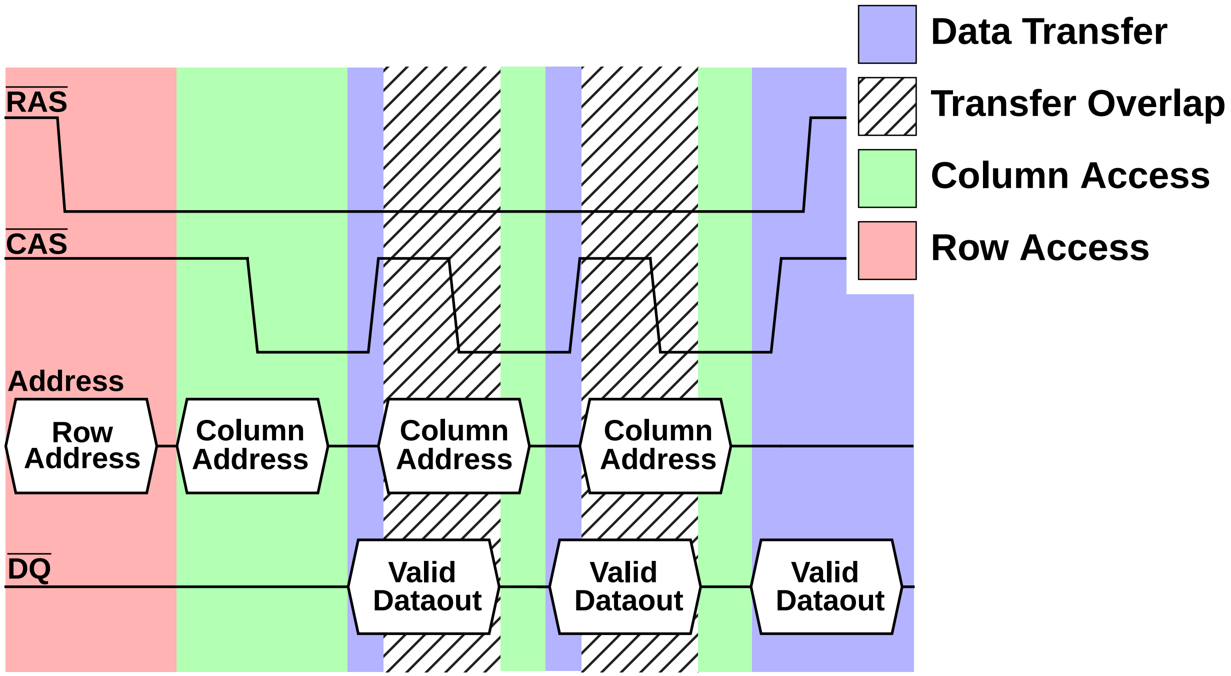
DRAM Primer

Read Timing for Fast Page Mode DRAM



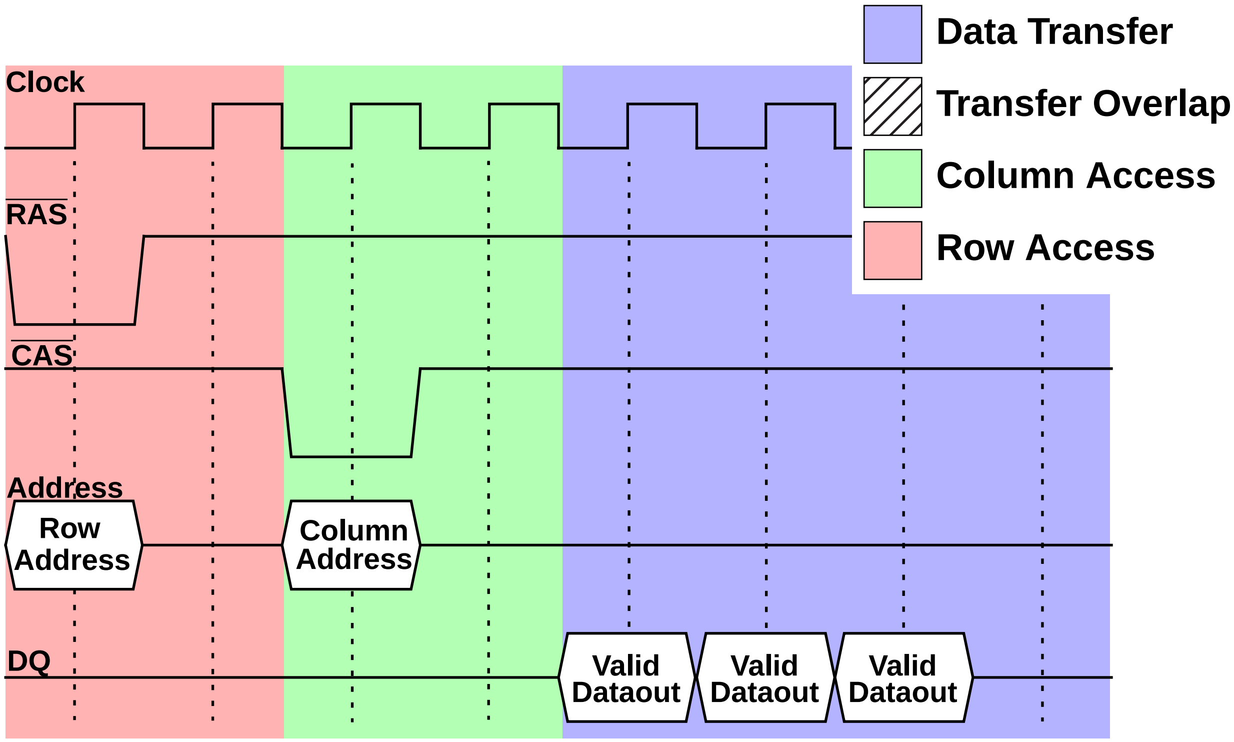
DRAM Primer

Read Timing for Extended Data Out DRAM



DRAM Primer

Read Timing for Synchronous DRAM



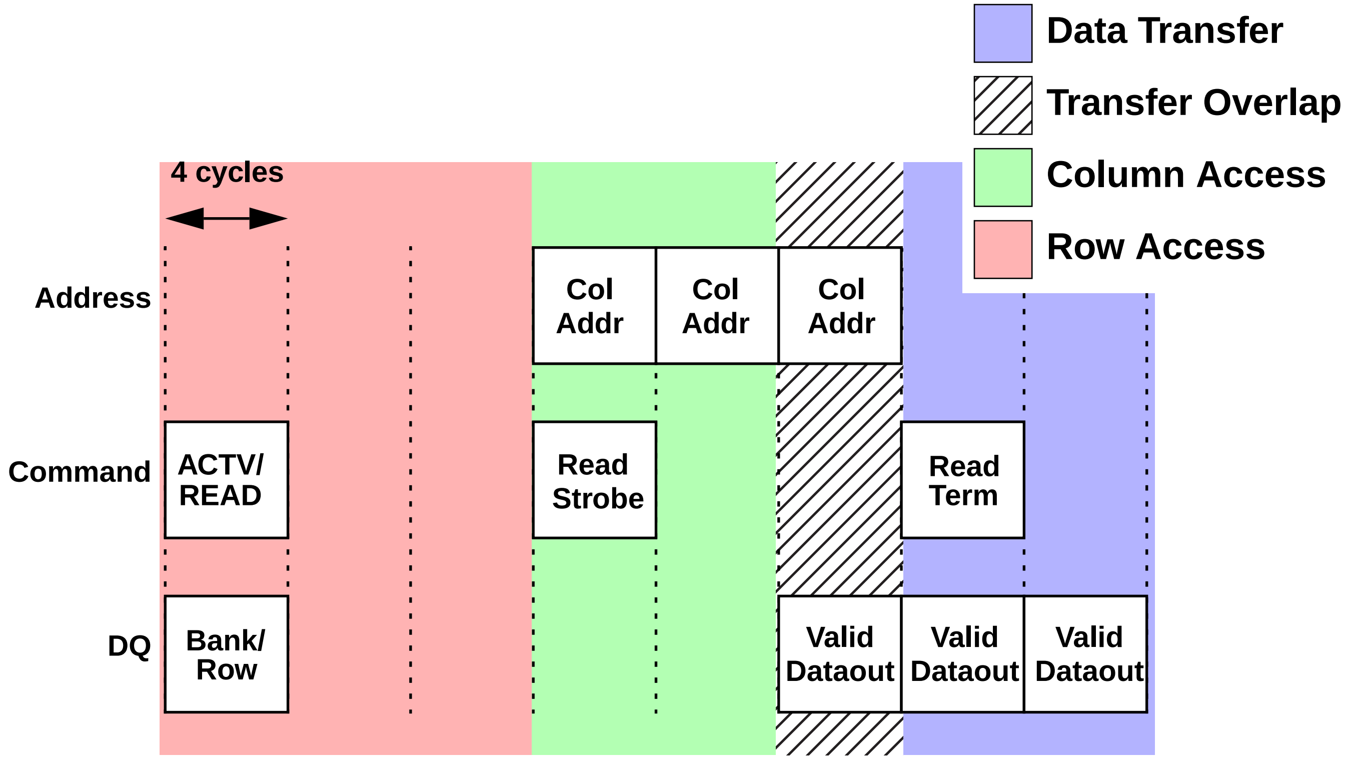
CONTEMPORARY DRAM ARCHITECTURES AND BEYOND

Bruce Jacob

University of Maryland

DRAM Primer

Read Timing for 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**

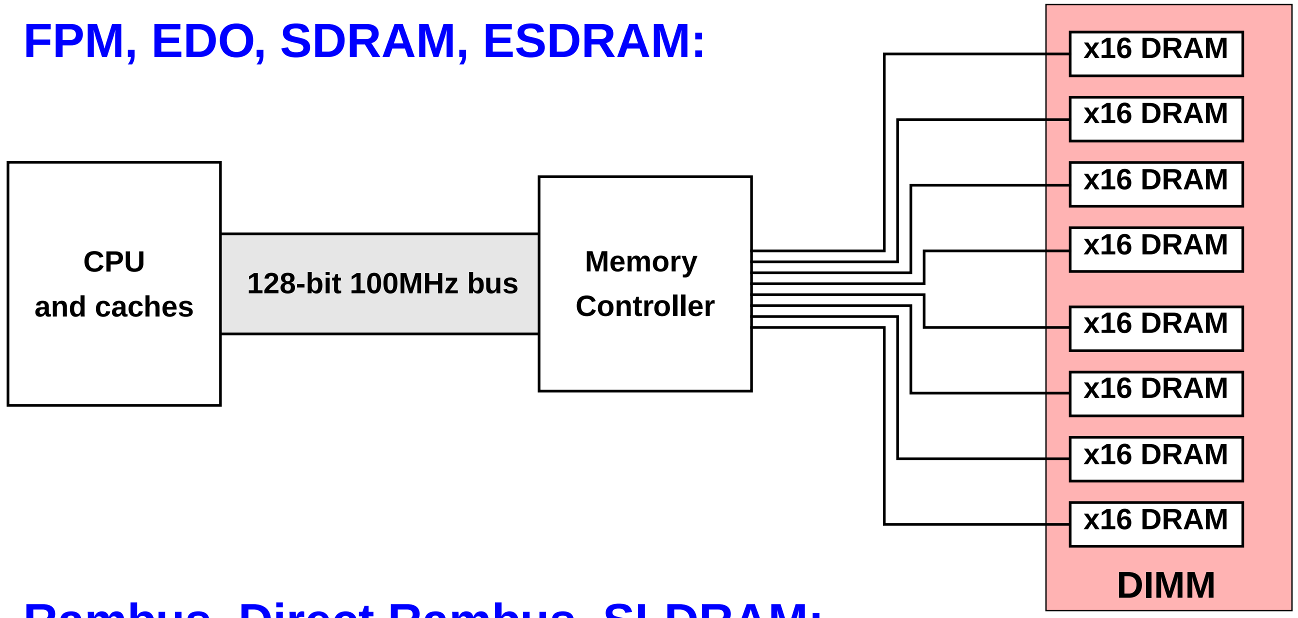
Main Memory: 8 64Mb DRAMs

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

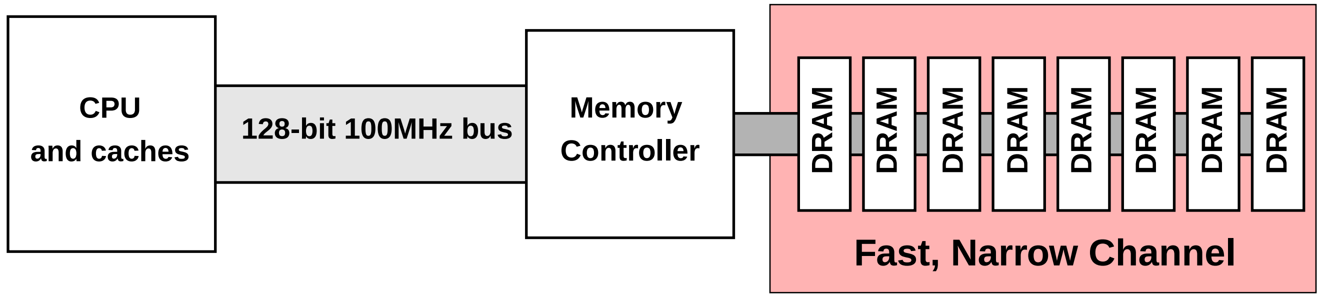
Represents a “typical” workstation

DRAM Configurations

FPM, EDO, SDRAM, ESDRAM:



Rambus, Direct Rambus, SLDRAM:

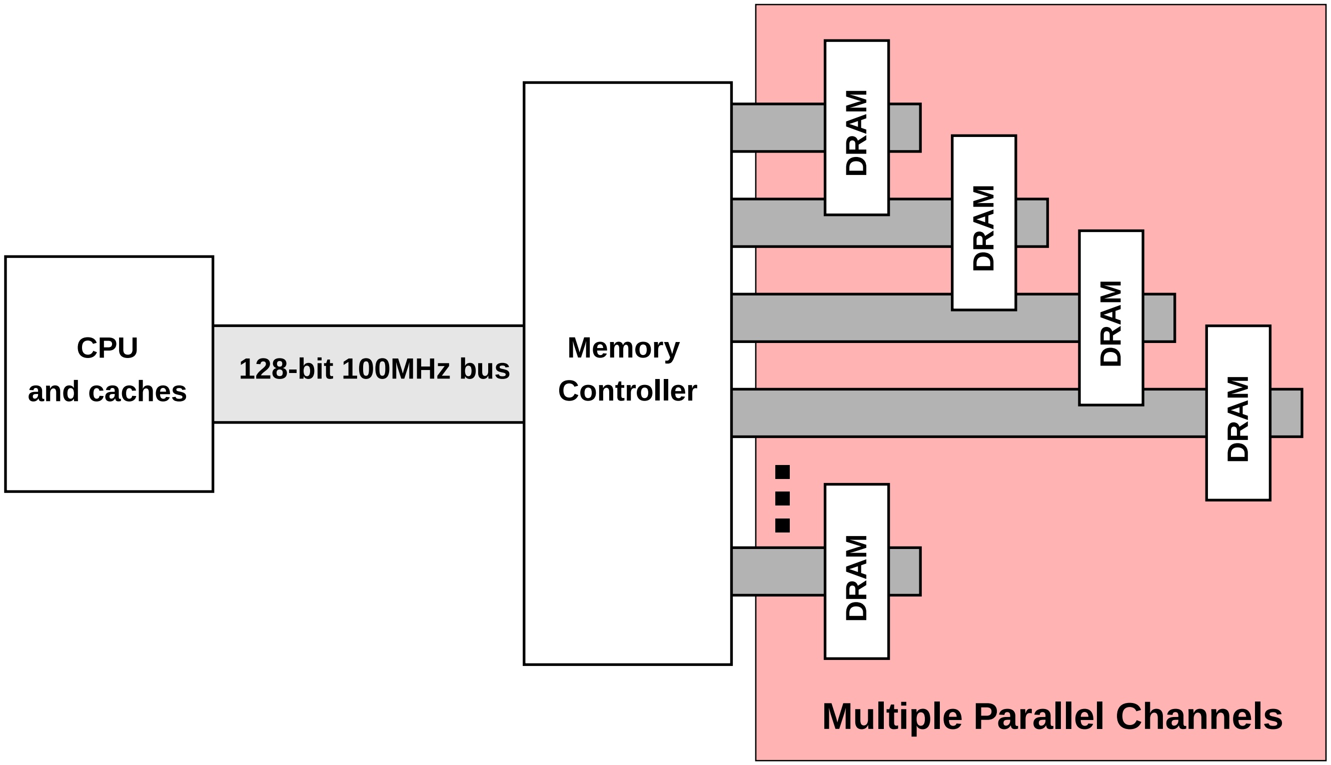


Note: TRANSFER WIDTH of Direct Rambus Channel

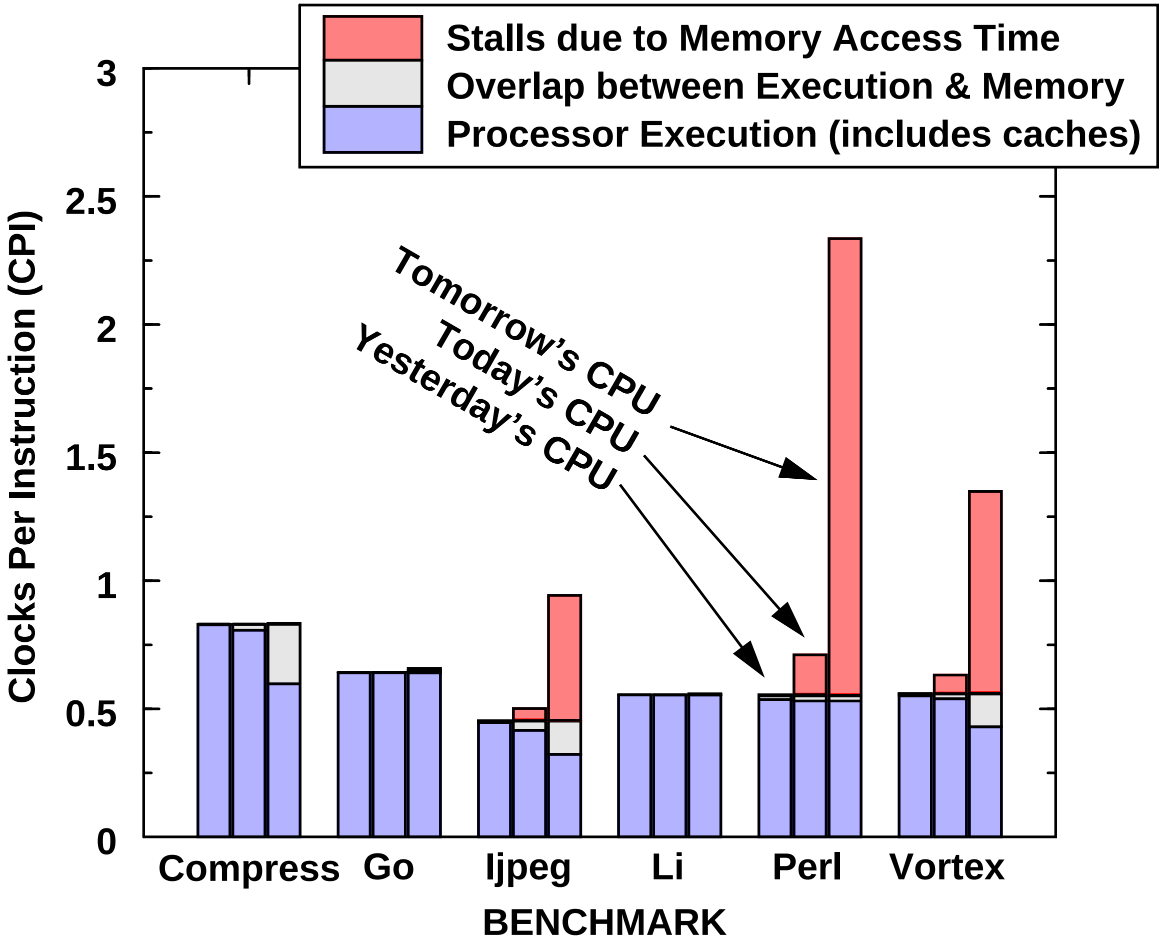
- **equals that of ganged FPM, EDO, etc.**
- **is 2x that of Rambus & SLDRAM**

DRAM Configurations

Strawman: Rambus, etc.



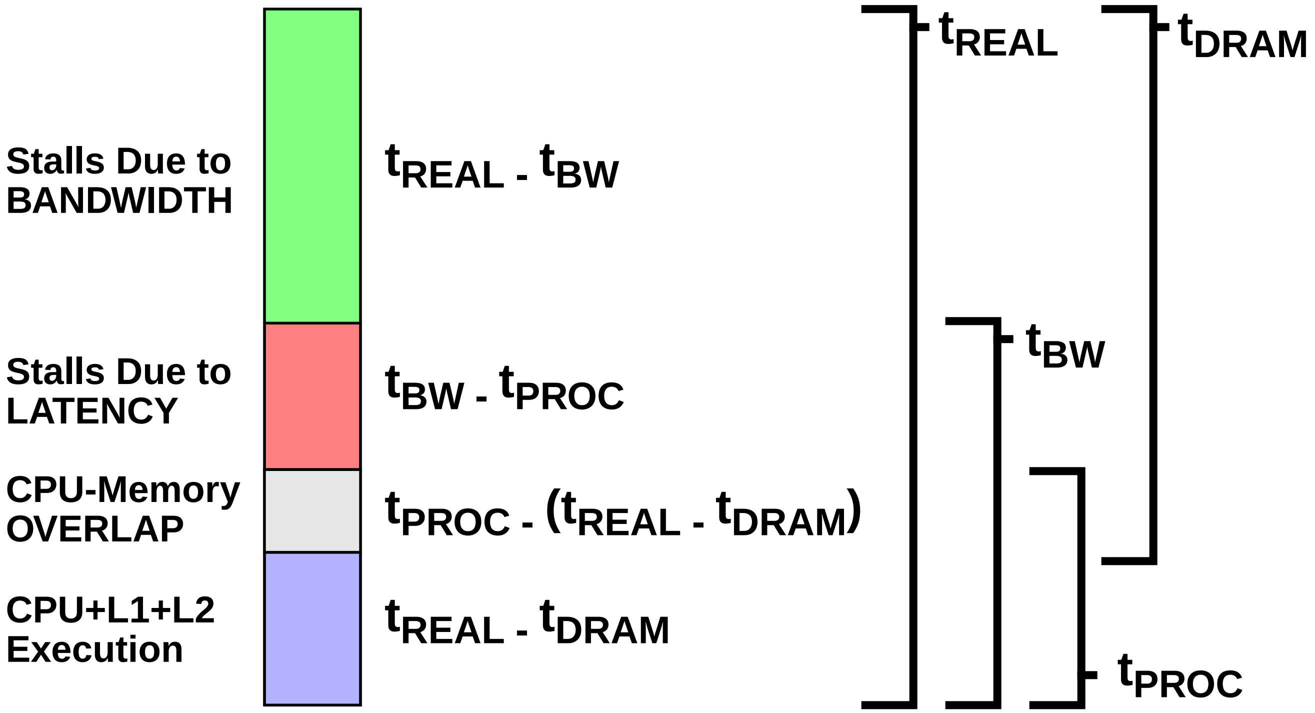
Overhead: Memory vs. CPU



Variable: speed of processor & caches

Definitions (var. on Burger, et al)

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



Stalls Due to BANDWIDTH — $t_{REAL} - t_{BW}$

Stalls Due to LATENCY — $t_{BW} - t_{PROC}$

CPU-Memory OVERLAP — $t_{PROC} - (t_{REAL} - t_{DRAM})$

CPU+L1+L2 Execution — $t_{REAL} - t_{DRAM}$

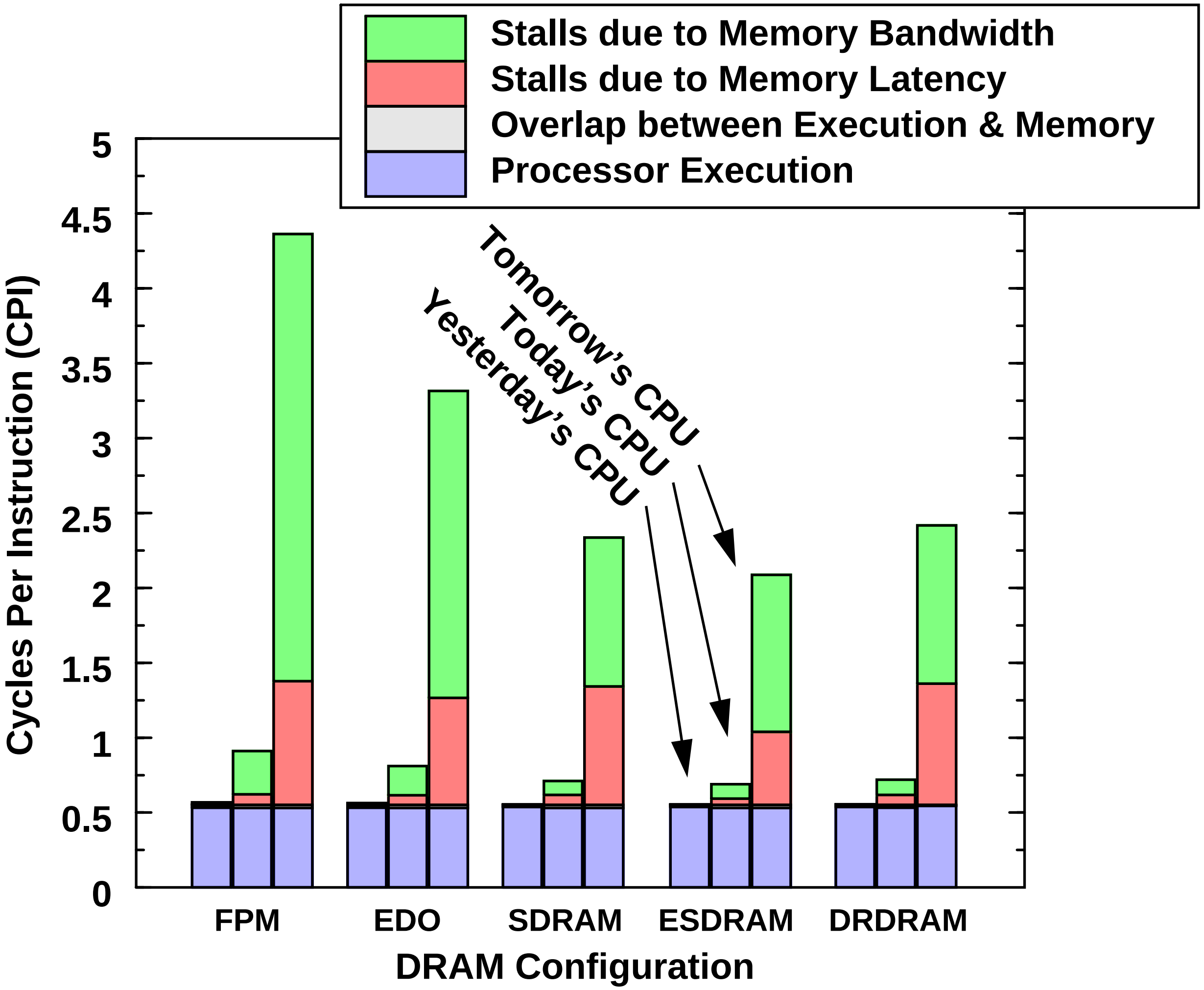
t_{REAL}

t_{DRAM}

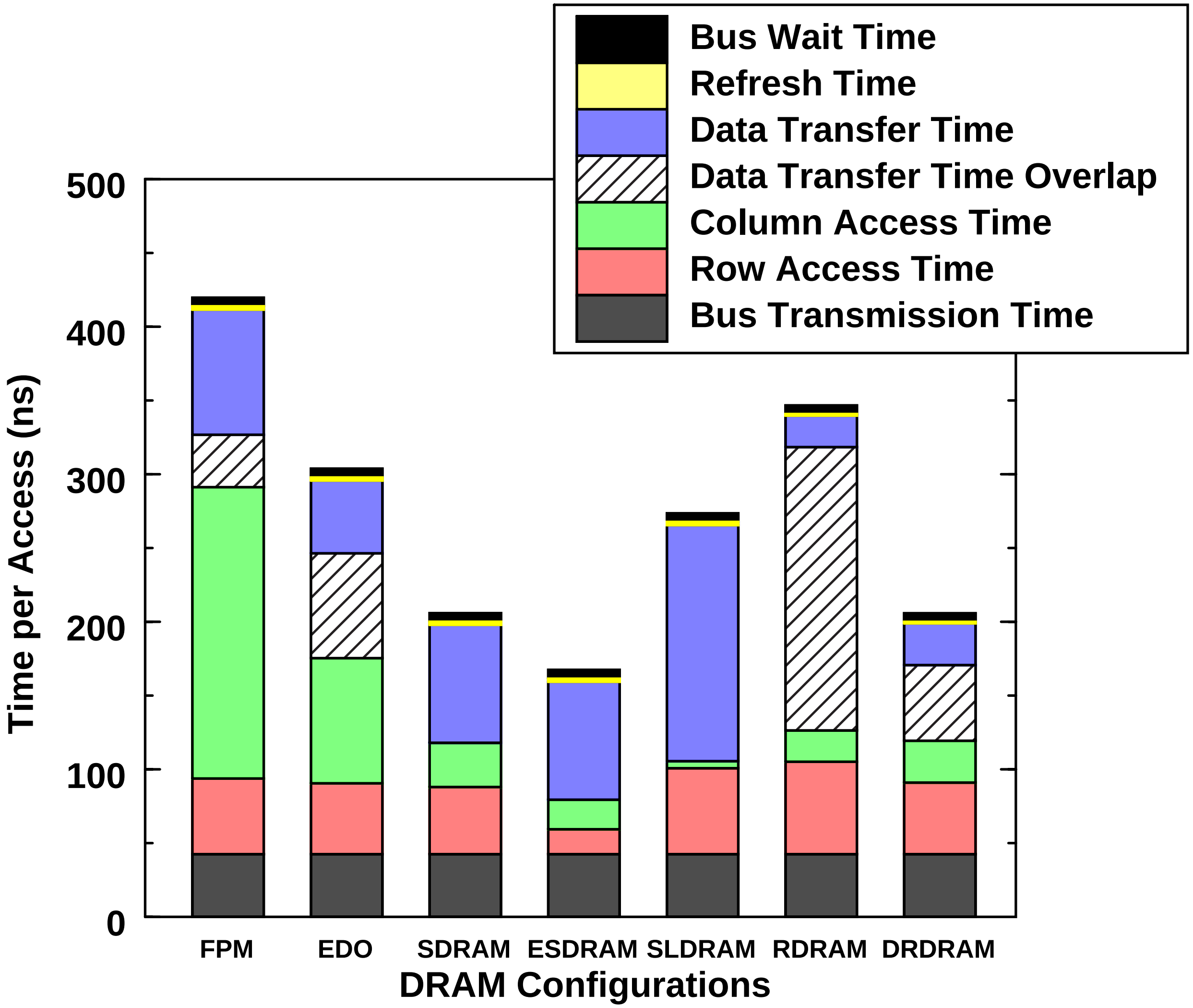
t_{BW}

t_{PROC}

Memory & CPU — PERL

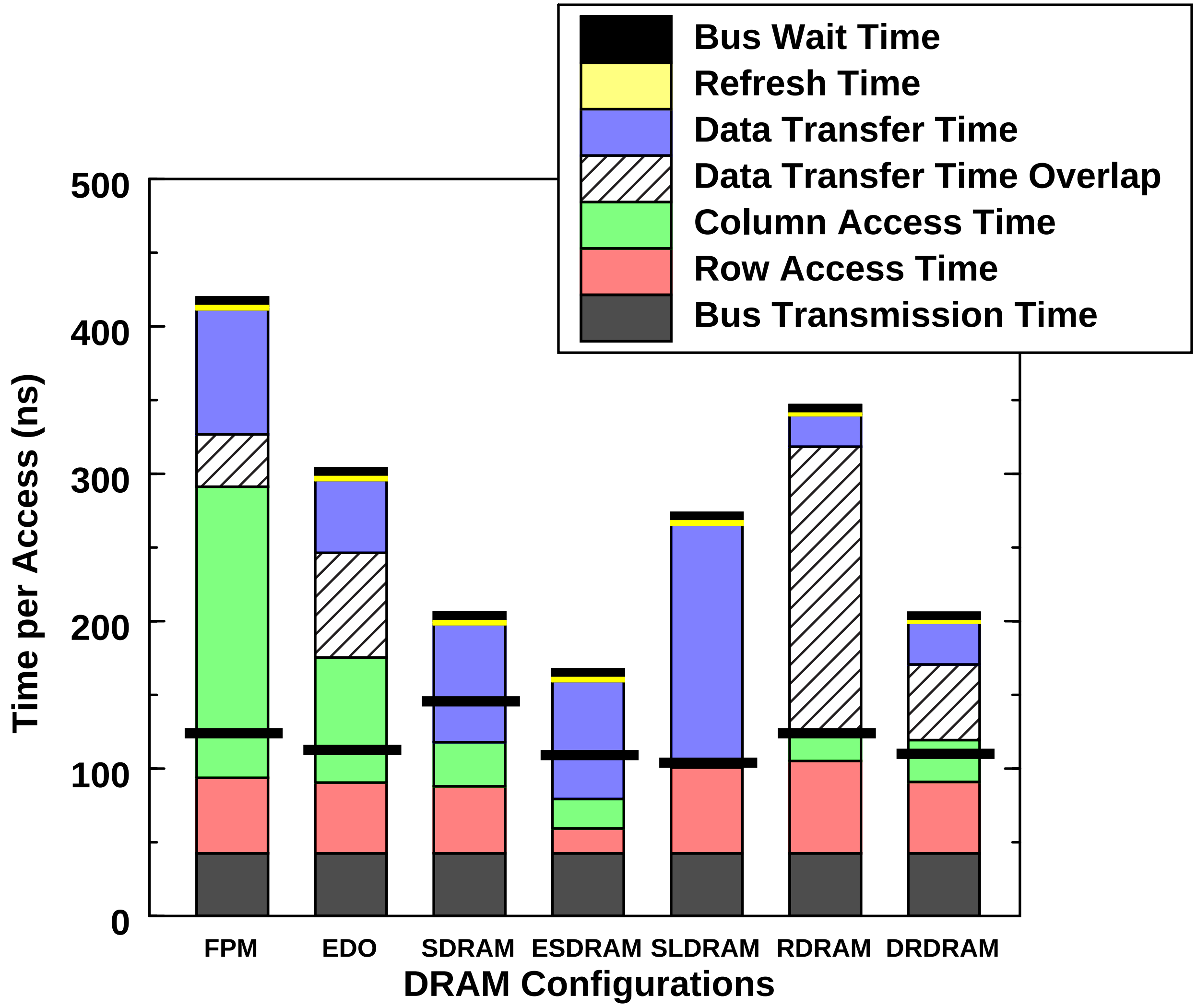


Average Latency of DRAMs



note: SLDRAM & RDRAM 2x data transfers

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**

1 DRDRAM = Multiple SDRAM

Conclusions

100MHz/128-bit Bus is Current Bottleneck

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

Current DRAMs Solving Bandwidth Problem
(but not Latency Problem)

There is Locality in DRAM Accesses
(but how important is this?)

SPECint ’95 Fits in 1MB Cache

Recent (Unfinished) Work

Investigation of Organization-Level Parameters:

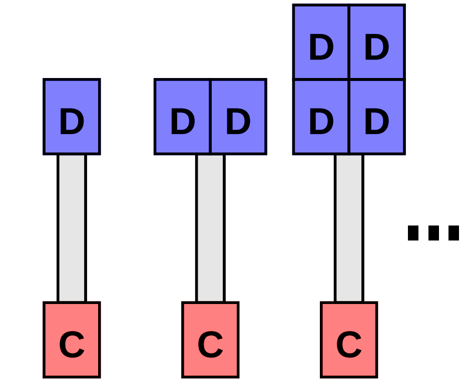
- **Channel widths & speeds, turnaround**
- **Independent vs. ganged channels**
- **Banks per channel, burst widths**

Detailed Study of DRDRAM vs. SDRAM in Highly Concurrent Environment

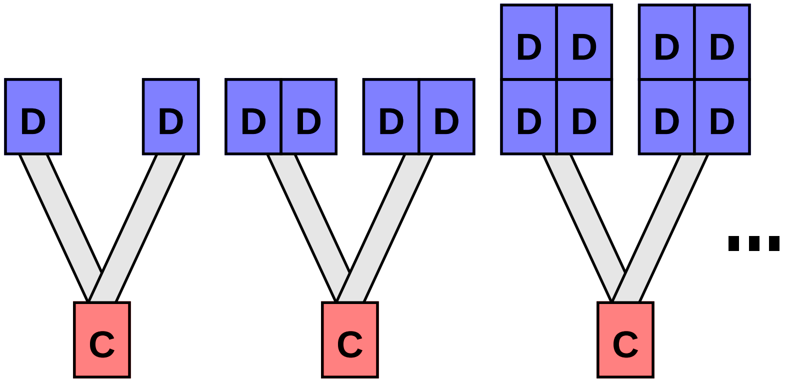
Embedded DRAM+DSP Architectures

Detailed Study of Multiprocessor Buses

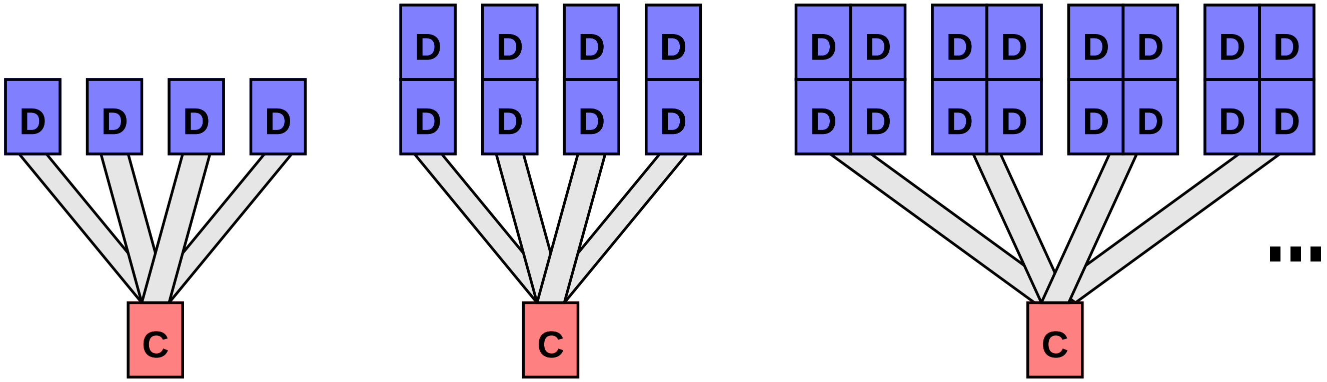
Channel/Bank Model



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



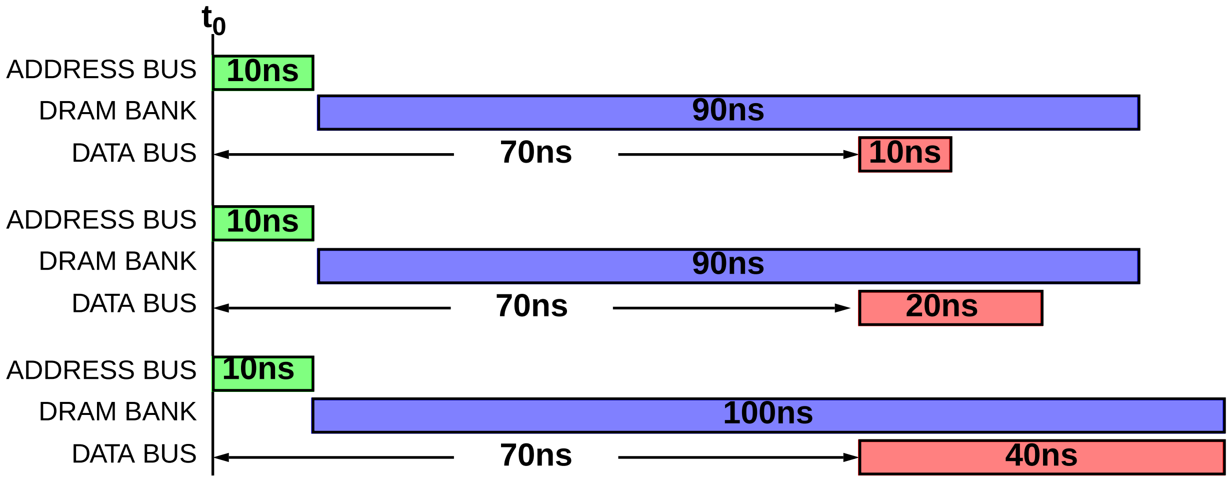
Two independent channels
Banking degrees of 1, 2, 4, ...



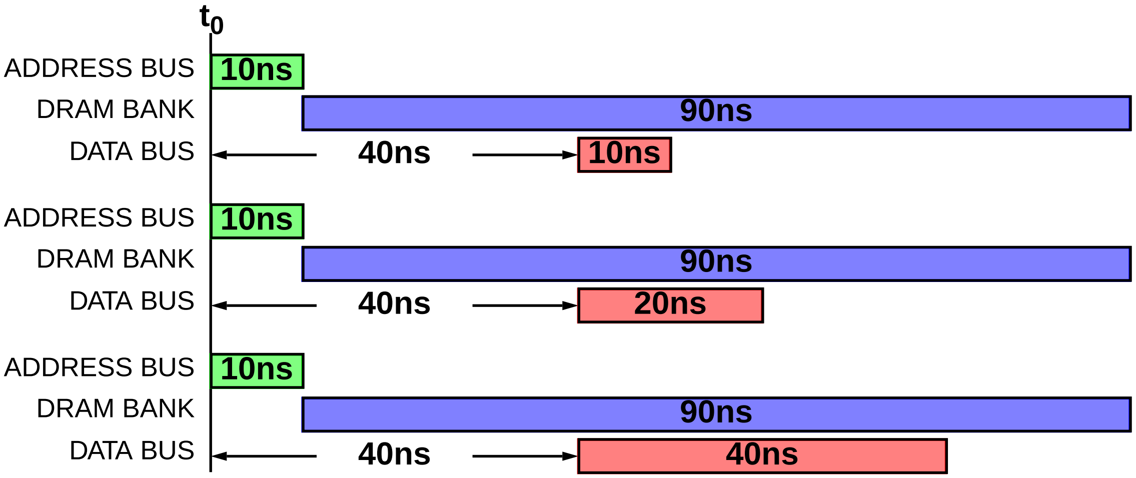
Four independent channels
Banking degrees of 1, 2, 4, ...

Read/Write Request Model

READ REQUESTS:

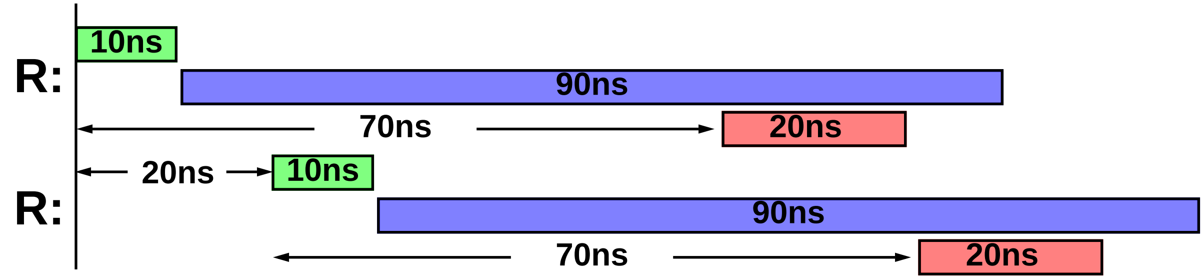


WRITE REQUESTS:

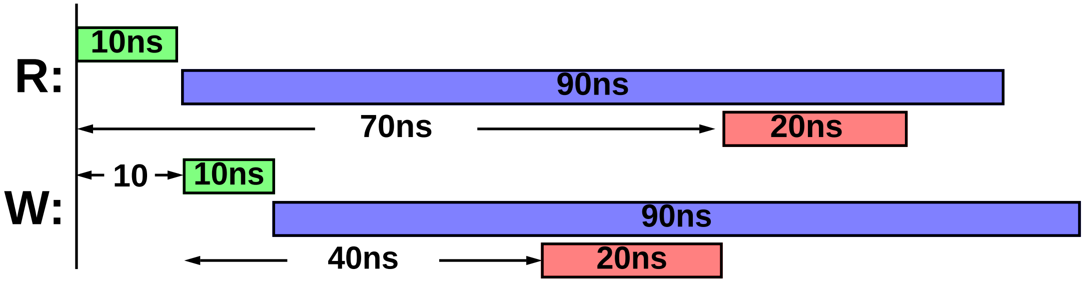


Concurrency Model

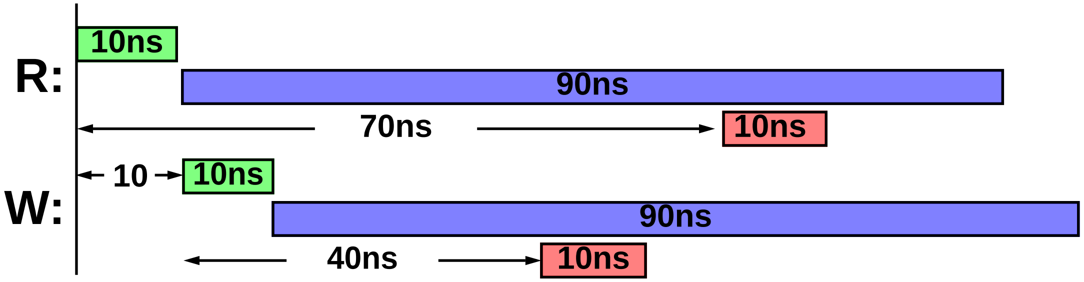
Legal if R/R to different banks:



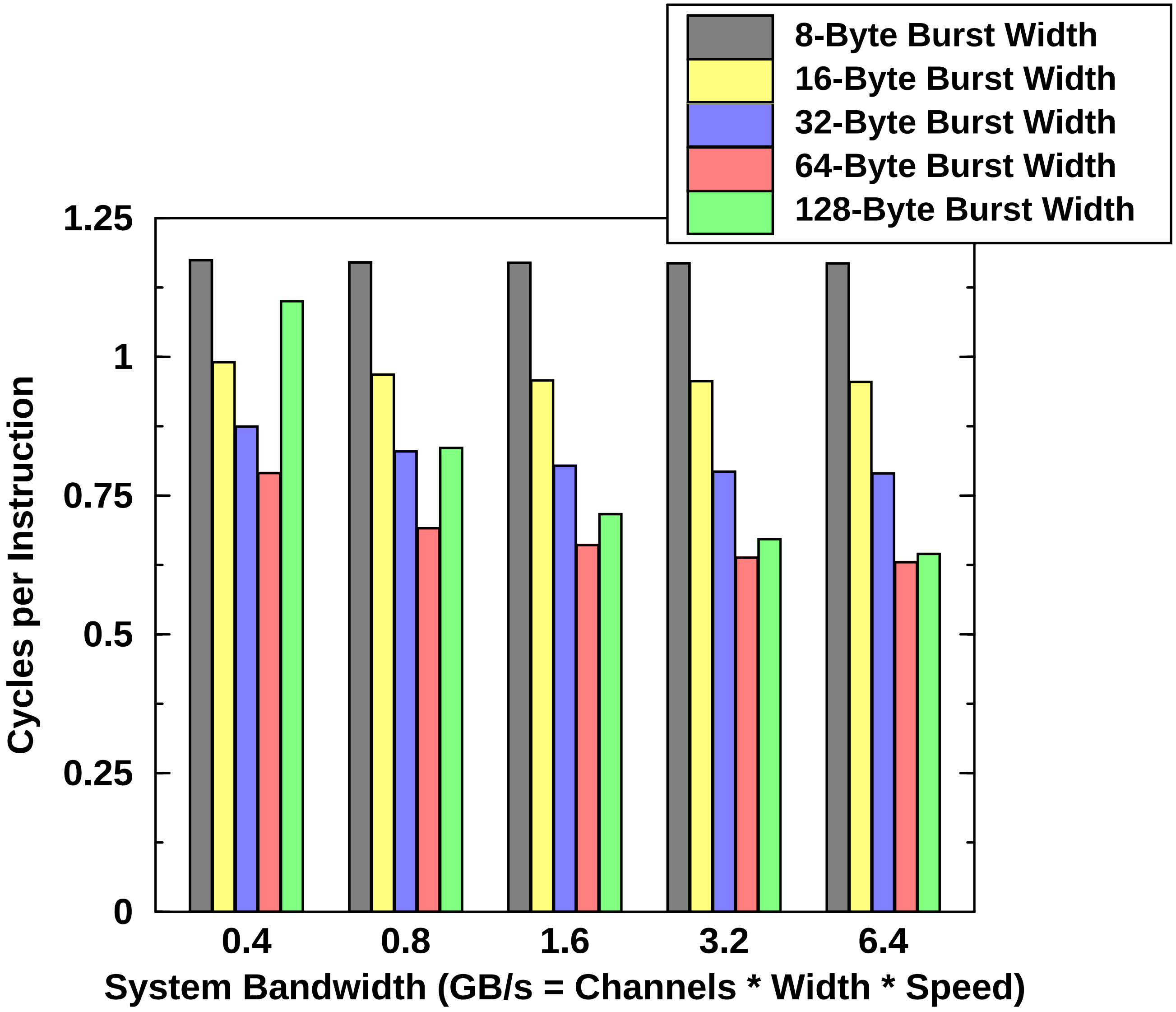
Legal if no turnaround and R/W to different banks:



Legal if turnaround ≤ 10ns and R/W to different banks:

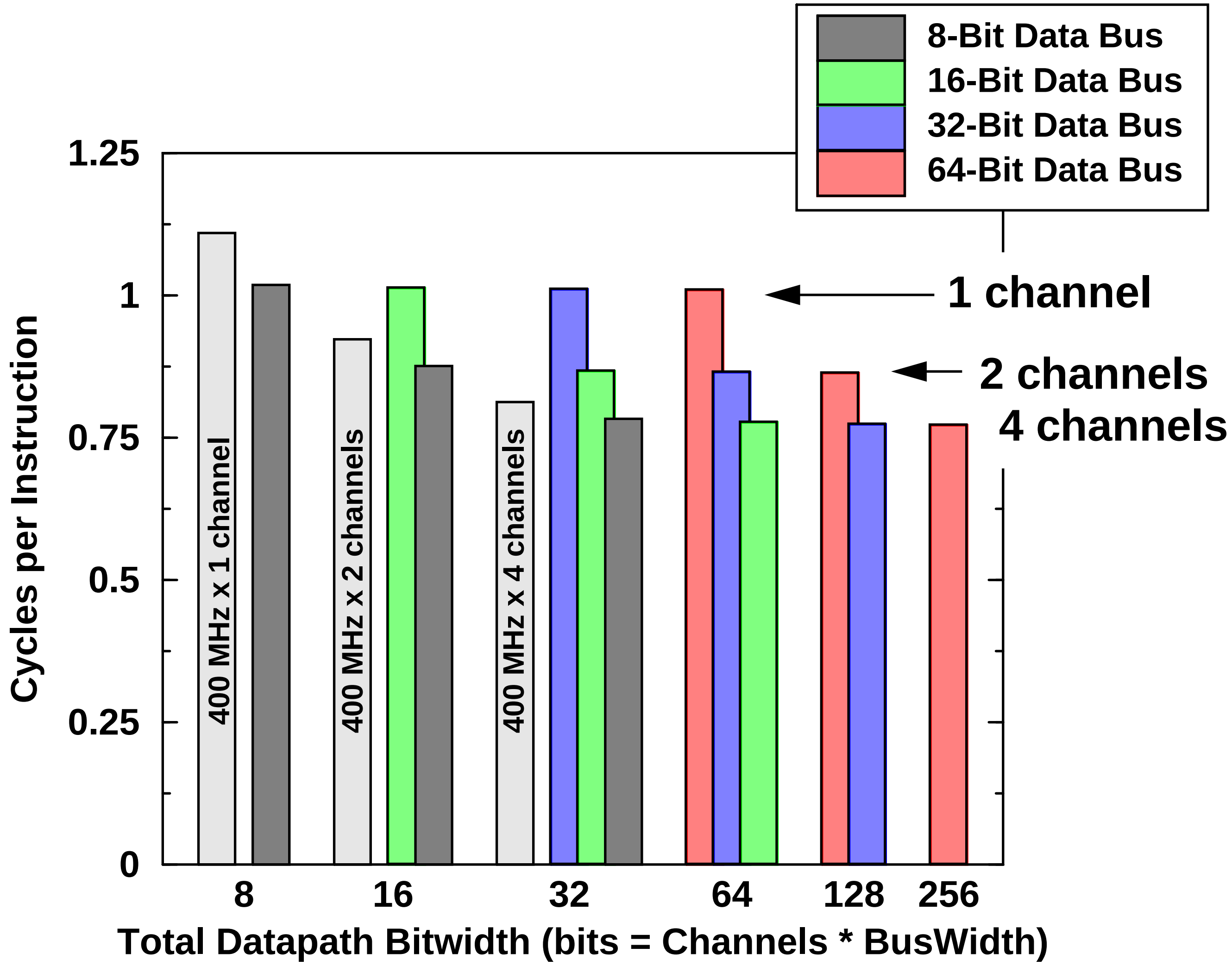


Bandwidth vs. Burst Width



PERL: 1 channel, 4 banks, 2GHz CPU

Exploiting Concurrency



PERL: 2 banks, 16-byte burst, 2GHz CPU

Conclusions

None yet ... preliminary data

CONTACT INFO:

Prof. Bruce Jacob

Electrical & Computer Engineering
University of Maryland, College Park
`http://www.ece.umd.edu/~blj/`
`blj@eng.umd.edu`