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


“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:

...  

```java
if ( INSTR.loadstore ) {
    if (L1_cache_miss( INSTR.daddr )) {
        if (L2_cache_miss( INSTR.daddr )) {

            INSTR.ready = now() + DRAM_LATENCY;

        }
    }
}
```

...
... or THIS

Fast Page Mode Read Cycle
Motivation

HERE’S WHAT YOU MISS:

DRAM LATENCY:

DATA TRANSFER

OVERLAP

COLUMN ACCESS

ROW ACCESS

BUS TRANSMISSION
Motivation

HERE’S WHAT YOU MISS:

- Data transfer?
- Overlap?
- Column access?
- Row access?
- Bus transmission?

DRAM LATENCY:
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

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
DRAM Primer

Read Timing for Fast Page Mode DRAM

- Data Transfer
- Transfer Overlap
- Column Access
- Row Access

RAS
CAS
Address
Row Address
DQ
Valid Dataout

Column Address
DRAM Primer

Read Timing for Extended Data Out DRAM
DRAM Primer

Read Timing for Synchronous DRAM

- Clock
- RAS
- CAS
- Address
- Row Address
- Column Address
- DQ
- Valid Dataout
- Data Transfer
- Transfer Overlap
- Column Access
- Row Access
DRAM Primer

Read Timing for Rambus DRAM

- **Command**: ACTV/READ
- **Address**: Bank/Row
- **DQ**: Read Strobe
- **Data Transfer**: 4 cycles
- **Transfer Overlap**
- **Column Access**
- **Row Access**

- **Valid Dataout**: Col Addr
- **Valid Dataout**: Col Addr
- **Valid Dataout**: Col Addr

---

**Bruce Jacob**
University of Maryland
DRAM Primer

Read Timing for Direct Rambus DRAM

Row [2:0]
Col [4:0]
Data [17:0]

ACT (this)

4 cycles

Col Addr
Col Addr
Col Addr
Col Addr

Valid Dataout
Valid Dataout
Valid Dataout
Valid Dataout

PRE (next)

ACT (next)

Data Transfer
Transfer Overlap
Column Access
Row Access

4 cycles
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

Stalls Due to BANDWIDTH
- $t_{\text{REAL}} - t_{\text{BW}}$

Stalls Due to LATENCY
- $t_{\text{BW}} - t_{\text{PROC}}$

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

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

$\text{t}_{\text{REAL}}$

$\text{t}_{\text{DRAM}}$

$\text{t}_{\text{BW}}$

$\text{t}_{\text{PROC}}$
Memory & CPU — PERL

- Stalls due to Memory Bandwidth
- Stalls due to Memory Latency
- Overlap between Execution & Memory
- Processor Execution

Newer DRAMs

Cycles Per Instruction (CPI)

- FPM
- EDO
- SDRAM
- ESDRAM
- DRDRAM

DRAM Configuration
Average Latency of DRAMs

- Bus Wait Time
- Refresh Time
- Data Transfer Time
- Data Transfer Time Overlap
- Column Access Time
- Row Access Time
- Bus Transmission Time

note: SLDRAM & RDRAM 2x data transfers
Average Latency of DRAMs

Bus Wait Time
Refresh Time
Data Transfer Time
Data Transfer Time Overlap
Column Access Time
Row Access Time
Bus Transmission Time

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)
HOW NOT TO CONFIGURE YOUR DRAM SYSTEM

Bruce Jacob
University of Maryland

ARCHITECTURAL COMPARISON

Normalized Execution Time (DDR2)

Benchmark

cc1, compress, go, jpeg, li, ...
DDR2 Study Results

Perl Runtime

Execution Time (Sec.)

Processor Frequency

- pc100
- ddr133
- ddr
- ddr2
- ddr2ems
- ddr2vc

Execution Time vs. Processor Frequency
**DDR2 Study Results**

**Architectural Comparison for Trace**

- Benchmark Metrics: ltp1w, ltp8w, xm_access, xm_cpu−mark, xm_gcc, xm_quake

- Metrics: Normalized Execution Time (DDR2)

- Comparison Types: ddr2, ddr2ems, ddr2vc
Concurrency Study: Timing

READ REQUEST TIMING:

$ t_0 $
Read/Write Request Shapes

**READ REQUESTS:**

- ADDRESS BUS: 1.25ns
- DRAM BANK: 90ns
- DATA BUS: 61.25ns
- DATA BUS: 10ns

**WRITE REQUESTS:**

- ADDRESS BUS: 1.25ns
- DRAM BANK: 90ns
- DATA BUS: 31.25ns
- DATA BUS: 10ns
- ADDRESS BUS: 1.25ns
- DRAM BANK: 90ns
- DATA BUS: 31.25ns
- DATA BUS: 20ns
- ADDRESS BUS: 1.25ns
- DRAM BANK: 90ns
- DATA BUS: 31.25ns
- DATA BUS: 40ns
Pipelined/Split Transactions

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

Read:

- 1.25ns
- 61.25 ns
- 20ns
- 90ns

(b) Legal if turnaround ≤ 8.75ns and R/W to different banks:

(note: write can start up to 7.5ns later if turnaround = 1.25ns)

Read:

- 1.25ns
- 61.25 ns
- 20ns
- 90ns

Write:

- 31.25 ns
- 20ns

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

Read:

- 1.25ns
- 61.25 ns
- 40ns
- 100ns

Write:

- 31.25 ns
- 40ns
Channels & Banks

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
The Bottom Line

Cycles per Instruction (CPI)

benchmarks: bzip, gcc, mcf, parser, perl, vpr, average

- Worst Organization
- Average Organization
- Best Organization
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
System Overhead

Benchmark = BZIP (SPEC 2000)
System Overhead

System overhead 10-40% over perfect memory

Cycles per Instruction (CPI)

Perfect Memory

System Bandwidth

(GB/s = 1 Channel * Width * 800Mhz)
8MSHRs, 32-Byte Burst

Benchmark = BZIP (SPEC 2000)

System Overhead graph showing system overhead for different configurations of banks per channel with a benchmark of BZIP (SPEC 2000).
It’s Not Queue Size ...

CONFIGURATIONS OF 2-BYTE CHANNELS

Black = infinite request queue,
Red = 32-entry request queue
... It’s Also Not Turnaround ...

Benchmark = BZIP (SPEC 2000), 32-byte burst, 16-bit bus
... It’s Related to **Concurrency**

![Diagram showing cycles per instruction (CPI) vs. system bandwidth (GB/s = Channels * Width * Speed). The benchmark is BZIP (SPEC 2000), 32-byte burst, 16-bit bus.](image)

Bruce Jacob
University of Maryland
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

Bruce Jacob
University of Maryland
Bandwidth vs. Burst Width

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

Benchmark = GCC (SPEC 2000), 2 banks/channel
Bandwidth vs. Burst Width

GOLDILOCKS PRINCIPLE

Wide channels (32/64-bit) want large bursts

Cycles per Instruction (CPI)

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

Benchmark = GCC (SPEC 2000), 2 banks/channel
Bandwidth vs. Burst Width

GOLDILOCKS PRINCIPLE

Narrow channels (8-bit) want small bursts

Cycles per Instruction (CPI)

System Bandwidth

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

Benchmark = GCC (SPEC 2000), 2 banks/channel
Bandwidth vs. Burst Width

GOLDILOCKS PRINCIPLE

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

Benchmark = GCC (SPEC 2000), 2 banks/channel
Focus on 3.2 GB/s — MCF

- 1 Bank per Channel
- 2 Banks per Channel
- 4 Banks per Channel
- 8 Banks per Channel

Benchmark = MCF (SPEC 2000)
Focus on 3.2 GB/s — MCF

Benchmark = MCF (SPEC 2000)

#Banks not particularly important with large bursts...
Focus on 3.2 GB/s — MCF

Benchmark = MCF (SPEC 2000)
Focus on 3.2 GB/s — MCF

Multiple channels not always a good idea

Benchmark = MCF (SPEC 2000)
Focus on 3.2 GB/s — MCF

<table>
<thead>
<tr>
<th>Burst Size</th>
<th>Channels</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>32-Byte</td>
<td>1 chan x 4 bytes</td>
<td>1-byte channels</td>
</tr>
<tr>
<td></td>
<td>2 chan x 2 bytes</td>
<td>2-byte channels</td>
</tr>
<tr>
<td></td>
<td>4 chan x 1 byte</td>
<td>4-byte channels</td>
</tr>
<tr>
<td>64-Byte</td>
<td>1 chan x 4 bytes</td>
<td>1-byte channels</td>
</tr>
<tr>
<td></td>
<td>2 chan x 2 bytes</td>
<td>2-byte channels</td>
</tr>
<tr>
<td></td>
<td>4 chan x 1 byte</td>
<td>4-byte channels</td>
</tr>
<tr>
<td>128-Byte</td>
<td>1 chan x 4 bytes</td>
<td>1-byte channels</td>
</tr>
<tr>
<td></td>
<td>2 chan x 2 bytes</td>
<td>2-byte channels</td>
</tr>
<tr>
<td></td>
<td>4chan x 1 byte</td>
<td>4-byte channels</td>
</tr>
</tbody>
</table>

Benchmark = MCF (SPEC 2000)
Focus on 3.2 GB/s — BZIP

Benchmark = BZIP (SPEC 2000)
Focus on 3.2 GB/s — BZIP

Benchmark = BZIP (SPEC 2000)
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