Modern DRAM Memory Systems

Brian T. Davis

MTU Interview Seminar

Advanced Computer Architecture Laboratory
University of Michigan

April 24, 2000
● Introduction
  ○ Memory system
  ○ Research objective

● DRAM Primer
  ○ Array
  ○ Access sequence
  ○ SDRAM
  ○ Motivation for further innovation

● Modern DRAM Architectures
  ○ DRDRAM
  ○ DDR2
  ○ Cache enhanced DDR2 low-latency variants

● Performance and Controller Policy Research
  ○ Simulation methodologies
  ○ Results

● Conclusions

● Future Work
Architecture Overview
- This is the architecture of most desktop systems
- Cache configurations may vary
- DRAM Controller is typically an element of the chipset
- Speed of all Busses can vary depending upon the system

DRAM Latency Problem
Research Objective

● Determine highest performance memory controller policy for each DRAM architecture

● Compare performance of various DRAM architectures for different classifications of applications, while each architecture is operating under best controller policy
One transistor & capacitor per bit in the DRAM (256 or 512MBit currently)

Three events in hardware access sequence

- Precharge
- Energize word line--based upon de-muxed row data
- Select bits from the row in sense-amps

Refresh is mandatory

Page and row are synonymous terminology
Arrays per Device

- Multiple arrays per device & aspect ratio
  - Larger arrays; larger bit lines; higher capacitance; higher latency
  - Multiple smaller arrays; lower latency; more concurrency (if interface allows)
  - Tradeoff--fewer & larger = cheaper--more & smaller = higher performance

- Controller policies
  - Close-Page-AutoPrecharge (CPA)
  - Open-Page (OP)
Fast-Page-Mode (FPM) DRAM Interface

- All signals required by DRAM array provided by DRAM controller
- Three events in FPM interface access sequence
  - Row Address Strobe - RAS
  - Column Address Strobe - CAS
  - Data response
- Dedicated interface - only a single transaction at any time
- Address bus multiplexed between row & column
SDRAM Interface

- All I/O synchronous rather than async--buffered on the device
- Split-transaction interface
- Allows concurrency in a pipelined-similar fashion - to unique banks
- Requires latches for address & data - low device overhead
- Double Data Rate (DDR) increases only data transition frequency
Devices per DIMM affects effective page size thus potentially performance
- Each device only covers a "slice" of the data bus
- DIMMs can be single or double sided - single sided shown
- Data I/O per device is a bond-out issue
  - Has been increasing as devices get larger
Motivation for a New DRAM Architecture

- SDRAM limits performance of high-performance processors
  - TPC-C 4-wide issue machines achieve CPI of 4.2-4.5 (DEC)
  - STREAM 8-wide machine--1Ghz: CPI of 3.6-9.7--5G: CPI of 7.7-42.0
  - PERL 8-wide machine--1Ghz: CPI of 0.8-1.1--5Ghz: CPI of 1.0-4.7

- DRAM array has essentially remained static for 25 years
  - Device size (x4) per 3 years - Moore’s law
  - Processors performance (not speed) 60% annually
  - Latency decreases at 7% annually

- Bandwidth vs. Latency
  - Potential bandwidth = (data bus width) * (operating frequency)
  - 64-bit desktop bus 100-133 MHz (0.8 - 1.064 GB/s)
  - 256-bit server (parity) bus 83-100 Mhz (2.666-3.2 GB/s)

- Workstation manufacturers migrating to enhanced DRAM
Modern DRAM Architectures

- DRAM architecture’s examined
  - PC100 - baseline SDRAM
  - DDR133(PC2100) - SDRAM 9 months out
  - Rambus -> Concurrent Rambus -> Direct Rambus
  - DDR2
  - Cache Enhanced Architecture - possible to any interface - here to DDR2

- Not all novel DRAM will be discussed here
  - SyncLink - death by standards organization
  - Cached DRAM - two-port notebook single-solution
  - MultiBanked DRAM - low-latency core w/ many small banks

- Common elements
  - Interface should enable parallelism between accesses to unique banks
  - Exploit the extra bits retrieved, but not requested

- Focus on DDR2 low-latency variants
  - JEDEC 42.3 Future DRAM Task Group
  - Low-Latency DRAM Working Group
DRDRAM RIMM/System Architecture

- Smaller arrays: 32 per 128Mbit device (4 Mbit Arrays; 1KByte page)
- Devices in series on RIMM rather than parallel
- Many more banks than in an equivalent size SDRAM memory system
- Sense-amps are shared between neighboring banks
- Clock flows both directions along channel
### Direct Rambus (DRDRAM) Channel

- **Narrow bus architecture**
- **All activity occurs in OCTCycles (4 clock cycles; 8 signal transitions)**
- **Three bus components**
  - Row (3 bits); Col (5 bits); Data (16 bits)
- **Allows 3 transactions to use the bus concurrently**
- **All signals are Double Data Rate (DDR)**
Four arrays per 512 Mbit device
Simulations assume 4 (x16) devices per DIMM
Few, large arrays--64MByte effective banks--8 KByte effective pages
DDR2 Interface

- Changes from current SDRAM interface
  - Additive Latency (AL = 2; CL = 3 in this figure)
  - Fixed burst size of 4
  - Reduce power considerations
- Leverages existing knowledge
EMS Cache-Enhanced Architecture

- Full SRAM cache array for each row
- Precharge latency can always be hidden
- Adds the capacity for No-Write-Transfer
- Controller requires no additional storage--only control for NW-Xfer
Virtual Channel Architecture

- Channels are SRAM cache on DRAM die - 16 channels = 16 line cache
- Read and write can only occur through channel
- Controller can manage channels in many ways
  - FIFO
  - Bus-master based
- Controller complexity & storage increase dramatically
- Designed to reduce conflict misses
<table>
<thead>
<tr>
<th></th>
<th>PC133</th>
<th>DDR2</th>
<th>DDR2_VC</th>
<th>DDR2_EMS</th>
<th>DRDRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Potential Bandwidth</td>
<td>1.064 GB/s</td>
<td></td>
<td>3.2 GB/s</td>
<td></td>
<td>1.6 GB/s</td>
</tr>
<tr>
<td>Interface</td>
<td>• Bus</td>
<td>• Bus</td>
<td></td>
<td>• Channel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 64 Data bits</td>
<td>• 64 Data bits</td>
<td></td>
<td>• 16 Data Bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 168 pads on DIMM</td>
<td>• 184 pads on DIMM</td>
<td></td>
<td>• 184 pads on RIMM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 133 Mhz</td>
<td>• 200 Mhz</td>
<td></td>
<td>• 400 Mhz</td>
<td></td>
</tr>
<tr>
<td>Latency to first 64 bits (Min. : Max)</td>
<td>(3 : 9) cycles</td>
<td>(3.5 : 9.5) cycles</td>
<td>(2.5 : 18.5) cycles</td>
<td>(3.5 : 9.5) cycles</td>
<td>(14 : 32) cycles</td>
</tr>
<tr>
<td></td>
<td>(22.5 : 66.7) nS</td>
<td>(17.5 : 47.5) nS</td>
<td>(12.5 : 92.5) nS</td>
<td>(17.5 : 47.5) nS</td>
<td>(35 : 80) nS</td>
</tr>
<tr>
<td>Latency Advantage</td>
<td></td>
<td>• 16 Line Cache / Dev; 1/4 row line size</td>
<td>• Cache Line per bank; line size is row size</td>
<td>• Many smaller banks</td>
<td>• Narrow Bus</td>
</tr>
<tr>
<td>Advantage</td>
<td>• Cost</td>
<td>• Cost</td>
<td>• Less Misses in “Hot Bank”</td>
<td>• Precharge Always Hidden</td>
<td>• Narrow Bus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Full Array BW Utilized</td>
<td>• Smaller Incremental granularity</td>
</tr>
<tr>
<td>Disadvantage</td>
<td></td>
<td></td>
<td>• Area (3-6%)</td>
<td>• Area (5-8%)</td>
<td>• Area (10%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Controller Complexity</td>
<td>• More conflict misses</td>
<td>• Sense Amps shared between adjacent banks</td>
</tr>
</tbody>
</table>
Comparison of Controller Policies

- **Close-Page Auto Precharge (CPA)**
  - After each access, data in sense-amps is discarded
  - ADV: Subsequent accesses in unique row/page: no precharge latency
  - DIS: Subsequent accesses in same row/page: must repeat access

- **Open-Page (OP)**
  - After each access, data in sense-amps is maintained
  - ADV: subsequent accesses in same row/page: page-mode access
  - DIS: Adjacent accesses in unique row/page: incurs precharge latency

- **EMS considerations**
  - No-Write Transfer mode - how to identify write only streams or rows

- **Virtual Channel (VC) considerations**
  - How many channels can the controller manage?
  - Dirty virtual channel writeback
Execution Driven Simulation

- SimpleScalar - standard processor simulation tool

**Advantages**
- Feedback from DRAM latency
- Parameter’s of system are easy modify with full reliability
- Confidence in results can be very high

**Disadvantages**
- SLOW to execute
- Limited to architectures which can be simulated by SimpleScalar
Trace Driven Simulation

 Advantages
● FAST to simulate
● Allows traces from SMP's or more complex architectures
● Appropriate for model verification, hit-rate

 Disadvantages
● No-feedback from access to subsequent accesses
● W/O timestamps is essentially a limit-study framework
● Not appropriate for time based results
● Simulation parameters limited to those of the gathered system
Results

● Execution driven based upon:
  ○ SimpleScalar
    ● Version 2.0 MSHR - Written by Todd Austin - Modified by Doug Burger - Customized for these simulations
    ● 8 Way Super Scalar / 2 Memory Ports
    ● 32K I/D split L1 caches
    ● 256K Unified L2
    ● 16 MSHRs provide concurrent memory access support

● Trace driven based upon:
  ○ IBM OLTP (On-Line Transaction Processing) traces
    ● SMP 1-way or 8-way processor - elements are cache snoop data
  ○ Transmeta Crusoe processor running Windows applications
    ● Includes processor, AGP graphics & I/O as access sources

● DRAM & controller models
  ○ SDRAM model (PC100 - DDR133)
  ○ DRDRAM model
  ○ DDR2 model (std, vc & ems)
### Bandwidth Benchmarks Runtime

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>random_walk</td>
<td>0.5</td>
</tr>
<tr>
<td>stream</td>
<td>0.75</td>
</tr>
<tr>
<td>stream_no_unroll</td>
<td>1.0</td>
</tr>
</tbody>
</table>

- pc100
- ddr133
- ddr2
- ddr2ems
- ddr2vc
Data Bus Utilization

Fraction

Benchmark

- pc100
- ddr133
- ddr2
- ddr2ems
- ddr2vc

cc1  compress  go  jpeg  li  linear_walk  mpeg2dec  mpeg2enc  pegwit  perl  random_walk  stream  stream_no_un
Adjacent Accesses to Same Bank

Trace

Fraction

oltp1w  oltp8w  xm_access  xm_cpumark  xm_gcc  xm_quake

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Legend:
- ddr2_cpa
- ddr2_cpa_remap
- ddr2_op
- ddr2_op_remap
- ddr2ems
- ddr2ems_remap
- ddr2vc
- ddr2vc_remap
Hit Rates

Fraction

Trace

oltp1w  oltp8w  xm_access  xm_cpumark  xm_gcc  xm_quake

ddr2_cpa  ddr2_op  ddr2ems_cpa  ddr2vc_cpa
Average Latency

NanoSeconds

Benchmark

cc1   compr  go    jpeg  li   linear_w   mp2d  mp2e  pegwit  perl  random_w   stream  stream_unroll

ddr2_cpa   ddr2_cpa_inv
ddr2_op
ddr2ms_cpa   ddr2ms_cpa_inv
ddr2ms_op
ddr2vc_cpa
Conclusions

● More bandwidth can be had, at a cost
● The target for architectural improvements must be latency
● Controller can significantly affect average latency
● DDR2 is evolutionary, but provides the required performance
● Cache Enhanced DRAM can improve performance, but the price for improvement is dependant upon market penetration
● Packetized interfaces incurs increased latency
Future Work

- VC controller performance
  - Cache line allocation policy(s) for Channels
  - When to write-back dirty channels - avoid maximal penalty
  - Price/Performance in Controller

- EMS controller performance
  - When to use no-write-transfer

- Controller onto processor die

- Embedded DRAM architectures

- SMP primary memory partitioning
Conventional DRAM

- Basic DRAM core (memory array) used in all DRAM memories
- Delay is propagation through all circuits, no pipelining
- Limit on memory array size due to bit line capacitance
- Remainder of row, accessed, but not used, is discarded
Conventional DRAM Upgrades

- **Fast Page Mode (FPM) DRAM**
  - Eliminates the RAS transition requirement between each access
  - Utilizes the sense-amp contents as cache

- **Extended Data Out (EDO) DRAM**
  - Latch added between the sense-amps and the output drivers
  - Allows parallel operation of two DRAM components
    - Output drivers function while next access is being done
    - Memory array (precharge or access) is somewhat overlapped

- **Burst EDO DRAM**
  - Burst capability for accessing large contiguous segments of a row
  - Toggling of the CAS line sequences to the next datum in the burst
Conventional (FPM) DRAM Interface

- Dedicated interface - only a single transaction at any time
- Address bus multiplexed between Row & Column
- All signals to req’d by DRAM array provided by DRAM controller
Synchronous DRAM (SDRAM)

- Make all I/O synchronous rather than async
- 66MHz SDRAM -> PC100 -> DDR133 (PC2100)
- Overhead is very low - latches for address & data
Interleaved Memory

- Relatively uncommon
- Used to get concurrency from asynchronous DRAM
Direct Rambus (DRDRAM) Device Architecture